TOWARDS ROBUST AND GENERALIZABLE MACHINE LEARNING FOR REAL-WORLD HEALTHCARE DATA WITH HETEROGENEITY

A Dissertation

by

ZEPENG HUO

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Bobak J. Mortazavi
Committee Members,	Xiaoning Qian
	Zhangyang Wang
	James Caverlee
Head of Department,	Scott Schaefer

December 2022

Major Subject: Computer Science

Copyright 2022 Zepeng Huo

ABSTRACT

The utility of machine learning for enhancing human well-being and health has risen to the core discussion in both research and real-world application in today's technological front-line. The fast-growing artificial intelligence industry has innovated health-related applications. Inversely the real-world challenges in accurate implementation of mobile and clinical health solutions have necessitated advancements in theoretical and algorithmic development. The increasing rate of digitizing medical records in hospitals has enable artificial intelligence with abundant data to train. In return the trained models have shown to lower the uncertainty in clinical decision making. However, as always, with opportunities comes new challenges. We have observed many discrepancies of compatibility between sophisticated machine learning models and the nuanced clinical needs, such as fairness, personalized treatment through precise phenotyping and data shift in longitudinal medical records. In modeling complex and heterogeneous health record-based machine learning and then extrapolating through remote health applications, I have identified the need for advanced, multimodal models that continually learn risk representation from varied, heterogeneous data sources. Broadly, I recognize a few gaps in different levels currently blocking us from enhancing continual machine learning for health: 1) domain-, 2) class-, and 3) personal-level heterogeneity in real-world healthcare data.

With the three proposed aims, I will target at using machine learning in a more robust and generalizable way towards real-world biomedical data and to enhance not only the prediction accuracy but also the interpretation of the results. The proposed work will largely benefit both machine learning domain as well as human-centered computing application. This dissertation will mostly introduce and elaborate how to bridge the gap between algorithm in the lab and the open-world challenges and hopeful will spur more research onto this interdisciplinary problem and bring about real world improvements.

DEDICATION

To my mother, my father, my grandfather, and my late grandmother and all the family and friends, locally in College Station or globally around the world, who supported me along the journey.

ACKNOWLEDGMENTS

I would like to thank my committee chair, Dr. Bobak Mortazavi, and my committee members, Dr. Xiaoning Qian, Dr. Zhangyang Wang, and Dr. James Caverlee, for their guidance and support throughout the course of this research. My thanks also go to all the help I have received from Computer Science department at Texas A&M, including my friends, labmates, classmates, colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis (or) dissertation committee consisting of Professor Bobak Mortazaiv [advisor] and Dr. Zhangyang Wang, Dr. James Caverlee of the Department of Computer Science and Engineering and Professor Xiaoning Qian of the Department of Electrical and Computer Engineering.

The data in Section 2.1 for in-house RealActivity motion dataset is collected by my labmate Arash Pakbin. The data in Section 3.2 from TOPCAT is processed by my labmate Nathan Hurley. The data in Section 4.2 for interstitial glucose readings is collected by Center for Translational Research in Aging and Longevity at Texas A&M University. The data in Section 3.3 is from electronic health records of patient visits provided by the Yale New Haven Health system.

Funding Sources

Graduate study was partially supported by a Graduate Research Assistantship from Texas A&M University. Other funding sources include Defense Advanced Research Projects Agenc (DARPA FA8750-18-2-0027), National Science Foundation (FY 2017 1648451), National Institutes of Health (R01EB028106-01) and National Institute of Biomedical Imaging and Bioengineering (#1R21EB028486-01).

NOMENCLATURE

AUC- PRC	area under precision-recall curve
AUC-ROC	area under the receiver operating curve
BA	balanced accuracy
BSS	Brier Skill Score
CBS	class-balanced sampling
CGM	Continuous glucose monitors
CKD	chronic kidney disorder
DAE	Denoising Autoencoder
DAH	Density Aware Hinge
ED	Emergency department
EHR	Electronic health records
EM	Expectation-Maximization
GRU	gated recurrent unit
ICU	intensive care units
IMU	inertial measurement units
IOT	Internet of Things
KCCQ	Kansas City Cardiomyopathy Questionnaire
LSTM	long short term memory
MEL	maximum entropy learning
MLE	maximum likelihood estimation
MLP	Multi-layer Perceptron
MoE	Mixture-of-Experts

PNs	Prior Networks
PPGR	Postprandial glycemic response
SGD	stochastic gradient descent
SVM	Support vector machine
t-SNE	t-distributed stochastic neighbor embedding
VAE	Variational Autoencoder
UQ	Uncertainty Quantification

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION AND LITERATURE REVIEW

To view heterogeneity issue in different granularity levels, I plan to introduce it in a top-down fashion. 1) First, There are numerous aspects to domain level heterogeneity that impact model performance. The definition medical domain can be a predefined cohort, type of disease, or data collected from one hospital that carries patient information of similar demographics. The real world medical implication of domain heterogeneity in healthcare data can be: domain shift (label/covariant distribution shift across train and test domains), hidden phenotype (intertwined disease/symptoms among patients), and data density discrepancy (imbalance between common disease and rare disease). If handled properly, the domain level heterogeneity research can shed light on the opportunity on personalized phenotyping, rare disease discovery or domain generalization. 2) Second, middle-scale heterogeneity in healthcare data we observe is class level heterogeneity. The class heterogeneity can lead to class imbalance issue, or fairness issues. For example, one common class/ethnic group may have abundantly more data therefore their information will dominate the learning models, which leads to the bias towards those bigger class or common ethnicity. I found that with more personalized techniques towards class-level heterogeneity handling, we can largely minimize the imbalance effect on the machine learning models for healthcare. The uniqueness of medical data imbalance is that the minority class (for example, mortality in intensive care units) is often more heterogeneous than the majority class (stable patients expected to be discharged). Through fine-grained phenotyping, I find the risk factors that drive high risk more scattered in a feature space, and ultimately guiding personalized risk estimates and thus personalized treatment decision. 3) Third, down to the lowest level of healthcare data, we observe personal level heterogeneity. In this heterogeneity we may observe that user's behaviors in remote health can change over time axis as context shift, or patient's inter-personal variability in EHR can hinder adapting a generic model to a person under transfer learning. If handled properly, the trained model can learn to accurately quantify this heterogeneity and then continually grow in capacity to account for new and unseen contexts, such as emerging disease detection, adverse drug reaction discovery or unknown anomaly detection.

Diving into some real world applications and problems, applying machine learning techniques to biomedical data to either maximize the clinical utility or general daily well-being tracking can be beneficial. But directly applying any off-the-shelf methods onto real-world dataset may render dissatisfaction because there are many aspects of heterogeneity of the data as we mentioned before. For some tangible examples, in intensive care unit (ICU) setting, models producing accurate risk prediction are often confounded by patient heterogeneity, where various patient co-morbidity may result in different driving risk factors for risk scores, lengths of stay within the ICU, and requested laboratory measurements and treatment decisions. This patient heterogeneity may weaken performance, as well as hinder any clinical interpretation of the results if not handled by care. Another example is in activity recognition with wearable sensor data. The problem can arise when the context information is shifted to unknown domain. In wearable computing, context-awareness helps recognize activities based on sensor measurements under different situations. Context can be defined as "any information that can be used to characterize the situation" or to improve recognition. Context-aware systems have previously been used in many applications, including activity recognition, online, personalized and adaptive activity classification, and healthcare applications. The definition of context heavily relies on domain knowledge, such as a user's tasks (e.g., spontaneous activity, engaged tasks) or a user's social environment (e.g., co-location of others, group dynamics), etc. However, in practice, predefined contexts may not always be available, or definitions of contexts may change in different environments. Additionally, new unknown contexts may emerge over time, realized as data heterogeneity. For these reasons, there is a general data insufficiency and lack of contextual information to develop accurate context-aware activity recognition systems that could adapt to these unknown contexts. All these can fall into the umbrella of heterogeneity of the data in the wild, and the thus it requires the machine learning to be able to handle them robustly.

In this dissertation, I propose to use three aims to tackle this problem. 1) First I propose that the framework should be able to capture or quantify the heterogeneity. As the first step, if the model can detect the heterogeneous component in the data, such as shifting of the user behavioral contexts by uncertainty quantification (UQ) or the hidden subtypes/phenotypes of the patients in a medical dataset, the model will have a better chance to adjust or adapt to them in the subsequent modeling. The heterogeneity can be demonstrated in different forms, such as data missingness, data imbalance, or domain shift. A model trained in the lab environment might need to retrain or re-calibrate when it is deployed in the real world environment, where the capturing of the heterogeneity be extremely beneficial as the first step in a pipeline. 2) Second I propose to divide and conquer the heterogeneity. As the term suggests, heterogeneity is a combination of different forms of data distributions under different environments or contexts. Therefore, the most intuitive way to tackle it is to break down into smaller problems and study each problem under a microscope, so to speak. Many ensemble-based learning frameworks have demonstrate their power in learning large and complex data so we are aiming to follow the same line of thought. However, for biomedical data there has to be extra process involved, in that the data is a reflection of a underlying mechanism of a user performing activities or a patient having an onset of a disease. So the model should be able to gear towards learning the underlying mechanism as well, rather than simply overfitting the data to get high accuracy. 3) Third, the model should be able to adapt or tailor to heterogeneity because heterogeneity itself can be dynamically changing over time. This makes the problem extra difficult since the first two aims may work under one particular circumstances but as time progresses, the model needs to manual tuning the parameters or end up not accurate anymore. There is a need to let the model adapt or generalize well once it is deployed in the real world setting.

The real-world heterogeneity in biomedical data can be reflected in many aspects, but without proper implementation and modeling the model might give flawed conclusion or subpar performance. A challenge in developing machine learning models for patient risk prediction involves addressing patient heterogeneity and interpreting the model outcome in clinical settings. Patient heterogeneity manifests as clinical differences among homogeneous patient subtypes in observational datasets. The discovery of such subtypes is helpful in precision medicine, where different risk factors from different patient would contribute differently to disease development and thus personalized treatment. In activity recognition task, the heterogeneity can manifest as user behavioral shifts. The context, which is to describe this user behavior, can be the underlying mechanism that can be informative for modeling. New unknown contexts may emerge over time but it can be agnostic to the model. For these reasons, there is a general data insufficiency and lack of contextual information to develop accurate context-aware activity recognition systems that could adapt to these unknown contexts. While existing works are not adequate to address the challenges such as a lack of context information in data while training the model, or the emergence of unknown contexts when the model is applied, we propose a data-driven approach with the capability to handle many aspects of data heterogeneity in biomedical data to achieve better algorithmic performance and better result interpretation. We break down the whole framework into three aims as follows:

1.1 Aim 1: Capturing and quantifying heterogeneity

The capturing of heterogeneity can be the leading step for any downstream machine learning model because without this step, the model may make wrong assumption about the situation and this will subsequently lead to poor performance. One method that can shine light in the capturing stage is uncertainty quantification (UQ) which takes the model's input to decide how much uncertainty will arise after the prediction is made. UQ has been critical for robust learning under different contexts (known or unknown) in mobile activity recognition, healthcare, signal processing, and manufacturing. Existing models with Gaussian process or Bayesian approximation methods either rely on the assumption that variables in a system can be characterized by explicit probabilistic relationships (e.g., Bayesian models) or rely on generating one best model in the learning algorithm. However, measuring the predictive uncertainty for deep neural networks remains a challenging problem, closely related to the problem of detecting test samples that are drawn sufficiently far away from the distribution of training samples. For example, Malinin & Gales proposed a framework called Prior Networks (PNs) for modeling predictive uncertainty that explicitly modeled distributional uncertainty, Hendrycks & Gimpel proposed a framework that utilized probabilities from softmax distributions and detected out-of-distribution examples, by introducing confidence scores based on density estimators, later improved by processing the input and output of DNNs in [8]. Lee et al. proposed a method for detecting abnormal test samples by including both out-of-distribution and adversarial samples, to obtain the class conditional Gaussian distributions by introducing confidence scores based on the Mahalanobis distance. Our method is different from these works as we aim to learn a distribution of deep models rather than one single best one.

Other than wearable sensing application, the heterogeneity which is realized as uncertainty can be also prominent in medical dataset. The data imbalance, for example, can be problematic since the area with fewer data which is the minority class's area, will cause the model to be more uncertain. The performance of modern machine learning models is often biased in evaluation by commonly reported metrics (such as area under the curve of the receiver operating characteristic: AUC-ROC), often reporting overly-optimistic findings as a result of the imbalance between those that observe medical adverse events and those that do not. The adverse event of interest is often in the minority class. For example, in mortality prediction, patients with higher risk represent a smaller fraction in the cohort compared to most of the people who survive (thankfully). Naively applying machine learning models may render dissatisfaction: the outcome of interest can be extremely costly, either through unnecessary medical intervention (type 1 error) or misdiagnosis (type 2 error). We propose a framework to address class imbalance density and make use of this imbalance to render density-aware training for improved risk prediction performance. First, we decouple the training of representation learning and classification. Traditionally, representation learning and classification are trained jointly, but by decoupling, class specific features are extracted and class-specific predictions made, removing a source of bias for the learned classifier. Second, the density differences are important to learn, not eliminate, when modeling. Patients with lower risk (majority) are often lower risk because they do not contain any of the common risk factors (e.g. lack of hypertension, diabetes, prior myocardial infarction), and hence, form a dense cluster. However, patients with higher risk (minority) may arrive at this high risk from different factors (e.g. renal failure versus respiratory distress), thus scattering them in the data space. Our approach is density-aware, by avoiding re-sampling or re-weighing pre-processing steps, and the decoupling approach improves risk prediction performance. We demonstrate this approach in two different medical data scenarios: a randomized clinical trial dataset and an electronic health record (observational) dataset.

1.2 Aim 2: Dividing and conquering heterogeneity

Patient heterogeneity in real-world clinical data poses potential challenges in actualizing machine learning models and further complicates clinical decision making. For example, in evaluating risk modeling, two patients with the exact same risk score from the model might have different underlying driving factors, potentially rendering completely different treatments. Electronic health records (EHRs) are an example of a dataset with vast, heterogeneous clinical records that contain intra-and inter-variability of measurements and treatments across patients. To date, many deep learning techniques have begun exploring this data, often within subsets of clinical questions or data type, for example deep embedding of patient representations, time series modeling of structured clinical data with recurrent neural networks, and feature extraction techniques with convolutional neural networks. However, these methods often depend on a single model with 'one-size-fits-all' approach. An opportunity exists to stratify patients heterogeneity and generate explainable risk prediction from model-generated separation of *subtypes*, which may aid in clinical decision making in response to risk model outputs.

In order to stratify the patient population into specific patient subtypes (i.e. homogeneous subgroups) through a learnable 'divide-and-conquer' mechanism, we employ a mixture-of-experts (MoE) model. As one realization of ensemble learning, traditional MoE models generally assume a model containing a mixture of Gaussian experts. With the advancement of deep learning, the Gaussian experts have been replaced by neural networks in many applications with more complex architectures and larger learning capacity. However, such a larger feature space and bigger cohort for these models may be difficult to interpret for clinical decision making. Unlike LASSO for linear model interpretation, the introduction of sparsity to deep learning models, e.g. dropout and sparse gating, traditionally aims to prevent overfitting or alleviate the computation overhead. We propose that while all these methods with sparsity achieve strong predictive performance in their tasks, sparsity is not responsible for separating latent groups and reducing redundancy among different parts of the model for explaining heterogeneity. Therefore, enhancing these techniques with added interpretation of the patient subgroups is needed for successfully applying MoE model to clinical settings.

This work proposes an MoE framework with a macroscopic-style sparse gating network to address patient heterogeneity and provide interpretation to aid in clinical decision making, while improving risk prediction. To enable automatic subtyping of patients, the sparse gating, when coupled with an MoE framework, is to monitor the training activation of each expert rather than individual neuron. The sparse gating network is a realization of *model sparsity*, extending the concept of conditional computation, which has a trainable activation mechanism based on input examples. Compared to *dropout*, this *model sparsity* better explains the derived prediction in MoE scenarios, due to the fact that the sparse instance-based activation constrains each expert rather than neuron to handle different subsets of training data (different subtypes) and thus better at interpretation. Because the gating network and expert networks are trained concurrently, the label information is embedded in the expert assignment in the gating network, making the representation learned for each patient supervised and informative. In the experiment, we study the derived patient representation through expert assignment, showing discovery of clinically meaningful and interpretable patient subtypes. We demonstrate that sparse MoE not only learns to 'divide-and-conquer' a real-world clinical dataset with vast patient heterogeneity, but also provides better interpretation than raw features or generic encoding on how the model derives latent groups for prediction.

1.3 Aim 3: Adapting and tailoring to heterogeneity

As we have seen the data in the wild, the heterogeneity can be dynamically changing as well. For example, data missingness can be one form of data heterogeneity and can cause problem for modeling. Normally the imputation proposed can be useful under one circumstance but when the circumstance changes the model needs to be re-calibrated. In wearable sensing applications, data is inevitable to be irregularly sampled or partially missing, which pose challenges for any downstream application. An unique aspect of wearable data is that it is time-series data and each channel can be correlated to another one, such as x, y, z axis of accelerometer. We argue that traditional methods have rarely made use of both times-series dynamics of the data as well as the relatedness of the features from different sensors. We propose a model, termed as DynImp, to handle different time point's missingness with nearest neighbors along feature axis and then feeding the data into a LSTM-based denoising autoencoder which can reconstruct missingness along the time axis. We experiment the model on the extreme missingness scenario (> 50% missing rate) which has not been widely tested in wearable data. Our experiments on activity recognition show that the method can exploit the multi-modality features from related sensors and also learn from history time-series dynamics to reconstruct the data under extreme missingness.

Traditional missing value imputation techniques include filling with the mean value of a feature, or using forward or backward imputation. These static imputation methods do not reflect the underlying time-varying dynamics of data and therefore may bias downstream predictions based on the estimated likelihood of imputed values. Additionally, in wearable sensing applications, the different sensing channels are often capturing the same events of interest, so multiple imputation approaches may be suited to addressing this kind of missing data. However, these techniques have remained static in imputation as well. For example, a Multi-layer Percep-tron (MLP)-based model for irregular time-series data handling that accounts for such multiple imputation, still interpolates only a single channel, handling time-dynamics but not able to model across all channels (in other words, multiple models would be needed). On the other hand, MissForest, a tree-based machine learning method that predicts missing values from related, non-missing data, overcomes these limitations but does not make use of time-varying dynamics. In addition, the rate of missing data in traditional studies is relatively low, ranging from 1.98% to 50.65%. A stronger imputation technique is needed for real-world applications with severe missingness testing. We propose to use a long-short-term-memory-based denoising autoencoder (LSTM-DAE) to learn more robust imputation strategies for remote sensing data. The model has an encoder and decoder architecture to embed signal data, then this encoded information is fed into the LSTM network to learn the time-varying dynamics of the data. This architecture robustly imputes missing data from related channels and latest dynamics being measured by all sensor channels, even in the presence of high rates of missing data. In order to demonstrate the utility of using both time-series dynamics and feature relatedness, we experiment on datasets with both inherent missing values and increased missing data (up to 60% missing across all the channels on a dataset with inherent 66% missingness), which surpasses traditional missing rate study by a large margin.

2. FIRST AIM: CAPTURING AND QUANTIFY HETEROGENEITY

2.1 A context-aware framework for uncertainty quantification under wearable sensing[1]

2.1.1 Introduction

2.1.1.1 Overview

Activity recognition in wearable computing faces two key challenges: i) activity characteristics may be context-dependent and change under different contexts or situations; ii) unknown contexts and activities may occur from time to time, requiring flexibility and adaptability of the algorithm. We develop a context-aware mixture of deep models termed the α-β network coupled with uncertainty quantification (UQ) based upon maximum entropy to enhance human activity recognition performance. We improve accuracy and F score by 10% by identifying high-level contexts in a data-driven way to guide model development. In order to ensure training stability, we have used a clustering-based pre-training in both public and in-house datasets, demonstrating improved accuracy through unknown context discovery.

In wearable computing, context-awareness helps recognize activities based on sensor measurements under different situations. Context can be defined as “any information that can be used to characterize the situation” [9] or to improve recognition [10]. Context-aware systems have previously been used in many applications, including activity recognition [11], online, personalized and adaptive activity classification [12], and healthcare applications [13, 14]. The definition of context heavily relies on domain knowledge, such as a user’s tasks (e.g., spontaneous activity, engaged tasks) or a user’s social environment (e.g., co-location of others, group dynamics), etc. However, in practice, pre-defined contexts may not always be available, or definitions of contexts may change in different environments. Additionally, new unknown contexts may emerge over time. For these reasons, there is a general data insufficiency and lack of contextual information to develop

[1] Reprinted with permission from Huo, Z., PakBin, A., Chen, X., Hurley, N., Yuan, Y., Qian, X., Wang, Z., Huang, S. and Mortazavi, B., 2020, June. Uncertainty quantification for deep context-aware mobile activity recognition and unknown context discovery. In International Conference on Artificial Intelligence and Statistics (pp. 3894-3904). PMLR. Copyright 2020 by the authors.

accurate context-aware activity recognition systems that could adapt to these unknown contexts. While existing works are not adequate to address the challenges such as a lack of context information in data while training the model, or the emergence of unknown contexts when the model is applied [15, 16], we propose a data-driven approach with context-awareness capability to achieve better activity recognition performance. Specifically, we develop an integrative α-β framework to simultaneously learn unknown contexts and the distribution of each user's specific activity likelihood within each context. In this framework, the α network is the context detector to learn a distribution over contexts as a mixture of weights, and the β network models activity recognition for context-specific sensor measurements. For example, given the sensor reading data from a user, the α network detects the context by generating a distribution over different contexts; then, each context has a dedicated β network that outputs a distribution over different activities.

We further extend our model with the ability to explore new unknown contexts by equipping the α-β network with uncertainty quantification (UQ) based on the maximum entropy learning (MEL) principal. MEL identifies the distribution of the parameters of a statistical model that bears the maximum uncertainty, rather than one single best model, as a principle to achieve robustness in prediction and modeling. The prediction model could refuse to predict on given data if the uncertainty for making a prediction on this data is higher than a threshold. This method adapts to data and effectively discovers unknown contexts with the UQ. This work allows for models trained in laboratory settings to extend to natural environments for monitoring behaviors and performances of users, as in Figure 2.1.

Our contributions are as follows. In this chapter, we propose a context-aware model for activity recognition. The context and activity are simultaneously modeled by dedicated networks. For unknown contexts, an overarching UQ method is applied to all the model parameters. This provides robustness in testing new context that our model can uniquely offer, beyond a traditional activity recognition technique. Finally, we demonstrate these findings in both a publicly-available benchmark dataset and an in-house dataset we collected to identify confounded versions of human motion, and make this data available for public use.

2.1.1.2 Related Work

Context-Awareness: Mixture of Experts Model In many applications of machine learning, heterogeneous data can be divided into smaller homogeneous groups that can be modeled more accurately. Context-awareness, as an example, plays an important role in improving the performance of activity recognition systems [17, 18]. In Mixture of Experts (MoE) models [19] such group-level clustering and modeling comes as a single training step, rather than splitting data a prior then building models. Elements are clustered based upon their relationship and the next level modeling accuracy. MoEs have successfully served different applications from classification and regression tasks [20][21] to phenotyping in medical datasets [22]. In [19][20][23], authors provided formulations of probabilities of observed given different experts. The MoE structure consists of two components: A gate and several experts. The gate is often modeled by Gaussian Mixture Models (GMM) [24][25] and neural networks [26], while expert modeling is more dependant on the application including SVM [26] [27]. This work used neural networks for both.

Different methods have been used in the literature in order to determine the number of experts for these models [20]: growing models where the experts with the worst performance decompose into a MoE themselves [28] [29], pruning models where they start with a large number of experts and reduce the number by combining/removing experts [30], and exhaustive search in cases where tree topologies are not overly complex [31].

Uncertainty Quantification: Previous Studies UQ has been critical for robust learning under different contexts (known or unknown) in mobile activity recognition [32], healthcare [33, 34], signal processing [35], and manufacturing [36]. Existing models with Gaussian process [32] or Bayesian approximation methods [37] either rely on the assumption that variables in a system can be characterized by explicit probabilistic relationships (e.g., Bayesian models) or rely on generating one best model in the learning algorithm. However, measuring the predictive uncertainty for deep neural networks remains a challenging problem, closely related to the problem of detecting test samples that are drawn sufficiently far away from the distribution of training samples. For example, Malinin & Gales proposed a framework called Prior Networks (PNs) for modeling predictive uncertainty that explicitly modeled distributional uncertainty [38], Hendrycks & Gimpel proposed a framework that utilized probabilities from softmax distributions and detected out-of-distribution examples, by introducing confidence scores based on density estimators [39], later improved by processing the input and output of DNNs in [8]. Lee et al. proposed a method for detecting abnormal test samples by including both out-of-distribution and adversarial samples, to obtain the class conditional Gaussian distributions by introducing confidence scores based on the Mahalanobis distance [40]. Our method is different from these works as we aim to learn a distribution of deep models rather than one single best one.

2.1.2 Methods

Contextual information can help model similar activities together, resulting in an improvement of activity recognition performance by reducing the search space of activities to recognize given a set of features. Our proposed α-β network integrates activity recognition with unsupervised context detection, as detailed in Section 2.1.3. As context can vary from person to person, and change over time for the same person, in Section 2.1.4, we present maximum entropy learning (MEL) based UQ in the α-β network to discover unknown contexts when needed.

We further extend our model with the ability to explore new unknown contexts by equipping the α-β network with uncertainty quantification (UQ) based on the maximum entropy learning (MEL) principal. MEL identifies the distribution of the parameters of a statistical model that bears the maximum uncertainty, rather than one single best model, as a principle to achieve robustness in prediction and modeling. The prediction model could refuse to predict on given data if the uncertainty for making a prediction on this data is higher than a threshold. This method adapts to data and effectively discovers unknown contexts with the UQ. This work allows for models trained in laboratory settings to extend to natural environments for monitoring behaviors and performances of users, as in Figure 2.1.

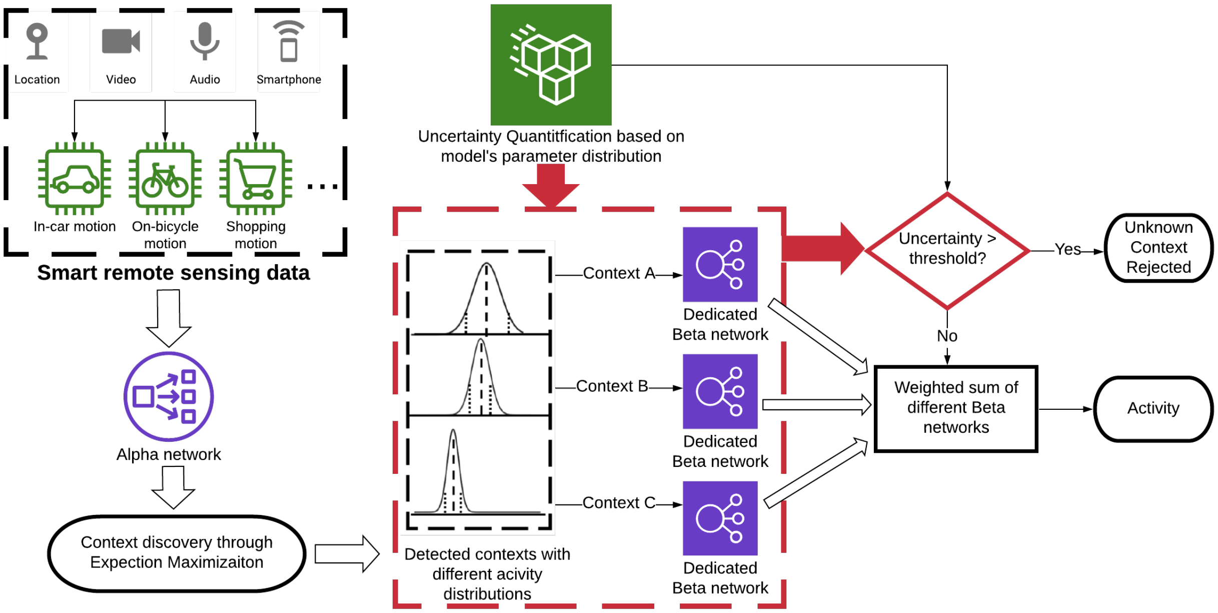


Figure 2.1: A conceptual overview of UQ integrated with the α-β network. Reprinted/adapted with permission from [1].

2.1.3 Context-Awareness Processing

In this work we develop a mixture of CNNs, the α-β network, where each mixture component is dedicated to one specific context. There are two types of networks: α and β. Given the sensor data, the α network detects context by generating a probability distribution over all known contexts. Each context has a dedicated β network that outputs a probability distribution over different activities. Our activity recognition problem features a latent context variable and can be formulated as:

$$\begin{aligned} \log p(ACTIVITY|\mathbf{X},\theta) &= \sum_{i=1}^{N} \log p(activity_i|\mathbf{x}_i,\theta) \\ &= \sum_{i=1}^{N} \log \sum_{c=1}^{N_c} p(activity_i|c_i = c, \mathbf{x}_i, \theta) p(c_i = c|\mathbf{x}_i, \theta), \end{aligned} \tag{2.1}$$

where θ denotes the mixture component parameters, N denotes the number of data samples, and N_c denotes the number of expected clusters (contexts) to which each data point may belong. Our objective is to maximize Eq. (2.1) with respect to θ. The log-likelihood has a lower bound, first formulated in [41]:

$$\log p(ACTIVITY|\mathbf{X}, \theta) \geq$$
$$\sum_{n=1}^{N} \sum_{c=1}^{N_c} q(c_i = c) \log \frac{p(activity_i|c_i = c, \mathbf{x}_i, \theta).p(c_i = c)}{q(c_i = c)},$$

where $q(\cdot)$ is the distribution over different contexts and $p(\cdot|context, x, \theta)$ is the distribution over activities given the context and the input. $q(\cdot)$ is modeled using the context-detecting α network, and each $p(\cdot|context, x, \theta)$ is modeled using a β network (each context has its own β network). While [41] used a supervised technique to define contexts as specific locations, this work provides for an unsupervised exploration of context. Additionally, in our implementation, we use a network for context recognition (α network) and a dedicated network for each context (β network) whereas [41] used only two networks regardless of the number of contexts. Following the EM algorithm, the lower bound in Eq. (2.2) can be maximized. Specifically, the loss, the negative of the lower bound, is minimized. In the E-step, $q(\cdot)$ is optimized which translates to optimizing α network while freezing β networks. In the M-step, θ (model parameters) need to be optimized which translates to optimizing β networks while freezing α network. The EM training alternates iterations of training either α network or β networks while keeping the other(s) fixed. It should be noted that no labeled contextual data is used in the training process for this α-β network.

2.1.4 Unknown Context Discovery

The α network enables context detection; however, in practice, contexts may change over time, or may not always be pre-defined. It is possible to improve context-aware systems by detecting the uncertainty of possible unknown contexts as a result of potential distribution mismatch between known and unknown contexts. To identify unknown contexts, we combine the feature extraction power of deep learning with the learning power of MEL to define a probabilistic mechanism for unknown context discovery.

While many of the current works focus on revising general deep models with a probabilistic evaluation of their model or prediction, here we have a different aim: We modify the α-β network to be adaptive to changing contexts hidden in data. We relax our expectation of identifying one single optimal model of the α-β network; rather, we consider solving for a full distribution over multiple models. The intuition is that many different models might generate relatively similar performance, so it would be better to estimate a distribution over parameters $p(\mathbf{w})$, from the output layer of the α networks that detects context. This aligns with the basic principal of MEL. Therefore, we equip the α-β network with UQ capacity based on the MEL principal by identifying the distribution of the parameters of a statistical model that bears the maximum uncertainty.

2.1.4.1 Uncertainty Quantification via Minimizing Relative Entropy

To learn the distribution of the α-β network parameters that encode maximum uncertainty, we employ the MEL formulation. There are two steps. First, we create constraints that encode information from the data. For example, for each sample we derive a loss function such that the expected prediction on this sample over all the possible model parameters matches the observed outcome on this sample, as in traditional ML. Second, on the top of this constraint structure, the learning objective of MEL is to learn the distribution of the model parameters with the maximal entropy in terms of the parameter posterior distribution. Thus, unlike traditional machine learning methods that estimate a single optimal setting of the parameter, MEL considers a more general problem of these methods by solving for a full distribution over multiple $p(\mathbf{w})$ values.

2.1.4.2 Analytical Details of MEL

To further illustrate this distribution approach, note that our context detection problem is also a classification problem where the response variable is denoted by y taking values for different contexts. Let $\mathbf{x}_n = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ be an input feature vector as an aggregate of all the measures from sensors for each window, and let $\mathcal{D}(\mathbf{x}_n|\mathbf{w})$ be the discriminant function parameterized by $\mathbf{w}$ implemented in the α network. Traditional learning machines such as the max-margin methods estimate the optimal $\hat{\mathbf{w}}$ that minimizes the classification error in predicting the labels of training examples as:

$$\hat{y} = sign\mathcal{D}\left(\mathbf{x}_n|\mathbf{w}\right). \tag{2.2}$$

Based on this line of thought, we can classify margin as $y_n \mathcal{D}(\mathbf{x}_n, \mathbf{w})$, and learn the optimal parameter setting $\mathbf{w}$ by the empirical loss and the regularization penalty as:

$$\begin{aligned} &\min_{(\mathbf{w},\gamma_n)} R(\mathbf{w}) + \sum_n L(\gamma_n) \\ &\text{s.t.} \quad y_n \mathcal{D}(\mathbf{x}_n \mid \mathbf{w}) - \gamma_n \geq \mathbf{0}, \quad \forall n. \end{aligned} \tag{2.3}$$

where $L(\gamma_n)$ is the loss function, a non-increasing and convex function of the margin, and $R(\mathbf{w})$ is the regularization penalty. Given $p(\mathbf{w})$, we can recast (2.3) as an integration where the classification constraints will also be applied in an expected sense. Instead of considering an expectation of the regularization penalty functions, we can apply a canonical penalty function for distributions, the negative entropy; minimizing the negative entropy is equivalent to maximizing the entropy. Hence, we use the Shannon entropy defined as $H(p(\mathbf{w})) = -\int p(\mathbf{w}) \log p(\mathbf{w}) d\mathbf{w}$. This gives us the following objective function to learn the distribution $p(\mathbf{w})$ over the parameters $\mathbf{w}$:

$$\begin{aligned} &\min_{p(\mathbf{w})} H(p(\mathbf{w})) \\ &\text{s.t.} \quad \int p(\mathbf{w})[y_n \mathcal{D}(\mathbf{x}_n, \mathbf{w}) - \gamma_n] d\mathbf{w} \geq \mathbf{0}, \quad \forall n. \end{aligned} \tag{2.4}$$

As a result, MEL no longer finds a fixed set of the parameters, but a distribution over them. Learning such a distribution of model parameters does not rely on assumptions on the model's mathematical form. It also does not rely on knowing a particular distribution as is needed in Bayesian learning frameworks. Therefore, MEL is more flexible than typical Bayesian learning methods [42, 43] to characterize uncertainties associated with complex models such as the α-β network here. To solve the MEL formulation (2.4), we could derive a Lagrangian $J(p)$, and take the derivatives with respect to $\mathbf{w}$ and set them to 0. To do that we first need to calculate the unconditional maximum of the problem (2.4) plus the constraints added with some multiplying factors (the Lagrange multipliers), which give the probabilities in a functional form with the Lagrange multipliers as parameters. Our UQ approach compares the uncertainty with a threshold to see whether a given sample should be detected as belonging to a new context or not. We defined a classification with rejection option as $\hat{y}_i^{Rej}$, where if a sample is rejected $\hat{y}_i^{Rej} = 0$, and if it is accepted $\hat{y}_i = \hat{y}$, where $\hat{y}$ corresponds to the classification of the ith sample. Note that, a sample is rejected when $p(\mathbf{w}|x_i) < \epsilon$, and ϵ is chosen through cross-validation.

The distribution of parameters forms a quantitative evaluation of model uncertainty, which could be further used in subsequent decision making by using probability laws to track the uncertainty propagation process. In this chapter, we create a rejection option, a flexibility enabled by the UQ capacity. The rejection option allows for the prediction model to refuse to generate a prediction if the uncertainty is higher than a given threshold. This is typically solved by estimating the class conditional probabilities and rejecting the samples that have lower posterior probability of class.

2.1.5 Experiments

In this section, we discuss experiments with our deep context-aware mixture of experts for activity detection coupled with maximum entropy based uncertainty quantification. We first introduce the datasets, and then the detailed implementation of our models. Then, we show various competitive baselines to demonstrate that our pipeline performs the best as evaluated by several different metrics.

2.1.6 Datasets

UCI data. We used the UCI OPPORTUNITY dataset [44] for context-aware human activity recognition. The dataset contained 18 different activities performed in five different contexts and sensed by 72 different sensors. Each of the 18 activities had one of the five contexts, but not all contexts contained every activity. Therefore, the UCI OPPORTUNITY dataset provided a realistic capture of a situation where not all of the human activities occurred with an equal likelihood in all contexts. The UCI OPPORTUNITY dataset has seven levels of hierarchical labels. Higher level labels described details such as subject posture, while lower level labels described the hand movements or interactions with other subjects. In this study, we chose a higher level label (e.g., cleaning time) as the context and a lower level (e.g., opening a door) label as the activity. We used all the body-worn sensors which included seven inertial measurement units (IMUs) and twelve 3D acceleration sensors. Five IMUs were on the upper body while two were on user's shoes. Accelerometers were on the upper body, hip, and leg, which translated to 133 columns in the raw dataset.

In-house data. To generate a more realistic experimental setting, we have collected our data to detect different types of human motion that may be confounded by environments factors. The dataset is made publicly available, serving as extra part of our contribution in this chapter. The motivations are three fold: 1) instead of collecting the data in a strict laboratory setting, we have loosened all the rules for subjects, including their choice of rest time and the pace of activity. 2) we have devised a set of much more realistic contexts to be detected. Those contexts can cover almost all the real world setting a subject might face. 3) we have collected a much nosier dataset compared to previous ones, with respect to the randomness we imposed on the data collection, such as a rare activity as walking with one shoe off in both outdoor (lawn) and indoor (hardwood covered by carpet). As a result we have 3 contexts with corresponding movements (1) outdoors: Crawling, Jogging, Riding Bike, Sprinting, Walking, Walking with One Shoe (simulated limping), Walking with Weight in Arms, Walking with Weight on Back. (2) Movements that happen indoors: Escalator Up, Escalator Down, Elevator Up, Elevator Down, Lying Down, Sitting, Stairs Down, Stairs Up, Standing, Walking, Walk with One Shoe (simulated limping), Cooking, Dancing, Eating, Reading, Sleeping, Talking on phone, Talking to Another Person, Using PC, and (3) movements that happen outdoors but in vehicles: Driving Car, Riding Car, Riding Bus, Reading, Device Usage. The movements, however, are not unique to each context, and the position of the phone may change through usage. We collected data on 20 people, each doing 32 activities while having 3 phones, one in hand, one in the pocket and one in the backpack, as those are the common places for phone positions. The applications used on the phones for data collection was *Readisens* [45]. The sensors used are: acceleration (in three axes), altitude, compass, gyroscope (in three axes), GPS information (latitude, longitude), screen time information, and phone speed. This data contains contextual information which is independent of the locations of the phone. Participants were given minimal instruction for the execution of the activities in order to allow for individual variation. Participants were also wearing clothes of choice and the order of the activities were randomized. This study was reviewed and approved by the Texas A&M Institutional Review Board (IRB # 2018-210D). The data has been made available for public use.[2]

2.1.7 Implementation

In UCI dataset, 19 different preprocessed sensors were fed into the network. Time series data were divided into non-overlapping segments of 1 second (30 samples). In each window, the features of sensors are concatenated. We used a five-fold cross-validation to evaluate our models with testing accuracy and micro F score as performance metrics. In our experiments, we used 3 convolutional layers followed by 3 fully connected layers for both α and β networks. Note that these two networks' architectures were different in terms of the number of neurons in the output layer. Networks were trained using stochastic gradient descent with an initial learning rate of 0.001 and a momentum of 0.9, which provided the best results in cross-validation.

Implementation of MEL. For maximum entropy classifier, we have the input from the parameter distribution derived from the α network. This network outputs $\hat{y}_i^{Rej}$ as a probability of rejection, which is compared against a threshold that is derived from cross-validation to further guide the β network for fine-grained activity detection under specific context. We defined a classification with rejection, where if a sample was rejected (if the uncertainty was higher than a specific threshold), the prediction model refused to generate a prediction by setting the predicted context to zero.

In the UCI OPPORTUNITY dataset we had five contexts: relaxing, coffee time, early morning, clean-up, and meal time. To test our unknown context discovery, we adopted a rotating strategy. In this strategy, we removed one context and its corresponding data from the training dataset at each rotation, and trained an $\alpha - \beta$ network only on the remaining known contexts. In this case, one context was assumed to be unknown and was treated as a hold-out to be used for unknown context discovery assessment. The final evaluation was conducted through comparison of activities

[2] https://github.tamu.edu/guangzhou92/RealActivity

that were sampled from both known and unknown contexts, to demonstrate the model's ability to distinguish the unknown context.

Pre-training. Initialization is an essential step for both the optimization and for the training of the neural networks. Without proper initialization, the model collapses into selecting only one specific β network while eliminating the contribution of others, as is observed in many other mixture network modeling. We have added pre-training to solve this problem and have compared it with regularization. We have shown that it is much more effective in terms of accuracy. Pre-training is an important stage in the α-β network training. The model could approach the base model of a single neural network classifier without proper pre-training because of the large gradients at the beginning of the training stage; large gradients cause the selector to saturate and select only one β network. Therefore, the full capabilities of the α-β network can only be achieved using proper pre-training, which proves useful in finding subgroups of data as well as in yielding a better performance. We used the idea presented in [46] to cluster the activity data. In detail, a base network was trained with sensor readings as input and activities as output. Next, the CNN segment of the network was used to embed the sensor readings into the features which have proven to be descriptive of the input data [47, 48]. Subsequently, we used K-means to cluster the input into a fixed number of clusters. Finally, we trained our α network to learn the mapping between sensor readings of activity to clusters.

2.1.8 Baseline

We compare our model against several baselines. The first baseline is a single β network, which is a component of the α-β network. Another baseline, for each specific number of contexts, is a β network which has wider hidden layers in order to have the number of parameters equal to the α-β output (similar size). In other words, for a α-β network with the number of clusters equal to k, given that α and β networks have the same size, the baseline is a β network with hidden layers $k + 1$ times as large to have roughly the same capacity. Finally, in order to demonstrate the performance of our UQ method, we design a similar probabilistic baseline, logistic regression, to output a rejection likelihood and we compare it with our method to show the better unknown con-

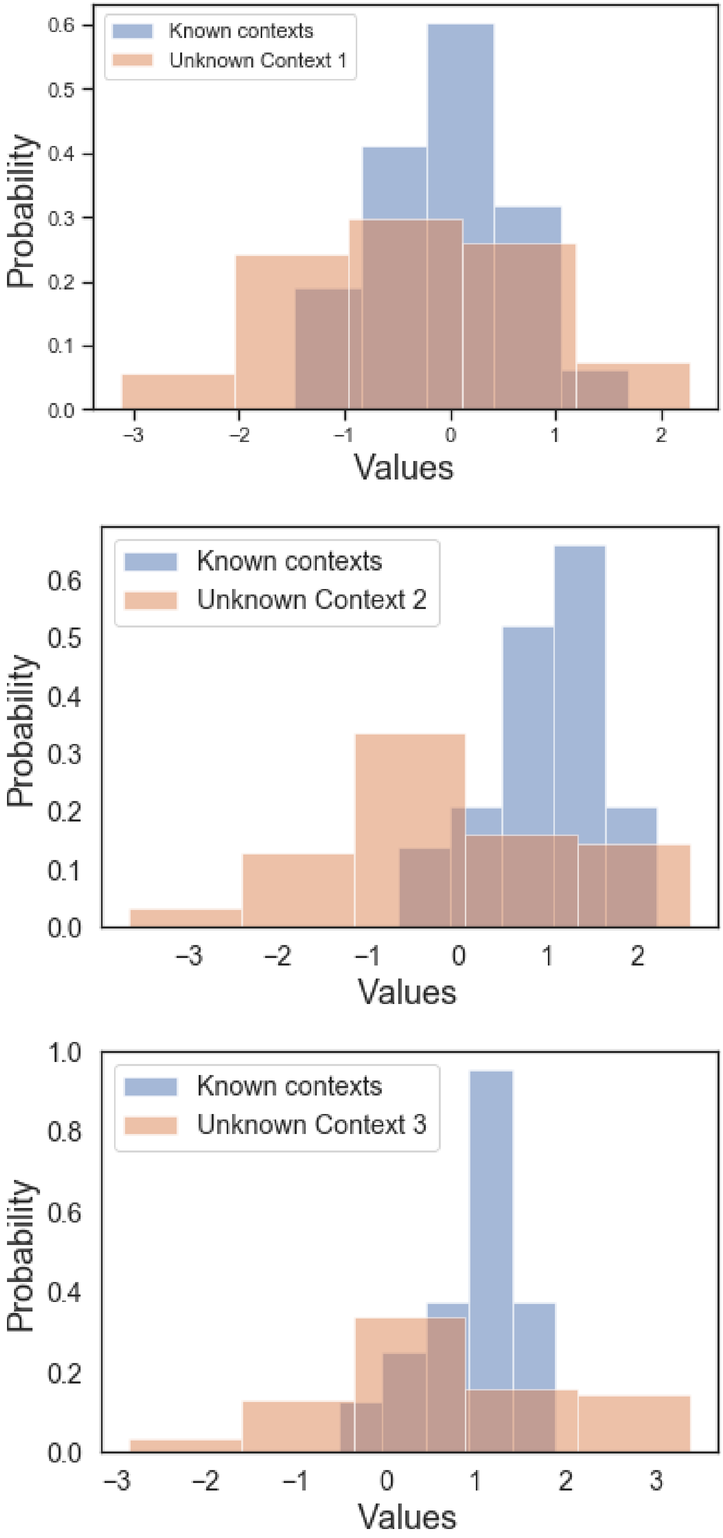


Figure 2.2: Predicted probability distributions when a context is removed v.s the aggregate of all known contexts within each rotation. Reprinted/adapted with permission from [1].

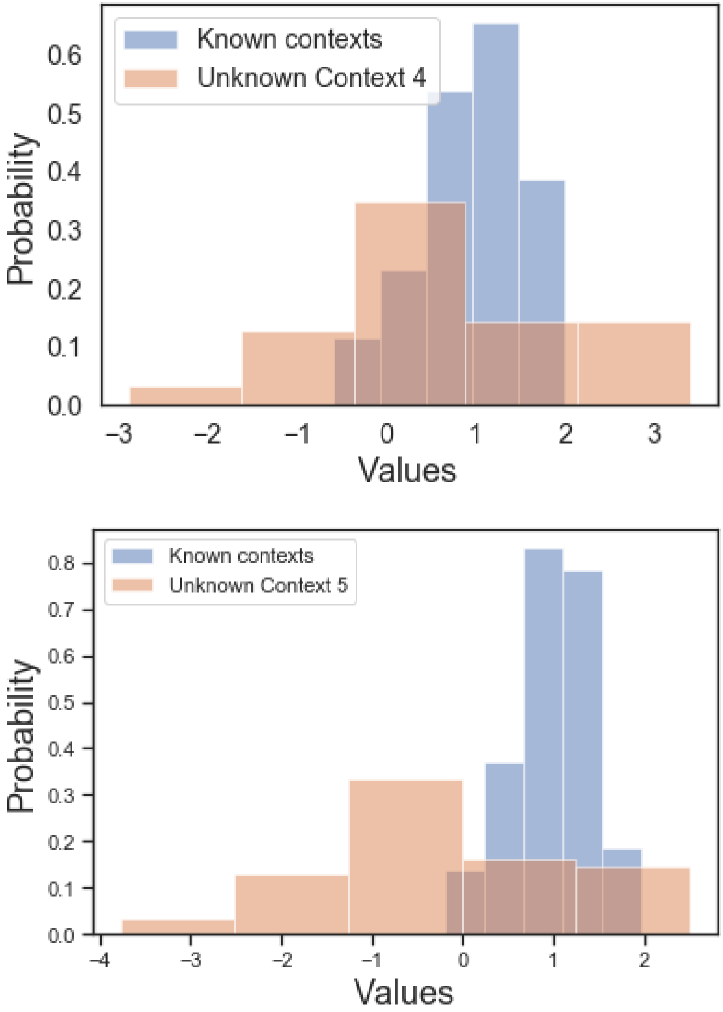


Figure 2.3: Predicted probability distributions when a context is removed v.s the aggregate of all known contexts within each rotation. Reprinted/adapted with permission from [1].

text discovery performance from our UQ pipeline. Baseline doesn't impose extra regularization, and all parameters were selected through cross-validation.

2.1.9 Result of UCI dataset: UQ vs. Baseline

For the performance of UQ against baseline, results are shown in Table 2.1. As can be seen, the proposed UQ method detects unknown contexts better than baseline in all evaluations.

Figure 2.2 and Figure 2.3 present a main result of UQ in this experiment from the UCI OP-

Table 2.1: UQ results VS. baseline for unknown context discovery where contexts are 1 = Relaxing, 2 = Coffee time, 3 = Early morning, 4 = Clean up, 5 = Sandwich time. Reprinted/adapted with permission from [1].

Performance measure	Context number that was removed each rotation				
	1	2	3	4	5
UQ					
Sensitivity	0.63	0.71	0.75	0.62	0.73
Specificity	0.72	0.72	0.76	0.79	0.82
Testing accuracy	0.67	0.71	0.75	0.69	0.77
F-score	0.67	0.72	0.77	0.70	0.78
Baseline					
Sensitivity	0.26	0.34	0.17	0.19	0.26
Specificity	0.75	0.79	0.81	0.80	0.80
Testing accuracy	0.66	0.72	0.64	0.64	0.69
F-score	0.26	0.34	0.17	0.20	0.27

PORTUNITY dataset. Here, the distribution of predicted probabilities of the removed context (the red histogram) is shown with the distribution for all the other contexts combined (the blue histogram). The UQ algorithm is able to detect the uncertainty for the unknown context, as the unknown context usually leads to smaller probabilities in comparison with the distribution of the known contexts.

2.1.10 Results of UCI dataset: α-β Network vs. Baseline

Table 2.2 compares the testing accuracy of the α-β network and the baseline (a single network that is equivalent to a single β network) for predicting labels in the UCI OPPORTUNITY dataset. The testing accuracies are averaged among all their corresponding bootstraps (bootstrapped 5 times). Table 2.2 shows that the mixture of the context-specific neural networks improved on the accuracy of the baseline from 86% to 96% when using nine contexts. Thus, our context-aware α-β network was able to find subgroups in the data in an unsupervised manner with different numbers of contexts This fact was reflected in the accuracy boost due to the subgroup modeling by different β networks.

Table 2.2: Accuracy (F score) for the α-β network and baseline which is a β network with the same number of parameters for each specific number of clusters. For 1 cluster both the models are the same. Reprinted/adapted with permission from [1].

Clusters	2	3	4
α-β	**0.89(0.90)**	**0.89(0.90)**	**0.91(0.91)**
baseline	0.81(0.81)	0.84(0.84)	0.84(0.83)

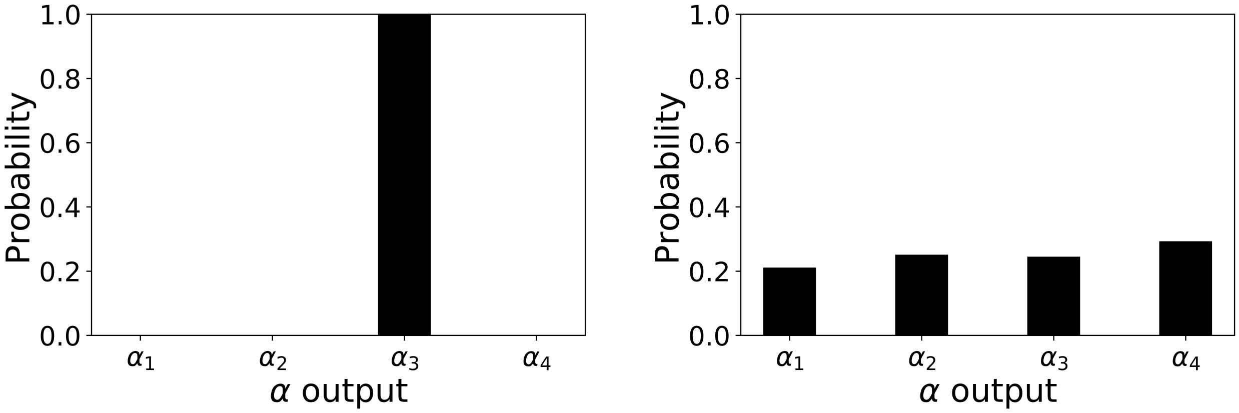
Clusters	5	6	7
α-β	**0.92(0.91)**	**0.91(0.92)**	**0.96(0.96)**
baseline	0.83(0.82)	0.86(0.86)	0.85(0.85)

Clusters	8	9	10
α-β	**0.96(0.96)**	**0.96(0.96)**	**0.95(0.95)**
baseline	0.84(0.84)	0.86(0.86)	0.86(0.86)

The α-β network is equipped with pre-training. The comparison in Figure 2.4 shows that pre-training is extremely important, as the model without pre-training results in selection of only one network, leading to model collapse. The effect of pre-training can also be seen in the network performance. Figure 2.5 shows that pre-training is much more effective than regularization since regularization just penalizes single network usage but does not use any knowledge about the data in the process. We took the best number of contexts in terms of accuracy and F-score, 9, and tried to train networks with the same number of contexts but without pre-training and by using only regularization with different regularization coefficients. Without proper initialization, the α-β network drops from 96% to 87% in accuracy and 0.96 to 0.87 in F-score. This is 4% less than the performance of an identical α-β network with proper initialization in Table 2.2. This is roughly equal to the performance achieved by a single β network (refer to baseline results in Table 2.2).

2.1.11 Results: In-house data

Having tested our methodology on the UCI OPPORTINITY dataset, we use our In-house dataset to further assess α-β Network with UQ in a noisier environment. We first measure the activity detection accuracy and F-score of the α-β network, shown in Table 2.3. It can be seen that our method still outperforms the baseline (84% vs. 80%). Note that our model did drop accuracy



(a) Without pre-training

(b) With pre-training

Figure 2.4: Average of α network output in testing without (a) and with (b) pre-training. The model collapses into a single network in the former. This shows the effectiveness of pre-training in making sure that the α network finds subgroups in the data and hence can take advantage of context-aware recognition. Reprinted/adapted with permission from [1].

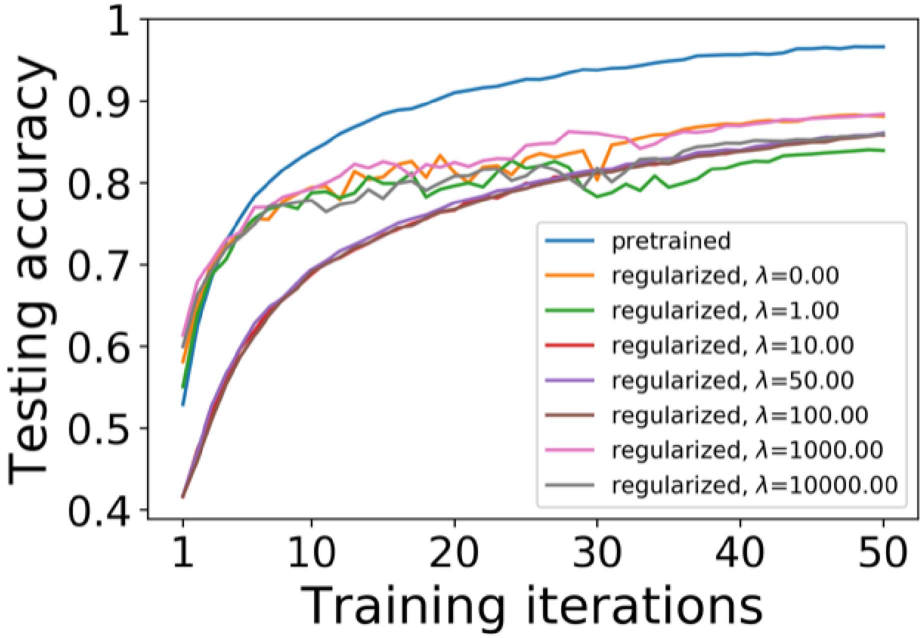


Figure 2.5: Comparison between pre-training and regularization. Reprinted/adapted with permission from [1].

compared to the average results from Table 2.2 (84.3% (std 0.015)), showing we indeed collected a noisier dataset which is more realistic. Secondly, to present the model's ability to detect new

Table 2.3: In house data, Accuracy and F score for the α-β network and baseline which is a β network with the same number of parameters for each specific number of clusters. For 1 cluster both the models are the same. Reprinted/adapted with permission from [1].

Performance Measures	Accuracy	F score
α-β Network	0.84	0.86
baseline	0.80	0.80

contexts on such a dataset, we present the UQ results as well, shown in Figure 2.6. It is clear that unknown contexts are more spread out, whereas known contexts have more concentrated predictive posterior probability distributions. We have further pushed the experimental setting harder to make it more realistic. That is, in this dataset with three contexts we removed 2 contexts as unknown contexts and just run the UQ solely on the seen contexts, and calculate the probability distribution to see if it can distinguish different contexts. As a result, the model can still find concentrated probability from known contexts but flat distribution for unknown ones, showing our superior performance on unknown context discovery.

2.2 Task-agnostic continual learning with Bayesian Novelty as Uncertainty Quantification[3]

In conventional machine learning, data points are assumed to be identically and independently distributed (*iid*) and all available at once. In contrast, the regime of continual learning (CL) presents the new challenge of incrementally accumulating knowledge from past experiences, mimicking human's ability to learn with non-*iid* data streams in widely varying contexts. CL sequentially learns novel concepts to achieve reliable predictions without *catastrophically forgetting* previously learned knowledge. It can be applied to many real-world applications, such as robotics [49], computer vision [50], autonomous driving [51], and healthcare monitoring [32, 52, 53, 54]. To this end, many CL methods have been developed in attempting to solve the *stability-plasticity dilemma* [55, 56, 57, 58, 59, 60, 61, 62, 63, 64].

Many existing CL methods assume that the data stream is explicitly divided into a sequence of

[3] Reprinted with permission from Ardywibowo, R., Huo, Z., Wang, Z., Mortazavi, B.J., Huang, S. and Qian, X., 2022, June. VariGrow: Variational Architecture Growing for Task-Agnostic Continual Learning based on Bayesian Novelty. In International Conference on Machine Learning (pp. 865-877). PMLR. Copyright 2022 by the authors.

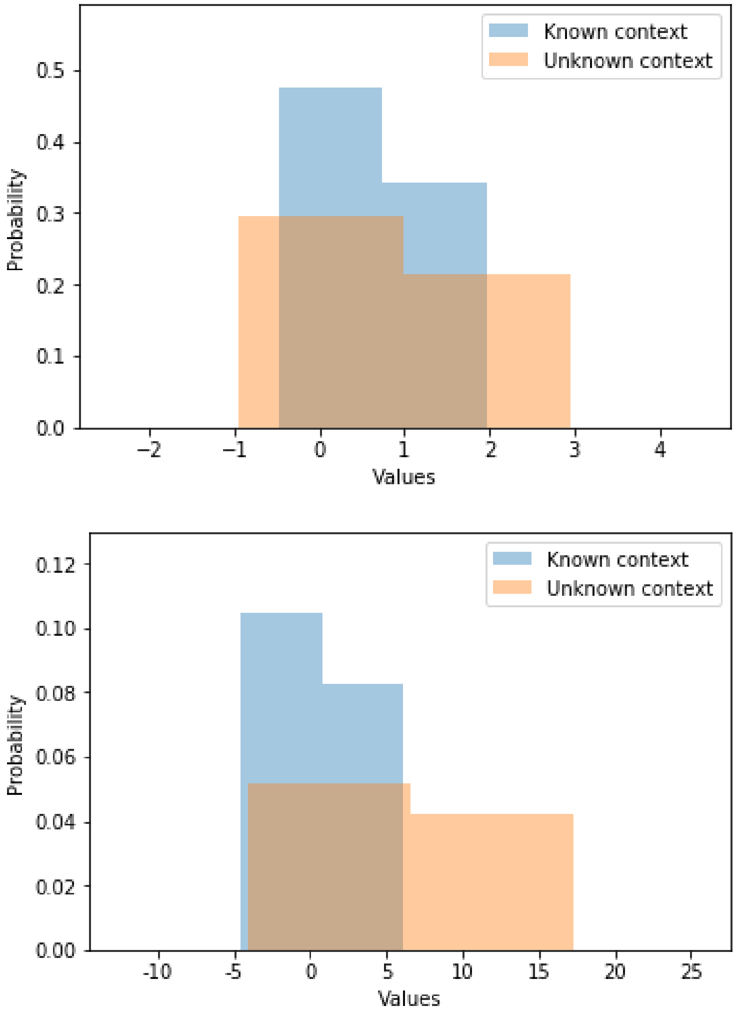


Figure 2.6: Predicted probability distributions when 1 context is removed vs. 2 contexts are removed. Reprinted/adapted with permission from [1].

transiting contexts, termed as **tasks**, with task information given at both *training* and *testing* time. In real-world scenarios, however, there is no clear transition boundary between different contexts or tasks, limiting the application of these CL methods in practice [65]. With this in mind, **task-agnostic CL** performs continual learning without requiring task IDs and their transitions. This new setting is challenging, dubbed as the *single-headed* setting, where existing task-agnostic CL methods have significantly lower performance compared to their task-aware counterparts [66, 67, 68, 69, 70]. In this chapter, we focus on task-agnostic CL for classification problems.

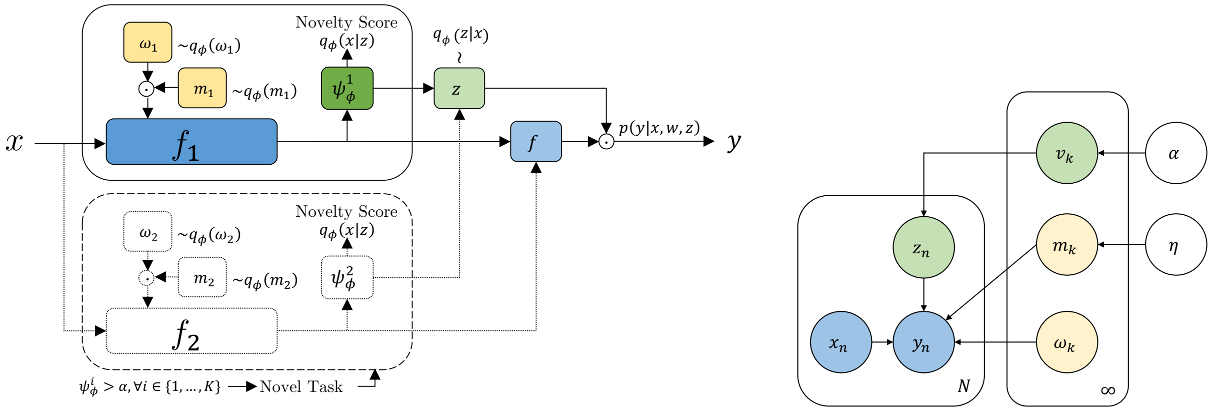
Existing CL methods can be broadly categorized into 1) regularization-based, 2) memory-based, and 3) expansion-based [71]. While regularization- and memory-based methods focus on retaining the knowledge learned from the old tasks, expansion-based methods lean towards better absorbing new knowledge and circumvent the *capability saturation* [72]. To the best of our knowledge, though, methods that tackle both catastrophic forgetting and capability saturation, under the task-agnostic CL paradigm, are lacking [73]. Bayesian inference offers a promising way to reconcile this problem, with old data points naturally being summarized by a posterior distribution that can be sequentially updated. The inherent uncertainty quantification capability enables effective task-agnostic CL without needing task information explicitly [74]. In particular, Bayesian nonparametrics offer a natural solution to the *stability-plasticity* dilemma by principally increasing model complexity as novel data arrives. However, the posterior distribution becomes intractable with large and complex datasets, and existing sequential variational approximations are not flexible enough to capture the complexity of these datasets [65, 75, 76, 77]. Promisingly, implicit variational inference enables flexible modeling of the posterior [78, 79, 80]. However, their application to dynamically growing architectures for task-agnostic CL has not been previously explored.

In this chapter, we propose VariGrow, a **Vari**ational architecture **Grow**ing framework for task-agnostic continual learning. To accomplish this, we first formulate model or network growing in terms of Bayesian nonparametric distributions that define an infinite mixture of expert distributions, which can be considered as having an expansion-based backbone. This consists of an expert distribution for each mixture component and a mixing distribution selecting from which expert the data originate. We then approximate these distributions using flexible implicit variational distributions, allowing us to more accurately capture the posterior at each incremental step. Specifically, we design a mixing distribution using energy-based novelty scores to determine the mixture component to which each data point belongs [81, 82]. This allows to dynamically decide whether to grow a new mixture component for novel instances, or to assign it to an existing one. Meanwhile, each component is handled by an expert distribution defined implicitly through Bayesian Neural Networks (BNNs) [78, 79, 80]. By deriving tractable approximations to the Kullback-Leibler (KL) divergence, we optimize the Evidence Lower Bound (ELBO) of our formulation through stochastic gradient-based techniques along with a sparsification trick to ensure expressiveness. We have tested VariGrow on several CIFAR and ImageNet-based benchmarks for the **strict task-agnostic** (without using the 'label trick' [69]) CL setting, which demonstrates its consistently competitive performance to existing task-agnostic CL methods. Interestingly, VariGrow even achieves comparable performances to task-aware counterparts.

2.2.1 Related Work

Continual Learning: Continual learning models aim to learn new knowledge without catastrophically forgetting previously learned information. Methods in this domain can be broadly categorized into three classes: 1) *memory-based* methods which store a subset of raw data or build a generative model to generate synthetic data for replay [57, 60, 83, 84], 2) *regularization-based* methods which focus on preserving old information when learning new ones by penalizing drastic changes to a model's parameters [56, 58, 85, 86], and 3) *expansion-based* methods which grow and assign new model components for different tasks, keeping unrelated model parameters fixed. The expansion can be based on neurons [87], layers [59, 88], or independent networks [62]. Most CL methods require the task information during training and/or testing. For example, in the *multi-head* setting, models would only need to predict among the classes in one task, instead of the whole class set [73].

Task-agnostic continual learning: In many real-world applications, the current task information is usually not given [65, 89]. Some methods proposed to tackle the task-agnostic setting, but



(a) (b)

Figure 2.7: VariGrow schematic: **(a)**: The dynamically growing construct illustrated for two expert components. The input $\boldsymbol{x}$ is passed into a Bayesian Neural Network f_1 with weights $\boldsymbol{\omega}_1 \sim q_\phi(\boldsymbol{\omega}_1)$ multiplied by a binary mask $\boldsymbol{m}_1 \sim q_\phi(\boldsymbol{m}_1)$, sparsifying the architecture. The output is used to compute an energy-based novelty score ψ_ϕ^1. As $\psi_\phi^1(\boldsymbol{x})$ exceeds a threshold α, VariGrow expands and creates a second mixture component. The novelty scores are then used to construct the mixing distribution and sample $z \sim q_\phi(z|x)$, determining which expert component is used to compute $p(y|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{z})$. Through differentiable reparameterizations and approximations, gradient-based optimization can be performed to learn the variational parameters that optimize the Evidence Lower Bound (ELBO). **(b)**: The graphical model of the nonparametric distribution that we approximate, consisting of mixture assignments z_n for each data point according to prior probability $p(z_n = k) = v_k$, and a mixture distribution where the expert parameters $\boldsymbol{w}_k = \{\boldsymbol{\omega}_k, \boldsymbol{m}_k\}$ are sampled from. Reprinted/adapted with permission from [2].

only during testing [62, 73, 90, 91]. There are recently developed methods assuming that task information is not given during training [65, 69, 92] but their performances are much lower compared to their task-aware counterparts. Furthermore, the model training in these methods have various drawbacks. [91] assumes that data in one batch comes from a single context (task), and assume that task labels are available during training. The training of UCB [92] is extremely slow due to their modified backpropagation formulation. In [65], CN-DPM has the least assumptions on the data stream; however, their method requires performing density estimation through generative modeling, which can be intractable and unstable [65, 93, 94, 95], causing their performance to be significantly lower than task-aware CL methods.

Variational Inference for CL with Anomaly Detection: Our VariGrow is motivated by the energy-based model (EBM), which maps an input to a single, non-probabilistic scalar called *energy* [81, 82]. This energy score has shown to outperform the softmax confidence score for OoD detection [39]. Some other works on OoD detection either develop deep generative models [95], unify probabilistic and non-probabilistic models [96], or add background classes to enhance OoD detection [97]. [98] analyzed non-stationary data using Bayesian neural networks and memory-based online variational Bayes by implementing 'Bayesian forgetting' to selectively forget knowledge not relevant to the current data distribution. [77] proposed a hierarchical Indian Buffet process (IBP) to allocate resources when learning new tasks. However, training would still require task information for online inference. [69] proposed Bayesian Gradient Descent to train neural networks, claiming that their closed-form update rule better fits task-agnostic training. However, a 'label trick' was used to implicitly infer new tasks from novel labels. [99] proposed VCL, a variational Bayesian interpretation of CL using exemplar data points and a KL divergence penalty to retain previous information. But VCL is a *multi-head* formulation and requires task labels both during training and testing. [89] proposed using likelihood-based mixture models to handle the multi-modality of the different tasks. However, likelihood-based models fail on complex datasets and often assign higher likelihoods to OoD data [95]. For this, they resort to use a pretrained model to extract features for more complex datasets such as CIFAR100 [100] and ImageNet [101].

2.2.2 Methodology

Let $\{\mathcal{D}_t\}_{i=t}^{T}$ be a stream of datasets with each $\mathcal{D}_t$ having input-output pairs $(\boldsymbol{x}, y)$. Bayesian learning places a prior distribution $p(\boldsymbol{\theta})$ on the model parameters $\boldsymbol{\theta}$. In continual learning (CL), the posterior distribution after observing $t+1$ datasets is obtained using Bayes' rule:

$$p(\boldsymbol{\theta}|\mathcal{D}_{1:t+1}) \propto p(\mathcal{D}_{t+1}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D}_{1:t}). \tag{2.5}$$

Here, the posterior obtained in the previous step t is treated as a prior for the current step $t+1$. As we observe more novel data points, the complexity of the evolving posterior given our dataset increases. Hence it is important that our model scales accordingly [102].

2.2.2.1 VariGrow

To this end, we design a dynamically growing model, VariGrow, parameterized as follows:

$$\begin{aligned} p(\boldsymbol{w}, \boldsymbol{z}|\mathcal{D}_{1:t+1}) &\propto p(\mathcal{D}_{t+1}|\boldsymbol{w}, \boldsymbol{z}_{t+1})p(\boldsymbol{w}, \boldsymbol{z}|\mathcal{D}_{1:t}), \\ p(\boldsymbol{w}, \boldsymbol{z}|\mathcal{D}_{1:t}) &= \prod_{i=1}^{t} p(\boldsymbol{w}_{\boldsymbol{z}_i}|\mathcal{D}_i)p(\boldsymbol{z}|\mathcal{D}_{1:t}). \end{aligned}$$

Here, $\boldsymbol{w}$ denote the parameters of an expert module such as a neural network, while $\boldsymbol{z}$ determines which expert mixture component $p(\boldsymbol{w}_{\boldsymbol{z}})$ to sample $\boldsymbol{w}$ from. This mixing strategy naturally enriches the model representation capacity when needed for continual learning. The schematic is shown in Figure. 2.7. When training with large and complex datasets, the posterior distribution is intractable and is typically approximated [103]. It is important that the distributions we use to approximate the posterior via variational inference are flexible and expressive. For CL in particular, one must ensure that these variational distributions can be robustly updated without requiring the access to previously observed datasets [99]. In our CL settings, one would expect to grow more components as we sequentially observe more data from novel tasks. There could be infinitely many expert mixture components, presenting additional challenges in inference [102]. To address the mentioned challenges, we define the following variational approximation to the above posterior:

$$q_{\phi}(\boldsymbol{w}, \boldsymbol{z}|\boldsymbol{x}) = \prod_{i=1}^{t} q_{\phi}(\boldsymbol{w}_{\boldsymbol{z}_i})q_{\phi}(\boldsymbol{z}|\boldsymbol{x}). \tag{2.6}$$

To obtain an ideal approximate solution, it is crucial that both $q_{\phi}(\boldsymbol{z}|\boldsymbol{x})$ and $q_{\phi}(\boldsymbol{w}_{\boldsymbol{z}})$ are expressive and flexible. To this end, we will define these distributions implicitly. Also, note that we make $q_{\phi}(\boldsymbol{z}|\boldsymbol{x})$ covariate-dependent, allowing us to assign individual data points to any mixture component. To deal with the potentially infinite number of expert modules, we can define $q_{\phi}(\boldsymbol{z}|\boldsymbol{x})$ through a set of K expert components, while the other mixture components can be defined in relation to these main components. We describe these two distributions in detail in the following subsections.

For the moment, let us assume that these two distributions are given. Optimizing the variational parameters $\boldsymbol{\phi}$ corresponds to minimizing the negative ELBO at each CL step t:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}_{t+1}) &= \mathbb{E}_{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w},\boldsymbol{z}|\boldsymbol{x})}[-\log p(\mathcal{D}_{t+1}|\boldsymbol{w}_{\boldsymbol{z}_{t+1}})] \\ &+ \mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w},\boldsymbol{z}|\boldsymbol{x}) \parallel q_{\boldsymbol{\phi}_t}(\boldsymbol{w},\boldsymbol{z}|\boldsymbol{x})).\end{aligned} \tag{2.7}$$

The expectation can be approximated using a single sample of $(\boldsymbol{w}, \boldsymbol{z}_{t+1}) \sim q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w},\boldsymbol{z}|\boldsymbol{x})$, and $\log p(\mathcal{D}_{t+1}|\boldsymbol{w}_{K+1}) = N\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{t+1}}[\log p(y|\boldsymbol{x},\boldsymbol{w}_{K+1})]$ can be approximated using minibatches, with N being the number of data points in $\mathcal{D}_{t+1}$. When a novel task is detected by our variational formulation, $\boldsymbol{z}_{t+1} > K$, and a new expert mixture component $K+1$ is created. So the ELBO becomes

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}_{t+1}) &= \mathbb{E}_{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w},\boldsymbol{z}|\boldsymbol{x})}[-\log p(\mathcal{D}_{t+1}|\boldsymbol{w}_{K+1})] \\ &+ \mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w}_{K+1}) \parallel p(\boldsymbol{w})) \\ &+ \mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}|\boldsymbol{x}) \parallel q_{\boldsymbol{\phi}_t}(\boldsymbol{z}|\boldsymbol{x})).\end{aligned} \tag{2.8}$$

Here, since $K+1$ indicates a new mixture component $q_{\boldsymbol{\phi}_t}(\boldsymbol{w}_{K+1}) = p(\boldsymbol{w})$, where $p(\boldsymbol{w})$ is the prior distribution on the expert parameters. Meanwhile when we observe data points assigned to an existing mixture component, $\boldsymbol{z}_{t+1} = k \in \{1,\ldots,K\}$, we have:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}_{t+1}) &= \mathbb{E}_{q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w},\boldsymbol{z}|\boldsymbol{x})}[-\log p(\mathcal{D}_{t+1}|\boldsymbol{w}_k)] \\ &+ n_k\mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{w}_k) \parallel q_{\boldsymbol{\phi}_t}(\boldsymbol{w}_k)) \\ &+ \mathrm{KL}(q_{\boldsymbol{\phi}_{t+1}}(\boldsymbol{z}|\boldsymbol{x}) \parallel q_{\boldsymbol{\phi}_t}(\boldsymbol{z}|\boldsymbol{x})),\end{aligned} \tag{2.9}$$

where n_k is the number of data points previously assigned to expert k. Intuitively, as more data points are assigned to expert k, we would expect the expert distribution $q_{\boldsymbol{\phi}}(\boldsymbol{w}_k)$ to approach the true corresponding posterior. Meanwhile, for new expert components, only a prior distribution is given, and the component is free to learn from novel data.

To evaluate and optimize the ELBO above, it is important that we define our variational distributions such that the KL terms defined above are easily computable, plus being flexible and expressive. So we adopt an energy-based mixing construct for the variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and define the expert weight distribution implicitly.

2.2.2.2 Energy-based Mixing Distribution

Here, we describe our specification for the expert mixing distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. By Bayes' rule, we have

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \frac{q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})q_{\boldsymbol{\phi}}(\boldsymbol{z})}{\sum_{i=1}^{\infty} q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z}=i)q_{\boldsymbol{\phi}}(\boldsymbol{z}=i)}. \tag{2.10}$$

Although one would typically find $q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})$ through density estimation, such as an agnostic method CN-DPM [65], this involves the difficult optimization process of training generative models which can be intractable and unstable to perform in practice [93, 94, 95]. Instead of relying on density estimation for $q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})$ we interpret it as an energy-based score function as in [82].

In energy-based models, the system is optimized such that $\boldsymbol{x}$ belonging to a particular mixture component k will have low free energy $\psi_{\boldsymbol{\phi}}^{k}(\boldsymbol{x})$ relative to component k [81, 82]. For example, in classification problems, the *Helmholtz free energy* relative to component k can be written w.r.t. the log-posterior predictive distribution $\ell_{\boldsymbol{\phi}}^{k}(\boldsymbol{x}, c) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{w}|\boldsymbol{z}=k)}[\log p(y=c|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{z}=k)]$:

$$\psi_{\boldsymbol{\phi}}^{k}(\boldsymbol{x}) = -T \log \left(\sum_{c=1}^{C} \exp(\ell_{\boldsymbol{\phi}}^{k}(\boldsymbol{x}, c)/T) \right).$$

Here, C is all the known classes until current CL step, and T a temperature parameter of the free energy of component k. Note that $\ell_{\boldsymbol{\phi}}^{k}(\boldsymbol{x}, c)$ can be estimated using a single sample of $\boldsymbol{w}$. Similar energy functions can be derived for other tasks [81]. We would expect higher energy for data points $\boldsymbol{x}$ not belonging to component k, allowing us to assign data points into their respective mixture components. Specifically, for $k \in \{1, \ldots, K\}$, we have

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}=k|\boldsymbol{x}) = \frac{\exp\left(-\psi_{\boldsymbol{\phi}}^{k}(\boldsymbol{x})\right)}{\sum_{i=1}^{K} \exp\left(-\psi_{\boldsymbol{\phi}}^{i}(\boldsymbol{x})\right) + e^{-\alpha}},$$

where α is a parameter controlling the concentration of the mixture components. Meanwhile, for

$k > K$, we have

$$q_{\phi}(\boldsymbol{z} = k|\boldsymbol{x}) = \frac{e^{-\alpha}}{2^{k-K}\big(\sum_{i=1}^{K} \exp\left(-\psi_{\phi}^{i}(\boldsymbol{x})\right) + e^{-\alpha}\big)}.$$

Note that $q_{\phi}(\boldsymbol{z} > K|\boldsymbol{x}) = e^{-\alpha}/(\sum_{i=1}^{K} \exp\left(-\psi_{\phi}^{i}(\boldsymbol{x})\right) + e^{-\alpha})$. Here, the number of active mixture components K can dynamically grow as more data come in. Specifically, when $q_{\phi}(\boldsymbol{z} > K|\boldsymbol{x}) > q_{\phi}(\boldsymbol{z}_i|\boldsymbol{x}), \forall i \in \{1, \dots, K\}$, we can allocate a new mixture component, growing a new expert module that implicitly defines this distribution. From an energy-based perspective, α can be seen as the average energy of all data points in the system. Such energy-based novelty scores allow us to identify and handle novel data points in a Bayesian fashion.

Having specified our mixing distribution, we now show how to approximate its KL term. Instead of computing the KL term of this distribution directly, we derive a tractable upper bound to the KL term. Specifically, we can divide the KL term into two parts as follows:

$$\begin{aligned} &\mathrm{KL}(q_{\phi_{t+1}}(\boldsymbol{z}|\boldsymbol{x})||q_{\phi_t}(\boldsymbol{z}|\boldsymbol{x})) = \\ &\qquad \sum_{i=1}^{K} q_{\phi_{t+1}}(\boldsymbol{z} = i|\boldsymbol{x}) \log \frac{q_{\phi_{t+1}}(\boldsymbol{z} = i|\boldsymbol{x})}{q_{\phi_t}(\boldsymbol{z} = i|\boldsymbol{x})} \\ &\qquad + \sum_{i=K+1}^{\infty} q_{\phi_{t+1}}(\boldsymbol{z} = i|\boldsymbol{x}) \log \frac{q_{\phi_{t+1}}(\boldsymbol{z} = i|\boldsymbol{x})}{q_{\phi_t}(\boldsymbol{z} = i|\boldsymbol{x})}. \end{aligned} \tag{2.11}$$

Then, by applying Jensen's inequality to the second term:

$$\begin{aligned} &\mathrm{KL}(q_{\phi_{t+1}}(\boldsymbol{z}|\boldsymbol{x})||q_{\phi_t}(\boldsymbol{z}|\boldsymbol{x})) \leq \\ &\qquad \sum_{i=1}^{K} q_{\phi_{t+1}}(\boldsymbol{z} = i|\boldsymbol{x}) \log \frac{q_{\phi_{t+1}}(\boldsymbol{z} = i|\boldsymbol{x})}{q_{\phi_t}(\boldsymbol{z} = i|\boldsymbol{x})} \\ &\qquad + q_{\phi_{t+1}}(\boldsymbol{z} > K|\boldsymbol{x}) \log \frac{q_{\phi_{t+1}}(\boldsymbol{z} > K|\boldsymbol{x})}{q_{\phi_t}(\boldsymbol{z} > K|\boldsymbol{x})} \end{aligned} \tag{2.12}$$

In other words, by defining the following $K + 1$ categorical distribution $q'_{\phi}(\boldsymbol{z}|\boldsymbol{x})$:

$$q'_{\phi}(\boldsymbol{z} = k|\boldsymbol{x}) = q_{\phi}(\boldsymbol{z} = k|\boldsymbol{x}), \;\; k \in \{1, \dots, K\} \tag{2.13}$$

Table 2.4: Results on CIFAR100-B0 benchmark (averaged over three runs). Parameters are counted by millions. *Dashes indicate results were not reported by the authors. Reprinted/adapted with permission from [2].

Method	5 Steps		10 Steps		20 Steps		50 Steps	
	Params.	Acc. (%)	Params.	Acc. (%)	Params.	Acc. (%)	Params.	Acc. (%)
Bound	11.2	80.40	11.2	80.41	11.2	81.49	11.2	81.74
iCaRL [83]	11.2	71.14	11.2	65.27	11.2	61.20	11.2	56.08
UCIR [104]	11.2	62.77	11.2	58.66	11.2	58.17	11.2	56.86
BiC [104]	11.2	73.10	11.2	68.80	11.2	66.48	11.2	62.09
WA [105]	11.2	72.81	11.2	69.46	11.2	67.33	11.2	64.32
PODNet [106]	11.2	66.70	11.2	58.03	11.2	53.97	11.2	51.19
AANets [63]	11.2	67.59	11.2	65.66	-	-	-	-
RPSNet [64]	60.6	70.50	56.5	68.60	-	-	-	-
DER [107]	2.89	75.55	4.96	74.64	7.21	73.98	10.15	72.05
CN-DPM [65] (Agnostic)	19.2	20.34	19.2	17.60	19.2	18.79	19.2	19.70
VariGrow (Agnostic)	2.97	75.50	4.88	75.04	7.30	74.03	10.25	72.21

$$q'_{\phi}(\boldsymbol{z} = K + 1|\boldsymbol{x}) = q_{\phi}(\boldsymbol{z} > K|\boldsymbol{x}), \tag{2.14}$$

we can use the KL divergence of this distribution as a tractable upper bound to the KL divergence of our original distribution:

$$\mathrm{KL}(q_{\phi_{t+1}}(\boldsymbol{z}|\boldsymbol{x})||q_{\phi_t}(\boldsymbol{z}|\boldsymbol{x})) \leq \mathrm{KL}(q'_{\phi_{t+1}}(\boldsymbol{z}|\boldsymbol{x})||q'_{\phi_t}(\boldsymbol{z}|\boldsymbol{x})).$$

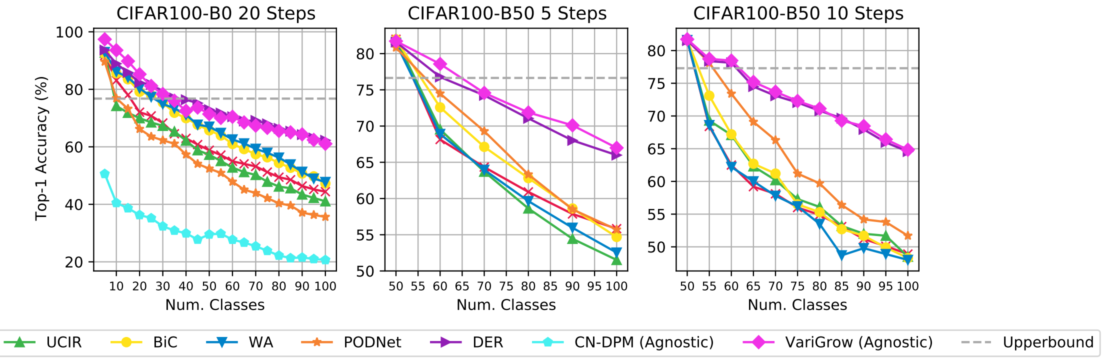


Figure 2.8: Class-incremental performance comparisons at each step for the CIFAR-100 dataset. Reprinted/adapted with permission from [2].

Table 2.5: Results on CIFAR100-B50 (averaged over three runs). Parameters are counted by millions. Reprinted/adapted with permission from [2].

Method	2 Steps		5 Steps		10 Steps	
	Params.	Acc. (%)	Params.	Acc. (%)	Params.	Acc. (%)
Bound	11.2	77.22	11.2	79.89	11.2	79.91
iCaRL [83]	11.2	71.33	11.2	65.06	11.2	58.59
UCIR [104]	11.2	67.21	11.2	64.28	11.2	59.92
BiC [104]	11.2	72.47	11.2	66.62	11.2	60.25
WA [105]	11.2	71.43	11.2	64.01	11.2	57.86
PODNet [106]	11.2	71.30	11.2	67.25	11.2	64.04
DER [107]	3.90	74.57	6.13	72.60	8.79	72.45
VariGrow (Agnostic)	**3.63**	**74.62**	**6.01**	**73.97**	**8.55**	**72.75**

2.2.3 Experiments

In this subsection, we conduct extensive experiments to validate the effectiveness of VariGrow. We evaluate our method on 3 datasets: CIFAR-100 [83], ImageNet-100 [83], and ImageNet-1000 [83], using two commonly used benchmark protocols. After detailing our experimental setups and implementation details in subsection 2.2.3.1, we present and discuss experimental results on the CIFAR-100 dataset and both ImageNet-100 and ImageNet-1000 datasets in subsections 2.2.3.2 and 2.2.3.3.

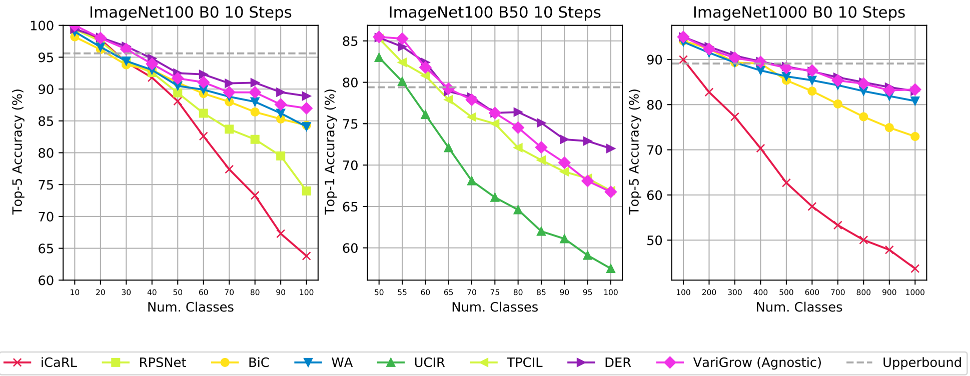


Figure 2.9: Class-incremental performance comparisons at each step for the ImageNet-100 and ImageNet-1000. Reprinted/adapted with permission from [2].

Table 2.6: Results on ImageNet-B0 (averaged over three runs). Parameters are counted by millions. Avg is the average accuracy (%) over steps. Last is the accuracy (%) evaluate on each task for the model at the last incremental step. *Dashes indicate results were not reported. Reprinted/adapted with permission from [2].

Method	ImageNet100-B0					ImageNet1000-B0				
	Params.	Top-1		Top-5		Params.	Top-1		Top-5	
		Avg	Last	Avg	Last		Avg	Last	Avg	Last
Bound	11.2	-	-	-	95.1	11.2	89.27	-	-	-
iCaRL [83]	11.2	-	-	83.6	63.8	11.2	38.4	22.7	63.7	44.0
BiC [104]	11.2	-	-	90.6	84.4	11.2	-	-	84.0	73.2
WA [105]	11.2	-	-	91.0	84.1	11.2	65.67	55.6	86.6	81.1
RPSNet [64]	-	-	-	87.9	74.0	-	-	-	-	-
AANets [63]	11.2	75.58	-	-	-	11.2	64.85	-	-	-
DER [107]	7.67	76.12	66.06	92.79	88.38	14.52	66.73	58.62	87.08	81.89
VariGrow (Agnostic)	7.82	76.04	65.87	92.51	88.17	14.89	66.58	58.47	86.88	81.70

2.2.3.1 Experimental Setups

Datasets: CIFAR-100 [100] consists of 60,000 32x32 pixel color images ranging over 100 classes. The dataset is divided into 50,000 training images with 500 images per class, and 10,000 images for evaluation with 100 images per class. ImageNet-1000 [101] is a large-scale dataset consisting of 1,000 classes, including about 1.2 million RGB images for training and 50,000 images for validation. ImageNet-100 [62, 83, 104] is a subset of it by selecting 100 classes from the ImageNet-1000 dataset.

Benchmark Protocols: For the **CIFAR-100**, we test our methods on two widely used protocols: 1) CIFAR100-B0 [62, 83]: a protocol which divides all 100 classes into 5, 10, 20, and 50 incremental steps with a fixed memory size of 2,000 exemplars over batches; 2) CIFAR100-B50 [62, 104]: a protocol which starts from a model trained on 50 classes, while the remaining 50 classes are divided into splits of 2, 5, and 10 incremental steps with 20 examples as memory per class. We compare the top-1 average incremental accuracy, which takes the average of the accuracy for each step. We follow similar protocols for **ImageNet-100**: 1) ImageNet100-B0 [62, 83]: the protocol trains the model in batches of 10 classes from scratch with a fixed memory size 2,000 over batches; 2) ImageNet100-B50 [62, 104]: the protocol starts from a model trained on 50 classes while the remaining 50 classes come in 10 steps with 20 exemplars per class. For fair comparisons, we use the same ImageNet subset and class order done by [83], [104], and [62]. For **ImageNet-1000**, we evaluate our method on the ImageNet1000-B0 benchmark [62, 83], that trains the model in batches of 100 classes with 10 steps in total and set a fixed memory size as 20,000 exemplars, with the same class order by [83] for ImageNet-1000. For both ImageNet-100 and ImageNet-1000, we compare the top-1 and top-5 average incremental accuracy, as well as the last step accuracy. For the task-agnostic setting during *training* and *testing* we hide task IDs, which is called *single-head* setting. The task-aware setting (i.e. *multi-head*) is using one prediction head for each task and effectively predicting the labels within a task instead the whole label pool. For baselines, we compare against various state-of-the-art (i) **task-aware** CL methods: iCaRL [83] is memory-based that picks exemplars by balancing the number of class labels. UCIR [104], uses normalized feature vectors for prediction. BiC [104] trains a bias correction layer on a validation set. WA [105] corrects biased weights by aligning the norm of the weight vectors of new classes to weight vectors of old classes. PODNet [106] uses a spatial distillation loss penalizing parameter changes. TPCIL [108] attempts to preserve the topology of the latent feature space, AANets [63] attempt to solve the stability-plasticity dilemma by proposing stable and plastic blocks, RPSNet [64] is a path selection algorithm that progressively chooses optimal paths as sub-network for the new classes. DER [107] dynamically grows the network using given task labels. We also benchmark against (ii) **task-agnostic** CL methods: CN-DPM [65], a hybrid expansion- and memory-based method, and UCB [92], a regularization-based BNN model. We were not able to reproducible UCB's results reliably, with an accuracy of only 40.34% on the CIFAR10/100 testing protocol. One agnostic method [68] was not considered due to the capacity can only handle smaller dataset.

Implementation Details: For all datasets, we adopt ResNet-18 [109] as the architecture of our expert modules, following RPSNet [64] and DER [62]. We run experiments on three different class orders and report the average of the results. In these experiments, we treat the exemplars variationally, following [99] and select new exemplars (i.e. coreset) as novel mixture components are encountered based on the herding selection strategy [110]. We also use these exemplars along with OoD datasets to further calibrate our energy-based novelty score, following the selection of [82].

We use Tiny-ImageNet [111] and LSUN [112] as OoD datasets for CIFAR-100 and ImageNet experiments respectively. We perform MAP estimation of the neural network weights $\boldsymbol{\omega}$ using Gaussian priors, equivalent to adding a 5×10^{-4} weight decay coefficient. We set $\lambda = 1$ for the prior distribution of $\boldsymbol{m}$, and set $T = 1$, and $\alpha = 18$ for the energy-based novelty scores. We optimize our formulation using SGD with a learning rate of 0.1, batch size of 128 for CIFAR-100, and 256 for ImageNet. We train for 120 epochs and decay the learning rate by 0.1 after 30, 60, and 90 epochs.

2.2.3.2 Evaluation on CIFAR100

Quantitative Results: Table 2.4 and Figure 2.8 (left) show the results for CIFAR100-B0. We can see that, without needing task labels nor task switching information, our method is competitive with state-of-the-art CL methods which are task-aware. Meanwhile, ours significantly outperforms CN-DPM, a task-agnostic online learning formulation, with an improvement of over +50%. Moreover, the margin between our method and CN-DPM continuously increases, indicating that our method performs better over longer continual learning episodes with fewer parameters with our sparsification. Note also that we are getting very close to the offline multi-task learning baseline (Bound). This demonstrates that VariGrow is able to learn from a non-*iid* data stream without much decrease in performance despite not having access to the entire dataset.

We further compare the performance of VariGrow on the CIFAR100-B50 benchmark in Table 2.5 and Figure 2.8 (middle, right), again showing that our method is competitive with task-aware continual learning methods. We note that DER is the most competitive task-aware method in our benchmarks but it has to grow the network architecture with given task switching information.

To further banchmark the efficacy of our method in handling task-agnosticism, we study the effects of two settings where task-agnosticism can occur. One setting involves removing the clear task boundaries, and instead gradually introducing data from novel tasks. This setting is similar to the fuzzy task boundary experiment conducted by [65]. The other setting assumes that data from previous tasks can be observed again in between task switches. Thus, our model needs to be able

Table 2.7: Results on ImageNet-B50 (averaged over three runs). Parameters are counted by millions. *Dashes indicate results were not reported by the authors. Reprinted/adapted with permission from [2].

Method	ImageNet100-B50				
	Params.	Top-1		Top-5	
		Avg	Last	Avg	Last
Bound	11.2	81.20	81.5	-	-
UCIR [104]	11.2	68.09	57.3	-	-
PODNet [106]	11.2	74.33	-	-	-
TPCIL [108]	11.2	74.81	66.91	-	-
DER [107]	8.87	77.73	72.06	94.01	91.64
VariGrow (Agnostic)	8.94	77.64	71.48	92.84	89.95

to distinguish these instances and correctly assign them to an existing expert instead of growing a new one. We denote both of these experiments as fuzzy and lookback respectively, and the performance of our method in these settings on the CIFAR-100 dataset can be seen in Table 2.8. As seen in Table 2.8, our method suffers only a slight degradation in accuracy in these settings compared to the traditional setting. We hypothesize that this is caused by stray datapoints being incorrectly assigned to the wrong expert.

Qualitative view on energy-based Novelty Score: We show the energy-based novelty score at each timestep for the CIFAR100-B50 10-step protocol in Figure 2.10. Here, we see that our novelty score clearly helps detect task changes, with significantly increased novelty scores after a new task is observed. Note also that VariGrow is able to correctly detect that there are 10 tasks with 10 observed peaks.

Table 2.8: Results on different task-agnostic settings on the CIFAR100-B50 benchmark. Reprinted/adapted with permission from [2].

Setting	Accuracy (%)	
	5 Steps	10 Steps
Baseline	73.97	72.45
Lookback Old Tasks	71.21	70.98
Fuzzy Boundaries	70.03	69.19

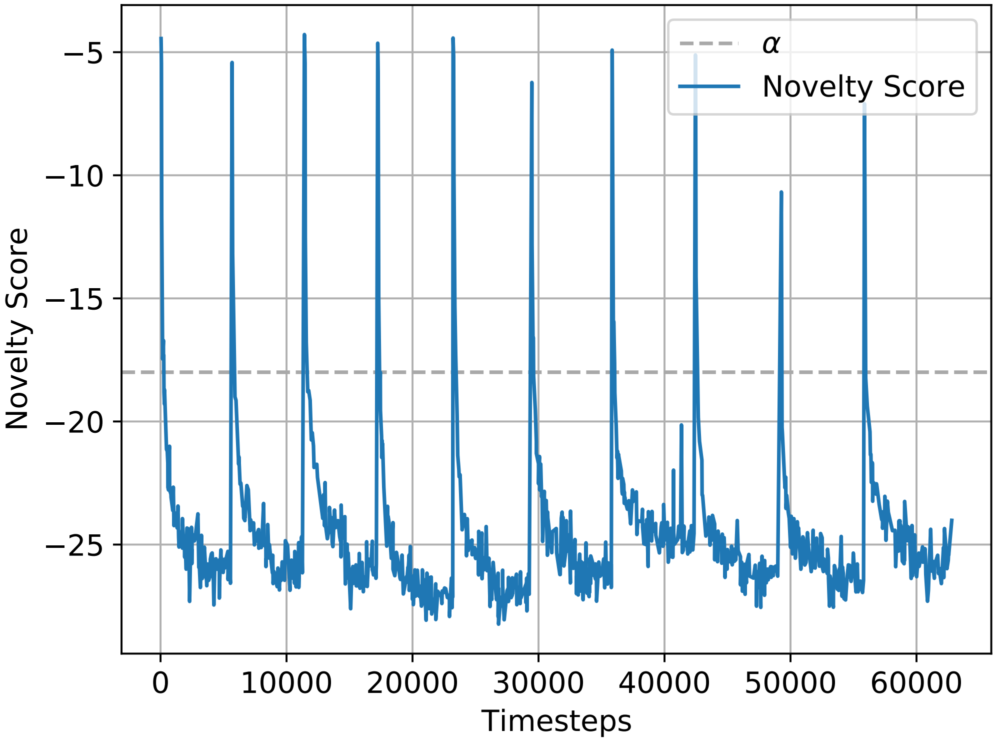


Figure 2.10: Novelty score values for each iteration on the CIFAR100-B50 10-step protocol. Reprinted/adapted with permission from [2].

2.2.3.3 Evaluation on ImageNet

We show results for the ImageNet-100 and ImageNet-1000 datasets in Tables 2.6, 2.7, and Figure 2.9. We see that our VariGrow is again competitive with task-aware methods for all splits on these two datasets, which are more complex compared to CIFAR100. We note that the gap in top-5 accuracy is smaller. We believe that this is because the top-5 accuracy is more tolerant to slightly inaccurate predictions and thus less sensitive to catastrophic forgetting.

3. SECOND AIM: DIVIDING AND CONQUERING HETEROGENEITY

3.1 Sparse Gated Mixture-of-Experts to Separate and Interpret Patient Heterogeneity in EHR data[1]

3.1.1 Introduction

3.1.1.1 Overview

A challenge in developing machine learning models for patient risk prediction involves addressing patient heterogeneity and interpreting the model outcome in clinical settings. Patient heterogeneity manifests as clinical differences among patients in observational datasets, e.g. in electronic health records (EHRs), leading to latent homogeneous patient subgroups. The discovery of such subgroups is helpful in precision medicine, where different risk factors from different patient would contribute differently to disease development and thus personalized treatment. Therefore the model development to account for patient heterogeneity requires a 'divide-and-conquer' mechanism. In medical terms, *subtypes* of patients have been critical in clinical decision making, but finding meaningful *subtypes* of patients is not always obvious from traditional 'one-size-fits-all' models. Furthermore the *subtypes* discovered should be able to be interpreted by the corresponding part of the model. In this paper, we use a Mixture-of-Experts (MoE) model, coupled with a sparse gating network, to handle patient heterogeneity for prediction and to aid interpretation of patient *subtype* separation. Different from traditional sparsity imposed on neuron activation in neural networks, we propose an expert-wise sparsity, leading to better macroscopic training footprints for modeling and interpretation. We demonstrate the performance of this technique in a publicly available intensive care unit (ICU) dataset, MIMIC-III, both through risk prediction metrics and evaluation on interpretability.

Patient heterogeneity in real-world clinical data poses potential challenges in actualizing ma-

[1] Reprinted with permission from Huo, Z., Zhang, L., Khera, R., Huang, S., Qian, X., Wang, Z. and Mortazavi, B.J., 2021, July. Sparse Gated Mixture-of-Experts to Separate and Interpret Patient Heterogeneity in EHR data. In 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI) (pp. 1-4). IEEE. Copyright 2021 by the authors.

chine learning models and further complicates clinical decision making [113]. For example, in evaluating risk modeling, two patients with the exact same risk score from the model might have different underlying driving factors, potentially rendering completely different treatments [114]. Electronic health records (EHRs) are an example of a dataset with vast, heterogeneous clinical records that contain intra- and inter-variability of measurements and treatments across patients. To date, many deep learning techniques have begun exploring this data, often within subsets of clinical questions or data type, for example deep embedding of patient representations [115], time series modeling of structured clinical data with recurrent neural networks [116, 117], and feature extraction techniques with convolutional neural networks [118]. However, these methods often depend on a single model with 'one-size-fits-all' approach. An opportunity exists to stratify patients heterogeneity and generate explainable risk prediction from model-generated separation of *subtypes*, which may aid in clinical decision making in response to risk model outputs.

In order to stratify the patient population into specific patient subtypes (i.e. homogeneous subgroups) through a learnable 'divide-and-conquer' mechanism [119], we employ a mixture-of-experts (MoE) model. As one realization of ensemble learning [120], traditional MoE models generally assume a model containing a mixture of Gaussian experts. With the advancement of deep learning, the Gaussian experts have been replaced by neural networks in many applications with more complex architectures and larger learning capacity [1, 28]. However, such a larger feature space and bigger cohort for these models may be difficult to interpret for clinical decision making. Unlike LASSO for linear model interpretation [121], the introduction of sparsity to deep learning models, e.g. dropout [122] and sparse gating [28], traditionally aims to prevent overfitting or alleviate the computation overhead. We propose that while all these methods with sparsity achieve strong predictive performance in their tasks, sparsity is not responsible for separating latent groups and reducing redundancy among different parts of the model for explaining heterogeneity. Therefore, enhancing these techniques with added interpretation of the patient subgroups is needed for successfully applying MoE model to clinical settings.

This work proposes an MoE framework with a macroscopic-style sparse gating network to address patient heterogeneity and provide interpretation to aid in clinical decision making, while improving risk prediction. To enable automatic subtyping of patients, the sparse gating, when coupled with an MoE framework, is to monitor the training activation of each expert rather than individual neuron. The sparse gating network is a realization of *model sparsity*, extending the concept of conditional computation [123], which has a trainable activation mechanism based on input examples. Compared to *dropout*, this *model sparsity* better explains the derived prediction in MoE scenarios, due to the fact that the sparse instance-based activation constrains each expert rather than neuron to handle different subsets of training data (different subtypes) and thus better at interpretation. Because the gating network and expert networks are trained concurrently, the label information is embedded in the expert assignment in the gating network, making the representation learned for each patient supervised and informative. In the experiment, we study the derived patient representation through expert assignment, showing discovery of clinically meaningful and interpretable patient subtypes. We demonstrate that sparse MoE not only learns to 'divide-and-conquer' a real-world clinical dataset with vast patient heterogeneity, but also provides better interpretation than raw features or generic encoding on how the model derives latent groups for prediction.

3.1.1.2 Related work

Heterogeneity is an important issue when analyzing EHR data. [124] have proposed HeteroMed based on Heterogeneous Information Network [125], and address the heterogeneity problem of various diagnose sources (laboratory tests, measurements, etc.). Similarly, [126] analyzed the same aspect of heterogeneous EHR data by using a probabilistic graphical model. However, here we are addressing another level of EHR heterogeneity in terms of different patient cohorts, among which ICU stay lengths, lab measurement frequencies, and phenotypes may vary drastically in particular for MIMIC III. These types of heterogeneity require an instance-wise divide-and-conquer regime, as opposed to feature-wise method [127], which has not been found in traditional methods. MoE is a good fit for this problem because each expert can be assigned to a unique cohort (a subtype) for personalized training and prediction.

The interpretability analysis in clinical data usually focuses on modeling patient representation and finding similar subtypes to understanding the correlation of risk factors and mortality rate. Some methods have been proposed for this purpose, such as deep representations [115, 128, 129], sparse encoding [5], and prototype matching [130]. In our paper, we couple the model prediction and interpretation together, as opposed to post hoc methods, where the learned patient representations (discovered subtypes) are supervised, providing additional predictive power.

We use MoEs to address the problem of heterogeneity and interpretability in clinical data. MoEs have been widely used with strong capacity in healthcare and medicine, such as ubiquitous computation [1, 131], and health informatics [132, 133]. However, these methods are usually not capable of finding patient subtypes due to the fact that each expert's specialty is not guaranteed, for example, by a sparsity metric. When learning redundant information among the experts, it is hard to interpret which expert is responsible for which cohort. We impose expert-level sparsity on modeling MoE to handle clinical heterogeneity and interpretability.

3.1.2 Methodology

In order to develop a model that tackles patient heterogeneity and provides additional interpretability, we propose a sparse MoE model, as illustrated in Fig. 3.1. The sparse MoE leverages the divide-and-conquer strength of the general MoE framework, with the addition of a sparse gating network trained concurrently, deriving a parsimonious training regime, leading to a learned patient representation that naturally subtypes. Since the expert assignment representation for each patient is through supervised learning, it is more informative and predictive than general encoding.

3.1.2.1 MoE training process

An MoE model approaches the divide-and-conquer regime for highly specialized training [28]. By applying a gating mechanism to distribute data to different experts we can couple the training of the group of expert networks and gating networks concurrently [134]. The training of this model is often framed as a maximum likelihood estimation (MLE) problem. For this work, we consider

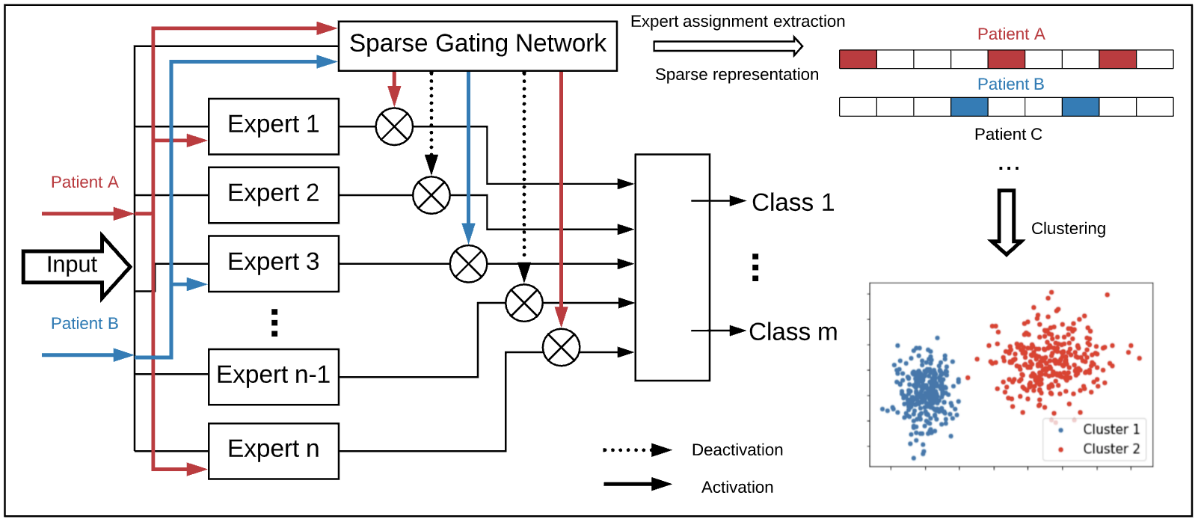


Figure 3.1: Sparse Mixture of Experts framework. On the left is risk prediction module with gating and expert network training concurrently, on the right is the sparse expert assignment module for interpretability. Reprinted/adapted with permission from [3].

maximizing the 'complete' log-likelihood [19, 135], by introducing the hidden variable $\mathbf{Z}$:

$$\begin{aligned} &\log L_c(\boldsymbol{\Theta}; \mathcal{D}) \propto \log P(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \Theta) \\ &= \sum_i \sum_j \mathbb{1}\{z_i = j\}\log\{g_j(\mathbf{x}_i, \mathbf{v})P(y|\mathbf{x}_i, \theta_j)\}, \end{aligned} \tag{3.1}$$

where $\boldsymbol{\Theta}$ is the model parameter set and $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ is the observational dataset. The function $g(\cdot)$ represents the gating network parameterized by $\mathbf{v}$, $\mathbb{1}\{\cdot\}$ denotes an indicator function and $P(y|\mathbf{x}, \theta_j)$ is the probability of an outcome given the input data as estimated by the selected expert. The hidden variable $\mathbf{Z} = \{z_1, ..., z_n\}$ is introduced as the indicator of expert training assignment [134], which allows for the gating function to be represented as:

$$g_j = P(z_i = j|\mathbf{x}_i) = \frac{e^{\beta_j}}{\sum_k e^{\beta_k}}, \tag{3.2}$$

where $\beta_j = \mathbf{v}_j^T \mathbf{x}_i,$, a multinomial logit representing the gating network's assignment z_i to expert j.

3.1.2.2 Parsimonious training of MoE

3.1.2.2.1 Sparse EM variant The aforementioned MLE problem for training MoE is usually solved by the Expectation-Maximization (EM) algorithm, which has many variants [135, 136]. However, previous work along this line did not adopt a parsimonious training regime [134, 136]. In this work, instead of feeding each data point (one patient's data) to all experts, we only activate a small group of experts for training each patient by imposing sparsity. The key idea is to assign each instance to experts parsimoniously, so that the one expert will be responsible for a small cohort of instances, providing a more localized estimation of outcomes and subtyping by evaluating the assignment of the data points to the individual experts. In order to accomplish this, we modify the standard EM algorithm to enforce sparsity in the assignment of patients to individual experts. This requires updating the EM steps incrementally (instead of a full expectation calculation in the E-step) to calculate a partial expectation. To accomplish this, we first apply a 'generalized' EM algorithm introduced by [137], where instead of a full maximization, they first redefine $L_c(\mathbf{\Theta}, \mathcal{D})$ as:

$$F(\tilde{P}(\mathbf{Z}), \mathbf{\Theta}) = \mathbb{E}_{\tilde{P}(\mathbf{Z})}[L_c(\mathbf{\Theta}; \mathcal{D})] + H(\tilde{P}(\mathbf{Z})), \tag{3.3}$$

where $H(\tilde{P}(\mathbf{Z})) = -\mathbb{E}_{\tilde{P}(\mathbf{Z})}[\log\tilde{P}(\mathbf{Z})]$ is the entropy of distribution $\tilde{P}(\mathbf{Z})$. The F function is defined over a specific value of the observed data $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$ and is guaranteed to be non-decreasing at each EM step. It is essential to note that $L_c(\mathbf{\Theta}, \mathcal{D})$ is a continuous function of $\mathbf{\Theta}$, from which one may assume that F is a continuous function of both $\mathbf{\Theta}$ and $\tilde{P}(\mathbf{Z})$. We will use this F function as a surrogate for the traditional Q function in EM algorithm. This modification allows for the sparse training of experts. For additional details of F in EM and the guarantee of the same global optimum as Q function, please see the supplement.

3.1.2.2.2 Modified gating function In order to use F, we adopt a modification of the gating function introduced in [135]:

$$g_j = P(z_i = j | \mathbf{x}_i) = \frac{\alpha_j e^{\beta_j}}{\sum_k \alpha_k e^{\beta_k}}, \tag{3.4}$$

where $\alpha_j \geq 0$ constructing the selection criterion for a data point's posterior to be updated in the E-step of the EM algorithm. In this work, we use the Top-k masking as the surrogate for α_j [28]:

$$\alpha_j = \text{Top-k}(\beta_j, k) = \begin{cases} 1 & \text{if } \beta_j \text{ in top } k \text{ value of all } \beta \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

A selection set for updating the posterior of z_i at the t-th E-step can be denoted by $S^{(t)}$. As noted in [28], Top-k masking seemingly imposes discontinuity in the output of the gating network; but in practice (and in our experiments), this does not pose an issue. However, to prevent the preference on an individual expert due to factors based upon initialization in early epochs, we adopt a load balance constraint on training the MoE:

$$L_{\text{load-balance}} = \text{CV}(\sum_j g_j(\mathbf{x})), \tag{3.6}$$

where CV stands for the coefficient of variant, also known as relative standard deviation (RSD). This constraint will force the gating network to have more evenly distributed expert assignments, thus avoiding training solely on favored experts. The final EM algorithm for training a sparse MoE, as well as enforcing a load-balanced sparse selection, is summarized as follows:

$$\begin{aligned} \textbf{E-step:} \quad & \text{Select one } z_i \in S^{(t)}\text{, so that} \\ & \tilde{P}_i^{(t)} = \arg\max_{\tilde{P}_i} F_i(\tilde{P}_i, \Theta^{(t-1)}). \\ & \text{Set the rest } \tilde{P}_j^{(t)} = \tilde{P}_j^{(t-1)} \text{ where } i \neq j \\ \textbf{M-step:} \quad & \Theta^{(t)} = \arg\max_{\Theta} F(\tilde{P}^{(t)}, \Theta) - \lambda L_{\text{load-balance}}, \end{aligned} \tag{3.7}$$

where λ is a tunable hyperparameter for the load balancing constraint. Selection set $S^{(t)}$ corresponds to the masking in Eq. (3.5).

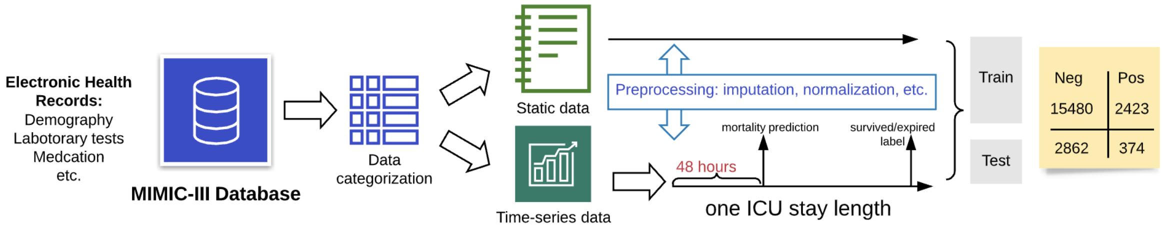


Figure 3.2: MIMIC-III data extraction pipeline. Reprinted/adapted with permission from [3].

3.1.2.3 Model specification of expert in MoE

The experts may be modeled in various forms in our method to train $P(y|\mathbf{x}_i, \theta_j)$ in Eq. (3.1), contrary to [28], whose expert layer only has MLPs. This enables us to train individual experts suitable for specific tasks, e.g. a convolutional neural network for computer vision, a recurrent neural network for natural language processing. For our EHR application, the data comes in the form of time series for each patient, therefore we adopt recurrent neural network architectures; we have chosen LSTM [138] for our expert framework. The gating network will first take the input $\mathbf{x}_i$ and output a vector of length n, which is the number of experts. Each component g_j in the vector is the probability of expert j being chosen. The Top-k masking will enforce activation of a parsimonious set of k experts, and thus deactivate the rest. This is corresponding to the E-step in Eq. (3.7). The activated expert will take in data point $\mathbf{x}_i$ and output $P(y|\mathbf{x}_i, \theta_j)$. In our time-series data we have T time stamps, i.e. $\{\mathbf{x}_i\}_{t>0}^{T}$, this will make LSTM have T-long sequence of hidden states $\{\mathbf{h}\}_{t>0}^{T}$:

$$
\begin{aligned}
i_t &= \sigma(\mathbf{x}_i^t W^{(xi)} + h_{t-1} W^{(hi)}), \\
f_t &= \sigma(\mathbf{x}_i^t W^{(xf)} + h_{t-1} W^{(hf)}), \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(\mathbf{x}_i^t W^{(xc)} + h_{t-1} W^{(hc)} + b^{(c)}), \\
o_t &= \sigma(\mathbf{x}_i^t W^{(xo)} + h_{t-1} W^{(ho)} + b^{(o)}), \\
h_t &= o_t \odot \sigma_h(c_t),
\end{aligned}
\tag{3.8}
$$

where the last time stamp of hidden state h_t is fed into a softmax function for the final probability $P(y|\mathbf{x}_i, \theta_j)$ which is the $\tilde{P}_i^{(t)}$ in Eq. (3.7). The single probability output is due to the fact that we are interested in many-to-one sequence-based prediction in medical settings, e.g. mortality prediction and phenotyping, rather than sequence-to-sequence prediction. The notations above carry the traditional LSTM convention, where i_t denotes the input gate, f_t for the forget gate, c_t for the cell state, and o_t for the output gate. So each expert's output will be determined as activated or deactivated by the sparse gating in Eq. (3.4). If activated, the parameters will be optimized through M-step in Eq. (3.7).

3.1.3 Experiments

Through the experiments in this subsection, we aim to illustrate that our sparse MoE model can tackle heterogeneous real-world clinical data for corresponding medical prediction for each individual patient, with good interpretability in terms of clinically meaningful patient stratification. We test our model on a publicly available EHR dataset, i.e. MIMIC-III dataset [139], in two tasks, which are mortality prediction and phenotyping. The mortality prediction is useful for risk factor control and preliminary prognosis, to eventually have better resource reallocation and patient outcomes in hospital. The phenotyping task is to classify patients by a group of separate binary labels, in which the same group of patients would share similar outcomes of interest (e.g. adverse events, commodities). Phenotyping can help personalized treatments and fine-grained diagnosis. We will use these two tasks to demonstrate the advantage of our sparse MoE in handling patient heterogeneity and interpretability, and further to show the interpretable patient stratification and subtyping.

3.1.3.1 MIMIC-III dataset

MIMIC-III is one of the largest clinical datasets that has been made publicly available. It contains multivariate time-series data from over 40,000 intensive care unit (ICU) stays. The types of data range from static demographics such as gender and age to rapidly changing measurements such as heart rate and arterial blood pressure. The heterogeneity is one of the major challenges when analyzing this dataset, due to the diverse patient health conditions, rapidly changing hazard ratio as well as the corresponding treatments. We focus on using only the first 48 hours after

ICU admission for the prediction of both tasks. The intuition is that for early risk prognosis and phenotyping, the precaution procedure can be undertaken, since the average ICU stay can be up to 100 to 200 hours [139]. The data extraction and experiment setup are shown in Fig. 3.2. We adopted the same data pre-processing steps as in a set of benchmark models in MIMIC-III (i.e. imputation, normalization, data masking, etc) [140]. We followed their splitting of whole dataset into training and testing sets of 17,903 and 3,236 instances, respectively. The positive (expired) and negative (survived) distribution is shown in Fig. 3.2.

3.1.3.2 Experimental setup

We design our experiments to answer the following practical questions:

- Can an ensemble method be better at 'divide-and-conquering' patient clinical heterogeneity to improve the risk prediction?
- Can *model sparsity* further improve the MoE model?
- Can the derived patient representation from proposed sparse MoE provide real-world clinical interpretability?

First, we will discuss our baselines in two categories, i.e. single network and ensemble methods, to answer the first question. For single network baselines, we first compare against a strong published baseline in [140], which is based on a bidirectional LSTM. Harutyunyan et al. also presented an improved baseline version (a channel-wise LSTM) for both mortality prediction and phenotyping, where each time-series feature is trained independently on the first layer and then concatenated in the second layer for all the features [140]. Another strong baseline for MIMIC-III mortality prediction is Deep Supervision [116], where the authors used target replication for the supervision of LSTM in each time stamp, and with changing loss function the model needed to predict replicated target variables along with outcome. Note these baselines are carried out with dropout tuning [116, 140], against which we compare the performance of our MoE with *model sparsity* as in the gating network. Furthermore, to design an ensemble method without sparsity as the ensemble baseline, we adopt the standard bidirectional LSTM architecture in the MIMIC-III benchmark [140] as

Table 3.1: The mortality prediction results by the proposed sparse MoE and the baselines in AUC-ROC and AUC-PR. (Mean and 95% CI) (S: single. E: ensemble). Reprinted/adapted with permission from [3].

Performance Measures	AUC-ROC	AUC-PR
Standard LSTM (S)	0.855 (0.835, 0.873)	0.485 (0.431, 0.537)
Channel-wise LSTM (S)	0.862 (0.844, 0.881)	0.515 (0.464, 0.568)
Deep Supervision (S)	0.856 (0.836, 0.875)	0.493 (0.438, 0.549)
LSTM majority vote (E)	0.807 (0.793, 0.811)	0.521 (0.501, 0.533)
Sparse-MoE (E)	**0.888** (0.879, 0.891)	**0.530** (0.497, 0.570)

our individual expert, and constrain the neurons in the ensemble to be comparable to the baseline LSTM (each expert is a smaller version of the original one), and the output is the majority vote of the experts (instead of imposing sparsity, we activate all experts), with which we aim to answer the second question. Finally, for our model, we introduce the sparse gating on top of the expert group, to demonstrate the importance of our imposed *model sparsity*. We extract the patient representation from the gating network of the expert assignment to answer the last question. We have adopted the metrics AUC-ROC and AUC-PR, which are especially useful in imbalanced binary/multi-class classification present in many healthcare applications, including MIMIC-III [140] For the detailed architecture setting in the model, please refer to the supplementary material.

3.1.3.3 Results and discussion

We discuss the results in two corresponding tasks. First, for mortality prediction, as can be seen in Table 3.1 (we denote the single network baselines by (S), and ensemble framework by (E)), our model captures high risk patients and provides superior performance in predicting mortality than all the baselines on both AUC-ROC and AUC-PR scores. 95% confidence intervals are also shown by bootstrapping the test set 100 times. This validates that an ensemble method is necessary. Moreover, the results from the LSTM majority voting shows that, without proper expert collaboration monitoring, i.e. constraining each expert to be responsible for a small cohort, the experts would sometimes disagree with each other. Interestingly, it is shown that it has much lower AUC-ROC, but better AUC-PR than other baselines. AUC-PR is known for its good application in medicine because it captures better characteristics in imbalanced dataset where the minority class is of interest (survived vs. expired) [141], and thus more applicable in medicine. So we speculate that LSTM majority voting is better at finding the expired patients (higher risk) than other baselines, but slightly worse in terms of finding a balance between two classes. However, the sparse gating mechanism in our new sparse MoE can properly constrain each expert to focus on a small subset of the dataset, thus less likely to collide and overfit, giving an overall improvement on both metrics.

We further examine the results in phenotyping, where the issue due to heterogeneity only worsens. For phenotyping, the model needs to classify each patient into 25 binary labels, each of which is a clinically meaningful acute care condition, and not mutually exclusive (one patient can have multiple conditions concurrently). As shown in Table 3.2, we have calculated the individual performance of the 25 phenotypes and the macro-average of them, which is less affected by class-imbalance than micro-average. We can see that phenotyping is better predicted through an ensemble of experts than a single model. Our sparse MoE again performs better than the single network baselines and the naive ensemble baseline of majority voting, demonstrating the necessity of enforcing model sparsity. Note that the ensemble baseline is worse than the single network counterpart in this scenario, indicating in a heterogeneous task such as phenotyping, the expert collaboration is much more needed with fine-tuned sparsity.

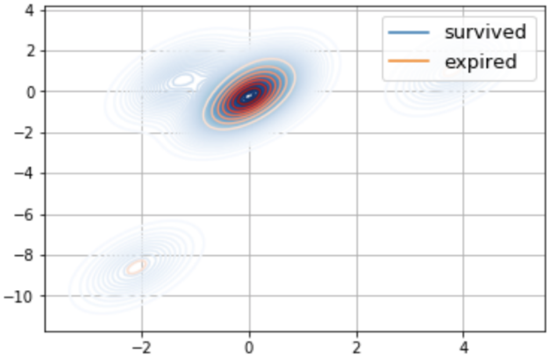


Figure 3.3: KDE of survived and expired patients from raw features. Reprinted/adapted with permission from [3].

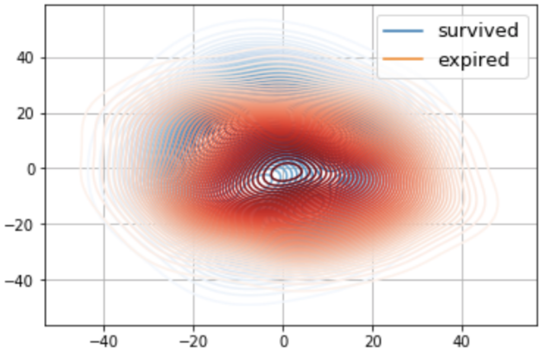


Figure 3.4: KDE of survived and expired patients from single expert. Reprinted/adapted with permission from [3].

Table 3.2: The individual phenotype prediction results of proposed method and the macro comparison with baselines in AUC-ROC through only first 48 hours. (S: single. E: ensemble). Reprinted/adapted with permission from [3].

Phenotype	type	AUC-ROC	Phenotype	type	AUC-ROC
Acute and unspecified renal failure	acute	0.766	Essential hypertension	chronic	0.642
Acute cerebrovascular disease	acute	0.902	Fluid and electrolyte disorders	acute	0.738
Acute myocardial infarction	acute	0.756	Gastrointestinal hemorrhage	acute	0.754
Respiratory failure; insufficiency; arrest	acute	0.821	Hypertension with complications	chronic	0.732
Chronic kidney disease	chronic	0.726	Other liver diseases	mixed	0.714
Chronic obstructive pulmonary disease	chronic	0.619	Other lower respiratory disease	acute	0.602
Complications of surgical/medical care	acute	0.695	Other upper respiratory disease	acute	0.66
Pleurisy; pneumothorax; pulmonary collapse	acute	0.657	Conduction disorders	mixed	0.655
Congestive heart failure; nonhypertensive	mixed	0.698	Pneumonia	acute	0.74
Coronary atherosclerosis and related	chronic	0.747	Cardiac dysrhythmias	mixed	0.634
Diabetes mellitus with complications	mixed	0.864	**All diseases (macro-averaged) (E)**		**0.727**
Diabetes mellitus without complication	chronic	0.756	baselines		
Disorders of lipid metabolism	chronic	0.685	LSTM Majority vote (macro-) (E)		0.651
Septicemia (except in labor)	acute	0.757	Deep Supervision (macro-) (S)		0.679
Shock	acute	0.847	Channel-wise LSTM (macro-) (S)		0.708

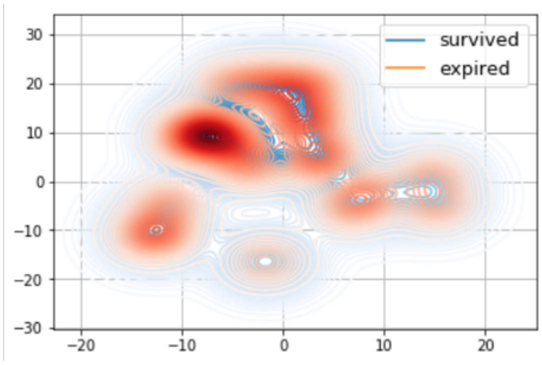


Figure 3.5: KDE of survived and expired patients from LSTM majority vote. Reprinted/adapted with permission from [3].

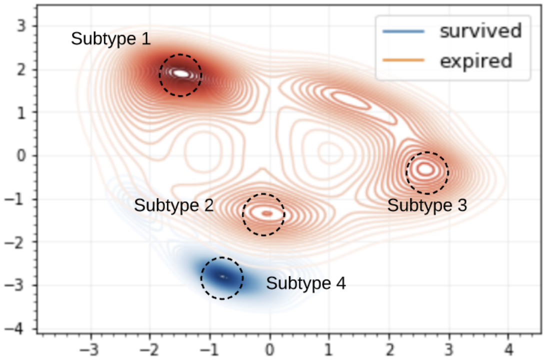


Figure 3.6: KDE of survived and expired patients from sparse MoE. Reprinted/adapted with permission from [3].

3.1.3.4 *Interpretability and heterogeneity*

In various clinical applications, patient subtyping is of great interest [5], due to the fact that personalized treatment can lead to better resource allocation, medication adaptation and ultimately patient outcomes [5]. Therefore patient stratification is a popular approach for clinical inter-

Table 3.3: Patient demographic and phenotype distribution. Reprinted/adapted with permission from [3].

Example subtype	1	2	3	4 (survived)
Gender (female)	45.4%	50%	62.7%	57.6%
Age (average)	65.2	88.6	69.1	77.1
# of patients in peak	88	36	43	78
Acute/unspecified renal failure	25.1% (17.4%, 32.7%)	**32.3%** (25.4%, 39.3%)	11.5% (2.3%, 20.7%)	30.8% (22.1%, 39.5%)
Acute cerebrovascular disease	9.2% (5.5%, 12.9%)	3.4% (0.5%, 6.3%)	**17.1%** (16.4%, 17.7%)	10.9% (7.7%, 14.0%)
Acute myocardial infarction	**24.3%** (23.7%, 24.9%)	20.8% (4.7%, 36.8%)	15.9% (14.6%, 17.1%)	14.4% (11.5%, 17.3%)
Cardiac dysrhythmias	32.8% (28.2%, 37.4%)	**37.1%** (32.4%, 41.8%)	30.2% (26.0%, 34.3%)	20.6% (17.8%, 23.5%)
Chronic kidney disease	11.3% (8.9%, 13.7%)	**21.8%** (3.8%, 39.8%)	7.0% (5.7%, 8.3%)	11.7% (9.8%, 13.5%)
Congestive heart failure	35.4% (29.0%, 41.8%)	**45.4%** (36.7%, 54.1%)	31.5% (23.2%, 39.7%)	23.0% (17.8%, 28.3%)
Coronary atherosclerosis	32.7% (25.1%, 40.3%)	**44.5%** (32.1%, 56.9%)	26.9% (21.5%, 32.2%)	27.6% (24.7%, 30.6%)
Disorders of lipid metabolism	**23.1%** (18.0%, 28.1%)	4.8% (0.9%, 8.6%)	**23.1%** (17.8%, 28.5%)	22.5% (18.7%, 26.3%)
Fluid and electrolyte disorders	28.2% (25.4%, 31.0%)	**53.2%** (45.4%, 61.1%)	47.0% (44.4%, 49.6%)	30.4% (23.2%, 37.7%)
Gastrointestinal hemorrhage	**14.2%** (8.9%, 19.6%)	11.4% (4.5%, 18.3%)	3.0% (0.4%, 5.6%)	10.2% (6.7%, 13.6%)
Pleurisy; pulmonary collapse	4.9% (3.2%, 6.7%)	1.3% (0.8%, 3.5%)	**13.1%** (9.1%, 17.1%)	6.2% (1.9%, 10.5%)
Pneumonia	13.5% (12.7%, 14.4%)	22.2% (17.6%, 26.9%)	**38.5%** (29.3%, 47.7%)	25.7% (23.9%, 27.5%)
Respiratory failure	17.2% (14.5%, 19.9%)	24.9% (16.0%, 33.8%)	**39.9%** (38.6%, 41.3%)	29.3% (23.8%, 34.7%)
Septicemia	4.9% (3.2%, 6.7%)	19.6% (18.9%, 20.3%)	**28.3%** (25.0%, 31.6%)	25.3% (23.3%, 27.3%)

pretability. Most work along this line of research involves clustering patients' clinical features to identify patient subtypes. However, raw feature clustering is unsupervised and thus identified clusters may not be directly amiable to provide useful clinical predictions or treatment suggestions. In our sparse MoE, we have implemented a gating network with the imposed model sparsity, concurrently trained with experts, to ensure that each expert is responsible for a subset of the overall heterogeneity in the dataset, making the assignment supervised. First, we study the mortality prediction task to show that the MoE can inherently 'divide-and-conquer' the hidden patient subtypes, and then use the phenotyping task to further show that larger-scale heterogeneity can be handled by ensemble learning. We note that subtypes here refer to the latent groups found by the model, and phenotypes are clinically labeled outcomes from professionals in MIMIC-III.

Discovering subtypes through visualization For mortality prediction, we compare the subtype discovery using 4 patient representations, i.e. 1) raw patient features, 2) single LSTM expert's last layer of hidden state representation, 3) fully activated expert group assignment, where the gating

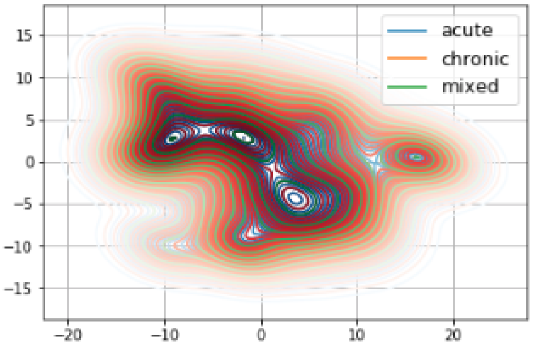


Figure 3.7: Kernel density estimation of acute/chronic/mixed phenotype patients from baseline. Reprinted/adapted with permission from [3].

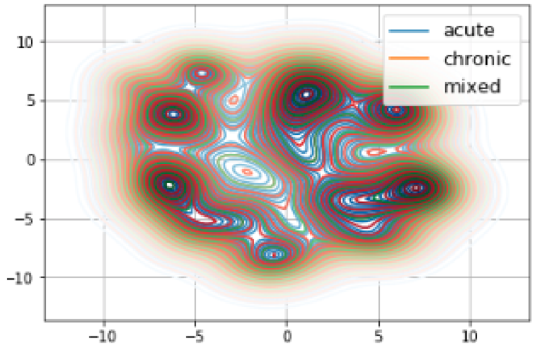


Figure 3.8: Kernel density estimation of acute/chronic/mixed phenotype patients from Sparse MoE. Reprinted/adapted with permission from [3].

network does not impose sparsity, and 4) lastly our sparse MoE assignments. We use fast-tSNE (t-Stochastic Neighborhood Embedding) [142] for dimension reduction as it has been shown to better capture the local and global structures of the data in the embedded space. We then use kernel density estimation (KDE) to visualize the aforementioned representations and plot the distributions of two groups of patients, i.e. survived and expired, shown in Figs. 3.3, 3.4, 3.5, 3.6. For raw feature embedding in Fig. 3.3, the two groups are almost completely overlapping, and slightly distinguishable in single expert's embedding in Fig. 3.4. The multiple LSTM framework produces few subtypes to some extent in Fig. 3.5. Compared to these results, the survived and expired patient groups by our sparse MoE are clearly separated in Fig. 3.6 inside which it has further separation of the expired patients (high risk) into finer grained types, allowing for the discovery of new patient subtypes with different causes.

Bridging discovered subtypes with phenotypes Furthermore, in Fig. 3.6 we pick four most distinctly discovered subtypes by sparse MoE to study their potential clinical interpretation, bridging the gap between discovered subtypes and clinical phenotypes. Note that we pick those 4 by visual appearances. In future we aim to propose some standardized metrics to choose distinctive peaks. We first extract the representative patients (the data points in the peaks) and cross-refer to the patient demographic statistics in MIMIC-III (note that our model does not have this information) and compare to check whether there are clinical differences between those subtypes. The demographic

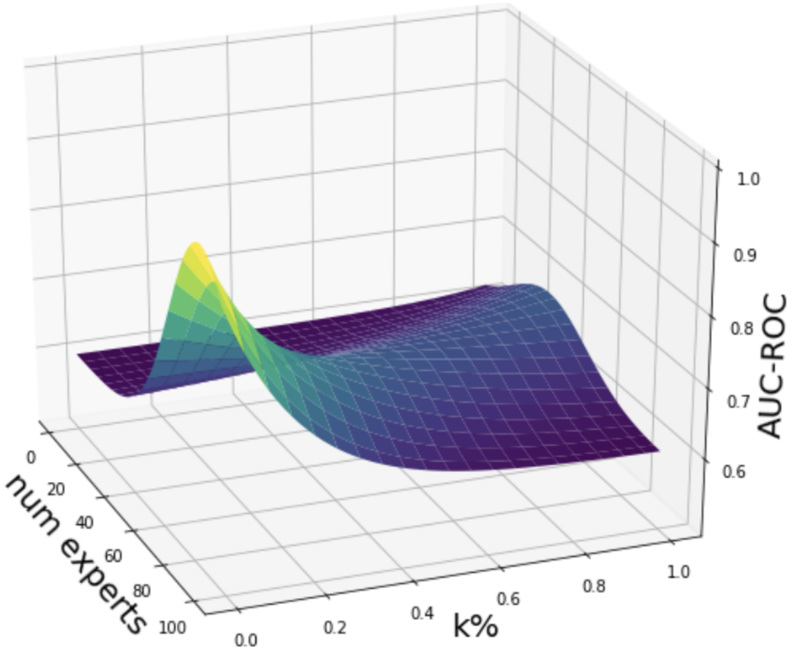


Figure 3.9: MIMIC-III data for tuning expert number and k% together.
Reprinted/adapted with permission from [3].

information is shown in the upper part of Table 3.3. As can be seen, subtype 2 is a significantly senior group, and subtype 3 is predominantly female patients. Moreover, we want to verify the discovered subtypes' clinical meaning. We then consolidate the phenotype information from each patient and compute the statistics within discovered subtypes, which is shown in the lower part of Table 3.3 (again the phenotype labels are agnostic to the model). We choose a set of example phenotypes that were recorded during the ICU stays. We calculate the percentages of the phenotypes in each discovered subtype of patients (with 95% confidence interval) and highlight the significantly higher percentage.

As can be seen, subtype 1 has a higher percentage of acute myocardial infarction, disorders of lipid metabolism and gastrointestinal hemorrhage than others. Subtype 2 has multiple higher percentage of phenotypes, indicating a more clinically risk group, which is justified by the older age. Subtype 3 generally has a higher percentage of respiratory related diseases (pneumonia, pleurisy, etc.). Subtype 4 (survived after ICU) is not highly associated with any of them. Therefore, the discovered subtypes can guide the clinical procedures and studies can be adopted accordingly to inspire new patient stratification routines.

Visualization in phenotyping task In the larger-scale heterogeneous task, i.e. phenotyping, we also derived the patient latent representation from our model and one of the baselines (channel-wise LSTM) to see if the 'divide-and-conquer' result is evident. We coarsely label each phenotype according to the acuity type listed in Table 3.2, i.e. acute, chronic and mixed. Shown in Figs. 3.7, 3.8, the phenotyping task is giving trouble to the model since it is more complicated than binary classification but Sparse MoE still seeks to find finer grained subtypes (more local peaks) among the patients whereas the baseline is more obtuse (peaks are lumped).

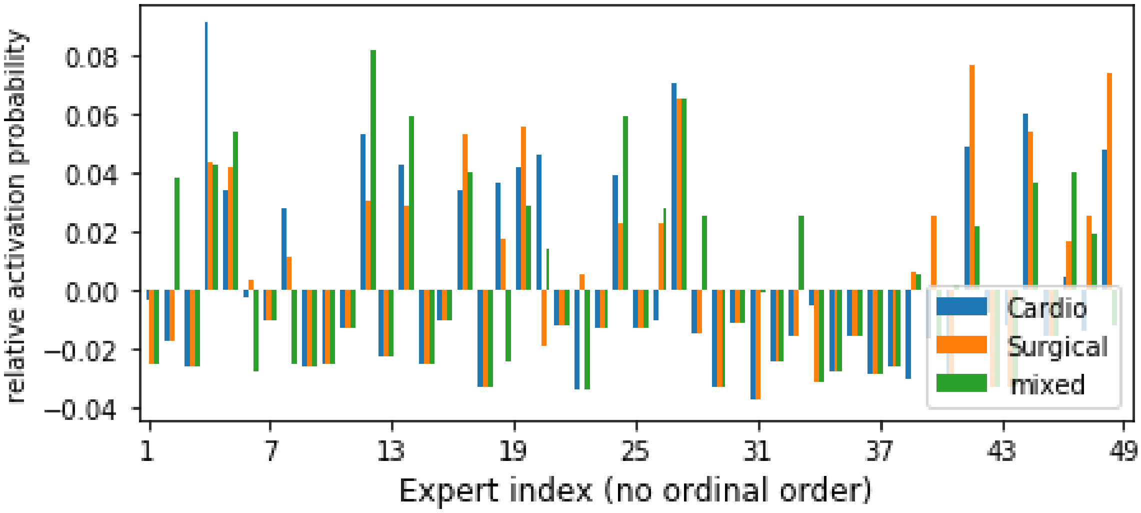


Figure 3.10: Expert activation by gating network. Reprinted/adapted with permission from [3].

Visualization of patient representations by expert assignment vector Here we show the usage of the expert assignment from the gating network for patient representation. The representation is a vector of probabilities of which one patient's data going through a specific expert network. Ideally, different patients with different medical conditions would have a unique set of experts to be activated. We have calculated the average activated probabilities on the 50 experts we choose from all the patient cohort as the baseline (50 is the number of total experts in our best experimental setting). We then extract three patients' expert assignment from the sparse gating network and subtract the average from them to demonstrate their excursion from the mean. Note that this representation is from mortality prediction model. The reason we choose these three patients is because they have similar risk scores (the last layer prediction, 1 being expired, and 0 being survived) but different assigned phenotypes (we look for their corresponding phenotype labels in MIMIC-III, since the mortality model does not know the phenotypes, we use it for proving that the model itself is *subtyping* patients). The first patient has risk score 0.67, and the phenotypes assigned are: 1) *Cardiac dysrhythmias* 2) *Diabetes mellitus without complication* 3) *Essential hypertension*. The second patient has risk score 0.64 and phenotype label is *Complications of surgical/medical care*. The third patient has risk score 0.7 and the assigned phenotypes are: 1) *Acute and unspecified renal failure* 2) *Chronic kidney disease* 3) *Congestive heart failure; nonhypertensive* 4) *Diabetes mellitus without complication* 5) *Hypertension with complications* 6) *Pleurisy; pneumothorax; pulmonary collapse*. We can coarsely define the first patient's issue is cardio related, second is from surgical care, third one is mixed with multiple high risk factors. We have shown the expert assignment in Fig. 3.10. As can be seen, the probabilities of each patient for activating an expert above average are sparse (there is no ordinal order from x-axis, which is just experts' indexes). Furthermore, their corresponding activated experts are different as well. The cardio patient has clearly one activated expert which stands out distinctively, whereas surgical patient and mixed patient each has some distinctively activated experts, but each of them are different from each other. For proof-of-concept, we want to know if the outstanding expert for cardio patient is mostly responsible for similar phenotypes. The index of outstanding expert is 42, and we examine the phenotype labels for all the patients going through this expert, as it turns out 100% of the cardio related phenotypes went through this expert (all patients going through this expert have at least one cardio phenotpype, such as *Coronary atherosclerosis and related*, *Congestive heart failure; nonhypertensive*, *Essential hypertension*, *Cardiac dysrhythmias*). So we can observe that the model intrinsically trains cardio-related patients to a sparse set of experts, which means the model is innately pushing similar patients to one expert without knowing their exact phenotype labels. So we have shown the utility of using expert assignment to explain the subtypes as the footprint of each patient's data in the network can be monitored better by this macroscopic style sparsity and thus provide a better understanding than neuron wise explanation. This can be especially helpful in precision medicine since the risk can be from different factors and only a score can not tell the whole story. While professional clinicians' opinion can be expensive and slow, the model generated subtypes can be of great help in finding difference in patients and recommend personalized treatment. In the future, we aim to study each expert's expertise with knowledge from professional clinicians to further shed light on this patient representation for practical usage.

3.1.3.5 Parameter analysis

Here we discuss the effect of sparsity tuning on the model performance, since it is the most important hyper-parameter in the model. We define $k\%$ as the percentage of the activated experts, i.e. k/n, where n is the total number of experts and k is the number of experts to be activated in each batch, shown in (3.5). We vary our MoE for different n and $k\%$ for the hyperparameter sensitivity study based on the corresponding evaluation metric by conducting the mortality prediction task as the performance measurement. We fit the distribution of the two variables into a kernel density estimation plot where the height of the plot is indicated by AUC-ROC. We have tuned the number of experts and the sparsity parameter simultaneously in a grid-search manner. The tuning shows the model would likely have a global optimum of the sparsity parameter as shown in Fig. 3.9. The trend shows that the number of experts should be large ($\sim$100) but the activated experts in each batch should remain low (0.1$\sim$0.2). The plot validates that a network with large capacity should carry the *model sparsity* so that the experts would work better at their corresponding specialty, rather than a fully activated regime. This echos with the experiment results above that a simple ensemble is not enough for a heterogeneous dataset as MIMIC-III, and a smart gating network is necessary when properly tuned for sparsity in order to conduct risk assessing and prediction.

3.1.4 Conclusions

In this chapter, we have developed a sparse Mixture-of-Experts framework, a realization of ensemble learning for heterogeneous patient data with interpretability for medical prediction. To make sure that each expert learns on a specialized, homogeneous subset of patients rather than general information, we have introduced a mechanism realized by a sparse gating network. The imposed sparsity pushes each expert to take part in a small portion of the overall heterogeneity. Therefore, the whole expert group can represent a large space of knowledge that may not have been achieved by existing methods. We have evaluated our method on the real-world clinical dataset with two tasks. The numerical results have shown that an ensemble is desirable for a heterogeneous dataset and have further improved with model sparsity. In addition, our visualizations have demonstrated better discovery of subtypes within the supervised learning manner as in the sparse gating network than other representations such as raw features. The discovered subtypes have been interpreted through clinical phenotype distributions, which indicate that the model intrinsically explores and distinguishes patients with different risk factors. This patient representation from sparse learning can shed light on more interpretable patient subtyping from a macroscopic point of view, whereas most of the existing work has not properly addressed the issue. For future work, we will continue the in-depth study of discovered patient subtypes and further explore the inner mechanism of patient stratification in the model. To use this framework in real medical settings for phenotype discovery, we need to show that meaningful patient groups can provide clinical practitioners actionable treatment decisions.

3.1.5 Supplementary Material

3.1.5.1 Consistency of the convergence of F and Q functions

In the paper we have used the following F function as a surrogate for the Q function in the E-step of the expectation-maximization (EM) algorithm that was introduced by [137]:

$$F(\tilde{P}(\mathbf{Z}), \mathbf{\Theta}) = \mathbb{E}_{\tilde{P}(\mathbf{Z})}[L_c(\mathbf{\Theta}; \mathcal{D})] + H(\tilde{P}(\mathbf{Z})). \tag{3.9}$$

As stated in the paper, $H(\tilde{P}(\mathbf{Z})) = -\mathbb{E}_{\tilde{P}(\mathbf{Z})}[\log\tilde{P}(\mathbf{Z})]$, which is the entropy of $\tilde{P}(\mathbf{Z})$. In [137], the F function is deemed as ‘variational free energy’ in statistical physics. Optimizing the F function then can be understood as maximizing the model evidence $L_c(\mathbf{\Theta}; \mathcal{D})$ or minimizing the surprise

$-\log\tilde{P}(\mathbf{Z})$. In [137], they demonstrated that the convergence achieved by $\tilde{P}^*(\mathbf{Z})$ and $\mathbf{\Theta}^*$ (that locally maximizes the F function) is transformable to the convergence for $\mathbf{\Theta}^*$ for the Q function.

Therefore the original EM algorithm was then transformed into the form as follows:

$$\begin{aligned} &\textbf{E-step: } Q(\mathbf{\Theta},\mathbf{\Theta}^{(t)}) = \mathbb{E}_{\tilde{P}(\mathbf{Z})}[L_c(\mathbf{\Theta},\mathcal{D})|\mathbf{\Theta}^{(t)}] \\ &\textbf{M-step: } \mathbf{\Theta} = \arg\max_{\mathbf{\Theta}^{(t)}} Q(\mathbf{\Theta},\mathbf{\Theta}^{(t)}) \\ \Rightarrow\ &\textbf{E-step: } \tilde{P}^{(t)} = \arg\max_{\tilde{P}} F(\tilde{P},\mathbf{\Theta}^{(t-1)}) \\ &\textbf{M-step: } \mathbf{\Theta}^{(t)} = \arg\max_{\mathbf{\Theta}} F(\tilde{P}^{(t)},\mathbf{\Theta}). \end{aligned} \tag{3.10}$$

Different from the original EM on the left, the empirical probability instead will be calculated $\tilde{P}(\mathbf{Z})$ that maximizes the F function in each E-step and use the probability as the posterior in M-step to update parameters $\mathbf{\Theta}$ (which is independent of the entropy term). Note that each iteration above is a GEM step (generalized EM) that makes sure that F increases or at least stays the same from one iteration $t-1$ to the next t, rather than a full log-likelihood maximization. Having the assumption that the local optimum of the F function is transformable to Q function, we can use the GEM for our training regime.

3.1.5.2 Assumptions for our sparsely updating EM algorithm

In literature [137, 143], the independence is assumed, i.e. $\mathbf{Z}$ can be factored as $(z_1, ..., z_n)$. We can see each component in $\mathbf{Z}$ can be considered as a latent variable with a learnable probability of an expert generating a data point (for generative Gaussian expert) or an expert being trained on an instance (neural network expert), so the process of updating one would not interfere another in the network. In addition, observational data $\{\mathbf{x}_i,\mathbf{y}_i\}_{i=1}^n$ usually are deemed as i.i.d. by default. So the $\tilde{P}(\mathbf{Z})$ would be able to factor as $\tilde{P}(\mathbf{Z}) = \prod_i \tilde{P}_i(z_i)$, which can ultimately give us the form $F(\tilde{P},\Theta) = \sum_i F_i(\tilde{P}_i,\Theta)$. Following Eq. (3.3), we can have $F_i(\tilde{P}_i,\Theta) = \mathbb{E}_{\tilde{P}_i}[L_c(\Theta,\{\mathbf{x}_i,\mathbf{y}_i\})] +$

$H(\tilde{P}_i)$. In this case, F function can be updated with one data point at a time:

$$\begin{aligned}
\textbf{E-step:} \quad & \text{Select one } z_i \in S^{(t)}, \\
& \text{so that } \tilde{P}_i^{(t)}(z_i) = \arg\max_{\tilde{P}_i} F_i(\tilde{P}_i, \Theta^{(t-1)}). \\
& \text{Set the rest } \tilde{P}_j^{(t)} = \tilde{P}_j^{(t-1)} \text{ where } i \neq j \\
\textbf{M-step:} \quad & \Theta^{(t)} = \arg\max_{\Theta} F(\tilde{P}^{(t)}, \Theta).
\end{aligned} \tag{3.11}$$

According to [137], the incremental version where $\tilde{P}^{(t)}$ was updated by one data point instead of all, has the same convergence as using all of them, albeit with different convergence rate. The E-step is not a full expectation any more, rather it is an incremental estimation of the posterior. So this justifies the method to train the EM sparsely. The selection set $S^{(t)}$ in Eq. (3.11) dictates which posterior of z_i is updated.

Table 3.4: Model architecture and hyperparameter comparison between the baselines and our sparse MoE framework. Reprinted/adapted with permission from [3].

Model	dimension	# of layers	dropout	bidirectional	time step (hour)
Standard LSTM[140]	512	2	0.3	Yes	1
Channel-wise LSTM[140]	128	2	0.3	Yes	1
Deep Supervision[116]	128	1	0	No	0.8
Sparse MoE (each expert)	32	2	0.3	Yes	1

3.1.5.3 Subtype discovery

In the paper we use sparse MoE expert assignments to represent each patient's data and use t-SNE (t-Stochastic Neighborhood Embedding) for dimension reduction into two-dimensional subspace for visualization. The subtype is then associated with a set of clinically defined phenotypes for distribution analysis, in order to validate the distinct difference in our discovered subtypes. The process is stochastic due to the following factors, such as random initialization of neural networks,

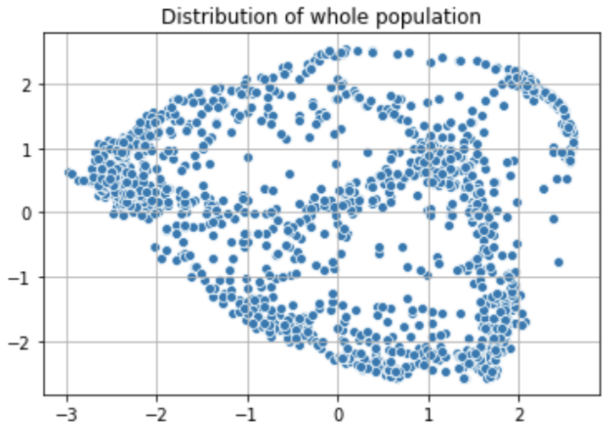


Figure 3.11: Raw t-SNE embedding. Reprinted/adapted with permission from [3].

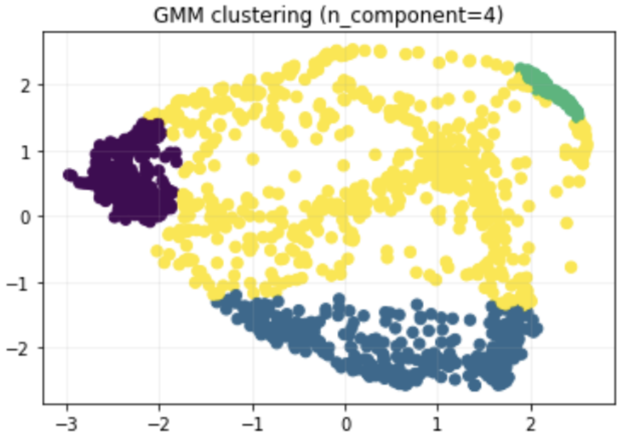


Figure 3.12: GMM clustering. Reprinted/adapted with permission from [3].

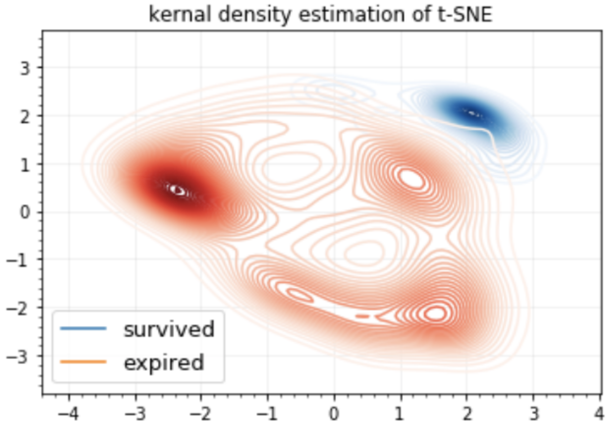


Figure 3.13: KDE of t-SNE. Reprinted/adapted with permission from [3].

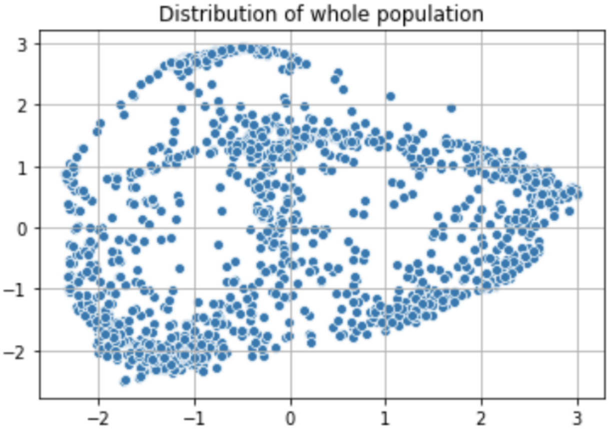


Figure 3.14: Raw t-SNE embedding. Reprinted/adapted with permission from [3].

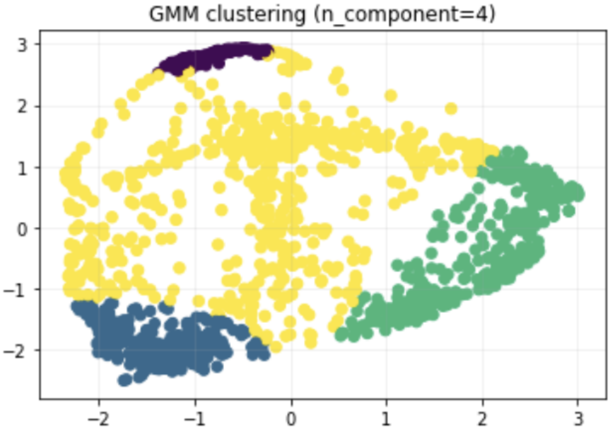


Figure 3.15: GMM clustering. Reprinted/adapted with permission from [3].

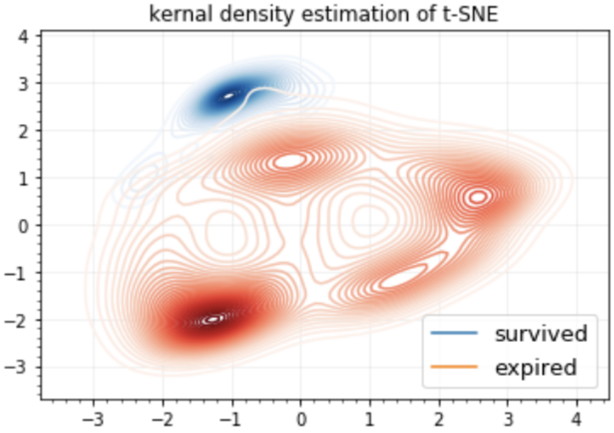


Figure 3.16: KDE of t-SNE. Reprinted/adapted with permission from [3].

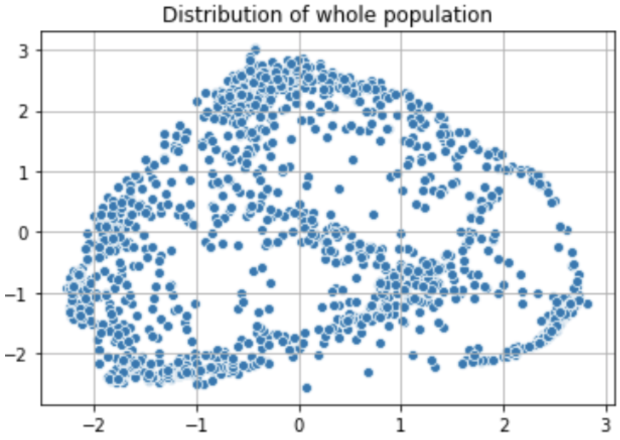


Figure 3.17: Raw t-SNE embedding. Reprinted/adapted with permission from [3].

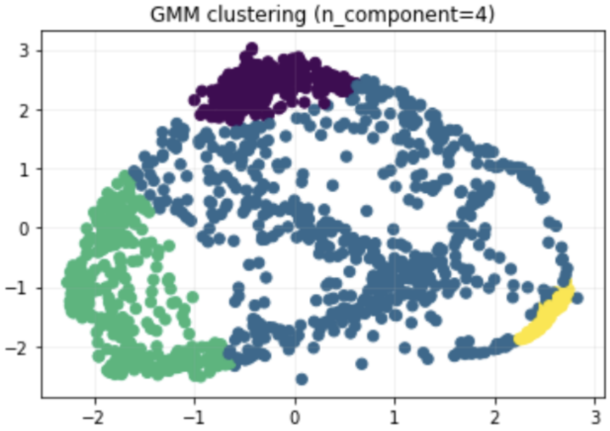


Figure 3.18: GMM clustering. Reprinted/adapted with permission from [3].

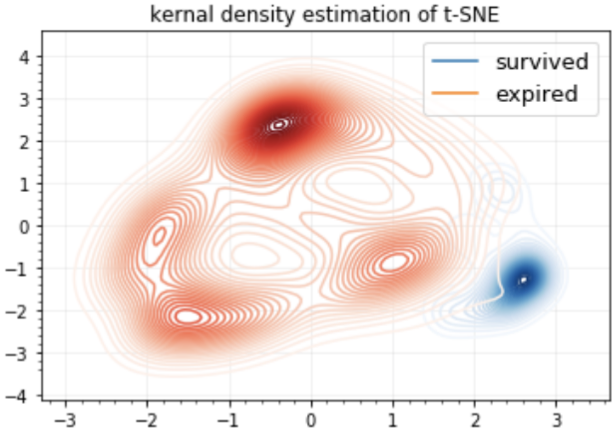


Figure 3.19: KDE of t-SNE. Reprinted/adapted with permission from [3].

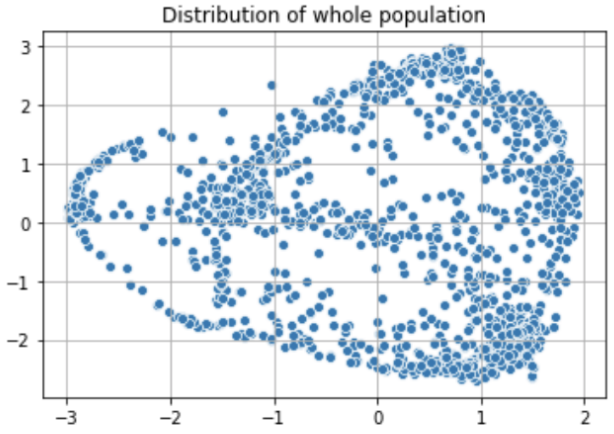


Figure 3.20: Raw t-SNE embedding. Reprinted/adapted with permission from [3].

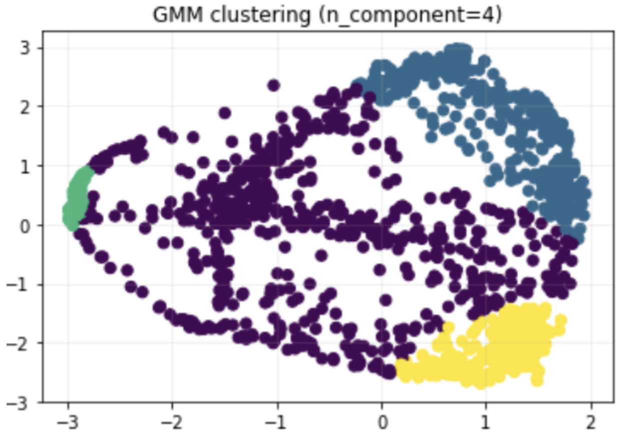


Figure 3.21: GMM clustering. Reprinted/adapted with permission from [3].

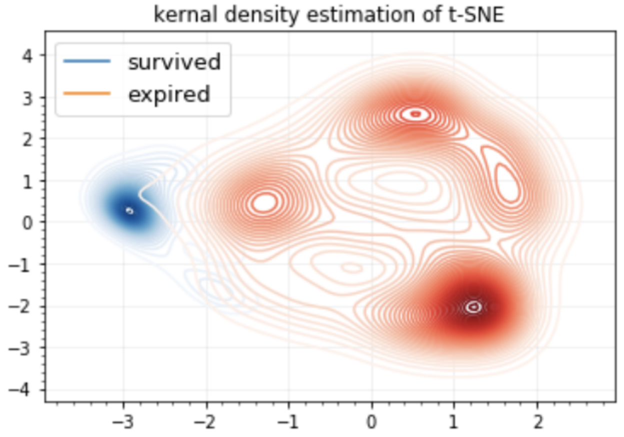


Figure 3.22: KDE of t-SNE. Reprinted/adapted with permission from [3].

Figure 3.23: Examples of four randomized trials of our model to show the model with ability to innately subtype. Reprinted/adapted with permission from [3].

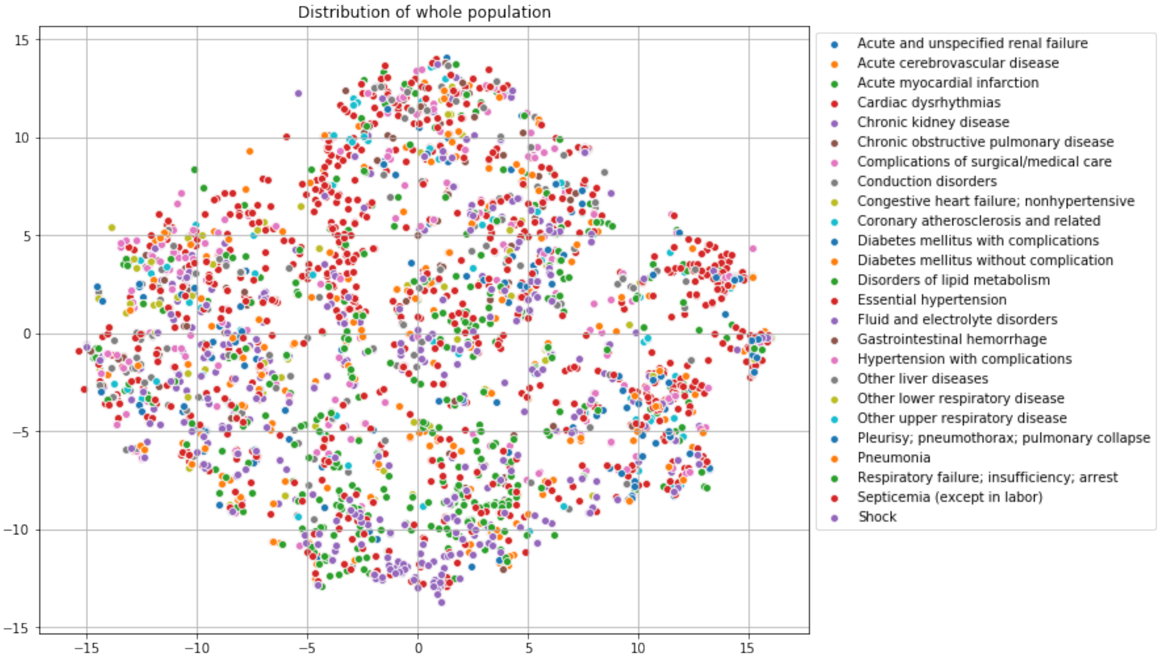


Figure 3.24: Phenotype embedding all 25 labels. Reprinted/adapted with permission from [3].

expert selection by gating network, neighbor selection in embedding. So the phenotype distribution is calculated by bootstrapping (with replacement) on the test set, from which the 95% confidence interval is calculated. Here we will study how stable the subtypes are in our model to have ablation study. We have four trial results shown in Fig. 3.23 as three columns, of which is raw t-SNE embeddding, Gaussian Mixture Model (GMM) clustering, and kernel density estimation (KDE) plots, respectively. As can be seen, all trials have the same subtype structures (minor variations) as shown in the paper, and the GMM clusters can also identify those groups of patients when given the number of clusters. Therefore, we have validated the experiments with multiple trials that the subtypes we have found are stable and clinically differentiable groups. As can be seen, although there are minor variations among those trials, all of them demonstrate distinguishable subtypes among the high-risk patients (expired) and the survived patients themselves can be captured as a cluster in both GMM and KDE.

Furthermore, the phenotyping task has the embedding plot from t-SNE as well. However, due to space limitation we cannot incorporate all the 25 labels in the paper. Therefore we include it

Table 3.5: MIMIC-III patient statistics and time-series feature type from the database tables. Reprinted/adapted with permission from [3].

before processing	after processing		
	train	test	
46,476 patients	15,331	2,763	patients
57,786 hospital admissions	17,903	3,236	ICU stays
61,532 ICU stays	17,903	3,236	Total samples

Variable	Mode	Variable	Mode
Capillary refill rate	categorical	Diastolic blood pressure	continuous
Glascow coma scale eye opening	categorical	pH	continuous
Glascow coma scale motor response	categorical	Heart Rate	continuous
Glascow coma scale verbal response	categorical	Glucose	continuous
Glascow coma scale total	categorical	Height	continuous
Mean blood pressure	continuous	Oxygen saturation	continuous
Respiratory rate	continuous	Systolic blood pressure	continuous
Temperature	continuous	Weight	continuous
Fraction inspired oxygen	continuous		

in Fig. 3.24. Note that the results here is done on only 48 hours worth of data, which makes the prediction more challenging than the full length as in [140].

3.1.5.4 Model/architecture details

For our sparse MoE model in the MIMIC-dataset, we have hyperparemeters for each experts, including the LSTM depth and width, as well as the hyperparameters for the whole architecture, including the number of experts. The optimal hyperparameters of each expert are given in Table 3.4, compared against the baselines. The time step indicates how much space between two readings for the 48 hours mortality prediction. Simply put, for one hour time interval, we have 48 readings. If the time-step needs to be smaller than that, we perform interpolation for data enrichment. For phenotyping, we specially note here that our experimental setting is constrained for the first 48 hours, as opposed to the full length in baseline [140]. Moreover, the best number of experts is 120 while the k% = 0.2, which means that for each batch of data, the number of activated experts is at most 24. We used the Adam optimizer to minimize the cross-entropy loss, with the learning rate = 1e-6 and epochs = 1000, and batch size = 32.

3.1.5.5 MIMIC-III data details

Here we provide the details of the MIMIC-III dataset [139], including raw time-series feature types and the total patients and ICU stays statistics, as shown in Table 3.5. For processing standards, such as removing patients with less than 48 hours ICU stay, splitting data into training and testing set, we followed the work in [140].

3.2 Density-Aware Personalized Training for Risk Prediction in Imbalanced Medical Data[2]

3.2.1 Introduction

3.2.1.1 Overview

Medical events of interest, such as mortality, often happen at a low rate in electronic medical records, as most admitted patients survive. Training models with this imbalance rate (class density discrepancy) may lead to suboptimal prediction. Traditionally this problem is addressed through ad-hoc methods such as resampling or reweighting but performance in many cases is still limited. We propose a framework for training models for this imbalance issue: 1) we first decouple the feature extraction and classification process, adjusting training batches separately for each component to mitigate bias caused by class density discrepancy; 2) we train the network with both a density-aware loss and a learnable cost matrix for misclassifications. We demonstrate our model's improved performance in real-world medical datasets (TOPCAT and MIMIC-III) to show improved AUC-ROC, AUC-PRC, Brier Skill Score compared with the baselines in the domain.
Machine learning-based medical risk prediction models continue to grow in popularity [115, 129, 144]. However, the performance of these models is often biased in evaluation by commonly reported metrics (such as area under the curve of the receiver operating characteristic: AUC-ROC), often reporting overly-optimistic findings as a result of the imbalance between those that observe

[2] Reprinted with permission from Huo, Z., Qian, X., Huang, S., Wang, Z. and Mortazavi, B.J., 2022. Density-Aware Personalized Training for Risk Prediction in Imbalanced Medical Data. Machine Learning for Healthcare Conference (MLHC). PMLR. Copyright 2022 by the authors.

medical adverse events and those that do not [145, 146, 147, 148]. The adverse event of interest is often in the minority class [149]. For example, in mortality prediction, patients with higher risk represent a smaller fraction in the cohort compared to most of the people who survive. Naively applying machine learning models may render dissatisfaction: the outcome of interest can be extremely costly, either through unnecessary medical intervention (type 1 error) or misdiagnosis (type 2 error). Furthermore, it is important not only to rank expired patients higher than survived patients w.r.t. probability output (e.g. AUC-ROC), but also the probability output is more calibrated [150].

While methods to tackle this imbalance issue via resampling or reweighting methods constitute a popular approach [151, 152], their applications in the context of medical data are often heuristic (or case by case) in nature. First, these techniques may give readjusted importance to the smaller class, but the weighting ratio remains ad-hoc from dataset to dataset, therefore, manual tuning might not be ideal. Second, apart from inter-class density discrepancy, one unique aspect of medical data is that even in the same risk group (same label), the patients may have different underlying comorbidities or risk factor characteristics that arrive at potentially high risk for various reasons, rendering intra-class heterogeneity [5]. This heterogeneity requires models to have a personalized training regime to distinguish the nuanced differences [3], to address the imbalance in a standardized/automated fashion. Rather than treating imbalanced densities as a problem, exploiting this information in training may enhance performance [153].

We propose a framework to address class imbalance density and make use of this imbalance to render density-aware training for improved risk prediction performance. First, we decouple the training of representation learning and classification. Traditionally, representation learning and classification are trained jointly [154], but by decoupling, class-specific features are extracted and class-specific predictions made, removing a source of bias for the learned classifier [155]. Second, the density differences are important to learn, not eliminate, when modeling. Patients with lower risk (majority) are often lower risk because they do not contain any of the common risk factors (e.g. lack of hypertension, diabetes, prior myocardial infarction), and hence, form a dense cluster. However, patients with higher risk (minority) may arrive at this high risk from different factors (e.g. renal failure versus respiratory distress), thus being scattered in the data space [3]. Our approach is density-aware, by avoiding re-sampling or re-weighing pre-processing steps, and the decoupling approach improves risk prediction performance. We demonstrate this approach in two different medical data scenarios: a randomized clinical trial dataset and an electronic health record (observational) dataset. We show that our method can achieve high predictive performance in these imbalanced medical datasets (imbalance ratio can range from $7 \sim 10$) and perhaps surprisingly it can also achieve superior calibration than the baselines without an extra set of calibration data.

Generalizable Insights about Machine Learning in the Context of Healthcare

As sophisticated models for increasingly large medical datasets are developed and promoted, evaluation of the predicted outcomes, through the use of an appropriate set of metrics is necessary. Medical data often contains low event rates for the major adverse events of interest. The primary measures of their performance, either threshold specific-based classification techniques, which may not properly account for the different costs of Type I and Type II errors, or the ability of the model to discriminate those at risk and those not, through the AUC-ROC, become overconfident in telling clinicians using the model who is not at risk. By leveraging the imbalance in the classes modeled, we are able to more accurately estimate those at risk, by more accurately identifying driving risk factors in the groups independently. As a result, this allows us to more concretely evaluate why they individuals are at risk (rather than simply being not at low-risk), and provide for better model calibration for medical decision making - through probabilistic interpretation of an occurrence in a frequentist perspective.

3.2.1.2 Related Work

Supervised learning methods on imbalance dataset tasks often re-balance data via re-sampling, such as oversampling [156], undersampling [157] classes. Others use synthetic samples to account for imbalance, where new samples are generated from perturbations of old samples [158, 159]. Another common approach is via re-weighting, which re-assigns training weights for each class based on criteria such as number of instances of each class [160], effective numbers [161] or the distance between loss [162]. However it is not clear the clinical utility of these methods since they were developed in non-medical datasets.

Medical models often focus on risk of adverse event estimation, which intrinsically carries data imbalance. Re-sampling has been widely applied [158, 163]. Cost-sensitive training is also applied, for example on Intensive Care Unit (ICU) data [164]. Hybrid approaches, which combine re-sampling and cost-sensitive training have also been applied [149]. These methods all are based upon the ad hoc tuning, weighting, or re-sampling to address imbalance, but do not learn from the imbalance information itself.

Furthermore, the imbalance issue in medical dataset not only affects the prediction, but also calibration [150]. The currently used metrics in medical modeling are usually not geared towards calibration and the metrics most widely used, such as AUC-ROC is susceptible to imbalance ratio [148]. The modern-day neural networks have achieved astonishing accuracy but studies have shown most methods are getting less and less calibrated [165]. Therefore we will demonstrate in our model the calibration is an extra contribution on top of handling imbalance prediction

3.2.2 Methods

In this subsection, we introduce our framework. The approach first separates the training data to different risk groups. Then, it uses a density-aware loss function to take into account the data density difference between majority class and minority class. Finally, it uses a learnable cost matrix to personalize misclassification. We stress that our framework is a training regime that can apply to different backbones (e.g. different neural network architectures) and we will later show in experiments this framework being used on real-world tabular as well as time-series medical data. The overall pipeline is shown in Fig 4.1.

3.2.2.1 Decoupling training for imbalance classes

In neural networks, we can coarsely define the last layer (or last few layers) [155] as the output classifier, since the output is used to determine the class of one specific instance. The previous layers of the network architecture can be deemed as the feature extractors or backbone. Traditionally

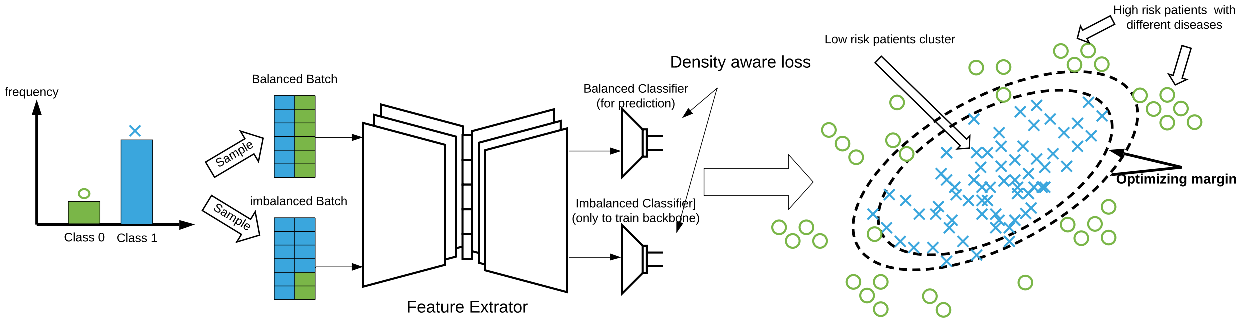


Figure 3.25: Overall pipeline. The original medical dataset is sampled with two distributions, one balanced and one imbalanced (default) batch. Both of the batches will go through the backbone but the classifiers will utilize different batches to optimize the density-aware loss, rendering personalized decision boundaries for different classes. Reprinted/adapted with permission from [4].

these two parts are trained jointly and the distinctions between them are ill-defined [154]. However, [155] showed that the classifier portion of the network is more susceptible to data imbalance, whereas the feature extractor is not, during training. Thus, we decouple training of the feature extractor and classifier.

Formally, let $\mathcal{I} = \{I_i\}$ be a set of inputs, and $\mathcal{Y} = \{y_i\}$ be the set of corresponding labels. For a typical objective function, we write:

$$\mathcal{L} = \frac{1}{N} \sum_{c=1}^{|C|} \sum_{i=1}^{|N_c|} l(h(I_{ci}), y_{ci}), \tag{3.12}$$

where $l(\cdot)$ is the loss function and $h(\cdot)$ is our model. $|C|$ signifies the number of total classes and $|N_c|$ the number of instances in one specific class. In an imbalance setting, the larger class with more training instances $|N_c|$ will dominate the loss and thus make the model biased. A naive way to tackle this issue is to adjust the sampling rate for the smaller class. For example, a *class-balanced sampling* (CBS) is proposed [166], where the instances from each class are sampled with equal probability so that the big class would not dominate the loss calculation and hence the density discrepancy will have much less effect. However, the CBS strategy will likely induce the ill-fitting problem because either the big class is under-sampled, inducing loss of information or small class is over-sampled, inducing over-fitting [167]. We propose that for a well-trained neural network, a set of abundant and diverse training instances is required, so that the model can generalize well in the testing set. A method is required to make use of the rich information in the big class but to ensure the smaller class is well represented as well.

Inspired by [168], we are proposing a solution to use both *class-balanced sampling* and *regular random sampling*, where the first would sample each instance to make sure each class has equal probability and the latter one samples each instance with equal probability. The function can be defined:

$$p_j = \frac{n_j^q}{\sum_{c=1}^{|C|} n_c^q}, \tag{3.13}$$

where p_j indicates the probability of sampling a data point from class j and the range for $q \in [0, 1]$ and $|C|$ is the number of classes. The *regular random sampling* entails the $q = 1$, meaning the probability will be proportional to the cardinality of the class j. The *class-balanced sampling* would entail $q = 0$ which means $p_j = 1/|C|$, and therefore each class is balanced. These two sampling strategies will generate two sets of batches, with each class's density built in, and we will train the feature extractor with both batches while the classifier will only train on the corresponding batch. In this way the rich information of big class will be preserved and at the same time the balanced classifier, which is eventually used for prediction during inference, will not be biased towards one class.

3.2.2.2 Density-aware outlier detection loss

To further make use of the inherent density information among the classes, we will introduce the density-aware training. There have been many cost-sensitive methods proposed to address the imbalance issue. One of the most popular ones is the focal loss [169]. This method focuses on the 'difficult' examples, which means the predicted probability of the example is far away from the true label. Based on previous discussion, we can treat the low risk patient as in-distribution data and high risk patients (with different underlying factors) as out-of-distribution data, and use the outlier detection technique to optimize the boundary [1].

By following this direction we propose a hinge loss based objective function. Hinge loss itself is less susceptible to density discrepancy among classes because it aims to optimize around support vectors, thus focusing the 'difficult' examples which are close to the decision boundary. However the traditional 'max-margin' training using the hinge loss did not take into account the class-wise density, which renders a non-personalized training. Our proposed personalized training is through a density-aware margin optimization [162]. This Density-Aware Hinge (DAH) loss can be written as follows:

$$\begin{aligned} \mathcal{L}_{DAH} &= \frac{1}{N}\sum_{c}^{|C|} \max(\max_{j\neq c}\{z_j\} - z_c + \Delta_c, 0) \\ \text{where } \Delta_c &= \frac{\text{K}}{|N_c|^{1/4}}, \text{ for } c \in \{1, ..., |C|\}, \end{aligned} \tag{3.14}$$

where $\mathcal{L}_{DAH}$ is the density-aware hinge loss, z_j is the model j-th element in the output vector, indicating the probability of this instance predicted to be j-th class, and z_c is the predicted output probability of the true class c. The form follows the traditional hinge loss, except the density-aware component Δ_c. The parameter K is a hyper-parameter, and $|N_c|$ is number of examples in class c. In [162], the exponential in $|N_c|^{1/4}$ is derived by the trade-off of optimizing all the margins between classes, so that the imbalanced test error can be smaller than a generalization error bound [170]. That is, $\gamma_j \propto |N_c|^{1/4}$, where γ_c is the margin in the hyper-plane for class c. Therefore we follow this tuning. The hyper-parameter K is usually tuned by normalizing the last hidden activation and last fully-connected layer's weight vectors' ℓ_2 norm to 1, as noted in [171].

In practice, the hinge loss may pose difficulty for optimization due to its non-smoothness [172]. First we derive the softmax from the original form and thus a relaxed form of hinge loss for smoothness is adopted to simulate the cross-entropy form:

$$\begin{aligned} &\mathcal{L}_{DAH} = \frac{1}{N}\sum_{c}^{|C|} - \log\sigma(z_c), \\ &\text{where } \sigma(z_c) = \frac{\text{exp}(z_c - \Delta_c)}{\text{exp}(z_c - \Delta_c) + \sum_{j\neq c}\text{exp}(z_j)}, \text{and } \Delta_c = \frac{\text{K}}{|N_c|^{1/4}}, \text{ for } c \in \{1, ..., |C|\} \end{aligned} \tag{3.15}$$

The 'max-margin' form is relaxed to a softmax function in the cross-entropy-like optimization. While some previous work [171, 173] adopted similar ideas, our proposed personalized margin Δ_c

Table 3.6: A typical cost matrix where the diagonal has zero cost, and C_{FN}, C_{FP} represents false negative cost and false positive cost, respectively. Reprinted/adapted with permission from [4].

Cost Matrix	Predicted as Positive	Predicted as Negative
True Positive	0	C_{FN}
True Negative	C_{FP}	0

can make use of the information in density discrepancy itself for training.

3.2.2.3 Trainable cost matrix

For personalized training, we propose to equip the density-aware loss with a trainable cost matrix. Traditionally the cost of training has been set static throughout the whole training process (e.g. false positive cost and false negative cost in binary classification). The default cost matrix can be seen as table 3.6, where the C_{FN}, C_{FP} were traditionally set to 1 (Note that the cost matrix here can only be applied to binary prediction). However this implies that the two types of cost are equal throughout the whole training [174]. But as we discussed before, the big class and small class would make the model more biased towards one versus the other due to the density disparity. But we want to use some mechanism to rebalance the training so that the model would be less biased. Thus instead of treating the costs as a prior knowledge, we make them as trainable parameters along with the model as well [175, page. 66]. In this way, the model will dynamically learn the cost to minimize the loss function. For an input and target pair (x, y), where the output of the model is $z = h(x)$, the loss function with incorporation of two costs under binary classification is proposed:

$$\begin{gathered}\mathcal{L}((x,y);h(\cdot)) = -\,y\log\sigma(C_{FN}z_{\max}) - (1-y)\log(1-\sigma(C_{FP}z_{\max})) \\ \text{subject to } C_{FN} > 0, C_{FP} > 0, C_{FN} > \theta C_{FP}\end{gathered} \tag{3.16}$$

The $z_{\max}$ indicates the largest logit along the output vector. The constraints above ensure that the two types of misclassification cost will always be positive [174] and due to the minority class is the prediction of interest (such as higher risk patients), we penalize more in the event of false negatives

Table 3.7: Summary of the datasets and the tasks. Reprinted/adapted with permission from [4].

Dataset	task	#instances	#features	IR
TOPCAT	Mortality hospitalization	1,767	86	7.92 1.71
MIMIC-III	Mortality Phenotyping	21,139	34	7.57 10.32

verse false positives. Here, θ can be tuned as a hyper-parameter.

In practice, when applying stochastic gradient descent (SGD), the parameters can only be updated without constraints. Here we relax the constrained problem as an unconstrained one, we thus rewrite:

$$C_{FN} = \theta C_{FP} + \mathcal{D}, \tag{3.17}$$

where $\mathcal{D}$ is a regularization term. Therefore we will only need to make sure $C_{FP} > 0$ during training. We propose to minimize the objective loss function in terms of $\log C_{FP}$ instead of C_{FP}:

$$\frac{\partial \mathcal{L}((x,y);h(\cdot))}{\partial \log C_{FP}} = C_{FP}\frac{\partial \mathcal{L}((x,y);h(\cdot))}{\partial C_{FP}}, \tag{3.18}$$

where the loss function can take the form as we defined above for density-aware training. Note that there are generally two ways to handle the constraints for optimization: reparameterization to an unconstrained minimization problem or projected gradient (PG) [176]. PG is to perform unconstrained gradient updates, then project back onto the feasible space after each update. PG directly solves the convex optimization problem, but the intermediate iterates can sometimes lead to a possibly less stable or too aggressive trajectory [177]. Ours is similar to reparameterization where numerical stability is more warranted in this regard.

3.2.3 Experiments

3.2.3.1 Datasets

In our experiment, we test our proposed model on two real-world medical datasets which include inherent imbalance issues and heterogeneous patients representations.

1) The first dataset is TOPCAT (Treatment of Preserved Cardiac Function Heart Failure with an Aldosterone Antagonist). TOPCAT is a multi-center, international, randomized, double-blind, placebo controlled trial sponsored by the U.S. National Heart, Lung, and Blood Institute [178]. TOPCAT collects patients from the United States, Canada, Brazil, Argentina, Russia, and Georgia between 2006 and 2013. The outcomes of interest were all-cause mortality and heart failure hospitalization through 3 years of follow-up. The data includes demographic and clinical data available from patients in addition to laboratory data, electrocardiography data, Kansas City Cardiomyopathy Questionnaire (KCCQ) scores (physical limitation score, symptom stability score, symptom frequency score, symptom burden score, total symptom score, self-efficacy score, quality of life score, social limitation score, overall summary score, and clinical summary score). The details of the variables are listed in supplementary Table A.1.

2) The second dataset is MIMIC-III (Medical Information Mart for Intensive Care). MIMIC-III is one of the largest clinical datasets that has been made publicly available [139]. It contains multivariate time-series data from over 40,000 intensive care unit (ICU) stays. The types of data range from static demographics such as gender and age to rapidly changing measurements such as heart rate and arterial blood pressure. The heterogeneity is one of the major challenges when analyzing this dataset, due to the diverse patient health conditions, rapidly changing hazard ratio as well as the corresponding treatments. We focus on using only the first 48 hours after ICU admission for the prediction of patient mortality and phenotyping. The intuition is that for early risk prognosis and phenotyping, the precaution procedure can be undertaken since the average ICU stay can be up to 100 to 200 hours [139]. We adopted the same data pre-processing steps as in a set of benchmark models in MIMIC-III [179](i.e. imputation, normalization, data masking, etc), where we used the same 17 clinical measurements and their derivations to construct in total 34 time-series features.

A summary of the datasets with the imbalance ratio (IR) is shown in Table 3.7. We test our model on three binary classifications (2 from TOPCAT, 1 from MIMIC-III) tasks and a multi-class classification task (MIMIC-III) to demonstrate our model can work under a variety of scenarios.

Note that only the phenotyping task in MIMIC-III is a multi-class multi-label scenario, so the imbalance ratio is calculated between the largest class and smallest class. The listing of phenotype labels used is in supplementary Table A.2, along with their medical type to indicate this is a heterogeneous set of labels that have many underlying driving factors. The model is not trained on a learnable cost matrix for phenotyping since the false positive and false negative is for binary classification, therefore, we solely rely on decoupling training and density aware loss.

3.2.3.2 Experimental Setup

For each of the datasets and tasks, we selected strong baselines from existing benchmarks.

1) Baselines for TOPCAT:

- RF [180]: a Random Forest based method which is originally tested on TOPCAT dataset
- U-RF [181]: a balanced Random Forest that randomly under-samples each boostrap sample to balance training
- R-MLP [152]: a Multi-layer Perceptron (MLP) model that uses reweighting in the training

For all the baseline with resampling or reweighting, we train the network on 80% of the data and tune the hyper-parameters including the weighting ratio in 10% of the data, and test on the rest 10%. For our model to have a strong neural work backbone, we construct a multi-layer Perceptron model as our backbone. The construction of the backbone is similar to R-MLP [152], training the neural network with 200 epochs, with learning as 0.001 and batch size as 64. More specifically this is a 4-layer fully connected NN, the first input layer is the same as number of features and each hidden layer has 28 neurons with one residual skip connection block and output layer has 2 neuron which is later measured on cross-entropy loss. We have our model train with our proposed decoupling and density aware loss, whereas R-MLP has under-sampling as their technique with some stochastic measures.

2) Baselines for MIMIC-III:

- GRU-D [182]: a Gated Recurrent Unit (GRU) based method where the model has a trainable decay component

- bi-LSTM [179]: a bi-directional Long Short-term Memory (LSTM) based method with channel-wise feature fusion
- flexEHR [183]: a GRU based method that uses word embedding technique to extract features.
- GRU-U [151]: a GRU based method that utilizes both trainable decay and undersampling technique for imbalance handling
- c-LSTM [179]: a channel-wise LSTM that process each variable independently in the first layer then fuse them in the second layer
- Deep Supervision [116]: an RNN based model that uses target replication for the supervision of LSTM in each time stamp, and with changing loss function the model needed to predict replicated target variables along with outcome

For the MIMIC-III dataset, we follow the same 80/10/10 splits. And we construct our backbone same as flexEHR [183] which is a GRU based method. We trained the models with 50 epochs with an early stopping threshold of 5 epochs with no increase in AUC-ROC on the validation set. The batch size is 128 and Adam optimizer is used with learning rate 0.001.

In addition to the traditional way of measuring probabilistic output of the medical models, i.e. area under the receiver operating curve (AUC-ROC), we argue that we need to incorporate the metrics that can represent the difficulties induced by imbalanced class densities. First AUC-ROC only measures the true positives (TP) and false positive (FP) relationship, which can present an overly optimistic view of an algorithm's performance if there is large skew in the class distribution [184]. On the other hand, area under precision-recall curve (AUC-PRC) can provide a more reliable interpretation under imbalance, due to the fact that they evaluate the fraction of true positives among positive predictions [185], and the precision-recall relationship will change when the test set's imbalance ratio changes, thus providing more sensitive evaluation [184]. Furthermore, in a medical model, the conventional way of measuring the model is through Brier score [186], which takes into consideration the calibration of the model. However, the Brier score is also susceptible to imbalance ratio [175]. We propose to use Brier Skill Score (BSS) [175], where the model takes the

Table 3.8: Results for TOPCAT dataset. Reprinted/adapted with permission from [4].

Task	Methods	AUC-ROC	AUC-PRC	BSS
Mortality	RF [180]	0.723 ± 0.003	0.512 ± 0.001	-0.357 ± 0.002
	U-RF [181]	0.752 ± 0.002	0.532 ± 0.002	-0.103 ± 0.003
	R-MLP [152]	0.736 ± 0.001	0.523 ± 0.005	-0.067 ± 0.003
	Ours	**0.794 ± 0.002**	**0.583 ± 0.002**	**0.166 ± 0.003**
Hospitalization	RF [180]	0.763 ± 0.005	0.657 ± 0.006	-0.008 ± 0.0004
	U-RF [181]	0.771 ± 0.005	0.674 ± 0.006	-0.005 ± 0.003
	R-MLP [152]	**0.789 ± 0.003**	0.661 ± 0.005	-0.012 ± 0.001
	Ours	0.788 ± 0.007	**0.711 ± 0.003**	**0.132 ± 0.002**

Table 3.9: Results for MIMIC-III dataset. Reprinted/adapted with permission from [4].

Task	Methods	AUC-ROC	AUC-PRC	BSS
Mortality	GRU-D [182]	0.852 ± 0.002	-	-
	bi-LSTM [179]	0.862 ± 0.004	0.515 ± 0.001	-0.801 ± 0.002
	flexEHR [183]	0.878 ± 0.004	0.513 ± 0.002	-1.105 ± 0.003
	GRU-U [151]	0.876 ± 0.006	0.532 ± 0.002	-
	Ours	**0.892 ± 0.001**	**0.586± 0.004**	**0.240 ± 0.003**

Task	Methods	Macro AUC-ROC	Micro AUC-ROC
Phenotyping (Multi-class, Multi-label)	c-LSTM [179]	0.708 ± 0.0023	0.725 ± 0.0053
	bi-LSTM [179]	0.770 ± 0.0081	0.791 ± 0.0048
	flexEHR [183]	0.755 ± 0.0052	0.814 ± 0.0071
	Deep Supervision [116]	0.679 ± 0.0074	0.713 ± 0.0061
	Ours	**0.771 ± 0.0061**	**0.821 ± 0.0049**

calculated Brier score and compare it to a reference point, i.e. a scaled Brier score by its maximum score under a non-informative model [187], to show the improvement:

$$BSS = 1 - \frac{BS}{BS_{\text{max}}} \tag{3.19}$$

We chose the reference BS_{max} to the the prevalence predictor to output the probability based on the imbalance ratio, i.e. $BS_{\text{max}} = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2$ and prediction f_t is replaced by the event rate and o_t is the outcome label of interest [188].

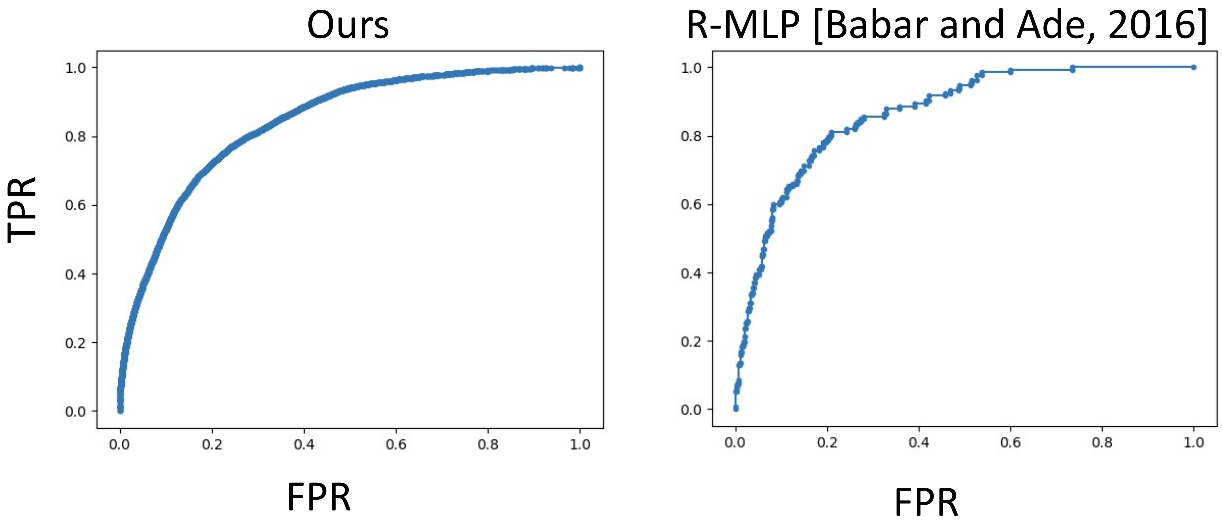


Figure 3.26: AUC-ROC comparison. Reprinted/adapted with permission from [4].

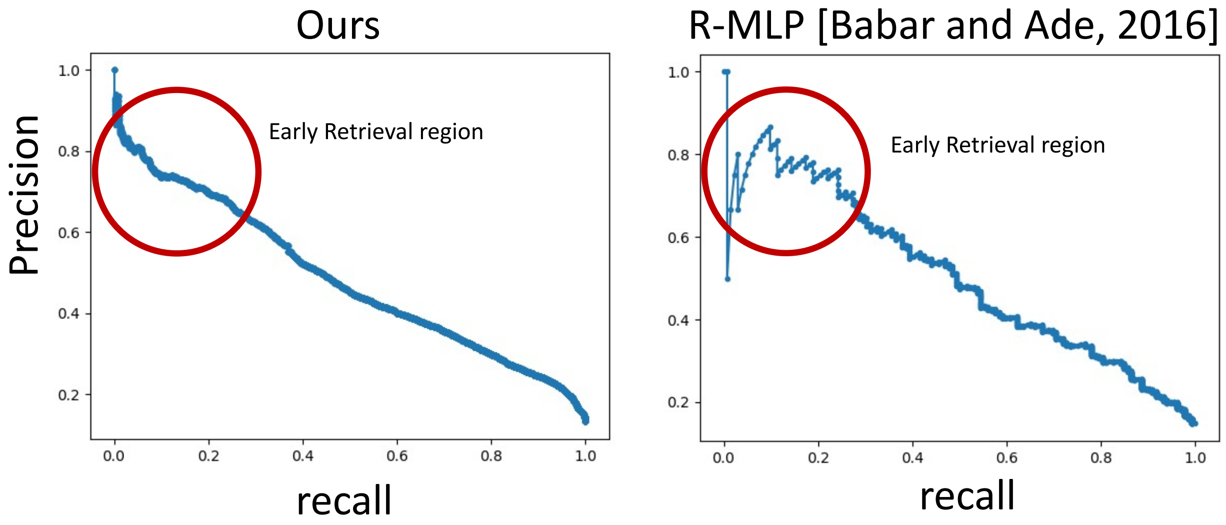


Figure 3.27: AUC-PRC comparison. Reprinted/adapted with permission from [4].

3.2.3.3 Results

First, for the TOPCAT dataset in Table 3.8, we have compared our model with the baselines and we repeated the experiments in a 5-fold cross-validation scenario and compute the 95% confidence intervals. In the mortality prediction task, we are performing better on three metrics, especially in the imbalance oriented metric, AUC-PRC and BSS. The margin improved on AUC-PRC is obvious, showing the model is sensitive on finding a balance between precision and recall, both of which measure the performance of the class of interest (minority class where the patients eventually expired). Furthermore, the baselines model all have negative BSS scores, showing that in this mortality prediction scenario, the imbalance can pose a big challenge for a model to calibrate. In fact negative BSS is not uncommon in existing work [189, 190], showing that many modern-day models can have high predictive power but are poor at calibration, as noted in [165]. We will later show that by comparing against with some post hoc calibration technique on the baselines, our model can still stand out on both prediction and calibration. Next for hospitalization prediction, our model also outperforms the baselines in two of the key metrics. The AUC-ROC is second to the best, after the same model backbone trained on resampling. We suspect this is due to the fact that the IR score is lower in this task, rendering less focus on difficulty induced by imbalance and resampling is designed to handle the example-wise difficulty. However as we discussed before, the AUC-ROC is not sensitive to class distribution so the majority class's performance can lead to the model having an overly optimistic evaluation. We compared the AUC-ROC plot and AUC-PRC plot of our model and the R-MLP baseline in Figure 3.26 and 3.27. As can be seen, the AUC-ROC plots of the two models are similar, however, the AUC-PRC plot shows on the upper region of the curve the baseline is performing rather unstably but our model gives a more smooth curve. In [185], this region is defined as *early retrieval* region, where is usually used to measure in information retrieval application for when results of interest account for a small portion of all the corpus [191, 192] (what is the model precision when recall rate is low). We can conclude our model has better performance on AUC-PRC is due to the better part on *early retrieval* where it can better handle the imbalance for the class of interest.

For the MIMIC-III dataset in Table 3.9, we also compared our model with the baselines (Note some metrics are empty due to the original model did not report those metrics and there is no publicly available code to replicate the results). First, for mortality prediction, our model is again better across the board among all the three metrics, especially on BSS evaluation on the model. It is perhaps surprising that traditional medical models were rarely optimized w.r.t. calibration, which however is an important metric for medicine [193]. For the phenotyping task, due to multi-

Table 3.10: Calibration study for TOPCAT dataset. Reprinted/adapted with permission from [4].

Methods	AUC-ROC	AUC-PRC	BSS
RF [180]	0.723 ± 0.003	0.512 ± 0.001	-0.357 ± 0.002
w/ Califorest [150]	0.734 ± 0.003	0.498 ± 0.001	-0.052 ± 0.002
R-MLP [152]	0.736 ± 0.001	0.523 ± 0.005	-0.067 ± 0.003
w/ Temperature scaling [165]	0.744 ± 0.003	0.518 ± 0.001	0.102 ± 0.002
Ours	**0.794 ± 0.002**	**0.583 ± 0.002**	**0.166 ± 0.003**

class, multi-label scenario, the previously existing methods did not adopt AUC-PRC (precision, recall are used mostly in binary classification) or BSS (Brier score is used in a 1-0 probabilistic output model for calibration purposes.) So we instead use macro AUC-ROC and micro AUC-ROC for model performance comparison. From the table, we can see our model is better than all the baselines on these two metrics, showing the multi-class multi-label imbalance scenario can also be handled by our framework.

In [165], the authors argued the confidence calibration, being the problem of predicting probability estimates representative of the true correctness likelihood, is important for classification models in many applications, but modern neural networks are becoming increasingly lacking in this respect. They proposed a *temperature scaling* method to calibrate the model which is a variant of Platt scaling [194]. The method is to use sigmoid function as the transformation for model's output into proper posterior probability. Our method of distribution-aware loss resembles the temperature scaling in that we have a component in the softmax to 'soften' the probability similarly to the temperature variable. Furthermore the component is label distribution aware, making it particularly suitable for calibration in an imbalanced setting. We aim to compare against this method. Furthermore, the tree-based methods all performed badly especially in terms of calibration. We will use a calibration specifically designed for tree models, i.e. Califorest [150], where the authors used Out-of-Bag (OOB) samples as the calibration set and each individual prediction is used to calculate the weights for the samples. We will use these two methods as the post hoc calibration method (i.e. the model is calibrated after finished training, this will theoretically give them more advantage since our model does not use any post hoc calibration) for the tree-based model and neural networks to study if the BSS metric for them can be increased to positive. We did our experiment on TOPCAT dataset and the task is hospital mortality prediction. The results are shown in Table 3.10. As we can see we have equipped the tree model, i.e. Random Forest, the Califorest and R-MLP the Temperature scaling to calibrate after the training, which renders improvement on BSS for both of the cases. The RF has not been able to push BSS to positive but the improvement is more substantial. The temperature scaling has pushed the neural network to have postive BSS, meaning the calibration is better than a model that outputs the prevalence of the events. However we should note that both of the post hoc calibration techniques have lowered the AUC-PRC, meaning the predictive power is compromised for the calibration. In our model we can observe high predictive power as well as calibration. We postulate that the decoupling training indeed separates the bias from imbalanced data to the classifier while the feature extractor maintains the power of absorbing all information. The label distribution-aware loss, acting similarly to the Temperature scaling (where the scaling factor is inherently tuned during training with our modified softmax formulation), is calibrating the balanced classifier without sacrifice of predictive power. To this end, without using the extra calibration dataset is not an issue anymore. We have tried to test the Temperature scaling on our model but we did not observe improvement on calibration but the AUC-ROC and AUC-PRC are slightly compromised as well.

Furthermore, as a proof-of-concept, we are particularly interested if our assumption holds, i.e. the patients who survived would be similar to each other where the patients who expire would be more dissimilar. We have extracted the embedded vector from our model whose backbone is based on [183], carrying 256-dimension last hidden layer (before classifier) on the mortality prediction task in MIMIC-III. And then we apply t-SNE [142] which is a visualization algorithm to embed the data into 2 dimensions. We plot the embedding along with their labels to show the density differences, shown in Figure 3.28. As we can see, the survived patients account for a small and condensed space where the patients who expired would form different peaks, indicating different local clusters (e.g. diseases/phenotypes). This can prove the assumption of the density discrepancy as training information can be truly captured.

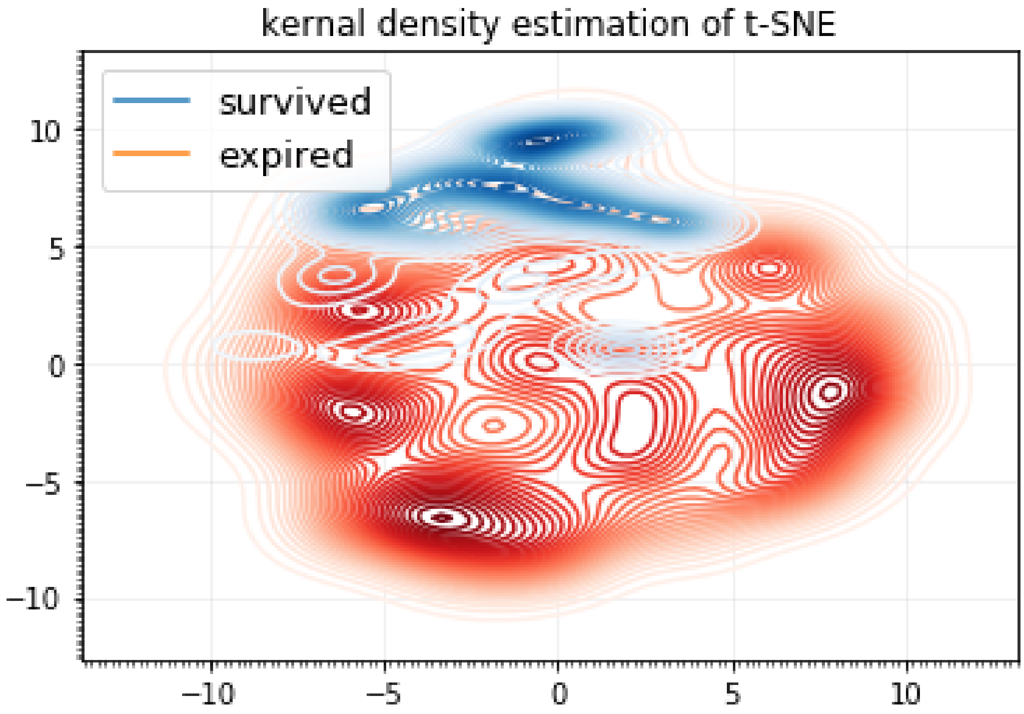


Figure 3.28: Density plot of survived/expired patients. Reprinted/adapted with permission from [4].

Table 3.11: Ablation study for TOPCAT dataset. Reprinted/adapted with permission from [4].

Task	Methods	AUC-ROC	AUC-PRC	BSS
Mortality	MLP	0.736 ± 0.004	0.523 ± 0.002	-0.067 ± 0.001
	MLP-TrainableCost	0.770 ± 0.003	0.541 ± 0.002	-0.188 ± 0.001
	MLP-decoupling	0.778 ± 0.002	0.569 ± 0.005	-0.480 ± 0.004
	MLP-FL	0.782 ± 0.004	0.541 ± 0.003	-0.080 ± 0.004
	MLP-DAH	0.779 ± 0.001	0.549 ± 0.005	0.111 ± 0.004
	MLP-Ours	**0.798** ± **0.002**	**0.589** ± **0.001**	**0.178** ± **0.002**

3.2.3.4 Ablation Study

We have a few components in our model such as decoupling training and density aware loss function. We are interested to know what makes the model improve and how can we dissect the model to demonstrate. We aim to study how does the decoupling help the prediction, and specifically what has the model learned. Also by comparing density aware loss with vanilla version (traditional cross-entropy loss) as well as another advanced version of loss function (focal loss

Table 3.12: Ablation study for MIMIC-III dataset. Reprinted/adapted with permission from [4].

Task	Methods	AUC-ROC	AUC-PRC	BSS
Mortality	GRU	0.871 ± 0.004	0.514 ± 0.003	-1.116 ± 0.005
	GRU-TrainableCost	0.879 ± 0.001	0.520 ± 0.002	-1.108 ± 0.005
	GRU-decoupling	**0.892 ± 0.003**	0.577 ± 0.002	-0.909 ± 0.002
	GRU-FL	0.875 ± 0.008	0.523± 0.007	-0.112 ± 0.007
	GRU-DAH	0.876 ± 0.003	0.534 ± 0.005	0.078 ± 0.003
	GRU-Ours	**0.892± 0.001**	**0.586± 0.004**	**0.240± 0.004**

[169]), we conduct a thorough comparison between them. In MIMIC, we have an existing strong backbone that we can apply our techniques on [183], which is based on a GRU model. In the TOPCAT dataset, to construct a strong backbone, we make use of the same MLP architecture as in R-MLP [152] with a residual skip connection block [109] that can be further decoupled or trained with different loss functions. We listed our ablation study in Table 3.11 and 3.12 in these two datasets both for mortality prediction.

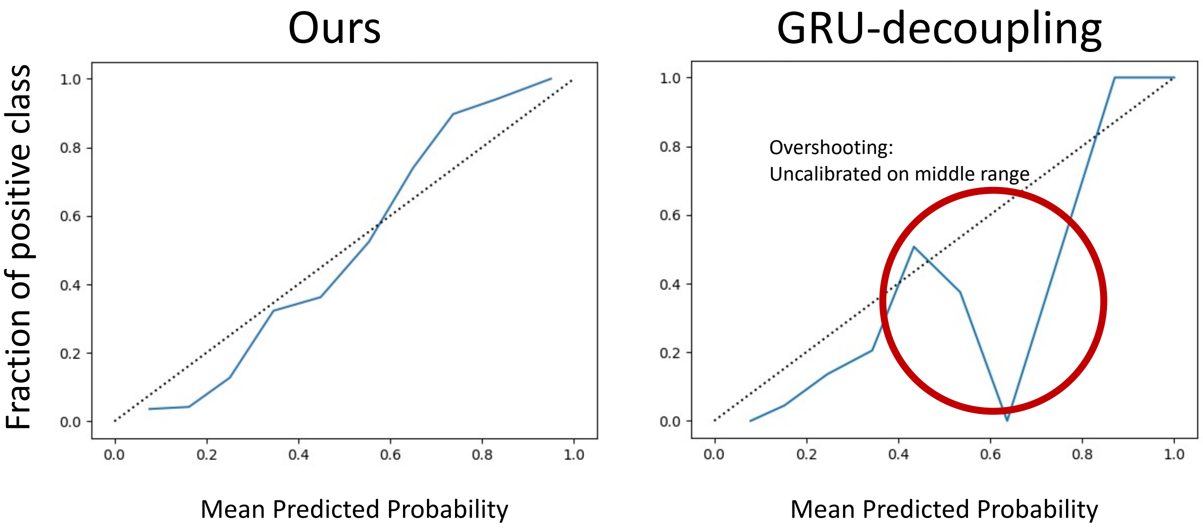


Figure 3.29: Calibration plot comparison. Reprinted/adapted with permission from [4].

First, for the TOPCAT dataset, we can see that when fully applying our framework on the backbone, the model would outperform all other variants in Table 3.11. Another finding is that the decoupling training is improving the AUC-PRC in a larger margin than others, suggesting that this way of training can largely avoid the imbalance issue by through a more distribution-aware metric. However the shortcoming of decoupling alone is that it is bad at calibration, where it is among the worst BSS metric in the methods. Second, when applying density aware loss alone, we can see the model can be better calibrated (i.e. positive BSS), which is usually an important aspect of a medical model [180] because the output probability can be evaluated as the risk score for further ranking. For the MIMIC-III dataset in Table 3.12, we can see that the decoupling itself can improve significantly and this method alone can give good AUC-ROC (tied as best). We are then interested to know how does this single trick compare to the full framework on improving BSS in terms of calibration. The comparison is in Fig 3.29 where we can see our model's calibration is closer to diagonal, rendering a more natural 'S' shape [195], where the baseline GRU-decoupling has poor calibrated range when the output probability is in mid/high range (which is the label of high-risk patients). This is showing the model is overshooting for this range of probability, likely due to an density discrepancy, because the model would assign overconfident probability to the patients in higher risk, requiring a density aware training. The over-confident prediction is prevalent in modern-day neural networks, where the mean predicted probability is higher than the fraction of positive class in a certain bin as noted in [165]. However, when equipped with the full framework, the performance of our model can increase significantly, especially on Brier Skill Score for calibration and rendering the plot to be closer to the diagonal (perfect calibration).

3.2.3.5 Parameter study

Since we have incorporated a trainable cost matrix, and we are interested in how does the parameter θ in Eq. 3.16 change the performance in the model. We have search on a space of $\{1, 5, 10, 25, 50, 100\}$ for θ, following [174]. On the TOPCAT dataset for mortality prediction we conduct the experiments and show it in Figure 3.30. We can see that AUC-ROC peaks at $\theta = 5$ while AUC-PRC can be $\theta = 10$. However, given the confidence interval's overlap, the significance for choosing $\theta = 10$ over $\theta = 5$ for AUC-PRC can be statistically minimal, therefore $\theta = 5$ is chosen.

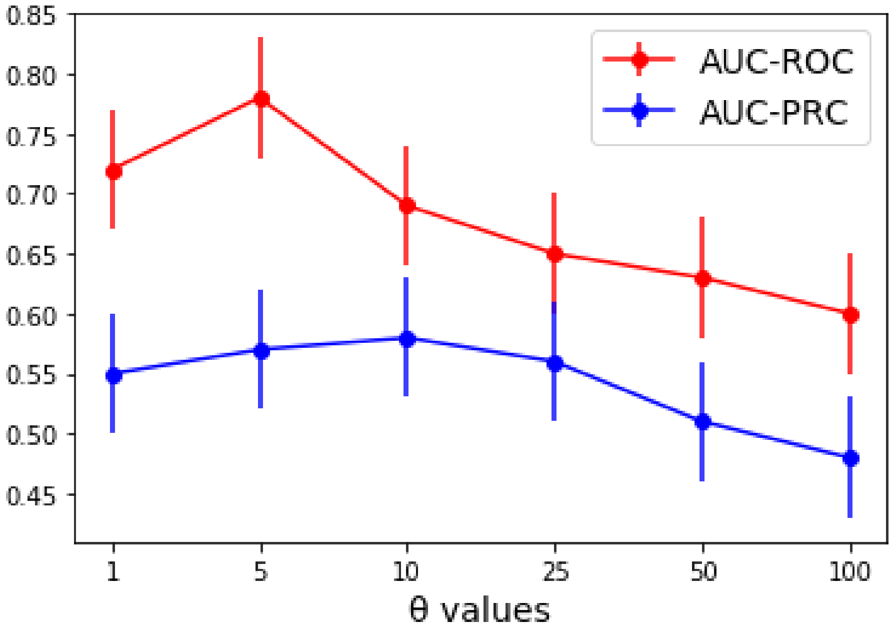


Figure 3.30: Tuning of θ for AUC-ROC and AUC-PRC. Reprinted/adapted with permission from [4].

3.2.4 Conclusion

We proposed a framework to treat class imbalance, which is prevalent in medical datasets. The introduced framework not only addresses imbalanced class densities but also makes use of the density discrepancy to train a model. The decoupled training method alleviated bias caused by the majority class, by ensuring faithful representation of the minority class. Further, we used a density-aware loss to personalize training of each class, specifically: learning that lower-risk patients arrive at low risk by calculation of the similar factors, forming a dense cluster in the data space, but high-risk patients are dissimilar, driving them to different regions of the data space. We demonstrated that our model, trained with this decoupling framework along with density-aware loss and learnable cost matrix, outperformed baseline approaches when applied to risk prediction in medical datasets. Furthermore, through experiments we find that traditional models were poorly calibrated, calling for more comprehensive evaluation, especially geared towards imbalance issues. Our framework overall has shown to be better at prediction as well as calibration, which can be of great use in the medical domain.

3.3 Sparse Embedding for Interpretable Hospital Admission Prediction[3]

3.3.1 Introduction

3.3.1.1 Overview

This work introduces a sparse embedding for electronic health record (EHR) data in order to predict hospital admission. We use a k-sparse autoencoder to embed the original registry data into a much lower dimension, with sparsity as a goal. Then, t-SNE is used to show the embedding of each patient's data in a 2D plot. We then demonstrate the predictive accuracy in different existing machine learning algorithms. Our sparse embedding performs competitively against the original data and traditional embedding vectors with an AUROC of 0.878. In addition, we demonstrate the expressive power of our sparse embedding, i.e. interpretability. Sparse embedding can discover more phenotypes in t-SNE visualization than original data or traditional embedding. The discovered phenotypes can be regarded as different risk groups, through which we can study the driving risk factors for each patient phenotype.

Emergency department (ED) admission is based on triage, which is a collection of data recorded by the nursing staff in a number of categories, including demographics, chief complaint, and vital signs. Patient disposition, either admission or discharge, will be made based on the triage data. Prediction of hospital admission with a precise model can help improve patient care and logistical efficiency [196]. Early identification of admission can allow hospitals to optimize healthcare resources and improve patient outcomes [197]. ED crowding, for example, is one of the problems that often occurs in hospitals, and it can incur higher readmission rates, longer hospital stays, and even higher mortality rate [198]. A reliable prediction model for ED admission can aid these problems.

Studies that predict ED admission by using triage data as input develop statistical models to predict an outcome of disposition [196, 197, 199]. Triage data usually includes demographics,

[3] Reprinted with permission from Huo, Z., Sundararajhan, H., Hurley, N.C., Haimovich, A., Taylor, R.A. and Mortazavi, B.J., 2019, July. Sparse embedding for interpretable hospital admission prediction. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 3438-3441). IEEE. Copyright 2019 by the authors.

vital signs, chief complaint, nursing notes, and early diagnostics. Additional models include hospital usage statistics and patient past medical history [199], achieving an area under the receiver operating characteristic (AUROC) curve of 0.849. Improvements through machine learning algorithms have been limited due to the poor interpretability of these more complex models [200]. Each patient may have different admission risk factors, and early identification of a patient phenotype is important to medical decision support. High-risk patients may need different treatment than low-risk patients. Therefore, patient stratification needs to be done along with the process of prediction. A potential phenotyping method projects patient data into a 2D plot by using visualization algorithms, and constructing patient phenotypes from the distribution. In each phenotype, a customized classification method can be proposed and used to provide a more fine grained prediction. However, the reasons that drive each patient into their corresponding phenotype, i.e. feature attribution in phenotyping, need additional exploration.

In this paper, we propose an interpretable framework to study hospital admission and explore the driving predictive risk factors for different patient phenotypes. First, triage data will be embedded into low dimension vectors with sparsity as a training goal. Sparse embedding has been proven useful for finding important features in a number of domains, including computer vision [201], natural language processing [202] and time series modeling [203]. In the word embedding domain, previous studies showed that sparsity can help preserve high level information and thus give more interpretability to each word's embedding when comparing their similarities. In [204], authors proposed an enhanced k sparse autoencoder, which uses two sets of penalty to enforce average and partial sparsity in the hidden layer. In general, sparsity forces models to concentrate useful information in a much lower dimensional space, which helps important features stand out when building a prediction model and also aids in grouping similar patients. After triage data is embedded, we use t-distributed stochastic neighbor embedding (t-SNE) [205] to visualize the data, which can be helpful for patient stratification. Phenotypes were extracted from the visualization of embedded data. In each phenotype, we build a customized model specifically for that cohort, and use feature ranking techniques to show what features are the most predictive in that cohort.

Finally, we use the important features to describe each phenotype, which is regarded as the process of feature attribution. Our contributions are as follows:

- We modify k-sparse autoencoding to further improve sparsity by pushing each dimension to either 1 or 0.
- We use a visualization technique to demonstrate patient risk group stratification.
- We identify low- vs. high-risk patient phenotypes that describe patients that are admitted.

3.3.1.2 Related Work

Medical data science using EHR data has become increasingly popular to understand high-dimensional data. In [115], authors used a denoising autoencoder to embed patient EHR data. The input was first 'corrupted' by Gaussian noise to account for missingness in medical settings. In [206], the authors aimed to model patient health state trajectories. They used Long short term memory (LSTM) for the progression of illness parameterized by time. In their model, medical intervention was incorporated to account for the course changes of illness. In [207], the authors used non-negative tensor factorization to extract patients phenotypes iteratively, by using the data from diabetic patients who are later diagnosed with chronic kidney disorder (CKD). In [208], the authors used a recurrent neural network (RNN) with skip gram embeddings to predict all diagnosis and medication categories in the EHR. In [209], the authors proposed a hierarchical framework by using a denoising autoencoder through a semi-supervised learning scenario to conduct phenotype stratification. The output was later fed into a random forest for prediction. By visualization of the results, they show their model's superiority of phenotype stratification. However, the demonstration of stratification was only in synthetic data.

3.3.1.3 Methodology

3.3.1.3.1 Sparse embedding We first introduce how we built a model for a k-sparse autoencoder with a loss function that penalized any deviation of the preset activation distribution for the hidden layer. We designed a secondary constraint in the training model to further push each dimension of the sparse embedding to be close to either 0 or 1, i.e. binarization. In this case, the sparse embedding was forced to carry the most important information for each patient and the new embedded vectors were regarded as the concentration of the original features.

Given a dataset D with size of D instances and F original features D $= [\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \ldots, \boldsymbol{X}_D]^T \in \mathbb{R}^{D\times F}$, we found the representations in the autoencoder's hidden layer that preserved the important information from the original features. We defined two mapping functions, and we framed the classical autoencoder objective function as:

$$\phi\colon \boldsymbol{X} \to \tilde{\boldsymbol{X}} \psi\colon \tilde{\boldsymbol{X}} \to \boldsymbol{X} \tag{3.20}$$

$$\underset{\phi,\psi}{\arg\min}\, \frac{1}{|D|} \sum\nolimits_{i=1}^{D} ||\boldsymbol{X}_i - (\phi \circ \psi)\, \boldsymbol{X}_i||_2^2 \tag{3.21}$$

Next, we enforced a k-sparse activation constraint that penalized the deviation of observed average activation value from the pre-defined average activation value:

$$\underset{\phi,\psi}{\arg\min} \sum_{h\in\phi,\psi} ||\max\left(0, \rho_{h,D} - \rho^*_{h,D}\right)||^2 \tag{3.22}$$

where $h \in H$ indicates neurons in the hidden layer and $\rho_{h,D}$ is the expected average activation values that parameterizes a Bernoulli distribution:

$$\sum\nolimits_{h=1}^{H} \tilde{\boldsymbol{X}}_h \sim \text{Bernoulli}\left(\rho_{h,D}\right) \tag{3.23}$$

Finally, we enforced a secondary constraint that further pushed each dimension to either 1 or 0 in order to achieve higher information concentration:

$$\underset{\phi,\psi}{\arg\min} \frac{1}{|D|} \sum\nolimits_{i=1}^{D} \sum\nolimits_{h=1}^{H} \left(\tilde{\boldsymbol{X}}_{i,h} \times \left(1 - \tilde{\boldsymbol{X}}_{i,h}\right)\right) \tag{3.24}$$

3.3.1.3.2 Visualization through t-SNE We used a popular visualization algorithm, t-SNE, to visualize the original data and embedded data distributions. t-SNE is a commonly used visualization

Table 3.13: Dataset Description. Reprinted/adapted with permission from [5].

Category	**Number of variables**	**Category**	**Number of variables**
Response variable	1	Past medical history	281
Demographic	9	Outpatient medications	48
Triage evaluation	13	Historical vitals	28
Chief complaint	200	Historical labs	379
Hospital usage statistic	4	Imaging/EKC counts	9
Total		972	

tool that can nonlinearly reduce data dimensionality while preserving the similarity among data points, and is well-suited for embedding high-dimensional data to two or three dimensions for visualization. After plotting patient data through t-SNE, we visually see the distributions.

3.3.1.3.3 classification with XGBoost We defined the patient cohorts in the visualization and built models to fit only that specific cohort. We used XGBoost, a gradient boosting classification algorithm and its internal feature ranking to determine important features for phenotype discovery.

3.3.2 Experiment

3.3.2.1 Data description

In this study we use health records of adult patients' visits to the Yale New Haven Health system, including one academic and two community emergency rooms, between March 2014 and July 2017. All three EDs are part of a single hospital system, utilizing the Emergency Severity Index (ESI) for triage. There were 560,468 patient visits in total. Each patient was assigned a final disposition, i.e. the response variable, either admission or discharge. 971 variables were collected during triage, serving as the patient information for the final medical decision-making. Data description is provided in Table 3.13.

3.3.2.2 Sparse embeddding

We embedded the 971 features, excluding the disposition and ESI of each patient, into 100 sparse dimensions using the k-sparse autoencoding introduced above. For the sake of comparison, we also built a traditional autoencoder to 100 dimensions to highlight the benefit of sparsity. We normalized the data to make all variables fall into the range of [0, 1] to align with the binary features that make up the majority of the dataset. We visualized the encoding and density of all the features in Fig 3.31. In Fig. 3.31, the y-axis represents patients and the x-axis represents each dimension of the data. The sparse embedding impact is visually apparent versus the traditional autoencoder, when reducing to 100 dimensions.

In order to prove the newly introduced sparsity did not lose important information, we then used XGBoost to predict the disposition of each patient through original data, traditional autoencoder output, and our sparse embedding. Using a five-fold cross-validation, we tested the AUROC of each embedding method and present those results in Table II. As we can see, the result from sparse embedding is still competitive. We will show what benefit we gain from sacrificing some marginal accuracy.

3.3.2.3 Visualization through t-SNE

Fig. 3.33 shows the visualization of the data when using t-SNE. We use kernel density estimation (KDE) to show the distribution. The red contour lines represent admitted patients, and the blue represent discharged patients. For original data and traditional autoencoder output, we can only see one peak of admitted patients, meaning that admitted patients are regarded as similar. While this aligns with medical intuition based upon risk, we expect phenotypes that are admitted for different reasons, independent of common risk factors. For our sparse embedding visualization, we can see that it has two peaks for admitted patients, one smaller more concentrated peak and one narrow but extended peak, as well as additional concentrations to the right of the figure. For the sake of comparison, we give one plot with only admitted patients. The left peak has an admission rate of 55% (812 out of 1474 patients), while the right peak has an admission rate of 45% (524 out of 1160 patients). We extract patients from the peaks of original data and the peaks from sparse embedding, and we use KDE to estimate the distribution in terms of ESI, as shown in Fig. 3.32 Given the distribution of ESI, we can see patient stratification when applying sparse embedding on the original data. Although both of the peaks are admitted patients, we can roughly define the

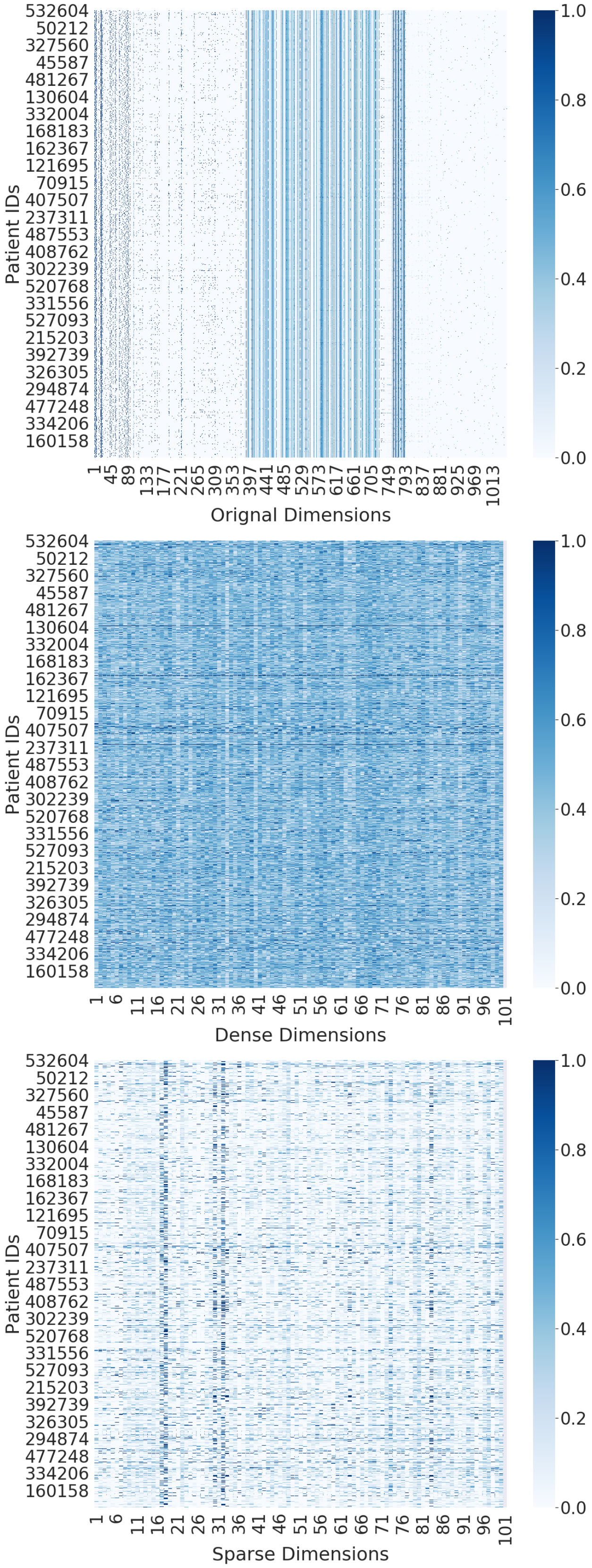


Figure 3.31: Comparison of original data (upper), traditional AE (middle) output and K-sparse embedding (bottom). Reprinted/adapted with permission from [5].

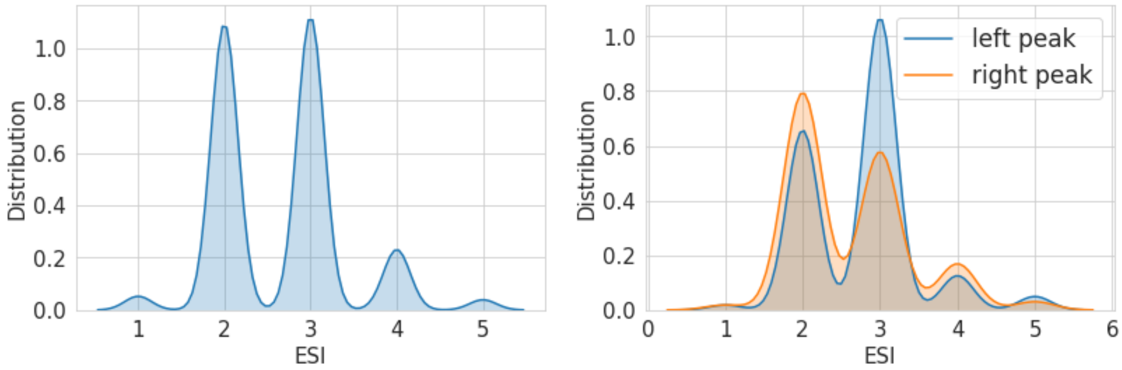


Figure 3.32: Comparison of ESI for the peak from original data and two most concentrated peaks in sparse embedding. Reprinted/adapted with permission from [5].

patients from the left group as high-risk, and patients from the right peak as low-risk, based upon ESI. This identifies different factors that drive admission not considered by ESI.

3.3.2.4 Classification with XGBoost

We built two XGBoost models to fit to new phenotype cohorts found in the concentrated peaks in the visual embeddings, to predict hospital admission. These models also return the feature importance of each model. We calculate the AUROC of admission prediction as well as the root mean square error (RMSE) of ESI, shown in Table III. In Fig. 3.34 we show the top 20 features and the importance value as ranked by XGBoost in each peak. Fig. 3.35 shows a few manually chosen representative features that vary between the two peaks. This figure shows that the higher-risk group is composed of elderly adults, who among other differences have more varied values for PTT, a measurement of bleeding risk. The low-risk group has a relatively younger cohort and, among other differences, higher eosinophil count. We therefore discover a younger cohort that might not generally be considered at risk for admission (low ESI) that has a high concentration of admissions as determined by the embedded models and can find unique phenotypes that are not apparent when modeling on the entire population.

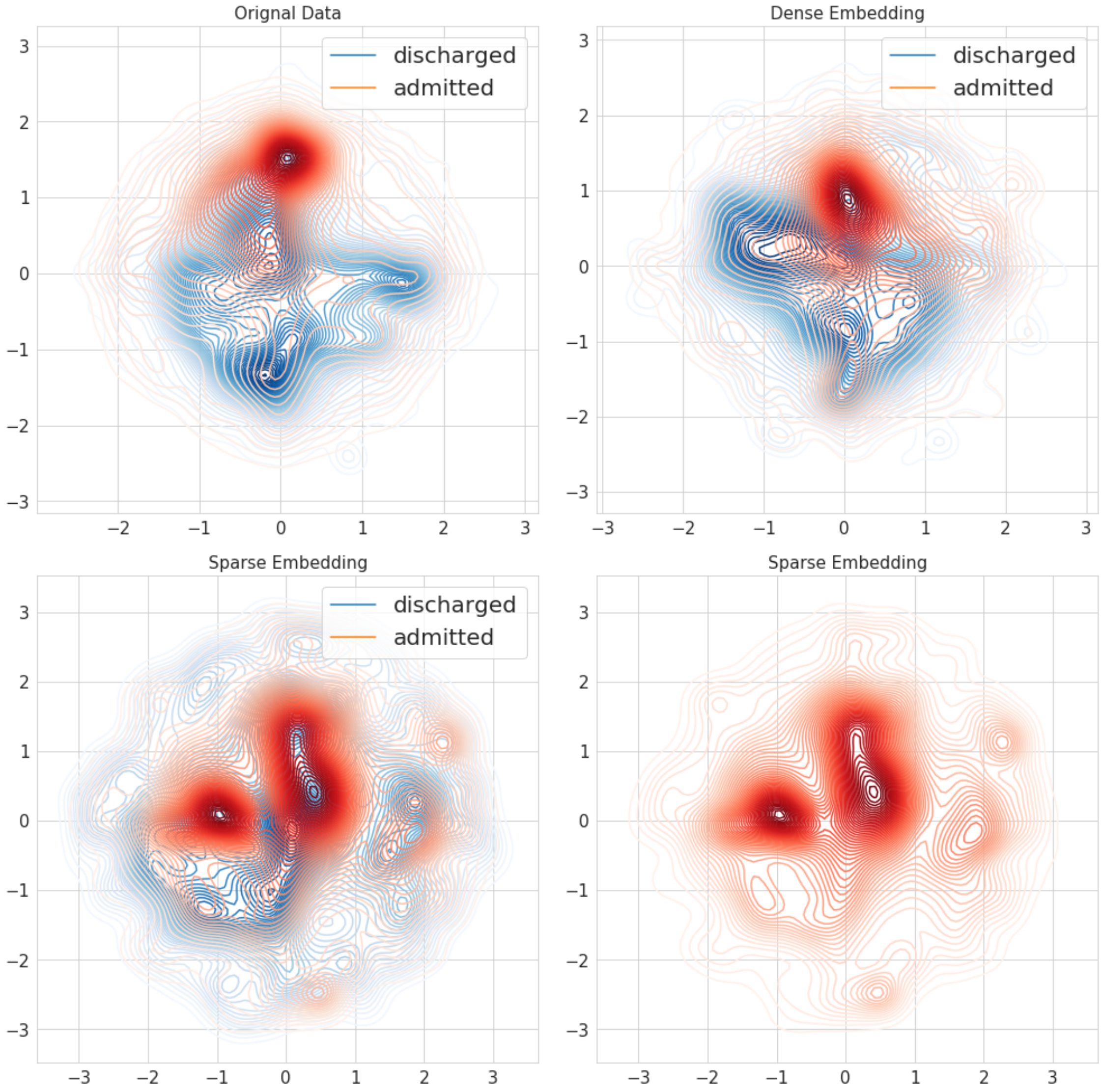


Figure 3.33: Visualization of data through t-SNE. Reprinted/adapted with permission from [5].

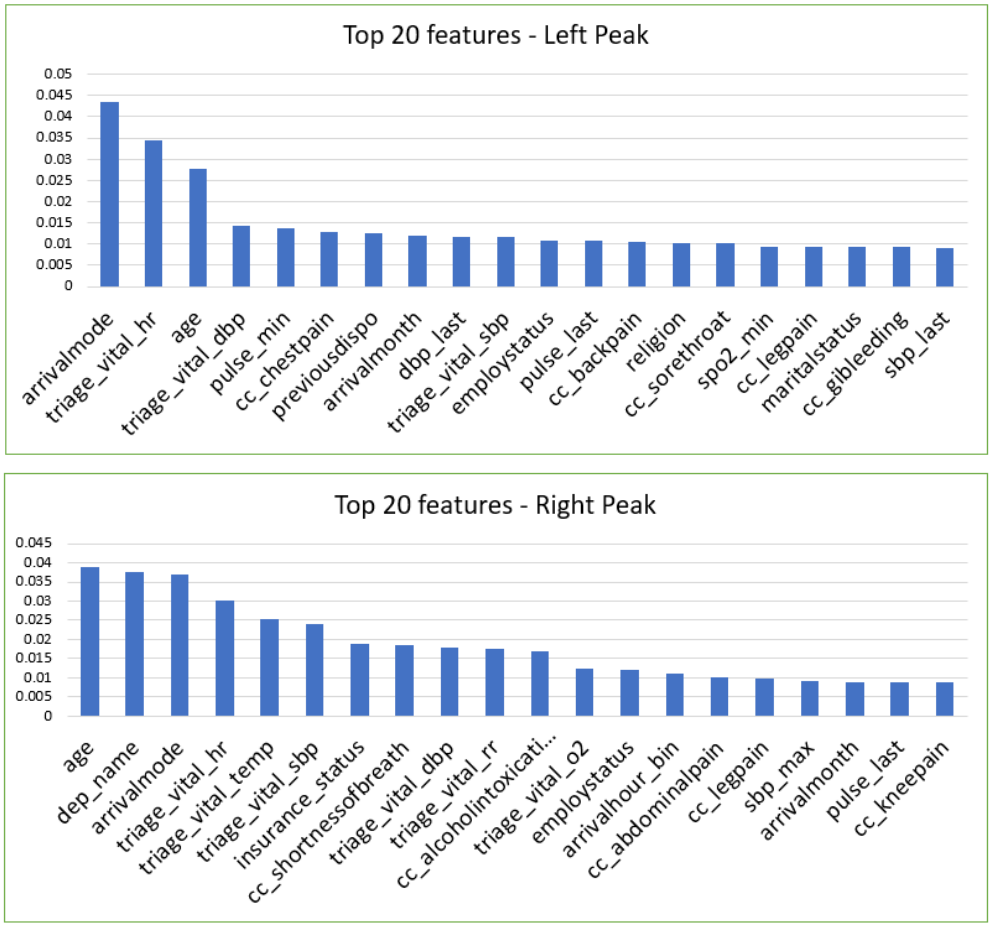


Figure 3.34: Top features as ranked by XGBoost in the two selected peaks from sparse embedding and their feature importance. Reprinted/adapted with permission from [5].

3.3.3 Conclusion

In this paper, we proposed an interpretable framework for hospital admission prediction using EHR data. The model has a hierarchical structure in which medical data are embedded into sparse vectors to force feature concentration. The embeddings were then visualized through t-SNE. Patient data distribution showed that sparse embedding can push hidden phenotypes in the admitted cohort to be more prominent, driven by the binarization of the features in the embedding. For future work, we will further investigate the effect of sparsification and binarization, with the premise that the lost accuracy can be better justified if we can quantify how much interpretability we can gain by using our model. One example can be that the binary embedded dimensions are indica-

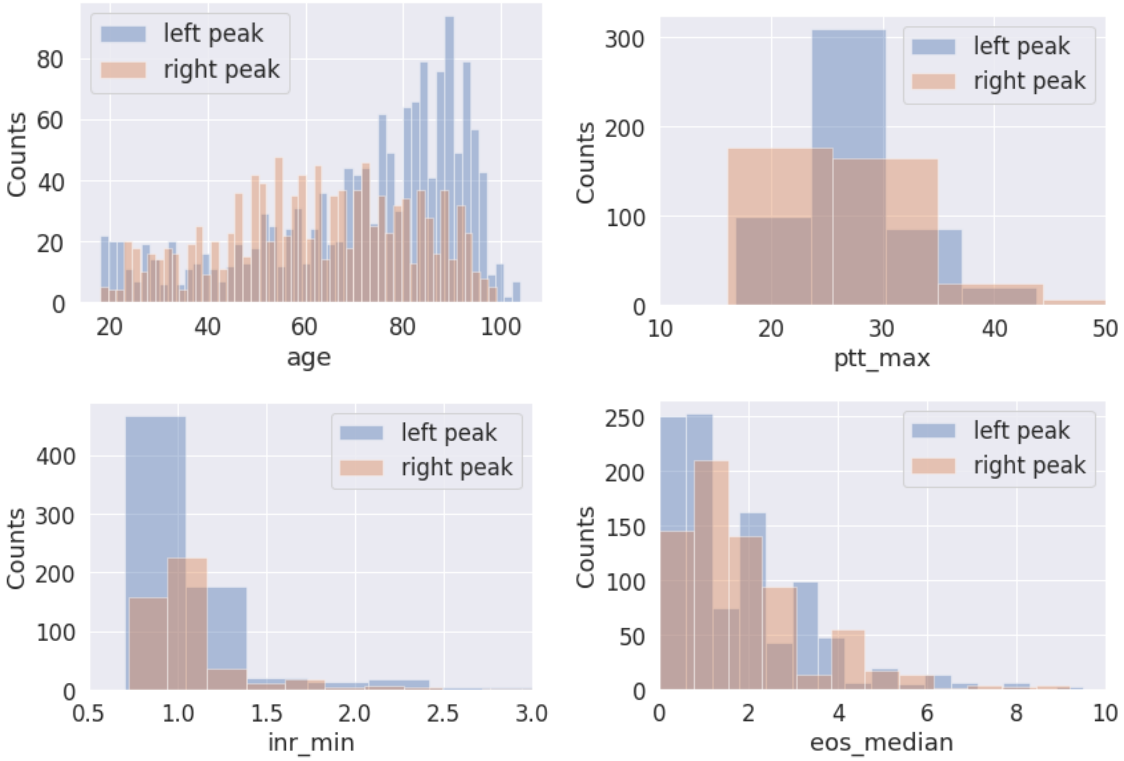


Figure 3.35: Age, Partial Thromboplastin Time, international normalized ratio, and Eosinophils difference between peaks in sparse embedding. Reprinted/adapted with permission from [5].

tors for phenotypes we found in the cohort, and sparsity can further help them to be more human interpretable.

4. THIRD AIM: ADJUSTING AND TAILORING TO HETEROGENEITY

4.1 DynImp: Dynamic Imputation for Wearable Sensing Data Through Sensory and Temporal Relatedness[1]

4.1.1 Introduction

4.1.1.1 Overview

Due to the changing dynamics in the data, the heterogeneity itself is constantly evolving and thus giving difficulty to modeling. A perfect example can be the data missingness in wearable sensors. Due to its temporal characteristics, the missingness may present different trends in different timestamps. Therefore a model which assumes the data is from i.i.d. might not be realistic to model the missingess.

We propose to use a long-short-term-memory-based denoising autoencoder (LSTM-DAE) to learn more robust imputation strategies for remote sensing data. The model has an encoder and decoder architecture to embed signal data, then this encoded information is fed into the LSTM network to learn the time-varying dynamics of the data. The overall structure is shown in Fig. 4.1. This architecture robustly imputes missing data from related channels and latest dynamics being measured by all sensor channels, even in the presence of high rates of missing data. In order to demonstrate the utility of using both time-series dynamics and feature relatedness, we experiment on datasets with both inherent missing values and increased missing data (up to 60% missing across all the channels on a dataset with inherent 66% missingness), which surpasses traditional missing rate study by a large margin.

[1] Reprinted with permission from Huo, Z., Ji, T., Liang, Y., Huang, S., Wang, Z., Qian, X. and Mortazavi, B., 2022, May. Dynimp: Dynamic Imputation for Wearable Sensing Data through Sensory and Temporal Relatedness. In ICASSP 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 3988-3992). IEEE. Copyright 2022 by the authors.

4.1.1.2 Related Work

There has been numerous studies on missing-data approaches which brought about promising results for down-stream modeling. MissForest can handle mixed types of features and of missing data value by using a tree-based method [210]. In [211], the authors proposed a bi-clustering based data imputation technique using the mean squared residual metric that estimates the degree of coherence between each recorded cell of the dataset. In [212], the authors present the an imputation method for missing data value in Internet of Things (IOT) device data, by applying context-based linear mean, binary search method as well as a Gaussian mixture model. There are many deep learning based methods as well, in [213], the authors a multi-layer perceptron coupled with interpolation technique. In [214], the authors chose not to directly impute, but rather treated the missing values as extra source of information and used an LSTM model to train and predict the time series data. This work will evaluate the strengths of each of these techniques in comparison to the work proposed here.

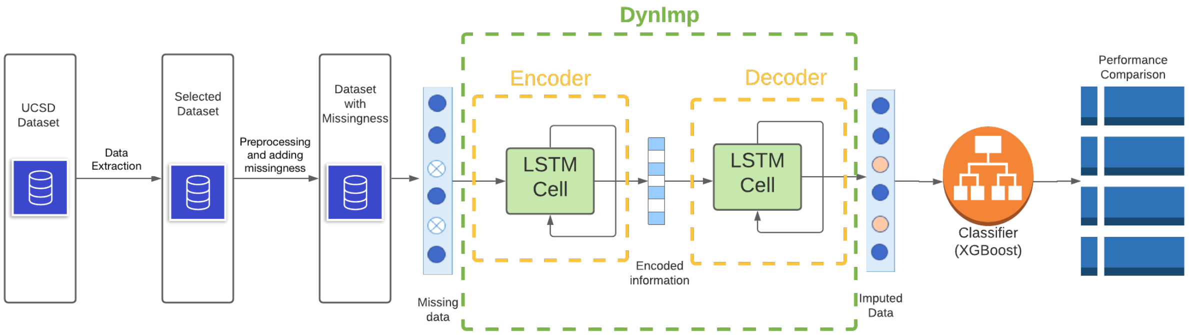


Figure 4.1: Pipeline for Dynamic Imputation. Reprinted/adapted with permission from [6].

4.1.2 Methodology

Our proposed architecture is illustrated in Fig. 1. We will discuss the building components of them in each following subsections. The architecture is split into an encoder/decoder architecture. The time-series information is modeled by a LSTM-based autoencoder, coupled with KNN-padding to exploit the feature relatedness in wearable data.

4.1.2.1 Denoising Autoencoder

The aims for autoencoder are delivered by learning representations (encoding) of a set of data and reconstructing (decoding) the data from these representations. Through this encoding and decoding process, the network can take care of data missingness and possible signal noise that lies within the data. A generic autoencooder would be formulated as:

$$\begin{aligned} \theta^*, \theta'^* &= \arg\min_{\theta,\theta'} \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}) \\ &= \arg\min_{\theta,\theta'} \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{x}^{(i)}, g_{\theta'}(f_\theta(\mathbf{x}^{(i)}))), \end{aligned} \tag{4.1}$$

where the encoding function is $\mathbf{y} = f_\theta(\mathbf{x}) = f(\mathbf{W}\mathbf{x} + \mathbf{b})$, and the decoding function to map the latent representation back is $\mathbf{z} = g_{\theta'}(\mathbf{y}) = g(\mathbf{W}'\mathbf{y} + \mathbf{b})$. L is represented as the loss function, which can be taken as the form of reconstruction cross-entropy:

$$L(\mathbf{x}, \mathbf{z}) = -\sum_{k=1}^{d}[\mathbf{x}_k \log \mathbf{z}_k + (1 - \mathbf{x}_k)\log(1 - \mathbf{z}_k)]. \tag{4.2}$$

The denoising autoencoder is to enforce some extra noise on the input vector so the input vector will be *corrupted*. This is based on the assumption that the core data representation from a large amount of training example will stay relatively stable even with background noises so the decoder will learn to distinguish that from the stochastic noise [5, 215]. The stochastic mapping can be written as $\tilde{\mathbf{x}} \sim q_D(\tilde{\mathbf{x}}|\mathbf{x})$. The mapping function can take various forms to corrupt the input vector. We used a stochastic dropout as the function

$$\begin{aligned} &r_j \sim \text{Bernoulli}(p), \\ &\tilde{\mathbf{x}} = \mathbf{r} \cdot \mathbf{x}, \end{aligned} \tag{4.3}$$

where $\mathbf{r}$ is a vector of independent Bernoulli random variables, each of which has probability p of being 1. And $*$ is an element-wise product, so this vector is sampled and multiplied element-wise with the input to give the *corrupted* signal.

4.1.2.2 Padding for Denoising Autoencoder

Traditional methods for training the neural networks would usually pad missing data with interpolated values or simply zeros [216]. However for a dynamic imputation scenario, the missingness should be captured with an adaptive manner, which means as time goes on, the imputation should also adjust with newer data coming in. We propose to integrate K-NN imputation with the neural network. Therefore, in each time point, the model will have different nearest neighbors, and the combination with autoencder can be used to capture this higher level of dynamics. Therefore our hidden layer function can be further written as:

$$h(\tilde{\mathbf{x}}^{(i)}) = g[\mathbf{W}' \cdot ((\mathbf{M}^t \odot \tilde{\mathbf{x}}^{(i)}) + \mathbf{P}) + \mathbf{b}], \tag{4.4}$$

where $\mathbf{M}^t$ is a masking matrix for indicating the missingness in input data, $\odot$ is element-wise production operator. Assume we have i-th data entry missing on t-th time point:

$$\mathbf{M}^t = \mathbb{1}_{(m,l)} \begin{cases} 0 & \text{if } i = m, t = l, \\ 1 & \text{otherwise}, \end{cases} \tag{4.5}$$

where $\mathbb{1}$ is an indicator function. Furthermore the imputation matrix with nearest neighbor padding $\mathbf{P}$ is as follows:

$$\mathbf{P}^t = \begin{cases} \frac{\sum_j \mathbf{x}^{(j)}}{k} & \text{if } j \text{ in top } k \text{ neighbors of } i, \\ 0 & \text{otherwise}. \end{cases} \tag{4.6}$$

The missing value would be from the mean of k closes neighbor's average, where the distance is measured in Euclidean distance. Therefore we have constructed our denoising autoencoder with nearest neighbor padding technique for time-series data.

Table 4.1: Results for comparison between baseline imputation and dynamic imputation (in BA and its confidence interval). Reprinted/adapted with permission from [6].

Level of missingness		Filled Mean	kNN imputer	Missforest	SparseSense [MLP]	Indicator Variable [LSTM]	DynImp [LSTM-DAE]
mild	10%	0.8259 ± 0.0069	0.8288 ± 0.0046	**0.8458** ± **0.0027**	0.8213 ± 0.0032	0.8248 ± 0.0085	0.838 ± 0.0095
	20%	0.8065 ± 0.0051	0.8001 ± 0.0069	0.8364 ± 0.0052	0.8108 ± 0.0077	0.8081 ± 0.0078	**0.8390** ± **0.0045**
medium	30%	0.7871 ± 0.0043	0.7802 ± 0.00097	0.8131 ± 0.0046	0.7901 ± 0.0060	0.7852 ± 0.0060	**0.8318** ± **0.0012**
	40%	0.7627 ± 0.0092	0.7570 ± 0.0106	0.7918 ± 0.0072	0.7647 ± 0.0079	0.7624 ± 0.0105	**0.826** ± **0.0042**
severe	50%	0.7367 ± 0.0087	0.7351 ± 0.0067	0.7541 ± 0.0066	0.7401 ± 0.0095	0.7401 ± 0.0104	**0.831** ± **0.0071**
	60%	0.7043 ± 0.0085	0.7087 ± 0.0086	0.7131 ± 0.0093	0.6907 ± 0.0122	0.6907 ± 0.0071	**0.8304** ± **0.0070**

4.1.2.3 LSTM-based Autoencoder

In order to exploit time-series dynamics, we propose that the fully connected layer in traditional autoencoder would need to replace the function $\mathbf{y} = f_\theta(\mathbf{x}) = f(\mathbf{W}\mathbf{x} + \mathbf{b})$ to a LSTM cell, where each input feature $\mathbf{x}$ has a time-stamp, i.e. $\mathbf{x}_t$.

First the LSTM cell will generate a decision vector and select the candidate information. For the current time stamp t, the vector I_t is a function of last hidden state h_{t-1} and input feature x_t, and the output gate will generate the hidden state h_t conditioned on the output:

$$
\begin{aligned}
I_t &= f_i(w_i x_t + w_i h_{t-1} + b_i), \\
F_t &= f_g(w_g x_t + w_g h_{t-1} + b_g), \\
\tilde{C}_t &= f_C(w_c x_t + w_c h_{t-1} + b_c), \\
C_t &= C_{t-1} F_t + \tilde{C}_t I_t, \\
Y_t &= f_o(w_o x_t + w_o h_{t-1} + b_o), \\
h_t &= Y_t f_h(C_t).
\end{aligned}
\tag{4.7}
$$

This hidden state will replace the original encoding function output $f_\theta(\mathbf{x}) = f(\mathbf{W}\mathbf{x} + \mathbf{b})$ and thus will become the encoded information that is later fed into decoder for reconstruction, following similar steps as above for time-series imputation.

Table 4.2: Results for variants of DynImp for missingness. Reprinted/adapted with permission from [6].

DynImp Variations	10%	20%	30%	40%	50%	60%
(0-padding)	0.8262	0.8219	0.8239	0.8236	0.8288	0.8299
(Filled mean)	0.8266	0.8224	0.8219	0.8245	0.8232	0.8254
(Interpolation)	0.8284	0.8103	0.8291	0.8201	0.8263	0.8290
(kNN)	**0.8380**	**0.8390**	**0.8318**	**0.8260**	**0.8310**	**0.8304**

4.1.3 Experiment

4.1.3.1 Dataset

We use the UCSD ExtraSensory dataset for these experiments [217]. The UCSD ExtraSensory dataset contains data from 60 users (34 female and 26 male). Data were collected through a user's personal smart phone (34 iPhone, 26 android). The sensors include high-frequency motion reactive sensor (accelerometer, gyroscope, magnetometer, etc.). We chose this dataset because the it collects motion-reactive sensor data and the dynamics of one sensor has potentiality to be recovered by another one due to underlying relatedness [217]. The inherent missingness is 66% for each sensor per their description. We also generate extra missingness at random across the sensors to stress the limits of the traditional methods, where we randomly mask out the data points in the 2D data matrix which has a feature column and time column. And the rate to delete the data is ranging from 10% to 60%.

4.1.3.2 Experimental Setup

We selected a group of features that represent data collected widely from wearable sensors [217]. We considered raw measurements for the accelerometer, gyroscope, and magnetometer across all three channels (axis). We also selected two features for the location sensor, which were the mean absolute longitude and latitude. The total number of features (including all the axes of each feature) is 14. We use a sliding window technique, where the window length is 24 minutes and features are grouped by 1 minute intervals. So for each window we will use the features from each sensor to predict the body movement label. Regarding to the labels, we have two concurrent labels for the users which are 4 body movements and 4 phone locations (where on the body). The 4 body movements labels which are mutually exclusive in nature and combined into one multiple class label, which are 'LYING_DOWN', 'SITTING', 'FIX_walking', and 'others'. With these 4 body movement labels, as well as their combinations with the 4 phone locations on the body, we have 16 labels in all the combinations. We used classification report from sklearn, and recorded balanced accuracy (BA) as the model performances evaluation [217]. In order to achieve a stabilized result and perform fair comparisons, we composed each kinds of testing with different setup mentioned above 10 times (each time with different seed for random missingness generation) and average the BA in validation set, and the corresponding confidence interval. Regards to different baseline imputation method and different state-of-the-arts, we picked:

- Filled mean: each missing value is computed from the average of the time-series for that sensor feature.
- kNN imputer: each missing value is computed by finding the nearest neighbors. The distance is measured by euclidean distance of the remaining values for that sensor feature to the neighbors.
- Missforest: a tree-based imputation method that learns to predict the missing values
- SparseSense (MLP) [213]: a neural network based framework that combines interpolation and Multi-Layer Perceptron to predict the missing value
- Indicator Variable (LSTM) [214]: a LSTM-based method to use missing value as extra features, an indicator (masking) feature vector to indicate the imputed value is missing or not, where 1 is missing and 0 is not.

4.1.3.3 Results

We put the results of experiment into Table 4.1, illustrating the comparison between baselines. The models performance were introduced regarding the balanced accuracy (BA) [217] in validation set. The row denotes for increasing random missingness generated in the data. From the result tables overall we can discern a pattern for traditional imputation methods: as the level of missingness increases, the model performance decreases drastically. We notice that Missforest can perform relatively well albeit marginal, when the missingness is mild, but still degrades quickly when the missingness enlarges. We hypothesize that when coupled with similar tree-based classifier, i.e. XGBoost, the Missforest can reconstruct the data that is best for a boosting classifier. One previous study [218] showed the Missforest coupled with XGboost can perform relatively well on mild missingness scenario (2.19% to 13.63%). But for our model, it outperforms not only the traditional methods but also the methods that utilizes the neural networks such as MLP and LSTM. The performs can hold even with extremely severe missingness.

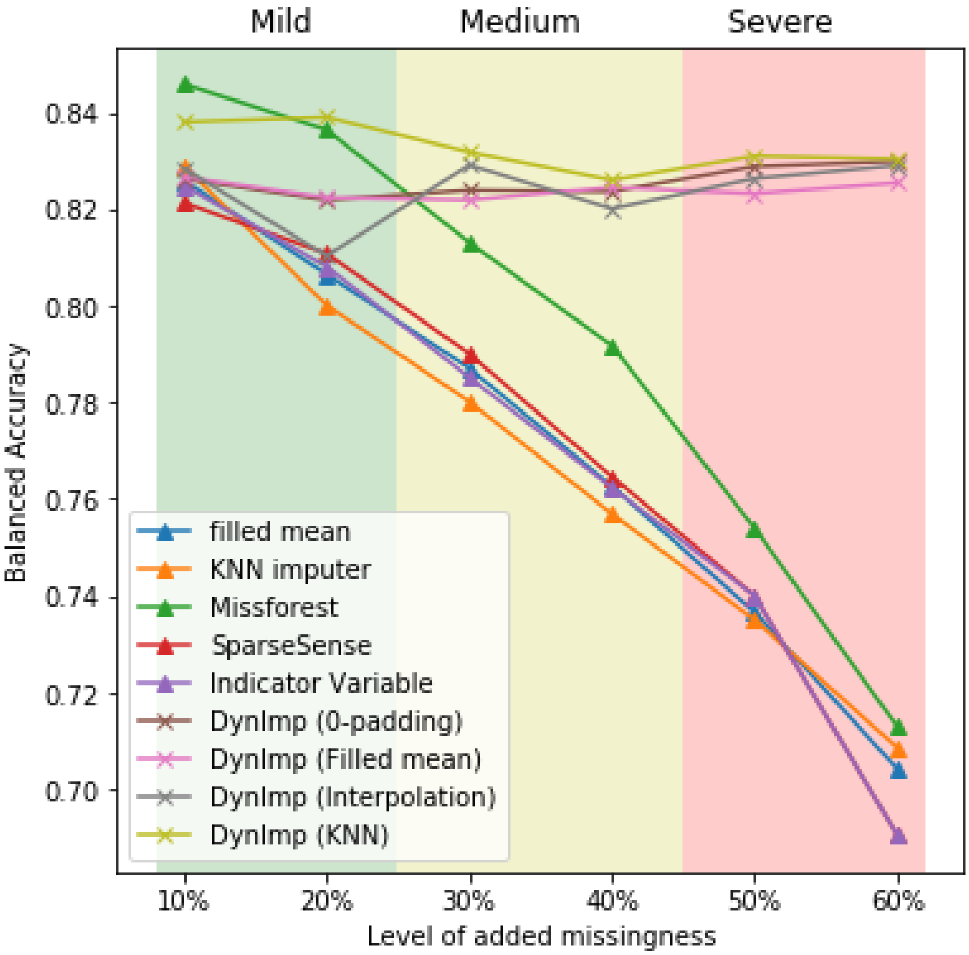


Figure 4.2: Results comparison. Reprinted/adapted with permission from [6].

4.1.3.4 Ablation Study

We have established the improvement of our model by making use of LSTM-DAE pipeline. Next in order to study the necessity of using nearest neighbor as the padding strategy in our architecture we conduct a comparison among the variants:

- DynImp (0-padding): the missingness is replaced by zeros of the time-series for that sensor feature before feeding to DynImp for training [216]
- DynImp (Filled mean): the missingness is replaced by average of the time-series for that sensor feature before feeding to DynImp for training [116]
- DynImp (Interpolation): the missingness is replaced by interpolation of before and after values before feeding to DynImp for training [213]
- DynImp (kNN): the missingness is replaced by kNN-imputer for $k = 5$ neigbhors, before feeding to DynImp

As we can see in Table 4.2 the superior performance of our proposed scenario, even with the same network architecture, the nearest neighbor method can further push the LSTM-DAE to have better performance than baselines. A clear illustration of all methods is also shown in Fig 4.2.

4.1.4 Conclusion

In this paper we proposed a dynamic imputation technique for remote sensing data. The method is through a trainable mechanism by the use of deep learning model to learn the missing dynamics along the time axis to impute the missing data. The model shows strong performance compared to baselines. While not particularly outstanding in mild missingness scenario, the model can maintain high prediction accuracy in some extreme missingness scenarios. We also tested on variations of the model to find out the best performing one. This dynamic imputation can be classifier-agnostic so it can be compatible with any downstream methods. In the future we aim to study how does the model perform in other scenarios in terms of different data types and features.

4.2 Using Continuous Glucose Monitors to Estimate the Macronutrient Composition of Meals[2]

Diet monitoring is an important component of interventions aiming to help patients manage blood glucose levels. However, existing diet monitoring methods are inaccurate and time-consuming. To address this problem, we propose a method to predict the composition of a meal automatically from its associated glucose response, as measured by a continuous glucose monitor (CGMs). The approach relies on the observation that the glucose response to a meal depends on its macronutrient composition (i.e., carbohydrates, proteins, and fats). To capture this information, our method computes the area-under-the-curve (AUC) at different time intervals during the glucose response to the meal using a family of Gaussian kernels. These AUCs are then passed to a multitask neural network trained on the glucose response to a set of standardized meals. To test the model, we conducted a study in which participants consumed nine meals with known macronutrients while wearing a CGM. We compared our model against two baseline methods, a linear regression model and a single-task neural network. Our model outperformed both baselines, as measured by the correlation coefficient between prediction and ground truth, and classification accuracy, in both subject-independent and subject-dependent cases. These findings provide support for using CGM signals to automatically predict food intake.

4.2.1 Introduction

One hundred and twenty million Americans are glucose intolerant. Glucose intolerance results from the body's inability to properly produce and respond to insulin sufficiently. Unless treated, glucose intolerance progresses from pre-diabetes to the severe form, known as type 2 diabetes mellitus (T2DM). This intolerance is often manifested by high glucose responses after a meal. Therefore, an essential component of clinical interventions for T2DM is to monitor dietary intake to control blood glucose levels after eating a meal. However, conventional methods for diet

[2] Reprinted with permission from Huo, Z., Mortazavi, B.J., Chaspari, T., Deutz, N., Ruebush, L. and Gutierrez-Osuna, R., 2019, May. Predicting the meal macronutrient composition from continuous glucose monitors. In 2019 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI) (pp. 1-4). IEEE. Copyright 2019 by the authors.

tracking rely on manual input, which is burdensome.

An unexplored opportunity to address the burden of logging diet has emerged with the advent of continuous glucose monitors (CGMs). CGMs are inserted under the skin (subcutaneously) with a small electrode to measure glucose levels in interstitial fluid. CGMs make it possible to capture the blood glucose response to a meal, which is known to depend on the macronutrient composition (i.e., carbohydrates, proteins, fats) of the meal. As an example, adding fat and protein to carbohydrates generally reduces the glucose response and slows down the return to baseline [219]. This suggests that the shape of the glucose response to a meal can be used to recover the macronutrient composition of the meal, and therefore be used to log food intake automatically.

To test this concept, we conducted a study in which healthy subjects were asked to consume a set of standardized meals (i.e., of known carbohydrates, proteins, and fats) while wearing a CGM. Then, we built a neural-network regression model to estimate the macronutrient composition from the measured post-prandial glucose response (PPGR) as captured by the CGM. Initial results from this model were presented at the 2019 IEEE Biomedical and Health Informatics (BHI) conference [7]. This manuscript builds upon on our earlier results by evaluating two techniques to improve the accuracy of the model, which normalize the glucose response relative to the subjects' fasting glucose levels and their body composition. We also present a post-processing technique that improves the interpretability of the results by classifying meal compositions into three categories: low, medium, and high amounts of each macronutrient. The contributions of this paper are as follows:

- We present a feature extraction technique that uses a family of Gaussian kernels to capture the shape of the glucose response
- We propose a multitask neural network model that predicts the macronutrient composition of a meal from the shape-related features, and
- We show that taking into consideration the body composition of subjects improves the accuracy of the model

The rest of this paper is organized as follows. subsection 4.2.2 discusses related research and findings. subsection 4.2.3 discusses the the multi-task learning method implemented, while subsection 4.2.4 discusses the experiments and findings. subsection 4.2.5 discusses the significance of those findings, while subsection 4.2.6 highlights the limitations and future direction opportunities, and subsection 4.2.7 concludes this work.

4.2.2 Related Work

4.2.2.1 Diet Monitoring Technology

Diet is typically tracked done using a written diary. This requires a high level of commitment and often leads to poor adherence [220]. Several technology-based approaches have been proposed to address this issue. As an example, computer vision techniques can be used to estimate the contents of a meal. Hassanejad et al. surveyed computer vision techniques and wearable sensor techniques in order to understand accurate food intake [221]. Their results show that vision based solutions provide opportunity, however additional work is done, including needing reference food image datasets, image segmentation is challenging, and while promising, most of this work needs additional validation outside of laboratory environments [221]. More recently, Zhu et al. addressed this image segmentation problem [222] and developing an end-to-end system that converts images of food to caloric estimates[223]. Their results show that known meals can be estimated within 209 kcals, providing accurate estimates of energies for meals that were prepared for participants. While their results are promising, they acknowledge a limitation of their system involves developing a larger database of training images for the system to recognize, which remains a limitation of vision-based solutions.

Wearable sensors can also be used to detect moments of dietary intake. For instance, Kalantarian et al. [224] surveyed body-worn accelerometers to identify moments of food intake by recognizing hand and arm gestures typical of eating behaviors, which presents a number of sensing-based solutions that focus first on detecting moments of dietary intake, then on the types of food ingested. This included their own work using a sensor-based necklace to detect swallows [225].

Using their method, they were able to achieve accurate determinations of swallows related to liquids versus solid foods. Other studies surveyed focused on acoustic measurements with body-worn microphones to characterize swallows and chewing [226, 227]. Body-worn accelerometers have also been used to identify moments of food intake by recognizing hand and arm gestures typical of eating behaviors [224]. One approach to the sensor-based solution was to use smart glasses to merge the wearable sensors domain with the computer vision approaches to the diet problem [228]. Zhang and Amft show that their system is able to translate from highly accurate laboratory detection of eating moments (94%) to free environments with good accuracy (77%), however the detection of the content of those meals was not studied. Some efforts involve instrumenting the utensils used for eating, such as the smart fork work by Kadomura et al. [229]. Their results show that eating behaviors can be detected, tracked, and modified, but do not focus on the contents of those meals. Additional sensor-based methods detect moments of dietary intake with some accuracy [230, 231, 232], demonstrating the significance of the diet intake problem, but are not suitable to estimate the contents of those meals, which is far more critical.

4.2.2.2 Measuring Blood Glucose

The most common method to measure blood glucose is through a glucometer, a device that requires pricking the finger with a lancet. This procedure is inconvenient and painful, and only measures glucose responses at specific times. CGMs dramatically reduce this burden and allow glucose levels to be measured automatically and continuously (every 5 to 15 minutes, depending on the CGM model). This frequent measurement makes it possible to capture detailed information about the glucose response to a meal that would otherwise be lost with a traditional glucometer. CGMs are commonly used for managing type-1 diabetes, where monitoring glucose levels is critical, but have yet to make an impact in the management of type-2 diabetes [233], the more prevalent of the two conditions (90%). Earlier CGM instruments required users to calibrate the sensor with a finger-prick measurement multiple times a day, but the newest generation of CGMs are able to monitor glucose levels uninterruptedly for up to 14 days without requiring calibration. Earlier CGMs were also limited by their high cost, but prices have dramatically fallen in the new generation of "flash" CGMs (i.e., Abbot Freestyle Libre) and many of these devices are now covered by Medicare. This broadens the range of applications of CGMs beyond its primary use in type-1 diabetes.

4.2.2.3 Modeling Glucose Responses

Differences in macronutrient composition in meals lead to varied PPGRs. For example, simple carbohydrates result in steep glucose spikes, while proteins and fats yield smoother, longer fluctuations [232, 234]. A study examining 38 foods with isoenergetic portions found that protein-rich foods produced the highest insulin secretion per gram of carbohydrate, followed by bakery products, snack foods, fruits, carbohydrate-rich foods, and cereals [235]. A number of studies have tried to estimate periods of hyperglycemia and hypoglycemia as measured by CGMs. One such model tries to learn and restrict what segments of glucose response can look like in T1D patients before and after meal intake, in order to estimate the most likely periods of hypo- and hyperglycemia [236]. By adding gut microbiome measurements, it is possible to personalize estimations of hypo- and hyperglycemia to improve accuracy [18]. In addition to the improvement in hypo- and hyperglycemia estimates provided by the gut microbiome, additional sources of information, such as daily physical activity, further improves the hypo- and hyperglycemia predictions [237]. Additional studies have focused on the variability present in glucose response as a result of a variety of different mixed meals (of varying energy, carbohydrates, protein, and fat) [238, 239]. Wolever and Jenkins, for example, studied the impact of proteins and fats on glycemic index, showing that the glucose response differences in meals with varied proteins and fats can be described through quantitative measures of the glycemic index of a meal [240]. Wolover and Bolognesi took meals of varied glycemic indices in order to predict the glucose and insulin response [238]. Their findings demonstrated that the carbohydrates alone did not describe the glucose response (nor did proteins or fats), but that the glycemic index did describe the variation.

Rozendaal et al. provided healthy participants varied/mixed meals to evaluate glucose response dynamics [239]. Their study found that area under the glucose response curve, as well as the height of the peak, width, and rise time were necessary measurements to describe the dynamics of the glucose response. We consider features that extract area under the curve measurements at different time frames to capture this information in our study. Gonzalez-Rodriguez et al. studied the glucose repsonse in adults to determine if there were differences in men and women [241]. Their study examined the impact of clinical factors such as age, sex, and body composition to describe glycemic responses, and concluded that CGMs provide important information about meal responses and that the clinical characteristics described some of the variability recorded [241]. In a landmark study [242, 243], researchers built a model that was capable of predicting the glucose response to a variety of meals, including a number of personalized characteristics. In their study, Zeevi et al. developed machine learning methods that looked at gut microbiome, physical activity levels, and a number of other characteristics to categorize and estimate the glucose responses from a variety of meals in a study of 100 individuals. Their study found that the glucose responses were driven by both the composition of the meal as well as characteristics that personalize the each individual, requiring accurate models to account for subject-to-subject variability in glucose response to similar foods [242]. In contrast with these studies, which focus on predicting glucose responses to different types of meals, our goal is the reverse: predicting meal composition from their post-prandial glucose responses.

4.2.3 Neural Network Model

We aimed to design a model that can estimate macronutrient quantities from the PPGR signal captured by the CGMs. The model would need to take features that represent the PPGR over the 8 hour study period, and inform us of the shape, amplitude, and general area under the curve, to represent the size of the meal. Zeevi et al. demonstrated further that the combination of of carbohydrates, proteins, and fats all influence the glucose response. As a result, we needed a model that could incorporate features of the PPGR and simultaneously learn the impact of carbohydrates, proteins, and fats on that signal, rather than learning each individually.

We designed and evaluated a multitask neural network to estimate the macronutrient composition of meals. The basic architecture of the model is illustrated in Fig. 4.3. The network takes the glucose response to a meal and predicts the amount of macronutrients: carbohydrates, protein, and

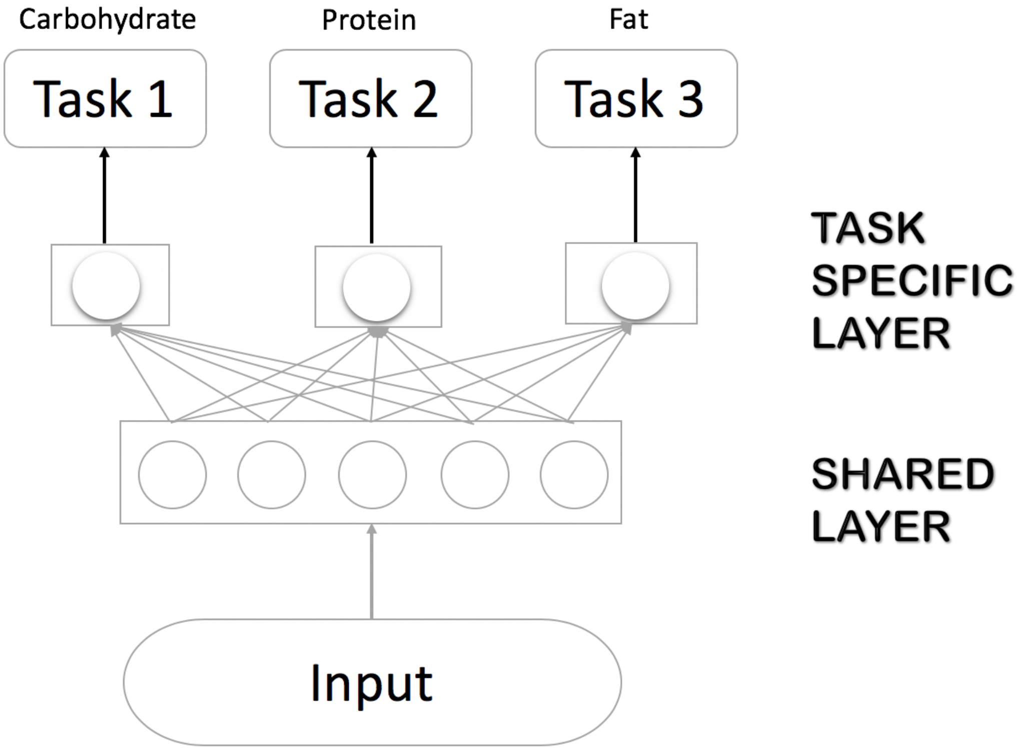


Figure 4.3: Structure of the multitask neural network. Reprinted/adapted with permission from [7].

fat. The network consists of a shared layer that generates a representation or embedding to capture information that is common to the three macronutrients and a task-specific layer that predicts the amount of each individual macronutrient. The hidden units use a Rectified Linear Unit (Relu) activation function to capture non-linear information in the glucose response, and a linear activation function for the task-specific layer to span the full range of macronutrient levels.

We extract a number of features from the glucose response, which are provided to the multitask neural network as inputs. These features represent the area-under-the curve (AUC) of the glucose response at various times over the 8 hours that follow consumption of a meal. The feature-extraction process is illustrated in Fig. 4.4. We distribute a family of Gaussian kernels uniformly over the glucose response, and then computed the AUC weighted by those kernels, with and without subtracting the baseline glucose level (see subsections 4.2.4.3 and subsections 4.2.4.4), measured at time zero after consuming the meal. This allows us to capture the initial rise time, the

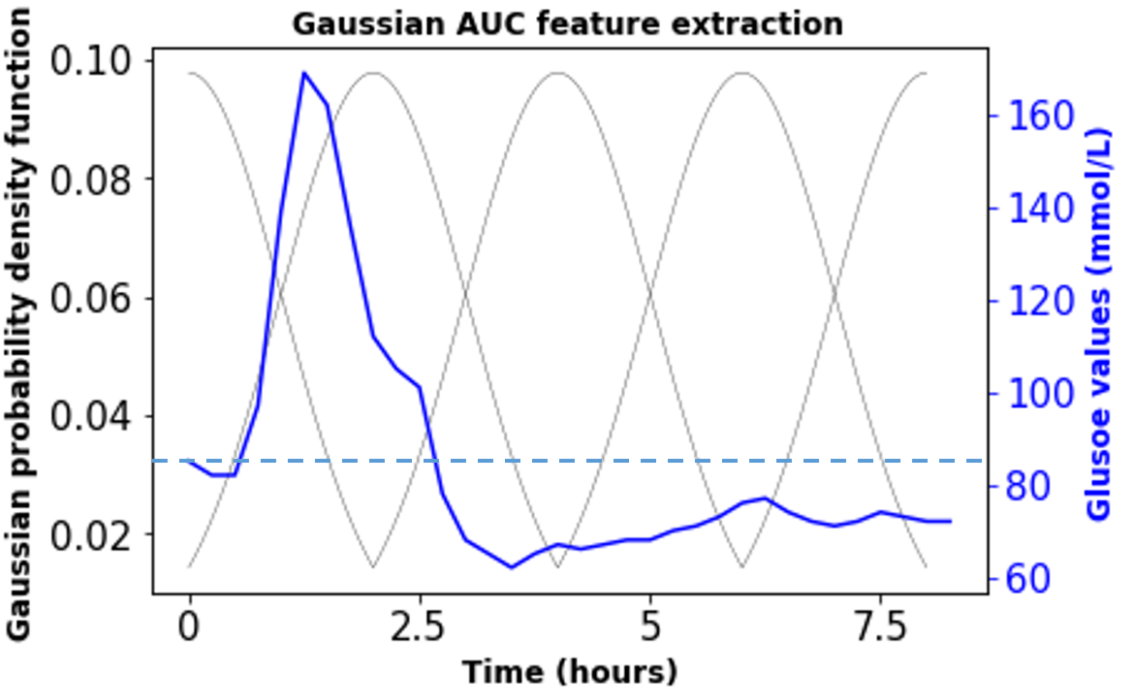


Figure 4.4: The extraction of 5 Gaussian AUC features, extended from [7] with the fasting glucose subtracted to generate relative AUC values, over an 8 hour window. Reprinted/adapted with permission from [7].

duration of the elevated glucose response, and the recovery back to the baseline glucose level. We set the standard deviation of the Gaussian kernel to σ = n/1.96 where n is the number of sensor readings within a time interval. Fig. 4.4 shows the case for five Gaussian kernels, but different numbers of kernels can be used to capture the glucose response at different levels of temporal resolution.

We train the neural network using back-propagation and the Huber loss function. The Huber loss uses a quadratic term for small errors and a linear term for larger errors, thus providing robustness to outliers. The neural network has a hyper-parameter: the number of neurons in the hidden layer, which we optimize using grid search through internal cross-validation, searched from 1 to the number of inputs provided to the model. Using this technique (details not shown), we determine that the optimal number of hidden neurons is equal to the number of inputs. We select these optimal hyper-parameters as those that maximize the Pearson correlation between ground truth (amount of each macronutrient) and predicted values.

Table 4.3: Macronutrient composition of the nine meals and their classification as low (1), medium (2), or high (3), for carbohydrates (C), proteins (P), and fats (F). Reprinted/adapted with permission from [7].

Meal Name	Carbohydrates (g)	Protein (g)	Fat (g)
C1P1F1	52.25	15	13
C2P2F2	94.75	30	26
C3P3F3	179.75	60	52
C1P2F2	52.25	30	26
C3P2F2	179.75	30	26
C2P1F2	94.75	15	26
C2P3F2	94.75	30	26
C2P2F1	94.75	30	13
C2P2F3	94.75	30	52

The network is a regression model that estimates a continuous value –the amount of each macronutrient, but can also be used as a classification model. This requires discretizing the network outputs into several categories [3]. In our case, we discretize each output into three levels, representing high, medium and low amounts of each macronutrient. We accomplish this by identifying two thresholds across the continuum of outputs. We optimize each threshold separately. First, we find the optimal threshold between low amounts and high/medium amounts of macronutrients by linearly sweeping the threshold forward, as in the development of a receiver operating characteristic (ROC) curve. We then repeated this process to find the ideal high vs. medium/low threshold by beginning at the maximum value and sweeping backward.

We evaluate the model using two measures. For the regression task, we use the Pearson correlation between ground truth macronutrient values and estimated values, for the classification task, we analyze the confusion matrix of the model's estimates, which we then convert into the conventional measures: precision and recall. All the code for this study can be found online[4].

4.2.4 Meal Macronutrient Results

This study proposes a technique to predict the amount of macronutrients in a meal from the shape of the glucose response of the meal, using continuous glucose monitors and a novel multitask

[3] With this technique, we avoid the problem of misclassifying the two extreme categories (i.e. classifying lows and highs, and vice versa), which would be a possibility if the network were trained as a ternary classification model.

[4] https://github.com/ZepengHuo/CGMmacronutrition

neural network. Our results indicate that the amount of carbohydrates (and to a lesser extent fats but not protein) can be predicted, as measured through regression and classification tasks. We validated our approach in a series of experiments. We find that subject-dependent models, where a separate model is trained for each subject, significantly outperform a subject-independent model, where a single model is trained by pooling data from multiple subjects. This is consisted with other studies which have shown that metabolism and therefore the glucose response to a meal is unique to each individual [242]. We find that the multitask network, which jointly predicts the amount of the three macronutrients, outperforms a single-task method that uses three separate networks (one for each macronutrient).

We explored several techniques to improve model performance. First, we find that subtracting fasting glucose levels improves performance when compared to using the absolute glucose response. Second, we find that normalizing the amount of macronutrients relative to body composition (e.g., body weight) also improves model performance. These results show that fasting glucose and body composition are part of the phenotype of each subject. Finally, we also find that features extracted a multiple levels of resolution do not improve performance compared to using features at a single time scale. We also examined evaluated the performance of the model when used as a classification task, where the amount of each macronutrients was discretized into three labels: low, medium and high. We find that obtaining class labels by training the neural network as a regression model, and then discretizing the model outputs, provides higher classification performance than training the network directly as a ternary classifier.

4.2.4.1 Data Collection

We recruited seven healthy subjects (not diagnosed with T2DM or pre-diabetes) ages 60-85 years and Body Mass Index in the range of 25-35. Each subject participated in 9 study days in which they consumed a predefined meal in a randomized design. Each study day lasted approximately 8 hours and the procedures on the study days were identical, with the only change being the macronutrient composition of the meal taken (e.g. varying low and high values of carbohydrates, proteins, and fats). Subjects were asked to fast for at least 8 hours prior to the meal intake on each

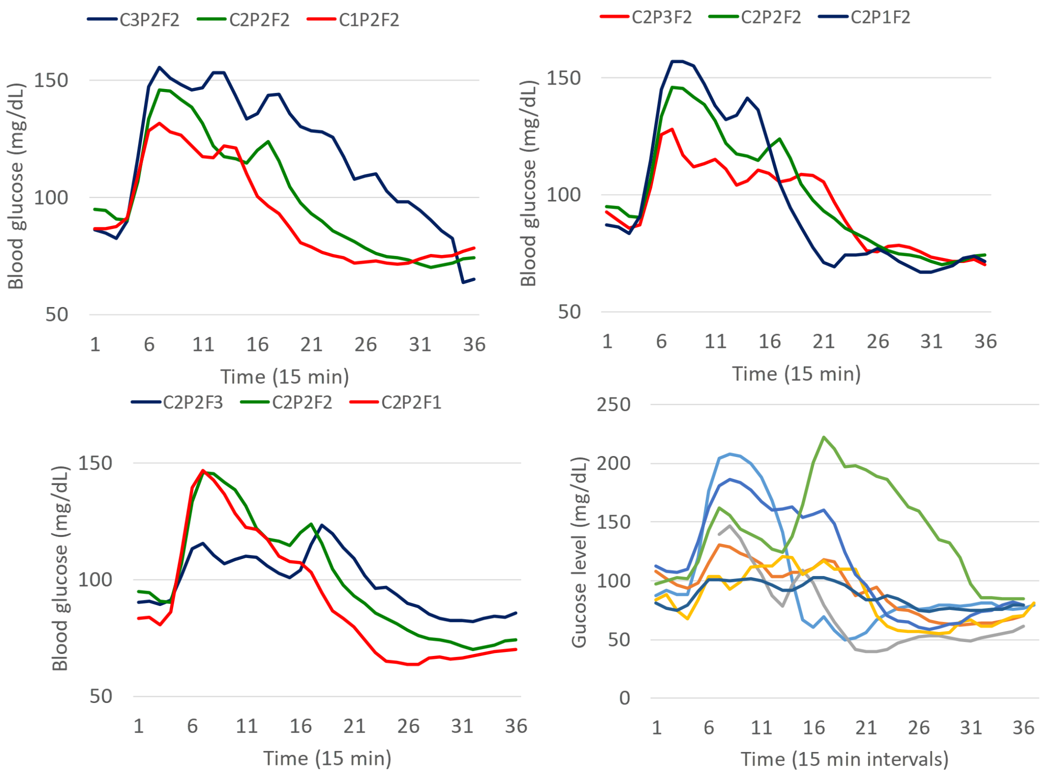


Figure 4.5: (a) Glucose response at increasing levels of carbohydrates, with protein and fat at fixed levels. (b) Glucose response at increasing levels of protein, with carbohydrates and fat at fixed levels. (c) Glucose response at increasing levels of fat, with carbohydrates and protein at fixed levels. (d) Individual variability for the C2P2F2 meal. X axis are samples (taken every 15 minutes) and Y axis are the blood glucose values. Reprinted/adapted with permission from [7].

study day, so that the first blood glucose reading would be their fasting glucose. The CGM was placed on the first study day and replaced every 2 weeks. After taking a baseline blood sample the morning of a study visit, a predefined meal was consumed. Subjects were then asked to remain in a sedentary state and were not allowed to eat anything for the following 8 hours to remove impact of physical activity on the PPGR, and also not eating anything else The blood samples served to validate the CGM reading accuracy, as well as collect additional information such as insulin to be used in future analysis out of the scope of this paper. The composition of the nine meals is shown in Table 4.3. Subjects were asked to remain sedentary in the laboratory environment, removing the confounding impacts of physical activity. It is important to note that a few subjects were missing data on some meals, as the CGM fell out of their arm. Five of the subjects are missing a total of seven meals because of errors in CGM readings, and no more than two meals were missed by

Table 4.4: Pooled Pearson Correlation and Statistical Significance of Macronutrient Results for carbohydrates (C), proteins (P), and fats (F). Reprinted/adapted with permission from [7].

Regression	**subject-independent**			**subject-dependent**		
	C	P	F	C	P	F
Linear	0.32^{**}	0.12	0.09	0.31^{**}	-0.29	-0.01
Multitask	0.31^{**}	0.14	0.21	$\mathbf{0.69}^{***}$	$\mathbf{0.23}^{*}$	$\mathbf{0.48}^{**}$

Significance: ***: $p < 0.0001$, **: $0.0001 \leq p < 0.05$, *: $0.05 \leq p < 0.1$

any one subject. This study was approved by the Texas A&M Institutional Review Board (IRB #2017-0886).

4.2.4.2 Variability in Glucose Response

Fig. 4.5 illustrates the meal-to-meal variability in modifying meal macronutrients and the subject-to-subject variability that exists within individual meals. Fig. 4.5 (a) shows the average response across subjects as we increase the amount of carbohydrates (C1, C2, C3) while maintaining the other two macronutrients at a fixed level (P2, F2). As shown, the glucose response becomes more pronounced at higher levels of carbohydrates, both in terms of the maximum value and the overall AUC. Fig. 4.5 (b) shows the average response across subjects as we increase the amount of protein (P1, P2, P3) while maintaining the other two macronutrients at a fixed level (C2, F2). As we increase the amount of protein, the glucose response becomes more moderate, with lower maximum levels and slower return to the baseline. Fig. 4.5 (c) shows the average response across subjects as we increase the amount of fat (F1, F2, F3) while maintaining the other two macronutrients at a fixed level (C2, P2). As in the case for protein, as we increase the amount of fat, the glucose response becomes more moderate, with lower maximum levels and slower return to the baseline. These results provide support to our overall strategy, as they show that the shape of the glucose response depends on the constituents of the meal. Finally, Fig. 4.5 (d) shows the response of each participant to the C2P2F2, which illustrates the high level of variability and the need to develop personalized models.

Table 4.5: Pooled Pearson correlation (with statistical significance) of four neural network models: multitask vs. single-task models for 5 and 17 Gaussian AUCs for carbohydrates (C), proteins (P), and fats (F). Reprinted/adapted with permission from [7].

	subject-independent			**subject-dependent**		
	C	P	F	C	P	F
5M	0.35^{***}	-0.09	-0.03	0.76^{***}	$\mathbf{0.32}^{**}$	$\mathbf{0.57}^{***}$
17M	0.44^{***}	-0.10	-0.08	$\mathbf{0.80}^{***}$	0.28^{**}	0.39^{***}
5S	0.11	0.09	0.01	0.38^{**}	-0.21	-0.15
17S	0.24*	-0.09	0.17	0.70^{***}	0.05	-0.08

Significance: ***: $p < 0.0001$, **: $0.0001 \leq p < 0.05$, *: $0.05 \leq p < 0.1$

4.2.4.3 Area Under the PPGR

In a first analysis, we analyzed the performance of the model as a regression task using 5 Gaussian kernels. Results are summarized in Table 2 in terms of the Pearson correlation between predictions and ground truth. We considered two types of models: subject-independent and subject-dependent. In the subject-independent case we computed 9 models, each model trained on 6 participants and tested on the remaining participant (leave one subject out cross-validation). The results in Table 2 represent the average response across those 7 models. In the subject-dependent case, we computed 9 separate models for each subject, each model trained on 8 meals and tested on the remaining meal (leave one meal out cross-validation). The results in Table 4.4 represent the average response across those 56 models (7 subjects × 9 models/subject minus the 7 meals with missing data.) For comparison purposes, we also included results for a linear regression model, which we used as a baseline. The subject-dependent model significantly outperformed the subject-independent model, as one might expect given the high degree of individual variability shown in Figure 3(d). In the subject-independent case, both models (linear and multitask neural network) achieved similar performance, and could (to some extent) predict the amount of carbohydrate but not the amount of protein and fat. Results for the subject-dependent models are markedly different. In this case, the multitask neural network significantly outperformed the linear model, and was able to predict the amount of carbohydrate and fat (but not protein).

4.2.4.4 Varying Input Features

In a second analysis, we examined whether extracting information from the glucose response at multiple levels of resolution would increase model performance. For this purpose we used three families of Gaussian kernels, consisting of 3 kernels, 5 kernels (as shown in Fig. 4.4) and 9 kernels, or a total of 17 features. Additionally, we subtracted the baseline glucose value from these AUC calculations. Given the poor results of the linear regression model, in this case we compared the multitask neural network against a single task neural network (a separate network for each of the three macronutrients). Results are shown in Table 4.5 for four different models:

- 5M: the previous multitask model with 5 Gaussian AUCs as inputs
- 17M: a multitask model with 17 Gaussian AUCs as inputs
- 5S: a single task model with 5 Gaussian AUCs as inputs
- 17S: a single task model with 17 Gaussian AUCs as inputs

By analyzing the two sets of results, we find that subtracting the baseline glucose levels (Table 4.5) improves model performance compared to using the absolute glucose level (Table 4.4). More specifically, we find that in the subject-independent case, AUC values relative to the baseline glucose level improve the prediction of carbohydrates, while in the subject-dependent case it improve carbohydrates, protein, and fat estimations and the level of statistical significance of the findings for the latter two. Further, adding features at multiple levels of resolution to the 5M model improved the prediction of carbohydrates for the four models, but not the prediction of fat and protein. Most importantly, the multitask model significantly outperformed the single task model for carbohydrate prediction (both in the subject-dependent and subject-independent cases) and the protein and fat prediction for the subject-independent model. This result clearly shows the advantage of the multitask architecture.

Table 4.6: Performance of the subject-dependent model after accounting for each subject's physiological parameters. Two approaches were used: adding the physiological parameters as inputs to the network, and normalizing the network's target value by the physiological parameter. Reprinted/adapted with permission from [7].

Regression	subject-independent		
	C	P	F
17 RMT	0.44^{***}	-0.10	-0.08
17 RMT + additional inputs	0.39^{***}	-0.04	0.20
17 RMT + normalized outputs	$\mathbf{0.62}^{***}$	$\mathbf{0.23}^{*}$	$\mathbf{0.30}^{**}$

Significance: ***: $p < 0.0001$, **: $0.0001 \leq p < 0.05$, *: $0.05 \leq p < 0.1$

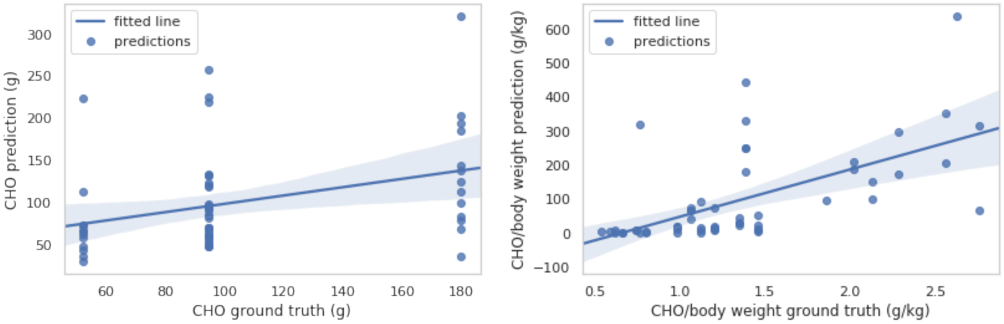


Figure 4.6: Best fit regression and 95% confidence interval of all test results pooled in a single regression plot when predicting (a) absolute amount of carbohydrates and (b) values relative to each subject's body weight for carbohydrates (CHO). Reprinted/adapted with permission from [7].

Table 4.7: Confusion matrix for ternary classification. Reprinted/adapted with permission from [7].

			Ground Truth							
			Carbohydrates			Protein			Fat	
		Low	*Medium*	*High*	*Low*	*Medium*	*High*	*Low*	*Medium*	*High*
Predicted	Direct Classification	*Low*	**6**	1	0	**3**	3	2	**6**	3	1
		Medium	5	**30**	5	7	**24**	7	7	**24**	4
		High	1	0	**8**	2	4	**4**	0	4	**7**
	subject-dependent Thresholds	*Low*	**11**	2	0	**7**	2	1	**10**	8	2
		Medium	1	**30**	0	5	**26**	5	3	**16**	2
		High	0	0	**12**	0	3	**7**	0	7	**8**

4.2.4.5 Improving Subject-Independent Models with Body Composition Features

Next, we examined whether the subject-independent models could be improved by including physiological parameters of each subject, e.g., body weight, lean body mass, fat mass, and body mass index. For this purpose, we considered two strategies: (1) using these physiological parameters as additional inputs to the multitask neural network, and (2) dividing the network's target values by the physiological parameters (i.e., normalization of outputs). Results are shown in Table 4.6 when using body weight as the physiological parameter; similar results (not included here) were obtained when using lean body mass, fat mass, and body mass index. Adding the body weight as an input to the model did not improve performance, arguably because of the small size of the training set and difficulty in optimally training the input weights in the network. However, normalizing the network's target values relative to the body weight significantly improved the model predictions for the three macronutrients. Thus, it was not the absolute amount of the macronutrients in the meal but the amount relative to each patient's physiological parameters (in this case body weight) that should be used as a target. Figure 4 illustrates the adjustment of the labels that lead to the improvement in the subject-independent model, as a result of normalizing the output, which has a correlation coefficient of 0.62 for carbohydrates (Figure 4b) versus 0.44 (Figure 4a).

4.2.4.6 Ternary Classification

In a final analysis, we evaluated the performance of the multitask neural network when used for ternary classification in the subject-dependent case, in which macronutrient compositions were collapsed into three levels: low, medium and high. As a reminder, for each of the 7 subjects, we trained 9 models in the subject-dependent case: one model trained on 8 meals and tested on the remaining 9th meal. We then determined the regression value threshold below which we would classify "low" and above which we would classify "not low" (medium or high), by sweeping this threshold from the minimum to maximum values provided in the training set, as in the creation of a receiver operating characteristic curve. We selected the lowest such threshold that optimized the f1 score for the low class for each macronutrient. We repeated this process for the highest such threshold that optimized the f1 score for the high class. Any value in between these thresholds was classified as the medium class. In order to evaluate performance of our multitask neural network with these subject-dependent thresholds, we compare our findings with a multitask neural network trained for classification (rather than correlation).

Results for the classification task are summarized in Table 4.7 (pooled across all subjects). We compare our approach, which applies a subject-dependent threshold to discretize model outputs into three categories (i.e., low, medium, high), against a direct classification approach where the neural network is trained as a classifier. For low carbohydrates (predicting low vs. medium/high), we have the following results: (1) the precision, recall, and f1 score for the low class for our subject-dependent threshold model are 0.85, 0.92, and 0.88; (2) the precision, recall, and f1 score for the direct classification model are 0.86, 0.50, and 0.63 respectively. For medium carbohydrates (vs. low/high), we have the following results: (1) the precision, recall, and f1 score for the low class for our subject-dependent threshold model are 0.97, 0.94, and 0.95; (2) the precision, recall, and f1 score for the direct classification model are 0.75, 0.97, and 0.85 respectively. For high carbohydrates (vs. low/medium), we have the following results: (1) the precision, recall, and f1 score for the low class for our subject-dependent threshold model are 1.00, 1.00, and 1.00; (2) the precision, recall, and f1 score for the direct classification model are 0.89, 0.62, and 0.73 respectively. We find that the direct classification results, while accurate at f1 scores of 0.63 (low), 0.85 (medium), and 0.73 (high), the model does not account for extra penalties for lows classified as highs and highs classified as lows. This is why our subject-dependent threshold model (picking subject-dependent thresholds from the multitask neural network regression model) has more accurate f1 scores at 0.88 (low), 0.95 (medium), and 1.00 (high) respectively, because it is trained initially as a regression model with optimal correlation coefficient.

We find the same classification improvements in proteins and fats. For proteins, the f1 score for our subject dependent threshold model vs. direct classification is: 0.64 vs. 0.30 for low (vs. medium/high), 0.78 vs. 0.70 for medium (vs. low/high), and 0.61 vs. 0.35 for high (vs. low/medium). With the poorer performance in estimating proteins we find that the bulk of correct predictions come from estimating the majority class: medium. For fats, we find the two techniques perform similarly: 0.61 vs. 0.52 for low (vs. medium/high), 0.62 vs. 0.72 for medium (vs. low/high), and 0.59 vs. 0.61 for high (vs. low/medium). These findings indicate that the direct classification tends to predict medium, while the subject-dependent threshold model varies low and high predictions more, but the prediction performances are relatively similar. In both findings, the classification results demonstrate accuracy in predicting low and high, even though the class imbalance tends towards medium. However, Table 4.7 shows that the subject-dependent threshold findings reduce the number of extreme misclassifications (lows as highs and highs as lows).

4.2.5 Discussion

This study proposes a technique to predict the amount of macronutrients in a meal from the shape of the glucose response of the meal, using continuous glucose monitors and a novel multitask neural network. Our results indicate that the amount of carbohydrates (and to a lesser extent fats but not protein) can be predicted, as measured through regression and classification tasks. We validated our approach in a series of experiments. We find that subject-dependent models, where a separate model is trained for each subject, significantly outperform a subject-independent model, where a single model is trained by pooling data from multiples subjects. This is consisted with other studies which have shown that metabolism and therefore the glucose response to a meal is unique to each individual [242]. We find that the multitask network, which jointly predicts the amount of the three macronutrients, outperforms a single-task method that uses three separate networks (one for each macronutrient). We explored several techniques to improve model performance. First, we find that subtracting fasting glucose levels improves performance when compared to using the absolute glucose response. Second, we find that normalizing the amount of macronutrients relative to body composition (e.g., body weight) also improves model performance. These results show that fasting glucose and body composition are part of the phenotype of each subject. Finally, we also find that features extracted a multiple levels of resolution do not improve performance compared to using features at a single time scale. We also examined evaluated the performance of the model when used as a classification task, where the amount of each macronutrients was discretized into three labels: low, medium and high. We find that obtaining class labels by training the neural network as a regression model, and then discretizing the model outputs, provides higher classification performance than training the network directly as a ternary classifier.

4.2.6 Limitations and Future Work

Our study was conducted in a controlled setting, where participants remained stationary following consumption of a meal. As such, the model does not account for physical activity following a meal, which is known to influence (i.e., reduce) the glucose response. Our approach extracts information over the eight hours that follow a meal, too long of a period in practice. It is possible that similar prediction results could be achieved by extracting information over a shorter time period (e.g., 2-4 hour) but this remains to be tested. Future work also needs to examine whether results obtained on liquid meals generalize to realistic meals, which are generally solid and are consumed over a longer period (compared to ingesting a liquid meal at one time). Finally, participants consumed each meal only once, so our study is unable to assess the levels of intra-individual variability, which are also known to exist [242]. All of these limitations will be assessed in a forthcoming study where participants will consume a number of solid and liquid meals over an extended period while carrying out with their daily lives and monitoring their physical activity with fitness trackers.

4.2.7 Discussion

This work showed that the carbohydrate and fat content of a meal can be estimated by analyzing post-prandial glucose responses. Our study is related to but different from that of Zeevi et al. [242]. In that seminal work, the authors showed that the glucose response of a meal can be predicted from the meal's contents. Instead, our study addresses the reverse problem: predicting meals' contents from their glucose response. As such, we believe our study is the first of its kind. While not definite, our results are encouraging and open the possibility of automatically tracking dietary intake, an essential component in the management of diabetes.

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

The increasing attention and utility of machine learning for enhancing human well-being and health has been the backbone research of this dissertation. Again, The fast-growing artificial intelligence industry has spurred some interesting health-related applications, and inversely the real-world need of mobile and clinical health problems have pushed forward many theoretical findings for algorithmic development. In this dissertation I have discussed the opportunities along with the challenges in the field. We have observed many discrepancies of compatibility between sophisticated machine learning models and the nuanced biomedical issues. I have discussed how we can bridge the gap between the pure theory and the problem in the wild, any tangible development would not render any satisfactory results.

For my research looking back for the dissertation, the main goal is to organically merge the two fronts together and therefore the real-world health application can benefit from the powerful machine learning models and at the same time the algorithm can be guided for social good in general. My two main pillars of research interests lie in mobile health and clinical health, which are seemingly different but very connected. 1) First the artificial intelligence applications have been seen in some real-world clinical settings, such as risk prediction, phenotyping, or personalized treatment. But either due to lack of in-depth machine learning expertise or clinical intuition, many models are still in a primitive stage. For example, many clinical prediction models assume the patient data is homogeneous or noise-free. However, in the real-world medical domain, this is still a big challenge even with the curated clinical dataset. My work has focused on many aspects of these discrepancies, such as data imbalance, missing values, hidden phenotypes. Many of those are inspired by the unique challenges of clinical data but the developments of such models also shed light on more theoretical model developments for the machine learning world at large. For one tangible example, the patient outcome of interest always if not all accounts for a smaller portion of the data, such as mortality in an ICU stay, or detection of COVID in a free-living environment. This data distribution discrepancy among the classes can bias the model towards outputting the prediction mostly aligned with the majority of the cohort, which in this case is the cohort without carrying the high risk factor. However, any useful model should output not only prediction aligned with some accuracy metric, but also differentiate the underlying latent subgroups that are hidden in the data. I have conducted various experiments on that aspect for this purpose.

2) Second, beyond what we have inside the hospital, the most abundant information sources for well-being monitoring are in free-living environments, which could not be captured by in-hospital devices. With the growing number of smartphones and wearable devices, we have seen the enormous opportunities of extending health modeling to many other aspects of people's lives. However, these opportunities are not going to present themselves unless we can solve many imminent challenges at hand. First in a free-living environment, users' behaviors are out of control compared to laboratory settings. We would envision many heterogeneous user behaviors and even within one user, the behavior would likely change over time. How can we take that into account to have a continually growing model for any given unknown context is of great interest. The uncertainty quantification aspect of the model becomes very relevant, which has been intensively studied through my research for mobile health application.

5.2 Future Work

The exciting trend of AI for health is the goal of my research, specifically targeting at data heterogeneity at different granularity levels. To advance in that direction, I have put together the pieces of some of my research outlooks in the following paths. First, I plan to continue the translational research for algorithms that are developed towards the tangible benefit of their utility towards human well-being monitoring, on both the population-level and personal-level heterogeneity. This requires combining some of the machine learning frontlines together such as continual learning, multimodal learning, domain adaptation, to bring about more intelligent systems that require minimal human intervention to operate under many noisy and uncertain real-world conditions. For example, how does data source from one domain become useful in another one or the data source from previous data collection trial be useful for the current one, through learning some time-invariant high-level representation. Second, I plan to dive deeper into the real-world data analytics. For example, in the mobile health domain, users might not be willing to share their raw data but how can we make use of tens of thousands of smart phone data without infringing on personal privacy. Federated learning or swarm intelligence can be of great help when we train the model locally and update the parameters remotely, for example through sharing the gradients instead of the raw data. Another example can be the clinical health domain, where the low interpretability has been the bottleneck for many machine learning models and therefore the adoption rate in the hospitals for such algorithms are currently low. To make the model more transparent without the sacrifice of accuracy is one of the key components toward AI for health. With many prominent opportunities and challenges at hand, I plan to continue to build a more robust and generalizable machine learning pipeline to be useful in a more dynamic and uncertain environment for the common good of human well-being.

REFERENCES

[1] Z. Huo, A. PakBin, X. Chen, N. Hurley, Y. Yuan, X. Qian, Z. Wang, S. Huang, and B. Mortazavi, "Uncertainty quantification for deep context-aware mobile activity recognition and unknown context discovery," in *International Conference on Artificial Intelligence and Statistics*, pp. 3894–3904, PMLR, 2020.

[2] R. Ardywibowo, Z. Huo, Z. Wang, B. J. Mortazavi, S. Huang, and X. Qian, "Varigrow: Variational architecture growing for task-agnostic continual learning based on bayesian novelty," in *International Conference on Machine Learning*, pp. 865–877, PMLR, 2022.

[3] Z. Huo, L. Zhang, R. Khera, S. Huang, X. Qian, Z. Wang, and B. J. Mortazavi, "Sparse gated mixture-of-experts to separate and interpret patient heterogeneity in ehr data," in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–4, IEEE, 2021.

[4] Z. Huo, X. Qian, S. Huang, Z. Wang, and B. J. Mortazavi, "Density-aware personalized training for risk prediction in imbalanced medical data," *Machine Learning for Healthcare Conference (MLHC). PMLR*, 2022.

[5] Z. Huo, H. Sundararajhan, N. C. Hurley, A. Haimovich, R. A. Taylor, and B. J. Mortazavi, "Sparse embedding for interpretable hospital admission prediction," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3438–3441, IEEE, 2019.

[6] Z. Huo, T. Ji, Y. Liang, S. Huang, Z. Wang, X. Qian, and B. Mortazavi, "Dynimp: Dynamic imputation for wearable sensing data through sensory and temporal relatedness," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3988–3992, IEEE, 2022.

[7] Z. Huo, B. J. Mortazavi, T. Chaspari, N. Deutz, L. Ruebush, and R. Gutierrez-Osuna, "Predicting the meal macronutrient composition from continuous glucose monitors," in *Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics*,

IEEE, 2019.

[8] S. Liang, Y. Li, and R. Srikant, “Principled detection of out-of-distribution examples in neural networks,” *arXiv preprint arXiv:1706.02690*, 2017.

[9] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, “Context-aware computing, learning, and big data in internet of things: a survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 1–27, 2018.

[10] A. K. Dey, “Understanding and using context,” *Personal and ubiquitous computing*, vol. 5, no. 1, pp. 4–7, 2001.

[11] D. Riboni and C. Bettini, “Cosar: hybrid reasoning for context-aware activity recognition,” *Personal and Ubiquitous Computing*, vol. 15, no. 3, pp. 271–289, 2011.

[12] J. Xu, L. Song, J. Y. Xu, G. J. Pottie, and M. Van Der Schaar, “Personalized active learning for activity classification using wireless wearable sensors,” *IEEE journal of selected topics in signal processing*, vol. 10, no. 5, pp. 865–876, 2016.

[13] J. Andreu-Perez, D. R. Leff, H. M. Ip, and G.-Z. Yang, “From wearable sensors to smart implants—toward pervasive and personalized healthcare,” *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 12, pp. 2750–2762, 2015.

[14] B. M. Spiegel, M. Kaneshiro, M. M. Russell, A. Lin, A. Patel, V. C. Tashjian, V. Zegarski, D. Singh, S. E. Cohen, M. W. Reid, *et al.*, “Validation of an acoustic gastrointestinal surveillance biosensor for postoperative ileus,” *Journal of Gastrointestinal Surgery*, vol. 18, no. 10, pp. 1795–1803, 2014.

[15] J. Y. Xu, H.-I. Chang, C. Chien, W. J. Kaiser, and G. J. Pottie, “Context-driven, prescription-based personal activity classification: methodology, architecture, and end-to-end implementation,” *IEEE journal of biomedical and health informatics*, vol. 18, no. 3, pp. 1015–1025, 2014.

[16] O. Steven Eyobu and D. Han, “Feature representation and data augmentation for human activity classification based on wearable imu sensor data using a deep lstm neural network,” *Sensors*, vol. 18, no. 9, p. 2892, 2018.

[17] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in *Aaai*, vol. 5, pp. 1541–1546, 2005.

[18] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[19] M. I. Jordan and R. A. Jacobs, “Hierarchical mixtures of experts and the em algorithm,” *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.

[20] S. E. Yuksel, J. N. Wilson, and P. D. Gader, “Twenty years of mixture of experts,” *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[21] A. Miech, I. Laptev, and J. Sivic, “Learnable pooling with context gating for video classification,” *arXiv preprint arXiv:1706.06905*, 2017.

[22] M. Courbariaux, C. Ambroise, C. Dalmasso, M. Szafranski, M. Consortium, *et al.*, “A mixture model with logistic weights for disease subtyping with integrated genome association study,” 2018.

[23] M. I. Jordan and L. Xu, “Convergence results for the em approach to mixtures of experts architectures,” *Neural networks*, vol. 8, no. 9, pp. 1409–1431, 1995.

[24] C. Yuan and C. Neubauer, “Variational mixture of gaussian process experts,” in *Advances in Neural Information Processing Systems*, pp. 1897–1904, 2009.

[25] A. Sharma, S. Saxena, and P. Rai, “A flexible probabilistic framework for large-margin mixture of experts,” *Machine Learning*, pp. 1–25, 2019.

[26] C. A. Lima, A. L. Coelho, and F. J. Von Zuben, “Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification,” *Information Sciences*, vol. 177, no. 10, pp. 2049–2074, 2007.

[27] L. Cao, “Support vector machines experts for time series forecasting,” *Neurocomputing*, vol. 51, pp. 321–339, 2003.

[28] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” *arXiv preprint arXiv:1701.06538*, 2017.

[29] R. Aljundi, P. Chakravarty, and T. Tuytelaars, “Expert gate: Lifelong learning with a network of experts,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3366–3375, 2017.

[30] R. A. Jacobs, F. Peng, and M. A. Tanner, “A bayesian approach to model selection in hierarchical mixtures-of-experts architectures,” *Neural Networks*, vol. 10, no. 2, pp. 231–241, 1997.

[31] C. M. Bishop and M. Svenskn, “Bayesian hierarchical mixtures of experts,” in *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pp. 57–64, Morgan Kaufmann Publishers Inc., 2002.

[32] R. Ardywibowo, G. Zhao, Z. Wang, B. Mortazavi, S. Huang, and X. Qian, “Adaptive activity monitoring with uncertainty quantification in switching Gaussian Process models,” *The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[33] A. Samareh and S. Huang, “UQ-CHI: An uncertainty quantification-based contemporaneous health index for degenerative disease monitoring,” *arXiv preprint arXiv:1902.08246*, 2019.

[34] N. Meghdadi, H. Niroomand-Oscuii, M. Soltani, F. Ghalichi, and M. Pourgolmohammad, “Brain tumor growth simulation: model validation through uncertainty quantification,” *International Journal of System Assurance Engineering and Management*, vol. 8, no. 3, pp. 655–662, 2017.

[35] E. Reynders, K. Maes, G. Lombaert, and G. De Roeck, “Uncertainty quantification in operational modal analysis with stochastic subspace identification: validation and applications,” *Mechanical Systems and Signal Processing*, vol. 66, pp. 13–30, 2016.

[36] S. Nannapaneni and S. Mahadevan, “Uncertainty quantification in performance evaluation of manufacturing processes,” in *2014 IEEE International Conference on Big Data (Big Data)*, pp. 996–1005, IEEE, 2014.

[37] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*, pp. 1050–1059, 2016.

[38] A. Malinin and M. Gales, “Predictive uncertainty estimation via prior networks,” in *Advances in Neural Information Processing Systems*, pp. 7047–7058, 2018.

[39] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” *arXiv preprint arXiv:1610.02136*, 2016.

[40] K. Lee, K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in *Advances in Neural Information Processing Systems*, pp. 7167–7177, 2018.

[41] R. Solis, A. Pakbin, A. Akbari, B. J. Mortazavi, and R. Jafari, “A human-centered wearable sensing platform with intelligent automated data annotation capabilities,” in *Proceedings of the International Conference on Internet of Things Design and Implementation*, pp. 255–260, 2019.

[42] S. Sun, Y. Liu, and L. Mao, “Multi-view learning for visual violence recognition with maximum entropy discrimination and deep features,” *Information Fusion*, 2018.

[43] C. Zhu and Z. Wang, “Semi-supervised soft margin consistency based multi-view maximum entropy discrimination,” *Applied Computing and Informatics*, 2018.

[44] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, *et al.*, “Collecting complex activity datasets in highly rich networked sensor environments,” in *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*, pp. 233–240, IEEE, 2010.

[45] Lockheed Martin, “Anonymized phone usage data collector,” 2019.

[46] J. Guérin, O. Gibaru, S. Thiery, and E. Nyiri, “Cnn features are also great at unsupervised classification,” *arXiv preprint arXiv:1707.01700*, 2017.

[47] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

[48] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International*

conference on machine learning, pp. 647–655, 2014.

[49] S. Thrun and T. M. Mitchell, “Lifelong robot learning,” in *The biology and technology of intelligent autonomous agents*, pp. 165–196, Springer, 1995.

[50] L. Li, Z. Jun, J. Fei, and S. Li, “An incremental face recognition system based on deep learning,” in *2017 Fifteenth IAPR international conference on machine vision applications (MVA)*, pp. 238–241, IEEE, 2017.

[51] J. M. Pierre, “Incremental lifelong deep learning for autonomous vehicles,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3949–3954, IEEE, 2018.

[52] R. Ardywibowo, S. Huang, S. Gui, C. Xiao, Y. Cheng, J. Liu, and X. Qian, “Switching-state dynamical modeling of daily behavioral data,” *Journal of Healthcare Informatics Research*, vol. 2, no. 3, pp. 228–247, 2018.

[53] R. Ardywibowo, *Analyzing Daily Behavioral Data for Personalized Health Management*. PhD thesis, 2017.

[54] Z. Jiang, R. Ardywibowo, A. Samereh, H. L. Evans, W. B. Lober, X. Chang, X. Qian, Z. Wang, and S. Huang, “A roadmap for automatic surgical site infection detection and evaluation using user-generated incision images,” *Surgical infections*, vol. 20, no. 7, pp. 555–565, 2019.

[55] R. Aljundi, P. Chakravarty, and T. Tuytelaars, “Expert gate: Lifelong learning with a network of experts,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3366–3375, 2017.

[56] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, “Memory aware synapses: Learning what (not) to forget,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

[57] D. Lopez-Paz and M.-A. Ranzato, “Gradient episodic memory for continual learning,” in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 6467–6476, Curran

Associates, Inc., 2017.

[58] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proc. of the national academy of sciences*, 2017.

[59] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” *ArXiv e-prints*, jun 2016.

[60] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” in *Advances in Neural Information Processing Systems*, pp. 2990–2999, 2017.

[61] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, “Lifelong learning with dynamically expandable networks,” in *International Conference on Learning Representations*, 2018.

[62] S. Yan, J. Xie, and X. He, “DER: Dynamically Expandable Representation for class incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3014–3023, 2021.

[63] Y. Liu, B. Schiele, and Q. Sun, “Adaptive aggregation networks for class-incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2544–2553, 2021.

[64] J. Rajasegaran, M. Hayat, S. H. Khan, F. S. Khan, and L. Shao, “Random path selection for incremental learning,” *CoRR*, vol. abs/1906.01120, 2019.

[65] S. Lee, J. Ha, D. Zhang, and G. Kim, “A neural Dirichlet process mixture model for task-free continual learning,” in *International Conference on Learning Representations*, 2020.

[66] D. Rao, F. Visin, A. A. Rusu, Y. W. Teh, R. Pascanu, and R. Hadsell, “Continual unsupervised representation learning,” 2019.

[67] R. Aljundi, “Continual learning in neural networks,” *arXiv preprint arXiv:1910.02718*, 2019.

[68] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, “Gradient based sample selection for online continual learning,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 11816–

11825, Curran Associates, Inc., 2019.

[69] C. Zeno, I. Golan, E. Hoffer, and D. Soudry, “Task agnostic continual learning using online variational Bayes,” 2018.

[70] X. He and H. Jaeger, “Overcoming catastrophic interference using conceptor-aided backpropagation,” in *International Conference on Learning Representations*, 2018.

[71] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural Networks*, vol. 113, pp. 54–71, 2019.

[72] S. Sodhani, S. Chandar, and Y. Bengio, “Toward training recurrent neural networks for lifelong learning,” *Neural computation*, vol. 32, no. 1, pp. 1–35, 2020.

[73] P. Kaushik, A. Gain, A. Kortylewski, and A. Yuille, “Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping,” *arXiv preprint arXiv:2102.11343*, 2021.

[74] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[75] D. M. Blei and M. I. Jordan, “Variational inference for dirichlet process mixtures,” *Bayesian analysis*, vol. 1, no. 1, pp. 121–143, 2006.

[76] D. Lin, “Online learning of nonparametric mixture models via sequential variational approximation,” *Advances in Neural Information Processing Systems*, vol. 26, pp. 395–403, 2013.

[77] S. Kessler, V. Nguyen, S. Zohren, and S. Roberts, “Hierarchical indian buffet neural networks for bayesian continual learning,” *arXiv preprint arXiv:1912.02290*, 2019.

[78] M. Yin and M. Zhou, “Semi-implicit variational inference,” in *International Conference on Machine Learning*, pp. 5660–5669, PMLR, 2018.

[79] M. K. Titsias and F. Ruiz, “Unbiased implicit variational inference,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 167–176, PMLR, 2019.

[80] D. Molchanov, V. Kharitonov, A. Sobolev, and D. Vetrov, “Doubly semi-implicit variational inference,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2593–2602, PMLR, 2019.

[81] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, “A tutorial on energy-based learning,” *Predicting structured data*, vol. 1, no. 0, 2006.

[82] W. Liu, X. Wang, J. Owens, and Y. Li, “Energy-based out-of-distribution detection,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[83] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “ICARL: Incremental classifier and representation learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

[84] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, “Learning to learn without forgetting by maximizing transfer and minimizing interference,” in *International Conference on Learning Representations*, 2018.

[85] M. K. Titsias, J. Schwarz, A. G. d. G. Matthews, R. Pascanu, and Y. W. Teh, “Functional regularisation for continual learning with Gaussian processes,” in *International Conference on Learning Representations*, 2019.

[86] J. Pomponi, S. Scardapane, V. Lomonaco, and A. Uncini, “Efficient continual learning in neural networks with embedding regularization,” *Neurocomputing*, vol. 397, pp. 139–148, 2020.

[87] M. Wortsman, V. Ramanujan, R. Liu, A. Kembhavi, M. Rastegari, J. Yosinski, and A. Farhadi, “Supermasks in superposition,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[88] J. Schwarz, J. Luketina, W. M. Czarnecki, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, “Progress & compress: A scalable framework for continual learning,” in *ICML*, 2018.

[89] P. Kirichenko, M. Farajtabar, D. Rao, B. Lakshminarayanan, N. Levine, A. Li, H. Hu, A. G. Wilson, and R. Pascanu, “Task-agnostic continual learning with hybrid probabilistic models,” in *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021.

[90] D. Abati, J. Tomczak, T. Blankevoort, S. Calderara, R. Cucchiara, and B. E. Bejnordi, “Con-

ditional channel gated networks for task-aware continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3931–3940, 2020.

[91] J. Rajasegaran, S. Khan, M. Hayat, F. S. Khan, and M. Shah, "iTAML: An incremental task-agnostic meta-learning approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13588–13597, 2020.

[92] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach, "Uncertainty-guided continual learning with bayesian neural networks," in *International Conference on Learning Representations*, 2020.

[93] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *International Conference on Learning Representations*, 2019.

[94] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for gans do actually converge?," in *International conference on machine learning*, pp. 3481–3490, PMLR, 2018.

[95] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?," in *International Conference on Learning Representations*, 2018.

[96] M. Ranzato, Y.-L. Boureau, S. Chopra, and Y. LeCun, "A unified energy-based framework for unsupervised learning," in *Artificial Intelligence and Statistics*, pp. 371–379, PMLR, 2007.

[97] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang, "Self-supervised learning for generalizable out-of-distribution detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 5216–5223, 2020.

[98] R. Kurle, B. Cseke, A. Klushyn, P. van der Smagt, and S. Günnemann, "Continual learning with bayesian neural networks for non-stationary data," in *International Conference on Learning Representations*, 2019.

[99] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *International Conference on Learning Representations*, 2018.

[100] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[101] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[102] N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker, *Bayesian nonparametrics*, vol. 28. Cambridge University Press, 2010.

[103] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[104] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[105] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13208–13217, 2020.

[106] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "PODNet: Pooled outputs distillation for small-tasks incremental learning," in *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2020.

[107] S. Yan, J. Xie, and X. He, "DER: Dynamically expandable representation for class incremental learning," 2021.

[108] X. Tao, X. Chang, X. Hong, X. Wei, and Y. Gong, "Topology-preserving class-incremental learning," in *European Conference on Computer Vision*, pp. 254–270, Springer, 2020.

[109] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[110] M. Welling, "Herding dynamical weights to learn," in *Proceedings of the 26th Annual In-*

ternational Conference on Machine Learning, pp. 1121–1128, 2009.

[111] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge,"

[112] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv e-prints*, pp. arXiv–1506, 2015.

[113] B. Groot Koerkamp, M. C. Weinstein, T. Stijnen, M. H. Heijenbrok-Kal, and M. M. Hunink, "Uncertainty and patient heterogeneity in medical decision models," *Medical Decision Making*, vol. 30, no. 2, pp. 194–205, 2010.

[114] Y. Wang, R. Chen, J. Ghosh, J. C. Denny, A. Kho, Y. Chen, B. A. Malin, and J. Sun, "Rubik: Knowledge guided tensor factorization and completion for health data analytics," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1265–1274, 2015.

[115] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep patient: an unsupervised representation to predict the future of patients from the electronic health records," *Scientific reports*, vol. 6, no. 1, pp. 1–10, 2016.

[116] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with lstm recurrent neural networks," *arXiv preprint arXiv:1511.03677*, 2015.

[117] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Distilling knowledge from deep networks with applications to healthcare domain," *arXiv preprint arXiv:1512.03542*, 2015.

[118] H.-C. Shin, L. Lu, L. Kim, A. Seff, J. Yao, and R. M. Summers, "Interleaved text/image deep mining on a large-scale radiology database for automated image interpretation," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3729–3759, 2016.

[119] X. Li, D. Zhu, and P. Levy, "Predicting clinical outcomes with patient stratification via deep mixture neural networks," *AMIA Summits on Translational Science Proceedings*, vol. 2020, p. 367, 2020.

[120] R. Avnimelech and N. Intrator, "Boosted mixture of experts: an ensemble learning scheme," *Neural computation*, vol. 11, no. 2, pp. 483–497, 1999.

[121] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[122] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[123] E. Bengio, P.-L. Bacon, J. Pineau, and D. Precup, "Conditional computation in neural networks for faster models," *arXiv preprint arXiv:1511.06297*, 2015.

[124] A. Hosseini, T. Chen, W. Wu, Y. Sun, and M. Sarrafzadeh, "Heteromed: Heterogeneous information network for medical diagnosis," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 763–772, 2018.

[125] J. Han, Y. Sun, X. Yan, and P. S. Yu, "Mining knowledge from databases: an information network analysis approach," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 1251–1252, 2010.

[126] R. Pivovarov, A. J. Perotte, E. Grave, J. Angiolillo, C. H. Wiggins, and N. Elhadad, "Learning probabilistic phenotypes from heterogeneous ehr data," *Journal of biomedical informatics*, vol. 58, pp. 156–165, 2015.

[127] M. Sushil, S. Šuster, K. Luyckx, and W. Daelemans, "Patient representation learning and interpretable evaluation using clinical notes," *Journal of biomedical informatics*, vol. 84, pp. 103–113, 2018.

[128] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: graph-based attention model for healthcare representation learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 787–795, 2017.

[129] J. Zhang, K. Kowsari, J. H. Harrison, J. M. Lobo, and L. E. Barnes, "Patient2vec: A personalized interpretable deep representation of the longitudinal electronic health record," *IEEE Access*, vol. 6, pp. 65333–65346, 2018.

[130] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *Thirty-Second AAAI Conference*

on Artificial Intelligence, 2018.

[131] Y.-S. Lee and S.-B. Cho, "Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data," *Neurocomputing*, vol. 126, pp. 106–115, 2014.

[132] E. D. Übeyli, "A mixture of experts network structure for breast cancer diagnosis," *Journal of medical systems*, vol. 29, no. 5, pp. 569–579, 2005.

[133] İ. Güler and E. D. Übeyli, "A modified mixture of experts network structure for ecg beats classification with diverse features," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 7, pp. 845–856, 2005.

[134] S.-K. Ng and G. J. McLachlan, "Using the em algorithm to train neural networks: misconceptions and a new algorithm for multiclass classification," *IEEE transactions on neural networks*, vol. 15, no. 3, pp. 738–749, 2004.

[135] L. Xu, M. I. Jordan, and G. E. Hinton, "An alternative model for mixtures of experts," in *Advances in neural information processing systems*, pp. 633–640, 1995.

[136] M. I. Jordan and R. A. Jacobs, "Hierarchies of adaptive experts," in *Advances in neural information processing systems*, pp. 985–992, 1992.

[137] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*, pp. 355–368, Springer, 1998.

[138] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[139] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[140] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. V. Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *arXiv preprint arXiv:1703.07771*, 2017.

[141] K. Boyd, K. H. Eng, and C. D. Page, "Area under the precision-recall curve: point estimates and confidence intervals," in *Joint European conference on machine learning and knowledge*

discovery in databases, pp. 451–466, Springer, 2013.

[142] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, and Y. Kluger, "Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data," *Nature methods*, vol. 16, no. 3, pp. 243–245, 2019.

[143] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[144] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, *et al.*, "Scalable and accurate deep learning with electronic health records," *NPJ Digital Medicine*, vol. 1, no. 1, pp. 1–10, 2018.

[145] J. A. Swets, "Roc analysis applied to the evaluation of medical imaging techniques.," *Investigative radiology*, vol. 14, no. 2, pp. 109–121, 1979.

[146] J. M. Lobo, A. Jiménez-Valverde, and R. Real, "Auc: a misleading measure of the performance of predictive distribution models," *Global ecology and Biogeography*, vol. 17, no. 2, pp. 145–151, 2008.

[147] N. R. Cook, "Use and misuse of the receiver operating characteristic curve in risk prediction," *Circulation*, vol. 115, no. 7, pp. 928–935, 2007.

[148] C. Huang, S.-X. Li, C. Caraballo, F. A. Masoudi, J. S. Rumsfeld, J. A. Spertus, S.-L. T. Normand, B. J. Mortazavi, and H. M. Krumholz, "Performance metrics for the comparative analysis of clinical risk prediction models employing machine learning," *Circulation: Cardiovascular Quality and Outcomes*, pp. CIRCOUTCOMES–120, 2021.

[149] D.-C. Li, C.-W. Liu, and S. C. Hu, "A learning method for the class imbalance problem with medical data sets," *Computers in biology and medicine*, vol. 40, no. 5, pp. 509–518, 2010.

[150] Y. Park and J. C. Ho, "Califorest: calibrated random forest for health data," in *Proceedings of the ACM Conference on Health, Inference, and Learning*, pp. 40–50, 2020.

[151] S. Wang, M. B. McDermott, G. Chauhan, M. Ghassemi, M. C. Hughes, and T. Naumann, "Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii," in *Proceedings of the ACM Conference on Health, Inference, and Learning*, pp. 222–

235, 2020.

[152] V. Babar and R. Ade, “Mlp-based undersampling technique for imbalanced learning,” in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pp. 142–147, IEEE, 2016.

[153] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, “Classification with class imbalance problem,” *Int. J. Advance Soft Compu. Appl*, vol. 5, no. 3, 2013.

[154] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, “Decoupling representation and classifier for long-tailed recognition,” *arXiv preprint arXiv:1910.09217*, 2019.

[155] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, “Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9719–9728, 2020.

[156] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen, *et al.*, “Dynamic sampling in convolutional neural networks for imbalanced data classification,” in *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pp. 112–117, IEEE, 2018.

[157] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[158] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[159] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *International Conference on Learning Representations*, 2018.

[160] C. Huang, Y. Li, C. C. Loy, and X. Tang, “Deep imbalanced learning for face recognition and attribute prediction,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 11, pp. 2781–2794, 2019.

[161] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective

number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

[162] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *arXiv preprint arXiv:1906.07413*, 2019.

[163] S. Bhattacharya, V. Rajan, and H. Shrivastava, "Icu mortality prediction: a classification algorithm for imbalanced datasets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[164] M. M. Rahman and D. N. Davis, "Addressing the class imbalance problem in medical datasets," *International Journal of Machine Learning and Computing*, vol. 3, no. 2, p. 224, 2013.

[165] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*, pp. 1321–1330, PMLR, 2017.

[166] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384, 2016.

[167] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19290–19301, 2020.

[168] J. Zhang, L. Liu, P. Wang, and C. Shen, "To balance or not to balance: A simple-yet-effective approach for learning with long-tailed distributions," *arXiv preprint arXiv:1912.04486*, 2019.

[169] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[170] C. Wei and T. Ma, "Improved sample complexities for deep networks and robust classification via an all-layer margin," *arXiv preprint arXiv:1910.04284*, 2019.

[171] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[172] J. Luo, H. Qiao, and B. Zhang, “Learning with smooth hinge losses,” *arXiv preprint arXiv:2103.00233*, 2021.

[173] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks.,” in *ICML*, vol. 2, p. 7, 2016.

[174] S. Roychoudhury, M. Ghalwash, and Z. Obradovic, “Cost sensitive time-series classification,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 495–511, Springer, 2017.

[175] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*, vol. 10. Springer, 2018.

[176] E. Amid and M. K. Warmuth, “Reparameterizing mirror descent as gradient descent,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 8430–8439, 2020.

[177] G. Raskutti and S. Mukherjee, “The information geometry of mirror descent,” *stat*, vol. 1050, p. 29, 2014.

[178] P. Bertram *et al.*, “Spironolactone for heart failure with preserved ejection fraction. treatment of preserved cardiac function heart failure with an aldosterone antagonist (topcat trial),” *N Engl J Med*, vol. 370, pp. 1383–92, 2014.

[179] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, “Multitask learning and benchmarking with clinical time series data,” *Scientific data*, vol. 6, no. 1, pp. 1–18, 2019.

[180] S. Angraal, B. J. Mortazavi, A. Gupta, R. Khera, T. Ahmad, N. R. Desai, D. L. Jacoby, F. A. Masoudi, J. A. Spertus, and H. M. Krumholz, “Machine learning prediction of mortality and hospitalization in heart failure with preserved ejection fraction,” *JACC: Heart Failure*, vol. 8, no. 1, pp. 12–21, 2020.

[181] M. Y. Arafat, S. Hoque, and D. M. Farid, “Cluster-based under-sampling with random forest for multi-class imbalanced classification,” in *2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pp. 1–6, IEEE, 2017.

[182] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[183] J. Deasy, A. Ercole, and P. Liò, “Impact of novel aggregation methods for flexible, time-sensitive ehr prediction without variable selection or cleaning,” *arXiv preprint arXiv:1909.08981*, 2019.

[184] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, 2006.

[185] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets,” *PloS one*, vol. 10, no. 3, p. e0118432, 2015.

[186] G. W. Brier *et al.*, “Verification of forecasts expressed in terms of probability,” *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

[187] E. W. Steyerberg, A. J. Vickers, N. R. Cook, T. Gerds, M. Gonen, N. Obuchowski, M. J. Pencina, and M. W. Kattan, “Assessing the performance of prediction models: a framework for some traditional and novel measures,” *Epidemiology (Cambridge, Mass.)*, vol. 21, no. 1, p. 128, 2010.

[188] N.-C. C. D. Center, “Brier skill scores, rocs, and economic value diagrams can report false skill,” 2005.

[189] A. P. Weigel, M. A. Liniger, and C. Appenzeller, “The discrete brier and ranked probability skill scores,” *Monthly Weather Review*, vol. 135, no. 1, pp. 118–124, 2007.

[190] S. J. Leadbetter, A. R. Jones, and M. C. Hort, “Assessing the value meteorological ensembles add to dispersion modelling using hypothetical releases,” *Atmospheric Chemistry and Physics*, vol. 22, no. 1, pp. 577–596, 2022.

[191] J. Hilden, “The area under the roc curve and its competitors,” *Medical Decision Making*, vol. 11, no. 2, pp. 95–101, 1991.

[192] J.-F. Truchon and C. I. Bayly, “Evaluating virtual screening methods: good and bad met-

rics for the “early recognition” problem,” *Journal of chemical information and modeling*, vol. 47, no. 2, pp. 488–508, 2007.

[193] B. Van Calster, D. J. McLernon, M. Van Smeden, L. Wynants, and E. W. Steyerberg, “Calibration: the achilles heel of predictive analytics,” *BMC medicine*, vol. 17, no. 1, pp. 1–7, 2019.

[194] J. Platt *et al.*, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[195] N. P. Fellowship and N. I. T. Grant, “B. jill venton, ph. d.,” *Journal Advisory Board*, vol. 2008, no. 2010, 2008.

[196] W. S. Hong, A. D. Haimovich, and R. A. Taylor, “Predicting hospital admission at emergency department triage using machine learning,” *PloS one*, vol. 13, no. 7, p. e0201016, 2018.

[197] S. Horng, D. A. Sontag, Y. Halpern, Y. Jernite, N. I. Shapiro, and L. A. Nathanson, “Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning,” *PloS one*, vol. 12, no. 4, p. e0174708, 2017.

[198] S. L. Bernstein, D. Aronsky, R. Duseja, S. Epstein, D. Handel, U. Hwang, M. McCarthy, K. John McConnell, J. M. Pines, N. Rathlev, *et al.*, “The effect of emergency department crowding on clinically oriented outcomes,” *Academic Emergency Medicine*, vol. 16, no. 1, pp. 1–10, 2009.

[199] Y. Sun, B. H. Heng, S. Y. Tay, and E. Seow, “Predicting hospital admissions at emergency department triage using routine administrative data,” *Academic Emergency Medicine*, vol. 18, no. 8, pp. 844–850, 2011.

[200] G. J. Katuwal and R. Chen, “Machine learning model interpretability for precision medicine,” *arXiv preprint arXiv:1610.09045*, 2016.

[201] J. Gui, Z. Sun, W. Jia, R. Hu, Y. Lei, and S. Ji, “Discriminant sparse neighborhood preserving embedding for face recognition,” *Pattern Recognition*, vol. 45, no. 8, pp. 2884–2893,

2012.

[202] M. Faruqui, Y. Tsvetkov, D. Yogatama, C. Dyer, and N. Smith, "Sparse overcomplete word vector representations," *arXiv preprint arXiv:1506.02004*, 2015.

[203] M. W. Fakhr, "Sparse locally linear and neighbor embedding for nonlinear time series prediction," in *2015 Tenth International Conference on Computer Engineering & Systems*, pp. 371–377, 2015.

[204] A. Subramanian, D. Pruthi, H. Jhamtani, T. Berg-Kirkpatrick, and E. Hovy, "Spine: Sparse interpretable neural embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[205] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[206] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Deepcare: A deep dynamic memory model for predictive medicine," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 30–41, Springer, 2016.

[207] J. Henderson, J. Ho, and J. Ghosh, "gamaid: Greedy cp tensor decomposition for supervised ehr-based disease trajectory differentiation," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3644–3647, IEEE, 2017.

[208] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor ai: Predicting clinical events via recurrent neural networks," in *Machine learning for healthcare conference*, pp. 301–318, PMLR, 2016.

[209] B. K. Beaulieu-Jones, C. S. Greene, *et al.*, "Semi-supervised learning of the electronic health record for phenotype stratification," *Journal of biomedical informatics*, vol. 64, pp. 168–178, 2016.

[210] D. J. Stekhoven and P. Bühlmann, "Missforest—non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012.

[211] F. O. de França, G. P. Coelho, and F. J. Von Zuben, "Predicting missing values with biclus-

tering: A coherence-based approach," *Pattern Recognition*, vol. 46, no. 5, pp. 1255–1266, 2013.

[212] X. Yan, W. Xiong, L. Hu, F. Wang, and K. Zhao, "Missing value imputation based on gaussian mixture model for the internet of things," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[213] A. Abedin, S. H. Rezatofighi, Q. Shi, and D. C. Ranasinghe, "Sparsesense: Human activity recognition from highly sparse sensor data-streams using set-based neural networks," *arXiv preprint arXiv:1906.02399*, 2019.

[214] Z. C. Lipton, D. C. Kale, R. Wetzel, *et al.*, "Modeling missing data in clinical time series with rnns," *Machine Learning for Healthcare*, vol. 56, 2016.

[215] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

[216] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1120–1124, 2014.

[217] Y. Vaizman, N. Weibel, and G. Lanckriet, "Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–22, 2018.

[218] G. Madhu, B. L. Bharadwaj, G. Nagachandrika, and K. S. Vardhan, "A novel algorithm for missing data imputation on machine learning," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 173–177, IEEE, 2019.

[219] E. Moghaddam, J. A. Vogt, and T. M. S. Wolever, "The Effects of Fat and Protein on Glycemic Responses in Nondiabetic Humans Vary with Waist Circumference, Fasting Plasma Insulin, and Dietary Fiber Intake," *The Journal of Nutrition*, vol. 136, pp. 2506–2511, 10 2006.

[220] F. Cordeiro, E. Bales, E. Cherry, and J. Fogarty, "Rethinking the mobile food journal: Exploring opportunities for lightweight photo-based capture," in *Proceedings of the 33rd An-*

nual ACM Conference on Human Factors in Computing Systems, pp. 3207–3216, ACM, 2015.

[221] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, "Automatic diet monitoring: a review of computer vision and wearable sensor-based methods," *International journal of food sciences and nutrition*, vol. 68, pp. 1–15, 01 2017.

[222] J. Chae, I. Woo, S. Kim, R. Maciejewski, F. Zhu, E. J. Delp, C. J. Boushey, and D. S. Ebert, "Volume estimation using food specific shape templates in mobile image-based dietary assessment," in *Computational Imaging IX*, vol. 7873, p. 78730K, International Society for Optics and Photonics, 2011.

[223] S. Fang, Z. Shao, D. A. Kerr, C. J. Boushey, and F. Zhu, "An end-to-end image-based automatic food energy estimation technique based on learned energy distribution images: Protocol and methodology," *Nutrients*, vol. 11, no. 4, p. 877, 2019.

[224] H. Kalantarian, N. Alshurafa, and M. Sarrafzadeh, "A survey of diet monitoring technology," *IEEE Pervasive Computing*, vol. 16, pp. 57–65, jan 2017.

[225] H. Kalantarian, N. Alshurafa, T. Le, and M. Sarrafzadeh, "Monitoring eating habits using a piezoelectric sensor-based necklace," *Computers in Biology and Medicine*, vol. 58, 01 2015.

[226] E. Sazonov, O. Makeyev, S. Schuckers, P. Lopez-Meyer, E. Melanson, and M. Neuman, "Automatic detection of swallowing events by acoustical means for applications of monitoring of ingestive behavior," *IEEE transactions on bio-medical engineering*, vol. 57, pp. 626–33, 09 2009.

[227] M. Farooq and E. Sazonov, "A novel wearable device for food intake and physical activity recognition," *Sensors*, vol. 16, p. 1067, 07 2016.

[228] R. Zhang and O. Amft, "Monitoring chewing and eating in free-living using smart eyeglasses," *IEEE journal of biomedical and health informatics*, vol. 22, no. 1, pp. 23–32, 2017.

[229] A. Kadomura, C.-Y. Li, K. Tsukada, H.-H. Chu, and I. Siio, "Persuasive technology to improve eating behavior using a sensor-embedded fork," in *Proceedings of the 2014 acm*

international joint conference on pervasive and ubiquitous computing, pp. 319–329, ACM, 2014.

[230] Y. Shen, J. Salley, E. Muth, and A. Hoover, "Assessing the accuracy of a wrist motion tracking method for counting bites across demographic and food variables," *IEEE journal of biomedical and health informatics*, vol. PP, 09 2016.

[231] V. Papapanagiotou, C. Diou, L. Zhou, J. Boer, M. Mars, and A. Delopoulos, "A novel chewing detection system based on ppg, audio, and accelerometry," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, pp. 1–1, 11 2016.

[232] J. Brand-Miller, K. Stockmann, F. Atkinson, P. Petocz, and G. Denyer, "Glycemic index, postprandial glycemia, and the shape of the curve in healthy subjects: Analysis of a database of more than 1000 foods," *The American journal of clinical nutrition*, vol. 89, pp. 97–105, 01 2009.

[233] R. Vigersky and M. Shrivastav, "Role of continuous glucose monitoring for type 2 in diabetes management and research," *Journal of Diabetes and its Complications*, vol. 31, no. 1, pp. 280 – 287, 2017.

[234] G. Freckmann, S. Hagenlocher, A. Baumstark, N. Jendrike, R. Gillen, K. Rössner, and C. Haug, "Continuous glucose profiles in healthy subjects under everyday life conditions and after different meals," *Journal of diabetes science and technology*, vol. 1, pp. 695–703, 09 2007.

[235] S. H. Holt, J. C. Miller, and P. Petocz, "An insulin index of foods: the insulin demand generated by 1000-kJ portions of common foods," *The American Journal of Clinical Nutrition*, vol. 66, pp. 1264–1276, 11 1997.

[236] I. Fox, L. Ang, M. Jaiswal, R. Pop-Busui, and J. Wiens, "Contextual motifs: Increasing the utility of motifs using contextual data," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 155–164, ACM, 2017.

[237] H. Mendes-Soares, T. Raveh-Sadka, S. Azulay, K. Edens, Y. Ben-Shlomo, Y. Cohen, T. Ofek, D. Bachrach, J. Stevens, D. Colibaseanu, *et al.*, "Assessment of a personalized

approach to predicting postprandial glycemic responses to food among individuals without diabetes," *JAMA network open*, vol. 2, no. 2, pp. e188102–e188102, 2019.

[238] T. M. Wolever and C. Bolognesi, "Prediction of glucose and insulin responses of normal subjects after consuming mixed meals varying in energy, protein, fat, carbohydrate and glycemic index," *The Journal of nutrition*, vol. 126, no. 11, pp. 2807–2812, 1996.

[239] Y. J. Rozendaal, A. H. Maas, C. van Pul, E. J. Cottaar, H. R. Haak, P. A. Hilbers, and N. A. van Riel, "Model-based analysis of postprandial glycemic response dynamics for different types of food," *Clinical Nutrition Experimental*, vol. 19, pp. 32–45, 2018.

[240] T. M. Wolever and D. J. Jenkins, "The use of the glycémie index in predicting the blood glucose response to mixed meals," *The American journal of clinical nutrition*, vol. 43, no. 1, pp. 167–172, 1986.

[241] M. González-Rodríguez, M. Pazos-Couselo, J. M. García-López, S. Rodríguez-Segade, J. Rodríguez-García, C. Túñez-Bastida, and F. Gude, "Postprandial glycemic response in a non-diabetic adult population: the effect of nutrients is different between men and women," *Nutrition & metabolism*, vol. 16, no. 1, p. 46, 2019.

[242] D. Zeevi, T. Korem, N. Zmora, D. Israeli, D. Rothschild, A. Weinberger, O. Ben-Yacov, D. Lador, T. Avnit-Sagi, M. Lotan-Pompan, *et al.*, "Personalized nutrition by prediction of glycemic responses," *Cell*, vol. 163, no. 5, pp. 1079–1094, 2015.

[243] J. A. Wagner, H. Tennen, and H. Wolpert, "Continuous glucose monitoring: a review for behavioral researchers.," *Psychosomatic medicine*, vol. 74 4, pp. 356–65, 2012.

APPENDIX A

SUPPLEMENT MATERIAL

Table A.1: List of Candidate Variables used for Predicting Mortality of TOPCAT dataset. Reprinted/adapted with permission from [4].

Variable Names	Definition
age_entry.x	Age entering the study
GENDER.x	Gender of the subject
RACE_WHITE	White or Caucasian
RACE_BLACK	Race: Black
RACE_ASIAN	Race: Asian
RACE_OTHER	Race: Other
ETHNICITY	Subject of Hispanic, Latino, or Spanish origin?
DYSP_CUR	Dyspnea: Present at screening?
DYSP_YR	Dyspnea: experienced in past year?
ORT_CUR	Orthopnea: Present at screening?
ORT_YR	Orthopnea: experienced in past year?
DOE_CUR	Dyspnea on exertion: Present at screening?
DOE_YR	Dyspnea on exertion: experienced in past year?
RALES_CUR	Rales present at screening?
RALES_YR	Rales: experienced in past year?
JVP_CUR	JVP: Present at screening?
JVP_YR	JVP: experienced in past year?
EDEMA_CUR	Edema: Present at screening?
EDEMA_YR	Edema: experienced in past year?
EF	Ejection Fraction

CHF_HOSP	Previous hospitalization for CHF
chfdc_dt3	Time Between randomization and Hospitalization for Cardiac Heart Failure (years)
MI	Previous myocardial infarction
STROKE	Previous Stroke
CABG	Previous Coronary artery bypass graft surgery
PCI	Previous Percutaneous Coronary Revascularization
ANGINA	Angina Pectoris
COPD	Chronic Obstructive Pulmonary Disease
ASTHMA	Asthma
HTN	Hypertension
PAD	Peripheral Arterial Disease
DYSLIPID	Dyslipidemia
ICD	Implanted cardioverter defibrillator
PACEMAKER	Pacemaker implanted
AFIB	Atrial fibrillation
DM	Diabetes Mellitus
treat_sp_cat	Treat for diabetes mellitus: other: specify (categorical variable)
SMOKE_EVER	Has subject ever been a smoker
QUIT_YRS	How many years since quitting
alcohol4_cat	How many Drinks do you consume per week (0/1-5/5-10/11+)
HEAVY_WK	Exercise: Heavy
MED_WK	Exercise: Medium
LIGHT_WK	Exercise: Light
LIGHT_MIN	Exercise: Light: Minutes
mets per week	Activity Level (mets per week)
cooking_salt_score	Cooking Salt Score

nyha_class_cat	NYHA class 3&4 vs 1&2
HR.x	Heart rate
SBP	Systolic blood pressure
DBP	Diastolic blood pressure
gfr	Glomerular Filtration Rate
NA_mmolL	Sodium: Result (mmol/L)
K_mmolL	Potassium: Result (mmol/L)
CL_mmolL	Chloride: Result (mmol/L)
CO2_mmolL	CO2: Result (mmol/L)
BUN_mgdL	Blood Urea Nitrogen: Result (mg/dL)
GLUCOSE_mgdL	Glucose: Result (mg/dL)
GLUCOSE_INDICATOR	Whether the glucose measured was fasting or random
WBC_k/μL	WBC count: Result (k/uL)
HB_gdL	Hemoglobin: Result (g/dL)
PLT_k/μL	Platelet Count: Result (k/uL)
ALT_UL	Alanine Aminotransferase: Results (U/L)
ALP_UL	Alkaline Phosphatase: Results (U/L)
AST_UL	Aspartate Aminotransferase: Results (U/L)
TBILI_mgdL	Total Bilirubin: Results (mg/dL)
ALB_gdL	Albumin: Results (g/dL)
urine_val_mgg	Urine Microalbumin/Creatinine Ratio: Result (mg/g)
QRS_DUR	QRS Duration
ECG_AFIB	Atrial fibrillation/Flutter
ECG_BBB2	Bundle Branch Block - Yes/No indicator
ECG_VPR	Ventricular paced rhythm
ECG_Q	Pathological Q waves
ECG_LVH	Left ventricular hypertrophy

drug.x	Treatment Group (Spironolactone or Placebo)
BMI	Body Mass Index
cigpacksperday	Number of cigarettes per day
phys_limit_score	KCCQ: Physical Limitation score
symp_stab_score	KCCQ: Symptom Stability score
symp_freq_score	KCCQ: Symptom Frequency score
symp_bur_score	KCCQ: Symptom Burden score
tot_symp_score	KCCQ: Total Symptom score
self_eff_score	KCCQ: Self-Efficacy score
qol_score	KCCQ: Quality of Life score
soc_limit_score	KCCQ: Social Limitation score
overall_sum_score	KCCQ: Overall Summary score
clin_sum_score	KCCQ: Clinical Summary score

Table A.2: Phenotype labels for MIMIC-III dataset. Reprinted/adapted with permission from [4].

Phenotype	type
Acute and unspecified renal failure	acute
Essential hypertension	chronic
Acute cerebrovascular disease	acute
Fluid and electrolyte disorders	acute
Acute myocardial infarction	acute
Gastrointestinal hemorrhage	acute
Respiratory failure; insufficiency; arrest	acute
Hypertension with complications	chronic
Chronic kidney disease	chronic
Other liver diseases	mixed
Chronic obstructive pulmonary disease	chronic
Other lower respiratory disease	acute
Complications of surgical/medical care	acute
Other upper respiratory disease	acute
Pleurisy; pneumothorax; pulmonary collapse	acute
Conduction disorders	mixed
Congestive heart failure; nonhypertensive	mixed
Pneumonia	acute
Coronary atherosclerosis and related	chronic
Cardiac dysrhythmias	mixed
Diabetes mellitus with complications	mixed
Diabetes mellitus without complication	chronic
Disorders of lipid metabolism	chronic
Septicemia (except in labor)	acute
Shock	acute