Conditional Diffusion Models for ECG Signal Denoising

Pie de Boer
Department of Advanced Computing Sciences
Faculty of Science and Engineering
Maastricht University
Maastricht, The Netherlands

***Abstract*—This study adapts the Super-Resolution via Repeated Refinement (SR3) conditional diffusion model for ECG noise removal. We aimed to develop a single model capable of eliminating electrode motion (EM), muscle artifacts (MA), and composite noise, which are particularly challenging to remove at low signal-to-noise ratios (SNRs) without data-driven methods.**

We employed Gram Angular Fields for lossless embedding of 1D signals into the image domain. The model's performance was compared against various filters, including an LMS adaptive filter, FIR low pass filter (LPF), moving average, and a hybrid filter (LPF with LMS), depending on the noise type. Initially, the model was trained on MIT-BIH Arrhythmia Database (ARDB) data with added noise from the MIT-BIH Noise Stress Test Database. We further assessed the model's performance on the MIT-BIH Atrial Fibrillation Database (AF) to evaluate its generalization ability.

Our model effectively tackled EM, MA, and composite noise across SNRs ranging from 0 to 15 decibels (dB). It outperformed baseline models in handling single noise types (EM, MA) at low SNRs (0 and 5 dB). While excelling in removing composite noise on ARDB across all SNRs, its effectiveness diminished for single noise types at high SNRs. Retraining our model on atrial fibrillation data significantly improved its ability to denoise these types of signals.

Our methodology presents an innovative yet computationally demanding approach to ECG noise removal, excelling in low SNR and multi-noise settings.

***Index Terms*—electrocardiography, deep learning, digital signal processing, artificial intelligence, noise reduction**

Introduction

Acquiring clean ECG signals is vital in healthcare, as signals with minimal noise can significantly improve the accuracy of diagnosing potential heart diseases [1]. In addition to the expected clinical settings, wearable devices like smartwatches can monitor ECG signals. Signal corruption can be expected with those devices, for example, due to their frequent movement [2].

Common sources of ECG signal corruption include powerline interference, baseline wander, electrode motion, and muscle artifact noise [3]. Using digital filters, such as high-pass and notch filters, power line interference and baseline wander can be (easily) removed [1]. However, eliminating electrode motion and muscle artifact noise without data-driven approaches is challenging due to their potential to distort the signal waveform and introduce unwanted artifacts [1]. Popular non-data-driven methods for electrode motion noise and muscle artifact noise removal considered in this paper are LMS adaptive filtering, moving average, and low pass filters [1], [4], [5]. These methods serve as a baseline to compare the performance of the conditional diffusion model. Unfortunately, traditional methods usually only work well when the ECG is not heavily corrupted and tend to fail when the noise has high amplitude [6].

In recent years, the strength of machine learning has been leveraged in various ways to denoise ECG signals. We will discuss some noteworthy examples relevant to our work's context. Starting with convolution neural networks, which have been successfully used by multiple researchers, outperforming traditional approaches such as filtering and wavelet-based approaches [7], [8], [9]. Fully convolutional denoising autoencoders (FCN DAE) have shown promising results in removing composite noise [3]. Inspired by the success of deep learning in the image processing domain, U-net models, popular for image segmentation [10], have been successfully adapted to denoise ECG signals with low power usage [11]. Using a U-Net, the researchers tackled multiple noise types, which included baseline wander, additive white Gaussian noise, electrode motion, and muscle artifact noise.

Going beyond 'standard' deep learning, researchers started investigating the potential application of generative models for denoising (and synthesizing) ECG signals. In 2023, researchers achieved enhanced noise removal performance by utilizing variational auto-encoders with a convolution mask, effectively addressing composite noise comprising baseline wander, muscle artifact, and electrode motion noise, surpassing other contemporary methods [12]. The generative adversarial network (GAN) emerged as another prominent model for denoising and synthesizing ECG signals. This model demonstrated its efficacy in denoising ECG signals while preserving crucial signal characteristics, outperforming autoencoder methods, as shown in this research [13].

To overcome challenges with training GANs, such as unstable training and decreased diversity due to their adversarial nature, researchers have investigated the viability of using (score-based) diffusion models instead [14]. In the previously mentioned work, researchers used a probabilistic diffusion

This thesis was prepared in partial fulfillment of the requirements for the Degree of Bachelor of Science in Data Science and Artificial Intelligence, Maastricht University. Supervisor(s): Pietro Bonizzi model for synthetic ECG signal generation, imputation, and forecasting. In another recent work from 2023, researchers used OpenAIs stable diffusion with time series imaging [15] to generate realistic ECG signals encompassing one heartbeat [16]. To our current understanding, only one study has employed a conditional diffusion model for denoising ECG signals [6].

In this study, we adapted the Super-Resolution via Repeated Refinement (SR3) model for ECG denoising. SR3 is a conditional diffusion model able to upscale blurry input images and can be used for other tasks such as inpainting [17]. The model relies on a U-Net that serves as a denoising function. Therefore, conditioned on a noisy input, it will learn how to recover a clean reconstructed image from pure Gaussian noise. By embedding 'slices' of our ECG signals using Gram Angular Fields (GAF), as described in [15], we transformed our 1D signal processing challenge into an image processing challenge. To the best of our knowledge, this is the first research to combine a conditional diffusion model with image embedding for ECG noise removal.

For an extensive overview of deep generative models used in biomedical signal processing focusing on ECGs and electroencephalograms (EEGs), we advise looking at the following work [18].

In our study, we aim to address the following research questions:

- Can the conditional diffusion model (SR3) effectively remove electrode motion and muscle artifact noise from ECG signals?
- Can the model (SR3) effectively address composite noise, comprising multiple types of noise corrupting a single 'raw' signal?
- Can the model (SR3) be effectively used for noise removal on different 'types' of ECG signals, such as those from certain heart patients?

Our contribution to the field is to provide a (novel) approach that leverages the advances found in computer vision in the biomedical signal processing domain. Using a simple embedding strategy, namely Gram Angular Fields, our approach allows researchers with an accessible approach to use the SR3 model for ECG signal denoising. Our methodology does not require a challenging redesign of model architectures, such as the U-Net denoising function. The approach works seamlessly with SR3 and the underlying U-Net, as it utilizes square images/embeddings with dimensions that are powers of 2.

We will begin our work with a brief introduction of the theoretical background of the SR3 conditional diffusion model, GAF embedding, and baseline methods. Afterward, we will discuss our implementation in detail, including frameworks, data preparation, and other critical attributes necessary for proper reproducibility. Moving on to the experiments, we discuss how we compared our models against the baseline methods on data from two datasets corrupted with different types of noise at low and high SNRs. Finally, the results will be presented, strengths and limitations will be discussed before drawing conclusions, and future directions will be suggested.

METHODS

A. Conditional Diffusion Model

SR3 is a conditional diffusion model that works directly with images [17]. The Denoising Diffusion Probabilistic Model (DDPM), which we will also refer to as the diffusion model, was introduced in 2020 for synthetic image generation [19]. It is worth first investigating 'traditional' diffusion models to understand conditional diffusion models more deeply. In diffusion models, two processes play a main role. In the forward process, Gaussian noise is gradually added in T steps to a clean image x_0. We can denote this process as $q(x_t \mid x_{t-1})$, also known as diffusion. In the reverse inference process, data is generated from noise in a (stochastic) iterative fashion. This happens according to $p_\theta(x_{t-1} \mid x_t)$. The reverse denoising process is achieved using a trainable network, for which researchers of the original paper utilize a U-Net [19]. Figure 1 shows the diffusion and inference processes for DDPM, where an image of a face x_0 is generated from pure Gaussian noise x_T.

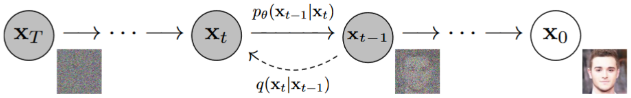
$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$

Fig. 1. Forward and backward process for the original diffusion model (DDPM). Taken from [19].

In the conditional diffusion process, however, we aim to learn a parametric approximation for $p(y \mid x)$. This can be done through a stochastic iterative refinement process, which maps a source image x to a target image $y \in \mathbb{R}^d$. The researchers of the SR3 model achieved this by adapting the DDPM [19] for conditional image generation. We will break down the essential workings of the model.

A target image y_0 is generated using conditional diffusion models in T refinement steps. Starting with an image of pure noise $y_T \sim \mathcal{N}(0, I)$ the model takes a denoising step obtaining y_{T-1} and repeats this process T times in order to obtain y_0. The denoising steps are learned according to the conditional transition distribution $p_\theta(y_{t-1} \mid y_t, x)$.

The conditioning on x makes the SR3 approach stand out from the DDPM mentioned earlier. Instead of just learning a network to denoise pure noise as with the DDPM, the reverse process is now conditioned on x, expressed as $p_\theta(y_{t-1} \mid y_t, x)$. The denoising (process) can be learned with a neural network f_θ that takes a source image x and a noisy target image $\tilde{y}$, striving to restore the noiseless target image y_0. For the exact details of the training and inference phase, we refer to the original SR3 paper [17]. We made no alterations concerning training and inference in our implementation. Figure 2 shows the conditional diffusion process for SR3. It is important to note that the visualization does not show the source image x.

B. Embedding

Since (conditional) diffusion models (often) work with images, we used a times series embedding named Gram Angular

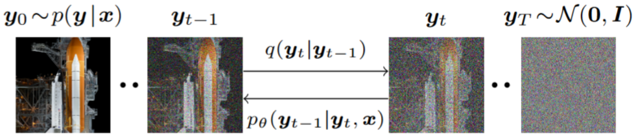
$\boldsymbol{y}_0 \sim p(\boldsymbol{y}\,|\,\boldsymbol{x})$
$\boldsymbol{y}_{t-1}$
$q(\boldsymbol{y}_t|\boldsymbol{y}_{t-1})$
$p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x})$
$\boldsymbol{y}_t$
$\boldsymbol{y}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

Fig. 2. The forward and backward processes for the conditional diffusion model are similar to the original DDPM. However, notice that we condition on both y_t and x in the backward process. In addition to that, our target image y_0 is eventually obtained from y given x through iterative steps of the denoising network, denoted as $p_\theta(y_{t-1} \mid y_t, x)$. Taken from [17].

Fields [15] for our ECG signals. The embedding process consists of three steps. Firstly, we apply min-max scaling to ensure the ECG signal is in the range $[-1, 1]$. As a second step, we take the polar coordinates of the normalized signal. Finally, we obtain the Gram Angular (Summation) Field from the polar coordinates. The Gram Angular Summation Field is a 2D square matrix of size $N * N$. The input length of the 1D signal determines the size of N.

Gram Angular Field embedding has the favorable properties of being bijective and preserving temporal dependency. This means that the 1D signal can be reconstructed from the 2D GAF embedding without any loss (except for potential numerical inaccuracies) while maintaining the sequence and relationship of data points over time. Figure 3 depicts the three steps of the embedding process.

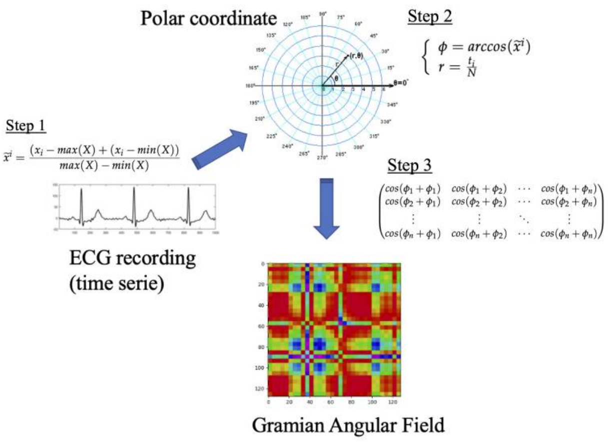
Polar coordinate
Step 2
$\phi = arccos(\tilde{x}^i)$
$r = \frac{t_i}{N}$
Step 1
$\tilde{x}^i = \frac{(x_i - max(X)) + (x_i - min(X))}{max(X) - min(X)}$
Step 3
$\begin{pmatrix} cos(\phi_1+\phi_1) & cos(\phi_1+\phi_2) & \cdots & cos(\phi_1+\phi_n) \\ cos(\phi_2+\phi_1) & cos(\phi_2+\phi_2) & \cdots & cos(\phi_2+\phi_n) \\ \vdots & \vdots & \ddots & \vdots \\ cos(\phi_n+\phi_1) & cos(\phi_n+\phi_2) & \cdots & cos(\phi_n+\phi_n) \end{pmatrix}$
ECG recording
(time serie)
Gramian Angular Field

Fig. 3. Visualization of the three (reversible) steps to obtain the Gram Angular Field from a 1D signal. Taken from [20].

C. SR3 for ECG Signal Denoising

In the SR3 paper, researchers aimed to reconstruct a high-resolution target image given a low-resolution source image. This relates to the previously mentioned x, representing the source image, and y_0, representing the target image. Figure 4 illustrates how SR3 upscales blurry input images to produce high-resolution outputs that closely resemble the reference images.

Instead of using low-resolution and high-resolution images, we devised the idea of using 2D embeddings of clean and noisy ECG signals. We used the model to reconstruct embeddings (images) of clean ECG signals y_0, given an embedding (image) of a noisy ECG signal x. Since we can loosely embed and de-embed our signals using Gram Angular Fields, we can convert our image back and forth between the image domain without (significant) loss. As long as the model is presented with square images, where the dimensions are a power of 2, no further adaptions are needed to use the SR3 model, and the model can directly be trained and used for inference. In our case, inference means generating an embedding (image) of a clean ECG signal y_0 from pure noise y_t given an embedding (image) of a noisy source ECG signal x. Training in this context means learning the parameterized function p_θ that can denoise y_t conditioned on x to obtain y_0.

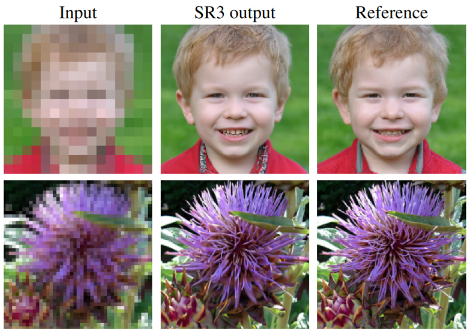

Fig. 4. Two low-resolution images used as input for the SR3 model (not our implementation). The first image, originally 16×16, was upscaled to 128×128, while the second image, initially 32×32, was upscaled to 256×256 Taken from [17].

D. Single-Shot and Multi-Shot Reconstructions

As indicated by the theoretical analysis and results in [6], employing conditional diffusion models for ECG noise removal benefits from averaging multiple reconstructions rather than relying on single outcomes. Since conditional diffusion models, thus SR3, rely on random Gaussian distributions, each reconstruction from separate runs can be considered independent.

Equation (1) shows how a multi-shot reconstruction can be built.

$$y_{\text{comb}} = \frac{1}{M} \sum_{m=1}^{M} y_m \tag{1}$$

Where:

- y_{comb} represents the ensemble of multiple shots.
- y_m represents the m-th reconstruction.
- M is the number of shots used.

Our m-th reconstruction y_m can be defined as the clean ECG signal y plus a reconstruction error ϵ_m, shown in Equation (2).

$$y_m = y + \epsilon_m \tag{2}$$

Using Equation (2) for the single reconstruction allows us to derive Equation (3), which shows that the reconstruction error of combined runs is lower or equivalent to the reconstruction error of a single run.

$$\left(\sum_{m=1}^{M} \frac{1}{M} \epsilon_m\right)^2 \leq \sum_{m=1}^{M} \frac{1}{M} \epsilon_m^2 \tag{3}$$

We reference to [6] if the derivation of Equation (3) is unclear.

E. Baseline Methods

Our choice of baseline methods depends on which type of noise we aim to remove from the ECG signal. To remove muscle artifact noise, we investigated using a moving average filter as suggested in the following work [4]. We also implemented a finite-impulse response (FIR) low-pass filter (LPF) for muscle artifact noise removal, as described in [5].

To remove electrode motion, we used a least means squares (LMS) adaptive filter as proposed in the earlier mentioned classical work on biomedical signal processing [1].

Since the signal with composite noise is corrupted with both MA and EM, it seemed appropriate to use a hybrid method as a baseline. We cascaded the LPF with LMS (in series) to remove composite noise since it gave better results than LMS followed by LPF. We decided not to investigate wavelet-based denoising approaches since previous studies mainly used those for Gaussian noise removal in ECG signals, such as in the following paper [21].

a) Moving Average Filter: A moving average filter works by averaging a fixed number of consecutive data points to smooth out short-term fluctuations. Each output value is calculated by taking the mean of the current point and a specified number of preceding points [22]. This process reduces noise and highlights longer-term trends in the data. Using a larger number of points smoothens the signal more drastically, which might come at a risk of losing essential signal characteristics. We have included Equation (4) for the moving average filter as sourced from [22].

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \tag{4}$$

Where:

- x represents the input signal (noisy ECG signal),
- y represents the output signal (denoised ECG signal),
- M represents the number of points used in the moving average.

The moving average filter exclusively incorporates data points preceding the calculated output sample.

b) FIR Low Pass Filter: The FIR Low Pass Filter utilizes a finite impulse response approach to attenuate high-frequency components while preserving low-frequency signals [23]. This leads to specific frequencies being attenuated beyond the specified cutoff frequency. Since MA might be focused at higher frequencies than the main characteristics of the ECG signal, this approach can help remove MA noise found in ECG signals. Choosing a correct cutoff frequency is essential to maintain important signal characteristics while eliminating noise from the signal.

c) LMS Adaptive Filtering: The LMS adaptive filter algorithm adjusts filter coefficients based on the difference/error $e(n)$ between the desired output $d(n)$ (clean ECG signal) and the actual output signal $y(n)$ (denoised ECG signal). It iteratively minimizes the mean square error by updating filter coefficients in the direction of the negative gradient of the error with respect to the coefficients [23]. The central equation (5) that describes this process is given below.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \tag{5}$$

Where:

- $\mathbf{w}(n)$ is the filter coefficient vector at iteration n,
- μ is the step size or adaptation constant,
- $e(n)$ is the error signal at iteration n,
- $\mathbf{x}(n)$ is the input signal vector at iteration n.

Figure 5 illustrates the principle of adaptive filtering. In the context of ECG denoising, $d(n)$ represents the target signal (clean ECG signal), $x(n)$ is the input (noisy ECG signal), $y(n)$ is the adaptive filter output (denoised ECG signal), and $e(n)$ is the reconstruction error, calculated as the difference between $d(n)$ and $y(n)$.

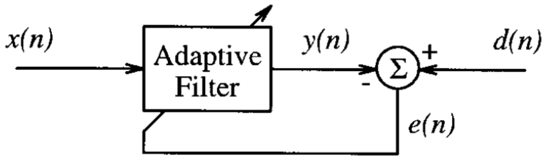

Fig. 5. Principle of adaptive filtering taken from [23].

F. Datasets

We utilized the MIT-BIH Arrhythmia Dataset (ARDB) [24] available through PhysioNet [25] for its high-quality samples and widespread acceptance in ECG signal research. We selected all records available from this dataset. Samples of 'pure' EM and MA noise were sourced from the MIT-BIH Noise Stress Test Database (NSTDB) [26]. Access to those samples facilitated the generation of a synthetic dataset. The MIT-BIH Atrial Fibrillation (AF) dataset [27] was selected since it includes signals exclusively from patients with atrial fibrillation. These signals present a unique challenge for our model due to the noise-like characteristics associated with this heart condition. We aimed to investigate the model's ability to preserve the underlying signal structure while specifically targeting artificially added noise. We acknowledge that specific signals within the ARDB dataset include segments exhibiting atrial fibrillation. However, upon reviewing the annotations for ARDB, it was found that only approximately 30 (small) segments were affected, indicating minimal bias when retraining on AF.

IMPLEMENTATION

G. Parameters of SR3 U-Net and Diffusion Model

We used the PyTorch-based implementation of the SR3 model available on GitHub [28]. The architecture of the U-Net, used as the denoising function, remained unchanged. We initialized the denoising network (U-Net) with the settings in Table I. We adopted these settings directly from [28] without making any changes.

TABLE I
INITIALIZATION PARAMETERS OF THE DENOISING FUNCTION (U-NET)

Parameter	Value
In channels	2
Out channels	1
Inner channels	32
Norm groups	32
Channel multipliers	[1, 2, 4, 8, 8]
Attention resolution	8
Residual blocks	3
Dropout	0
Noise level embedding	True
Image size	128

We implemented a noise schedule for the diffusion model with the settings listed in Table II. Although we experimented with fewer steps to expedite the diffusion process, it resulted in noisier reconstructions. We investigated the noise schedule proposed in [6] to expedite inference, but it resulted in poorer outcomes. In the end, we kept the original settings from the GitHub implementation [28].

TABLE II
NOISE SCHEDULE SETTINGS

Parameter	Value
Beta start	1×10^{-6}
Beta end	1×10^{-2}
Number of steps	2000
Schedule	Linear

We initialized the diffusion model with the settings detailed in Table III. By extracting slices of 128 samples, we generated embeddings of size 128 by 128. Given that we worked with grayscale images, our channel count remained one. We kept SR3's original loss function as we did not identify the need for modifications.

TABLE III
DIFFUSION MODEL INITIALIZATION PARAMETERS

Parameter	Value
Denoising function	As described in I
Image size	(128, 128)
Channels	1
Loss type	L1
Conditional	True
Diffusion configuration	As described in II

Using the parameters for initialization of the U-Net (denoising network) and the diffusion model, given the SR3 implementation from GitHub [28] with the suggested noise schedule in table II, allows for the exact reconstruction of the denoising network and diffusion model we used for training and inference.

For a detailed description of the network architecture used, we advise checking the GitHub implementation [28] and, most importantly, the original SR3 paper [28].

H. Single-Shot and Multi-Shot Reconstructions

While computationally demanding due to SR3's lengthy sampling process, Equation (3) shows that single-shot reconstruction errors provide an upper bound for the errors of multiple-shot reconstructions. Since SR3 utilizes a high number of diffusion steps, namely 2000 (see Table II), we limited the number of runs to six. We saved both the individual reconstructions and built multiple-shot reconstructions (from six runs) according to Equation (1). Deploying the multi-shot approach in practical scenarios, such as integration into a software package, may encounter challenges due to the prolonged sampling times required for reconstruction generation. However, this approach provides valuable insights into the potential quality of reconstructions. As a result, we will investigate both single-shot and multiple-shot reconstructions, as outlined in further detail in the Experiments section.

I. Data Preparation

a) Datasets: Using ARDB data, we generated three different training datasets to train the model iteratively for EM, MA, and composite noise removal. We iterated over all records in ARDB (healthy patients and those with arrhythmias) to generate slices of length 192, corresponding to approximately two heartbeats at a sample rate of 128 Hz. This process resulted in approximately 57000 samples. The first 50000 samples were used for training, and the last 7000 were used for validation. To improve our methodology, we should have ensured the data split was not solely based on the sample count but also patient records. Our current splitting method might introduce bias, as it could train the model on early segments from one patient and validate it on later segments from the same patient. However, we anticipate minimal impact, as this likely only occurred with one record.

We applied the same procedure to the AF dataset, resulting in 50000 samples for training and 8000 for validation.

b) Preparation of Slices: The noisy signals were generated by either adding MA or EM noise to a preprocessed 'cleaned' ECG signal or by taking the raw signal and adding both MA and EM to it. If we trained our model with composite noise samples, the ground truth would still be the preprocessed clean signal. Below is an overview of the steps involved.

- **Signal Cleaning:**
  - Take a full-length ECG signal (Lead I).
  - Apply preprocessing steps:
    - Remove baseline wander – high-pass filter at 0.5 Hz using `scipy.signal.butter`.
    - Remove power-line interference – notch filter at 60 Hz using `scipy.signal.iirnotch`.
- **Data Segmentation:**

- Slice the preprocessed ECG signal into samples of length 192.

- **Noise Addition:**
  - Randomly select a slice of length 192 from the noise sample.
  - Add the noise sample(s) to the preprocessed ECG signal at a desired signal-to-noise ratio (SNR).

c) Noise Addition: We adapted the approach for adding noise from [8]. The noise was added according to Equation (6).

$$X_n = X + noise \times \lambda \tag{6}$$

Where:

- X is the original ECG signal,
- X_n is the ECG signal with noise added,
- $noise$ is the pure noise sample,
- λ is a hyperparameter controlling the SNR of the noisy signal.

Where λ was computed according to Equation (7).

$$\lambda = \frac{\text{RMS}(X)}{\text{RMS(noise)} \cdot 10^{\left(\frac{0.1 \cdot a}{2}\right)}} \tag{7}$$

In Equation (7), a signifies the desired SNR in dB.

Individual noise (EM, MA) was added to our data at an SNR of 3 dB. Composite noise was added at a slightly higher SNR of 5 dB since composite noise might more heavily corrupt the signals. Our decision for the SNRs was motivated by generating noisy 'enough' signals that pose a clear challenge to the model (and baseline methods) while maintaining 'some' signal characteristics intact. We assumed that adding noise at much lower SNRs might corrupt the signal to the extent that the model would struggle to converge and become unable to reconstruct clean signals. Using multiple SNRs could diversify the training data by incorporating noisy and less noisy samples, possibly yielding improved performance across all SNRs. However, opting for this approach would further extend the already lengthy training times, exacerbated by the fact that we train the model from scratch, as explained later.

J. Embedding

We used the Gram Angular Fields embedding from the pyts package [29] to generate 'images' for the SR3 model. The 1D signal was reconstructed by taking the diagonal of the Gram Angular Summation matrix. We only used the Gram Angular Summation matrix, as grayscale images reduced the computational cost of training. To ensure square images with power-of-2 dimensions, we took the first 128 samples from our 192-length ECG slices.

K. Training

a) Training Configuration: Since neither the ARDB nor AF training datasets mentioned earlier could be loaded in RAM, we iterated over subsets of 2500 samples. We shuffled the clean and noisy data in our training and validation sets to avoid bias in how specific samples were presented to our model. We chose a batch size of 16 to avoid overloading our GPU memory. We used the Adam optimizer with a learning rate of 1×10^{-4}, mirroring the SR3 implementation [28]. The model was trained on subsets of 2500 samples for 30 epochs each. We saved the model for each interval (30 epochs) until the entire training set was exhausted. This allowed for easy comparison when using the model for inference to see if a lower loss function due to more training epochs in the image domain (always) corresponds to a better reconstruction in the 1D signal domain. Based on our investigation of the loss function's convergence, we selected 30 epochs. We trained the model from scratch on an NVIDIA K80 available through Google Cloud.

b) Training Approach: The model was trained to iteratively learn how to remove multiple individual noise types (MA, EM) and composite noise. Firstly, we aimed to train our model to remove MA noise on samples from ARDB. Afterward, we retrained the same model to remove EM noise on samples from ARDB. With the retrained model, we now investigated if it could remove EM noise while still being able to remove MA. Lastly, we took the previously mentioned model and trained it to remove composite noise, and we checked whether it could now remove both composite noise and individual noise (MA, EM) on ARDB. Later on, we will refer to this model as model 1.

To investigate the generalization capability and the model's understanding of underlying signal characteristics, we compared model 1 on samples exclusively of patients with atrial fibrillation with composite noise added. We also retrained model 1 on those types of samples (AF with composite noise). We will refer to the retrained model as model 2.

L. Baseline models

All baseline methods were run in Matlab. We used the LMS Adaptive Filter from their DSP System Toolbox [30] with default settings. The moving average filter was implemented with a window size of 4, which yielded better results than the suggested size of 8 from [4]. For the finite impulse response low pass filter, we used fir1 [31], a window-based filter design with the settings listed in table IV. These settings were based on [5], with the cutoff frequency slightly adjusted for improved performance. We combined the LPF and LMS adaptive filter in series for composite noise removal.

TABLE IV
PARAMETERS FOR THE FIR1 LOW PASS FILTER

Parameter	Value
Sampling rate (f_s)	128 Hz
Cutoff frequency (f_c)	40.60 Hz
Filter length (N)	14
Kaiser window parameter (β)	0

EXPERIMENTS

M. Model Comparisons

To evaluate the SR3 models for noise removal, we compared them with baseline methods specific to each noise type, as outlined in Table V.

TABLE V
NOISE TYPES AND ACCOMPANYING BASELINE METHODS

Noise Type	Baseline Models
Electrode Motion	LMS Adaptive Filter
Muscle Artifact	FIR Low Pass Filter, Moving Average
Composite	Hybrid (LPF → LMS)

We extracted 128-sample slices of ECG signals (from the 192-sample slices in the validation dataset) specifically chosen to have a visible (P)QRS complex, see Fig. 14 and Fig. 17 in the Appendix. This ensured distinct signal characteristics were visible, allowing for a better evaluation of the impact of different noise types on the signal and the effectiveness of our models for noise removal. We selected SNRs ranging from 0 to 15 dB, evenly distributed, to investigate model performance across a spectrum of noisy and relatively clean signals. Table VI outlines the investigated noise types and their corresponding SNRs.

TABLE VI
NOISE TYPES AND SNRS FOR SAMPLES INVESTIGATED

Noise Type	SNR (dB)	Dataset
Electrode Motion	0, 5, 10, 15	ARDB
Muscle Artifact	0, 5, 10, 15	ARDB
Composite	0, 5, 10, 15	ARDB
Composite	0, 5, 10, 15	AF

As stated in the Methods/Implementation, we investigated both single-shot and multiple-shot reconstructions. In the first part of the Results section (comparative analysis), we showcase the performance of the six-shot reconstructions against baseline methods for the samples listed in Table VI. In the second part of the Results, showcasing the results of statistical analysis (described in the next paragraph), we investigate the six (individual) single-shot reconstructions for each of the samples listed in Table VI.

We assessed model performance using root mean square error (RMSE), peak signal-to-noise ratio (PSNR), and mean absolute error (MAE). However, upon analysis, we found that RMSE and PSNR did not offer additional insights compared to MAE. Therefore, we only present results using MAE.

N. Statistical Analysis

We conducted two statistical tests to delve deeper into the single-shot performance of our models. We produced six individual reconstructions or estimated clean ECGs for each sample listed in Table VI. Model 1 was exclusively trained on ARDB, while model 2 was retrained on AF. These tests are designed to address two specific questions.

- **Q1**: Are the differences in MAEs for model (1 or 2) across various SNRs statistically significant?

By addressing the first question, we aimed to determine whether there was a statistical difference in our models' single-shot performance given varying input noise levels. Ideally, we expected a gradual reduction in MAE for the estimated clean ECGs with lower input noise levels. The absence of such a trend might suggest the need to train the model across multiple SNRs to achieve optimal performance in low- and high-noise settings. We employed the Kruskal-Wallis test to examine the performance of a single model across SNRs.

- **Q2**: Are the differences in MAEs between models (1 vs. 2) at a fixed SNR statistically significant?

With the second question, we aim to understand whether retraining on AF samples led to a potential decrease in performance on the original ARDB data. We utilized the Wilcoxon rank sum test to compare the models.

O. Relevance for RQs

Investigating the noise removal capability given different individual noise types helps us to answer our first research question. We strive to answer the second research question by examining the model's ability to remove composite noise. By evaluating model performance on the AF dataset, we aim to assess how well the models can remove artificially added noise from various types of ECGs. While ARDB included healthy and arrhythmic cases and offered insights, investing model performance on the AF dataset helped us shape a more comprehensive answer to our third research question. Investigating different SNRs gives us further insight into whether our model is a favorable choice compared to the baseline methods at some or all SNRs.

RESULTS

P. Comparative Analysis of SR3 and Baseline Methods

The plots in this subsection display the MAEs of the estimated denoised ECGs generated by the baseline methods and SR3 models 6-shot reconstructions (constructed according to Equation 1) given different noise types (MA, EM, and composite) on ARDB (see Figures 6, 7, and 8), as well as composite noise on AF (see Figure 9) at varying SNRs.

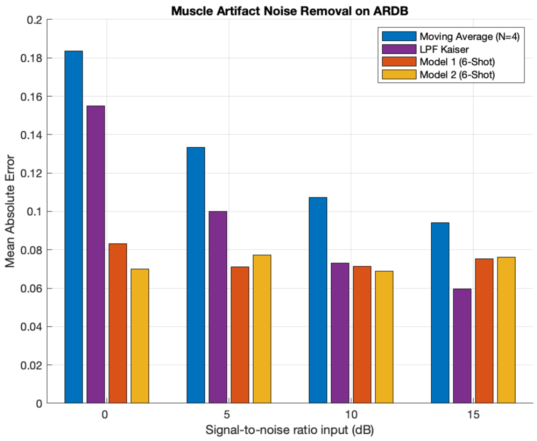

Fig. 6. MA noise removal on ARDB dataset. Model 1 was trained only on ARDB, and model 2 was retrained on AF.

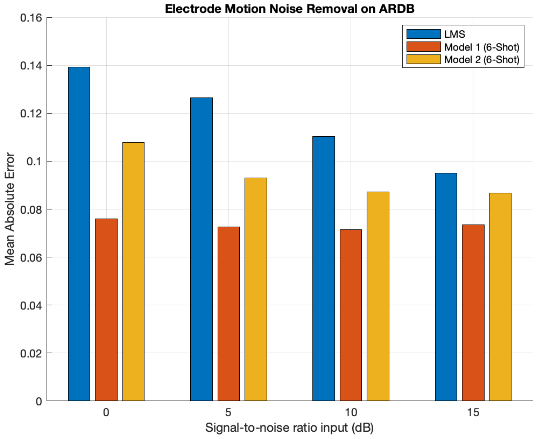

Fig. 7. EM noise removal on ARDB dataset. Model 1 was trained only on ARDB, and model 2 was retrained on AF.

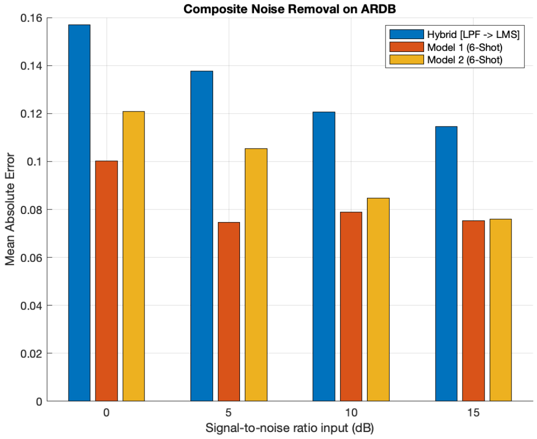

Fig. 8. Composite noise removal on ARDB dataset. Model 1 was trained only on ARDB, and model 2 was retrained on AF.

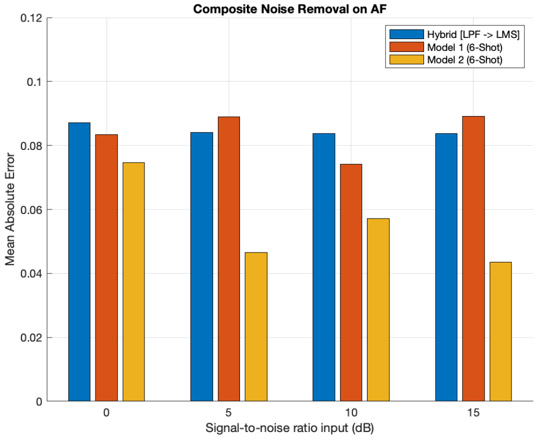

Fig. 9. Composite noise removal on AF dataset. Model 1 was trained only on ARDB, and model 2 was retrained on AF.

Q. Impact of Retraining on AF for Model Performance

Tables VII, VIII, IX and X show the results of the Wilcoxon rank sum test, where we compared the MAEs of each of our (six) single-shot reconstructions, generated for each sample listed in Table VI. These results highlight whether retraining on AF caused a statistically significant difference in performance or not, given certain noise types on specific datasets at varying SNRs.

TABLE VII
MODEL COMPARISON FOR MUSCLE ARTIFACT NOISE ON ARDB; WILCOXON RANK SUM TEST

SNR	P-Value	Model 1 (Median MAE)	Model 1 (IQR MAE)	Model 2 (Median MAE)	Model 2 (IQR MAE)
0	0.240	0.087	0.038	0.078	0.011
5	0.394	0.072	0.003	0.082	0.018
10	0.310	0.072	0.003	0.069	0.012
15	0.818	0.074	0.005	0.078	0.013

TABLE VIII
MODEL COMPARISON FOR ELECTRODE MOTION NOISE ON ARDB; WILCOXON RANK SUM TEST

SNR	P-Value	Model 1 (Median MAE)	Model 1 (IQR MAE)	Model 2 (Median MAE)	Model 2 (IQR MAE)
0	0.015	0.076	0.013	0.127	0.039
5	0.009	0.073	0.004	0.104	0.021
10	0.026	0.072	0.007	0.090	0.012
15	0.004	0.076	0.006	0.092	0.013

TABLE IX
MODEL COMPARISON FOR COMPOSITE NOISE ON ARDB; WILCOXON RANK SUM TEST

SNR	P-Value	Model 1 (Median MAE)	Model 1 (IQR MAE)	Model 2 (Median MAE)	Model 2 (IQR MAE)
0	0.002	0.112	0.012	0.128	0.021
5	0.026	0.076	0.011	0.111	0.014
10	0.394	0.075	0.034	0.091	0.009
15	0.937	0.078	0.009	0.079	0.018

TABLE X
MODEL COMPARISON FOR COMPOSITE NOISE ON AF; WILCOXON RANK SUM TEST

SNR	P-Value	Model 1 (Median MAE)	Model 1 (IQR MAE)	Model 2 (Median MAE)	Model 2 (IQR MAE)
0	1.000	0.085	0.012	0.090	0.038
5	0.002	0.103	0.052	0.053	0.015
10	0.065	0.079	0.017	0.065	0.019
15	0.002	0.120	0.012	0.050	0.029

R. *Performance Across Various SNRs with a Fixed Noise Type*

Figures 10, 11, 12 and 13 display the results of the Kruskal-Wallis tests for specific noise types and datasets. These figures illustrate whether the different SNRs result in significant differences in single-shot performance across the investigated SNRs.

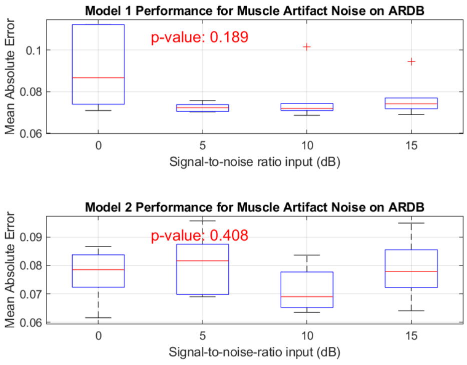

Fig. 10. Results of the Kruskal-Wallis tests showcasing single-shot model performance across various SNRs given MA noise on ARDB.

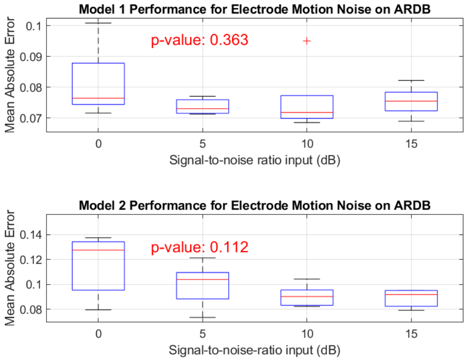

Fig. 11. Results of the Kruskal-Wallis tests showcasing single-shot model performance across various SNRs given EM noise on ARDB.

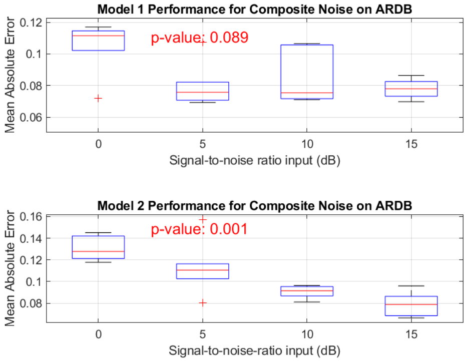

Fig. 12. Results of the Kruskal-Wallis tests showcasing single-shot model performance across various SNRs given composite noise on ARDB.

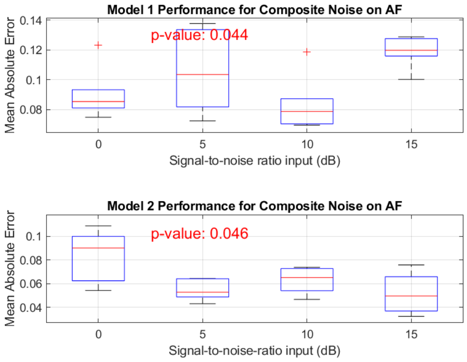

Fig. 13. Results of the Kruskal-Wallis tests showcasing single-shot model performance across various SNRs given composite noise on AF.

S. *Interpretation of Results*

a) Muscle Artifact Noise Removal on ARDB: Our models outperform baseline methods for MA noise removal at low SNRs, particularly 0 and 5 dB. However, a well-designed low-pass filter might yield slightly better results for a relatively clean signal (SNR 15 dB), as shown in Fig. 6. There was no significant difference in performance between model 1 and model 2 (p-value > 0.05); see Table VII. Both models showed statistically invariant results across different SNRs based on Kruskal-Wallis, as seen in Fig. 10. This suggests that training the models across various SNRs may be essential to achieve the desired gradual decrease in MAE as noise levels decrease.

b) Electrode Motion Noise Removal on ARDB: Model 1 performs well across all SNRs when examining EM noise, with a clear advantage apparent at low SNRs, as depicted in Fig. 7. Model 2 experiences a performance decline across all SNRs ($p < 0.05$), as illustrated in Table VIII. This indicates that training solely on the target distribution (ARDB) might yield optimal performance due to reduced model complexity. Similar to MA noise, both models exhibit invariance to the SNR of the input ($p > 0.05$) as illustrated in Fig. 11.

c) Composite Noise Removal on ARDB: For composite noise removal, the model trained solely on ARDB demonstrates superior performance across all SNRs, as shown in Fig. 8. Our retrained model (model 2) also outperforms the hybrid model across all SNRs. Although we observe a slight performance decline compared to model 1, significant only at SNR 0 and 5 dB; see Table IX. Model 2 exhibits statistically distinct results across SNRs, as depicted in Fig. 12. As with MA and EM noise types, model 1 appears to be statistically invariant to the SNR of the input.

d) Composite Noise Removal on AF: We achieved improved performance across all SNRs after retraining on AF compared to the baseline hybrid model; see Fig. 9. Inspection of Fig. 17 in the Appendix helps us better understand the results from model 1 and model 2. We observe that model 1 smoothened the baseline flutter characteristic of AF and struggled to reconstruct the second QRS complex. Model 2, however, preserved the original signal's morphology better. It retained the bump around the 50th sample and more accurately captured the second QRS complex around the 120th sample, particularly at SNR 10 and 15 dB. The performance of model 2 was statistically significant (improved) compared to model 1 for SNR 5 and 15 dB; see Table X. We hypothesize that the absence of a statistical difference at SNR 0 dB could be attributed to the high noise level.

Discussion

T. State of the art

We are the first to successfully adapt an image super-resolution diffusion model for ECG noise removal, demonstrating effective removal of various noise types and excellent performance in high-noise settings.

The closest competitors for model comparison would be (other) diffusion models and auto-encoders. In [6] (2023), using conditional diffusion for signal denoising, researchers showed excellent performance in high noise settings and removed multiple noise types; our findings align with this. However, we cannot directly compare whether our model improved upon theirs due to differences in metrics and investigated signals. Additionally, their reconstructions appear (relatively) insensitive to the input SNRs, as demonstrated in their results, exhibiting behavior similar to our model. Their approach required a complex architecture update for 1D signals, whereas our method directly adapted a model from the image processing domain. Neither their work nor ours addressed the high cost of the sampling process in diffusion models. When comparing our method to a U-Net-only approach as suggested in [11], which effectively addressed multiple types of noise, their method is the preferred choice if computational efficiency is a priority.

U. Shortcomings (and Fixes)

a) SNR Invariant Results: Due to computational constraints, we trained the models at fixed SNRs for each type of noise (EM, MA) and composite noise. Individual noise was introduced at an SNR of 3 dB and composite noise at an SNR of 5 dB.

Upon examining model 1's performance on ARDB for MA noise removal, unintended behavior is evident. In Figure 6, the reconstruction error is lower at SNR 5 dB than at SNR 15 dB. We believe this is caused by the similarity between the signals at SNR 5 dB and those affected at SNR 3 dB, which the model was exposed to during training.

We propose augmenting the training data with noisy signals at a higher SNR (e.g., 10 dB) to address this issue. This approach would expose the model to a broader range of noisy signals during training, potentially resulting in model performance that is more consistent with our expectations.

b) Decreased Performance Model 2: We believe that the slight decrease in performance on ARDB for both EM and composite noise is due to the increased complexity of the model, which must effectively handle noise added to signals from two underlying distributions (ARDB and AF). The best performance might be achieved by using specialized models based on the distribution of the ECG signals, such as a model specific to patients with atrial fibrillation. However, implementing this approach would require additional patient-specific information in real-world settings.

c) Improved Research Methodology: Some ARDB records contain atrial fibrillation segments identified through a detailed annotation review. Initial training on a dataset of exclusively healthy patients is preferable to properly assess the impact of retraining on the AF dataset.

Additionally, data splitting for ARDB and AF should be based on entire patient records, not just sample counts, to ensure segments from a single patient do not appear in both training and validation sets, preventing overfitting.

Conclusion

In our research, we were able to (successfully) adapt the SR3 conditional diffusion model for multi-type noise removal in ECG signals. Using Gram Angular Fields, the model could be used without changes to the denoising network or the diffusion model settings. Our models addressed both individual and composite noise in ECGs from healthy individuals and those with arrhythmias (as observed in ARDB). Retraining the model on a dataset containing only atrial fibrillation records significantly improved its capability to remove noise from such ECGs. Our models were predominantly advantageous over baseline methods for noisy signals (SNR 0 and 5 dB) and in composite noise settings.

RQ1: The SR3 model can be used to remove individual noise types (EM, MA) at low and high SNRs, although its use seems (only) beneficial compared to baseline methods at low SNRs of 0 and 5 dB.

RQ2: Using our methodology, we could denoise raw signals with composite noise added and outperformed the baseline method, which used a cascade of filters (LPF followed by LMS) across all SNRs investigated.

RQ3: Our model effectively denoised ECG signals from healthy individuals and those with arrhythmias (as seen in ARDB). However, retraining on AF samples significantly enhanced performance for these types of signals.

Our approach contributes to healthcare by providing a model that effectively removes challenging noise types, such as composite and electrode motion, in high-noise settings. However, further comparisons are required to determine if it outperforms current state-of-the-art methods. SR3's computational requirements may limit its practical application, but this could change as hardware improves and diffusion models become faster.

We believe the models could be improved by training on low and high SNRs, provided sufficient computational resources are available. This likely ensures better performance than baseline methods at high SNRs, such as for muscle artifact removal. Models tailored to specific ECG signals, like those from atrial fibrillation patients, may be preferable. Retraining on particular data (AF samples) improved performance for those samples but sometimes reduced performance on the original data, as seen with EM removal on ARDB.

References

[1] L. Sörnmo and P. Laguna, in *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Elsevier, 2005, pp. 440–443.

[2] S. Nagai, D. Anzai, and J. Wang, "Motion artifact removal for wearable ecg using stationary wavelet multi-resolution analysis," in *2017 IEEE 5th International Symposium on Electromagnetic Compatibility (EMC-Beijing)*, 2017, pp. 1–5.

[3] F. Shi, "A review of noise removal techniques in ecg signals," in *2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, 2022, pp. 237–240.

[4] R. M. Rangayyan and J. W. & Sons, "Biomedical signal analysis : a case-study approach," 2015.

[5] B. C. and M. Uplane, "High frequency electromyogram noise removal from electrocardiogram using fir low pass filter based on fpga," *Procedia Technology*, vol. 25, December 2016.

[6] H. Li, G. Ditzler, J. Roveda, and A. Li, "Descod-ecg: Deep score-based diffusion model for ecg baseline wander and noise removal," *IEEE Journal of Biomedical and Health Informatics*, p. 1–11, 2024.

[7] C. T. Arsene, R. Hankins, and H. Yin, "Deep learning models for denoising ecg signals," in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[8] E. Brophy, B. Hennelly, M. De Vos, G. Boylan, and T. Ward, "Improved electrode motion artefact denoising in ecg using convolutional neural networks and a custom loss function," *IEEE Access*, vol. 10, pp. 54 891–54 898, 2022.

[9] Q. Zhang, L. Fu, and L. Gu, "A cascaded convolutional neural network for assessing signal quality of dynamic ECG," *Comput. Math. Methods Med.*, vol. 2019, p. 7095137, October 2019.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[11] L. Hu, W. Cai, Z. Chen, and M. Wang, "A lightweight u-net model for denoising and noise localization of ecg signals," *Biomedical Signal Processing and Control*, vol. 88, p. 105504, 2024.

[12] Y. Xia, C. Chen, M. Shu, and R. Liu, "A denoising method of ecg signal based on variational autoencoder and masked convolution," *Journal of Electrocardiology*, vol. 80, pp. 81–90, 2023.

[13] B. Xu, R. Liu, M. Shu, X. Shang, and Y. Wang, "An ecg denoising method based on the generative adversarial residual network," *Computational and Mathematical Methods in Medicine*, vol. 2021, p. 5527904, April 2021.

[14] N. Neifar, A. Ben-Hamadou, A. Mdhaffar, and M. Jmaiel, "Diffecg: A versatile probabilistic diffusion model for ecg signals synthesis," 2024.

[15] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," 2015.

[16] E. Adib, A. Fernandez, F. Afghah, and J. J. Prevost, "Synthetic ecg signal generation using probabilistic diffusion models," 2023.

[17] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," 2021.

[18] N. Neifar, A. Mdhaffar, A. Ben-Hamadou, and M. Jmaiel, "Deep generative models for physiological signals: A systematic literature review," 2023.

[19] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020.

[20] C. Camara, P. Peris-Lopez, M. Safkhani, and N. Bagheri, "ECG identification based on the gramian angular field and tested with individuals in resting and activity states," *Sensors (Basel)*, vol. 23, no. 2, p. 937, January 2023.

[21] F. Samann and T. Schanze, "An efficient ecg denoising method using discrete wavelet with savitzky-golay filter," *Current Directions in Biomedical Engineering*, vol. 5, no. 1, pp. 385–387, 2019.

[22] S. W. Smith, *The scientist and engineer's guide to digital signal processing*, 1st ed. San Diego, Calif.: California Technical Pub., 1997.

[23] V. Madisetti, "The digital signal processing handbook, second edition - 3 volume set," Boca Raton, 2009.

[24] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[25] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. E215–20, June 2000.

[26] G. B. Moody, W. E. Muldrow, and R. G. Mark, "A noise stress test for arrhythmia detectors," *Computers in Cardiology*, vol. 11, pp. 381–384, 1984.

[27] G. B. Moody and R. G. Mark, "A new method for detecting atrial fibrillation using R-R intervals," *Computers in Cardiology*, vol. 10, pp. 227–230, 1983.

[28] L. Jiang. (2021) "Image-Super-Resolution-via-Iterative-Refinement: Unofficial implementation of Image Super-Resolution via Iterative Refinement by Pytorch". https://github.com/Janspiry/Image-Super-Resolution-via-Iterative-Refinement. Accessed: April 30, 2024.

[29] J. Faouzi and H. Janati, "pyts: A python package for time series classification," *Journal of Machine Learning Research*, vol. 21, no. 46, pp. 1–6, 2020.

[30] The MathWorks Inc., "`dsp.LMSFilter`, compute output, error, and weights of least mean squares LMS adaptive filter," https://www.mathworks.com/help/dsp/ref/dsp.lmsfilter-system-object.html, Accessed: April 30, 2024.

[31] The MathWorks, Inc., "Window-based fir filter design; `fir1`," https://www.mathworks.com/help/signal/ref/fir1.html, Accessed: April 30, 2024.

Appendix

Figures 14, 15, 16 and 17 show the estimated clean ECG signals generated by the baseline methods (see Table V) and six-shot reconstructions of SR3 model 1 (ARDB trained) and model 2 (AF retrained) for the samples listed in Table VI.

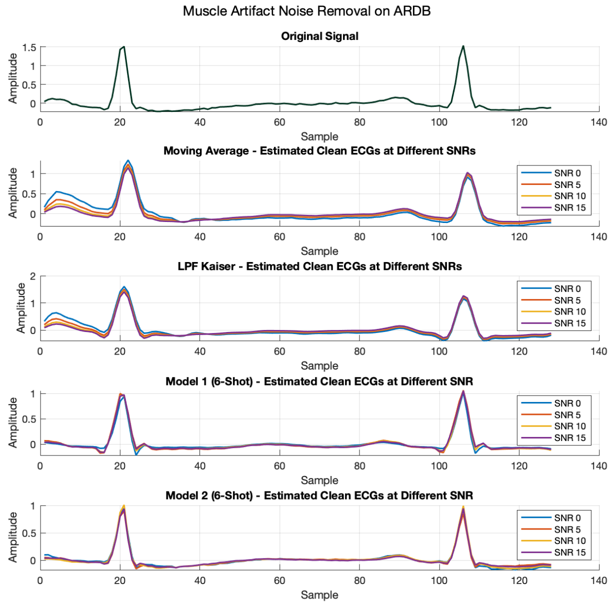

Fig. 14. Plots of estimated clean ECG signals for MA noise removal on ARDB at various SNRs.

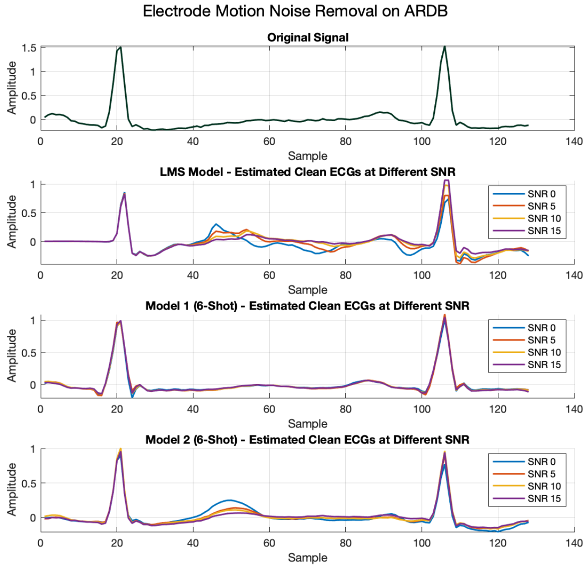

Fig. 15. Plots of estimated clean ECG signals for EM noise removal on ARDB at various SNRs.

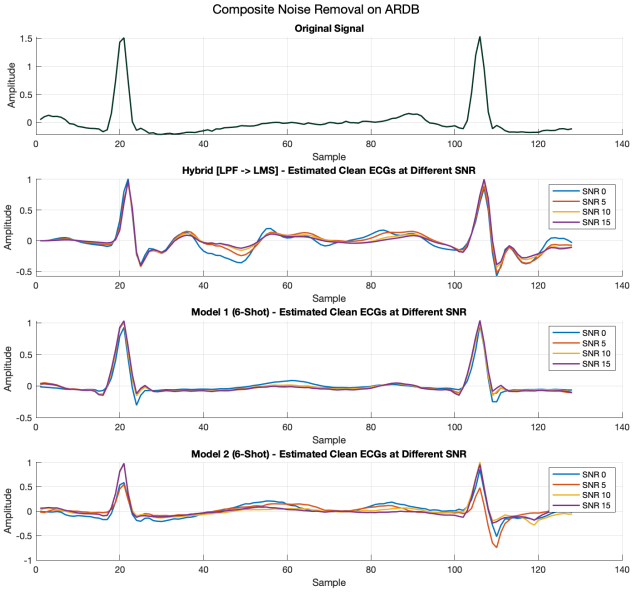

Fig. 16. Plots of estimated clean ECG signals for composite noise removal on ARDB at various SNRs.

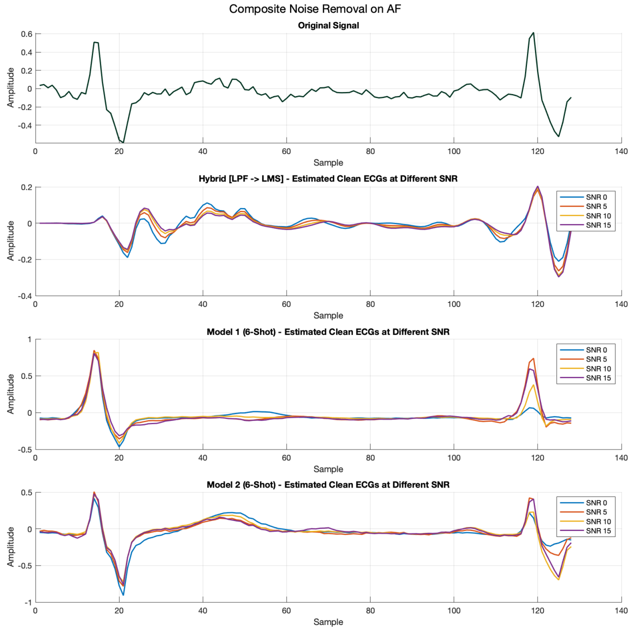

Fig. 17. Plots of estimated clean ECG signals for composite noise removal on AF at various SNRs.