Leveraging an ECG Beat Diffusion Model for Morphological Reconstruction from Indirect Signals

Lisa Bedin[*][†]
Ecole Polytechnique

Gabriel Cardoso[†]
Ecole Polytechnique, IHU Liryc

Josselin Duchateau
Bordeaux University Hospital, IHU Liryc

Remi Dubois
IHU Liryc

Eric Moulines
Ecole Polytechnique

Abstract

Electrocardiogram (ECG) signals provide essential information about the heart's condition and are widely used for diagnosing cardiovascular diseases. The morphology of a single heartbeat over the available leads is a primary biosignal for monitoring cardiac conditions. However, analyzing heartbeat morphology can be challenging due to noise and artifacts, missing leads, and a lack of annotated data. Generative models, such as denoising diffusion generative models (DDMs), have proven successful in generating complex data. We introduce `BeatDiff`, a lightweight DDM tailored for the morphology of multiple leads heartbeats. We then show that many important ECG downstream tasks can be formulated as conditional generation methods in a Bayesian inverse problem framework using `BeatDiff` as priors. We propose `EM-BeatDiff`, an Expectation-Maximization algorithm, to solve this conditional generation tasks without fine-tuning. We illustrate our results with several tasks, such as removal of ECG noise and artifacts (baseline wander, electrode motion), reconstruction of a 12-lead ECG from a single lead (useful for ECG reconstruction of smartwatch experiments), and unsupervised explainable anomaly detection. Experiments show that the combination of `BeatDiff` and `EM-BeatDiff` outperforms SOTA methods for the problems considered in this work.

1 Introduction

Electrocardiograms (ECG) are essential tools for diagnosing cardiac conditions. Two main types of diagnostics can be obtained from an ECG signal: rhythm-based and morphology-based. Rhythm-based diagnostics focus on the frequency and regularity of heartbeats, while morphology-based diagnostics focus on the shape and amplitude of the various waves and segments of the ECG signal; see [40]. Many critical cardiac conditions can be diagnosed by analyzing the morphology of the different phases of a single beat ([66, 67, 20, 39]). For example, an increase in the ST segment suggests a myocardial infarction ([93]), while a long QT syndrome is associated with an increased risk of sudden death ([92]). In fact, patients who have survived events similar to sudden death often have abnormal intracardiac signals, even during sinus rhythm; see [34].

Most generative models for ECG literature attempt to accurately represent the rhythm, i.e. the time at which the individual ECG events occur; see e.g., [30, 99, 79, 23, 104, 100, 3]. We focus in this paper on the morphology of a single heartbeat.

Code available at `https://github.com/LisaBedin/BeatDiff`
Pretrained models available at `https://huggingface.co/lbedin/BeatDiff`
*Correspondence: `lisa.bedin@polytechnique.org` †Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

The standard for ECG recording systems includes 12 leads, which are obtained using 9 electrodes. The use of multiple leads is required because pathologies may manifest only in one lead. For example, T wave inversions—which can indicate Arrhythmogenic Right Ventricula Dysplasia—might only appear on one or two precordial leads ([102]). Furthermore, the specific precordial lead in which a disease is detected has an important diagnostic value about the disease location in the heart ([52]).

Multi-lead ECGs can be affected by noise and artefacts, i.e. unwanted signals caused primarily by variations in potential and impedance at the electrode-skin interface, but also by other factors such as environmental interference, movement of the subject; see e.g., [101, 47, 14, 56] and the references therein. These artefacts overlap in the spectral range of interest and manifest as morphological features that resemble inherent aspects of the ECG or disease-specific aspects. Therefore, methods that can accurately reconstruct 12-lead ECGs from partial observations are central to analysing the morphology of heartbeats. In this work, we address classical problems such as baseline wander or motion artefacts; we are also interested in the reconstruction of missing leads to be able to reconstruct the 12-lead representation from a reduced number of electrodes, which is an essential precursor for the reconstruction of 12-lead ECGs from smartwatch measurements; see [90].

The heartbeat morphology reconstruction problems are naturally formulated as Bayesian linear inverse problems ([87, 38, 21]). The observation vector is an affine transformation of the signal of interest, which is influenced by additive noise. This affine function is only partially known (it may depend on unknown parameters) and the corresponding inverse problem is usually ill-posed (in the missing lead case, the affine function is not invertible). Inverse Bayesian problems require the use of a prior distribution for the signal to be reconstructed. Recently, the use of generative models to define priors has enabled us to achieve many successes in various areas.; see e.g.,[5, 57, 64, 96]. Early works in this area used flow models or GANs. There has been a recent increase in interest in using diffusion models as a prior in Bayesian inverse problems ([83, 17, 85, 45, 46, 13, 98]). Many techniques have been recently proposed. We focus here on `MCGDiff` ([13]), which constructs unbiased estimates of posterior distributions using Sequential Monte Carlo (SMC) methods, a.k.a. particle filter.

Our main contributions are as follow

- We introduce `BeatDiff` a new light-weight DDM model designed to generate 12-leads heart beat morphology. In comparison to [3], `BeatDiff` has a significanty lower memory footprint and faster generation speed. `BeatDiff` has shown superior performance to state-of-the-art ECG generation methods accross all the metrics that we have considered.
- We then show how `BeatDiff` can be used as a prior to address various challenges in heartbeat morphology reconstruction from partial observations. We show how the `MCGDiff` method can be combined with Monte Carlo Expectation Maximization (MCEM) algorithm to compute maximum likelihood estimate of the unknown parameters of the inverse linear model (e.g., noise level estimation, noise and artifact model, etc.), leading to a new full-fledged algorithm for conditional ECG generation, called `EM-BeatDiff`.
- We demonstrate the effectiveness of our approach by comparing it to state-of-the-art methods. Our algorithm outperforms the current best approaches on multiple evaluation metrics specifically designed for ECGs, and offers new paths that have the potential to lead to novel applications.

Related works: The use of generative models ([50, 51, 33]) as informative priors in solving Bayesian inverse problems has attracted significant interest ([4, 96, 88, 42, 77, 103, 74]). In particular, DDMs have been demonstrated as a particularly suitable choice of prior for solving inverse problems ([83, 17, 85, 45, 46]). DDMs are generative models that transform a simple reference distribution into the training data distribution through a denoising process called denoising diffusion. These models are capable of generating high-quality realistic samples on par with the best Generative Adversarial Networks (GANs) ([32]) in terms of image and audio generation, without the intricacies of adversarial training ([81, 86, 83, 84, 9]). In this article, we follow the approach proposed in [13, 98], for sampling solutions to an inverse problem using a Sequential Monte Carlo (SMC) algorithm that guides the denoising process of a pretrained diffusion model. This method is accompanied by a series of theoretical guarantees in realistic scenarios.

Generative modeling, denoising methods, and automatic anomaly detection algorithms are commonly used for ECG analysis. In particular, DDMs have been demonstrated to be capable of generating realistic ECGs: [2] focuses on generating a single healthy beat for a single ECG lead, [3] generates a

10-second period conditioned on various complementary ECG information. Additionally, numerous methods address the denoising problem in ECGs; see e.g., [79, 56, 15]. Classical approaches like Dower matrices ([59]) are used to reconstruct missing leads in ECGs. [97, 43] rely on neural networks to detect anomalies, and [76] use adversarial autoencoders for unsupervised anomaly detection. However, to our knowledge, there is no method that addresses all these problems with a single pretrained model.

2 `BeatDiff` - a generative model for heartbeat morphology

Denoising Diffusion Generative Models (DDM): We briefly describe in this section the DDMs and introduce some basic notations which are required below; see [24, 37, 86, 83, 85, 44, 19] and the references therein for theory and practical implementation details. We focus on the variance-exploding (VE) framework ([86]). In the *forward path* an initial state X_0 is sampled from the data distribution $\mathsf{q}_{\mathrm{data}}$ and independent Gaussian noise with zero-mean and increasing variance is added to generate subsequent states $X_k = X_{k-1} + \rho_k \varepsilon_k$, where $k \in \mathbb{N}^*$, $\rho_k > 0$, and $\varepsilon_k \sim \mathcal{N}(0, \mathrm{I})$. The joint p.d.f. of the Markov chain is $\mathsf{q}_{0:K}(x_{0:K}) = \mathsf{q}_{\mathrm{data}}(x_0) \prod_{k=1}^{K} q_k(x_k | x_{k-1})$, where $q_k(\cdot | x_{k-1}) = \mathcal{N}(x_{k-1}, \rho_k^2 \mathrm{I})$ and $K \in \mathbb{N}^*$. The conditional distribution of X_k given X_s with $k > s \geq 0$ is given by $\mathsf{q}_{k|s}(\cdot | x_s) = \mathcal{N}(x_s, (\upsilon_k^2 - \upsilon_s^2) \mathrm{I})$ with $\upsilon_k^2 = \sum_{j=1}^{k} \rho_j^2$ (and $\upsilon_0^2 = 0$). The number of forward steps K is chosen such that $\upsilon_K^2 = \upsilon_{\mathrm{max}}^2$ is far larger than the variance of $\mathsf{q}_{\mathrm{data}}$. With such choice, $\mathsf{q}_{K|0}(\cdot | x_0)$ is close to the reference distribution $q_{\mathrm{ref}} = \mathcal{N}(0, \upsilon_{\mathrm{max}}^2 \mathrm{I})$. We learn for each state X_k a denoiser $\mathcal{D}_{0|k}^{\varphi}$ with parameters φ trained to minimize $\mathcal{L}_{\mathcal{D}}(\varphi) := \sum_{k=1}^{K} \gamma_k^2 \mathbb{E}_{X_0 \sim \mathsf{q}_{\mathrm{data}}, \epsilon \sim \mathcal{N}(0, \mathrm{I})} \left[\| \mathcal{D}_{0|k}^{\varphi}(X_0 + \upsilon_k \epsilon, \upsilon_k) - X_0 \|^2 \right]$, where $\{\gamma_k\}_{k \in [1:K]}$ is a sequence of appropriately defined positive weights. We denote the result of this minimization as φ^*. In the backward path, we sample $x_K \sim q_{\mathrm{ref}}$ and for $k = K$ to $k = 2$ we sample x_{k-1} given x_k with

$$p_{k-1|k}(x_{k-1} | x_k) = \mathcal{N}\left(x_{k-1}; \boldsymbol{\mu}_{k-1}(x_k, \mathcal{D}_{0|k}^{\varphi^*}(x_k, \upsilon_k)), \eta_{k-1}^2 \, \mathrm{I}_d \right)$$

where the variances are hyperparameters $\eta = \{\eta_k\}_{k \in \mathbb{N}}$ satisfying $\eta_k^2 \leq \upsilon_k^2$ and $\boldsymbol{\mu}_{k-1}(x_k, x_0) := x_0 + (\upsilon_{k-1}^2 / \upsilon_k^2 - \eta_{k-1}^2 / \upsilon_k^2)^{1/2} (x_k - x_0)$. Finally, we sample $x_0 \sim p_0(\cdot | x_1) := \mathcal{N}(\mathcal{D}_{0|1}^{\varphi^*}(x_1, \upsilon_1), \eta_0^2 \, \mathrm{I})$. To keep the notations simple, we remove in the sequel the dependence in η and φ^*. For $k \in [0 : K-1]$, we denote by $\mathsf{p}_k(x_k)$ the marginal distribution of X_k:

$$\mathsf{p}_k(x_k) := \int q_{\mathrm{ref}}(x_K) \prod_{s=K}^{k+1} p_{s-1|s}(x_{s-1} | x_s) \mathrm{d}x_{k+1:K} .$$

`BeatDiff` **model:** In the standard ECG, the augmented limb leads (AVL, AVR, AVF) can be obtained from a known linear combination of the limb leads (I, II, III) ([59, Vol 1, Chapter 11]). Hence, it is standard practice to select either the augmented leads or the limb leads to model the ECG ([3, 35]). We exclude the augmented leads and use the leads (I, II, III, V1–6). We denote by $L = 9$ the number of leads, and by T the maximal heartbeat duration (expressed in number of samples).

Various factors, including age (A), sex (S) and the RR interval, which is the reciprocal of heart rate, influence the morphology of the heartbeat; see e.g., [60, 75, 6]. Therefore, we use the DDM described above to approximate the distribution $\mathsf{q}_{\mathrm{data}}$ of heartbeats over the retained leads *conditionally* on the patient characteristics $\mathcal{P} := (\mathrm{A}, \mathrm{S}, \mathrm{RR})$. The denoiser of `BeatDiff` $\mathcal{D}_{0|k}^{\varphi}$ takes as input $(x, \upsilon_k, \mathrm{e}_{\mathcal{P}})$ where x is the $L \times T$ matrix of single heartbeat samples, υ_k is the k-th step diffusion variance and $\mathrm{e}_{\mathcal{P}}$ encodes the patient features. We obtain $\mathrm{e}_{\mathcal{P}}$ from $\mathcal{P} = (\mathrm{A}, \mathrm{S}, \mathrm{RR})$ by first one-hot-coding the Boolean variable S and then embedding it using a fully connected 2-layer network. For υ_k we use the Fourier positional encoding ([91]) of $\log(\upsilon_k)$ as in [24]. For $\mathcal{D}_{0|k}^{\varphi}$ we use a modified 1d Unet, the specific details are given in Appendix B.1.4. The model has 10^6 parameters. Compared to [3], the inference time is 400 times shorter and the memory footprint is 900 times smaller.

`BeatDiff` **training** We utilize the PhysioNet Challenge dataset ([31, 68, 69]), comprising 43,101 12-lead ECGs. The pre-processing of [8] is used which consists of normalization of the sampling frequency, detection of R peaks to identify heartbeats, segmentation of the heartbeats. We obtain 214,460 single-beat ECGs, each with $T = 176$ samples and $L = 9$ (I, II, III, V1–V6). See Appendix B.1.1 for details. Each patient (and *the entirety of its recordings*) is attributed to one of the three datasets: Training, Cross-validation (CV) or Test. During training, a batch of size b

is constituted by firstly drawing b patients and then selecting randomly one of the beats for each given patient. For testing and cross-validation, due to the significant variability between patients in comparison to the variability between heartbeats, we randomly select a single beat per patient for model evaluation. The entire network $\mathcal{D}^{\varphi}_{0|k}$ is trained to minimize $\mathcal{L}_{\mathcal{D}}$ using the Adam optimizer ([49]) with a batch-size 2^{10} on the healthy training set, and the best model in terms of $\mathcal{L}_{\mathcal{D}}$ over the cross-validation set is retained. See Appendix B.1.4 for details.

3 `EM-BeatDiff` - conditional heartbeat generation from indirect measurements

We present `EM-BeatDiff` a method that allows us to sample heartbeat morphology from partial observations, focusing on a class of problems that can be formulated as Bayesian linear inverse problems. Our approach is based on Monte-Carlo guided diffusion (`MCGDiff`), introduced in [13] (see also [89, 65]), which is used in combination with `BeatDiff`.

Monte Carlo Guided Diffusion (`MCGDiff`): In many applications of interest, the objective is to sample from a distribution $\phi_0(x_0) := g_0(x_0)\mathsf{p}_0(x_0)/\mathcal{Z}$, where g_0 is a nonnegative potential function, p_0, the marginal of the diffusion model at time 0, and $\mathcal{Z} := \int g_0(x)\mathsf{p}_0(x)\mathrm{d}x$ is the normalizing constant. For example, in a Bayesian setting, g_0 is the likelihood function (the conditional distribution of the observation given the current value of the state x_0) and $\mathsf{p}_0(x_0)$ is the prior distribution of the state. In such case, $\phi_0(x_0)$ is the posterior distribution of the state x_0 given the current observation. A simple idea for sampling the posterior $\phi_0(x_0)$ is to use sampling importance resampling (SIR, [72]), where p_0 is used as the instrumental distribution. However, this method may be inefficient since the instrumental distribution neglects the potential g_0.

We define a distribution over the path space

$$\phi_{0:K}(x_{0:K}) := \mathcal{Z}^{-1} g_0(x_0) \textstyle\prod_{k=1}^{K} p_{k-1|k}(x_{k-1}|x_k) q_{\mathrm{ref}}(x_K).$$

In [13], a sequence of positive intermediate potentials $\{g_k\}_{k\in[1:K]}$ with $g_K \equiv 1$ was introduced to guide the backward Markov chain to regions where the potential g_0 is large. The path space distributions may be equivalently rewritten as

$$\begin{aligned}\phi_{0:K}(x_{0:K}) &\propto q_{\mathrm{ref}}(x_K) \textstyle\prod_{k=1}^{K} \frac{g_{k-1}(x_{k-1})}{g_k(x_k)} p_{k-1|k}(x_{k-1}|x_k)\\ &\propto q_{\mathrm{ref}}(x_K) \textstyle\prod_{k=1}^{K} \omega_k(x_k)\hat{p}_{k-1|k}(x_{k-1}|x_k)\,, \end{aligned} \tag{3.1}$$

where, for $k \in [1:K]$, $\hat{p}_{k-1|k}(\cdot|x_k) := g_{k-1}(\cdot)p_{k-1|k}(\cdot|x_k)/\mathcal{Z}_k(x_k)$, and $\mathcal{Z}_k(x_k) := \int g_{k-1}(x')p_{k-1|k}(x'|x_k)\mathrm{d}x$, and $\omega_k(x_k) := \mathcal{Z}_k(x_k)/g_k(x_k)$. We implicitly assume that these formulas have a closed form. Sampling according to (3.1) passes through the intermediate distributions $\phi_k(x_k) := \int q_{\mathrm{ref}}(x_K) \prod_{s=k+1}^{K} \omega_s(x_s)\hat{p}_{s-1|s}(x_{s-1}|x_s)\mathrm{d}x_{k+1:K}$ for each $k \in [1:K]$, which verifies

$$\phi_{k-1}(x_{k-1}) \propto g_{k-1}(x_{k-1})\mathsf{p}_{k-1}(x_{k-1}) \propto \textstyle\int \omega_k(x)\hat{p}_{k-1|k}(x_{k-1}|x)\phi_k(x)\mathrm{d}x\,. \tag{3.2}$$

Each ϕ_{k-1} thus has the same structure as ϕ_0: a product of a potential function and the marginal law at time $k-1$ of the backward diffusion.

It remains to approximate this sequence of distributions. For this purpose, we use Sequential Monte Carlo (SMC); see [25, 16]. Suppose that we have at iteration k a *particle approximation* $\phi^M_k = M^{-1}\sum_{j=1}^{M}\delta_{\xi^j_k}$ of ϕ_k through a set of $M \in \mathbb{N}_{>0}$ particles $\xi^{1:M}_k$, initialized with $\xi^{1:M}_K \sim q_{\mathrm{ref}}^{\times M}$. Plugging this approximation into Equation (3.2) gives

$$\phi^{\cdot}_{k-1} \propto \textstyle\sum_{j=1}^{M} \omega_k(\xi^j_k)\hat{p}_{k-1|k}(\cdot|\xi^j_k)\,.$$

Hence, to obtain $\xi^{1:M}_{k-1}$, we first sample M ancestors according to $I^{1:M}_{k-1} \sim \mathrm{Cat}\big(\{\omega_k(\xi^j_k)/\sum_{i=1}^{M}\omega_k(\xi^i_k)\}_{j=1}^{M}\big)^{\times M}$, then we sample new particles $\xi^{1:M}_{k-1} \sim \{\hat{p}_{k-1|k}(\cdot|\xi^{I^j_{k-1}}_k)\}_{j=1}^{M}$, leading to $\phi^M_{k-1} = M^{-1}\sum_{j=1}^{M}\delta_{\xi^j_{k-1}}$. Algorithm is given in Appendix A.1.1.

Bayesian inverse problems We assume that the $d_y \times 1$ vector of observations Y (noisy/partial heartbeat) is given by

$$Y = A_\theta X_0 + b_\theta + D_\theta \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathrm{I}), \quad X_0 \sim \mathsf{p}_0 \tag{3.3}$$

where A_θ is a $d_y \times d_x$ matrix (for selecting the lead/time observed indices), b_θ is a $d_y \times 1$ vector (modeling hearbeat artifacts; e.g., baseline wander, electrode motion), D_θ is a $d_y \times d_y$ invertible matrix (the variance of the noise), and $\theta \in \Theta$ is a vector of unknown parameters. Define by $g_0^{\theta,y}(x_0)$ the likelihood of the observation, given by $g_0^{\theta,y}(x_0) := \mathcal{N}(y; A_\theta x + b_\theta, D_\theta D_\theta^\top)$. Given an observation y and a value of the parameter θ, we may sample X_0 from the posterior $X_0|y, \theta$, with p.d.f. $\phi_0^{\theta,y}(x_0) = \mathsf{p}_0(x_0) g_0^{\theta,y}(x_0)/\mathcal{Z}^{\theta,y}$, $\mathcal{Z}^{\theta,y} = \int g_0^{\theta,y}(x)\mathsf{p}_0(x)\mathrm{d}x$ is the normalizing constant. We use `MCGDiff` with the intermediate potentials $\{g_k^{\theta,y}\}_{k\in[0:K]}$ defined as

$$g_k^{\theta,y}(x) = \mathcal{N}(y; A_\theta x + b_\theta, \Sigma_{k,\theta}), \tag{3.4}$$

where the sequence of covariance matrices $\Sigma_{k,\theta}$ are specified in Appendix A.2.1. For this choice of potentials, $\hat{p}_{k-1|k}^{\theta,y}$ and $\omega_k^{\theta,y}$ admit closed forms given in Appendix A.2.2.

`MCGDiff` allows to sample $X_0|y, \theta$ for a known parameter θ. When θ is unknown, we maximize the penalized marginal log-likelihood

$$\theta^* = \operatorname*{argmax}_{\theta \in \mathbb{R}^d} \left(l(\theta) + \mathrm{Pen}(\theta)\right), \quad l(\theta) := \log \mathcal{Z}^{\theta,y} = \log \int g_0^{y,\theta}(x)\mathsf{p}_0(x)\mathrm{d}x \tag{3.5}$$

where $\mathrm{Pen}(\theta)$ is a penalty. The best-known method for optimizing the marginal log-likelihood (3.5) is the expectation maximization algorithm (EM); see [61]. The EM iterates between two main steps: expectation (E) and maximization (M). Starting from an initial guess θ_0, the EM algorithm alternates between: (E) compute the surrogate function $\mathrm{Q}(\theta; \theta_i) := \int \log g_0^{y,\theta}(x_0)\phi_0^{\theta_i,y}(x_0)\mathrm{d}x_0$; and (M) solve for $\theta_{i+1} := \arg\max_{\theta\in\Theta} \mathrm{Q}(\theta; \theta_i) + \mathrm{Pen}(\theta)$. Under general conditions, the sequence of parameter estimates $(\theta_i)_{i\in\mathbb{N}}$ converges to a stationary point θ_* of the marginal penalized likelihood; see [61, Chapter 3]. In this setting, the E-step is untractable; we approximate the surrogate function in the (E) step using `MCGDiff` with the current parameter θ_i and the sequence of intermediate potentials defined in (3.4). Such scheme becomes a specific instance of the Monte Carlo EM algorithm (MCEM), initially introduced in [95] and further analyzed in [54, 26, 53].

The `EM-BeatDiff` algorithm: We combine the `BeatDiff` for the prior and `MCGDiff` algorithms for posterior sampling, with MCEM steps for parameter inference. The only slight difference is that the observations are gathered in a matrix $\mathbf{Y}$ of size $\tilde{L} \times \tilde{T}$ - where $\tilde{L}$ is the number of observed leads and $\tilde{T}$ the number of observed samples on each lead. The state we are attempting to reconstruct is a matrix of size $L \times T$. The observation equation takes the form

$$Y = A_\theta X_0 \bar{A}_\theta + B_\theta + D_\theta \epsilon \bar{D}_\theta,$$

where (A_θ, D_θ) and $(\bar{A}_\theta, \bar{D}_\theta)$ are $\tilde{L} \times L$ and $T \times \tilde{T}$ matrices, B_θ is a $\tilde{L} \times \tilde{T}$ matrix, and ϵ is a $L \times T$ matrix with i.i.d. standard Gaussian entries. The model (3.3) is obtained by applying the vectorization operator to $\mathbf{Y}$. The full algorithm is detailed in Appendix A.3. Note that `EM-BeatDiff` is also applicable to "standard" ECG signals and could be used in combination with the ECG generative model of [3], for example.

4 Experimental validation

`BeatDiff` **evaluation:** We begin by assessing the impact of `BeatDiff` in a classifier improvement task. This involves comparing the performance of a classifier trained on a severely unbalanced dataset with that of the same classifier trained on a balanced dataset. The balancing is achieved by augmenting the minority class with new examples from a generative model. Results for sex classification from heartbeat are reported in Table 1, including model size, inference time, F1 score, total accuracy, and AUC score are shown for the balanced dataset with `BeatDiff` and alternative ECG generation models from [1, 3], retrained to generate ECGs conditioned on sex. Diffusion-based models `BeatDiff` and SSDM ([3]) outperform WGAN ([1]), with `BeatDiff` being 400 times faster than [3]. See implementation details in Appendices B.3.1, B.3.2 and B.4.

Table 1: Evaluation of ECG generation models for balancing sex-imbalanced datasets in heartbeat classification task. F and M refer to the number of female and male real heartbeats in the training set. Confidence intervals are obtained by re-initializing the classifier training and the generated data used to balance the datasets.

Model	Size	Inference	F = 10%M			F = 5%M		
	(Mb, ↓)	Time (ms, ↓)	F1 (%, ↑)	Acc. (%, ↑)	AUC (%, ↑)	F1 (%, ↑)	Acc. (%, ↑)	AUC (%, ↑)
SSDM [3]	39×10^3	7.5×10^4	76 ± 0	69 ± 1	77 ± 1	74 ± 0	64 ± 0	71 ± 1
WGAN [1]	27	3.8×10^{-2}	76 ± 0	69 ± 1	77 ± 1	74 ± 0	64 ± 1	72 ± 1
`BeatDiff`	42	1.6×10^2	$\mathbf{78 \pm 0}$	$\mathbf{73 \pm 1}$	$\mathbf{79 \pm 1}$	$\mathbf{77 \pm 1}$	$\mathbf{72 \pm 1}$	$\mathbf{76 \pm 1}$
Unbalanced	-	-	76 ± 0	69 ± 1	74 ± 3	74 ± 1	64 ± 1	70 ± 3
Balanced	-	-	82 ± 0	81 ± 0	86 ± 1	82 ± 0	81 ± 0	86 ± 1

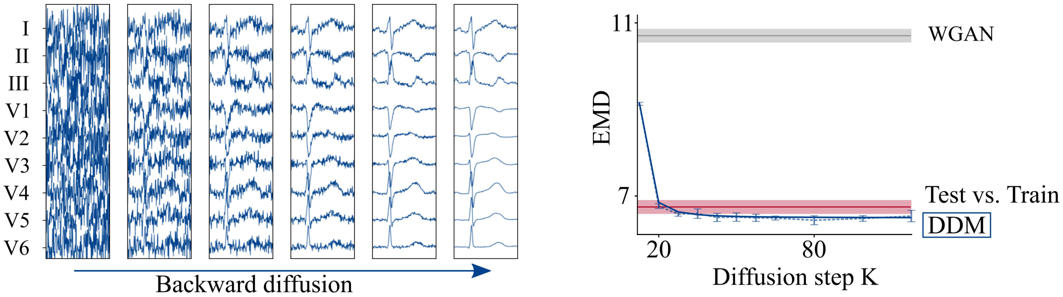


Figure 1: *Left:* heartbeat generation along backward diffusion steps. *Right:* EMD between generated ECG distribution and real ECG distribution. EMD vs. test (resp. train) in plain (resp. dotted) line. EMD for DDM with different number of diffusion steps, in blue. DDM for WGAN model in gray. EMD between test and train distributions in red. Error bars correspond to different training batches of size 2864.

We also use the L2-Earth Mover's Distance (EMD) ([29]) to evaluate the dissimilarity between the predicted and target distributions, excluding SSDM due to computational limitations. The EMD is computed from the generated set for both the test set and batches of the training set of the same size. Our results in figure 1 show that a few diffusion steps are sufficient to generate an accurate prediction distribution, with `BeatDiff` performing better than [1] in replicating the real data distribution. In Appendix B.6.1 we present a third evaluation of the generated ECGs' quality using the out-of-distribution score proposed in [18].

Prediction of Corrected QT: Both the EMD and Classifier Enhancement tasks are concerned with how different the generated ECGs are from the ECGs in the dataset. We are now focusing on the question "Is the algorithm able to correctly capture underlying physiological mechanism?". To do so, we evaluate `EM-BeatDiff` on the prediction of corrected QT, which is an important clinical indicator obtained from the ECG; see [7].

The QT interval is the duration between the Q wave, which marks the beginning of ventricular depolarization, and the end of the T wave, which signifies the completion of ventricular repolarization. This interval depends on heart rate: as the heart rate increases (and the RR interval decreases), the QT interval tends to shorten. Understanding the relationship between the RR and QT intervals is crucial for diagnosing and managing various cardiac conditions. For instance, a prolonged QT interval may indicate an increased risk of arrhythmias, such as Torsades de Pointes, while a shortened QT interval can be associated with conditions like hypercalcemia. Moreover, certain medications are known to prolong the QT interval; see [58].

We use `EM-BeatDiff` to generate the T-Wave from a given patient QRS complex (the sequence of waves (Q, R, S) with negative, positive, and negative deflections, respectively) and heart rate (RR). Each test ECG is trimmed to focus solely on the QRS complex. Then, for RR values ranging from 0.6 s to 1.2 s, or equivalently for heart rates ranging from 43 to 100 beats per minute, we sample x from the conditional distribution of the ECGs over all leads given the RR and the observed QRS as illustrated in figure 2. The configuration of the related inverse problem is given in Table 2 under the name QT.

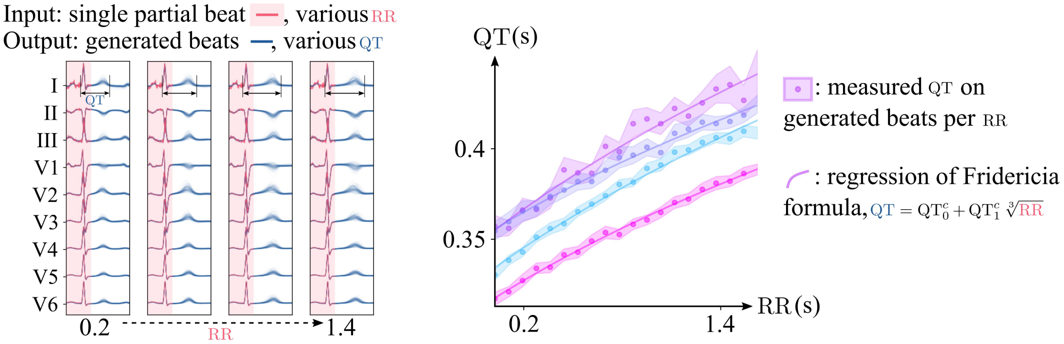


Figure 2: *Left:* Example of T-wave prediction (blue) conditioned on Q-wave (red) for different value of RR. *Right:* QT as a function of RR for 4 patients. QT measured in 100 generated samples (resp. regressed with Fridericia formula) displayed in dots with 95%-CLT bars (resp. curve).

Table 2: Configurations used for `EM-BeatDiff` for each task.

Task	$(\bar{L}, \bar{T})$	θ	A_θ	$\bar{A}_\theta$	B_θ	D_θ	$\bar{D}_\theta$
QT	$(L, 70)$	$\sigma_{1:L}$	$\mathrm{I}_{L\times L}$	$\mathrm{I}_{T\times\bar{T}}$	$\mathbf{0}_{L,\bar{T}}$	$\mathrm{diag}(\sigma_{1:L})$	$\mathrm{I}_{T\times\bar{T}}$
AR	(L, T)	$(\sigma_{1:L}, \vartheta_{1:K,1:L})$	$\mathrm{I}_{L\times L}$	$\mathrm{I}_{T\times T}$	(4.1)	$\mathrm{diag}(\sigma_{1:L})$	$\mathrm{I}_{T\times T}$
ML (SW)	$(1, T)$	σ	$\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times T}$	$\mathbf{0}_{\bar{L},T}$	$\sigma\,\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times T}$
ML (V1-6)	$(3, T)$	$\sigma_{1:\bar{L}}$	$\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times T}$	$\mathbf{0}_{\bar{L},T}$	$\mathrm{diag}(\sigma_{1:\bar{L}})\,\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times T}$
AD (MI)	$(3, T)$	$(\sigma_{1:\bar{L}}, \vartheta_{1:K,1:\bar{L}})$	$\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times T}$	(4.1)	$\mathrm{diag}(\sigma_{1:\bar{L}})\,\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times T}$
AD (LAE)	$(3, T)$	$(\sigma_{1:\bar{L}}, \vartheta_{1:K,1:\bar{L}})$	$\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times T}$	(4.1)	$\mathrm{diag}(\sigma_{1:\bar{L}})\,\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times T}$
AD (LAD)	$(L, 106)$	$\sigma_{1:L}$	$\mathrm{I}_{L\times L}$	$\left[\mathbf{0}_{\bar{T},T-\bar{T}}; \mathrm{I}_{\bar{T}\times\bar{T}}\right]^T$	(4.1)	$\mathrm{diag}(\sigma_{1:L})\,\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times\bar{T}}$
AD (LQT)	$(L, 70)$	$\sigma_{1:L}$	$\mathrm{I}_{L\times L}$	$\mathrm{I}_{T\times T}$	$\mathbf{0}_{L,\bar{T}}$	$\mathrm{diag}(\sigma_{1:L})\,\mathrm{I}_{L\times L}$	$\mathrm{I}_{T\times\bar{T}}$

To evaluate this, we rely on well-known empirical formulas from [7, 27, 73]. These formulas introduce coefficients called “corrected QT” denoted as QT_0^c and QT_1^c, which depend on the patient and are determined from ECGs measured during stress test. We regress the intercept QT_0^c and slope QT_1^c of the Fridericia formula from [27], which states that $\mathrm{QT} = \mathrm{QT}_0^c + \mathrm{QT}_1^c \sqrt[3]{\mathrm{RR}}$, from the generated curves. As shown in figure 2, we observe a consistent trend between the observed and regressed curves for four patients. Additionally, Table 10 indicates a high R^2-score of 0.98 between observed and expected QT curves in the test set. `EM-BeatDiff` generates QT for different RR that follow the Fridericia formula, one of the most correlated with patient QT vs. RR behaviour, without explicitly encoding this relationship during training. This demonstrates the ability of `EM-BeatDiff` to capture underlying physiological mechanisms.

Artifact removal (AR): Many solutions for removing ECG artifacts such as baseline wander–a low-frequency artifact caused mainly by respiration and body movements– or electrode motion–also a low-frequency artifact caused by bad electrode contact– have been proposed so far, most often based on adaptive filter, time-frequency (and most notably empirical mode decomposition) and time-scale decomposition; see [22, 94, 101, 55, 14] and the references therein. We use in this experiment a sparse representation of the artifacts in a dictionary, and propose to use penalized MLE with DDM prior to estimate the artifacts and denoise the ECG. In this case, we set for a given $\bar{L} \in \{1, L\}$ and $\bar{T} \in \{1, T\}$

$$B_\theta = [b_1^\theta, \ldots, b_{\bar{L}}^\theta]^\top \quad \text{with} \quad b_{\ell,t}^\theta = \textstyle\sum_{i=1}^K \vartheta_{j,\ell} c_j(t) \quad \text{for} \quad \ell, t \in [1:\bar{L}] \times [1:\bar{T}]\,, \tag{4.1}$$

with $\{c_j\}_{j=1}^J$ be a known set of functions (such as B-splines, a Fourier or a wavelet basis). We choose a Fourier basis in the experiments as expressed in Equation (B.1). The other parameters are given in Table 2 under the name AR.

We therefore remove the artifacts by subtracting a vector assumed to have a sparse representation on an appropriate basis. We use the sparse group LASSO penalty defined as

$$\mathrm{Pen}(\theta) = \lambda_1 \textstyle\sum_{j=1}^J \left(\sum_{\ell=1}^L (\vartheta_{j,\ell})^2\right)^{1/2} + \lambda_2 \sum_{j=1}^J \sum_{\ell=1}^L |\vartheta_{j,\ell}| \tag{4.2}$$

which leads to parsimony at both the group and individual levels, in order to promote the selection of the same functions (e.g. Fourier frequency) over all the leads; see [28, 78]. The first term promotes

Table 3: Evaluation of several reconstruction metrics for the AR task on beats corrupted with artifacts from MIT-BIH database from [62], with 95%-CLT intervals over the test-set.

	Baseline Wander			Electrode Motion		
Method	SSD (↓)	MAD (↓)	Cos. (×100, ↑)	SSD (↓)	MAD (↓)	Cos. (×100, ↑)
DeScoD [3]	4.37 ± 8.19	0.31 ± 0.15	95.20 ± 0.36	0.27 ± 0.01	0.27 ± 0.01	92.73 ± 0.25
EM-BeatDiff	$\mathbf{0.14 \pm 0.01}$	$\mathbf{0.24 \pm 0.02}$	$\mathbf{96.69 \pm 0.22}$	$\mathbf{0.18 \pm 0.01}$	$\mathbf{0.26 \pm 0.01}$	$\mathbf{95.42 \pm 0.19}$

Table 4: Evaluation of ECG generation models for the missing lead retrieval task, with 95%-CLT intervals over the test-set.

	Smartwatch			V1–6		
Method	SSD (↓)	MAD (↓)	Cos. (×100, ↑)	SSD (↓)	MAD (↓)	Cos. (×100, ↑)
EkGAN [41]	1.63 ± 0.47	0.36 ± 0.03	$\mathbf{91.33 \pm 0.38}$	2.10 ± 0.83	$\mathbf{0.35 \pm 0.03}$	$\mathbf{93.42 \pm 0.00}$
EM-BeatDiff	$\mathbf{1.03 \pm 0.05}$	$\mathbf{0.35 \pm 0.01}$	86.02 ± 0.99	$\mathbf{1.10 \pm 0.06}$	0.36 ± 0.01	87.78 ± 0.98

group sparsity: it keeps or removes the projections of observations on c_j across all leads. The second term promotes global sparsity. Details are given in Appendix B.2.2.

The evaluation of `EM-BeatDiff` in the AR task on 12-lead ECGs contaminated with per-lead independent noise from the MIT-BIH Noise Stress Test database ([62]) is presented in Table 3. Despite not being exposed to MIT-BIH Noise during training, `EM-BeatDiff` outperforms DeScoD ([56]) –a conditional DDM specifically trained to remove baseline wander– according to the following metrics: Sum of the square of the distances (SSD), Absolute maximum distance (MAD), and Cosine similarity (Cos.) ([63], Appendix B.5). Visualizations of ECGs obtained with `EM-BeatDiff` and DeScoD are provided in figure 3, and failure cases of DeScoD are discussed in Appendix B.6.4.

Missing Leads Reconstruction (ML): In resource-limited clinical settings like ambulatory care, electrode placements can vary from six-lead montages and reduced Frank or EASI configurations to single-lead setups. Similarly, in non-clinical settings, smartwatches like the Apple Watch provide a single lead ECG by measuring the potential between the wrist and the finger of the opposite hand. Recent studies such as [90] showed that this single lead ECG is essentially equivalent to the lead I ECG recorded in a 12 lead ECG. Several papers have addressed the reconstruction of ECGs from a single lead with deep learning, indicating potential applicability in ECG reconstruction from smartwatch single-lead ECG; see e.g., [82, 80, 41].

In the first experiment, we evaluate the performance of `EM-BeatDiff` in reconstructing V1-6 from the limb leads (I, II, III). This corresponds to the setting ML (V1-6) in Table 2. The second task we consider is generating II, III, V1-6 from lead I, which we refer to as the Smartwatch task. This corresponds to setting ML (SW) in Table 2. In Table 4, `EM-BeatDiff` outperforms EkGAN ([41]), a deep learning-based methods designed and trained for ECG missing lead reconstruction according to SSD, MAD and Cos. Unlike EkGAN, `EM-BeatDiff` does not require any task-specific training.

Cardiac Anomaly Detection (AD): In this section, we propose using `EM-BeatDiff` for detecting cardiac abnormalities. Our evaluation methodology consists of evaluating `EM-BeatDiff` 's capacity to detect four distinct medical conditions: Myocardial Infarction (MI), Left Anterior Descending artery (LAD), Left Atrial Enlargement (LAE), and Long QT syndrome (LQT). These anomalies were selected due to their typical association with localized alterations in P-Wave, QRS, or T-Wave morphologies. To incorporate patient-specific ECG data, we consider three distinct conditioning settings: (I, II, III), QRS, and ST. Conditioning on the limb leads (I, II, III) suggests that the abnormality is more prominently manifested in the precordial leads than the limb leads. Conditioning on QRS indicates that the abnormality is evident in the T-wave, while conditioning on the ST segment implies that the abnormality is present either in the QRS or the P-wave.

For each conditioning type and medical condition, we generate samples from the posterior distribution using `EM-BeatDiff`. Th anomaly score is the $1 - R^2$ metric between the mean of the generated ECGs from the posterior distribution and the observed ECG over the non-conditional ECG segment.

First, we conduct an ablation study to determine the optimal setting for each medical condition, as detailed in Appendix B.6.3. The chosen optimal settings are presented in Table 2, designated as AD

Table 5: AUC obtained using the proposed anomaly detection score $(1 - R^2)$ for each medical conditioning. See Table 2 for details on the inverse problem in hand. Confidence intervals are obtained by running 10 times EM-BeatDiff per heartbeat.

Model	MI	LAD	LAE	LQT
AAE [76]	80.23	82.69	74.87	70.96
EM-BeatDiff	**84.82 ± 0.01**	**93.06 ± 0.03**	**79.02 ± 0.07**	**84.73 ± 0.04**

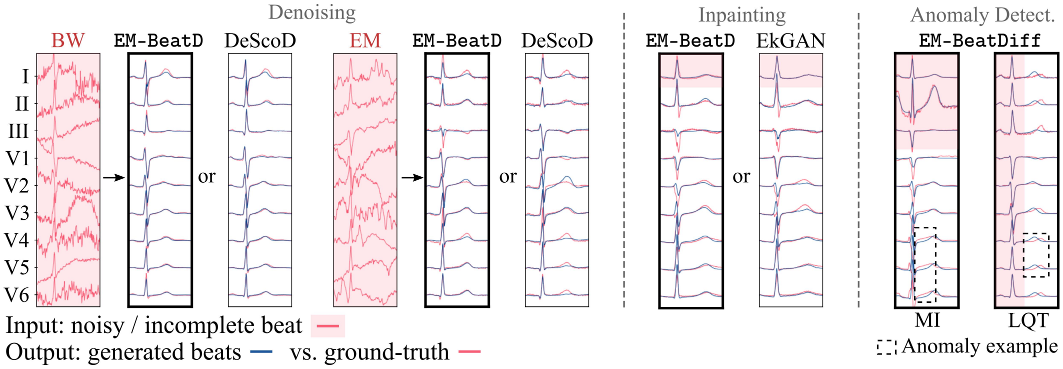


Figure 3: Illustration of EM-BeatDiff on the denoising, inpaiting and anomaly detection tasks. The red background indicate the parts of the ECG that are observed through y. The red ECGs corresponds to the real ECG and the blue ECGs corresponds to each algorithm reconstructed ECG.

followed by the respective medical condition name. Table 5 shows that EM-BeatDiff outperforms AAE ([76]), which uses as anomaly score the MSE between the output of an Adversarial Auto Encoder and the input ECG, according to the AUC of the anomaly score. See Appendix B.3.5 for implementation details. A key advantage of the aforementioned approach is its ability to function in an interactive manner, unlike methods that rely on training on a specific setting. By simply selecting a different set of ECG leads for conditioning, the posterior can be regenerated in a near-online fashion following visual assessment of the posterior and the patient ECGs as in the MI case in Figure 3.

5 Conclusion

In this work, we have described a flexible method that addresses several challenges in heart beat morphology, including baseline wander and electrode-motion removal, missing lead reconstruction and anomaly detection, all formulated as Bayesian linear inverse problems. Our method utilises BeatDiff a DDM pre-trained to generate the heartbeat morphology on 12 leads, as a prior for sampling solutions to inverse problems with EM-BeatDiff. Several evaluation metrics show the effectiveness of EM-BeatDiff compared to baseline solutions, which contrary to EM-BeatDiff require specific training for the specific task in hand.

Our approach also enables new applications, such as generating a 12-lead ECG using a subset of electrodes, including 12-lead heartbeat morphology reconstruction from smartwatch measurements as an example. Another example is the anomaly detection algorithm which can enable diagnostic of long QT syndrome or other diseases that specifically alter repolarization. In this paper, BeatDiff was trained only on healthy ECGs. However, BeatDiff could be trained on a dataset containing ECGs presenting pathologies by conditioning on the specific pathology as discussed below.

6 Discussion

This paper focus on generating 12-lead healthy heartbeats from partial measurements, e.g., limb leads only, samples corrupted with eletrocde artifacts, see Tables 3 and 4. We show that EM-BeatDiff can also be used to classify abnormal heartbeats in an unsupervised manner: we generate healthy counterparts of abnormal heartbeats and use the distance as an anomaly score. The flexibility of our method allows the detection of various heart conditions (MI, LQT, LAD, LAE) by reconstructing 12 leads from limb leads only, QRS only, or ST only, see Tables 11 and 12.

Risk of hallucination We would like to point out that this anomaly detection tool is semi-white-box: rather than outputting an abnormality score, our approach is also able to show the healthy counterparts of an abnormal signal and highlight where they differ from the patient's signals. This could theoretically enable cardiologists to rule out abnormalities that are not relevant. However, there is still a risk of hallucinations. We have shown that the generated signals are close to the real signals for healthy patients, but a clinical study must be performed before clinical use.

10s ECG signals Although our study focused on heartbeat morphology, we would like to discuss the feasibility of generating realistic longer samples that are more than a beat. We trained a diffusion model to produce 5-second ECGs. We then applied our sampling algorithm to reconstruct 12-lead ECGs from limb leads only and limb leads + V2 + V4. This second setting is similar to AliveCor's recent system Kardia12L*. We found that reconstructing 12 leads from multi-beat ECGs is more complex and requires precordial leads for reasonable results (see figure 11). A special study is needed to build a relevant diffusion model and adapt the algorithm's parameters in order to generate longer 12-lead ECGs from limb leads only. This adaptation is complex due to the need for larger models and more particles for posterior sampling.

Arrhythmic data Testing `EM-BeatDiff`'s ability to reconstruct arrhythmic ECGs is valuable as it would enable the use of additional simulated leads for abnormality detection from portable devices such as smartwatches or AliveCor products. We trained a diffusion model on 5-second ECGs with Atrial Fibrillation (AF) from PhysioNet and successfully predicted a reasonable 12-lead ECG from limbs + V2 + V4 (Kardia12L setting) (see figure 12). A larger AF dataset would yield more interesting results, but this experiment shows `EM-BeatDiff`'s potential for generating rhythmic abnormalities.

Heartbeat segmentation We would like to discuss the use of external segmentation tools for segmenting heartbeats before applying `EM-BeatDiff`. The segmentation of heartbeats is a common practice in the literature. Many works, including the baselines we analyzed in our paper, such as [56], and more recent papers [48], rely on external tools to segment the common 10-second clinical ECG signal into heartbeats. The heartbeat settings of `EM-BeatDiff` could also be used for arrhythmic data such as Premature Ventricular Contractions (PVC), even if they significantly alter the ECG phase. Indeed, one could detect PVCs using available methods such as [12] and remove them from inference.

7 Broader Impact

We demonstrate various ways in which `BeatDiff` and `EM-BeatDiff` can be utilized to address various heartbeat morphology analysis tasks. It is important to note that all our results are currently in prototype stage, and before any implementation in a clinical environment, a prior impact assessment and clinical trial must be conducted. This notably includes verifying performance on other datasets that better represent patient characteristics, as well as conditioning all the hyperparameters chosen in this study on this dataset.

8 Acknowledgment

This work was supported by the French National Research Agency (ANR- 10-IAHU04-LIRYC). We would like to thank PNY Technology for the computing power made available to us for this study. The work of LB and GC has been partly funded by the European Union (ERC-2022-SYG-OCEAN-101071601). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

[*] `https://alivecor.com/products/kardia12l`

References

[1] Adib, E., Afghah, F., and Prevost, J. J. (2022). Arrhythmia classification using cgan-augmented ecg signals*. *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1865–1872.

[2] Adib, E., Fernandez, A. S., Afghah, F., and Prevost, J. J. (2023). Synthetic ecg signal generation using probabilistic diffusion models. *IEEE Access*, 11:75818–75828.

[3] Alcaraz, J. M. L. and Strodthoff, N. (2023). Diffusion-based conditional ecg generation with structured state space models. *Computers in Biology and Medicine*, 163:107115.

[4] Arjomand Bigdeli, S., Zwicker, M., Favaro, P., and Jin, M. (2017). Deep mean-shift priors for image restoration. *Advances in Neural Information Processing Systems*, 30.

[5] Bai, Y., Chen, W., Chen, J., and Guo, W. (2020). Deep learning methods for solving linear inverse problems: Research directions and paradigms. *Signal Processing*, 177:107729.

[6] Ball, R. L., Feiveson, A. H., Schlegel, T. T., Stare, V., and Dabney, A. R. (2014). Predicting "heart age" using electrocardiography. *Journal of personalized medicine*, 4(1):65–78.

[7] Bazett, H. (1997). An analysis of the time-relations of electrocardiograms. *Annals of noninvasive electrocardiology*, 2(2):177–194.

[8] Bear, L. R., Svehlikova, J., Bergquist, J. A., Good, W. W., Rababah, A., Coll-Font, J., Macleod, R. S., Van Dam, E., and Dubois, R. (2021). Impact of baseline drift removal on ecg beat classification and alignment. In *2021 Computing in Cardiology (CinC)*, volume 48, pages 01–04. IEEE.

[9] Benton, J., Shi, Y., De Bortoli, V., Deligiannidis, G., and Doucet, A. (2022). From denoising diffusions to denoising markov models. *arXiv preprint arXiv:2211.03595*.

[10] Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-López, F., Pedregosa, F., and Vert, J.-P. (2021). Efficient and modular implicit differentiation. *arXiv preprint arXiv:2105.15183*.

[11] Brammer, J. C. (2020). biopeaks: a graphical user interface for feature extraction from heart- and breathing biosignals. *Journal of Open Source Software*, 5(54):2621.

[12] Cai, Z., Wang, T., Shen, Y., Xing, Y., Yan, R., Li, J., and Liu, C. (2022). Robust pvc identification by fusing expert system and deep learning. *Biosensors*, 12(4).

[13] Cardoso, G., el idrissi, Y. J., Corff, S. L., and Moulines, E. (2024). Monte carlo guided denoising diffusion models for bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations*.

[14] Chatterjee, S., Thakur, R. S., Yadav, R. N., Gupta, L., and Raghuvanshi, D. K. (2020). Review of noise removal techniques in ecg signals. *IET Signal Processing*, 14(9):569–590.

[15] Chiang, H.-T., Hsieh, Y.-Y., Fu, S.-W., Hung, K.-H., Tsao, Y., and Chien, S.-Y. (2019). Noise reduction in ecg signals using fully convolutional denoising autoencoders. *IEEE Access*, 7:60806–60813.

[16] Chopin, N., Papaspiliopoulos, O., et al. (2020). *An introduction to sequential Monte Carlo*, volume 4. Springer.

[17] Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*.

[18] Ciosek, K., Fortuin, V., Tomioka, R., Hofmann, K., and Turner, R. E. (2020). Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*.

[19] Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M. (2023). Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[20] Das, M. K. and Zipes, D. P. (2009). Fragmented qrs: a predictor of mortality and sudden cardiac death. *Heart rhythm*, 6(3):S8–S14.

[21] Dashti, M. and Stuart, A. M. (2017). The bayesian approach to inverse problems. In *Handbook of uncertainty quantification*, pages 311–428. Springer.

[22] de Pinto, V. (1992). Filters for the reduction of baseline wander and muscle artifact in the ecg. *Journal of electrocardiology*, 25:40–48.

[23] Delaney, A. M., Brophy, E., and Ward, T. E. (2019). Synthesis of realistic ecg using generative adversarial networks.

[24] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

[25] Doucet, A., De Freitas, N., Gordon, N. J., et al. (2001). *Sequential Monte Carlo methods in practice*, volume 1. Springer.

[26] Fort, G. and Moulines, E. (2003). Convergence of the monte carlo expectation maximization for curved exponential families. *The Annals of Statistics*, 31(4):1220–1259.

[27] Fridericia, L. (1921). Die systolendauer im elektrokardiogramm bei normalen menschen und bei herzkranken. *Acta Medica Scandinavica*, 54(1):17–50.

[28] Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*.

[29] Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016). Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29.

[30] Golany, T., Freedman, D., and Radinsky, K. (2021). Ecg ode-gan: Learning ordinary differential equations of ecg dynamics via generative adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 134–141.

[31] Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220. Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[32] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

[33] Gui, J., Sun, Z., Wen, Y., Tao, D., and Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE transactions on knowledge and data engineering*.

[34] Haïssaguerre, M., Hocini, M., Cheniti, G., Duchateau, J., Sacher, F., Puyo, S., Cochet, H., Takigawa, M., Denis, A., Martin, R., et al. (2018). Localized structural alterations underlying a subset of unexplained sudden cardiac death. *Circulation: Arrhythmia and Electrophysiology*, 11(7):e006120.

[35] Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., Tison, G. H., Bourn, C., Turakhia, M. P., and Ng, A. Y. (2019). Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1):65–69.

[36] Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. (2023). Flax: A neural network library and ecosystem for JAX.

[37] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

[38] Idier, J. (2013). *Bayesian approach to inverse problems*. John Wiley & Sons.

[39] Iuliano, S., Fisher, S. G., Karasik, P. E., Fletcher, R. D., Singh, S. N., of Veterans Affairs Survival Trial of Antiarrhythmic Therapy in Congestive Heart Failure, D., et al. (2002). Qrs duration and mortality in patients with congestive heart failure. *American heart journal*, 143(6):1085–1091.

[40] Jameson, J. L., Fauci, A. S., Kasper, D. L., Hauser, S. L., Longo, D. L., and Loscalzo, J. (2018). McGraw-Hill Education, New York, NY.

[41] Joo, J., Joo, G., Kim, Y., Jin, M.-N., Park, J., and Im, H. (2023). Twelve-lead ecg reconstruction from single-lead signals using generative adversarial networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 184–194. Springer.

[42] Kaltenbach, S., Perdikaris, P., and Koutsourelakis, P.-S. (2023). Semi-supervised invertible neural operators for bayesian inverse problems. *Computational Mechanics*, pages 1–20.

[43] Kang, J. and Wen, H. (2022). A Study on Several Critical Problems on Arrhythmia Detection using Varying-Dimensional Electrocardiography. *Physiological Measurement*, 43(6):064007.

[44] Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*.

[45] Kawar, B., Elad, M., Ermon, S., and Song, J. (2022). Denoising diffusion restoration models. 35:23593–23606.

[46] Kawar, B., Vaksman, G., and Elad, M. (2021). Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769.

[47] Kearney, K., Thomas, C., and McAdams, E. (2007). Quantification of motion artifact in ecg electrode design. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1533–1536. IEEE.

[48] Kim, Y., Lee, M., Yoon, J., Kim, Y., Min, H., Cho, H., Park, J., and Shin, T. (2023). Predicting future incidences of cardiac arrhythmias using discrete heartbeats from normal sinus rhythm ecg signals via deep learning methods. *Diagnostics*, 13(17).

[49] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[50] Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

[51] Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2020). Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979.

[52] Kuchar, D. L., Ruskin, J. N., and Garan, H. (1989). Electrocardiographic localization of the site of origin of ventricular tachycardia in patients with prior myocardial infarction. *Journal of the American College of Cardiology*, 13(4):893–900.

[53] Kuntz, J., Lim, J. N., and Johansen, A. M. (2023). Particle algorithms for maximum likelihood training of latent variable models. In *International Conference on Artificial Intelligence and Statistics*, pages 5134–5180. PMLR.

[54] Levine, R. A. and Casella, G. (2001). Implementations of the monte carlo em algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439.

[55] Li, H. and Boulanger, P. (2021). An automatic method to reduce baseline wander and motion artifacts on ambulatory electrocardiogram signals. *Sensors*, 21(24):8169.

[56] Li, H., Ditzler, G., Roveda, J., and Li, A. (2023). Descod-ecg: Deep score-based diffusion model for ecg baseline wander and noise removal. *IEEE Journal of Biomedical and Health Informatics*, pages 1–11.

[57] Li, H., Schwab, J., Antholzer, S., and Haltmeier, M. (2020). Nett: Solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005.

[58] Li, M. and Ramos, L. G. (2017). Drug-induced qt prolongation and torsades de pointes. *Pharmacy and Therapeutics*, 42(7):473.

[59] Macfarlane, P. W., Van Oosterom, A., Pahlm, O., Kligfield, P., Janse, M., and Camm, J. (2010). *Comprehensive electrocardiology*. Springer Science & Business Media.

[60] Malik, M., Hnatkova, K., Kowalski, D., Keirns, J. J., and van Gelderen, E. M. (2013). Qt/rr curvatures in healthy subjects: sex differences and covariates. *American Journal of Physiology-Heart and Circulatory Physiology*, 305(12):H1798–H1806.

[61] McLachlan, G. J. and Krishnan, T. (2007). *The EM algorithm and extensions*. John Wiley & Sons.

[62] Moody, G. B., Muldrow, W., and Mark, R. G. (1984). A noise stress test for arrhythmia detectors. *Computers in cardiology*, 11(3):381–384.

[63] Nygaard, R., Melnikov, G., and Katsaggelos, A. K. (2001). A rate distortion optimal ecg coding algorithm. *IEEE Transactions on biomedical engineering*, 48(1):28–40.

[64] Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. (2020). Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56.

[65] Phillips, A., Dau, H.-D., Hutchinson, M. J., De Bortoli, V., Deligiannidis, G., and Doucet, A. (2024). Particle denoising diffusion sampler. *arXiv preprint arXiv:2402.06320*.

[66] Porthan, K., Viitasalo, M., Toivonen, L., Havulinna, A. S., Jula, A., Tikkanen, J. T., Väänänen, H., Nieminen, M. S., Huikuri, H. V., Newton-Cheh, C., et al. (2013). Predictive value of electrocardiographic t-wave morphology parameters and t-wave peak to t-wave end interval for sudden cardiac death in the general population. *Circulation: Arrhythmia and Electrophysiology*, 6(4):690–696.

[67] Ramírez, J., Orini, M., Mincholé, A., Monasterio, V., Cygankiewicz, I., Bayes de Luna, A., Martínez, J. P., Pueyo, E., and Laguna, P. (2017). T-wave morphology restitution predicts sudden cardiac death in patients with chronic heart failure. *Journal of the American Heart Association*, 6(5):e005310.

[68] Reyna, M. A., Sadr, N., Alday, E. A. P., Gu, A., Shah, A. J., Robichaux, C., Rad, A. B., Elola, A., Seyedi, S., Ansari, S., Ghanbari, H., Li, Q., Sharma, A., and Clifford, G. D. (2021). Will two do? varying dimensions in electrocardiography: The physionet/computing in cardiology challenge 2021. In *2021 Computing in Cardiology (CinC)*, volume 48, pages 1–4.

[69] Reyna, M. A., Sadr, N., Alday, E. A. P., Gu, A. P., Shah, A. J., Robichaux, C., Rad, A. B., Andoni, Elola, Seyedi, S., Ansari, S., Ghanbari, H., Qiao, Li, Sharma, A., and Clifford, G. D. (2022). Issues in the automated classification of multilead ecgs using heterogeneous labels and populations. *Physiological Measurement*, 43.

[70] Ribeiro, A. H., Ribeiro, M. H., Paixão, G. M., Oliveira, D. M., Gomes, P. R., Canazart, J. A., Ferreira, M. P., Andersson, C. R., Macfarlane, P. W., Meira Jr, W., et al. (2020). Automatic diagnosis of the 12-lead ecg using a deep neural network. *Nature communications*, 11(1):1760.

[71] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer.

[72] Rubin, D. B. (1987). The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The sir algorithm. *Journal of the American Statistical Association*, 82(398):543–546.

[73] Sagie, A., Larson, M. G., Goldberg, R. J., Bengtson, J. R., and Levy, D. (1992). An improved method for adjusting the qt interval for heart rate (the framingham heart study). *The American journal of cardiology*, 70(7):797–801.

[74] Sahlström, T. and Tarvainen, T. (2023). Utilizing variational autoencoders in the bayesian inverse problem of photoacoustic tomography. *SIAM Journal on Imaging Sciences*, 16(1):89–110.

[75] Salama, G. and Bett, G. C. (2014). Sex differences in the mechanisms underlying long qt syndrome. *American Journal of Physiology-Heart and Circulatory Physiology*, 307(5):H640–H648.

[76] Shan, L., Li, Y., Jiang, H., Zhou, P., Niu, J., Liu, R., Wei, Y., Peng, J., Yu, H., Sha, X., and Chang, S. (2022). Abnormal ecg detection based on an adversarial autoencoder. *Frontiers in Physiology*, 13.

[77] Shin, H. and Choi, M. (2023). Physics-informed variational inference for uncertainty quantification of stochastic differential equations. *Journal of Computational Physics*, page 112183.

[78] Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245.

[79] Singh, P. and Pradhan, G. (2020). A new ecg denoising framework using generative adversarial network. *IEEE/ACM transactions on computational biology and bioinformatics*, 18(2):759–764.

[80] Smith, G. H., Van den Heever, D. J., and Swart, W. (2021). The reconstruction of a 12-lead electrocardiogram from a reduced lead set using a focus time-delay neural network. *Acta Cardiologica Sinica*, 37(1):47.

[81] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR.

[82] Sohn, J., Yang, S., Lee, J., Ku, Y., and Kim, H. C. (2020). Reconstruction of 12-lead electrocardiogram from a three-lead patch-type device using a lstm network. *Sensors*, 20(11):3278.

[83] Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. In *International Conference on Learning Representations*.

[84] Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021b). Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428.

[85] Song, Y., Shen, L., Xing, L., and Ermon, S. (2022). Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*.

[86] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021c). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

[87] Stuart, A. M. (2010). Inverse problems: a bayesian perspective. *Acta numerica*, 19:451–559.

[88] Su, J., Xu, B., and Yin, H. (2022). A survey of deep learning approaches to image restoration. *Neurocomputing*, 487:46–65.

[89] Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. S. (2023). Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*.

[90] van der Zande, J., Strik, M., Dubois, R., Ploux, S., Alrub, S. A., Caillol, T., Nasarre, M., Donker, D. W., Oppersma, E., and Bordachar, P. (2023). Using a smartwatch to record precordial electrocardiograms: a validation study. *Sensors*, 23(5):2555.

[91] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[92] Viskin, S. (1999). Long qt syndromes and torsade de pointes. *The Lancet*, 354(9190):1625–1633.

[93] Vogel, B., Claessen, B. E., Arnold, S. V., Chan, D., Cohen, D. J., Giannitsis, E., Gibson, C. M., Goto, S., Katus, H. A., Kerneis, M., et al. (2019). St-segment elevation myocardial infarction. *Nature reviews Disease primers*, 5(1):39.

[94] Wang, Z., Wong, C. M., da Cruz, J. N., Wan, F., Mak, P.-I., Mak, P. U., and Vai, M. I. (2014). Muscle and electrode motion artifacts reduction in ecg using adaptive fourier decomposition. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1456–1461. IEEE.

[95] Wei, G. C. and Tanner, M. A. (1990). A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704.

[96] Wei, X., van Gorp, H., Gonzalez-Carabarin, L., Freedman, D., Eldar, Y. C., and van Sloun, R. J. (2022). Deep unfolding with normalizing flow priors for inverse problems. *IEEE Transactions on Signal Processing*, 70:2962–2971.

[97] Wen, H. and Kang, J. (2021). Hybrid Arrhythmia Detection on Varying-Dimensional Electrocardiography: Combining Deep Neural Networks and Clinical Rules. In *2021 Computing in Cardiology (CinC)*. IEEE.

[98] Wu, L., Trippe, B. L., Naesseth, C. A., Blei, D. M., and Cunningham, J. P. (2023). Practical and asymptotically exact conditional sampling in diffusion models.

[99] Xia, Y., Wang, W., and Wang, K. (2023a). Ecg signal generation based on conditional generative models. *Biomedical Signal Processing and Control*, 82:104587.

[100] Xia, Y., Xu, Y., Chen, P., Zhang, J., and Zhang, Y. (2023b). Generative adversarial network with transformer generator for boosting ecg classification. *Biomedical Signal Processing and Control*, 80:104276.

[101] Zhang, D. (2006). Wavelet approach for ecg baseline wander correction and noise reduction. In *2005 IEEE engineering in medicine and biology 27th annual conference*, pages 1212–1215. IEEE.

[102] Zhang, L., Liu, L., R Kowey, P., and H Fontaine, G. (2014). The electrocardiographic manifestations of arrhythmogenic right ventricular dysplasia. *Current Cardiology Reviews*, 10(3):237–245.

[103] Zhihang, X., Yingzhi, X., and Qifeng, L. (2023). A domain-decomposed vae method for bayesian inverse problems. *arXiv preprint arXiv:2301.05708*.

[104] Zhu, F., Ye, F., Fu, Y., Liu, Q., and Shen, B. (2019). Electrocardiogram generation with a bidirectional lstm-cnn generative adversarial network. *Scientific reports*, 9(1):6734.

Contents

A Theoretical appendix

A.1 Sequential Monte Carlo samplers

A.1.1 SMC Algorithm

In this section we first provide the SMC algorithm 1.

Algorithm 1: SMC

```
Input: observation $y$, number of diffusion steps $K$, number of particles $M$
Operations involving index i are repeated for $i \in [1:M]$
Initialization: $\xi_K^i \sim q_{\mathrm{ref}}$
for $k = K-1$ to $0$ do
    $I_k^i \sim \mathrm{Cat}\big(\{\omega_k(\xi_{k+1}^j)/\sum_{i=1}^M \omega_k(\xi_{k+1}^i)\}_{j=1}^M\big)$
    $\xi_k^i \sim \hat{p}_k^y(\cdot|\xi_{k+1}^{I_k^i})$
end for
Output: $\xi_0^{1:M}$
```

A.2 `MCGDiff`

A.2.1 Covariance matrix

Simplified setting Following [13, Section 2.1], we first give explicitly the potentials for the simplified case

$$Y = \overline{\mathrm{V}}^T X + \sigma_y \mathrm{S}^{-1}\varepsilon\,, \quad \varepsilon \sim \mathcal{N}(0, \mathrm{I}_{\mathsf{d_y}})\,, \tag{A.1}$$

where $\overline{\mathrm{V}} \in \mathbb{R}^{\mathsf{d}_x \times \mathsf{d_y}}$ is an orthonormal matrices and $\mathrm{S} \in \mathbb{R}^{\mathsf{d_y} \times \mathsf{d_y}}$ is diagonal. For $i \in [1:\mathsf{d_y}]$, we define $\tau_i := \min\{k \in [1:K] | \upsilon_k > \sigma_y/s_i\}$ where s_i is the i-th element of the diagonal matrix S. τ_i is defined such that the i-th coordinate of Y, $Y[i]$ and the i-th coordinate of $\overline{\mathrm{V}}^T X_{\tau_i}$ follow the same distribution. This is the fundamental idea of the `MCGDiff` algorithm for the noisy version and refer to [13, Section 2.1] for a detailed explanation.

We define, for $k \in [1:K]$, $R_k \in \mathbb{R}^{\mathsf{d_y} \times \mathsf{d_y}}$ a diagonal matrix with values

$$R_k[i,i] = \begin{cases} (\upsilon_k^2 - \upsilon_{\tau_i}^2)^{1/2} & \text{if} \quad k > \tau_i\,, \\ \sigma_y/s_i & \text{if} \quad k \leq \tau_i\,. \end{cases}$$

We can finally define the `MCGDiff` potentials when the measurement models are of the type A.1. For $k \in [1:K]$, define

$$g_l^y(x) := \mathcal{N}(y; \overline{\mathrm{V}}x, \mathrm{R}_k\mathrm{R}_k^T)\,.$$

Note that if $k < \min\{\tau_i | i \in [1:\mathsf{d_y}]\}$, then $g_k^y(x) = g_0^y(x)$.

General A_θ and diagonal D_θ. Even though A.1 corresponds to a simplified version of 3.3, `MCGDiff` can also be applied to the case where $D_\theta = \sigma_y\,\mathrm{I}$ and thus

$$Y = \mathrm{A}_\theta X + b_\theta + \sigma_y \varepsilon$$

where $\varepsilon \sim \mathcal{N}(0_{\mathsf{d}_y}, \mathrm{I}_{\mathsf{d}_y})$ and $\sigma \geq 0$ and the singular value decomposition (SVD) $\mathrm{A}_\theta = \mathrm{U}_\theta \mathrm{S}_\theta \overline{\mathrm{V}}_\theta{}^T$, where $\overline{\mathrm{V}}_\theta \in \mathbb{R}^{\mathsf{d}_x \times \mathsf{d_y}}$, $\mathrm{U}_\theta \in \mathbb{R}^{\mathsf{d}_y \times \mathsf{d_y}}$ are two orthonormal matrices, and $\mathrm{S}_\theta \in \mathbb{R}^{\mathsf{d_y} \times \mathsf{d_y}}$ is diagonal.

Set $\mathsf{b} = \mathsf{d}_x - \mathsf{d_y}$. Multiplying the measurement equation by $\mathrm{S}_\theta^{-1}\mathrm{U}_\theta^T$ and substracting $\mathrm{S}_\theta^{-1}\mathrm{U}_\theta^T b_\theta$ yields

$$\mathbf{Y} := \mathrm{S}_\theta^{-1}\mathrm{U}_\theta^T\,(Y - b_\theta) = \overline{\mathrm{V}}_\theta X + \sigma_y \mathrm{S}_\theta^{-1}\tilde{\varepsilon}\,, \quad \tilde{\varepsilon} \sim \mathcal{N}(0, \mathrm{I}_{\mathsf{d_y}})\,.$$

Therefore, it is possible to use the potentials defined above which yields

$$g_k^{\theta,y}(x) := \mathcal{N}(y; \mathrm{A}_\theta x + b_\theta, \underbrace{\mathrm{U}_\theta \mathrm{S}_\theta^2 \mathrm{R}_{k,\theta}^2 \mathrm{U}_\theta^T}_{:=\Sigma_{k,\theta}})\,.$$

A.2.2 Proposal Potential and Weight

Using conjugate formulas we compute the proposal kernel and the weights defined in Section 3 used in SMC algorithm

$$
\begin{aligned}
\hat{p}^{\theta,y}_{k|k+1}(x_k|x_{k+1}) &= \frac{g^y_k(x_k)p_{k|k+1}(x_k|x_{k+1})}{\int g^y_k(z)p_{k|k+1}(z|x_{k+1})\mathrm{d}z} \\
&= \mathcal{N}\left(x_k; \mathrm{W}_{k,\theta}\left\{A^T_\theta\Sigma^{-1}_{k,\theta}(y-b_\theta)+\upsilon_k^{-2}\boldsymbol{\mu}_k(x_{k+1},\mathcal{D}_{0|k}(x_{k+1}))\right\}, \mathrm{W}_{k,\theta}\right)\,, \\
\omega^{\theta,y}_{k+1}(x_{k+1}) &= \frac{\int g^y_k(z)p_{k|k+1}(z|x_{k+1})\mathrm{d}z}{g^y_{k+1}(x_{k+1})} = \frac{\mathcal{N}(y; A_\theta\boldsymbol{\mu}_k(x_{k+1},\mathcal{D}_{0|k}(x_{k+1}))+b_\theta, \Sigma_{k,\theta}+\upsilon_k^2 A_\theta A_\theta^T)}{\mathcal{N}(y; A_\theta x_{k+1}+b_\theta, \Sigma_{k+1,\theta})}\,,
\end{aligned}
$$

where $\mathrm{W}_{k,\theta} := \left(\upsilon_k^{-2}\,\mathrm{I}+A^T_\theta\Sigma^{-1}_{k,\theta}A_\theta\right)^{-1} = \left(\upsilon_k^{-2}\,\mathrm{I}+\overline{\mathrm{V}}_\theta\mathrm{R}^{-2}_{k,\theta}\overline{\mathrm{V}}^T_\theta\right)^{-1}$.

A.3 Monte Carlo Expectation Maximization (MCEM)

Algorithm 2: MCEM

```
    Input: observation $y$, number of diffusion steps $K$, number of particles $M$, regularization parameters $(\lambda_1, \lambda_2)$ (4.2), M step optimization parameters $(N_{\mathrm{M}}, \gamma)$, total number of iterations $N_{\mathrm{EM}}$, initial parameters $\theta_0$.
    for $k = 1$ to $N_{\mathrm{EM}}$ do
        $\xi^t_{1:M} \leftarrow \texttt{MCGDiff}(\theta_{t-1}, K, y, M)$
        $\theta_t \leftarrow$ M-step$(\theta_{t-1}, \xi^t_{1:M}, (\lambda_1, \lambda_2), N_{\mathrm{M}}, \gamma, y)$ (Algorithm 3)
    end for
    Output: $\xi_0^{1:M}$
```

Algorithm 3: M-step (implemented using [10])

```
    Input: initial parameter $\theta$, particles $\xi_{1:M}$, regularization parameters $(\lambda_1, \lambda_2)$, number of gradient steps $N_{\mathrm{M}}$, learning rate $\gamma$, observation $y$.
    for $k = 1$ to $N_{\mathrm{M}}$ do
        $\theta \leftarrow \mathrm{Prox}_{\lambda_1\|\cdot\|_1+\lambda_2\|\cdot\|_2}(\theta + \gamma\nabla F(\theta; \xi_{1:M}, y))$ (see Equation (A.2) for definition of $F$)
    end for
    Output: $\theta$
```

In algorithm 3, F is the empirical error defined as follow for parmater θ when provided observation y and particles $\xi_{1:M}$

$$
F(\theta; \xi_{1:M}, y) = \frac{1}{M}\sum_{i=1}^{M}\|A_\theta\xi_i\bar{A}_\theta + B_\theta + D_\theta\epsilon\bar{D}_\theta - y\|_2^2\,. \tag{A.2}
$$

B Numerical appendix

B.1 `BeatDiff`

B.1.1 Preprocessing Implementation Details

Our preprocessing follows:

- Align the recording-frequency of all ECGs to 250 Hz by performing down or up sampling. Thus, two consecutive points in the ECG are separated by 4ms.
- Extract R peaks from the ECG. The first principal component is extracted channel-wise from the entire ECG. Subsequently, this extracted component is processed through a Savitzky-Golay filter, characterized by an order of 3 and a window length of 15. The extraction of R-peaks is then carried out based on the methodology proposed in [11].
- Select the window $[\mathrm{R}-192\,\mathrm{ms}, R+512\,\mathrm{ms}]$ containing the QRS. This window corresponds to 176 time-points as $(192+512)/4 = 176$.

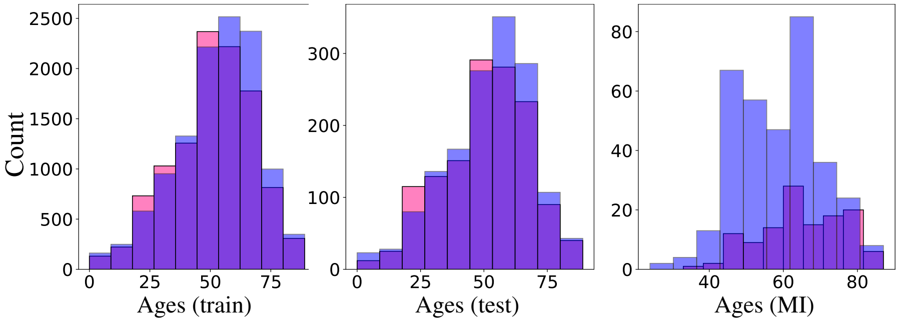
- ECGs are not normalized, unless otherwise specified for comparing to baseline methods or for improving visual clarity in figures.

B.1.2 Dataset statistics

Table 6: Distribution of patients, gender and number of recorded beats among train, test and MI sets.

	Train	CV	Test	MI
All (patients)	22580	2723	2864	468
Male (patients)	11722	1399	1497	343
Female (patients)	10858	1324	1367	125
All (beats)	214460	25694	27221	44911
Mean (beats)	9.5 +/- 0.1	9.4 +/- 0.2	9.5 +/- 0.2	96 +/- 5

Figure 4: Female (pink), male (blue) ages histograms in training (left), test (middle), MI (right) sets.



B.1.3 Backward generation

For ECG generation in Section 4 we follow the scheduling prposed by [44] which consists of, for a total number of backward steps $\tilde{K} \in [1 : K]$, defining for $t \in [1 : \tilde{K}]$

$$\upsilon_t = \left[\upsilon_K^{1/\rho} + \frac{t}{\tilde{K}}\left(\upsilon_1^{1/\rho} - \upsilon_K^{1/\rho}\right)\right]^\rho$$

Throughout our experiments, we used $\rho = 5$.

B.1.4 Network Architecture Details

In this work, we follow closely the architecture from [24], but adapting it to the case of the ECG. Denoising diffusion consists of using a single network to learn several denoising networks, one for each level of corruption $\{\upsilon_k\}_{k=1}^K$. We denote those different instances of the same network as $\{\mathcal{D}^\theta_{0|k}\}_{k=1}^K$. We describe below the different adaptations that we used for the specific case of the ECG. We start by describing how we embed the conditioning variables, namely the patient information $\mathrm{A}, \mathrm{S}, \mathrm{RR}$ but also the creation of a positional embedding on the signal and the noise embedding. We then describe the parameterization used in [24] but formalized in [44] of the F_θ and finally we describe precisely the structure of the denoising network.

Encoding of patient features ($\mathrm{e}_\mathcal{P}$): As explained in Section 2, we use the $\mathcal{P} = (\mathrm{A}, \mathrm{S}, \mathrm{RR})$ as a conditioning variable for $\mathcal{D}_{0|k}$. We encode the sex S as a boolean feature $\tilde{\mathrm{S}}$. For the numerical features, we choose the following normalization:

$$\tilde{\mathrm{A}} = (\mathrm{A} - 50)/50, \quad \widetilde{\mathrm{RR}} = (\mathrm{RR} - 500)/500\,.$$

This values are chosen so that $\tilde{\mathrm{A}} = 0$ for a 50 year old patient and $\mathrm{RR} = 0$ if the patient heart rate is of 120 bpm. The resulting vector obtained by concatenating $\tilde{\mathrm{S}}, \tilde{\mathrm{A}}, \widetilde{\mathrm{RR}}$ is fed into a two-layer dense network, yielding a 192×1 vector called $\tilde{\mathrm{e}}_\mathcal{P}$. The final embedding is obtained by passing $\tilde{\mathrm{e}}_\mathcal{P}$ through one MLP with SiLu activation and 2×192 neurons and a second linear layer projecting back to $\mathbb{R}^{192}$. This procedure leads to an embedding vector $\mathrm{e}_\mathcal{P}$.

Time conditioning ($\mathrm{e}_{\mathcal{T}}$)**:** We are interested in generating a fixed ECG beat. Therefore, the time wise position ($t \in [1 : \mathcal{T}]$) of each event is important to determine what is the event that we want to model in this moment. Indeed, we expect that (this can vary slightly for each signal) for $t < 50$ we observe the P-wave, for $t \in [50 : 100]$ we should observe the QRS-wave and for $t > 100$ we should observe the T-Wave. Each of this phenomenon possesses unique distinctive characteristics. This means that the data is not translation invariant. Indeed, we created each of the ECG beats by placing the R peak at the position $t = 75$.

Convolutional neural networks are translation invariant, therefore, we want to add information about the position of a certain value with respect to the whole beat window. To do so, we use positional embedding, first introduced in [91]. For a given embedding dimension $c \in \mathbb{N}$ and maximum sequence length $\mathcal{T} \in \mathbb{N}$, Positional encoding generates, for each sequence time $t \in [1 : \mathcal{T}]$ a $\mathrm{PosEnc}(t) \in \mathbb{R}^c$ by

$$\mathrm{PosEnc}(t)\,[l] = \begin{cases} \sin(1000^{-(2r/c)}t) & \text{if} \quad \ell = 2r\,, \\ \cos(1000^{-(2r/c)}t) & \text{if} \quad \ell = 2r + 1\,. \end{cases}$$

For the time embedding, we set $c = 192$ and we obtain a vector $\mathrm{e}_{\mathcal{T}} = (\mathrm{PosEnc}(1), \cdots, \mathrm{PosEnc}(\mathcal{T})) \in \mathbb{R}^{\mathcal{T} \times 192}$.

Noise level conditioning (e_{υ_k})**:** For encoding the noise level, we follow [24] and used also Positional encoding to generate a first embedding $\tilde{e}_{\upsilon_k} = \mathrm{PosEnc}(4^{-1}\log(\upsilon_k)) \in \mathbb{R}^{192}$. The final embedding is obtained by passing $\tilde{e}_{\upsilon_k}$ through one MLP with SiLu activation and 2×192 neurons and a second linear layer projecting back to $\mathbb{R}^{192}$. This procedure leads to an embedding vector e_{υ_k}.

Final conditioning vector ($\mathrm{e}_{\mathrm{cond}}$)**:** We combine $(\mathrm{e}_{\mathcal{P}}, \mathrm{e}_{\mathcal{T}}, \mathrm{e}_{\upsilon_k})$ into a single matrix $\mathrm{e}_{\mathrm{cond}} \in \mathbb{R}^{\mathcal{T} \times 192}$ by broadcasting (repeating across the first dimension) $\mathrm{e}_{\mathcal{P}}$ and e_{υ_k} into $(\mathcal{T}, 192)$ matrices and then defining $\mathrm{e}_{\mathrm{cond}} := \mathrm{SiLu}(\mathrm{e}_{\mathcal{P}} + \mathrm{e}_{\mathcal{T}} + \mathrm{e}_{\upsilon_k})$

Denoising network design: We use the definition of the Denoising network used in [24] and which is called the F net decomposition in [44]:

$$\mathcal{D}^{\theta}_{0|k}(x, \upsilon_k, \mathrm{e}_{\mathrm{cond}}) = \mathrm{c}_{\mathrm{skip}}(\upsilon_k)x + \mathrm{c}_{\mathrm{out}}(\upsilon_k)\,\mathrm{F}_{\theta}(\mathrm{c}_{\mathrm{in}}(\upsilon_k)x, \mathrm{e}_{\mathrm{cond}})\,.$$

where x is a 9×176 matrix corresponding to the noisy ECG beat, $\mathrm{c}_{\mathrm{in}}(\upsilon_k) = (\upsilon_k^2 + \sigma_{\mathrm{data}}^2)^{-1/2}$, $\mathrm{c}_{\mathrm{skip}}(\upsilon_k) = (\upsilon_k^2 + \sigma_{\mathrm{data}}^2)^{-1}\sigma_{\mathrm{data}}^2$, $\mathrm{c}_{\mathrm{out}}(\upsilon_k) = \upsilon_k\sigma_{\mathrm{data}}(\upsilon_k^2 + \sigma_{\mathrm{data}}^2)^{-1/2}$, and σ_{data} is the (estimated) empirical standard deviation of $\mathsf{q}_{\mathrm{data}}$. The key idea of this decomposition is that what is expected of the neural network is different for small υ_k and large υ_k.

For small υ_k, $\mathrm{c}_{\mathrm{skip}}(\upsilon_k) \approx 1$ and $\mathrm{c}_{\mathrm{out}}(\upsilon_k) \approx 0$, thus $\mathcal{D}^{\theta}_{0|k}(x, \upsilon_k, \mathrm{e}_{\mathrm{cond}}) \approx x$, which is expected since x is already a good reconstruction of the original data. On the contrary, when υ_k is large, then $\mathrm{c}_{\mathrm{skip}}(\upsilon_k) \approx 0$ and $\mathrm{c}_{\mathrm{out}} \approx 1$, thus $\mathcal{D}^{\theta}_{0|k}(x, \upsilon_k, \mathrm{e}_{\mathrm{cond}})$ relies heavily on the network F_{θ} to provide a good reconstruction.

The input scaling $\mathrm{c}_{\mathrm{in}}(\upsilon_k) = (\upsilon_k^2 + \sigma_{\mathrm{data}}^2)^{-1/2}$ is introduced so that $\mathrm{c}_{\mathrm{in}}(\upsilon_k)x$ has always the same standard deviation. As x is a realization of $X_k \sim X_0 + \upsilon_k\epsilon_k$ with $\epsilon_k \sim \mathcal{N}(0, \mathrm{I})$, $X_0 \sim \mathsf{q}_{\mathrm{data}}$ and σ_{data} is and approximation of the standard deviation of $\mathsf{q}_{\mathrm{data}}$, we expect $\mathrm{c}_{\mathrm{in}}(\upsilon_k)X_k$ to have an standard deviation of approximately 1.

F_{θ} **architecture** The F_{θ} is an evolution of the UNet firstly introduced in [71] and we follow the one proposed by [24]. We illustrate the general architecture of our F_{θ} of depth 2 in figure 5. More (or less) depth can be obtained by adding (or removing) Down blocks and Up blocks. Each block (Down, Up, Middle) is an instantiation of the general UNet block, whose architecture is shown in figure 6.

EncoderBlock: The EncoderBlock consists of a 1d Convolutional layer with kernel size 3 and padding and stride 1, with 192 kernels.

DecoderBlock: The Decoder block consist of a GroupNorm layer, followed by SiLu activation and finally a 1d Convolutional layer with kernel size 3 and padding and stride 1, with 9 kernels.

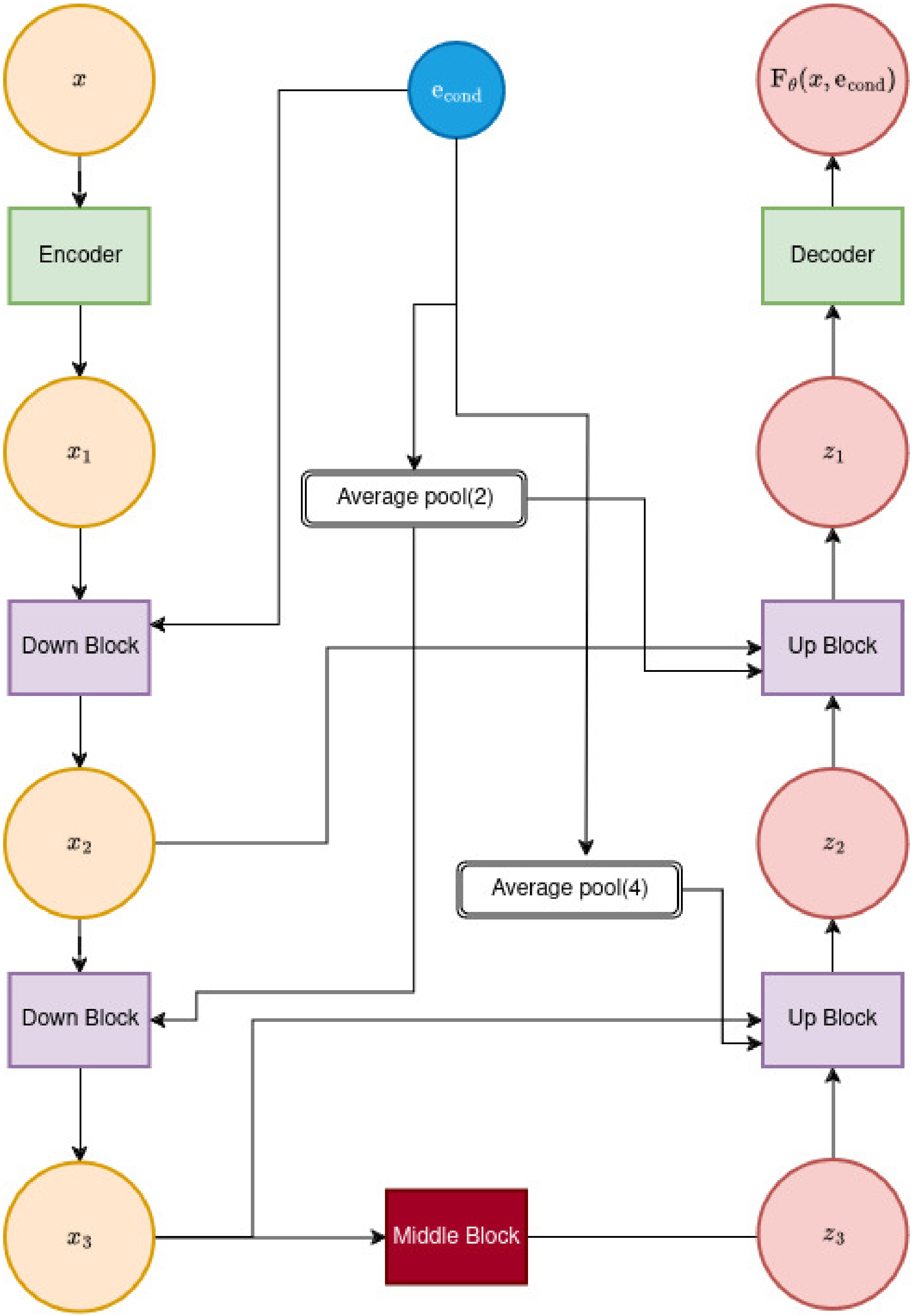


Figure 5: Illustration of F_θ architecture for a UNet of depth 2.

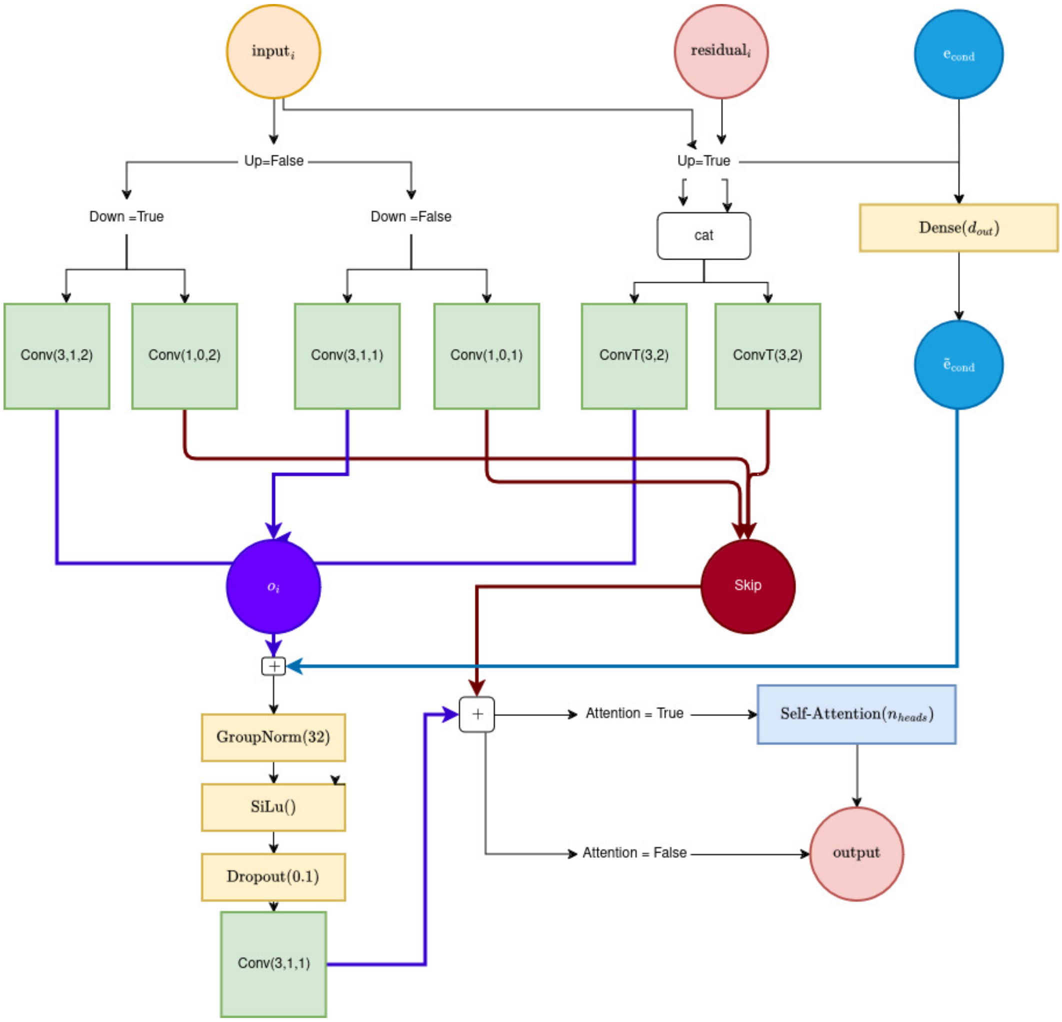


Figure 6: Illustration of a UNet block. Inputs: (Up, Down, d_out, Attention, n_{heads}).

UNet block: The terms in figure 6 describing the UNet with parameters (Up, Down, d_{out}, Attention, n_{heads}) block corresponds to:

- $Conv(k, p, s)$ means a 1d-convolutional layer with kernel size k, padding p and strides s,
- $ConvT(k, s)$ means a 1d transposed convolutional layer with kernel size k and stride s using the padding configuration "Same".

The number of output channels for the convolution layers are always d_{out}. From UNet blocks, we can construct

- DownBlock(d_{out}, n_{heads}, Attention): UNetBlock(False, True, d_{out}, Attention, n_{heads}),
- UpBlock(d_{out}, n_{heads}, Attention): UNetBlock(True, False, d_{out}, False, n_{heads}),
- MiddleBlock($d_{out}, n_{heads}, Attention$): Stack of two UNetBlock(False, False, d_{out}, Attention, n_{heads}).

The final configuration retained for `BeatDiff` is given in Table 7. In Appendix B.1.5 we tested running a deeper architecture that is given in Table 8 . Each output from the U-Net blocks undergoes a multi-head attention layer [91], with the number of heads equal to the original dimension divided by 64. The entire network $\mathcal{D}^{\theta}_{0|k}$ is trained to minimize $\mathcal{L}_{\mathcal{D}}$ through stochastic gradient descent on the healthy training set, and the best model is selected using the cross-validation set.

Layer Name	Parameters	Output Dimension
EncoderBlock		$\mathcal{T} \times 192$
DownBlock	$(d_{out}, n_{heads}, Attention) = (192, 0, False)$	$(\mathcal{T}/2) \times 192$
MiddleBlock	$(d_{out}, n_{heads}, Attention) = (192, 3, True)$	$(\mathcal{T}/2) \times 192$
UpBlock	$(d_{out}, n_{heads}, Attention) = (192, 0, False)$	$\mathcal{T} \times 192$
DecoderBlock		$\mathcal{T} \times 9$

Table 7: Final configuration of `BeatDiff` .

Layer Name	Parameters	Output Dimension
EncoderBlock		$\mathcal{T} \times 192$
DownBlock	$(d_{out}, n_{heads}, Attention) = (192, 0, False)$	$(\mathcal{T}/2) \times 192$
DownBlock	$(d_{out}, n_{heads}, Attention) = (384, 6, True)$	$(\mathcal{T}/4) \times 384$
MiddleBlock	$(d_{out}, n_{heads}, Attention) = (384, 6, True)$	$(\mathcal{T}/4) \times 384$
UpBlock	$(d_{out}, n_{heads}, Attention) = (192, 6, True)$	$(\mathcal{T}/2) \times 192$
UpBlock	$(d_{out}, n_{heads}, Attention) = (192, 0, False)$	$\mathcal{T} \times 192$
DecoderBlock		$\mathcal{T} \times 9$

Table 8: Configuration of deeper network tested.

Optimization: We use the Adam optimizer [49] with the following configuration

- learning rate: 10^{-4},
- Number of epochs: 10^4,
- Batch Size: 1024.

We also use exponential moving average of the network parameters with coefficient 0.9999.

Forward diffusion parameters: For the (forward diffusion) we used the following parameters:

- $\sigma_{\min} = 2 \times 10^{-4}$,
- $\sigma_{\max} = 80$,
- $\sigma_{\text{data}} = 0.5$,
- Importance law of σ for training: $\operatorname{Log}\mathcal{N}(-1.2, 1.2^2\,\mathrm{I})$.

B.1.5 Deeper or Unconditioned Denoisers networks

In this section we test two alternative architectures: a DDM unconditioned on the patient information $\mathcal{P}$ and a deeper DDM with configuration given in Table 8. We find that conditioning over A, S, RR leads to smaller EMD. No substantial improvements were observed when utilizing a deeper network.

B.2 `EM-BeatDiff` parameters

B.2.1 Number of particles

As the number of particles, denoted as M, increases, we observe a corresponding decrease in the discrepancy between the target posterior distribution and the distribution of particles generated by algorithm 1. A critical question arises: what is the optimal value for M that strikes a balance between accuracy and computational efficiency? To approach this question, we first selected a patient from the test dataset and used algorithm 1 to generate 10^3 samples with a high particle count of $M = 10^4$. We consider these samples as our reference representing the target posterior distribution. We then generated 10^3 samples with algorithm 1 for different values of M and calculated the Earth Mover's Distance (EMD) relative to the reference samples. This process helps us to evaluate the convergence of the distribution generated by the algorithm to the posterior as M varies. Figure 7 illustrates the relationship between M and the EMD. From this analysis, $M = 50$ provides an effective equilibrium that provides a reasonable approximation to the posterior distribution while ensuring manageable inference times.

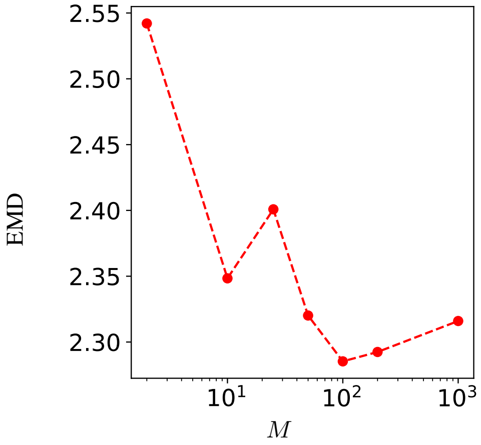


Figure 7: EMD distance between 1000 samples from algorithm 1 with M particles and 1000 samples of algorithm 1 with 10^5 particles, that is considered the standard samples.

B.2.2 Artifacts removal parameters

Choice of artifact basis: We choose the following Fourier basis for removing baseline wander and electrode motion. For $j \in [1, J]$ and $t \in [1, T]$

$$c_j(t) = \begin{cases} \sin(2\frac{j}{J}t(f^a_{\text{max}} - f^a_{\text{min}})/f^s) & \text{if} \quad j \leq J/2\,, \\ \cos(2\frac{j}{J}t(f^a_{\text{max}} - f^a_{\text{min}})/f^s) & \text{else}\,, \end{cases} \tag{B.1}$$

where $J = 200$ is the number of Fourier function is the basis, $f^s = 250$Hz is the sampling frequency, $f^a_{\text{min}} = 0$Hz and $f^a_{\text{max}} = 1$Hz is the typical range of frequency of baseline wander and electrode motion artifact.

Regularization parameters: In Table 9 we display the parameters used in the EM algorithm inside `EM-BeatDiff`. (N_{M}, γ) indicates the number of gradient steps and the learning rate used in the M-step of the EM algorithm algorithm 2. N_{EM} indicates the total number of EM steps used and N_{M} the number of iterations per M step.

Table 9: Parameters used fo `EM-BeatDiff` .

Name	(λ_1, λ_2) from (4.2)	N_{EM}	N_{M}
QT	(0, 1)	10	1
AR (BW)	(10, 10)	10	5
AR (EM)	(10, 5)	10	5
ML (SW)	(0, 1)	10	1
ML (V1-6)	(0, 1)	10	1
AD (MI)	(1, 1)	10	1
AD (LAE)	(1, 1)	10	1
AD (LAD)	(1, 1)	10	1
AD (LQT)	(1, 1)	10	1

B.3 Baseline methods and networks

In this section, we provide implementation details of the adaptations that were needed to test the existing baselines to the problem in hand.

B.3.1 WGAN [1]

In [1], the WGAN is conditioned on 15 categorical heart disease labels. These labels are embedded into a vector of size 100 and concatenated with the latent variable before being inputted into the generator. They are also embedded into a vector of length T (where T is the temporal length of the signal) and then concatenated with the cardiac signal (fake or real) before being inputted into the critic. Embedding maps variables with a finite number of possible values (i.e., categorical variables) into a vectorized representation. However, since in our DDM we condition on scalar variables($\mathcal{P} = (\mathrm{A}, \mathrm{S}, \mathrm{RR})$), in order to compare the results obtained with our DDM and the WGAN, we instead use a multi-layer perceptron (MLP) with the following architecture: a linear layer from 4 to 864, a 1D normalization layer, LeakyReLU, and a linear layer from 864 to 64. This MLP maps the 4-size feature vector $(\tilde{\mathrm{A}}, \tilde{\mathrm{S}}, \tilde{\mathrm{RR}})$ to a 64-vector, which is then used in the same way as the embedding was in the original paper.

B.3.2 SSSD [3]

We adapt the approach described in [3]. We first used the same training procedure to train a network on the same training set as ours. We added conditioning on Sex and changed the sampling frequency from 100hz to 250hz to match ours. To compare with our approach, we generate a 10s according to patient characteristics and using the NSR label from the Physionet dataset. We then use the procedure described in Appendix B.1.1 to extract heartbeats from generated ECG. For all the generation done with SSSD, we use a DDIM [83] schedule with 100 steps and $\eta = 0.01$.

B.3.3 DeScoD [56]

The model proposed in [56] is trained to denoise the beats from the PhysioNet training set, to which noise from the MIH dataset was added. The provided code [†] was modified to train the model on 9-lead ECGs instead of 1-lead ECGs. The 9-lead preprocessed PhysioNet training set, as described in Appendix B.1.1, was used for clean ECGs, and independent random noise was added to each lead. The training procedure followed [56], where noise was sampled from 80% of the first lead of baseline wander noise from the MIT-BIH database [62] and multiplied by a random factor uniformly sampled in $[0.1, 20]$. At test time, the noise was sampled from the remaining 10% of the second lead of the noise, and no multiplication factor was used. The model was run 10 times per ECG, and the average of the 10 outputs was evaluated.

B.3.4 EkGAN [41]

We train the model proposed in [41] to reconstruct I,II,III, V1–6 leads from I (with lr=0.0001 for 100 epochs and then applied weight decay of 0.95 per epoch). For all the inpainting experiments we normalize the ECGs by the max absolute value.

B.3.5 AAE [76]

The model proposed in [76] was trained on the training set described in Appendix B.1.1. The architecture of the model was kept the same, except for the input channels, which were modified to $L = 9$ instead of $L = 1$.

B.4 Classifier network for Classifier Enhancement task

The classifier used for the sex classification task is defined below, using the Flax library [36].

```
 class Classifier(nn.Module):
"""A simple CNN model."""
n_class: int = 2
@nn.compact
def __call__(self, x):
     x = nn.Conv(features=64, kernel_size=(3,))(x)
     x = nn.relu(nn.LayerNorm()(x))
     x = nn.avg_pool(x, window_shape=(2,), strides=(2,))
```

[†] https://github.com/HuayuLiArizona/Score-based-ECG-Denoising

```
        x = nn.Conv(features=128, kernel_size=(3,))(x)
        x = nn.relu(nn.LayerNorm()(x))
        x = nn.avg_pool(x, window_shape=(2,), strides=(2,))
        x = nn.Conv(features=256, kernel_size=(3,))(x)
        x = nn.relu(nn.LayerNorm()(x))
        x = nn.avg_pool(x, window_shape=(2,), strides=(2,))
        x = x.mean(axis=-2)  # flatten
        x = nn.Dense(features=256)(x)
        x = nn.relu(nn.LayerNorm()(x))
        x = nn.Dense(features=self.n_class)(x)
        return x
```

All classifiers were executed with a batch size of 4096, Adam optimizer [49] with learning rate of 0.001 for 10^5 steps. All classifiers achieved 100% accuracy on the training set. Networks weights were initialized always using the same seed.

B.5 Evaluation Metrics

Sum of squared deviations (SSD): The sum of squared deviations between two arrays $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$ is defined as

$$\mathrm{SSD}(x, y) = \textstyle\sum_{i=1}^{d} (x_i - y_i)^2 \,.$$

Maximum absolute deviation (MAD): The maximum absolute deviation between two arrays $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$ is defined as

$$\mathrm{MAD}(x, y) = \max_{i \in [1:d]} |x_i - y_i| \,.$$

Cosine distance (Cos.): The cosine distance between two arrays $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$ is defined as

$$\mathrm{Cos.}(x, y) = \tfrac{x^T y}{\|x\| \|y\|} \,.$$

B.6 Additional Results

B.6.1 Out of distribution (OOD) score [18]

To quantify how unlikely each generated ECG is with respect to the training distribution, we used the OOD-score proposed by [18]. Their method involves using a randomly initialized network, which remains unchanged throughout the process, to produce a "random prior" by associating each training data point (images in the original paper, real or generated ECGs in our case) with a random pattern. Subsequently, a second network is trained to learn this random prior distribution, meaning that the output of the network for a training data point should be close (in terms of L2 distance) to the random pattern from the first network. After training the second network, the OOD-score for an input data point is the distance between the outputs of the two networks. The OOD-score boxplots and the resulting classification ROC curve in figure 8 show that the OOD-scores of the generated ECGs are close to those of the test ECGs, and that the scores for MI ECGs are significantly higher than those for the test and generated ECG. The authors demonstrate the relevance of their score for out-of-distribution data detection by training on four classes of the CIFAR dataset and verifying that, at test time, the score effectively distinguishes test data with the same classes as the training data from those with different classes. In our case, we adopt the same residual network architectures proposed in [18], but replace the 2D convolutions with 1D convolutions, as unidimensional residual networks are known for their efficiency in ECG classification [70]. We use 10 bootstraps and train the corresponding networks for 100 epochs with the Adam optimizer (learning rate=0.001) on healthy patients from the training set.

Table 10: R^2-score between QT measured vs. regressed (intercept: $\mathrm{QT}_0^{\mathrm{c}}$, slope: $\mathrm{QT}_1^{\mathrm{c}}$) as a function of RR, in generated samples, with 95%-CLT intervals over the test-set.

METHOD	R^2-SCORE	EXPRESSION
Framingham	0.88 ± 0.03	$\mathrm{QT} = \mathrm{QT}_0^c + 0.154(1 - \mathrm{RR})$
Bazett	0.47 ± 0.04	$\mathrm{QT} = \mathrm{QT}_1^c \sqrt{\mathrm{RR}}$
Baz. (offset)	0.98 ± 0.00	$\mathrm{QT} = \mathrm{QT}_0^c + \mathrm{QT}_1^c \sqrt{\mathrm{RR}}$
Fridericia	0.94 ± 0.02	$\mathrm{QT} = \mathrm{QT}_1^c \sqrt[3]{\mathrm{RR}}$
Frid. (offset)	0.98 ± 0.00	$\mathrm{QT} = \mathrm{QT}_0^c + \mathrm{QT}_1^c \sqrt[3]{\mathrm{RR}}$

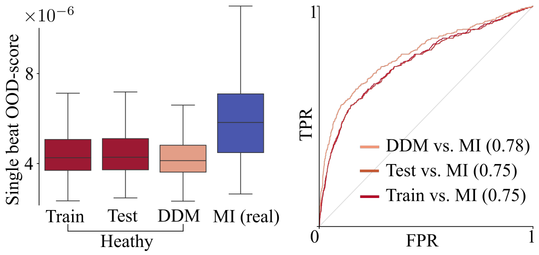


Figure 8: Out-of-distribution evaluation. **Left.** Box-plot of OOD-score for train, test, generated (Gen) and MI heart beats. **Right.** ROC curves for classification between train/test/gen and MI based on OOD-score.

B.6.2 Prediction of QT from RR

In this section we provide supplementary results for the experiments of the prediction of corrected QT: we provide the R^2-score between QT measured vs. regressed (intercept: $\mathrm{QT}_0^{\mathrm{c}}$, slope: $\mathrm{QT}_1^{\mathrm{c}}$) as a function of RR, in generated samples, with 95%-CLT intervals over the test-set, for several corrected QT formulas in Table 10.

B.6.3 Ablation Study cardiac anomaly

In this section, we consider for each medical condition the AUC score of the anomaly detection task while varying the way that we use the conditioning ECGs.

The configurations are shown in Table 11 and we describe now for each configuration their electrophysiological motivation.

- I, II, III: This choice of configuration implies generating the precordial leads V1–6 from the limb leads. It is coherent when the abnormality is expected to manifest in a localized way in one of the precordial leads.

- QRS: This choice of configuration implies generating the ST segment (ventricle repolarization) conditionally on the QRS observation over all the leads. It is particularly pertinent when an T-wave abnormality is expected.

- ST: This choice of configuration implies generating the QRS and P-wave from the ST segment. It is particularly coherent when the abnormality is expected in the beginning of the signal (i.e., the P-wave and QRS).

In Table 12 we display the AUC scores obtained using $1 - R^2$ between the patient signal and the mean posterior signal from `EM-BeatDiff` over the non-observed part of the signals.

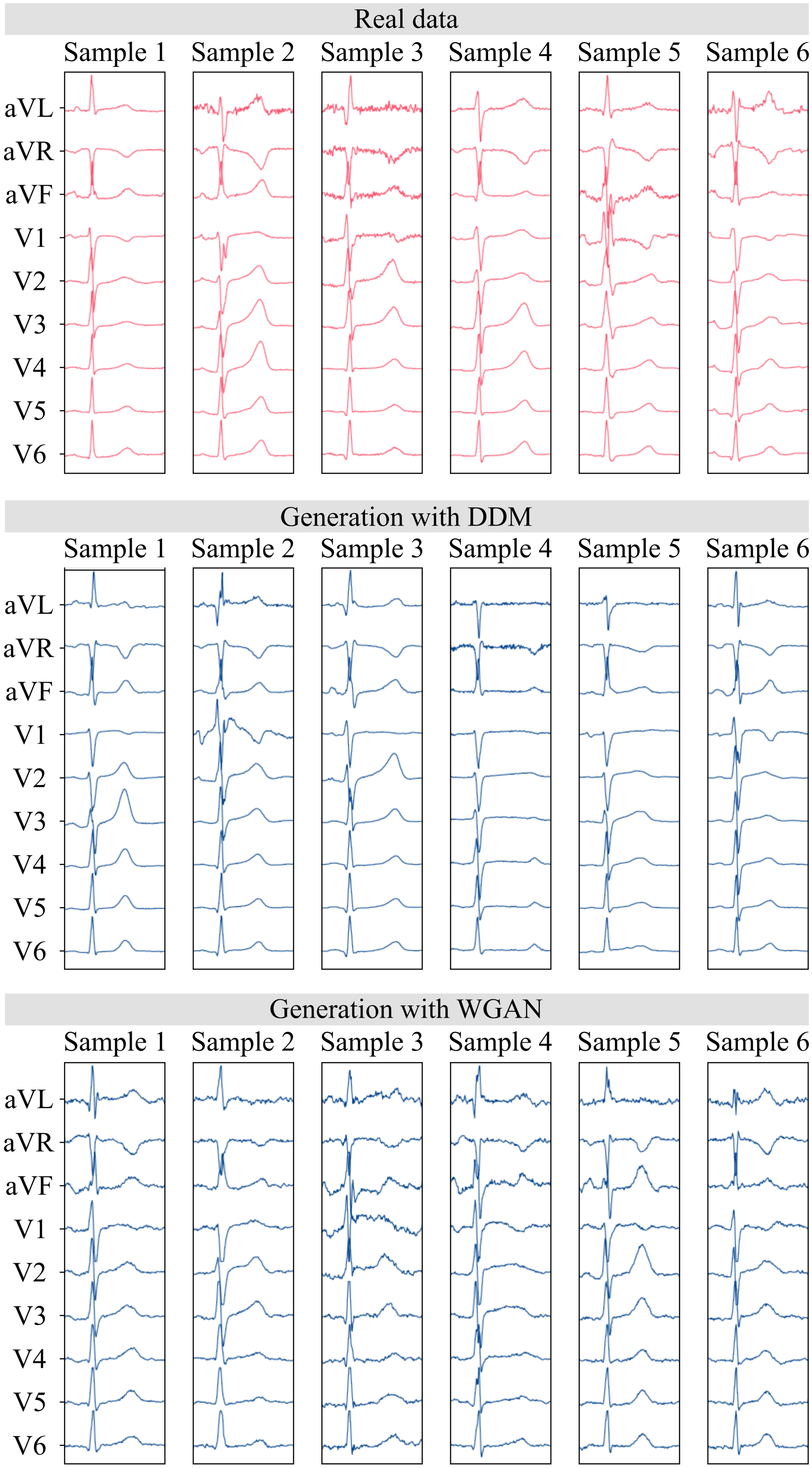


Figure 9: Real and generated ECG heart beat with DDM and WGAN.

Table 11: Configurations tested in the ablation study.

Conditioning	$(\bar{L}, \bar{T})$	A_θ	$\bar{A}_\theta$	B_θ	D_θ	$\bar{D}_\theta$
I, II, III	$(3, T)$	$\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{T\times T}$	(4.1)	$\mathrm{diag}(\sigma_{1:\bar{L}})$	$\mathrm{I}_{T\times T}$
QRS	$(L, 70)$	$\mathrm{I}_{L\times L}$	$\mathrm{I}_{\bar{T}\times T}$	(4.1)	$\mathrm{diag}(\sigma_{1:L})$	$\mathrm{I}_{T\times \bar{T}}$
ST	$(L, 106)$	$\mathrm{I}_{L\times L}$	$\left[\mathbf{0}_{\bar{T},T-\bar{T}}; \mathrm{I}_{\bar{T}\times\bar{T}}\right]^T$	(4.1)	$\mathrm{diag}(\sigma_{1:L})\,\mathrm{I}_{\bar{L}\times L}$	$\mathrm{I}_{\bar{T}\times T}$

Table 12: Anomaly detection abblation study. Confidence intervals are obtained by running 10 times `EM-BeatDiff` per heartbeat.

Conditioning	MI	LAD	LAE	LQT
I, II, III	$\mathbf{84.82 \pm 0.01}$	91.63 ± 0.03	$\mathbf{79.02 \pm 0.07}$	77.40 ± 0.11
QRS	81.88 ± 0.02	70.45 ± 0.09	62.89 ± 0.06	$\mathbf{84.73 \pm 0.04}$
ST	84.05 ± 0.01	$\mathbf{93.06 \pm 0.03}$	78.33 ± 0.05	79.72 ± 0.06

B.6.4 Failure cases of DeScoD in artifact removal

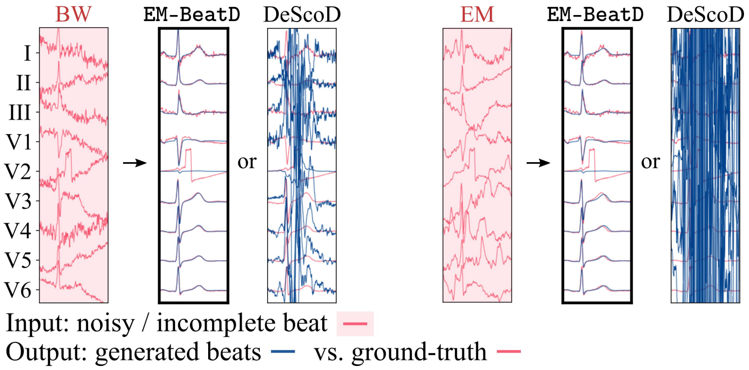


Figure 10: Failure case of DeScoD ([56]) on baseline wander (left) and electrode motion artifact (right).

Some (rare) ECGs in the test database already contain artifacts before the addition of noise from the MIT-BIH dataset. DeScoD is unable to effectively denoise these ECGs and produces inconsistent results because the noise in these ECGs is outside of the training domain of the model. The example in figure 10 illustrates that training an artifact removal model in a supervised manner is specific to the MIT-BIH database and does not allow for the removal of artifacts not found in this database, even if they share the same characteristics (low frequency). On the other hand, our approach, which is not trained in a supervised manner, is more generalizable.

B.6.5 10s ECGs and arrythmic data

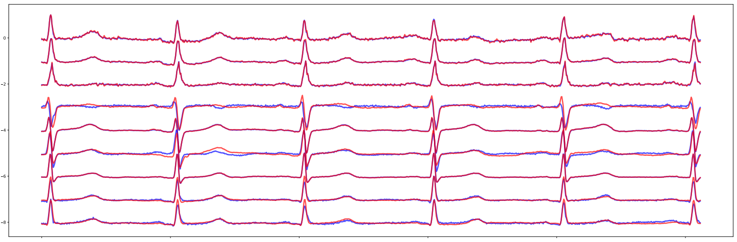


Figure 11: 9 lead (I, II, III, V1–V6 from top to bottom) 5 second healthy ECG reconstruction. Red indicates the ground-truth and blue the generated ECGs conditionned on leads I, II, III, V2 and V4.

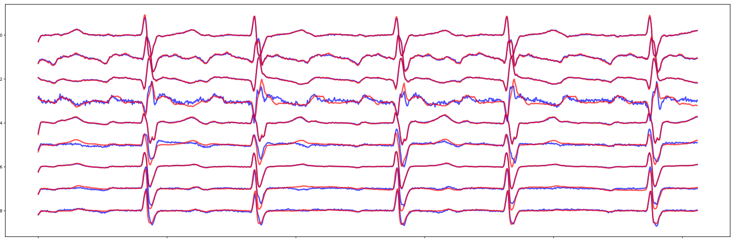


Figure 12: 9 lead (I, II, III, V1–V6 from top to bottom) 5 second AF (Atrial fibrillation) ECG reconstruction. Red indicates the ground-truth and blue the generated ECGs conditionned on leads I, II, III, V2 and V4.

B.7 Computational resources

All the experiments were run in an internal server equipped with 8 A40 Nvidia GPUs, each with 46Gb of available memory. The server CPU has 72 threads and a total live memory of 378 Gb. All the data creation and preprocessing task were used CPU workers while all the neural network related tasks used GPU workers.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We show numerically that our generative model is either on pair with the current models or better, while being lightweight. We show that using this model for solving inverse problems without additional training results in performance on par with models trained for the specific inverse problem reconstruction problem.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a limitation sections in the main paper Section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper introduces no new theoretical results, but a method that combines other methods such as Sequential Monte Carlo and Expectation Maximization algorithm. We provide the relevant references where theoretical results can be found.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailled description of architecture, preprocessing and hyper parameters. Furthermore, we provide a anonymous git link for the code. Reproducing our experiments is possible provided that a comparable computational budget is available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use for all the experiments the open source data from the Physionet Challange 2021 and the code is available via an anonymous github. Detailed description of the architecture, training procedure and hyper-parameters are provided in the supplementary material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed description of the architecture, training procedure and hyper-parameters are provided in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide confidence intervals for all the metrics that are either obtained by testing the algorithm on different data or by repeating the experiment with different initialization when pertinent. We did not provide training with several different train test splits due to the high-computational demands for the training of each model, but all the evaluation of the models and the inverse problem parts possesses the equivalent confidence intervals.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide in Table 1 an overview of all the computational burden of each tested model. In Appendix B.7 we describe the hardware used for the experimental section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The database used is available on open source for some years and has been anonymized.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We describe in the Limitations section that any usage of the tools described in this paper on real patients need to pass through the appropriate clinical trials and that in no way we intend the available code to be used *per se* in real patient data.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: N/A

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the baselines and data providers are properly cited and all the license and terms of used are respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and model are documented both in the appendix of the current paper and in the git repository. We will continue to work to improve the documentation of the anonymous git repository.[‡]

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

[‡] Anonymous code available at `https://anonymous.4open.science/r/ecg_inpainting-44A6`

Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.