4-2024

Improving Adaptation of Deep Learning with Inductive Bias

Xiajun Jiang
xj7056@rit.edu

Recommended Citation

Jiang, Xiajun, "Improving Adaptation of Deep Learning with Inductive Bias" (2024). Thesis. Rochester Institute of Technology. Accessed from

Improving Adaptation of Deep Learning with Inductive Bias

by

Xiajun Jiang

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in Computing and Information Sciences

B. Thomas Golisano College of Computing and
Information Sciences

Rochester Institute of Technology
Rochester, New York
April, 2024

Improving Adaptation of Deep Learning with Inductive Bias

by

Xiajun Jiang

Committee Approval:

We, the undersigned committee members, certify that we have advised and/or supervised the candidate on the work described in this dissertation. We further certify that we have reviewed the dissertation manuscript and approve it in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Computing and Information Sciences.

Dr. Linwei Wang — Date
Dissertation Advisor

Dr. Qi Yu — Date
Dissertation Committee Member

Dr. Nathan Cahill — Date
Dissertation Committee Member

Dr. Rui Li — Date
Dissertation Committee Member

Dr. Dan Phillips — Date
Dissertation Defense Chairperson

Certified by:

Dr. Pengcheng Shi — Date
Ph.D. Program Director, Computing and Information Sciences

Improving Adaptation of Deep Learning with Inductive Bias

by

Xiajun Jiang

Submitted to the
B. Thomas Golisano College of Computing and Information Sciences Ph.D. Program in
Computing and Information Sciences
in partial fulfillment of the requirements for the
Doctor of Philosophy Degree
at the Rochester Institute of Technology

Abstract

Human intelligence is strong at adapting to a small number of observations, partially because of the human ability to 1) use given knowledge and 2) distill knowledge from related but different data to guide learning for future tasks, where such ability is the inductive bias during learning.

Deep learning shows a promising solution to artificial intelligence. However, generalizing or adapting deep learning models to heterogeneous tasks remains an open question. Existing *data-driven* models often ignore prior knowledge about the underlying problems of interest, or have limitations in incorporating complex knowledge into neural networks. The *one-size-fit-all* formula assumes the training and testing data follow the same distribution, while the heterogeneity within the training data and the distribution shift from training time to test time lead to generalization error.

In this dissertation, we approached these challenges from the perspective of improving the adaptation with inductive bias, primarily examining the following three research questions: 1) how to learn to adapt with unknown knowledge that can be learned from data, 2) how to adapt deep learning models with known prior knowledge, and 3) how to learn to identify hybrid knowledge with both known prior and unknown errors.

To answer the first research question, we proposed a novel concept of learning to adapt to diverse dynamic environments in high-dimensional long-term time series forecasting. To answer the second research question, we first designed neural functions to model the spatiotemporal physics relationships defined on geometrical domains. We then proposed to improve the learning of neural networks given partially known physics with a hybrid state-space framework. For the last research question, we proposed a hybrid gray-box modeling combining the strength of learning to identify unknown errors from data and adapting with known physics.

In this dissertation, we proposed several novel adaptation methods with good adaptation ability by drawing ideas from different well-studied areas such as variational inference (*e.g.* variational Bayes), image reconstruction (*e.g.* electrocardiographic imaging), time-series forecasting (*e.g.* sequential latent variable models), and few-shot learning (*e.g.* feedforward meta-learning). We evaluated our algorithms on synthetic data and real data in both general and clinical settings, and show that our approach yields significant improvement over existing methods. This, furthermore, opens the door for many new directions of research related to adaptation.

Acknowledgments

The fear of the LORD is the beginning of wisdom, and the knowledge of the Holy One is understanding. – Proverbs 9:10 NASB

I first give my thanks to God for guiding me throughout the journey of my Ph.D. study. God is the creator of the universe and the nature philosophy. His words have enlightened my way of seeking the truth. May God's light shine before men so that the world may see the good works His hand has made. May the Holy One be glorified!

I want to give thanks to my family, especially my beloved wife Ling. Our marriage started in the middle of my Ph.D. journey, and I have a strong belief that God made her to be my better half. I can always find happiness in her even if it is a tiny achievement in my life, and find support in her when troubles come and my soul is weary. When I reflect on our memories, my life has been enriched more than I can imagine since God joined her to me, as her patience, hope, and endurance have already changed me from a boy to a man. God's love never fails, and so as hers. My little girl, Jolynn, who is to come into this world soon, you are a surprise to us when I don't know where to go. We believe that God sent you to us at the right time, and we can't wait to see you face-to-face. May you grow in God's grace, be full of graciousness, and be a bright and beautiful lady.

I would like to express my gratitude to my advisor Professor Linwei Wang. I can't find any better words to describe how she changed my life. She made me realize how a true scientist should be, as she always humbles herself no matter how much knowledge she has. Her mentoring, both professional and personal, has taught me to be a better problem solver and strengthened my faith to continue exploring knowledge in the future. It was my pleasure to study and work under her advisement, for there is always freedom to explore new topics of my interest that ultimately shaped my dissertation and my future research directions. I hope to continue the collaboration with her, for I am grateful to have such a wonderful advisor along with my academic journey.

I want to thank Professor Shi for his support and guidance throughout my Ph.D. study, especially his insightful suggestions for my career plans and his humorous way of sharing his experiences with the young generation. I want to thank my committee members, Professor Rui Li, Professor Qi Yu, and Professor Nathan Cahill for their valuable feedback and discussions on my dissertation. I am especially grateful for Professor Rui Li's support from the very beginning of my Ph.D. study. His unique way of linking mathematical language to real-world concepts benefits me in solving complex problems. I also want to thank Min-Hong, Siyuan, and Charles for their support to Ph.D. students to create a warm and friendly environment where everyone feels at home.

I would also like to acknowledge CBL. I always learned a lot from my lab mates. Sandesh, the one

with solid mathematical foundations, contributed so much to my work of geometrical modeling. Prashnna, one of my most diligent lab mates, helped me with the foundation of variational autoencoder and supported my career direction. Jwala, proficient in mathematical modeling and coding, taught me a lot about implementation skills. Zhiyuan, the best man at my wedding, always told me to be patient and never give up when pursuing research goals. Ryan, one of the best programmers in the lab, collaborated with me on so many projects (I will always remember the paper deadlines that we worked on together). I was so happy to have all of you, Maryam, Nilesh, Pradeep, Omar, Bipin, Ruby, Sumeet, Will, and others who have been supportive and helped me in numerous ways. Although the pandemic separated us physically during most of my Ph.D. study, this connection among us became even more valuable.

I would give thanks to my parents, for their raising me with great love and for unconditional support in my pursuit of academic life. I also give thanks to God for providing me with another family in Christ, especially Mitchell & Anna, Fei & Eliana, Jack, Gus, Max, Pastor Yang, and Pastor Herring, for their great support in my spiritual and worldly life. Many brothers and sisters also witnessed my growth throughout these years. I am so grateful to have all of you in my life.

Praise the Lord for all His grace upon my Ph.D. journey, for the Lord is good, his lovingkindness is everlasting, and His faithfulness to all generations.

Author Publications

† indicates that a modified version of this publication is included in this dissertation.

† **Xiajun Jiang**, Ryan Missel, Maryam Toloubidokhti, Karli Gillette, Anton J. Prassl, Gernot Plank, B. Milan Horacek, John L. Sapp, Linwei Wang. Hybrid Neural State-Space Modeling for Supervised and Unsupervised Electrocardiographic Imaging. Accepted at IEEE Transactions on Medical Imaging (IEEE TMI). 2024.

† ***Xiajun Jiang**, *Ryan Missel, Zhiyuan Li, Linwei Wang. Sequential Latent Variable Models for Few-Shot High-Dimensional Time-Series Forecasting. In Eleventh International Conference on Learning Representations (ICLR), 2023. (* equal contribution)

† **Xiajun Jiang**, Maryam Toloubidokhti, Jake Bergquist, Brian Zenger, Wilson W Good, Rob S MacLeod, Linwei Wang. Improving Generalization by Learning Geometry-Dependent and Physics-Based Reconstruction of Image Sequences. In IEEE Transactions on Medical Imaging (IEEE TMI). IEEE, vol. 42, no. 2, pp. 403-415, Feb. 2023.

- **Xiajun Jiang**, Zhiyuan Li, Ryan Missel, Md Shakil Zaman, Brian Zenger, Wilson W Good, Rob S MacLeod, John L Sapp, Linwei Wang. Few-shot Generation of Personalized Neural Surrogates for Cardiac Simulation via Bayesian Meta-Learning. In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, pp. 46-56, 2022.

- **Xiajun Jiang**, Jess Tate, Jake Bergquist, Akil Narayan, Rob MacLeod, Linwei Wang. Uncertainty Quantification of Cardiac Position on Deep Graph Network ECG. In 2022 Computing in Cardiology (CinC). IEEE, vol. 498, pp. 1-4, Sep. 2022.

- ***Xiajun Jiang**, *Ryan Missel, Maryam Toloubidokhti, Zhiyuan Li, Omar Gharbia, John L Sapp, Linwei Wang. Label-Free Physics-Informed Image Sequence Reconstruction with Disentangled Spatial-Temporal Modeling. In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, pp. 361-371, 2021. (* equal contribution)

- **Xiajun Jiang**, Sandesh Ghimire, Jwala Dhamala, Zhiyuan Li, Prashnna Kumar Gyawali, Linwei Wang. Learning Geometry-Dependent and Physics-Based Inverse Image Reconstruction. In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, pp. 487-496, 2020.

- Maryam Toloubidokhti, Yubo Ye, Ryan Missel, **Xiajun Jiang**, Nilesh Kumar, Ruby Shrestha, Linwei Wang. DATS: Difficulty-Aware Task Sampler for Meta-Learning Physics-Informed Neural Networks. In Twelfth International Conference on Learning Representations (ICLR), 2024.
- Yubo Ye, Huafeng Liu, **Xiajun Jiang**, Maryam Toloubidokhti, Linwei Wang. A Spatial-Temporally Adaptive PINN Framework for 3D Bi-Ventricular Electrophysiological Simulations and Parameter Inferenc. In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, pp. 163-172, 2023.
- Zhiyuan Li, **Xiajun Jiang**, Ryan Missel, Prashnna Kumar Gyawali, Nilesh Kumar, Linwei Wang. Continual Unsupervised Disentangling of Self-Organizing Representations. In Eleventh International Conference on Learning Representations (ICLR), 2023.
- Maryam Toloubidokhti, Ryan Missel, **Xiajun Jiang**, Niels Otani, Linwei Wang. Neural State-Space Modeling with Latent Causal-Effect Disentanglement. In International Workshop on Machine Learning in Medical Imaging (MLMI). Springer, pp. 338-347, 2022.
- Prashnna K Gyawali, Jaideep Vitthal Murkute, Maryam Toloubidokhti, **Xiajun Jiang**, B Milan Horacek, John L Sapp, Linwei Wang. Learning to Disentangle Intersubject Anatomical Variations in Electrocardiographic Data. In IEEE Transactions on Biomedical Engineering (IEEE TBME). IEEE, vol. 69, no. 2, pp. 860-870, Feb. 2022.
- Maryam Toloubidokhti, Prashnna K Gyawali, Omar A Gharbia, **Xiajun Jiang**, Jaume Coll Font, Jake A Bergquist, Brian Zenger, Wilson W Good, Dana H Brooks, Rob S MacLeod, Linwei Wang. Deep Adaptive Electrocardiographic Imaging with Generative Forward Model for Error Reduction. In International Conference on Functional Imaging and Modeling of the Heart (FIMH). Springer, pp. 471-481, 2021.

Contents

1 Introduction **1**

1.1 Overview 1

1.2 Contribution 3

2 Background **5**

2.1 Electrocardiographic Imaging (ECGI) 5

2.1.1 Traditional ECGI Approaches 5

2.1.2 Data-Driven ECGI Approaches 6

2.2 Time Series Forecasting 8

2.2.1 Sequential Latent Variable Models (SLVMs) 8

2.2.2 Autoregressive dynamics 9

2.3 Few-Shot Learning 9

2.3.1 General few-shot learning 9

2.3.2 Few-shot time-series forecasting 10

3 What-how Framework **11**

3.1 Introduction 11

3.2 Unifying conceptual framework for learning latent dynamics . . . 13

3.3 Few-shot forecasting via Bayesian meta-learning . . . 14

3.4 Experiments on benchmark image sequences . . . 16

3.5 Experiments on Complex Physics Simulations . . . 21

3.6 Conclusions and discussion . . . 23

4 Adaptation with Physics – Geometry **24**

4.1 Introduction . . . 24

4.2 Methods . . . 26

4.2.1 Encoding-Decoding with ST-GCNNs . . . 27

4.2.2 Learning Latent Geometry-Dependent Physics . . . 30

4.2.3 Deterministic and Stochastic Formulations . . . 30

4.3 Simulation Data Experiments . . . 32

4.3.1 Models, Data, and Training . . . 32

4.3.2 Generalization to Unseen Heart Rotations . . . 33

4.3.3 Generalization to New Geometry . . . 37

4.3.4 Training and Testing on Multiple Geometries . . . 38

4.3.5 Effect of Stochastic IB Formulation . . . 39

4.3.6 Effect of Alternative Model Architectures . . . 39

4.4 Real Data Experiments . . . 42

4.4.1 Experimental Data Description . . . 42

4.4.2 Evaluation & Results . . . 43

4.5 Discussion and Conclusions 45

5 Adaptation with Physics – Forward Imaging Physics 47

5.1 Introduction 47

5.2 Preliminary: Bayesian Filtering 49

5.3 Hybrid SSM for Neural Filtering 51

5.3.1 Decoder: Hybrid SSM 51

5.3.2 Encoder: Neural Bayesian Filtering 52

5.4 Learning the Hybrid SSM 54

5.4.1 Unsupervised Learning of Hybrid SSM 54

5.4.2 Combined Supervised and Unsupervised Learning 54

5.5 Experiments and Results on Unsupervised Learning of Hybrid SSM 56

5.5.1 Models, Baselines, and Evaluation Metrics 56

5.5.2 Simulation-Data Experiments 57

5.5.3 Real-Data Experiments 60

5.6 Experiments and Results on Combined Learning of Hybrid SSM 63

5.6.1 Models, Baselines, and Evaluation Metrics 63

5.6.2 Combining with Subject-Specific Simulation Data 64

5.6.3 Combining with Non-Subject-Specific Simulation data 64

5.7 Conclusions and Discussion 65

6 Identifiable Hybrid Model 68

6.1 Introduction 68

6.2 Problem Formulation 70
6.3 Methodology 71
6.3.1 Hybrid Modeling of Cardiac EP Process 71
6.3.2 Learning to Identify 73
6.4 Experiments and Results 73
6.5 Conclusion 75

7 Conclusion and Future Works 76

7.1 Future Works 76
7.2 Ethical Impact 77

Appendices 92

A Appendix for Chapter 3 93

A.1 Derivation of (3.7) 93
A.2 Gated Recurrent Unit (GRU) Set-Conditioning 94
A.3 Extra Constraint to Meta-Model 94
A.4 Additional Gravity-16 Results 94
A.5 Turbulent Flow Visualizations 95
A.6 Additional Cardiac Figure 97
A.7 Data Details 97

List of Figures

2.1 Sequential latent-variable models for forecasting high-dimensional sequences. A) time-varying state estimation models infer the latent state from the current and past observations. B1) Time-varying state identification models infer the system parameters from the current and past observations. B2) Time-invariant system identification models predict the whole sequence with a global system parameter and an initial state inferred from observations. C) The proposed meta-SLVM framework that learns to predict diverse dynamics with system parameters inferred from few-shot support sets and latent dynamic model conditioned on inferred parameters. 8

3.1 Forecasting on gravity-16 under (blue) full dynamics and (orange) single dynamics training. The three meta-models had better forecasting performance compared to all baselines. DKF/VRNN/KVAE and meta-DKF had strong reconstruction and near-term forecasting ability but incorrect long-term forecasting (marked by red vertical lines). GRU-res/NODE/RGN-res and their fine-tuned versions showed difficulty in describing mixed gravity. 18

3.2 A: Comparison with baselines trained on mixed-physics. B: Forecasting examples. The meta-model showed superior performance to all baselines. 19

3.3 A: t-SNE plot of support-set embedding **c** from stochastic (left) and deterministic (right) meta-models. The framework was able to recognize and separate the three dynamics in well-structured clusters. B: Generated forecasting by sampling the distribution of **c** given the same z_0. The stochastic model allows changes within and across dynamics by sampling on **c**. 21

3.4 Visual examples (A) and performance metrics of meta-GRU versus other baselines trained on the meta-training set for forecasting electrical dynamics on the heart. The meta-model predicts better signal propagation blocks than other baseline models. 22

4.1 Outline of the ST-GCNN inverse imaging network. The size of the feature maps follows the format of #Vertices×#Feature×#Time. 27

4.2 Illustration of the structure of one ST-GCNN layer followed by a spatial pooling/unpooling layer from Fig. 4.1. C_1 and C_2 are the numbers of channels before/after the spatial/regular convolution. T_1 and T_2 are the temporal dimensions before/after the temporal convolution. 27

4.3 Summary of average performance with respect to different percentages of full training data among the three comparison models. 34

4.4 Examples of reconstructed electrical activity trained on 25% of full training data, tested at rotation x = 10°. The MSE value is shown for each model at each timestep. Both of the two baseline models showed high errors on the free wall of the left ventricle during depolarization/repolarization (arrows), while our presented model predicted the most accurate signal propagation pattern and the scar location (circles). 34

4.5 Summary of average performance with respect to geometrical variations among the three comparison models. 36

4.6 Visual examples of reconstructed electrical activity trained on one single geometry and tested on the heart with a rotation at z = 10°. The MSE value is shown for each model at each timestep. Both baseline models showed substantial errors in locating the scar, while our model predicted the most accurate signal propagation pattern and the scar location (circles). 36

4.7 Convergence of reconstruction accuracy on new geometry by the fine-tuned (red) *vs.* retrained model (yellow). 37

4.8 Examples of reconstructed electrical activity by the fine-tuned model at epoch = 100, the retrained model at epoch = 100, and the retrained model at epoch = 300 on A) patient #2 and B) patient #3. The MSE value is shown for each model at each timestep. The fine-tuned model has the most accurate reconstruction of the signal propagation pattern and the scar location (circles). 37

4.9 Performance of our ST-GCNN method *vs.* ECGI. 38

4.10 A) Effect of residual blocks; B) Effect of hyperparameter value of β; C) Comparison of model performance at $\beta = 0.01$. 40

4.11 Performance of GCNN and non-linear fully-connected (FCN) latent inverse mapping. 40

4.12 Examples of the relation between the norm of $\mathbf{z}_b(j) \cdot \hat{\mathbf{h}}(\mathbf{u}(i,j))$ (y-axis) and the distance $\mathbf{u}(i,j)$ between vertex i and j on the bipartite graph (x-axis). 41

4.13 A flexible epicardial sock array encircled the isolated heart, which was perfused from a second support animal through the aorta with blood returned under suction from the right ventricle. The human-torso-shaped tank was filled with electrolytic fluid consistent with human torso conductivity and contained 192 embedded Ag/AgCl electrodes. The recording system sampled cage and torso potentials simultaneously. Bipolar stimulation was initiated from intramural plunge needles. 42

4.14 Performance of comparison models on measured data: A) Convergence of fine-tuned *vs.* retrained model; B) Final accuracy of various models. 44

4.15 Examples of reconstructed electrical activation sequences stimulated from A) LV base and B) LV free wall. The SCC value is shown for each model at each timestep. The fine-tuned model and the retrained model at epoch = 200 had similar performance. 44

5.1 Overview of the presented network: ODE-GCN. The dashed box means the data-driven term in the mixed loss may not be always available. 50

5.2 Overview of the sampled pacing locations (black) for each heart geometry. 55

5.3 Performance of the proposed model with respect to different data availability in training (right side of the vertical dashed line) and the four baseline models (left side of the vertical dashed line). 55

5.4 Visual examples of reconstructed electrical activity among four comparison models. Both epicardial and endocardial surfaces are presented. The arrow indicates the pacing site location. The color bar shows the scaled range of the signal since it is unitless in synthetic data. The U-SSM is trained on 50% of the full set. The MSE value is shown for each model at each timestep. The U-SSM showed better performance in predicting signal propagation and scar localization (circled) compared to the three ECGI baselines. The S-SSM demonstrated an upper bound in both signal propagation prediction and scar localization. 58

5.5 Summary of average performance on *in-silico* data among three comparison models with respect to different temporal models. 59

5.6 Visual examples of reconstructed electrical activations on *in-vivo* data (Unit: mm; ss: subject-specific; nss: non-subject-specific). The arrows are predicted pacing sites and the dots highlight actual pacing sites. The localization error is shown for each model. The U-SSM and both subject-specific and non-subject-specific M-SSM showed generally better accuracy in early activation localization among all comparison models. 61

6.1 Summary of signal reconstruction performance on $\mathcal{M}_{\text{Hybrid}}$, $\mathcal{M}_{\text{NN}}$, and $\mathcal{M}_{\text{PHY}}$. . . . 74

6.2 Visual examples of the action potential obtained from $\mathcal{M}_{\text{Hybrid}}$ versus $\mathcal{M}_{\text{NN}}$ and $\mathcal{M}_{\text{PHY}}$. $\mathcal{M}_{\text{Hybrid}}$ had the best details compared to the baselines. $\mathcal{M}_{\text{NN}}$ only had the information of depolarization and repolarization, while $\mathcal{M}_{\text{PHY}}$ only showed the signal propagation blocks. 75

A.1 Comparison with models with and without constraint. 95

A.2 Examples of models with and without constraint. 95

A.3 Example of forecasting *known* turbulent flow dynamics. 95

A.4 Example of forecasting *unknown* turbulent flow dynamics. 95

A.5 Example of forecasting *unknown* electrical dynamics on the heart. 97

List of Tables

3.1 Comparison of the presented meta-models with all baselines trained on the meta-training set for gravity-16 data. The improvement of meta-GRU-res (best-performing) over its closest baseline is statistically significant in all metrics ($p < 0.01$, paired t-test). 17

3.2 Comparison with baselines trained on single dynamics in meta-training data on gravity-16. 20

3.3 Performance metrics of meta-GRU-res models with fixed vs. variable k values 20

4.1 Comparison of Architectures 40

5.1 Summary of localization errors for the origin of ventricular activation in each subject.
(Unit: mm. Paired p-test is performed between ECGI Tikhonov and other comparison models. nss: non-subject-specific) 61

A.1 Comparison of the presented meta and base models on known and unknown dynamics. 96

Chapter 1

Introduction

1.1 Overview

Deep learning has shown great promise in a broad range of domains such as computer vision [48,103], speech recognition [51], and more recently modeling and inference of complex systems [30,76,92,131]. Standard deep learning follows a general framework that the neural function learns to fit the direct mapping between data samples and their corresponding labels. This framework, however, faces two critical challenges. First, a large set of well-labeled data needs to be available to train a model with sufficiently good performance before the model is used for future tasks. Second, the learned model is not able to be directly applied to future tasks that are related but different from the training tasks. These two challenges hinder the standard deep learning in adaptation.

Human intelligence, on the other side, has a strong ability to adapt to a small set of observations. This is partially because humans can use knowledge in learning. We call such ability as *inductive bias*. This ability can be further categorized into two types. The first type is that humans are able to incorporate given knowledge to guide learning, which is called *known inductive bias*. For instance, physicists can predict the state of objects in different dynamic systems given physics knowledge and related factors to each system. The other type is that humans are able to extract knowledge from observations for future tasks, considered as *unknown inductive bias*. For the same example of the dynamic systems, suppose small sets of observations from each system are available, even one with limited physics knowledge can summarize the key factors that cause the difference among these dynamics. The two types of inductive bias enable humans to learn in a data-efficient and adaptive formula, providing a perspective from human intelligence on adaptation.

To improve the adaptation of deep learning, we need to better understand the bottlenecks in standard deep learning. The first bottleneck in standard deep learning is that the models are fixed and specific to training tasks, assuming that the training and testing data follow the same distribution. However, this task-specific formula prevents the model from being directly applied to related but different tasks at test time. It further faces challenges in training a model on tasks where the task-related factors change over time. In both scenarios, the task-specific model is not able to extract the shared knowledge under heterogeneous tasks. The second bottleneck of standard deep learning lies in its black-box nature that it learns a direct mapping over the dataset. However, the prior knowledge about the underlying problems of interest has limited contribution to the deep learning model. There is no specific design to represent prior knowledge in neural networks. The learning objective of neural networks is usually an abstract function that is not guided by prior knowledge.

In this dissertation, our main objective is to approach challenges in deep learning models from the perspective of *improving adaptation with inductive bias.* We examine key research questions including: 1) how to learn to adapt with unknown knowledge that can be learned from data, 2) how to adapt deep learning models with known prior knowledge, and 3) how to learn to identify hybrid knowledge both known prior and unknown errors. In the first aspect, the goal is to learn – instead of a single neural network – a learning rule that is able to summarize the knowledge underlying the data from related tasks about how to adapt a neural network to different data distributions. In the second aspect, the prior knowledge of the data can be fully known or partially known. Therefore, the goal is to impose prior knowledge into neural networks when it is fully known, or leverage the partially known knowledge in a hybrid neural formulation to improve learning. In the third aspect, we consider a more hybrid setting where the underlying knowledge of the data contains both known prior knowledge and unknown errors. Therefore, the goal is to learn the rule of identifying the task by given knowledge and estimating the error from data.

We first resolve the research question of learning to adapt to heterogeneous tasks. Instead of learning a *single* model for all heterogeneous tasks or multiple models for each individual, we proposed to learn a learning rule of summarizing knowledge from related but different tasks about adaptation across different data distributions. We then address the research question about adaptation with known prior knowledge, where we primarily focus on incorporating physics-based inductive bias into neural networks. Depending on whether the inductive bias is perfectly known, we explicitly designed neural functions for spatiotemporal relationships in perfectly known physics, then leveraged imperfect or inexact physics rules to guide learning in neural functions. Finally, to answer the third research question, we proposed a hybrid gray-box modeling combining the foundation of learning to identify knowledge from data in the first research question and the strength of adapting with given knowledge in the second research question.

1.2 Contribution

The goal of this research is to improve the adaptation of deep learning models with inductive bias. To this end, we present four major contributions:

What-how Framework: We proposed to learn the inductive bias of how to adapt in a *what-how framework* for the adaptation of neural functions: 1) *what* is the dynamic environment underlying a small set of observations and 2) *how* to adapt the neural function to the learned environment. We implemented the framework via Bayesian meta-learning with a latent dynamic function conditioned on knowledge derived from few-shot support series and a meta-model learning to extract such dynamic-specific knowledge via feed-forward embedding of the support set. We demonstrated improved performance with various sizes of support sets on diverse forecasting tasks and further proved that the framework is agnostic to the latent dynamic function of choice [58]. We further presented the application of the *what-how* framework on cardiac signal simulation with improved personalization and predictive accuracy [57].

Adaptation with Physics – Geometry: We presented a novel spatiotemporal graph convolutional neural network to reconstruct non-Euclidean image sequences. This approach describes the spatiotemporal variables (unknowns and measurements) over their separate geometrical domains, and learns the inverse mapping in between as a function of the graphical embedding of these two geometrical domains. We first proved the feasibility of the framework [56]. In a series of generalization tasks with increasing difficulty, we demonstrated the improved ability of the network to generalize across geometrical changes underlying the data with reduced training data and fewer geometrical variations needed [61].

Adaptation with Physics – Forward Imaging Physics: We proposed a novel hybrid state-space modeling framework to solve image sequence reconstruction where the dynamic function of the source variable is unknown, and the label of the data is absent or partially available. We leveraged the advantage of structured state-space modeling in data-driven learning, with a partially learnable state-space function to model physics and Bayesian filtering strategies to enable supervision by physics without labels. We proved the feasibility of unsupervised learning utilizing physics [60]. We further presented that the proposed method is able to accumulate knowledge from minimal unlabeled observations and combine the strength of physics-based constraints and data-driven learning with better reconstruction performance on partially labeled data [59].

Identifiable Hybrid Model: We present a novel hybrid modeling framework to describe a personalized cardiac digital twin as a combination of a physics-based known expression augmented by neural network modeling of its unknown gap to reality. We then present a novel meta-learning framework to enable the separate identification of both the physics-based and neural components in the hybrid model. We demonstrate the feasibility and generality of this hybrid modeling framework with a proof-of-concept in synthetic experiments.

Chapter 2

Background

2.1 Electrocardiographic Imaging (ECGI)

Cardiac electrical sources produce time-varying voltage signals on the body surface, following quasi-static electromagnetism [94]. Given a pair of heart and torso geometries represented by their enclosing surfaces, the governing physics can be numerically approximated to relate signals on the heart $\mathbf{X}_t$ to those on the body surface $\mathbf{Y}_t$ [119]:

$$\mathbf{Y}_t = \mathbf{H}(\mathbf{g})\mathbf{X}_t \quad \forall t \in \{1, ..., T\}. \tag{2.1}$$

where $\mathbf{X}_t = [x_t(1), x_t(2), \ldots, x_t(M)]^T$ represents electrical potentials on M vertices of the heart mesh, and $\mathbf{Y}_t = [y_t(1), y_t(2), \ldots, y_t(N)]^T$ the electrical potentials on N vertices of the torso mesh, at time instant t. The forward operator $\mathbf{H}(\mathbf{g})$ defines the physics of their relationship, dependent on the given heart-torso geometry $\mathbf{g}$. Specifically, the signal on each torso vertex can be computed as a linear combination of signals on all heart vertices $y_t(i) = \sum_j x_t(j) \cdot h(g(i,j))$ for $i = 1, 2, \ldots N$ and $j = 1, 2, \ldots M$, where linear coefficients $h(g(i,j))$ are known to be inversely proportional to the relative distance between torso vertex i and heart vertex j [6, 96]. ECGI then addresses the reconstruction of $\mathbf{X}_{1:T}$ from given measurements of $\mathbf{Y}_{1:T}$.

2.1.1 Traditional ECGI Approaches

In classic methods, imaging physics is incorporated as prior knowledge in the forward operator $\mathbf{H}(\mathbf{g})$. Numerical optimization and statistical inference methods are then used to seek the inverse solution that best fits the measurements under $\mathbf{H}(\mathbf{g})$, in combination with certain constraints $\mathcal{R}(\mathbf{X}_t)$ about $\mathbf{X}_t$. Representative constraints include the smoothness of the solution in space and time at different orders of derivatives [106], such as Tikhonov regularization [35, 108], truncated singular value decomposition (SVD) [88], and spatiotemporal regularization [16]. Other constraints consider the sparsity of the cardiac signal in the gradient domain by L1 norm [43] and total variance [124]. Alternatively, model-based regularization encodes *a priori* physiological knowledge about the spatiotemporal dynamics of the solution, including logistic functions [117] and step jump functions [95] to describe the activation of action potential and 3D electrophysiological simulation model [47, 86, 119] or network-based representation [42] of spatiotemporal propagation of action potential for transmural sources throughout the myocardium.

The various existing works discussed above can be conceptually summarized in a general state-space framework, where different types of constraints $\mathcal{R}(\mathbf{X}_t)$ can be translated to designs of different dynamic transition functions. For instance, temporal smoothness of $\mathbf{X}_t$ can be described by $\mathbf{X}_t = \mathbf{X}_{t-1} + \omega_t$ which assumes that $\mathbf{X}_t$ changes minimally over consecutive time frames and ω_t is a noise term. If no temporal constraint is imposed, the dynamic transition function can be simply interpreted as $\mathbf{X}_t = \omega_t$. For instance, the spatial smoothness constraint for $\mathbf{X}_t$ [16, 108] can be interpreted as a Gaussian assumption for ω_t, and the spatial sparsity constraint [43, 124] can be summarized as a Laplacian assumptions for ω_t. Alternatively, the dynamic transition function in model-based regularization can be translated as PDEs [117, 119] or neural networks trained from such PDEs [42]. In general, while simpler functions (*e.g.*, smoothness) have limited capacity to express rich knowledge, complex functions (*e.g.*, PDEs of action potential propagation) are associated with increased structural and parameter assumptions and thus a higher risk of introducing errors when applied to an individual subject.

In traditional Bayesian filtering, after the choice of the dynamic transition function is made, it is kept fixed to provide predictions that constrain the estimation given ECG data. Incorrect assumptions may thus lead to incorrect estimations; furthermore, Bayesian filtering is typically performed independently across different ECG data, with a missed opportunity to pull knowledge across data to refine the dynamic transition function to improve its application to future ECG data.

2.1.2 Data-Driven ECGI Approaches

As in many other areas, deep learning ECGI methods have started to show great promise [41, 61, 64, 89, 99]. These approaches typically learn a direct inverse mapping using data pairs of signals on the body and the heart. For instance, fully connected layers have been used to learn the temporal correlation between body and heart potentials from the sequence of ECG data and reconstruct heart potentials in future timesteps [53,63,79]. Long short-term memory (LSTM) networks [52] have been used to learn the sequence-to-sequence relationship between ECGs and cardiac transmembrane potentials (TMPs) [41].

Unlike classic ECGI methods, these data-driven approaches also have the following challenges. First, few data-driven ECGI approaches considered the fact that the inverse mapping should be specific to the underlying geometry. As a result, the learned inverse mapping has to be restricted to the same geometry on which the training was performed. This largely limits the clinical value of these approaches to be applied across patients. To address this challenge, investigators have performed the learning of the inverse mapping between BSPs and activation maps offline and transferred the results onto patient-specific anatomies to achieve fast personalized predictions online [44]. A similar approach is to learn an inverse mapping that is invariant to geometry by removing geometry-related information from the input ECG data using an information bottleneck [41]. However, this approach requires additional training data that represent variations from different geometries. Alternatively, the geometry can be incorporated by conditioning the reconstructions of electrical potentials on 2D image scans of the heart shape [5]. It is not clear how to extend this approach to consider the most important geometrical factors in ECGI—the relative position between the heart and torso.

Another challenge in these data-driven approaches is the large requirement of pairs of $\mathbf{X}_t$ and $\mathbf{Y}_t$ to train. In practice, however, cardiac electrical activity $\mathbf{X}_t$ is hardly available. Training on simulated data offers a potential alternative, as adopted in many existing data-driven ECGI models [41,61,63]. However, the generalizability of the resulting model to real data remains to be tested.

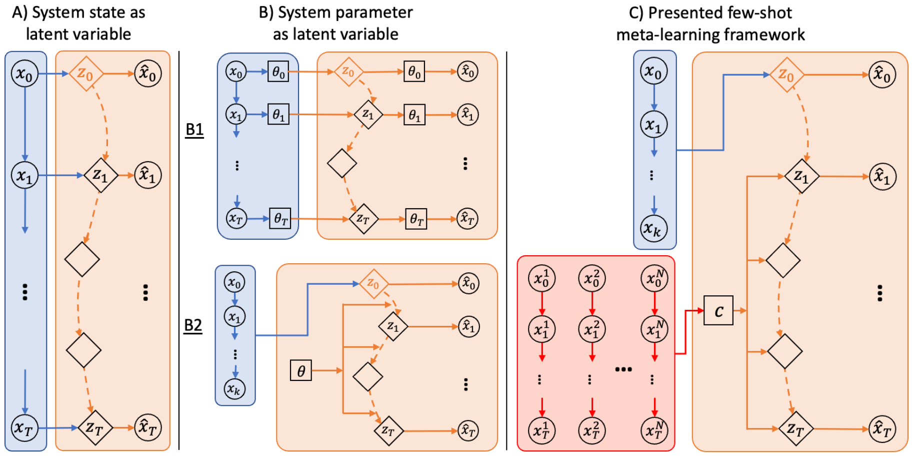


Figure 2.1: Sequential latent-variable models for forecasting high-dimensional sequences. A) time-varying state estimation models infer the latent state from the current and past observations. B1) Time-varying state identification models infer the system parameters from the current and past observations. B2) Time-invariant system identification models predict the whole sequence with a global system parameter and an initial state inferred from observations. C) The proposed meta-SLVM framework that learns to predict diverse dynamics with system parameters inferred from few-shot support sets and latent dynamic model conditioned on inferred parameters.

2.2 Time Series Forecasting

2.2.1 Sequential Latent Variable Models (SLVMs)

Among the first SLVMs is the variational recurrent neural networks (VRNN) [21], followed by a series of deep state-space models (SSMs) [71, 73, 77] focused on modeling the dependence of the posterior and transitional density of the latent state $\mathbf{z}_k$ on past latent states $\mathbf{z}_{<k}$ and observations $\mathbf{x}_{<k}$ (Fig. 2.1A) – resembling the deep extensions of the classic Kalman filter [71] and particle filter [77]. An alternative line of deep SSMs aims to infer the parameters of the latent dynamic function instead [37, 62, 70, 101]. Existing approaches along this line assumed linear latent dynamics, where the linear transition matrix at each time frame k is modeled as a linear combination of a set of global matrices. The linear coefficients are modeled to be time-varying and inferred from observations $\mathbf{x}_{\leq k}$ as illustrated in Fig. 2.1B1. In both formulations, the latent dynamic function's reliance on inferred time-varying variables reduces its ability to forecast without near-term observations.

In parallel, a set of models (Fig. 2.1B2) have been presented that aims to learn a latent dynamic function that forecasts a sequence using only an inferred initial state, in stochastic [107, 127] or deterministic forms [13]. The resulting latent dynamic function is strong at forecasting, albeit only a single function is learned at a time. We build on and advance this formulation of learning to learn a dynamic-specific function from few-shot observations.

Modeling switching dynamics in SLVMs, often based on the formulation in Fig. 2.1A, shares the presented idea of using context variables to control the latent dynamics [10, 74]. They however are concerned with the switching of dynamics within a time series, whereas we are interested in learning to learn dynamics from k-shot support series.

Sequential neural processes (SNPs), based on SLVM formulation in Fig. 2.1A [97, 114], are underlined by Bayesian meta-learning similar to the presented work. They are originally designed for supervised learning of a regression function over time instead of forecasting. In this work, we will extend SNP to realize a meta-version of the SLVM formulation in Fig. 2.1A, as a counterpart to be compared with the presented meta-SLVM in Fig. 2.1C.

2.2.2 Autoregressive dynamics

Autoregressive models are also popular for modeling and forecasting dynamics, especially for approximating physics-based simulations [93, 120]. Some recent works have focused on generalizing across dynamics by, for instance, disentangling spatial and temporal modeling [29] or learning dynamic-specific functions in addition to a global dynamic function [128]. A recent autoregressive model considered ”meta-learning” dynamics by using task-embedding to condition the forecaster [121], although this task encoder is trained separately from the forecasting model via weak supervision and it infers the task from the observed frames of a forecasting series. Moreover, autoregressive models cannot support the controlled generation of time series.

2.3 Few-Shot Learning

2.3.1 General few-shot learning

Few-shot learning has seen substantial progress with static data, including weight initialization [36, 129], model optimizers [102], and feed-forward models to condition [39] or parameterize the primary networks [12, 115]. Among these, feed-forward meta-models replace test-time optimization with simple feed-forward passes using support data. It also has an interesting high-level relation to Exemplar VAE [87] where the few-shot support samples can be viewed as the exemplar. It thus constitutes the basis of the presented few-shot forecasting methods.

2.3.2 Few-shot time-series forecasting

Meta-learning is well studied in univariate time-series forecasting [83] including recent deep-learning advances [90]. Few-shot forecasting for high-dimensional time-series, however, has not been attempted to our knowledge.

Chapter 3

What-how Framework

In this chapter, we focused on the methodology of a *what-how* framework that allows adaptation to heterogeneity across tasks at training and testing time published in ICLR 2023 [58]. Its direct application on personalized neural surrogates for cardiac simulation is detailed in MICCAI 2022 [57].

3.1 Introduction

In many applications, an ultimate goal is to forecast the future states or trajectories of a dynamic system from its high-dimensional observations such as a series of images. Compared to the relatively well-studied *univariate* time-series forecasting [78, 91, 109], *high-dimensional* time-series forecasting raises new challenges: it requires the extraction of the dynamics of an abstract latent state that is not directly observed [13].

Sequential latent variable models (SLVMs) provide an attractive solution that, unlike autoregressive models, abstracts a latent dynamic function $\mathbf{z}_i = f(\mathbf{z}_{<i}; \boldsymbol{\theta}_z)$ with state $\mathbf{z}_i$ and parameter $\boldsymbol{\theta}_z$, along with $\mathbf{z}_i$'s emission to observations $\mathbf{x}_i = g(\mathbf{z}_i)$ [21]. This pair of learned models can support long-term forecasting given only initial frames of observations, as well as the controlled generation of new dynamics. Critical bottlenecks however remain in reaching these goals.

The earlier formulation of SLVMs relies on a natural extension of the static LVMs: as illustrated in Fig. 2.1A, the latent state $\mathbf{z}_i$ is modeled as the latent variable for the generation of $\mathbf{x}_i$, and a sequential encoder is used to facilitate the inference of $\mathbf{z}_i$ from current and past observations $\mathbf{x}_{\leq i}$ [21, 71]. Recent works have argued to instead model and infer the parameter of the latent dynamic function, often modeled as time-varying linear coefficients $\boldsymbol{\theta}_{z,i}$ [37,62,70,101]. This results in an LVM formulation as illustrated in Fig. 2.1B1, where the latent variable $\boldsymbol{\theta}_{z,i}$ is inferred at each i from observations $\mathbf{x}_{\leq i}$. While strong at time-series reconstructions and classifications, a fundamental limitation makes these SLVMs less suited for long-term forecasting: the latent dynamic function has a limited ability to forecast without near-term observations to support the inference of $\mathbf{z}_i$ or $\boldsymbol{\theta}_{z,i}$.

This limitation in the mainstream SLVMs raises a natural question: are we able to relax the assumption of linear dynamic function and directly infer its $\boldsymbol{\theta}_z$? Works adopting this idea have emerged: as illustrated in Fig. 2.1B2, by modeling a single $\boldsymbol{\theta}_z$ – either deterministic [13, 107] or stochastic [127] – $f(\mathbf{z}_{<i}; \boldsymbol{\theta}_z)$ can be asked to predict a time sequence using only an inferred initial state. This formulation has shown strong long-term forecasting, although with its own fundamental limitation: it learns a *single dynamic function* global to all training sequences. This would not only require all training data to share identical latent dynamics, but also has difficulty to forecast test sequences with dynamics different from or unknown to the training.

In this chapter, we answer this important open question of *long-term forecasting for diverse dynamics.* We first present a conceptual framework of SLVMs to unify existing works, and identify an intuitive solution to the underlying critical gap via meta-learning: instead of *learning a single dynamic function*, we can learn to pull knowledge across datasets of different dynamics and *learn to adapt a dynamic function* to few-shot high-dimensional time-series. We then present a Bayesian meta-learning framework as illustrated in Fig. 2.1C: instead of being a single fixed function as in Fig. 2.1B2, we let the latent dynamic function be conditioned on knowledge derived from few-shot support time-series via a feed-forward set-embedding meta-model; given k-shot time-series of a specific dynamics, the model is asked to forecast for query time-series using only the initial frames, meta-learned across dynamics. We develop this framework to be agnostic to the latent dynamic functions of choice, and with the flexibility to forecast with a variable size of k.

We evaluated the presented framework in benchmark image sequences with mixed physics including bouncing balls [37], pendulum [13], and mass-spring [13]. We further applied it to forecasting complex physics of turbulence flow [121] and electrical dynamics over 3D geometrical meshes of the heart. We compared the presented work with SLVMs representative of each of the formulations in Fig. 2.1A-B, along with a recent autoregressive model designed to forecast for diverse dynamics [29]. Each baseline model was trained on 1) the large meta-training set with diverse dynamics, and 2) each dynamics individually, both with and without fine-tuning to k-shot support data. Representative SLVMs were further tested in their feed-forward or optimization-based meta-extensions. Results demonstrated clear margins of improvements by the presented work in forecasting diverse dynamics, with added ability to recognize clusters of distinct dynamics and allow controlled time-series generation given only initial conditions.

3.2 Unifying conceptual framework for learning latent dynamics

We first describe an LVM framework that unifies existing works under two choices of probabilistic graphical models (PGMs). It includes a dynamic function of latent $\mathbf{z}_k$ and its emission to data $\mathbf{x}_k$: $\mathbf{z}_k = f(\mathbf{z}_{<k}; \boldsymbol{\theta}_z), \mathbf{x}_k = g(\mathbf{z}_k)$, where $\boldsymbol{\theta}_z$ represents the parameter of the latent dynamic function.

System states as latent variables: One natural choice of the latent variable is the latent state $\mathbf{z}_k$ underlying the observations $\mathbf{x}_k$. This gives rise to the PGM as illustrated in Fig. 2.1A, where the marginal likelihood of an observed sequence $\mathbf{x}_{0:T}$ can be expressed as:

$$p(\mathbf{x}_{0:T}) = \int_{\mathbf{z}_{0:T}} p(\mathbf{x}_0|\mathbf{z}_0)p(\mathbf{z}_0) \prod\nolimits_{i=1}^{T} p(\mathbf{x}_i|\mathbf{z}_i)p(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{<i})d\mathbf{z}_{0:T}, \tag{3.1}$$

where $p(\mathbf{x}_i|\mathbf{z}_i)$ describes emission and $p(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{<i})$ describes latent dynamics. To facilitate inference, a variational approximation of the posterior density $q(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ is often modeled as $q(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) = \prod_{i=1}^{T} q(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{\leq i})$. The evidence lower bound (ELBO) of (3.1) is:

$$\log p(\mathbf{x}_{0:T}) \geq \sum\nolimits_{i=0}^{T} \mathbb{E}_{q(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{\leq i})} [\log p(\mathbf{x}_i|\mathbf{z}_i)] - \mathrm{KL}(q(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{\leq i})||p(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{<i})). \tag{3.2}$$

Existing works adopting this PGM [21,71,73] differ primarily in how $p(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{<i})$ and $q(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{\leq i})$ are modeled. The first term above encourages reconstruction using the inferred $q(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{\leq i})$ at each time frame i; this weakens the latent dynamic function underlying $p(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{<i})$ that is constrained only by the KL-divergence term. This leads to limited ability to forecast without near-term $\mathbf{x}_{\leq i}$ to support the inference of $q(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x}_{\leq i})$.

System parameters as latent variables: An alternative choice of the latent variable is the parameters themselves of the LVM equation, especially $\boldsymbol{\theta}_z$ of the latent dynamic function. This gives rise to the PGM in Fig. 2.1B, where the marginal likelihood of $\mathbf{x}_{0:T}$ can now be expressed as:

$$p(\mathbf{x}_{0:T}) = \int_{\mathbf{z}_0} p(\mathbf{x}_0|\mathbf{z}_0)p(\mathbf{z}_0)d\mathbf{z}_0 \int_{\boldsymbol{\theta}_z} \prod\nolimits_{i=1}^{T} p(\mathbf{x}_i|\mathbf{z}_i)|_{\mathbf{z}_i=f(\mathbf{z}_{<i};\boldsymbol{\theta}_z)} p(\boldsymbol{\theta}_z)d\boldsymbol{\theta}_z, \tag{3.3}$$

where the observations are explained by an initial latent state $\mathbf{z}_0$ and parameter $\boldsymbol{\theta}_z$ of the latent dynamic function. With a variational approximation of the posterior density $q(\boldsymbol{\theta}_z, \mathbf{z}_0)$ and an assumption of their prior densities $p(\mathbf{z}_0)$ and $p(\boldsymbol{\theta}_z)$, the ELBO of (3.3) becomes:

$$\log p(\mathbf{x}_{0:T}) \geq \mathbb{E}_{q(\boldsymbol{\theta}_z,\mathbf{z}_0)}\left[\log p(\mathbf{x}_{0:T}|\mathbf{z}_0,\boldsymbol{\theta}_z)\right] - \mathrm{KL}(q(\mathbf{z}_0)||p(\mathbf{z}_0)) - \mathrm{KL}(q(\boldsymbol{\theta}_z)||p(\boldsymbol{\theta}_z)), \tag{3.4}$$

This covers different lines of existing works depending on how $q(\boldsymbol{\theta}_z)$ and $p(\boldsymbol{\theta}_z)$ are modeled. In a series of works [37,62,70,101], $\boldsymbol{\theta}_z$ is modeled as time-varying system parameters $\boldsymbol{\theta}_{z,0:T}$. This involves intricate temporal modeling of $q(\boldsymbol{\theta}_{z,i}|\mathbf{x}_{\leq i})$ and $p(\boldsymbol{\theta}_{z,i}|\mathbf{z}_{\leq i})$ over time as illustrated in Fig. 2.1B1. Because the latent dynamic function relies on time-varying $\boldsymbol{\theta}_{z,0:T}$, its forecasting again relies on near-term observations to support the inference of $\boldsymbol{\theta}_{z,i}$. Alternatively, $q(\boldsymbol{\theta}_z)$ can be simply assumed to be global across observations and the dynamic function becomes a Bayesian neural network as presented by [127]. As a more special case, $\boldsymbol{\theta}_z$ can be deterministic which leads to the latent ODE model presented by [107]. If we further assume $\mathbf{z}_0$ to be deterministic, we arrive at the set of deterministic encoding-decoding network with latent dynamic functions examined by [13]. This set of formulations, as summarized in Fig. 2.1B2, shares the advantage of strong long-term forecasting, albeit a fundamental limitation in learning a single dynamic function at a time.

In Section 3.4, we will include representative models from each PGM to provide empirical evidence for the identified limitations. With this basis, we derive an intuitive solution to the identified critical gaps by extending the PGM in Fig. 2.1B2 to the presented PGM in Fig. 2.1C: instead of learning a single dynamic function, we will learn to adapt a latent dynamic function to few-shot support time-series.

3.3 Few-shot forecasting via Bayesian meta-learning

Consider a dataset $\mathcal{D}$ of high-dimensional time-series with M similar but distinct underlying dynamics: $\mathcal{D} = \{\mathcal{D}_j\}_{j=1}^{M}$. For each $\mathcal{D}_j$, we consider disjoint few-shot *support* series $\mathcal{D}_j^s = \{\mathbf{x}_{0:T}^{s,1}, \mathbf{x}_{0:T}^{s,2}, ..., \mathbf{x}_{0:T}^{s,k}\}$ and *query* series $\mathcal{D}_j^q = \{\mathbf{x}_{0:T}^{q,1}, \mathbf{x}_{0:T}^{q,2}, ..., \mathbf{x}_{0:T}^{q,l}\}$ where $k \ll l$. Instead of maximizing the marginal likelihood of $\mathbf{x}_{0:T}$ for all $\mathbf{x}_{0:T} \in \mathcal{D}$ as in (3.3), we formulate a meta-objective to learn to maximize the marginal likelihood of $\mathbf{x}_{0:T}^q$ for all query series $\mathbf{x}_{0:T}^q \in \mathcal{D}_j^q$ when conditioned on support series $\mathcal{D}_j^s$, for all dynamics $j \in \{1, 2, ..., M\}$:

$$p(\mathbf{x}_{0:T}^q|\mathcal{D}_j^s) = \int_{\mathbf{c}} p(\mathbf{x}_{0:T}^q|\mathbf{c})p(\mathbf{c}|\mathcal{D}_j^s)d\mathbf{c}, \quad \mathbf{x}_{0:T}^q \in \mathcal{D}_j^q \tag{3.5}$$

where $p(\mathbf{x}_{0:T}^q|\mathbf{c})$, though similar to (3.3), is now conditioned on (thus adapted to) knowledge derived from support series of a specific dynamics. $p(\mathbf{c}|\mathcal{D}_j^s)$ is the meta-model describing how to extract such dynamic-specific knowledge from few-shot support set $\mathcal{D}_j^s$.

Set-conditioned latent dynamic functions: We model $p(\mathbf{x}^q_{0:T}|\mathbf{c})$ based on (3.3) as:

$$p(\mathbf{x}^q_{0:T}|\mathbf{c}) = \int_{\mathbf{z}_0} p_{\theta_x}(\mathbf{x}_0|\mathbf{z}_0)p(\mathbf{z}_0)d\mathbf{z}_0 \prod_{i=1}^{T} p_{\theta_x}(\mathbf{x}_i|\mathbf{z}_i)|_{\mathbf{z}_i=f(\mathbf{z}_{i-1},\mathbf{c};\boldsymbol{\theta}_z)}, \tag{3.6}$$

where the latent dynamic function is parameterized by $\boldsymbol{\theta}_z$ but conditioned on embedding $\mathbf{c}$ from the support set. To focus on $\mathbf{c}$, we assume $\boldsymbol{\theta}_z$ to be deterministic and global as in [13, 107]. As an example, we can describe $\mathbf{z}_i = \tilde{\mathbf{z}}_{i-1} + \Delta\mathbf{z}_i$ where $\Delta\mathbf{z}_i$ conditions gated recurrent units (GRUs) [20] on $\mathbf{c}$, as detailed in Appendix A.2. This conditioning can be generalized to other functional forms of $f(\cdot)$, which we will demonstrate in experiments.

Meta-model for amortized variational inference: We model $p_\zeta(\mathbf{c}|\mathcal{D}^s_j)$ with a meta-model parameterized by ζ in the form of feed-forward embedding of support set $\mathcal{D}^s_j$. Specifically, each support sequence $\mathbf{x}^s_{0:T} \in \mathcal{D}^s_j$ is first encoded through a neural function $h_\phi(\mathbf{x}^s_{0:T})$ with blocks of interlaced spatial convolution and temporal compression layers. To extract knowledge shared by the set, the embedding from all sequences in $\mathcal{D}^s_j$ is aggregated by an averaging function: $\frac{1}{k}\sum_{\mathbf{x}^s_{0:T}\in\mathcal{D}^s_j} h_\phi(\mathbf{x}^s_{0:T})$, where k is the size of the support set. The value of k can be fixed or variable in our framework. This set embedding parameterizes $p_\zeta(\mathbf{c}|\mathcal{D}^s_j) \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}^2_c)$ via separate linear layers.

To enable inference, we approximate the posterior density $p(\mathbf{c}|\mathcal{D}^s_j, \mathbf{x}^q_{0:T})$ as $q_\zeta(\mathbf{c}|\mathcal{D}^s_j \cup \mathbf{x}^q_{0:T})$, sharing the same meta set-embedding model by augmenting $\mathcal{D}^s_j$ with $\mathbf{x}^q_{0:T}$. The ELBO of (3.5) across all dynamics $\mathcal{D} = \{\mathcal{D}_j\}_{j=1}^M$ can then be derived as:

$$\begin{aligned}\sum_{j=1}^{M} \sum_{\mathbf{x}^q_{0:T}\in\mathcal{D}^q_j} \log p(\mathbf{x}^q_{0:T}|\mathcal{D}^s_j) \quad &\geq \sum_{j=1}^{M} \sum_{\mathbf{x}^q_{0:T}\in\mathcal{D}^q_j} \mathbb{E}_{q_\phi(\mathbf{z}^q_0), q_\zeta(\mathbf{c}|\mathcal{D}^s_j\cup\mathbf{x}^q_{0:T})}[\log p_{\theta_x}(\mathbf{x}^q_{0:T}|\mathbf{z}^q_0, \mathbf{c})] \\ &-\mathrm{KL}(q_\phi(\mathbf{z}^q_0|\mathbf{x}^q_{0:l_{z_0}})||p(\mathbf{z}_0)) - \mathrm{KL}\left(q_\zeta(\mathbf{c}|\mathcal{D}^s_j \cup \mathbf{x}^q_{0:T})||p_\zeta(\mathbf{c}|\mathcal{D}^s_j)\right),\end{aligned} \tag{3.7}$$

where $q_\phi(\mathbf{z}^q_0|\mathbf{x}^q_{0:l_{z_0}}) \sim \mathcal{N}(\boldsymbol{\mu}_{z_0}, \boldsymbol{\sigma}^2_{z_0})$ is parameterized by an encoder with $l_{z_0} = 2$ in all experiments. $p(\mathbf{z}_0)$ is assumed to be $\mathcal{N}(0, \mathbf{I})$. The likelihood term is estimated with reparameterization trick [69], and the KL-divergence terms are calculated analytically.

The optimization of (3.7) is realized via episodic training where, in each training episode, data in each dynamic set $\mathcal{D}_j$ is divided into disjoint support set $\mathcal{D}^s_j$ and query set $\mathcal{D}^q_j$. For each query series across all dynamics, starting with an initial latent state $\mathbf{z}_0$ (inferred from l_{z_0} frames) and k-shot support embedding $\mathbf{c}$, the latent dynamic function is asked to propagate forward to forecast the entire sequence of $\mathbf{z}_{0:T}$ and their corresponding high-dimensional observations $\mathbf{x}_{0:T}$.

3.4 Experiments on benchmark image sequences

Data: We first considered benchmark images generated with controllable physics, including bouncing ball [37], Hamiltonian pendulum [13], and Hamiltonian mass-spring systems [13]. Details of data generation are available in Appendix A.7. To intentionally create data with diverse dynamics, we included 1) a bouncing ball dataset with 16 different directions of gravity, each with 3000 samples simulated using a combination of different initial positions and velocities (*gravity-16*); and 2) a mixed-physics dataset consisting of bouncing balls under 4 gravity directions, and pendulums and mass springs each with four different values of friction coefficients of $0, 0.05, 0.1, 0.15$ (*mixed-physics*). Each physics with a unique parameter includes 3000 samples.

Models: We considered baseline models representative of each formulation outlined in Fig. 2.1. This includes VRNN [21] and DKF [71] representing Fig. 2.1A, DVBF [62] and KVAE [37] representing Fig. 2.1B1, and three models representing Fig. 2.1B2 with latent dynamic functions as residual GRUs (GRU-res), neural ordinary differential equation (NODE), and residual Recurrent Generative Networks (RGN-res) [13]. We also considered a recent autoregressive model designed to tackle forecasting diverse dynamics [29]. All baseline models were 1) trained using the entire meta-training data consisting of mixed dynamics, 2) trained in 1) and further fine-tuned to the meta-test k-shot support set ($k = 15$) (except for [29] as we were uncertain about a proper approach of fine-tuning due to its specialized architecture), and 3) trained individually for each single dynamics, with and without fine-tuning to the meta-test k-shot support set ($k = 15$).

For each of the global latent dynamic models (GRU-res, NODE, and RGN-res), we extended it into our few-shot framework. While few-shot learning with the rest of the SLVMs is not yet reported in literature, we further selected DKF as a representative of the SLVM in Fig. 2.1A and extended it into a feed-forward meta-formulation via a variant of the SNP (meta-DKF). We also attempted optimization-based meta-learning of MAML [36] to the DKF and GRU-res models, although challenges of stability and convergence as noted in literature [2, 80] were encountered, suggesting that MAML extensions to SLVMs may not be trivial due to issues such as vanishing gradient issues over the complex computation graph.

All GRE-res, NODE, and RGN-res based models were trained to forecast for a sequence of 20 frames using only the first 3 frames. We investigated k-shot forecasting when k is fixed at different values of $k = 1, 5, 10, 15$, or allowed to be variable at both meta-training and -test with 15 as the upper limit. For VRNN, DKF, DVBF, and KVAE, we used their public implementations for training and evaluation. Similar network components with the meta-models were scaled to

Table 3.1: Comparison of the presented meta-models with all baselines trained on the meta-training set for gravity-16 data. The improvement of meta-GRU-res (best-performing) over its closest baseline is statistically significant in all metrics ($p < 0.01$, paired t-test).

PGM type	Model	MSE↓	VPT-MSE↑	Dist↓	VPT-Dist↑
Fig. 2.1C	meta-GRU-res	**1.44(0.34)e-2**	**0.68(0.26)**	**2.88(1.45)**	**0.97(0.07)**
	meta-NODE	1.60(0.26)e-2	0.58(0.22)	6.10(2.63)	0.80(0.12)
	meta-RGN-res	1.59(0.24)e-2	0.56(0.21)	6.97(3.08)	0.76(0.13)
Fig. 2.1B2	GRU-res	1.63(0.21)e-2	0.50(0.17)	10.4(3.30)	0.61(0.09)
	GRU-res finetune	1.65(0.24)e-2	0.50(0.18)	9.35(3.33)	0.66(0.12)
	NODE	1.69(0.18)e-2	0.48(0.16)	10.9(3.32)	0.59(0.08)
	NODE finetune	1.70(0.19)e-2	0.48(0.17)	10.4(3.23)	0.61(0.09)
	RGN-res	1.70(0.17)e-2	0.47(0.16)	11.2(3.39)	0.58(0.09)
	RGN-res finetune	1.72(0.19)e-2	0.47(0.17)	10.0(3.36)	0.62(0.11)
Fig. 2.1B1	DVBF	2.32(14.4)e-2	0.02(0.10)	45.3(0.00)	0.00(0.00)
	DVBF finetune	2.33(13.4)e-2	0.02(0.10)	45.3(0.00)	0.00(0.00)
	KVAE	3.37(1.36)e-2	0.24(0.19)	4.81(3.61)	0.57(0.29)
Fig. 2.1A	meta-DKF	3.80(0.59)e-2	0.10(0.11)	7.35(3.26)	0.70(0.25)
	DKF	3.84(0.59)e-2	0.10(0.11)	7.39(3.21)	0.69(0.25)
	DKF finetune	3.85(0.58)e-2	0.10(0.11)	7.51(3.26)	0.69(0.25)
	VRNN	1.78(10.9)e-2	0.24(0.11)	23.1(21.6)	0.51(0.07)
	VRNN finetune	2.15(12.2)e-2	0.21(0.16)	8.31(11.6)	0.75(0.19)
Autoregressive	Donà et al	3.52(0.26)e-2	0.001(0.01)	13.7(3.05)	0.06(0.15)

have comparable parameter scales. Because of their reliance on observed time frames to support prediction, 8 observed frames were exposed to the encoder to reconstruct the 8 frames and forecast the additional 12 frames.

Metrics: We considered four quantitative metrics on meta-test series. We included the commonly used mean squared error (MSE) of forecasted images, and the recently proposed metric of Valid Prediction Time (VPT) that measures how long the predicted object's trajectory remains close to the ground truth trajectory based on the MSE (VPT-MSE) [13]. Because pixel-level MSE does not necessarily well capture the quality of the predicted dynamics due to the small object size on the image, we further introduced two new metrics: distance (Dist) between the ground truth

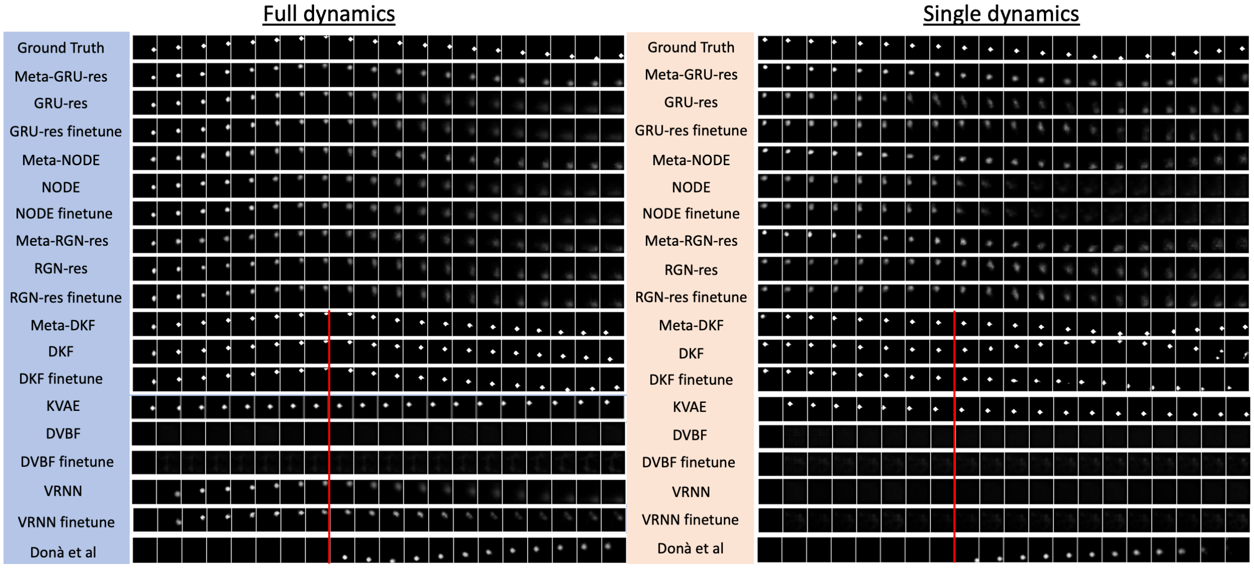
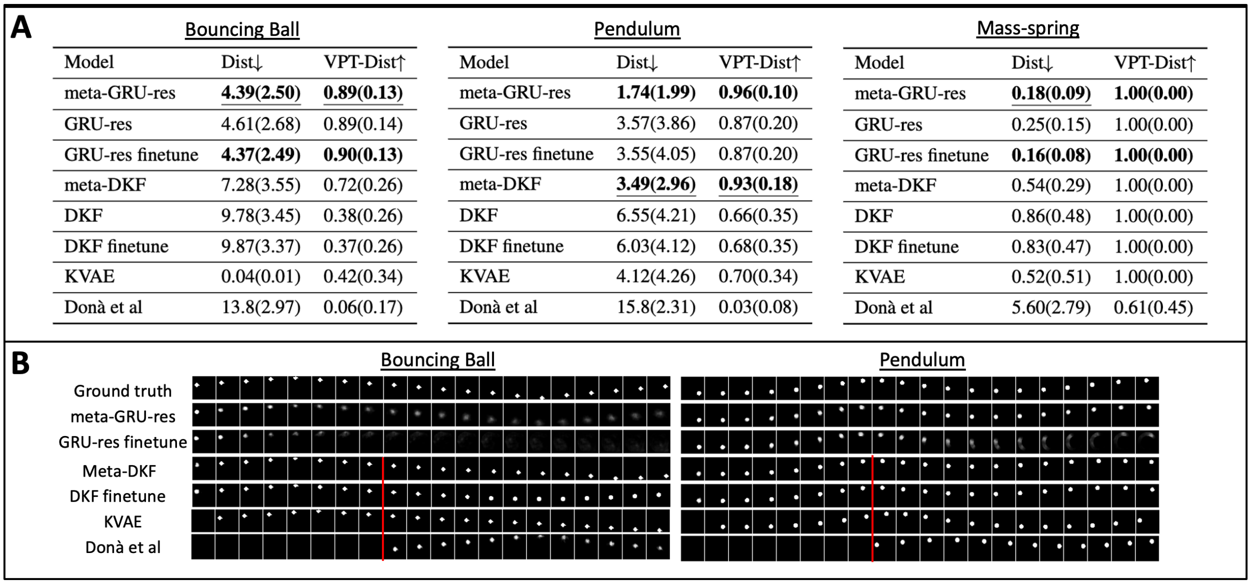


Figure 3.1: Forecasting on gravity-16 under (blue) full dynamics and (orange) single dynamics training. The three meta-models had better forecasting performance compared to all baselines. DKF/VRNN/KVAE and meta-DKF had strong reconstruction and near-term forecasting ability but incorrect long-term forecasting (marked by red vertical lines). GRU-res/NODE/RGN-res and their fine-tuned versions showed difficulty in describing mixed gravity.

and predicted location of the moving object; and VPT determined based on this distance error (VPT-Dist).

Comparison with baseline models trained on full dynamics: For *gravity-16* data, we used 10 gravity in meta-training, 2 in meta-validation, and 4 in meta-testing. Table 3.1 summarizes the quantitative test performance of the three k-shot meta-models obtained with $k = 15$, in comparison to each of the baseline models trained from the full meta-training set. We include complete results across all models in Appendix A.4 with Table A.1. Visual examples for these quantitative results are in Fig. 3.1 (shaded blue): all the baseline models, including their fine-tuned and meta-versions, struggled with limited forecasting ability, especially evidenced by the error in predicting the movement of the ball over time (Dist and VPT-Dist). For DKF/VRNN/KVAE and meta-DKF, there were strong reconstruction and near-term forecasting from partially observed frames (marked by red vertical lines), but incorrect forecasting further away from the observed frames. GRU-res/NODE/RGN-res and their fine-tuned versions exhibited difficulty to describe mixed gravity.

For *mixed-physics* data, for each of the three physics, we included three dynamic settings in meta-



Bouncing Ball

Model	Dist↓	VPT-Dist↑
meta-GRU-res	**4.39(2.50)**	**0.89(0.13)**
GRU-res	4.61(2.68)	0.89(0.14)
GRU-res finetune	**4.37(2.49)**	**0.90(0.13)**
meta-DKF	7.28(3.55)	0.72(0.26)
DKF	9.78(3.45)	0.38(0.26)
DKF finetune	9.87(3.37)	0.37(0.26)
KVAE	0.04(0.01)	0.42(0.34)
Donà et al	13.8(2.97)	0.06(0.17)

Pendulum

Model	Dist↓	VPT-Dist↑
meta-GRU-res	**1.74(1.99)**	**0.96(0.10)**
GRU-res	3.57(3.86)	0.87(0.20)
GRU-res finetune	3.55(4.05)	0.87(0.20)
meta-DKF	**3.49(2.96)**	**0.93(0.18)**
DKF	6.55(4.21)	0.66(0.35)
DKF finetune	6.03(4.12)	0.68(0.35)
KVAE	4.12(4.26)	0.70(0.34)
Donà et al	15.8(2.31)	0.03(0.08)

Mass-spring

Model	Dist↓	VPT-Dist↑
meta-GRU-res	**0.18(0.09)**	**1.00(0.00)**
GRU-res	0.25(0.15)	1.00(0.00)
GRU-res finetune	**0.16(0.08)**	**1.00(0.00)**
meta-DKF	0.54(0.29)	1.00(0.00)
DKF	0.86(0.48)	1.00(0.00)
DKF finetune	0.83(0.47)	1.00(0.00)
KVAE	0.52(0.51)	1.00(0.00)
Donà et al	5.60(2.79)	0.61(0.45)

Figure 3.2: A: Comparison with baselines trained on mixed-physics. B: Forecasting examples. The meta-model showed superior performance to all baselines.

training and left out one in meta-testing. Fig. 3.2A summarized the test results of the presented meta-GRU-res (with variable k) with representative baseline models. Visual examples are shown in Fig. 3.2B. As shown, meta-DKF, DKF, and DVBF again demonstrated limited ability for long-term forecasting across all physics. KVAE, VRNN, and the finetuned global latent GRU-res were more successful with the mass spring and pendulum systems with relatively simpler dynamics, yet they struggled with the gravity system. The presented meta-GRU-res model consistently outperformed all the baselines across all dynamics, with a larger gain in more complex dynamics.

Comparison with baseline models trained on single dynamics: Table 3.2 summarizes the performance of representative baseline models when trained on a single gravity on *gravity-16* data in comparison to meta-GRU. As shown, in both test dynamics known and unknown to the training, the meta-models outperformed the single-dynamic baselines, suggesting the added benefits of learning across dynamics. This margin of improvements remained even when the single-dynamics baselines were fine-tuned to the k-shot support series of unknown test dynamics. Visual examples of these baselines are also shown in Fig. 3.1 (orange shade).

Ablation study: Table 3.3 summarized the effect of k on k-shot forecasting using the meta-GRU-res model. As expected, model performance improved as the size of k increased. Even with $k = 5$, however, the performance was significantly better than all the base models summarized in Table

Table 3.2: Comparison with baselines trained on single dynamics in meta-training data on gravity-16.

Model	Dynamics	MSE↓	VPT-MSE↑	Dist↓	VPT-Dist↑
meta-GRU-res	known	**1.43(0.34)e-2**	**0.68(0.26)**	**2.86(1.44)**	**0.97(0.06)**
	unknown	**1.45(0.33)e-2**	**0.67(0.25)**	**2.96(1.49)**	**0.97(0.07)**
GRU-res	known	1.80(0.29)e-2	0.46(0.23)	6.86(3.95)	0.77(0.19)
	unknown	1.99(0.277)e-2	0.37(0.18)	8.07(3.56)	0.69(0.17)
GRU-res finetune	unknown	2.03(0.27)e-2	0.35(0.17)	8.51(3.42)	0.66(0.18)
meta-DKF	known	3.81(0.59)e-2	0.10(0.11)	7.37(3.27)	0.70(0.25)
	unknown	3.80(0.59)e-2	0.10(0.11)	7.30(3.21)	0.70(0.25)
DKF	known	3.74(0.55)e-2	0.10(0.11)	8.37(3.79)	0.63(0.28)
	unknown	3.79(0.52)e-2	0.09(0.10)	8.72(3.75)	0.60(0.27)
DKF finetune	unknown	3.82(0.52)e-2	0.09(0.10)	8.77(3.77)	0.59(0.27)
KVAE	known	3.42(1.30)e-2	0.39(0.34)	5.05(3.57)	0.5(0.34)
	unknown	3.46(1.36)e-2	0.22(0.19)	5.17(3.91)	0.53(0.28)
Donà et al	known	3.58(0.33)e-2	0.00(0.01)	13.7(3.36)	0.07(0.18)
	unknown	3.56(0.34)e-2	0.00(0.01)	14.1(3.84)	0.08(0.19)

Table 3.3: Performance metrics of meta-GRU-res models with fixed vs. variable k values

K	Mode	MSE↓	VPT-MSE↑	Dist↓	VPT-Dist↑
1	Fixed	1.80(0.21)e-2	0.44(0.16)	10.6(3.40)	0.60(0.10)
5	Fixed	1.53(0.36)e-2	0.61(0.25)	3.49(1.89)	0.94(0.10)
10	Fixed	1.46(0.34)e-2	0.65(0.26)	3.08(1.58)	0.96(0.08)
15	Fixed	**1.44(0.34)e-2**	**0.68(0.26)**	**2.88(1.45)**	**0.97(0.07)**
	Variable	**1.50(0.34)e-2**	**0.64(0.25)**	**3.44(1.80)**	**0.94(0.10)**

3.1. Allowing k to be variable had no noticeable effect on model performance. This flexibility highlights the practicality of the presented framework to forecast with any given size of support series.

Latent embedding and generation of diverse dynamics: Fig. 3.3A shows the distribution of the latent embedding $\mathbf{c}$ obtained from a randomly-selected support set, in comparison to a

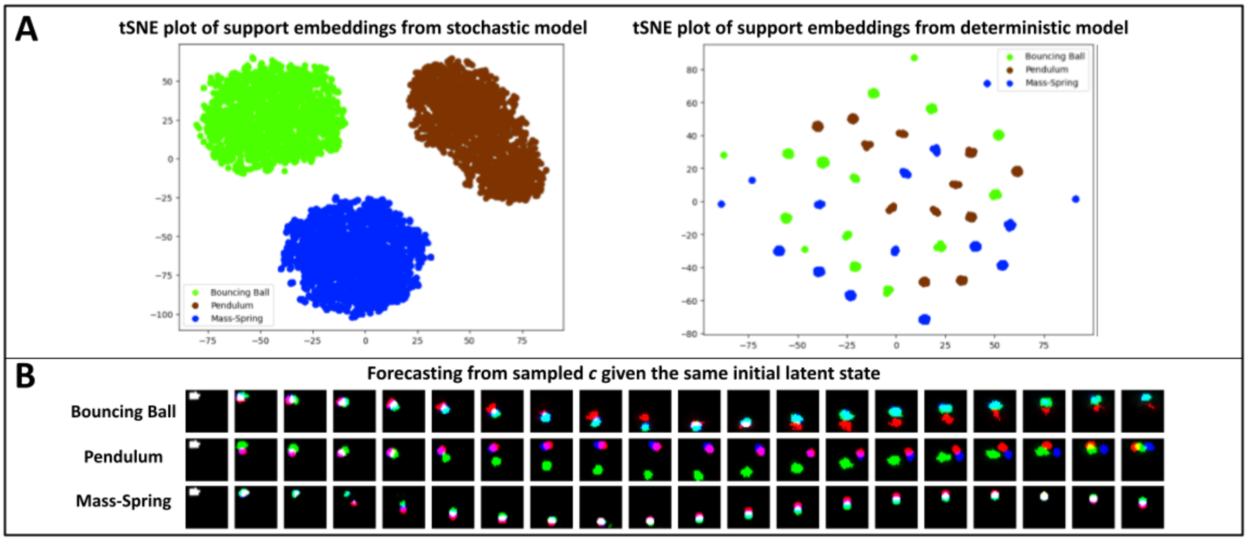


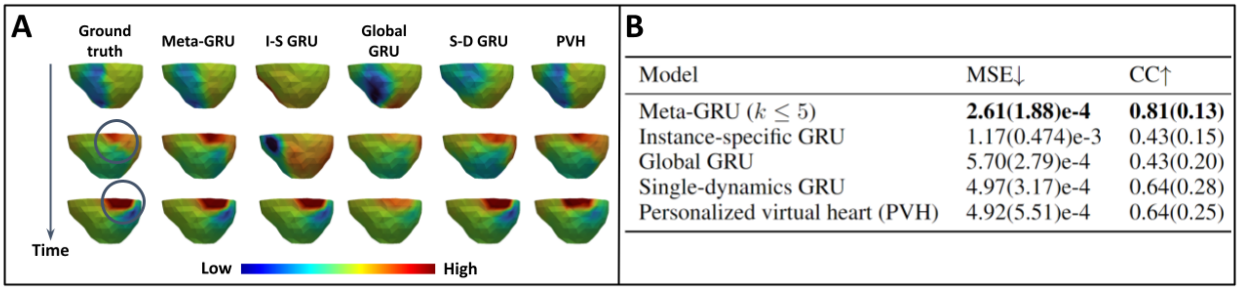
Figure 3.3: A: t-SNE plot of support-set embedding $\mathbf{c}$ from stochastic (left) and deterministic (right) meta-models. The framework was able to recognize and separate the three dynamics in well-structured clusters. B: Generated forecasting by sampling the distribution of $\mathbf{c}$ given the same z_0. The stochastic model allows changes within and across dynamics by sampling on $\mathbf{c}$.

deterministic version of the presented meta-model on *mixed-physics* data. As shown, the presented framework was able to recognize and separate the three dynamics using the k-shot support set: given an initial $\mathbf{z}_0$, it was then able to generate different time-series within the same dynamics as well as across dynamics by sampling the distribution of $\mathbf{c}$ (Fig. 3.3B). This was not possible with its deterministic counterpart.

3.5 Experiments on Complex Physics Simulations

We then considered learning and forecasting two more complex physics-based dynamics: turbulent flow dynamics and cardiac electrical dynamics.

Turbulent flow dynamics: We customized the meta-GRU model to a dataset of turbulent flow dynamics, simulated with 25 varying buoyant forces acting on the fluid. Each dynamic contains 64×64 velocity fields of turbulent flows. We use 20 dynamics in meta-training and meta-validation with 80-20 split, and the rest 5 in meta-testing. We followed the experimental setup in [121] with an observed window (20 frames in theirs vs. 5 in ours) and a prediction roll-out of 20 frames. Despite using a smaller number of observed frames, the presented meta-GRU model obtained a rooted MSE



Model	MSE↓	CC↑
Meta-GRU ($k \leq 5$)	**2.61(1.88)e-4**	**0.81(0.13)**
Instance-specific GRU	1.17(0.474)e-3	0.43(0.15)
Global GRU	5.70(2.79)e-4	0.43(0.20)
Single-dynamics GRU	4.97(3.17)e-4	0.64(0.28)
Personalized virtual heart (PVH)	4.92(5.51)e-4	0.64(0.25)

Figure 3.4: Visual examples (A) and performance metrics of meta-GRU versus other baselines trained on the meta-training set for forecasting electrical dynamics on the heart. The meta-model predicts better signal propagation blocks than other baseline models.

(RMSE) of 0.26 ± 0.05 on seen dynamics and 0.49 ± 0.05 on unseen dynamics, in comparison to respective RMSEs of 0.42 ± 0.01 and 0.51 ± 0.02 reported in [121], all reported on the 20 roll-out frames. We included trajectory visualizations in Appendix A.5.

Cardiac electrical dynamics: The propagation of electrical waves in the heart is governed by reaction-diffusion partial differential equations (PDEs) [1]. While direct PDE-based simulation holds clinical potential (*e.g.*, for virtually testing treatment response), its patient-specific parameters are difficult to estimate and its computational cost is high. Although neural approximations provide a promising computationally-efficient alternative [38], how to personalize such a neural model remains an open challenge where existing models are typically trained for a PDE with given parameter configurations. Here, we apply the presented framework for few-shot learning of a personalized neural model that can be used to efficiently forecast how a patient-specific heart may respond to electrical simulations at different locations.

We simulate electrical propagation originating from various locations in a 3D heart mesh, with 15 settings of PDE parameters representing 15 dynamics with different locations of injury to the heart muscle. We use 9 dynamics in meta-training, 3 in meta-validation, and 3 in meta-testing, with disjoint time series with different initial conditions (meta-training: 450; meta-test: 2,020). Each time series describes 3D+T propagation of electrical wave with blocks at locations of muscle injury specific to each dynamics (see an example if Fig. 3.4A column 1). The quality of the forecast series is measured by its MSE and spatial correlation coefficient (CC) with the actual time series.

We adopted a graph-CNN encoder/decoder and an ODE-GRU latent dynamic function $\mathbf{z}_k = f(\mathbf{z}_{<k}; \boldsymbol{\theta}_z)$ similar to that described in [60]. We trained it with a global $\boldsymbol{\theta}_z$ (global GRU), an individual $\boldsymbol{\theta}_z$ for each PDE parameter (single-dynamics GRU), a conditioned $f(\mathbf{z}_{i-1}, \mathbf{c}; \boldsymbol{\theta}_z)$ with $\mathbf{c}$ encoded from individual training series (instance-specific GRU), and the presented framework

(meta GRU) with k varying between 1 and 5. We further added a strong personalized virtual heart (PVH) baseline using the original PDE simulation, with the PDE parameter optimized by a SOTA approach from k-shot support series as described in [26].

As shown in Fig. 3.4A and additional examples in Appendix A.6, only meta-GRU was able to accurately forecast the propagation block while the other baseline models missed the correct locations of muscle injury specific to a subject (black circles in column 1): note that the single-dynamics GRU performed well on the training dynamics Fig. 3.4A, but fails on unknown dynamics (Appendix A.6). Unable to identify injury to patient-specific heart, the forecasting model will be of little value in clinical tasks such as personalized prediction and treatment planning. This gain in forecasting performance by meta-GRU is quantitatively summarized in Fig. 3.4B across all meta-test time-series. Note that meta-GRU exhibited a notable margin of improvement even versus the PVH: PVH takes on average 5 minutes to forecast each series, versus 0.24 seconds by the meta-GRU; moreover, to optimize PDE parameters of the PVH on average required 100 calls to the PDEs (*i.e.*, $\sim$ 10 hours), versus 0.032 seconds for meta-GRU to adapt to patient-specific dynamics. This substantial gain in efficiency without loss of accuracy holds significant value for clinical applications.

3.6 Conclusions and discussion

In this chapter, we present a sequential LVM framework to unify existing approaches to learning latent dynamics, identify their limitations associated with the underlying choices of PGMs, and provide empirical evidence for the identified limitations. We further identify meta-learning as an intuitive solution to the identified open gaps, present a framework for few-shot high-dimensional time-series forecasting, and demonstrate that its performance gain is agnostic to the underlying choice of latent dynamic functions. **Limitations:** An avenue of future work is to expand the latent dynamic functions in this framework, especially those integrating strong inductive biases based on physics such as Hamiltonian mechanics [13].

Chapter 4

Adaptation with Physics – Geometry

In this chapter, we describe how to explicitly design neural functions for spatiotemporal relationships in physics. We considered electrocardiographic imaging (ECGI) systems and focused on incorporating geometry into neural networks. This work has been published in MICCAI 2020 [56] and IEEE TMI 2023 [61].

4.1 Introduction

Deep learning has shown state-of-the-art performance across a variety of image reconstruction tasks [30,40,76,126,131,133]. In some tasks, the imaging physics is partially known and modulated by specific parameters. For instance, the heart or brain generates potentials that can be measured on the body or skull surface [42,81]. This gives rise to (forward and inverse) mapping operators following the underlying quasi-static electromagnetic theory, but specific to the geometry on which the sources and measurements reside (*e.g.*, the heart and body surface).

In the context of Euclidean deep learning, one would attempt to learn such inverse mapping without the knowledge of the underlying geometry, such as the heart shape and relative position of between heart and torso [5,41,53,55,63,79]. This approach, as we will show, increases the need for training data and produce an inverse mapping not generalizable across geometries. It is possible to tackle the latter issue by using an information bottleneck to remove geometrical information from input data and thus make this inverse mapping invariant to geometrical factors [41]. Such an approach, unfortunately, requires even more training data to represent the variations arising from different geometries.

An interesting open question is thus whether learning such inverse mapping as a function of the underlying geometry would reduce the need of training data and improve the generalization of the learned function. Graph convolutional neural networks (GCNN) provide a promising approach to describe non-Euclidean variables defined over geometrical domains [15]. Significant efforts have been made in GCNN, such as node- and graph-level classifications, graph embedding, and graph generation [123]. However, to our knowledge, no previous works have reported learning inverse mappings between spatiotemporal variables defined on two separate graphs.

In this chapter, we present a novel network to reconstruct non-Euclidean image sequences by directly learning inverse mapping as a function of the underlying geometry of the problem. To describe the spatiotemporal variables (unknowns and measurements) over their respective geometrical domains, we first introduce an encoding-decoding architecture consisting of spatial-temporal graph convolutional neural networks (ST-GCNN) defined separately on each domain. To model the geometry-dependent physics in between, we then learn the inverse mapping as a function defined on a bipartite graph over the graphical embedding of these two geometrical domains with the functional form informed by the underlying physics. We focus on the generalization ability of this non-Euclidean image reconstruction network from two aspects. First, previous studies [41] based on Euclidean deep networks described that a stochastic formulation of the same network, based on the theory of information bottleneck (IB) [116], could improve the generalization ability of the network by removing from the input data geometry information that is not present in the output solutions. By allowing the inverse mapping to change with the underlying geometry in the presented non-Euclidean network, we anticipate that this particular benefit of the IB would be reduced. To test this conjecture, we develop the non-Euclidean network in both deterministic and stochastic formulations, and investigate their reconstruction performance differences. Second, we test the presented network in a series of generalization tasks with increasing difficulty, in comparison to Euclidean baselines with and without a geometry-invariant bottleneck [41], to test its ability to 1) reduce the diversity of the training data needed on the same geometry, 2) reduce the number of geometrical variations needed in the training data, and 3) train across different geometry (*i.e.* different graphs), a process that is not possible with Euclidean learning unless these geometries are pre-registered. Finally, in *in-vivo* animal experiments, we demonstrate the ability of the presented network – after training on simulation data – to be quickly fine-tuned to a small amount of *in-vivo* data. All experiments are performed in the application of reconstructing spatiotemporal electrical potentials on the ventricular surface from body-surface potentials (commonly known as electrocardiographic imaging (ECGI)) [119].

The main contributions of this chapter include the following:

1. We present, to our knowledge, the first ST-GCNN approach to learning an inverse mapping between non-Euclidean variables that is geometry dependent and informed by the underlying imaging physics.
2. We present the first geometric deep learning approach to ECGI [119] that addresses the importance of geometry-specific mapping, which has been widely established in the ECGI literature [100,119], but only rarely considered in emerging machine or deep learning solutions to ECGI.
3. We investigate the generalization ability of our method, both by examining its stochastic formulations based on the theory of IB, and by experimentation in a series of generalization tasks with increasing difficulty. We provide evidence for its ability to generalize without the stochastic formulation, to learn from a small amount of training data, as well as to learn and test across multiple different geometries.
4. We further investigate the ability of our method to be fine-tuned to a small amount of data on a new geometry.
5. We perform in-depth examinations into the effects of different neural network architecture designs, and introduce random edge dropping within the presented framework.

4.2 Methods

To respect the geometry-dependent physics behind the problem, our method learns a geometry-dependent inverse mapping by 1) describing $\mathbf{X}_t$ and $\mathbf{Y}_t$ in their respective geometrical domains, and 2) explicitly modeling their relationship at the latent space as a function of the geometry. We realize our method in an encoder-decoder architecture with ST-GCNNs as summarized in Fig. 4.1: an ST-GCNN encoder embeds $\mathbf{Y}_t$ over the torso geometry, and an ST-GCNN decoder generates $\mathbf{X}_t$ over the ventricular geometry; at the latent space, the relationship between the latent variables of $\mathbf{Y}_t$ and $\mathbf{X}_t$ – as informed by the actual imaging physics – is assumed to be linear with coefficients as a function over the graph embedding of the two geometries. Following past ECGI works that showed the importance of including the temporal dimension in the reconstruction, we consider reconstructing the spatiotemporal signals on the heart over time.

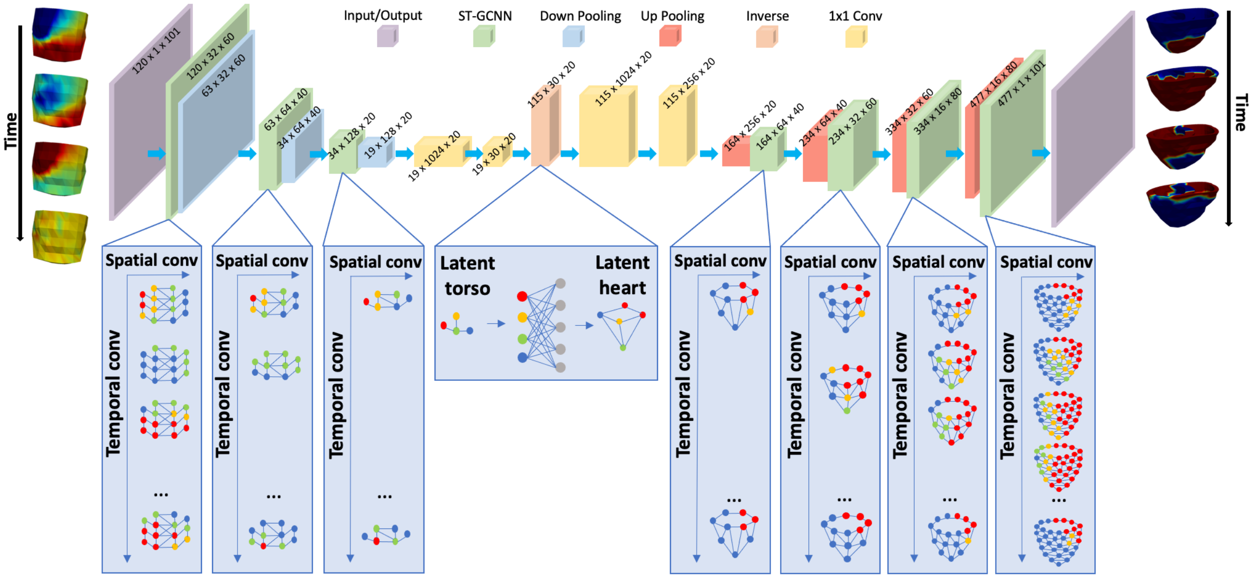


Figure 4.1: Outline of the ST-GCNN inverse imaging network. The size of the feature maps follows the format of #Vertices×#Feature×#Time.

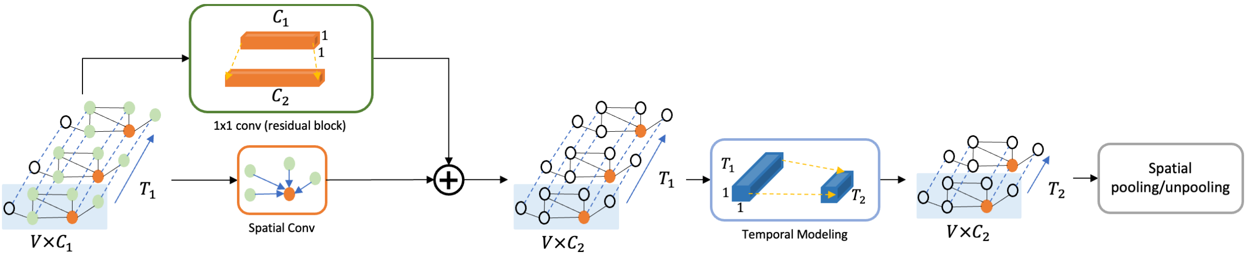


Figure 4.2: Illustration of the structure of one ST-GCNN layer followed by a spatial pooling/unpooling layer from Fig. 4.1. C_1 and C_2 are the numbers of channels before/after the spatial/regular convolution. T_1 and T_2 are the temporal dimensions before/after the temporal convolution.

4.2.1 Encoding-Decoding with ST-GCNNs

As $\mathbf{X}_t$ and $\mathbf{Y}_t$ are temporal sequences that exist within a 3D geometry, we describe their generation/embedding with ST-GCNNs that consist of interlaced graph convolution in space and regular compression in time as illustrated in Fig. 4.1.

Geometrical Representation in Graphs

Triangular meshes of the heart and torso surfaces are represented as two separate undirected graphs: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{U}, \mathbf{F})$, where vertices $\mathcal{V}$ consist of all V mesh nodes and edges $\mathcal{E}$ describe the vertex connection as defined by the triangular mesh. $\mathbf{U} \in [0,1]^{V \times V \times 3}$ consists of edge attributes $\mathbf{u}(i,j)$ between vertex i and j as normalized differences in their 3D coordinates if an edge exists. $\mathbf{F} \in \mathbb{R}^{V \times M \times T}$ represents the time sequences of features across all vertices, where the feature represents the signals at the input/output level when $M = 1$ and feature maps in the middle layers, where M is the size of the feature dimension.

During encoding and decoding, we apply hierarchical graph representations that coarsen as it gets closer to the bottleneck of the two geometries. In contrast to general graphs, this hierarchical representation should satisfy a constraint that the topology of the geometry must be preserved in its hierarchical representations to prevent non-physical spatial propagation of signals. The hierarchical representations are obtained by a specialized mesh coarsening method in the Computational Geometry Algorithms Library (CGAL) [18], defined prior to the training of the ST-GCNN.

Spatial Graph Convolution

We use a continuous spline kernel for spatial convolution such that it can be applied across different graphs [34]. For each channel of the feature map at each time instant, the convolution kernel is defined as:

$$g_l(\mathbf{u}) = \sum_{\mathbf{p} \in \mathcal{P}} w_{\mathbf{p},l} B_{\mathbf{p}}(\mathbf{u}), \tag{4.1}$$

where $1 \leq l \leq C$ and C is the number of channels. The spline basis $B_{\mathbf{p}}(\mathbf{u}) = \prod_{r=1}^{d} N_{r,p_r}^{m}(\mathbf{u})$ with N_{r,p_r}^{m} denoting d, an open B-spline basis of degree m based on equidistant knot vectors, $\mathcal{P} = (N_{1,r}^{m})_r \times \ldots \times (N_{d,r}^{m})_r$ is the Cartesian product of the B-spline bases, and $w_{\mathbf{p},l}$ are trainable parameters.

Given kernel $\mathbf{g} = (g_1, \ldots, g_C)$ and graph node features $\mathbf{f} \in \mathbb{R}^{V \times M}$ at each time instant, spatial convolution for vertex $i \in \mathcal{V}$ with its neighborhood $N(i)$ is defined as:

$$(f_l * g_l)(i) = \sum_{j \in N(i), \mathbf{p} \in \mathcal{P}(\mathbf{u}(i,j))} f_l(j) \cdot g_l(\mathbf{u}(i,j)). \tag{4.2}$$

Since the B-spline basis in the (4.1) is conditioned on local geometry, the learned kernel can be applied across graphs and the convolution incorporates geometrical information within the graph.

This spatial convolution is independently applied to each time instant of the signal sequence in parallel.

To make the network deeper and more expressive in feature representation, residual blocks are introduced to pass the input of spatial convolution through a skip connection with 1D convolution. Fig. 4.2 illustrates all components in an ST-GCNN layer.

Temporal Modeling

The spatial convolution is interlaced with temporal modeling. Common approaches for temporal modeling include using a standard convolutional kernel [72] or a recurrent unit of time sequences [52]. Here, we apply directly the fully connected layers on time sequences, where the parameter of the layer is shared across each vertex and feature. The size of the output of the fully connected layer is set to compress the time sequence in dimension in the encoder, while expanding in the decoder. The geometry graph remains the same for the complete temporal sequences. In Section 4.3.6, this temporal modeling will be compared with alternative RNN or CNN options in experimental evaluations.

Pooling and Unpooling

Pooling and unpooling in space are carried out on the hierarchical graph representation of the two geometries described in Section 4.2.1. Using $\mathcal{G}_o$ to denote a graph with N_1 vertices and $\mathcal{G}_c$ its coarsened graph with N_2 vertices, we use a binary matrix $\mathbf{P} \in \mathbb{R}^{N_1 \times N_2}$, where $\mathbf{P}_{ij} = 1$ if vertex i in $\mathcal{G}_o$ is grouped to vertex j in $\mathcal{G}_c$, and $\mathbf{P}_{ij} = 0$ otherwise. Here, each vertex on $\mathcal{G}_o$ is grouped to its nearest vertex on $\mathcal{G}_c$. Given a feature map $\mathbf{f}_o \in \mathbb{R}^{N_1 \times M}$ over $\mathcal{G}_o$ and $\mathbf{f}_c \in \mathbb{R}^{N_2 \times M}$ over $\mathcal{G}_c$, the pooling operation is defined by $\mathbf{f}_c = \mathbf{P}_n^T \mathbf{f}_o$ and the unpooling operation is defined by $\mathbf{f}_o = \mathbf{P}\mathbf{f}_c$, where $\mathbf{P}_n^T$ is column normalized from $\mathbf{P}$.

Summary

As summarized in Fig. 4.2, each ST-GCNN block consists of spatial graph convolution, temporal compression, and spatial pooling/unpooling, as described above. Denoting the latent features of the body-surface signals as $\mathbf{z}_b$ and those of the heart-surface signals as $\mathbf{z}_h$, respectively, we can represent the encoder and decoder as $\mathbf{z}_b = E_\theta(\mathbf{Y})$ and $\hat{\mathbf{X}} = D_\phi(\mathbf{z}_h)$, where θ and ϕ are parameters of the encoder and decoder, respectively.

4.2.2 Learning Latent Geometry-Dependent Physics

As explained in (2.1), $\mathbf{Y}_t$ on one torso vertex can be represented as a linear combination of $\mathbf{X}_t$ from all heart vertices, where the coefficients are determined by the relative position between each pair of torso-heart vertices. We assume this linearity to hold between $\mathbf{z}_h$ and $\mathbf{z}_b$ in the latent space. Specifically, for $\mathbf{z}_h(i)$ on vertex i of the latent heart mesh, we define it as a linear combination of latent features $\mathbf{z}_b(j)$ across all vertices j of the latent torso mesh.

One option to learn this linear mapping between $\mathbf{z}_b$ and $\mathbf{z}_h$ is a fully connected layer. However, the resulting learned relationship will not consider the underlying geometry and, more importantly, will not be applicable to different heart-torso geometries with different numbers of graph vertices. Instead, we explicitly model the learnable coefficients as functions of the relative position between embedded heart and torso geometries. To do so, we construct a bipartite graph such that an edge with attribute $\mathbf{u}(i,j)$ exists between each pair of heart and torso vertices in their graph embeddings. We then define $\mathbf{z}_h(i)$ as a linear combination of $\mathbf{z}_b(j)$ across all vertices j, where linear coefficients $\hat{\mathbf{h}}$ are learnable as a function over $\mathbf{u}(i,j)$:

$$\mathbf{z}_h(i) = \sum_j \mathbf{z}_b(j) \cdot \hat{\mathbf{h}}(\mathbf{u}(i,j)). \tag{4.3}$$

Exploiting the similarity between (4.3) and (4.2), we describe (4.3) using spline convolution with the geometry-dependent coefficients $\hat{\mathbf{h}}$ learned as the spline convolution kernel. We denote the geometry-dependent inverse function as $\mathbf{z}_h = h_\rho(\mathbf{z}_b)$ with network parameter ρ. Aside from being physics-informed, this geometric parameterization allows the learned function to generalize across different geometries.

4.2.3 Deterministic and Stochastic Formulations

As explained earlier, we developed both a deterministic and stochastic formulation of the non-Euclidean encoding-decoding networks, in order to investigate whether the use of IB theory is still necessary for improving the ability of the network to generalize to different geometries.

For the deterministic model, its parameters θ, ρ and ϕ are optimized by minimizing the mean square error between the reconstructed $\hat{\mathbf{X}}^{(i)}$ on training data $\{\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\}_{i=1}^{N}$:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{X}^{(i)} - D_\phi\left(h_\rho\left(E_\theta\left(\mathbf{Y}^{(i)}\right)\right)\right)||_2^2. \tag{4.4}$$

For the stochastic model, $\mathbf{z}_b$ is modeled with a Gaussian distribution whose mean and variance are obtained by neural networks: $p_\theta(\mathbf{z}_b|\mathbf{Y}) = \mathcal{N}(\mathbf{z}_b|\boldsymbol{\mu}_b(\mathbf{Y}), \boldsymbol{\sigma}_b^2(\mathbf{Y}))$. We apply reparameterization $\mathbf{z}_b = \boldsymbol{\mu}_b + \boldsymbol{\sigma}_b \odot \boldsymbol{\epsilon}$ as described in [69], where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and $\odot$ is Hadamard product. We draw a random sample from the distribution of $\mathbf{z}_b$, and then apply $\mathbf{z}_h = h_\rho(\mathbf{z}_b)$ to obtain the sample for $\mathbf{z}_h$. The decoder then reconstructs from this sample. From the theory of information bottleneck [116], we minimize:

$$loss_{IB} = -I(\mathbf{X}; \mathbf{z}_h) + \beta I(\mathbf{z}_b; \mathbf{Y}) \tag{4.5}$$

where $I(\mathbf{X}; \mathbf{z}_h)$ is the mutual information between the output and latent features of the heart signals, $I(\mathbf{z}_b; \mathbf{Y})$ is the mutual information between the measurement and latent features of the torso signals, and β is the multiplier of the KL-divergence term. For the first term in (4.5) we have:

$$\begin{aligned}
I(\mathbf{X}; \mathbf{z}_h) &= \int p(\mathbf{X}, \mathbf{z}_h) \log \frac{p(\mathbf{X}|\mathbf{z}_h)}{p(\mathbf{X})} d\mathbf{X} d\mathbf{z}_h \\
&= H(\mathbf{X}) + \int p(\mathbf{X}, \mathbf{z}_h) \log p(\mathbf{X}|\mathbf{z}_h) d\mathbf{X} d\mathbf{z}_h \\
&= H(\mathbf{X}) + \int p(\mathbf{z}_h) p(\mathbf{X}|\mathbf{z}_h) \log \frac{p(\mathbf{X}|\mathbf{z}_h)}{q(\mathbf{X}|\mathbf{z}_h)} d\mathbf{X} d\mathbf{z}_h \\
&+ \int p(\mathbf{z}_h) p(\mathbf{X}|\mathbf{z}_h) \log q(\mathbf{X}|\mathbf{z}_h) d\mathbf{X} d\mathbf{z}_h \\
&= H(\mathbf{X}) + D_{KL}(p(\mathbf{X}|\mathbf{z}_h)||q(\mathbf{X}|\mathbf{z}_h)) \\
&+ \int p(\mathbf{X}, \mathbf{Y}, \mathbf{z}_h) \log q(\mathbf{X}|\mathbf{z}_h) d\mathbf{X} d\mathbf{Y} d\mathbf{z}_h \\
&\geq E_{p(\mathbf{X},\mathbf{Y})}[E_{p(\mathbf{z}_b|\mathbf{Y})}[\log q(\mathbf{X}|\mathbf{z}_h)]],
\end{aligned}$$

where $\mathbf{z}_h = \hat{\mathbf{H}}(\mathbf{g})\mathbf{z}_b$ and $H(\mathbf{X}) = \int p(\mathbf{X}) \log p(\mathbf{X}) d\mathbf{X}$. We set $q_\phi(\mathbf{X}|\mathbf{z}_h)$ to be a Gaussian distribution parameterized by the decoder: $q_\phi(\mathbf{X}|\mathbf{z}_h) = \mathcal{N}(\mathbf{X}|\mu_h, \sigma_h^2)$.

For the second term in (4.5):

$$\begin{aligned}
I(\mathbf{z}_b;\mathbf{Y}) &= \int p(\mathbf{Y},\mathbf{z}_b)\log\frac{p(\mathbf{z}_b|\mathbf{Y})}{p(\mathbf{z}_b)}d\mathbf{Y}d\mathbf{z}_b \\
&= \int p(\mathbf{Y},\mathbf{z}_b)\log\frac{p(\mathbf{z}_b|\mathbf{Y})r(\mathbf{z}_b)}{r(\mathbf{z}_b)p(\mathbf{z}_b)}d\mathbf{Y}d\mathbf{z}_b \\
&= \int p(\mathbf{Y})p(\mathbf{z}_b|\mathbf{Y})\log\frac{p(\mathbf{z}_b|\mathbf{Y})}{r(\mathbf{z}_b)}d\mathbf{Y}d\mathbf{z}_b \\
&- D_{KL}(p(\mathbf{z}_b)||r(\mathbf{z}_b)) \\
&\leq \int p(\mathbf{X},\mathbf{Y})p(\mathbf{z}_b|\mathbf{Y})\log\frac{p(\mathbf{z}_b|\mathbf{Y})}{r(\mathbf{z}_b)}d\mathbf{X}d\mathbf{Y}d\mathbf{z}_b \\
&= E_{p(\mathbf{X},\mathbf{Y})}[D_{KL}(p(\mathbf{z}_b|\mathbf{Y})||r(\mathbf{z}_b))],
\end{aligned}$$

where the KL divergence is analytically available when $r(\mathbf{z}_b)$ is set to be a standard Gaussian distribution: $r(\mathbf{z}_b) = \mathcal{N}(\mathbf{z}_b|\mathbf{0},\mathbf{I})$ and the latent distribution of torso graph embedding as a Gaussian distribution: $p_\theta(\mathbf{z}_b|\mathbf{Y}) = \mathcal{N}(\mathbf{z}_b|\boldsymbol{\mu}_b(\mathbf{Y}))$. Therefore, we have:

$$\begin{aligned}
loss_{IB} &= -I(\mathbf{X};\mathbf{z}_h) + \beta I(\mathbf{z}_b;\mathbf{Y}) \\
&\leq E_{p(\mathbf{X},\mathbf{Y})}[-E_{p_\theta(\mathbf{z}_b|\mathbf{Y})}[\log q_\phi(\mathbf{X}|\mathbf{z}_h)]] \\
&+ \beta D_{KL}(p_\theta(\mathbf{z}_b|\mathbf{Y})||r(\mathbf{z}_b)) = \mathcal{L}_{IB}.
\end{aligned}$$

4.3 Simulation Data Experiments

In controlled simulation experiments, we evaluated the performance of the reconstruction network in a series of generalization tasks, with increasing difficulty in terms of how close the test geometry is to those in training data. We further assessed how the performance of the network changed as the diversity of the training data decreased. We compared the performance of our network to that of a Euclidean encoding-decoding network, as described by Ghimire *et al.* [41] in a deterministic formulation as well as a stochastic formulation with a geometry-invariant bottleneck. In a subset of experiments where Euclidean networks do not apply (Section 4.3.4, training across multiple geometries), we further compared the performance of our network to classic ECGI approaches utilizing known physics-based forward operators.

4.3.1 Models, Data, and Training

Our presented network consisted of three ST-GCNN blocks and two standard convolutional layers in the encoder, one spline convolutional layer in the latent inverse mapping, and four ST-GCNN blocks and two standard convolutional layers in the decoder, as detailed in Fig. 4.1. We used B-spline basis degree of $m = 1$ with kernel size of $k_1 = k_2 = k_3 = 3$ in all graph convolution layers. The spatial and temporal dimensions in each level of the encoder were $[120, 63, 34, 19]$ and $[101, 60, 40, 20]$, respectively; and $[115, 164, 234, 334, 477]$ and $[20, 40, 60, 80, 101]$ in the decoder, respectively. We used ELU activation [22], an ADAM optimizer [68], and a learning rate of 5×10^{-4}. The Euclidean baselines followed the architectures presented in [41] consisting of cascaded LSTMs and fully-connected layers.

For training, we generated pairs of simulated potentials on a heart-torso mesh. We simulated realistic spatiotemporal propagation sequences of action potentials by the Aliev-Panfilov model [1], considering a combination of 38 origins of activation and 16 spatial distributions of scar tissue (totaling 531 data samples for a single geometry). We then rotated the heart by -2° to 2° around the z-axis, obtaining approximately 2700 sets of body-surface potentials. All body-surface potentials were corrupted with 20 dB SNR of Gaussian noises before performing inverse imaging.

Testing data were generated in a similar fashion, with additional geometry changes as detailed below. The reconstruction accuracy was measured by the mean square error (MSE), spatial correlation coefficient (SCC), and temporal correlation coefficient (TCC) between the reconstructed and actual potential sequence on the heart surface. While MSE measures the quantitative errors of the reconstructed signals on the heart, the SCC and TCC measure the correlation between the reconstructed and reference signals. We considered the correlation both for spatial signals at each time instance (SCC) and temporal signals at each location of the heart (TCC).

4.3.2 Generalization to Unseen Heart Rotations

In this set of experiments, we first applied the trained models to 21,771 sets of body-surface potentials generated by rotating the heart by -20° to 20° around the z-axis, a range far outside that considered in training. We then tested the trained models on a different set of 64,782 body-surface potentials generated from novel heart rotations around the x-axis (-20° to +40°) and y-axis (-20° to +40°), types of rotations not seen in training. In both experiments, we examined the change of performance of the trained models when 1) we randomly sampled the training data with respect to the combinations of sites of activation and scar locations by a rate of 2%, 4%, 10%, 25%, 50%, 75%, and 100% of the complete training data, and 2) we reduced the number of rotations around z-axis in the training data from -2° to 2° (five geometries), to -1° to 1° (three geometries), and eventually to no rotation at all (one single geometry).

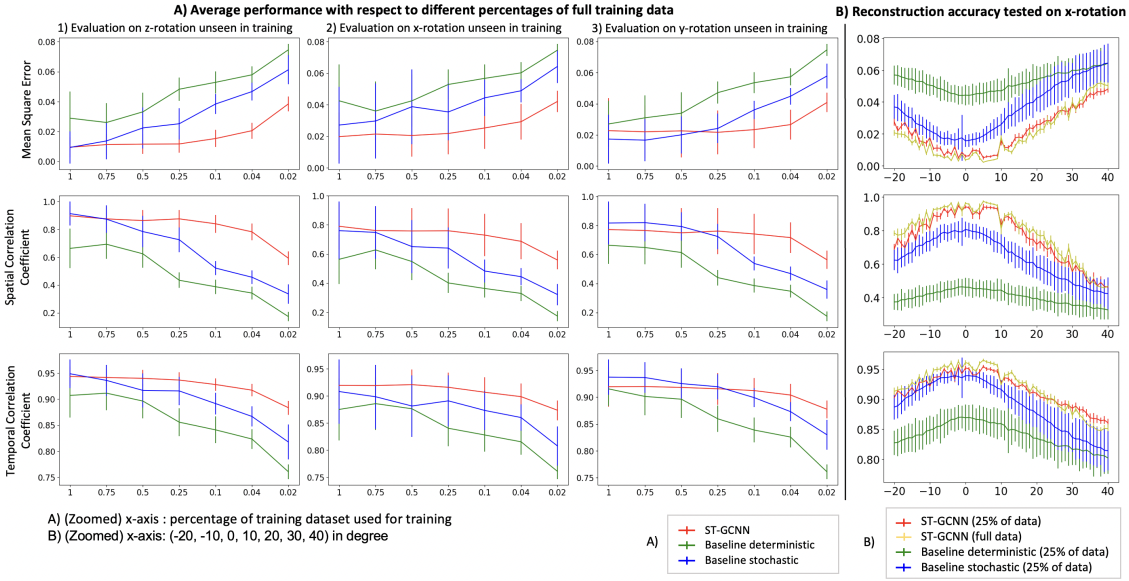


Figure 4.3: Summary of average performance with respect to different percentages of full training data among the three comparison models.

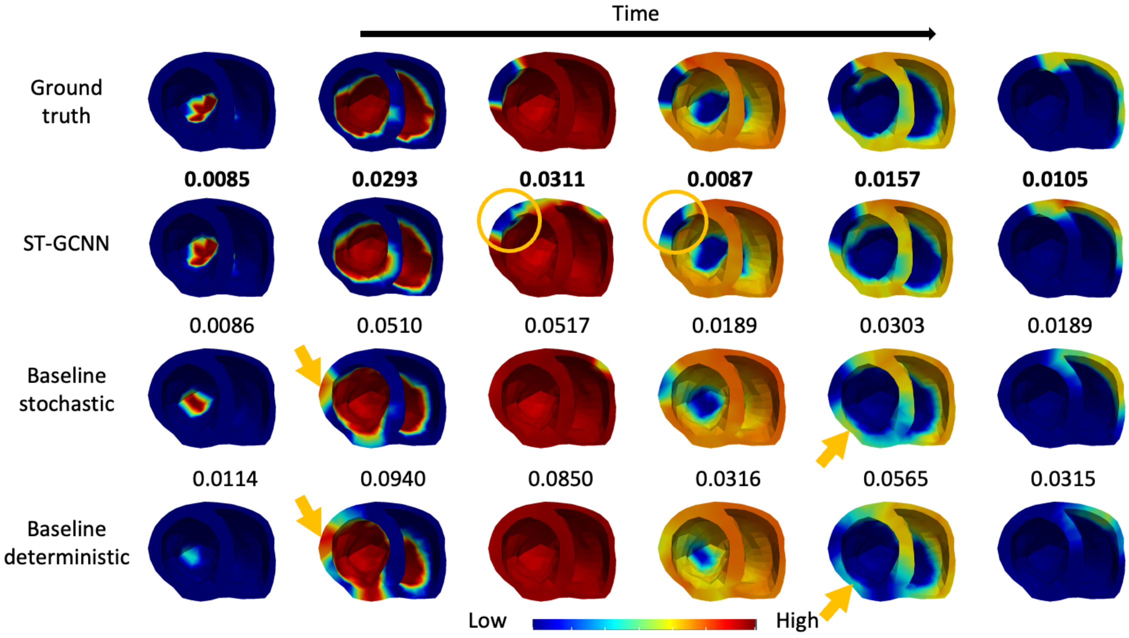


Figure 4.4: Examples of reconstructed electrical activity trained on 25% of full training data, tested at rotation x = 10°. The MSE value is shown for each model at each timestep. Both of the two baseline models showed high errors on the free wall of the left ventricle during depolarization/repolarization (arrows), while our presented model predicted the most accurate signal propagation pattern and the scar location (circles).

Generalization with reduced diversity in training data

Fig. 4.3A1 shows quantitative metrics of the three models on z-axis rotations unseen in the training set (21,771 cases), against the number of combinations of earliest activation sites and scar locations. The decrease in performance (reduced accuracy and increased standard deviation) in the face of reduced training data was significantly slower by our method (red) compared to the deterministic (green) and stochastic (blue) Euclidean baselines. Fig. 4.3A2-3 lists the performance of the three models on geometry with novel x- and y-rotations, showing a similar trend. While the performance of the ST-GCNN was similar and occasionally worse than the stochastic Euclidean baseline (*e.g.*, generalizing to y-rotations in Fig. 4.3A3) at the full diversity of the training data, it outperformed both baselines in all metrics as the training diversity decreased to 25%. In fact, its performance did not show notable change in performance until the number of training cases was reduced to below 5-10% of the complete set.

As an example, Fig. 4.3B shows the detailed quantitative metrics of the three models trained on 25% of the full training data, against x-axis rotations of the heart as measured by differences from training data. The presented method (red) outperformed the deterministic (green) and stochastic (blue) Euclidean baseline in all metrics, and its performance changed only slightly compared to that using full training data (yellow). Fig. 4.4 provides visual examples of the comparisons. The presented ST-GCNN model predicted the most accurate signal propagation pattern and scar location, while both baseline models showed errors on the free wall of the left ventricle.

Generalization with reducing training geometry

Fig. 4.5A1 summarizes quantitative metrics of the three models against the number of geometrical models in training. Our method (red) again showed a slower decrease in the average error and an increase in standard deviation than the two Euclidean baselines. The testing results on novel x- and y-rotations showed a similar trend (Fig. 4.5A2-3). Note that these experiments were performed using training data that contained the full diversity, as described in the section above, thus the performance of ST-GCNN was lower than the stochastic Euclidean baseline when generalizing to y-rotations of the heart using five or three geometries.

Fig. 4.5B shows quantitative metrics of the three models when trained on a single geometry and tested on data following z-axis rotations of the heart. The decrease in performance compared to using more geometries (yellow curve) was more evident compared to the results in Fig. 4.3B, although it was much less significant compared to the two baselines. Fig. 4.6B provides visual

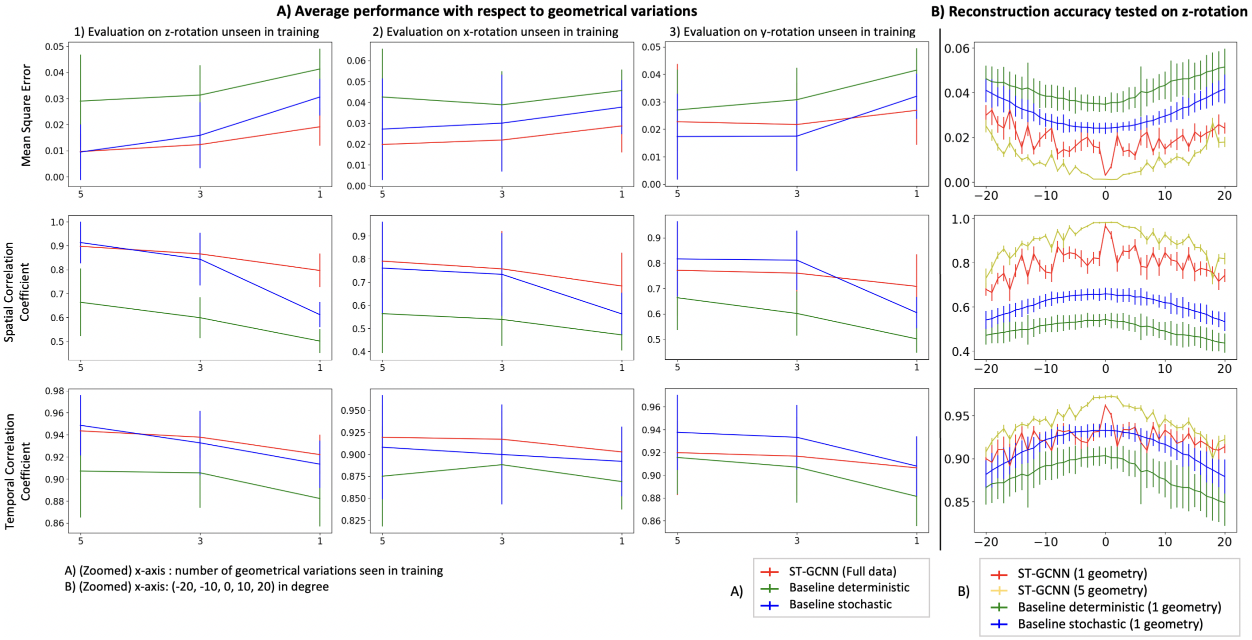


Figure 4.5: Summary of average performance with respect to geometrical variations among the three comparison models.

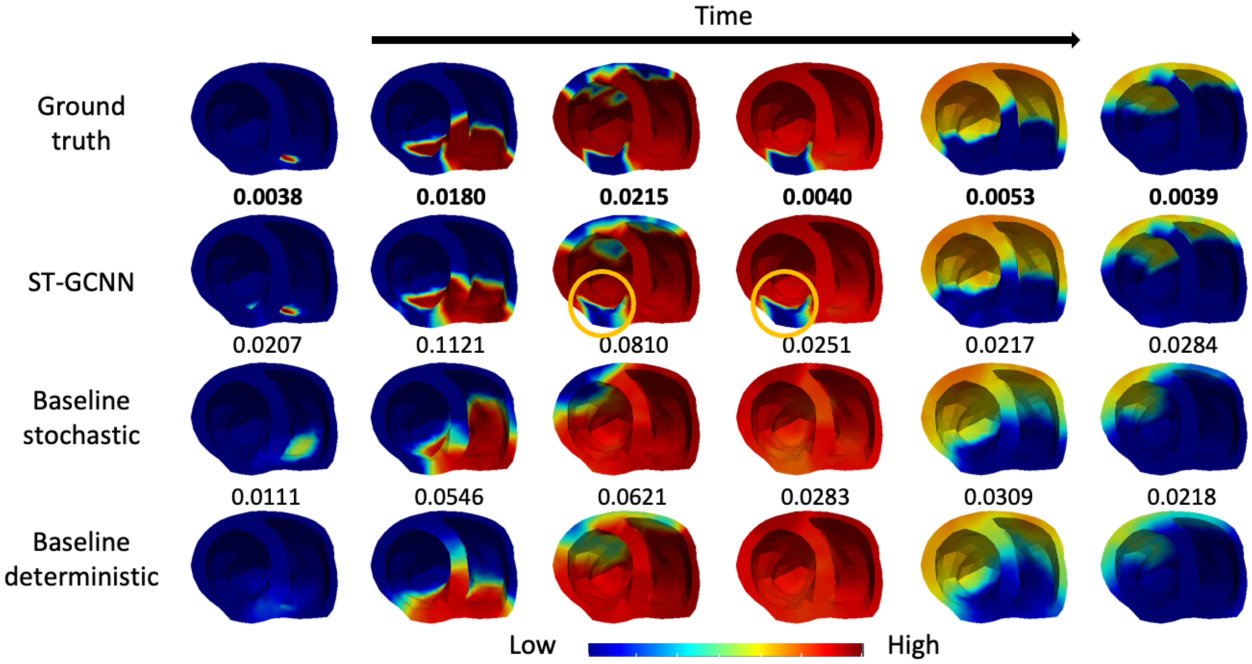


Figure 4.6: Visual examples of reconstructed electrical activity trained on one single geometry and tested on the heart with a rotation at z = 10°. The MSE value is shown for each model at each timestep. Both baseline models showed substantial errors in locating the scar, while our model predicted the most accurate signal propagation pattern and the scar location (circles).

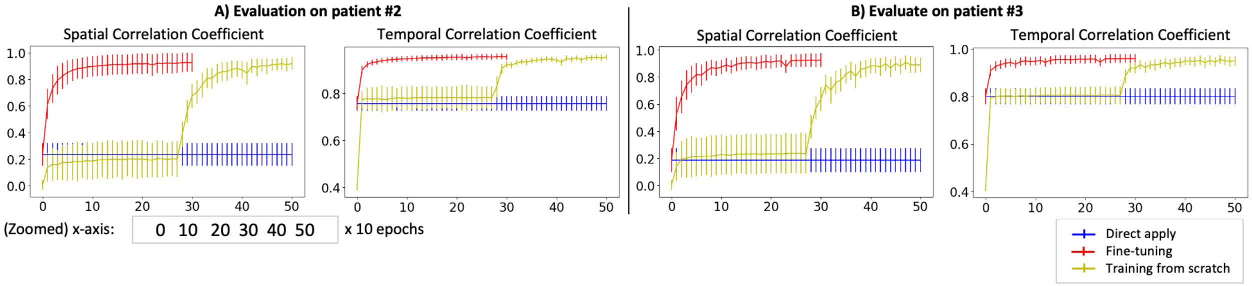


Figure 4.7: Convergence of reconstruction accuracy on new geometry by the fine-tuned (red) *vs.* retrained model (yellow).

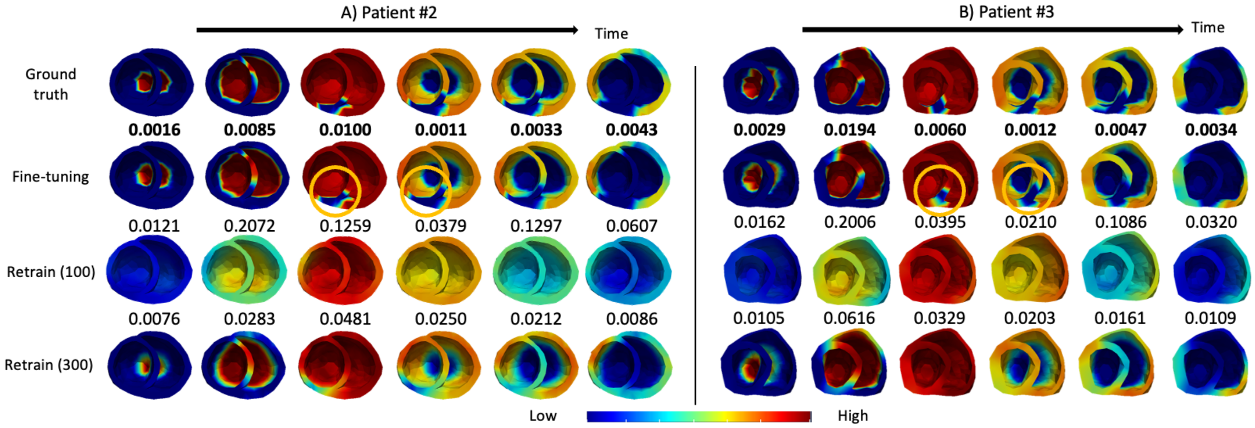


Figure 4.8: Examples of reconstructed electrical activity by the fine-tuned model at epoch = 100, the retrained model at epoch = 100, and the retrained model at epoch = 300 on A) patient #2 and B) patient #3. The MSE value is shown for each model at each timestep. The fine-tuned model has the most accurate reconstruction of the signal propagation pattern and the scar location (circles).

examples of the reconstructed image sequences. Similar to the example shown in Fig. 4.4, ST-GCNN was the most accurate at predicting signal propagation and locating the scar, while both baseline models indicated incorrect scar locations.

4.3.3 Generalization to New Geometry

We then applied the model to 491 and 444 sets of simulated potentials generated on each of two new heart-torso meshes, respectively. This scenario is realistic in that the network trained on one group of patients will be applied to new patients. The blue flat line in Fig. 4.7 shows the accuracy

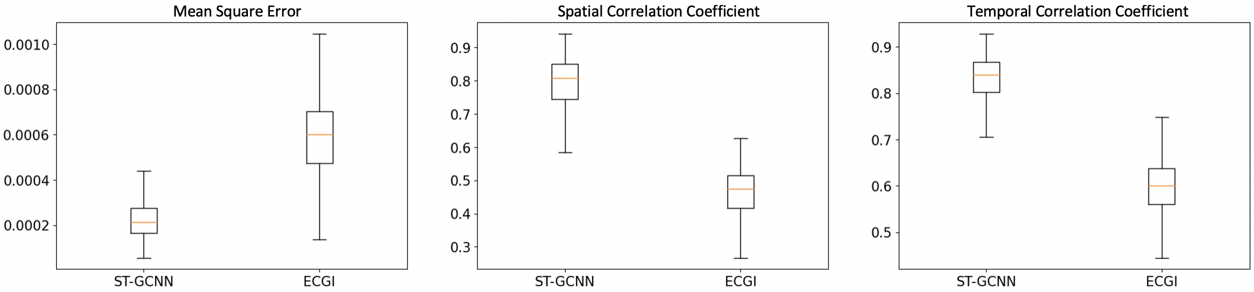


Figure 4.9: Performance of our ST-GCNN method *vs.* ECGI.

of the results, trained on the full data set as described earlier and directly applied to the test data from the two new geometries. The performance (patient #2: MSE = 0.11, SCC = 0.23, TCC = 0.76; patient #3: MSE = 0.08, SCC = 0.19, TCC = 0.80) showed a larger decrease compared to those in Section 4.3.2, yet it is satisfactory overall. Note that Euclidean networks cannot be applied directly on a new geometry without re-training [41]. We then fine-tuned the model on a small set of data for each new geometry (122 and 111 sets, respectively), in comparison to retraining the model from scratch. As shown in Fig. 4.7, the fine-tuned model was much faster (< 100 epochs) to converge than the retrained model (> 300 epochs). Fig. 4.8 provides examples of the results. The fine-tuned model had the most accurate reconstruction of both the signal propagation pattern and the scar location among all the models we compared.

4.3.4 Training and Testing on Multiple Geometries

We further evaluated the ability of our model to be trained and tested across different geometries of the heart and torso, *i.e.*, across different input graphs. Such cross-training is not possible with Euclidean models unless the different heart and torso geometries are pre-registered. Even after registration, the learning will completely miss the different geometry information underlying the data. Therefore, in this set of experiments, we compared our method with traditional ECGI methods that rely on first building a forward mapping operator based on the given geometry, and then optimizing for the reconstruction, given the forward operator and applying second-order Tikhonov regularization [108]. Since unipolar extracellular potential, instead of transmembrane potential, is more commonly used as the source model in traditional ECGI methods, we chose to apply the presented method to reconstruct extracellular potential as well. Specifically, we considered the meshes in Section 4.3.1 and Section 4.3.3 for the simulation of extracellular potentials. We trained our model on a small subset of data for each geometry (50, 50, and 50 sets, respectively). We then evaluated the model on each geometry used in the training set but with rotations unseen in the training set. Fig. 4.9 shows that our method had a much better reconstruction accuracy than the ECGI method.

4.3.5 Effect of Stochastic IB Formulation

As observed in Section 4.3.2, in the Euclidean baselines, the stochastic model outperformed the deterministic one. This result was consistent with published results [41], which attributed this gain to the theory of information bottleneck in helping remove geometry-related information from the latent space. With our network trained as a function of the geometry, we expected that the stochastic IB formulation would bring less significant benefits. The results in Fig. 4.10B verified this assertion, as the performance of the stochastic model changed only minimally when using different values of β. Using $\beta = 1 \times 10^{-2}$ as an example (Fig. 4.10C), the performance gap between the stochastic and deterministic models was marginal.

4.3.6 Effect of Alternative Model Architectures

Temporal Modeling

We investigated several alternatives for temporal modeling in our network including: 1) interlaced spatial graph convolution and local temporal convolution [72] (L-Conv), which uses a 5×1 standard convolutional kernel to slide through the time sequence on each node and feature, 2) interlaced spatial graph convolution and regular Long Short-Term Memory (LSTM) networks [52] on temporal sequences, and 3) graph LSTM (GCN-LSTM), which replaces the fully connected operator in LSTM [52] with graph convolution so that the layer can operate on graph data. Table. 4.1 shows that the presented ST-GCNN model was the most efficient and achieved the best performance.

Residual Blocks

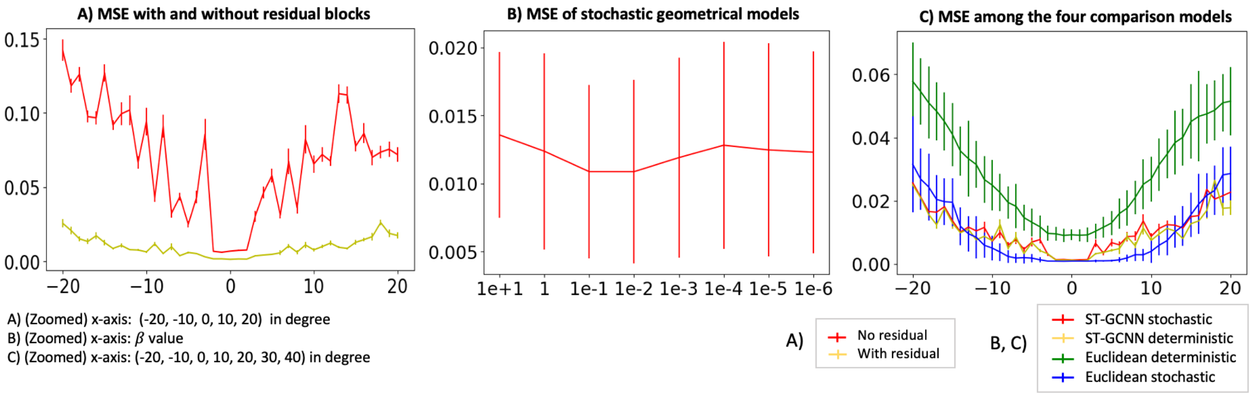
We further trained a geometric network without residual blocks on the dataset used in Section 4.3.1 and tested it on the same dataset of rotations as described in Section 4.3.2. Fig. 4.10A summarizes the mean square error of two networks against the change in heart rotations from the training data. As shown, without the residual blocks, the network was not able to accurately reconstruct heart potentials.

Table 4.1: Comparison of Architectures

Architecture Type	Time Complexity[a]	Average MSE[b]
L-Conv	12.1 min	0.0144
LSTM	29.5 min	0.0118
GCN-LSTM	32.6 min	0.0139
ST-GCNN	**9.5 min**	**0.0096**

[a] Average time per epoch.

[b] Average mean square error tested on all z-rotations with all models trained for 300 epochs. The other metrics followed the same trend.



A) (Zoomed) x-axis: (-20, -10, 0, 10, 20) in degree
B) (Zoomed) x-axis: β value
C) (Zoomed) x-axis: (-20, -10, 0, 10, 20, 30, 40) in degree

Figure 4.10: A) Effect of residual blocks; B) Effect of hyperparameter value of β; C) Comparison of model performance at $\beta = 0.01$.

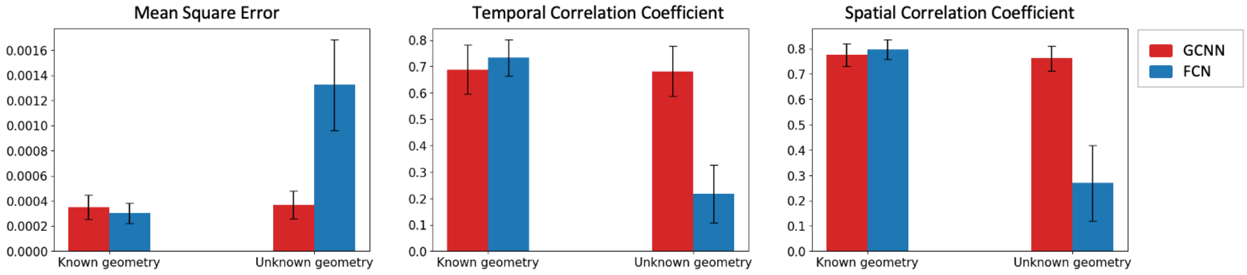


Figure 4.11: Performance of GCNN and non-linear fully-connected (FCN) latent inverse mapping.

Latent inverse mapping

We investigated a more general modeling option at the latent space for the relationship between latent embedding $\mathbf{Z}_h$ and $\mathbf{Z}_b$: a fully-connected layer with ELU activation [22]. This describes $\mathbf{Z}_h$

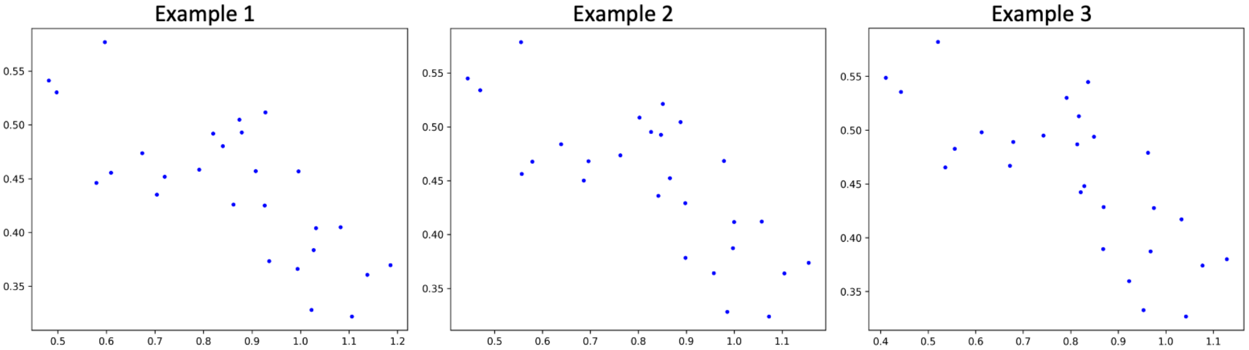


Figure 4.12: Examples of the relation between the norm of $\mathbf{z}_b(j)\cdot\hat{\mathbf{h}}(\mathbf{u}(i,j))$ (y-axis) and the distance $\mathbf{u}(i,j)$ between vertex i and j on the bipartite graph (x-axis).

as a general nonlinear function of $\mathbf{Z}_b$ without considering the underlying geometry. We trained this network on 50 data samples on one single geometry and tested it on 3,095 data samples from the same geometry but with five different rotations of the heart. As shown in Fig. 4.11, the resulting network – while performing well on the training geometry – struggled with testing data from different geometries.

To further understand the learned latent inverse mapping in (4.3), we examined empirical evidence on how the coefficient of the linear inverse mapping $\hat{\mathbf{h}}(\mathbf{u}(i,j))$ changed with the relative distance between vertex i and j in the latent torso and heart graphs. Due to the high dimensionality of $\mathbf{z}_h$ and $\mathbf{z}_b$, this was difficult to observe directly. Instead, we set all $\mathbf{z}_b(j)$ on the torso graphs to be identical constants, chose random vertex i on the heart graph, and obtained the norm of $\hat{\mathbf{h}}(\mathbf{u}(i,j))$ for each j which describes how each $\mathbf{z}_b(j)$ contributed to $\mathbf{z}_h(i)$ for different vertices j. Fig. 4.12 shows three examples of randomly selected i on the heart graph: consistent with the known physics, the contribution of each $\mathbf{z}_b(j)$ to $\mathbf{z}_h(i)$ decreased as the distance of (i, j) increased. This suggests that the inverse mapping learned geometry-dependent functions consistent with the underlying physics.

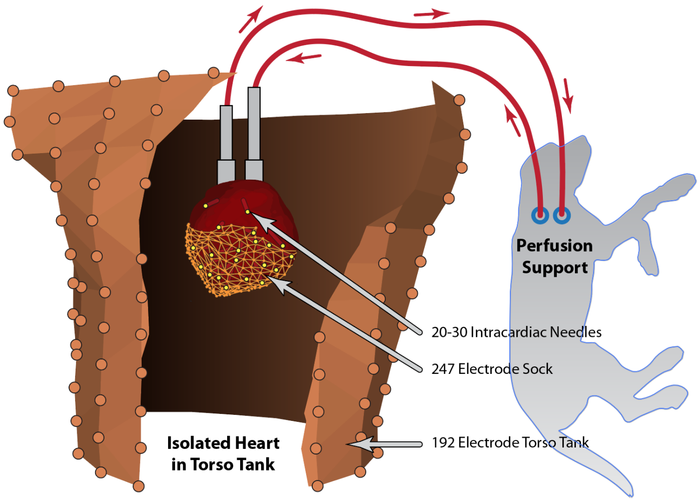


Figure 4.13: A flexible epicardial sock array encircled the isolated heart, which was perfused from a second support animal through the aorta with blood returned under suction from the right ventricle. The human-torso-shaped tank was filled with electrolytic fluid consistent with human torso conductivity and contained 192 embedded Ag/AgCl electrodes. The recording system sampled cage and torso potentials simultaneously. Bipolar stimulation was initiated from intramural plunge needles.

4.4 Real Data Experiments

4.4.1 Experimental Data Description

Torso Tank Experimental Preparation

The experimental data sets used in this study were acquired from a modified Langendorff-perfused torso tank preparation [11]. As illustrated in Fig. 4.13, an isolated canine heart was suspended within a human-shaped torso tank and perfused via arterial blood from a second support dog. Blood was returned to the support dog from a right ventricular cannula to the jugular vein. The human-shaped torso tank was filled with an electrolytic solution (resistivity was 500 Ω-cm), which approximates the electrical conductivity of a human torso. The animals were under deep anesthesia using procedures approved by the Institutional Animal Care and Use Committee of the University of Utah and conformed to the Guide for the Care and Use of Laboratory Animals.

Signal Acquisition

Cardiac activation was generated with bipolar stimulation using plunge needles at five sites: left ventricular (LV) base, LV Apex, LV freewall, LV septum, and right ventricular (RV) free wall. All stimulation was initiated near the endocardium and signals were recorded for five seconds.

Cardiac potentials were recorded using an epicardial sock with 247 electrodes (inter-electrode spacing 6.5±1.3 mm) stretched over the ventricles of the heart. The torso tank had 192 silver/silver-chloride electrodes (with inter-electrode spacing 40.2±16.8 mm) distributed across the outer surface. All signals were referenced to a Wilson's Central Terminal and were simultaneously sampled at 1000 Hz. Signals were filtered, annotated, and post-processed using PFEIFER [105].

Geometric Model Creation

The surface geometries were constructed based on electrode locations acquired during each experiment. Template geometries for both the torso tank and epicardial sock were registered using known correspondence points, which were measured using a 3D mechanical digitizer (Microscribe from Immersion Corp). The epicardial sock registration was further refined as described previously [25].

4.4.2 Evaluation & Results

120 out of 192 torso-tank measurements were selected for inverse imaging, to be consistent with the number of input measurements used in the synthetic training data. The measured QRST signals were downsampled using polyphase filtering to the length of the simulated training signals. The epi-endocardial geometry used in Section 4.3.1 was registered to the epicardial sock geometry with transition, rotation, and scaling operations. The measured epicardial potential and the sites of stimulation were registered to this epi-endocardial model, to provide reference data for evaluation. The inverse imaging results were evaluated by MSE, SCC, and TCC metrics against the measured epicardial potentials.

We carried out cross-validation by leaving out the signals from one stimulated activation sequence (5 sequences) each time. We first directly applied a model trained on simulated data, considering a subset from those described in Section 4.3.1 including 72 different combinations of activation origins and scar tissues. We then fine-tuned the trained model using measured signals from the remaining four stimulation sites (20 sequences). Finally, we retrained the same model from scratch

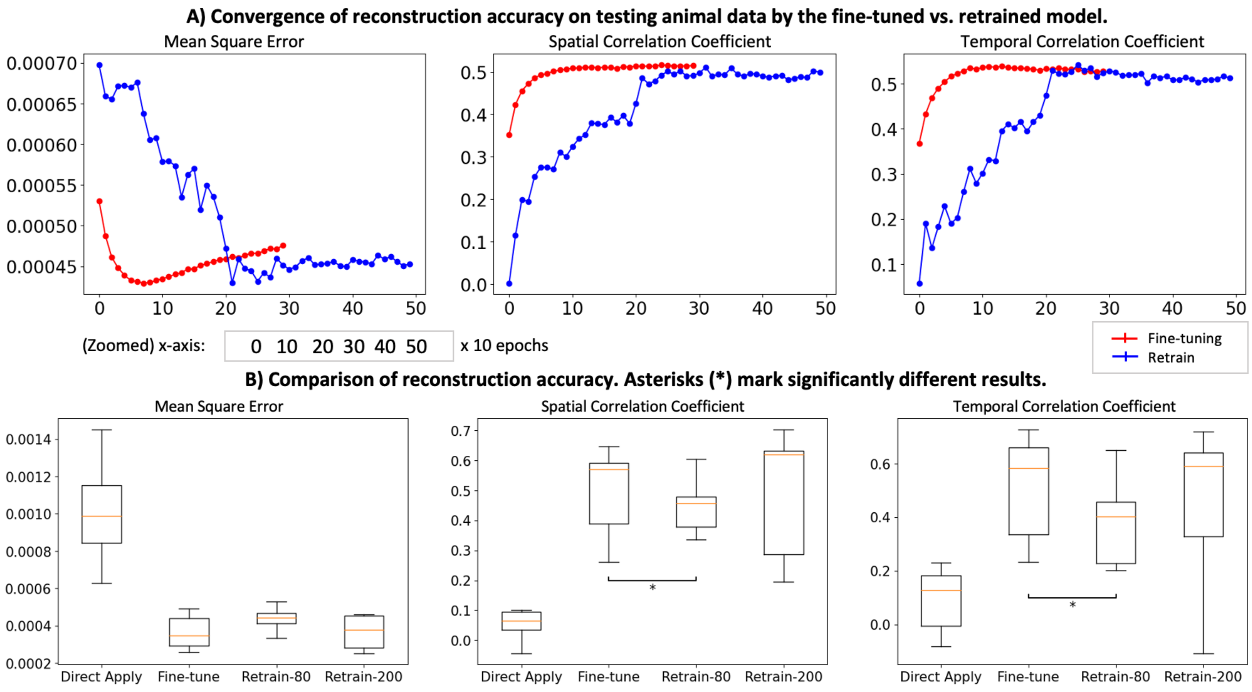


Figure 4.14: Performance of comparison models on measured data: A) Convergence of fine-tuned *vs.* retrained model; B) Final accuracy of various models.

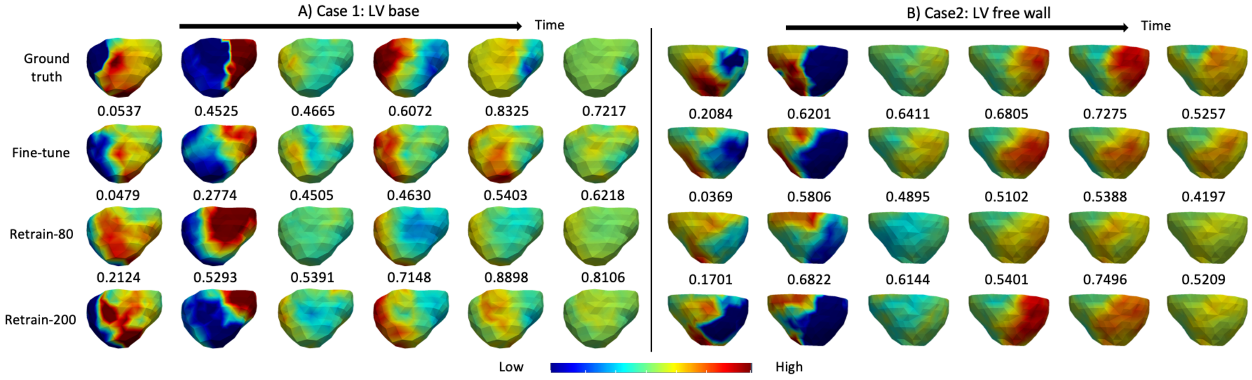


Figure 4.15: Examples of reconstructed electrical activation sequences stimulated from A) LV base and B) LV free wall. The SCC value is shown for each model at each timestep. The fine-tuned model and the retrained model at epoch = 200 had similar performance.

using the same experimental data. Fig. 4.14A illustrates the quantitative metrics on the test data, averaged over the cross-validation folds as the training of the fine-tuned and re-trained models converged. Similar to the results presented in Section 4.3.3, the fine-tuned model took many fewer epochs (80 epochs) to converge than the retrained model (200 epochs). Fig. 4.14B summarizes quantitative metrics obtained by these models. As shown, all metrics of the fine-tuned model were significantly better than the retrained model at epoch 80 ($p = 0.05$ (MSE), 0.013 (SCC), and 0.001 (TCC), paired t-tests), and moderately better than the retrained model after convergence ($p = 0.32$ (MSE), 0.10 (SCC), and 0.14 (TCC), paired t-tests). Fig. 4.15 provides visual examples of two paced activations. The fine-tuned model and the retrained model at epoch = 200 had similar performance on the prediction of propagation of activation.

4.5 Discussion and Conclusions

We have presented a novel non-Euclidean network for learning geometry-dependent and physics-based inverse mapping between spatiotemporal variables mapped to 3D geometrical domains. We demonstrated its ability to improve generalization to unseen geometrical variations in comparison to its Euclidean alternatives, to directly apply to new geometry in a way that is not possible with Euclidean approaches, and to be quickly fine-tuned to a new geometry using a small amount of data. To our knowledge, this is the first report of a geometry-dependent non-Euclidean inverse imaging network. Our method is general for problems with spatiotemporal data living on graphs and linked with a linear imaging operator. Future studies will extend its application to other problems that fall into this category as well as incorporate more general physics. Furthermore, as observed in the results using data from experiments, there is still a performance gap between models based on simulation and measured data, even after fine-tuning. Given the challenges in obtaining labeled data in the types of application considered in this study (*e.g.*, measurement of whole-heart electrical potential), there is considerable motivation to investigate how to supervise the network with the governing physics in addition to data-driven losses.

Several aspects of the presented ST-GCNN could also be further investigated. One of the challenging factors lies in the hierarchical graph representations of the heart-torso mesh. One critical consideration in this problem is to prevent non-physical spatial propagation of signals, which requires the coarsened graph in hierarchical graph representations to preserve the topology of the geometry. Unlike down-sampling strategies on 2D Euclidean spaces, there is no established automatic method for down-sampling node features on realistic geometrical spaces. Existing graph pooling methods use the clustering method based on graph topological structure to coarsen the graph [28, 113]. However, we found significant structural information loss that, for instance, introduced holes on the right ventricle (RV) of the coarsened heart mesh. We also noticed that the activation did not always propagate over the surface of the coarsened heart. Therefore, we adopted a specialized mesh-coarsening approach from CGAL [18], to preserve the topological information of the coarsened heart and torso mesh. This mesh coarsening method also allowed us to control the down-sampling rate to prevent the unacceptable loss of structural information.

There are various choices of source models to represent the electrical activity of the heart in existing ECGI approaches, including heart surface potentials [53,63,79], or transmembrane potential defined on the volumetric mesh of the heart [27,40,41,42,119]. We based this study on the former because this is the most common formulation and the one used in commercial systems and also because surface-based methods are more straightforward to implement. Any further extension of the ST-GCNN to a more complete volumetric representation of the cardiac electrical sources will require appropriate hierarchical graph representations, which will certainly become more challenging. Furthermore, the size of the graph and thus the computational cost of training the ST-GCNN can also be expected to increase substantially.

There are many sources of geometric variation that we did not evaluate; in addition to variations between subjects, breathing can expand and contract the torso and alter the location of the heart relative to the torso; the heart geometry and position change every time the heart beats; regular human posture and activities may also cause slight transition or rotation of the heart. Considering all these scenarios of geometrical variation in any data-driven method could be challenging, given the volume and variation in the required geometric and signal information required for training. In this study, we simplified the geometrical variation to heart rotations, following the most common settings in the previous reports [41].

Chapter 5

Adaptation with Physics – Forward Imaging Physics

In this chapter, we propose how to leverage imperfect or inexact physics rules to guide learning in neural functions. We focused on the forward imaging physics in ECGI systems and used that to improve the learning of neural networks to solve the inverse problem. The proof-of-concept work has been published in MICCAI 2021 [60]. The complete study has been published in IEEE TMI 2024 [59].

5.1 Introduction

ECGI, in analogy to tomographic imaging, aims to computationally reconstruct electrical potential signals of the heart from non-invasive external observations (*i.e.*, body-surface ECGs) [119]. The forward *imaging physics* between the cardiac sources and their ECG measurements is relatively well-understood. Given heart and thorax geometrical meshes, the forward operator can be numerically constructed by solving the governing equations for *quasi-static* electromagnetic fields [9]. The inference/optimization of the inverse solutions given ECG measurements, however, is challenging due to the ill-posedness of the problem. A significant amount of effort in ECGI literature has focused on regularization techniques to leverage prior knowledge of cardiac electrical activity. Examples include spatial and/or temporal smoothness [16, 108] or sparsity [43, 124] of the solution, as well as *a priori* physiological knowledge about the spatiotemporal dynamics of the solution [42, 117, 119]. Many of these existing works can be represented as special cases within the general framework of state-space modeling (SSM) [132], where the prior knowledge can be expressed in a *dynamic transition function* that makes a prediction of the system state to constrain the inverse solution. Solving a *maximum a posteriori* (MAP) optimization, this prediction can then be corrected by the actual observations (*i.e.*, ECG data) to obtain an optimal reconstruction of the system state. Such a framework provides flexibility in explicitly incorporating various spatiotemporal knowledge about the system, such as complex partial differential equations (PDEs) describing the electrical dynamics of the cardiac system [119].

However, the traditional SSM framework faces several challenges. First, the reliance on strong structural and parameter assumptions in the state-space functions – although representing general knowledge about the systems – introduces potential errors when applied to individual subjects. Second, state-space Bayesian filtering is typically applied independently to each set of ECG observations to estimate the corresponding cardiac electrical activity, while the prior knowledge as described in the state-space functions is kept fixed. Even if information across multiple sets of ECG data from the same subject may indicate errors in the prior knowledge, they are not assimilated to refine the state-space functions for improving future estimations.

In recent years, traditional ECGI approaches have been increasingly replaced by data-driven machine-learning approaches that extract information from data without needing sophisticated prior knowledge or physics-based modeling [5, 41, 61]. Its blackbox model however lacks interpretability, and training requires large amounts of labeled data – in the form of electrical activity measured on the heart surface – that are typically not available in the real world. While it is clear that SSM frameworks naturally complement these limitations, direct integration of traditional SSM framework (with fixed state-space functions) with data-driven learning is challenging for ECGI and the attempt has been limited.

In this chapter, we propose a novel hybrid SSM framework for ECGI to leverage the advantage of state-space formulations in data-driven learning, with partially-learnable state-space functions and unsupervised Bayesian filtering strategies. We follow structured state-space modeling to leverage the physics-based forward emission function, such that the learning is supervised by the known physics free from the need of *labeled* data of cardiac electrical activity. In the meantime, instead of completely fixed state-space functions, we introduce neural modeling of the dynamic transition function (to predict the signal propagation in the heart) and the associated Bayesian filtering strategy. In an initial publication [60], we demonstrated the proof-of-concept of learning this hybrid SSM in an unsupervised fashion across subjects on *in-silico* data, followed by its fine-tuning on *in-vivo* data. In this work, we formalize two new scenarios to improve the learning of this hybrid

SSM: 1) Unsupervised learning where, when previous ECG observations from the same subject accumulate without *ground truth* recordings in the heart, the hybrid SSM utilizes these past ECG data to improve its future ECGI solutions on the same individual; and 2) Combined supervised and unsupervised learning where, when both *in-silico* simulation and clinical ECG observations are available, simulation data corresponding to different rhythms can be generated to augment the learning. In both scenarios, we develop the hybrid SSM to take ECG observations and reconstruct the epicardial and endocardial extracellular potentials.

To elaborate, in the scenario of unsupervised Bayesian filtering across different ECG observations on the same subject, the learning does not require any *a priori* data of cardiac electrical activity. Instead of relying on potentially incorrect prior assumptions in fixed SSM functions, we allow part of the state-space functions to be learnable to accumulate knowledge from past data of an individual subject. Compared to independently applying Bayesian filtering with a fixed SSM, we show that the learnable SSM can accumulate knowledge from unsupervised Bayesian filtering across a (small) number of ECG observations, and thus deliver improved ECGI results on new ECG observations. We demonstrated this via both *in-silico* and *in-vivo* experiments.

Furthermore, the neural formulation of the SSM allows it to be supervised by paired data of ECG observation and its corresponding cardiac electrical activity when such data are available. This allows the integration of supervised data-driven learning utilizing available data of cardiac electrical activity (*e.g.* via *in-silico* simulation), with unsupervised Bayesian filtering utilizing only ECG data (*e.g.* in typical clinical settings). In this scenario, the same learnable state space functions are supervised by ground truth of electrical cardiac data when available (*i.e.*, data-driven loss), along with supervision by ECG data via the physics-based forward operator (*i.e.*, physics-based loss). On *in-vivo* test data for human subjects, we demonstrated that this mixed-loss SSM allowed us to combine simulated data – either from the same subject or from other subjects – with real ECG data of the subject to improve ECGI performance over using either model alone.

5.2 Preliminary: Bayesian Filtering

The traditional SSM for ECGI consists of a prior dynamic transition function and an emission function:

$$\textit{Transition function: } \mathbf{X}_t = f(\mathbf{X}_{t-1}) + w_t, \tag{5.1}$$

$$\textit{Emission function}\text{: } \mathbf{Y}_t = \mathbf{H}\mathbf{X}_t + \nu_t \tag{5.2}$$

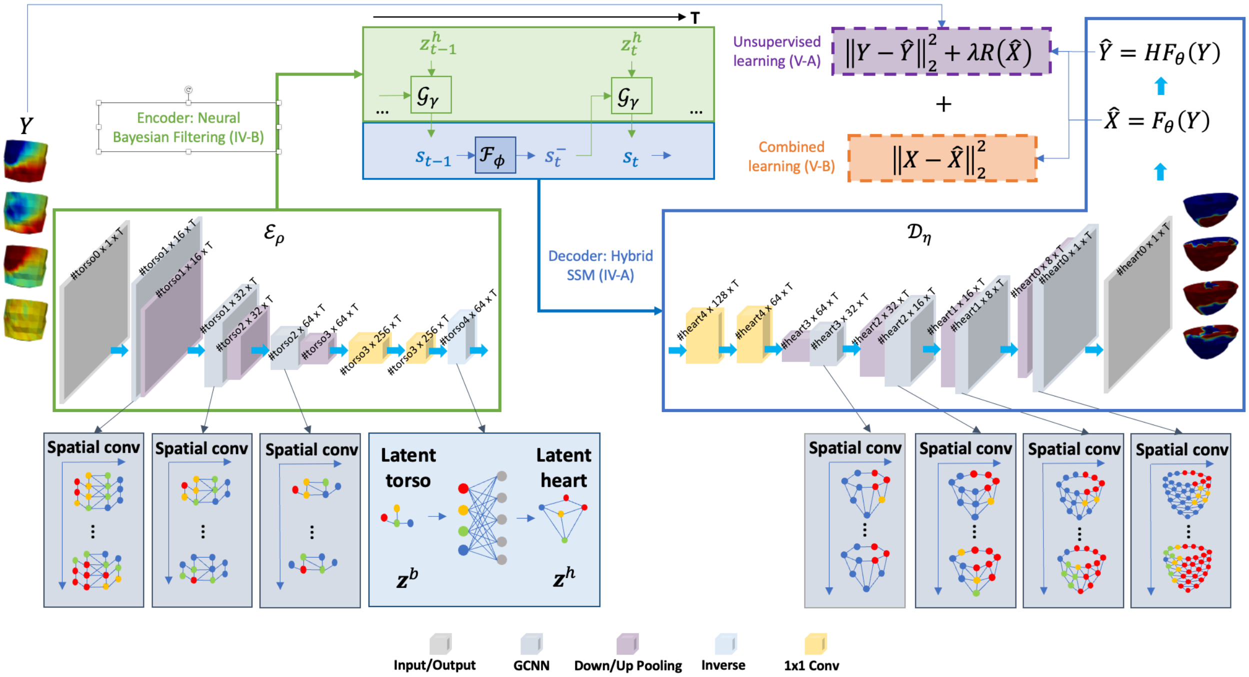


Figure 5.1: Overview of the presented network: ODE-GCN. The dashed box means the data-driven term in the mixed loss may not be always available.

where $\mathbf{X}_t$ is the heart signal and $\mathbf{Y}_t$ is the body surface potential at time t, and w_t and ν_t are noise variables introduced to account for modeling errors in these two functions. The transition function $f(\cdot)$ is typically designed to describe prior knowledge about $\mathbf{X}_t$ as discussed earlier. Based on this SSM, Bayesian filtering involves iterative estimations following a two-step procedure below:

$$\textit{Prior prediction:}\ \mathbf{X}_t^- = f(\mathbf{X}_{t-1}), \tag{5.3}$$

$$\textit{MAP estimation:}\ \hat{\mathbf{X}}_t = \mathbf{X}_t^- + \mathbf{K}_t(\mathbf{Y}_t - \mathbf{H}\mathbf{X}_t^-), \tag{5.4}$$

where the prediction in (5.3) leverages the prior assumption in the SSM transition function in (5.1), and the MAP estimation in (5.4) utilizes the Kalman gain $\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}^T(\mathbf{H}\mathbf{P}_t^-\mathbf{H}^T + \mathbf{R}_\nu)^{-1}$ ($\mathbf{P}_t^-$ is the covariance of $\mathbf{X}_t$) to optimally weigh the prior prediction with its discrepancy to observed $\mathbf{Y}_t$.

Typically the state-space functions are kept fixed when the above Bayesian filtering is applied to different observations of $\mathbf{Y}_{1:T}$. This means that the knowledge of the discrepancy between the prior assumption and the observations, while present in each of such observations, is not being utilized to refine the prior assumptions nor to improve future estimations.

5.3 Hybrid SSM for Neural Filtering

In this work, we present a novel hybrid SSM framework and realize it with an encoding-decoding network to make the state-space function and the associated Bayesian filtering strategy learnable. Specifically, the decoder will model the SSM in (5.1) and (5.2) as a generator network, and the encoder will learn the Bayesian filtering in (5.3) and (5.4) as an inference network.

5.3.1 Decoder: Hybrid SSM

The decoder (as illustrated in blue boxes in Fig. 5.1) models state-space functions (5.1) and (5.2) as:

$$\textit{Transition}: \begin{cases} \dfrac{d\mathbf{s}_t}{dt} = \mathcal{F}_\phi(\mathbf{s}_t), & \text{Temporal Transition}, \quad (5.5) \\ \mathbf{X}_t = \mathcal{D}_\eta(\mathbf{s}_t), & \text{Spatial Decoding}, \quad (5.6) \end{cases}$$

$$\textit{Emission}: \mathbf{Y}_t = \mathbf{H}\mathbf{X}_t, \tag{5.7}$$

where $\mathbf{s}_t \in \mathbb{R}^{V \times M}$ is the state of the latent dynamics at time t, V is the number of nodes ($\sim$60, depending on the anatomy of the subject) on the sparse heart mesh and M is the feature dimension (64 for all subjects) in the latent space. Here we introduce spatial-temporally disentangled neural modeling of the transition function in Equation (5.1). Specifically, we introduce neural functions $\mathcal{F}_\phi$ to model the temporal dynamics of cardiac electrical activity in a lower-dimensional latent space, and $\mathcal{D}_\eta$ to model the generation of cardiac electrical activity from this latent space. Both ϕ and η are learnable parameters. We preserve the physics-based emission function leveraging the forward operator $\mathbf{H}$ to supervise the learnable SSM with ECG data without labeled data of cardiac electrical activity. We term this a hybrid SSM.

Latent Dynamic Transition with Neural ODEs: We chose to model the latent dynamics of cardiac electrical activity with a neural ODE due to its ability to handle irregular data availability. Given that the temporal resolution of ECG varies, the ability to propagate forward indefinitely without requiring data allows the transition models' temporal resolution to be decoupled from that of the available ECG. In data settings with high density, both spatially and temporally, the NODE also has the advantage of an $O(1)$ memory cost in its forward solution with the usage of the adjoint method [111].

More specifically, as illustrated in Fig. 5.1, the dynamics of the latent variable $\mathbf{s}_t$ of $\mathbf{X}_t$ can be described by numerically integrating the neural ODE function $\mathcal{F}_\phi$ in Equation (5.5) as:

$$\mathbf{s}_t = \mathbf{s}_{t-1} + \int_{t-1}^{t} \mathcal{F}_\phi(\mathbf{s}_\tau) d\tau. \tag{5.8}$$

We implemented the ODE function as a stack of multi-layer perceptrons and activation functions. We then apply a fourth-order Runge-Kutta solver [17] to solve the ODE on $[t-1, t]$ to obtain the prediction of the current state $\mathbf{s}_t$.

Spatial Decoding via GCNNs: As heart signals live on a 3D non-Euclidean geometry, we describe the spatial decoding model from the learned latent space to the reconstruction $\mathbf{X}_t$ with a stack of GCNNs $\mathcal{D}_\eta$ with learnable parameters η.

As detailed in our previous work [61], we represent the triangular mesh of the heart as an undirected graph. Note that the number of vertices of the heart graph varies among different subjects. The spatial decoding is performed over hierarchical graph representations of the heart geometry obtained by a specialized mesh coarsening method [18] to preserve the topology of the geometry.

Following the previous work [60], we use the graph convolution with a continuous spline kernel across graphs [34], then introduce residual blocks to make the network deeper and more expressive.

Emission to ECG Data: The emission from $\mathbf{X}_t$ to $\mathbf{Y}_t$ employs the physics-based forward operator $\mathbf{H}$. This allows us to supervise the hybrid SSM with this physics operator and observed ECG data, rather than data of cardiac electrical activity that is rarely available in practice. Indeed the physics operator may have errors, which may degrade the performance of the hybrid-SSM compared to a neural network that is fully supervised by cardiac electrical activity data. However note that such data is hardly available in practice, thus the potential performance drop of the hybrid SSM will be in exchange for a significant advantage in terms of real-world applicability. We will compare the presented hybrid-SSM with fully-supervised baselines in the experiments.

5.3.2 Encoder: Neural Bayesian Filtering

We then use the encoder network (as illustrated in green boxes in Fig. 5.1) to learn Bayesian filtering:

$$\textit{Prediction function: } \mathbf{s}_t^- = \hat{\mathbf{s}}_{t-1} + \textstyle\int_{t-1}^{t} \mathcal{F}_\phi(\mathbf{s}_\tau) d\tau, \tag{5.9}$$

$$\textit{Estimation function: } \hat{\mathbf{s}}_t = \mathcal{G}_\gamma(\mathbf{z}_t^h, \mathbf{s}_t^-), \tag{5.10}$$

where $\mathbf{z}_t = \mathcal{E}_\rho(\mathbf{Y}_t)$, $\mathbf{z}_t \in \mathbb{R}^{V \times M}$ is the latent embedding of the observation $\mathbf{Y}_t$, V is the number of nodes (~60, depending on the anatomy of the subject) on the sparse heart mesh and M is the feature dimension (64 for all subjects) in the latent space, and the prediction of the latent state $\mathbf{s}_t^-$ is modeled as the solution to the neural ODE in Equation (5.5) given the previous state estimation $\hat{\mathbf{s}}_{t-1}$. The encoder consists two parts: the spatial embedding function $\mathcal{E}_\rho$ encodes $\mathbf{Y}_t$ to a latent variable $\mathbf{z}_t^h$, and the correction function $\mathcal{G}_\gamma$ combines $\mathbf{s}_t^-$ with the embedded $\mathbf{z}_t^h$ to get an optimal state estimation $\hat{\mathbf{s}}_t$.

GCNNs for Embedding Observed Data: As torso signals $\mathbf{Y}_t$ also live on 3D geometry, we follow a similar design in Section 5.3.1 to describe the spatial embedding function $\mathcal{E}_\rho$ with a stack of GCNNs with learnable parameters ρ. The geometry of the torso is also represented as an undirected graph, where edges and vertices are defined similarly as described in Section 5.3.1. For torso graphs on different subjects, the number of vertices is the same but the edge attributes are different. To learn the relationship between heart and torso, we follow [61] and assume the linearity to hold between $\mathbf{X}_t$ and $\mathbf{Y}_t$ in the latent space during inverse imaging. We construct a bipartite graph between the graph embedding of the heart and torso geometry: the edge attribute $\mathbf{u}(i, j)$ between torso vertex i and heart vertex j describes their relative geometrical relationship. Using spline convolution, we model the latent representation $\mathbf{z}^h(i)$ on vertex i of the latent heart mesh as a linear combination of latent representation $\mathbf{z}^b(j)$ across all vertices j of the latent torso mesh as:

$$\mathbf{z}^h(i) = \sum_j \mathbf{z}^b(j) \cdot \hat{\mathbf{h}}(\mathbf{u}(i, j)), \tag{5.11}$$

where $\hat{\mathbf{h}}$'s are learned as the spline convolution kernel.

Latent Dynamics Correction with GCN-GRU: The final estimation of the latent variable $\hat{\mathbf{s}}_t$ is given as a weighted combination of the latent representation $\mathbf{z}_t^h$ of the measurement $\mathbf{Y}_t$ on the heart, and the prediction of the latent dynamics $\mathbf{s}_t^-$. This is achieved by a Gated Recurrent Unit (GRU) cell [19] whose underlying architecture contains GCN layers as both the hidden state and input to the cell are graphs. We call it a *GCN-GRU* cell and denote it as $\mathcal{G}_\gamma$. Specifically, each linear operator in the regular GRU cell is replaced by GCN:

$$\begin{aligned}
\mathbf{r}_t &= \sigma\left(G_{z1}(\mathbf{z}_t^h) + G_{s1}(\mathbf{s}_t^-)\right), \\
\mathbf{g}_t &= \sigma\left(G_{z2}(\mathbf{z}_t^h) + G_{s2}(\mathbf{s}_t^-)\right), \\
\mathbf{n}_t &= \tanh\left(G_{z3}(\mathbf{z}_t^h) + \mathbf{r} \odot G_{s3}(\mathbf{s}_t^-)\right), \\
\hat{\mathbf{s}}_t &= (1 - \mathbf{g}_t) \odot \mathbf{n}_t + \mathbf{g}_t \odot \mathbf{s}_t^-,
\end{aligned} \tag{5.12}$$

where $\sigma(\cdot)$ is the sigmoid function, $\{G_{zi}, G_{si}\}_{i=1}^3$ are GCN operators, and $\gamma = \{z_i, s_i\}_{i=1}^3$ are learnable parameters.

5.4 Learning the Hybrid SSM

The presented encoder-decoder as described in Section 5.3 is denoted as $\boldsymbol{F}_\theta$, where θ are learnable parameters. We consider the following two scenarios in learning $\boldsymbol{F}_\theta$. First, we learn $\boldsymbol{F}_\theta$ without the supervision of ground truth cardiac signals $\mathbf{X}$, where the loss function (which we call physics-loss) between the input body surface potentials $\mathbf{Y}$ and the output of the emission model $\hat{\mathbf{Y}}$ provides an alternative supervision to the model. Second, we introduce mixed-loss by combining unsupervised learning on *in-vivo* real data when ground truth data $\mathbf{X}$ is not available, with data-driven learning on synthetic data when ground truth data $\mathbf{X}$ are available.

5.4.1 Unsupervised Learning of Hybrid SSM

Traditional methods seek to find a solution $\mathbf{X}_t$ that minimizes the fitting of $\mathbf{Y}_t$ while satisfying constraints $\mathcal{R}(\mathbf{X}_t)$:

$$\arg\min_{\mathbf{X}_t} \sum_{t=1}^{T} ||\mathbf{Y}_t - \mathbf{H}\mathbf{X}_t||_2^2 + \lambda \mathcal{R}(\mathbf{X}_t), \tag{5.13}$$

where the regularization parameter λ is often empirically tuned: it usually increases from an extremely small value until the error of reconstructing $\mathbf{Y}_t$ stops decreasing significantly. The proposed physics-loss (as illustrated in purple dashed boxes in Fig. 5.1) utilizes the physics-informed relationship in (5.13) to supervise the neural function $\hat{\mathbf{X}}_t = \boldsymbol{F}_\theta(\mathbf{Y}_t)$ of the hybrid SSM, where $\boldsymbol{F} = \{\mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{D}\}$ and $\theta = \{\rho, \phi, \gamma, \eta\}$. The optimization problem in (5.13) is then converted to learning parameters of $\boldsymbol{F}_\theta(\mathbf{Y}_t)$ for:

$$\arg\min_{\theta} \sum_{i=1}^{N} \sum_{t=1}^{T} ||\mathbf{Y}_t^i - \mathbf{H}\boldsymbol{F}_\theta(\mathbf{Y}_t^i)||_2^2 + \lambda \mathcal{R}(\boldsymbol{F}_\theta(\mathbf{Y}_t^i)), \tag{5.14}$$

where N is the number of data points. While different choices of $\mathcal{R}(\mathbf{X}_t)$ can be considered, as a proof of concept we adopt the popular Laplacian smoothing over the heart geometry, and λ is empirically tuned as in (5.13).

5.4.2 Combined Supervised and Unsupervised Learning

The neural formulation of the SSM allows it to be learned with supervision when partial data of cardiac electrical activity $\mathbf{X}_t$ are available. In this scenario, we propose mixed-loss (as illustrated in orange dashed boxes in Fig. 5.1) that combines the strength of both the physics loss in (5.14) and

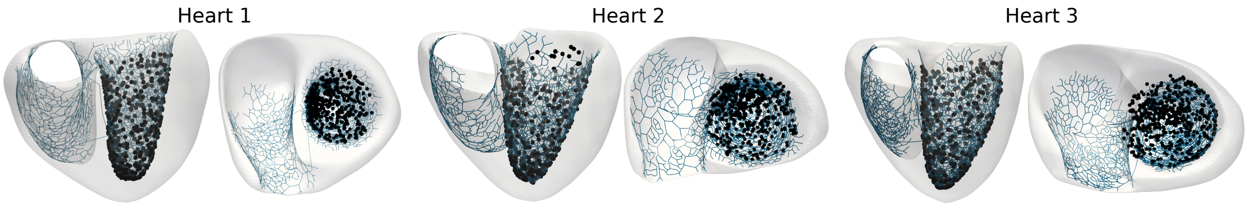


Figure 5.2: Overview of the sampled pacing locations (black) for each heart geometry.

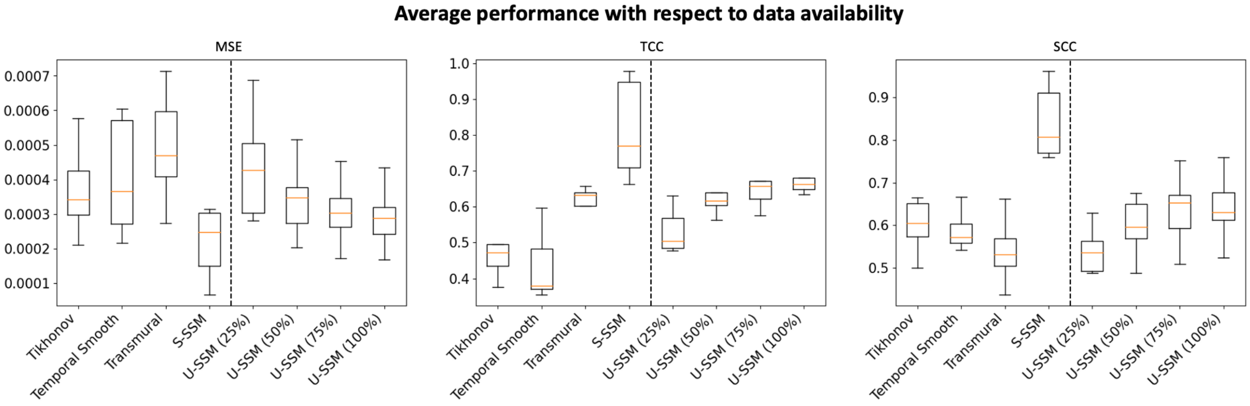


Figure 5.3: Performance of the proposed model with respect to different data availability in training (right side of the vertical dashed line) and the four baseline models (left side of the vertical dashed line).

data-driven learning. Specifically, the learnable state-space function is supervised by the *in-silico* data with ground truth $\mathbf{X}_t$ available, along with the same physic loss in (5.14) for *in-vivo* ECG data without $\mathbf{X}_t$. Additionally, we ask the physics loss to be satisfied on *in-silico* data as well. In summary, the mixed-loss can be represented as follows:

$$\begin{aligned} \arg\min_{\theta} \sum_{i=1}^{N} \sum_{t=1}^{T} ||\mathbf{Y}_t^i - \mathbf{H}\boldsymbol{F}_\theta(\mathbf{Y}_t^i)||_2^2 + \lambda\mathcal{R}(\boldsymbol{F}_\theta(\mathbf{Y}_t^i)) \\ + \mu^i||\mathbf{X}_t^i - \boldsymbol{F}_\theta(\mathbf{Y}_t^i)||_2^2, \end{aligned} \tag{5.15}$$

where μ^i is 0 when there is no available $\mathbf{X}^i$, and non-zero when $\mathbf{X}^i$ is available for sample $\mathbf{Y}^i$. μ^i is empirically determined by balancing the scale of the reconstruction error terms of $\mathbf{Y}^i$ and $\mathbf{X}^i$ in Equation (5.15) such that the measurements and the source have similar contribution to the supervision.

5.5 Experiments and Results on Unsupervised Learning of Hybrid SSM

In this set of experiments, we demonstrate that when the hybrid SSM is trained to do unsupervised Bayesian filtering on multiple data from the same subject, its performance on new data increases and improves over independent ECGI solutions on the same data. In this setting, the hybrid SSM does not need the supervision of $\mathbf{X}_t$. This represents a common use scenario that, as multiple ECG data from the same individual are collected over time, the hybrid SSM keeps refining the learned function of the underlying cardiac system for future use on the same subject. We tested this on both simulation and *in-vivo* data.

5.5.1 Models, Baselines, and Evaluation Metrics

Our presented hybrid SSM contains the following components: a spatial embedding network with three GCNN blocks, two standard convolutional layers, and a spline convolutional inverse mapping; a neural-ODE-based latent transition model with four linear layers; a correction model with a *GCN-GRU* cell; and a spatial decoder with two standard convolutional layers and four GCNN blocks, with 1,051,594 parameters in total. We used ELU activation [22] for most of the layers and the hyperbolic tangent function for the last layer of the neural function.

In this set of experiments, the model is supervised only by the ECG data through the forward operator via the loss function in Equation (5.14): in other words, the model is unsupervised (by the label $\mathbf{X}_t$). We call this unsupervised model U-SSM. The network parameter is optimized by Adam optimizer [68] with a learning rate of 5×10^{-4} and a learning rate scheduler decreasing the learning rate every 200 epochs with a decay rate of 0.5. We compared the performance of unsupervised SSM with classic ECGI approaches utilizing three different types of dynamic transition functions as temporal constraints: 1) no temporal constraints which correspond to second-order Tikhonov regularization [108], and 2) temporal smoothness constraint and 3) transmural constraint [33]. We also trained a data-driven version of the same hybrid SSM (we call this model S-SSM) where the hybrid SSM is supervised by available cardiac data $\mathbf{X}_t$ corresponding to the ECG data.

5.5.2 Simulation-Data Experiments

Data and Training

We considered three human heart-torso meshes, with 448, 475, 480 heart nodes and 120, 120, 120 torso nodes respectively. For each heart, we considered two different spatial configurations of scar tissues, resulting in total six *subject-specific* hearts for training and testing the hybrid SSMs. On each heart, we simulated realistic spatiotemporal propagation sequences of action-potentials by the Aliev-Panfilov model [1] considering different origins of activation. To obtain the extracellular potential on the heart surface from volumetric action potential, we obtained the forward operator by solving Possoin's equation using the coupled meshfree method and boundary element methods as described in [14, 75, 119]. The forward operator from the heart-surface extracellular potential $\mathbf{X}_t$ to body-surface potential $\mathbf{Y}_t$ was then obtained using the open-source SCIRun toolkit solving the Laplace equation using the boundary element method [54]. On the simulated body-surface potential, 30 dB Gaussian noises were added for reconstructing $\mathbf{X}_t$. Because the action potential simulated by the Aliev-Panfilov model was unitless in both amplitude and time, the generated signals on the heart and torso were also unitless. Specifically, in our experiments, we considered the depolarization process that was downsampled in time and represented by 65 discrete time steps. To generate disjoint training and testing sets for each subject, we partition the data based on the locations of activation origins. To examine the performance of the U-SSM as the number of training data increases, we randomly sampled the simulated data with an increasing number of origins (25, 50, 75, 100).

We also considered simulation data generated from a different pipeline on the three healthy subjects. For these simulations, each bi-ventricular mesh of targeted resolution 1200 µm was equipped with rule-based fibers [8] and a physiologically-detailed His-Purkinje system representing the cardiac conduction system [45]. Universal ventricular coordinates were computed for spatial navigation [7, 46]. On each heart, simulations of the trans-membrane voltages were run using a reaction-Eikonal model in monodomain formulation without diffusion using CARPentry [84, 118] considering 500 different origins of activation evenly sampled across the entirety of the endocardium (see Figure 5.2). Automated sampling of activation sites was facilitated using the universal ventricular coordinates in open-source *meshtool* [85]. Cellular dynamics within the heart were modeled using the Mitchell-Schaeffer ionic model with a set membrane voltage of $-86.2\,\mathrm{mV}$ and a plateau of $40\,\mathrm{mV}$ [82]. Heart conductivities were assigned according to [104] and the torso conductivity was assigned a value of $0.22\,\mathrm{S\,m^{-1}}$ [67]. Conduction velocities within the myocardium were assigned $0.6\,\mathrm{m\,s^{-1}}$ with an off-axis ratio of 4:2:1. A general conduction velocity of $2.0\,\mathrm{m\,s^{-1}}$ was prescribed within the His-

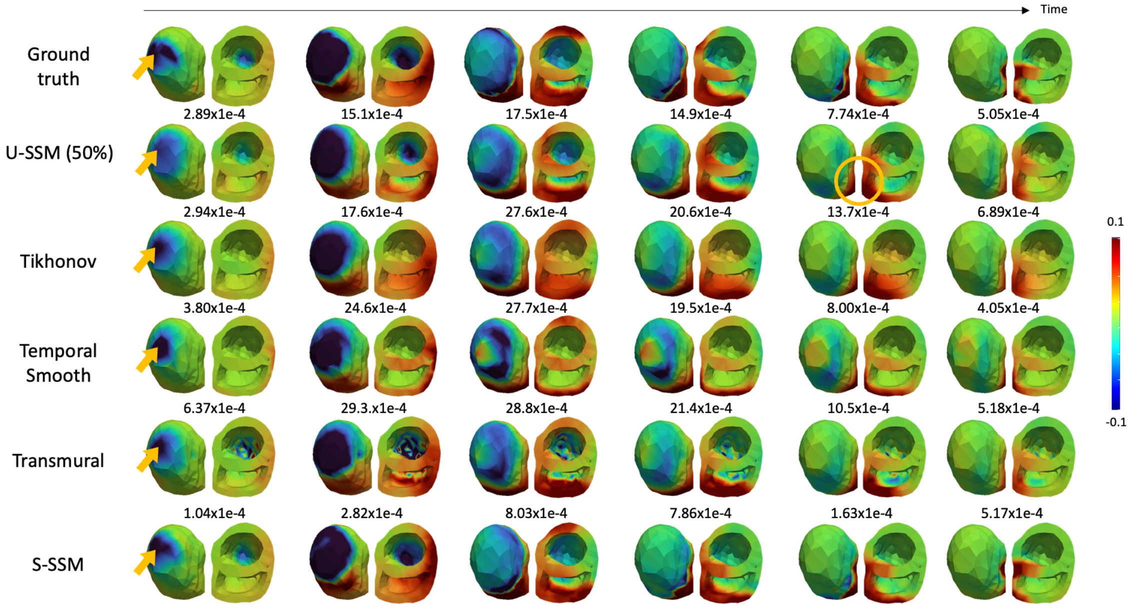


Figure 5.4: Visual examples of reconstructed electrical activity among four comparison models. Both epicardial and endocardial surfaces are presented. The arrow indicates the pacing site location. The color bar shows the scaled range of the signal since it is unitless in synthetic data. The U-SSM is trained on 50% of the full set. The MSE value is shown for each model at each timestep. The U-SSM showed better performance in predicting signal propagation and scar localization (circled) compared to the three ECGI baselines. The S-SSM demonstrated an upper bound in both signal propagation prediction and scar localization.

Purkinje System, with slight deviations assigned in the fascicular branches. An anterograde and retrograde delay of 8 ms and 3 ms, respectively, was assigned. The simulated transmembrane voltages were rescaled to $[0, 1]$, and we followed the similar steps mentioned above to obtain extracellular potentials and then 120-lead ECGs. Therefore, the generated signals on the heart and torso were all unitless. The sampling frequency was 1000 Hz. The temporal dimension was 600 discrete time steps and then downsampled to 200 discrete time steps. The three subjects have the spatial dimensions of 350, 350, 350 on the heart and 120, 120, 120 on the torso and we consider the depolarization process represented by 90 discrete time steps. For each subject, we randomly sampled the simulated data with an increasing number of origins (100, 200, 300, 400) to examine the performance of the U-SSM with an increasing number of training data.

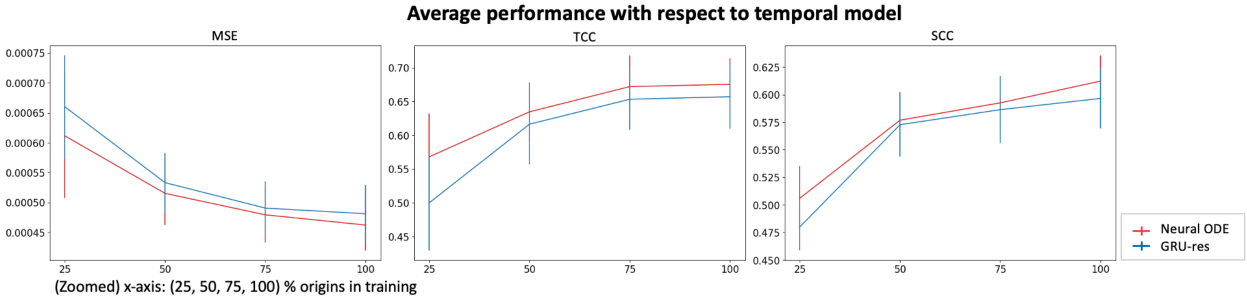


Figure 5.5: Summary of average performance on *in-silico* data among three comparison models with respect to different temporal models.

Evaluation Metrics

The accuracy of ECGI solutions in reconstructing $\mathbf{X}_t$ was measured by the mean square error (MSE), spatial correlation coefficient (SCC), and temporal correlation coefficient (TCC) between the reconstructed and actual potential sequence on the heart surface. While MSE measures the quantitative errors of the reconstructed signals on the heart, the SCC and TCC measure the correlation between the reconstructed and reference signals. We considered the correlation for both spatial signals at each time instance (SCC) and temporal signals at each location of the heart (TCC).

Results

Fig. 5.3 summarizes the quantitative metrics of the four models with respect to different numbers of training data. The reconstruction accuracy of the U-SSM surpassed the three ECGI baselines as the data availability increased. Note that the TCC of the U-SSM was better than the Tikhonov and temporal smooth ECGI baselines even when the training dataset is small among all subjects (25% training data). The SCC by the U-SSM outperformed all three ECGI baselines when there were over 50% training samples. The S-SSM set up a strong upper bound for the performance of the U-SSM, even at a small training size of 25. Note that the U-SSM's performance is approaching that of the supervised model as the number of unlabeled data increases. Fig. 5.4 provides visual examples demonstrating improved signal propagation patterns and scar locations obtained by the U-SSM. Notice that the U-SSM shows better reconstruction details of the signal propagation on the endocardial surface compared to all ECGI baseline models.

Alternative Architecture of Latent Transition Functions

As mentioned in Section 5.3.1, we chose the neural ODE as the latent dynamic transition function because of its generality as a continuous dynamic model, which has the advantage of decoupling the transition model's temporal resolution from that of the available ECG measurements, to accommodate irregularly sampled time-series data, and to handle missing data in time – the latter two scenarios, however, are not commonly expected in the setting of ECGI. Nevertheless, it is important to note that the presented hybrid SSM is agnostic to specific choices of function architectures or types for latent dynamic transitions. To demonstrate this, we investigated an alternative for temporal modeling: $\mathbf{s}_t^- = \hat{\mathbf{s}}_{t-1} + \Delta\mathbf{s}_t$, where $\Delta\mathbf{s}_t$ is given by GRU (denoted as GRU-res):

$$\begin{aligned}
\mathbf{s}_{t-1}^{(1)} &= \mathrm{ELU}(\boldsymbol{\alpha}_1\hat{\mathbf{s}}_{t-1} + \boldsymbol{\beta}_1), \quad \mathbf{g}_{t-1} = \sigma(\mathbf{W}_1\mathbf{s}_{t-1}^{(1)} + \mathbf{b}_1)\\
\mathbf{s}_{t-1}^{(2)} &= \mathrm{ELU}(\boldsymbol{\alpha}_2\hat{\mathbf{s}}_{t-1} + \boldsymbol{\beta}_2), \quad \mathbf{h}_{t-1} = \mathrm{ELU}(\mathbf{W}_2\mathbf{s}_{t-1}^{(2)} + \mathbf{b}_2)\\
\tilde{\mathbf{s}}_{t-1} &= \boldsymbol{\alpha}_3\hat{\mathbf{s}}_{t-1} + \boldsymbol{\beta}_3,\\
\Delta\mathbf{s}_t &= (1 - \mathbf{g}_{t-1}) \odot (\mathbf{W}_3\tilde{\mathbf{s}}_{t-1} + \mathbf{b}_3) + \mathbf{g}_{t-1} \odot \mathbf{h}_{t-1},
\end{aligned}$$

where $\boldsymbol{\theta}_z = \{\mathbf{W}_i, \mathbf{b}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i\}_{i=1}^3$ are learnable parameters of the dynamic function [58]. We compared GRU-res with the proposed method (denoted as Neural ODE) on a subset of synthetic data in Section 5.5.2 with changing of origin. Fig. 5.5 summarizes the quantitative results of the two temporal models with respect to different numbers of training data. The reconstruction accuracy of GRU-res is compatible with Neural ODE and has a similar trend with increasing training data. This proves that hybrid SSM was not affected by the choice of latent dynamic models.

5.5.3 Real-Data Experiments

Data and Training

We then repeated the evaluation and comparison experiments on *in-vivo* 120-lead ECG data from three subjects with premature ventricular contraction (PVC) on structurally healthy hearts [33] (termed as Healthy 1, Healthy 2, and Healthy 3), which have been made available at EDGAR [3], and three post-infarction subjects [3, 110] (termed as Post 1, Post 2, and Post 3), all undergoing ventricular pacing. The healthy subjects have the same geometry as the geometry of the three healthy subjects used in Section 5.5.2. The body surface potentials of the three healthy subjects are sampled at 2000 Hz and further downsampled using polyphase filtering to 201 steps, with the depolarization represented in 90 steps used in experiments. The heart-torso geometry of the three

Table 5.1: Summary of localization errors for the origin of ventricular activation in each subject. (Unit: mm. Paired p-test is performed between ECGI Tikhonov and other comparison models. nss: non-subject-specific)

Type	Model	Post 1	Post 2	Post 3	Healthy 1	Healthy 2	Healthy 3	Overall	p-value
Non-ML ECGI	Tikhonov	33.87(17.81)	51.60(12.07)	39.09(39.05)	22.49(20.78)	44.69(26.60)	49.82(15.01)	40.26(22.69)	/
	Transmural	30.63(17.50)	38.00(16.40)	14.19(11.46)	25.56(24.86)	30.75(13.68)	64.65(5.45)	33.97(20.79)	0.321
Supervised Baselines	S-SSM	41.33(20.35)	45.39(16.98)	50.70(6.41)	35.82(7.18)	38.15(19.80)	39.16(23.51)	44.00(12.22)	0.481
	ST-GCNN	34.05(18.65)	62.41(22.31)	65.90(23.50)	46.11(18.76)	30.59(11.87)	55.67(13.47)	49.12(20.81)	0.165
	Euclidean	46.83(28.39)	56.75(16.04)	53.10(20.44)	39.82(15.28)	49.43(8.21)	47.47(9.74)	48.90(15.26)	0.129
	MARS	39.69(21.79)	43.40(47.43)	37.37(23.49)	29.97(24.13)	32.30(15.02)	48.11(26.05)	38.48(25.04)	0.797
U-SSM	25%	39.68(19.82)	48.91(19.26)	32.11(16.23)	28.32(17.01)	33.47(11.98)	42.59(18.17)	37.51(15.69)	0.628
	50%	37.42(20.02)	44.82(27.66)	26.09(8.63)	25.96(21.27)	32.58(11.74)	41.42(21.11)	34.71(17.88)	0.352
	100%	35.12(19.55)	39.70(21.47)	26.82(27.66)	24.85(19.38)	33.16(13.90)	38.41(17.73)	33.01(18.18)	0.228
M-SSM	25%	31.41(16.61)	32.50(23.76)	36.75(12.79)	18.14(5.02)	31.88(13.29)	28.19(12.76)	29.81(13.83)	0.060
	50%	24.45(17.76)	30.02(16.04)	35.22(14.08)	15.97(6.84)	29.55(15.26)	25.44(13.42)	26.77(13.81)	0.017
	100%	**23.23(13.96)**	**27.20(16.32)**	**25.34(7.65)**	**13.16(1.57)**	29.29(13.39)	**25.28(9.85)**	**23.92(11.03)**	0.003
	nss	**26.66(14.93)**	**24.79(7.98)**	24.55(11.13)	**24.52(20.02)**	**33.35(13.79)**	**27.34(12.58)**	26.87(12.09)	0.014
F-SSM	25%	36.48(22.22)	39.06(18.94)	42.54(15.13)	25.09(4.35)	30.58(12.14)	35.59(17.31)	34.89(14.51)	0.334
	50%	43.10(27.05)	32.28(18.23)	40.64(19.63)	23.76(4.44)	29.94(12.63)	35.68(12.29)	34.23(15.60)	0.289
	100%	31.51(14.89)	39.29(20.28)	37.14(18.74)	22.00(3.04)	29.48(11.62)	28.25(10.64)	31.28(13.09)	0.100
	nss	32.31(17.08)	44.40(3.76)	**21.61(5.10)**	26.66(19.84)	34.51(11.72)	30.99(11.48)	31.75(12.56)	0.115

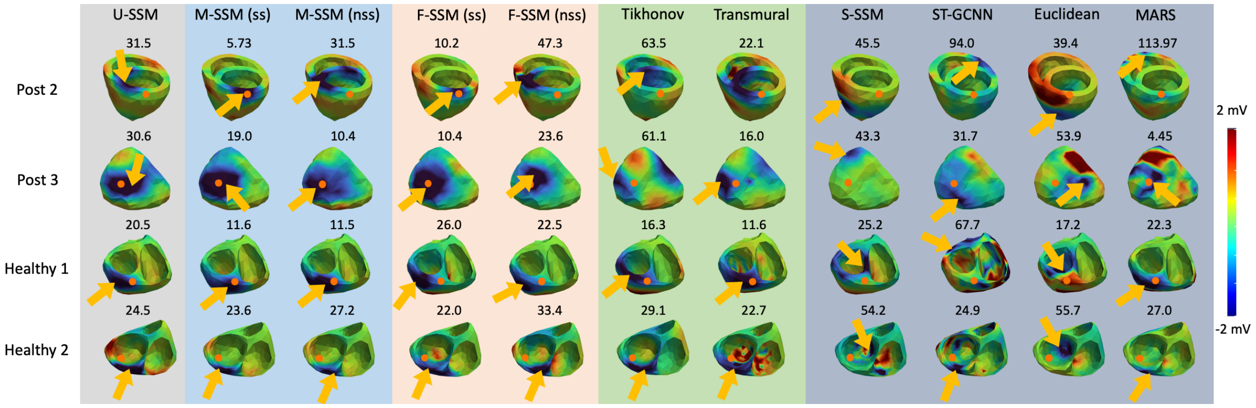


Figure 5.6: Visual examples of reconstructed electrical activations on *in-vivo* data (Unit: mm; ss: subject-specific; nss: non-subject-specific). The arrows are predicted pacing sites and the dots highlight actual pacing sites. The localization error is shown for each model. The U-SSM and both subject-specific and non-subject-specific M-SSM showed generally better accuracy in early activation localization among all comparison models.

subjects with post-infarction is the same as the geometry of the three subjects with scar tissues used in Section 5.5.2. The body surface potentials of the three subjects with post-infarction are sampled at 1000 Hz and further downsampled using polyphase filtering to 101 steps, with the depolarization represented in 65 steps used in experiments. For each subject, patient-specific heart torso geometry was extracted from computed-tomography images, on which extracellular potential signals were reconstructed from ECG data. Each healthy subject contains ECG data acquired from

21, 14, and 17 different origins of pacing, respectively. Each post-infarct subject contains ECG data acquired from 22, 20, and 23 different origins of pacing respectively. For each subject, we randomly left out four sites for testing. Because the S-SSM cannot be directly trained on *in-vivo* data due to lack of $\mathbf{X}_t$, we considered directly applying the S-SSM trained on the complete *in-silico* dataset in Section 5.5.2 to *in-vivo* data. Similarly, we also evaluated existing data-driven methods, such as ST-GCNN [61], Euclidean method [41], and MARS [89], by training them on the same *in-silico* dataset and testing on *in-vivo* data. We also considered ECGI with the second-order Tikhonov regularization and transmural regularization in comparison. To examine the performance of the U-SSM given different training data availability, we randomly selected 50% and 25% of the training data on each subject. All results below are reported on the same four test cases each on the six subjects totaling 24 cases.

Evaluation Metrics

Due to the lack of *in-vivo* measurement of heart-surface potential, quantitative accuracy was measured by the Euclidean distance between the reconstructed origins of the activation and the known sites of pacing. To localize the site of the earliest activation by the steepest descent of the reconstructed extracellular potential during depolarization in itself presents non-trivial challenges [24]. We consider 5 mm as the best case target in clinical settings, since it is the diameter of the ablation catheter, and thus the best ablation resolution. We thus resorted to manual identification of the region of the earliest activation by visual inspection of the reconstruction electrograms sequence, and the centroid of the identified region was calculated as the site of the earliest activation.

Results

Table 5.1 summarizes the localization errors for the early activation site with respect to different amounts of training data. Directly applying supervised data-driven models trained on simulation data to *in-vivo* data (supervised baselines) had a worse overall localization error (S-SSM: 44.00 ± 12.22, ST-GCNN: 49.12 ± 20.81, Euclidean: 48.90 ± 15.26, MARS: 38.48 ± 25.04) than the Tikhonov (40.26 ± 22.69) and transmural-based regularization (33.97 ± 20.79) ECGI baselines, highlighting the challenges of generalization when training an ECGI reconstruction network supervised by simulation data. When using 25% of the full training set, the overall localization accuracy of the U-SSM (37.51 ± 15.69) was comparable with the Tikhonov method (40.26 ± 22.69) and less accurate than transmural-based regularization (33.97 ± 20.79). The accuracy of U-SSM surpassed the Tikhonov

ECGI baseline when the training data increased (25%: 37.51 ± 15.69, 50%: 34.71 ± 17.88, 100%: 33.01 ± 18.18). When using the full training data, the accuracy of U-SSM became comparable with the transmural-based ECGI baseline. Fig. 5.6 provides visual examples.

This set of experiments demonstrates that the presented U-SSM can learn to perform unsupervised Bayesian filtering on individual hearts, with a reconstruction accuracy higher than the supervised ECGI networks trained on simulation data and traditional ECGI methods with a similar regularization choice, and on par with traditional ECGI methods using more advanced regularization options.

5.6 Experiments and Results on Combined Learning of Hybrid SSM

In this set of experiments, we demonstrate the use scenario for the hybrid SSM to leverage available simulation data – either from different subjects or from the same subject – and combine it with *in-vivo* ECG data available on a subject, to improve ECGI performance than using either data alone.

5.6.1 Models, Baselines, and Evaluation Metrics

We considered two specific experimental settings. For each subject on whom to test the hybrid SSM, we combined unsupervised training using *in-vivo* ECG data with supervised training on *in-silico* data 1) generated on this specific subject, versus 2) generated from two different subjects. This helps us test if we need to generate subject-specific simulation data for a subject in order to use this mixed-loss SSM (we call this model M-SSM).

For the three healthy subjects and the three post-infarction subjects, the simulation and *in-vivo* data are as described in Sections 5.5.2 and 5.5.3. For each healthy subject, we chose the same 400 origins in training as in Section 5.5.2, and the same selection of four pacing sites for testing and the rest for training on *in-vivo* data as in Section 5.5.3. For each post-infarction subject, we chose one configuration of scar tissue with the same 100 origins in training as in Section 5.5.2, and the same selection of four pacing sites for testing and the rest for training on *in-vivo* data as in Section 5.5.3. All results below are reported on the same *in-vivo* test cases each on the six subjects totaling 24 cases.

Similar to Section 5.5.3, we compared the M-SSM with: 1) ECGI with second-order Tikhonov regularization and transmural-based regularization, 2) existing supervised data-driven models (S-SSM, ST-GCNN, Euclidean, and MARS) by training on *in-silico* data and testing on *in-vivo* data, and 3) unsupervised fine-tuning of the same supervised SSM on *in-vivo* data using the physics-based loss in (5.14) (we call this F-SSM). Similar to Section 5.5.3, we also examine the performance of the M-SSM concerning the change in data availability of *in-vivo* data by randomly selecting 50% and 25% of *in-vivo* data in training on each subject. We continued to use the Euclidean distance of pacing site localization similar to that in Section 5.5.3.

5.6.2 Combining with Subject-Specific Simulation Data

Table 5.1 shows localization errors of early activation sites among the comparison models. The M-SSM demonstrated the best overall localization accuracy (23.92 ± 11.03) among all comparison models. Note that directly applying supervised data-driven models trained on simulation data to *in-vivo* data showed worse overall performance (S-SSM: 44.00 ± 12.22, ST-GCNN: 49.12 ± 20.81, Euclidean: 48.90 ± 15.26, MARS: 38.48 ± 25.04) than both M-SSM and F-SSM (31.28 ± 13.09). When using 25% of the full training set, the overall performance of the M-SSM (29.81 ± 13.83) was better than the Tikhonov baseline (40.26 ± 22.69) and the transmural-based regularization baseline (33.97 ± 20.79). In Post 1, 2, 3, and Healthy 1, 3, the performance of the M-SSM using 25% of the full training set (Post 1: 31.41 ± 16.61, Post 2: 32.50 ± 23.76, Post 3: 36.75 ± 12.79, Healthy 1: 18.14 ± 5.02, Healthy 3: 28.19 ± 12.76) was even better than the F-SSM using the full training set (Post 1: 31.51 ± 14.89, Post 2: 39.29 ± 20.28, Post 3: 37.14 ± 18.74, Healthy 1: 22.00 ± 3.04, Healthy 3: 28.25 ± 10.64). The overall localization accuracy of both M-SSM (25%: 29.81 ± 13.83, 50%: 26.77 ± 13.81, 100%: 23.92 ± 11.03) and F-SSM (25%: 34.89 ± 14.51, 50%: 34.23 ± 15.60, 100%: 31.28 ± 13.09) showed an increasing trend with the increase in data availability. Fig. 5.6 provides visual examples of reconstructions of early activation.

5.6.3 Combining with Non-Subject-Specific Simulation data

As summarized in Table 5.1, the non-subject-specific M-SSM showed a better localization accuracy (26.87 ± 12.09) in early activation sites in comparison with the Tikhonov baseline (40.26 ± 22.69) and transmural-based regularization baseline (33.97 ± 20.79), even though there is a small reduction of performance compared to the subject-specific M-SSM (23.92 ± 11.03). The overall performance of the non-subject-specific M-SSM in Post 1, 2, and all Healthy subjects (Post 1: 26.66 ± 14.93, Post 2: 24.79 ± 7.98, Healthy 1: 24.52 ± 20.02, Healthy 2: 33.35 ± 13.79, Healthy 3: 26.87 ± 12.09) was better than F-SSM (Post 1: 32.31 ± 17.08, Post 2: 44.40 ± 3.76, Healthy 1: 26.66 ± 19.84, Healthy 2: 34.51 ± 11.72, Healthy 3: 31.75 ± 12.56), while the performance of F-SSM was better in Post 3 (21.61 ± 5.10) than M-SSM (24.55 ± 11.13). Visual examples in Fig. 5.6 show that the non-subject-specific M-SSM also improved the early activation localization.

This set of experiments demonstrated that the hybrid SSM allows us to combine supervised training using simulation data and unsupervised training using *in-vivo* data, to obtain improved performance over using either data alone. It can utilize non-subject-specific simulation data which reduces the burden of having to conduct subject-specific *in-silico* simulations.

5.7 Conclusions and Discussion

We have presented a novel hybrid SSM framework for ECGI to leverage the advantage of state-space formulations in data-driven learning, with partially learnable state-space functions and unsupervised Bayesian filtering strategies. We demonstrated its improved ECGI performance under unsupervised Bayesian filtering on both *in-silico* and *in-vivo* experiments and under mixed-loss SSM setting on *in-vivo* data combined with simulation data either from the same subject or from other subjects. Our method is general for problems with spatiotemporal data with disentangled dynamic transition and spatial emission. Future studies will extend its application to other problems that fall into this category.

Several aspects of the presented hybrid SSM framework can be further investigated. One of the challenging factors exists in the subject-specific nature of Bayesian filtering, in which a critical component is that it accumulates the information of each subject. One may naively consider learning the knowledge shared across subjects by training the model on different subjects together. However, we found the improvement in reconstruction was marginal. How to model diverse dynamics across subjects with few-shot data remains an open question.

There are various choices of source models to represent the cardiac electrical activity in existing ECGI approaches, including heart surface potentials [53,63,79], or transmembrane voltages defined on the volumetric mesh of the heart [27,40,41,42,119]. Our study is based on the former because this is the most common formulation and the one used in commercial systems. Further extension of the hybrid SSM to the volumetric representation of the cardiac electrical activity will bring challenges in building appropriate hierarchical graph representations for spatial decoding. Furthermore, the substantial increase of the size of the graph can also lead to a higher computational cost of training the hybrid SSM.

This work uses Laplacian smoothing as the regularization in the objective function because this is one of the most popular choices and the Laplacian operator can be directly derived given the heart geometry. When using the Laplacian operator, the reconstruction accuracy of hybrid SSM was lower compared to traditional methods using more advanced regularization with small training data and increased to be comparable with the transmural-based method when more data were available. However, many alternative constraints on cardiac electrical activities such as spatial sparsity [43,124] and PDEs [117,119], have not been investigated. Future works will seek to better incorporate physiological knowledge into the hybrid SSM by tackling challenges in the modeling of spatial and temporal gradients to make physics-informed neural networks.

The two major scenarios of hybrid SSM are 1) unsupervised learning of hybrid SSM when previous ECG observations from the same subject accumulate, and 2) combined supervised and unsupervised learning when both in-silico simulation and clinical ECG observations are available. In both scenarios, the hybrid SSM is unsupervised on the real data, meaning that all it requires is past ECG data from the same subjects. Because no label (actual knowledge about the underlying conditions) is required, in theory, ECG observations under any arrhythmic conditions can be used such as sinus rhythm, pacing, or various arrhythmic conditions. In our experiments, we considered pacing rhythms primarily because of the availability of data. In future works, we will investigate the general applicability of the hybrid SSM to various ECG conditions owing to its unsupervised nature.

We considered the localization of the pacing site by manual identification of the earliest activation region as the quantitative evaluation of the proposed method on *in-vivo* data. The ECGI validation remains an open question as the challenges come from the choice of the implementation methods, the unavailability of *in-vivo* data, and variations in clinical application interest [23]. As one of the typical local comparison metrics, the estimation of activation time and site from reconstructed electrograms itself is a difficult problem that is investigated among many groups [31,32]. Future works will investigate more generalized and automated localization methods for pacing and alternative metrics based on the corresponding arrhythmic conditions.

A complete heart signal has two main stages: depolarization and repolarization, in which using ECGI to detect the time of electrical activation and recovery of local cardiac tissue from transmembrane voltages or extracellular potentials remains an open question. Our study only considered the depolarization stage in all experiments, given the limitation of the neural network in describing complex dynamics. Future works will explore more advanced neural networks to describe the exact physical processes.

Ideally, the performance of hybrid SSM should keep improving with the increase of available training data. However, we noticed a common phenomenon in our experiments that improvement of performance slowed down when the number of training data was above a certain amount. Future work on the hybrid SSM will investigate whether the hybrid SSM is sufficiently expressive for the data governed by physics equations.

Chapter 6

Identifiable Hybrid Model

In this chapter, we focused on learning to identify hybrid knowledge with both the physics-based model and neural components.

6.1 Introduction

Personalized virtual heart models, such as those describing the electrophysiological (EP) process of the heart, have shown significant progress in risk stratification [4], treatment planning [130], and outcome predictions [112]. Effective personalization of virtual heart models, especially estimation of model parameters pertaining to patient-specific tissue properties, however, remains a critical challenge due to the ill-posed nature of the inverse problem, the myriad of modeling assumptions involved, and the computational cost associated with these models.

Many efforts have been devoted to personalizing the parameters of virtual EP models of the heart. Earlier works have focused on iterative optimization/inference to minimize the discrepancy between model outputs and measured data [112, 122]. Despite significant progress, the iterative nature of these approaches involving multiple runs of the EP model makes it less appealing for clinical use. More importantly, they attribute discrepancies between model outputs and data observations solely to the model parameters being optimized, essentially assuming the absence of structural or other errors in the model. This overlooks unknown errors within a virtual EP model associated with its structural assumptions and simplifications, which may in turn lead to erroneous parameter identification if this unknown error is non-negligible. We refer to this as *white-box* models.

Recent machine learning (ML) and deep learning (DL) advances have brought success in the personalization of virtual heart models. Examples include learning the input-output relationship between the parameters and outputs of the EP model [65], or the recent meta-learning approach to learn to adapt a neural network as a surrogate for the EP model [57]. These data-driven approaches bypass the underlying physical principles governing the cardiac EP process, but instead heavily rely on the availability of large datasets on the input-output relationship being learned. Since these data (*e.g.,* tissue property as model input, or spatiotemporal activity of action potential as model output) are not always available in *in-vivo* settings, most data-driven approaches resort to simulated data for supervision. As a result, its training involves computationally prohibitive data generation, while its deployment to real data faces challenges of generalization. We refer to this as *black-box* models which are further limited in interpretability.

To bridge the gap between white-box and black-box modeling, recent works have proposed the use of physics-informed neural networks (PINNs) in personalized virtual heart models [50]. In PINNs, the neural network's output is governed by a partial differential equation (PDE) with known mathematical expressions, representing prior knowledge and removing the need for data supervision; the parameter of the PDE can be optimized at the same time as the PINN is being trained, achieving a personalized PINN and PDE at the same time. However, although *informed* by a given PDE in the training loss, the PINN is still a black-box function; moreover, the constraining PDE assumes exact and perfect mathematical descriptions of the underlying system of interest: this *parallel* integration of white- and black-box modeling thus, unfortunately, inherits their respective limitations: the PINN has limited interpretability, while its personalization may suffer if the constraining white-box models are imperfect. Additionally, the joint optimization of the PINN and PDE parameters has to be performed for each individual subject, limiting its clinical applicability.

To overcome the above challenges, we propose a novel hybrid modeling approach towards personalized virtual heart models, replacing existing *white-box* or *black-box* modeling approaches with a *gray-box* approach. Unlike existing *physics-informed* gray-box models, we further move towards a *physics-integrated* gray-box model that explicitly hybridize physiological models with neural network models inside the digital twin. While the concept of hybrid models has emerged in various domains including virtual heart models [49, 66, 98, 125], a critical roadblock is the assumption of direct supervision on the variables being modeled, which is not applicable in virtual heart models where the modeled variable (*e.g.,* spatiotemporal propagation of action potentials in the heart) is often only partially or indirectly observed. To this end, we further address the challenge of *unsupervised* identification of hybrid models by a novel meta-learning strategy to separately identify the parameters of the physiological model and its gaps to observed data. At training time, the proposed method for meta-learning hybrid personalized cardiac EP (HyPer-EP) models does not require the ground truth of the variables being modeled, leveraging prior physiology while learning to identify its gap to observed data. At test time, HyPer-EP enables the personalization of a hybrid cardiac digital twin – composed of an interpretable physiological component and a neural component accounting for its errors – using subject-specific data via efficient feedforward computations. We demonstrate the feasibility and generality of HyPer-EP with two examples of instantiations, providing evidence for their feasibility and benefits over physics-based or neural modeling alone in synthetic experiments.

6.2 Problem Formulation

Consider the goal of obtaining a personalized model $\mathcal{M}(\theta)$ describing the EP process of the ventricles in the form of the spatiotemporal propagation of action potentials $\mathbf{X}_{0:T}$, with patient-specific parameter θ and observations $\mathbf{Y}_{0:T} = g(\mathbf{X}_{0:T})$.

In a white-box approach, $\mathcal{M}$ is a known mathematical expression $\mathcal{M}_{\text{PHY}}$ and, when given measurements $\mathbf{Y}_{obs}$, the value of θ is optimized to fit the output of $\mathcal{M}_{\text{PHY}}$ to $\mathbf{Y}_{obs}$ via data-fitting metrics such as mean-squared-errors (MSE):

$$\hat{\theta} = \arg\min_{\theta} ||g(\mathcal{M}_{\text{PHY}}(\theta)) - \mathbf{Y}_{obs}||_2^2 \tag{6.1}$$

where all potential errors in $\mathcal{M}_{\text{PHY}}$ due to model assumption and simplifications are neglected, and its difference from $\mathbf{Y}_{obs}$ is solely attributed to parameter θ.

In a black-box approach, $\mathcal{M}$ is often a deep neural network (DNN) $\mathcal{M}_{\phi}$ and its weight parameters ϕ is typically learned given a large number of paired data $\{\theta^i, \mathbf{X}^i_{0:T}\}_{i=1}^N$ in a supervised loss using, for instance, MSE:

$$\hat{\phi} = \arg\min_{\phi} \sum_{i=1}^{N} ||\mathcal{M}_{\phi}(\theta^i) - \mathbf{X}^i_{0:T}||_2^2 \tag{6.2}$$

where $\{\theta^i, \mathbf{X}^i_{0:T}\}_{i=1}^N$ are often obtained via simulation data because they are not readily available in practice, raising challenges of generalization to real data.

In the recently-emerged PINN approach, a DNN $\mathcal{M}_{\phi}$ with weight parameters ϕ, is supervised by a partial differential equation (PDE) $\mathcal{M}_{\text{PHY}}$ with known mathematical expressions and potentially unknown parameter θ. Given the available data on $\mathbf{x}_{0:T}$, both ϕ and θ of the two models can be simultaneously optimized:

$$\{\hat{\phi}, \hat{\theta}\} = \arg\min_{\phi,\theta}\{||\mathcal{M}_\phi - \mathbf{X}_{0:T}||_2^2 + \lambda||\mathcal{M}_{\mathrm{PHY}}(\mathcal{M}_\phi;\theta)||_2^2\} \tag{6.3}$$

where the first term fits $\mathcal{M}_\phi$'s output to available data (data-fitting loss), and the second term encourages $\mathcal{M}_\phi$'s output to follow the governing PDE specified by $\mathcal{M}_{\mathrm{PHY}}$ (PDE residual loss). With this *parallel* integration, $\mathcal{M}_\phi$ is still a black box while $\mathcal{M}_{\mathrm{PHY}}$ is assumed to represent exact knowledge of the system. Furthermore, because θ is unique to each individual, this joint optimization of ϕ and θ must be repeated for each given set of observations $\mathbf{x}_{0:T}$.

6.3 Methodology

In this work, we propose a novel Hybrid Personalized (HyPer) modeling framework to address the limitations of purely white-box or black-box models while marrying their respective strengths. Unlike the parallel integration in current *physics-informed* approaches as described in Section 6.2, HyPer is underpinned by a *physics-integrated* hybrid model $\mathcal{M}_{\mathrm{Hybrid}}$ consisting of known mathematical expression $\mathcal{M}_{\mathrm{PHY}}$ augmented by an unknown neural component $\mathcal{M}_\phi$ to account for its potential gap to reality, each parameterized by learnable parameters. This hybrid model is then situated within the latent space of an encoding-decoding architecture to bridge the variables being modeled to their indirect observations in the data space, enabling a novel *unsupervised* learning paradigm with a learn-to-identify meta-learning formulation to address the identifiability issue associated with the separate identification of $\mathcal{M}_{\mathrm{PHY}}$ and $\mathcal{M}_\phi$ in the hybrid model. This hybrid generative modeling and the learn-to-identify inference strategy constitute the backbone of HyPer, which we elaborate on below in the context of cardiac EP models (referred to as HyPer-EP).

6.3.1 Hybrid Modeling of Cardiac EP Process

The proposed hybrid model $\mathcal{M}_{\mathrm{Hybrid}}$ is a combination of known mathematical expression $\mathcal{M}_{\mathrm{PHY}}$ and an unknown neural function $\mathcal{M}_\phi$, where the latter is intended to capture potentially unmodeled complexities or errors inherent in the simplified white-box representation $\mathcal{M}_{\mathrm{PHY}}$:

$$\mathcal{M}_{\mathrm{Hybrid}} = \mathcal{M}_{\mathrm{PHY}} + \mathcal{M}_\phi \tag{6.4}$$

Note that Equation (6.4) denotes a general framework where the hybridization of $\mathcal{M}_{\mathrm{Hybrid}}$ and $\mathcal{M}_{\mathrm{PHY}}$ can be realized in various strategies.

In this chapter, we use HyPer-EP to bridge simple and data-generating physics. We consider the single-variable Eikonal model as $\mathcal{M}_{\text{PHY}}$ due to its popularity associated with its simplicity and fast computation for personalized EP modeling. While the Eikonal model computes only the arrival time of activation wavefront in space, we use $\mathcal{M}_\phi$ to bridge its gap to the spatiotemporal action potential depolarization and repolarization process.

$\underline{\mathcal{M}_{\text{PHY}}}$: We consider the isotropic but heterogeneous Eikonal PDE given by:

$$|\nabla T(\mathbf{r})|\theta(\mathbf{r}) = 1 \tag{6.5}$$

where $T(\mathbf{r})$ denote the arrival time of activation wavefront at spatial location $\mathbf{r}$, and $\theta(\mathbf{r})$ denotes the local conduction velocity at $\mathbf{r}$. Given the initial locations of electrical activation and the heterogeneous conduction velocity $\theta(\mathbf{r})$ across the myocardium of the heart, Equation (6.5) can be solved in real-time to describe the isotropic propagation of action potential wavefront through the myocardium. It however does not model the realistic depolarization and repolarization dynamics of local action potential, nor the anisotropic spatial diffusion due to fiber orientation, which will be accommodated in an unknown neural component.

$\underline{\mathcal{M}_\phi}$: We model $\mathcal{M}_\phi$ to take inputs from Eikonal's output $T(\mathbf{r})$ and convert it to action potential $\mathbf{x}_{0:T}$ throughout the myocardium over time $[0, T]$:

$$\mathbf{X}_{0:T} = \mathcal{M}_\phi(T(\mathbf{r})). \tag{6.6}$$

Because $\mathbf{x}_{0:T}$ lives on a 3D geometry of the heart, we represent the myocardial mesh with an undirected k-nearest-neighbors (kNN) graph: each node of the myocardial mesh represents a vertex in the graph, and an edge is formed between a mesh node and its k nearest node neighbors as measured by Euclidean distance; the edge attributes between vertices are defined as the normalized differences in their 3D coordinates if an edge exists. On a given graph, $\mathcal{M}_\phi$ is realized as a spatial-temporal graph convolutional neural network (ST-GCNN) built on the spline-GCNN [56, 61] with interlaced graph convolution and temporal feature extraction operations:

$$\mathbf{f} * \mathbf{g} = \sum_{j \in N(i)} \mathbf{f}(j) \cdot \sum_{\mathbf{p} \in \mathcal{P}} \omega_{\mathbf{p}} B_{\mathbf{p}}(\mathbf{u}(i, j)) \tag{6.7}$$

where $\mathbf{f}$ is graph node features at each time instant, $\mathbf{u}(i, j)$ is the edge attribute between vertex i and j, $\mathbf{g}(\cdot)$ is the convolution kernel, $B_{\mathbf{p}}(\cdot)$ is the spline basis with its the Cartesian product $\mathcal{P}$ and $\omega_{\mathbf{p}}$ are trainable parameters. And the temporal feature extraction operation is implemented by the fully connected layers. The choice of spline-GCNN with its spatial convolution kernels allows the modeling across different hearts at both training and test times. This hybrid formulation leverages the fast conduction physics described by the Eikonal model, while allowing data-driven modeling of its gap to reality. Once $\theta(\mathbf{r})$ and ϕ are each identified for $\mathcal{M}_{\text{PHY}}$ and $\mathcal{M}_{\phi}$, respectively, a personalized hybrid cardiac EP model will be obtained.

6.3.2 Learning to Identify

The identification of $\mathcal{M}_{\text{Hybrid}}$ requires simultaneous identification of the parameter of $\mathcal{M}_{\text{PHY}}$ and $\mathcal{M}_{\text{NN}}$. Formally, we cast this into a meta-learning formulation. Consider a dataset $\mathcal{D}$ of action potentials with M similar but distinct underlying dynamics: $\mathcal{D} = \{\mathcal{D}_j\}_{j=1}^{M}$. For each $\mathcal{D}_j$, we consider disjoint few-shot context instances $\mathcal{D}_j^s = \left\{\mathbf{Y}_{0:T}^{s,1}, \mathbf{Y}_{0:T}^{s,2}, \ldots, \mathbf{Y}_{0:T}^{s,k}\right\}$ and query instances $\mathcal{D}_j^q = \left\{\mathbf{Y}_{0:T}^{q,1}, \mathbf{Y}_{0:T}^{q,2}, \ldots, \mathbf{Y}_{0:T}^{q,d}\right\}$, where $k \ll d$. Then we formulate a meta-objective to learn to identify the underlying true parameter vector θ of $\mathcal{M}_{\text{PHY}}$ from k-shot context instances $\mathcal{D}_j^s$, such that the identified HyPer is able to forecast for any query instances in $\mathcal{D}_j^q$ given only an estimate of its initial state $\hat{\mathbf{x}}_{0,j}^q$. More specifically, we have a feed-forward meta-model $\mathcal{G}_\zeta(\mathcal{D}_j^s)$ to learn to identify θ for dynamics j as:

$$\hat{\theta}_j = \mathcal{G}_\zeta(\mathcal{D}_j^s) = \frac{1}{k} \sum_{\mathbf{X}_{0:T}^s \in \mathcal{D}_j^s} \xi_\zeta(\mathbf{Y}_{0:T}^s) \tag{6.8}$$

where an embedding is extracted from each individual context instance via a meta-encoder ξ_ζ and gets aggregated across $\mathcal{D}_j^s$ to extract knowledge shared by the set. k is the size of the context set, and its value can be fixed or variable which we will demonstrate in the ablation study.

Given the inferred $\hat{\mathbf{X}}_{0,j}^q$ and θ, we minimize the forecasting accuracy on the query instances.

$$\left\{\hat{\phi}, \hat{\zeta}\right\} = \arg\min_{\phi,\zeta} \sum_{j=1}^{M} \sum_{\mathbf{Y}_{0:T}^q \in \mathcal{D}_j^q} \left\| \mathbf{Y}_{0:T}^q - g(\hat{\mathbf{X}}_{0:T}^q) \right\|_2^2 \tag{6.9}$$

6.4 Experiments and Results

Our proof-of-concept experiments were run on synthetic data generated by the two-variable Aliev-Panfilov model [1] for both instantiations.

$$\begin{aligned} \frac{du}{dt} &= \nabla(D\nabla u) + k * u(1-u) * (u-a) - uv \\ \frac{dv}{dt} &= -e(k * u(u-a-1) + v), \end{aligned} \tag{6.10}$$

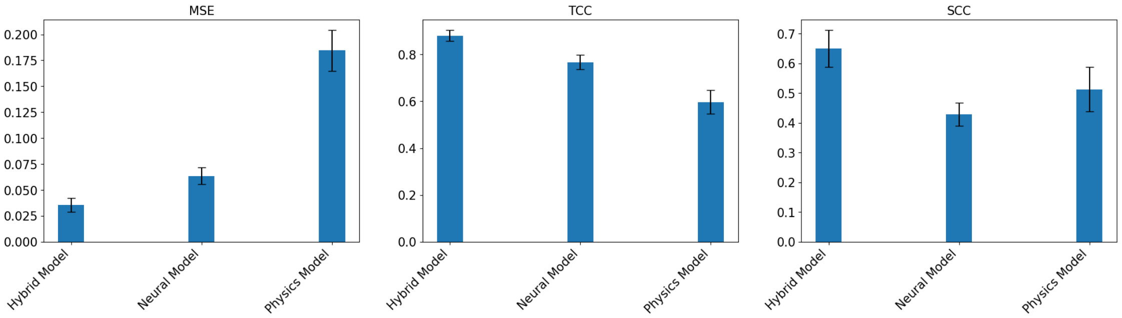


Figure 6.1: Summary of signal reconstruction performance on $\mathcal{M}_{\text{Hybrid}}$, $\mathcal{M}_{\text{NN}}$, and $\mathcal{M}_{\text{PHY}}$.

where u represents action potential, v recovery current, D the conductivity tensor, and the rest of the parameters controlling the temporal shape of the action potential. In particular, parameter a is known to control the excitability of heart tissue, where an increased value of a results in a reduced action potential duration and amplitude until an inability to activate. We consider parameter a as spatially varying to mimic regions of infarcted tissue for different subjects.

We use the signal generated as by the full Aliev-Panfilov model (Equation (6.10)) as the ground truth of the experiments. The signal is generated on 1,862 volumetric heart meshes with 186 different points of activation, each repeated for how many different parameter settings.

In HyPer-EP, as described earlier the partial physics $\mathcal{M}_{\text{PHY}}$ represents the Eikonal model and $\mathcal{M}_{\text{NN}}$ that has two linear layers followed by three layers of interlaced graph convolution for spatial features and 1D convolution to recover the signal from the activation time map. The meta-encoder is modeled using another three layers of interlaced graph convolution to extract spatial features and 1D convolution to aggregate temporal features, then averaging on all context samples to estimate the parameter mask. Hyper-EP is trained to, given the initial excitation point of a query sample and parameter θ estimated from $k = 5$ context samples, to reconstruct the action potential sequence for the query example. Hyper-EP was trained on three parameter settings in total with around 200 data samples, and tested on five parameter settings with around 60 data samples in each parameter setting.

Fig. 6.1 compares the performance of using only $\mathcal{M}_{\text{PHY}}$, $\mathcal{M}_{\text{NN}}$, and $\mathcal{M}_{\text{Hybrid}}$ in the same meta-learning framework, considering metrics of mean squared error (MSE), spatial correlation coefficient (SCC), and temporal correlation coefficient (TCC) between the reconstructed and true action potential. Visual examples are shown in Fig. 6.2. These results demonstrate the advantages of the hybrid model over either physics-based or neural-network modeling alone in learning personalized

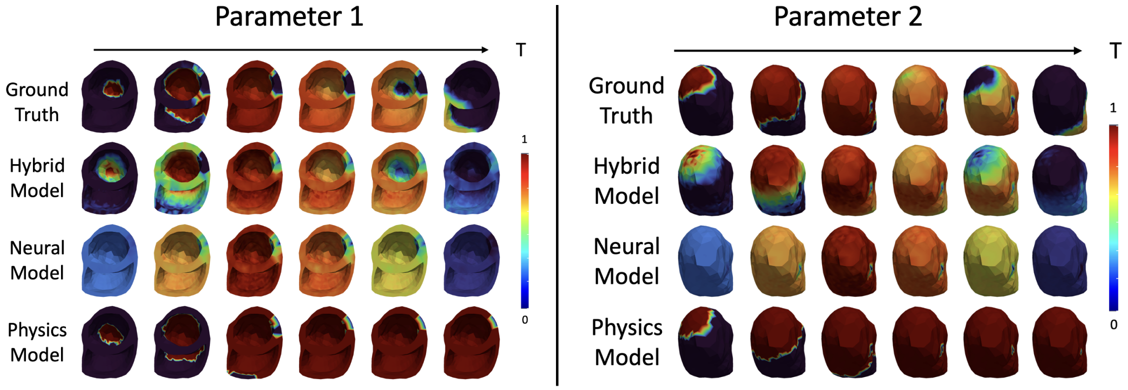


Figure 6.2: Visual examples of the action potential obtained from $\mathcal{M}_{\text{Hybrid}}$ versus $\mathcal{M}_{\text{NN}}$ and $\mathcal{M}_{\text{PHY}}$. $\mathcal{M}_{\text{Hybrid}}$ had the best details compared to the baselines. $\mathcal{M}_{\text{NN}}$ only had the information of depolarization and repolarization, while $\mathcal{M}_{\text{PHY}}$ only showed the signal propagation blocks.

cardiac EP models.

6.5 Conclusion

In this paper, we present a HyPer-EP framework that hybridizes physics-based modeling of prior knowledge with data-driven modeling of errors in prior physics and demonstrates the feasibility of meta-learning how to identify both components to realize personalized hybrid modeling of cardiac EP. A proof of concept is presented on the example of HyPer-EP on cardiac synthetic data. Future works will investigate more extensive experimental evaluations as well as real-data use.

Chapter 7

Conclusion and Future Works

In this thesis, we pointed out and reviewed important aspects of improving the adaptation of deep learning models with inductive bias. We approached the challenges by examining key research questions including: 1) how to learn to adapt with unknown knowledge that can be learned from data, 2) how to adapt deep learning models with known prior knowledge, and 3) how to learn to identify hybrid knowledge with both known prior and unknown errors. We demonstrated several advanced methods to learn to adapt with the summarized knowledge from data, incorporate known physics knowledge into deep learning, and a hybrid model that combines the strengths of both, with comprehensive experiments to demonstrate their improved performance.

7.1 Future Works

With the foundation of existing works, the future goal is to improve adaptive deep-learning models and focus on interdisciplinary directions of fundamental methodological developments and their applications to health care. Below are the three major aspects towards the goal:

Hybrid Modeling: Hybrid modeling has great significance at the intersection of deep learning and real-world systems, showing a promising future direction to expand the methodology and application of our initial works in hybrid SSM and Hyper-EP. However, two critical bottlenecks need to be overcome. The first one comes from the high-dimensional nature of the variables in physics, where these variables are usually defined in more complicated and detailed domains in general scientific applications. Toward this, we will investigate modeling the physics in abstract latent space, which can be realized by physics-based neural operators and learning strategies. The second one lies in the unmeasured effects between existing physics models and real-world observations, where such error effects and physics parameters are not able to be directly identified. Toward this, we plan to extend the Hyper-EP into a setting with weaker supervision. We plan to follow the two directions to develop physics-informed neural networks for complex systems.

Learning to Adapt Anytime Anywhere: There is a growing interest in high-dimensional time series forecasting, in which our initial work of *what-how* framework can be expanded as an emerging future direction. Two major issues need to be tackled. The first issue is that the assumption that the training and testing tasks follow the same distribution is not always true Toward this, we plan to further enable the inductive bias of learning to adapt to out-of-distribution data at test time, which is related to domain invariance learning. The second issue is that the dynamic parameters change over time within the time series at training time. We will investigate the inductive bias of continually adjusting the learned environmental knowledge from sequentially-arrived dynamics controlled by time-varying parameters, which links to continuous learning and online learning.

Personalized Health Care: One of the biggest goals in health care is to improve the precision of each individual, making personalized health care a future direction with great potential. The two future directions of methodology above can play a critical role in personalized health care. Many real-world clinical models are usually built on a limited batch of data that does not reflect the complete demographics. Toward this, we will explore the application of the first direction to improve the personalization of clinical models on diverse subjects. In addition, medical experiments cannot be directly done on patients and are usually performed offline repeatedly at discrete time instants. To this end, we plan to apply the inductive bias of continuous adaptation to develop real-time tracking systems for patients' health progressions.

7.2 Ethical Impact

The presented work is applicable to a broad spectrum of social problems such as personalized health care, business analysis, and ecological sustainability. This impact can be further enhanced by various practices such as respecting different demographics, transparency of bias, as well as continuous reflection and improvement of ethical practices. The outcome of the presented research work is expected to disseminate a broader impact on social trustworthiness and responsibility.

Bibliography

[1] Rubin R Aliev and Alexander V Panfilov. A simple two-variable model of cardiac excitation. Chaos, Solitons & Fractals, 7(3):293–301, 1996.

[2] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. arXiv preprint arXiv:1810.09502, 2018.

[3] Kedar Aras, Wilson Good, Jess Tate, Brett Burton, Dana Brooks, Jaume Coll-Font, Olaf Doessel, Walther Schulze, Danila Potyagaylo, Linwei Wang, et al. Experimental data and geometric analysis repository—edgar. Journal of electrocardiology, 48(6):975–981, 2015.

[4] Hermenegild J Arevalo, Fijoy Vadakkumpadan, Eliseo Guallar, Alexander Jebb, Peter Malamas, Katherine C Wu, and Natalia A Trayanova. Arrhythmia risk stratification of patients after myocardial infarction using personalized heart models. Nature communications, 7(1):11437, 2016.

[5] Tania Bacoyannis, Julian Krebs, Nicolas Cedilnik, Hubert Cochet, and Maxime Sermesant. Deep learning formulation of ecgi for data-driven integration of spatiotemporal correlations and imaging information. In International Conference on Functional Imaging and Modeling of the Heart, pages 20–28. Springer, 2019.

[6] Roger C Barr, Maynard Ramsey, and Madison S Spach. Relating epicardial to body surface potential distributions by means of transfer coefficients based on geometry measurements. IEEE Transactions on biomedical engineering, BME-24(1):1–11, 1977.

[7] Jason Bayer, Anton J Prassl, Ali Pashaei, Juan F Gomez, Antonio Frontera, Aurel Neic, Gernot Plank, and Edward J Vigmond. Universal ventricular coordinates: A generic framework for describing position within the heart and transferring data. Medical image analysis, 45:83–93, 2018.

[8] Jason D Bayer, Robert C Blake, Gernot Plank, and Natalia A Trayanova. A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models. Annals of biomedical engineering, 40:2243–2254, 2012.

[9] Laura R Bear, Leo K Cheng, Ian J LeGrice, Gregory B Sands, Nigel A Lever, David J Paterson, and Bruce H Smaill. Forward problem of electrocardiography: is it solved? Circulation: Arrhythmia and Electrophysiology, 8(3):677–684, 2015.

[10] Philip Becker-Ehmck, Jan Peters, and Patrick Van Der Smagt. Switching linear dynamics for variational bayes filtering. In International Conference on Machine Learning, pages 553–562. PMLR, 2019.

[11] J.A. Bergquist, J. Coll-Font, B. Zenger, L.C. Rupp, W.W. Good, D.H. Brooks, and R.S. MacLeod. Improving localization of cardiac geometry using ecgi. In Comput in Cardiol 2020, volume 47, pages 1–4, Sep 2020.

[12] Luca Bertinetto, J. F Henriques, J Valmadre, P. H. S. Torr, and Vedaldi A. Learning feed-forward one-shot learners. In Neural Information Processing Systems, 2016.

[13] Aleksandar Botev, Andrew Jaegle, Peter Wirnsberger, Daniel Hennes, and Irina Higgins. Which priors matter? benchmarking models for learning latent dynamics. In Advances in Neural Information Processing Systems, 2021.

[14] Carlos Alberto Brebbia, José Claudio Faria Telles, and Luiz C Wrobel. Boundary element techniques: theory and applications in engineering. Springer Science & Business Media, 2012.

[15] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine, 34(4):18–42, 2017.

[16] Dana H Brooks, Ghandi F Ahmad, Robert S MacLeod, and George M Maratos. Inverse electrocardiography by simultaneous imposition of multiple constraints. IEEE Transactions on Biomedical Engineering, 46(1):3–18, 1999.

[17] John Charles Butcher. Numerical methods for ordinary differential equations. John Wiley & Sons, 2016.

[18] Fernando Cacciola. Triangulated surface mesh simplification. In CGAL Editorial Board, editor, CGAL User and Reference Manual. 3.3 edition, 2007.

[19] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

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

[21] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In Advances in Neural Information Processing Systems, 2015.

[22] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289, 2015.

[23] Matthijs Cluitmans, Dana H Brooks, Rob MacLeod, Olaf Dössel, María S Guillem, Peter M Van Dam, Jana Svehlikova, Bin He, John Sapp, Linwei Wang, et al. Validation and opportunities of electrocardiographic imaging: from technical achievements to clinical applications. Frontiers in physiology, 9:1305, 2018.

[24] Matthijs JM Cluitmans, Pietro Bonizzi, Joël MH Karel, Marco Das, Bas LJH Kietselaer, Monique MJ de Jong, Frits W Prinzen, Ralf LM Peeters, Ronald L Westra, and Paul GA Volders. In vivo validation of electrocardiographic imaging. JACC: Clinical Electrophysiology, 3(3):232–242, 2017.

[25] Jaume Coll-Font and Dana H Brooks. Tracking the position of the heart from body surface potential maps and electrograms. Frontiers in Physiology, 9:1727, 2018.

[26] Jwala Dhamala, Sandesh Ghimire, John L Sapp, B Milan Horáček, and Linwei Wang. High-dimensional bayesian optimization of personalized cardiac model parameters via an embedded generative model. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 499–507. Springer, 2018.

[27] Jwala Dhamala, Sandesh Ghimire, John L Sapp, B Milan Horáček, and Linwei Wang. Bayesian optimization on large graphs via a graph convolutional generative model: Application in cardiac model personalization. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 458–467. Springer, 2019.

[28] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. IEEE transactions on pattern analysis and machine intelligence, 29(11):1944–1957, 2007.

[29] Jérémie Donà, Jean-Yves Franceschi, Sylvain Lamprier, and Patrick Gallinari. Pde-driven spatiotemporal disentanglement. arXiv preprint arXiv:2008.01352, 2020.

[30] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 38(2):295–307, 2015.

[31] Josselin Duchateau, Mark Potse, and Remi Dubois. Spatially coherent activation maps for electrocardiographic imaging. IEEE Transactions on Biomedical Engineering, 64(5):1149–1156, 2016.

[32] Burak Erem, Dana H Brooks, Peter M Van Dam, Jeroen G Stinstra, and Rob S MacLeod. Spatiotemporal estimation of activation times of fractionated ecgs on complex heart surfaces. In 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 5884–5887. IEEE, 2011.

[33] Burak Erem, Jaume Coll-Font, Ramon Orellana, Petr Stovicek, and Dana Brooks. Using transmural regularization and dynamic modeling for noninvasive cardiac potential imaging of endocardial pacing with imprecise thoracic geometry. IEEE transactions on medical imaging, 33:726–38, 03 2014.

[34] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 869–877, 2018.

[35] Carlos Figuera, Víctor Suárez-Gutiérrez, Ismael Hernández-Romero, Miguel Rodrigo, Alejandro Liberos, Felipe Atienza, María S Guillem, Óscar Barquero-Pérez, Andreu M Climent, and Felipe Alonso-Atienza. Regularization techniques for ecg imaging during atrial fibrillation: a computational study. Frontiers in physiology, 7:466, 2016.

[36] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning, pages 1126–1135. PMLR, 2017.

[37] Marco Fraccaro, Simon Kamronn, Ulrich Paquetz, and Ole Winthery. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In Advances in Neural Information Processing Systems, 2017.

[38] Stefania Fresca, Andrea Manzoni, Luca Dedè, and Alfio Quarteroni. Pod-enhanced deep learning-based reduced order models for the real-time simulation of cardiac electrophysiology in the left atrium. Frontiers in physiology, page 1431, 2021.

[39] M. Garnelo, D. Rosenbaum, C. J. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. J. Rezende, , and S. M. A. Eslami. Conditional neural processes. In International conference on machine learning, 2018.

[40] Sandesh Ghimire, Jwala Dhamala, Prashnna Kumar Gyawali, John L Sapp, Milan Horacek, and Linwei Wang. Generative modeling and inverse imaging of cardiac transmembrane potential. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 508–516. Springer, 2018.

[41] Sandesh Ghimire, Prashnna Kumar Gyawali, Jwala Dhamala, John L Sapp, Milan Horacek, and Linwei Wang. Improving generalization of deep networks for inverse reconstruction of image sequences. In International Conference on Information Processing in Medical Imaging, pages 153–166. Springer, 2019.

[42] Sandesh Ghimire, John L Sapp, B Milan Horáček, and Linwei Wang. Noninvasive reconstruction of transmural transmembrane potential with simultaneous estimation of prior model error. IEEE transactions on medical imaging, 38(11):2582–2595, 2019.

[43] Subham Ghosh and Yoram Rudy. Application of l1-norm regularization to epicardial potential solution of the inverse electrocardiography problem. Annals of biomedical engineering, 37(5):902–912, 2009.

[44] Sophie Giffard-Roisin, Hervé Delingette, Thomas Jackson, Jessica Webb, Lauren Fovargue, Jack Lee, Christopher A Rinaldi, Reza Razavi, Nicholas Ayache, and Maxime Sermesant. Transfer learning from simulations on a reference anatomy for ecgi in personalized cardiac resynchronization therapy. IEEE Transactions on Biomedical Engineering, 66(2):343–353, 2018.

[45] Karli Gillette, Matthias AF Gsell, Julien Bouyssier, Anton J Prassl, Aurel Neic, Edward J Vigmond, and Gernot Plank. Automated framework for the inclusion of a his–purkinje system in cardiac digital twins of ventricular electrophysiology. Annals of Biomedical Engineering, 49(12):3143–3153, 2021.

[46] Karli Gillette, Matthias AF Gsell, Anton J Prassl, Elias Karabelas, Ursula Reiter, Gert Reiter, Thomas Grandits, Christian Payer, Darko Štern, Martin Urschler, et al. A framework for the generation of digital twins of cardiac electrophysiology from clinical 12-leads ecgs. Medical Image Analysis, 71:102080, 2021.

[47] Bin He, Guanglin Li, and Xin Zhang. Noninvasive imaging of cardiac transmembrane potentials within three-dimensional myocardium by means of a realistic geometry anisotropic heart model. IEEE Transactions on Biomedical Engineering, 50(10):1190–1202, 2003.

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

[49] Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S Sukhatme. Neuralsim: Augmenting differentiable simulators with neural networks. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 9474–9481. IEEE, 2021.

[50] Clara Herrero Martin, Alon Oved, Rasheda A Chowdhury, Elisabeth Ullmann, Nicholas S Peters, Anil A Bharath, and Marta Varela. Ep-pinns: Cardiac electrophysiology characterisation using physics-informed neural networks. Frontiers in Cardiovascular Medicine, 8:768419, 2022.

[51] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine, 29(6):82–97, 2012.

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

[53] Joe Horvath, Lu Shien, Tommy Peng, Avinash Malik, Mark Trew, and Laura Bear. Deep learning neural nets for detecting heart activity. arXiv preprint arXiv:1901.09831, 2019.

[54] SCI Institute, 2016. SCIRun: A Scientific Computing Problem Solving Environment, Scientific Computing and Imaging Institute (SCI), Download from: http://www.scirun.org.

[55] Mingfeng Jiang, Feng Liu, Yaming Wang, Guofa Shou, Wenqing Huang, and Huaxiong Zhang. A hybrid model of maximum margin clustering method and support vector regression for noninvasive electrocardiographic imaging. Computational and Mathematical Methods in Medicine, 2012, 2012.

[56] Xiajun Jiang, Sandesh Ghimire, Jwala Dhamala, Zhiyuan Li, Prashnna Kumar Gyawali, and Linwei Wang. Learning geometry-dependent and physics-based inverse image reconstruction. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 487–496. Springer, 2020.

[57] Xiajun Jiang, Zhiyuan Li, Ryan Missel, Md Shakil Zaman, Brian Zenger, Wilson W Good, Rob S MacLeod, John L Sapp, and Linwei Wang. Few-shot generation of personalized neural surrogates for cardiac simulation via bayesian meta-learning. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII, pages 46–56. Springer, 2022.

[58] Xiajun Jiang, Ryan Missel, Zhiyuan Li, and Linwei Wang. Sequential latent variable models for few-shot high-dimensional time-series forecasting. In The Eleventh International Conference on Learning Representations, 2023.

[59] Xiajun Jiang, Ryan Missel, Maryam Toloubidokhti, Karli Gillette, Anton J Prassl, Gernot Plank, B Milan Horáček, John L Sapp, and Linwei Wang. Hybrid neural state-space modeling for supervised and unsupervised electrocardiographic imaging. IEEE Transactions on Medical Imaging, 2024.

[60] Xiajun Jiang, Ryan Missel, Maryam Toloubidokhti, Zhiyuan Li, Omar Gharbia, John L Sapp, and Linwei Wang. Label-free physics-informed image sequence reconstruction with disentangled spatial-temporal modeling. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 361–371. Springer, 2021.

[61] Xiajun Jiang, Maryam Toloubidokhti, Jake Bergquist, Brian Zenger, Wilson W Good, Rob S MacLeod, and Linwei Wang. Improving generalization by learning geometry-dependent and physics-based reconstruction of image sequences. IEEE Transactions on Medical Imaging, 2022.

[62] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. In International Conference on Learning Representations, 2017.

[63] Amel Karoui, Mostafa Bendahmane, and Nejib Zemzemi. A spatial adaptation of the time delay neural network for solving ecgi inverse problem. In International Conference on Functional Imaging and Modeling of the Heart, pages 94–102. Springer, 2019.

[64] Amel Karoui, Mostafa Bendahmane, and Nejib Zemzemi. Cardiac activation maps reconstruction: a comparative study between data-driven and physics-based methods. Frontiers in Physiology, 12:686136, 2021.

[65] Victoriya Kashtanova, Ibrahim Ayed, Nicolas Cedilnik, Patrick Gallinari, and Maxime Sermesant. Ep-net 2.0: Out-of-domain generalisation for deep learning models of cardiac elec-

trophysiology. In International Conference on Functional Imaging and Modeling of the Heart, pages 482–492. Springer, 2021.

[66] Victoriya Kashtanova, Mihaela Pop, Ibrahim Ayed, Patrick Gallinari, and Maxime Sermesant. Simultaneous data assimilation and cardiac electrophysiology model correction using differentiable physics and deep learning. Interface Focus, 13(6):20230043, 2023.

[67] David UJ Keller, Frank M Weber, Gunnar Seemann, and Olaf Dössel. Ranking the influence of tissue conductivities on forward-calculated ecgs. IEEE Transactions on Biomedical Engineering, 57(7):1568–1576, 2010.

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

[69] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[70] Alexej Klushyn, Richard Kurle, Maximilian Soelch, Botond Cseke, and Patrick van der Smagt. Latent matters: Learning deep-state-space models. In Advances in Neural Information Processing Systems, 2021.

[71] Rahul G. Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In Association for the Advancement of Artificial Intelligence, 2017.

[72] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[73] L Li, J Yan, X Yang, and Y Jin. Learning interpretable deep state space model for probabilistic time series forecasting. In International Joint Conference on Artificial Intelligence, 2019.

[74] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In Artificial Intelligence and Statistics, pages 914–922. PMLR, 2017.

[75] G. R. Liu. Mesh free methods: moving beyond the finite element method. CRC Press, Boca Raton, Fla, 2003.

[76] Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. IEEE Signal Processing Magazine, 35(1):20–36, 2018.

[77] Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. Advances in Neural Information Processing Systems, 30, 2017.

[78] S Makridakis, E Spiliotis, and V Assimakopoulos. S. Statistical and machine learning forecasting methods: Concerns and ways forward. PLoS ONE, 13, 2018.

[79] Avinash Malik, Tommy Peng, and Mark L Trew. A machine learning approach to reconstruction of heart surface potentials from body surface potentials. In 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pages 4828–4831. IEEE, 2018.

[80] Bhairav Mehta, Ankur Handa, Dieter Fox, and Fabio Ramos. A user's guide to calibrating robotic simulators. In Conference on Robot Learning, pages 1326–1340. PMLR, 2021.

[81] Christoph M Michel and Micah M Murray. Towards the utilization of eeg as a brain imaging tool. Neuroimage, 61(2):371–385, 2012.

[82] Colleen C Mitchell and David G Schaeffer. A two-current model for the dynamics of cardiac membrane. Bulletin of mathematical biology, 65(5):767–793, 2003.

[83] P. Montero-Manso, G. Athanasopoulos, R. J. Hyndman, and T. S Talagala. FFORMA: Feature-based forecast model averaging. International Journal of Forecasting, 36, 2020.

[84] Aurel Neic, Fernando O Campos, Anton J Prassl, Steven A Niederer, Martin J Bishop, Edward J Vigmond, and Gernot Plank. Efficient computation of electrograms and ecgs in human whole heart simulations using a reaction-eikonal model. Journal of computational physics, 346:191–211, 2017.

[85] Aurel Neic, Matthias AF Gsell, Elias Karabelas, Anton J Prassl, and Gernot Plank. Automating image-based mesh generation and manipulation tasks in cardiac modeling workflows using meshtool. SoftwareX, 11:100454, 2020.

[86] Bjørn Fredrik Nielsen, Marius Lysaker, and Per Grøttum. Computing ischemic regions in the heart with the bidomain model—first steps towards validation. IEEE Transactions on Medical Imaging, 32(6):1085–1096, 2013.

[87] Sajad Norouzi, David J Fleet, and Mohammad Norouzi. Exemplar vae: Linking generative models, nearest neighbor retrieval, and data augmentation. Advances in Neural Information Processing Systems, 33:8753–8764, 2020.

[88] Yoshiwo Okamoto, Yasuaki Teramachi, and Toshimitus Musha. Limitation of the inverse problem in body surface potential mapping. IEEE transactions on biomedical engineering, BME-30(11):749–754, 1983.

[89] Önder Nazim Onak, Taha Erenler, and Yesim Serinagaoglu Dogrusoz. A novel data-adaptive regression framework based on multivariate adaptive regression splines for electrocardiographic imaging. IEEE Transactions on Biomedical Engineering, 69(2):963–974, 2021.

[90] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. Meta-learning framework with applications to zero-shot time-series forecasting. In AAAI Conference on Artificial Intelligence, 2021.

[91] B. N. Oreshkina, D. Carpov, N. Chapados, and Y. Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In International Conference on Learning Representations, 2020.

[92] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2536–2544, 2016.

[93] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. arXiv preprint arXiv:2010.03409, 2020.

[94] R. Plonsey. Bioelectric Phenomena. McGraw Hill, New York, 1969.

[95] AJ Pullan, LK Cheng, MP Nash, CP Bradley, and DJ Paterson. Noninvasive electrical imaging of the heart: theory and model development. Annals of biomedical engineering, 29:817–836, 2001.

[96] A.J. Pullan, L.K.Cheng, M.P. Nash, D.H. Brooks, A. Ghodrati, and R.S. MacLeod. The inverse problem of electrocardiography. In P.W. Macfarlane, A. van Oosterom, O. Pahlm, P. Kligfield, M. Janse, and J. Camm, editors, Comprehensive Electrocardiology, pages 299–344. Springer Verlag, London, UK, 2010.

[97] Shenghao Qin, Jiacheng Zhu, Jimmy Qin, Wenshuo Wang, and Ding Zhao. Recurrent attentive neural process for sequential data. In arXiv, 2019.

[98] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. arXiv preprint arXiv:2001.04385, 2020.

[99] Abhejit Rajagopal, Vincent Radzicki, Hua Lee, and Shivkumar Chandrasekaran. Nonlinear electrocardiographic imaging using polynomial approximation networks. APL bioengineering, 2(4), 2018.

[100] Charulatha Ramanathan, Raja N Ghanem, Ping Jia, Kyungmoo Ryu, and Yoram Rudy. Noninvasive electrocardiographic imaging for cardiac electrophysiology and arrhythmia. Nature medicine, 10(4):422–428, 2004.

[101] Syama Sundar Rangapuram, Matthias Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In Neural Information Processing Systems, 2018.

[102] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In International Conference on Learning Representations (ICLR), 2016.

[103] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[104] David E Roberts and Allen M Scher. Effect of tissue anisotropy on extracellular potential fields in canine myocardium in situ. Circulation Research, 50(3):342–351, 1982.

[105] A. Rodenhauser, W.W. Good, B. Zenger, J.D. Tate, K. Aras, B.M. Burton, and R.S. MacLeod. PFEIFER: Preprocessing framework for electrograms intermittently fiducialized from experimental recordings. J. Open Source Software, 3(21):472, September 2018.

[106] Miguel Rodrigo, Andreu M Climent, Alejandro Liberos, Ismael Hernández-Romero, Angel Arenal, Javier Bermejo, Francisco Fernández-Avilés, Felipe Atienza, and Maria S Guillem. Solving inaccuracies in anatomical models for electrocardiographic inverse problem resolution by maximizing reconstruction quality. IEEE transactions on medical imaging, 37(3):733–740, 2017.

[107] Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent odes for irregularly-sampled time series. In Neural Information Processing Systems, 2019.

[108] Y Rudy and HS Oster. The electrocardiographic inverse problem. Critical reviews in biomedical engineering, 20(1-2):25, 1992.

[109] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 36:1181–1191, 2020.

[110] John L Sapp, Fady Dawoud, John C Clements, and B Milan Horáček. Inverse solution mapping of epicardial potentials: quantitative comparison with epicardial contact mapping. Circulation: Arrhythmia and Electrophysiology, 5(5):1001–1009, 2012.

[111] Lev Semenovich, LW Neustadt, and KN Trirogoff. The Mathematical Theory of Optimal Processes. Interscience Publishers, 1962.

[112] Maxime Sermesant, Radomir Chabiniok, Phani Chinchapatnam, Tommaso Mansi, Florence Billet, Philippe Moireau, Jean-Marc Peyrat, K Wong, Jatin Relan, Kawal Rhode, et al. Patient-specific electromechanical models of the heart for the prediction of pacing acute effects in crt: a preliminary clinical validation. Medical image analysis, 16(1):201–215, 2012.

[113] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3693–3702, 2017.

[114] Gautam Singh, Jaesik Yoon, and Youngsung Son. Sequential neural processes. In Neural Information Processing Systems, 2019.

[115] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In Computer Vision and Pattern Recognition, 2018.

[116] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. arXiv preprint physics/0004057, 2000.

[117] Peter M Van Dam, Thom F Oostendorp, André C Linnenbank, and Adriaan Van Oosterom. Non-invasive imaging of cardiac activation and recovery. Annals of biomedical engineering, 37(9):1739–1756, 2009.

[118] EJ Vigmond, R Weber Dos Santos, AJ Prassl, M Deo, and G Plank. Solvers for the cardiac bidomain equations. Progress in biophysics and molecular biology, 96(1-3):3–18, 2008.

[119] Linwei Wang, Heye Zhang, Ken CL Wong, Huafeng Liu, and Pengcheng Shi. Physiological-model-constrained noninvasive reconstruction of volumetric myocardial transmembrane potentials. IEEE Transactions on Biomedical Engineering, 57(2):296–315, 2009.

[120] Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. arXiv preprint arXiv:2002.03061, 2020.

[121] Rui Wang, Robin Walters, and Rose Yu. Meta-learning dynamics forecasting using task inference. arXiv preprint arXiv:2102.10271, 2021.

[122] Ken CL Wong, Maxime Sermesant, Kawal Rhode, Matthew Ginks, C Aldo Rinaldi, Reza Razavi, Hervé Delingette, and Nicholas Ayache. Velocity-based cardiac contractility personalization from images using derivative-free optimization. Journal of the mechanical behavior of biomedical materials, 43:35–52, 2015.

[123] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2020.

[124] Jingjia Xu, Azar Rahimi Dehaghani, Fei Gao, and Linwei Wang. Noninvasive transmural electrophysiological imaging based on minimization of total-variation functional. IEEE transactions on medical imaging, 33(9):1860–1874, 2014.

[125] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Learning physics constrained dynamics using autoencoders. Advances in Neural Information Processing Systems, 35:17157–17172, 2022.

[126] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep admm-net for compressive sensing mri. In Proceedings of the 30th international conference on neural information processing systems, pages 10–18, 2016.

[127] C. Yildiz, M. Heinonen, and H. Lahdesmaki. ODE^{2}VAE: Deep generative second order odes with bayesian neural networks. In Neural Information Processing Systems, 2020.

[128] Yuan Yin, Ibrahim Ayed, Emmanuel de Bézenac, Nicolas Baskiotis, and Patrick Gallinari. Leads: Learning dynamical systems that generalize across environments. Advances in Neural Information Processing Systems, 34:7561–7573, 2021.

[129] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. Advances in neural information processing systems, 31, 2018.

[130] Sohail Zahid, Kaitlyn N Whyte, Erica L Schwarz, Robert C Blake III, Patrick M Boyle, Jonathan Chrispin, Adityo Prakosa, Esra G Ipek, Farhad Pashakhanloo, Henry R Halperin, et al. Feasibility of using patient-specific models and the "minimum cut" algorithm to predict optimal ablation targets for left atrial flutter. Heart rhythm, 13(8):1687–1698, 2016.

[131] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE transactions on image processing, 26(7):3142–3155, 2017.

[132] Yiheng Zhang, Alireza Ghodrati, and Dana H Brooks. An analytical comparison of three spatio-temporal regularization methods for dynamic linear inverse problems in a common statistical framework. Inverse Problems, 21(1):357, 2005.

[133] Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. Nature, 555(7697):487, 2018.

Appendices

Appendix A

Appendix for Chapter 3

A.1 Derivation of (3.7)

$$
\begin{aligned}
&\textstyle\sum_{j=1}^{M} \sum_{\mathbf{x}_{0:T}^{q} \in \mathcal{D}_{j}^{q}} \log p(\mathbf{x}_{0:T}^{q} | \mathcal{D}_{j}^{s}) \\
&= \textstyle\sum_{j=1}^{M} \sum_{\mathbf{x}_{0:T}^{q} \in \mathcal{D}_{j}^{q}} \log \int_{\mathbf{c}} p(\mathbf{x}_{0:T}^{q} | \mathbf{c}) p_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s}) d\mathbf{c} \\
&= \textstyle\sum_{j=1}^{M} \sum_{\mathbf{x}_{0:T}^{q} \in \mathcal{D}_{j}^{q}} \log \int_{\mathbf{c}} \int_{\mathbf{z}_0} \left[p_{\theta_x}(\mathbf{x}_0 | \mathbf{z}_0, \mathbf{c}) \prod_{i=1}^{T} p_{\theta_x}(\mathbf{x}_i | \mathbf{z}_i, \mathbf{c}) \right] \\
&\cdot \frac{p(\mathbf{z}_0)}{q_{\phi}(\mathbf{z}_0 | \mathbf{x}_{0:l_{z_0}}^{q})} \frac{p_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s})}{q_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s} \cup \mathbf{x}_{0:T}^{q})} \cdot q_{\phi}(\mathbf{z}_0 | \mathbf{x}_{0:l_{z_0}}^{q}) q_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s} \cup \mathbf{x}_{0:T}^{q}) d\mathbf{z}_0 d\mathbf{c} \\
&\geq \textstyle\sum_{j=1}^{M} \sum_{\mathbf{x}_{0:T}^{q} \in \mathcal{D}_{j}^{q}} \int_{\mathbf{c}} \int_{\mathbf{z}_0} \Bigg[\log p_{\theta_x}(\mathbf{x}_{0:T}^{q} | \mathbf{z}_0^{q}, \mathbf{c}) - \log \frac{q_{\phi}(\mathbf{z}_0 | \mathbf{x}_{0:l_{z_0}}^{q})}{p(\mathbf{z}_0)} \\
&- \log \frac{q_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s} \cup \mathbf{x}_{0:T}^{q})}{p_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s})} \Bigg] \cdot q_{\phi}(\mathbf{z}_0 | \mathbf{x}_{0:l_{z_0}}^{q}) q_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s} \cup \mathbf{x}_{0:T}^{q}) d\mathbf{z}_0 d\mathbf{c} \\
&= \textstyle\sum_{j=1}^{M} \sum_{\mathbf{x}_{0:T}^{q} \in \mathcal{D}_{j}^{q}} \mathbb{E}_{q_{\phi}(\mathbf{z}_0^{q}), q_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s} \cup \mathbf{x}_{0:T}^{q})} [\log p_{\theta_x}(\mathbf{x}_{0:T}^{q} | \mathbf{z}_0^{q}, \mathbf{c})] \\
&- \mathrm{KL}(q_{\phi}(\mathbf{z}_0^{q} | \mathbf{x}_{0:l_{z_0}}^{q}) || p(\mathbf{z}_0)) - \mathrm{KL}\left(q_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s} \cup \mathbf{x}_{0:T}^{q}) || p_{\zeta}(\mathbf{c} | \mathcal{D}_{j}^{s}) \right)
\end{aligned}
$$

A.2 Gated Recurrent Unit (GRU) Set-Conditioning

We condition the GRU cell of the transition function through (A.1),

$$\begin{aligned}
&\mathbf{z}_{i-1}^{(1)} = \mathrm{ELU}(\boldsymbol{\alpha}_1\mathbf{z}_{i-1} + \boldsymbol{\beta}_1\mathbf{c} + \boldsymbol{\gamma}_1), \quad \mathbf{g}_{i-1} = \sigma(\mathbf{W}_1\mathbf{z}_{i-1}^{(1)} + \mathbf{b}_1) \\
&\mathbf{z}_{i-1}^{(2)} = \mathrm{ELU}(\boldsymbol{\alpha}_2\mathbf{z}_{i-1} + \boldsymbol{\beta}_2\mathbf{c} + \boldsymbol{\gamma}_2), \quad \mathbf{h}_{i-1} = \mathrm{ELU}(\mathbf{W}_2\mathbf{z}_{i-1}^{(2)} + \mathbf{b}_2) \\
&\tilde{\mathbf{z}}_{i-1} = \boldsymbol{\alpha}_3\mathbf{z}_{i-1} + \boldsymbol{\beta}_3\mathbf{c} + \boldsymbol{\gamma}_3, \quad \Delta\mathbf{z}_i = (1 - \mathbf{g}_{i-1}) \odot (\mathbf{W}_3\tilde{\mathbf{z}}_{i-1} + \mathbf{b}_3) + \mathbf{g}_{i-1} \odot \mathbf{h}_{i-1},
\end{aligned} \tag{A.1}$$

where $\boldsymbol{\theta}_z = \{\mathbf{W}_i, \mathbf{b}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{\gamma}_i\}_{i=1}^{3}$ are learnable parameters of the dynamic function.

A.3 Extra Constraint to Meta-Model

We considered adding an additional regularization to the set embedding $p_\zeta(\mathbf{c}|\mathcal{D}_j^s)$ so that it is constrained to a reasonable range. Since the true posterior density $p_\zeta(\mathbf{c}|\mathcal{D}_j^s)$ is unknown, we assume that it is bounded by a standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$. Therefore, the objective function of the model becomes:

$$\begin{aligned}
\arg\min \sum\nolimits_{j=1}^{M} \sum_{\mathbf{x}_{0:T}^q \in \mathcal{D}_j^q} &\mathbb{E}_{q_\phi(\mathbf{z}_0^q), q_\zeta(\mathbf{c}|\mathcal{D}_j^s \cup \mathbf{x}_{0:T}^q)}[\log p_{\theta_x}(\mathbf{x}_{0:T}^q|\mathbf{z}_0^q, \mathbf{c})] \\
&- \mathrm{KL}(q_\phi(\mathbf{z}_0^q|\mathbf{x}_{0:l_{z_0}}^q)||p(\mathbf{z}_0)) - \mathrm{KL}\left(q_\zeta(\mathbf{c}|\mathcal{D}_j^s \cup \mathbf{x}_{0:T}^q)||p_\zeta(\mathbf{c}|\mathcal{D}_j^s)\right) \\
&- \mathrm{KL}\left(p_\zeta(\mathbf{c}|\mathcal{D}_j^s)||\mathcal{N}(0, \mathbf{I})\right)
\end{aligned}$$

We applied the extra constraint to the proposed meta-GRU-res model for both fixed and variable k at different value of $k = 1, 5, 9, 15$ and evaluated on *gravity-16* dataset. Fig. A.1 summaries the quantitative test performance of the two models trained with and without the constraint on the set embedding. The constraint generally had no noticeable effect on model performance. Visual examples at $k = 1$ are also shown in Fig. A.2. It shows that when k is small, the model with the constraint had a slightly better performance.

A.4 Additional Gravity-16 Results

Here we provide the complete results of all baseline models trained on *gravity-16* in Table A.1 when split between the known dynamics during training and the unknown ones during testing. We note that the VRNN fails to converge in any of the single dynamics cases, in contrast to its full

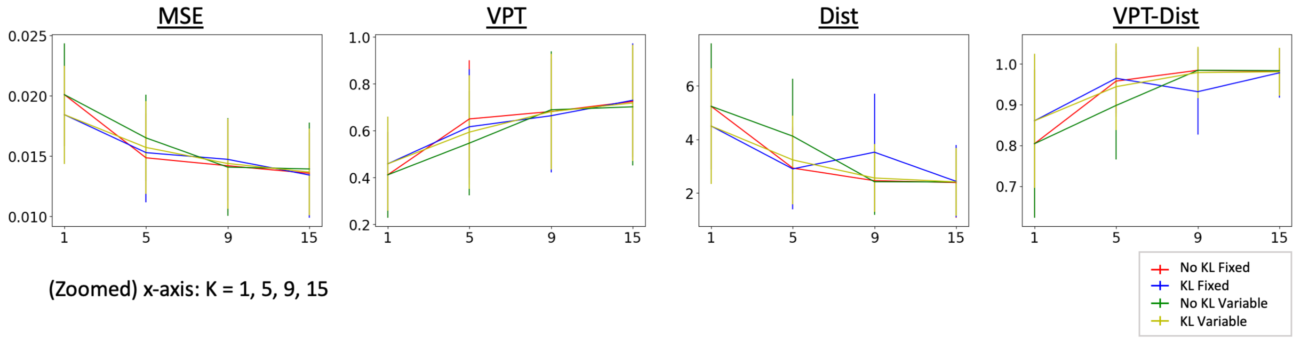


Figure A.1: Comparison with models with and without constraint.

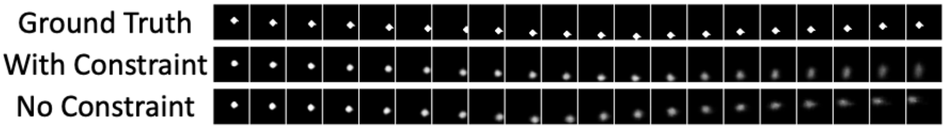


Figure A.2: Examples of models with and without constraint.

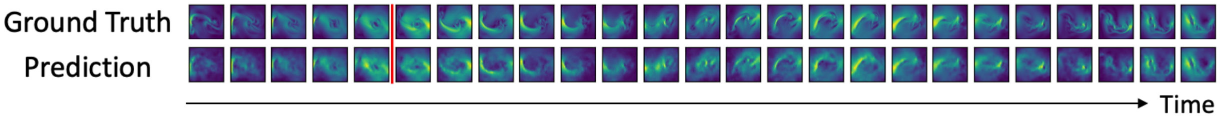


Figure A.3: Example of forecasting *known* turbulent flow dynamics.

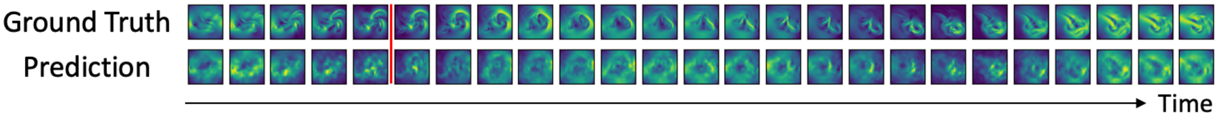


Figure A.4: Example of forecasting *unknown* turbulent flow dynamics.

dynamics training, which is likely due to the lower data availability in these settings. The other baselines DKF and KVAE see similar decreases in performance compared to their full meta-training set performance.

A.5 Turbulent Flow Visualizations

Here we provide visualizations of predicted trajectories for both known and unknown buoyancy factors on the turbulent flow dataset within Fig. A.3 and Fig. A.4.

Table A.1: Comparison of the presented meta and base models on known and unknown dynamics.

Model	Dynamics	MSE↓	VPT-MSE↑	Dist↓	VPT-Dist↑
meta-GRU-res	known	**1.43(0.34)e-2**	**0.68(0.26)**	**2.86(1.44)**	**0.97(0.06)**
	unknown	**1.45(0.33)e-2**	**0.67(0.25)**	**2.96(1.49)**	**0.97(0.07)**
meta-NODE	known	1.60(0.26)e-2	0.58(0.23)	6.05(2.65)	0.80(0.12)
	unknown	1.62(0.26)e-2	0.57(0.22)	6.23(2.54)	0.80(0.12)
meta-RGN-res	known	1.59(0.25)e-2	0.57(0.21)	6.89(3.08)	0.76(0.13)
	unknown	1.60(0.23)e-2	0.56(0.20)	7.23(3.09)	0.75(0.13)
GRU-res	known	1.80(0.29)e-2	0.46(0.23)	6.86(3.95)	0.77(0.19)
	unknown	1.99(0.277)e-2	0.37(0.18)	8.07(3.56)	0.69(0.17)
GRU-res finetune	unknown	2.03(0.27)e-2	0.35(0.17)	8.51(3.42)	0.66(0.18)
NODE	known	1.97(0.21)e-2	0.30(0.20)	10.6(4.13)	0.60(0.17)
	unknown	2.06(0.22)e-2	0.27(0.17)	11.1(3.71)	0.57(0.16)
NODE finetune	unknown	2.07(0.21)e-2	0.27(0.17)	11.2(3.72)	0.55(0.16)
RGN-res	known	1.93(0.22)e-2	0.34(0.20)	10.1(4.11)	0.61(0.18)
	unknown	2.03(0.22)e-2	0.30(0.16)	10.8(3.72)	0.57(0.16)
RGN-res finetune	unknown	2.05(0.22)e-2	0.29(0.16)	10.7(3.62)	0.56(0.16)
meta-DKF	known	3.81(0.59)e-2	0.10(0.11)	7.37(3.27)	0.70(0.25)
	unknown	3.80(0.59)e-2	0.10(0.11)	7.30(3.21)	0.70(0.25)
DKF	known	3.89(0.32)e-2	0.08(0.06)	10.7(3.17)	0.46(0.20)
	unknown	3.88(0.32)e-2	0.08(0.06)	10.7(3.23)	0.45(0.20)
DKF finetune	unknown	3.89(0.32)e-2	0.08(0.06)	10.8(3.24)	0.45(0.20)
VRNN	known	2.32(14.6)e-2	0.02(0.11)	45.3(0.00)	0.00(0.00)
	unknown	2.32(14.6)e-2	0.01(0.08)	45.3(0.00)	0.00(0.00)
VRNN finetune	unknown	2.34(14.5)e-2	0.04(0.15)	45.3(0.00)	0.00(0.00)
DVBF	known	2.32(14.3)e-2	0.01(0.02)	45.2(0.00)	0.00(0.00)
	unknown	2.43(14.1)e-2	0.01(0.07)	45.3(0.00)	0.00(0.00)
DVBF finetune	unknown	2.35(14.1)e-2	0.01(0.08)	45.3(0.00)	0.00(0.00)
KVAE	known	3.42(1.30)e-2	0.39(0.34)	5.05(3.57)	0.5(0.34)
	unknown	3.46(1.36)e-2	0.22(0.19)	5.17(3.91)	0.53(0.28)
Donà et al	known	3.58(0.33)e-2	0.00(0.01)	13.7(3.36)	0.07(0.18)
	unknown	3.56(0.34)e-2	0.00(0.01)	14.1(3.84)	0.08(0.19)

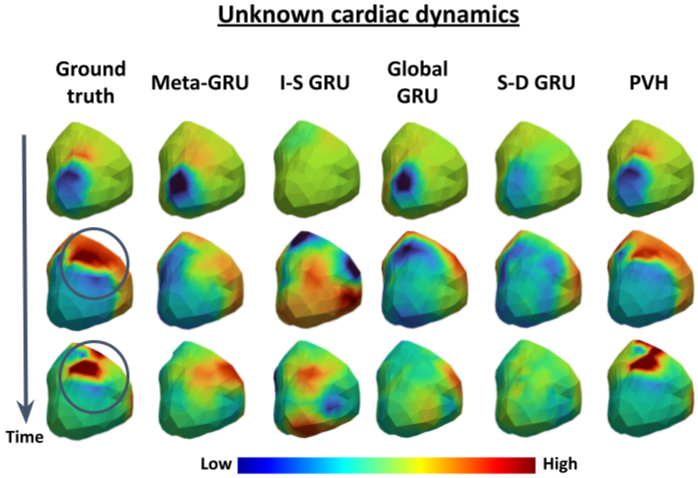


Figure A.5: Example of forecasting *unknown* electrical dynamics on the heart.

A.6 Additional Cardiac Figure

Here we provide an additional result on the cardiac experiments in Figure A.5, this time highlighting the performance of the model on unknown dynamic cardiac dynamics when compared against the baselines. We use a varying support set k size between 1 to 5 samples. The proposed model manages to effectively model around the scar tissue present in the ground truth. The baselines, besides the expensive personalized virtual heart (PVH), are unable to account for the specific dynamics of this subject and propagate over it.

A.7 Data Details

In this section, we give the specific data sizes and splits used for training throughout the experiments, as well as generation procedures and considerations for each.

For the bouncing balls, we leveraged the PyMunk Physics Engine (`www.pymunk.org`) to perform simulations under various gravity following [37]. For pendulum and mass-spring systems, we leveraged the Hamiltonian Dynamics Suite presented in [13]. The suite's default physical parameters were used and friction coefficients were introduced to build non-energy conserving systems. For data consistency, we extracted the red color channel of pendulum and mass-spring systems to generate gray-scale images.

For *gravity-16* experiments, we generated bouncing balls dynamic trajectories on 32×32 images. All dynamics are under 16 different gravitational constants, where $g = 3 + \epsilon, \epsilon \sim U(0, 1)$ and the direction of each gravity is evenly distributed over the 2D space. We took the trajectories samples at every $\Delta t = 0.2$ intervals. The initial position of the ball is set to a 16×16 region centered in the image, and the initial velocity is randomly sampled from $[0, 10]$ on both x and y directions. Each gravity setting has 3,000 samples in total. For *gravity-16* data, we used 10 gravity in meta-training, 2 in meta-validation, and 4 in meta-testing. We also left out samples from both meta-training and meta-validation sets (∼1500 for each gravity) to evaluate the performance of the model on the known dynamics.

For *mixed-physics* experiments, we generated a mixed-physics dataset consisting of bouncing balls under 4 gravity directions, and pendulums and mass springs each with four different values of friction coefficients of $0, 0.05, 0.1, 0.15$. The bouncing ball dataset is similar to *gravity-16* experiment except for the 4 gravity directions. Both pendulums and mass springs took the trajectories samples at every $\Delta t = 0.2$ intervals. In pendulums, the mass of the particle $m = 0.5$, the gravitational constant $g = 3$ and the pivot length $l = 1$. The friction coefficients are chosen in $0, 0.05, 0.1, 0.15$. In mass springs, the mass of the particle $m = 0.5$, the spring force coefficient $k = 2$. The friction coefficients are chosen in $0, 0.05, 0.1, 0.15$. For each of the three physics, we included three dynamic settings in meta-training and left out one in meta-testing.

For the turbulent flow dataset, we generated the tasks as given by the instructions and scripts found within the official code repository from [121], `https://github.com/Rose-STL-Lab/Dynamic-Adaptation-Network`. We split the seen and unseen buoyancy factors according to the same task split used within their work and directly compare RMSE values based on their implementation and magnitude coefficients.

For cardiac electrical dynamics, we generated 3D electrical signal propagation in the heart simulated by the Aliev-Panfilov model [1] on 3 heart meshes and a total of 12 different tissue parameters (4 on each heart) representing different injury to the heart muscle. This was treated as 12 tasks in meta-learning. All 12 tasks appeared in meta-training and -testing, with disjoint time sequences resulting from different external stimulations (meta-training: 300; meta-test: 2,020).