Xi'an Jiaotong-Liverpool University
西交利物浦大学

radarODE-MTL: A Multi-Task Learning Framework with Eccentric Gradient Alignment for Robust Radar-Based ECG Reconstruction

- **Part 1 – Background and Problems**
- **Part 2 – Proposed Methods**
- **Part 3 – Result Evaluations**

Yuanyuan Zhang

GitHub: https://github.com/amazing0844 LinkedIn: https://www.linkedin.com/in/yuanyuan-16

Part 1 - Background and Problems

- **Part 1 – Background**
- **Part 2 – Existing Methods and Limitations**

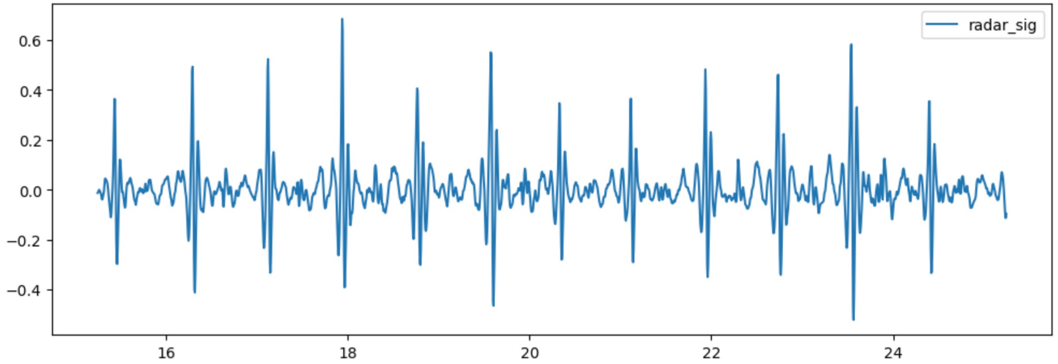
Background

General Obstacles:

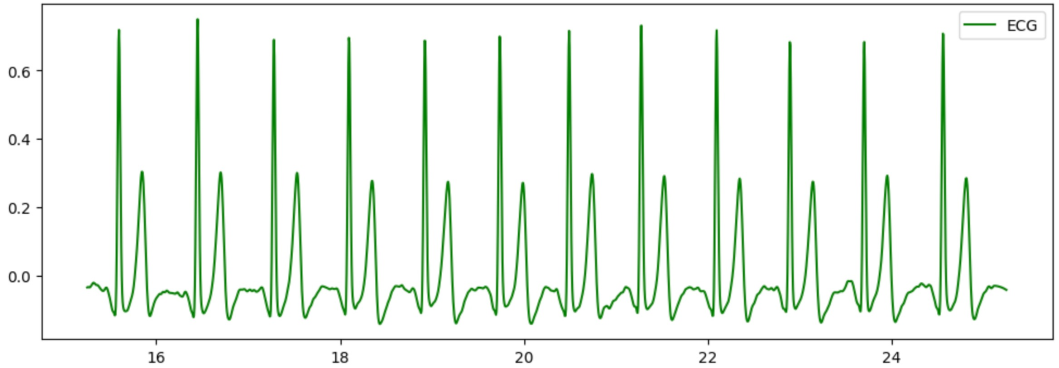
1. To extract fine-grained cardiac features (i.e., ECG, SCG)
2. To mitigate the real-world noise (i.e., body movement, multi-path propagation)
3. To utilize the limited data with proper data augmentation methods.
4. To enable the transformation of ECG recovery with the support of theoretical signal model.

Project Specification:

This project aims to recover the accurate ECG signals from the radar signals.



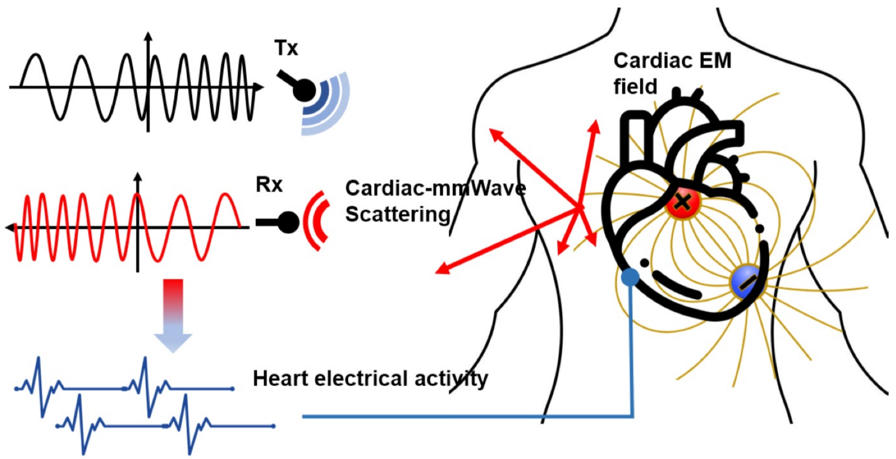
Radar signals (mechanical)



ECG (electrical)

Existing Methods

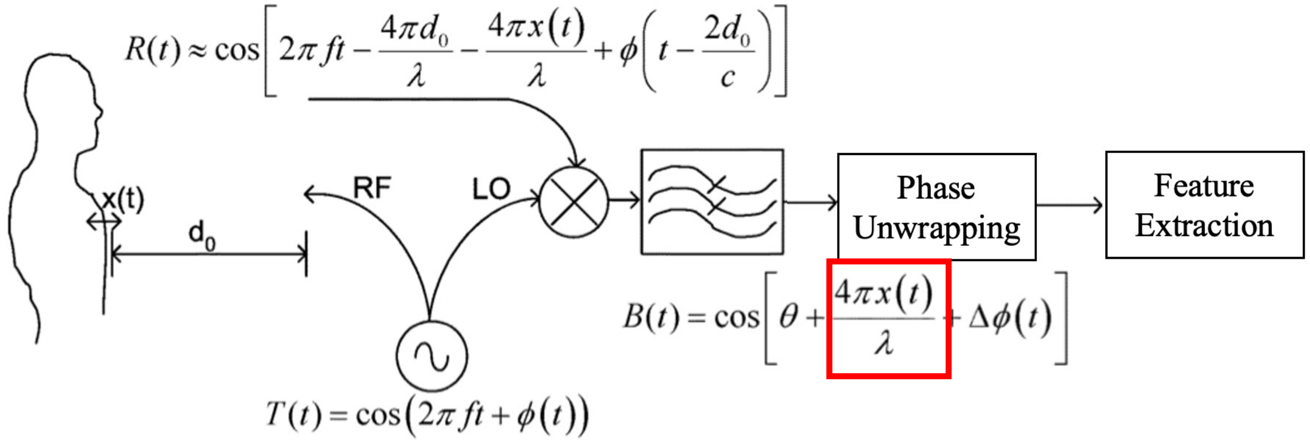
First Category: Sensing the EM field [1]



Limitations:

1. Complex model due to Green Function and biological ionic concentration
2. The solution is hard to be calculated and is vulnerable to changing environment

Second Category: Sensing the vibration from chest region



Limitations:

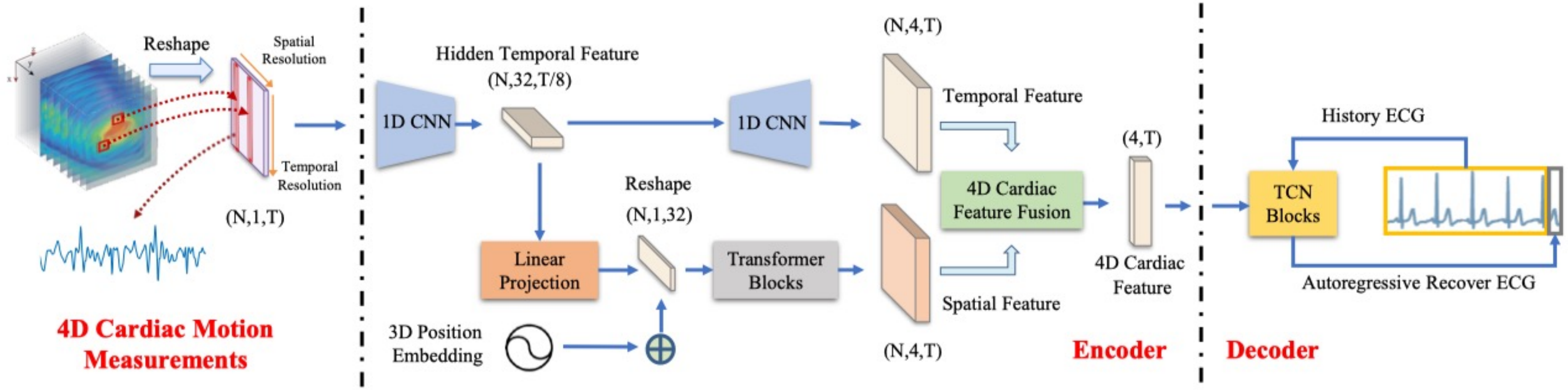
1. Lack of the theoretical model for radar-ECG signal decoupling (from mechanical domain to electrical domain)
2. Rely on purely data-driven methods

[1] C. Xu, H. Li, Z. Li, H. Zhang, A. S. Rathore, X. Chen, K. Wang, M.-c. Huang, and W. Xu, “CardiacWave: A mmWave-based scheme of non-contact and high-definition heart activity computing,” Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), vol. 5, no. 3, pp. 1–26, Sep. 2021.

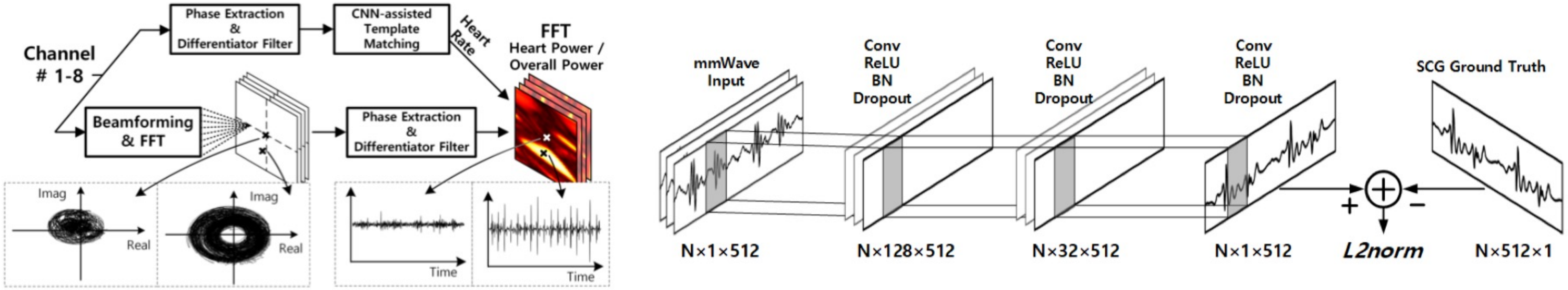
Existing Methods

Examples on the methods from the second category:

[2]

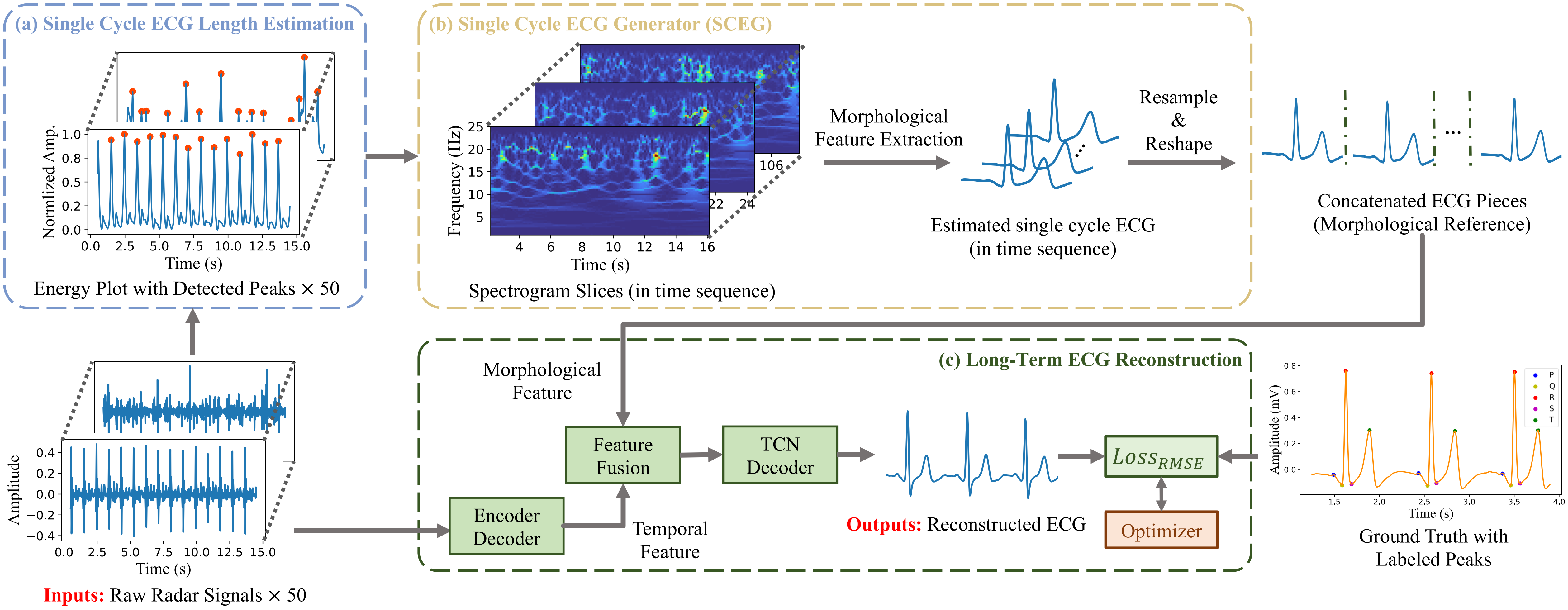


[3]



[2] J. Chen, D. Zhang, Z. Wu, F. Zhou, Q. Sun, and Y. Chen, "Contactless electrocardiogram monitoring with millimeter wave radar," IEEE Transactions on Mobile Computing, pp. 1–17, Oct. 2022.
[3] U. Ha, S. Assana, and F. Adib, "Contactless seismocardiography via deep learning radars," in Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 114, Apr. 2020.

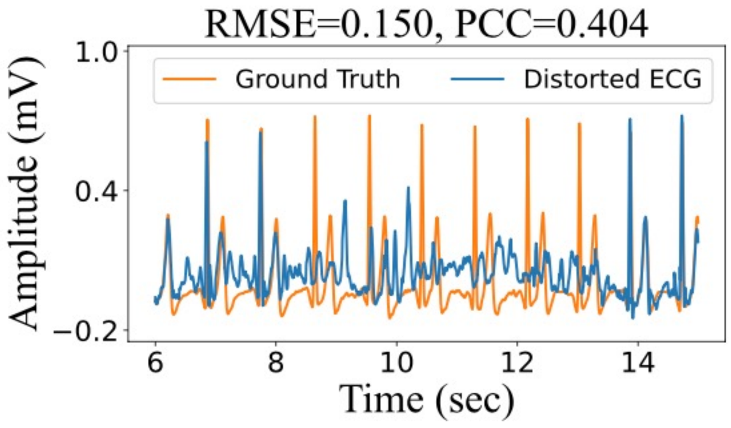
Existing Methods: radarODE



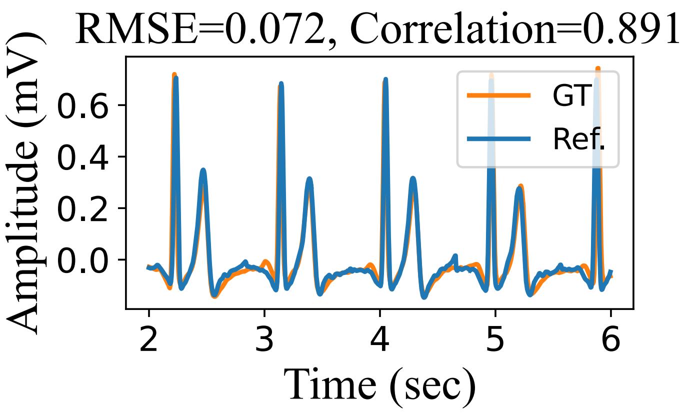
Research Gap

Research gap in radar-based ECG recovery:

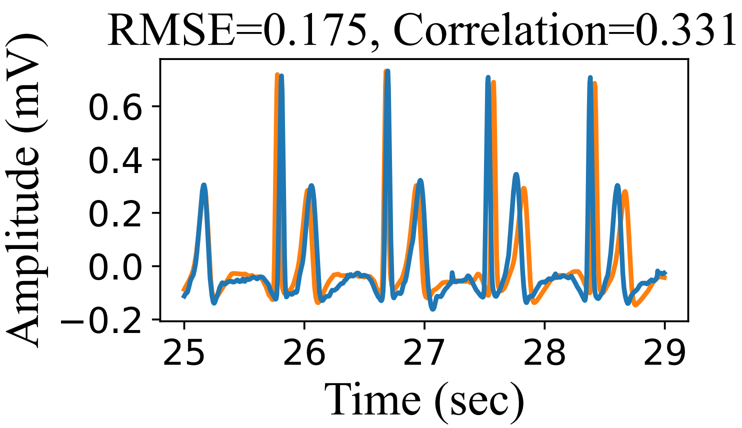
1. The domain transformation (radar to ECG) lacks theoretical explanation.
2. Noise robustness is never considered or validated in ECG recovery, especially for the DL model itself.
3. The misalignment issue in the method based on single cardiac cycle.



Distorted by noise



Well Aligned



Misaligned

Part 2 – Proposed Methods and Results

- Part 1 – Preliminary
- Part 2 – Overview of radarODE-MTL
- Part 3 – Framework Design
- Part 4 – EGA optimization strategy
- Part 5 – Results and Analyses

radarODE with Multi-task Learning (MTL)

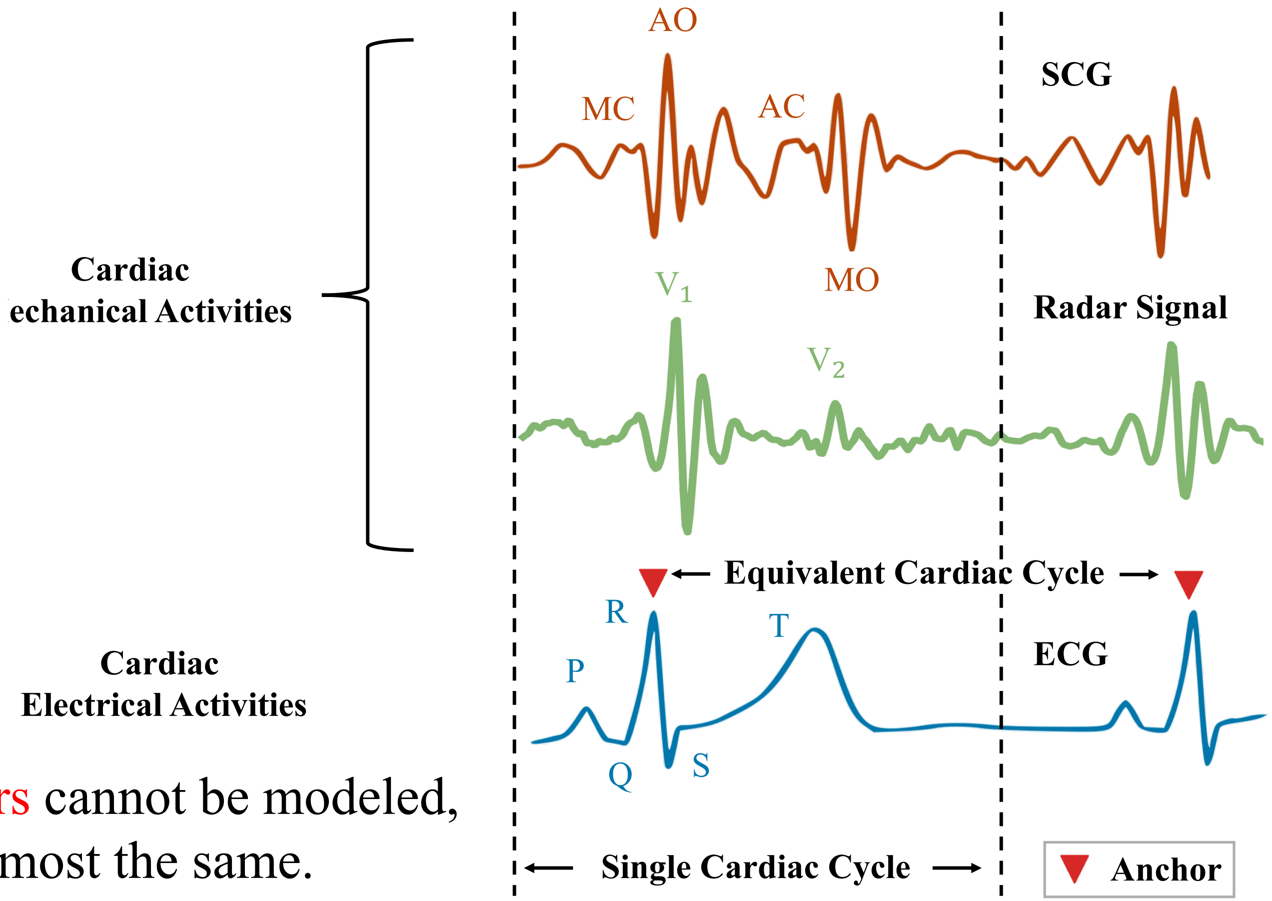
Contributions:

1. Deconstruct the radar-based ECG reconstruction into three tasks and designs an MTL framework named radarODE-MTL in an end-to-end manner, avoiding the reintroduction of the time-domain noises and hence improving the noise robustness.
2. A novel optimization strategy is proposed for MTL learning by eccentrically aligning the task gradients, aiming to adaptively alter the task priority to ensure efficient learning progress for the tasks with disparate complexity while preventing the negative transfer phenomenon at the same time.
3. To the best of our knowledge, this is the first work that investigates the noise robustness of the deep-learning model in radar-based ECG recovery. Sufficient experiments show that the proposed radarODE-MTL with eccentric gradient alignment (EGA) outperforms other frameworks and optimization strategies under various noise conditions.

- Task 1: The reconstruction of the morphological features of single-cycle ECG pieces (has been realized in radarODE).
- Task 2: The prediction of the length of each single cardiac cycle.
- Task 3: The identification of the R peaks in terms of long-term monitoring periods.

Task Deconstruction & Problem Statement

- Task 1: The reconstruction of the morphological features aims to learn the mapping function $\mathcal{F}(\cdot)$ only for the single-cycle cardiac signals from $x(t)$ to the ECG pieces $x_{ecg}(t) = \mathcal{F}(x(t))$.
- Task 2: The detection of R peaks (Anchors) is equivalent to find $\mathbf{R} = \{T_1^1, T_1^2, \cdots, T_1^K\}$ according to the central frequency f_1^k of v_1^k, as shown in Figure 1.
- Task 3: The prediction of the length of a single cardiac cycle is equivalent to find the distance between successive anchors (i.e., peak-to-peak interval (PPI)) using differentiation, with $\mathbf{PPI} = diff(\mathbf{R})$, as shown in Figure 1.



Advantages:

1. The transformation between arbitrary radar/ECG pairs cannot be modeled, while ECG patterns within every cardiac cycle are almost the same.
2. Direct estimation of the anchors and cardiac cycle lengths avoid the impact of error accumulation in long-term ECG recovery.
3. The latent information needed in different tasks can be broadcasted across layers to improve the generalization of the model and the performance of every single task.

Preliminary - Signal Model

Signal Model:

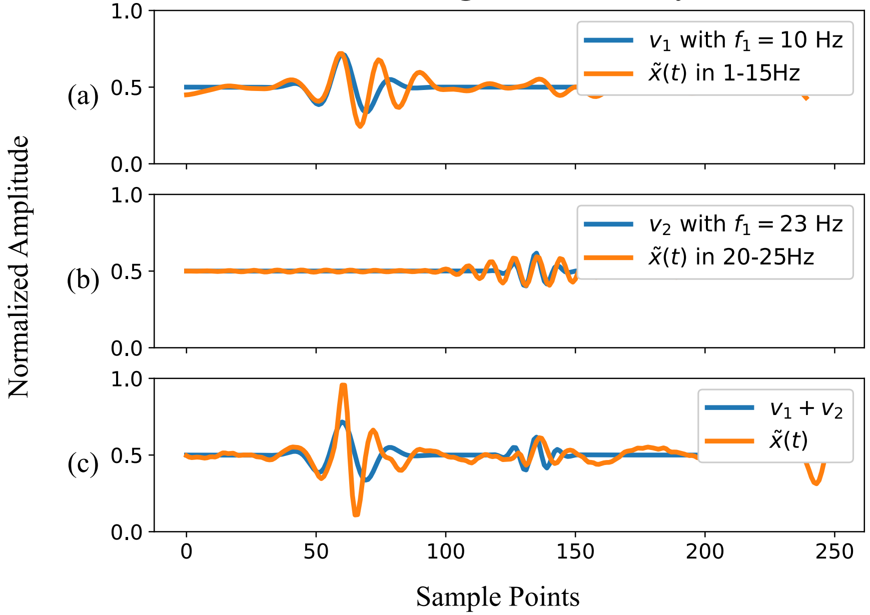
1. The cardiac vibration x(t) can be extracted from the phase variation of the received radar signal as:

$$\Delta\phi(t) = \frac{4\pi x(t)}{\lambda} \quad \text{with} \quad x(t) = x_c(t) + n_{abr}(t) + n_{con}(t)$$

n_{abr}: abrupt noises (e.g., RBM), ruining all the cardiac features

n_{con}: constant noises (e.g., random radar directions, long-distance monitoring, thermal noise), causing low SNR.

True Radar signal with Synthesis



2. Based on the previous intuition, we model the $x(t)$ as two prominent vibrations for K cardiac cycles as:

$$x(t) = \sum_{k=1}^{K} v_1^k(t) + \sum_{k=1}^{K} v_2^k(t) + n_{abr}(t) + n_{con}(t) \quad \text{with} \quad \begin{aligned} v_1^k(t) &= \mathrm{a}_1^k \cos(2\pi f_1^k t)\exp\left(-\frac{(t-T_1^k)^2}{{b_1^k}^2}\right) \\ v_2^k(t) &= \mathrm{a}_2^k \cos(2\pi f_2^k t)\exp\left(-\frac{(t-T_2^k)^2}{{b_1^k}^2}\right) \end{aligned}$$

a_1, a_2 → amplitude of the vibrations, T_1, T_2 → when the vibrations happen

b_1, b_2 → length/width of the vibration, f_1, f_2 → central frequency of the vibration

Signal Pre-processing to Spectrograms

Localization of T_1, T_2 using synchrosqueezed wavelet transform (SST) [6]:

1. Calculate the wavelet transform of $x(t)$:

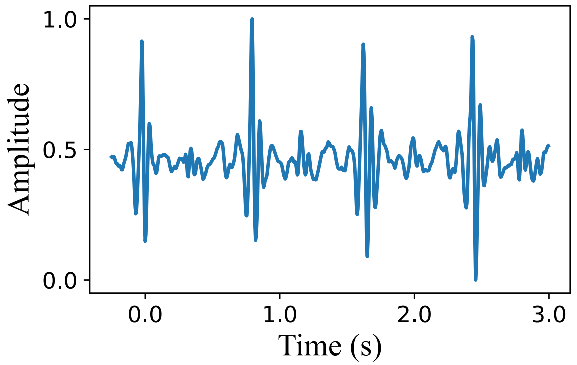
$$W_{\tilde{x}}(a,b) = \int \tilde{x}(t) a^{-1/2} \psi^* \left(\frac{t-b}{a} \right) dt$$

2. Calculate the candidate instantaneous frequency as:

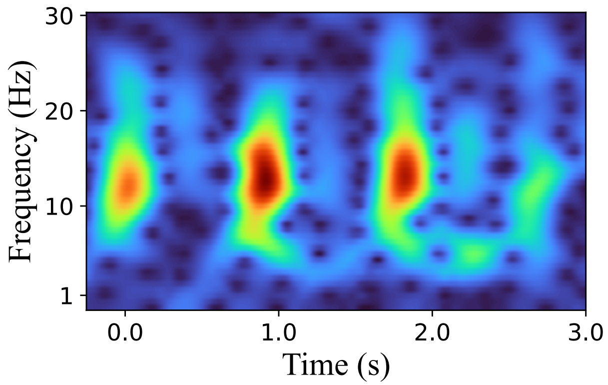
$$f_{\tilde{x}}(a,b) = -2\pi i \left(W_{\tilde{x}}(a,b) \right)^{-1} \frac{\partial W_{\tilde{x}}(a,b)}{\partial b}$$

3. Concentrate the energy along the candidate instantaneous frequency:

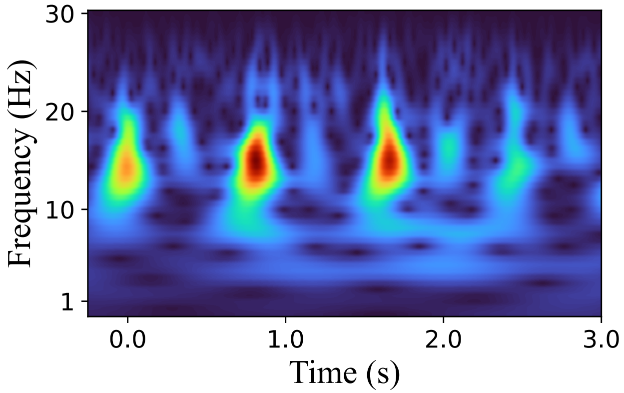
$$T_{\tilde{x}}(2\pi f, b) = \int_{A(b)} W_{\tilde{x}}(a,b) a^{-3/2} \delta \left(2\pi f_{\tilde{x}}(a,b) - 2\pi f \right) df$$



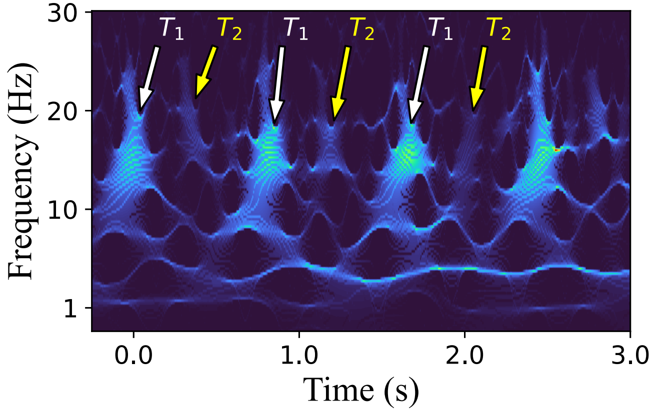
Radar Signal $\tilde{x}(t)$



STFT



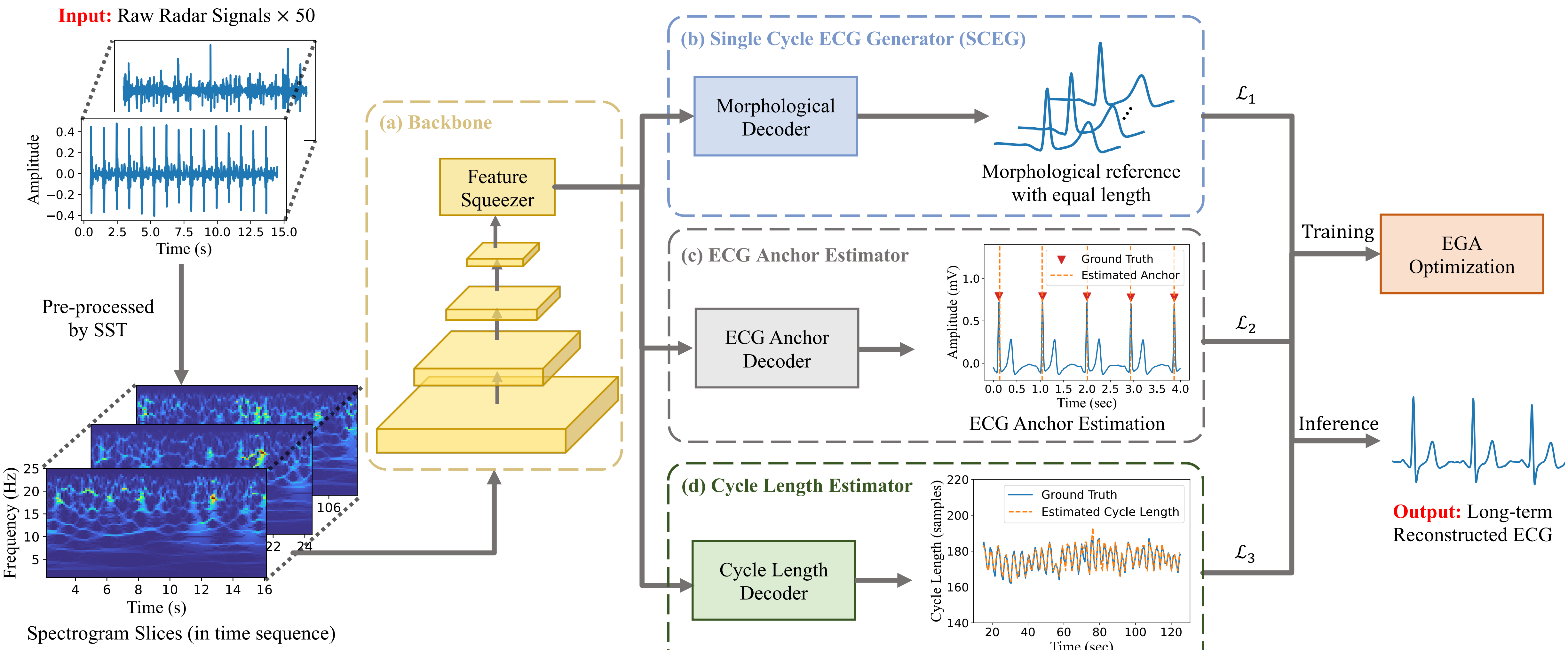
CWT



SST

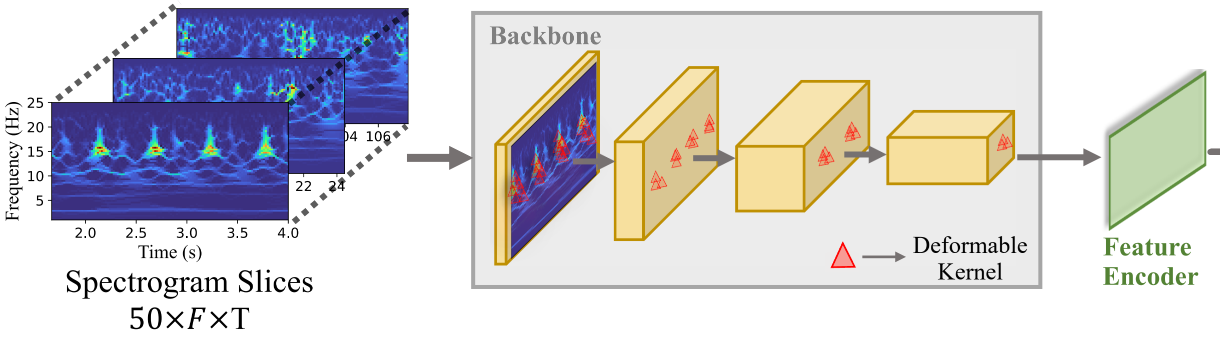
[6] I. Daubechies, J. Lu, and H.-T. Wu, "Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool," Applied and computational harmonic analysis, vol. 30, no. 2, pp. 243–261, Aug. 2011.

radarODE with Multi-task Learning (MTL)

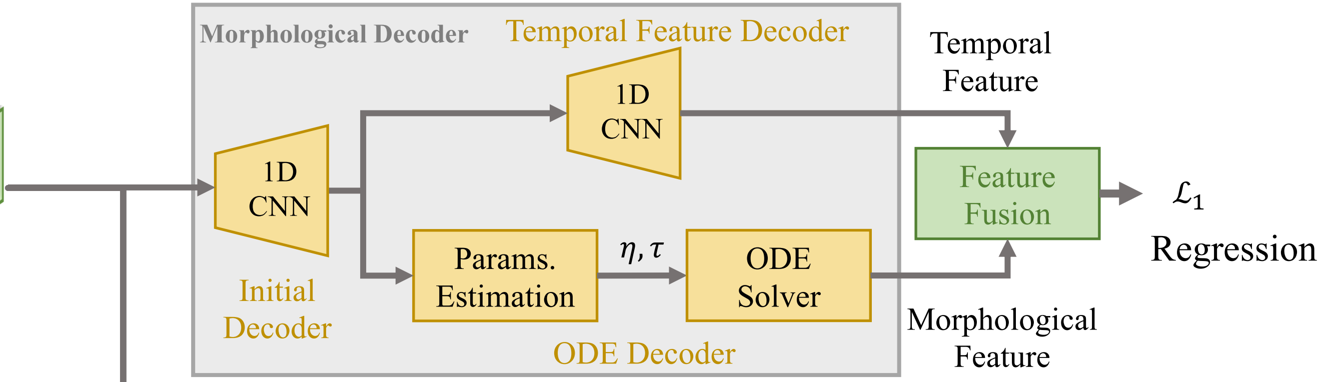


radarODE Framework

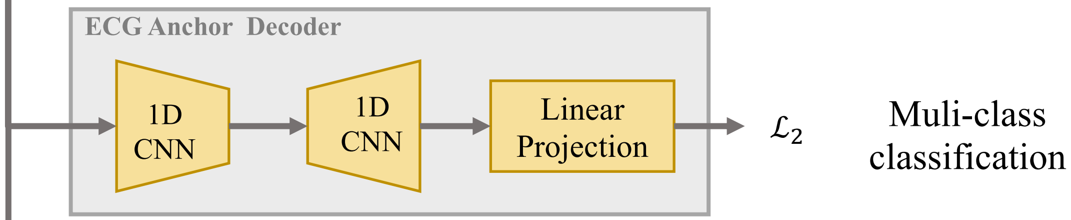
Backbone & Feature Encoder:



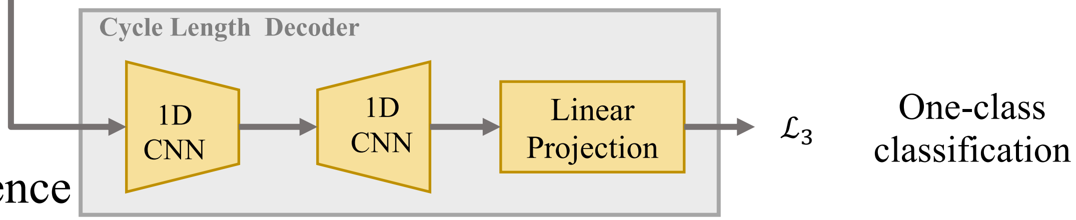
Single Cycle ECG Generator (SCEG):

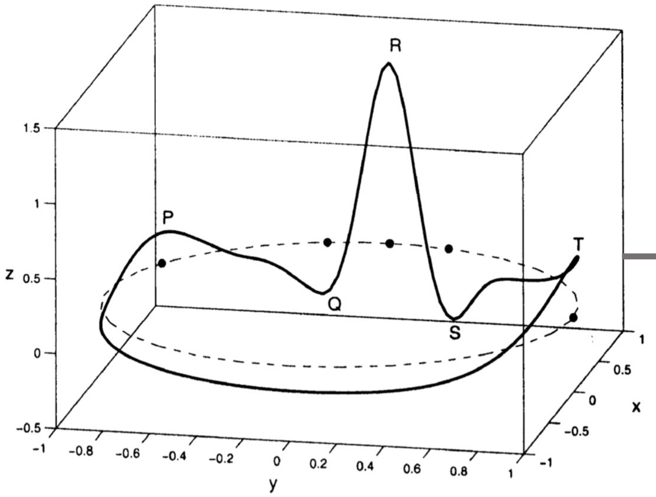


ECG Anchor Estimator:

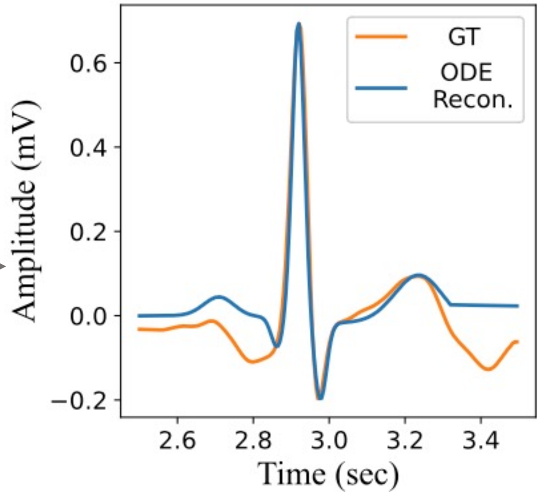


Cycle Length Estimator:



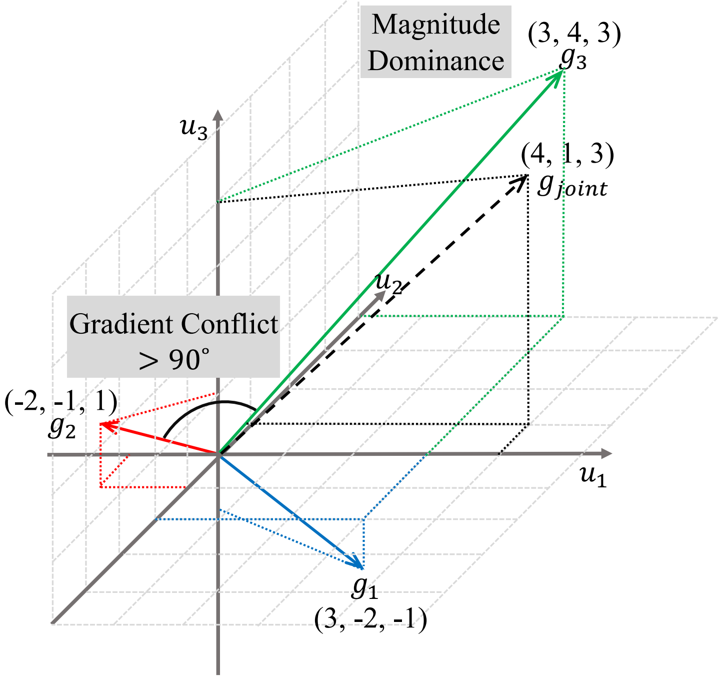


Result from ODE solver

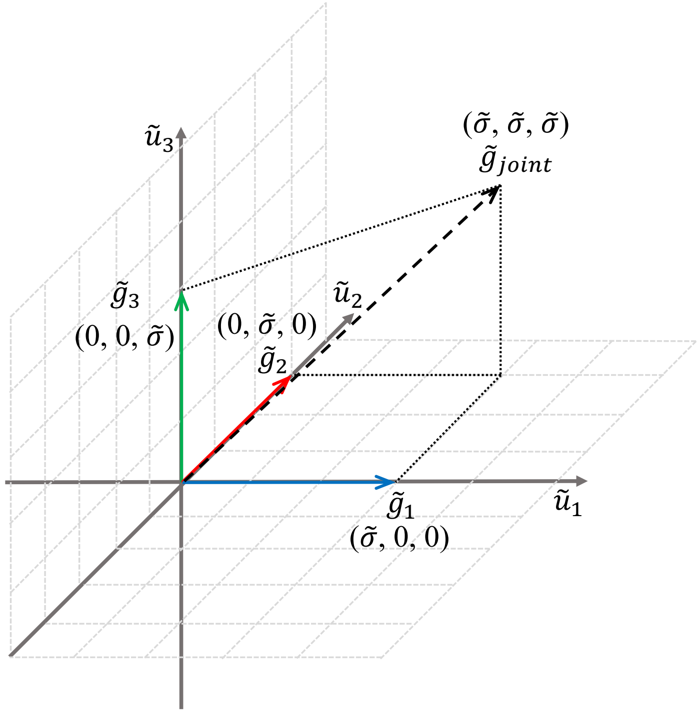


Rigid Morphological reference

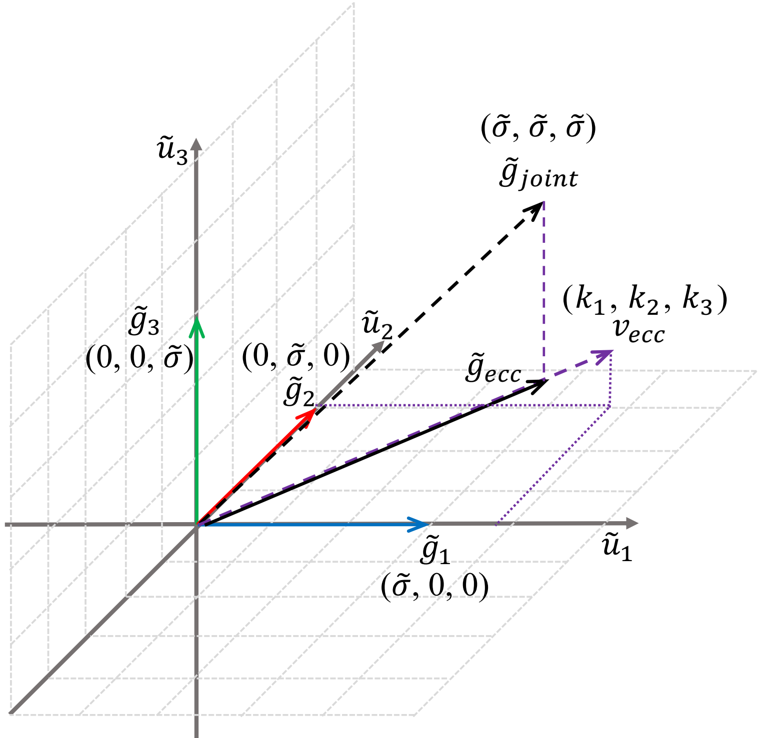
Eccentric gradient alignment (EGA)



(a) Original gradient space with gradient conflict and magnitude dominance



(b) The projection of the original gradient space into the orthogonal space with equal "learning rate"



(c) The implementation of eccentric gradient alignment to slant the joint gradient to the hard task by introducing the eccentric vector

Measurement of learning progress, with normalization (sum = task number)

$$k_i(t) = \mathrm{softmax}(lr_i(t-1)) = \frac{n \exp\left(lr_i(t-1)/T\right)}{\sum_{j=1}^{n} \exp\left(lr_j(t-1)/T\right)} \quad \text{with} \quad lr_i(t-1) = \frac{\mathcal{L}_i(t-1)}{\mathcal{L}_i(t_{warm})}$$

Eccentric gradient alignment (EGA)

Define a MTL problem for n tasks:

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^m} \left\{ \mathcal{F}(\theta) \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i(\theta) \right\} \quad \theta \in \mathbb{R}^m$$ for shared parameter space

The original task-specific gradient and the gradient matrix :

$$\boldsymbol{g}_i = \nabla_\theta \mathcal{L}_i(\theta), i \in [n] \qquad \boldsymbol{G} = \{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_i\} \in \mathbb{R}^{n \times m}$$

Projection to orth. space

1. Formulate the projection problem:

$$\min \|\boldsymbol{g}_{joint} - \tilde{\boldsymbol{g}}_{joint}\|_2^2 \;\text{ s.t. }\; \tilde{\boldsymbol{G}}\tilde{\boldsymbol{G}}^\intercal = \boldsymbol{I} \quad \text{with} \quad \begin{aligned} \boldsymbol{g}_{joint} &= \boldsymbol{G}^\intercal \boldsymbol{w} \\ \tilde{\boldsymbol{g}}_{joint} &= \tilde{\boldsymbol{G}}^\intercal \boldsymbol{w} \end{aligned} \quad \boldsymbol{w} = [1, \cdots, 1]^\intercal$$

Then we have:

$$\left\|\boldsymbol{g}_{joint} - \tilde{\boldsymbol{g}}_{joint}\right\|_2^2 = \left\|\boldsymbol{G}^\intercal \boldsymbol{w} - \tilde{\boldsymbol{G}}^\intercal \boldsymbol{w}\right\|_2^2 \leq \|\boldsymbol{G}^\intercal - \tilde{\boldsymbol{G}}^\intercal\|_F^2 \|\boldsymbol{w}\|_2^2$$

2. The final projection problem:

$$\min_{\tilde{\boldsymbol{G}}} \|\boldsymbol{G} - \tilde{\boldsymbol{G}}\|_F^2 \quad \text{s.t.} \quad \tilde{\boldsymbol{G}}\tilde{\boldsymbol{G}}^\intercal = \boldsymbol{I}$$

Algorithm 1 EGA Optimization Strategy for MTL

Input: Loss values for n tasks $[\mathcal{L}_1, \cdots, \mathcal{L}_i], i \in [n]$,
T for softmax and t_{warm} for warmup epoch
INITIALIZATION:
- Unit eccentric vector $\boldsymbol{v}_{ecc} = [1, \cdots, 1]^\intercal \in \mathbb{R}^n$
FOR EACH ITERATION:
- Get the current epoch as t
- Get task-specific gradient $\boldsymbol{g}_i = \nabla_\theta \mathcal{L}_i(\theta), i \in [n]$
- Form gradient matrix $\boldsymbol{G} = \{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_i\} \in \mathbb{R}^{n \times m}$
- Eigen decomposition of Gram matrix as in (13): $\boldsymbol{G}\boldsymbol{G}^\intercal = \boldsymbol{U}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\intercal\right)\boldsymbol{U}^\intercal$ with eigenvalues $\boldsymbol{\lambda}$
- Get scaling factor for gradient normalization: $\tilde{\sigma} = \min(\sqrt{\boldsymbol{\lambda}})$
- Calculated the orthogonal and normalized gradient matrix as in (15): $\tilde{\boldsymbol{G}} = \tilde{\sigma}\boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\intercal\boldsymbol{G}$

if $t = t_{warm}$ **then**
  - Record the loss values for all the tasks $\mathcal{L}_i(t_{warm})$
else if $t > t_{warm}$ **then**
  - Calculate the task difficulty weights as in (17): $k_i(t) = \text{softmax}(lr_i(t-1))$
  - Form eccentric vector $\boldsymbol{v}_{ecc} = [k_1, \cdots, k_i]^\intercal$
end if
Output: $\tilde{\boldsymbol{g}}_{ecc} = \tilde{\boldsymbol{G}}^\intercal \boldsymbol{v}_{ecc}$ for optimization

Eccentric gradient alignment (EGA)

3. Find the solution for:

$$\min_{\tilde{G}} \|G - \tilde{G}\|_F^2 \quad \text{s.t.} \quad \tilde{G}\tilde{G}^\intercal = I$$

4. Bring the solution from orthogonal Procrustes problem using SVD:

$$\begin{array}{lll} \min & \|A\Omega - B\|_F & A, B \in R^{m\times n} \\ \text{s.t.} & \Omega^\top \Omega = I & C = B^\top A \end{array} \quad \text{with solution} \quad \begin{array}{c} U\Sigma V^\top = C \\ \Omega = V U^\top \end{array}$$

5. The new gradient matrix (using eigen decomposition) is:

$$\tilde{G} = UV^\intercal \qquad GG^\intercal = U\left(\Sigma\Sigma^\intercal\right)U^\intercal$$

6. The final solution is:

$$\tilde{G} = U\Sigma^{-1}U^\intercal G$$ with unit singular values

The gradient conflict resolved □

Algorithm 1 EGA Optimization Strategy for MTL

1: **Input:** Loss values for n tasks $[\mathcal{L}_1, \cdots, \mathcal{L}_i], i \in [n]$, T for softmax and t_{warm} for warmup epoch

INITIALIZATION:

2: - Unit eccentric vector $v_{ecc} = [1, \cdots, 1]^\intercal \in \mathbb{R}^n$

FOR EACH ITERATION:

3: - Get the current epoch as t

4: - Get task-specific gradient $g_i = \nabla_\theta \mathcal{L}_i(\theta), i \in [n]$

5: - Form gradient matrix $G = \{g_1, \cdots, g_i\} \in \mathbb{R}^{n\times m}$

6: - Eigen decomposition of Gram matrix as in (13): $GG^\intercal = U\left(\Sigma\Sigma^\intercal\right)U^\intercal$ with eigenvalues λ

7: - Get scaling factor for gradient normalization: $\tilde{\sigma} = \min(\sqrt{\lambda})$

8: - Calculated the orthogonal and normalized gradient matrix as in (15): $\tilde{G} = \tilde{\sigma}U\Sigma^{-1}U^\intercal G$

9: **if** $t = t_{warm}$ **then**

10: - Record the loss values for all the tasks $\mathcal{L}_i(t_{warm})$

11: **else if** $t > t_{warm}$ **then**

12: - Calculate the task difficulty weights as in (17): $k_i(t) = \text{softmax}(lr_i(t-1))$

13: - Form eccentric vector $v_{ecc} = [k_1, \cdots, k_i]^\intercal$

14: **end if**

15: **Output:** $\tilde{g}_{ecc} = \tilde{G}^\intercal v_{ecc}$ for optimization

Eccentric gradient alignment (EGA)

Gradient Normalization

1. The magnitude of each gradient ∝ eigenvalues $\boldsymbol{\lambda}$ [4]

$$\tilde{\boldsymbol{G}} = \boldsymbol{U}\boldsymbol{V}^{\intercal} \quad \text{with} \quad \boldsymbol{G}\boldsymbol{G}^{\intercal} = \boldsymbol{U}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\intercal}\right)\boldsymbol{U}^{\intercal}$$

2. Get scaling factor for gradient normalization:

$$\tilde{\sigma} = \min(\sqrt{\boldsymbol{\lambda}})$$

3. The new gradient matrix is:

$$\tilde{\boldsymbol{G}} = \tilde{\sigma}\boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\intercal}\boldsymbol{G}$$

The magnitude dominance resolved □

Algorithm 1 EGA Optimization Strategy for MTL

Input: Loss values for n tasks $[\mathcal{L}_1, \cdots, \mathcal{L}_i], i \in [n]$,
T for softmax and t_{warm} for warmup epoch
INITIALIZATION:
- Unit eccentric vector $\boldsymbol{v}_{ecc} = [1, \cdots, 1]^{\intercal} \in \mathbb{R}^n$
FOR EACH ITERATION:
- Get the current epoch as t
- Get task-specific gradient $\boldsymbol{g}_i = \nabla_\theta \mathcal{L}_i(\theta), i \in [n]$
- Form gradient matrix $\boldsymbol{G} = \{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_i\} \in \mathbb{R}^{n \times m}$
- Eigen decomposition of Gram matrix as in (13): $\boldsymbol{G}\boldsymbol{G}^{\intercal} = \boldsymbol{U}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\intercal}\right)\boldsymbol{U}^{\intercal}$ with eigenvalues $\boldsymbol{\lambda}$
- Get scaling factor for gradient normalization: $\tilde{\sigma} = \min(\sqrt{\boldsymbol{\lambda}})$
- Calculated the orthogonal and normalized gradient matrix as in (15): $\tilde{\boldsymbol{G}} = \tilde{\sigma}\boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\intercal}\boldsymbol{G}$

if $t = t_{warm}$ **then**
  - Record the loss values for all the tasks $\mathcal{L}_i(t_{warm})$

else if $t > t_{warm}$ **then**
  - Calculate the task difficulty weights as in (17): $k_i(t) = \text{softmax}(lr_i(t-1))$
  - Form eccentric vector $\boldsymbol{v}_{ecc} = [k_1, \cdots, k_i]^{\intercal}$

end if

Output: $\tilde{\boldsymbol{g}}_{ecc} = \tilde{\boldsymbol{G}}^{\intercal}\boldsymbol{v}_{ecc}$ for optimization

[4] D.Senushkin,N.Patakin,A.Kuznetsov,andA.Konushin,"Independent component alignment for multi-task learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Sep. 2023

Eccentric gradient alignment (EGA)

Introduce eccentric vector

1. Run several warmup epoch until the epoch t_{warm}

2. Record the loss values for all the tasks $\mathcal{L}_i(t_{warm})$

3. Calculate learning rate:

$$lr_i(t-1) = \frac{\mathcal{L}_i(t-1)}{\mathcal{L}_i(t_{warm})}$$ (small lr_i for fast learning rate)

4. Calculate task difficulty weight and $\boldsymbol{v}_{ecc}$:

$$k_i(t) = \text{softmax}(lr_i(t-1)) = \frac{n \exp\left(lr_i(t-1)/T\right)}{\sum_{j=1}^{n} \exp\left(lr_j(t-1)/T\right)}$$

$$\boldsymbol{v}_{ecc} = [k_1, \cdots, k_i]^\intercal$$ (small T enlarge discrepancy）

5. Get the final joint gradient:

$$\tilde{\boldsymbol{g}}_{ecc} = \tilde{\boldsymbol{G}}^\intercal \boldsymbol{v}_{ecc}$$

The difficulty imbalance resolved □

Algorithm 1 EGA Optimization Strategy for MTL

1: **Input:** Loss values for n tasks $[\mathcal{L}_1, \cdots, \mathcal{L}_i], i \in [n]$, T for softmax and t_{warm} for warmup epoch
INITIALIZATION:
2: - Unit eccentric vector $\boldsymbol{v}_{ecc} = [1, \cdots, 1]^\intercal \in \mathbb{R}^n$
FOR EACH ITERATION:
3: - Get the current epoch as t
4: - Get task-specific gradient $\boldsymbol{g}_i = \nabla_\theta \mathcal{L}_i(\theta), i \in [n]$
5: - Form gradient matrix $\boldsymbol{G} = \{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_i\} \in \mathbb{R}^{n \times m}$
6: - Eigen decomposition of Gram matrix as in (13): $\boldsymbol{G}\boldsymbol{G}^\intercal = \boldsymbol{U}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\intercal\right)\boldsymbol{U}^\intercal$ with eigenvalues $\boldsymbol{\lambda}$
7: - Get scaling factor for gradient normalization: $\tilde{\sigma} = \min(\sqrt{\boldsymbol{\lambda}})$
8: - Calculated the orthogonal and normalized gradient matrix as in (15): $\tilde{\boldsymbol{G}} = \tilde{\sigma}\boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\intercal\boldsymbol{G}$
9: **if** $t = t_{warm}$ **then**
10: - Record the loss values for all the tasks $\mathcal{L}_i(t_{warm})$
11: **else if** $t > t_{warm}$ **then**
12: - Calculate the task difficulty weights as in (17): $k_i(t) = \text{softmax}(lr_i(t-1))$
13: - Form eccentric vector $\boldsymbol{v}_{ecc} = [k_1, \cdots, k_i]^\intercal$
14: **end if**
15: **Output:** $\tilde{\boldsymbol{g}}_{ecc} = \tilde{\boldsymbol{G}}^\intercal \boldsymbol{v}_{ecc}$ for optimization

Part 3 – Result Evaluations

- **Part 1 – EGA Performance**
- **Part 2 – Long-term ECG Recovery Quality**
- **Part 3 – Noise Robustness**

Result Evaluation

EGA Performance: on radar-based ECG recovery

Remarks:

1. Impact of different T values, as the estimation of the intrinsic task difficulties.
2. Difficulties cannot be obtained in advance, except by doing pre-exp.
3. Gaps between loss balancing and gradient balancing methods.
4. Uneven improvements occur for many methods.
5. Potential improvement of the deep-learning model using 'scale law' or other MTL architecture.

TABLE II
COMPARISON OF DIFFERENT OPTIMIZATION STRATEGIES ON RADARODE-MTL

Methods \ Tasks	ECG Shape Recovery RMSE (mV) ↓	ECG Shape Recovery PCC ↑	Cycle Length Estimation PPI Error (ms) ↓	ECG Anchor Estimation Timing Error (ms) ↓	ECG Anchor Estimation MDR ↓	$\Delta m\% \uparrow$
Single-task baseline	0.106	86.6%	9.6	7.5	6.67%	0%
Loss Balancing Methods						
Equal Weight	0.125	79.7%	**8.0**	9.7	5.51%	-0.73%
UW [43]	**0.066**	88.5%	11.2	**5.5**	6.44%	5.95%
GLS [42]	0.087	87.3%	14.1	6.7	4.32%	-4.65%
DWA [41]	0.133	80.7%	8.3	6.4	5.33%	4.84%
STCH [40]	0.070	88.0%	13.9	**5.5**	**3.28%**	3.90%
Gradient Balancing Methods						
GradDrop [39]	0.126	76.5%	8.3	7.6	5.71%	1.53%
GradVac [37]	0.115	81.0%	9.4	9.6	5.89%	-4.33%
CAGrad [38]	0.107	84.2%	10.2	6.2	3.98%	6.72%
IMTL [36]	0.088	89.4%	9.3	6.0	6.22%	8.90%
MoCo [35]	0.179	61.0%	8.7	6.8	4.27%	-5.72%
Aligned-MTL [34]	0.092	87.9%	10.0	6.9	3.52%	10.26%
EGA ($T = 0.1$)	0.119	79.0%	10.6	6.8	3.34%	2.74%
EGA ($T = 0.5$)	0.082	**89.6%**	9.9	6.3	4.19%	12.10%
EGA ($T = 1.0$)	0.085	87.4%	8.5	7.2	4.31%	**13.95%**
EGA ($T = 1.5$)	0.105	82.9%	8.1	6.3	5.13%	10.95%
EGA ($T = 2.0$)	0.091	86.3%	9.2	7.3	4.01%	10.78%

Bold and underline represent the best and the second best results, respectively.

Result Evaluation

EGA Performance: on NYU-v2 (scene understanding)

Remarks:

1. Competitive performance of EGA
2. Dramatic changes of MoCo and STCH
3. Intrinsic task difficulties is hard estimated in prior.

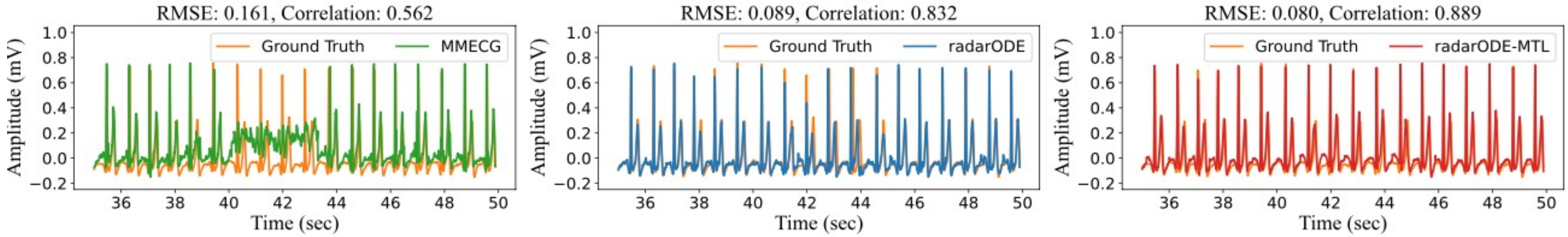
TABLE III
COMPARISON OF DIFFERENT OPTIMIZATION STRATEGIES ON NYUV2

Method	Segmentation ↑		Depth Estimation ↓		Surface Normal Prediction					$\Delta m\%$ ↑
					Angle Distance ↓		Within t° ↑			
	mIoU	Pixel Acc.	Abs. Err.	Rel. Err.	Mean	Median	11.25	22.5	30	
Single-task baseline	52.08	74.11	0.4147	0.1751	23.83	17.36	34.34	60.22	71.47	0.0%
Loss Balancing Methods										
Equal Weight	**53.36**	74.94	0.3953	0.1672	24.35	17.55	34.22	59.64	70.71	1.74%
UW [43]	53.33	**75.43**	0.3878	0.1639	24.03	17.24	34.80	60.33	71.31	2.92%
GLS [42]	53.04	74.68	0.3951	0.1600	24.03	17.30	34.78	60.17	71.28	2.69%
DWA [41]	53.12	75.23	0.3883	0.1615	24.26	17.60	34.25	59.51	70.62	2.55%
STCH [40]	52.87	74.78	0.3915	0.1615	23.27	16.34	**36.61**	62.33	72.98	**4.00%**
Gradient Balancing Methods										
GradDrop [39]	53.26	75.39	**0.3856**	0.1577	24.17	17.49	34.43	59.78	70.95	3.26%
GradVac [37]	52.47	74.88	0.3907	0.1633	24.10	17.23	34.83	60.35	71.33	2.46%
CAGrad [38]	52.19	74.07	0.3976	0.1634	23.83	17.16	34.89	60.65	71.77	2.09%
IMTL [36]	52.34	74.35	0.3897	0.1579	23.76	17.00	35.28	60.92	71.89	3.24%
MoCo [35]	52.78	74.59	0.3858	0.1612	23.34	16.51	36.21	61.90	72.65	3.94%
Aligned-MTL [34]	52.19	74.17	0.3911	0.1605	23.44	16.73	35.45	61.74	72.70	3.23%
EGA ($T = 0.1$)	52.16	74.23	0.3944	0.1651	23.32	16.62	35.87	61.81	72.72	2.84%
EGA ($T = 0.5$)	51.82	73.98	0.3904	0.1614	23.41	16.66	35.87	61.65	72.51	3.11%
EGA ($T = 1.0$)	51.75	74.38	0.3913	0.1609	**23.09**	**16.29**	36.54	**62.51**	**73.22**	3.71%
EGA ($T = 1.5$)	52.37	74.65	0.3950	**0.1571**	23.15	16.46	36.07	62.22	73.07	3.96%
EGA ($T = 2.0$)	52.18	74.23	0.3922	0.1605	23.28	16.61	35.77	61.95	72.86	3.40%

Bold and underline represent the best and the second best results, respectively.

Result Evaluation

Long-term Recovered ECG



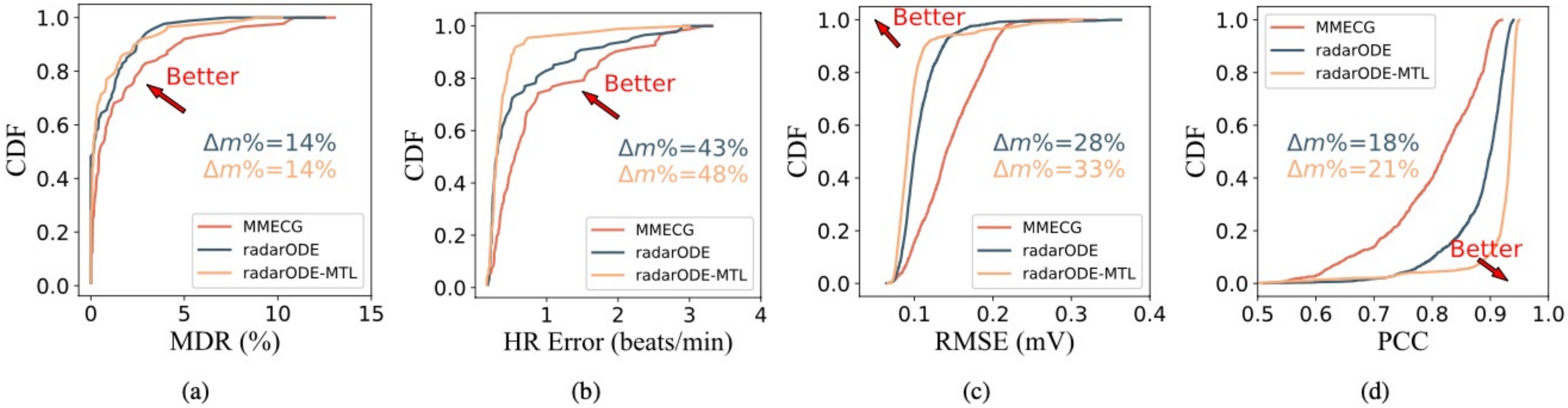
(a) (b) (c)

Remarks:

1. MMECG can do nothing during noise distortion
2. radarODE has the misalignment issue.
3. radarODE-MTL fix the issue and preserve the morphological accuracy.

Result Evaluation

Long-term Recovered ECG

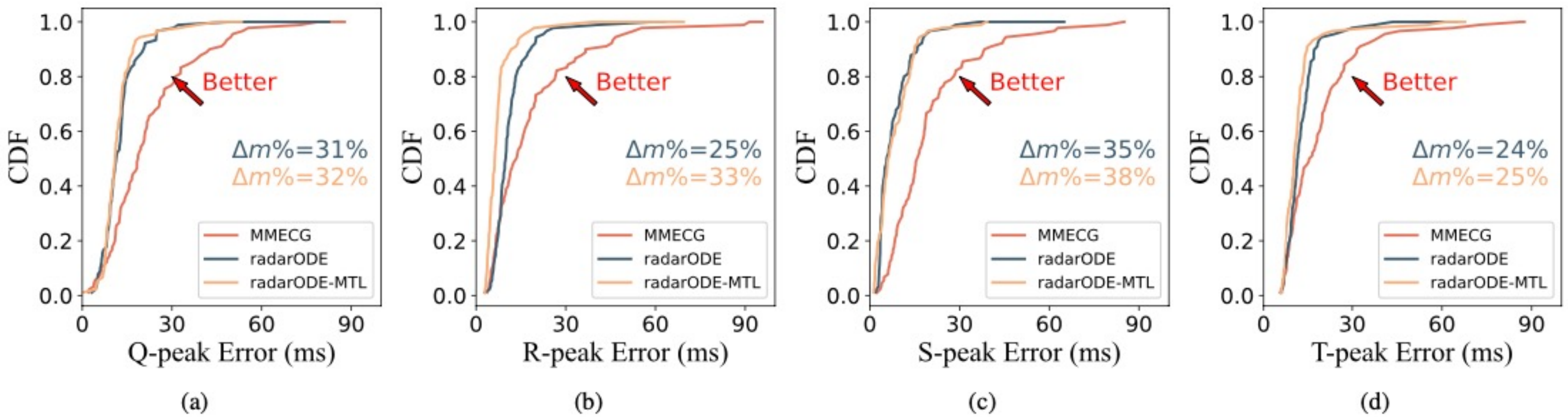


Remarks:

1. Similar performances in MDR.
2. Good performance in HR error for all the three methods (with median error as 0.6, 0.3, 0.3bpm)
3. More improvement on RMSE than PCC

Result Evaluation

Long-term Recovered ECG: Peak Accuracy



Remarks:

1. Similar performances for QST peaks (different from expectation).
2. Large improvement on R peak accuracy, due to the anchor prediction.

Result Evaluation

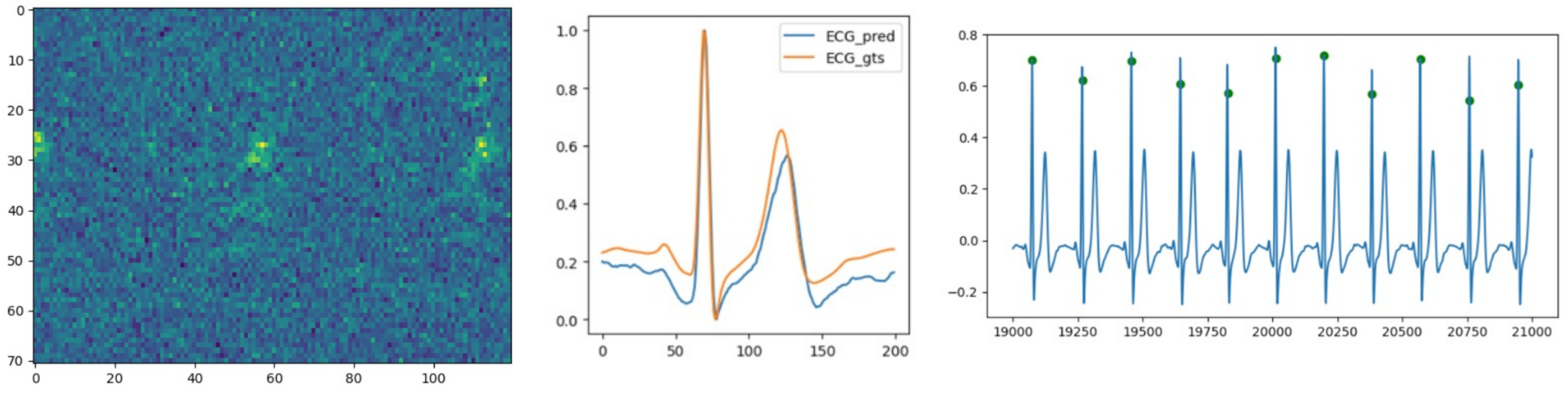
Noise Robustness: Constant Noise

TABLE III
COMPARISON OF THE FRAMEWORKS UNDER DIFFERENT SNR

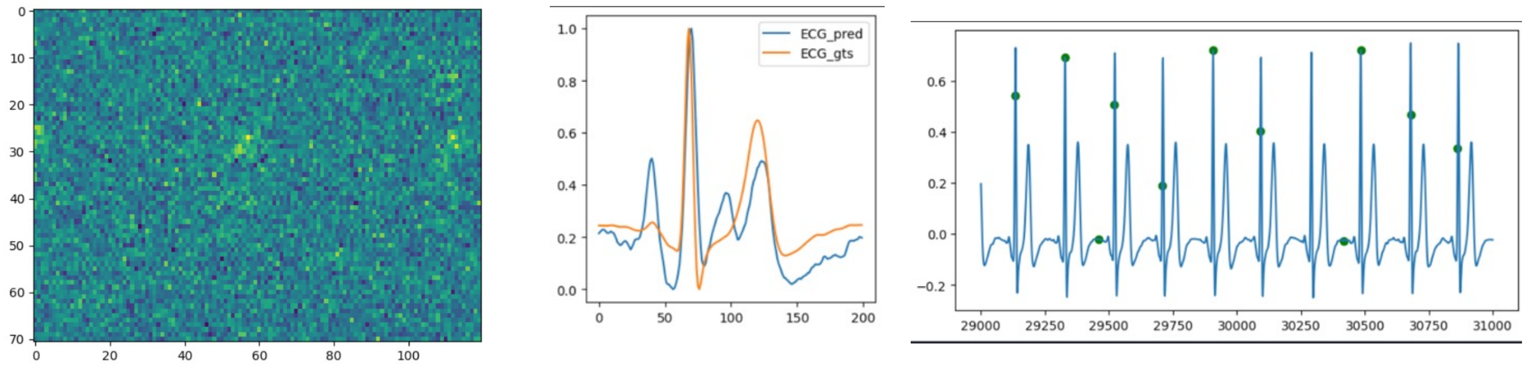
SNR	RMSE (mV) ↓	PCC ↑	Peak Error (ms) ↓	MDR ↓	$\Delta m\%$[1] ↑
		MMECG [30]			
Baseline	0.107	83.75%	9.45	4.52%	0.0%
6 dB	0.107	82.60%	9.76	4.37%	-0.44%
3 dB	0.107	82.64%	9.85	4.84%	-3.20%
0 dB	0.109	80.00%	11.80	4.92%	-10.14%
−1 dB	0.114	78.55%	12.20	5.32%	-15.02%
−2 dB	0.120	74.32%	14.64	5.59%	-25.42%
−3 dB	0.127	62.45%	21.28	6.40%	-52.82%
		radarODE			
Baseline	0.091	83.53%	9.08	4.03%	0.0%
6 dB	0.093	83.30%	9.12	4.36%	-2.83%
3 dB	0.095	83.01%	9.01	4.70%	-5.23%
0 dB	0.101	82.21%	9.89	5.86%	-16.86%
−1 dB	0.116	79.66%	11.90	5.36%	-24.17%
−2 dB	0.157	70.87%	13.95	6.19%	-48.74%
−3 dB	-	-	-	-	Failed
		radarODE-MTL			
Baseline	0.089	85.03%	8.22	4.08%	0.0%
6 dB	0.088	85.31%	8.18	4.20%	-0.31%
3 dB	0.089	84.29%	8.31	4.27%	-1.87%
0 dB	0.091	83.77%	8.03	4.76%	-4.58%
−1 dB	0.093	84.01%	8.10	5.10%	-7.33%
−2 dB	0.093	84.51%	8.02	5.45%	-9.18%
−3 dB	0.094	84.96%	8.19	6.02%	-13.3%

1. $\Delta m\%$ is calculated for each framework based on the baseline.
2. The ECG recovery fails if PCC$<$ 60%, according to the empirical observation of the morphological ECG features.

-3 dB



-9 dB



Result Evaluation

Noise Robustness: Abrupt Noise

TABLE V
COMPARISON OF THE FRAMEWORKS UNDER ABRUPT NOISES

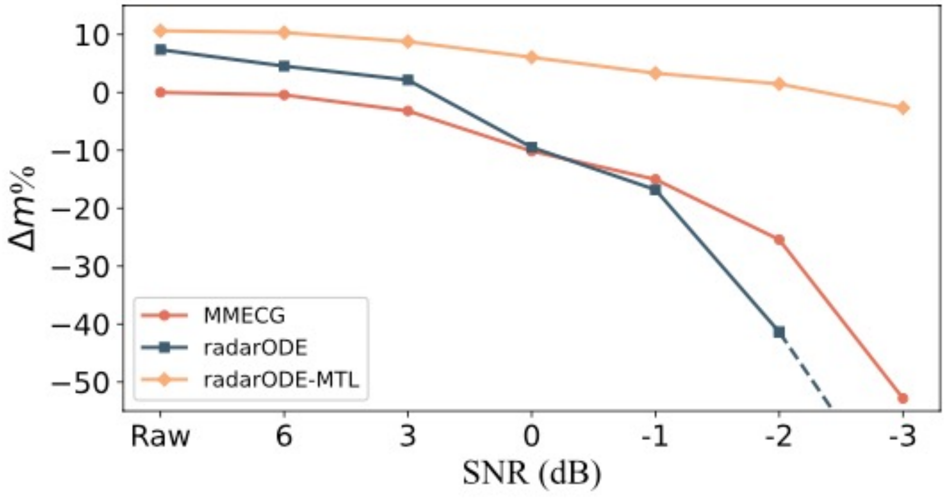
Duration	RMSE (mV) ↓	PCC ↑	Peak Error (ms) ↓	MDR ↓	$\Delta m\%^{1}$ ↑	RMSE (mV) ↓	PCC ↑	Peak Error (ms) ↓	MDR ↓	$\Delta m\%$ ↑
MMECG [5]:	Mild Body Movement (0 dB)					Extensive Body Movement (−9 dB)				
Baseline	0.107	83.75%	9.45	4.52%	0.0%	0.107	83.75%	9.45	4.52%	0.0%
1 sec	0.107	85.53%	10.84	4.82%	-4.94%	0.107	84.05%	10.93	4.82%	-5.62%
2 sec	0.110	82.64%	11.31	5.02%	-8.76%	0.108	79.01%	12.31	5.23%	-13.10%
3 sec	0.114	76.87%	15.56	5.92%	-27.71%	0.116	75.09%	12.50	9.56%	-40.66%
radarODE [10]:	Mild Body Movement (0 dB)					Extensive Body Movement (−9 dB)				
Baseline	0.091	83.53%	9.08	4.03%	0.0%	0.091	83.53%	9.08	4.03%	0.0%
1 sec	0.091	83.49%	9.12	4.36%	-2.31%	0.095	82.96%	9.15	4.33%	-3.19%
2 sec	0.092	83.39%	9.82	4.64%	-6.23%	0.098	82.16%	9.31	4.97%	-8.86%
3 sec	0.095	83.01%	10.01	5.7%	-14.19%	0.102	81.87%	9.66	7.39%	-25.46%
radarODE-MTL:	Mild Body Movement (0 dB)					Extensive Body Movement (−9 dB)				
Baseline	0.089	85.03%	8.22	4.08%	0.0%	0.089	85.03%	8.22	4.08%	0.0%
1 sec	0.090	84.62%	7.87	4.42%	-1.52%	0.090	84.31%	8.28	4.08%	-0.82%
2 sec	0.090	84.78%	8.29	4.44%	-2.56%	0.091	84.21%	8.32	4.41%	-3.15%
3 sec	0.091	84.44%	8.34	5.12%	-7.39%	0.095	84.17%	8.43	5.10%	-8.74%

1. $\Delta m\%$ is calculated for each framework based on the corresponding baseline.

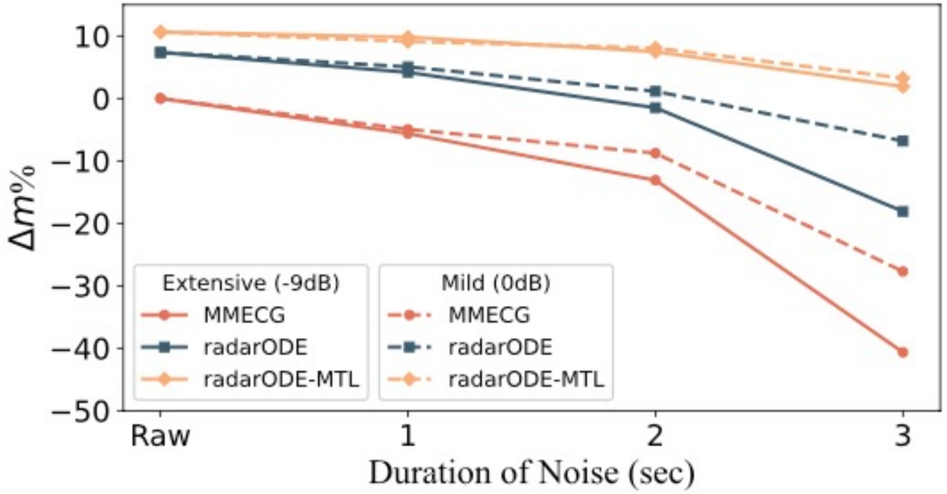
Result Evaluation

Remarks:

1. The ODE model embedded in the ODE decoder could generate uncorrupted ECG pieces for single cardiac cycles under noises and resists certain noise distortion as shown by the acceptable performance of radarODE under mild noise conditions.
2. It is necessary to consider the noise-robustness when designing the deep-learning model, because both MMECG and radarODE reveal a severe degradation in the performance, especially for the low SNR scenarios.
3. The deconstruction of the ECG recovery task in radarODE-MTL could effectively resist the noises, because the morphological feature is protected by the ODE decoder, and the peak accuracy can be compensated from the adjacent cardiac cycles with less noise distortion.



(a)



(b)

Q&A

Thanks for your time!