On Learning and Generalization to Solve Inverse Problem of Electrophysiological Imaging

by

Sandesh Ghimire

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
October 2020

On Learning and Generalization to Solve Inverse Problem of Electrophysiological Imaging

by

Sandesh Ghimire

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. Nathan Cahill — Date
Dissertation Committee Member

Dr. Rui Li — Date
Dissertation Committee Member

Dr. Niels Otani — Date
Dissertation Committee Member

Dr. David A. Borkholder — Date
Dissertation Defense Chairperson

Certified by:

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

On Learning and Generalization to Solve Inverse Problem of Electrophysiological Imaging

by

Sandesh Ghimire

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

In this dissertation, we are interested in solving a linear inverse problem: inverse electrophysiological (EP) imaging, where our objective is to computationally reconstruct personalized cardiac electrical signals based on body surface electrocardiogram (ECG) signals. EP imaging has shown promise in the diagnosis and treatment planning of cardiac dysfunctions such as atrial flutter, atrial fibrillation, ischemia, infarction and ventricular arrhythmia.

Towards this goal, we frame it as a problem of learning a function from the domain of measurements to signals. Depending upon the assumptions, we present two classes of solutions: 1) Bayesian inference in a probabilistic graphical model, 2) Learning from samples using deep networks. In both of these approaches, we emphasize on learning the inverse function with good generalization ability, which becomes a main theme of the dissertation. In a Bayesian framework, we argue that this translates to appropriately integrating different sources of knowledge into a common probabilistic graphical model framework and using it for patient specific signal estimation through Bayesian inference. In learning from samples setting, this translates to designing a deep network with good generalization ability, where good generalization refers to the ability to reconstruct inverse EP signals in a distribution of interest (which could very well be outside the sample distribution used during training). By drawing ideas from different areas like functional analysis (e.g. Fenchel duality), variational inference (e.g.

Variational Bayes) and deep generative modeling (e.g. variational autoencoder), we show how we can incorporate different prior knowledge in a principled manner in a probabilistic graphical model framework to obtain a good inverse solution with generalization ability. Similarly, to improve generalization of deep networks learning from samples, we use ideas from information theory (e.g. information bottleneck), learning theory (e.g. analytical learning theory), adversarial training, complexity theory and functional analysis (e.g. RKHS). We test our algorithms on synthetic data and real data of the patients who had undergone through catheter ablation in clinics and show that our approach yields significant improvement over existing methods. Towards the end of the dissertation, we investigate general questions on generalization and stabilization of adversarial training of deep networks and try to understand the role of smoothness and function space complexity in answering those questions.

We conclude by identifying limitations of the proposed methods, areas of further improvement and open questions that are specific to inverse electrophysiological imaging as well as broader, encompassing theory of learning and generalization.

Acknowledgments

Life is but a collection of experiences.

I have beautiful memories of the time in Rochester as a PhD student at RIT. I am grateful to the members with whom I shared this beautiful part of my life and my PhD is complete because of all of them. I take this opportunity to thank all of them.

First and foremost, I am grateful to my wonderful advisor, Prof. Linwei Wang. I feel extremely lucky to work under her advisement. She is an excellent advisor and has helped me academically, personally and professionally. More importantly, she allowed me to explore and understand my intellectual curiosity, and my true nature, so that I could follow my passion, realize my potential and weaknesses and create my own path in life. I enjoyed juggling and wrestling with a wide variety of ideas, in breadth and in depth, and I enjoyed discussing them in great detail with Prof Wang. I fully enjoyed research. It would not have been possible without full support of my advisor.

I am very thankful to Prof. Shi for his wise suggestions and guidance, both personal and academic. I am grateful to my PhD committee members, Prof. Nathan Cahill, Prof. Niels Otani and Prof. Rui Li for their feedback, wonderful discussion and guidance. I will especially remember deep discussions about the implications of several mathematical results in deep learning with Prof. Nathan Cahill, and his elaborate comments on my writing which helped me improve it. I am grateful to Prof. Rui Li for appreciating my work and encouraging me right from the time of RPA until my PhD defense. Similarly, Prof. Niels Otani was very supportive and enthusiastic, especially about the electrophysiology and its modeling. It always encouraged me and boosted my energy. I am thankful to Prof. Borkholder for smoothly conducting the PhD defense.

I would like to thank my labmates: Jwala, Prashnna, Pradeep, Nilesh, Xiajun, Zhiyuan, Omar, Mohammed, Maryam, Shakil, Roland for all the interesting discussions, reading sessions, Q&A, collaborations and numerous occasions of solving problems together. I would also like to remember Kishan who used to sit near and spark many interesting discussions.

I would also like to express my gratitude to IBM team. I am thankful to Dr. Tanveer, Dr. Mehdi, Dr. Satyananda, Dr. Alex, Dr. Ken, Dr. Joy and whole IBM Medical Sieve Radiology Group for the wonderful opportunity to work in the real world problem, for the guidance and mentorship. I also remember fellow interns at that time.

The circle of friends who made the Rochester life fun, easy and intellectually stimulating have played a major role in my successful completion of PhD. I am very much thankful to Shusil and Utsav for their help and guidance. I would like to thank Sushant, Prashnna, Aayush, Kishan, Manoj, Pradeep, Nikhil for creating an intellectually stimulating and alive environment. I am thankful to Jwala and Kushal for all the help they provided from the start of my PhD. I want to express gratitude to Krishna, Tejan, Dilip, Prasanna, Deep, Shradha, Laxmi, Bhawani, Sunita, Akhter and others for the supportive and cordial environment. I am thankful to everyone for helping out each other, most of all to get through everyday challenges of being a student and an independent adult.

Last, but most importantly, I would like to remember and be thankful to my parents, Krishna Kumar Ghimire and Dipa Kumari Ghimire for their love, care and instilling in me a habit to dream and work hard for it. I want to remember my brother, Sandip and my girlfriend, Binita for their love and support.

To the love, sacrifice and dream of my parents.

Author Publications

Following is the list of refereed publications of the author. * indicates that some part of the publication is included in the dissertation.

1. * **Ghimire, S.**, Gyawali, P. K., Wang, L. Reliable Estimation of Kullback–Leibler Divergence by Controlling Discriminator Complexity in the Reproducing Kernel Hilbert Space. *arXiv preprint arXiv:2002.11187.* Under Review

2. ***Ghimire S**, Gyawali PK, Dhamala J, Sapp JL, Horacek M, Wang L. Improving Generalization of Deep Networks for Inverse Reconstruction of Image Sequences. In *International Conference on Information Processing in Medical Imaging* 2019 Jun 2 (pp. 153-166). Springer, Cham.

3. * **Ghimire S**, Sapp JL, Horáček BM, Wang L. Noninvasive Reconstruction of Transmural Transmembrane Potential With Simultaneous Estimation of Prior Model Error. *IEEE Transactions on Medical Imaging.* 2019 Mar 20;38(11):2582-95.

4. * **Ghimire S**, Dhamala J, Gyawali PK, Sapp JL, Horacek M, Wang L. Generative Modeling and Inverse Imaging of Cardiac Transmembrane Potential. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* 2018 Sep 16 (pp. 508-516). Springer, Cham.

5. * **Ghimire S**, Wang L. Deep Generative Model and Analysis of Cardiac Transmembrane Potential. In 2018 *Computing in Cardiology Conference (CinC)* 2018 Sep 23 (Vol. 45, pp. 1-4). IEEE.

6. * **Ghimire S**, Sapp JL, Horacek M, Wang L. A Variational Approach to Sparse Model Error Estimation in Cardiac Electrophysiological Imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* 2017 Sep 10 (pp. 745-753). Springer, Cham.

7. Jiang X, **Ghimire S**, Dhamala J, Li Z, Gyawali PK, Wang L. Learning Geometry-Dependent and Physics-Based Inverse Image Reconstruction.
   Accepted in *MICCAI 2020.*

8. Gyawali PK, **Ghimire S**, Bajracharya P, Li Z, Wang L. Semi-supervised Medical Image Classification with Global Latent Mixing. Accepted in *MICCAI 2020.* arXiv:2005.11217. 2020.

9. Gyawali P, Li Z, Knight C, **Ghimire S**, Horacek BM, Sapp J, Wang L. Improving Disentangled Representation Learning with the Beta Bernoulli Process. In 2019 *IEEE International Conference on Data Mining (ICDM)* 2019 Nov 8 (pp. 1078-1083). IEEE.

10. Gyawali PK, Li Z, **Ghimire S**, Wang L. Semi-supervised Learning by Disentangling and Self-ensembling over Stochastic Latent Space. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* 2019 Oct 13 (pp. 766-774). Springer, Cham.

11. Dhamala J, **Ghimire S**, Sapp JL, Horáček BM, Wang L. 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* 2019 Oct 13 (pp. 458-467). Springer, Cham.

12. Dhamala J, **Ghimire S**, Sapp JL, Horáček BM, Wang L. 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* 2018 Sep 16 (pp. 499-507). Springer, Cham.

13. **Ghimire S**, Wang L. L0 Norm Based Sparse Regularization for Non-invasive Infarct Detection using ECG Signal. In 2017 *Computing in Cardiology (CinC)* 2017 Sep 24 (pp. 1-4). IEEE.

14. **Ghimire S**, Dhamala J, Coll-Font J, Tate JD, Guillem MS, Brooks DH, MacLeod RS, Wang L. Overcoming Barriers to Quantification and Comparison of Electrocardiographic Imaging Methods: A Community-based Approach. In 2017 *Computing in Cardiology (CinC)* 2017 Sep 24 (pp. 1-4). IEEE.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Dubito, ergo cogito, ergo sum.
(I doubt, therefore I think, therefore I am.)
- René Descartes

In this dissertation, we consider an inverse problem of estimating cardiac electrical signals from body surface electrocardiograms. We look at this problem from the perspective of learning theory and 1) propose several approaches to solve the problem, 2) ask the fundamental questions about the problem and algorithms. Although we are firmly grounded in this inverse problem, the fundamental questions raised while solving the problem has lifted the problem to a more general nature and has helped us answer some questions related to the learning theory itself which we discuss at the latter portion of the thesis.

1.1 Motivating Problem

The heart is an electromechanical system. Rhythmic contraction of the heart is induced by coordinated electrical propagation throughout the heart. Compared to advances

in imaging technologies for cardiac structures, however, there is inadequate progress in our ability to observe electrical activity of the heart. Current clinical practice to assess individual's cardiac electrophysiology is mainly restricted to either remote body surface electrocardiograms (ECGs), or invasive catheter mapping on the heart surface (epicardium and endocardium) with limited spatial resolution.

Computational electrophysiological (EP) imaging aims to fill this gap by computationally reconstructing subject-specific cardiac source dynamics from noninvasive ECG. It has shown promise in the diagnosis of cardiac dysfunctions such as atrial flutter [89], atrial fibrillation [26], ischemia [70], infarction [107] and ventricular arrhythmia [106]. However, computational EP imaging is an ill posed inverse problem. The source of this ill-posedness is the fact that ECG is an integral effect of all the electrical sources inside heart. If we seek solution throughout heart transmurally, the source of electrical activity is distributed transmurally throughout the heart and due to the law of electromagnetism, different configurations of cardiac electrical sources could result in the same ECG observation on the body surface, making the inverse estimation ill posed [18, 85]. This problem is further exacerbated by the lack of sufficient measurements of ECG on the body surface. Solving this inverse problem to perform EP imaging is the main goal of this dissertation.

It is clear that solving for the EP signals based solely on the ECG data is impossible, especially in cases where we are looking for the transmural source. Luckily, we have other sources of knowledge. For example, we know about the physiology of human hearts. We have rich but general knowledge about the pattern of electrical signal propagation over time and throughout the heart, although specific propagation pattern might be different among different individuals. Similarly, the cardiac electrical field propagates in a manner similar to a wavefront propagation and therefore spatial gradient of electrical voltage is sparse at a time instant. We would benefit immensely by integrating these knowledge while solving the inverse problem.

In this dissertation, we want to develop a principled way to look at the inverse problem. Towards that goal, we view the inverse problem as a learning problem. From this perspective, the goal is to learn a function that maps an ECG measurement to the

cardiac electrical signal. Then, we propose two types of methods to learn the inverse function: 1) Inference in a probabilistic graphical model (PGM), 2) Learning from the samples. By using rich literature of Bayesian Inference, Learning Theory and Machine Learning in general, we propose several methods to solve the inverse problem. In this process, we answer some fundamental questions related to algorithms and learning in general while we leave some as open questions.

1.2 Mathematical Formulation

We start with a general setup of inverse problem and specialize it to inverse EP imaging whenever appropriate. Let $\mathcal{X}$ and $\mathcal{Y}$ be, respectively, the signal domain and the measurement domain in a general inverse problem. In the inverse EP imaging, $\mathcal{X}$ represents the domain of cardiac electrical signal and $\mathcal{Y}$ denotes the ECG measurement domain. There exists a certain model, deterministic or probabilistic, that maps each $x \in \mathcal{X}$ to $y \in \mathcal{Y}$. In a deterministic setting, we denote this relation by a function g such that $y = g(x)$. In a fully probabilistic setting, we denote this relation by a conditional distribution $p_{\mathcal{Y}|\mathcal{X}}(y|x)$. Then, framed as a learning problem, inverse problem is the problem of obtaining the inverse conditional relationship $p_{\mathcal{X}|\mathcal{Y}}(x|y)$. These ideas are pictorially illustrated in Fig. (1.1). We can simplify the problem to the task of obtaining a deterministic inverse function, f which maps y to x. In a unified way, we can connect the deterministic prediction and probabilistic prediction as :

$$f(y) = \int x p_{\mathcal{X}|\mathcal{Y}}(x|y) dx \tag{1.1}$$

i.e. the deterministic inverse mapping is the mean of the probabilistic conditional distribution.

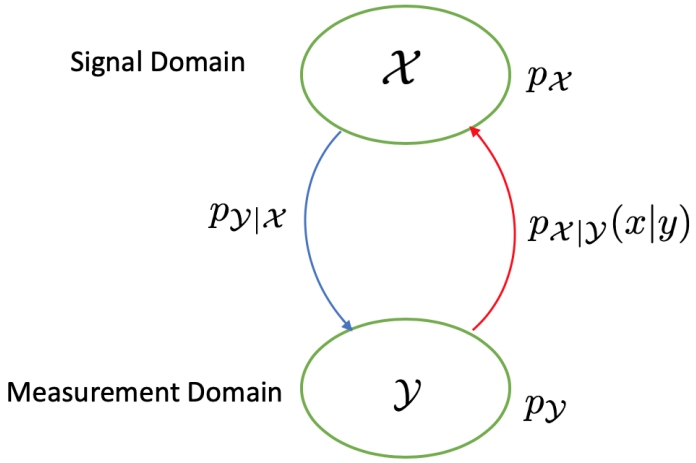


Figure 1.1: A high level representation of the inverse problem

.

1.3 Learning and Generalization

There are two key questions: 1) How to solve this inverse problem framed as learning problem? 2) What does it mean to solve an inverse problem? Following two subsections are dedicated to elaborate how we answer these questions.

1.3.1 Learning

To solve the learning problem, we approach from two general perspectives: 1) Using probabilistic graphical models and inference, 2) Learning directly from samples using deep networks. These two learning approaches have been depicted in Fig. 1.2.

Inference in Probabilistic Graphical Model

In this approach, we create a probabilistic graphical model, directed graphical models [11], to be precise. As shown in Fig.1.2(A), the signal becomes the random variable

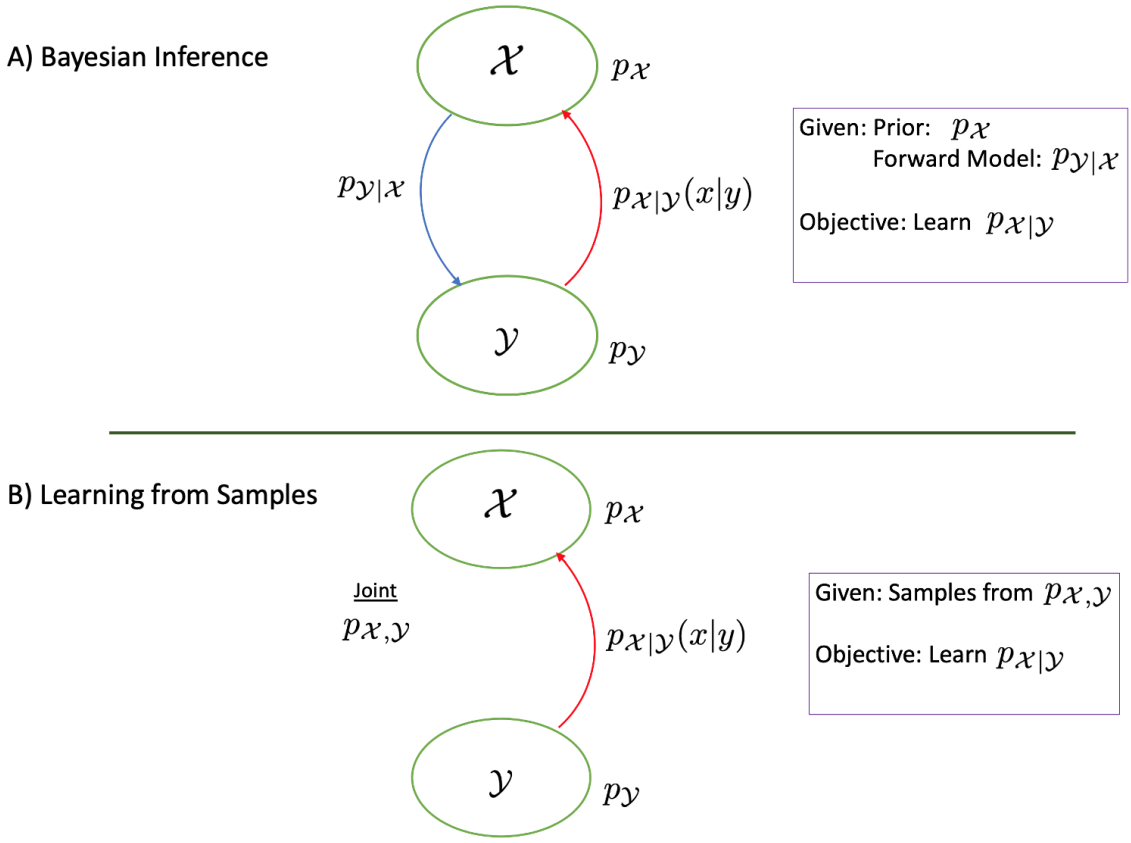


Figure 1.2: Pictorial representation of two ways of solving the inverse problem. Essential difference lies in defining the joint distribution and algorithm used to obtain the conditional distribution

.

x with its domain $\mathcal{X}$. The the prior distribution, $p_{\mathcal{X}}(x)$ encodes our belief we want to enforce about the signal. The forward model, is incorporated into the likelihood function $p_{\mathcal{Y}|\mathcal{X}}(y|x)$ meaning the y that are consistent with the forward model are more likely. We incorporate our bias and prior knowledge through the modeling of likelihood and prior distribution and carry out Bayesian inference to estimate the inverse conditional distribution, $p_{\mathcal{X}|\mathcal{Y}}(x|y)$, which is what we want. This general framework is always the same in using PGM and inference. We usually have freedom in defining relations through graphical model, choice of conditional distributions and prior distributions

where we can inject useful prior knowledge as bias.

Learning from Samples

Fig.1.2(B) shows another general method of learning the inverse solution. In this approach, we are interested in designing a learning algorithm which takes in samples, $\{(x_i, y_i)\}$, from the joint distribution $p(x, y)$ and directly outputs a conditional distribution $p_{\mathcal{X}|\mathcal{Y}}$. Comparing with the inference in PGM, we could think of this method as directly computing required posterior distribution, where the notion of prior and likelihood are only implicit. Correspondingly, we do not utilize knowledge in the form of prior or likelihood distributions. Nor do we enjoy the benefit of encoding human knowledge about relationship between known random variables into the graphical model. Rather, we try to learn everything in an automatic way using pairs of data in the heart and observed on the body surface. We still use information of physics and physiology, but we do so by feeding physics-based simulation data to the algorithm. In a way, this approach provides us more flexibility but less structure in our learning.

1.3.2 Generalization

Suppose we have found a solution using one of the approaches; how do we evaluate the solution? We propose to do so with the generalization ability of the solution. Since we have two general approaches to obtain the inverse solution, the definition of generalization also takes different form in each approach. This is particularly true because in probabilistic model, we do not have a training and test sets as in typical machine learning. Therefore, the ability to generalize refers to how good the solution is in average when it is measured throughout the distribution of interest. However, in learning from samples, we have training and test set and we need to compare the performance in those two sets to understand generalization.

Inference in Probabilistic Graphical Model

In this case, good generalization means that the statistical inference yields good solution in all test cases. Since there is no training set, all the samples from distribution of interest are like the test cases. To concretely explain this notion, we we define the following quantity as the goodness:

$$Goodness = E_{p(x,y)}[-\ell(f(y), x)] \tag{1.2}$$

where $\ell : \mathcal{X} \times \mathcal{X} \rightarrow R$ measures the discrepancy between the prediction of the learnt function, i.e., $f(y)$ and the ground truth signal value x. Let us call the distribution where we evaluate the inverse function as the 'distribution of interest' and denote it by $p^*(x, y)$. It is reasonable to define generalization to be the goodness measure in the distribution of interest $p^*(x, y)$, i.e. replacing $p(x, y) = p^*(x, y)$ in eq.(1.2). We say that the inverse solution has good generalization if $-\ell(f(y), x)$ is high throughout the distribution of interest regardless of how the inverse solution was obtained. In a Bayesian setting, generalization corresponds to average accuracy during testing since there is no training set.

To obtain good solution in all test cases, we need to overcome the lack of information in the forward model, which is achieved by integrating information from different sources using probabilistic graphical model. Later, we show that this is possible by appropriate modeling of information through prior distribution and hierarchical representation in graphical model. One key challenge in solving for the inverse solution using graphical model and inference is to be able to adapt a common prior knowledge to different patients. We address this problem using hierarchical graphical model as a prior and adapting hyper-parameters.

Learning from Samples

In learning function from samples, we take supervised learning approach where we reserve a set of data to learn the inverse function and other set of data to test how well

the learnt function generalizes to the test data. Therefore, we differentiate between the goodness in the training and the whole distribution of interest or just the test distribution not used in training. In this setup, we often define generalization gap as the difference between the two goodness measures as:

$$Generalization \quad gap = E_{p_{train}(x,y)}[-\ell(f(y), x)] - E_{p^*(x,y)}[-\ell(f(y), x)] \tag{1.3}$$

The definition of generalization gap in this sense matches that in the machine learning literature and the smaller the generalization gap, the better. Improving generalization requires us to think from the perspective of learning theory, sampling and complexity and answer some fundamental questions.

1.4 Research Questions and Contributions

To solve the inverse problem of EP imaging, we intend to design learning algorithms that would yield an inverse function from ECG data to the cardiac electrical signal while maintaining good generalization ability.

1.4.1 Inference in Probabilistic Graphical Inference

In the first approach, we argue that good generalization can be achieved if we can adapt the population knowledge by integrating it with the data using a probabilistic framework and performing Bayesian inference. But, this adaptation is challenging, especially if we have to do it for each personalized case. That brings us to our first key question.

Q. How can generalization be improved in a PGM and inference framework by facilitating adaptation of prior knowledge for each personalized case?

Looking at a finer scale helps us divide this question into sub-questions which we answer in next chapters. The physiological knowledge like the general shape of the

TMP or differential equation guiding propagation of TMP is rich and has been popular in inverse EP imaging. To apply this knowledge for the personalized estimation is difficult because good physiological models come with a set of parameters that differ among individuals, pathological conditions and tissue properties, about which we do not have prior knowledge. Fixing these parameter values to standard values might introduce model error while their simultaneous adaptation is challenging due to their high dimensionality and complex relation with cardiac electrical signal. Hence, it is challenging to generalize this knowledge to patient specific inference. This brings us to our first set of research questions:

Q. 1.a) How can the population knowledge be adapted for patient specific inference?
Q. 1.b) How can the prior knowledge about the sparsity in the gradient domain and dynamics of TMP signals be combined in principled way?

Joint inference of error and signal:
To answer these research questions, we propose principled probabilistic modeling of the electrophysiology as well as error that might be introduced into the EP model. To address the challenge due to introduction of additional unknown random variable, we add knowledge about the sparsity of this error random variable in the gradient domain using heavy tail distributions and variational approximations. After we have incorporated different sources of information by exploiting the hierarchical representation ability of the probabilistic graphical methods, we perform inference of all unknown random variables at the presence of ECG data. To overcome difficulty in tractability of posterior distribution, we use variational posterior distribution as well as expectation maximization to jointly learn some estimates and posterior distribution. By using Fenchel duality, we also introduce tractability of Gaussian distribution. All in all, the inference amounts to iterative update of distribution parameters associated with the unknown random variables. At the end, we obtain a learning algorithm of iterative nature which yields an estimate of distribution of TMP given ECG data. We describe the details in chapter 4.

Moving a little bit further towards data-driven approach and circumventing the issues

in adapting population knowledge using traditional representation (like simulation dynamic model), we ask if we can use an alternative representation to describe generation of EP signals such that the representation would also help in simultaneous adaptation during inference. This brings us to our second research question.

Q. 2. How can we improve generalization with an alternative representation of prior knowledge such that it allows efficient inference?

Deep generative model prior:
To answer this question, we take advantage of the recent breakthrough in deep learning. First we use a variational autoencoder (VAE) to learn a generative model of transmembrane potential (cardiac electrical signal), TMP in an unsupervised way. This way we learn a distribution of the latent generative factor, z, and conditional dependence between latent factor and TMP signal, x, *i.e.* $p(x|z)$. Once we have trained VAE, we then use conditional relation between latent factor and TMP, $p(x|z)$, as conditional prior of TMP. The machine-learnt functions describing relationships between random variables are expressive and amenable to inference. We show that we can perform inference in the resulting graphical model by using expectation maximization and exploiting gradient descent feature of deep network. This is described in chapter 5.

1.4.2 Learning from Samples

In the second half of this dissertation, we reach the fully data-driven machine learning approach; i.e., our inverse function relies solely on data samples. Here, we are interested in finding an algorithm that takes in a large number of ECG-TMP pairs and gives out a good estimate of conditional distribution of TMP signals on ECG in both training and test set. Typically, a neural network is used to parameterize the conditional distribution. So, we are interested in understanding the generalization ability of such a neural network function as an inverse solution. We again pose question about the generalization ability of the inverse solution as follows:

Q. How can we learn from samples an inverse EP imaging function that can

also generalize well outside sample distribution?

We go one step deeper into this question. Based on the theory of learning, we identify two independent factors that affect the generalization in the purely sample based approach: 1) possibility of shift in training and test distribution of the input space, 2) smoothness and regularity properties of the conditional distribution. These two factors are somewhat orthogonal to each other. The former is related to the distribution of the measurement space, $\mathcal{Y}$ while the latter is related to regularity property of the inverse function. Correspondingly, we have finer level questions:

Q. 3. a) How can we understand and improve generalization when there is possibility of shift in training and test distribution?

Q.3. b) How can we understand and quantify the role of smoothness and regularity properties of the neural networks regarding generalization?

Learning an invariant representation:
To answer the first question, we introduce the idea of learning an invariant representation. In order to achieve good generalization to the test set that is shifted from the training set, we propose to learn a representation that is invariant to the shift in those distributions. We proposed two ways of enforcing invariance: 1) adversarial training, and 2) learning minimal, sufficient representation using information bottleneck principle. The details are described in chapter 7.

Role of smoothness:
To answer the second question and formally treat the role of smoothness and regularity properties of neural networks, we apply the notion of variation from Analytical Learning Theory. We show theoretically that introducing stochasticity in the latent representation reduces the variation of the decoder which helps in learning functions with good generalization properties. Experimentally we show that the generalization ability of a neural network can be improved by using a stochastic latent space and employing the information bottleneck principle to learn a minimal, sufficient representation. We derive the variational lower bound of the information bottleneck objective

as the loss function which is easily optimized with respect to the parameters of the neural network by using stochastic gradient descent. We support our arguments with carefully designed experiments. The details of this approach are described in chapter 6.

Complexity and control:
Continuing further in this direction, we seek to investigate the role of smoothness regarding generalization of neural networks. We seek to understand how to quantify the degree of smoothness and precisely how it is connected to the generalization ability. Our last contribution is only a precursor in this direction. We establish that the notion of complexity could be a good measure of smoothness. We also show that minimizing this complexity measure eradicates a pathology of generative adversarial networks (GANs) by stabilizing the training. This is described in chapter 8.

Below is a graphical overview and organization of the contributions in this dissertation.

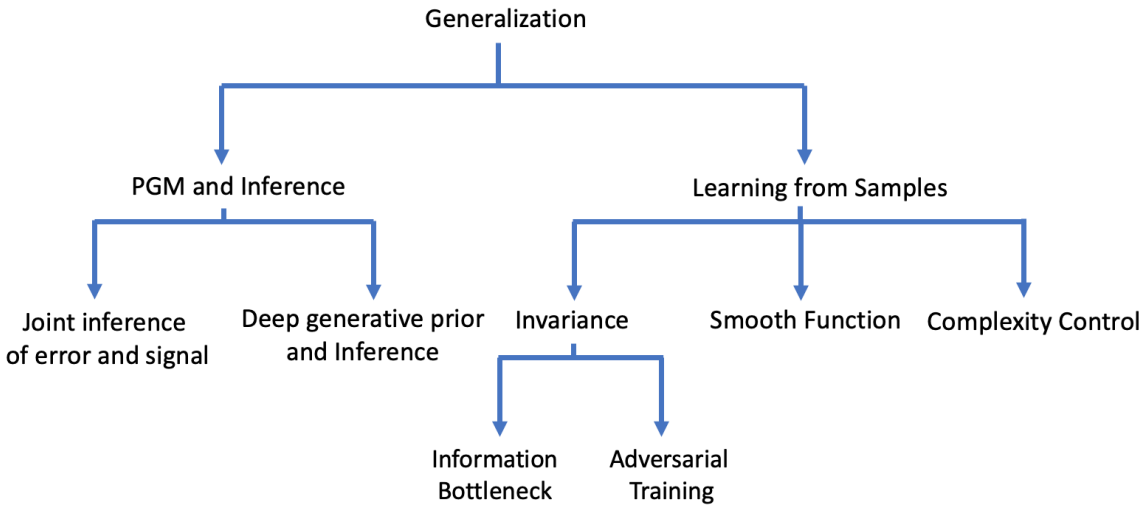


Figure 1.3: Overview of dissertation

.

Chapter 2

Foundation Literature

If I have seen further than others, it is by standing upon the shoulders of giants.
- Issac Newton

2.1 Probabilistic graphical model and Inference

2.1.1 Introduction

Probabilistic graphical models are diagrammatic representation of the joint distributions of random variables in the form of a graph. Through this diagram, it is easy to express the relation between different random variables thereby providing a structure to the dependence between different random variables. Simple inspection of the graph can provide insights, for example of conditional independence between different random variables. Complex manipulations required to perform inference can be understood as graphical manipulations [11].

By the definition, a graphical model (see Fig. 2.1) is a graph containing nodes and edges. The nodes represent random variables while the edges represent relations be-

tween random variables. These graphical models can be divided into directed and undirected depending on the the presence or absence (respectively) of the directionality of the edge in the graph. There is one more type of graphical model known as factor graph which contains a relation node between random variable nodes. This factor graph, therefore, generalizes both the directed and undirected graphs and posses unique ability to construct graphical model with hybrid structure: containing a subgraph as directed and another subgraphs as undirected graph.

2.1.2 Directed Graphical Models and Inference

A directed graphical model a directed graph with nodes as the random variables and directed edges encoding the conditional dependence. This graphical model provides information about the generative process and dependence. Using rules like D-separation [84], we can identify independence through graphical model. These knowledge can be used to quickly write down the joint distribution from the graphical model.

Expectation Maximization

Consider a graphical model as shown in Fig. 2.1(a). Suppose our objective is to maximize $p(\mathbf{y}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. It might be difficult to integrate out the latent variable $\mathbf{z}$. In such cases, we first note that maximizing $p(\mathbf{y}|\boldsymbol{\theta})$ is same as maximizing $\log p(\mathbf{y}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Then we decompose $\log p(\mathbf{y}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ as follows:

$$\log p(\mathbf{y}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q||p) \tag{2.1}$$

$$\text{where } \mathcal{L}(q, \theta) = \int q(\mathbf{z}) \log(\frac{p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})}) \tag{2.2}$$

$$KL(q||p) = \int q(\mathbf{z}) \log(\frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})}) \tag{2.3}$$

To maximize the likelihood $p(\mathbf{y}|\boldsymbol{\theta})$, first the KL divergence is minimized by setting $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})$, then eq. (2.2) is maximized with respect to $\boldsymbol{\theta}$ by maximizing $\int q(\mathbf{z}) \log p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ with the updated $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})$. Therefore,

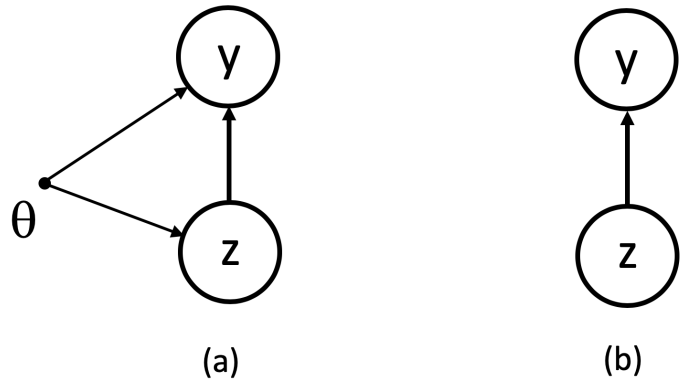


Figure 2.1: z and y are random variable while θ is a parameter in (a) and no parameter in (b). Typically, 'parameter' is reserved when we intend to infer only a point estimate, whereas by a circular node, we typically represent a random variable whose distribution is of interest. The graphical model in (a) immediately enables us to decompose distribution in the following way: $p(y, z|\theta) = p(y|z, \theta)p(z|\theta)$

.

expectation maximization [28] is a procedure to alternatively update posterior of the latent random variable and the value of parameter to be optimized with respect to. Also, note that this procedure is useful when we have a situation where it is difficult to integrate out the latent variable, but it is relatively simple to find the posterior of the latent variable z given parameter and data y, *i.e.,* $p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})$. In many situations, the posterior $p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})$ is also intractable. In such cases, we can use other methods to obtain $q(\mathbf{z})$ as a close approximation of $p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})$ instead of using the exact posterior. This is the strategy we use later for our inference.

Variational Bayes

To understand variational Bayes, we imagine a generative process where $\mathbf{y}$ is generated by $\mathbf{z}$ as shown in Fig.2.1(b). Our objective is to obtain a posterior distribution $p(\mathbf{z}|\mathbf{y})$.

Similar to expectation maximization, we can decompose log likelihood as follows:

$$\log p(\mathbf{y}) = \mathcal{L}(q) + KL(q||p) \tag{2.4}$$

$$\text{where } \mathcal{L}(q) = \int q(\mathbf{z}) \log(\frac{p(\mathbf{y}, \mathbf{z})}{q(\mathbf{z})}) \tag{2.5}$$

$$KL(q||p) = \int q(\mathbf{z}) \log(\frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{y})}) \tag{2.6}$$

Note that the decomposition is similar to EM above, except for the parameter $\boldsymbol{\theta}$. Here, all the parameters are absorbed into the variable $\mathbf{z}$ and we want joint posterior of all of them given $\mathbf{y}$. Unlike before, our objective here is to find $q(\mathbf{z})$ as close to posterior as possible. Therefore, we are interested in minimizing eq.(2.6). Since $\log p(\mathbf{y})$ is constant in eq. (2.4), we can equivalently maximize $\mathcal{L}(q)$, also known as evidence lower-bound (ELBO), with respect to q. The distribution $q(z)$ is the variational distribution and therefore this type of procedure to obtain posterior distribution is called Variational Bayes.

Later we will talk about large scale method to minimize eq.(2.5) using a so called recognition neural network. Here, we will discuss about a very popular method to obtain approximation of the posterior by using mean field approximation. Minimization of eq.(2.5) with respect to arbitrary distribution, q is very difficult. Therefore, we need to make further assumptions to restrict the class of distribution on which we can minimize for the posterior distribution. Mean field approximation refers to the assumption that the posterior $q(\mathbf{z})$ is independent in its components, i.e. $q(\mathbf{z}) = \prod_i q(\mathbf{z}_i)$. With this assumption, we can simplify optimization of eq.(2.5) to simple computation of expectations as follows:

$$\begin{aligned}
\mathcal{L}(q) &= \int \prod q_i \Big[\log p(\mathbf{y}, \mathbf{z}) - \sum_i \log q_i \Big] d\mathbf{z} \\
&= \int q_j \Big[\int \log p(\mathbf{y}, \mathbf{z}) \prod_{i \neq j} q_i d\mathbf{z}_i \Big] d\mathbf{z}_j - \sum_j \int q_j \log q_j d\mathbf{z}_j \\
&= \int q_j \log \widetilde{p}(\mathbf{y}, \mathbf{z}_j) d\mathbf{z}_j - \int q_j \log q_j d\mathbf{z}_j - \sum_{i \neq j} \int q_i \log q_i d\mathbf{z}_i + const.
\end{aligned} \tag{2.7}$$

where $\log \widetilde{p}(\mathbf{y}, \mathbf{z}_j) d\mathbf{z}_j = \int \log p(\mathbf{y}, \mathbf{z}) \prod_{i \neq j} q_i d\mathbf{z}_i = E_{i \neq j}[\log p(\mathbf{y}, \mathbf{z})]$. Our objective is to maximize $\mathcal{L}$ in eq.(2.7) with respect to q. If we fix all the q_is except q_j, we can see

that eq.(2.7) can be easily maximized by setting $q_j = \widetilde{p}(\mathbf{y}, \mathbf{z}_j)$ because the first two terms produce negative of KL divergence. Thus, we can obtain posterior distribution satisfying mean field approximation by simply updating posterior distribution in each component as follows:

$$\log q_j(\mathbf{z}_j) = E_{i \neq j}[\log p(\mathbf{y}, \mathbf{z})] + const. \tag{2.8}$$

Variational Bayes is most effective in cases where the posterior is known to be of certain form. For example, any distribution with only quadratic terms in the exponent must be a Gaussian distribution. This can be achieved by using conjugate priors in the Bayesian inference. In such cases, variational Bayes corresponds to the updates of the parameters of distributions q_j. In a directed graphical model, this corresponds to the update of parameters of each node sequentially and in a cyclic manner.

Other methods

Other popular methods of inference are sampling based methods and neural network based methods. Sampling based methods require some form of sampling from the joint distribution and yields samples from the posterior distribution. These methods are effective if we need samples from the posterior or if we need to compute expectation with respect to the posterior distributions. Some commonly used sampling methods for the inference are Markov chain Monte Carlo(MCMC) and its variants, Gibbs sampling, slice sampling and other hybrid methods [11].

With the recent breakthrough in deep learning, many algorithms have emerged combining ideas from variational inference and sampling methods. Some recent works in this direction are variational autoencoder [58], adversarial variational Bayes [75] , variational inference with normalizing flow [90], inverse autoregressive flow [59], etc.

2.2 Deep Generative Models

Deep generative models loosely refer to the graphical models with deep neural connections. Consequently, samples from complex distributions can be generated by using trained neural networks. Deep belief networks, deep Boltzmann machine, variational autoencoder and generative adversarial networks are examples of deep generative models [42]. We briefly review variational autoencoder (VAE) as it is the most relevant one for our work.

2.2.1 Variational Autoencoder

Suppose our objective is to obtain a posterior distribution $p(\mathbf{z}|\mathbf{y})$ in the graphical model Fig. 2.1(b). As described earlier in the Variational Bayes section, we can do so by maximizing evidence lower-bound (ELBO) $\mathcal{L}$ with respect to a variational distribution $q(\mathbf{z}|\mathbf{y})$. Unlike before, in a deep generative model, the variational posterior approximation is parameterized with neural network and we maximize $\mathcal{L}$ in eq. (2.5) with respect to parameters of the neural network. This network is also called recognition network. Unlike variational inference, however, a variational autoencoder is primarily concerned with autoencoding. Therefore, both the conditional distributions $p(\mathbf{y}|\mathbf{z})$ and $q(\mathbf{z}|\mathbf{y})$ are assumed unknown and are parameterized with neural networks. Then, we maximize $\mathcal{L}$ with respect to both of these network parameters. With these parameterization, we can write the ELBO, $\mathcal{L}$ as:

$$\mathcal{L}(q) = E_{q_\phi(\mathbf{z}|\mathbf{y})}\big[\log p_\theta(\mathbf{y}|\mathbf{z})\big] - KL(q_\phi(\mathbf{z}|\mathbf{y})||p(\mathbf{z})) \tag{2.9}$$

where θ and ϕ represent parameters of neural networks. The recognition network is called encoder and likelihood network is called the decoder. The first term in the ELBO (eq.(2.9)) corresponds to reconstruction of $\mathbf{y}$ by first passing through the encoder and then through the decoder. This resembles the loss in a traditional autoencoder. In the VAE, however, we have an additional second term which behaves as a regularization term corresponding to minimization of the KL divergence between the posterior and the

isotropic Gaussian distribution. This term tries to make the variance of the posterior distribution high (one) as much possible.

Chapter 3

Background and Related Works

But although all our knowledge begins with experience,
it does not follow that it arises from experience.
- Immanuel Kant

3.1 Electrophysiological imaging

The heart is divided into four chambers: left and right atrium and left and right ventricles. Right atrium receives deoxygenated blood from the body through veins and pass it to the right ventricle which pumps blood to the lungs, where blood is oxygenated. The oxygenated blood is collected at left atrium which is then passed, through mitral valve, to the left ventricle which pumps it throughout the body. This constitutes a cycle. The heart is an electromechanical system. The electrical conduction system plays crucial role in the mechanical contraction and expansion of the cardiac muscles.

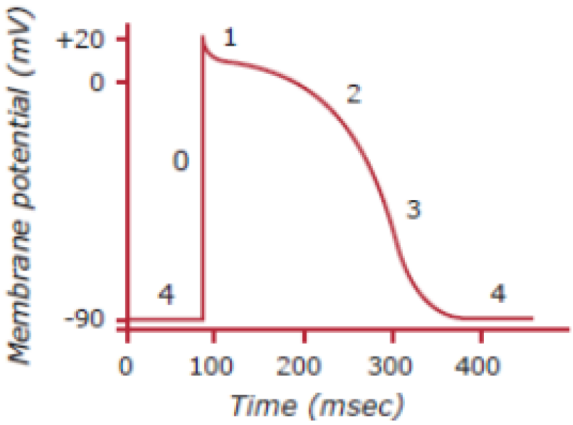


Figure 3.1: Schematic diagram of transmembrane potential (TMP)

.

3.1.1 Cardiac Electrophysiology

Ion channels embedded within the cellular membrane of cardiac muscle cells facilitate the propagation of cardiac electrical signals. The membrane current resulting from the membrane conductance change produces potential difference across the cell membrane (between intracellular and extracellular space), known as the cardiac transmembrane potential (TMP) or action potential. At rest, this potential difference maintains approximately 90mV and changes to about +30mV under large electrical stimulus, called depolarization. TMP remains at this high voltage stage for a while (plateau) before returning to the resting state (repolarization) (Fig 3.1). Furthermore, diffusion of action potential between the cells allows the cell-to-cell transmission of the activation without attenuation along distance from the starting cells. The action potential, or TMP dynamics, represents the electrical activity within a single myocyte, purkinje fibres or nodes over time, and its propagation throughout the heart constitutes the whole image of cardiac electrical activity.

The electrical conduction system consists of some key nodes for transmission of electrical signals in the heart (see Fig.3.2). In a normal sinus rhythm, the electrical signal arises at sinoatrial (SA) node in the right atrium causing contraction of atria. Then

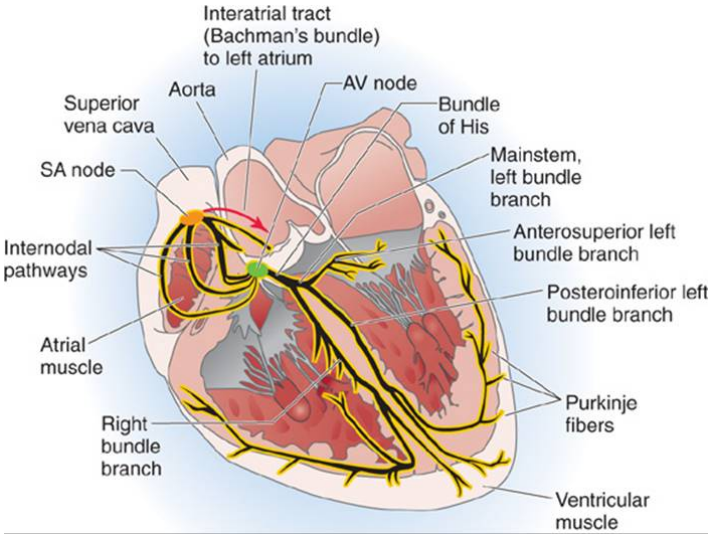


Figure 3.2: Fibre diagram of the heart [62]

.

the signal reaches atrioventricular(AV) node in the septum. After a delay, the electrical signal is conducted through the left and right bundle of His to the respective Purkinje fibers for each side of the heart, as well as to the endocardium at the apex of the heart, then finally to the ventricular epicardium; causing its contraction.

Several methods have been proposed to model the action potential dynamics, as shown in Fig. 3.1, with the trade off between complexity, accuracy and computational cost. Biophysical models, for example, are very detailed models considering the microscopic level ionic interactions within the cardiac cell and through cell membranes, and therefore, contains large number of parameters. Eikonal models, on the other hand, is a macroscopic model focusing only on the electrical wavefront and cannot model TMP propagation. Phenomenological models include details in between these two types of modeling and are computationally attractive. Considering the balance between model plausibility and computational feasibility, in this proposal, we choose the Aliev-Panfilov

model [2] described by two differential equations.

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla(\mathbf{D}\nabla u) + ku(u-1)(u-a) - uv, \\ \frac{\partial v}{\partial t} &= -\varepsilon(u,v)(v + ku(u-a-1)) \end{aligned} \tag{3.1}$$

where u is the transmembrane potential, v is the vector of recovery current, $\mathbf{D}$ is the diffusion tensor, k controls the repolarization, and a controls the excitability of the cells. These equations can be numerically solved over the discrete mesh of the ventricles as described in [109] to arrive at:

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{M}^{-1}\mathbf{K}\mathbf{u} + g_1(\mathbf{u},\mathbf{v}), \quad \frac{\partial \mathbf{v}}{\partial t} = g_2(\mathbf{u},\mathbf{v}) \tag{3.2}$$

Matrices $\mathbf{M}$ and $\mathbf{K}$ encode the 3-D myocardial structure and its conductive anisotropy. We use this model as a prior knowledge about the cardiac electrophysiological signals.

3.1.2 Electrocardiography

The cardiac electrical activity produces an electric field around it. The electric potential can be measured on the body surface as electrocardiogram (ECG); the process is called electrocardiography (EKG). Classical ECG recording systems consisted of three electrodes on the left arm, right arm and left leg, from which three limb voltages V_I, V_{II} and V_{III} are calculated. This system was modified to an extended, and more popular, version called 12 lead ECG consisting of six limb recordings and six precordial recordings.

For the purpose of inferring electrophysiological signals, we need denser ECG signals. Therefore, we use a high density body surface potential maps (BSPM)s. BSPMs use tens to hundreds of ECG electrodes. In our case, we use 120 lead BSPM (also called 120 lead ECG). Fig. 3.3 shows a schematic of a normal ECG signal of a single lead. It consists of following main segments: 1) P wave, 2) PR segment, 3) QRS complex, 4) ST segment, and 5) T wave. The P wave corresponds to the atrial depolarization, PR segment to the propagation of the activation through AV node and the Purkinje fiber,

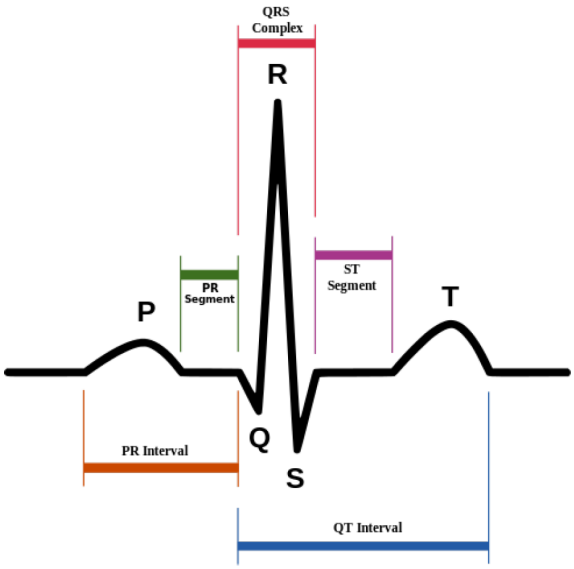


Figure 3.3: Schematic diagram of ECG signal

.

QRS complex to the depolarization of the ventricles, ST segment to the stage when all the myocytes are at the plateau and all regions in the ventricles are at depolarized state, and T wave to the re-polarization of the ventricles.

3.1.3 Forward and Inverse Problem

To obtain quantitative relation between the cardiac electric sources and the BSP on the body surface, we use quasi static approximation of the electromagnetic theory. Within the volume of myocardium, Ω_h, the bidomain theory [77] describes the distribution of extracellular potential, $\mathbf{y}_{te}$ as a result of the gradient of the action potential, $\mathbf{u}$ as:

$$\nabla((\mathbf{D}_i(r) + \mathbf{D}_e(\mathbf{r}))\nabla\mathbf{y}_{te}(\mathbf{r})) = \nabla(-\mathbf{D}_i(\mathbf{r})\nabla\mathbf{u}(\mathbf{r})) \quad \forall\mathbf{r} \in \Omega_h \tag{3.3}$$

where $\mathbf{r}$ is the position vector corresponding to the spatial coordinate, $\mathbf{D}_i$ and $\mathbf{D}_e$ are intracellular and extracellular conductivity tensors, and their summation $\mathbf{D}_k = \mathbf{D}_i + \mathbf{D}_e$ is the bulk conductivity tensor. Within the region between the heart surface and the

body surface, denoted by Ω_t/h, we assume no source of electrical activation, hence we have:

$$\nabla((\mathbf{D}_t(\mathbf{r}))\nabla\mathbf{y}_t(\mathbf{r})) = 0 \quad \forall r \in \Omega_t/h \tag{3.4}$$

where $\mathbf{D}_t$ is the torso conductivity tensor. For simplicity, we assume $\mathbf{D}_k$ and $\mathbf{D}_t$ to be isotropic and only $\mathbf{D}_i$ to be anisotropic; consequently, $\mathbf{D}_k$ and $\mathbf{D}_t$ become scalars σ_k and σ_t respectively. Assuming isotropic and homogeneous conductivity, the forward relationship between cardiac action potential and the body surface voltage data can be described with following Poisson equation within the heart and Laplace equation external to the heart:

$$\sigma_k\nabla^2\mathbf{y}_{tk}(\mathbf{r}) = \nabla(-\mathbf{D}_i(\mathbf{r})\nabla\mathbf{u}(\mathbf{r})) \quad \forall\mathbf{r} \in \Omega_h \tag{3.5}$$

$$\sigma_t\nabla^2\mathbf{y}_t(\mathbf{r}) = 0 \quad \forall\mathbf{r} \in \Omega_t/h \tag{3.6}$$

We solve eq. (3.5) and (3.6) with coupled meshfree and boundary element methods (BEM) [108]. BEM gives us a linear biophysical relation on a subject specific heart-torso model derived from tomographic images:

$$\mathbf{y}(t) = \mathbf{H}\mathbf{u}(t) \tag{3.7}$$

where $\mathbf{H}$ is the transfer matrix specific to individual's heart torso geometry and is assumed time invariant for simplicity.

The inverse electrophysiological imaging refers to the problem of estimating $\mathbf{u}(t)$ time sequence based on the information in BSP time sequence $\mathbf{y}(t)$. This is an ill posed inverse problem. Several approaches have been proposed to solve this inverse problem over the years.

3.2 Related Works

Noninvasive electrophysiological (EP) imaging aims at a mathematical reconstruction of cardiac electrical sources from high density electrocardiogram (ECG) signals. To

solve EP imaging, two types of sources models are used: 1) surface-based source models where source is sought in the form of electrical potential on the epicardium and/or endocardium [30, 44, 86], or activation time on the ventricular surface [52, 96, 104]; and 2) volumetric source model where the source is sought in the form of action potential [49,50,79,109], or current density/activation front [65] throughout the myocardial wall. Reconstructing surface sources is ill posed due to sparse measurement, attenuation and smoothing of the electric field while reaching the torso surface. In addition to these difficulties, the volumetric source reconstruction is plagued with additional issue of non-unique solution, i.e. multiple sources give rise to same ECG recording [18] even if all the previous mentioned problems were mitigated. In this sense, the seeking for transmural electric source throughout the myocardium is even more ill-posed; and therefore, surface source reconstruction can be thought as an implicit regularization.

The success of noninvasive EP imaging, therefore, largely relies on an effective incorporation of prior knowledge about the solutions via regularization techniques. Representative constraints include the smoothness of the electrical potential in space and/or time at different orders of derivatives, enforced through techniques such as Tikhonov regularization [91], truncated SVD [81], and spatio-temporal regularization [17]. Other constraints exploit sparsity of the cardiac signal in a certain domain, such as the gradient domain, by utilizing L1 norm [41] or total variation [114] as the regularization cost. Similar constraints on smoothness and sparsity can be incorporated within a probabilistic formulation where they enter into the equation as the prior distribution on the source signal. For example, Gaussian prior [95] is used for smoothness and total variation prior for sparsity [115], while generalized Gaussian prior [87] adapts between smoothness and sparsity.

Alternatively, model based regularization has been used to encode a priori physiological knowledge about the electrical propagation inside the heart. Examples include step jump functions [86] and logistic functions [104] to describe the activation of action potential, and parameterized curves modeling the wavefront velocity as trigonometric functions and the potential as a step response of a second order linear system [40]. When estimating transmural sources throughout the myocardium, 3D EP simulation

models of the spatiotemporal propagation of action potential have been used to provide dynamic constraints of the inverse problem [49, 79, 109].

PART I

PROBABILISTIC MODEL AND INFERENCE

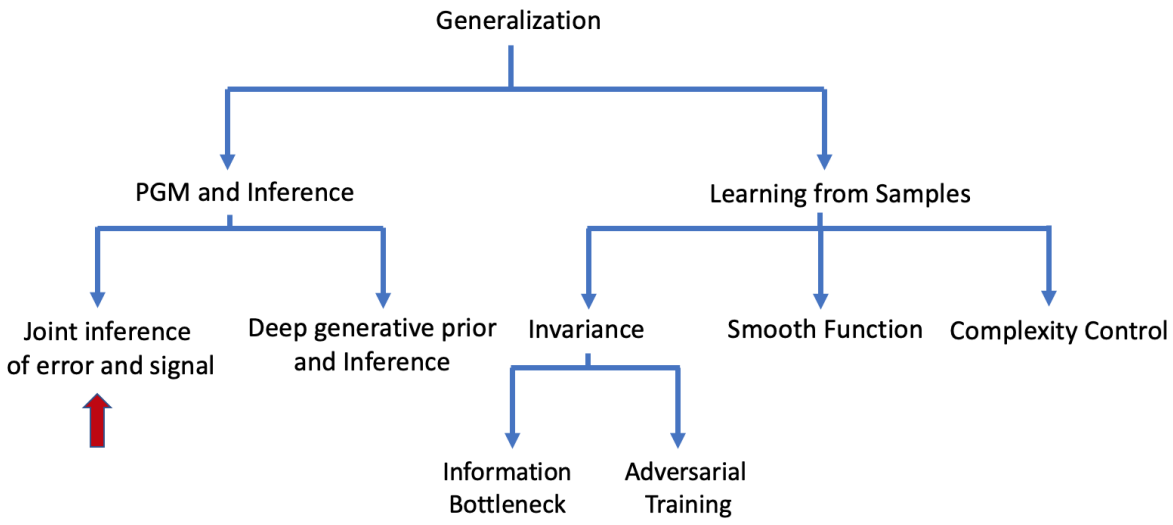
Prologue to Part I

In part I, we try to look at the inverse solution methods based on probabilistic graphical models and Bayesian inference. To improve generalization, we focus on integration of several knowledge sources and adaptation of common prior. These are achieved by modeling the error and modeling the prior in Chapters 4 and 5 respectively.

Chapter 4

Learning by Inferring Model Error

Knowledge rests not upon truth alone, but upon error also.
- Carl Jung



4.1 Introduction

Since we are interested in inferring transmembrane potential throughout the myocardium, we use volumetric source model as discussed before. To overcome ill-posedness in solving inverse problem, Alieve-Panfilov model based simulation model [109] can provide a general population knowledge about the behavior of cardiac electrophysiological signals through the differential equations. We want to generalize this knowledge to help in the inference of patient specific TMP. This requires addressing a challenge: model parameters controlling the shape of transmembrane potential (TMP) vary across the heart depending upon whether the underlying tissue is healthy or diseased, and depending upon the origin of excitation of TMP, and are unknown a priori. In the absence of prior knowledge about these parameters, common practice is to assume values commonly used in literature, introducing errors in the models. Such model inaccuracies and their effect on the inverse solution have been studied. Erem et al [30], for example, used convex relaxation of the original problem to study how the solution differs if the model assumption about a uniform TMP amplitude throughout the heart is violated due to the presence of infarction, and Xu et al [115] investigated uncertainty in the inverse solution due to model errors, and showed the importance of considering the resulting solution uncertainty in addition to a point estimate. While these works have highlighted the importance of acknowledging prior model errors in EP imaging, addressing these errors remains a challenge.

Ideally, we would want to estimate patient specific parameters in the EP model in addition to TMP. Since this task is very difficult, we take a slightly different approach by estimating error introduced due to the error in parameters. In this chapter, we present a probabilistic framework to allow principled estimation of the prior model error while reconstructing transmural TMP under the constraint of the EP simulation model. However, simultaneous estimation of both the error and TMP is still challenging, and might require additional source of information to address it. To overcome this challenge, we exploit the low-dimensional nature of cardiac wavefront propagation to formulate a sparse representation for the model error. We then present a Bayesian inference method to estimate the posterior distribution of transmural TMP and the sparse error

of the prior EP model. Building upon our previous work [38], we provide a rigorous treatment to the inference by explicitly introducing error random variable and jointly inferring its posterior distribution. This enables proper estimation of model uncertainty as a combination of propagated uncertainty from previous time and the estimated uncertainty at the present time.

We evaluate the performance of the presented method on simulated and real data on its ability to detect and correct model errors resulting from the presence of myocardial infarction and unknown excitation points. We compare its performance with the previously-described model-constrained approach to TMP reconstruction [109] that does not consider errors in the a priori model. The main contributions of this paper include:

1. We present a new probabilistic approach to EP imaging that is able to estimate the error in the prior model by leveraging the sparsity of model errors.

2. We show that the presented method can detect and correct errors in prior model predictions, improving the accuracy of the estimated TMP signals in the presence of unknown infarction and excitation locations.

3. We provide theoretical and experimental analysis relating the performance of the presented method to the interplay between ECG data and the singular value decomposition of the forward matrix.

4. We relate the presented method to algorithms in machine learning community, such as relevance vector machines (RVM) and Empirical Bayes, to provide further insights into the nature of the solution.

This chapter includes parts from author's journal and conference publications [37, 38].

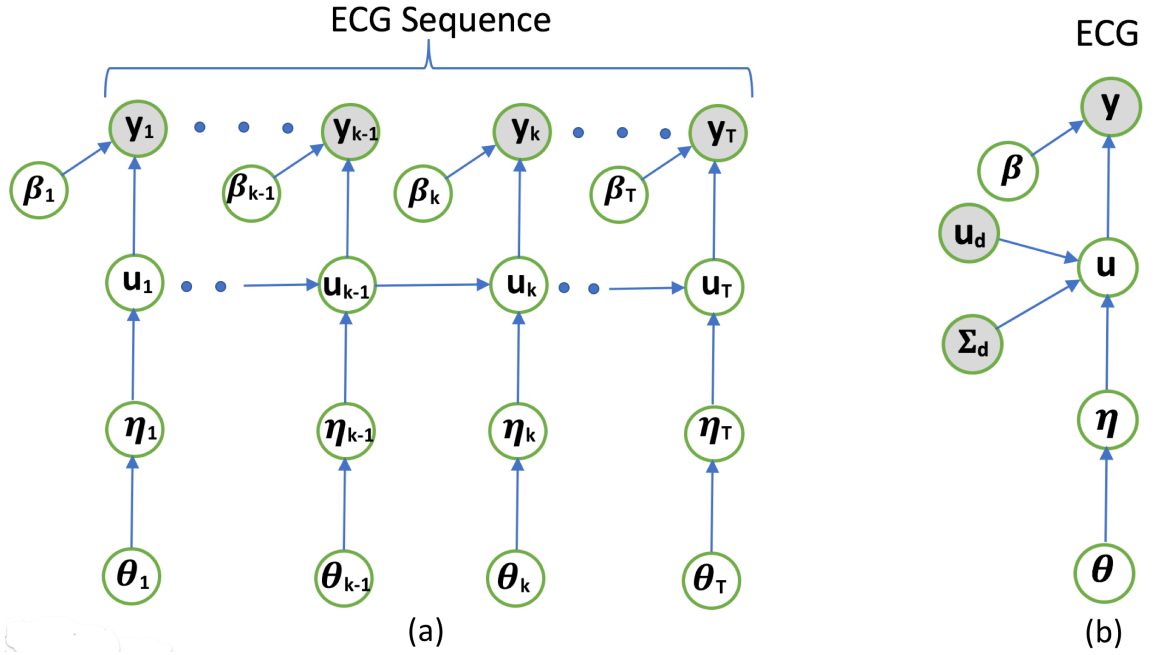


Figure 4.1: Probabilistic graphical models of (a) ECG sequence and (b) ECG at a time instant.

4.2 Probabilistic Formulation for EP Imaging

We represent the generation of ECG sequence by a probabilistic graphical model as shown in Fig.4.1(a), where TMP $\mathbf{u}_k$ is the latent random variable generating ECG data through the linear measurement model, and the hidden state $\mathbf{u}_k$ is related to the previous state by the prior EP model (eq.3.2). Solving it numerically over time provides:

$$\mathbf{u}_k = f(\mathbf{u}_{k-1}) \tag{4.1}$$

where f denotes the routine for numerically solving eq.(3.2) and does not necessarily have a closed form. Furthermore, to account for modeling errors in the prior model given by eq.(4.1), we introduce a prediction error $\boldsymbol{\eta}_k$ through:

$$\mathbf{u}_k = f(\mathbf{u}_{k-1}) + \boldsymbol{\eta}_k \tag{4.2}$$

While existing works often model $\boldsymbol{\eta}_k$ as a known Gaussian noise with a pre-defined variance [109], we assume $\boldsymbol{\eta}_k$ to be unknown with a prior distribution parameterized by $\boldsymbol{\theta}_k$. The joint posterior distribution of $\mathbf{u}_k$ for all time instants is analytically intractable because of the lack of closed form solution of eq.(4.1). Therefore, we sequentially solve for the marginal probability density function (pdf) of $\mathbf{u}_k$ for each time instant given ECG data till present time, $\mathbf{y}_{1,..k}$. Since $\mathbf{u}_k$ depends on previous ECG data $\mathbf{y}_{1,..k-1}$ through $\mathbf{u}_{k-1}$ (see Fig.4.1(a)), given the posterior distribution of $\mathbf{u}_{k-1}$, $\mathbf{u}_k$ is independent of $\mathbf{y}_{1,..k-1}$. This brings us to the graphical model in Fig.4.1(b) for the generation of ECG data at time instant k. Components of this graphical model are detailed below:

Likelihood

ECG data $\mathbf{y}_k$ is generated from TMP $\mathbf{u}_k$ through the linear measurement model described earlier considering a zero-mean Gaussian noise with variance β_k^{-1}:

$$p(\mathbf{y}_k|\mathbf{u}_k, \beta_k) = \mathcal{N}(\mathbf{y}_k|\mathbf{H}\mathbf{u}_k, \beta_k^{-1}\mathbf{I}) \tag{4.3}$$

where β_k is modeled with the conjugate Gamma prior:

$$p(\beta_k|c, d) = \frac{d^c}{\Gamma(c)}\beta_k^{-1}\exp(-d\beta_k) \tag{4.4}$$

Conditional prior of $\mathbf{u}_k$

We model the prior of action potential $\mathbf{u}_k$ conditioned on previous ECG data as well as the prediction error $\boldsymbol{\eta}_k$. Given the posterior distribution of action potential $\mathbf{u}_{k-1}$ at the previous time instant, $p(\mathbf{u}_{k-1}|\beta_{1...k-1}, \mathbf{y}_{1..k-1}, \boldsymbol{\theta}_{1..k-1})$, we have:

$$\begin{aligned} &p(\mathbf{u}_k|\boldsymbol{\eta}_k, \beta_{1...k-1}, \mathbf{y}_{1..k-1}, \boldsymbol{\theta}_{1..k-1}) \\ &= \int p(\mathbf{u}_k|\mathbf{u}_{k-1}, \boldsymbol{\eta}_k)p(\mathbf{u}_{k-1}|\beta_{1...k-1}, \mathbf{y}_{1..k-1}, \boldsymbol{\theta}_{1..k-1})d\mathbf{u}_{k-1} \end{aligned} \tag{4.5}$$

where $p(\mathbf{u}_k|\mathbf{u}_{k-1}, \boldsymbol{\eta}_k)$ can be defined as $\mathcal{N}(\mathbf{u}_k|f(\mathbf{u}_{k-1}) + \boldsymbol{\eta}_k, \mathbf{0})$ based on the prior EP model in eq.(4.2). Because f is not in a closed form, the integral in eq.(4.5) cannot be

solved analytically but has to be approximated numerically. To do so, we sample from the posterior distribution of $\mathbf{u}_{k-1}$ and pass them through the physiological model f. The mean $\mathbf{u}_d$ and covariance $\boldsymbol{\Sigma_d}$ of $f(\mathbf{u}_{k-1})$ are then approximated from the output samples. Let $\boldsymbol{\omega_k}$ be the Gaussian approximation of $f(\mathbf{u}_{k-1})$, *i.e.*, $\boldsymbol{\omega_k} \sim \mathcal{N}(\boldsymbol{\omega}_k|\mathbf{u}_d, \boldsymbol{\Sigma_d})$, we have:

$$
\begin{aligned}
& p(\mathbf{u}_k|\boldsymbol{\eta}_k, \beta_{1\ldots k-1}, \mathbf{y}_{1..k-1}, \boldsymbol{\theta}_{1..k-1}) \\
&= \int \mathcal{N}(\mathbf{u}_k|f(\mathbf{u}_{k-1}) + \boldsymbol{\eta}_k, \mathbf{0}) p(\mathbf{u}_{k-1}|\beta_{1\ldots k-1}, \mathbf{y}_{1..k-1}, \boldsymbol{\theta}_{1..k-1}) d\mathbf{u}_{k-1} \\
&= \int \mathcal{N}(\mathbf{u}_k|\boldsymbol{\omega_k} + \boldsymbol{\eta}_k, \mathbf{0}) \mathcal{N}(\boldsymbol{\omega}_k|\mathbf{u}_d, \boldsymbol{\Sigma_d}) d\omega_k \qquad (4.6) \\
&= \mathcal{N}(\mathbf{u}_k|\mathbf{u}_d + \boldsymbol{\eta}_k, \boldsymbol{\Sigma_d}) \qquad (4.7)
\end{aligned}
$$

where eq.(4.6) uses the law of unconscious statistician (LOTUS) about the transformation of random variables : $\int g(f(u))p(u)du = \int g(\omega)p(\omega)d\omega$ if $\omega = f(u)$.

Error model

Finally, we model the prediction error $\boldsymbol{\eta}_k$ with a prior distribution $p(\boldsymbol{\eta}_k|\boldsymbol{\theta}_k)$. Because we model $\boldsymbol{\eta}_k$ independently for each time instant, below we drop k from the formulation for the sake of simplicity.

To model $\boldsymbol{\eta}$, we exploit its low-dimensional structure by considering the physiological phenomenon that TMP wavefront (which can be thought as spatial gradient of TMP) is spatially localized. It is therefore reasonable to assume the spatial gradient vector of $\mathbf{u}_k$ to be sparse with a lot of zeros, as illustrated in the examples in Fig. 4.2. At any time instant, the difference between the gradient of true TMP and that predicted by an erroneous model would capture the difference in their wavefronts, which would be localized in space and can be reasonably approximated by a sparse representation. This is demonstrated in Fig. 4.2, where the actual wavefront (left column) is delayed by the annotated infarct region when moving from the apex towards the base of the ventricles. In comparison, propagation produced by a prediction model unaware of the infarct (middle column) does not exhibit this delay. The difference between these

two wavefront, computed as the difference of TMP gradient, is also sparse (spatially localized) at any time instant as illustrated in the last column of Fig. 4.2.

One common practice to enforce sparsity is to use L1 penalty and correspondingly laplacian prior distribution. More recently, L_p norm ($0 \leq p < 1$) [1] has been used to generate sparse solutions [19, 27] in compressed sensing. Both of these cases can be incorporated within a single framework of Generalized normal distribution with L_p norm in the exponent:

$$p_{gn}(\boldsymbol{\eta}|\alpha) = \Big(\frac{p}{2\alpha\Gamma(1/p)}\Big)^n \exp\Big(-\Big(\frac{\|\mathbf{D}\boldsymbol{\eta}\|_p}{\alpha}\Big)^p\Big) \tag{4.8}$$

where α is the hyperparameter and $\mathbf{D}$ is the 3D spatial gradient operator. As we decrease p from 2 towards 0, the tail of this distribution gets heavier encouraging sparser solutions. One key difficulty in calculating the posterior distribution using generalized normal prior is the presence of the L_p norm in exponent of eq.(4.8). Hence, to perform principled inference, we derive a variational lower-bound of eq.(4.8) below.

Theorem 1. *Let $\boldsymbol{x} = (x_1, x_2, ...x_n)$ be a vector with independent components each following a generalized normal distribution with the same parameters α and p, with a joint pdf: $p(\boldsymbol{x}|\alpha) = \Big(\frac{p}{2\alpha\Gamma(1/p)}\Big)^n \exp\Big(-\Big(\frac{\|\boldsymbol{x}\|_p}{\alpha}\Big)^p\Big)$.*
Then, $p(\boldsymbol{x}|\alpha) = \sup_{\boldsymbol{\lambda}>0} \frac{C}{\alpha^n} \exp\Big(-\frac{\boldsymbol{x}^T\boldsymbol{\Lambda}\boldsymbol{x}}{2} - \frac{2-p}{2}\big(\frac{\alpha^2}{p}\big)^{\frac{p}{p-2}} \sum_i \lambda_i^{\frac{p}{p-2}}\Big)$
where $C = (\frac{p}{2\alpha\Gamma(1/p)})^n$ and $\boldsymbol{\Lambda} = diag(\boldsymbol{\lambda})$

The proof of Theorem 1 is provided in the Appendix. It makes use of the Fenchel-Lagrange duality and uses conjugate of a convex function to derive a variational lower bound. Fig.4.3 illustrates the crux of Theorem 1: the red curve represents a function with the negative of L_p norm, raised to the pth power, in the exponent – the generalized normal distribution is composed of such functions in each component. This function is lowerbounded by functions with a negative quadratic term (like Gaussian) in the exponent. So, essentially we have replaced a complicated function with a family of simpler

[1] L_p norm is not a norm for ($0 \leq p < 1$) in strict sense because it does not satisfy the triangle inequality which is easy to verify noting non-convexity of unit ball in L_p space. Here, we refer to it as a norm for the sake of convenience.

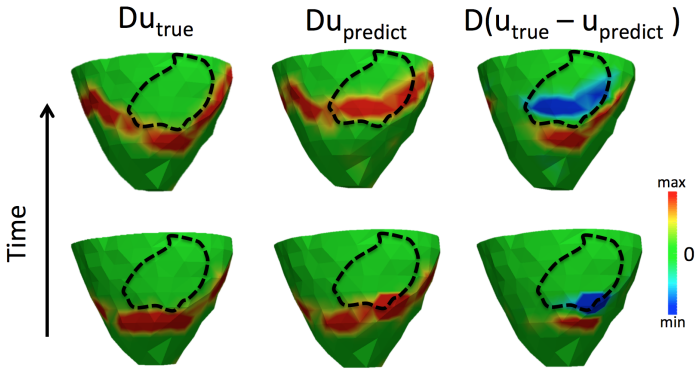


Figure 4.2: Spatial gradient of true and predicted TMP and their difference.

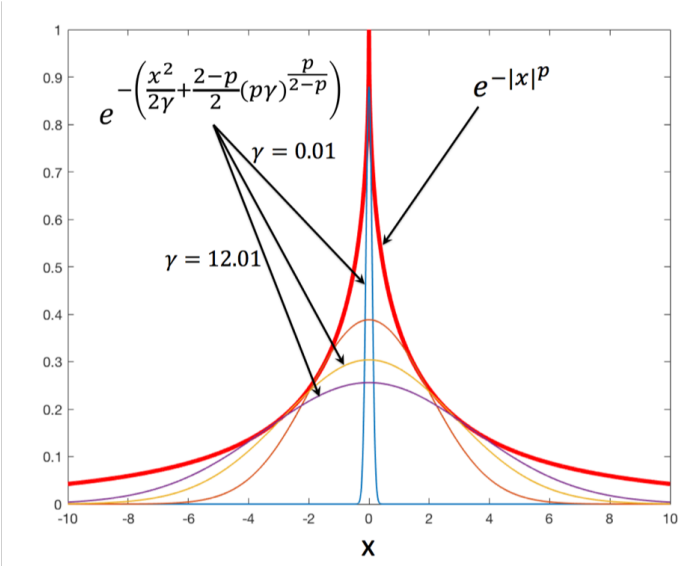


Figure 4.3: At each point x, and fixed p, there exists a lowerbounding Gaussian-like function that tangentially touches $\exp(-|x|^p)$.

lower bounding functions. Obviously, it comes with the additional set of variational parameters, each corresponding to one Gaussian-like function. However, the advantage of this formulation is that, conditioned on fixed variational parameter, the function is Gaussian (multiplied by some constant). This brings forth the tractability of Gaussian distributions and the consequent computational advantage during the development of the inference algorithm that will be elaborated in section III-B.

Setting $\mathbf{x} = \mathbf{D}\boldsymbol{\eta}$ in Theorem 1, we obtain,

$$p_{gn}(\boldsymbol{\eta}|\alpha) = \sup_{\boldsymbol{\lambda}>0} \frac{C}{\alpha^n} \exp\left(-\frac{\mathbf{u}^T\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}\mathbf{u}}{2} - \frac{2-p}{2}\left(\frac{\alpha^2}{p}\right)^{\frac{p}{p-2}} \sum_i \lambda_i^{\frac{p}{p-2}} \right) \quad (4.9)$$

Dropping the supremum in eq.(4.9) gives us a lower bound for the generalized normal distribution for any $\boldsymbol{\lambda}$. This lower bound is used as the prior distribution of $\boldsymbol{\eta}$, $p(\boldsymbol{\eta}|\boldsymbol{\theta})$, treating $\boldsymbol{\lambda}$ as a variational parameter to be optimized during the inference.

$$p(\boldsymbol{\eta}|\boldsymbol{\theta}) = \frac{C}{\alpha^n} \exp(-\frac{\boldsymbol{\eta}^T\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}\boldsymbol{\eta}}{2} - \frac{2-p}{2}(\frac{\alpha^2}{p})^{\frac{p}{p-2}} \sum_i \lambda_i^{\frac{p}{p-2}}) \quad (4.10)$$

where $\boldsymbol{\theta} = \{\alpha, \boldsymbol{\lambda}\}$.

By definition, the gradient operator $\mathbf{D}$ has a null space: a vector containing all ones (say $\mathbf{1}$). Using this $\mathbf{D}$ thus will fail to correct the constant bias in TMP. To address this issue, we augment $\mathbf{D}$ with one more row of all ones, i.e. $\mathbf{D} = \left(\mathbf{D}^T, \mathbf{1}\right)^T$.

4.3 Joint inference of transmural TMP and prediction errors

As the inference is iteratively carried out for each time instant, we drop k from equations for simplicity. Given the probabilistic formulation described in the previous section, we have the following joint pdf of interest:

$$\begin{aligned} &p(\mathbf{y}, \mathbf{u}, \boldsymbol{\eta}, \beta|\mathbf{u}_d, \boldsymbol{\Sigma}_d, \theta, c, d) = \\ &p(\mathbf{y}|\mathbf{u}, \beta)p(\mathbf{u}|\boldsymbol{\eta}, \mathbf{u}_d, \boldsymbol{\Sigma}_d)p(\boldsymbol{\eta}|\boldsymbol{\theta})p(\beta|c, d) \end{aligned} \quad (4.11)$$

We make notation uncluttered by writing this distribution as $p(\mathbf{y}, \mathbf{u}, \boldsymbol{\eta}, \beta|\theta)$ where $\mathbf{u}_d, \boldsymbol{\Sigma}_d, c, d$ are understood as given. We are interested in jointly estimating the random variables $\mathbf{u}, \boldsymbol{\eta}, \beta$ as well as parameter $\boldsymbol{\theta} = \{\boldsymbol{\lambda}, \alpha\}$. We propose to do this in two steps. First, we estimate the parameter $\boldsymbol{\theta}$ as the maximum likelihood estimate by integrating

out the variables $\mathbf{u}, \boldsymbol{\eta}, \beta$, *i.e.*,

$$\hat{\boldsymbol{\theta}} = \underset{\theta}{argmax}\, p(\mathbf{y}|\theta), \tag{4.12}$$
$$\text{where } p(\mathbf{y}|\theta) = \int p(\mathbf{y}, \mathbf{u}, \boldsymbol{\eta}, \beta|\theta) d\mathbf{u} d\boldsymbol{\eta} d\beta$$

Once we obtain the optimum $\hat{\boldsymbol{\theta}}$, we then compute posterior distribution $p(\mathbf{u}, \eta, \beta|\mathbf{y}, \hat{\boldsymbol{\theta}})$. However, eq.(4.12) is difficult to solve due to the need to integrate out the random variables $\mathbf{u}, \boldsymbol{\eta}$ and β. We therefore present an iterative procedure which yields us both the optimum $\hat{\boldsymbol{\theta}}$ and $p(\mathbf{u}, \eta, \beta|\mathbf{y}, \hat{\boldsymbol{\theta}})$.

We decompose $\log p(\mathbf{y}|\boldsymbol{\theta})$ as:

$$\log p(\mathbf{y}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q||p) \tag{4.13}$$
$$\text{where } \mathcal{L}(q, \theta) = \int q(\mathbf{u}, \boldsymbol{\eta}, \beta) \log(\frac{p(\mathbf{y}, \mathbf{u}, \boldsymbol{\eta}, \beta|\boldsymbol{\theta})}{q(\mathbf{u}, \boldsymbol{\eta}, \beta)}) \tag{4.14}$$
$$KL(q||p) = \int q(\mathbf{u}, \boldsymbol{\eta}, \beta) \log(\frac{q(\mathbf{u}, \boldsymbol{\eta}, \beta)}{p(\mathbf{u}, \boldsymbol{\eta}, \beta|\mathbf{y}, \boldsymbol{\theta})}) \tag{4.15}$$

Since Kullback-Leibler divergence $KL(q||p)$ between q and $p(\mathbf{u}, \boldsymbol{\eta}, \beta|\mathbf{y}, \boldsymbol{\theta})$ is non-negative, $\mathcal{L}$ is the lowerbound of $\log\ p(\mathbf{y}|\boldsymbol{\theta})$ with the gap given by $KL(q||p)$. To maximize $\log p(\mathbf{y}|\boldsymbol{\theta})$, we can thus minimize the KL divergence gap $KL(q||p)$ and maximize the lowerbound $\mathcal{L}$. We achieve this by two alternating optimization: i) Posterior approximation where we fix $\boldsymbol{\theta}$ and minimize KL by making q as close to true posterior $p(\mathbf{u}, \boldsymbol{\eta}, \beta|\mathbf{y}, \boldsymbol{\theta})$ as possible via Variational Bayes, and ii) Parameter optimization, where we fix q and maximize the lowerbound $\mathcal{L}$ with respect to $\boldsymbol{\theta}$. This style of alternatively estimating parameter and posterior distribution of hidden variable is known as expectation maximization (EM).

Posterior approximation of $\mathbf{u}$,$\boldsymbol{\eta}, \beta$: Given the estimate at previous iteration, $\boldsymbol{\theta}_{old} = \{\boldsymbol{\lambda}_{old}, \alpha_{old}\}$, true posterior distribution is:

$$p(\mathbf{u}, \boldsymbol{\eta}, \beta|\mathbf{y}, \boldsymbol{\theta}_{old}) \propto p(\mathbf{y}|\beta, \mathbf{u}) p(\mathbf{u}|\boldsymbol{\eta}) p(\boldsymbol{\eta}|\boldsymbol{\lambda}_{old}, \alpha_{old}) p(\beta) \tag{4.16}$$

Note that, through the variational lower-bound $p(\boldsymbol{\eta}|\boldsymbol{\theta})$ derived in Theorem 1, $p(\boldsymbol{\eta}|\boldsymbol{\lambda}_{old}, \alpha_{old})$ becomes Gaussian when conditioned on known values of $\boldsymbol{\lambda}_{old}$ and α_{old}. In another word,

by combining Theorem 1 and the EM algorithm, we are able to replace a complex distribution (Lp norm in the exponent) with a Guassian distribution and greatly simplify calculation of the posterior distribution (and its approximation).

The approximated joint distribution $q(\mathbf{u}, \boldsymbol{\eta}, \beta)$ is obtained using Variational Bayes with mean field approximation: $q(\mathbf{u}, \boldsymbol{\eta}, \beta) = q(\mathbf{u}, \boldsymbol{\eta})q(\beta)$. Note that we only assume the independence to exist between β and $(\boldsymbol{\eta}, \mathbf{u})$, not between $\boldsymbol{\eta}$ and $\mathbf{u}$ since the action potential and model error is closely related. From eq.(4.16), Variational Bayes yields:

$$\begin{aligned}\log q(\mathbf{u}, \boldsymbol{\eta}) &= E_{q(\beta)}[\log[p(\mathbf{y}|\beta, \mathbf{u})p(\mathbf{u}|\boldsymbol{\eta})p(\boldsymbol{\eta}|\boldsymbol{\lambda}_{old}, \alpha_{old})]] + c \\ &= -\frac{1}{2}[E_{q(\beta)}[\beta](\mathbf{y} - \mathbf{H}\mathbf{u})^T(\mathbf{y} - \mathbf{H}\mathbf{u}) + \boldsymbol{\eta}^T\mathbf{D}^T\boldsymbol{\Lambda}_{old}\mathbf{D}\boldsymbol{\eta} \\ &+ (\mathbf{u} - \mathbf{u}_d - \boldsymbol{\eta})^T{\boldsymbol{\Sigma}_{\boldsymbol{d}}}^{-1}(\mathbf{u} - \mathbf{u}_d - \boldsymbol{\eta})] + c\end{aligned}$$

$$q(\mathbf{u}, \boldsymbol{\eta}) = \mathbf{C}\exp(-\frac{1}{2}[\bar{\beta}(\mathbf{y} - \mathbf{H}\mathbf{u})^T(\mathbf{y} - \mathbf{H}\mathbf{u}) \tag{4.17}$$

$$+ (\mathbf{u} - \mathbf{u}_d - \boldsymbol{\eta})^T{\boldsymbol{\Sigma}_{\boldsymbol{d}}}^{-1}(\mathbf{u} - \mathbf{u}_d - \boldsymbol{\eta}) + \boldsymbol{\eta}^T\mathbf{D}^T\boldsymbol{\Lambda}_{old}\mathbf{D}\boldsymbol{\eta}]) \tag{4.18}$$

where $\bar{\beta} = E_{q(\beta)}[\beta]$ and $q(\mathbf{u}, \boldsymbol{\eta})$ is jointly Gaussian. Marginal distributions $q(\mathbf{u})$ and $q(\boldsymbol{\eta})$ can then be analytically derived:

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\bar{\mathbf{u}}, \boldsymbol{\Sigma}_u)$$

where $\boldsymbol{\Sigma}_u^{-1} = \beta\mathbf{H}^T\mathbf{H} + (\boldsymbol{\Sigma}_{\boldsymbol{d}} + \boldsymbol{\Lambda}_{old}^{-1})^{-1}$, $\bar{\mathbf{u}} = \boldsymbol{\Sigma}_u(\beta\mathbf{H}^T\mathbf{y} + (\boldsymbol{\Sigma}_{\boldsymbol{d}} + \boldsymbol{\Lambda}_{old}^{-1})^{-1}\mathbf{u}_d)$.

$$q(\boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\eta}|\bar{\boldsymbol{\eta}}, \boldsymbol{\Sigma}_\eta)$$

where $\boldsymbol{\Sigma}_\eta^{-1} = \mathbf{D}^T\boldsymbol{\Lambda}_{old}\mathbf{D} + \mathbf{H}^T(\beta_{old}^{-1}\mathbf{I} + \mathbf{H}\Sigma_d\mathbf{H}^T)^{-1}\mathbf{H}$, $\bar{\boldsymbol{\eta}} = \boldsymbol{\Sigma}_\eta\mathbf{H}^T(\beta_{old}^{-1}\mathbf{I} + \mathbf{H}\Sigma_d\mathbf{H}^T)^{-1}(\mathbf{y} - \mathbf{H}\mathbf{u}_d)$.

Using Variational Bayes, $q(\beta)$ can be calculated as,

$$\log q(\beta) = E_{q(\mathbf{u},\boldsymbol{\eta})}[\log[p(\mathbf{y}|\beta, \mathbf{u})p(\beta)]] + c \tag{4.19}$$

$$\begin{aligned}&= -\beta[d + \frac{1}{2}E_{q(\mathbf{u},\boldsymbol{\eta})}[(\mathbf{y} - \mathbf{H}\mathbf{u})^T(\mathbf{y} - \mathbf{H}\mathbf{u})] \\ &+ (c - 1 + \frac{m}{2})log[\beta] + c\end{aligned} \tag{4.20}$$

$$q(\beta) = Gamma(c + \frac{m}{2}, d + \frac{1}{2}E_{q(\mathbf{u})}[(\mathbf{y} - \mathbf{H}\mathbf{u})^T(\mathbf{y} - \mathbf{H}\mathbf{u}))$$
$$\bar{\beta} = \frac{m + 2c}{2d + \|\mathbf{y} - \mathbf{H}\bar{\mathbf{u}}\|^2 + tr(\mathbf{\Sigma}_u \mathbf{H}^T\mathbf{H})} \tag{4.21}$$

Parameter optimization: Fixing the posterior approximation q obtained from the previous step, maximization of $\mathcal{L}$ in eq.(4.14) is equivalent to maximization of $E_{q(\mathbf{u},\boldsymbol{\eta},\beta)}[\log p(\mathbf{y}, \mathbf{u}, \boldsymbol{\eta}, \beta|\boldsymbol{\theta})]$ with respect to $\boldsymbol{\theta}$. In $\log p(\mathbf{y}, \mathbf{u}, \boldsymbol{\eta}, \beta|\boldsymbol{\theta})$, the only term depending on $\boldsymbol{\theta}$ is $\log(p(\boldsymbol{\eta}|\boldsymbol{\theta}))$. In taking expectation of this term, we can marginalize out $\mathbf{u}, \beta$, leaving the optimization of $E_{q(\boldsymbol{\eta})}[\log(p(\boldsymbol{\eta}|\boldsymbol{\theta}))]$ which is achieved by equating its first derivative to zero:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \left(E_{q(\boldsymbol{\eta})}[\log(p(\boldsymbol{\eta}|\boldsymbol{\theta}))] \right) = \mathbf{0} \tag{4.22}$$

The details of derivations are in appendix B. The complete algorithm is summarized in Algorithm 1.

Limiting case of p Choosing p close to zero makes our prior sparser which we expect to work better. Fortunately, we can derive the limiting case expression for $p \to 0$ in Algorithm 1:

$$\text{If } p \to 0 \text{ then } s \to 1, \lambda_i \to \frac{1}{tr([\bar{\boldsymbol{\eta}}\bar{\boldsymbol{\eta}}^T + \mathbf{\Sigma}_{\boldsymbol{\eta}}]\mathbf{d}_i\mathbf{d}_i^T)} \tag{4.23}$$

We report results using Algorithm 1 with this limiting case in all experiments unless stated otherwise. The effect of different values of p is investigated in section 4.7.

Upon convergence, $q(\mathbf{u})$ provides the posterior distribution of TMP at the current time instant, k. It will then be used to predict the prior distribution of TMP at the next time instant, $k + 1$, as described in the previous section.

4.4 Reducing Computational Cost

A main computational cost of the presented method comes from the inversion of matrices listed in steps 8-10 in Algorithm 1. In specific, let $\mathbf{H} \in M \times N$ where $M \sim 120$ and

Algorithm 1 Data Corrected Posterior Distribution Algorithm

procedure DATA CORRECTED POSTERIOR($\mathbf{u}_d$,Σ_d)
  Initialize p, c, d, $\boldsymbol{\lambda}$, β, $\lambda_{threshold}$
  m=no. of rows in H
  n=no. of rows in D
  $\mathbf{D} = \begin{pmatrix} \mathbf{D} \\ \mathbf{1}^T \end{pmatrix}$
  while $r < maxIteration$ & $\bar{\mathbf{u}}$ does not converge **do**
    $\boldsymbol{\Lambda} = diag(\boldsymbol{\lambda})$
    $\mathbf{P}_u = (\boldsymbol{\Sigma_d} + (\mathbf{D}^T \boldsymbol{\Lambda} \mathbf{D})^{-1})^{-1}$
    $\boldsymbol{\Sigma}_u = (\beta \mathbf{H}^T \mathbf{H} + \mathbf{P}_u)^{-1}$
    $\bar{\mathbf{u}} = \boldsymbol{\Sigma}_u(\beta \mathbf{H}^T \mathbf{y} + \mathbf{P}_u \mathbf{u}_d)$
    $\mathbf{P}_\eta = (\beta^{-1}\mathbf{I} + \mathbf{H}\boldsymbol{\Sigma_d}\mathbf{H}^T)^{-1}$
    $\boldsymbol{\Sigma}_\eta = (\mathbf{D}^T \boldsymbol{\Lambda} \mathbf{D} + \mathbf{H}^T \mathbf{P}_\eta \mathbf{H})^{-1}$
    $\bar{\boldsymbol{\eta}} = \boldsymbol{\Sigma}_\eta \mathbf{H}^T \mathbf{P}_\eta (\mathbf{y} - \mathbf{H}\mathbf{u}_d)$
    $s = N/(\sum_i \lambda_i^{\frac{p}{p-2}})$
    for i in 1 to n **do**
      $\lambda_i = \left(\frac{s}{tr([\bar{\boldsymbol{\eta}}\bar{\boldsymbol{\eta}}^T + \boldsymbol{\Sigma_\eta}]\mathbf{d}_i \mathbf{d}_i^T)} \right)^{\frac{2-p}{2}}$
    end for
    $\boldsymbol{\lambda}_{n+1}$=max($\lambda_{threshold}$,max($\boldsymbol{\lambda}_{1:n}$))
    $\beta = \frac{m+2c-1}{2d + \|\mathbf{y} - \mathbf{H}\bar{\mathbf{u}}\|^2 + tr(\boldsymbol{\Sigma}_u \mathbf{H}^T \mathbf{H})}$
  end while
  return $\bar{\mathbf{u}}$, $\boldsymbol{\Sigma_u}$ ▷ Posterior mean and covariance
end procedure

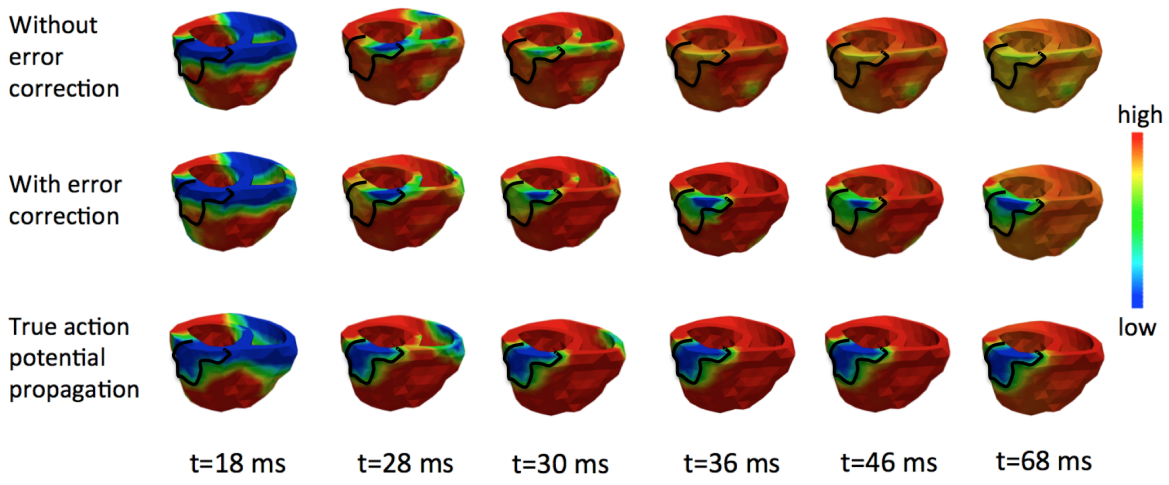


Figure 4.4: Comparison of TMP propagation sequences between simulated ground truth and reconstructions with and without model error correction. Scar region has been delineated with black contour.

$N \sim 2000$; steps 8-10 require three inversions of matrices of size $N \times N$. To reduce this cost, we rearrange equations in those steps and equivalently invert $M \times M$ matrices instead of $N \times N$:

$$\begin{aligned} \boldsymbol{\Sigma}_u &= (\beta \mathbf{H}^T \mathbf{H} + \boldsymbol{\Sigma_p}^{-1})^{-1} \\ &= \boldsymbol{\Sigma_p} - \boldsymbol{\Sigma_p} \mathbf{H}^T (\mathbf{H} \boldsymbol{\Sigma_p} \mathbf{H}^T + \beta^{-1} \mathbf{I})^{-1} \mathbf{H} \boldsymbol{\Sigma_p} \qquad (4.24) \\ \bar{\mathbf{u}} &= \boldsymbol{\Sigma}_u (\beta \mathbf{H}^T \mathbf{y} + \boldsymbol{\Sigma_p}^{-1} \mathbf{u}_d) \\ &= \boldsymbol{\Sigma}_p \mathbf{H}^T (\mathbf{H} \boldsymbol{\Sigma_p} \mathbf{H}^T + \beta^{-1} \mathbf{I})^{-1} \mathbf{y} + (\beta \boldsymbol{\Sigma}_p \mathbf{H}^T \mathbf{H} + \mathbf{I})^{-1} \boldsymbol{u}_d \qquad (4.25) \end{aligned}$$

where $\boldsymbol{\Sigma_p} = \boldsymbol{\Sigma_d} + (\mathbf{D}^T \boldsymbol{\Lambda} \mathbf{D})^{-1}$.

Proof. Proof is in Appendix C. □

Using this reformulation we reduced the computational time by ∼30% using Tesla K20m GPU (5GB), 2.2 GHz processor and using matlab inbuilt functions supporting GPU. As described earlier, we reduce cost by decreasing the number of inversion of heavy matrices. Therefore, in a setup where matrix inversion is made very efficient

using GPU or alternative parallel architecture and/or low level programming language, a smaller gain may be expected by this rearrangement.

4.5 Connection with Sparse Bayesian Learning

We note that the prior distribution on the model error (eq.(4.10)) is a variational distribution with a quadratic term in the exponent. This is reminiscent of works in sparse Bayesian learning (SBL), where a zero mean Gaussian prior with unknown variance is used to enforce sparsity [102, 112]. If we rearrange the presented error prior in eq.(4.10) in the form of SBL, we will obtain:

$$p(\boldsymbol{\eta}|\boldsymbol{\theta}) = p_N(\boldsymbol{\eta}|\boldsymbol{\theta})p_{sbl}(\boldsymbol{\theta}) \tag{4.26}$$

$$= \mathcal{N}(\mathbf{0}, (\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}))\exp(-\Psi(\boldsymbol{\theta})) \tag{4.27}$$

where,

$$\Psi(\boldsymbol{\theta}) = \log\frac{\alpha^n}{Z} + \frac{1}{2}\log|\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}| + \frac{2-p}{2}(\frac{\alpha^2}{p})^{\frac{p}{p-2}}\sum_i \lambda_i^{\frac{p}{p-2}} \tag{4.28}$$

In this form, the presented prior is similar to the SBL-variant presented in [111], where the prior covariance is represented as a linear combination of basis matrices with unknown weights modeled with a hyperprior. Here, the precision instead of covariance matrix is expressed as a linear combination of basis matrices, with the basis matrices being the outer product of the columns($\boldsymbol{d}_i$) of the gradient matrix $\mathbf{D}^T$, i.e. the precision matrix is given by $\sum_{i=1}^{n+1}\boldsymbol{\lambda}_i\boldsymbol{d}_i\boldsymbol{d}_i^T$. This results naturally from assuming the gradient of TMP (wavefront) to be sparse.

Parameter estimation for the additional row of D

As described earlier, we add one more row of ones to the matrix $\mathbf{D}$, *i.e.,* $\boldsymbol{d}_{n+1} = \mathbf{1}$. Our inference procedure alternatingly estimates parameters and random variables.

Parameters $\boldsymbol{\lambda}_i$s are estimated from the ECG through $\boldsymbol{\eta}$ (see line 16 of Algorithm 1). Then $\boldsymbol{\eta}$ and $\boldsymbol{u}$ are updated according to $\boldsymbol{\lambda}_i$ (see line 8 through 13 in Algorithm 1). The whole precision matrix, given by $\sum_{i=1}^{n+1} \boldsymbol{\lambda}_i \boldsymbol{d}_i \boldsymbol{d}_i^T$ affects how much the inverse TMP estimate ($\mathbf{u}$) should adjust prediction of dynamic model ($\mathbf{u}_d$) according to ECG data ($\mathbf{y}$) (see line 10 of Algorithm 1). Intuitively, when the value of $\boldsymbol{\lambda}_i \boldsymbol{d}_i \boldsymbol{d}_i^T$ is high, less correction will occur for the i-th element in $\boldsymbol{d}_i^T \mathbf{u}$ and the estimated value will be more heavily determined by the model prediction. However, since the vector of ones, i.e. $\mathbf{1}$, lies in the null space of the forward matrix $\mathbf{H}$, the last $\boldsymbol{\lambda}_{n+1}$ – corresponding to $\mathbf{1}$ added to matrix $\mathbf{D}$ – cannot be estimated from the ECG during our inference. Therefore, we heuristically set this $\boldsymbol{\lambda}_{n+1}$ high such that when $\mathbf{u}$ is estimated, the bias $\mathbf{1}^T\mathbf{u}$ is only minimally corrected with respect to prediction from previous time instant. This is based on the assumption that initial $\mathbf{u}$ we start from has the bias ($\mathbf{1}^T\mathbf{u}$) approximately correct and we maintain the bias in the same range throughout. This is a reasonable assumption because 1) we are focusing on the error in the gradient of $\mathbf{u}$, and 2) we do not have any other source of information to learn this bias. Note that we want $\boldsymbol{\lambda}_{n+1}$ to be sufficiently high but not too high so as to put heavy constraint on the inverse estimate. To maintain a high value of $\boldsymbol{\lambda}_{n+1}$, we always keep it above a threshold $\boldsymbol{\lambda}_{threshold}$. Above this threshold, we set $\boldsymbol{\lambda}_{n+1}$ to be $max(\boldsymbol{\lambda}(1:n))$. This helps in gradually increasing $\boldsymbol{\lambda}_{n+1}$ over the iteration as other values of $\boldsymbol{\lambda}$ increase and reaches much high value than $\boldsymbol{\lambda}_{threshold}$.

4.6 Synthetic Experiment 1: Errors in Model Parameters

We first evaluate the ability of the presented method to detect and correct model errors arising from model parameters that represent tissue properties. In specific, we consider the presence of local myocardial infarcts unknown to the prior physiological model. Experiments were carried out on three image-derived heart-torso models, including 34 settings of myocardial infarcts of various sizes and locations in the ventricles. In specific, we divided each left ventricle into 17 segments according to the American

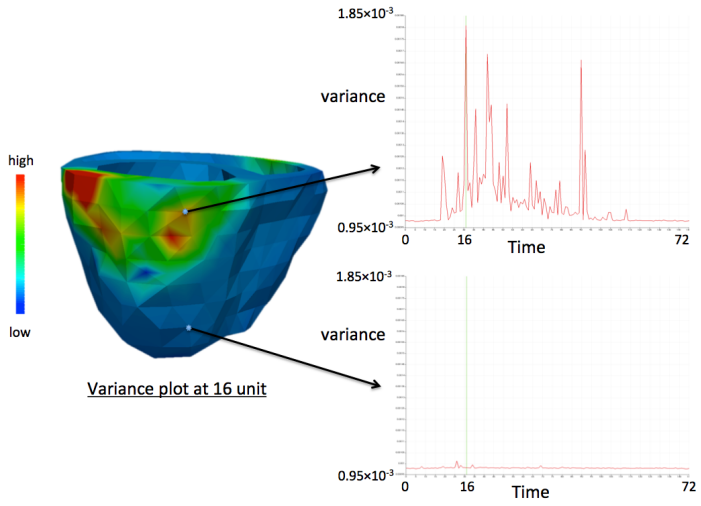


Figure 4.5: Examples of variance plots. Left: spatial plot at one time instant. Right: Temporal plot at selected locations.

Heart Association (AHA) recommendations [82], and set each infarct to two of the 17 segments. 120-lead ECG data was then simulated and corrupted with 20dB noise for inverse reconstruction of 3D transmural TMP signals. The inverse reconstruction utilized a prior EP model without knowledge of the presence of the infarct. From the reconstructed TMP signals, activation time was calculated as the time of the steepest upstroke and the region of infarct was extracted from where TMP signals have duration below 50% of normal values. Quantitative accuracy of the solutions was evaluated using two metrics: correlation coefficient between the true and reconstructed activation time, and Dice coefficient between the true and estimated regions of infarcts. Using these metrics, we also compared the performance of the presented method against model-constrained EP reconstruction without correcting model errors as described in [109].

Figure 4.4 shows examples of the simulated and reconstructed TMP sequences. As shown in the ground truth (bottow row), the TMP propagation was blocked at the region of an infarct located at the basal infero-lateral region of the heart. Without model error correction (top row), the reconstructed TMP sequence was not able to reflect this conduction block until after the depolarization stage. In comparison, the

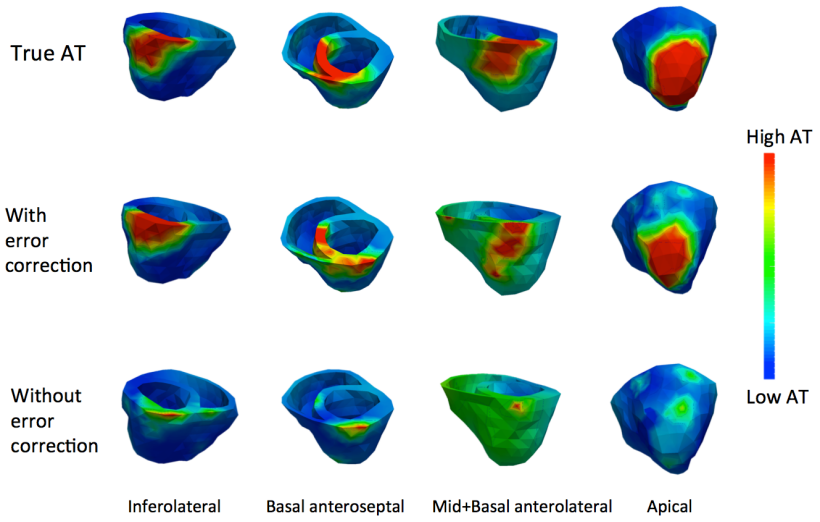


Figure 4.6: Comparison of activation time reconstructed with and without model error correction at different scar settings.

presented method (middle row) was able to detect and correct the prior model error at an early stage of the depolarization, capturing the conduction block at the correct location of the heart.

To better understand how model uncertainty helps in correcting the posterior estimate of TMP, we note that, in line 10 of Algorithm 1, the prediction from the previous time instant, $\mathbf{u}_d$ is multiplied by the precision matrix $\mathbf{P}_u$, i.e., the inverse of the covariance matrix $\boldsymbol{\Sigma_d} + (\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D})^{-1}$. The covariance matrix is the sum of the propagated uncertainty $\boldsymbol{\Sigma_d}$ from the previous step and model error $(\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D})^{-1}$ estimated at this time step, capturing true uncertainty in the model predicted TMP. We plot variances, diagonal elements of the covariance matrix, in Fig. 4.5. As shown, the presented method detected high uncertainty (variance) in the predicted TMPs at the infarcted region, but low uncertainty at the healthy region. Also note that the variance was particularly high at the boundary of the infarct, which was a natural result of modeling the prediction error to be sparse in the spatial gradient domain.

Fig. 4.6 shows additional examples of activation time maps derived from the reconstructed TMP sequence, with and without model error correction, in comparison to

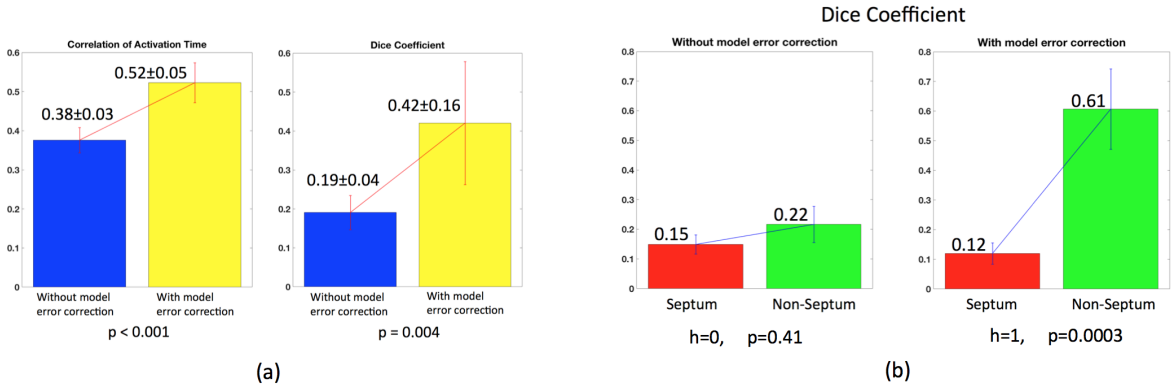


Figure 4.7: a) Quantitative comparisons of reconstructions obtained with and without error correction, at the presence of infarcts unknown to the prior model. b) Quantitative comparisons when reconstructing septal and non-septal infarcts. Left: without model error correction; Right: with model error correction

the simulated ground truth. As shown, at the presence of infarcts at different locations of the heart, the presented method was able to more closely reconstruct the conduction block despite the absence of knowledge about these infarcts in the prior EP model.

Fig. 4.7 summarize the quantitative comparison between the results obtained with and without model error correction. As shown in Fig. 4.7.a., accuracy of the presented method – in both activation time and infarct detection – is significantly higher than that without error correction. Noting the high standard deviation of the presented results in Fig. 4.7.a., we further compare the performance (Dice coefficient) of the methods regarding whether the infarcts were septal. As shown in Fig.4.7.b., 1) in both methods, the performance was poor when the infarct is septal, and 2) the correction of model error brought a significant improvement in accuracy when the infarct was non-septal. This suggests that the ability to reconstruct septal information in the heart may be fundamentally limited by its observability in surface ECG data, while for cases where this observability is not an issue, the presented method performs well. We further analyze the sensitivity of algorithm on infarct settings in greater detail in section 4.10.

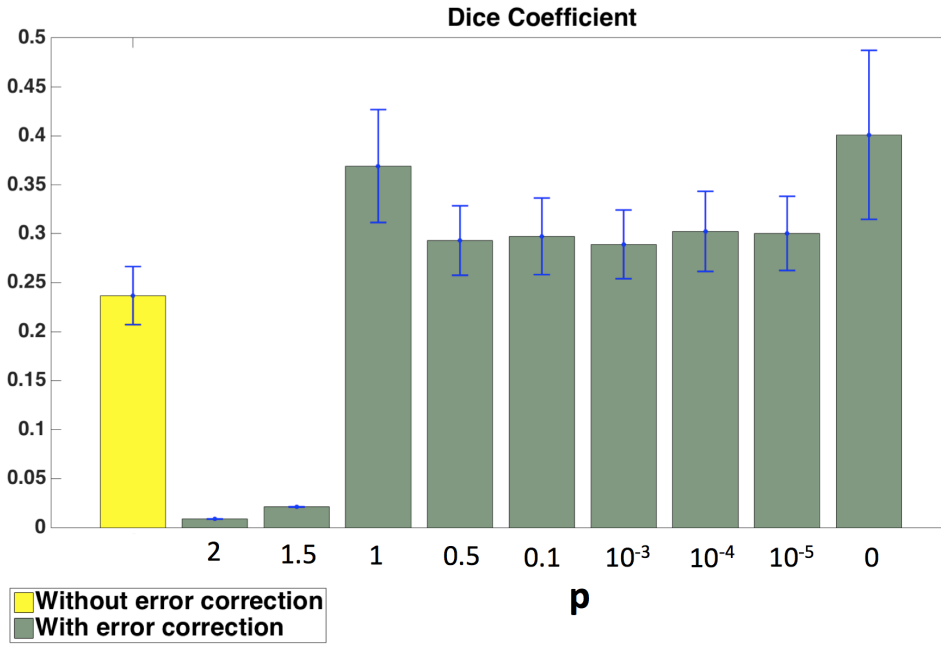


Figure 4.8: Sensitivity of the presented algorithm with respect to different values of p in the generalized Gaussian prior

4.7 Sensitivity Analysis

Sensitivity to the value of p

The generalized normal distribution would enforce sparsity for values $0 \leq p \leq 1$. We performed analysis on a single geometry to understand the sensitivity of the presented algorithm to different values of p ranging between 0 and 2. As shown in Fig. 4.8, with any value of p within the range of $0 \leq p \leq 1$, the presented method performed better than that did not consider model error correction. However, contrary to expectation, the performance of the presented method did not improve as we decreased p from 1 to 0. In fact, the presented method performed better when $p = 1$ in comparison to $0 < p < 1$, although the best was obtained at the limiting case $p \to 0$ as derived before. We report results using this limiting case of the algorithm throughout this paper.

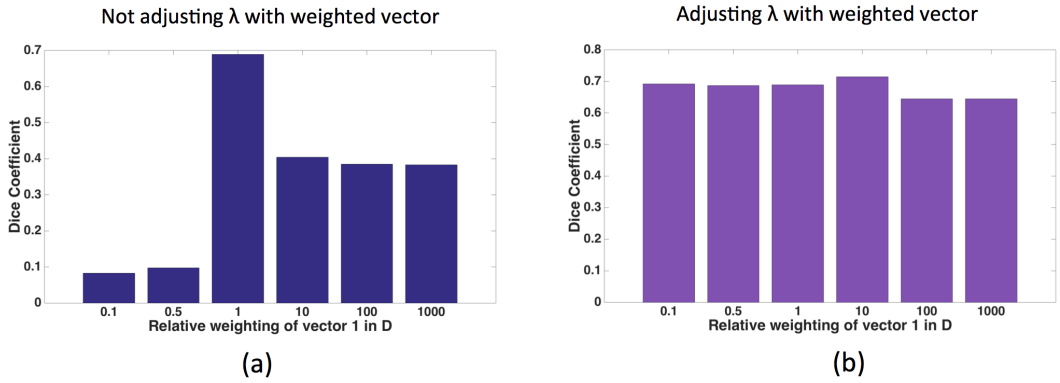


Figure 4.9: Sensitivity of the presented algorithm with respect to different weighting factors of the added vector of ones to the gradient matrix $\mathbf{D}$.

Sensitivity to the added vector in D

As described in subsection 4.5, to preserve a bias term that is applicable to the prior electrophysiological model, we added one row of ones to the gradient operator $\mathbf{D}$ and heuristically updated the corresponding $\boldsymbol{\lambda}_{n+1}$. The vector of ones and updating heuristic of $\boldsymbol{\lambda}_{n+1}$ was empirically found to work well. To understand the effect of this strategy on the presented algorithm, we investigated two scenarios. First, we replaced the vector of ones with different weighting factors while keeping the value of $\boldsymbol{\lambda}_{n+1}$ fixed. The performance of the presented algorithm is summarized in Fig. 4.9(a). As shown, the performance dropped when the weighting factor was either higher or lower than one, although the drop was much more significant if the weighting factor was less than one. This is because a high weighting factor imposes too high a bias constraint and does not allow much change to TMP in accordance to ECG, while a low weighting imposes a weak bias constraint which may allow the TMP solution to wander beyond a feasible range. The latter causes a much bigger problem because, once the value of TMP goes beyond the range feasible for the prior electrophysiological model, the model prediction becomes unstable and may even crash.

We then tested the second scenario where we adjusted $\boldsymbol{\lambda}_{n+1}$ accordingly when multiplying the row of ones with a weighting factor. As summarized in Fig. 4.9(b), with

simultaneous adjustment of $\boldsymbol{\lambda}_{n+1}$ following the strategy adopted in the presented algorithm, the performance remained more or less unchanged over the range of weighting factors tested.

4.8 Synthetic Experiments 2: Errors in Initial Conditions

We then evaluate the ability of the presented method to detect and correct model errors arising from the initial condition of the prior EP model – locations of the earliest excitation points in the ventricles. In each of the three patient-specific geometrical models, we considered the following error settings: 1) the prior EP model missed one excitation point from the ground truth, 2) the prior EP model included an extra excitation point not in the ground truth, and 3) the excitation point in the prior EP model was at a different location from the ground truth. In all cases, 120-lead ECG data was simulated and corrupted with 20dB noise for TMP reconstruction. Quantitative accuracy of the reconstructed TMP sequence in comparison to the ground truth was measured by two metrics: 1) normalized mean square error, and 2) correlation coefficient.

Fig. 4.10 shows an example of the reconstructed and simulated TMP sequence where the simulated TMP started from two excitation points while the TMP reconstruction was constrained by a prior EP model starting with only one of the excitation points. While the reconstructed TMP was unable to capture the missing excitation point without model error correction, the presented method was able to quickly correct that error 20ms into the depolarization. In comparison, as shown in Fig. 4.11, we found that it was more difficult for both methods to correct an extra excitation point that was not in the ground truth. Quantitative comparison between the two methods is summarized in Fig. 4.12, showing a statistically significant improvement brought by the presented method.

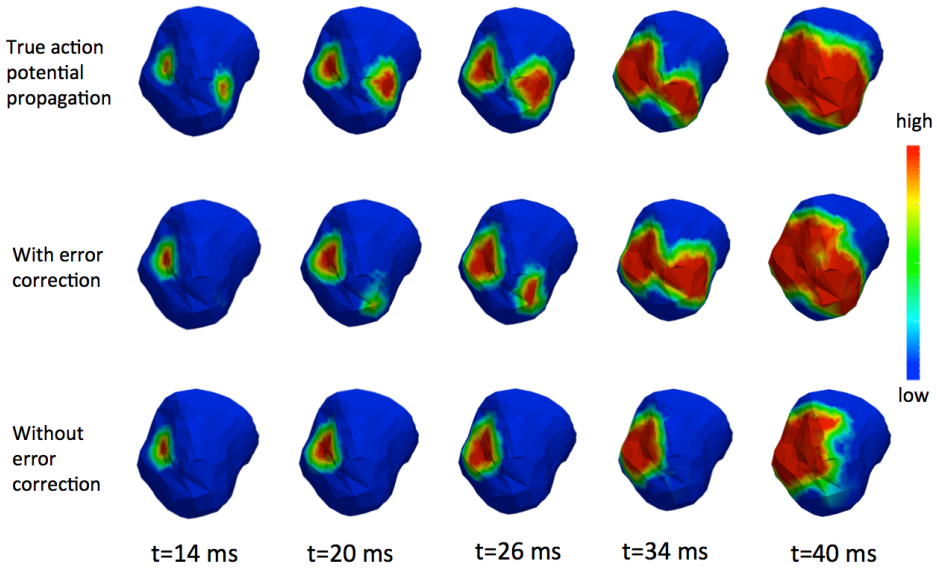


Figure 4.10: Reconstructed versus true TMP propagation when the prior model missed one of the two excitation points.

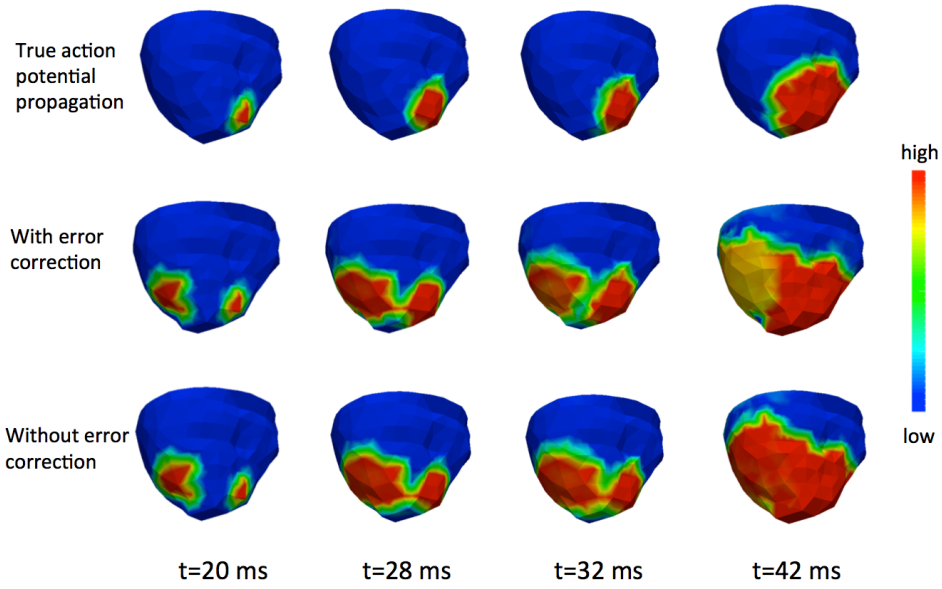


Figure 4.11: Reconstructed versus true TMP propagation when the prior model included an extra excitation point absent in the ground truth.

4.9 Real Data experiments

We performed real data experiments on two patients who underwent catheter ablation

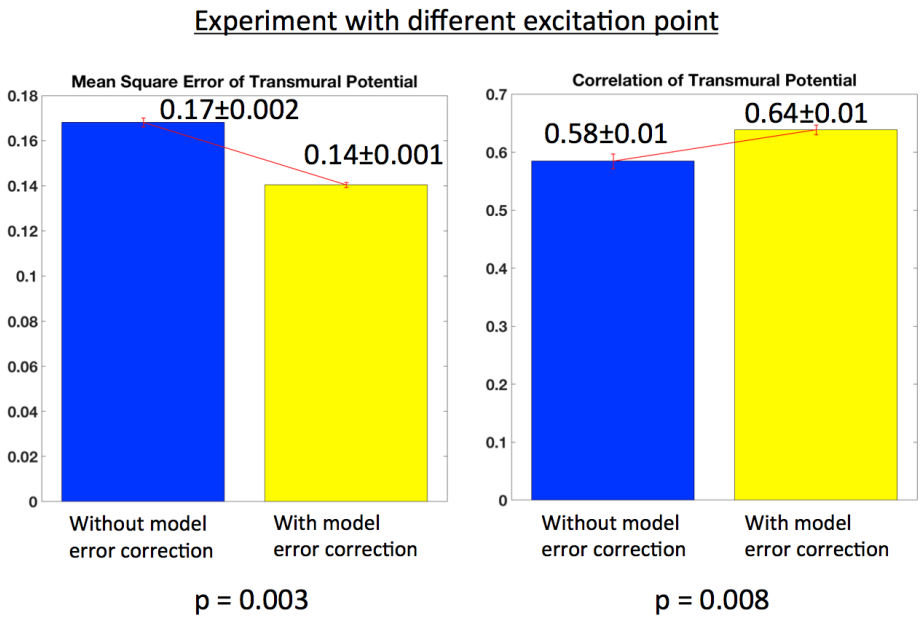


Figure 4.12: Quantitative comparisons of reconstructions obtained with and without model error correction, at the presence of model errors in excitation points.

Table 4.1: Dice Coefficients between infarcted regions extracted from reconstructed TMP and bipolar voltage maps.

	Case 1	Case 2
Without Model Error Correction	0.2406	0.1053
With Model Error Correction	**0.3053**	**0.2237**

due to post-infarction ventricular arrhythmia [92]. For each patient, patient-specific heart-torso geometry was extracted from CT images, on which transmural TMP signals were reconstructed from 120-lead ECG data acquired during sinus rhythm. From the reconstructed TMP signals, the region of infarct was identified as where the duration of TMP falls below 50% of the normal value. The obtained region of infarct was compared with in-vivo bipolar voltage maps, and reconstructions obtained with and without model error correction were compared.

These results are visually summarized in Fig. 6.6 and quantitatively summarized in

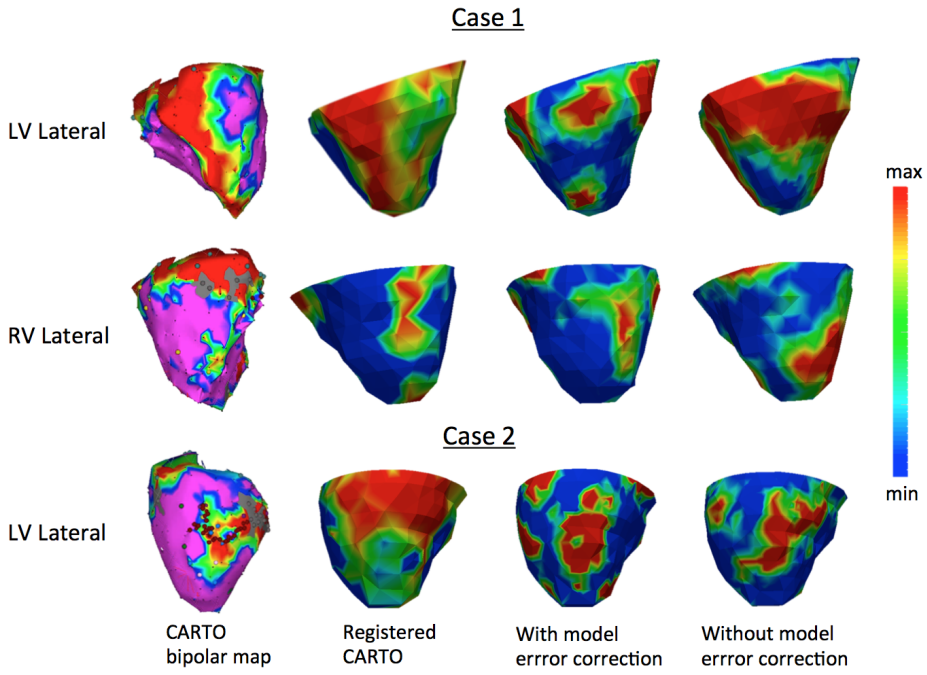


Figure 4.13: Regions of infarcts extracted from reconstructed TMP sequences in reference to in-vivo bipolar voltage maps.

Table 4.1 in terms of the Dice coefficient between the detected infarct region and the low-voltage region (≤ 1.5 mV) in bipolar voltage maps. In case 1, infarct reconstructed considering model error (third column) is visually closer to the bipolar voltage map registered to the CT-derived mesh. In case 2, both reconstructions were visually less consistent with the bipolar voltage map. The Dice coefficients in Table 4.1 suggests that both methods performed less satisfactorily in case 2, although the presented method was able to bring evident improvement in both cases.

We noted that the improvement in inverse reconstructions brought by the presented method was not as significant in real data as it was in simulated data. This might be attributed to several reasons. First, the forward matrix H was treated as known in simulated experiments while, in real-data experiments, the accuracy of inverse reconstructions is directly affected by the errors in the forward matrix itself. Second, if the error between the true TMP propagation and that from the prior EP model is too

high, the assumption of sparse model error might not hold. This error can come from multiple sources. For example, realistic infarcts may have much more complex spatial distributions than the simple-shaped infarcts used in simulated experiments. The number and location of excitation points are also less predictable in real-data experiments in comparison to simulated settings. Third, the correspondence of bipolar voltage to the reconstructed TMP sequence is not straightforward. As a result, we resorted to a secondary comparison where we identified infarcts from both data for comparison. These intermediate steps might be another source of errors. Finally, the registration of the bipolar voltage map to CT-derived mesh may introduce additional errors that further compounded the validation process.

4.10 Discussion

4.10.1 Algorithm Performance vs. Error Observability

As observed in section IV-A, the performance of the presented method changes with the location of the infarct. If we decompose the observed ECG for each case into the following two components: $\mathbf{y}_k = \mathbf{H}(\mathbf{u}_{prediction} + \boldsymbol{\eta}) = \mathbf{y}_{prediction} + \mathbf{y}_{\boldsymbol{\eta}}$, it is clear that — given the same dynamic prediction model unaware of any infarct settings — the difference in ECG data from different infarct settings are introduced by the model error $\boldsymbol{\eta}$. Therefore, here we attempt to rationalize how the observation of error $\boldsymbol{\eta}$ on ECG might be related to the quality of the estimation results. To do so, we revisit the approach for maximum-likelihood estimation of parameter $\boldsymbol{\theta}$ derived in eq.(4.12), and reformulate it to focus on the error observation in ECG $\mathbf{y}_\eta$ (rather than the overall ECG observation $\mathbf{y}$ used in eq. (4.12).

Consider that $\boldsymbol{\eta}$ is observed on the surface ECG data as the data error $\mathbf{y}_{\boldsymbol{\eta}} = \boldsymbol{H}\boldsymbol{\eta}$, we have $p(\mathbf{y}_{\boldsymbol{\eta}}|\boldsymbol{\eta}, \beta) = \mathcal{N}(\mathbf{y}_{\boldsymbol{\eta}}|\mathbf{H}\boldsymbol{\eta}, \beta^{-1}\mathbf{I})$, where the prior density of $\boldsymbol{\eta}$ is characterized by hyperparameters $\boldsymbol{\lambda}$ and α as defined in eq.(4.27) and eq.(4.28). As mentioned in section 4.3, our optimization scheme is to first obtain a parameter that maximizes the

likelihood of $\mathbf{y}$ after marginalizing over intermediate random variables. Following the same line of derivation, we marginalize over $\boldsymbol{\eta}$ to obtain $\mathbf{y}_{\boldsymbol{\eta}}$ as a Gaussian distribution characterized by $\boldsymbol{\lambda}$ and α as $p(\mathbf{y}_{\boldsymbol{\eta}}|\boldsymbol{\Lambda}, \beta) = \mathcal{N}(\mathbf{y}_{\boldsymbol{\eta}}|\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}})$, where:

$$\boldsymbol{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}} = \beta^{-1}\mathbf{I} + \mathbf{H}(\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D})^{-1}\mathbf{H}^T \tag{4.29}$$

Note that marginalization over $\boldsymbol{\eta}$ is now possible because unlike before, where we also had another latent variable $\mathbf{u}$, now we only have $\boldsymbol{\eta}$ as latent variable. We obtain parameter estimation equations as

$$\begin{aligned}\hat{\boldsymbol{\lambda}}, \hat{\alpha} &= \underset{\boldsymbol{\lambda},\alpha}{argmax} \quad \log[p(\mathbf{y}_{\boldsymbol{\eta}}|\boldsymbol{\Lambda}, \beta)p(\boldsymbol{\lambda}, \alpha)] \\ &= \underset{\boldsymbol{\lambda},\alpha}{argmin} \quad [\mathbf{y}_{\boldsymbol{\eta}}^T\boldsymbol{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}}^{-1}\mathbf{y}_{\boldsymbol{\eta}} + \log|\boldsymbol{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}}| + 2\Psi(\boldsymbol{\lambda}, \alpha)]\end{aligned} \tag{4.30}$$

Now, we want to understand how this optimization leads to better performance in certain error cases than others. Note that we want to analyze difference in performance with respect to ECG error $\mathbf{y}_{\boldsymbol{\eta}}$. In estimating optimal $\hat{\boldsymbol{\lambda}}$, the only term that constrains $\boldsymbol{\lambda}$ to fit ECG data error is the first term, $\mathbf{y}_{\boldsymbol{\eta}}^T\boldsymbol{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}}^{-1}\mathbf{y}_{\boldsymbol{\eta}}$, which we call data fitting constraint. We decompose it into terms that do and do not depend on $\boldsymbol{\lambda}$ in following result.

Result 1. *If $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ is the singular value decomposition, $\boldsymbol{z} = \boldsymbol{U}^T\boldsymbol{y}_{\eta}$, and $\langle ., . \rangle$ denotes inner product, then,*

$$\boldsymbol{y}_{\boldsymbol{\eta}}^T\boldsymbol{\Sigma}_{\boldsymbol{y}_{\boldsymbol{\eta}}}^{-1}\boldsymbol{y}_{\boldsymbol{\eta}} = \beta\boldsymbol{y}_{\boldsymbol{\eta}}^T\boldsymbol{y}_{\boldsymbol{\eta}} - \left\langle \boldsymbol{z}\boldsymbol{z}^T, \boldsymbol{S}(\beta^{-1}\boldsymbol{V}^T\boldsymbol{D}^T\boldsymbol{\Lambda}\boldsymbol{D}\boldsymbol{V} + \boldsymbol{S}^T\boldsymbol{S})^{-1}\boldsymbol{S}^T \right\rangle$$

Proof is in the appendix D.

Our argument here is that if there are multiple minima, then it would be difficult for the algorithm to find the true minima. Let's say $\boldsymbol{\lambda}^*, \alpha^*$ minimizes eq.(4.30) and let $\mathbf{y}_{\boldsymbol{\eta}}\boldsymbol{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}}^{-1}\mathbf{y}_{\boldsymbol{\eta}} = C^*$ at this minima. Because only the second term in Result 1 depends on $\boldsymbol{\lambda}$, the data fitting constraint $\mathbf{y}_{\boldsymbol{\eta}}\boldsymbol{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}}^{-1}\mathbf{y}_{\boldsymbol{\eta}} = C^*$ is satisfied by all the $\boldsymbol{\lambda}$s such that the inner product $\left\langle \boldsymbol{z}\boldsymbol{z}^T, \mathbf{S}(\beta^{-1}\boldsymbol{V}^T\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}\boldsymbol{V} + \mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T \right\rangle$ remains unchanged. Note that in this inner product, zero elements in $\boldsymbol{z}$ will mask the matrix $\mathbf{S}(\beta^{-1}\boldsymbol{V}^T\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}\boldsymbol{V} + \mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T$ that contains $\boldsymbol{\lambda}$. Therefore, if $\boldsymbol{z}$ is highly sparse, there will be a large number of $\boldsymbol{\lambda}$ values that could satisfy data fitting constraint, and therefore would be the minimizer

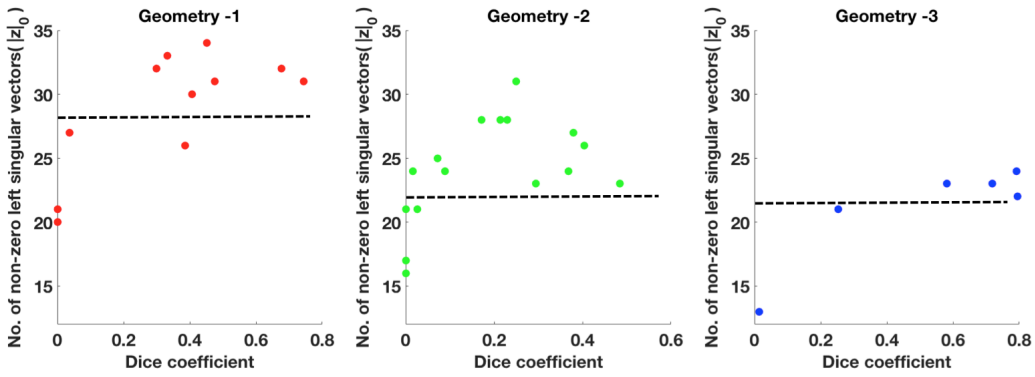


Figure 4.14: Values of $|\mathbf{z}|_0 = |\boldsymbol{U}^T\mathbf{y}_\eta|_0$ versus Dice coefficient in three geometrical models, where $\boldsymbol{U}$ is a matrix of left singular vectors in $\mathbf{H}$.

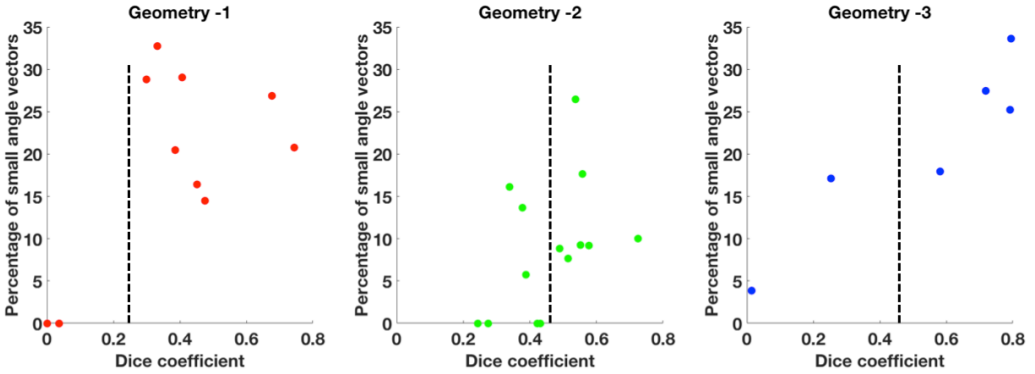


Figure 4.15: Percentage of relevant vectors (columns in $\mathbf{H}$ that have a small angle with the ECG error vector) in the reconstructed region of infarct.

of eq.(4.30), thereby increasing the feasible solution space. We use L0 norm of $\boldsymbol{z}$, denoted by $|\boldsymbol{z}|_0$ to quantify how dense the vector is. Above analysis suggests that a lower value of $|\boldsymbol{z}|_0 = |\boldsymbol{U}^T\mathbf{y}_\eta|_0$ – a small number of left eigenvectors of $\mathbf{H}$ present in $\mathbf{y}_\eta$ – may correspond to a higher difficulty to find the true optimum. Note that a lower value of $|\boldsymbol{z}|_0$ is only sufficient to ensure multiple minima, but is not necessary because, even if $\boldsymbol{z}$ is not sparse at all, the inner product might be small depending on the alignment of two matrices.

To test this hypothesis, we carried out experiments in various settings of infarcts con-

sidered in section IV. In each case, we set the vector $\boldsymbol{\eta}$ to be one in the infarct and zero elsewhere. We calculated $\mathbf{y}_\eta = \mathbf{H}\boldsymbol{\eta}$ and then plot $|\boldsymbol{z}|_0 = |\boldsymbol{U}^T\mathbf{y}_\eta|_0$ against the Dice coefficient of the solution obtained earlier. As shown in Fig. 4.14, we observed that whenever $|\boldsymbol{z}|_0$ is low – for example below the threshold annotated in figure – the Dice coefficient of the obtained results is also low. This is in agreement with our hypothesis. Additionally, we found that most of the settings with septal infarcts had a low value of $|\boldsymbol{z}|_0$, explaining the difficulty of reconstruction in these cases. For the cases where $|\boldsymbol{z}|_0$ is high, however, the Dice coefficient is mixed. This result of mixed dice coefficient for high $|\boldsymbol{z}|_0$ is also consistent with our argument above that the smaller value of $|\boldsymbol{z}|_0$ is only sufficient but not necessary. This is because from Result 1 higher value of $|\boldsymbol{z}|_0$ may also lead to smaller inner product depending on the two matrices. Thus, the experimental results support our theory.

We do caution that the above tests were conducted with the following simplifications: 1) we assumed that the model error is either zero (in healthy region) or one (in infarcted region), 2) we considered only one time instant, and 3) we assumed that the Dice coefficient is a direct measure of the reconstruction accuracy. More rigorous experimental testings may be devised in the future for the presented theoretical analysis.

4.10.2 Relation to Relevance Determination

We now examine the presented method from the perspective of relevance determination, and lay out its similarity with and difference from relevance vector machines [102]. As the matrix $\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}$ in eq. (4.29) is approximately block diagonal, we re-express the covariance matrix of the data error $\mathbf{y}_\eta$ as: $\boldsymbol{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}} = \beta^{-1}\mathbf{I} + \sum_k \mathbf{H}_k\mathbf{A}_k^{-1}\mathbf{H}_k^T$ where $\mathbf{A}_k$ is the k-th block of $\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}$. Following the reasoning in [102], if data error $\mathbf{y}_\eta$ is generated by a Gaussian distribution, the empirical covariance $\mathbf{y}_\eta\mathbf{y}_\eta^T$ must be approximately equal to the covariance $\boldsymbol{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}}$. Therefore, if $\mathbf{A}_k$ were a 1×1 block (say a_k), we would be estimating a_k such that $\beta^{-1}\mathbf{I} + \sum_k h_k a_k^{-1} h_k^T$ matches the $\mathbf{y}_{\boldsymbol{\eta}}\mathbf{y}_{\boldsymbol{\eta}}^T$. This would be the same as the relevance vector machines [102] and automatic relevance determination [69], which work by selecting relevant columns h_k's that are most closely aligned to the data

vector while driving the rest a_k^{-1} towards 0. Here, given the block matrices $\mathbf{A}_k$s that couples the columns of $\mathbf{H}$, we speculate that, instead of a single column, the presented method is forced to choose a set of columns such that the covariance is close to the data covariance. Consequently, only a portion of the columns in the solution will be closely aligned to the data vector due to block selection.

To experimentally examine this mechanism of block selection, we carried out experiments in a setting similar to that described in the previous subsection. In each experiment, We computed the angle between each column in $\mathbf{H}$ and $\mathbf{y}_\eta$, focusing particularly on those columns with small angles to $\mathbf{y}_\eta$ (i.e., relevant vectors). Fig. 4.15 shows the percentage of small-angle columns out of all columns in the reconstructed infarct, plotted against the Dice coefficient. We note that having a higher percentage of relevant vectors in the solution was related to higher Dice coefficients, suggesting the nature of relevance determination in the presented method. At the same time, we also note that the percentage of relevant vectors remains moderate even when the Dice coefficient is high, supporting our speculation of block selection.

4.10.3 Limitations and Future Work

We observed limited performance of the presented method in real data experiments. Compared to synthetic experiments where the prior model error was controlled to one source, model errors in real data experiments can arise from multiple sources, such as the error in the prior dynamic model, and the error in the forward measurement model that relates the TMP in the heart to ECG data on the body surface. Investigation of methods that can detect and correct errors in the forward measurement model is an interesting direction of future work, such as those presented in [29].

We may need to consider additional prior knowledge about the error or better model error, for example, by considering temporal correlation. Future work may also consider an alternative approach to incorporate prior physiological knowledge, for example, through a data-learnt generative model extracting knowledge from physiological models but with latent factors that can be more easily adapted to ECG data while retaining

complex relationship [32].

With an interest in understanding why the presented method performs differently in different cases, we presented mathematical justifications and initial empirical support that the performance of the presented method is related to how the model error is observed on ECG data. We hope that this result will encourage researchers in inverse electrophysiological imaging to look closely into the relatively unexplored area of how and why a new reconstruction method performs differently in different pathological conditions. This also raises an open question: can we devise reconstruction methods for electrophysiological imaging that are less sensitive (in terms of performance) to the particular type of clinical application of interest?

Finally, the presented method performs inference sequentially using only past ECG data. This is largely limited by the nature of the prior EP model as it is not possible to reverse the model in time. This further suggests alternatives means to extracting knowledge from the EP models without explicitly utilizing these models within the inference.

4.11 Conclusions

We presented a Bayesian framework to jointly infer from ECG data the posterior distribution of TMP signals and the error in the prior EP model, exploiting the sparse nature of error in the gradient domain. We have shown that by considering and correcting the error in the prior model, we can improve TMP reconstruction. Future work will focus on alternative means to incorporating prior physiological knowledge such that the model elements to be estimated from ECG data is more expressive in generating the TMP sequence.

4.12 Summary and Answers to Research Questions

We observed that we could improve generalization by incorporating the prior knowledge about the spatio-temporal dynamics of TMP as well as knowledge of sparsity in the gradient domain. However, there were some challenges in realizing this idea. To answer the challenges we posed these two questions:

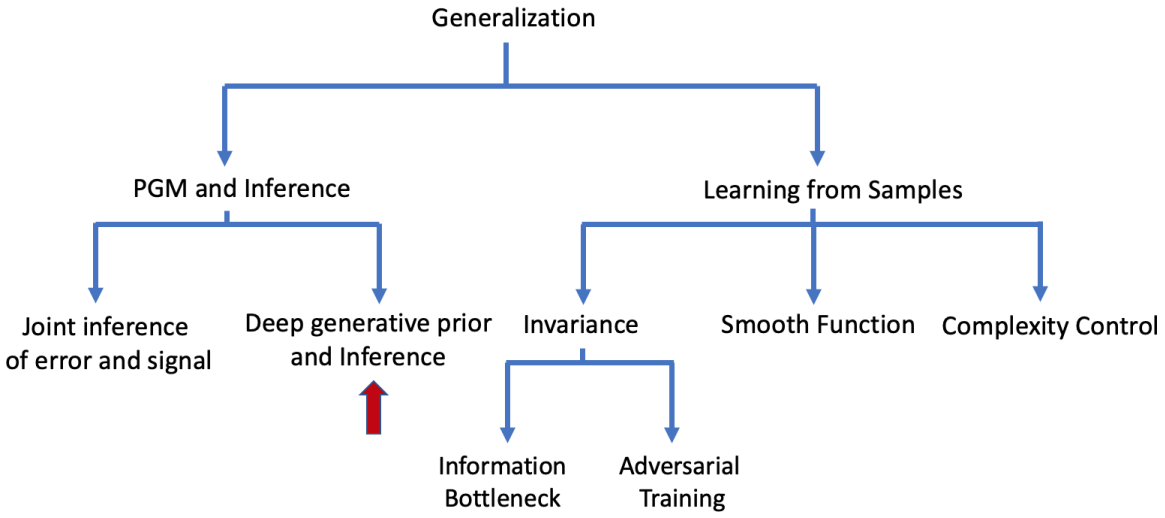
Q. 1.a) How can the population knowledge be adapted for patient specific inference?
Q. 1.b) How can the prior knowledge about the sparsity in the gradient domain and dynamics of TMP signals be combined in principled way?

In this chapter, we answered both of these questions by introducing an error random variable in the graphical model and solving for the joint distribution of the inverse signal and the error via clever combination of variational Bayes and expectation maximization. With this modeling and inference, we showed that the inverse estimate has good generalization abilities in different settings of synthetic and real data experiments. At the end of the chapter, we also investigated that there is certain inductive bias in the algorithm because of assumptions (for example sparsity) and modeling of the solution. Therefore, the solution works better in certain situations than others.

Chapter 5

Learning by Adapting Deep Generative Model

How is it we have so much information, but know so little?
- Noam Chomsky



5.1 Introduction

To solve the inverse problem of estimating patient specific TMP from ECG data, we want to generalize the prior knowledge about the dynamics of TMP. To do so, we employ a model constrained inference framework where the dynamics of TMP is represented in the form of EP simulation model based on differential equations. But, these models are controlled by high-dimensional parameters often associated with local tissue properties and the origin of electrical activation that are unknown a priori. To fix these model parameters in optimization/inference, as is common in existing approaches, model errors may be introduced decreasing the accuracy of the estimated electrical activity [109]. To adapt these model parameters to the observed data, as is desired for accurate inference, is however difficult due to their high-dimensionality and nonlinear relationship with the observed ECG data [38]. In the last chapter we focused on estimating errors introduced in the model due to error in parameters. In this chapter, we take a different approach; replace the conventional physiological models with a deep generative model that is trained to generate the spatiotemporal dynamics of transmembrane potential (TMP) from a low-dimensional set of generative factors. These generative factors can be viewed as a low-dimensional abstraction of the high-dimensional physical parameters, which allows us to efficiently adapt the prior physiological knowledge to the observed ECG data (through inference of the generative factors) for an improved reconstruction of TMP dynamics.

In specific, the presented method consists of two novel contributions. First, to obtain a generative model that is sufficiently expressive to reproduce the temporal sequence of 3D spatial TMP distributions, we adopt a novel sequence-to-sequence variational auto-encoder (VAE) [16] with cascaded long short-term memory (LSTM) networks. This VAE is trained on a large database of simulated TMP dynamics originating from various myocardial locations and with a wide range of local tissue properties. Second, once trained, the VAE decoder describes the likelihood of the TMP conditioned on a low-dimensional set of generative factors, while the encoder learns the posterior distributions of the generative factors conditioned on the training data. We utilize these two components within the Bayesian inference, and present a variation of the

expectation-maximization (EM) algorithm to jointly estimate the generative factors and transmural TMP signals from observed ECG data. In a set of synthetic and real-data experiments, we demonstrate that the presented method is able to improve the accuracy of transmural EP imaging in comparison to statistical inference either constrained by a conventional physiological model [109] or without physiological constraints.

This chapter includes parts from author's conference publications [32, 39].

5.2 Generative Modeling of TMP via Sequential VAE

To learn to generate the spatiotemporal TMP sequences, we use a sequential variation of VAE [58] based on the use of LSTM networks [16].

5.2.1 VAE Architecture:

The architecture of the sequential VAE is summarized in the red block in Fig. 5.1. Both the encoder and the decoder consists of two layers of LSTM, where the second layer includes separate mean and variance networks. The spatial dimension decreases from the original TMP signal $\mathbf{U}$ to the latent representation $\mathbf{Z}$, while the temporal relationship is modeled by the LSTMs. Note that while the random variables in a standard VAE are vectors, a sequential VAE deals with matrices. By defining the conditional distribution of a matrix as the product of distributions over its columns, we obtained the likelihood distribution $p_\theta(\mathbf{U}|\mathbf{Z})$ and the variational posterior distribution $q_\phi(\mathbf{Z}|\mathbf{U})$ as:

$$p_\theta(\mathbf{U}|\mathbf{Z}) = \prod_k \mathcal{N}(\mathbf{U}_{:,k}|\mathbf{M}_\theta(\mathbf{Z})_{:,k}, diag(\mathbf{S}_\theta(\mathbf{Z})_{:,k})) \tag{5.1}$$

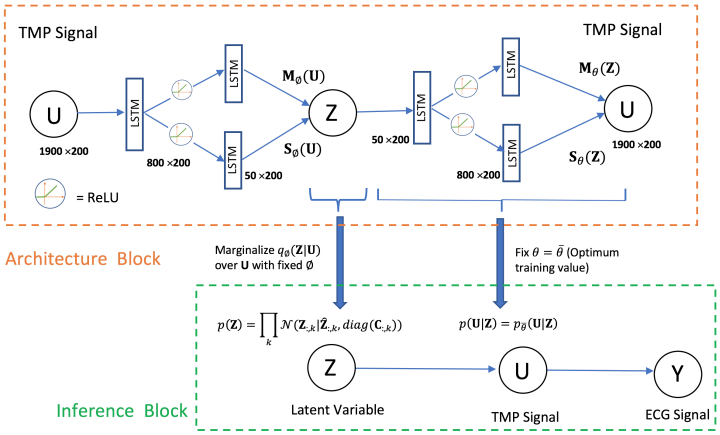


Figure 5.1: Red block: VAE architecture. Green block: graphical model in inference.

$$q_\phi(\mathbf{Z}|\mathbf{U}) = \prod_k \mathcal{N}(\mathbf{Z}_{:,k}|\mathbf{M}_\phi(\mathbf{U})_{:,k}, diag(\mathbf{S}_\phi(\mathbf{U})_{:,k})) \tag{5.2}$$

where $\mathbf{M}_\phi(\mathbf{U})$ and $\mathbf{S}_\phi(\mathbf{U})$ are output from the mean and variance networks of the encoder parameterized by ϕ, and $\mathbf{M}_\theta(\mathbf{Z})$ and $\mathbf{S}_\theta(\mathbf{Z})$ are output from the mean and variance networks of the decoder parameterized by θ.

5.2.2 VAE Training:

Training of the VAE is performed by maximizing the variational lower bound on the likelihood of the training data given as:

$$\mathcal{L}_{ELB}(\theta, \phi; \mathbf{U}^{(i)}) = -KL(q_\phi(\mathbf{Z}|\mathbf{U}^{(i)})||p_\theta(\mathbf{Z})) + E_{q_\phi(\mathbf{Z}|\mathbf{U}^{(i)})}(\log p_\theta(\mathbf{U}^{(i)}|\mathbf{Z})) \tag{5.3}$$

where $p_\theta(\mathbf{Z})$ is an isotropic Gaussian prior. The calculation of the KL divergence and cross entropy loss for the presented sequential architecture is carried out in a manner similar to that described in [58]. The training data is generated by the Aliev-Panfilov (AP) model [2], simulating spatiotemporal TMP sequences originated from different ventricular locations with different tissue properties.

5.3 Transmural EP Imaging

The biophysical relationship between cardiac TMP, $\mathbf{U}$ and body-surface ECG, $\mathbf{Y}$ can be described by a a linear measurement model: $\mathbf{Y} = \mathbf{HU}$, where $\mathbf{H}$ is specific to the heart-torso model of an individual. To estimate $\mathbf{U}$ from $\mathbf{Y}$ is severely ill-posed and requires the regularization from additional knowledge about $\mathbf{U}$.

5.3.1 Probabilistic Modeling of the Inverse Problem:

We formulate the inverse problem in the form of statistical inference. We define the likelihood distribution of $\mathbf{Y}$ given $\mathbf{U}$ by assuming zero-mean measurement errors with variance β^{-1}:

$$p(\mathbf{Y}|\mathbf{U}, \beta) = \prod_k \mathcal{N}(\mathbf{Y}_{:,k}|\mathbf{H}\mathbf{U}_{:,k}, \beta^{-1}\mathbf{I}) \tag{5.4}$$

To incorporate physiological knowledge about $\mathbf{U}$, we model its prior distribution conditioned on $\mathbf{Z}$ using the VAE decoder with trained parameter $\bar{\theta}$:

$$p_{\bar{\theta}}(\mathbf{U}|\mathbf{Z}) = \prod_k \mathcal{N}(\mathbf{U}_{:,k}|\mathbf{M}_{\bar{\theta}}(\mathbf{Z})_{:,k}, diag(\mathbf{S}_{\bar{\theta}}(\mathbf{Z})_{:,k})) \tag{5.5}$$

To further utilize the knowledge about the generative factor $\mathbf{Z}$ learned by the VAE from a large training dataset, we also utilize the VAE-encoded marginal posterior distribution of $\mathbf{Z}$ as its prior distribution in Bayesian inference. In specific, we approximate samples from this marginalized distribution to be Gaussian:

$$p(\mathbf{Z}) = \prod_k \mathcal{N}(\mathbf{Z}_{:,k}|\bar{\boldsymbol{Z}}_{:,k}, diag(\mathbf{C}_{:,k})) \tag{5.6}$$

With this, we complete the statistical formulation of our problem. Our goal is to estimate the joint posterior distributions $p(\mathbf{U}, \mathbf{Z}|\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{U})p(\mathbf{U}|\mathbf{Z})p(\mathbf{Z})$.

5.3.2 Inference:

Due to the presence of a deep neural network, the posterior $p(\mathbf{U}, \mathbf{Z}|\mathbf{Y})$ is analytically intractable. To address this issue, we note that conditioned on $\mathbf{Z}$, the distribution of $\mathbf{U}$ is Gaussian in each column; thus, $p(\mathbf{U}|\mathbf{Y}, \mathbf{Z})$ is analytically available. We leverage this fact and employ a variant of the expectation maximization (EM) algorithm to obtain the maximum a posteriori (MAP) estimate of $\mathbf{Z}$ along with the posterior distribution of $\mathbf{U}$ given the MAP estimate of $\mathbf{Z}$.

E-step: Conditioned on an estimated value of $\mathbf{Z}$ (say $\hat{\mathbf{Z}}$), we calculate posterior of $\mathbf{U}$ as $\hat{p}(\mathbf{U}|\mathbf{Y}, \hat{\mathbf{Z}}) = \prod_k \mathcal{N}(\mathbf{U}_{:,k}|\hat{\boldsymbol{U}}_{:,k}, \hat{\boldsymbol{\Sigma}}_{:,:,k})$, with the covariance and mean of the k^{th} column of $\mathbf{U}$ as:

$$\hat{\boldsymbol{\Sigma}}_{:,:,k} = (\beta \mathbf{H}^T \mathbf{H} + \mathbf{D}_k^{-1})^{-1}, \qquad \hat{\boldsymbol{U}}_{:,k} = \hat{\boldsymbol{\Sigma}}_{:,:,k}(\beta \mathbf{H}^T \mathbf{Y}_{:,k} + \mathbf{D}_k^{-1} \mathbf{m}_k) \tag{5.7}$$

where $\mathbf{D}_k = diag(\mathbf{S}_\theta(\hat{\mathbf{Z}})_{:,k})$, and $\mathbf{m}_k = \mathbf{M}_{\bar{\theta}}(\hat{\mathbf{Z}})_{:,k}$ and $\mathbf{S}_{\bar{\theta}}(\hat{\mathbf{Z}})_{:,k}$ are the k^{th} column output of the VAE decoder network when $\hat{\mathbf{Z}}$ is input to it.

M-step: Given $\hat{p}(\mathbf{U}|\mathbf{Y}, \hat{\mathbf{Z}})$, we update $\mathbf{Z}$ by maximizing $E_{\hat{p}(\mathbf{U}|\mathbf{Y}, \hat{\mathbf{Z}})} \log(p(\mathbf{Y}, \mathbf{U}, \mathbf{Z}))$

$$\mathcal{L} = E_{\prod_k \mathcal{N}(\mathbf{U}_{:,k}|\hat{\boldsymbol{U}}_{:,k}, \hat{\boldsymbol{\Sigma}}_{:,:,k})}[\log(p_{\bar{\theta}}(\mathbf{U}|\mathbf{Z}))] + \log(p(\mathbf{Z})) + constant \tag{5.8}$$

Realizing that a complete optimization of $\mathcal{L}$ with respect to $\mathbf{Z}$ would be expensive, we instead take a few gradient descent steps towards the optimum. The gradient of the second term is analytically available. The gradient of the first term is calculated by backpropagation through the decoder network.

The EM steps iterate until convergence, at which we obtain both the MAP value of $\mathbf{Z}$ and the posterior distribution of $\mathbf{U}$ conditioned on $\mathbf{Z}$ and $\mathbf{Y}$.

5.4 Synthetic Experiments:

Synthetic experiments are carried out on two image-derived human heart-torso models. On each heart, the VAE is trained using around 850 simulated TMP signals consid-

ering approximately 50 different origins of ventricular activation in combination with 17 different tissue property configurations. As an initial study, here we focus on tissue properties representing local regions of myocardial scars with varying sizes and locations.

The presented method incorporating the trained VAE model is then tested on simulated 120-lead ECG data from three different settings, each with 20 experiments. The three settings include 1) presence of myocardial scar not included in training data, 2) origin of ventricular activation different from those used in training, and 3) both myocardial scar and activation origin not seen in training. In all experiments, the performance of the presented method is compared to 0-order Tikhonov regularization with temporal constraint (Greensite method) [44] and conventional EP model constrained inference with fixed parameters [109].

The reconstruction accuracy is measured with three metrics: 1) normalized RMSE given by the ratio of Frobenius norm of the error matrix to that of the truth TMP matrix, 2) Euclidean distance between the reconstructed and true origins of ventricular activation, and 3) Dice coefficient of the reconstructed S_1 and true regions of scar S_2 as $=2|S_1 \cap S_2|/(|S_1|+|S_2|)$. In the two physiologically constrained methods, region of scar is defined based on absence or delay of activation and shortening of action potential duration; in Greensite method, since the reconstructed signal no longer preserves the temporal shape of TMP, the region of scar is defined based on the peak amplitude of the signal.

Computational cost: Training of the VAE takes approximately 40 hours on a 4 GB Nvidia Quadro P1000 GPU. Generation of training data for each heart takes about 7 hours and inference around 30 minutes on Quadcore CPU.

TMP generation: Fig. 5.2 shows examples of local TMP signals generated by the trained VAE decoder against TMP signals simulated by the AP model [2]. Note that, when generating from a isotropic Gaussian (Fig. 5.2 right), noisy rather than meaningful TMP signals may also be generated. In comparison, when sampling from the approximated posterior distribution of $\mathbf{Z}$ as described in equation (5.6), the generated

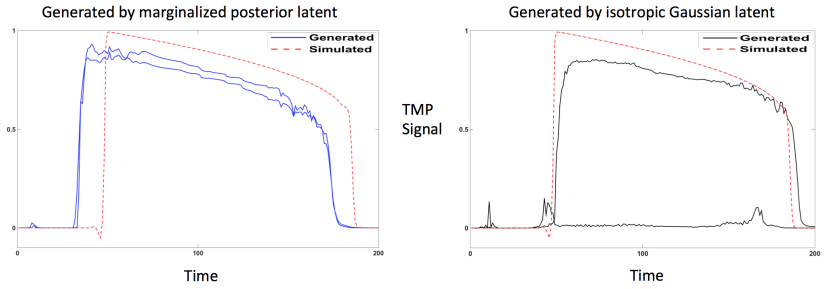
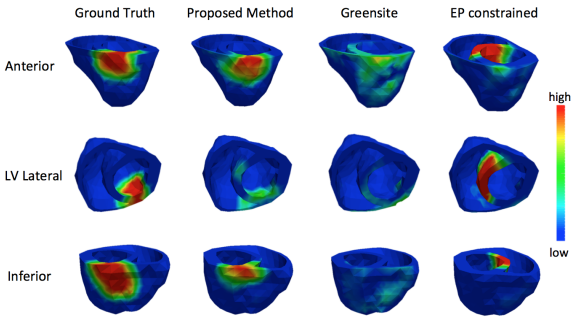


Figure 5.2: Examples of TMP signals generated by samples from two different distributions: Left- marginalized posterior density encoded by the VAE ; Right- isotropic Gaussian.



figures

Figure 5.3: Snapshots of early TMP pattern reconstructed by the three methods in comparison to the ground truth. The origin of activation is noted on the left in each row.

signals closely resemble the simulated TMP signals.

Imaging TMP from various origins: Fig. 5.3 shows a snapshot from the early stage of ventricular activation reconstructed by the three methods in comparison to the ground truth. Since the EP model constrained approach assumes general sinus-rhythm activation, it introduces model error that incorrectly dominates the results. The simple Greensite method, free from erroneous model assumption, actually does a better job in comparison. By adapting model generative factors to the data, the presented method demonstrates a significantly improved ability to reconstruct TMP sequence resulting from unknown origins.

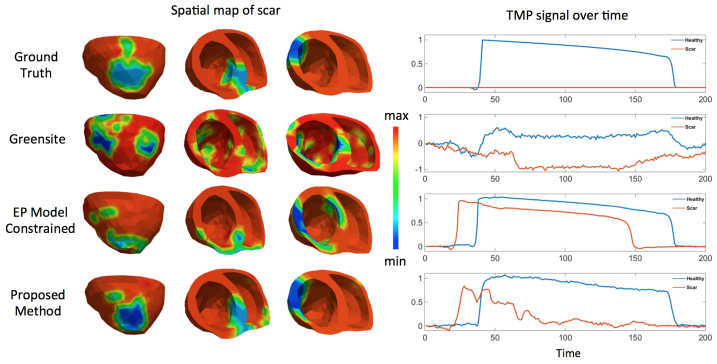


Figure 5.4: Spatial distributions of scar tissues and temporal TMP signals obtained by the three methods in comparison to the ground truth.

Imaging TMP at the presence of myocardial scar: Fig. 6.3 shows the spatial distribution of scar tissue obtained by the three different methods, along with temporal TMP signals reconstructed in healthy and scar regions, in comparison to the ground truth. Without prior physiological knowledge, the Greensite method is not able to preserve the temporal TMP shape, resulting in high RMSE error as shown in Table 1. By thresholding the maximum amplitude of the reconstructed signals, the identified region of scar has high false positives and resembles poorly with the ground truth. The EP model constrained approach does a better job in retaining the temporal TMP shape. However, without prior knowledge about the scar, the model error again affects the accuracy of TMP reconstruction, especially in the early stage of activation when a smaller amount of ECG data is available for correcting the model error. The presented method, in comparison, is able to recognize the presence of scar tissue, adapting the physiological constraint for improved TMP reconstructions and scar identifications.

Summary: Table 1 summarizes the quantitative comparison of the three methods tested in the three settings as described earlier. Although the test cases were not seen by the VAE during training, the proposed method shows a significant improvement in inverse reconstruction (paired t-test, $p<0.001$) when compared with the other two methods in all settings and metrics except with Euclidean distance using Greensite method, where improvement is only marginal. It shows the importance of physiological knowledge and its adaptation to observed data during model-constrained inference.

	Greensite	EP constrained	Proposed Method
Normalized RMSE	1.005 ± 0.006	0.3 ± 0.04	$\mathbf{0.23 \pm 0.05}$
Dice coefficient	0.19 ± 0.04	0.25 ± 0.09	$\mathbf{0.52 \pm 0.2}$

	Greensite	EP constrained	Proposed Method
Normalized RMSE	1.001 ± 0.003	0.28 ± 0.05	$\mathbf{0.11 \pm 0.08}$
Euclidean Distance	18.5 ± 10.96	39.47 ± 6.3	$\mathbf{14.37 \pm 14.0}$

	Greensite	EP constrained	Proposed Method
Normalized RMSE	1.005 ± 0.003	0.39 ± 0.03	$\mathbf{0.29 \pm 0.09}$
Dice coefficient	0.20 ± 0.07	0.21 ± 0.05	$\mathbf{0.48 \pm 0.24}$
Euclidean Distance	18.7 ± 9.3	65.5 ± 11.02	$\mathbf{17.89 \pm 10.6}$

Table 5.1: Quantitative accuracy of the three methods in three settings. Test data is simulated with **1) Top**: scar not in VAE training, **2) Middle**: activation origin not in training, **3) Bottom**: both myocardial scar and activation origin not in training.

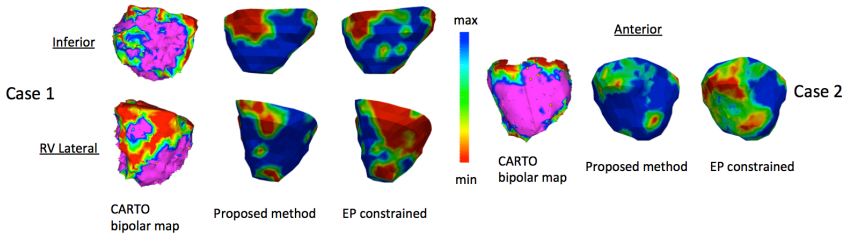


Figure 5.5: Real-data experiments: regions of scar tissues identified by the presented method and conventional EP model constrained method, in comparison to bipolar voltage data (red: scar core; green: scar border; purple: healthy tissue).

5.5 Real data Experiments:

Two case studies are performed on real-data from patients who underwent catheter ablation due to scar-related ventricular arrhythmia. Spatiotemporal TMP is reconstructed from 120-lead ECG data using the presented method and the EP model constrained method. In Fig. 5.5, scar regions (red regions with low voltage) identified

from the reconstructed TMP are compared with scar regions (red regions) in the in-vivo bipolar voltage data. In both cases, while the scar tissue identified by two methods are generally in similar locations, the presented method shows less false positives and higher qualitative consistency with bipolar voltage maps.

5.6 Study of Architecture in Learning Representation

To investigate the effect of different architecture choices in learning representation, we experiment with three architectures: Language, svs and sss architecture. Language architecture is of the same form as used in Language translation [100], and has deterministic latent space. Other two architectures have stochastic latent vector as described before. Fig. 5.6 shows a general architecture for a stochastic model at the bottom half. As shown, both encoder and decoder has two layer LSTMs. Both, svs and sss are in the stochastic setting where there are two networks for mean(M) and variance(S) while the Language architecture does not have variance network. The major difference in three architectures is explained in the top half of the Fig. 5.6. In the language model, the output from last hidden unit of LSTM is directly fed to the decoder and then subsequent predictions are computed recurrently. The svs architecture uses additional fully connected layers to map sequence of latent codes into a vector – hence the name sequence to vector to sequence (svs). In the sss architecture, however, the hidden codes from all units are represented as a matrix latent code from which input TMP signal is reconstructed through a mirrored architecture.

5.6.1 Implementation details

Training and test sets of transmembrane potential (TMP) were generated by using Aliev Panfilov model [2] on a human-torso geometry model. By varying two parameters: origin of excitation and tissue properties representing myocardial scar, we gen-

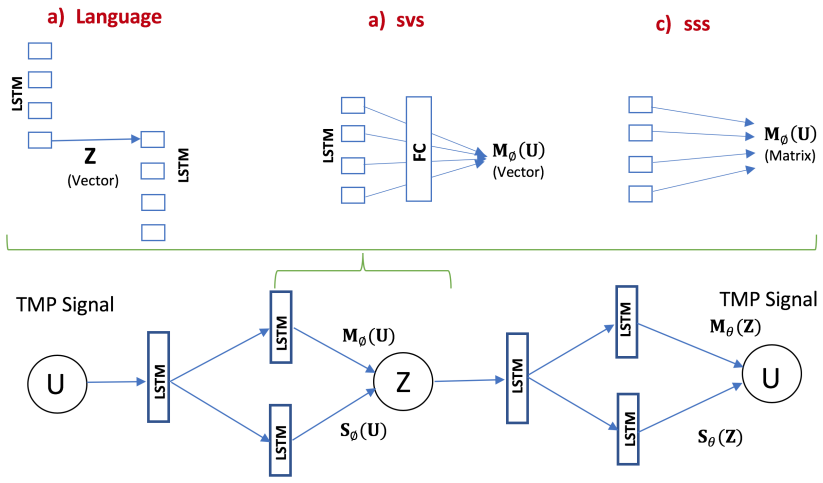


Figure 5.6: Bottom: Common skeleton for three architectures, Top: Three architectures of differing in their ways of converting output from last layers of LSTM to latent representation

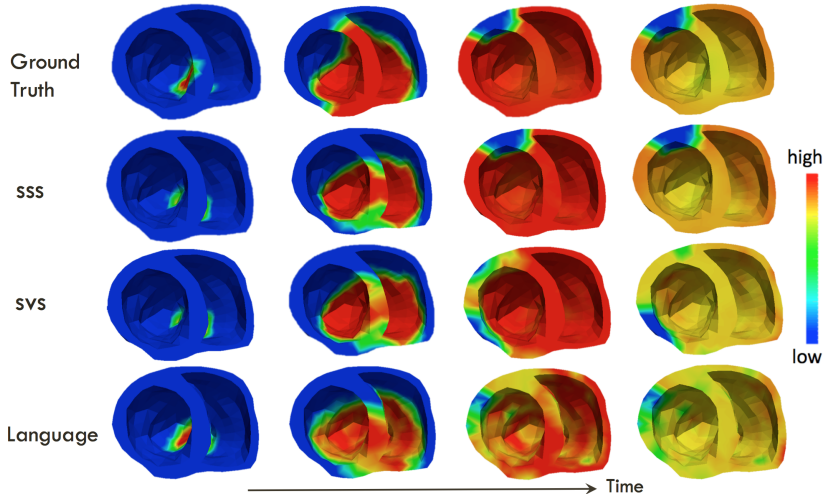


Figure 5.7: Comparison of transmembrane potential propagation

erated about 600 simulation data with the combination of 17 different tissue property configurations and 35 different origins of excitation. To test generalization ability, test data were selected with different origin of excitation than those used in training.

We used ReLU activation function in both encoder and decoder, ADAM optimizer and a flat learning rate of 10^{-3} in all three architectures.

Figure 5.8: Comparison of reconstruction using three architectures

5.6.2 Comparison and Discussion

We compare three architectures in their ability to generalize in new test cases. We measure the reconstruction accuracy with four metrics: 1) mean square error (MSE) of TMP, 2) correlation of TMP, 3) dice coefficient of the scar region, 4) correlation of activation time. Fig. 5.7 compares TMP propagation reconstructed by using three architectures with the ground truth. The Language model matches the ground truth better than other architectures at the beginning of the propagation sequence. However, later on, other two methods are qualitatively better. The scar region, however, seems to be better identified by sss architecture compared to other two.

The graphs on Fig. 5.8 shows average of 20 tests, each performed by randomly drawing 200 samples from the test set. It is interesting that the Language model performs quite

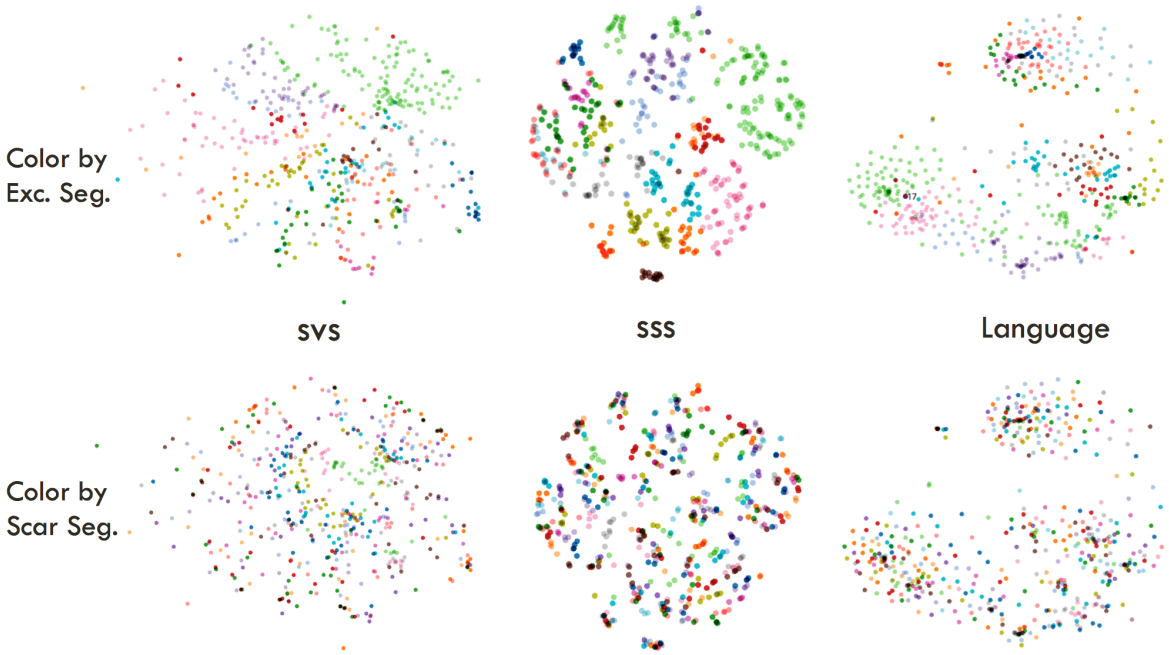


Figure 5.9: Visualization of point cloud in the latent space corresponding training and test data

good when measured with mean square error and correlation of TMP. But, when we measure dice coefficient of scar and correlation of activation time derived from the reconstructed TMP, the Language architecture performs the worst. It suggests that Language model might be good at preserving temporal consistency but not so much at learning underlying factors. On the other hand, svs and sss architectures seem to be better learning underlying factors, which might be because of the stochastic latent space in these two architectures.

We also visualized the latent representation of the whole dataset- training and test set- of three different architectures. In Fig. 5.9, top row shows latent point cloud colored according to the segment where origin of excitation lies. Similarly, bottom row shows latent point cloud colored according to the segment where scar lies. The heart is divided into 17 segments according to American heart association (AHA) standard and each color denotes one segment of the heart where scar/origin lies. We observe that the latent representation is clustered by the location of origin of excitation in all

three architectures, but not by the location of scar region. It might be because there were not many examples of scar regions from the same segment for the network to generalize. We need further analysis.

The results are thought-provoking. However, we caution that the work is preliminary in that we tried it on a single geometry and relatively small dataset for a deep network. We leave some open questions triggered by these observations: Why does a method performs better with respect to RMSE error but not so well with respect to error of origin of excitation? Does stochasticity play a role in better representation? Why did the neural network better represented the origin of excitation than the scar region?

5.7 Discussion and Conclusions:

To our knowledge, this is the first work that integrates a generative network learned from numerous examples into a statistical inference framework to allow the adaptation of prior physiological knowledge via a small number of generative factors. The results show the ability of this concept to improve model-constrained inference.

One interesting future direction of research is proper modeling of the prior distribution of latent space after unsupervised training. At the moment, we use a naive approach to estimate the prior distribution as Gaussian by sampling from the marginal distribution of the decoder. The marginal posterior distribution is a rich distribution. Therefore, we must be creative in better representing it. This would be a good future direction of research. Second, since the present formulation is in a personalized setting, we intend to extend this architecture to learn a geometry-invariant generative model that can be trained on multiple heart models and applied on a new subject.

5.8 Summary and Answers to Research Questions

In this chapter, we looked for a better way to represent patterns of TMP signals, x in the prior distribution $p_{\mathcal{X}}$. We started with the following question:

Q. 2. How can we improve generalization with an alternative representation of prior knowledge such that it allows efficient inference?

We saw that a deep generative model based on variational autoencoder trained on simulated samples of transmembrane potential (TMP) can be used as a custom generative model. Using this generative model, we use a custom hierarchical prior for the transmembrane potential. The prior automatically learns the relationship between latent factors and the generated TMP signals. We also showed that using this prior helps both the learning useful pattern and adapting generative model based on the ECG signal via efficient inference. During experiments, we showed that this model indeed helps to generalize well in the test distribution, even if it is selected to be outside of the training samples used in learning the prior model.

PART II

LEARNING FROM SAMPLES

Prologue to Part II

We now change gear and take a purely data-driven approach. Here, we are interested in directly computing conditional distribution of the TMP given ECG data, like the posterior distribution of TMP given ECG in a Bayesian framework. Here, however, we do not have the structure of probabilistic graphical model, nor do we have tools from Bayesian inference. This approach is, therefore, more direct and more flexible. Deep learning is a quintessential example of this approach. One fundamental question that arises when dealing with any learning algorithm is its ability to generalize outside the examples used for learning the conditional distribution. And there are theories that deal with generalization while learning from samples. However, in case of deep learning, traditional theories like statistical learning theories seem inadequate in analysis and design of better learning algorithm and architectures [116]. Although huge effort is devoted to understand and improve generalization of deep network, we have not yet reached a consensus or a well understood/agreed theory. On the other hand, although the field of medical imaging and computational physiology has seen a flood of work employing deep learning, the amount of work in improving generalization of deep networks remains extremely small.

Since the main theme of this dissertation is to improve upon the generalization of learning algorithm for the inverse EP imaging, we focus in this second part in better understanding generalization and improving it. To improve generalization, we approach from two general directions:
1. Address distribution mismatch: If there is discrepancy between training and test set, the generalization might be poor if we do not address it. To address it, we propose the idea of invariance: to learn a common representation space where the discrepancy of the projection of training and test set is minimum. We enforce invariance via two ideas: a) Information Bottleneck, b) Adversarial training.

2. Learning simple functions: Another direction that is complementary to distribution mismatch is that of learning a simple function (classifier or regressor). This notion has its root in statistical learning theory and in Occam's Razor. The ideas is that simpler functions generalize better. However, what is a simple function? In this direction, we invoke two ideas a) By using analytical learning theory, we propose that a smooth

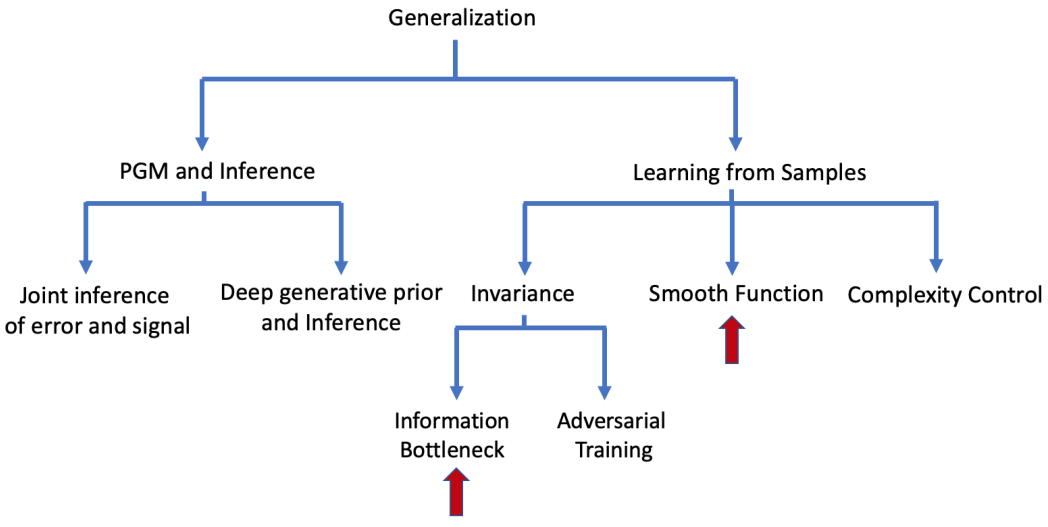
function is a simple function and can help in generalization, b) We propose an entirely different network architecture merging the ideas from deep learning and kernel methods. Now, using this new architecture, we ensure that the function lies in Reproducing Kernel Hilbert Space and the we explicitly penalize the complexity of the function.

The organization of the second part is as follows. In chapter 6, we try to improve generalization by information bottleneck and analytical learning theory, in chapter 7, we improve generalization by improving invariance via adversarial training and in chapter 8, we construct function in RKHS and control the complexity. Even though we were primarily focused in solving the inverse problem of electrophysiological imaging, in the second part, we took on a general problem of improving generalization. Therefore, we also test the idea of invariance to improve generalization in X-ray image classification (Ch. 7). Similarly, in our last chapter we tackle the problem of stability in adversarial training, which is a precursor towards our bigger goal of quantifying the role of complexity to improve generalization, again an objective broader than solving inverse problems.

Chapter 6

Learning with Generalization in Deep Networks

Nothing is more practical than a good theory.
- Vladimir N. Vapnik



6.1 Introduction

There has been an upsurge of deep learning approaches for traditional image reconstruction problems in computer vision and medical imaging [67]. Examples include image denoising [71], inpainting [83], and medical image reconstructions across a variety of modalities such as magnetic resonance imaging [117] and computed tomography [47]. Despite state-of-the-art performances brought by these deep neural networks, their ability to reconstruct from data not seen in the training distribution is not well understood. To date, very limited work has investigated the generalization ability of these image reconstruction networks from a theoretical perspective, or provided insights into what aspects of representation and learning may improve the ability of these networks to generalize outside the training data.

In this paper, we take an information theoretic perspective – along with analytical learning theory – to investigate and improve the generalization ability of deep image reconstruction networks. Let $\boldsymbol{x}$ be the original image and $\boldsymbol{y}$ be the measurement obtained from $\boldsymbol{x}$ by some transformation process. To reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}$, we adopt a common deep encoder-decoder architecture [83, 117] where a latent representation $\boldsymbol{w}$ is first inferred from $\boldsymbol{y}$ before being used for the reconstruction of $\boldsymbol{x}$. Our objective is to learn transformations that are general, possibly learning the underlying generative process rather than focusing on every detail in training examples. To this end, we propose that the generalization ability of a deep reconstruction network can be improved from two means: 1) the ability to generalize to data $\boldsymbol{y}$ that are generated from $\boldsymbol{x}$ (and thereby $\boldsymbol{w}$) outside the training distribution; and 2) the ability to generalize to unseen variations in data $\boldsymbol{y}$ that are introduced during the measurement process but irrelevant to $\boldsymbol{x}$.

For the first type of generalization ability, we hypothesize that it can be improved by using stochastic instead of deterministic latent representations. We support this hypothesis by the analytical learning theory [56], showing that stochastic latent space helps to learn a decoder that is less sensitive to perturbations in the latent space and thereby leads to better generalization. For the second type of generalization ability, we

hypothesize that it can be improved if the encoder compresses the input measurement into a minimal latent representation (*codes* in information theory), containing only the necessary information for $\boldsymbol{x}$ to be reconstructed. To obtain a minimal representation from $\boldsymbol{y}$ that is maximally informative of $\boldsymbol{x}$, we adopt the information bottleneck theory formulated in [103] to maximize the mutual information between the latent code $\boldsymbol{w}$ and $\boldsymbol{x}$, $I(\boldsymbol{x}, \boldsymbol{w})$, while putting a constraint on the mutual information between $\boldsymbol{y}$ and $\boldsymbol{w}$ such that $I(\boldsymbol{w}, \boldsymbol{y}) < I_0$. This can be achieved by minimizing the following objective:

$$loss_{IB} = -I(\boldsymbol{x}; \boldsymbol{w}) + \beta I(\boldsymbol{w}; \boldsymbol{y}) \tag{6.1}$$

where β is the Lagrange multiplier. Based on these two primary hypotheses, we present a deep image reconstruction network optimized by a variational approximation of the information bottleneck principle with stochastic latent space.

While the presented network applies for general reconstruction problems, we test it on the sequence reconstruction of cardiac transmembrane potential (TMP) from high-density body-surface electrocardiograms (ECGs) [109]. Given the sequential nature of the problem, we use long short-term memory (LSTM) networks in both the encoder and decoder, with two alternative architectures to compress the temporal information into vector latent space. We tackle two specific challenges regarding the generalization of the reconstruction. First, because the problem is ill-posed, it has been important to constrain the reconstruction with prior physiological knowledge of TMP dynamics [33, 44, 109]. This however made it difficult to generalize to physiological conditions outside those specified by the prior knowledge. By using the stochastic latent space, we demonstrate the ability of the presented method to generalize outside the physiological knowledge provided in the training data. Second, because the generation of ECGs depends on heart-torso geometry, it has been difficult for existing methods to generalize beyond a patient-specific setting. By the use of the information bottleneck principle, we demonstrate the robustness of the presented network to geometrical variations in ECG data and therefore a unique ability to generalize to unseen subjects. These generalization abilities are tested in two controlled synthetic datasets as well as a real-data feasibility study.

This chapter includes parts from author's conference publication [35].

6.2 Related Work

Deep neural networks have become popular in medical image reconstructions across different modalities such as computed tomography [47], magnetic resonance imaging [117], and ultrasound [68]. Some of these inverse reconstruction networks are based on an encoder-decoder structure [47, 117], similar to that investigated in this paper. Among these, the presented work is the closest to Automap [117] in that the output image is reconstructed directly from the input measurements without any intermediate domain-specific transformations. However, these existing works have not investigated either the use of stochastic architectures or the information bottleneck principle to improve the ability of the network to generalize outside the training distributions.

The presented variational formulation of the information bottleneck principle is closely related to that presented in [1]. However, our work differs in three primary aspects. First, we investigate image reconstruction tasks in which the role of information bottleneck has not been clearly understood. Second, we define generalization ability in two different categories, and provide theoretical as well as empirical evidence on how stochastic latent space can improve the network's generalization ability in a way different from the information bottleneck. Finally, we extend the setting of static image classification to image sequences, in which the latent representation needs to be compressed from temporal information within the whole sequence.

To learn temporal relationship in ECG/TMP sequences, we consider two sequence encoder-decoder architectures. One is commonly used in language translation [100], where the code from the last unit of the last LSTM encoder layer is used as the latent vector representation to reconstruct $\boldsymbol{x}$. We also present a second architecture where fully connected layers are used to compress all the hidden codes of the last LSTM layer into a latent vector representation. This is in concept similar to the attention mechanism [7] to selectively use information from all the hidden LSTM codes for decoding. We experimentally compare the generalization ability of using stochastic versus deterministic latent vectors in both architectures, which has not been studied before.

In the application area of cardiac TMP reconstruction, most related to this paper are works constraining the reconstruction with prior temporal knowledge in the form of physics-based simulation models of TMP [109] and, more recently, generative models learned from physics-based TMP simulation [33]. This however to our knowledge is the first work that investigated the use of deep learning for the direct inference of TMP from ECG. This method will also have the unique potential to generalize outside the patient-specific settings and outside pathological conditions included in the prior knowledge.

6.3 Methodology

Body-surface electrical potential is produced by TMP in the heart. Their mathematical relation is defined by the quasi-static approximation of electromagnetic theory [85] and, when solved on patient-specific heart-torso geometry, can be derived as: $\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t)$, where $\boldsymbol{y}(t)$ denotes the time-varying body-surface potential map, $\boldsymbol{x}(t)$ the time-varying TMP map over the 3D heart muscle, and $\boldsymbol{H}$ the measurement matrix specific to the heart-torso geometry of a subject [109]. The inverse reconstruction of $\boldsymbol{x}$ from $\boldsymbol{y}$ at each time instant is ill-posed, and a popular approach is to reconstruct TMP time sequence constrained by prior physiological knowledge of its dynamics [33, 44, 109]. This is the setting considered in this study, in which the deep network learns to reconstruct with prior knowledge from pairs of $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ generated by physics-based simulation. Note that it is not possible to obtain real TMP data for training, which further highlights the importance of the network to generalize. In what follows, we use $\boldsymbol{x}$ and $\boldsymbol{y}$ to represent sequence matrices with each column denoting the potential map at one time instant.

Given the joint distribution of TMP and ECG given by $p(\boldsymbol{x}, \boldsymbol{y})$, the encoder gives us a conditional distribution $p(\boldsymbol{w}|\boldsymbol{y})$. These together defines a joint distribution of ($\boldsymbol{x}, \boldsymbol{y}$,$\boldsymbol{w}$):

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) = p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x}, \boldsymbol{y})p(\boldsymbol{w}|\boldsymbol{y}) \tag{6.2}$$

The first term in $loss_{IB}$ in eq.(6.1) is given by

$$I(\boldsymbol{x};\boldsymbol{w}) = \int p(\boldsymbol{x},\boldsymbol{w}) \log(\frac{p(\boldsymbol{x}|\boldsymbol{w})}{p(\boldsymbol{x})}) d\boldsymbol{x} d\boldsymbol{w} = H(\boldsymbol{x}) + \int p(\boldsymbol{x},\boldsymbol{w}) \log(p(\boldsymbol{x}|\boldsymbol{w})) d\boldsymbol{x} d\boldsymbol{w}$$

where $p(\boldsymbol{x}|\boldsymbol{w}) = \int \frac{p(\boldsymbol{x},\boldsymbol{w},\boldsymbol{y})}{p(\boldsymbol{w})} d\boldsymbol{y} = \int \frac{p(\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{w}|\boldsymbol{y})}{p(\boldsymbol{w})} d\boldsymbol{y}$ is intractable. Letting $q(\boldsymbol{x}|\boldsymbol{w})$ to be the variational approximation of $p(\boldsymbol{x}|\boldsymbol{w})$, we have:

$$\begin{aligned}\int p(\boldsymbol{x},\boldsymbol{w}) \log(p(\boldsymbol{x}|\boldsymbol{w})) d\boldsymbol{x} d\boldsymbol{w} &= \int p(\boldsymbol{w})[p(\boldsymbol{x}|\boldsymbol{w}) \log \frac{p(\boldsymbol{x}|\boldsymbol{w})}{q(\boldsymbol{x}|\boldsymbol{w})} + p(\boldsymbol{x}|\boldsymbol{w}) \log q(\boldsymbol{x}|\boldsymbol{w})] d\boldsymbol{x} d\boldsymbol{w} \\ &= \int p(\boldsymbol{w}) D_{KL}(p(\boldsymbol{x}|\boldsymbol{w})||q(\boldsymbol{x}|\boldsymbol{w})) d\boldsymbol{w} + \int p(\boldsymbol{x},\boldsymbol{w}) \log q(\boldsymbol{x}|\boldsymbol{w}) d\boldsymbol{x} d\boldsymbol{w} \qquad (6.3)\end{aligned}$$

where the KL divergence in the first term is non-negative. This gives us:

$$I(\boldsymbol{x};\boldsymbol{w}) \geq \int p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \log q(\boldsymbol{x}|\boldsymbol{w})] d\boldsymbol{x} d\boldsymbol{y} d\boldsymbol{w} = E_{p(\boldsymbol{x},\boldsymbol{y})}[E_{p(\boldsymbol{w}|\boldsymbol{y})}[\log q(\boldsymbol{x}|\boldsymbol{w})]] \qquad (6.4)$$

The second term in $loss_{IB}$ in eq.(6.1) is given by

$$\begin{aligned}I(\boldsymbol{y};\boldsymbol{w}) &= \int p(\boldsymbol{y},\boldsymbol{w}) \log(\frac{p(\boldsymbol{w}|\boldsymbol{y})}{p(\boldsymbol{w})}) d\boldsymbol{y} d\boldsymbol{w} = \int p(\boldsymbol{y},\boldsymbol{w}) \log[\frac{p(\boldsymbol{w}|\boldsymbol{y})r(\boldsymbol{w})}{r(\boldsymbol{w})p(\boldsymbol{w})}] d\boldsymbol{y} d\boldsymbol{w} \\ &= \int p(\boldsymbol{y})p(\boldsymbol{w}|\boldsymbol{y}) \log(\frac{p(\boldsymbol{w}|\boldsymbol{y})}{r(\boldsymbol{w})}) d\boldsymbol{y} d\boldsymbol{w} - D_{KL}(p(\boldsymbol{w})||r(\boldsymbol{w})) \qquad (6.5) \\ &\leq \int p(\boldsymbol{y})p(\boldsymbol{w}|\boldsymbol{y}) \log(\frac{p(\boldsymbol{w}|\boldsymbol{y})}{r(\boldsymbol{w})}) d\boldsymbol{y} d\boldsymbol{w} = E_{p(\boldsymbol{y})}[D_{KL}(p(\boldsymbol{w}|\boldsymbol{y})||r(\boldsymbol{w}))] \qquad (6.6)\end{aligned}$$

Combining eq.(6.4) and eq.(6.6), we have

$$loss_{IB} \leq E_{p(\boldsymbol{x},\boldsymbol{y})}[-E_{p(\boldsymbol{w}|\boldsymbol{y})}\big[\log q(\boldsymbol{x}|\boldsymbol{w})] + \beta D_{KL}(p(\boldsymbol{w}|\boldsymbol{y})||r(\boldsymbol{w}))\big] = \mathcal{L}_{IB} \qquad (6.7)$$

which gives us $\mathcal{L}_{IB}$ to be minimized as an upper bound of the information bottleneck objective $loss_{IB}$ formulated in eq.(6.1).

Parameterization with neural network:

We model both $p(\boldsymbol{w}|\boldsymbol{y})$ and $q(\boldsymbol{x}|\boldsymbol{w})$ as Gaussian distributions, with mean and variance parameterized by neural networks:

$$p_{\boldsymbol{\theta}_1}(\boldsymbol{w}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{t}_{\theta_1}(y), \boldsymbol{\sigma_t}^2(\boldsymbol{y})) \qquad q_{\boldsymbol{\theta}_2}(\boldsymbol{x}|\boldsymbol{w}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{g}_{\theta_2}(\boldsymbol{w}), \boldsymbol{\sigma_x}^2(\boldsymbol{w})) \qquad (6.8)$$

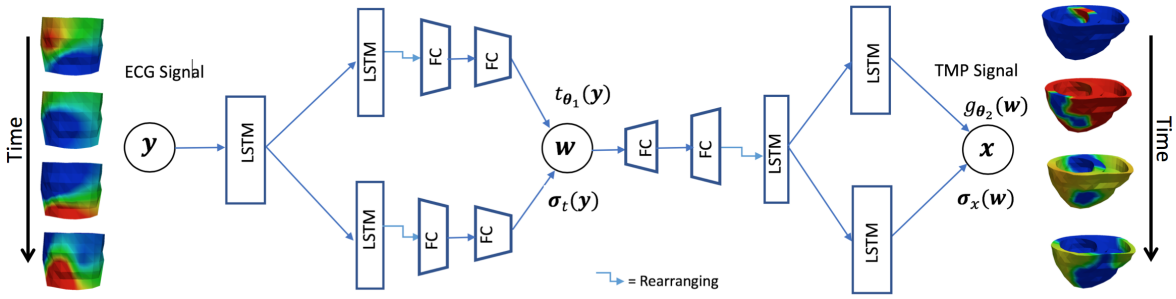


Figure 6.1: Illustration of the presented *svs stochastic* architecture, where both the encoder and the decoder consists of mean and variance networks.

where $\boldsymbol{\sigma_x}^2$ denotes a matrix that consists of the variance of each corresponding element in matrix $\boldsymbol{x}$. This is based on the implicit assumption that each elements in $\boldsymbol{x}$ is independent and Gaussian, and similarly for $\boldsymbol{w}$. This gives us:

$$\mathcal{L}_{IB}(\boldsymbol{\theta}) = E_{p(\boldsymbol{x},\boldsymbol{y})}[-E_{p_{\boldsymbol{\theta_1}}(\boldsymbol{w}|\boldsymbol{y})}[\log q_{\boldsymbol{\theta}_2}(\boldsymbol{x}|\boldsymbol{w})] + \beta.D_{KL}(p_{\boldsymbol{\theta}_1}(\boldsymbol{w}|\boldsymbol{y})||r(\boldsymbol{w}))]$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$. We use reparameterization $\boldsymbol{w} = \boldsymbol{t} + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}$ as described in [58] to compute the inner expectation in the first term. The KL divergence in the second term is analytically available for two Gaussian distributions. We obtain:

$$\begin{aligned}\mathcal{L}_{IB}(\boldsymbol{\theta}) = E_{p(\boldsymbol{x},\boldsymbol{y})}\Big[E_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\Big(\sum_i \frac{1}{\boldsymbol{\sigma_x}_i^2}(x_i - g_i(\boldsymbol{t} + \boldsymbol{\sigma_t} \odot \boldsymbol{\epsilon}))^2 + \log \boldsymbol{\sigma_x}_i^2\Big) \\ + \beta.D_{KL}(p_{\boldsymbol{\theta}_1}(\boldsymbol{w}|\boldsymbol{y})||\mathcal{N}(\boldsymbol{w}|\mathbf{0},\boldsymbol{I}))\Big] \end{aligned} \tag{6.9}$$

where g_i is the i^{th} function mapping latent variable to the i^{th} element of mean of $\boldsymbol{x}$, such that $\boldsymbol{g}_{\boldsymbol{\theta}_2} = [g_1, g_2 ... g_U]$. The deep network is trained to minimize $\mathcal{L}_{IB}(\boldsymbol{\theta})$ in eq.(6.9) with respect to network parameters $\boldsymbol{\theta}$.

Network architectures:

The sequence reconstruction network is realized using long short-term memory (LSTM) neural networks in both the encoder and decoder. To compress the time sequence into a latent vector representation, we experiment with two alternative architectures.

First, based on the commonly-used sequence-to-sequence language translation model [100],we consider a *svs-L* architecture that employs the hidden code of the last unit in the last encoding LSTM layer as the latent vector representation for reconstructing TMP sequences. Second, we propose a *svs* architecture where two fully connected layers are used to compress all the hidden codes of the last LSTM layer into a vector representation. In the decoder, this latent representation is expanded by two fully-connected layers before being fed into LSTM layers as shown in Fig. 6.1.

6.4 Statistical versus Analytical Learning Theory

Learning theory deals with the analysis of a theory of learning. Statistical learning theory [14,105] is the foundational learning theory from the beginning of machine learning championed by Vapnik. Statistical learning theory assumes that in any learning process a training and test set is an instant of many possible draw from the distribution of data and provides a measure of upper bound of how bad things can go in expectation and probability. Two remarks are in order. First, since the argument is in probability, if we talk about the specific learning problem instance, we may not be able to say anything about the problem instance. Second, since it gives us upper bound of how bad things can go, the statistical theory is the worst case analysis of the whole class of problem. Statistical learning theory provides generalization bound based on the complexity (like a measure of size) of the class of functions like VC dimension, Rademacher averages, etc. This comes as a drawback in analyzing neural networks because they have high complexity; therefore, the bounds are large and are not very useful. Recently, Zhang et. al. [116] empirically showed that any arguments in terms of sample complexity of the function does not take us too far in case of neural networks because they have enough capacity to memorize the whole random dataset, and yet generalizes well when trained on data with pattern. We cannot apply statistical learning theory to explain good generalization behavior of neural networks, let alone talk about how to improve them.

Therefore, we take a very recently proposed framework of analytical learning the-

ory [56]. It is fundamentally different from classic statistical learning theory in that it is strongly instance-dependent. While statistical learning theory deals with data-independent generalization bounds or data-dependent bounds for certain hypothesis space, analytical learning theory provides the bound on how well a model learned from a dataset should perform on true (unknown) measures of variable of interest. This makes it aptly suitable for measuring the generalization ability of stochastic latent space for the given problem and data, like ours.

6.5 Encoder-Decoder Learning from the Perspective of Analytical Learning Theory

In this section we look at the encoder-decoder inverse reconstructions using analytical learning theory [56]. We start with a general framework and then show that having a stochastic latent space with regularization helps in generalization.

Let $\boldsymbol{z} = (\boldsymbol{y}, \boldsymbol{x})$ be an input-output pair, and let $D_n = \{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, ..., \boldsymbol{z}^{(n)}\}$ denote the total set of training and validation data where $Z_m \subset D_n$ be the validation set. During training, a neural network learns the parameter $\boldsymbol{\theta}$ by using an algorithm $\mathcal{A}$ and dataset D_n, at the end of which we have a mapping $h_{\mathcal{A}(D_n)}(.)$ from $\boldsymbol{y}$ to $\boldsymbol{x}$. Typically, we stop training when the model performs well in the validation set. To evaluate this performance, we define a prediction error function, $\ell(\boldsymbol{x}, h_{\mathcal{A}(D_n)}(\boldsymbol{y}))$ based on our notion of the goodness of prediction. The average validation error is given by $E_{Z_m}\ell(\boldsymbol{x}, h_{\mathcal{A}(D_n)}(\boldsymbol{y}))$. However, there exists a so-called generalization gap between how well the model performs in the validation set versus in the true distribution of the input-output pair. To be precise, let $(\mathcal{Z}, \mathcal{S}, \mu)$ be a measure space with μ being a measure on $(\mathcal{Z}, \mathcal{S})$. Here, $\mathcal{Z} = \mathcal{Y} \times \mathcal{X}$ denotes the input-output space of all the observations and inverse solutions. The generalization gap is given by $\Delta_g = E_\mu \ell(\boldsymbol{x}, h_{\mathcal{A}(D_n)}(\boldsymbol{y})) - E_{Z_m}\ell(\boldsymbol{x}, h_{\mathcal{A}(D_n)}(\boldsymbol{y}))$. Theorem 1 in [56] provides an upper bound on the generalization gap Δ_g in terms of data distribution in the latent space and properties of the decoder.

Theorem 2 ([56]). *For any ℓ, let $(\mathcal{T}, f)$be a pair such that $\mathcal{T} : (\mathcal{Z}, \mathcal{S}) \to ([0,1]^d$,*

$\mathcal{B}([0,1]^d))$ is a measurable function, $f : ([0,1]^d, \mathcal{B}([0,1]^d)) \to (\mathbb{R}, \mathcal{B}(\mathbb{R}))$ is of bounded variation as $V[f] < \infty$, and $\ell(\boldsymbol{x}, h(\boldsymbol{y})) = (f \circ \mathcal{T})(\boldsymbol{z}) \forall \boldsymbol{z} \in \mathcal{Z}$, where $\mathcal{B}(A)$ indicates the Borel σ- algebra on A. Then for any dataset pair (D_n, Z_m) and any $\ell(\boldsymbol{x}, h_{\mathcal{A}(D_n)}(\boldsymbol{y}))$,

$$\Delta_g = E_\mu \ell(\boldsymbol{x}, h_{\mathcal{A}(D_n)}(\boldsymbol{y})) - E_{Z_m} \ell(\boldsymbol{x}, h_{\mathcal{A}(D_n)}(\boldsymbol{y})) \leq V[f] \mathcal{D}^*[\mathcal{T}_* \mu, \mathcal{T}(Z_m)]$$

where $\mathcal{T}_* \mu$ is pushforward measure of μ under the map $\mathcal{T}$.

For an encoder-decoder setup, $\mathcal{T}$ is the encoder that maps the observation to the latent space and f becomes the composition of loss function and decoder that maps the latent representation to the reconstruction loss. Theorem 1 provides two ways to decrease the generalization gap in our problem: by decreasing the variation $V[f]$ or the discrepancy $\mathcal{D}^*[\mathcal{T}_* \mu, \mathcal{T}(Z_m)]$. Here, we show that stochasticity of the latent space helps decrease the variation $V[f]$. The variation of f on $[0,1]^d$ in the sense of Hardy and Krause [48] is defined as: $V[f] = \sum_{k=1}^{d} \sum_{1 \leq j_1 < \ldots < j_k \leq d} V^k[f_{j_1 \ldots j_k}]$ where $V^k[f_{j_1 \ldots j_k}]$ is defined with following proposition.

Proposition 1 ([56])**.** *Suppose that $f_{j_1, .. j_k}$ is a function for which $\partial_{1,\ldots k} f_{j_1, .. j_k}$ exists on $[0,1]^k$. Then,*

$$V^k[f_{j_1 \ldots j_k}] \leq \sup_{\boldsymbol{t}_{j_1}, .., \boldsymbol{t}_{j_k} \in [0,1]^k} |\partial_{1,\ldots k} f_{j_1, .. j_k}(\boldsymbol{t}_{j_1}, .., \boldsymbol{t}_{j_k})|$$

If $\partial_{1,\ldots k} f_{j_1, .. j_k}$ is also continuous on $[0,1]^k$, then,

$$V^k[f_{j_1 \ldots j_k}] = \int_{[0,1]^k} |\partial_{1,\ldots k} f_{j_1, .. j_k}(\boldsymbol{t}_{j_1}, .., \boldsymbol{t}_{j_k})| dt_{j_1} .. dt_{j_k}$$

In our case, f is the prediction error ℓ as a function of latent representations $\boldsymbol{t}$:

$$\ell(\boldsymbol{x}, h(\boldsymbol{y})) = ||\boldsymbol{x} - \boldsymbol{g}_{\boldsymbol{\theta}_2}(\boldsymbol{t})||_F^2 = \sum_i (\boldsymbol{x}_i - g_i(\boldsymbol{t}))^2 = \sum_i \ell_i \tag{6.10}$$

where $||\boldsymbol{a}||_F$ denotes the Frobenius norm of matrix $\boldsymbol{a}$, and $\boldsymbol{g}_{\boldsymbol{\theta}_2}$ maps the latent space to the estimated $\bar{\boldsymbol{x}}$. Theorem 1 and Proposition 1 implies that if the cross partial derivative of the loss with respect to the latent vector at all order is low in all directions throughout the latent space, then the approximated validation loss would be closer to the actual loss over the true unknown distribution of the dataset. Intuitively, we want the loss curve as a function of latent representation to be flat if we want a good generalization.

Using stochastic latent space:

In our formulation, the latent vector is stochastic with the cost function given by eq.(6.9). Using reparameterization $\boldsymbol{\eta} = \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}$, the inner expectation of the first term in the loss function $\mathcal{L}_{IB}$ is given by

$$\begin{aligned} T_1 &= E_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\sum_i \frac{1}{\boldsymbol{\sigma}_{xi}^2}(\boldsymbol{x}_i - g_i(\boldsymbol{t} + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}))^2] \\ &= \sum_i \frac{1}{\boldsymbol{\sigma}_{xi}^2}(\boldsymbol{x}_i - g_i(\boldsymbol{t} + \boldsymbol{\eta}))^2 = \sum_i \frac{1}{\boldsymbol{\sigma}_{xi}^2} E_{\boldsymbol{\epsilon}}[\ell_i(\boldsymbol{x}_i, \boldsymbol{t} + \boldsymbol{\eta})] \end{aligned}$$

Result 2.

$$\begin{aligned} T_1 = \sum_i \frac{1}{\boldsymbol{\sigma}_{xi}^2} \Big[& \ell_i(\boldsymbol{x}_i, \boldsymbol{t}) + \langle \boldsymbol{\sigma}_t \odot E_\epsilon[\boldsymbol{\epsilon}], \frac{\partial}{\partial t}\ell_i(\boldsymbol{x}_i, \boldsymbol{t})\rangle \\ & + \frac{1}{2}\langle [\boldsymbol{\sigma}_t \otimes \boldsymbol{\sigma}_t] \odot E_\epsilon[\boldsymbol{\epsilon} \otimes \boldsymbol{\epsilon}], \Big[\frac{\partial^2}{\partial \boldsymbol{t}_{j_1}, \partial \boldsymbol{t}_{j_2}}\ell_i(\boldsymbol{x}_i, \boldsymbol{t})\Big]\rangle \\ & + .. + \frac{1}{k!}\langle [\boldsymbol{\sigma}_t \otimes^k \boldsymbol{\sigma}_t] \odot E_\epsilon[\boldsymbol{\epsilon} \otimes^k \boldsymbol{\epsilon}], \Big[\frac{\partial^k}{\partial \boldsymbol{t}_{j_1}, .., \partial \boldsymbol{t}_{j_k}}\ell_i(\boldsymbol{x}_i, \boldsymbol{t})\Big]\rangle + ..\Big] \end{aligned}$$

where $[\boldsymbol{\sigma}_t \otimes^k \boldsymbol{\sigma}_t]$ *denotes* k *order tensor product of a vector* $\boldsymbol{\sigma}_t$ *by itself.*

Proof. Using Taylor series expansion for $\ell_i(\boldsymbol{x}_i, \boldsymbol{t} + \boldsymbol{\eta})$,

$$\begin{aligned} E_\epsilon[\ell_i(\boldsymbol{x}_i, \boldsymbol{t} + \boldsymbol{\eta})] = \ & \mathrm{E}_\epsilon\Big[\ell_i(\boldsymbol{x}_i, \boldsymbol{t}) + \langle \boldsymbol{\eta}, \tfrac{\partial}{\partial \boldsymbol{t}}\ell_i(\boldsymbol{x}_i, \boldsymbol{t})\rangle + \tfrac{1}{2}\langle [\boldsymbol{\eta} \otimes \boldsymbol{\eta}], \Big[\tfrac{\partial^2}{\partial \boldsymbol{t}_{j_1}, \partial \boldsymbol{t}_{j_2}}\ell_i(\boldsymbol{x}_i, \boldsymbol{t})\Big]\rangle \\ & + ... + \frac{1}{k!}\langle [\boldsymbol{\eta} \otimes^k \boldsymbol{\eta}], \Big[\frac{\partial^k}{\partial \boldsymbol{t}_{j_1}, .., \partial \boldsymbol{t}_{j_k}}\ell_i(\boldsymbol{x}_i, \boldsymbol{t})\Big]\rangle + ..\Big] \end{aligned} \tag{6.11}$$

We move expectation operator inside both brackets and take expectation of only the first term in the inner product. Using $\boldsymbol{\eta} = \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}$, we get $E_\epsilon[\boldsymbol{\eta} \otimes^k \boldsymbol{\eta}] = [\boldsymbol{\sigma}_t \otimes^k \boldsymbol{\sigma}_t] \odot E_\epsilon[\boldsymbol{\epsilon} \otimes^k \boldsymbol{\epsilon}]$. Using these in eq.(6.11) yields the required result. □

The first term of Result 1, $\ell_i(\boldsymbol{x}_i, \boldsymbol{t})$ (after ignoring $\frac{1}{\boldsymbol{\sigma}_{xi}^2}$), would be the only term in the cost function if the latent space were deterministic. The rest of the terms are additional in stochastic training. Each of these terms is an inner product of two tensor, the first being $[\boldsymbol{\sigma}_t \otimes^k \boldsymbol{\sigma}_t] \odot E_\epsilon[\boldsymbol{\epsilon} \otimes^k \boldsymbol{\epsilon}]$, and the second being the k^{th} order partial derivative

tensor $\left[\frac{\partial^k}{\partial \boldsymbol{t}_{j_1},..,\partial \boldsymbol{t}_{j_k}}\ell_i(\boldsymbol{x}_i,\boldsymbol{t})\right]$. We can thus consider the first tensor as providing penalizing weights to different partial derivatives in the second tensor. Since each inner product is added to the cost, we are minimizing them during optimization. This gives two important implications:

1. For sufficiently large samples, $E_\epsilon[\boldsymbol{\epsilon} \otimes^k \boldsymbol{\epsilon}]$ must be close to central moments of isotropic Gaussian. However, in practice, the number of samples of ϵ remains constant. As we move to the higher order moment tensors, we can expect that they do not converge to that of the standard Gaussian. This, luckily, works in our favor. Since we are minimizing $\frac{1}{k!}\langle[\boldsymbol{\sigma}_t \otimes^k \boldsymbol{\sigma}_t] \odot E_\epsilon[\boldsymbol{\epsilon} \otimes^k \boldsymbol{\epsilon}], \left[\frac{\partial^k}{\partial \boldsymbol{t}_{j_1},..,\partial \boldsymbol{t}_{j_k}}\ell_i(\boldsymbol{x}_i,\boldsymbol{t})\right]\rangle$ for each order, the inner product can be vanished for arbitrary ϵ only by driving partial derivative tensors towards zero. Therefore, minimizing the sum of all the inner product for arbitrary ϵ would minimize most of the terms in the partial derivative tensor. From Proposition 1, this corresponds to minimizing the variation of function ℓ_i, and consequently variation of the total error function ℓ according to eq.(6.10). Hence, additional terms in the stochastic latent space formulation contributes to decreasing the variation $V[f]$ and consequently the generalization gap.

2. Not all the partial derivatives are equally weighted in the cost function. Due to the presence of weighting tensor $[\boldsymbol{\sigma}_t \otimes^k \boldsymbol{\sigma}_t]$ in the first tensor of inner product, different partial derivative terms are penalized differently according to the value of $\boldsymbol{\sigma}_t$. Combination of the KL divergence term in eq.(6.9) with T_1 tries to increase standard deviation, $\boldsymbol{\sigma}_t$ towards 1 whenever it does not significantly increase the cost T_1: higher value of $\boldsymbol{\sigma}_t$ penalizes the partial derivatives of a certain direction more heavily, making the cost flatter in some directions than other.

Strictly speaking, Proposition 1 requires cross partial derivatives to be small throughout the domain of latent variable, which is not included in the above analysis. It however should not significantly affect the observation that, compared to deterministic formulation, the stochastic formulation decreases the variation $V[f]$.

6.6 Experiments & Results

Since it is not possible to obtain real TMP data, the reconstruction network is trained on simulated data pairs of $\boldsymbol{y}$ and $\boldsymbol{x}$. We focus on evaluating three generalization tasks of the network: to learn how to reconstruct under the prior physiological knowledge given in simulation data while generalizing to 1) unseen pathological conditions in $\boldsymbol{x}$, 2) unseen geometrical variations in $\boldsymbol{y}$ that are irrelevant to $\boldsymbol{x}$, and 3) real clinical data.

6.6.1 Generalizing outside the training distribution of TMP

Dataset and implementation details:

We simulated training and test sets using three human-torso geometry models. Spatiotemporal TMP sequences were generated using the Aliev-Panfilov (AP) model [2], and projected to the body-surface potential data with 40dB SNR noises. Two parameters were varied when simulating the TMP data: the origin of excitation and abnormal tissue properties representing myocardial scar. Training data were randomly selected with regard to these two parameters. Test data were selected such that values in these two parameters differed from those used in training in four levels: 1) Scar: Low, Exc: Low, 2) Scar: Low, Exc: High, 3) Scar:High, Exc: Low, and 4) Scar: High, Exc: High, where Scar/Exc indicates the parameter being varied and High/Low denotes the level of difference (therefore difficulty) from the training data. For example, Scar: Low, Exc: High test ECG data was simulated with region of scar similar to training but origin of excitation very different from that used in training.

For all four models being compared (svs stochastic/deterministic and svs-L stochastic/deterministic), we used ReLU activation functions in both the encoder and decoder, ADAM optimizer [57], and a learning rate of 10^{-3}. Each neural network was trained on approximately 2500 TMP simulations on each geometry. In addition to the four neural networks, we included a classic TMP inverse reconstruction method (Greensite) designed to incorporate temporal information [44]. On each geometry, approximately

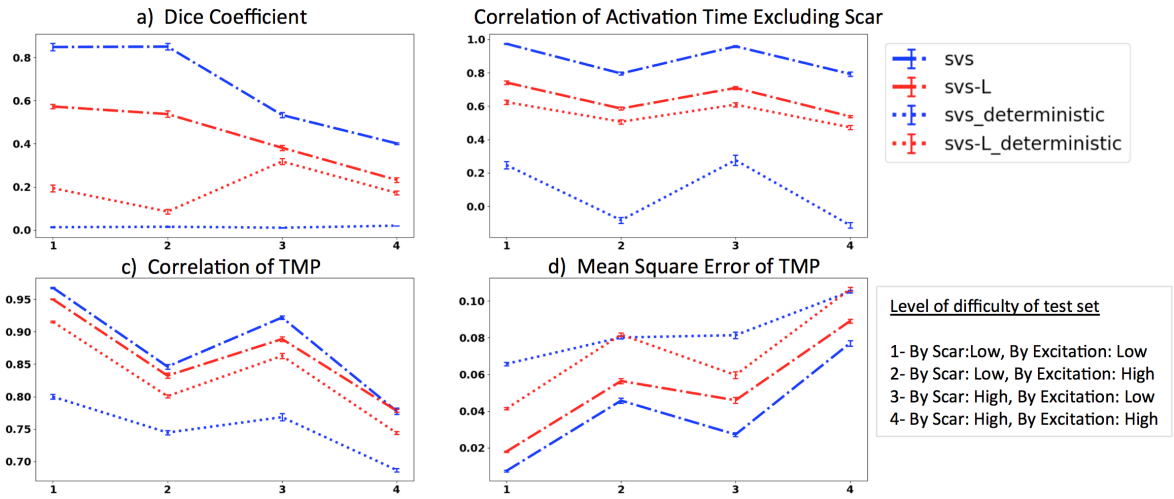


Figure 6.2: Reconstruction accuracy of different architectures at the presence of test data at different levels of pathological differences from training data.

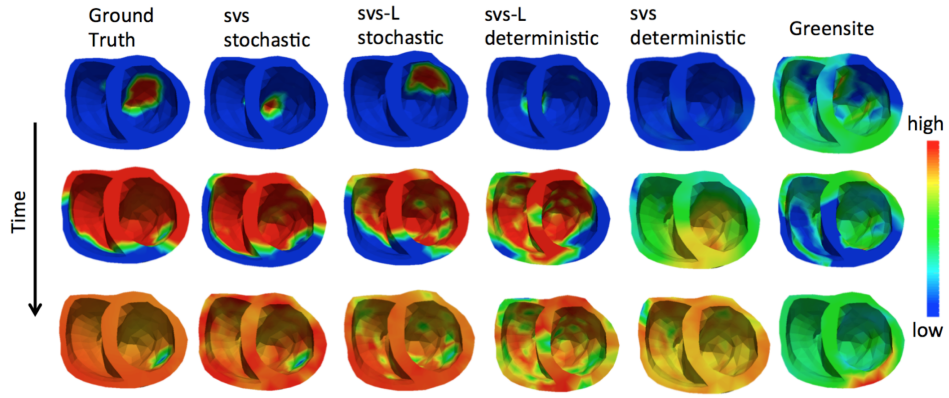


Figure 6.3: Examples of TMP sequences reconstructed by different methods being compared.

300 cases were tested for each of the four difficulty levels. We report the average and standard deviation of the results across all three geometry models.

Results:

The reconstruction accuracy was measured with four metrics: 1) mean square error (MSE) of the TMP sequence, 2) correlation of the TMP sequence, 3) correlation of

Table 6.1: Accuracy of different architectures at reconstructing unseen pathological conditions

Method \Metric	MSE	TMP Corr.	AT Corr.	Dice Coeff.
svs stochastic	$\mathbf{0.037 \pm 0.021}$	$\mathbf{0.885 \pm 0.061}$	$\mathbf{0.885 \pm 0.072}$	$\mathbf{0.645 \pm 0.181}$
svs deterministic	0.075 ± 0.013	0.77 ± 0.038	0.12 ± 0.13	0.01 ± 0.006
svs-L stochastic	0.068 ± 0.023	0.838 ± 0.053	0.601 ± 0.074	0.28 ± 0.154
svs-L deterministic	0.067 ± 0.02	0.84 ± 0.053	0.57 ± 0.052	0.165 ± 0.092
Greensite	–	–	0.514 ± 0.006	0.138 ± 0.005

TMP-derived activation time (AT), and 4) dice coefficients of the abnormal scar tissue identified from the TMP sequence. As summarized in Figure 6.2 and Table 1, in all test cases with different levels of pathological differences from the training data, the stochastic version of each architecture was consistently more accurate than its deterministic counterpart. In addition, most of the networks delivered a higher accuracy than the classic Greensite method (which does not preserve TMP signal shape and thus its MSE and correlation of TMP was not reported), and the accuracy of the *svs* stochastic architecture was significantly higher than the other architectures. These observations are reflected in the examples of reconstructed TMP sequences in Fig. 6.3.

6.6.2 Generalization to geometrical variations irrelevant to TMP

Dataset and implementation details:

TMP data were simulated as described in the previous section, but on a single heart-torso geometry. ECG data were simulated from TMP with controlled geometrical variations by rotating the heart along Z-axis at different angles (-20 degree to +20 degree at the interval of 1 degree). We trained the network to reconstruct TMP using ECG simulated by i) using five rotation angles from -2 degree to 2 degree, ii) ten rotation angles from -4 degree to +5 degree. We then compared the stochastic and deterministic *svs* networks on test ECG generated by the rest of the rotation angles. The network architecture and training details were the same as described in the previous section. Test ECG sets at each rotation angle were generated from 250 TMP signals with different tissue properties and origins of excitation and we report the mean and standard deviation of results for each angle.

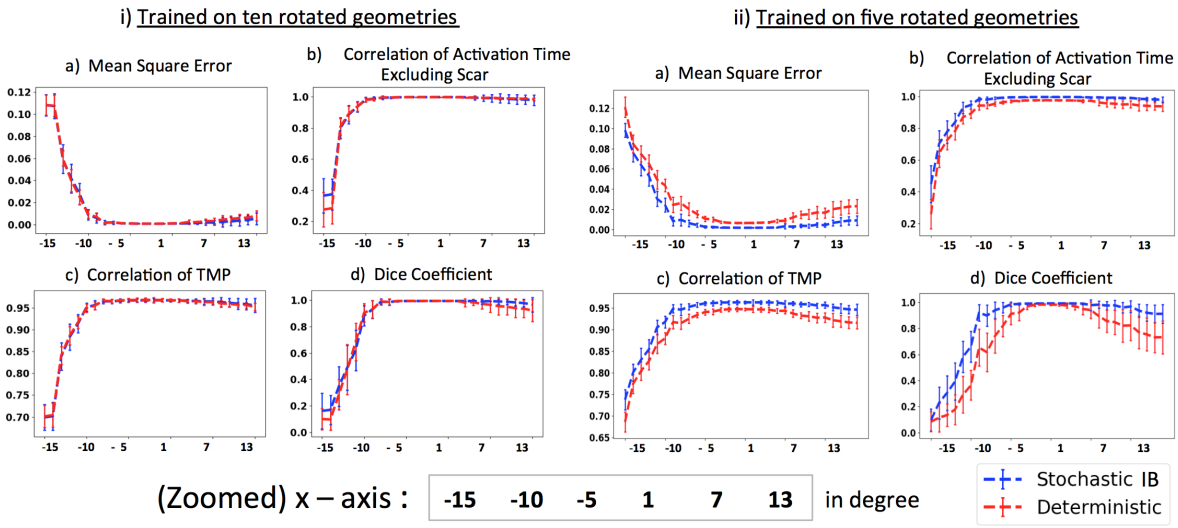


Figure 6.4: Comparison of TMP reconstruction by stochastic *vs.* deterministic networks using training data with a i) high and ii) low amount of variations in geometrical factors irrelevant to TMP. Values along the x axis shows the degree of rotation of the heart relative to the training set, *i.e.,* cases in the center of the x-axis are the closest to the training data.

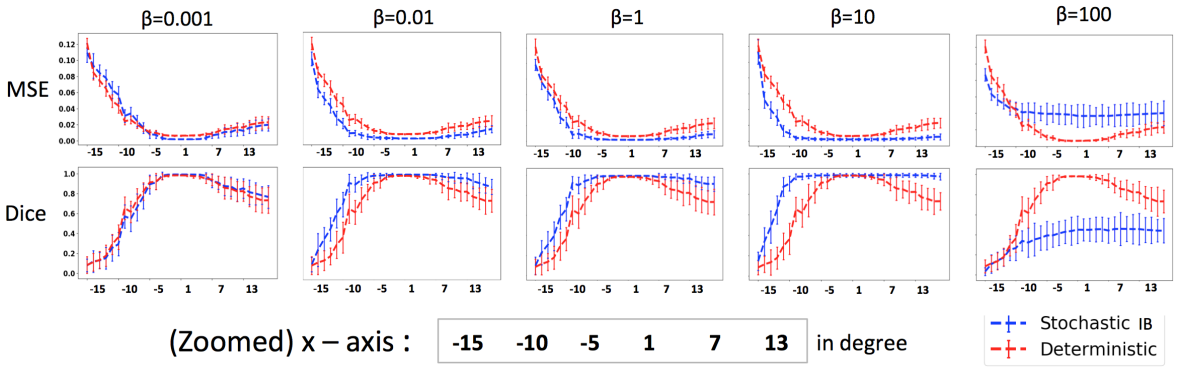


Figure 6.5: Comparison of stochastic *vs.* deterministic architectures at different values of β. At $\beta = 10$, the error stays low and flat for a large range of deviation in angles in stochastic architecture.

Results:

As summarized in Fig. 6.4(ii), when trained on a small interval of five rotation values, the stochastic information bottleneck consistently improves the ability of the network

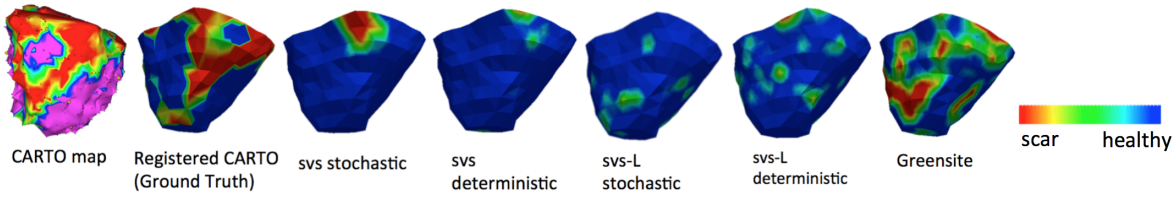


Figure 6.6: Comparison of scar region identified by different architectures and the Greensite method with reference to *in vivo* voltage maps.

to generalize to geometrical values outside the training distribution. This margin of improvement also increases as we move further away from the training set, *i.e.* as we go left or right from the centre, and seems to be more pronounced when measuring the dice coefficient of the detected scar. When trained on a larger interval of ten rotation values, however, this performance gap diminishes as shown in Fig. 6.4(i). This suggests that the encoder-decoder architecture with compressed latent space can naturally learn to remove variations irrelevant to the network output, although the use of stochastic information bottleneck allows the network to generalize from a smaller number of training examples.

To understand how the parameter β in the information bottleneck loss $\mathcal{L}_{IB}$ plays a role in generalization, we repeated the above experiments with different values of β. As shown in Fig. 6.5, as we increase β, the generalization ability of the network first increases and then degrades reaching optimum value at $\beta = 10$.

6.6.3 Generalization to real data: a feasibility study

Finally, we tested the presented networks – trained on simulated data as described earlier – on clinical 120-lead ECG data obtained from a patient with scar-related ventricular tachycardia. From the reconstructed TMP sequence, the scar region was delineated based on TMP duration and compared with low-voltage regions from *in-vivo* mapping data. As shown in Fig. 6.6, because the network is directly transferred from the simulated data to real data, the reconstruction accuracy is in general lower than that in synthetic cases. However, similar to the observations in synthetic cases, the

svs stochastic model is able to reconstruct the region of scar that is the closest to the *in-vivo* data.

6.7 Conclusion

To our knowledge, this is the first work that theoretically investigate the generalization of inverse reconstruction networks through the two different perspectives of stochasticity and information bottleneck, supported by carefully designed experiments in real-world applications. Note that the upper bound $\mathcal{L}_{IB} \geq loss_{IB} + D_{KL}(p(\boldsymbol{w})||r(\boldsymbol{w}))$. Therefore, minimizing $\mathcal{L}_{IB}$ puts an additional constraint on the marginal $p(\boldsymbol{w})$ to be close to a predefined $r(\boldsymbol{w})$. It is possible that the choice of $r(\boldsymbol{w})$ might also play a role in generalization and will be reserved for future investigations. Future works will also extend the presented study to a wider variety of medical image reconstruction problems.

6.8 Summary and Answer to Research Questions

Q. 3. a) How can we understand and improve generalization when there is possibility of shift in training and test distribution?

When we know that the test data may be shifted from the training data due to the presence of nuisance factors, like geometric variation as shown in this chapter, then, we may be able to counter the effect of that variation by controlling the flow of information using information bottleneck principle. We showed that this, in turn, helps the network learn representations that are invariant to such nuisance factors and improves generalization. We showed in the experiments that this strategy improves generalization ability under low data situation.

Another situation of data shift that we investigated is the shift in the training and test distribution due to the variation in the source. In such situation, we learn invariant

representation by adversarial training strategy. This is particularly successful if there is good variation among different sources and the factor of variation remains similar in the new test data from future sources.

Q.3. b) How can we understand and quantify the role of smoothness and regularity properties of the neural network regarding generalization?

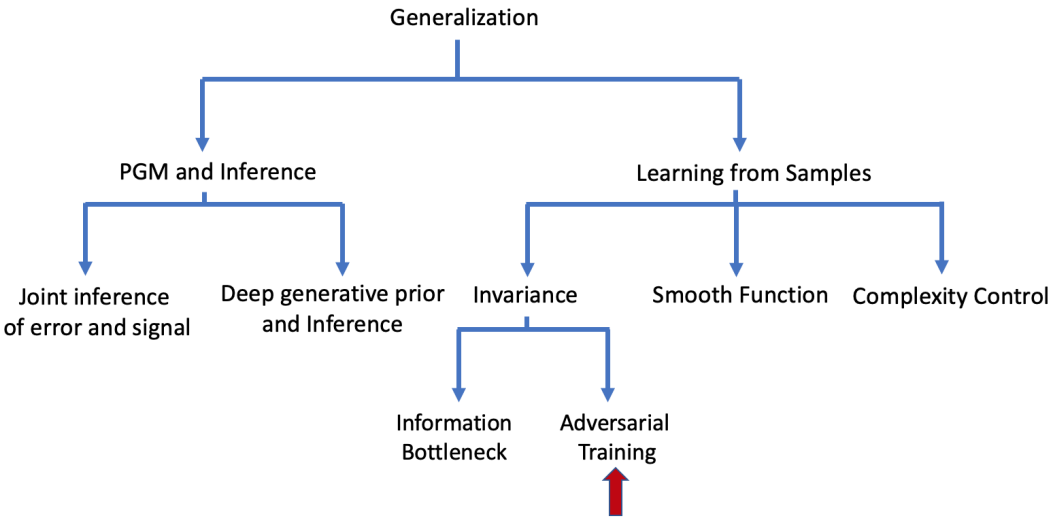
By using analytical learning theory, we argued that if the decoder is smooth, it helps reduce the generalization gap. We then theoretically established the link between random perturbation (stochasticity) in the latent space (as in VAE type reparameterization) and the smoothness of the decoder. Through experiments, we showed that the stochasticity in the latent space does help to improve generalization of the reconstruction network.

Chapter 7

Generalization via Invariance

...the search for constancy, the tendency towards certain invariants, constitutes a characteristic feature and immanent function of perception. This function is as much a feature of perception of objective experience as it is a condition of objective knowledge.

- Ernst Cassirer



7.1 Introduction

Automatic interpretation and disease detection in chest X-ray images is a potential use case for artificial intelligence in reducing the costs and improving access to healthcare. This modality is the most commonly prescribed in the world, not only in the context of clinical examination, but also for routine screening and even legal procedures such as health surveys for immigration purposes. Therefore, analysis of X-ray images through several computer vision algorithms has been an important topic of research in the past. Recently, deep learning based image classification [53, 88] has found important application in this area.

Most of these deep learning approaches are trained and tested on the same dataset and/or a single source. This is an unrealistic assumption in the case of medical image analysis. In radiology, we can always expect different images coming from different scanner, population, or image settings and therefore we can expect test images are different from the ones used in training. In non-quantitative imaging modalities, such as X-ray, this inconsistency of images across datasets is even more drastic. Another source of variability comes from the patients: one can always expect that the X-ray images of new patients in the future would be somewhat different from the images the network is trained on. Therefore, variability in test images should be considered a natural setting and models should be trained in such a way that it would work equally well when test images are different from the trained ones.

Unfortunately, the current methods are not geared towards addressing the scenario of test images being different from the training set. We found that the current popular deep learning architectures in medical imaging suffer from drop in performance when an X-ray image from a new source dataset is tested on a model trained on different dataset (see Table 7.2). This is a significant hurdle for adaptation of AI in the practice of radiology. The question of generalization across different sources of X-ray images, therefore, is an important clinical problem that needs our prompt attention. Recently, this need has been realized and the radiology editorial board has encouraged testing in *external* test set [13]. However, there are limited works looking at the issue of this

source generalization in medical imaging. Some of the works known to have tried to answer the question of generalization by intensity normalization and adding Gaussian noise layers to neural networks [60] while others use simple ensemble strategy as in [73].

Towards the resolution of this issue, we look at this problem from the perspective of generalization to out-of-distribution test cases under certain assumptions. We note that the this problem of generalization cannot be tackled by using rich literature in the field of statistical learning theory (SLT) [105] because SLT starts from the fundamental assumption that the training and testing data are randomly selected (iid assumption) from the same distribution. To address this generalization problem, we need to think about how to generalize when iid assumption is invalid and test sets are from a different distribution. Drawing ideas from causality and invariant risk minimization [3], we propose that the key is to learn features that are invariant in several X-ray datasets, and would be valid features even for the new test cases. We achieve feature invariance by using adversarial penalization strategy. To learn this invariant representation, we need different X-ray datasets that essentially contain similar diseases, but are different due to different practical image acquisition and other nuisance factors. Our second contribution in this paper is to develop and provide such a mixed dataset combining four different public sets of images, with new labeling for pneumonia and consolidation, to study this problem and for the benefit of the community.

In this chest X-ray image classification task, we train the network with data from multiple sources and test on an X-ray dataset from an entirely different source to assess generalization ability. We show that the proposed method does help in generalization. We also perform experiments using Grad-CAM [93] to localize the regions in X-ray that are attended by the network during classification. Using Grad-CAM, we qualitatively evaluate and compare the behaviour of the baseline and the proposed method.

This chapter includes parts from author's conference publication [36]. This was a joint work with the team from IBM Research, Almaden, San Jose.

7.2 Related Work

Generalization is an old topic in machine learning. Earlier works on generalization concentrated on statistical learning theory [14,105], studying the worst case generalization bound based on the capacity of the classifier. Later on, other view points emerged like PAC Bayes [72], information theoretic [113] and stability based methods [15]. Modern works on generalization, however, find statistical learning theory insufficient [116] and propose other theories from analytical perspectives [56]. Our work is quite different from these works. Most of these works are about in-source generalization and assume that data is independent and identically distributed (i.i.d) both in training and testing. We, however, start with the assumption that the training and testing could be from different distributions, but share some common, causal features. Based on the principles of Invariant Risk Minimization [3], we propose the idea that learning invaraint features from multiple sources could lead to learning causal features which would help in generalization to new sources.

Another closely related area to our work is that of domain adaptation [31, 94], and its application in medical imaging [20]. In a domain adaptation setting, the data is available from source and target domains; but, the labels are available only from the source domain. The objective is to learn to adapt knowledge from source to predict label of the target. Although similar in spirit, our work is quite different from domain adaptation in that we do not have target data to adapt to during training. Rather than adapting from source to target, we are interested in generalization to any new data.

We draw idea of distribution matching using GANs from unsupervised domain adaptation [31]. Other ideas of distribution matching like Maximum Mean Discrepancy (MMD) [63, 64] are related to our work. In comparison, the adversarial approach has been found to be very powerful and easily extendable to more than two sources, which is cumbersome to realize using MMD.

7.3 Method

7.3.1 Main Idea

Our key idea to improve generalization stems from the intuition of ignoring irrelevant features in X-ray images and focus on important, causal features. Imagine a radiologist trying to diagnose pneumonia from the X-ray images. What enables her to generalize her knowledge to a case she has never seen before? We argue that she can generalize because she focuses on key features relevant to pneumonia while ignoring other irrelevant features that might vary with the source of X-ray image. We can think of the key features in X-ray image that indicate pneumonia as causal features, while the features that vary from sources to sources but are not relevant for the purpose of pneumonia as non-causal features.

Causation as Invariance Following reasoning similar to [3], we argue that extracting invariant features from many different sources would help the network focus on extracting causal features. This would help the network generalize to new sources in the future assuming that it would extract causal features from the new X-ray images obtained in the future.

To force a network to learn invariant features, we propose an architecture as shown in Fig. 7.1 based on adversarial penalization strategy. It has three major components: Feature extractor, Discriminator and Classifier. Drawing ideas from unsupervised domain adaptation [31], we train the discriminator to classify which source the image was obtained from just using the latent features extracted by the feature extractor. The discriminator is trained to well identify the source from the features. The feature extractor, however, is trained adversarially to make it very difficult for the discriminator to classify among sources. This way, we force the feature extractor network to extract features from the X-ray images that are invariant across different sources for if there were any element in the latent feature that is indicative of the source, it would be easier for the discriminator to identify the sources. At the end, we expect the feature extractor and discriminator to reach an equilibrium where the feature extractor

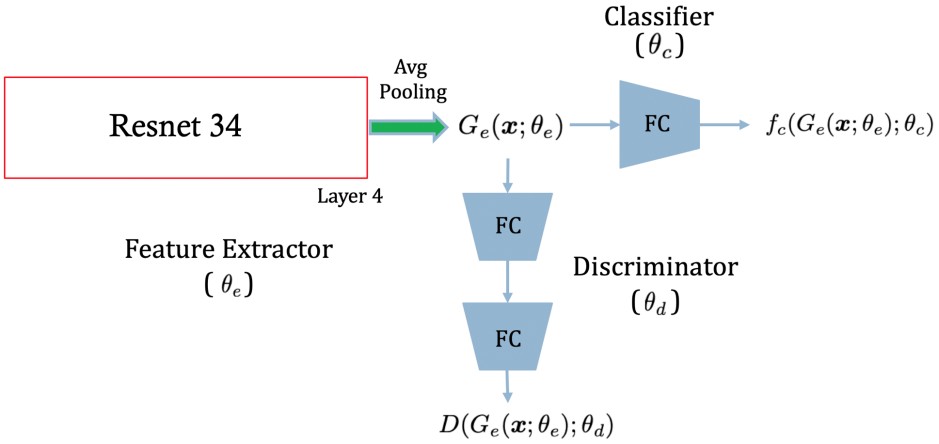


Figure 7.1: Proposed architecture to learn source invariant representation while simultaneously classifying disease labels

generates features that are invariant to the sources. Meanwhile, the same features are fed to the disease classifier which is trained to properly identify disease. Hence, the features must be source invariant and at the same time discriminative enough of the disease. Next, we describe three main components of our network.

1. Feature extractor: The feature extractor is the first component that takes in the input X-ray image and gives a latent representation. In Fig. 7.1, the feature extractor consists of a Resnet 34 [51] architecture up to layer 4 followed by global average pooling layer.

2. Discriminator: the discriminator consists of fully connected layers which take in features after global average pooling layer and try to classify which of the sources the image is obtained from. If adversarial training reaches equilibrium, it would mean that feature representation from different sources are indistinguishable (source invariant).

3. Classifier: The output of the feature extractor network should not only be source invariant, but also be discriminative to simultaneously classify X-ray images according to the presence or absence of disease. In our simple model, we simply use a fully connected layer followed by sigmoid as the classifier.

7.3.2 Training

From Fig. 7.1, the disease classification loss and source classification (discrimination) loss are respectively defined as:

$$\mathcal{L}_p(\theta_e, \theta_c) = \underset{p(\boldsymbol{x},\boldsymbol{y})}{E}[\ell_{BCE}(f_c(G_e(x;\theta_e);\theta_c), y)] \tag{7.1}$$

$$\mathcal{L}_s(\theta_e, \theta_d) = \underset{p(\boldsymbol{x},\boldsymbol{y_s})}{E}[\ell_{CE}(D(G_e(x;\theta_e);\theta_d), y_s)] \tag{7.2}$$

where $\ell_{BCE}(\hat{y}, y) = y \log \hat{y} + (1-y)\log(1-\hat{y})$ is the binary cross entropy loss and similarly $\ell_{CE}(\hat{y}, y) = \sum_i (y_s)_i \log(\hat{y}_s)_i + (1-(y_s)_i)\log(1-(\hat{y}_s)_i)$ is the cross entropy loss. We train extractor, classifier and discriminator by solving following min-max problem.

$$\hat{\theta}_e, \hat{\theta}_c = \underset{\theta_e,\theta_c}{argmin}\ \mathcal{L}_p(\theta_e, \theta_c) - \lambda \mathcal{L}_s(\theta_e, \hat{\theta}_d), \quad \hat{\theta}_d = \underset{\theta_d}{argmin}\ \mathcal{L}_s(\hat{\theta}_e, \theta_d) \tag{7.3}$$

It is easy to note that this is a two player min-max game where two players are trying to optimize an objective in opposite directions: note the negative sign and positive sign in front of loss $\mathcal{L}_s$ in eq.(7.3). Such min-max games in GAN literature are notorious for being difficult to optimize. However, in our case optimization was smooth as there was no issue with stability.

To perform adversarial optimization, two methods are prevalent in the literature. The first method, originally proposed in [43], trains the discriminator while freezing feature extractor and then freezes discriminator to train feature extractor while inverting the sign of loss. The second approach was proposed in [31], which uses a gradient reversal layer to train both the discriminator and feature extractor in a single pass. Note that the former method allows multiple updates of the discriminator before updating the feature extractor while the latter method does not. Many works in GAN literature reported that this strategy helped in learning better discriminators. In our experiments, we tried both and found no significant difference between the two methods in terms of stability or result. Hence, we used gradient reversal because it was time efficient. To optimize the discriminator, it helps if we have balanced dataset from each source. To account for imbalanced dataset from each source, we resample data from the source with

small size until the source of largest size is exhausted. By such resampling, we ensure that there is a balanced stream of data from each source to train the discriminator.

7.3.3 Grad-CAM Visualization

Grad-CAM [93] identifies important locations in an image for the downstream tasks like classification. It visualizes the last feature extraction layer of a neural network scaled by the backpropagated gradient and interpolated to the actual image size.

In this paper, we use Grad-CAM to visualize which location in the X-ray is being attended by the neural network when we train with and without adversarial penalization. Our hypothesis is that a method that extracts source invariant features should be extracting more relevant features to the disease to be identified, whereas a network which was trained without specific guidance to extract source invariant features would be less focused in the specific diseases and may be attending to irrelevant features in the input X-ray image. Using Grad-CAM, we qualitatively verify this hypothesis.

7.4 Datasets and Pneumonia/Consolidation Labeling Scheme

To learn invariant features from X-ray dataset for detecting signs of diseases, we need labeled datasets from several different sources. We have created such a dataset using publicly available images and generating labels when necessary. We include recently released large datasets of chest X-ray images like the ChestXray14 dataset released by NIH [110], MIMIC-CXR dataset [55] released by MIT Laboratory for Computational Physiology, a part of the Institute for Medical Engineering and Science , and CheXpert dataset released by researchers at Stanford [53]. We also have access to a smaller internally curated dataset of images originating from Deccan Hospital in India.

We are interested in classification task detecting signs of pneumonia and consolida-

tion in chest X-ray images. Consolidation is a symptom of disease (occuring when alveoli is filled with something other than air, such as blood) whereas Pneumonia is a disease often causing consolidation. Radiologists use consolidation, potentially with other signs and symptoms, to diagnose pneumonia. In a radiology report, both of these may be mentioned. Therefore, we have used both to build a dataset of pneumonia/consolidation.

We have used all four datasets listed above. The Stanford CheXpert dataset [53] is released with images and labels, but without accompanying reports . The NIH dataset is also publicly available with only images and no reports. A subset of 16,000 images from this dataset were examined by our radiologists and prepared reports. For the MIMIC dataset, we have full-fledged reports provided under a consortium agreement to us for the MIMIC-4 collection recently released [54]. For Deccan collection, we have the reports along with images. For the NIH, MIMIC and Deccan datasets, we used our natural language processing (NLP) labeling pipeline [23, 66], to find positive and negative examples in the reports whereas for the Stanford dataset, we used the labels provided by the Stanford team.

Using NLP generated and available labels (for CheXpert), we created training dataset by including images with positive indication of pneumonia or consolidation in our positive set and those with no indication of pneumonia or consolidation in the negative set. Table 7.1 lists the number of images from each class for each dataset.

7.5 Experiments and Results

We use four datasets as shown in Table.7.1. We use simple Resnet-34 architecture with classifier as our baseline so that enforcement of invariance through discriminator is the only difference between baseline and proposed method. Experiments using both the architecture use a leave-one-dataset-out strategy: we trained on three of the four datasets and left one out. Each experiment has two test sets: 1)in-source test that draws from only the unseen samples from datasets used for training, 2) out-of-source

Table 7.1: The distribution of the datasets used in the paper. The breakdown of the Positive (pneumonia/consolidation) and Negative (not pneumonia/consolidation) cases.

Leave out Dataset	**Train**		**Test**	
	Positive	Negative	Positive	Negative
Stanford	15183	123493	1686	13720
MIMIC	83288	49335	23478	13704
NIH	1588	6374	363	1868
Deccan Hospital	50	1306	12	379
Total	**100109**	**180508**	**25539**	**29671**

Table 7.2: The classification results in terms of area under ROC curve from baseline ResNet34 model, and our proposed architecture. Each row lists a leave-one-dataset-out experiment.

Leave out Dataset	**Baseline**		**Proposed Architecture**	
	in-source test	out-of-source test	in-source test	out-of-source test
Stanford	0.74	0.65	0.74	0.70
MIMIC	0.80	0.64	0.80	0.64
NIH	0.82	0.73	0.71	0.76
Deccan Hospital	0.73	0.67	0.75	0.70

test set, only including test samples from the fourth dataset that is not used in training. Note that all images from all sources are resized to 512x512.

The results of the classification experiments are listed in Table 7.2. We have chosen the area under ROC curve (AUC-ROC) as the classification metric since this is the standard metric in computer-aided diagnosis. The first observation is that in all experiments, both for baseline and for our proposed architecture, the AUC-ROC curve decreases as we move from in-source test set to the out-of-source test set as expected. However, this drop in accuracy is generally smaller in our proposed architecture. For example,

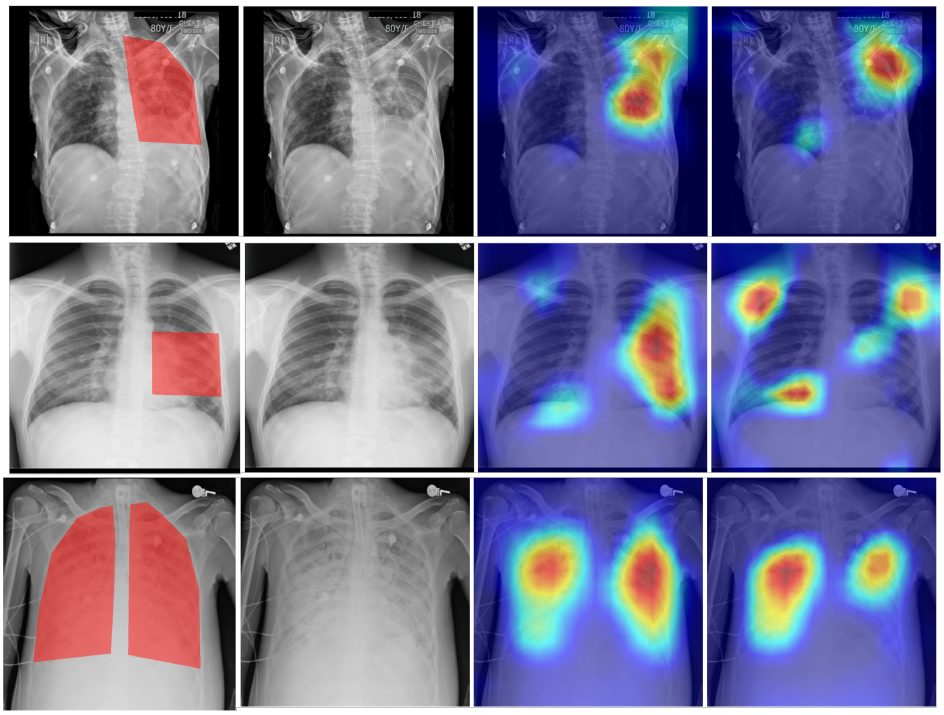


Figure 7.2: The qualitative comparison of the activation maps of the proposed and the baseline models with the annotation of an expert radiologist. The first column shows the region marked by the expert as the area of lung affected by pneumonia. The second column shows the original image for reference. The third and fourth columns are the Grad-CAM activation of the proposed and baseline models respectively.

when the Stanford dataset is left out of training, in the baseline method the difference between in-source and out-of-source tests is 0.09 (from 0.74 to 0.65), whereas in our proposed architecture, the drop in AUC-ROC is only 0.05 (from 0.74 to 0.70). While the performance on the in-source test stays flat, we gain 5% improvement in area under ROC curve, from 0.65 to 0.7, for out-of-source test.

Similar pattern holds in both the case of NIH and Deccan datasets: in both cases, the drop in performance due to out-of-source testing is smaller for the proposed architecture compared with the baseline classifier. Surprisingly for the NIH dataset, the out-of-source testing results in higher accuracy, which we interpret as heavy regularization during training. In case of the MIMIC dataset, the performance remains the same for

baseline and the proposed method.

Figure 7.2 shows Grad-CAM visualization to qualitatively differentiate between the regions or features focused by a baseline model and the proposed model while classifying X-ray images. Three positive examples and their activation maps are shown. The interpretation of activation maps in chest X-ray images is generally challenging. However, the evident pattern is that the heatmaps from the proposed method (third column) tends to agree more than the baseline (forth column) with the clinician's marking in the first column. Furthermore, proposed method shows fewer spurious activations. This is especially true in row 2 wherein the opacity from the shoulder blades are falsely highlighted as lungs pneumonia.

7.6 Conclusion and future work

We tackled the problem of out of source generalization in the context of chest X-ray image classification problem by proposing an adversarial penalization strategy to obtain a source-invariant representation. The availability of multiple public datasets allowed us to test our method through leave-one-dataset-out training and then testing on the left out dataset. In experiments, we show that the proposed algorithm provides improved generalization compared to the baseline. In the course of this work, we developed labeling methods and applied to the text reports accompanying these datasets to find positive samples for pneumonia/consolidation. These pneumonia/consolidation label lists constitute a new resource for the community and will be released publicly.

It is important to note that the performance on the in-source test set does not necessarily increase in our method. Mostly it stays flat except in one case, namely the NIH set, where the baseline beats the proposed method in the in-source test. This can be understood as a trade-off between in-source and out-of-source performance induced by the strategy to learn invariant representation, *i.e.*, by learning invariant features our objective is to improve on the out-of-source test cases even if in-source performance degrades. A possible route for further examination is the impact of the size of the

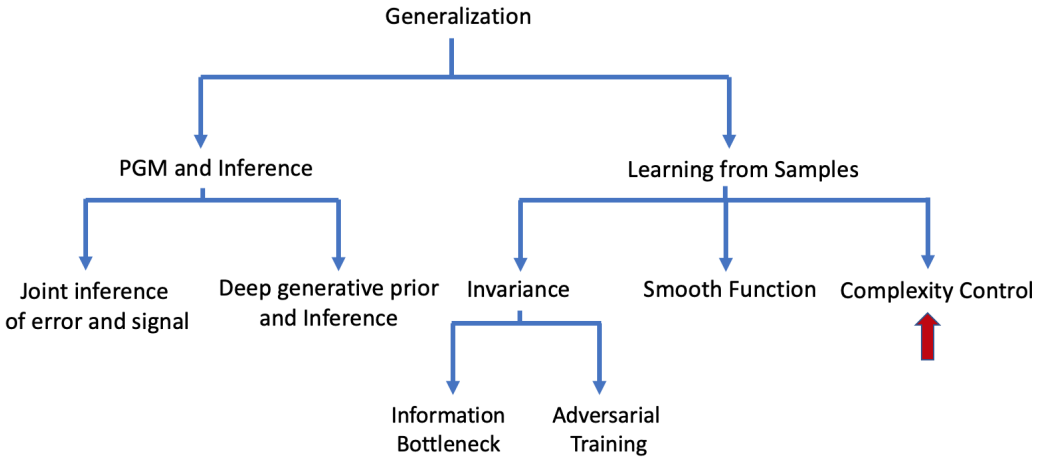
training datasets and left-out set on the behaviour of the model. It is noteworthy that we have kept the feature extractor and classifier components of our current architecture fairly simple to avoid excessive computational cost owing to adversarial training and large data and image size. A more sophisticated architecture might enhance the disease classification performance and is left as future work.

Chapter 8

Complexity Control

I claim that many patterns of Nature are so irregular and fragmented, that, compared with Euclid—a term used in this work to denote all of standard geometry—Nature exhibits not simply a higher degree but an altogether different level of complexity . . . The existence of these patterns challenges us to study these forms that Euclid leaves aside as being "formless," to investigate the morphology of the "amorphous."

- Benoit Mandelbrot



In the previous chapters, we saw that smoothness of the neural function from the input space to the output space is a crucial component to improve generalization. In the literature, there are other works including that of Bartlett et al. [8] who also point to the importance of smoothness of neural functions in generalization. Comparing this idea of smoothness with other works in the field of adversarial robustness [24] and stability of generative adversarial networks [101] made me realize that the problem of lack of regularity of neural functions is a fundamental one. The fact that we do not care about the regularity of neural functions during training, and that there is no direct way to control it might lie at the core of numerous pathology of neural networks including adversarial examples, instability in training of GANs and difficulty in generalization. For example, let's take Cohen et al. [24] regarding existence of adversarial examples and remedy via randomized smoothing. Their arguments can be summarized in simple terms as follows. Adversarial examples exist if there exists an examples $x+\epsilon$ near x such that they have different labels (say y and y'). But, that is only possible if the function from x to y fluctuates very rapidly within a small neighborhood of ϵ, implying that the function is not very regular. Once this is understood, the solution is simple: just make the function smooth, which they do by something called randomized smoothing, and provide a guarantee. Once we understand this, we start to see connections between randomized smoothing and ensemble methods in semi-supervised learning, thereby essentially connecting adversarial robustness with generalization. Similarly, Hoang et al. [101] argue that GANs are unstable because as the training progresses, the discriminator should become more and more steeper. To understand this, let's assume x_1 and x_2 lie in two different distributions. As the two distributions come closer, the distance $x_1 - x_2 = \epsilon$ gets smaller and smaller, but the discriminator must always label them as 0 and 1 (without loss of generality) respectively. Consequently, the discriminator function must become extremely steep to the extent of being infinitely steep resulting into instability. Obviously, some kind of gradient penalization is always needed while training GANs and gradient penalization seems to do the trick.

While these things are to some extent understood and some ad-hoc fixes based on gradient penalization, gradient clipping, ensemble etc have been proposed, these solutions do not address the fundamental problem: there is no way to enforce regularity

throughout the input space during training of neural networks. Towards the direction of principled enforcement, we propose a different kind of neural network which lies in Reproducing Kernel Hilbert Space (RKHS). In addition, we provide a principled way to control the complexity of the function space providing a way to control the regularity of neural functions. We test this network in the application of estimating KL divergence via adversarial training, essentially demonstrating the efficacy of our construction to stabilize adversarial training.

Obviously, as a next step, we intend to extend this theory to improve generalization as well; and then to adversarial robustness. However, proving that regularity can solve all these problems and that this technique works for all of them is a daunting task, and is beyond the scope of this dissertation.

8.1 Introduction

Calculating Kullback–Leibler (KL) divergence from data samples is an essential component in many machine learning problems that involve Bayesian inference or the calculation of mutual information. In small data regime, this problem has been studied using variational technique and convex optimization [78]. In the presence of ever-increasing data, several neural network models have been proposed which require estimation of KL divergence such as total correlation variational autoencoder (TC-VAE) [21], adversarial variational Bayes (AVB) [74], information maximizing GAN (InfoGAN) [22], and amortized MAP [97]. These large scale models have imposed the following new requirements on estimating KL divergence: 1. Scalability: The estimation algorithm should be able to compute KL divergence from a large amount of data samples. 2. Minibatch compatibility: The algorithm should be compatible with minibatch-based optimization and allow backpropagation (or other ways of optimizing the rest of the network) based on the estimated value of KL divergence.

These needs make classic methods such as [78] impractical, but were met by modern neural network based methods such as variational divergence minimization (VDM)

[80], mutual information neural estimation (MINE) [9], and GAN-based KL estimation [76, 97]. A key attribute of these methods is that they are based on updating a neural-net discriminator function to estimate KL divergence from a subset of samples, which makes them scalable and minibatch compatible. We, however, noted that even in simple toy examples, these methods tended to be either unreliable (high fluctuation of estimates, as in GAN based approach by [76]), or unstable (discriminator yields infinity, as in MINE and VDM) (see Table.8.1). This behavior exacerbated when increasing the size of the discriminator. Similar observations of instability of VDM and MINE have been reported in the literature [76, 98].

In this paper, we attempt to provide a theoretical underpinning for the core problem of the large fluctuation in the GAN based estimation of KL divergence. We approach this problem from the perspective of sample complexity, and propose that these fluctuations are a consequence of not controlling the complexity of the discriminator function. This direction has not been explored in existing works, and it faces the open question of how to properly measure the complexity of the large function space represented by neural networks. Note that naive approaches to bound complexity by the number of parameters would neither be guaranteed to yield tight bound, nor be easy to implement because it requires dynamically changing the size of the network during optimization.

We introduce the following contributions to resolve this challenge. First, to be able to compute the complexity of the discriminator function space, we propose a novel construction of the discriminator such that it lies in a smooth function space, the Reproducing Kernel Hilbert Space(RKHS). Leveraging sample complexity analysis and mean embedding of RKHS, we then bound the probability of the error of KL-divergence estimates in terms of the complexity of RKHS space. This further allows us to theoretically substantiate our main proposition that not controlling the complexity of the discriminator may lead to high fluctuation in estimation. Finally, we propose a scalable way to control the complexity of the discriminator based on the obtained error probability bound. In controlled experiments, we demonstrate that failing to control the complexity of the discriminator function leads to fluctuation in KL divergence estimates, and that the proposed method decreases such fluctuations.

This chapter includes parts from author's publication [34].

8.2 Related Works

Nguyen et al [78] used variational function to estimate KL divergence from samples of two distribution using convex risk minimization (CRM). They used the RKHS norm of the variational function as a way to both measure and penalize the complexity of the variational function. However, their work required handling all data at once and solving a convex optimization problem that could not be scaled. VDM reformulates the f-Divergence objective using Fenchel duality and uses a neural network to represent the variational function [80]. It is in concept close to [78], while the use of neural network and adversarial optimization made the estimation scalable. It however did not control the complexity of the neural-net function, resulting in unstable estimations.

One area of modern application of KL-divergence estimation is in computing mutual information which, as shown in MINE [9], is useful in applications such as stabilizing GANs or realizing the information bottleneck principle. MINE also optimizes a lower bound, but tighter, to KL divergence (Donsker-Varadhan representation). Similar to VDM, MINE uses a neural network as the dual variational function: it is thus scalable, but without complexity control and unstable.

Another use of KL divergence is scalable variational inference (VI) as shown in AVB [76]. VI requires KL divergence estimation between the posterior and the prior, which becomes nontrivial when an expressive posterior distribution is used and requires sample based scalable estimation. AVB solved it using GAN based adversarial formulation and a neural network discriminator. Similarly, [97] used GAN based adversarial formulation to obtain KL divergence in amortized inference.

To disentangle latent representations in VAE, [21] proposed TC-VAE which penalized the KL divergence between marginal latent distribution and the product of marginals in each dimension. This KL divergence was computed by minibatch based sampling strategy that gives a biased estimate. None of the existing works considered the theo-

retical underpinning of unreliable KL-divergence estimates, or mitigating the problem by controlling the complexity of the discriminator function.

8.3 Preliminaries

Reproducing Kernel Hilbert Space: Let $\mathcal{H}$ be a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ defined on non-empty space $\mathcal{X}$. It is a Reproducing Kernel Hilbert Space (RKHS) if the evaluation functional, $\delta_x : \mathcal{H} \to \mathbb{R}$, $\delta_x : f \mapsto f(x)$, is linear continuous $\forall x \in \mathcal{X}$. Every RKHS, $\mathcal{H}_K$, is associated with a unique positive definite kernel, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, called reproducing kernel [10], such that it satisfies:

1. $\forall x \in \mathcal{X}, K(., x) \in \mathcal{H}_K$ (*Membership property*)
2. $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}_K,\ \langle f, K(., x)\rangle_{\mathcal{H}_K} = f(x)$ (*Reproducing property*)

RKHS is often studied using a specific integral operator. Let $\mathcal{L}_2(d\rho)$ be a space of functions $f : \mathcal{X} \to \mathbb{R}$ that are square integrable with respect to a Borel probability measure $d\rho$ on $\mathcal{X}$, we define an integral operator $L_K : \mathcal{L}_2(d\rho) \to \mathcal{L}_2(d\rho)$ [6, 25]: $(L_K f)(x) = \int_{\mathcal{X}} f(y)K(x, y)d\rho(y)$ This operator will be important in constructing a function in RKHS and in computing sample complexity.

Mean Embedding in RKHS: Let $f : \mathcal{X} \to \mathbb{R}$ be a function in RKHS, $\mathcal{H}_K$, and p be a Borel probability measures on $\mathcal{X}$. If $E_{x\sim p}\sqrt{K(x, x)} < \infty$, then we have $\mu_p \in \mathcal{H}_K$ called the mean embedding of the distribution p and defined as [10,45,99]: $E_{x\sim p} f = \langle f, \mu_p\rangle_{\mathcal{H}_K}$ The condition for the existence of mean embedding is readily satisfied since we assume $\sup_{x,t} K(x, t) < \infty$.

8.4 Problem Formulation and Contribution

GAN-based Estimation of KL Divergence: Let $p(x)$ and $q(x)$ be two probability density functions in space $\mathcal{X}$ and we want to estimate their KL divergence using finite samples from each distribution in a scalable and minibatch compatible manner. As shown in [76, 97], this can be achieved by using a discriminator function. First, a discriminator $f : \mathcal{X} \to \mathbb{R}$ is trained with the objective:

$$f^* = \max_f [E_{p(x)} \log \sigma(f(x)) + E_{q(x)} \log(1 - \sigma(f(x)))] \tag{8.1}$$

where σ is the Sigmoid function given by $\sigma(x) = \frac{e^x}{1+e^x}$. Then it can be shown [76, 97] that the KL divergence $KL(p(x)||q(x))$ is given by:

$$KL(p(x)||q(x)) = E_{p(x)}[f^*(x)] \tag{8.2}$$

Sources of Error: Typically, a neural network is used as the discriminator. This implies that we are considering the space of functions represented by the neural network of given architecture as the hypothesis space, over which the maximization occurs in eq.(8.1). We thus must rewrite eq.(8.1) as

$$f_h^* = \max_{f \in h} [E_{p(x)} \log \sigma(f(x)) + E_{q(x)} \log(1 - \sigma(f(x)))] \tag{8.3}$$

where h is the discriminator function space. Furthermore, we have only a finite number of samples, say m, from the distribution p and q. Then, under finite sample, the optimum discriminator is

$$f_h^m = \max_{f \in h} \Big[\frac{1}{m} \sum_{x_i \sim p(x_i)} \log \sigma(f(x_i)) + \frac{1}{m} \sum_{x_j \sim q(x_j)} \log(1 - \sigma(f(x_j))) \Big] \tag{8.4}$$

Similarly, we write KL estimate obtained from, respectively, infinite and finite samples as:

$$KL(f) = E_{p(x)}[f(x)], \quad KL_m(f) = \frac{1}{m} \sum_{x_i \sim p(x_i)} [f(x)] \tag{8.5}$$

With these definitions, we can now write the error in estimation as:

$$KL_m(f_h^m) - KL(f^*) = \underbrace{KL_m(f_h^m) - KL(f_h^m)}_{\text{Deviation-from-mean error}} + \underbrace{KL(f_h^m) - KL(f_h^*)}_{\text{Discriminator induced error}} + \underbrace{KL(f_h^*) - KL(f^*)}_{Bias} \tag{8.6}$$

This equation decomposes total estimation error into three terms: 1) deviation from the mean error, 2) error in KL estimate by the discriminator due to using finite samples in optimization eq.(8.4), and 3) bias when the considered function space,h, does not contain optimal function. We leave quantification of second and third term as future work. Here, we concentrate on quantifying the probability of deviation-from-mean error which is directly related to observed variance of the KL estimate.

Overview of Contributions: Note that the deviation is the difference between a random variable and its mean. Based on this observation, we can bound the probability of this error using concentration inequality and the complexity of function space of f_h^m. This requires overcoming the open challenge of measuring the complexity of neural networks function space. To this end, we propose to construct a function out of neural network such that it lies on RKHS. This is our first contribution (Section 8.5). Then, we proceed to bound the probability of deviation-from-mean error through the covering number of the RKHS space. Lemma 1 and Theorem 2 are our contribution (Section 8.6). Then we provide insight into how the optimization of eq.(8.4) might affect discriminator function space $\mathcal{H}_K$. Using ideas from mean embedding, we prove Lemma 3 and Theorem 3 and provide a geometric insight (Section 8.7). This allows us to present a complete story of how the optimization setup might encourage increase in the complexity of $\mathcal{H}_K$ and how to control it (Section 8.8).

8.5 Constructing f in RKHS

To construct a function in RKHS, we use an operator T related to integral operator L_K by $L_K = TT^*$ [6]. The following theorem due to [5] paves a way for us to construct a neural function in RKHS.

Theorem 3. *[[5] Appendix A] A function $f \in \mathcal{L}_2(d\rho)$ is in Reproducing Kernel Hilbert Space, say $\mathcal{H}_K$ if and only if it can be expressed as*

$$\forall x \in \mathcal{X}, f(x) = \int_{\mathcal{W}} g(w)\psi(x,w)d\tau(w), \tag{8.7}$$

for a certain function $g : \mathcal{W} \to R$ such that $||g||^2_{\mathcal{L}_2(d\tau)} < \infty$. The RKHS norm of f satisfies $||f||^2_{\mathcal{H}_K} \leq ||g||^2_{\mathcal{L}_2(d\tau)}$ and the kernel K is given by

$$K(x,t) = \int_{\mathcal{W}} \psi(x,w)\psi(t,w)d\tau(w) \tag{8.8}$$

Theorem 3 gives us a condition when a square integrable function is guaranteed to lie in RKHS. We simply choose $g(w) = \mathbf{1}$, a constant unit function over the domain $\mathcal{W}$. This means that we can convert a square integrable neural network function $f : \mathcal{X} \to \mathbb{R}$ into a function in RKHS, if we make some weights in the neural network stochastic and average over them. Here, we make the last layer of the neural network to be drawn from Gaussian distribution, whose parameters are learnt during training. More precisely, we consider $\psi(x,w) = \phi_\theta(x)^T w$, where $\phi_\theta(x)$ denotes neural network transformation until the last layer, and w is the last linear layer sampled from Gaussian distribution. While in principle any layer could be made stochastic, we chose this architecture to reduce the computational cost of sampling. The kernel K, as defined in eq.(8.8), can be obtained as:

$$K_\theta(x^*,t^*) = \int_{\mathcal{W}} \phi_\theta(x^*)^T w w^T \phi_\theta(t^*)d\tau(w) = \phi_\theta(x^*)^T(\bar{w}\bar{w}^T + \Sigma)\phi_\theta(t^*) \tag{8.9}$$

where $\bar{w}$ and Σ denote the mean and covariance of w. We sometimes denote the kernel K by K_θ to emphasize that it is a function of neural network parameters, θ.

With this construction, our discriminator function f lies in RKHS denoted by $\mathcal{H}_K$. With $g(w) = \mathbf{1}$, it is easy to verify that $||g||^2_{L_2(d\tau)} = 1$ since w is sampled from a normal distribution. The inequality in Theorem 3 gives us $||f||^2_{\mathcal{H}_K} \leq 1$. It is interesting that the RKHS norm of function f is upper-bounded by 1. Traditionally, kernel K remains fixed and the norm of the function f determines the complexity of the function space. For example, [78] penalized the $||f||_{\mathcal{H}_K}$ as a way to control the function space while

estimating KL divergence. In our RKHS formulation of neural networks, the nature of the problem has changed: $||f||_{\mathcal{H}_K}$ cannot increase beyond 1, but the RKHS itself changes during training since it is determined by the kernel that depends on neural parameters θ. Therefore, the challenge becomes teasing out how neural parameters θ affects the complexity of the discriminator function space and how that affects the deviation-from-mean error in eq.(8.6).

8.6 Bounding the Error Probability of KL Estimates

In this section, we first bound the probability of deviation-from-mean error in terms of the covering number in Lemma 1. We then use an estimate of the covering number of RKHS due to [25] to relate the bound to kernel K_θ in Theorem 4, identifying the role of neural networks in this error bound.

Lemma 1. *Let $f^m_{\mathcal{H}_K}$ be the optimal discriminator function in a RKHS $\mathcal{H}_K$ which is M-bounded. Let $KL_m(f^m_{\mathcal{H}_K}) = \frac{1}{m}\sum_i f^m_{\mathcal{H}_K}(x_i)$ and $KL(f^m_{\mathcal{H}_K}) = E_{p(x)}[f^m_{\mathcal{H}_K}(x)]$ be the estimate of KL divergence from m samples and that by using true distribution $p(x)$ respectively. Then the probability of error at some accuracy level, ϵ is lower-bounded as:*

$$Prob.(|KL_m(f^m_{\mathcal{H}_K}) - KL(f^m_{\mathcal{H}_K})| \leq \epsilon) \geq 1 - 2\mathcal{N}(\mathcal{H}_K, \frac{\epsilon}{4\sqrt{S_K}}) \exp(-\frac{m\epsilon^2}{4M^2})$$

where $\mathcal{N}(\mathcal{H}_K, \eta)$ denotes the covering number of a RKHS space $\mathcal{H}_K$ with disks of radius η, and $S_K = \sup_{x,t} K(x,t)$ which we refer as kernel complexity

Proof. Let $\ell_z(f) = E_{p(x)}[f(x)] - \frac{1}{m}\sum_i f(x_i)$ denotes the error in the estimate such that we want to bound $|\ell_z(f)|$. We have,

$$\ell_z(f_1) - \ell_z(f_2) = E_{p(x)}[f_1(x) - f_2(x)] - \frac{1}{m}\sum_i f_1(x_i) - f_2(x_i)$$

We know $E_{p(x)}[f_1(x) - f_2(x)] \leq ||f_1 - f_2||_\infty$ and $\frac{1}{m}\sum_i f_1(x_i) - f_2(x_i) \leq ||f_1 - f_2||_\infty$. Using the triangle inequality, we obtain $|\ell_z(f_1) - \ell_z(f_2)| \leq 2||f_1 - f_2||_\infty$. Now, consider $f \in \mathcal{H}_K$, then,

$$|f(x)| = |\langle K_x, f\rangle| \leq ||f||||K_x|| = ||f||\sqrt{K(x,x)} \tag{8.10}$$

This implies the RKHS space norm and ℓ_∞ norm of a function are related by

$$||f||_\infty \leq \sqrt{S_K}||f||_{\mathcal{H}_K} \tag{8.11}$$

Hence, we have:

$$|\ell_z(f_1) - \ell_z(f_2)| \leq 2\sqrt{S_K}||f_1 - f_2||_{\mathcal{H}_K} \tag{8.12}$$

The idea of the covering number is to cover the whole RKHS space $\mathcal{H}_K$ with disks of some fixed radius η, which helps us bound the error probability in terms of the number of such disks. Let $\mathcal{N}(\mathcal{H}_K, \eta)$ be such disks covering the whole RKHS space. Then, for any function f in $\mathcal{H}_K$, we can find some disk, D_j with centre f_j, such that $||f - f_j||_{\mathcal{H}_K} \leq \eta$. If we choose $\eta = \frac{\epsilon}{2\sqrt{S_K}}$, then from eq.(8.12), we obtain,

$$\sup_{f \in D_j} |\ell_z(f)| \geq 2\epsilon \implies |\ell_z(f_j)| \geq \epsilon \tag{8.13}$$

Using the Hoeffding's inequality, $\text{Prob.}(|\ell_z(f_j)| \geq \epsilon) \leq 2e^{-\frac{m\epsilon^2}{2M^2}}$ and eq.(8.13),

$$\text{Prob.}(\sup_{f \in D_j} |\ell_z(f)| \geq 2\epsilon) \leq 2e^{-\frac{m\epsilon^2}{2M^2}} \tag{8.14}$$

Applying union bound over all the disks, we obtain,

$$\begin{aligned} \text{Prob.}(\sup_{f \in \mathcal{H}} |\ell_z(f)| \geq 2\epsilon) &\leq 2\mathcal{N}(\mathcal{H}, \frac{\epsilon}{2\sqrt{S_K}})e^{-\frac{m\epsilon^2}{2M^2}} \\ \text{Prob.}(\sup_{f \in \mathcal{H}} |\ell_z(f)| \leq \epsilon) &\geq 1 - 2\mathcal{N}(\mathcal{H}, \frac{\epsilon}{4\sqrt{S_K}})e^{-\frac{m\epsilon^2}{8M^2}} \end{aligned} \tag{8.15}$$

which proves the lemma.

On M-boundedness of $f^m_{\mathcal{H}_K}$

To prove the lemma, we assumed that $f^m_{\mathcal{H}_K}$ is M bounded. To see why this is reasonable, from eq.8.11, we have $||f^m_{\mathcal{H}_K}||_\infty \leq \sqrt{S_K}||f^m_{\mathcal{H}_K}||_{\mathcal{H}_K}$. Since by construction, $||f^m_{\mathcal{H}_K}||_{\mathcal{H}_K} \leq 1$, $f^m_{\mathcal{H}_K}$ is bounded if S_K is bounded, which is true by assumption and seems to hold true in experiments. □

Remark 1. *We derived the error bound based on the Hoeffding's inequality by assuming that our only knowledge about f is that it is bounded. If we have other knowledge, for example, if we know the variance of f, we could use Bernstein's inequality instead of Hoeffding's inequality with minimal change to the proof. To the extent we are interested in the contribution of neural network in error bound, however, there is not much gain by using one inequality or the other. Hence, we stick with Hoeffding's inequality and note other possibilities.*

Remark 2. *Note that in Lemma 1, the radius of disks are inversely related to the the quantity, S_K, meaning that if S_K is high, we would need large number of disks to fill the RKHS space. Hence, it denotes a quantity that reflects the complexity of the RKHS space. We, therefore, term it kernel complexity. Also in eq. 8.11 and the discussion about the M-boundedness, we see that the maximum value $|f(x)|$ depends on S_K, again providing insight into how S_K may control both maximum fluctuation and the boundedness.*

Lemma 1 bounds the probability of error in terms of the covering number of the RKHS space. Note that the radius of the disc is inversely related to S_K which indicates how complex the RKHS space defined by the kernel K_θ is. Here K_θ depends on the neural network parameters θ. Therefore, we denote S_K as a function of θ as $S_K(\theta)$ and term it kernel complexity. Next, we use Lemma 2 due to [25] to obtain an error bound in estimating KL divergence with finite samples in Theorem 4.

Lemma 2 ([25])**.** *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a $\mathcal{C}^\infty$ Mercer kernel and the inclusion $I_K : \mathcal{H}_K \hookrightarrow \mathcal{C}(\mathcal{X})$ is the compact embedding defined by K to the Banach space $\mathcal{C}(\mathcal{X})$. Let B_R be the ball of radius R in RKHS $\mathcal{H}_K$. Then $\forall \eta > 0, R > 0, h > n$, we have*

$$\ln \mathcal{N}(I_K(B_R), \eta) \leq \left(\frac{RC_h}{\eta}\right)^{\frac{2n}{h}} \tag{8.16}$$

where $\mathcal{N}$ gives the covering number of the space $I_K(B_R)$ with discs of radius η, and n represents the dimension of inputs space $\mathcal{X}$. C_h is given by $C_h = C_s\sqrt{||L_s||}$ where L_s is a linear embedding from square integrable space $\mathcal{L}_2(d\rho)$ to the Sobolev space $H^{h/2}$ and C_s is a constant.

To prove Lemma 2, the RKHS space is embedded in the Sobolev Space $H^{h/2}$ using L_s and then the covering number of the Sobolev space is used. Thus the norm of L_s and the degree of Sobolev space, $h/2$, appears in the covering number of a ball in $\mathcal{H}_K$. In Theorem 4, we use this Lemma to bound the estimation error of KL divergence.

Theorem 4. *Let $KL(f^m_{\mathcal{H}_K})$ and $KL_m(f^m_{\mathcal{H}_K})$ be the estimates of KL divergence obtained by using true distribution $p(x)$ and m samples respectively as described in Lemma 1, then the probability of error in the estimation at the error level ϵ is given by:*

$$Prob.(|KL_m(f^m_{\mathcal{H}_K}) - KL(f^m_{\mathcal{H}_K})| \leq \epsilon) \geq 1 - 2\exp\left[\left(\frac{4RC_s\sqrt{S_K(\theta)||L_s||}}{\epsilon}\right)^{\frac{2n}{h}} - \frac{m\epsilon^2}{4M^2}\right]$$

Proof. Lemma 2 gives the covering number of a ball of radius R in a RKHS space. If we consider the hypothesis space to be a ball of radius R in Lemma 1 , we can apply Lemma 2 in it. We fix the radius of discs to $\eta = \frac{\epsilon}{4\sqrt{S_K}}$ in Lemma 1 and substitute $C_h = C_s\sqrt{||L_s||}$ to obtain the required result. □

Theorem 2 shows that the error increases exponentially with the radius of the RKHS space, complexity of the kernel $S_K(\theta)$, and the norm of Sobolev space embedding $||L_s||$. Since we have $||f||_{\mathcal{H}_K} \leq 1$, we can consider our hypothesis space to be a ball of radius 1. To bound $||L_s||$, we need to compute higher order derivatives of $K(x,t)$, which we leave as future work. This allows us to focus on kernel complexity $S_K(\theta)$, which is exponentially related to the probability of deviation-from-mean error.

Note that to bound the deviation-from-mean error, we used union bound and therefore, the bound does not explicitly depend on the function $f^m_{\mathcal{H}_K}$, but only depends on the complexity S_K of the function space $\mathcal{H}_K$. However, the optimization of discriminator (eq.(8.4)) also impacts the complexity $S_K(\theta)$. To understand this effect, in the

next section, we present an upper bound on the objective in eq.(8.4), and give some geometric intuition connecting the optimization objective with the kernel complexity $S_K(\theta)$. Using this intuition, we further argue that the optimization of eq.(8.4) may encourage increment in the complexity, $S_K(\theta)$, thereby increasing the probability of deviation from the mean.

8.7 Mean Embedding Upper Bound

In addition to deriving complexity bound, another advantage of using RKHS is that it allows us to use mean embedding representation. This helps us derive some geometrical insights into the maximization objective in eq.(8.4), on which we give an upper bound in Theorem 5.

Theorem 5. *Let $f \in \mathcal{H}_{K_\theta}$ be a function in RKHS $\mathcal{H}_{K_\theta}$. Then we have the following upper bound on the objective of KL divergence estimation:*

$$\frac{1}{m}\sum_{x_i \sim p(x)} \log \sigma(f(x_i)) + \frac{1}{m}\sum_{x_j \sim q(x)} \log(1-\sigma(f(x_j))) \leq \log \sigma[\langle \mu_p^m - \mu_q^m, f\rangle_{\mathcal{H}_K}] \quad (8.17)$$

and the KL divergence is given by $KL = \langle \mu_p^m, f\rangle$ where μ_p^m and μ_q^m represent mean embedding of m samples from distributions $x_i \sim p(x)$ and $x_j \sim q(x)$ with respect to $\mathcal{H}_K$.

The following Lemma is useful to prove this theorem.

Lemma 3. $E_{p(x)} \log \sigma(f(x)) + E_{q(y)} \log(1-\sigma(f(y))) \leq \log \sigma[E_{p(x)}(f(x)) - E_{q(y)}(f(y))]$

Proof.

$$\begin{aligned}
E_{p(x)} \log \sigma(f(x)) + E_{q(x)} \log(1 - \sigma(f(x))) &= E_{p(x)} \log \sigma(f(x)) + E_{q(x)} \log(\frac{1}{1+\exp(f)}) \\
&= E_{p(x)} \log \sigma(f(x)) + E_{q(x)} \log(\frac{\exp(-f)}{1+\exp(-f)}) \\
&= E_{p(x)} \log \sigma(f(x)) + E_{q(x)} \log \sigma(-f(x)) \\
&\leq \log \sigma[E_{p(x)}(f(x))] + \log \sigma[E_{q(x)}(-f(x))] \\
&\leq \log \sigma[E_{p(x)}(f(x)) - E_{q(x)}(f(x))
\end{aligned}$$

where we used the fact that $\log \sigma$ is a concave function and applied Jensen's inequality in last two lines and linearity of expectation in the last line. □

Proof of Theorem 5. If f lies in the RKHS, then there exists some μ_p^m and μ_q^m such that

$$\frac{1}{m} \sum_{x_i \sim p(x)} f(x_i) = \langle \mu_p^m, f \rangle_{\mathcal{H}_K}, \;\; \frac{1}{m} \sum_{x_j \sim q(x)} f(x) = \langle \mu_q^m, f \rangle_{\mathcal{H}_K} \tag{8.18}$$

Applying eq.(8.18) to the Lemma 3 for finite samples, we obtain required result. □

Geometric Intuition: Theorem 5 tells us that the upper bound (MEBUB) to the objective is $\log \sigma$ of the inner product between $\mu_p^m - \mu_q^m$ and f. The inner product and KL divergence estimates have been depicted geometrically in Fig. 8.1. When the objective is maximized, MEBUB may also increase which leads to an increase in the inner product since $\log \sigma$ is monotonic. When this happens, nothing prevents the midpoint of the mean embeddings, *i.e.*, $\frac{\mu_p^m + \mu_q^m}{2}$, from going away from the origin in Fig. 8.1. In the next section, we show how this affects kernel complexity S_K.

8.8 Fitting Pieces and Complexity Control

Theorem 4 shows that the error bound of the KL estimate is exponentially controlled by the kernel complexity $S_K(\theta) = \sup_{x,t} K_\theta(x,t)$. Since the mean of a vector is upper

bounded by supremum,

$$||\mu_p^m + \mu_q^m||_{\mathcal{H}_K} = \sqrt{\frac{1}{m^2}\sum_{i,j} K_\theta(y_i, y_j) + 2K_\theta(y_i, z_j) + K_\theta(z_i, z_j)} \tag{8.19}$$

$$\leq 2\sqrt{\sup_{x\in\{Y,Z\}, t\in\{Y,Z\}} K_\theta(x,t)} = 2\sqrt{S_K(\theta)} \tag{8.20}$$

As the training progresses in maximizing the objective in eq.(8.4), the algorithm tries to do two things: 1) align f with $\mu_p^m - \mu_q^m$ and 2) increase norms $||\mu_p^m - \mu_q^m||_{\mathcal{H}_K}$ and $||f||_{\mathcal{H}_K}$. For fixed $\langle \mu_p^m, \mu_q^m \rangle_{\mathcal{H}_K} / (||\mu_p^m||.||\mu_q^m||)$, we can show that the ratio $||(\mu_p^m - \mu_q^m)||_{\mathcal{H}_K} / ||\mu_p^m + \mu_q^m||_{\mathcal{H}_K}$ also remains unchanged. Under this assumption, we could say that maximizing eq.(8.4) could lead to increment of $||\mu_p^m + \mu_q^m||_{\mathcal{H}_K}$, and nothing would stop the network from going this way. When this happens, the inequality in eq.(8.20) suggests that $S_K(\theta)$ also increases, thereby increasing the probability of deviation-from-the-mean error in the KL estimate by Theorem 4. In other words, as we train the neural discriminator, the neural network parameters θ change such that the complexity of RKHS might itself keep increasing which causes exponential growth in the sample complexity.

To control the complexity of RKHS space, we can control $S_K(\theta)$ during the training of the neural network. To do this in a scalable way compatible with neural networks, we use gradient descent based optimization. Computation of gradient of $S_K(\theta)$ w.r.t θ is straightforward using definition of $K_\theta(x,t)$ and can be easily realized by using backpropagation. Ideally, $S_K(\theta)$ is $max.K_\theta(x,t)$ over all the data-pairs $(x,t) \in \mathcal{X} \times \mathcal{X}$, which requires passing all the datapoints through neural network. Instead, we simply compute supremum over the minibatch matrix which contains the $2b \times 2b$ entries corresponding to every pair in $2b$ elements (b from each distribution $p(x)$ and $q(x)$). This is obviously a lowerbound – denoted by $S_{mini}(\theta)$ – of $S_K(\theta)$. To penalize the RKHS space that are high in complexity, we add a regularization term with parameter λ to maximize a modified objective:

$$\frac{1}{m}\sum_{x_i\sim p(x_i)} \log\sigma(f(x_i)) + \frac{1}{m}\sum_{x_j\sim q(x_j)} \log(1-\sigma(f(x_j))) - \lambda.S_{mini}^{\gamma} \tag{8.21}$$

where γ is an estimation of $\frac{n}{h}$ and treated as a hyperparameter. Optimization of eq.(8.21) w.r.t. neural network parameters θ allows dynamic control of the complexity

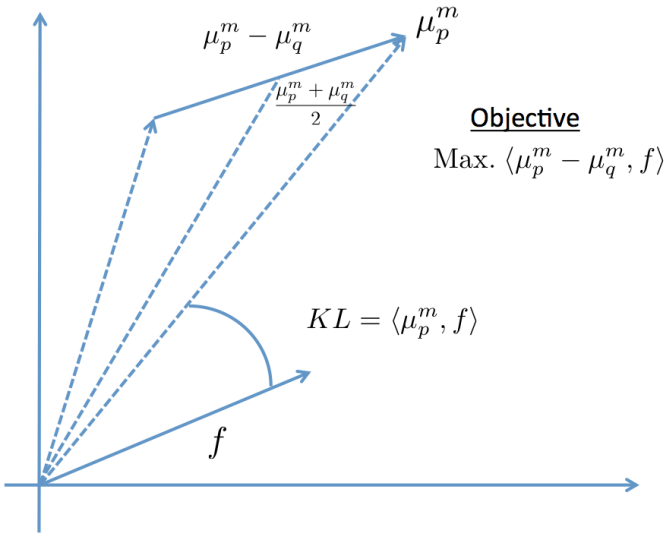


Figure 8.1: Geometrically representing mean embeddingsof two distributions, their relation to maximization objective and KL divergence.

of the discriminator function on the fly in a scalable and efficient way. Complete algorithm is given in Algorithm 2.

8.9 Experimental Results

Experimental Setup: We assume that we have finite sets of samples from two distributions. We further assume that we are required to apply minibatch based optimization. We consider estimating KL divergence in a simple case of two Gaussian distributions in 2D, where we know the analytical KL divergence between the two distributions as the ground truth. We consider three different pairs of distributions corresponding to true KL divergence values of $1.3, 13.8$ and 61.1, respectively and use $m = 5000$ samples from each distribution to estimate KL in the finite case.

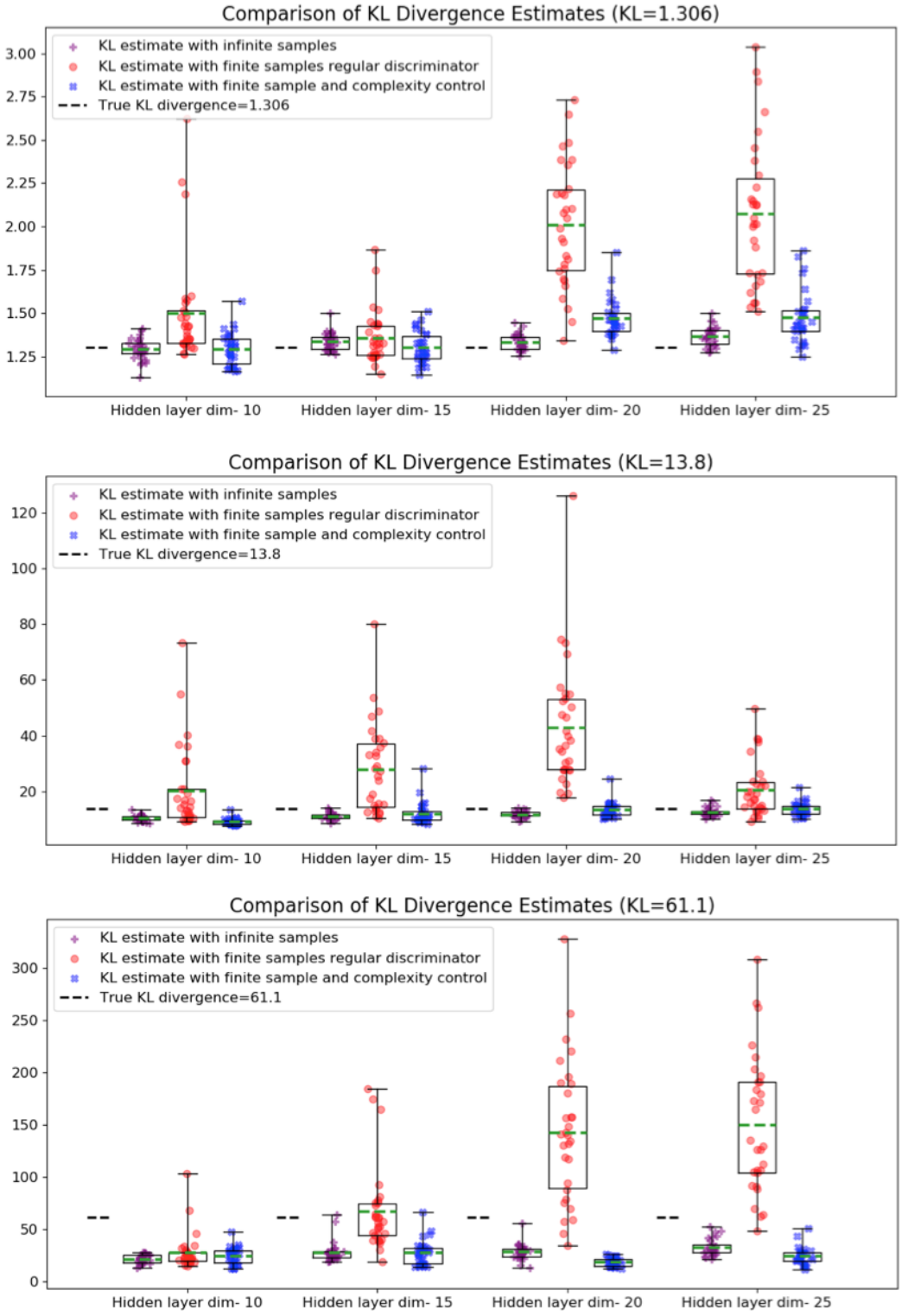


Figure 8.2: Comparison of KL divergence estimates (y-axis) using i) infinite samples (purple), ii) finite samples and a normal neural network discriminator (red), and iii) finite sample and the presented RKHS discriminator with complexity control (blue).

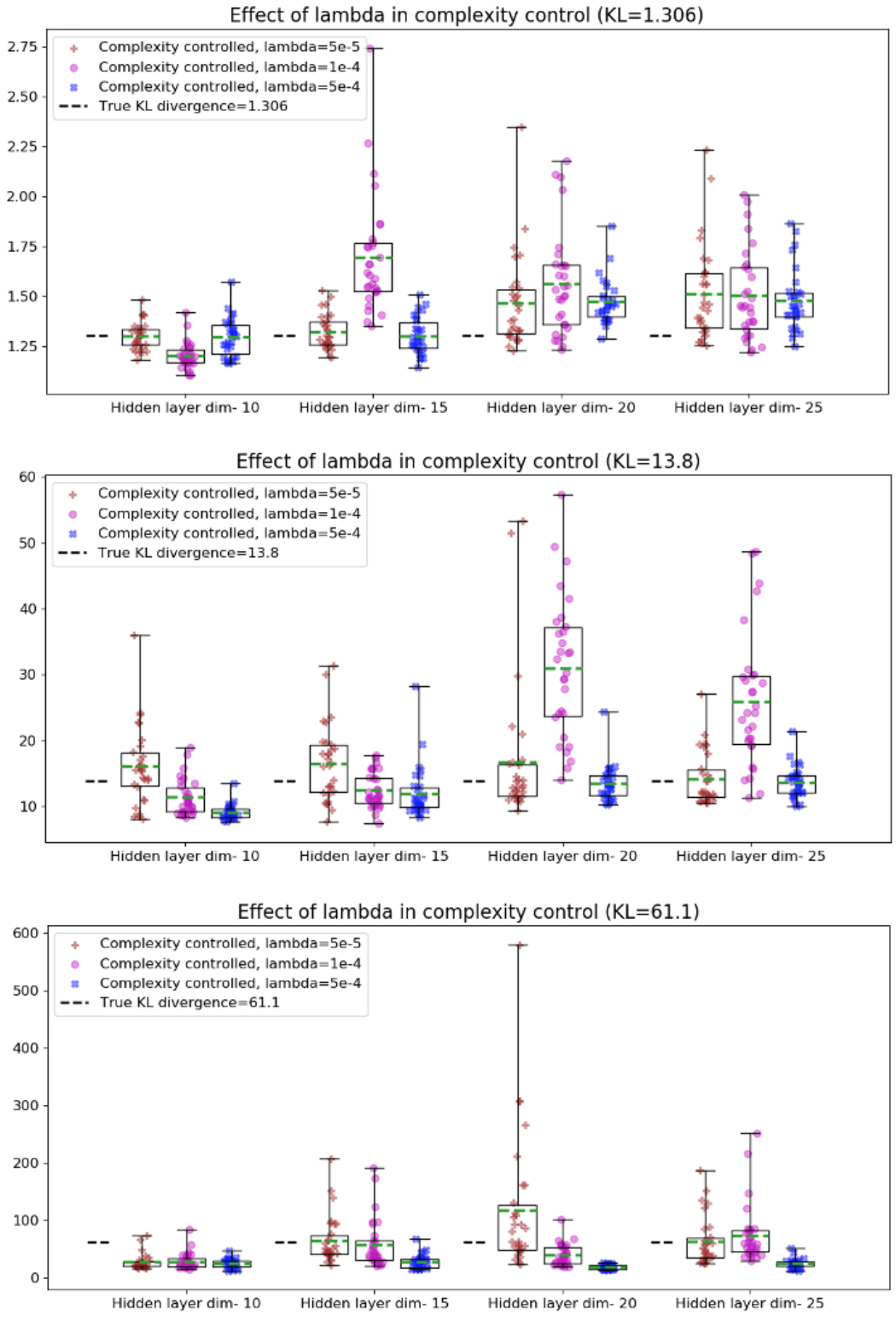


Figure 8.3: The effect of the regularization parameter λ in KL estimates (y-axis) plotted against the varying hidden layer dimension for each KL divergence value.

Algorithm 2 KL divergence estimation with complexity control

Fix minibatch size, b, hyperparameter γ, number of samples m, $flat\_n = 100$, $idx = 0$,$\ell_{min} = \infty$
Initialize the neural network parameters θ, last layer $w \sim \mathcal{N}(\bar{w}, LL^T)$, such that $\bar{w} = 0, LL^T = \mathrm{I}$
for iteration $iter$ in 1 to $iter_{max}$ **do**
    $kl_{sum} = 0, \ell_{adv} = 0$, $n\_batch = (m/b)$
    for iteration k in 1 to n_batch **do**
        Sample minibatch $\{x_i\}_{i=1}^b$ from $p(x)$ and $\{y_i\}_{i=1}^b$ from $q(x)$, and $J = \{\{x_i\}_{i=1}^b, \{y_i\}_{i=1}^b\}$
        For each x_i, y_i, sample $\epsilon \sim \mathcal{N}(0, I)$ and obtain samples $\{w_j\}_{j=1}^d$ where $w_j = \bar{w} + L\epsilon_j$
        $f(x)_i, = \frac{1}{d}\sum_j \phi_\theta(x_i)^T w_j$, $f(y)_i = \frac{1}{d}\sum_j \phi_\theta(y_i)^T w_j$
        $loss_d = -\frac{1}{b}\sum_i \log\sigma(f(x)_i) + \log(1 - f(y)_i)$
        $S_{mini} = \max_{x\in J, t\in J} \phi_\theta(x)^T(\bar{w}\bar{w}^T + \Sigma)\phi_\theta(t)$
        Backpropagate $loss = loss_d + \lambda.S_{mini}^\gamma$ and update $\theta, \bar{w}, L$
        $kl_{sum} = kl_{sum} + \frac{1}{b}\sum_i \log\sigma(f(x)_i)$
        $\ell_{adv} = \ell_{adv} + loss_d$
    end for
    $\ell = \ell_{adv}/n\_batch$, $kl_{iter} = kl_{sum}/n\_batch$
    if $\ell < \ell_{min}$ **then** $kl = kl_{iter}$, $idx = iter$
    else if $iter > idx + flat\_n$ **then return** kl
    end if
end forreturn kl

As the discriminator, we use a fully connected neural network with two hidden layers. The number of hidden units are varied to understand the effect of the discriminator complexity on the fluctuation of the KL estimate. The dimensions are kept identical between the neural-net discriminator and the RKHS discriminator, the latter being different only in that its last layer is stochastic. Full architecture details is provided in Appendix B. We perform random estimation experiment 30 times and report the mean, standard deviation, scatter and box plots.

Finite *v.s.* Infinite Samples: In infinite samples experiment, we assume that we can continuously sample from the model generating data from the two given distributions. The results of KL estimates using infinite samples is shown in Fig. 8.2 and Table.8.1, in comparison with estimates using finite samples without controlling the complexity of the neural-net discriminator. We observe that when we use infinite samples (purple), we obtain an estimate with low variance and values close to the analytical truth in KL = 1.3 and KL = 13.8 and an underestimate when KL = 61.1. In contrast, when we use finite samples without controlling the complexity of the neural-net discriminator (red), the estimates fluctuated heavily confirming our hypothesis: we need to control the complexity of the function when the number of samples is finite, or else the probability of estimation error increases.

Complexity Control: Fig. 8.2 and Table.8.1 compare the estimation of KL divergence with and without controlling the discriminator complexity. With discriminator complexity penalized (blue in Fig. 8.2) in eq. (8.21), the KL estimates are much more reliable (low variance) and closer to the estimates from infinite samples. Note in Fig.8.2 that, without complexity penalization, the erratic behavior of the KL estimator worsens as the number of hidden layers increases in the discriminator. This is consistent with our theory because increasing the number of hidden layers increases the complexity of the discriminator. This highlights the need of higher degree penalization of the discriminator complexity as a neural network with increased capacity is used to estimate higher values of KL divergence.

Effect of Regularization Parameter: Table 8.2 and Fig. 8.3 show the effect of the regularization parameter λ that tunes the level of complexity control in eq. (8.21). Both in the table and the figure, the fluctuation in estimates decreases as we increase the value of λ. Fig. 8.3 shows the results of these experiments varying the latent dimension and shows that the pattern is consistent in different in all cases. Furthermore, as can be seen in Fig. 8.3, for a discriminator with low complexity (*e.g.*, latent dimension = 10), a smaller value of λ is sufficient to yield low-variance estimate. As the size of the hidden layer increases, we need to penalize the complexity aggressively with a higher value of λ in order to obtain the same level of consistency. This further supports our

Table 8.1: Comparison of KL-divergence estimates using different methods; hidden layer dimension = 25.

Method	**True KL**		
	1.3	13.8	61.1
MINE	Unstable	Unstable	Unstable
VDM	Unstable	Unstable	Unstable
Infinite sample	1.36 ± 0.05	12.58 ± 1.49	32.4 ± 7.87
NN Disc	2.07 ± 0.42	20.63 ± 9.82	149.9 ± 65
Complexity control	1.47 ± 0.15	13.64 ± 2.39	24.04 ± 8.2

Table 8.2: The effect of the regularization parameter $\boldsymbol{\lambda}$; hidden layer dimension = 20

$\boldsymbol{\lambda}$	**True KL**		
	1.3	13.8	61.1
5e-5	1.46 ± 0.22	16.65 ± 10.4	116.7 ± 116
1e-4	1.56 ± 0.25	30.97 ± 10.5	39.17 ± 18.5
5e-4	1.47 ± 0.11	13.44 ± 2.68	18.36 ± 3.9

theory.

Underestimation for High KL Divergence We observe in Fig.8.2 and Table. 8.1 that, for KL = 61.1, results from both infinite samples and finite samples with complexity control give underestimated KL divergences even though they reduce fluctuation significantly. This is not surprising since we were focusing on deviation-from-mean error. The total estimation error consists of two additional errors: discriminator induced error and the bias (see eq.8.6). For the small KL divergence, simply controlling complexity was sufficient to minimize all the errors, but for higher value, it is no longer sufficient. The underestimation might be either because of the bias or error induced by

incorrect discriminator function. High bias might be caused if we control the function space too much such that the optimum discriminator $f^*_{\mathcal{H}_K}$ is not close to true discriminator function, f^* (see bias-variance trade-off [11, 25]). It would be an interesting future direction to quantify all three error terms in eq.(8.6).

8.10 Conclusions & Discussion

We have shown that using a regular neural network as a discriminator in estimating KL divergence results in unreliable estimation if the complexity of the function space is not controlled. We then showed a solution by penalizing the kernel complexity in a scalable way using neural networks.

The idea of constructing a neural-net function in RKHS and complexity control could also be useful in stabilizing GANs, or potentially in improving generalization of neural networks. Several papers have identified issues with the stability of GANs [61, 74, 101]. One common understanding is that, in its raw form, we do not enforce the discriminator function to be smooth or regular around the neighborhood of its inputs. Currently, the most successful way to stabilize GANs is to enforce smoothness by gradient penalization. Even in variations like Wasserstein GAN [4, 46] and MMD GAN [12], gradient penalty is crucial to achieve stable results. On the light of the present analysis, we believe that the gradient penalty can be thought as one way to control the complexity of the discriminator. The objective and nature of optimization is such that the complexity of discriminator is bound to increase and therefore some way of decreasing complexity is a must. Similarly, generalization of neural network classifiers and regressors could be improved with complexity control.

8.11 Summary and Answer to Research Questions

Q.3. b) How can we understand and quantify the role of smoothness and regularity properties of the neural network regarding generalization?

In the last chapter, we showed that smoothness helps in improving genearlization. But, when using neural networks, how to precisely define and quantify the degreee of smoothness and its role in generalization? Towards answering this question, this chapter investigates a functional analysis based approach and proposes the complexity of function space as a better measure of whether a neural network behaves well and whether it generalizes well. However, this chapter only establishes the importance of complexity control in stabilizing the neural network. Our conjecture is that this complexity measure lies at the core of both the stability and generalization of neural network. However, this chapter does not provide sufficient evidence for the latter.

Chapter 9

Conclusion and Future work

If you can approach the world's complexities, both its glories and its horrors, with an attitude of humble curiosity, acknowledging that however deeply you have seen, you have only scratched the surface, you will find worlds within worlds, beauties you could not heretofore imagine, and your own mundane preoccupations will shrink to proper size, not all that important in the greater scheme of things.

- Daniel C. Dennett

In this dissertation, we were interested in solving the inverse problem of electrophysiological imaging. Framing it as a learning problem helped us analyze it, understand the challenges and apply theories from Bayesian inference and learning theory to solve it under different settings. In chapter 1, we presented a unifying perspective, which distilled the problem to learning a mapping or a conditional distribution between two domains: measurement and signal. Using the forward model and prior distribution of the electrophysiological signal led us to Bayesian inference in the PGM framework. On the other hand, using samples from the joint distribution allowed us to use learning theory. In each of the methods, we focused on improving generalization of the learned inverse function.

Below we summarize the contributions of this dissertation and room for improvement

in the future:

1. In chapter 4, we modeled the error that might be introduced in the prior (dynamic) model as a random variable in a PGM framework. We introduced ideas like variational lower bound of the generalized Gaussian distribution in modeling sparsity of the error random variable and combined variational Bayes with expectation maximization during inference. This method achieved good improvement over previous method that was unable to model error in the dynamic model. However, it did not model the error in the forward model. We speculate that the error in the forward model could have some role in the algorithm's poor performance in real data experiments. It may, therefore, be helpful to model that error and if possible adapt it according to the data. This could be a possible research direction in the future. Another direction is to incorporate additional knowledge like that of geometry in the PGM framework. This may help in estimating the error in forward model with the help of geometric parameterization of the forward model. Geometric parameterization may also help in adapting the error of the forward model.

2. In chapter 5, we proposed to learn a generative model of the TMP to use as a prior model in a PGM framework. A variational autoencoder learnt a generative model from simulation examples of the cardiac TMP signals. Using the latent variable prior distribution, we performed the Bayesian inference by applying Expectation Maximization and gradient descent. One limitation of this method was that it required personalized simulation and learning of prior distribution; also, inference could only be achieved in same heart-torso geometry. One interesting future direction would be to learn a geometry dependent prior distribution so that it could help generalize knowledge to new geometry in the future. At the same time, the generative model could benefit from multiple signal pairs from different heart-torso geometry.

3. In chapter 6, we took a purely data based approach where a neural network learned an inverse solution from samples of joint distribution. We propose two ideas to improve generalization: 1) Learning invariant representations, 2) Learn-

ing smooth conditional functions. Learning invariant representation is based on the out of distribution (OOD) assumption i.e. we may get test examples that are different from the samples that were used in the training. Such discrepancy may be caused by a nuisance factor or may be a part of the data distribution. When shift in the distribution is caused by nuisance factor, like geometric factor in one of the experiments, we showed that using information bottleneck principle to distill only the useful information helps in robustness against nuisance factors and improves generalization. Another strategy was to improve smoothness of the decoder network to improve generalization. We supported this idea based on analytical learning theory and showed that simple stochasticity in the latent space helps to learn smooth decoder functions (functions with low variation), which in turn helps generalization due to analytical learning theory.

However, there is yet to formally define what smoothness means and how to quantify its degree. Quantifying smoothness and quantitatively relating it with generalization might be a good future direction work. Similarly, formalizing the notion of invariance to improve generalization could be a good direction to pursue. One promising work in this direction is that of Invariance Risk Minimization (IRM) [3]. Building upon IRM could be an interesting research direction.

4. Bayesian methods use the prior knowledge and the Bayesian inference strategies but are computationally expensive and slow at the test time because of the need to perform Bayesian inference for every test example. Direct learning from samples using deep networks is fast but does not incorporate the prior knowledge or utilizes the Bayesian inference. May be there is a way to combine best of the both worlds which makes Bayesian inference faster and/or explicitly incorporates forward model while learning from samples. Exploring this combination could be a good future direction work.

5. In chapter 8, we move a little bit towards the general question of how to quantify the connection between the smoothness of functions and generalization. Towards this end, we started with the hypothesis that the function complexity is an encompassing notion that affects both the stability and generalization of neural

networks. While we demonstrated how complexity plays role in stability in chapter 8, we have not presented compelling argument regarding generalization. To test the hypothesis regarding generalization, especially in the context of inverse problem, we need to extend our setup for a multi variable function. The formulation of a function in RKHS was much simpler as a single variable function. It is not straightforward how to generalize this construction for the multi variate function. That could be a good future direction of research.

9.1 Broader Future Directions

We started with a specific question firmly grounded on solving the inverse problem of electrophysiological imaging. Later, it enabled us to ask fundamental question about learning and generalization. Perhaps, it is the nature of research. We answered specific questions about learning the inverse function and general questions about learning and generalization by the end of dissertation. Subsequently, the thoughts, discussion and rumination during the course of research described in this dissertation sparked off general research directions as summarized below.

Generalization by incorporating geometry

In the problem of electrophysiological imaging, our ultimate goal is to be able to perform imaging in a new real patient. Therefore, we would like to learn a function that can generalize to a new heart-torso geometry. Without incorporating geometric information, being able to perform accurate inference in a new geometry would require training in a lot of examples (when learning from samples) or learning an accurate prior (in case of inference in PGM models). A much better way is to incorporate learning conditioned on the geometry and other available personalized parameters. Conditioning on geometry and personalized parameters would help in much better generalization to new patients/cases.

Generalization via transfer learning and domain adaptation

Our ultimate goal is to perform the inverse electrophysiological imaging in the real patients. To apply data driven methods like machine learning would require large amount of real data, which is not currently available. To circumvent this problem, we have been working with the simulated data. However, training on simulation and transferring to the real world has not yielded impressive results even though the inverse imaging works fine in the simulation data. To resolve this issue, we need to transfer learning from simulated to real data. Hence, we need to understand transfer learning at a fundamental level to resolve the question of how to apply it in this context. Another related approach is to adapt the learning to the real data based on a few available real data. This falls under the domain adaptation regime where the target domain is the real domain and simulation is the source domain.

Generalization via meta learning

To be able to generalize the knowledge (of inverse function) learnt from a few real data (from a few patients) or simulation data to yet unseen patient who may arrive in the future, we may need to use ideas from meta learning. Previously suggested ideas incorporating geometry, transfer learning and domain adaptation are still useful, but may be wanting when it comes to the real patient that was never seen before. In meta learning, we expect that the new case will be different. The goal of meta learning is to devise a way to handle novel difference by keeping track of changes. The idea is to learn pattern of how things are changing in the current set of data and predict how they might change in the future. Meta learning may prove to be very useful in the context of out of distribution generalization.

Connection between complexity, generalization and stability of GAN

Several research works in generalization and adversarial robustness have pointed out the importance of the smoothness of the neural functions in the generalization and adversarial robustness. In the last chapter, we conjectured that the notion of complexity is yet another connecting thread between generalization and adversarial robustness. We provided evidence that the complexity control stabilizes adversarial training. However, this was a very specific setting - stability of adversarial training to estimate KL divergence from samples of two distributions. We also argued that the training was such that complexity was bound to increase. In other scenarios of GAN training, does similar phenomenon hold? Does controlling complexity also help generalization as we conjectured? These questions are important unanswered questions. Moreover, complexity was presented as a quantitative measure of something related to smoothness that affects generalization. If complexity is not a good indicator of generalization of deep networks, we have to ask again how to quantify the measure of smoothness and its effect on generalization.

OOD generalization versus uncertainty quantification

Out of distribution set of data is a huge set, it contains everything that was not used during training. Obviously, any learning algorithm would not be able to generalize to all the data outside of the training distribution, but we may be able to generalize to some data. For the data for which generalization is not possible, may be it is possible to flag the prediction by observing the uncertainty. In summary, we should consider out of distribution generalization and uncertainty quantification under the same framework. Currently, however, there exists two lines of research one trying to improve OOD generalization and other trying to quantify uncertainty when test data lies outside distribution. It is also possible that the ideas in two direction are incompatible or competing with each other.

It is important to develop a unifying theory which considers both generalization and

uncertainty quantification in a common framework, and provides a mechanism to differentiate different types of OOD test data. One promising research direction on OOD generalization is invariant risk minimization (IRM) [3]; it would be interesting to investigate how probabilistic reasoning and uncertainty quantification could be incorporated into this framework to develop a unified theory.

Bibliography

[1] Alex Alemi, Ian Fischer, Josh Dillon, and Kevin Murphy. Deep variational information bottleneck. In ICLR, 2017.

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

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. arXiv preprint arXiv:1907.02893, 2019.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In International Conference on Machine Learning, pages 214–223, 2017.

[5] Francis Bach. Breaking the curse of dimensionality with convex neural networks. The Journal of Machine Learning Research, 18(1):629–681, 2017.

[6] Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. The Journal of Machine Learning Research, 18(1):714–751, 2017.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[8] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In Advances in Neural Information Processing Systems, pages 6240–6249, 2017.

[9] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In International Conference on Machine Learning, pages 531–540, 2018.

[10] A. Berlinet and C. Thomas-Agnan. Reproducing Kernel Hilbert Spaces in Probability and Statistics. Springer US, 2011.

[11] Christopher M Bishop. Pattern recognition and machine learning. springer, 2006.

[12] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In International Conference on Learning Representations, 2018.

[13] David A Bluemke, Linda Moy, Miriam A Bredella, Birgit B Ertl-Wagner, Kathryn J Fowler, Vicky J Goh, Elkan F Halpern, Christopher P Hess, Mark L Schiebler, and Clifford R Weiss. Assessing radiology research on artificial intelligence: A brief guide for authors, reviewers, and readersâfrom the radiology editorial board, 2020.

[14] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In Summer School on Machine Learning, pages 169–207. Springer, 2003.

[15] Olivier Bousquet and André Elisseeff. Stability and generalization. Journal of machine learning research, 2(Mar):499–526, 2002.

[16] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349, 2015.

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

[18] Martin Burger, Kent-André Mardal, and Bjørn Fredrik Nielsen. Stability analysis of the inverse transmembrane potential problem in electrocardiography. Inverse Problems, 26(10):105012, 2010.

[19] Rick Chartrand and Wotao Yin. Iteratively reweighted algorithms for compressive sensing. In 2008 IEEE ICASSP, pages 3869–3872. IEEE, 2008.

[20] Cheng Chen, Qi Dou, Hao Chen, and Pheng-Ann Heng. Semantic-aware generative adversarial nets for unsupervised domain adaptation in chest x-ray segmentation. In International Workshop on Machine Learning in Medical Imaging, pages 143–151. Springer, 2018.

[21] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In Advances in Neural Information Processing Systems, pages 2610–2620, 2018.

[22] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in neural information processing systems, pages 2172–2180, 2016.

[23] Anni Coden, Daniel Gruhl, Neal Lewis, Michael Tanenblatt, and Joe Terdiman. Spot the drug! An unsupervised pattern matching method to extract drug names from very large clinical corpora. In 2012 IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology, pages 33–39. IEEE, 2012.

[24] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In International Conference on Machine Learning, pages 1310–1320, 2019.

[25] Felipe Cucker and Steve Smale. On the mathematical foundations of learning. Bulletin of the American mathematical society, 39(1):1–49, 2002.

[26] Phillip S Cuculich, Yong Wang, Bruce D Lindsay, Mitchell N Faddis, Richard B Schuessler, Ralph J Damiano Jr, Li Li, and Yoram Rudy. Noninvasive characterization of epicardial activation in humans with diverse atrial fibrillation patterns. Circulation, 122(14):1364–1372, 2010.

[27] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. CPAM, 63(1):1–38, 2010.

[28] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1):1–22, 1977.

[29] Burak Erem, Jaume Coll-Font, Ramon Martinez Orellana, Petr St'ovicek, and Dana H 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(3):726–738, 2014.

[30] Burak Erem, P van Dam, and D Brooks. Identifying model inaccuracies and solution uncertainties in non-invasive activation-based imaging of cardiac excitation using convex relaxation. IEEE TMI, (99), 2014.

[31] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37, pages 1180–1189. JMLR. org, 2015.

[32] 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 MICCAI, pages 508–516. Springer, 2018.

[33] 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 MICCAI, pages 508–516. Springer, 2018.

[34] Sandesh Ghimire, Prashnna K Gyawali, and Linwei Wang. Reliable estimation of kullback-leibler divergence by controlling discriminator complexity in the reproducing kernel hilbert space. arXiv preprint arXiv:2002.11187, 2020.

[35] 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.

[36] Sandesh Ghimire, Satyananda Kashyap, Joy T Wu, Alexandros Karargyris, and Mehdi Moradi. Learning invariant feature representation to improve generalization across chest x-ray datasets. arXiv preprint arXiv:2008.04152, 2020.

[37] 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.

[38] Sandesh Ghimire, John L Sapp, Milan Horacek, and Linwei Wang. A variational approach to sparse model error estimation in cardiac electrophysiological imaging. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 745–753. Springer, 2017.

[39] Sandesh Ghimire and Linwei Wang. Deep generative model and analysis of cardiac transmembrane potential. In 2018 Computing in Cardiology Conference (CinC), volume 45, pages 1–4. IEEE, 2018.

[40] Alireza Ghodrati, Dana H Brooks, Gilead Tadmor, and Robert S MacLeod. Wavefront-based models for inverse electrocardiography. IEEE transactions on biomedical engineering, 53(9):1821–1831, 2006.

[41] 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.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. 2016.

[43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[44] Fred Greensite and Geertjan Huiskamp. An improved method for estimating epicardial potentials from the body surface. IEEE TBME, 45(1):98–104, 1998.

[45] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. Journal of Machine Learning Research, 13(Mar):723–773, 2012.

[46] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5767–5777. Curran Associates, Inc., 2017.

[47] Yo Seob Han, Jaejun Yoo, and Jong Chul Ye. Deep residual learning for compressed sensing ct reconstruction via persistent homology analysis. arXiv preprint arXiv:1611.06391, 2016.

[48] Godfrey H. Hardy. On double fourier series and especially those which represent the double zeta-function with real and incommensurable parameters. Quart. J. Math, 37(5), 1906.

[49] 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.

[50] Bin He and Dongsheng Wu. Imaging and visualization of 3-d cardiac electric activity. IEEE Transactions on Information Technology in Biomedicine, 5(3):181–186, 2001.

[51] K He, X Zhang, S Ren, and J Sun. Deep residual learning for image recognition. computer vision and pattern recognition (cvpr). In 2016 IEEE Conference on, volume 5, page 6, 2015.

[52] Geertjan Huiskamp and Fred Greensite. A new method for myocardial activation imaging. IEEE Transactions on Biomedical Engineering, 44(6):433–446, 1997.

[53] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn L. Ball, Katie S. Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. CoRR, abs/1901.07031, 2019.

[54] Alistair E W Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. MIMIC-CXR: A large publicly available database of labeled chest radiographs. arXiv:1901.07042 [cs.CV], 2019.

[55] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. Scientific Data, 3:160035, 2016.

[56] Kenji Kawaguchi and Yoshua Bengio. Generalization in machine learning via analytical learning theory. arXiv preprint arXiv:1802.07426, 2018.

[57] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. ICLR, 2015.

[58] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. ICLR, 2013.

[59] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In Advances in neural information processing systems, pages 4743–4751, 2016.

[60] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. CoRR, abs/1706.02515, 2017.

[61] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. arXiv preprint arXiv:1705.07215, 2017.

[62] W. Jonathan Lederer. Medical physiology a cellular and molecular approach, updated 2nd ed. `https://doctorlib.info/physiology/medical-physiology-molecular/22.html`.

[63] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In Advances in Neural Information Processing Systems, pages 2203–2213, 2017.

[64] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In International Conference on Machine Learning, pages 1718–1727, 2015.

[65] Zhongming Liu, Chenguang Liu, and Bin He. Noninvasive reconstruction of three-dimensional ventricular activation sequence from the inverse solution of distributed equivalent current density. IEEE transactions on medical imaging, 25(10):1307–1318, 2006.

[66] Edward Loper and Steven Bird. NLTK: the natural language toolkit. arXiv:cs/0205028 [cs.CL], 2002.

[67] 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.

[68] Adam C Luchies and Brett C Byram. Deep neural networks for ultrasound beamforming. IEEE transactions on medical imaging, 37(9):2010–2021, 2018.

[69] David JC MacKay. Bayesian interpolation. Neural computation, 4(3):415–447, 1992.

[70] Robert S MacLeod, Martin Gardner, Robert M Miller, and B MILAN HORÁCUEK. Application of an electrocardiographic inverse solution to localize ischemia during coronary angioplasty. Journal of cardiovascular electrophysiology, 6(1):2–18, 1995.

[71] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In Advances in neural information processing systems, pages 2802–2810, 2016.

[72] David A McAllester. Some pac-bayesian theorems. Machine Learning, 37(3):355–363, 1999.

[73] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg C Corrado, Ara Darzi, et al. International evaluation of an ai system for breast cancer screening. Nature, 577(7788):89–94, 2020.

[74] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In International Conference on Machine Learning, pages 3481–3490, 2018.

[75] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 2391–2400. JMLR. org, 2017.

[76] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In International Conference on Machine Learning (ICML), 2017.

[77] Jukka Nenonen, B Milan Horáček, et al. Simulated epicardial potential maps during paced activation reflect myocardial fibrous structure. Annals of biomedical engineering, 26(6):1022–1035, 1998.

[78] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. IEEE Transactions on Information Theory, 56(11):5847–5861, 2010.

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

[80] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In Advances in neural information processing systems, pages 271–279, 2016.

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

[82] American Heart Association Writing Group on Myocardial Segmentation, Registration for Cardiac Imaging:, Manuel D Cerqueira, Neil J Weissman, Vasken Dilsizian, Alice K Jacobs, Sanjiv Kaul, Warren K Laskey, Dudley J Pennell, John A Rumberger, Thomas Ryan, et al. Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart: a statement for healthcare professionals from the cardiac imaging committee of the council on clinical cardiology of the american heart association. Circulation, 105(4):539–542, 2002.

[83] 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.

[84] Judea Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible inference. 1988.

[85] R Plonsey. Bioelectric phenomena. 1969.

[86] 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(10):817–836, 2001.

[87] Azar Rahimi, John Sapp, Jingjia Xu, Peter Bajorski, Milan Horacek, and Linwei Wang. Examining the impact of prior models in transmural electrophysiological imaging: A hierarchical multiple-model bayesian approach. IEEE TMI, 35(1):229–243, 2016.

[88] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225, 2017.

[89] 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, 2004.

[90] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. International Conference in Machine Learning, 2015.

[91] Y Rudy and HS Oster. The electrocardiographic inverse problem. Critical reviews in biomedical engineering, 20(1-2):25–45, 1992.

[92] John L Sapp, Fady Dawoud, John C Clements, and B Milan Horáček. Inverse solution mapping of epicardial potentials: quantitative comparison to epicardial contact mapping. Circulation: Arrhythmia and Electrophysiology, pages CIRCEP–111, 2012.

[93] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, pages 618–626, 2017.

[94] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In Advances in Neural Information Processing Systems, pages 2110–2118, 2016.

[95] Yesim Serinagaoglu, Dana H Brooks, and Robert S MacLeod. Bayesian solutions and performance analysis in bioelectric inverse problems. IEEE TBME, 52(6):1009–1020, 2005.

[96] Yesim Serinagaoglu, Dana H Brooks, and Robert S MacLeod. Improved performance of bayesian solutions for inverse electrocardiography using multiple information sources. IEEE Transactions on Biomedical Engineering, 53(10):2024–2034, 2006.

[97] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. ICLR, 2017.

[98] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. arXiv preprint arXiv:1910.06222, 2019.

[99] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. Journal of Machine Learning Research, 11(Apr):1517–1561, 2010.

[100] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112, 2014.

[101] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. Improving generalization and stability of generative adversarial networks. In International Conference on Learning Representations, 2019.

[102] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. JMLR, 1(Jun):211–244, 2001.

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

[104] 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.

[105] Vladimir N Vapnik. An overview of statistical learning theory. IEEE transactions on neural networks, 10(5):988–999, 1999.

[106] L. Wang, O. A. Gharbia, B. M. Horacek, S. Nazarian, , and J. L. Sapp. Non-invasive epicardial and endocardial electrocardiographic imaging of scar-related ventricular tachycardia. EP Europace, page under review after major revision, 2018.

[107] Linwei Wang, Fady Dawoud, Sai-Kit Yeung, Pengcheng Shi, KenC L Wong, Huafeng Liu, and Albert C Lardo. Transmural imaging of ventricular action potentials and post-infarction scars in swine hearts. IEEE TMI, 32(4):731–747, 2013.

[108] Linwei Wang, Heye Zhang, Ken CL Wong, Huafeng Liu, and Pengcheng Shi. Electrocardiographic simulation on personalised heart-torso structures using coupled meshfree-bem platform. International Journal of Functional Informatics and Personalised Medicine, 2(2):175–200, 2009.

[109] Linwei Wang, Heye Zhang, Ken CL Wong, Huafeng Liu, and Pengcheng Shi. Physiological-model-constrained noninvasive reconstruction of volumetric myocardial transmembrane potentials. IEEE TBME, 57(2):296–315, 2010.

[110] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In IEEE Conference on Computer Vision and Pattern Recognition, pages 2097–2106, 2017.

[111] David Wipf and Srikantan Nagarajan. A unified bayesian framework for meg/eeg source imaging. NeuroImage, 44(3):947–966, 2009.

[112] David P Wipf, Bhaskar D Rao, and Srikantan Nagarajan. Latent variable bayesian models for promoting sparsity. IEEE Transactions on Information Theory, 57(9):6236–6255, 2011.

[113] Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In Advances in Neural Information Processing Systems, pages 2524–2533, 2017.

[114] 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.

[115] Jingjia Xu, John L Sapp, Azar Rahimi Dehaghani, Fei Gao, and Linwei Wang. Variational bayesian electrophysiological imaging of myocardial infarction. In International Conference on MICCAI, pages 529–537. Springer, 2014.

[116] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. International Conference on Learning and Representation, 2017.

[117] 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 of Chapter 4

A.1 Derivation of Variational Lower Bound

Lemma 4.

$$-|x|^p = \sup_{\gamma>0} \left(-\frac{x^2}{2\gamma} - (\frac{2-p}{2})(\frac{1}{p\gamma})^{\frac{p}{p-2}} \right)$$

Proof. We use Fenchel-Legendre duality for convex functions to prove this theorem. Let's define a variable y as $y = x^2$ and a function f as:

$$f(y) = -|x|^p = -y^{p/2},\ \forall y > 0 \tag{A.1}$$

which is convex function of y in the domain $y > 0$. Hence, Fenchel-Legendre duality is used to obtain

$$f(y) = \sup_{\lambda}(\lambda y - f^*(\lambda)) \tag{A.2}$$

where conjugate function $f^*(\lambda)$ is given by

$$f^*(\lambda) = \sup_{y>0}(\lambda y - f(y)) = \sup_{y>0}(\lambda y + y^{p/2}) \tag{A.3}$$

Supremum in eq.(A.3) is obtained at $\hat{y} = [(-\lambda)\frac{2}{p}]^{\frac{2}{p-2}}$. Substituting $\hat{y}$ back in A.3, we get,

$$f^*(\lambda) = (-\lambda)^{\frac{p}{p-2}}(\frac{2-p}{2})(\frac{2}{p})^{\frac{p}{p-2}} = (-\lambda)^{\frac{p}{p-2}} z(p) \tag{A.4}$$

where, $z(p) = (\frac{2-p}{2})(\frac{2}{p})^{\frac{p}{p-2}}$. Note that y is always positive in its domain, and thus λ is always negative. Substituting value of $f^*(\lambda)$ back to equation A.2,

$$f(y) = \sup_{\lambda<0}(\lambda y - (-\lambda)^{\frac{p}{p-2}} z(p)) \tag{A.5}$$

Putting $\lambda = -1/2\gamma$, we have,

$$\begin{aligned} f(y) &= \sup_{\gamma>0}(-\frac{y}{2\gamma} - (\frac{1}{2\gamma})^{\frac{p}{p-2}} z(p)) \\ &= \sup_{\gamma>0}(-\frac{y}{2\gamma} - \frac{2-p}{2}(\frac{1}{p\gamma})^{\frac{p}{p-2}}) \end{aligned} \tag{A.6}$$

Setting $y = x^2$ completes the proof. □

Lemma 5.

$$\exp(\frac{-|x|^p}{\zeta}) = \sup_{\tau>0} \exp(-\frac{x^2}{2\tau}) \exp(-\frac{2-p}{2}(\frac{1}{p\tau})^{\frac{p}{p-2}} \zeta^{\frac{2}{p-2}})$$

Proof. Multiplying both sides of Lemma 1 by $1/\zeta$ yields

$$\frac{f(y)}{\zeta} = \sup_{\gamma>0}(-\frac{y}{2\gamma\zeta} - \frac{2-p}{2\zeta}(\frac{1}{p\gamma})^{\frac{p}{p-2}}).$$

Setting $\tau = \zeta\gamma$, we have,

$$\begin{aligned} \frac{-|x|^p}{\zeta} &= \sup_{\tau>0}(-\frac{y}{2\tau} - \frac{2-p}{2\zeta}(\frac{\zeta}{p\tau})^{\frac{p}{p-2}}) \\ &= \sup_{\tau>0}(-\frac{y}{2\tau} - \frac{2-p}{2}(\frac{1}{p\tau})^{\frac{p}{p-2}} \zeta^{\frac{2}{p-2}}) \end{aligned} \tag{A.7}$$

Taking exponent and replacing $y = x^2$ completes the proof. □

Proof of Theorem 1

Proof. Using Lemma 2 in $p(\mathbf{x}|\alpha) = \frac{C}{\alpha^N}\exp(\frac{-\sum_i |x_i|^p}{\alpha^p})$ yields

$$\begin{aligned} p(\mathbf{x}|\alpha) &= \sup_{\tau>0} \frac{C}{\alpha^N}\exp\Big(-\sum_i \frac{x_i^2}{2\tau_i} - \sum_i \frac{2-p}{2}(\frac{\alpha^2}{p\tau_i})^{\frac{p}{p-2}}\Big) \\ &= \sup_{\lambda>0} \frac{C}{\alpha^N}\exp\Big(-\frac{\mathbf{x}^T\mathbf{\Lambda}\mathbf{x}}{2} - \frac{2-p}{2}(\frac{\alpha^2}{p})^{\frac{p}{p-2}}\sum_i \lambda_i^{\frac{p}{p-2}}\Big) \end{aligned}$$

where $\lambda_i = 1/\tau_i$ and $\mathbf{\Lambda} = diag(\boldsymbol{\lambda})$ □

A.2 Calculation of λ

$$\begin{aligned} &\frac{\partial}{\partial\lambda_i}\left(E_{q(\mathbf{u},\boldsymbol{\eta})}\left[\log\left(p(\boldsymbol{\eta}|\alpha,\boldsymbol{\lambda})\right)\right]\right) = 0 \\ &\frac{\partial}{\partial\lambda_i}\left(E_{q(\mathbf{u},\boldsymbol{\eta})}\left[\frac{\boldsymbol{\eta}^T\mathbf{D}^T\mathbf{\Lambda}\mathbf{D}\boldsymbol{\eta}}{2} + \frac{2-p}{2}(\frac{\alpha^2}{p})^{\frac{p}{p-2}}\sum_i \lambda_i^{\frac{p}{p-2}}\right]\right) = 0 \\ &tr(E_{q(\boldsymbol{\eta})}[\boldsymbol{\eta}\boldsymbol{\eta}^T]\mathbf{d}_i\mathbf{d}_i^T) = (\frac{\lambda_i\alpha^p}{p})^{\frac{2}{p-2}} \\ &\lambda_i = \frac{p}{\alpha^p}\frac{1}{(tr([\bar{\boldsymbol{\eta}}\bar{\boldsymbol{\eta}}^T + \mathbf{\Sigma}_{\boldsymbol{\eta}}]\mathbf{d}_i\mathbf{d}_i^T))^{\frac{2-p}{2}}} \end{aligned} \tag{A.8}$$

The variational parameter λ depends on another parameter α. To obtain an optimum value of α, we repeat the same process but take derivative with respect to alpha.

$$\begin{aligned} &\frac{\partial}{\partial\alpha}\left(E_{q(\mathbf{u},\boldsymbol{\eta})}\left[Nlog(\alpha) + \frac{2-p}{2}(\frac{\alpha^2}{p})^{\frac{p}{p-2}}\sum_i \lambda_i^{\frac{p}{p-2}}\right]\right) = 0 \\ &\frac{p}{\alpha^p} = \left(\frac{N}{\sum_i \lambda_i^{\frac{p}{p-2}}}\right)^{\frac{2-p}{2}} \end{aligned} \tag{A.9}$$

Using equation A.9 into A.8, we finally obtain,

$$\lambda_i = \left(\frac{s}{tr([\bar{\boldsymbol{\eta}}\bar{\boldsymbol{\eta}}^T + \mathbf{\Sigma}_{\boldsymbol{\eta}}]\mathbf{d}_i\mathbf{d}_i^T)}\right)^{\frac{2-p}{2}}$$

where

$$s = \frac{N}{\sum_i \lambda_i^{\frac{p}{p-2}}}$$

A.3 Reducing Computational Cost

$$\begin{aligned}\boldsymbol{\Sigma}_u &= (\beta \mathbf{H}^T\mathbf{H} + \boldsymbol{\Sigma_p}^{-1})^{-1} \\ &= \boldsymbol{\Sigma_p} - \boldsymbol{\Sigma_p}\mathbf{H}^T(\mathbf{H}\boldsymbol{\Sigma_p}\mathbf{H}^T + \beta^{-1}\mathbf{I})^{-1}\mathbf{H}\boldsymbol{\Sigma_p} \qquad \text{(A.10)} \\ \bar{\mathbf{u}} &= \boldsymbol{\Sigma}_u(\beta \mathbf{H}^T\mathbf{y} + \boldsymbol{\Sigma_p}^{-1}\mathbf{u}_d) \\ &= \boldsymbol{\Sigma}_p\mathbf{H}^T(\mathbf{H}\boldsymbol{\Sigma_p}\mathbf{H}^T + \beta^{-1}\mathbf{I})^{-1}\mathbf{y} + (\beta\boldsymbol{\Sigma}_p\mathbf{H}^T\mathbf{H} + \mathbf{I})^{-1}\boldsymbol{u}_d \qquad \text{(A.11)}\end{aligned}$$

where $\boldsymbol{\Sigma_p} = \boldsymbol{\Sigma_d} + (\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D})^{-1}$.

Proof. Eq.(A.10) readily follows from Woodbury Inverse Identity. To prove eq.(A.11), we prove each term. For the first term, we multiply both sides of $\boldsymbol{\Sigma}_u^{-1} = \beta\mathbf{H}^T\mathbf{H} + \boldsymbol{\Sigma_p}^{-1}$on the right with $\boldsymbol{\Sigma}_p\mathbf{H}^T$ to obtain eq.(A.12); and multiply it with $\boldsymbol{\Sigma}_u$ on the left and $(\mathbf{H}\boldsymbol{\Sigma_p}\mathbf{H}^T + \beta^{-1}\mathbf{I})^{-1}\mathbf{y}$ on the right to obtain eq.(A.13):

$$\boldsymbol{\Sigma}_u^{-1}\boldsymbol{\Sigma}_p\mathbf{H}^T = \beta\mathbf{H}^T(\mathbf{H}\boldsymbol{\Sigma_p}\mathbf{H}^T + \beta^{-1}\mathbf{I}) \qquad \text{(A.12)}$$

$$\boldsymbol{\Sigma}_p\mathbf{H}^T(\mathbf{H}\boldsymbol{\Sigma_p}\mathbf{H}^T + \beta^{-1}\mathbf{I})^{-1}\mathbf{y} = \beta\boldsymbol{\Sigma}_u\mathbf{H}^T\mathbf{y} \qquad \text{(A.13)}$$

To prove the second term, we start with the definition:

$$\boldsymbol{\Sigma}_u^{-1} = \boldsymbol{\Sigma}_p^{-1}(\beta\boldsymbol{\Sigma}_p\mathbf{H}^T\mathbf{H} + \mathbf{I}) \qquad \text{(A.14)}$$

and multiply it with $\boldsymbol{\Sigma}_u$ on the left and $\boldsymbol{\Sigma}_p{}^{-1}\mathbf{u}_d$ on the right:

$$(\beta\boldsymbol{\Sigma}_p\mathbf{H}^T\mathbf{H} + \mathbf{I})^{-1}\mathbf{u}_d = \boldsymbol{\Sigma}_u\boldsymbol{\Sigma}_p{}^{-1}\mathbf{u}_d \qquad \text{(A.15)}$$

□

A.4 Proof of Result 1

Proof. $\mathbf{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}} = \beta^{-1}\mathbf{I} + \mathbf{H}\mathbf{A}^{-1}\mathbf{H}^T$ where, $\mathbf{A} = (\mathbf{D}^T\mathbf{\Lambda}\mathbf{D})$
Using Woodbury identity,

$$\begin{aligned}
\mathbf{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}}^{-1} &= \beta\mathbf{I} - \mathbf{H}(\mathbf{A} + \mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T \\
&= \beta\mathbf{I} - \boldsymbol{U}\mathbf{S}(\beta^{-1}\boldsymbol{V}^T\mathbf{A}\boldsymbol{V} + \mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\boldsymbol{U}^T \\
\therefore \mathbf{y}_{\boldsymbol{\eta}}^T\mathbf{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}}^{-1}\mathbf{y}_{\boldsymbol{\eta}} &= \beta\mathbf{y}_{\boldsymbol{\eta}}^T\mathbf{y}_{\boldsymbol{\eta}} - tr(\boldsymbol{z}\boldsymbol{z}^T\mathbf{S}(\beta^{-1}\boldsymbol{V}^T\mathbf{A}\boldsymbol{V} + \mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T) \\
&= \beta\mathbf{y}_{\boldsymbol{\eta}}^T\mathbf{y}_{\boldsymbol{\eta}} - \left\langle \boldsymbol{z}\boldsymbol{z}^T, \mathbf{S}(\beta^{-1}\boldsymbol{V}^T\mathbf{A}\boldsymbol{V} + \mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T \right\rangle
\end{aligned}$$

where $\mathbf{H} = \boldsymbol{U}\mathbf{S}\boldsymbol{V}^T$ is the singular value decomposition and $\boldsymbol{z} = \boldsymbol{U}^T\mathbf{y}_{\eta}$. Finally, replacing $\mathbf{A} = (\mathbf{D}^T\mathbf{\Lambda}\mathbf{D})$, we obtain,

$$\mathbf{y}_{\boldsymbol{\eta}}^T\mathbf{\Sigma}_{\mathbf{y}_{\boldsymbol{\eta}}}^{-1}\mathbf{y}_{\boldsymbol{\eta}} = \beta\mathbf{y}_{\boldsymbol{\eta}}^T\mathbf{y}_{\boldsymbol{\eta}} - \left\langle \boldsymbol{z}\boldsymbol{z}^T, \mathbf{S}(\beta^{-1}\boldsymbol{V}^T\mathbf{D}^T\mathbf{\Lambda}\mathbf{D}\boldsymbol{V} + \mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T \right\rangle$$

□

Appendix B

Appendix of Chapter 8

B.1 Details of experimental setup

Neural RKHS discriminator architecture (Proposed method)

```
Fully connected
Leaky ReLU
Fully connected
Leaky ReLU
```

For the RKHS discriminator, this gives $\phi_\theta(x)$ for the input data x. Then, $f(x)$ needs to defined as in line 8 of Algorithm 1. Similarly, total loss with complexity penalization is computed as line 11 in Algorithm 1.

Neural network discriminator architecture

```
Fully connected
Leaky ReLU
Fully connected
Leaky ReLU
Fully connected
```

For the Neural net discriminator, this would directly give $f(x)$ for the input data x. Also, loss would be defined by line 9, no penalization as in line 11 of Algorithm 1.

Learning rate: 5×10^{-3}
γ : 0.05
No. of samples from each distribution: 5000
Minibatch size: 50
Hyperparameter selection: The hyperparameters like learning rate and γ were selected by first estimating KL divergence at a mid value like 13. Then, same value was used in all experiments.