Boosting Phonocardiogram Classification Performance with Function Generated Data

Naoki Nonaka[a,b] NAOKI.NONAKA@RIKEN.JP
Hiroshi Seki[c] HSEKI@AMI.INC
Tomohiro Komatsu[c] KOMATSU.TOMOHIRO@AMI.INC
Jun Seita[a,b] JUN.SEITA@RIKEN.JP
[a] *Center for Interdisciplinary Theoretical and Mathematical Sciences, RIKEN*
[b] *Center for Integrative Medical Sciences, RIKEN*
[c] *Research and Development Department, AMI Inc.*

Abstract

Deep neural networks require large datasets, yet medical phonocardiogram (PCG) data are scarce due to privacy and disease rarity. To address this challenge in PCG analysis, we present a function-generated PCG pipeline that synthesizes S1/S2 heart sounds with modulated noise to emulate aortic stenosis (AS), aortic regurgitation (AR), and mitral regurgitation (MR). Across eight architectures, we compare real-only training, synthetic-only, and synthetic pretraining followed by real fine-tuning (Syn→Real). Syn→Real consistently improves AUROC with average gains of +15.3% (AS), +17.0% (AR), +17.1% (MR) on BMD-HS, and +7.1%, +8.8%, +6.1% on a private cohort ($8,564$ recordings). Furthermore, we show Syn→Real is competitive with pretraining on out-of-domain real data, and combining it with multi-stage real fine-tuning yields the best overall performance, highlighting the complementary value of synthetic and real PCGs. While synthetic-only training generalizes poorly, pretraining on function-generated PCGs consistently improves PCG classification over training from scratch, offering a practical path to mitigate data-collection burdens and potentially reduce privacy and ethical exposure.

Keywords: Synthetic medical data, Phonocardiogram (PCG) analysis, Deep neural networks (DNNs), Cardiac disease classification

Data and Code Availability In this study, we use the BMD-HS dataset (Ali et al., 2024)[1] and a confidential private dataset. Code is available through our GitHub repository[2].

Institutional Review Board (IRB) This study involved human subjects, and we received approval from RIKEN Institutional Review Board (ID: Y2023-048).

1. Introduction

Training robust Deep Neural Network (DNN) models in the medical domain often requires large-scale datasets, a critical barrier when addressing rare or complex pathologies. This limitation is particularly evident in phonocardiogram (PCG) analysis—the computational analysis of heart sound recordings for cardiac abnormality detection. The primary challenge in PCG-based diagnosis lies in the difficulty of acquiring sufficient pathological heart sound data, such as those associated with aortic regurgitation (AR), aortic stenosis (AS), and mitral regurgitation (MR). This data scarcity problem is further compounded by several factors: varying recording environments that introduce inconsistencies in data quality, privacy regulations governing patient data that restrict data sharing and collection, and challenges in curating datasets that accurately represent diverse clinical scenarios. These factors collectively highlight the need for developing methodologies that reduce reliance on large real-world datasets while maintaining diagnostic reliability.

Synthetic data has emerged as a powerful tool to address data scarcity across domains. In computer vision, frameworks like SYNAuG (Ye-Bin et al., 2023) mitigate class imbalance by augmenting datasets with synthetic samples before fine-tuning on real data. Similarly, domain-randomized synthetic scenes (Tremblay et al., 2018) and biomechanical human

[1] https://github.com/sani002/BMD-HS-Dataset
[2] https://github.com/seitalab/SynPCG

models (Varol et al., 2017) have proven effective for tasks requiring generalization. These successes hinge on synthesizing data that captures essential domain characteristics—a principle equally critical for medical applications.

Recent work demonstrates synthetic data's potential in biomedical signal analysis. For electrocardiograms (ECGs), pretraining on domain-informed synthetic signals followed by real-data fine-tuning has achieved 33% accuracy gains in classifying rare arrhythmias (Nonaka and Seita, 2024a). However, parallel advances in PCG analysis remain lacking. While early PCG synthesis used simplified mathematical models (Almasi et al., 2011) and recent GAN-based approaches (Narváez and Percybrooks, 2020) generate normal heart sounds, they fail to capture the acoustic complexity of pathological murmurs. This gap persists despite the clinical urgency to detect valvular disorders—prevalent conditions that remain underdiagnosed due to data scarcity.

Our work bridges this gap by establishing a synthetic-to-real paradigm for PCG-based detection of aortic and mitral valve pathologies. Drawing on medical expertise, we combine several simple functions to emulate the charateristics of a normal state and AS, AR and MR state PCGs. We systematically evaluate eight DNN architectures, demonstrating that synthetic pretraining followed by real-data fine-tuning improves classification accuracy by up to 17.1% compared to real-data-only training. By extending synthetic data principles from ECG (Nonaka and Seita, 2024a) and computer vision (Ye-Bin et al., 2023) to cardiac acoustics, we show that pretraining using synthesized data improves PCG classification where large real PCG corpora are scarce, reducing dependence on costly data collection and potentially mitigating privacy and ethical risks, while respecting the anatomical and acoustic specificity of auscultation.

2. Related Work

Recent advances in synthetic data have demonstrated its value in training DNNs across domains. In computer vision, fractal-generated images (Kataoka et al. (2022, 2020)) and domain-randomized scenes (Tremblay et al. (2018)) reduce reliance on real-world datasets. Synthetic human models like SMPL (Varol et al. (2017)) enable scalable pose estimation, while frameworks like SYNAuG (Ye-Bin et al. (2023)) address class imbalance by augmenting imbalanced datasets with synthetic data followed by real-data fine-tuning. These efforts highlight synthetic data's dual role in mitigating scarcity and imbalance.

In medical applications, synthetic data has advanced imaging and biosignal analysis. GANs augment chest X-rays (Salehinejad et al. (2018)), synthesize skin lesions (Ghorbani et al. (2020)), and generate tumors for segmentation (Hu et al. (2023)). For electrocardiograms (ECGs), recent work demonstrates synthetic data's utility: Nonaka and Seita (2024a) pretrain models on synthetic abnormal ECG signals (e.g., AF, WPW), achieving 33% classification gains after fine-tuning on limited real data. Nonaka and Seita (2024b) further show that self-supervised pretraining with domain-informed synthetic ECGs (e.g., Gaussian curve-based waveforms) matches real-data pretraining for Transformers. Synthetic electronic health records (Biswal et al. (2021); Naseer et al. (2023)) also address privacy concerns. However, phonocardiogram (PCG) synthesis—critical for cardiac diagnostics—remains underexplored, particularly for abnormal murmurs.

Early PCG synthesis relied on mathematical models like ODE-based systems (Almasi et al. (2011); Kapen et al. (2020)), which capture normal signals but lack pathological complexity. Data-driven approaches now dominate: GANs with wavelet transforms (Narváez and Percybrooks (2020)) and adversarial biosignal models (Dissanayake et al. (2022)) synthesize normal PCGs but neglect abnormal cases. Scaling efforts like WaveNet-based PCG synthesis (Jamshidi et al. (2024)) and denoising frameworks (González-Rodríguez et al. (2023)) focus on fidelity or preprocessing, not pathological classification. In parallel, the literature includes parametric murmur synthesis (Debiais et al. (1997)) and hemoacoustic simulations of valvular lesions (e.g., aortic stenosis; Zhu et al. (2019)), which can achieve high acoustic fidelity. Nevertheless, open and reproducible pipelines tailored for large-scale, disease-labeled PCG training remain limited, and comparable evaluations are hindered by heterogeneous implementations and labeling conventions. Thus, scalable, pathology-aware synthesis strategies for PCG are still underdeveloped.

Our work addresses this gap by evaluating medically informed synthetic PCG signals for DNN-based classification of aortic and mitral valve pathologies. Building on successes in ECG synthesis (Nonaka and Seita (2024a,b)) and conceptually aligned with imbalance-correction frameworks such as SYNAuG (Ye-Bin et al. (2023)), we show that synthetic pretraining followed by real-data fine-tuning improves accuracy in data-scarce settings across multiple model families. This establishes synthetic-to-real pretraining as an effective approach for abnormal PCG pathologies, while remaining compatible with alternative synthesizers as they become available.

3. Method: Function Generated PCG

The PCG exhibits foundational attributes that are widely recognized, establishing it as a robust basis for synthesizing cardiac acoustic data in biomedical research. These attributes are characterized by distinct acoustic waveforms generated by mechanical heart activity, marked by primary sound components labeled as S1, S2, S3, and S4. The most prominent features include the “lub-dub” sounds (S1 and S2), which correspond to the closure of the atrioventricular and semilunar valves, respectively. This structured acoustic profile, often accompanied by murmurs or additional sounds in pathological conditions, enables the synthesis of PCG signals with high fidelity. The standard heart rate range aligns with that of ECG, typically 50 to 80 beats per minute.

Signal Model for S1/S2. Motivated by the physiology of valve closure followed by damped tissue–blood vibrations, we model each heart sound (S1, S2) as a short oscillatory burst constructed by multiplying a sinusoidal carrier with an *asymmetric (split) Gaussian* envelope. For $S \in \{\mathrm{S1}, \mathrm{S2}\}$ and $\tau = t - \mu_S$, we synthesize

$$x_S(t) \;=\; A_S \, g_{\mathrm{asym}}\big(\tau;\, \sigma_{\mathrm{rise}},\, \sigma_{\mathrm{decay}}\big)\, \sin\!\big(2\pi f_S t + \phi_S\big),$$

with a peak–normalized envelope

$$g_{\mathrm{asym}}(\tau; \sigma_{\mathrm{rise}}, \sigma_{\mathrm{decay}}) = \begin{cases} \exp\!\big(-\frac{1}{2}(\tau/\sigma_{\mathrm{rise}})^2\big)\,, \tau < 0, \\ \exp\!\big(-\frac{1}{2}(\tau/\sigma_{\mathrm{decay}})^2\big)\,, \tau \geq 0. \end{cases}$$

Here A_S is an amplitude scale, μ_S the onset/peak time, f_S the carrier frequency and ϕ_S the phase. The asymmetry ($\sigma_{\mathrm{rise}} < \sigma_{\mathrm{decay}}$) captures the faster attack and slower decay observed clinically; the split-Gaussian affords smoothness and closed-form spectral behavior, while the sinusoidal carrier concentrates energy in physiologically plausible bands.

3.1. Synthesis of Normal State PCG

PCGs were synthesized at a sampling rate of 8 kHz over 10-second intervals to match the technical specifications of the BMD-HS dataset (Ali et al., 2024). Synthesis proceeded cycle by cycle. Within each cardiac cycle, one S1 burst and one S2 burst were generated independently using the signal model above (component-specific $\mu_S, f_S, \sigma_{\mathrm{rise}}, \sigma_{\mathrm{decay}}, A_S, \phi_S$) and then superimposed in the time domain. An overview of the synthesis pipeline is provided in Algorithm 1. We iteratively generated cardiac cycles until the total duration exceeded 80,000 samples (10 seconds at 8 kHz). Excess samples were truncated to maintain consistent length, after which ambient noise (e.g., respiratory artifacts, skin friction) were added to emulate real-world recording conditions. Full algorithmic details are provided in Appendix A, and a representative synthesized PCG, including pathological murmurs and physiological variability, is shown in Appendix B.

Algorithm 1: PCG Synthesize by Superimposing Asymmetric Peaks and Noise

```
Input: target_length
Output: pcg: Synthesized PCG signal
$pcg \leftarrow []$;
$pcg\_params \leftarrow GenerateParams()$;
while $length(pcg) < target\_length$ do
    $beat \leftarrow GenerateBeat(pcg\_params)$;
    $pcg \leftarrow pcg + beat$;
    $pcg\_params \leftarrow$
      $PerturbParams(pcg\_params)$;
end
if $length(pcg) > target\_length$ then
    $pcg \leftarrow Trim(pcg, target\_length)$;
end
return $pcg$;
```

3.2. Synthesis of Pathological State PCG

To generate pathological PCGs, we extend Algorithm 1 by integrating clinical knowledge of valvular heart disease. We target three abnormalities: aortic stenosis (AS), aortic regurgitation (AR), and mitral regurgitation (MR). These conditions exhibit distinct acoustic signatures in PCG recordings, characterized by differences in murmur timing, intensity envelope, and spectral composition. All synthesized PCGs were generated at 8 kHz over 10-second intervals to align with the normal-state synthesis.

Pathologic murmurs were modeled as band-limited noise segments whose amplitude is shaped by task-specific envelopes and temporally anchored to the cardiac cycle (relative to S1/S2). This construction preserves consistency with the S1/S2 burst model above while reproducing known timing patterns.

Aortic stenosis (AS). AS is characterized by narrowing of the aortic valve opening, restricting blood ejection from the left ventricle to the aorta. The condition develops due to thickening and calcification of valve leaflets (aging, inflammation, or congenital abnormalities), reducing valve mobility. On auscultation, AS presents as an ejection murmur that peaks in early to mid-systole between S1 and S2.[3] We synthesize this by modulating white noise with an asymmetric bell-shaped envelope (crescendo–decrescendo across systole) and then applying a moving-average filter to obtain mid- to high-frequency content that closely matches the physiological AS murmur.

Aortic regurgitation (AR) / Mitral regurgitation (MR). AR and MR result from incomplete aortic and mitral valve closure during diastole and systole, respectively. AR manifests as a high-frequency diastolic decrescendo murmur following S2, while MR presents as a holosystolic murmur with uniform intensity from S1. Both are synthesized as band-limited noise: AR uses asymmetric bell curve modulation for the decrescendo pattern, whereas MR employs sustained amplitude for the quasi-steady murmur. Moving average filters smooth both signals to reproduce their characteristic high-frequency content (details in Appendix A).

4. Real-world PCG Data

This study compares the classification performance of DNN models trained using synthesized data with those trained solely on real-world data, in order to validate the efficacy of synthesized data in PCG classification. We evaluated the classification performance with two datasets, one publicly available BMD-HS dataset and one private dataset.

4.1. BMD-HS

The BMD-HS dataset (Ali et al., 2024) is a comprehensive collection of 864 heart sound recordings categorized into five common disease classes, designed for developing advanced machine learning models for automated heart sound classification and diagnosis. Unique to BMD-HS is its multi-label annotation system, capturing a diverse range of co-existing diseases and unique disease states, addressing a limitation in existing datasets that often lack comprehensive information on heart sound evaluations and disease categorizations. All recordings are of uniform duration, collected using the same stethoscope and standardized recording positions to minimize bias. Critically, diagnoses are confirmed via echocardiograms, enhancing the dataset's reliability. The BMD-HS dataset aims to improve cardiovascular disease diagnosis and management by providing a robust and clinically-sourced resource for cardiac health research.

4.2. Private PCG Dataset

The dataset comprises heart sound recordings collected from seven medical facilities in Japan, designed to support the development of machine learning models for automated detection of valvular heart diseases. Each case includes four recordings captured from standardized anatomical positions, yielding a total of $8,564$ recordings derived from $2,156$ unique cases. The dataset is annotated with multi-label classifications for AS, AR, and MR, with all diagnoses validated via echocardiograms to ensure clinical accuracy. By adhering to consistent recording protocols—including uniform equipment use and predefined anatomical positions—the dataset minimizes bias and variability, enhancing its reliability for research. This structured approach not only captures coexisting valvular pathologies but also provides a robust, clinically grounded resource to advance cardiac health research, particularly in correlating acoustic patterns with specific cardiovascular conditions. Further details are in Appendix C.

5. Experiment

To evaluate the effectiveness of training with synthesized PCG data, we formulated the problem as three independent binary classification tasks (Normal vs. AS, Normal vs. AR, and Normal vs. MR) and conducted a comparative analysis of abnormal PCG classification models across three types of abnormalities, comparing three training scenarios: (1) using only real-world data ("Real"), (2) using only synthesized data ("Syn"), and (3) fine-tuning a model pre-trained on synthesized data with real-world data ("Syn→Real"). Furthermore, to evaluate the qual-

[3] An example of synthesized AS PCG and real-world AS PCG are shown in Figure 5 in Appendix B.

ity of the synthesized PCGs, we conducted human evaluation.

5.1. Model Architectures

We conducted our experiments with various categories of DNN architectures, including CNN-based architectures, RNN-based architectures, and Transformer and its variants. Regarding CNN-based architectures, we examined ResNet18, ResNet34, ResNet50 (He et al., 2016), EfficientNet-B0, and EfficientNet-B1 (Tan and Le, 2019).[4] In addition to CNN-based architectures, we incorporated RNN-based architectures, including LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Chung et al., 2014). Furthermore, we evaluated Transformer (Vaswani et al., 2017)[5] for their effective performance in handling long sequence tasks.

5.2. Data Split

In this study, we maintained consistent data splits for the preliminary experiments and main experiment. All splits were performed at the patient level (patient-wise; group-stratified), ensuring that all recordings from each patient remained in a single partition. The splits for the real-world datasets used in the main experiments were prepared as follows. First, patients were divided into a train/validation set and a test set at an 8:2 ratio. Subsequently, the patients in the train/validation set were divided into a train set and a validation set at an 8:2 ratio, repeated six times to obtain six train/validation pairs. Out of these six pairs, one was used for hyperparameter tuning in abnormal PCG classification, while the remaining five were employed for training abnormal PCG classification models with the determined hyperparameters.[6]

5.3. Preliminary Experiments

We conducted four preliminary experments to determine the conditions for the abnormal PCG classification experiment. We first determined the optimal learning rate, then tuned the data-augmentation hyperparameters, selected the embedding module via experiments, and finally optimized the network-architecture hyperparameters (e.g., depth and number of heads). We used a single train and validation set pair from BMD-HS dataset for a preliminary experiment, and all the evaluation during preliminary experiment was conducted using validation set to avoid leakage from test set. Further details of the preliminary experiments are in Appendix D.

5.4. Classification of Abnormal PCGs

For each of the three binary classification tasks (Normal vs. AS, Normal vs. AR, and Normal vs. MR), we compared the three training scenarios as follows. Across all settings, we conducted five independent training runs using five distinct training/validation splits and evaluated each resulting model on a common real-world test set. AUROC served as the primary metric, with AUPRC, F1-score, sensitivity, specificity, PPV, and NPV additionally reported. For threshold-based metrics, a fixed decision threshold of 0.5 was used, and all reported values represent the mean across the five runs.

In the “Real” setting, to mitigate class imbalance, we assigned the positive class a sample weight equal to the negative-to-positive class ratio ($N_{\text{neg}}/N_{\text{pos}}$) computed on the training set. In the “Syn” setting, models were trained on $10,000$ synthetic samples ($5,000$ positive and $5,000$ negative) using the same hyperparameters as in the “Real” setting. In the “Syn→Real” setting, we initialized from the models trained under the “Syn” setting and fine-tuned them on the corresponding real-world training data, applying the same positive class reweighting as in the “Real” setting.

To assess transfer across datasets, we further considered two sequential protocols using the pair of real datasets, BMD-HS (A) and Private (B). (i) *“Real → Real”*: pre-train on A and fine-tune on B (and vice versa), and evaluate on the target (D2) test set. (ii) *“Syn → Real → Real”*: pre-train on 10,000 synthetic samples as above, then fine-tune on A and subsequently on B (and the reverse), evaluating on the final target. At each real-data stage, we reused the hyperparameters of the “Real” setting and applied positive-class reweighting computed on the current target training split. All protocols followed the same five-run procedure and common real-world test set described above; we report the mean across runs.

As for preprocessing, we applied scaling by subtracting the mean value from each sample and dividing by the standard deviation. As for data augmen-

[4] We convert 2D convolutional and pooling layers to its 1D counterparts following (Nonaka and Seita, 2021).

[5] We compare both causal and non-causal attention. Detailed in Appendix H.

[6] Details of a data split for the preliminary experiment and pretraining are in Appendix D.

Table 1: Sample size of each dataset.

Class	Real-world						Synthesized	
	BMD-HS			Private				
	Train	Val.	Test	Train	Val.	Test	Train	Val.
Normal	160	40	56	4,483	1,159	1,420	5,000	1,000
AS	216	32	48	270	92	85	5,000	1,000
AR	244	56	64	297	98	107	5,000	1,000
MR	200	40	64	440	132	148	5,000	1,000
Total *	568	120	176	5,459	1,393	1,712	20,000	4,000

* Due to the existence of samples with multiple diseases, the total number of samples is not equal to the simple sum of Normal, AR, AS, and MR cases.

tation, we applied random shifting, random masking, random flip, random scaling, random signal stretching and breathing sound addition with parameters related to each processing determined through hyper-parameter search. The batch size was set to 512[7], and the maximum number of epochs was set to 500, with validation conducted every five epochs. Early stopping was applied if the validation loss did not improve for five consecutive evaluations. We used Adam (Kingma and Ba, 2014) as the optimizer with the learning rate determined by hyper-parameter search. Regarding the loss function, we used binary cross entropy loss.

5.5. Synthetic Data Quality Evaluation

To comprehensively assess the quality of synthesized PCG signals, we conducted three complementary experiments. First, we performed a blind test with three medical professionals experienced in clinical heart sound analysis to evaluate perceptual quality and realism.[8] Second, we conducted a waveform feature-based similarity assessment between the synthetic and real-world datasets to quantify acoustic characteristics.[9] Third, we performed zero-shot classification experiments, where models trained exclusively on synthetic data were evaluated on real-world data, and vice versa, to assess cross-domain generalization performance.[10] Through these multifaceted experiments, we established a comprehensive quality evaluation framework for synthetic data based on their similarity to real-world PCG signals.

The blind test focused on four cardiac conditions: Normal, aortic stenosis (AS), aortic regurgitation (AR), and mitral regurgitation (MR). For each condition, 15 real-world samples from the BMD-HS dataset and 15 synthetic samples were randomly selected. All recordings exceeding 10 seconds were truncated to 10-second segments via random cropping for standardization. Participants performed two assessments: (1) distinguishing whether each sample was real or synthetic (binary classification) and (2) rating the perceived naturalness on a 5-point Likert scale. The inclusion of domain experts and balanced sample pairing ensured clinically grounded evaluation of synthesized PCG signal perceptual quality.

6. Result

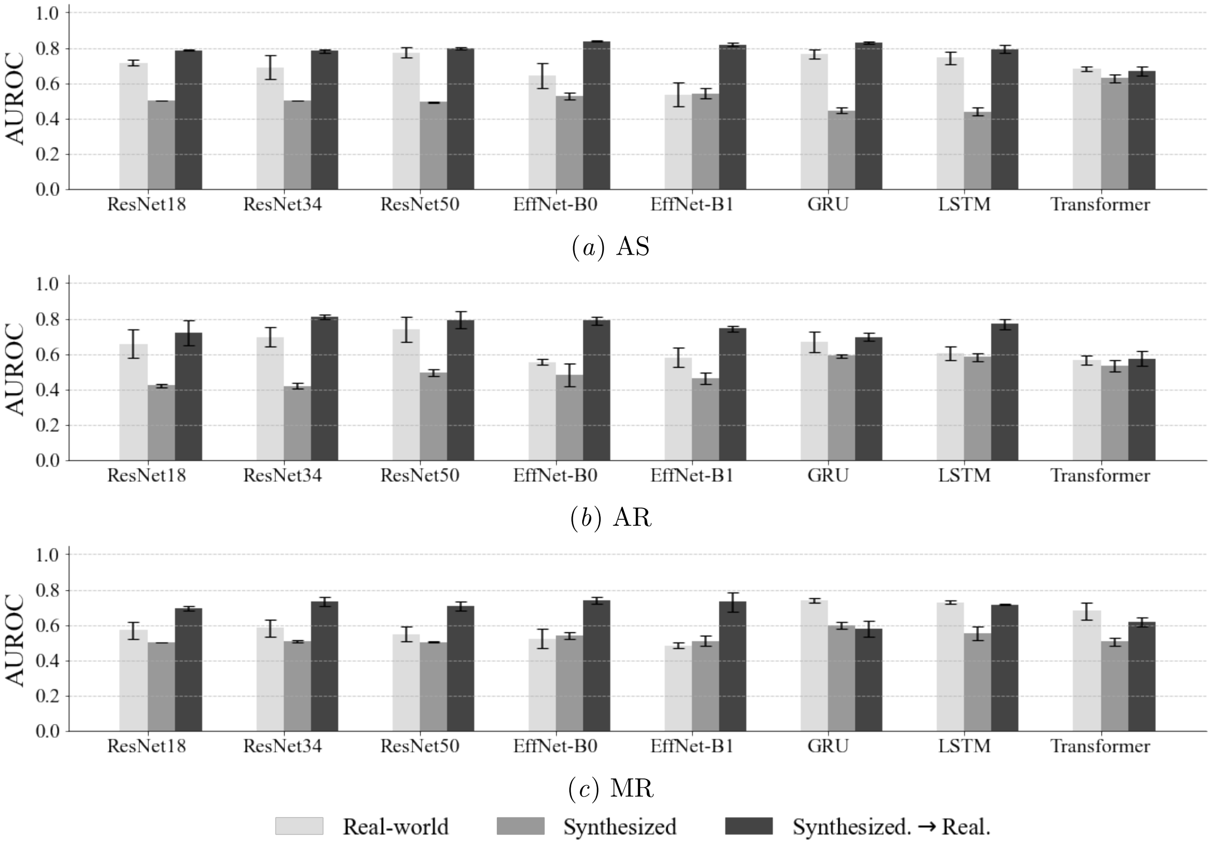
In this section, we present a obtained experimental results for abnormal PCG classification experiments. We first present result for abnormal PCG classification with three different training data setting, namely "Real", "Syn" and "Syn → Real", for three abnormal PCG classes respectively. We then report cross-dataset sequential fine-tuning and, finally, assess the quality of the synthetic data.

6.1. Classification of Abnormal PCG

6.1.1. BMD-HS

On BMD-HS (Figure 1), synthetic pretraining consistently improves AUROC over training on real data alone. For AS, the best "Real" model (ResNet50) reaches 0.7730, whereas the best "Syn→Real" model

[7] Batch sizes were set to 256 (ResNet50) and 128 (EfficientNet-B0/B1) due to GPU memory constraints.
[8] None of the these experts are authors on this paper.
[9] Details are shown in Appendix J.
[10] Details are shown in Appendix K.



(a) AS

(b) AR

(c) MR

Figure 1: AUROC comparison of models trained on real data ("Real") and those pre-trained on synthetic data before fine-tuning on real data ("Syn → Real") for AS, AR, and MR with BMD-HS dataset. Synthetic pretraining led to average improvements of 15.32% in AS, 16.96% in AR, and 17.10% in MR, showing its effectiveness in enhancing model performance. Error bars represent SEM (n=5).

(EfficientNet-B0) attains 0.8380 ($\Delta = +0.0650$); the mean relative gain across eight architectures is +15.32%. For AR, the best "Real" model (ResNet50) records 0.7393, while "Syn→Real" with ResNet34 yields 0.8105 ($\Delta = +0.0712$); the mean relative gain is +16.96%. For MR, although the best "Syn→Real" (EfficientNet-B0, 0.7388) is comparable to the best "Real" (GRU, 0.7405; $\Delta \approx -0.0017$), the average across eight architectures still improves by +17.10%.[11]

6.1.2. Private PCG Dataset

On the private dataset (Figure 2), synthetic pretraining also improves AUROC over real-only training, with mean relative gains of 7.11% (AS), 8.75% (AR), and 6.08% averaged over eight architectures and n=5 runs. For AS, the best "Real" model (GRU) attains AUROC 0.8357, whereas the best "Syn→Real" model (GRU) reaches 0.8917 (+0.0560 absolute). For AR, performance increases from 0.5978 (GRU, "Real") to 0.6598 (GRU, "Syn→Real"; +0.0620). For MR, the best score rises from 0.6110 (ResNet50, "Real") to 0.6835 (EfficientNet-B0, "Syn→Real"; +0.0725).[12]

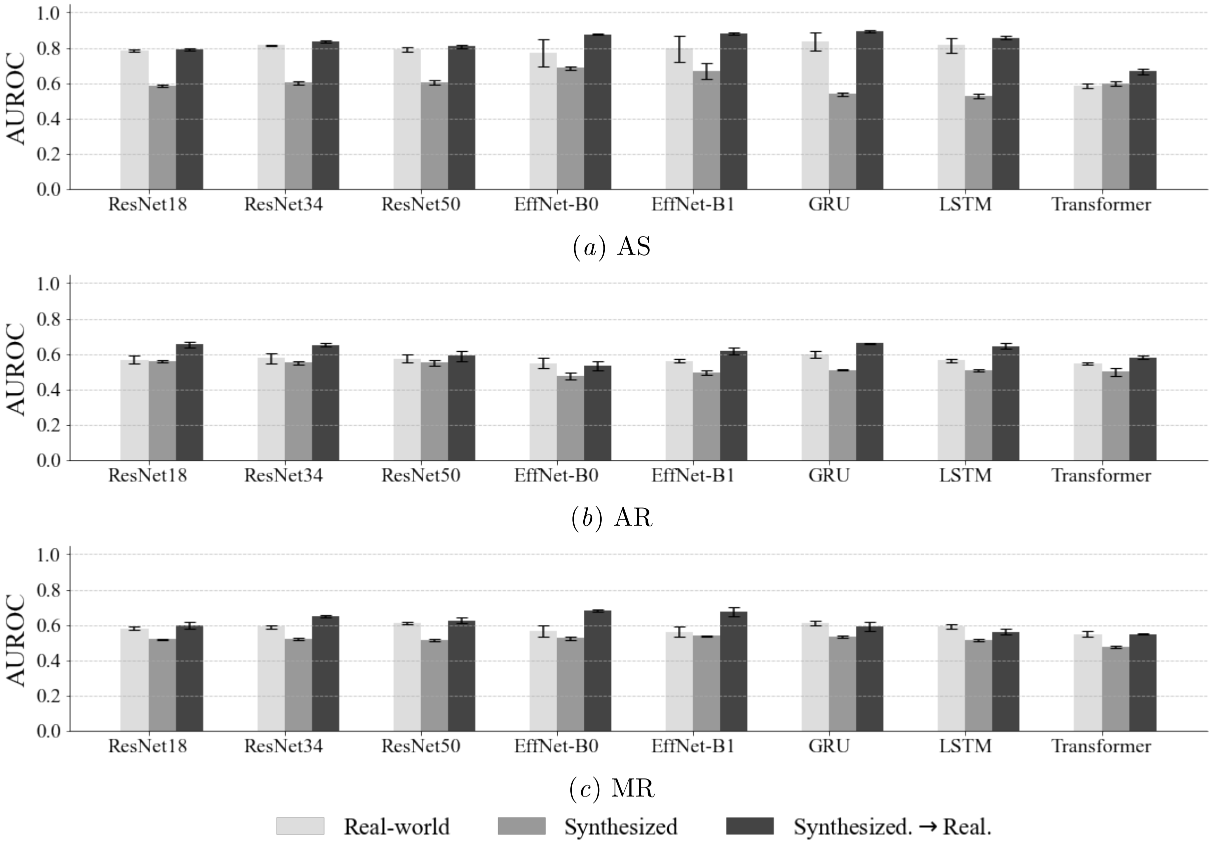
6.1.3. Cross-dataset sequential fine-tuning

We further evaluate cross-dataset transfer via sequential fine-tuning: Table 2 summarizes mean AUROC.[13] In all six dataset–target pairs,

[11] Details are in Appendix F (Table 10–Table 30).

[12] Details are in Appendix F (Table 31–Table 51).

[13] Detailed tables are in the Appendix G.



(*a*) AS

(*b*) AR

(*c*) MR

Figure 2: AUROC comparison of models trained on real data ("Real") and those pre-trained on synthetic data before fine-tuning on real data ("Syn → Real") for AS, AR, and MR with the private dataset. Synthetic pretraining led to average improvements of 7.11% in AS, 8.75% in AR, and 6.08% in MR, showing its effectiveness in enhancing model performance. Error bars represent SEM (n=5).

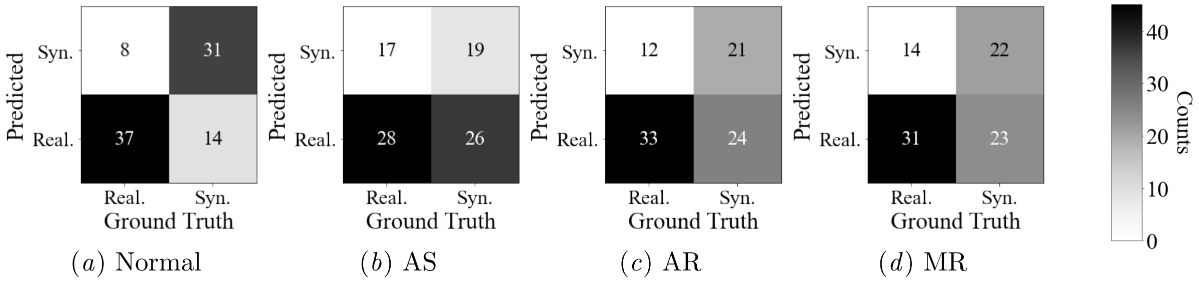
"Syn→Real→Real" achieves the best mean AUROC, surpassing "Real→Real" and the "Real" baseline.

6.2. Synthetic Data Quality Evaluation

The comprehensive evaluation revealed that our synthetic PCG signals capture perceptually salient features while exhibiting quantitative characteristics within real-world variability, though with limitations in cross-dataset transferability. **Human discrimination experiments** revealed that auditory-only evaluation resulted in misclassifications (as shown in Figure 3), though misidentification rates decreased under visualization conditions (detailed in Appendix I). Evaluators achieved high accuracy for normal heart sounds but found pathological conditions more challenging. When relying solely on auditory perception, experts classified samples as real based on respiratory sounds and as synthetic based on rhythm irregularities between S1 and S2. The **quantitative analysis** (presented in Appendix J) demonstrated that the distance between the two real datasets was consistently larger than the distances between synthetic and real data across all five metrics, indicating that synthetic data falls within the natural variation observed between real-world PCG datasets. However, **zero-shot cross-dataset evaluation** revealed limited performance, with AUROC values of 0.50–0.51 across most conditions, except for AS classification on the private dataset (TSTR AUROC= 0.60), suggesting limited preservation of disease-specific discriminative features for zero-shot transfer (detailed in Appendix K). These findings collectively indicate that while our synthetic PCG

Table 2: Mean AUROC for each setting.

Dataset	Target	Real	Syn	Syn→Real	Real→Real	Syn→Real→Real
BMD-HS	AS	0.6934	0.5096	0.7891	0.8165	0.8241
	AR	0.6336	0.4983	0.7336	0.7415	0.7852
	MR	0.6068	0.5267	0.6896	0.6472	0.7116
Private	AS	0.7322	0.6008	0.8260	0.8020	0.8288
	AR	0.5666	0.5180	0.6166	0.5834	0.6225
	MR	0.5818	0.5173	0.6162	0.5845	0.6252



(*a*) Normal (*b*) AS (*c*) AR (*d*) MR

Figure 3: Results of human evaluation for each PCG class.

signals preserve perceptually prominent features detectable by human experts and maintain quantitative characteristics within real-world variability ranges, they do not yet achieve sufficient quality for robust zero-shot cross-dataset transfer.

7. Discussion and Conclusion

Our experimental results demonstrate that pretraining with synthetic heart sound data contributed to improved classification performance on real-world data. Across eight architectures, mean relative AUROC gains are 15.32% (AS), 16.96% (AR), and 17.10% (MR) on BMD-HS (Figure 1), and 7.11% (AS), 8.75% (AR), and 6.08% (MR) on the private dataset (Figure 2). In our cross-dataset evaluation, synthetic pretraining is competitive with pretraining on real data from another source, and combining it with multi-stage real fine-tuning yields the best performance (Table 2), showing the complementary value of synthetic and real data in our framework. This effectiveness stems from synthetic data, generated with medical expertise to embed characteristic acoustic patterns, providing a better initialization than random.

In practice, this is especially valuable for PCG, where large, labeled real-world datasets are scarce. Pretraining on synthetic data mitigates the burden of extensive data collection and reduces privacy and ethical risks associated with patient data acquisition and sharing. Used as an intermediate learning step, it enables more efficient model development with less dependence on massive real-world datasets. However, models trained solely on synthetic data performed well in synthetic-domain evaluations but generalized poorly to real-world classification,[14] indicating that synthetic data serves best as a pretraining resource, while fine-tuning on real data remains essential for optimal performance in practical applications.

Limitations This study has three main limitations: (i) designing and scaling abnormal-PCG synthesis algorithms is labor-intensive; (ii) the relationship between synthetic-PCG fidelity and downstream performance is not well characterized; and (iii) we did not evaluate joint training on mixed synthetic and real data, so its benefits or pitfalls remain unknown. These factors currently constrain scalability and practical applicability.

[14] See Appendix E for synthetic-only training evaluated on synthetic data, and Appendix K (TSTR/TRTS) for evaluations on real data.

Acknowledgments

We would like to express our sincere gratitude to the participating hospitals for their invaluable cooperation in data collection for this study. This paper is based on results obtained from a project, JPNP23019, subsidized by the New Energy and Industrial Technology Development Organization (NEDO).

References

Shams Nafisa Ali, Afia Zahin, Samiul Based Shuvo, Nusrat Binta Nizam, Shoyad Ibn Sabur Khan Nuhash, Sayeed Sajjad Razin, SM Sani, Farihin Rahman, Nawshad Binta Nizam, Farhat Binte Azam, et al. Buet multi-disease heart sound dataset: A comprehensive auscultation dataset for developing computer-aided diagnostic systems. *arXiv preprint arXiv:2409.00724*, 2024.

Ali Almasi, Mohammad B Shamsollahi, and Lotfi Senhadji. A dynamical model for generating synthetic phonocardiogram signals. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5686–5689. IEEE, 2011.

Siddharth Biswal, Soumya Ghosh, Jon Duke, Bradley Malin, Walter Stewart, Cao Xiao, and Jimeng Sun. Eva: Generating longitudinal electronic health records using conditional variational autoencoders. In *Machine Learning for Healthcare Conference*, pages 260–282. PMLR, 2021.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Fabienne Debiais, L-G Durand, P Pibarot, and Robert Guardo. Time—frequency analysis of heart murmurs. part i: Parametric modelling and numerical simulations. *Medical and Biological Engineering and Computing*, 35(5):474–479, 1997.

Theekshana Dissanayake, Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Generalized generative deep learning models for biosignal synthesis and modality transfer. *IEEE Journal of Biomedical and Health Informatics*, 27 (2):968–979, 2022.

Amirata Ghorbani, Vivek Natarajan, David Coz, and Yuan Liu. Dermgan: Synthetic generation of clinical skin images with pathology. In *Machine learning for health workshop*, pages 155–170. PMLR, 2020.

Cristóbal González-Rodríguez, Miguel A Alonso-Arévalo, and Eloísa García-Canseco. Robust denoising of phonocardiogram signals using time-frequency analysis and u-nets. *IEEE Access*, 11: 52466–52479, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification, 2017. URL https://arxiv.org/abs/1609.09430.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Qixin Hu, Yixiong Chen, Junfei Xiao, Shuwen Sun, Jieneng Chen, Alan L Yuille, and Zongwei Zhou. Label-free liver tumor segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7422–7432, 2023.

Ainaz Jamshidi, Muhammad Arif, Sabir Ali Kalhoro, and Alexander Gelbukh. Synthetic time series data generation for healthcare applications: A pcg case study. *arXiv preprint arXiv:2412.16207*, 2024.

Pascalin Tiam Kapen, Mohamadou Youssoufa, Serge Urbain Kouam Kouam, Momo Foutse, André Rodrigue Tchamda, and Ghislain Tchuen. Phonocardiogram: A robust algorithm for generating synthetic signals and comparison with real life ones. *Biomedical Signal Processing and Control*, 60:101983, 2020.

Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. In

Asian Conference on Computer Vision (ACCV), 2020.

Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. *International Journal of Computer Vision (IJCV)*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Pedro Narváez and Winston S Percybrooks. Synthesis of normal heart sounds using generative adversarial networks and empirical wavelet transform. *Applied Sciences*, 10(19):7003, 2020.

Ahmed Ammar Naseer, Benjamin Walker, Christopher Landon, Andrew Ambrosy, Marat Fudim, Nicholas Wysham, Botros Toro, Sumanth Swaminathan, and Terry Lyons. Scoehr: Generating synthetic electronic health records using continuous-time diffusion models. In *Machine Learning for Healthcare Conference*, pages 489–508. PMLR, 2023.

Naoki Nonaka and Jun Seita. In-depth benchmarking of deep neural network architectures for ecg diagnosis. In *Machine Learning for Healthcare Conference*, pages 414–439. PMLR, 2021.

Naoki Nonaka and Jun Seita. Boosting ecg classification with synthesized data. In *1st International Workshop on Data-Centric Artificial Intelligence (DEARING) at ECML-PKDD 2024*, 2024a.

Naoki Nonaka and Jun Seita. Efficient self-supervised pretraining with simple synthesized ecg. In *Proceedings of the Machine Learning for Health Symposium (ML4H)*, 2024b.

Hojjat Salehinejad, Shahrokh Valaee, Tim Dowdell, Errol Colak, and Joseph Barfett. Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 990–994. IEEE, 2018.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018.

Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

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*, 30, 2017.

Moon Ye-Bin, Nam Hyeon-Woo, Wonseok Choi, Nayeong Kim, Suha Kwak, and Tae-Hyun Oh. Exploiting synthetic data for data imbalance problems: Baselines from a data perspective. *arXiv preprint arXiv:2308.00994*, 2023.

Chi Zhu, Jung-Hee Seo, and Rajat Mittal. Computational modeling and analysis of murmurs generated by modeled aortic stenoses. *Journal of Biomechanical Engineering*, 141(4):041007, 2019.

Appendix A. Details of the PCG Synthesis Algorithm

This section delineates the algorithm employed for PCG data synthesis. We first present a comprehensive description of the algorithm, followed by an exposition of the parameters utilized in the synthesis process. Finally, we provide additional exemplars of the synthesized PCG data to illustrate the efficacy of our approach.

A.1. Design Rationale and Modeling Choices

This subsection explains the modeling choices underlying the synthesis algorithm.

Envelope choice: asymmetric (split) Gaussian. S1/S2 are brief transients produced by rapid valve closure followed by damped tissue–blood vibrations. They exhibit an asymmetric time course—rapid onset and slower decay. A split (asymmetric) Gaussian

$$g_{\text{asym}}(\tau;\sigma_{\text{rise}},\sigma_{\text{decay}}) = \begin{cases} \exp\left(-\dfrac{\tau^2}{2\sigma_{\text{rise}}^2}\right), & \tau < 0, \\ \exp\left(-\dfrac{\tau^2}{2\sigma_{\text{decay}}^2}\right), & \tau \geq 0, \end{cases}$$

where $\sigma_{\text{rise}} < \sigma_{\text{decay}}$, captures this asymmetry with two interpretable time scales. The envelope is smooth with a matched value and slope at the peak (C^1 continuity), which reduces artificial high-frequency content and makes subsequent filtering/resampling stable.

Sinusoidal modulation for oscillatory bursts. Valve–tissue–blood dynamics behave as a lightly damped resonator, concentrating energy in characteristic bands. Multiplying g_{asym} by $\sin(2\pi f_s t + \phi_s)$ yields an oscillatory burst whose center f_s directly controls the spectral location of S1/S2 energy. Additional weak carriers can be superposed to emulate subtle multi-band structure without altering the envelope.

Cycle-anchored construction. Generating one S1 and one S2 burst per cycle and superposing them affords explicit control over heart rate, S1–S2 and S2–next-S1 intervals, and within-cycle variability. This anchoring is essential for placing murmurs (e.g., AS in systole; AR in early diastole; MR from S1 to S2) with correct timing relative to S1/S2.

Murmur modeling as amplitude-shaped, band-limited noise. Regurgitant or stenotic jets are turbulent and therefore noise-like. We synthesize murmurs by shaping white noise with lesion-specific amplitude envelopes and short time-domain smoothing (moving-average/FIR) to emphasize the clinically relevant band. This provides compact control over timing (window placement), intensity (envelope peak), and shape (crescendo/decrescendo/plateau) with few parameters.

A.2. Details of the Algorithm

```
Algorithm 2: Generate PCG for a Single Heartbeat
Input: $beat\_params$
Output: $pseudo\_pcg$: Synthesized PCG signal
$pseudo\_pcg \leftarrow \text{Zeros}(beat\_duration)$
/* Generate S1 Component                    */
$i\_wave \leftarrow GetAsymmetricPeak(beat\_params)$
$pseudo\_pcg \leftarrow$
  $ConcatSound(pseudo\_pcg, i\_wave, beat\_params)$

/* Generate S2 Component                    */
$ii\_wave \leftarrow GetAsymmetricPeak(beat\_params)$
$pseudo\_pcg \leftarrow$
  $ConcatSound(pseudo\_pcg, ii\_wave, beat\_params)$

/* Add noise and fluctuation                */
$pseudo\_pcg \leftarrow$
  $AddBaselineFluctuations(pseudo\_pcg)$
$pseudo\_pcg \leftarrow AddWhiteNoise(pseudo\_pcg)$
return $pseudo\_pcg$
```

The outline of algorithm for synthesizing PCG signals corresponding to a single cardiac cycle is shown in Algorithm 2. PCG synthesis is achieved through a sequential process that generates both primary heart sounds: S1 (first heart sound) and S2 (second heart sound). The algorithm initializes by creating a zero-filled array of predetermined beat duration. Subsequently, it employs an asymmetric peak generation function to synthesize the S1 component, which is then concatenated with the primary signal array according to specified beat parameters. An identical process is utilized for the S2 component generation. To enhance signal fidelity and better approximate physiological recordings, the algorithm incorporates two forms of signal modification: baseline fluctuations, which account for low-frequency physiological variations, and white noise addition, which simulates inherent recording artifacts and sensor noise. The algorithm accepts beat-specific parameters as input and outputs a complete synthetic PCG signal that exhibits characteristics consistent with physiological heart sounds.

Algorithm 3: GetAsymmetricPeak

```
Input: $peak\_freq$, $peak\_height$,
       $peak\_duration$, $neg\_side\_ratio$
Output: $wave$: Synthesized asymmetric peak
        wave
/* Generate base sine wave                  */
$sine\_wave \leftarrow GenSineWave(peak\_freq)$
/* Generate asymmetric bell curve           */
$y \leftarrow$
  $AsymmetricBell(peak\_duration, neg\_side\_ratio)$

/* Shift wave to align peak                 */
$wave \leftarrow$
  $ShiftWave(wave, shift\_len, shift\_direction)$
/* Normalize to peak height                 */
$wave \leftarrow NormalizeHeight(wave)$
return $wave$
```

Building upon the primary PCG synthesis framework, the asymmetric peak generation algorithm, shown in Algorithm 3, provides a detailed methodology for synthesizing the individual S1 and S2 components through a multi-step signal processing approach. Initially, the algorithm generates a fundamental sine wave at a specified peak frequency, which serves as the base carrier signal. This is followed by the creation of an asymmetric bell curve envelope, parameterized by the peak duration and negative side ratio, which modulates the amplitude characteristics of the resulting waveform. The algorithm then performs a temporal shift operation to achieve precise peak alignment, ensuring proper temporal positioning of the sound component within the cardiac cycle. Finally, the waveform undergoes amplitude normalization to achieve the desired peak height, resulting in a properly scaled asymmetric peak that accurately represents the morphological characteristics of heart sound components. This algorithmic approach accepts four critical parameters: peak frequency, peak height, peak duration, and negative side ratio, enabling fine-tuned control over the generated waveform's characteristics.

Algorithm 4: Generate Asymmetric Bell

```
Input: $gen\_len$, $std\_pos$, $neg\_ratio$
Output: $asymmetric\_bell$: Generated
        asymmetric bell curve
/* Compute standard deviation for
   negative side                            */
$std\_neg \leftarrow std\_pos \times neg\_ratio$
/* Generate x values                        */
$x \leftarrow \mathrm{Linspace}(-10, 10, gen\_len)$
/* Compute scaling factors                  */
$scaling\_pos \leftarrow \frac{1}{std\_pos \times \sqrt{2\pi}}$
$scaling\_neg \leftarrow \frac{1}{std\_neg \times \sqrt{2\pi}} \times \frac{std\_neg}{std\_pos}$
/* Compute asymmetric bell curve            */
foreach $x_i \in x$ do
    if $x_i \geq 0$ then
        $asymmetric\_bell[i] \leftarrow$
          $scaling\_pos \times \exp(-0.5 \times (x_i/std\_pos)^2)$
    else
        $asymmetric\_bell[i] \leftarrow$
          $scaling\_neg \times \exp(-0.5 \times (x_i/std\_neg)^2)$
    end
end
return $asymmetric\_bell$
```

Expanding on the asymmetric peak generation process, the Generate Asymmetric Bell Curve algorithm, shown in Algorithm 4, defines the shape of the amplitude envelope used for S1 and S2 synthesis. The algorithm constructs an asymmetric Gaussian-like curve by separately defining the standard deviation for its positive and negative sides. It first calculates the negative-side standard deviation as a scaled version of the positive-side standard deviation, controlled by the negative side ratio parameter. A set of linearly spaced x-values is then generated over a predefined range, ensuring adequate resolution for waveform synthesis. The algorithm computes separate scaling factors for the positive and negative regions to normalize the probability density function, maintaining amplitude continuity. For each x-value, it evaluates a Gaussian function with the respective standard deviation, producing an asymmetric bell-shaped curve that serves as the amplitude envelope. This approach enables flexible shaping of the temporal structure of heart sounds, facilitating realistic PCG synthesis that accurately captures the physiological variations observed in cardiac auscultation.

A.3. Algorithm for Synthesis of Pathological State PCG

A.3.1. AS State PCG

Algorithm 5: Generate AS state PCG for a Single Heartbeat

```
Input: $beat\_params$
Output: $pseudo\_pcg$: Synthesized AS state PCG signal
$pseudo\_pcg \leftarrow \text{Zeros}(beat\_duration)$
/* Generate S1 Component */
$i\_wave \leftarrow GetAsymmetricPeak(beat\_params)$
$pseudo\_pcg \leftarrow ConcatSound(pseudo\_pcg, i\_wave, beat\_params)$

/* Generate S2 Component */
$ii\_wave \leftarrow GetAsymmetricPeak(beat\_params)$
$pseudo\_pcg \leftarrow ConcatSound(pseudo\_pcg, ii\_wave, beat\_params)$

/* Generate AS noise Component */
$as\_noise \leftarrow GenerateASnoise(beat\_params)$
$pseudo\_pcg \leftarrow ConcatSound(pseudo\_pcg, as\_noise)$
/* Add noise and fluctuation */
$pseudo\_pcg \leftarrow AddBaselineFluctuations(pseudo\_pcg)$
$pseudo\_pcg \leftarrow AddWhiteNoise(pseudo\_pcg)$
return $pseudo\_pcg$
```

Details of the algorithm to synthesize AS state PCG is shown in Algorithm 5 and Algorithm 6. The overall procedure is the same as the synthesis of normal state PCG with Algorithm 1, but with Algorithm 5 instead of Algorithm 2. Algorithm 5, which generates AS state PCG for a single heartbeat, initializes an empty signal and sequentially constructs the key components of the PCG: the first heart sound (S1), the second heart sound (S2), and the AS-associated noise. The heart sounds (S1 and S2) are synthesized using an asymmetric peak function (Algorithm 3) and appended to the signal. The AS noise component is generated separately using the GenerateASnoise function (Algorithm 6), which creates a bell-shaped noise waveform modulated by white noise and scaled based on given amplitude parameters. The noise is then smoothed with Algorithm 7 and optionally shifted in time according to the defined parameters. After assembling these com-

Algorithm 6: GenerateASnoise

```
Input: $beat\_params$
Output: $noise$: AS_noise
$wn\_noise \leftarrow \text{WhiteNoise}(beat\_params.beat\_duration)$
/* Scale white noise with given parameter */
$max\_wn\_amp \leftarrow beat\_params.max\_wn\_amp$
$min\_wn\_amp \leftarrow beat\_params.min\_wn\_amp$
$scaled\_wn\_noise \leftarrow wn\_noise * (max\_wn\_amp - min\_wn\_amp)$
/* Bell curved noise */
$bell\_curve \leftarrow GetAsymmetricPeak(beat\_params)$
$as\_noise \leftarrow scaled\_wn\_noise \times (bell + min\_wn\_amp)$
/* Smooth noise */
$as\_smoothing\_size \leftarrow beat\_params.as\_smooth$
$as\_noise \leftarrow NoiseSmoothing(as\_noise, as\_smoothing\_size)$
/* Shift AS noise */
$shift\_len \leftarrow beat\_params.shift\_len$
$direction \leftarrow beat\_params.shift\_direction$
$as\_noise \leftarrow ShiftWave(as\_noise, shift\_len, direction)$
return $as\_noise$
```

Algorithm 7: NoiseSmoothing

```
Input: $noise\_seg$, $window\_width$
Output: $smoothed\_noise$: Smoothed noise segment
/* Compute moving average */
$kernel \leftarrow \mathrm{Ones}(window\_width)/window\_width$
$smoothed\_noise \leftarrow \mathrm{Convolve}(noise\_seg, kernel)$
return $smoothed\_noise$
```

ponents, the synthesized PCG undergoes additional processing, including baseline fluctuations and white noise addition, to enhance realism. The final result is a pseudo PCG signal that simulates the acoustic characteristics of a heartbeat with AS.

A.3.2. AR State PCG

Algorithm 8: AR state PCG synthesis

```
Input: target_length
Output: $pcg$: Synthesized AR state PCG signal
$pcg \leftarrow []$; $pcg\_params \leftarrow GenerateParams()$;
 while $length(pcg) < target\_length$ do
    $beat \leftarrow GenerateBeat(pcg\_params)$;
    $pcg \leftarrow pcg + beat$;
    /* Generate AR noise and add in between S1 and S2 */
    $ar\_noise \leftarrow GenerateARnoise(pcg, pcg\_params)$;
    $pcg \leftarrow ConcatSound(pcg, ar\_noise)$
    $pcg\_params \leftarrow PerturbParams(pcg\_params)$;
end
if $length(pcg) > target\_length$ then
    $pcg \leftarrow Trim(pcg, target\_length)$;
end
return $pcg$;
```

Details of the algorithm to synthesize AR state PCG are shown in Algorithm 8 and Algorithm 9. The AR state PCG synthesis algorithm constructs a continuous PCG signal by generating heartbeats iteratively until the target length is reached, similar to Algorithm 1. The key difference is that Algorithm 8 inserts AR-specific noise between the previously concatenated S2 and the S1 of the newly generated beat. This noise is generated by the Algorithm 9, which creates a bell-shaped noise envelope modulated by white noise. The noise amplitude is scaled based on predefined minimum and maximum levels, simulating the murmur associated with AR, similar to Algorithm 6. The final PCG signal is trimmed to match the target duration before returning the synthesized output. This structured approach ensures realistic simulation of AR-related heart sound characteristics.

Algorithm 9: GenerateARnoise

```
Input: $beat\_params$
Output: $noise$: AR_noise
$wn\_noise \leftarrow \mathrm{WhiteNoise}(beat\_params.beat\_duration)$
/* Scale white noise with given parameter */
$max\_wn\_amp \leftarrow beat\_params.max\_wn\_amp$
$min\_wn\_amp \leftarrow beat\_params.min\_wn\_amp$
$scaled\_wn\_noise \leftarrow wn\_noise * (max\_wn\_amp - min\_wn\_amp)$
/* Bell curved noise */
$bell\_curve \leftarrow GetAsymmetricPeak(beat\_params)$
$ar\_noise \leftarrow scaled\_wn\_noise \times (bell + min\_wn\_amp)$
/* Smooth noise */
$ar\_smoothing\_size \leftarrow beat\_params.ar\_smooth$
$ar\_noise \leftarrow NoiseSmoothing(ar\_noise, ar\_smoothing\_size)$
return $ar\_noise$
```

A.3.3. MR State PCG

Details of the algorithm to synthesize MR state PCG are shown in Algorithm 10 and Algorithm 11. Similar to the synthesis of an aortic stenosis (AS) state PCG, the overall procedure (Algorithm 10) follows the same framework as the synthesis of a normal state PCG using Algorithm 1, but replaces Algorithm 2 with Algorithm 10. To model the pathological MR noise, a segment of white noise is generated and scaled according to predefined amplitude constraints, as shown in Algorithm 10. This noise component is then temporally aligned by shifting it to begin after the S1 onset. Finally, the synthesized signal undergoes post-processing to introduce baseline fluctuations and white noise, enhancing its realism. This approach effectively captures the key acoustic characteristics of MR pathology while maintaining physiological plausibility.

Algorithm 10: Generate MR state PCG for a Single Heartbeat

```
Input: $beat\_params$
Output: $pseudo\_pcg$: Synthesized MR state PCG signal
$pseudo\_pcg \leftarrow$ Zeros($beat\_duration$)
/* Generate S1 Component */
$i\_wave \leftarrow GetAsymmetricPeak(beat\_params)$
$pseudo\_pcg \leftarrow$
  $ConcatSound(pseudo\_pcg, i\_wave, beat\_params)$

/* Generate S2 Component */
$ii\_wave \leftarrow GetAsymmetricPeak(beat\_params)$
$pseudo\_pcg \leftarrow$
  $ConcatSound(pseudo\_pcg, ii\_wave, beat\_params)$

/* Generate MR noise Component */
$mr\_noise \leftarrow GenerateMRnoise(beat\_params)$
$pseudo\_pcg \leftarrow$
  $ConcatSound(pseudo\_pcg, mr\_noise)$
/* Add noise and fluctuation */
$pseudo\_pcg \leftarrow$
  $AddBaselineFluctuations(pseudo\_pcg)$
$pseudo\_pcg \leftarrow AddWhiteNoise(pseudo\_pcg)$
return $pseudo\_pcg$
```

Algorithm 11: GenerateMRnoise

```
Input: $beat\_params$
Output: $noise$: MR Noise
$wn\_noise \leftarrow$
  WhiteNoise($beat\_params.mr\_duration$)
/* Scale white noise with given
   parameter */
$max\_wn\_amp \leftarrow beat\_params.max\_wn\_amp$
$min\_wn\_amp \leftarrow beat\_params.min\_wn\_amp$
$mr\_noise \leftarrow wn\_noise * (max\_wn\_amp -$
  $min\_wn\_amp) + min\_wn\_amp$
/* Smooth noise */
$mr\_smoothing\_size \leftarrow beat\_params.mr\_smooth$
$mr\_noise \leftarrow$
  $NoiseSmoothing(mr\_noise, mr\_smoothing\_size)$

/* Shift start of MR noise */
$shift\_len \leftarrow beat\_params.S1\_start$
$direction \leftarrow$ positive
$mr\_noise \leftarrow$
  $ShiftWave(mr\_noise, shift\_len, direction)$
return $mr\_noise$
```

A.4. Parameters of the Algorithm

In this section, we show parameter settings used to synthesize PCG data (shown in Table 3). For each sample, initial values of peak parameters were randomly selected by sampling from uniform distribution or Gaussian distribution, as shown in Algorithm 12. These base parameters are then further perturbed to introduce controlled variations in the signal characteristics, as shown in Algorithm 13. The perturbation process applies noise to each parameter individually, following two possible schemes: For parameters requiring Gaussian noise, random values are drawn from a normal distribution with a specified standard deviation and added to the base value. For parameters requiring uniform noise, random values are sampled from a uniform distribution within specified minimum and maximum bounds and added to the base value. This two-step randomization process ensures both diversity in the initial parameter space and fine-grained local variations around these base values, contributing to the synthetic dataset's richness and variability.

Table 3: Parameters used for PCG synthesis.

	Category	Base value	Per-sample perturbation	Per-beat perturbation
S1	frequency	24.00	$\mathcal{N}(0,\ 2.5)$	$\mathcal{N}(0,\ 1.0)$
	duration	0.75	$\mathcal{U}(-0.25,\ 0.25)$	$\mathcal{N}(0,\ 0.05)$
	height	2.50	$\mathcal{U}(-0.50,\ 0.50)$	$\mathcal{N}(0,\ 0.025)$
	neg_ratio	0.25	$\mathcal{N}(0,\ 0.05)$	$\mathcal{N}(0,\ 0.005)$
	shift	0.20	$\mathcal{N}(0,\ 0.05)$	$\mathcal{N}(0,\ 0.005)$
S2	frequency	60.00	$\mathcal{N}(0,\ 5.0)$	$\mathcal{N}(0,\ 1.0)$
	duration	0.25	$\mathcal{U}(-0.10,\ 0.10)$	$\mathcal{N}(0,\ 0.025)$
	height	3.00	$\mathcal{U}(-0.50,\ 0.50)$	$\mathcal{N}(0,\ 0.05)$
	neg_ratio	0.25	$\mathcal{N}(0,\ 0.05)$	$\mathcal{N}(0,\ 0.005)$
	shift	0.55	$\mathcal{N}(0,\ 0.075)$	$\mathcal{N}(0,\ 0.005)$

Algorithm 12: GenerateParams

```
Input: None
Output: pcg_param: Parameters for each sample
$pcg\_param \leftarrow$ empty dictionary
foreach $param\_name \in pcg\_parameters$ do
    /* Get parameter config for parameter */
    $param\_config \leftarrow pcg\_parameters[param\_name][per\text{-}sample]$
    /* Get base value for parameter */
    $param\_val \leftarrow param\_config[\text{“}base\_values''\,]$
    /* Sample noise from specified distribution */
    $noise \sim param\_config[\text{“}per\text{-}sample\_perturbation''\,]$
    /* Apply noise to parameter value */
    $pcg\_param[param\_name] \leftarrow param\_val + noise$
end
return $pcg\_param$
```

Algorithm 13: PerturbParams: Perturbing per sample parameters

```
Input: per-sample_params: A namespace containing per sample parameters
Output: per-beat_param: A namespace containing perturbed parameters
$per - beat\_param \leftarrow$ empty dictionary
foreach $(param\_name, per\text{-}sample\_param\_val) \in per\text{-}sample\_params$ do
    /* Get parameter config for parameter */
    $param\_config \leftarrow pcg\_parameters[param\_name][per\text{-}beat]$
    /* Sample noise from specified distribution and add to per_sample parameter */
    $noise \sim param\_config[\text{“}per\text{-}beat\_perturbation''\,]$
    $perturbed\_param \leftarrow per\text{-}sample\_param\_val + noise$;
    $perturbed\_param[key] \leftarrow perturbed\_param$
end
return $perturbed\_param$
```

A.5. Parameter Selection Procedure

In determining the synthesis parameters for the PCG synthesis algorithm presented in Table 3, we established a systematic approach based on expert knowledge and physiological considerations. The baseline parameters were carefully determined through comprehensive feedback from medical experts, ensuring clinical relevance and accuracy. When addressing the variability of parameters, we deliberately designed the algorithm to maintain larger inter-patient variations compared to beat-to-beat variations within individual patients, which better reflects the natural physiological differences observed among different subjects in clinical settings.

Appendix B. Visualization of PCG Data

In this section, we present representative examples of both real-world and synthesized PCG recordings for the four diagnostic categories: Normal, Aortic Stenosis (AS), Aortic Regurgitation (AR), and Mitral Regurgitation (MR) (Figure 4, Figure 5, Figure 6, Figure 7). Each condition is shown separately for real and synthetic data to illustrate their morphological characteristics and facilitate visual comparison. These datasets, comprising real and synthetic PCGs across the four categories, were subsequently employed to conduct DNN training experiments reported in the main text.

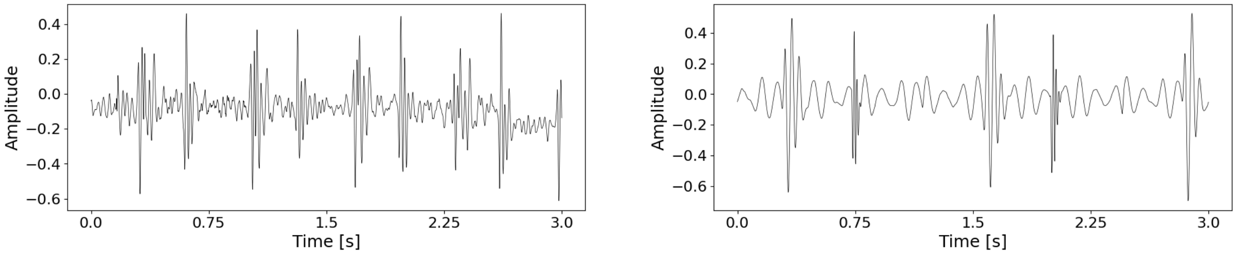
Appendix C. Details of the Private Dataset

Table 4: Details of the private dataset demographics

	Female	Male	Total
Size	881 (40.9%)	1, 275 (59.1%)	2, 156 (100.0%)
Age	70.4	69.2	69.7
AS	53 (6.0%)	61 (4.8%)	114 (5.3%)
AR	38 (4.3%)	90 (7.1%)	128 (5.9%)
MR	66 (7.5%)	116 (9.1%)	182 (8.4%)

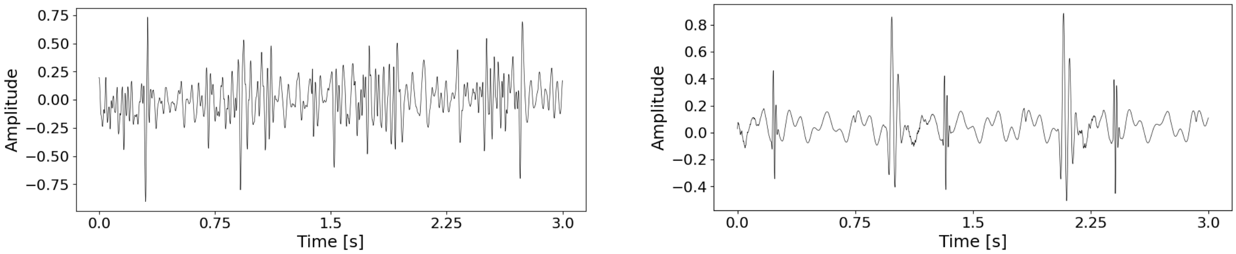
Table 4 summarizes demographic and clinical characteristics of a private dataset comprising 2, 156 participants. The dataset exhibits a gender imbalance, with 1, 275 males (59.1%) and 881 females (40.9%). Participants' mean age is 69.7 years, with females slightly older (70.4) than males (69.2). Three clinical conditions—AS, AR, and MR—are reported with frequencies and prevalence rates. AS affects 114 participants (5.3% overall), with higher female prevalence (6.0% vs. 4.8% in males). AR and MR demonstrate higher prevalence among males: AR occurs in 7.1% of males (90 cases) versus 4.3% of females (38 cases), totaling 128 cases (5.9% overall), while MR is present in 9.1% of males (116 cases) compared to 6.0% of females (66 cases), totaling 182 cases (8.4% overall). All percentages represent within-group proportions (gender-specific for conditions, total sample for demographic distributions).

Subjects were selected based on the following criteria: individuals who consented to provide data for research aimed at developing medical devices for early detection of cardiac diseases and for use in research and development at Japanese research institutions and companies, and who were able to undergo (or had undergone) cardiac ultrasound and blood tests. Data collection was performed using the latest heart sound and electrocardiogram device under development at AMI Inc. Measurements were taken non-invasively by placing the device on the anterior chest wall in a hospital room. Four measurement sites were used: 2nd Right Intercostal Space at the Sternal Border (2RSB), 2nd Left Intercostal Space at the Sternal Border (2LSB), 4th Left Intercostal Space at the Sternal Border (4LSB), and 5th Intercostal Space at the Left Midclavicular Line (5LMCL). The total measurement time for all four sites was approximately 5 minutes. During the measurements, various sources of noise were recorded, which may include human voices (the subject's own, guidance voices, and surrounding voices), alarm sounds emitted from measurement equipment and peripheral equipment, biological sounds such as stomach sounds, breathing sounds, and rubbing noises.



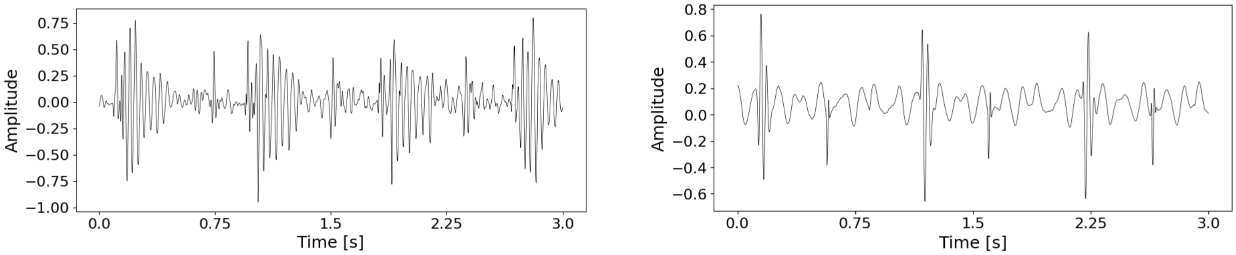
(a) Real-world (b) Synthesized

Figure 4: Examples of real-world and synthesized normal state PCG.



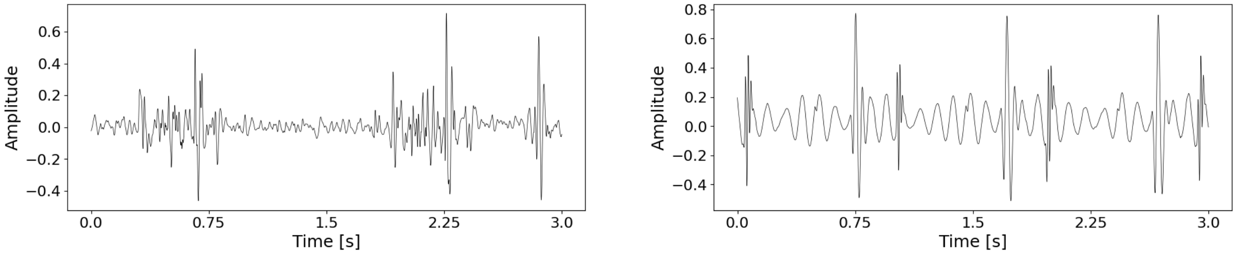
(a) Real-world (b) Synthesized

Figure 5: Examples of real-world and synthesized PCG of AS.



(a) Real-world (b) Synthesized

Figure 6: Examples of real-world and synthesized PCG of AR.



(a) Real-world (b) Synthesized

Figure 7: Examples of real-world and synthesized PCG of MR.

Appendix D. Details of the Preliminary Experiments

D.1. Search for a Optimal Learning Rate

To determine the optimal learning rate for PCG classification, we conducted a comprehensive exploration using the MR classification task from the BMD-HS dataset. The investigation involved testing ten different learning rates ranging from 1e-6 to 1e-3 (specifically: 1e-6, 2e-6, 5e-6, 1e-5, 2e-5, 5e-5, 1e-4, 2e-4, 5e-4, and 1e-3) with a ResNet18 architecture and Adam optimizer. The model was trained on the training set, and the optimal learning rate was selected based on the highest $F1$-score achieved on the validation set.

Table 5: Result of a optimal learning rate search

Learning rate	$F1$-score	AUROC	AUPRC
$1E-03$	0.167	0.430	0.347
$5E-04$	0.000	0.413	0.348
$2E-04$	0.513	0.455	0.362
$1E-04$	0.503	0.455	0.364
$5E-05$	0.652	0.727	0.592
$2E-05$	0.672	0.747	0.619
$1E-05$	0.656	0.748	0.613
$5E-06$	0.000	0.455	0.411
$2E-06$	0.000	0.465	0.416
$1E-06$	0.000	0.467	0.418

As shown in Table 5, 2E-05 achieved the best performance across all evaluated metrics, with an $F1$-score of 0.672, AUROC of 0.747, and AUPRC of 0.619. The results demonstrated a clear pattern where extremely low learning rates (1E-06 to 5E-06) failed to learn effectively, resulting in $F1$-scores of 0.000, while very high learning rates (5E-04 to 1E-03) also performed poorly with $F1$-scores below 0.200. The model showed stable and relatively good performance in the middle range of learning rates (1E-05 to 5E-05), with $F1$-scores above 0.650, indicating this range as the optimal zone for the PCG classification task. Based on these results, we adopted a learning rate of 2E-05 for subsequent experiments.

D.2. Search for a Optimal Augmentation

This section provides an overview of the data augmentation strategies employed in the study. We first introduce the types of augmentation functions utilized to enhance dataset diversity and robustness, followed by a summary of the parameter ranges explored to control the intensity and variation of these transformations. Additionally, we outline the experimental framework designed to systematically evaluate and select optimal augmentation parameters, ensuring their alignment with the goals of improving model generalization without introducing detrimental artifacts. The obtained augmentation setting was used for the abnormal PCG classification experiment.

D.2.1. Augmentations

In this work we employed six augmentations, namely breathing sound addition, random masking, random shifting, random flipping, random extension, and random scaling. We explain the details of each augmentation procedure and parameters associated with each augmentation procedure.

The breathing sound addition procedure enhances PCG data by synthesizing and integrating simulated respiratory noise. This involves generating a physiologically plausible breathing waveform through white noise modulated by a sine wave to mimic inhalation-exhalation patterns, with randomized breath durations. The synthesized sound is smoothed using a windowing function, extended to match the PCG length, and scaled to randomized amplitudes (within predefined limits) before being added to the original recording. If the breathing sound exceeds the PCG duration, a random segment is selected for alignment. The output combines the original cardiac signals with controlled breathing interference.

To enhance the robustness of models against data loss or artifacts commonly encountered in real-world recordings, a random masking procedure is employed. This technique simulates transient data gaps by selectively obscuring contiguous segments of the PCG signal during augmentation. A mask_ratio parameter controls the proportion of the signal to be masked (e.g., 0.2 for 20%). For each sample, the absolute mask_width is calculated based on the signal length and the specified mask_ratio. A random starting index, mask_start, is then chosen to ensure the masked segment remains within the signal boundaries. By introducing variability through the occlusion of random signal regions, this augmentation strategy encourages models to learn robust features that are invariant to partial data loss, improving their generalization performance on real-world PCG recordings where such gaps or artifacts may be present.

To enhance the robustness of the models to minor temporal variations in the PCG signal, a random shifting augmentation is applied. This technique introduces small, random shifts in the signal's timing. A shift_ratio parameter controls the maximum proportion of the signal length that can be shifted. For each sample, a random shift ratio is selected within the defined maximum. This ratio is then used to calculate the shift_size in number of samples. The signal is then either shifted to the left or right with equal probability. Padding with zeros is used to maintain the original signal length after the shift. Specifically, for a left shift, zeros are prepended to the signal, and the resulting signal is truncated to the original length. Conversely, for a right shift, zeros are appended, and the beginning of the shifted signal is truncated. This process effectively simulates minor misalignments or variations in the timing of the heart sounds within the recording, improving the model's ability to generalize to such variations in real-world data.

To augment the dataset with variations in signal polarity, a random flipping technique is employed. This method randomly inverts the PCG signal with a specified flip_rate probability. For each sample, a random number is generated. If this number is less than the flip_rate, the entire PCG signal is multiplied by -1, effectively flipping it vertically. This augmentation simulates scenarios where the recording electrodes might be inadvertently placed with reversed polarity, ensuring that the trained models are insensitive to such signal inversions and can accurately analyze PCG data regardless of its polarity.

To introduce variations in the temporal characteristics of the PCG signal, a signal stretching augmentation is implemented. This technique slightly alters the duration of the signal, simulating variations in heart rate or recording speed. A stretch_ratio parameter controls the extent of stretching, allowing for both expansion and compression of the signal. For each sample, a random stretch factor is determined within a range defined by the stretch_ratio. Linear interpolation is then used to resample the signal to the new length. If the stretched signal is longer than the original, a random segment is extracted to match the original length. Conversely, if the stretched signal is shorter, it is padded with zeros on both sides, with the padding distribution randomized, to restore the original length. This augmentation helps to improve the model's robustness to minor variations in the timing and duration of heart sounds within the PCG recording.

To introduce variations in the amplitude of the PCG signal, a random scaling augmentation is applied. This method simulates fluctuations in signal strength that might occur due to variations in contact with the patient or other recording artifacts. A scale_ratio parameter controls the range of possible scaling factors. For each sample, a scaling factor is generated using a sine wave with a random frequency within a range determined by the scaler_freq parameter. This sine wave modulates the scaling factor, creating smooth variations in amplitude across the signal. The PCG signal is then multiplied by this dynamically changing scaling factor. This augmentation encourages the model to be robust to variations in signal amplitude and improves generalization performance by exposing the model to a wider range of realistic signal intensities.

D.2.2. Augmentation parameter

Table 6: Search range of augmentation parameters.

	Sampling type	min	max
mask_ratio	uniform	0.00	0.90
shift_ratio	uniform	0.00	0.90
flip_rate	uniform	0.00	0.90
breathing_scale	log uniform	0.25	4.00
stretch_ratio	log uniform	0.25	4.00
scale_ratio	log uniform	0.25	4.00

To identify optimal augmentation parameters for the PCG classification task, we conducted an exhaustive hyperparameter search for each classification setting (AR, MR, AS) using the BMD-HS dataset. With a fixed learning rate of 2E-05, ResNet18 model, and Adam optimizer, we independently explored augmentation configurations for each task. Each search was constrained to a maximum duration of 24 hours, with the selection criteria focused on maximizing the $F1$-score on the validation set. The minimum and maximum range of each parameter and sampling strategies are shown in Table 6. This comprehensive approach enabled us to fine-tune augmentation strategies specific to the unique characteristics of each classification task, allowing the model to achieve tailored performance across different PCG classification scenarios.

The augmentation parameter search revealed distinct optimal configurations for each PCG classification task (AR, MR, AS). The results are shown in Table 7. For AR classification, the optimal parameters included a low mask ratio of 0.039, a relatively high shift ratio of 0.618, minimal flip rate of 0.019, high breathing scale of 0.598, moderate stretch ratio of 0.343, and scale ratio of 0.315. In contrast, AS classification favored a very low mask ratio of 0.018, a moderate shift ratio of 0.435, high flip rate of 0.585,

Table 7: Result of augmentation parameter search.

	AR	AS	MR
mask_ratio	0.039	0.018	0.112
shift_ratio	0.618	0.435	0.242
flip_rate	0.019	0.585	0.489
breathing_scale	0.598	0.618	0.264
stretch_ratio	0.343	0.253	0.311
scale_ratio	0.315	1.398	0.925

high breathing scale of 0.618, lower stretch ratio of 0.253, and notably high scale ratio of 1.398. The MR classification showed yet another unique configuration, with a higher mask ratio of 0.112, lower shift ratio of 0.242, substantial flip rate of 0.489, moderate breathing scale of 0.264, moderate stretch ratio of 0.311, and scale ratio of 0.925. These divergent optimal parameters underscore the importance of task-specific augmentation strategies in improving classification performance across different PCG scenarios.

D.3. Comparison of Embedding Modules

To identify the optimal embedding module for RNN-based and Transformer models in the PCG classification task, we conducted a grid search using the MR classification setting of the BMD-HS dataset. We compared four settings: (i) no embedding module[15], (ii) Linear embedding, (iii) STFT embedding, and (iv) combined Linear+STFT embedding.

The *Linear embedding* module divides the input time-series data into fixed-length chunks and transforms each chunk into an embedding vector via a linear projection. The *STFT embedding* module applies a short-time Fourier transform (STFT) to the input signal to extract frequency-domain features. If necessary, the frequency dimension is compressed through a linear projection, and each time frame is then converted into an embedding vector. The *Linear+STFT embedding* module integrates the features produced by the linear and STFT embeddings.

Table 8 summarizes the results of the embedding module search. For the GRU model, the best performance was obtained with the Linear+STFT embedding (0.7689). The LSTM model achieved the highest score with the STFT embedding (0.7398). Finally, the Transformer model performed best with the STFT embedding (0.7794). Based on these results, we conducted subsequent experiments using the embedding module that achieved the highest performance for each model.

D.4. Search for a Optimal Architecture

To identify optimal architectural hyperparameters for RNN- and Transformer-based PCG classifiers, we performed a hyperparameter search on the MR task of BMD-HS. For the RNNs, we searched embedding dimensions $d_{\text{emb}} \in \{2^k \mid k = 1, \dots, 15\}$, patch lengths $\{25, 50, 100, 250, 500\}$, RNN depth $\{1, \dots, 8\}$, and hidden sizes $d_{\text{hid}} \in \{2^k \mid k = 1, \dots, 15\}$. For the Transformers, we searched $d_{\text{emb}} \in \{2^k \mid k = 1, \dots, 12\}$, patch lengths $\{25, 50, 100, 250, 500\}$, number of heads $h \in \{2^k \mid k = 1, \dots, 12\}$, and depth $\{1, \dots, 8\}$. The feed-forward dimension was set to $4\, d_{\text{emb}}$. Each model was tuned for 12 hours, and the configuration with the lowest validation loss was selected.

As for GRU, the selected parameters were 512 for d_{emb}, 250 for length of patches, 5 for depth of the RNN layer, and 1024 for the hidden size. For LSTM, the selected parameters were 256 for d_{emb}, 250 for length of patches, 2 for depth of the RNN layer, and 2048 for the hidden size. For Transformer, the selected parameters were 128 for d_{emb}, 25 for length of patches, 2 for number of heads, 3 for depth. We used these settings for the main experiment to evaluate the efficacy of synthesized data.

Appendix E. Performance on Synthesized Data

To evaluate the efficacy of synthetic PCG data in training DNNs, eight distinct architectures—including EfficientNet variants, ResNets, recurrent models (GRU, LSTM), and Transformer—were trained and tested exclusively on synthesized PCG data representing AS, AR, and MR. Performance was quantified using the area under the receiver operating characteristic curve (AUROC), with results summarized in Table 9. The experiment aimed to assess baseline classification capabilities on synthetic data, independent of fine-tuning on real-world samples.

The results reveal substantial variation across models and pathologies. Recurrent networks (GRU,

[15] We did not test the no-embedding setting for Transformer models, since in this case the raw signal length must be treated as the sequence length, which incurs high computational cost.

Table 8: Result of embedding module search.

	None	Linear	STFT	Linear+STFT
GRU	0.5250	0.5905	0.7376	0.7689
LSTM	0.4770	0.5691	0.7398	0.5718
Transformer	-	0.5314	0.7794	0.7163

Table 9: Classification results for model trained and evaluated with synthesized data (AUROC)

	AS	AR	MR
EfficientNet-B0	1.0000 ± 0.0000	0.5134 ± 0.0068	1.0000 ± 0.0000
EfficientNet-B1	0.9018 ± 0.1965	0.5115 ± 0.0077	0.9025 ± 0.1949
GRU	1.0000 ± 0.0000	1.0000 ± 0.0000	0.9998 ± 0.0003
LSTM	0.9999 ± 0.0001	0.9999 ± 0.0001	0.9999 ± 0.0000
ResNet18	0.9999 ± 0.0001	0.9993 ± 0.0005	0.9999 ± 0.0001
ResNet34	1.0000 ± 0.0000	0.9996 ± 0.0003	1.0000 ± 0.0001
ResNet50	1.0000 ± 0.0000	0.9054 ± 0.2113	0.9999 ± 0.0001
Transformer	0.9976 ± 0.0033	0.7104 ± 0.2341	0.9993 ± 0.0004

LSTM) achieved nearly perfect AUROC scores across all classes (≥ 0.999), indicating that temporal modeling can be highly effective in recognizing synthetic PCG patterns. ResNet18 and ResNet34 also demonstrated excellent performance (AUROC ≥ 0.999 for most classes), confirming the strength of convolutional architectures. In contrast, EfficientNet variants showed inconsistent results: EfficientNet-B0 classified AS and MR perfectly but failed on AR (0.5134), while EfficientNet-B1 exhibited moderate yet unstable performance with large variance across classes. The Transformer model performed well for AS and MR (≈ 0.998 and 0.999) but only moderately for AR (0.7104 ± 0.2341), suggesting class-specific challenges.

Appendix F. Detailed Results of PCG Classification

F.1. BMD-HS Dataset

F.1.1. MR Classification

The experimental results demonstrate significant performance improvements when models are pre-trained on synthetic PCG data and fine-tuned on limited real-world data (Syn→Real), compared to training exclusively on real-world data. As shown in Tables 10, 11 and 12, the "Syn→Real" approach yielded average relative gains of 17.1% in AUROC, 73.4% in $F1$-score, and 20.7% in AUPRC across eight architectures. Convolutional architectures exhibited the most consistent improvements, with EfficientNet-B1 achieving a remarkable 426.8% $F1$-score gain and ResNet variants showing $22-29\%$AUROC improvements. In contrast, recurrent (GRU, LSTM) and Transformer displayed marginal or negative gains (-22.1% to -1.71 AUROC). Notably, models trained solely on synthetic data ("Syn") underperformed real-data baselines (AUROC: 51.6% vs 59.8%), emphasizing the necessity of real-world fine-tuning. The substantial performance boost in EfficientNet and ResNet architectures – particularly with limited real training samples ($n = 568$) – supports the hypothesis that synthetic data effectively compensates for medical data scarcity. These findings align with the theoretical framework presented in the abstract, demonstrating that hybrid synthetic-real training can enhance cardiac sound classification while mitigating dataset size constraints inherent to clinical settings.

In addition to AUROC, $F1$-score, and AUPRC, we report clinically important metrics—sensitivity (Table 13), specificity (Table 14), positive predictive value (PPV) (Table 15), and negative predictive value (NPV) (Table 16)—to address the critical concerns of false positives and false negatives in medical diagnostics. These metrics are vital for clinical safety,

Table 10: MR PCG Classification Result (BMD-HS / AUROC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.5230 ± 0.1177	0.5368 ± 0.0426	0.7388 ± 0.0389	41.262
EfficientNet-B1	0.4825 ± 0.0354	0.5095 ± 0.0649	0.7297 ± 0.1242	51.233
GRU	0.7405 ± 0.0273	0.5975 ± 0.0406	0.5770 ± 0.1042	−22.080
LSTM	0.7291 ± 0.0268	0.5551 ± 0.0854	0.7166 ± 0.0118	−1.714
ResNet18	0.5686 ± 0.1055	0.5000 ± 0.0000	0.6945 ± 0.0338	22.142
ResNet34	0.5841 ± 0.1085	0.5066 ± 0.0148	0.7333 ± 0.0521	25.544
ResNet50	0.5479 ± 0.0920	0.5027 ± 0.0040	0.7086 ± 0.0563	29.330
Transformer	0.6787 ± 0.1112	0.5050 ± 0.0525	0.6182 ± 0.0539	−8.914
Average	0.6068	0.5267	0.6896	17.100

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 11: MR PCG Classification Result (BMD-HS / $F1$-score)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.3418 ± 0.3152	0.5333 ± 0.0000	0.5998 ± 0.0333	75.483
EfficientNet-B1	0.1127 ± 0.2355	0.4880 ± 0.1015	0.5937 ± 0.0753	426.797
GRU	0.5922 ± 0.0411	0.5277 ± 0.0052	0.3475 ± 0.2971	−41.321
LSTM	0.6075 ± 0.0498	0.4852 ± 0.0553	0.6151 ± 0.0208	1.251
ResNet18	0.5384 ± 0.0438	0.5333 ± 0.0000	0.5546 ± 0.0321	3.009
ResNet34	0.4030 ± 0.2554	0.5333 ± 0.0000	0.5810 ± 0.0482	44.169
ResNet50	0.3101 ± 0.2924	0.5333 ± 0.0000	0.5791 ± 0.0486	86.746
Transformer	0.5650 ± 0.0839	0.5239 ± 0.0171	0.5154 ± 0.0339	−8.779
Average	0.4338	0.5197	0.5483	73.419

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

as a model with low sensitivity risks missing true cases (false negatives), and low specificity risks overdiagnosis (false positives). From Table 13 and Table 14, the "Syn→Real" setup achieves a over 30.0% average gain in both sensitivity and specificity, indicating fewer missed diagnoses and fewer spurious alerts compared to using real data alone. Similarly, Table 15 and Table 16 show a 60.1% increase in PPV and a 32.4% increase in NPV, further demonstrating that incorporating synthetic data can strengthen diagnostic reliability and improve overall clinical utility. Furthermore, Figure 8 presents confusion matrices for the model architectures that achieved the highest AUROC in MR classification using the BMD-HS dataset under both "Real" and "Syn→Real" conditions.

F.1.2. AS Classification

The "Syn→Real" training paradigm also improved AS classification, though with more variable gains across architectures compared to MR results. As shown in Tables 17, 18 and 19, the approach achieved

Table 12: MR PCG Classification Result (BMD-HS / AUPRC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.3889 ± 0.0842	0.3933 ± 0.0486	0.6127 ± 0.1565	57.547
EfficientNet-B1	0.3650 ± 0.0079	0.4480 ± 0.1026	0.5440 ± 0.1176	49.041
GRU	0.6280 ± 0.0452	0.4230 ± 0.0234	0.3953 ± 0.0763	−37.054
LSTM	0.5968 ± 0.0469	0.4147 ± 0.0528	0.5745 ± 0.0508	−3.737
ResNet18	0.4349 ± 0.0975	0.3683 ± 0.0102	0.6061 ± 0.0501	39.365
ResNet34	0.4273 ± 0.0891	0.3897 ± 0.0241	0.5922 ± 0.1511	38.591
ResNet50	0.4257 ± 0.0739	0.3475 ± 0.0324	0.5521 ± 0.1514	29.692
Transformer	0.5353 ± 0.0941	0.3852 ± 0.0384	0.4934 ± 0.0673	−7.827
Average	0.4752	0.3962	0.5463	20.702

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 13: MR PCG Classification Result (BMD-HS / Sensitivity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.5656 ± 0.4661	1.0000 ± 0.0000	0.7719 ± 0.0364	36.474
EfficientNet-B1	0.2031 ± 0.3985	0.8594 ± 0.2812	0.6656 ± 0.1354	227.720
GRU	0.7094 ± 0.1260	0.9656 ± 0.0319	0.5031 ± 0.4246	−29.081
LSTM	0.7625 ± 0.1200	0.7312 ± 0.1604	0.7219 ± 0.0588	−5.325
ResNet18	0.8687 ± 0.1611	1.0000 ± 0.0000	0.6062 ± 0.0702	−30.218
ResNet34	0.6156 ± 0.4176	1.0000 ± 0.0000	0.5969 ± 0.0864	−3.038
ResNet50	0.4562 ± 0.4185	1.0000 ± 0.0000	0.6531 ± 0.0965	43.161
Transformer	0.6719 ± 0.0719	0.9687 ± 0.0407	0.6719 ± 0.0943	0.000
Average	0.6066	0.9406	0.6488	29.962

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

average relative improvements of 15.3% in AUROC, 42.9% in $F1$-score, and 19.4% in AUPRC. Convolutional networks again demonstrated superior synthetic data utilization, with EfficientNet-B1 showing 189.9% $F1$-score improvement and EfficientNet-B1 achieving 70.7% AUPRC gain. ResNet variants exhibited consistent but more modest enhancements (6.7 − 18.6% $F1$ gains), while recurrent architectures and Transformer showed mixed results - GRU improved AUROC by 12.2% but Transformer gained only 2.4%. The extreme gains in EfficientNet models (B0: 96.3% $F1$, B1: 70.7% AUPRC) with minimal real-world data ($n = 568$) reinforce the architectural sensitivity observed in MR analysis. While average "Syn→Real" improvements were lower than MR benchmarks (42.9% vs 73.4% F1 gain), the maintained superiority over Real-only baselines ($F1$: 54.8% vs 43.4%) confirms synthetic data's cross-pathology utility. These findings further substantiate our hypothesis that synthetic-to-real training effectiveness depends on both model architecture and target condition characteristics.

Table 14: MR PCG Classification Result (BMD-HS / Specificity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	568	0	568	-
EfficientNet-B0	0.5143 ± 0.4481	0.0000 ± 0.0000	0.5411 ± 0.0373	5.211
EfficientNet-B1	0.7982 ± 0.3991	0.1268 ± 0.2536	0.6625 ± 0.2002	-17.001
GRU	0.6089 ± 0.1663	0.0321 ± 0.0369	0.5946 ± 0.3334	-2.349
LSTM	0.5786 ± 0.1141	0.2750 ± 0.2232	0.6429 ± 0.0696	11.113
ResNet18	0.2125 ± 0.2761	0.0000 ± 0.0000	0.6714 ± 0.0652	215.953
ResNet34	0.4768 ± 0.4158	0.0000 ± 0.0000	0.7429 ± 0.0566	55.810
ResNet50	0.6161 ± 0.3742	0.0000 ± 0.0000	0.6607 ± 0.0772	7.239
Transformer	0.5804 ± 0.1706	0.0125 ± 0.0091	0.4643 ± 0.1570	-20.003
Average	0.5482	0.0558	0.6225	31.997

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 15: MR PCG Classification Result (BMD-HS / PPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	568	0	568	-
EfficientNet-B0	0.2504 ± 0.2128	0.3636 ± 0.0000	0.4907 ± 0.0277	95.966
EfficientNet-B1	0.1727 ± 0.2159	0.3542 ± 0.0188	0.5642 ± 0.1108	226.694
GRU	0.5248 ± 0.0579	0.3632 ± 0.0026	0.2947 ± 0.1876	-43.845
LSTM	0.5136 ± 0.0363	0.3734 ± 0.0458	0.5396 ± 0.0320	5.062
ResNet18	0.4054 ± 0.0773	0.3636 ± 0.0000	0.5167 ± 0.0315	27.454
ResNet34	0.3392 ± 0.1790	0.3636 ± 0.0000	0.5732 ± 0.0407	68.986
ResNet50	0.2458 ± 0.2040	0.3636 ± 0.0000	0.5266 ± 0.0332	114.239
Transformer	0.4960 ± 0.0913	0.3591 ± 0.0088	0.4256 ± 0.0439	-14.194
Average	0.3685	0.3630	0.4914	60.045

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

In medical diagnostics, false positives and negatives must be carefully tracked, so in addition to AUROC, $F1$-score, and AUPRC, we report clinically relevant metrics—namely sensitivity (Table 20), specificity (Table 21), PPV (Table 22), and NPV (Table 23)—to ensure clinical safety by assessing the likelihood of missed or incorrect diagnoses. Across eight models evaluated under the "Syn→Real" setting, we observe on average an 26.1% gain in sensitivity, a 15.1% gain in specificity, a 47.2% gain in PPV, and a 33.6% gain in NPV relative to training on real data alone. Furthermore, Figure 9 shows confusion matrices for the top-performing architectures under both "Real" and "Syn→Real" conditions when classifying AS from the BMD-HS dataset, reinforcing the improved clinical efficacy of these methods.

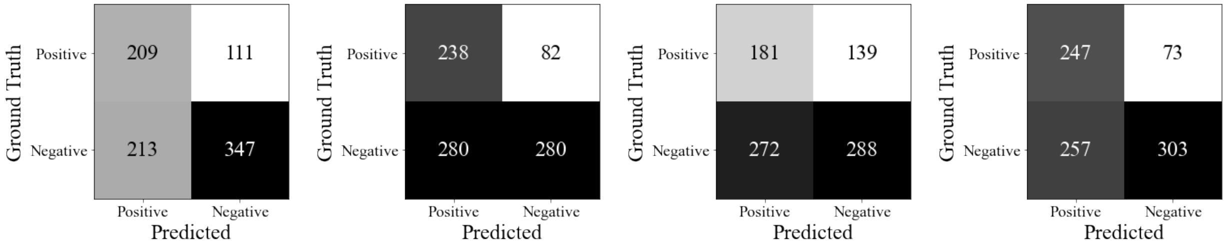
F.1.3. AR Classification

The "Syn→Real" strategy demonstrated measurable but more modest improvements for AR classification compared to MR and AS outcomes, revealing condition-specific performance patterns. As shown

Table 16: MR PCG Classification Result (BMD-HS / NPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	568	0	568	-
EfficientNet-B0	0.4252 ± 0.3561	0.0000 ± 0.0000	0.8055 ± 0.0325	89.440
EfficientNet-B1	0.5094 ± 0.2547	0.1224 ± 0.2448	0.7744 ± 0.0821	52.022
GRU	0.7927 ± 0.0358	0.3273 ± 0.2915	0.7737 ± 0.1525	-2.397
LSTM	0.8189 ± 0.0447	0.5906 ± 0.1419	0.8036 ± 0.0210	-1.868
ResNet18	0.6863 ± 0.3683	0.0000 ± 0.0000	0.7504 ± 0.0209	9.340
ResNet34	0.4306 ± 0.3601	0.0000 ± 0.0000	0.7658 ± 0.0332	77.845
ResNet50	0.5554 ± 0.2892	0.0000 ± 0.0000	0.7729 ± 0.0349	39.161
Transformer	0.7391 ± 0.1029	0.5600 ± 0.3929	0.7059 ± 0.0329	-4.492
Average	0.6197	0.2000	0.769	32.381

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.



(a) GRU ("Real") (b) GRU ("Syn→Real") (c) EfficientNet-B0 ("Real") (d) EfficientNet-B0 ("Syn→Real")

Figure 8: Confusion matrices for the model architectures (GRU and EfficientNet-B0) that achieved the highest AUROC in MR classification using the BMD-HS dataset under "Real" and "Syn→Real" conditions, respectively. The values represent the sum of results across five independent trials on the test set.

in Tables 24, 25 and 26, the approach yielded average gains of 17.0% AUROC, 8.2% $F1$-score, and 13.3% AUPRC – the lowest relative improvements across all three pathologies. Convolutional architectures maintained their synthetic data compatibility, with EfficientNet-B0 achieving standout 53.9% AUPRC and 42.5% AUROC gains, while ResNet34 showed consistent $15 - 27\%$ improvements across three metrics. Recurrent models exhibited unexpected variance, with LSTM gaining 25.5% $F1$-score but GRU declining -5.8%. Attention-based architectures again underperformed, particularly Transformers showing -2.0% AUPRC degradation. These results complete our tri-pathology analysis, demonstrating that while synthetic pretraining consistently enhances PCG classification, its effectiveness scales with both architectural compatibility and pathological signature distinctness – critical considerations for clinical implementation.

In addition to AUROC, $F1$-score, and AUPRC, we report performance using clinically relevant metrics—sensitivity (Table 27), specificity (Table 28), PPV (Table 29), and NPV (Table 30)—since these measures are critical in medical diagnostics for characterizing false positive/negative rates and ensuring patient safety. From the averages in Table 27–Table 30, the "Syn→Real" approach yields a 2.3% decrease in sensitivity, a 22.4% increase in specificity,

Table 17: AS PCG Classification Result (BMD-HS / AUROC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.6454 ± 0.1578	0.5281 ± 0.0435	0.8380 ± 0.0097	29.842
EfficientNet-B1	0.5351 ± 0.1477	0.5421 ± 0.0602	0.8177 ± 0.0218	52.813
GRU	0.7648 ± 0.0525	0.4460 ± 0.0324	0.8296 ± 0.0178	8.473
LSTM	0.7434 ± 0.0743	0.4388 ± 0.0478	0.7937 ± 0.0494	6.766
ResNet18	0.7165 ± 0.0351	0.5000 ± 0.0000	0.7857 ± 0.0074	9.658
ResNet34	0.6904 ± 0.1511	0.5000 ± 0.0000	0.7833 ± 0.0217	13.456
ResNet50	0.7730 ± 0.0644	0.4932 ± 0.0050	0.7950 ± 0.0149	2.846
Transformer	0.6785 ± 0.0281	0.6287 ± 0.0482	0.6699 ± 0.0556	-1.268
Average	0.6934	0.5096	0.7891	15.323

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 18: AS PCG Classification Result (BMD-HS / $F1$-score)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.3212 ± 0.3016	0.4286 ± 0.0000	0.6305 ± 0.0437	96.295
EfficientNet-B1	0.2082 ± 0.2925	0.4240 ± 0.0102	0.6035 ± 0.0556	189.865
GRU	0.5236 ± 0.0405	0.4361 ± 0.0091	0.5875 ± 0.0241	12.204
LSTM	0.5394 ± 0.0722	0.4321 ± 0.0086	0.5713 ± 0.0537	5.914
ResNet18	0.5056 ± 0.0707	0.4286 ± 0.0000	0.5996 ± 0.0365	18.592
ResNet34	0.5249 ± 0.0891	0.4286 ± 0.0000	0.5844 ± 0.0153	11.335
ResNet50	0.5444 ± 0.0651	0.4286 ± 0.0000	0.5809 ± 0.0271	6.705
Transformer	0.4636 ± 0.0188	0.4329 ± 0.0058	0.4748 ± 0.0476	2.416
Average	0.4539	0.4299	0.5791	42.916

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

a 17.5% increase in PPV, and a 35.9% increase in NPV relative to the "Real" baseline, indicating that training with synthetic data can substantially reduce false positives (and thus improve specificity, PPV, and NPV) while slightly lowering sensitivity. Furthermore, Figure 10 presents confusion matrices for the top-performing architectures (by AUROC) under both "Real" and "Syn→Real" training, highlighting the clinical impact of these additional metrics and the potential for synthetic data to enhance classification robustness in a patient-safe manner.

F.2. Private Dataset

In this section, we present the results of the three binary classification tasks distinguishing Aortic Stenosis (AS), Aortic Regurgitation (AR), and Mitral Regurgitation (MR). Detailed numerical results, including per-class metrics and aggregated scores, are reported in the following subsections.

Table 19: AS PCG Classification Result (BMD-HS / AUPRC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.4594 ± 0.2343	0.2992 ± 0.0360	0.7312 ± 0.0349	59.164
EfficientNet-B1	0.3522 ± 0.1807	0.3178 ± 0.0490	0.6011 ± 0.1268	70.670
GRU	0.6483 ± 0.0781	0.2493 ± 0.0122	0.6963 ± 0.0393	7.404
LSTM	0.6209 ± 0.0858	0.2546 ± 0.0346	0.6390 ± 0.1164	2.915
ResNet18	0.5600 ± 0.0851	0.2669 ± 0.0068	0.6147 ± 0.0447	9.768
ResNet34	0.5231 ± 0.2015	0.2585 ± 0.0166	0.5809 ± 0.1005	11.050
ResNet50	0.5884 ± 0.1717	0.2548 ± 0.0223	0.6222 ± 0.0642	5.744
Transformer	0.5408 ± 0.0346	0.4169 ± 0.0865	0.4769 ± 0.0666	−11.816
Average	0.5366	0.2898	0.6203	19.362

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 20: AS PCG Classification Result (BMD-HS / Sensitivity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.5250 ± 0.4349	1.0000 ± 0.0000	0.8125 ± 0.0475	54.762
EfficientNet-B1	0.3417 ± 0.4285	0.9750 ± 0.0500	0.8042 ± 0.0386	135.353
GRU	0.7208 ± 0.0741	1.0000 ± 0.0000	0.8042 ± 0.0408	11.570
LSTM	0.7000 ± 0.0974	0.9792 ± 0.0132	0.7375 ± 0.1322	5.357
ResNet18	0.7875 ± 0.1740	1.0000 ± 0.0000	0.7750 ± 0.0677	−1.587
ResNet34	0.8458 ± 0.1288	1.0000 ± 0.0000	0.7875 ± 0.0534	−6.893
ResNet50	0.7542 ± 0.1346	1.0000 ± 0.0000	0.7417 ± 0.0429	−1.657
Transformer	0.6958 ± 0.1219	1.0000 ± 0.0000	0.7792 ± 0.0937	11.986
Average	0.6713	0.9943	0.7802	26.111

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

F.2.1. MR Classification

The experimental results demonstrate significant performance improvements when models are pre-trained on synthetic PCG data and fine-tuned on limited real-world data (Syn→Real), compared to training exclusively on real-world data. As shown in Tables 31, 32 and 33, across eight tested DNN architectures, the "Syn→Real" approach yielded average relative gains of 6.08% in AUROC, 29.19% in $F1$-score, and 22.94% in AUPRC. Notably, the EfficientNet-B0 and B1 models exhibited the most significant performance increases, with $F1$-score gains exceeding 60% in some cases. Similar to the trends observed across BMD-HS dataset, these results reinforce the overall benefit of the "Syn→Real" training strategy for enhancing the performance of medical DNNs.

In addition to AUROC, $F1$-score, and AUPRC, we report performance using clinically relevant metrics, including sensitivity(Table 34), specificity (Table 35), PPV (Table 36), and NPV (Table 37). Overall, with the "Syn→Real" setting, we observed a 16.3% increase in sensitivity, a 5.06% increase in specificity,

Table 21: AS PCG Classification Result (BMD-HS / Specificity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.6547 ± 0.3715	0.0000 ± 0.0000	0.7062 ± 0.1005	7.866
EfficientNet-B1	0.7547 ± 0.3874	0.0172 ± 0.0344	0.6687 ± 0.1071	−11.395
GRU	0.6109 ± 0.0869	0.0297 ± 0.0318	0.6484 ± 0.0557	6.138
LSTM	0.6594 ± 0.1098	0.0422 ± 0.0291	0.6875 ± 0.0989	4.261
ResNet18	0.4484 ± 0.3665	0.0000 ± 0.0000	0.6922 ± 0.0915	54.371
ResNet34	0.4234 ± 0.3488	0.0000 ± 0.0000	0.6594 ± 0.0481	55.739
ResNet50	0.5828 ± 0.2961	0.0000 ± 0.0000	0.6922 ± 0.0712	18.771
Transformer	0.5156 ± 0.1217	0.0172 ± 0.0206	0.4375 ± 0.0676	−15.147
Average	0.5812	0.0133	0.649	15.076

[*] The relative improvement from “Real” to “Syn→Real”, calculated as $(c - a)/a \times 100$.

[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 22: AS PCG Classification Result (BMD-HS / PPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.2414 ± 0.2131	0.2727 ± 0.0000	0.5219 ± 0.0661	116.197
EfficientNet-B1	0.1625 ± 0.2162	0.2710 ± 0.0034	0.4877 ± 0.0630	200.123
GRU	0.4145 ± 0.0370	0.2789 ± 0.0067	0.4645 ± 0.0307	12.063
LSTM	0.4458 ± 0.0711	0.2772 ± 0.0060	0.4785 ± 0.0466	7.335
ResNet18	0.4032 ± 0.1072	0.2727 ± 0.0000	0.4961 ± 0.0571	23.041
ResNet34	0.4037 ± 0.1107	0.2727 ± 0.0000	0.4667 ± 0.0251	15.606
ResNet50	0.4496 ± 0.0934	0.2727 ± 0.0000	0.4828 ± 0.0559	7.384
Transformer	0.3565 ± 0.0296	0.2762 ± 0.0043	0.3425 ± 0.0316	−3.927
Average	0.3597	0.2743	0.4676	47.228

[*] The relative improvement from “Real” to “Syn→Real”, calculated as $(c - a)/a \times 100$.

[†] The total number of synthesized and real-world data used for training the model, respectively.

a 29.5% increase in PPV, and a 0.42% increase in NPV across the eight models evaluated. Furthermore, Figure 11 presents confusion matrices for the model architectures that achieved the highest AUROC in MR classification using the private dataset under both “Real” and “Syn→Real” conditions.

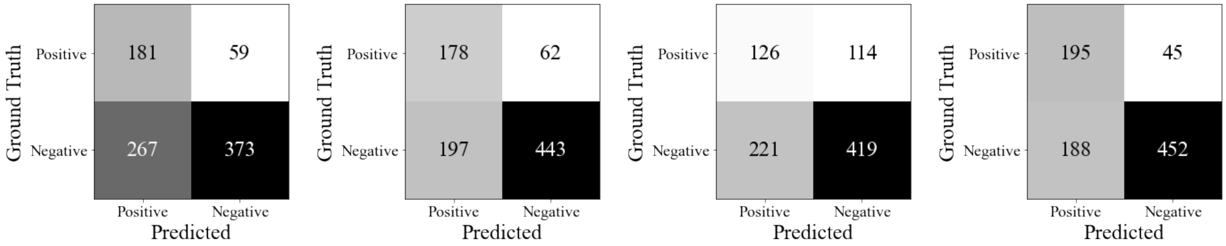
F.2.2. AS Classification

The experimental results demonstrate significant performance improvements when models are pre-trained on synthetic PCG data and fine-tuned on limited real-world data (Syn→Real), compared to training exclusively on real-world data. As shown in Tables 38, 39 and 40, the “Syn→Real” approach yielded average relative gains of 7.11% in AUROC, 16.95% in $F1$-score, and 22.11% in AUPRC across eight architectures. Specifically, for AUROC, the “Syn→Real” method achieved the highest gain of 14.46% with the Transformer architecture, improving from 0.5818 to 0.6659. For all eight architecture evaluated, we observed improvement with “Syn→Real” over “Real” setting with improvements ranging from 0.99% to

Table 23: AS PCG Classification Result (BMD-HS / NPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.6515 ± 0.3349	0.0000 ± 0.0000	0.9106 ± 0.0112	39.770
EfficientNet-B1	0.6116 ± 0.3112	0.1294 ± 0.2588	0.9000 ± 0.0162	47.155
GRU	0.8544 ± 0.0250	0.6000 ± 0.4899	0.8992 ± 0.0149	5.243
LSTM	0.8548 ± 0.0352	0.6498 ± 0.3434	0.8802 ± 0.0361	2.971
ResNet18	0.5095 ± 0.4160	0.0000 ± 0.0000	0.8929 ± 0.0181	75.250
ResNet34	0.5282 ± 0.4313	0.0000 ± 0.0000	0.8936 ± 0.0183	69.178
ResNet50	0.6919 ± 0.3462	0.0000 ± 0.0000	0.8779 ± 0.0080	26.883
Transformer	0.8239 ± 0.0215	0.6000 ± 0.4899	0.8451 ± 0.0517	2.573
Average	0.6907	0.2474	0.8874	33.628

[*] The relative improvement from “Real” to “Syn→Real”, calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.



(a) ResNet50 (“Real”) (b) ResNet50 (“Syn→Real”) (c) EfficientNet-B0 (“Real”) (d) EfficientNet-B0 (“Syn→Real”)

Figure 9: Confusion matrices for the model architectures (ResNet50 and EfficientNet-B0) that achieved the highest AUROC in AS classification using the BMD-HS dataset under “Real” and “Syn→Real” conditions, respectively. The values represent the sum of results across five independent trials on the test set.

14.46%. For AUPRC, we observed decrease ranging from 3.8% to 5.3% with ResNet family. For $F1$-score, the largest gain of 33.18% was observed with the Transformer architecture, increasing from 0.1103 to 0.1469. Although some individual architectures experienced negative gains, the average across all architectures demonstrates the effectiveness of the synthetic pretraining and real-world fine-tuning strategy.

In addition to AUROC, $F1$-score, and AUPRC, we report performance using clinically relevant metrics, including sensitivity(Table 41), specificity (Table 42), PPV (Table 43), and NPV (Table 44). Overall, with the “Syn→Real” setting, we observed a 8.86% increase in sensitivity, a 5.02% increase in specificity, a 18.4% increase in PPV, and a 0.37% increase in NPV across the eight models evaluated. Furthermore, Figure 12 presents confusion matrices for the model architectures that achieved the highest AUROC in AS classification using the private dataset under both “Real” and “Syn→Real” conditions.

F.2.3. AR Classification

The experimental results demonstrate significant performance improvements when models are pre-trained on synthetic PCG data and fine-tuned on limited real-world data (Syn→Real), compared to training exclusively on real-world data. As shown in Tables 45, 46 and 47, the “Syn→Real” approach

Table 24: AR PCG Classification Result (BMD-HS / AUROC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	568	0	568	-
EfficientNet-B0	0.5545 ± 0.0402	0.4818 ± 0.1413	0.7903 ± 0.0500	42.525
EfficientNet-B1	0.5813 ± 0.1164	0.4628 ± 0.0711	0.7428 ± 0.0364	27.783
GRU	0.6706 ± 0.1306	0.5889 ± 0.0224	0.6956 ± 0.0537	3.728
LSTM	0.6026 ± 0.0854	0.5833 ± 0.0502	0.7682 ± 0.0630	27.481
ResNet18	0.6573 ± 0.1774	0.4209 ± 0.0200	0.7193 ± 0.1641	9.432
ResNet34	0.6968 ± 0.1268	0.4201 ± 0.0419	0.8105 ± 0.0346	16.317
ResNet50	0.7393 ± 0.1626	0.4957 ± 0.0445	0.7924 ± 0.1045	7.183
Transformer	0.5667 ± 0.0611	0.5329 ± 0.0759	0.5737 ± 0.0930	1.235
Average	0.6336	0.4983	0.7366	16.960

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 25: AR PCG Classification Result (BMD-HS / $F1$-score)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	568	0	568	-
EfficientNet-B0	0.5333 ± 0.0000	0.4102 ± 0.0664	0.6022 ± 0.1001	12.920
EfficientNet-B1	0.5639 ± 0.0683	0.4051 ± 0.0981	0.5467 ± 0.0724	-3.050
GRU	0.4955 ± 0.1110	0.5277 ± 0.0635	0.4669 ± 0.2655	-5.772
LSTM	0.4626 ± 0.0624	0.4638 ± 0.0644	0.5804 ± 0.1438	25.465
ResNet18	0.4957 ± 0.2902	0.5082 ± 0.0194	0.5540 ± 0.1663	11.761
ResNet34	0.5020 ± 0.2913	0.5023 ± 0.0510	0.6359 ± 0.0364	26.673
ResNet50	0.5730 ± 0.1628	0.5243 ± 0.0196	0.5397 ± 0.3037	-5.811
Transformer	0.4657 ± 0.0458	0.4931 ± 0.0775	0.4804 ± 0.0606	3.156
Average	0.5115	0.4793	0.5508	8.168

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

yielded average relative gains of 8.75% in AUROC, 23.87% in $F1$-score, and 27.43% in AUPRC across eight architectures. For AUROC, the largest improvement was observed with ResNet18, achieving a 15.32% gain, increasing from 0.5673 to 0.6542. While EfficientNet-B0 showed a decrease in AUROC, others, like EfficientNet-B1 and LSTM, demonstrated substantial improvements. In terms of $F1$-score, the "Syn→Real" approach showed more consistent improvements, with the largest gain of 66.85% seen in EfficientNet-B1, rising from 0.0893 to 0.1490. ResNet18, ResNet34, GRU, and LSTM also showed considerable gains. For AUPRC, we observed LSTM exhibiting the largest improvement of 60.81%, increasing from 0.0694 to 0.1116. Despite some minor decreases in performance for specific architectures, the overall average gains across all metrics highlight the benefits of the "Syn→Real" training strategy.

In addition to AUROC, $F1$-score, and AUPRC, we report performance using clinically relevant metrics, including sensitivity(Table 48), specificity (Table 49), PPV (Table 50), and NPV (Table 51). Overall, with

Table 26: AR PCG Classification Result (BMD-HS / AUPRC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.4274 ± 0.0606	0.3947 ± 0.1038	0.6578 ± 0.0758	53.907
EfficientNet-B1	0.4556 ± 0.0919	0.3776 ± 0.0878	0.5503 ± 0.0950	20.786
GRU	0.5880 ± 0.1491	0.4143 ± 0.0083	0.4855 ± 0.1398	-17.432
LSTM	0.5005 ± 0.0968	0.4294 ± 0.0426	0.6293 ± 0.0553	25.734
ResNet18	0.5536 ± 0.1940	0.3274 ± 0.0131	0.5905 ± 0.1569	6.665
ResNet34	0.5808 ± 0.1387	0.3251 ± 0.0307	0.6684 ± 0.0543	15.083
ResNet50	0.6274 ± 0.1782	0.3911 ± 0.0570	0.6488 ± 0.1014	3.411
Transformer	0.5097 ± 0.0621	0.4049 ± 0.0548	0.4993 ± 0.0201	-2.040
Average	0.5304	0.3831	0.5912	13.264

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 27: AR PCG Classification Result (BMD-HS / Sensitivity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	1.0000 ± 0.0000	0.5031 ± 0.1831	0.6719 ± 0.1893	-32.810
EfficientNet-B1	0.9469 ± 0.1062	0.5062 ± 0.2393	0.6219 ± 0.2278	-34.322
GRU	0.4844 ± 0.1234	0.8187 ± 0.2024	0.6937 ± 0.3531	43.208
LSTM	0.4687 ± 0.0677	0.5969 ± 0.2638	0.5875 ± 0.1813	25.347
ResNet18	0.6812 ± 0.3657	0.9062 ± 0.0778	0.5969 ± 0.2116	-12.375
ResNet34	0.6625 ± 0.3444	0.9000 ± 0.1474	0.6594 ± 0.0835	-0.468
ResNet50	0.6281 ± 0.2570	0.9500 ± 0.0250	0.5437 ± 0.2813	-13.437
Transformer	0.5781 ± 0.2181	0.8094 ± 0.2308	0.6156 ± 0.1783	6.487
Average	0.6812	0.7488	0.6238	-2.296

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

the "Syn→Real" setting, we observed a 29.5% increase in sensitivity, a 1.79% decrease in specificity, a 19.6% increase in PPV, and a 0.76% increase in NPV across the eight models evaluated. Furthermore, Figure 13 presents confusion matrices for the model architectures that achieved the highest AUROC in AR classification using the private dataset under both "Real" and "Syn→Real" conditions.

Appendix G. Additional Cross-Dataset and Synthetic Pretraining Experiments

G.1. Experimental Setup (overview)

We assessed whether synthetic data pre-training can substitute for, or complement, pre-training on a different real dataset. We evaluated two real datasets (BMD-HS and Private) and three targets (AS, AR, MR). All classification task conditions (data splits,

Table 28: AR PCG Classification Result (BMD-HS / Specificity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	568	0	568	-
EfficientNet-B0	0.0000 ± 0.0000	0.4946 ± 0.2600	0.7018 ± 0.1488	inf
EfficientNet-B1	0.1536 ± 0.3071	0.4786 ± 0.2671	0.6625 ± 0.2087	331.315
GRU	0.7304 ± 0.1644	0.2929 ± 0.2144	0.4786 ± 0.3258	-34.474
LSTM	0.6750 ± 0.1440	0.4964 ± 0.3063	0.7821 ± 0.0925	15.867
ResNet18	0.5339 ± 0.4402	0.0536 ± 0.0855	0.7107 ± 0.0790	33.115
ResNet34	0.5964 ± 0.3395	0.0536 ± 0.0898	0.7661 ± 0.0671	28.454
ResNet50	0.6929 ± 0.3472	0.0429 ± 0.0376	0.8500 ± 0.0809	22.673
Transformer	0.5125 ± 0.2802	0.2000 ± 0.2498	0.4839 ± 0.2045	-5.580
Average	0.4868	0.2641	0.6795	22.137

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 29: AR PCG Classification Result (BMD-HS / PPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	568	0	568	-
EfficientNet-B0	0.3636 ± 0.0000	0.3871 ± 0.0618	0.5793 ± 0.0718	59.323
EfficientNet-B1	0.4197 ± 0.1121	0.3740 ± 0.1027	0.5487 ± 0.0773	30.736
GRU	0.5509 ± 0.1598	0.4024 ± 0.0245	0.3600 ± 0.1946	-34.652
LSTM	0.4747 ± 0.0968	0.4263 ± 0.0431	0.6114 ± 0.0594	28.797
ResNet18	0.4291 ± 0.2645	0.3539 ± 0.0106	0.5338 ± 0.1196	24.400
ResNet34	0.4271 ± 0.2480	0.3498 ± 0.0195	0.6255 ± 0.0558	46.453
ResNet50	0.6145 ± 0.1360	0.3621 ± 0.0136	0.5411 ± 0.2708	-11.945
Transformer	0.4310 ± 0.0725	0.3696 ± 0.0282	0.4159 ± 0.0416	-3.503
Average	0.4638	0.3782	0.527	17.451

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

preprocessing, optimization settings, and evaluation protocol) are identical to those described in the main text. Models include CNNs (ResNet-18/34/50, EfficientNet-B0/B1) and sequence models (GRU, LSTM, Transformer).

We compare five training schemas:

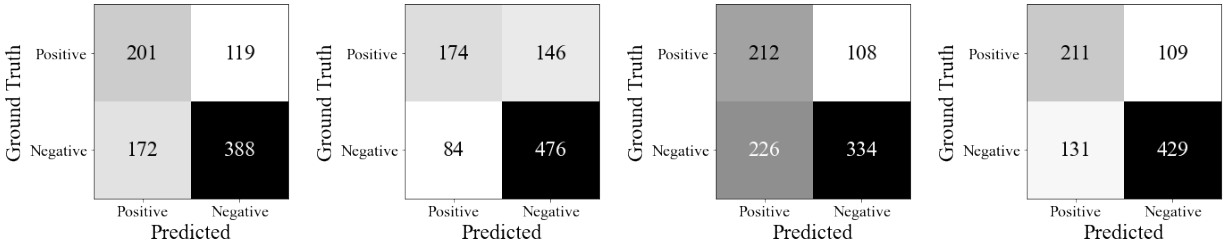
- **Real**: train on the target real-world dataset only (as in the main text).
- **Syn**:train on the synthetic dataset only (as in the main text).
- **Syn→Real**: pretrain on synthetic data, then fine-tune on the target real dataset (as in the main text).
- **Real→Real**: pretrain on the other real dataset and fine-tune on the target real dataset.
- **Syn→Real→Real**: pretrain on synthetic data, fine-tune on BMD-HS, then fine-tune again on Private (or vice versa, depending on the target).

Table 30: AR PCG Classification Result (BMD-HS / NPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	568	0	568	-
EfficientNet-B0	0.0000 ± 0.0000	0.6085 ± 0.1046	0.8061 ± 0.0670	*inf*
EfficientNet-B1	0.1670 ± 0.3340	0.5810 ± 0.1030	0.7857 ± 0.0879	370.479
GRU	0.7097 ± 0.0518	0.7695 ± 0.0580	0.7846 ± 0.0799	10.554
LSTM	0.6847 ± 0.0389	0.6872 ± 0.0254	0.7760 ± 0.0565	13.334
ResNet18	0.4598 ± 0.3821	0.2353 ± 0.2358	0.7673 ± 0.0928	66.877
ResNet34	0.7978 ± 0.0892	0.2420 ± 0.2268	0.7999 ± 0.0298	0.263
ResNet50	0.6188 ± 0.3154	0.5079 ± 0.2701	0.7822 ± 0.0782	26.406
Transformer	0.5444 ± 0.2727	0.4888 ± 0.3461	0.6851 ± 0.0488	25.845
Average	0.4978	0.5150	0.7734	35.943

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.

[†] The total number of synthesized and real-world data used for training the model, respectively.



(*a*) ResNet50 ("Real") (*b*) ResNet50 ("Syn→Real") (*c*) ResNet34 ("Real") (*d*) ResNet34 ("Syn→Real")

Figure 10: Confusion matrices for the model architectures (ResNet50 and ResNet34) that achieved the highest AUROC in AR classification using the BMD-HS dataset under "Real" and "Syn→Real" conditions, respectively. The values represent the sum of results across five independent trials on the test set.

Unless noted otherwise, we report mean AUROC (± standard deviation in Tables 52, 53, 54, 55, 56 and 57) across repeated runs.

G.2. Implications

The implications from the results on Table 2 are:

1. **Cross-dataset transfer helps.** "Real→Real" consistently improves over training solely on the target dataset ("Real"), indicating that representations learned from a related real dataset transfer effectively.

2. **Synthetic pretraining is competitive with real-source pretraining.** "Syn→Real" performs on par with "Real→Real" overall: slightly worse on BMD-HS (AS/AR), comparable or better on BMD-HS (MR) and on the Private dataset across AS/AR/MR.

3. **Sequential fine-tuning is best.** "Syn→Real→Real" yields the strongest results across all conditions, showing that synthetic pretraining and multi-stage fine-tuning on multiple real datasets are complementary.

Collectively, these findings show that synthetic data provides a strong and scalable foundation for transfer learning, comparable to leveraging real data from a different source, and that combining synthetic

Table 31: MR PCG Classification Result (Private / AUROC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	$5,459$	0	$5,459$	-
EfficientNet-B0	0.5655 ± 0.0692	0.5249 ± 0.0182	0.6835 ± 0.0133	20.866
EfficientNet-B1	0.5612 ± 0.0605	0.5365 ± 0.0091	0.6748 ± 0.0539	20.242
GRU	0.6093 ± 0.0283	0.5330 ± 0.0169	0.5931 ± 0.0557	-2.659
LSTM	0.5900 ± 0.0273	0.5119 ± 0.0119	0.5599 ± 0.0372	-5.102
ResNet18	0.5793 ± 0.0197	0.5172 ± 0.0054	0.5984 ± 0.0433	3.297
ResNet34	0.5888 ± 0.0169	0.5221 ± 0.0153	0.6483 ± 0.0199	10.105
ResNet50	0.6110 ± 0.0175	0.5152 ± 0.0163	0.6247 ± 0.0386	2.242
Transformer	0.5492 ± 0.0363	0.4775 ± 0.0156	0.5472 ± 0.0074	-0.364
Average	0.5818	0.5173	0.6162	6.079

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 32: MR PCG Classification Result (Private / $F1$-score)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	$5,459$	0	$5,459$	-
EfficientNet-B0	0.1550 ± 0.0871	0.1643 ± 0.0050	0.2575 ± 0.0130	66.129
EfficientNet-B1	0.1011 ± 0.0964	0.1535 ± 0.0424	0.2410 ± 0.0341	138.378
GRU	0.1798 ± 0.0357	0.1633 ± 0.0016	0.1840 ± 0.0252	2.336
LSTM	0.1810 ± 0.0141	0.1633 ± 0.0037	0.1621 ± 0.0216	-10.442
ResNet18	0.1768 ± 0.0080	0.1609 ± 0.0037	0.1888 ± 0.0200	6.787
ResNet34	0.1785 ± 0.0069	0.1627 ± 0.0022	0.2135 ± 0.0102	19.608
ResNet50	0.1901 ± 0.0141	0.1627 ± 0.0021	0.2110 ± 0.0271	10.994
Transformer	0.1634 ± 0.0123	0.1625 ± 0.0034	0.1630 ± 0.0056	-0.245
Average	0.1657	0.1617	0.2026	29.193

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

pre-training with multi-stage real fine-tuning maximizes performance.

G.3. Detailed Results

We report detailed values for "Real→Real" and "Syn→Real→Real" settings. For BMD-HS dataset, results for MR, AS, and AR are shown in Tables 52, 53 and 54 respectively. For Private dataset, results for MR, AS, and AR are shown in Tables 55, 56 and 57, respectively. The results for "Real", "Syn", and "Syn→Real" settings are from Tables 10, 17, 24, 31, 38 and 45 respectively.

Table 33: MR PCG Classification Result (Private / AUPRC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.1098 ± 0.0188	0.0899 ± 0.0091	0.1990 ± 0.0112	81.239
EfficientNet-B1	0.1127 ± 0.0309	0.0952 ± 0.0113	0.1981 ± 0.0539	75.776
GRU	0.1193 ± 0.0201	0.0932 ± 0.0046	0.1206 ± 0.0266	1.090
LSTM	0.1132 ± 0.0080	0.0957 ± 0.0080	0.1067 ± 0.0098	-5.742
ResNet18	0.1139 ± 0.0054	0.0817 ± 0.0047	0.1132 ± 0.0179	-0.615
ResNet34	0.1184 ± 0.0078	0.0806 ± 0.0049	0.1505 ± 0.0206	27.111
ResNet50	0.1196 ± 0.0108	0.0838 ± 0.0080	0.1263 ± 0.0101	5.602
Transformer	0.1060 ± 0.0111	0.0831 ± 0.0082	0.1050 ± 0.0023	-0.943
Average	0.1141	0.0879	0.1399	22.940

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 34: MR PCG Classification Result (Private / Sensitivity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.3608 ± 0.1873	0.9000 ± 0.0650	0.4595 ± 0.0866	27.356
EfficientNet-B1	0.2216 ± 0.2403	0.6865 ± 0.3177	0.5095 ± 0.0614	129.919
GRU	0.5243 ± 0.1843	0.9608 ± 0.0051	0.4703 ± 0.0802	-10.299
LSTM	0.5797 ± 0.1428	0.8919 ± 0.0487	0.4135 ± 0.0865	-28.670
ResNet18	0.5068 ± 0.1195	0.9446 ± 0.0156	0.5162 ± 0.0809	1.855
ResNet34	0.5000 ± 0.1855	0.9270 ± 0.0207	0.5514 ± 0.1171	10.280
ResNet50	0.5405 ± 0.1523	0.9095 ± 0.0379	0.5527 ± 0.0594	2.257
Transformer	0.5419 ± 0.1084	0.9486 ± 0.0199	0.5297 ± 0.0909	-2.251
Average	0.472	0.8961	0.5004	16.306

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Appendix H. Ablation: Causal vs. Non-causal Self-Attention

We investigated the impact of self-attention directionality (non-causal vs. causal) on the performance of the Transformer used in this study.

Configuration. In the causal variant, a strict lower-triangular mask was applied to the self-attention in each layer, allowing each time step t to attend only to tokens at $\leq t$. The architecture, preprocessing, data splits, optimization procedures (including learning rate schedules and batch sizes), and windowing strategy were kept identical to the non-causal variant. Evaluation was conducted on two datasets (BMD-HS and Private) across three targets (MR, AS, AR), and AUROC (mean ± standard deviation) was reported. Three training and evaluation conditions were examined: "Real", where training and evaluation were performed on real data; "Syn", where both training and evaluation used synthetic data; and "Syn$rightarrow$Real", where training was

Table 35: MR PCG Classification Result (Private / Specificity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.7584 ± 0.1244	0.1421 ± 0.0809	0.8014 ± 0.0469	5.670
EfficientNet-B1	0.8471 ± 0.1675	0.3875 ± 0.2763	0.7331 ± 0.0895	−13.458
GRU	0.6090 ± 0.1338	0.0721 ± 0.0136	0.6541 ± 0.0656	7.406
LSTM	0.5436 ± 0.1314	0.1454 ± 0.0515	0.6561 ± 0.0500	20.695
ResNet18	0.6010 ± 0.1100	0.0726 ± 0.0239	0.6257 ± 0.0635	4.110
ResNet34	0.6118 ± 0.1786	0.1038 ± 0.0209	0.6586 ± 0.0851	7.650
ResNet50	0.6060 ± 0.1423	0.1224 ± 0.0496	0.6465 ± 0.0605	6.683
Transformer	0.5228 ± 0.0860	0.0795 ± 0.0183	0.5320 ± 0.0818	1.760
Average	0.6375	0.1407	0.6634	5.064

[*] The relative improvement from “Real” to “Syn→Real”, calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 36: MR PCG Classification Result (Private / PPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.0990 ± 0.0496	0.0905 ± 0.0029	0.1816 ± 0.0126	83.434
EfficientNet-B1	0.0735 ± 0.0601	0.0934 ± 0.0059	0.1600 ± 0.0258	117.687
GRU	0.1112 ± 0.0150	0.0893 ± 0.0009	0.1151 ± 0.0144	3.507
LSTM	0.1084 ± 0.0085	0.0899 ± 0.0020	0.1013 ± 0.0098	−6.550
ResNet18	0.1092 ± 0.0085	0.0879 ± 0.0019	0.1161 ± 0.0117	6.319
ResNet34	0.1118 ± 0.0084	0.0892 ± 0.0011	0.1341 ± 0.0095	19.946
ResNet50	0.1177 ± 0.0111	0.0894 ± 0.0014	0.1310 ± 0.0180	11.300
Transformer	0.0968 ± 0.0049	0.0889 ± 0.0017	0.0968 ± 0.0011	0.000
Average	0.1034	0.0898	0.1295	29.456

[*] The relative improvement from “Real” to “Syn→Real”, calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

conducted on synthetic data and evaluation on real data.

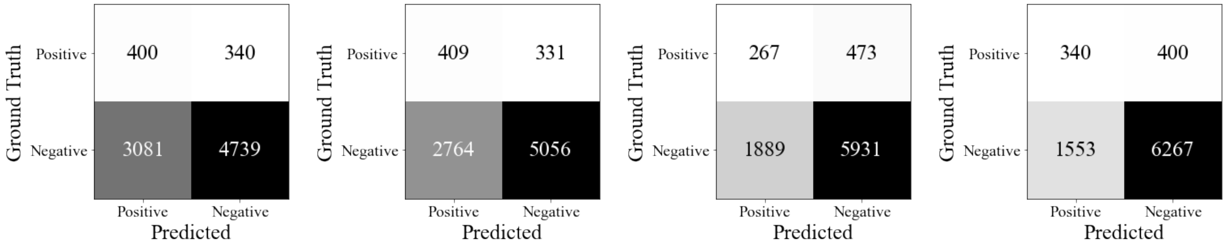
Results. The results are summarized in Table 58. Under the “Real” condition, differences between the two variants were minimal, with AUROC differences less than 0.01 across all tasks and datasets. In the “Syn*rightarrow*Real” condition, MR showed consistent improvement (BMD-HS: +0.055, Private: +0.024), AS exhibited dataset-dependent behavior (−0.019 for BMD-HS, +0.012 for Private), and AR demonstrated small and inconsistent changes (+0.020 for BMD-HS, −0.008 for Private). Under the “Syn” condition, notable improvements were observed for MR (approximately +0.10 on both datasets), while degradation was seen for AS on BMD-HS.

Table 37: MR PCG Classification Result (Private / NPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.9274 ± 0.0078	0.9364 ± 0.0122	0.9404 ± 0.0061	1.402
EfficientNet-B1	0.9221 ± 0.0103	0.9413 ± 0.0190	0.9402 ± 0.0054	1.963
GRU	0.9332 ± 0.0099	0.9503 ± 0.0056	0.9287 ± 0.0087	−0.482
LSTM	0.9336 ± 0.0092	0.9361 ± 0.0124	0.9223 ± 0.0058	−1.210
ResNet18	0.9286 ± 0.0054	0.9282 ± 0.0246	0.9320 ± 0.0069	0.366
ResNet34	0.9311 ± 0.0103	0.9379 ± 0.0095	0.9406 ± 0.0093	1.020
ResNet50	0.9347 ± 0.0085	0.9355 ± 0.0060	0.9384 ± 0.0072	0.396
Transformer	0.9241 ± 0.0070	0.9416 ± 0.0171	0.9232 ± 0.0028	−0.097
Average	0.9294	0.9384	0.9332	0.420

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.

[†] The total number of synthesized and real-world data used for training the model, respectively.



(a) ResNet50 ("Real") (b) ResNet50 ("Syn→Real") (c) EfficientNet-B0 ("Real") (d) EfficientNet-B0 ("Syn→Real")

Figure 11: Confusion matrices for the model architectures (ResNet50 and EfficientNet-B0) that achieved the highest AUROC in MR classification using the private dataset under "Real" and "Syn→Real" conditions, respectively. The values represent the sum of results across five independent trials on the test set.

Appendix I. Details of the Human Evaluation

I.1. Auditory Quality Assessment

The human evaluation results, shown in Figure 14, demonstrate nuanced perceptual characteristics of synthetic PCG signals across cardiac conditions. While real-world normal heart sounds received significantly higher naturalness ratings (4.44 vs 3.58 for synthetic), this pattern reversed in pathological cases: synthetic samples for AS, AR, and MR achieved marginally higher naturalness scores (AS: 3.96 vs 3.62; AR: 3.98 vs 3.67; MR: 3.84 vs 3.64) than their real counterparts. This paradoxical improvement in perceived quality for synthetic pathological signals may reflect either enhanced acoustic regularity in synthesized murmurs or potential confounding from clinical expectations of "cleaner" pathological signatures. The binary classification results corroborate this pattern, with synthetic AS signals proving most challenging to distinguish (45% discrimination accuracy, approaching chance levels), followed by AR and MR ($54 - 55\%$ accuracy). Notably, normal heart sounds showed higher discriminability (61% accuracy), suggesting synthetic normals retain subtle artifacts perceptible to experts. Across all conditions, substantial false positive rates ($34-48\%$ of synthetic samples misclassified as real) indicate clinically

Table 38: AS PCG Classification Result (Private / AUROC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.7711 ± 0.1731	0.6850 ± 0.0204	0.8761 ± 0.0092	13.617
EfficientNet-B1	0.7939 ± 0.1622	0.6682 ± 0.0987	0.8816 ± 0.0179	11.047
GRU	0.8357 ± 0.1176	0.5382 ± 0.0195	0.8917 ± 0.0168	6.701
LSTM	0.8139 ± 0.0927	0.5247 ± 0.0274	0.8570 ± 0.0215	5.295
ResNet18	0.7844 ± 0.0150	0.5847 ± 0.0146	0.7922 ± 0.0133	0.994
ResNet34	0.8145 ± 0.0084	0.6021 ± 0.0232	0.8359 ± 0.0133	2.627
ResNet50	0.7907 ± 0.0248	0.6055 ± 0.0260	0.8078 ± 0.0202	2.163
Transformer	0.5818 ± 0.0283	0.5978 ± 0.0295	0.6659 ± 0.0309	14.455
Average	0.7732	0.6008	0.826	7.112

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.

[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 39: AS PCG Classification Result (Private / $F1$-score)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.2910 ± 0.1638	0.1200 ± 0.0131	0.3606 ± 0.0502	23.918
EfficientNet-B1	0.3035 ± 0.1717	0.1141 ± 0.0111	0.3492 ± 0.0328	15.058
GRU	0.2718 ± 0.0946	0.1022 ± 0.0034	0.3273 ± 0.0514	20.419
LSTM	0.2373 ± 0.0674	0.1055 ± 0.0040	0.2851 ± 0.0341	20.143
ResNet18	0.2526 ± 0.0510	0.1004 ± 0.0014	0.2466 ± 0.0444	−2.375
ResNet34	0.2660 ± 0.0300	0.1060 ± 0.0055	0.3165 ± 0.0265	18.985
ResNet50	0.2763 ± 0.0590	0.1047 ± 0.0046	0.2936 ± 0.0478	6.261
Transformer	0.1103 ± 0.0116	0.1020 ± 0.0056	0.1469 ± 0.0135	33.182
Average	0.2511	0.1069	0.2907	16.949

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.

[†] The total number of synthesized and real-world data used for training the model, respectively.

meaningful realism. These findings collectively suggest that while synthetic generation of normal heart sounds requires refinement, the approach achieves particular success in replicating pathological acoustics – a critical advancement given the diagnostic importance of abnormal murmurs in cardiac assessment.

I.2. Evaluation with Inter-Rater Agreement

The main text reported aggregate totals across all 90 evaluations (Figure 3), which is useful for a compact overview. In this appendix we report confusion matrix with inter-rater agreement. Here, for each item, we use the majority label of three raters as a predicted label. The results are shown in Figure 15. The trend of easy identification in Normal and difficult identification in MR is consistent with the results presented in the main body.

Table 40: AS PCG Classification Result (Private / AUPRC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	$5,459$	0	$5,459$	-
EfficientNet-B0	0.2784 ± 0.1297	0.0808 ± 0.0069	0.3640 ± 0.0299	30.747
EfficientNet-B1	0.2794 ± 0.1254	0.0919 ± 0.0264	0.3699 ± 0.0402	32.391
GRU	0.2973 ± 0.1203	0.0566 ± 0.0012	0.3975 ± 0.0363	33.703
LSTM	0.2855 ± 0.1177	0.0502 ± 0.0035	0.3278 ± 0.0552	14.816
ResNet18	0.3277 ± 0.0221	0.0639 ± 0.0027	0.3108 ± 0.0196	-5.157
ResNet34	0.3651 ± 0.0180	0.0685 ± 0.0057	0.3512 ± 0.0141	-3.807
ResNet50	0.3234 ± 0.0466	0.0652 ± 0.0044	0.3064 ± 0.0331	-5.257
Transformer	0.0680 ± 0.0095	0.0867 ± 0.0151	0.1220 ± 0.0402	79.412
Average	0.2781	0.0705	0.3187	22.106

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 41: AS PCG Classification Result (Private / Sensitivity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	$5,459$	0	$5,459$	-
EfficientNet-B0	0.4729 ± 0.2373	0.9059 ± 0.0357	0.6518 ± 0.0274	37.830
EfficientNet-B1	0.5200 ± 0.2624	0.9365 ± 0.0456	0.7035 ± 0.0417	35.288
GRU	0.7529 ± 0.0640	0.8847 ± 0.0336	0.7341 ± 0.0524	-2.497
LSTM	0.7412 ± 0.0414	0.8729 ± 0.0410	0.6941 ± 0.0595	-6.355
ResNet18	0.6306 ± 0.0540	0.9435 ± 0.0137	0.6094 ± 0.0753	-3.362
ResNet34	0.6588 ± 0.0288	0.9318 ± 0.0319	0.6824 ± 0.0258	3.582
ResNet50	0.6071 ± 0.0425	0.9224 ± 0.0253	0.6282 ± 0.0231	3.475
Transformer	0.4800 ± 0.0606	0.8212 ± 0.0368	0.4941 ± 0.0941	2.938
Average	0.6079	0.9024	0.6497	8.863

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

I.3. Visual Assessment through Spectral Analysis

To further examine the quality of synthesized PCG signals beyond auditory assessment, we conducted visual comparisons between the generated and real PCG samples using time-frequency analysis methods. We employed two widely-used signal processing techniques: Continuous Wavelet Transform (CWT) and Short-Time Fourier Transform (STFT). These methods allowed us to visualize the time-frequency characteristics of both real and synthesized PCG signals, enabling detailed comparison of their spectral content and temporal patterns. The examples of visualized data is shown in Figure 16. By examining the spectrograms and scalograms produced by STFT and CWT respectively, we could evaluate how well the synthesized PCG signals captured the essential frequency components and temporal dynamics of real heart sounds, including S1 and S2 characteristics and their transitions.

The time-frequency analysis evaluation showed consistent patterns across all cardiac conditions, with

Table 42: AS PCG Classification Result (Private / Specificity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.9305 ± 0.0350	0.3003 ± 0.1041	0.8942 ± 0.0269	-3.901
EfficientNet-B1	0.9248 ± 0.0410	0.2333 ± 0.1164	0.8772 ± 0.0189	-5.147
GRU	0.7544 ± 0.1548	0.1934 ± 0.0431	0.8505 ± 0.0414	12.739
LSTM	0.7281 ± 0.1474	0.2309 ± 0.0636	0.8315 ± 0.0321	14.201
ResNet18	0.8140 ± 0.0596	0.1192 ± 0.0224	0.8152 ± 0.0628	0.147
ResNet34	0.8247 ± 0.0319	0.1791 ± 0.0673	0.8615 ± 0.0168	4.462
ResNet50	0.8467 ± 0.0443	0.1778 ± 0.0599	0.8575 ± 0.0296	1.276
Transformer	0.6224 ± 0.0446	0.2525 ± 0.0514	0.7246 ± 0.0651	16.420
Average	0.8057	0.2108	0.839	5.025

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 43: AS PCG Classification Result (Private / PPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.2103 ± 0.1061	0.0643 ± 0.0069	0.2513 ± 0.0414	19.496
EfficientNet-B1	0.2160 ± 0.1115	0.0608 ± 0.0058	0.2331 ± 0.0249	7.917
GRU	0.1704 ± 0.0610	0.0543 ± 0.0017	0.2135 ± 0.0419	25.293
LSTM	0.1434 ± 0.0406	0.0561 ± 0.0022	0.1804 ± 0.0242	25.802
ResNet18	0.1609 ± 0.0397	0.0530 ± 0.0007	0.1597 ± 0.0427	-0.746
ResNet34	0.1676 ± 0.0226	0.0562 ± 0.0029	0.2065 ± 0.0199	23.210
ResNet50	0.1817 ± 0.0461	0.0555 ± 0.0024	0.1928 ± 0.0358	6.109
Transformer	0.0624 ± 0.0060	0.0544 ± 0.0028	0.0872 ± 0.0098	39.744
Average	0.1641	0.0568	0.1906	18.353

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

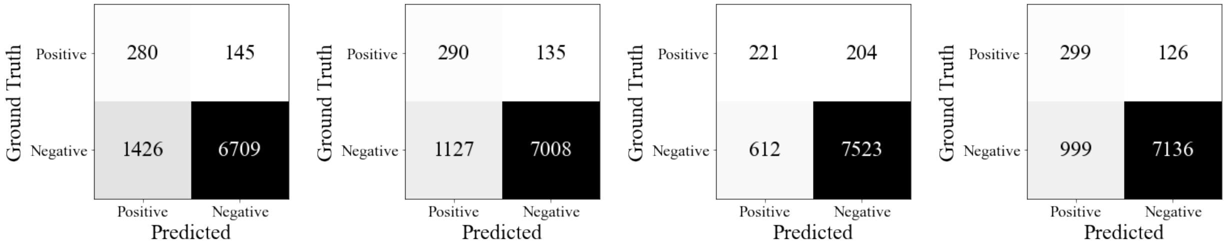
evaluators generally demonstrating higher accuracy in identifying real PCG signals compared to the auditory assessment. For normal heart sounds, 43 out of 45 real-world samples were correctly identified, while 15 synthetic samples were classified as real-world data, indicating that visual features made the distinction more apparent than auditory cues alone. Similar patterns were observed across pathological conditions, with high accuracy in real-world sample identification (MR: 42/45, AR: 40/45, AS: 43/45) and relatively consistent rates of synthetic samples being classified as real (MR: 12, AR: 12, AS: 13). This contrasts with the auditory evaluation where AS samples showed the lowest discriminative ability - in the visual assessment, experts maintained high accuracy across all conditions. These results suggest that while our synthetic PCG signals achieve considerable acoustic similarity to real samples, their time-frequency characteristics reveal more distinguishable patterns, particularly when examined through CWT and STFT visualizations.

Table 44: AS PCG Classification Result (Private / NPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.9718 ± 0.0108	0.9835 ± 0.0036	0.9800 ± 0.0016	0.844
EfficientNet-B1	0.9743 ± 0.0121	0.9867 ± 0.0039	0.9827 ± 0.0023	0.862
GRU	0.9818 ± 0.0083	0.9698 ± 0.0056	0.9840 ± 0.0026	0.224
LSTM	0.9811 ± 0.0043	0.9722 ± 0.0030	0.9812 ± 0.0032	0.010
ResNet18	0.9769 ± 0.0022	0.9759 ± 0.0027	0.9758 ± 0.0027	−0.113
ResNet34	0.9789 ± 0.0013	0.9809 ± 0.0031	0.9811 ± 0.0015	0.225
ResNet50	0.9763 ± 0.0023	0.9775 ± 0.0014	0.9778 ± 0.0017	0.154
Transformer	0.9582 ± 0.0033	0.9638 ± 0.0064	0.9651 ± 0.0037	0.720
Average	0.9749	0.9763	0.9785	0.366

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.

[†] The total number of synthesized and real-world data used for training the model, respectively.



(a) ResNet34 ("Real") (b) ResNet34 ("Syn→Real") (c) EfficientNet-B1 ("Real") (d) EfficientNet-B1 ("Syn→Real")

Figure 12: Confusion matrices for the model architectures (ResNet34 and EfficientNet-B1) that achieved the highest AUROC in AS classification using the private dataset under "Real" and "Syn→Real" conditions, respectively. The values represent the sum of results across five independent trials on the test set.

Appendix J. Quantitative Similarity Between Real and Synthetic PCG Data

Data and preprocessing. To quantify the similarity between real and synthetic PCG signals, we assembled 1,500 recordings in total. The real cohort comprised 1,000 recordings: 500 randomly sampled from the BMD-HS dataset and 500 from our private dataset. The synthetic cohort comprised 500 recordings, with 125 samples each generated for Normal, aortic stenosis (AS), aortic regurgitation (AR), and mitral regurgitation (MR) conditions. All signals were resampled to 4,000Hz and trimmed to a fixed 10-s duration prior to analysis.

Feature extraction. From every recording we extracted 10 scalar features intended to capture both temporal and spectral characteristics of heart sounds. Time-domain features (7) were: mean, standard deviation, skewness, kurtosis, maximum, minimum, and root-mean-square (RMS). Frequency-domain features (3) were: spectral centroid, spectral spread, and spectral entropy.

Distributional comparison. For each feature dimension (e.g., mean, standard deviation), we approximated the empirical marginal distribution of feature

Table 45: AR PCG Classification Result (Private / AUROC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.5490 ± 0.0600	0.4731 ± 0.0412	0.5352 ± 0.0582	−2.514
EfficientNet-B1	0.5613 ± 0.0251	0.4958 ± 0.0266	0.6200 ± 0.0432	10.458
GRU	0.5978 ± 0.0490	0.5105 ± 0.0062	0.6598 ± 0.0088	10.371
LSTM	0.5638 ± 0.0231	0.5047 ± 0.0145	0.6453 ± 0.0408	14.456
ResNet18	0.5673 ± 0.0537	0.5599 ± 0.0121	0.6542 ± 0.0369	15.318
ResNet34	0.5754 ± 0.0597	0.5502 ± 0.0215	0.6509 ± 0.0199	13.121
ResNet50	0.5736 ± 0.0500	0.5505 ± 0.0335	0.5884 ± 0.0621	2.580
Transformer	0.5448 ± 0.0140	0.4995 ± 0.0486	0.5787 ± 0.0218	6.223
Average	0.5666	0.5180	0.6166	8.752

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 46: AR PCG Classification Result (Private / $F1$-score)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.1009 ± 0.0495	0.0882 ± 0.0519	0.0846 ± 0.0559	−16.155
EfficientNet-B1	0.0893 ± 0.0548	0.0860 ± 0.0427	0.1490 ± 0.0237	66.853
GRU	0.1213 ± 0.0415	0.1158 ± 0.0049	0.1593 ± 0.0075	31.327
LSTM	0.1233 ± 0.0135	0.1158 ± 0.0070	0.1619 ± 0.0337	31.306
ResNet18	0.1214 ± 0.0232	0.1190 ± 0.0019	0.1621 ± 0.0145	33.526
ResNet34	0.1245 ± 0.0333	0.1202 ± 0.0019	0.1650 ± 0.0146	32.530
ResNet50	0.1279 ± 0.0329	0.1179 ± 0.0050	0.1240 ± 0.0442	−3.049
Transformer	0.1096 ± 0.0134	0.1062 ± 0.0218	0.1256 ± 0.0204	14.598
Average	0.1148	0.1086	0.1414	23.867

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

values within each dataset (BMD-HS, Private, Synthetic) using histograms, and then computed pairwise distances between the corresponding histograms.

Four complementary metrics were used:

- Total variation distance (TVD): measures the separation between two probability distributions (range 0–1; equivalently, one half of the L1 distance between the normalized histograms).
- Kullback–Leibler (KL) divergence: quantifies the information loss when one distribution is used to approximate another.
- L1 norm: Manhattan distance between the distributions.
- L2 norm: Euclidean distance between the distributions.
- Fréchet Audio Distance (FAD): quantifies the distance between audio distributions by comput-

Table 47: AR PCG Classification Result (Private / AUPRC)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	$5,459$	0	$5,459$	-
EfficientNet-B0	0.0769 ± 0.0158	0.0563 ± 0.0030	0.0757 ± 0.0222	-1.560
EfficientNet-B1	0.0745 ± 0.0073	0.0671 ± 0.0093	0.0956 ± 0.0256	28.322
GRU	0.0909 ± 0.0320	0.0618 ± 0.0009	0.1367 ± 0.0197	50.385
LSTM	0.0694 ± 0.0047	0.0631 ± 0.0022	0.1116 ± 0.0175	60.807
ResNet18	0.0835 ± 0.0258	0.0778 ± 0.0037	0.1017 ± 0.0264	21.796
ResNet34	0.0843 ± 0.0189	0.0768 ± 0.0100	0.1120 ± 0.0090	32.859
ResNet50	0.0765 ± 0.0173	0.0741 ± 0.0090	0.0833 ± 0.0216	8.889
Transformer	0.0673 ± 0.0031	0.0678 ± 0.0119	0.0794 ± 0.0111	17.979
Average	0.0779	0.0681	0.0995	27.435

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 48: AR PCG Classification Result (Private / Sensitivity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	$5,459$	0	$5,459$	-
EfficientNet-B0	0.1794 ± 0.1163	0.4280 ± 0.3253	0.1570 ± 0.1300	-12.486
EfficientNet-B1	0.1720 ± 0.1130	0.5028 ± 0.4197	0.2953 ± 0.0563	71.686
GRU	0.3346 ± 0.1577	0.5290 ± 0.0894	0.4467 ± 0.1339	33.503
LSTM	0.4262 ± 0.0850	0.5944 ± 0.0765	0.4168 ± 0.1206	-2.205
ResNet18	0.3925 ± 0.0731	0.9477 ± 0.0210	0.6280 ± 0.1537	60.000
ResNet34	0.3589 ± 0.2347	0.9290 ± 0.0403	0.5869 ± 0.0520	63.527
ResNet50	0.2841 ± 0.1132	0.8804 ± 0.0765	0.3159 ± 0.1387	11.193
Transformer	0.3664 ± 0.1491	0.5813 ± 0.2523	0.4056 ± 0.1204	10.699
Average	0.3143	0.6741	0.4065	29.490

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.

ing the Fréchet distance between feature embeddings extracted using VGGish (Hershey et al. (2017)).

For each metric we averaged the distances across the 10 feature dimensions to obtain a single summary value per dataset pair.

Results. Table 59 summarizes the average distances between the two real-world datasets and between each real dataset and the synthetic data (lower is closer).

Interpretation. Across three of the four metrics (TVD= 0.44, KL= 1.60, L1= 0.87), the synthesized data are closer to BMD-HS than to the private dataset. L2 distances are of comparable magnitude across all comparisons (0.23–0.35), with the smallest value observed for private vs synthesized dataset (0.23), and FAD was smallest between the private and synthesized dataset. Notably, the distance between the two real datasets (BMD-HS vs Private) is consistently larger than either real-vs-synthesized comparison across all metrics (TVD= 0.66, KL= 3.75,

Table 49: AR PCG Classification Result (Private / Specificity)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	$5,459$	0	$5,459$	-
EfficientNet-B0	0.8684 ± 0.0634	0.5695 ± 0.3197	0.8794 ± 0.0811	1.267
EfficientNet-B1	0.8653 ± 0.0890	0.4876 ± 0.3992	0.8232 ± 0.0229	-4.865
GRU	0.7468 ± 0.1123	0.4958 ± 0.0773	0.7230 ± 0.1036	-3.187
LSTM	0.6344 ± 0.0791	0.4219 ± 0.0741	0.7558 ± 0.0480	19.136
ResNet18	0.6562 ± 0.0723	0.0680 ± 0.0339	0.5892 ± 0.1236	-10.210
ResNet34	0.7408 ± 0.1813	0.0982 ± 0.0407	0.6299 ± 0.0449	-14.970
ResNet50	0.7993 ± 0.0632	0.1312 ± 0.0567	0.7730 ± 0.0777	-3.290
Transformer	0.6602 ± 0.1256	0.4046 ± 0.2063	0.6720 ± 0.0616	1.787
Average	0.7464	0.3346	0.7307	-1.792

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.

[†] The total number of synthesized and real-world data used for training the model, respectively.

Table 50: AR PCG Classification Result (Private / PPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	$10,000$	$10,000$	-
n real-world	$5,459$	0	$5,459$	-
EfficientNet-B0	0.0757 ± 0.0221	0.0507 ± 0.0266	0.0610 ± 0.0328	-19.419
EfficientNet-B1	0.0627 ± 0.0335	0.0528 ± 0.0169	0.1000 ± 0.0133	59.490
GRU	0.0778 ± 0.0168	0.0652 ± 0.0015	0.1014 ± 0.0120	30.334
LSTM	0.0725 ± 0.0071	0.0643 ± 0.0036	0.1011 ± 0.0178	39.448
ResNet18	0.0724 ± 0.0142	0.0635 ± 0.0010	0.0943 ± 0.0094	30.249
ResNet34	0.0881 ± 0.0133	0.0643 ± 0.0009	0.0962 ± 0.0085	9.194
ResNet50	0.0843 ± 0.0166	0.0632 ± 0.0022	0.0794 ± 0.0188	-5.813
Transformer	0.0661 ± 0.0028	0.0590 ± 0.0093	0.0747 ± 0.0088	13.011
Average	0.075	0.0604	0.0885	19.562

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.

[†] The total number of synthesized and real-world data used for training the model, respectively.

L1= 1.33, L2= 0.35, FAD= 17.26). Taken together, these results indicate that the synthetic PCG data reproduce key statistical and spectral characteristics of real PCG signals and fall within the natural variability observed across real-world datasets under the features and metrics considered.

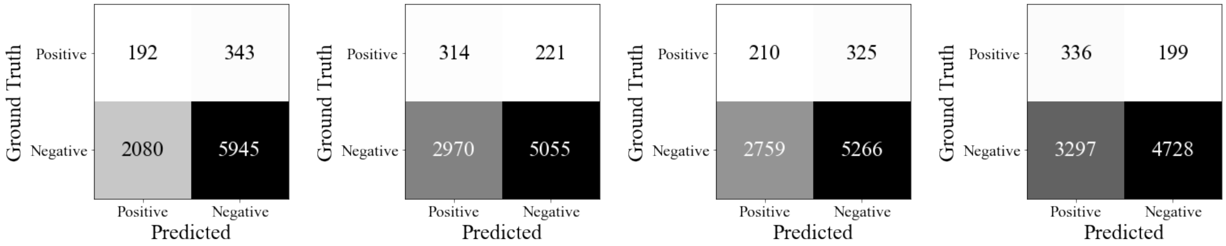
Appendix K. Zero-Shot Transfer Evaluation of Synthetic PCG Data

This appendix evaluates the extent to which synthetic PCG data preserves disease-specific discriminative structures under zero-shot transfer settings, specifically Train-on-Synthetic-Test-on-Real (TSTR) and Train-on-Real-Test-on-Synthetic (TRTS). The objective here is to assess whether synthetic data generalizes directly to clinical data without fine-tuning,

Table 51: AR PCG Classification Result (Private / NPV)

Architecture	*a.* **Real**	*b.* **Syn**	*c.* **Syn→Real**	**Gain (%)**[*]
n function generated[†]	0	10,000	10,000	-
n real-world	5,459	0	5,459	-
EfficientNet-B0	0.9410 ± 0.0046	0.9410 ± 0.0124	0.9403 ± 0.0046	−0.074
EfficientNet-B1	0.9402 ± 0.0026	0.9459 ± 0.0205	0.9461 ± 0.0034	0.627
GRU	0.9444 ± 0.0061	0.9407 ± 0.0021	0.9521 ± 0.0043	0.815
LSTM	0.9431 ± 0.0042	0.9398 ± 0.0051	0.9515 ± 0.0075	0.891
ResNet18	0.9417 ± 0.0058	0.9473 ± 0.0095	0.9613 ± 0.0093	2.081
ResNet34	0.9472 ± 0.0071	0.9550 ± 0.0105	0.9582 ± 0.0036	1.161
ResNet50	0.9440 ± 0.0048	0.9479 ± 0.0187	0.9447 ± 0.0056	0.074
Transformer	0.9404 ± 0.0029	0.9384 ± 0.0086	0.9449 ± 0.0062	0.478
Average	0.9428	0.9445	0.9499	0.757

[*] The relative improvement from "Real" to "Syn→Real", calculated as $(c - a)/a \times 100$.
[†] The total number of synthesized and real-world data used for training the model, respectively.



(*a*) ResNet34 ("Real") (*b*) ResNet34 ("Syn→Real") (*c*) ResNet18 ("Real") (*d*) ResNet18 ("Syn→Real")

Figure 13: Confusion matrices for the model architectures (ResNet34 and ResNet18) that achieved the highest AUROC in AR classification using the private dataset under "Real" and "Syn→Real" conditions, respectively. The values represent the sum of results across five independent trials on the test set.

which represents a distinct evaluation axis from the main findings presented in the manuscript, where synthetic data pre-training followed by real data fine-tuning demonstrated performance improvements.

Preprocessing, Models, and Training Conditions Acoustic preprocessing and input representations were identical to those employed in the downstream classification experiments described in the main manuscript. Eight distinct deep neural network (DNN) architectures, used in main body, were adopted. For each condition, AUROC was computed for each model, and the mean AUROC across the eight models is reported. For **TSTR**, models were trained exclusively on synthetic data (with training and validation performed within the synthetic domain) and directly evaluated on the test set of real data. For **TRTS**, models were trained exclusively on real data (with training and validation performed within the real domain) and directly evaluated on the test set of synthetic data. Critically, neither setting utilized any data, statistics, or hyperparameter tuning information from the target (evaluation) domain. That is, no fine-tuning or domain adaptation was performed, thus evaluating pure zero-shot transfer.

Results The results are shown in Table 60. In summary, AUROC values remained around 0.50 across most conditions, indicating limited discriminative performance under zero-shot transfer. The excep-

Table 52: BMD-HS / MR / AUROC (Extended from Table 10). Mean ± SD across runs.

Model	Real	Syn	Syn→Real	Real→Real	Syn→Real→Real
ResNet18	0.5686 ± 0.1055	0.5000 ± 0.0000	0.6945 ± 0.0338	0.5914 ± 0.1588	0.7012 ± 0.0424
ResNet34	0.5841 ± 0.1085	0.5066 ± 0.0148	0.7333 ± 0.0521	0.6334 ± 0.0751	0.7295 ± 0.0307
ResNet50	0.5479 ± 0.0920	0.5027 ± 0.0040	0.7086 ± 0.0563	0.5249 ± 0.0767	0.6929 ± 0.0429
EffNetB0	0.5230 ± 0.1177	0.5368 ± 0.0426	0.7388 ± 0.0389	0.6469 ± 0.0747	0.7393 ± 0.0263
EffNetB1	0.4825 ± 0.0354	0.5095 ± 0.0649	0.7297 ± 0.1242	0.5759 ± 0.0839	0.7017 ± 0.1140
GRU	0.7405 ± 0.0273	0.5975 ± 0.0406	0.5770 ± 0.1042	0.7539 ± 0.0147	0.7549 ± 0.0437
LSTM	0.7291 ± 0.0268	0.5551 ± 0.0854	0.7166 ± 0.0118	0.7379 ± 0.0119	0.7338 ± 0.0222
Transformer	0.6787 ± 0.1112	0.5050 ± 0.0525	0.6182 ± 0.0539	0.7133 ± 0.0179	0.6392 ± 0.0498
Mean	0.6068	0.5267	0.6896	0.6472	0.7116

Table 53: BMD-HS / AS / AUROC (Extended from Table 17). Mean ± SD across runs.

Model	Real	Syn	Syn→Real	Real→Real	Syn→Real→Real
ResNet18	0.7165 ± 0.0351	0.5000 ± 0.0000	0.7857 ± 0.0074	0.8200 ± 0.0096	0.8284 ± 0.0078
ResNet34	0.6904 ± 0.1511	0.5000 ± 0.0000	0.7833 ± 0.0217	0.8356 ± 0.0099	0.8277 ± 0.0149
ResNet50	0.7730 ± 0.0644	0.4932 ± 0.0050	0.7950 ± 0.0149	0.8293 ± 0.0063	0.8234 ± 0.0130
EffNetB0	0.6454 ± 0.1578	0.5281 ± 0.0435	0.8380 ± 0.0097	0.8508 ± 0.0149	0.8639 ± 0.0120
EffNetB1	0.5351 ± 0.1477	0.5421 ± 0.0602	0.8177 ± 0.0218	0.8439 ± 0.0231	0.8677 ± 0.0135
GRU	0.7648 ± 0.0525	0.4460 ± 0.0324	0.8296 ± 0.0178	0.8307 ± 0.0143	0.8385 ± 0.0107
LSTM	0.7434 ± 0.0743	0.4388 ± 0.0478	0.7937 ± 0.0494	0.8123 ± 0.0139	0.8136 ± 0.0095
Transformer	0.6785 ± 0.0281	0.6287 ± 0.0482	0.6699 ± 0.0556	0.7092 ± 0.0412	0.7294 ± 0.0322
Mean	0.6934	0.5096	0.7891	0.8165	0.8241

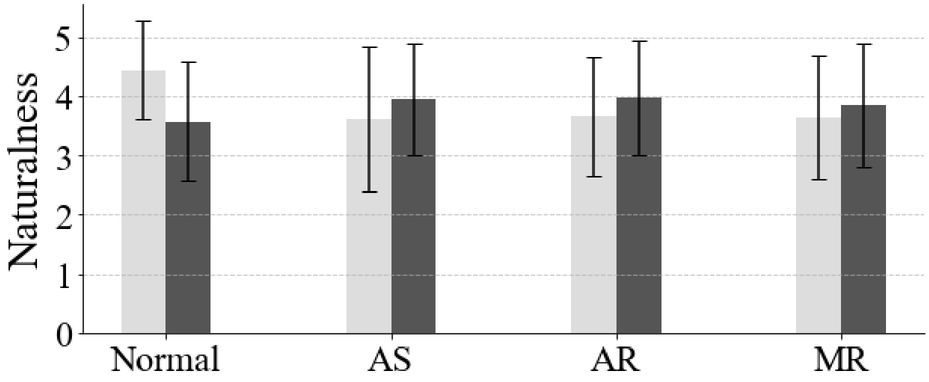


Figure 14: Naturalness of real-world and synthesized data evaluated by human.

tion was AS on Private dataset under TSTR, which showed a modest elevation to 0.6008; however, this level of performance is insufficient for clinical generalization.

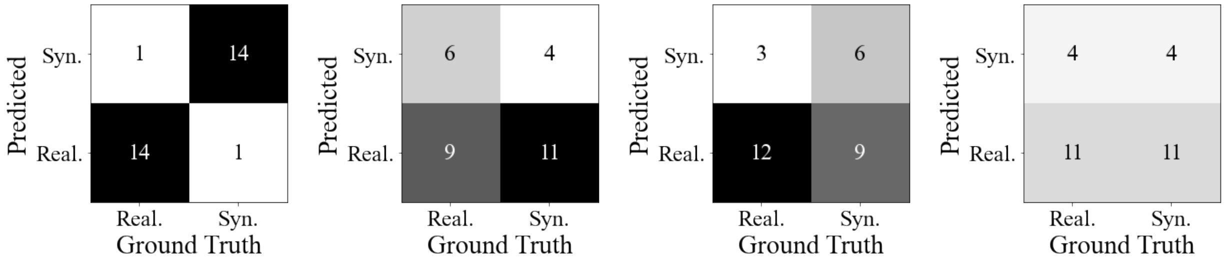
Discussion These results suggest that disease-specific features (e.g., systolic timing of murmurs and envelope morphology) are not sufficiently aligned between synthetic and real domains under zero-shot synthetic-to-real and real-to-synthetic transfer. However, the principal conclusion presented in the main

Table 54: BMD-HS / AR / AUROC (Extended from Table 24). Mean ± SD across runs.

Model	Real	Syn	Syn→Real	Real→Real	Syn→Real→Real
ResNet18	0.6573 ± 0.1774	0.4209 ± 0.0200	0.7193 ± 0.1641	0.8325 ± 0.0282	0.8011 ± 0.0466
ResNet34	0.6968 ± 0.1268	0.4201 ± 0.0419	0.8105 ± 0.0346	0.8331 ± 0.0302	0.8139 ± 0.0248
ResNet50	0.7393 ± 0.1626	0.4957 ± 0.0445	0.7924 ± 0.1045	0.8414 ± 0.0128	0.8157 ± 0.0408
EffNetB0	0.5545 ± 0.0402	0.4818 ± 0.1413	0.7903 ± 0.0500	0.8009 ± 0.0376	0.8236 ± 0.0288
EffNetB1	0.5813 ± 0.1164	0.4628 ± 0.0711	0.7428 ± 0.0364	0.7172 ± 0.1550	0.8066 ± 0.0180
GRU	0.6706 ± 0.1306	0.5889 ± 0.0224	0.6956 ± 0.0537	0.6695 ± 0.0822	0.7831 ± 0.0283
LSTM	0.6026 ± 0.0854	0.5833 ± 0.0502	0.7682 ± 0.0630	0.6412 ± 0.0427	0.7943 ± 0.0221
Transformer	0.5667 ± 0.0611	0.5329 ± 0.0759	0.5737 ± 0.0930	0.5963 ± 0.0133	0.6434 ± 0.0342
Mean	0.6336	0.4983	0.7336	0.7415	0.7852

Table 55: Private / MR / AUROC (Extended from Table 31). Mean ± SD across runs.

Model	Real	Syn	Syn→Real	Real→Real	Syn→Real→Real
ResNet18	0.5793 ± 0.0197	0.5172 ± 0.0054	0.5984 ± 0.0433	0.5845 ± 0.0370	0.6122 ± 0.0212
ResNet34	0.5888 ± 0.0169	0.5221 ± 0.0153	0.6483 ± 0.0199	0.5890 ± 0.0184	0.6440 ± 0.0220
ResNet50	0.6110 ± 0.0175	0.5152 ± 0.0163	0.6247 ± 0.0386	0.6024 ± 0.0264	0.6173 ± 0.0145
EffNetB0	0.5655 ± 0.0692	0.5249 ± 0.0182	0.6835 ± 0.0133	0.5784 ± 0.0550	0.6869 ± 0.0136
EffNetB1	0.5612 ± 0.0605	0.5365 ± 0.0091	0.6748 ± 0.0539	0.5639 ± 0.0937	0.6842 ± 0.0458
GRU	0.6093 ± 0.0283	0.5330 ± 0.0169	0.5931 ± 0.0557	0.6034 ± 0.0342	0.6267 ± 0.0205
LSTM	0.5900 ± 0.0273	0.5119 ± 0.0119	0.5599 ± 0.0372	0.5840 ± 0.0299	0.5887 ± 0.0450
Transformer	0.5492 ± 0.0363	0.4775 ± 0.0156	0.5472 ± 0.0074	0.5705 ± 0.0178	0.5416 ± 0.0206
Mean	0.5818	0.5173	0.6162	0.5845	0.6252



(a) Normal (b) AS (c) AR (d) MR

Figure 15: Results of inter-rater human evaluation for each PCG class.

manuscript concerns the utility of synthetic data pre-training for subsequent fine-tuning on real data. The low zero-shot performance observed in this appendix does not negate the efficacy of pre-training. In the transfer learning literature, it is well established that even partially misaligned domains can facilitate optimization during fine-tuning through the acquisition of low-level acoustic representations and favorable initialization. This interpretation is consistent with the behavior observed in our study. Thus, these findings clarify both the limitations and utility of synthetic PCG data: while synthetic data are beneficial for learning distributional properties and low-level representations, they are insufficient for generalizing disease-specific discrimination without fine-tuning.

Table 56: Private / AS / AUROC (Extended from Table 38). Mean ± SD across runs.

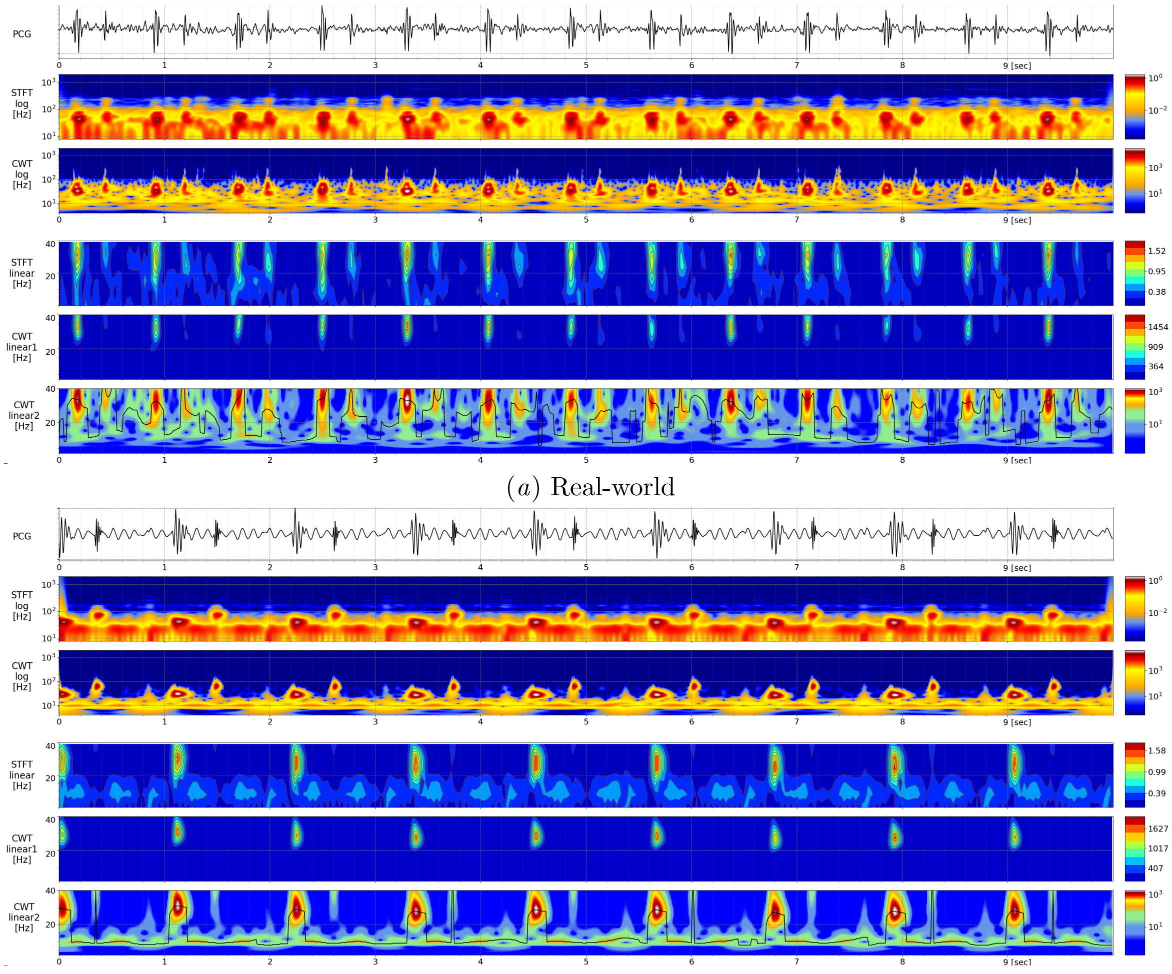
Model	Real	Syn	Syn→Real	Real→Real	Syn→Real→Real
ResNet18	0.7844 ± 0.0150	0.5847 ± 0.0146	0.7922 ± 0.0133	0.7698 ± 0.0234	0.7930 ± 0.0145
ResNet34	0.8145 ± 0.0084	0.6021 ± 0.0232	0.8359 ± 0.0133	0.8050 ± 0.0177	0.8374 ± 0.0176
ResNet50	0.7907 ± 0.0248	0.6055 ± 0.0260	0.8078 ± 0.0202	0.7881 ± 0.0208	0.8068 ± 0.0270
EffNetB0	0.7711 ± 0.1731	0.6850 ± 0.0204	0.8761 ± 0.0092	0.8468 ± 0.0229	0.8690 ± 0.0146
EffNetB1	0.7939 ± 0.1622	0.6682 ± 0.0987	0.8816 ± 0.0179	0.8785 ± 0.0065	0.8792 ± 0.0144
GRU	0.8357 ± 0.1176	0.5382 ± 0.0195	0.8917 ± 0.0168	0.8914 ± 0.0146	0.8933 ± 0.0129
LSTM	0.8139 ± 0.0927	0.5247 ± 0.0274	0.8570 ± 0.0215	0.8339 ± 0.0318	0.8530 ± 0.0146
Transformer	0.5818 ± 0.0283	0.5978 ± 0.0295	0.6659 ± 0.0309	0.6022 ± 0.0388	0.6509 ± 0.0240
Mean	0.7322	0.6008	0.8260	0.8020	0.8288

Table 57: Private / AR / AUROC (Extended from Table 45). Mean ± SD across runs.

Model	Real	Syn	Syn→Real	Real→Real	Syn→Real→Real
ResNet18	0.5673 ± 0.0537	0.5599 ± 0.0121	0.6542 ± 0.0369	0.6022 ± 0.0500	0.6616 ± 0.0249
ResNet34	0.5754 ± 0.0597	0.5502 ± 0.0215	0.6509 ± 0.0199	0.5823 ± 0.0382	0.6433 ± 0.0359
ResNet50	0.5736 ± 0.0500	0.5505 ± 0.0335	0.5884 ± 0.0621	0.5568 ± 0.0398	0.6116 ± 0.0557
EffNetB0	0.5490 ± 0.0600	0.4731 ± 0.0412	0.5352 ± 0.0582	0.5781 ± 0.0582	0.6014 ± 0.0272
EffNetB1	0.5613 ± 0.0251	0.4958 ± 0.0266	0.6200 ± 0.0432	0.6006 ± 0.0179	0.6016 ± 0.0226
GRU	0.5978 ± 0.0490	0.5105 ± 0.0062	0.6598 ± 0.0088	0.6127 ± 0.0375	0.6493 ± 0.0236
LSTM	0.5638 ± 0.0231	0.5047 ± 0.0145	0.6453 ± 0.0408	0.5753 ± 0.0391	0.6341 ± 0.0405
Transformer	0.5448 ± 0.0140	0.4995 ± 0.0486	0.5787 ± 0.0218	0.5596 ± 0.0168	0.5773 ± 0.0138
Mean	0.5666	0.5180	0.6166	0.5834	0.6225

Table 58: Causal vs. Non-causal Transformer (AUROC, mean ± SD).

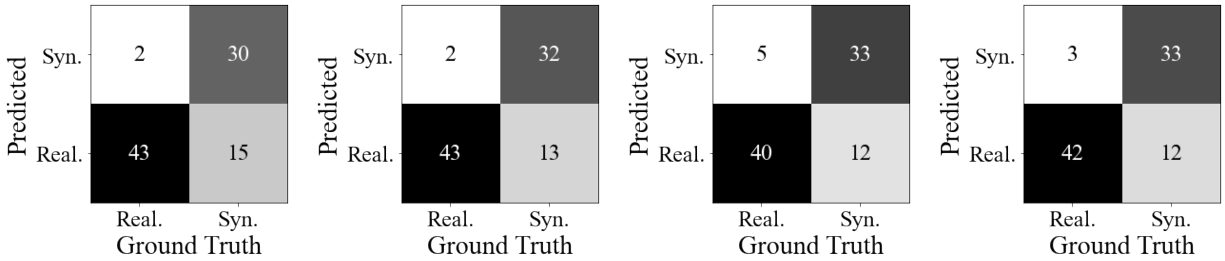
Dataset	Target	Causal/Non-causal	Real	Syn	Syn→Real
BMD-HS	MR	Non-causal	0.6787 ± 0.1112	0.5050 ± 0.0525	0.6182 ± 0.0539
		Causal	0.6708 ± 0.1155	0.6037 ± 0.0805	0.6732 ± 0.0768
	AS	Non-causal	0.6785 ± 0.0281	0.6287 ± 0.0482	0.6699 ± 0.0556
		Causal	0.6771 ± 0.0274	0.5798 ± 0.0865	0.6508 ± 0.0415
	AR	Non-causal	0.5667 ± 0.0611	0.5329 ± 0.0759	0.5737 ± 0.0930
		Causal	0.5640 ± 0.0549	0.5489 ± 0.0760	0.5939 ± 0.0900
Private	MR	Non-causal	0.5492 ± 0.0363	0.4775 ± 0.0156	0.5472 ± 0.0074
		Causal	0.5511 ± 0.0395	0.5763 ± 0.0280	0.5713 ± 0.0368
	AS	Non-causal	0.5818 ± 0.0283	0.5978 ± 0.0295	0.6659 ± 0.0309
		Causal	0.5763 ± 0.0280	0.6138 ± 0.0383	0.6782 ± 0.0208
	AR	Non-causal	0.5448 ± 0.0140	0.4995 ± 0.0486	0.5787 ± 0.0218
		Causal	0.5481 ± 0.0157	0.4975 ± 0.0537	0.5710 ± 0.0200



(a) Real-world

(b) Synthesized

Figure 16: Examples of visualized data used for human evaluation.



(a) Normal (b) AS (c) AR (d) MR

Figure 17: Results of human evaluation for each PCG class given visualized data.

Table 59: Average distributional distances between datasets (averaged over 10 features).

	BMD-HS vs Private	BMD-HS vs Synthesized	Private vs Synthesized
TVD	0.66	0.44	0.53
KL divergence	3.75	1.60	3.56
L1 norm	1.33	0.87	1.07
L2 norm	0.35	0.25	0.23
FAD	17.26	17.22	16.85

Table 60: Mean AUROC across eight models for TSTR and TRTS conditions.

Dataset	Target	TSTR*	TRTS
BMD-HS	AR	0.4983	0.5036
	AS	0.5096	0.5035
	MR	0.5267	0.5011
Private	AR	0.5180	0.5101
	AS	0.6008	0.5372
	MR	0.5173	0.5016

* Equivalent to "Syn" setting. The values are from Tables 10, 17, 24, 31, 38 and 45.