

ABSTRACT

HICKEY, JIMMY. Transfer Learning and Survival Analysis Methods with Biomedical Applications. (Under the direction of Emily Hector and Jonathan Williams).

Electronic Health Record (EHR) systems have been adopted by a majority of hospitals and office-based doctors in the United States. These systems track every patient encounter from admission to discharge. While large research hospitals are flush with data, smaller hospitals may lack necessary data to reliably make predictions or inferences. Transfer learning takes information learned in a source domain and uses it to aid in making predictions or inferences in a target domain, accounting for the differences between the domains. In Chapter 2 we develop *RECaST*, a Bayesian transfer learning method that provides uncertainty quantification for predictions while maintaining data privacy between domains. We apply our method to leverage EHR data between hospitals to predict patient shock in the ICU. In Chapter 3 we extend this framework to consider multivariate outcomes and online transfer learning with sequential targets. These methods are validated in simulation studies and in an analysis of dental data to predict periodontal outcomes.

While EHR data represents everything that happens in a healthcare system, cohort studies are carefully designed experiments that follow a group of patients to observe whether they experience a particular event. Survival analysis methods are used to determine patient risk and expected time until an event occurs. In clinical settings with limited available data, it is often preferable to judiciously partition the event time space into a limited number of intervals well suited to the prediction task at hand. In Chapter 4 we develop a novel hierarchical method to learn, from the data, a set of cut points defining such a partition without making parametric assumptions. Avoiding placing parametric assumptions on the event density tends to improve predictive performance. We demonstrate improved predictive performance on three real-world observational datasets, including a large, newly harmonized stroke risk prediction dataset.

© Copyright 2024 by Jimmy Hickey

All Rights Reserved

Transfer Learning and Survival Analysis Methods with Biomedical Applications

by
Jimmy Hickey

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Statistics

Raleigh, North Carolina
2024

APPROVED BY:

Matthew Engelhard

Emily Griffith

Brian Reich

Kazufumi Ito

Emily Hector
Co-chair of Advisory Committee

Jonathan Williams
Co-chair of Advisory Committee

DEDICATION

To Linh.

BIOGRAPHY

Jimmy thinks it is odd to talk about himself in the third person like the Seinfeld character of the same name, but he'll persevere. He grew up in Arlington Heights, Illinois before graduating from Winona State University in 2018 with Bachelor degrees in Computer Science, Physics, and Mathematics. During this time he proudly served as a peer tutor and a software developer for Digi International. Following this, Jimmy worked for Mayo Clinic in the Genomics Systems Unit. In 2020 he earned a Master's of Statistics from North Carolina State University. During his doctoral studies, he worked with faculty at Duke University and with an interdisciplinary group of statisticians and computer scientists at Sandia National Laboratories. Jimmy earned his doctorate in Statistics in 2024 under the direction of Dr. Jon Williams and Dr. Emily Hector.

ACKNOWLEDGEMENTS

I'd like to start by thanking my advisors Dr. Jon Williams and Dr. Emily Hector. To quote scientists completing an equally challenging feat as this dissertation, the opening of the Voyager Golden Record: "We step out of our solar system into the universe seeking only peace and friendship, to teach if we are called upon, to be taught if we are fortunate." I am incredibly fortunate to have been advised by the best the field has to offer. I am grateful for the additional feedback given to me along my journey from my committee members Dr. Brian Reich and Dr. Emily Griffith.

I would like to extend a special thank you to Dr. Marie Davidian and Dr. Ana-Maria Staicu for the opportunity to work on the NHLBI Integrated Biostatistical Training Program for CVD Research. Thank you to Hillary Mulder for guiding me through this program. I am greatly appreciative of Dr. Chuan Hong for her guidance through my first publication. A special thanks goes to Dr. Matt Engelhard for getting down in the mud with me to wrestle with Python and for practicing our Zoom drawings.

Thank you to Lyndsay Shand, Derek Tucker, and all of the folks at Sandia National Laboratories that I was lucky enough to work with.

I would like to thank my friends, who understand that extracurriculars are sometimes just as important as curriculars. Good game to the Stats Department Tennis League: my partner, Emmett "Em-net" Kendall, and my opponents, Eric Yanchenko and Brandon "B-Money" Feng. To Neil Dey, thank you for entertaining whatever entirely irrelevant curiosity I brought with me each day. To Naomi Giertych, I am grateful for your constant support and excellent crafting prowess. For preserving my sanity through the medium of tea, thank you Jonathan Fabish. To Alvin Sheng, I am thankful for our time working on the Statistics in the Community projects. Thank you to David Elsheimer for all of our group projects and for our fierce chess rivalry. I am grateful for the two best Stat Buddies a guy could ask for, Jake Koerner and Nate Wiecha. For their guidance as elder statesmen and stateswomen of the department, thank you to Drew Hollis, Cole Manschot, and Grace Rhodes.

Finally, I would like to thank my family. To my parents, Jim and Ann, and my sisters, Kaeley and Moira, thank you for understanding that "fine" is a perfectly acceptable answer to "How are things going?", even if it's my answer every single time. And thank you to my grandmother Hiya for teaching me that. To Linh, thank you for your constant love and support. Thank you for every meal you've made, for every joke you've told, and for every sitcom we've watched.

TABLE OF CONTENTS

List of Tables	vii
List of Figures	xi
Chapter 1 Introduction	1
1.1 Transfer Learning	2
1.2 Survival Timeline Partitioning	4
Chapter 2 Transfer Learning with Uncertainty Quantification: Random Effect Calibration of Source to Target (RECaST)	5
2.1 Introduction	5
2.1.1 Our Contributions	6
2.2 Related Work	8
2.3 RECaST Framework	10
2.4 Continuous Response Data	13
2.4.1 Model and Estimation	13
2.4.2 Remarks on Implementation	14
2.4.3 Theoretical Guarantees	15
2.5 Binary Response Data	17
2.5.1 Model and Estimation	17
2.5.2 Remarks on Implementation	18
2.6 Simulation Study	19
2.6.1 Objectives and Setup	19
2.6.2 Continuous Response Results	20
2.6.3 Binary Response Results	23
2.6.4 Robustness of RECaST	25
2.7 eICU Data	28
2.8 Concluding Remarks	31
Chapter 3 Multivariate and Online Transfer Learning with Uncertainty Quantification 33	
3.1 Introduction	33
3.2 Related Work	35
3.3 Summary of the RECaST Framework	37
3.4 Multivariate RECaST	38
3.4.1 Data Generating Mechanism	38
3.4.2 Multivariate Cauchy	39
3.4.3 Multivariate Normal Copula with Cauchy Marginals	41
3.5 Online RECaST	43
3.6 Simulation Study	46
3.6.1 Overview	46
3.6.2 Multivariate Single Target Simulations	47
3.6.3 Multivariate Online Simulations	50
3.7 Dental Data Analysis	56

3.8	Concluding Remarks	61
Chapter 4	Adaptive Discretization for Event Prediction (ADEPT)	62
4.1	Introduction	62
4.2	Methods	64
4.2.1	Setup and Notation	64
4.2.2	Piecewise Constant Density	65
4.2.3	Smooth Relaxation of Piecewise Density	66
4.2.4	Learning Procedure	66
4.3	Implementation Details	67
4.3.1	Baseline Model: Discrete-Time Neural Network	67
4.3.2	Performance Quantification	68
4.3.3	Hyperparameter Tuning	69
4.4	Simulation Examples	72
4.4.1	Learning Two Intervals	72
4.4.2	Learning Four Intervals	74
4.5	Data Analysis	75
4.5.1	Real-World Data Sources	75
4.5.2	Results	75
4.6	Conclusion	76
References		79
APPENDIX		88
Appendix A	Chapter 2 Supplementary Material	89
A.1	Proofs	89
A.2	Bounding Continuous Integral	94
A.3	MCMC Implementation Details	94
A.4	Neural Network Training Procedure	95
A.5	Additional Tables for Section 2.6.4	96
A.6	Additional Robustness Results	96
A.7	Comparative eICU Results	101
A.8	eICU Feature Descriptions	102
A.9	Multivariate Cauchy Gibbs Sampler	103
A.10	Multivariate Gaussian Copula Finite Integrals	104
A.11	Online Mean Derivation	104

LIST OF TABLES

Table 2.1	Empirical coverage (standard error) at the 95% nominal level, averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	22
Table 2.2	Out-of-sample RMSE (standard error) averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations.	23
Table 2.3	Empirical coverage (standard error) at the 95% nominal level, averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	24
Table 2.4	Out-of-sample AUC (standard error) averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	25
Table 2.5	Out of sample RMSE (standard error) averaged over 300 source and target data sets when the generating models are neural networks and the target model parameters are generated as $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.025\mathbf{I})$. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	26
Table 2.6	Out-of-sample AUC (standard error) averaged over 300 source and target data sets when the generating models are neural networks and the target model parameters are generated as $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.025\mathbf{I})$. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	27
Table 2.7	Empirical coverage (standard error) at the 75% nominal level for a binary response, averaged over 300 source and target data sets when the generating models are neural networks and the target model parameters are generated as $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.025\mathbf{I})$. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	28
Table 2.8	Out of sample RMSE (standard error) averaged over 300 source and target data sets when the source and target neural network weight matrices are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	28
Table 2.9	Out-of-sample AUC (standard error) averaged over 300 source and target data sets when the source and target neural network weight matrices are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	29
Table 2.10	Empirical coverage (standard error) at the 75% nominal level for a binary response, averaged over 300 source and target data sets when the source and target neural network weight matrices are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	29
Table 2.11	Out-of-sample AUC (standard error) [empirical coverage at the 80% nominal level] averaged over 300 target training and testing data sets for each target data setting of the eICU data. All reported values are multiplied by 100.	31

Table 3.1	Performance metrics for $\Theta_{T,1} = \Theta_{S,1} + U(0, a)$ and $\Theta_{T,2} = \Theta_{S,2} + U(b, 0)$ with $\Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. The reported values are averaged over 100 source and target data sets: average Mahalanobis distance (standard error of Mahalanobis distances) [empirical coverage at the 95% nominal level]. For univariate RECaST the marginal coverages are reported.	48
Table 3.2	Performance metrics for $\Theta_{T,1} = \Theta_{S,1} + U(0, a)$ and $\Theta_{T,2} = \Theta_{S,2} + U(b, 0)$ with $\Sigma = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 0 & 0 \\ 0.5 & 0 & 1 \end{bmatrix}$. The reported values are averaged over 100 source and target data sets: average Mahalanobis distance (standard error of Mahalanobis distances) [empirical coverage at the 95% nominal level]. For univariate RECaST the marginal coverages are reported.	49
Table 3.3	Performance metrics for $\Theta_{T,1} = \Theta_{S,1} + U(0, a)$ and $\Theta_{T,2} = \Theta_{S,2} + U(b, 0)$ with $\Sigma = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0 & 0 \\ -0.5 & 0 & 1 \end{bmatrix}$. The reported values are averaged over 100 source and target data sets: average Mahalanobis distance (standard error of Mahalanobis distances) [empirical coverage at the 95% nominal level]. For univariate RECaST the marginal coverages are reported.	49
Table 3.4	Performance metrics for $\Theta_T = c \cdot \Theta_S$. The reported values are averaged over 100 source and target data sets: average Mahalanobis distance (standard error of Mahalanobis distances) [empirical coverage at the 95% nominal level]. For univariate RECaST, the marginal coverages are reported.	50
Table 3.5	Mahalanobis distances for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. The reported values are: Average Mahalanobis distance (standard error of Mahalanobis distances)	52
Table 3.6	Mahalanobis distances for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 0 & 0 \\ 0.5 & 0 & 1 \end{bmatrix}$. The reported values are: Average Mahalanobis distance (standard error of Mahalanobis distances)	53
Table 3.7	Mahalanobis distances for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0 & 0 \\ -0.5 & 0 & 1 \end{bmatrix}$. The reported values are: Average Mahalanobis distance (standard error of Mahalanobis distances)	53
Table 3.8	Empirical coverage values at the 95% nominal level for multivariate, online RECaST and empirical coverage values averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. For univariate RECaST the marginal coverages are reported.	53
Table 3.9	Empirical coverage values at the 95% nominal level for multivariate, online RECaST and empirical coverage values averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 0 & 0 \\ 0.5 & 0 & 1 \end{bmatrix}$. For univariate RECaST the marginal coverages are reported.	54
Table 3.10	Empirical coverage values at the 95% nominal level for multivariate, online RECaST and empirical coverage values averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0 & 0 \\ -0.5 & 0 & 1 \end{bmatrix}$. For univariate RECaST the marginal coverages are reported.	54

Table 3.11	Posterior means of α for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$: average posterior mean (standard deviation of the posterior means).	55
Table 3.12	Posterior means of α for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 \end{bmatrix}$: average posterior mean (standard deviation of the posterior means).	55
Table 3.13	Posterior means of α for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0.5 \\ -0.5 & 0.5 & 1 \end{bmatrix}$: average posterior mean (standard deviation of the posterior means).	56
Table 3.14	HP Data feature and outcome summaries stratified by racial group. For age, CAL, and PD the 25 th , 50 th , and 75 th percentiles are presented.	58
Table 3.15	Performance metrics averaged over 10-fold cross validation. We report the joint empirical coverage at the 95% nominal level; for univariate RECaST the marginal coverages are reported. For the online methods we report the posterior mean of α averaged across the fold. The source population is the White participants. The target of interest is the Asian participants where 10% of the data was used for training in each cross validation fold ($n_{T_2} = 105$). For the online methods, the informative prior is created using the corresponding multivariate RECaST method with the Black or African American participants.	59
Table 3.16	Performance metrics averaged over 10-fold cross validation. We report the joint empirical coverage at the 95% nominal level; for univariate RECaST the marginal coverages are reported. For the online methods we report the posterior mean of α averaged across the fold. The source population is the White participants. The target of interest is the Native American and Alaska Native participants where 10% of the data was used for training in each cross validation fold ($n_{T_2} = 15$). For the online methods, the informative prior is created using the corresponding multivariate RECaST method with the Black or African American participants.	60
Table 3.17	Performance metrics averaged over 10-fold cross validation. We report the joint empirical coverage at the 95% nominal level; for univariate RECaST the marginal coverages are reported. For the online methods we report the posterior mean of α averaged across the fold. The source population is the White participants. The target of interest is the Native American and Alaska Native participants where 10% of the data was used for training in each cross validation fold ($n_{T_2} = 15$). For the online methods, the informative prior is created using the corresponding multivariate RECaST method with the Asian participants.	60
Table 4.1	Performance metrics for synthetic data. We report average metrics across 5-fold cross validation with standard errors in parentheses. In bold are the highest CI values for each setting.	72

Table 4.2	Test-set performance metrics for real-world data averaged across 5-fold cross validation with standard errors in parentheses. In bold are the highest CI, highest AUC, and lowest IBS models for each data set.	78
Table A.1	Empirical coverage (standard error) at the 95% nominal level for a continuous response, averaged over 300 source and target data sets when the when the target model parameters are generated as $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}_{p \times 10 + 10 \times 1}(\mathbf{0}, 0.025\mathbf{I})$. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	96
Table A.2	Empirical coverage (standard error) at the 95% nominal level for a continuous response, averaged over 300 source and target data sets when the source and target neural network weight matrices are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	96
Table A.3	Out of sample RMSE (standard error) averaged over 300 source and target data sets when the source data generating parameters are orthogonal to the target data generating parameters. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	97
Table A.4	Empirical coverage (standard error) at the 95% nominal level, averaged over 300 source and target data sets when the source data generating parameters are orthogonal to the target data generating parameters. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	97
Table A.5	Out-of-sample AUC (standard error) averaged over 300 source and target data sets when the source and target model parameter vectors are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	98
Table A.6	Empirical coverage (standard error) at the 75% nominal level, averaged over 300 source and target data sets when the source and target model parameter vectors are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	98
Table A.7	The reported values are: average out-of-sample RMSE (standard deviation). These summaries are over all 300 different source and target data sets for each target sample size when the target data had more features than the source.	99
Table A.8	Empirical coverage (standard error) at the 95% nominal level, averaged over 300 source and target data sets when the target data had more features than the source. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	99
Table A.9	Out-of-sample AUC (standard error) averaged over 300 source and target data sets when the target data had more features than the source. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	100
Table A.10	Empirical coverage (standard error) at the 75% nominal level, averaged over 300 source and target data sets when the target data had more features than the source. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.	100
Table A.11	Descriptions of the features from the eICU Collaborative Research Database used in the shock data analysis.	102

LIST OF FIGURES

Figure 2.1	Reliability curves of the nominal coverage versus the empirical coverage, averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations. The left panel shows an easy setting: $n_T = 100$ and $\sigma_{TL}^2 = 0.25$. The right panel shows a difficult setting: $n_T = 20$ and $\sigma_{TL}^2 = 4$	21
Figure 2.2	The left panel displays the reliability curve of the nominal versus empirical out-of-sample coverage of prediction sets averaged over 300 target-testing data sets; the right panel reports the out-of-sample receiver operating characteristic (ROC) curve averaged pointwise over 300 target-testing data sets. The legend also reports the AUC (standard error) averaged over the same 300 target-testing data sets. Note that we cut the reliability curve at a nominal coverage of 0.8 because there are very few observations with higher coverage, undermining the reliability of coverage estimation at higher nominal levels.	30
Figure 3.1	The outcome measurements for each participant’s first visit.	57
Figure 4.1	The event time space partitioned by three cut points into four intervals.	65
Figure 4.2	A training plot of the training and validation loss at each epoch for the two interval simulation example. The vertical lines represent drops in sigmoid temperature τ and the accompanying new value of τ	70
Figure 4.3	The event times and observed times of the two interval data. The true cut point is at time 67.	71
Figure 4.4	Event times and observed times of the four interval data. The true cut points are at 10, 30, and 70.	73
Figure 4.5	The DTNN (red, dashed) and ADEPT learned (black, solid) cut points.	77
Figure A.1	Results for TransRF, glmtrans, and WDGRL on the eICU data set. The left panel displays the reliability curve of the nominal versus empirical out-of-sample coverage of prediction sets averaged over 300 target-testing data sets; the right panel reports the out-of-sample receiver operating characteristic (ROC) curve averaged pointwise over 300 target-testing data sets. The legend also reports the AUC (standard error) averaged over the same 300 target-testing data sets. Note that we cut the reliability curve at a nominal coverage of 0.8 because there are very few observations with higher coverage, undermining the reliability of coverage estimation at higher nominal levels.	101

CHAPTER

1

INTRODUCTION

It is necessary to apply statistical methods to biomedical problems in order to properly assess risk factors in disease diagnosis and survival prediction. These insights can help clinicians offer better care to patients, assist pharmaceutical companies to create medications, and guide policy maker decisions on public health. There are many sources of medical data; we will focus on Electronic Health Records (EHRs), observational dental data, and cohort studies.

EHR systems have become ubiquitous in the United States health care system, being used by 89.9% of office-based physicians (Myrick et al. 2022) and 96% of non-federal acute care hospitals (of the National Coordinator for Health Information Technology 2017). EHR systems contain data from each patient interaction in a health care system including demographic information, allergies, prescriptions, diagnoses, clinical notes, and more. In Chapter 2 we use EHR data from the publicly available, multi-center eICU Collaborative Database (Pollard et al. 2018). By combining Internal Classification of Diseases 10 (ICD-10) codes, we are able to create a diagnosis for patients that go into *shock*. In this database, shock is a rare outcome occurring in less than 2% of patients. Because of this, some hospitals lack the data to properly learn shock diagnoses. We address this problem by using transfer learning.

A longitudinal, observational data set of routine dental checkups is provided by the Health-Partners Institute in Minneapolis Minnesota (Guan et al. 2020). This study tracks demographic, dental, and insurance features of patients with at least 8 years of continual dental insurance

during the study period. Two outcomes are measured which are important to diagnosing periodontitis; however, these outcomes are difficult to measure and time consuming. In Chapter 3, we predict these outcomes prior to a patient’s first visit to the dentist. We also propose an online modeling approach to handle the underrepresentation of certain racial groups in the collected data.

Cohort studies, on the other hand, are carefully designed longitudinal studies that are often concerned with measuring a certain outcome. In Chapter 4 we examine three cohort studies. The first is the German Breast Cancer Study Group data set which is publicly available and introduced by Schumacher et al. (1994). This studies an endpoint of recurrence free survival. The second is the Assay of Serum Free Light Chain data set which is publicly available and introduced by Dispenzieri et al. (2012). It studies the relationship between nonclonal serum immunoglobulin free light chains and mortality. The final data set is three stroke risk cohorts pooled together. This combined dataset consists of the Framingham Offspring Study (Feinleib et al. 1975), The Atherosclerosis Risk in Communities Study (Investigators 1989), and the Multi-Ethnic Study of Atherosclerosis (Bild et al. 2002). A properly designed cohort study can produce high quality data that will be useful in improving patient care. A downside to cohort studies is that they are often expensive to run and require a long time horizon to gather data. By learning a partition of the survival timeline, we are able to provide accurate survival prediction even in cases where data are limited.

1.1 Transfer Learning

In this section we present a general transfer learning framework that will be used throughout Chapter 2. For consistency, this survey uses definitions and notation from past surveys (Pan and Yang 2010; Weiss et al. 2016). To offer a concrete example as notation is introduced, our goal is to train a learner to predict whether a patient will be diagnosed with shock.

A *domain* \mathcal{D} is characterized by two parts: a feature space \mathcal{X} and a marginal probability distribution $P(\mathbf{X})$, where $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \mathcal{X}$. In our example, the feature vector \mathbf{x}_i is the collection of measurements taken when patient i is admitted to the ICU, where there are n total patients. We observe a certain sample of patients, \mathbf{X} , from the all possible patient vectors, \mathcal{X} . Given a specific domain \mathcal{D} , a *task* \mathcal{T} is also characterized by two part: a label space \mathcal{Y} and a conditional distribution $P(\mathbf{Y} | \mathbf{X})$. The distribution of $P(\mathbf{Y} | \mathbf{X})$ is learned from feature and label pairs $\{\mathbf{x}_i, y_i\}$ where $\mathbf{x}_i \in \mathbf{X}$ and $y_i \in \mathcal{Y}$ and is responsible for predicting y_i based on \mathbf{x}_i . In our example, $\mathcal{Y} = \{\text{false}, \text{true}\}$ would be the set of labels that a patient could take such that $y_i = \text{false}$ means that patient i was not diagnosed with shock and $y_i = \text{true}$ means that they were. The predictive distribution $P(\mathbf{Y} | \mathbf{X})$ is the learner that predicts the discharge status

likelihood from the measurements such as a logistic model or a neural network.

Using these definitions, we can create two different domains: the source domain and target domain. First, define the *source domain data* as the observed features and labels pairs, $D_S = \{(\mathbf{x}_{S_1}, y_{S_1}), \dots, (\mathbf{x}_{S_{n_S}}, y_{S_{n_S}})\}$ where $\mathbf{x}_{S_i} \in \mathcal{X}_S$ and $y_{S_i} \in \mathcal{Y}_S$. Also define the source predictive distribution as $P_S(\mathbf{Y}_S | \mathbf{X}_S)$. Similarly, define *target domain data* as the observed features and labels pairs, $D_T = \{(\mathbf{x}_{T_1}, y_{T_1}), \dots, (\mathbf{x}_{T_{n_T}}, y_{T_{n_T}})\}$ where $\mathbf{x}_{T_i} \in \mathcal{X}_T$ and $y_{T_i} \in \mathcal{Y}_T$ and the target predictive distribution $P(Y_T | \mathbf{X}_T)$. With these definitions in place, we can define transfer learning. Given a source domain \mathcal{D}_S with a corresponding source task \mathcal{T}_S and a target domain \mathcal{D}_T with a corresponding target task \mathcal{T}_T , *transfer learning* aims to improve the learning of the target predictive distribution, $P(Y_T | \mathbf{X}_T)$, by incorporating knowledge from \mathcal{D}_S and \mathcal{T}_S , where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$. If $\mathcal{D}_S = \mathcal{D}_T$ and $\mathcal{T}_S = \mathcal{T}_T$, then this problem becomes a standard machine learning problem. For traditional machine learning applications to EHR data, see Shickel et al. (2017).

In Chapter 2 we introduce RECaST to address the transfer learning questions where there is covariate shift between the source and target $P(\mathbf{X}_S) \neq P(\mathbf{X}_T)$ and there is context feature bias $P(Y_S | \mathbf{X}_S) \neq P(Y_T | \mathbf{X}_T)$. We accomplish this by introducing a random effect term that captures the relationship between the source and target data models. Using a Bayesian approach to learning the distribution of this random effect allows us to learn the posterior predictive distribution of new observations which facilitates uncertainty quantification. Our method is able to accurately predict outcomes on unseen data and provide credible intervals for those predictions. We are able to do this without making assumptions on the source model or requiring access to the source data. These are especially meaningful benefits in medical applications. Many transfer learning methods require the source model to be of a specific form (for example, a neural network). If the model is not in this form, the target hospital would not be able to use it. RECaST allows the target hospital to use the source hospital’s model without any changes. The target hospital can do this without having any patient level data from the source hospital, alleviating privacy concerns.

In Chapter 3 we extend the RECaST framework to consider multivariate outcomes and to model sequential target data sets. The multivariate approach provides a source agnostic method to transfer learning while providing Bayesian uncertainty quantification of the joint outcomes. Sequential target data sets are modeled using an online framework where the current target is informed by the previous targets through the prior distribution. Through simulation studies and data analyses, we demonstrate the improvements in both predictive performance and uncertainty quantification when compare to univariate RECaST.

1.2 Survival Timeline Partitioning

Recently developed survival analysis methods improve upon older approaches by predicting the probability of event occurrence in each of a number pre-specified time intervals. This approach tends to improve prediction performance, particularly when data are plentiful. However, in clinical settings with limited available data, it is often preferable to judiciously partition the event time space into a limited number of intervals well suited to the prediction task at hand. For example, it may be useful to partition your dataset daily, but if there is not enough data to support that many intervals then the approach is not feasible. Learning a partition of the timeline based on the data allows for tuning of the proper number intervals that can reliably based on the size of the data set.

In Chapter 4 we develop a method to learn a set of cut points defining such a partition based on data. By avoiding strong parametric assumptions on the event density, we are able to apply this method to a wide array of applications. Our approach facilitates clinical decision-making by learning time intervals that are most appropriate for each task, in the sense that they provide more accurate risk prediction. In practice, better risk prediction can influence how long a drug is prescribed, how long monitoring takes place, or how long the patient remains anxious after receiving their prognosis.

CHAPTER

2

TRANSFER LEARNING WITH UNCERTAINTY QUANTIFICATION: RANDOM EFFECT CALIBRATION OF SOURCE TO TARGET (RECAST)

2.1 Introduction

The use of artificial intelligence and machine learning (ML) is frequently limited in practice by a shortage of available training data and insufficient computational resources. To address these difficulties, transfer learning has developed as a powerful idea for leveraging the resources at leading institutions such as research hospitals (e.g., institutions having high quality data, exceptional research clinicians, high performance computing environments, etc.) to facilitate implementation of ML technologies in resource scarce settings such as small or rural hospitals. Developments in transfer learning methodologies are necessary to overcome resource allocation inequities, and they will likely drive the next decade of innovation in ML technologies.

Transfer learning consists broadly of two elements. The first is one or more *target* population(s) of interest that are associated with data sets for which there are resource limitations preventing the training of sophisticated models (e.g., a small hospital). The second is a *source* population (or populations) that is separate but in some way related to the target population. The source is associated with extensive data and/or resources for training sophisticated ML models. The premise of transfer learning is to use trained source models to aid in the training of target models. The source and targets are each composed of two components: a *domain*, denoted \mathcal{D} , and a *task*, denoted \mathcal{T} . A domain $\mathcal{D} := \{\mathcal{X}, P(x)\}$ consists of a feature space \mathcal{X} and a marginal probability distribution $P(x)$ over $x \in \mathcal{X}$. A task $\mathcal{T} := \{\mathcal{Y}, P(y | x)\}$ is composed of a label space \mathcal{Y} and a conditional distribution $P(y | x)$ over $y \in \mathcal{Y}$ given $x \in \mathcal{X}$. Traditional ML is described by the source and target sharing the same domain, $\mathcal{D}_S = \mathcal{D}_T$, and sharing the same task, $\mathcal{T}_S = \mathcal{T}_T$. Transfer learning problems arise when the source and target domains and/or the source and target tasks are similar but different. We propose a new Bayesian transfer learning framework termed Random Effect Calibration of Source to Target (RECaST) for source and target data sets that share the same outcome space but possibly have different feature-to-outcome mappings.

2.1.1 Our Contributions

Early efforts in transfer learning focused on using labeled data to learn about unlabeled data from the *same* population (see Joachims (1999) and Vapnik (2009) for examples). In contrast, modern transfer learning methods explore how knowledge from one source domain can be applied to a *different* target domain. In this spirit, we consider transfer learning in the supervised learning problem that dominates ML applications. Our proposed method uses information from the source and target features and labels to build a predictive model that can be applied to obtain predictions of labels for new target data features of interest. The use of target labels is common across transfer learning and is sometimes referred to as *inductive* transfer learning (Pan and Yang 2010). For example, a method is proposed in Donahue et al. (2014a) to generalize a model built on ImageNet data for use on different labeled target data sets. A neural network is fine-tuned in Shao et al. (2019) to identify and classify machine faults. In Goussies et al. (2014), a decision forest is proposed that uses mixed information gain and label propagation to improve image and gesture recognition in the target domain.

RECaST is a Bayesian framework applied to the transfer learning setting where the feature-to-outcome mappings $P(y | x)$ may differ between the source and target. For example, source and target hospitals might record largely the same patient data features, but nuances in clinician practices/procedures, inconsistencies in data quality, population disparities, etc. may affect

the suitability of using the source mapping as the target mapping. RECaST uses an estimated source model in tandem with the target data to estimate the distributions of a random effect that links the two domains. It then uses the estimated posterior distribution of the random effect parent parameters to construct a posterior predictive distribution of the outcome variable associated with a new target feature. The posterior predictive credible sets obtained through RECaST deliver critical quantification of prediction uncertainty that is lacking in most existing frameworks.

Two primary advantages of RECaST are its scalability, requiring estimation of only 2-3 parameters with no tuning parameters, and that it is agnostic to the source model specification. Importantly, RECaST only requires the source model and parameter estimates, not the source data itself; this is an immense benefit to applications with privacy concerns, such as with medical data. Further, we show that RECaST is asymptotically valid in the canonical case of distinct source and target Gaussian linear models, in that the coverage of prediction sets are guaranteed to asymptotically achieve their stated nominal level of significance.

To evaluate our proposed RECaST approach, we design synthetic simulation studies with both continuous and binary response data reflecting a variety of difficulty levels of transfer learning problems. Next, we investigate the performance of RECaST in real data simulations that arise by permuting real patient data from the multi-center eICU Collaborative Research Database (Pollard et al. 2018). A variety of both point-valued and set-valued prediction metrics are considered, including the empirical coverage of prediction sets. The performance of RECaST is compared to other state-of-the-art transfer learning approaches, including other source-free methods that do not require the source data while learning the target model. These include freeze-unfreeze approaches that are popular for neural networks, as well as a method based on adapting a random forest built on the source data to the target data (Gu et al. 2022). In some cases, it may be possible to have access to both the source and target data. As such we also compare RECaST to methods that require both data sets during training. These include an adversarial learning method (Shen et al. 2018), a method based on penalized GLMs (Tian and Feng 2022), and a popular weighting approach used on clinical data (Wiens et al. 2014).

The remainder of our paper is organized as follows. We discuss related works in transfer learning in Section 2.2. In Section 2.3, we develop the theoretical basis for RECaST and its uncertainty quantification. We then develop Bayesian parameter estimation and prediction procedures in both the continuous and binary response cases in Sections 2.4 and 2.5, respectively. We conduct extensive simulation studies in Section 2.6 by exploring transfer learning problems of a range of difficulties. Section 2.7 considers a real data analysis for predicting shock in ICU data. Section 2.8 concludes. Proofs and computational details are provided in the Appendix. Throughout the paper we keep to the convention in the statistical literature of using

(\cdot) for innermost grouping followed by $\{\cdot\}$ and finally $[\cdot]$. Thus, an expression with many nested parentheses respects the ordering $[\{\{\cdot\}\}]$.

2.2 Related Work

General survey papers on transfer learning topics include Pan and Yang (2010); Lu et al. (2015); Weiss et al. (2016); Dube et al. (2020). For hospital disease risk and mortality prediction problems, Wiens et al. (2014), Gong et al. (2015), and Desautels et al. (2017) propose transfer learning approaches based on training algorithms using a learned weighted combination of source and target patient observations. These methods learn many parameters and require access to the source data. RECaST may at first glance appear similar to density ratio estimation, a common approach to transfer learning. Density ratio transfer learning methods, such as the one described in Stojanov et al. (2019), seek to learn the relationship between the source and target data via a ratio of their densities; however, these methods require joint access to the source and target data – a limitation avoided by RECaST. In Paul et al. (2016), Raghu et al. (2019), and Ahishakiye et al. (2021), approaches are considered to improve classification accuracy for medical imaging tasks using pre-trained deep neural networks (DNNs) on the ImageNet database (Deng et al. 2009). In the context of ICU patient monitoring, in Shickel et al. (2021) a data augmenting-based transfer learning approach is built for fitting a single-layer recurrent neural network trained on electronic health records (EHR) and wearable device data. Their model is limited in scope to only predicting the binary response of successful versus unsuccessful discharge from a hospital. Implemented in Gao and Cui (2021) is a transfer learning strategy for precision medicine in survival analysis with clinical omics data sets via freezing layers of a pre-trained Cox neural network. Developed in Lee et al. (2012) is a method using support vector machines to predict surgical mortality. Another approach, from Gu et al. (2023), is to generate additional synthetic target data from a source data set and adjust for heterogeneity in order to predict extreme obesity from medical records and genomics data. An example of low-dimensional representation transfer learning is given in Maurer et al. (2015), and *online* transfer learning is considered in Zhao et al. (2014); Wu et al. (2017a). These applied methods are useful in modeling specific pieces of EHR data for prediction, but lack uncertainty quantification. Additionally, some require the learning of many parameters and access to the entire source data set.

Bayesian transfer learning adaptations include Baxter (1998), Raina et al. (2006a), Wohlert et al. (2018), Bueno et al. (2020), Chandra and Kapoor (2020), Yang et al. (2020), Zhou et al. (2020), Abba et al. (2023a); all except Baxter (1998) and Raina et al. (2006a) are based on priors specified from neural network models fitted to source data sets. A posterior distribution fitted

to a source DNN model is used as a prior on the parameters for the target task in Wohlert et al. (2018), and the model is trained using mean field variational Bayes (for a reference on variational Bayes, see Zhang et al. 2017). Boosting approaches to transfer learning are considered by Freund and Schapire (1999), Dai et al. (2007), and Desautels et al. (2017). In Abba et al. (2023a), a penalized complexity prior between the source and target tasks is considered. While uncertainty quantification for predictions in transfer learning applications is mostly absent in the literature, approximate inference from Bayesian neural networks is used in Roy et al. (2022) to quantify uncertainty in parameter estimates and predictions to account for misaligned feature distributions. This approach, referred to as U-SFAN, is related to our RECaST framework in that it is source-free, but it requires that the source model is a neural network. Another difference is that U-SFAN focuses on using uncertainty in the source domain to guide uncertainty quantification in the target model, whereas RECaST provides uncertainty quantification directly based on the target predictions themselves.

It is important to note the difference between source-free transfer learning methods and “source-free domain adaption” (SFDA) methods: RECaST aims to use labels in the target domain in tandem with a model built in the source domain to learn about the target domain. SFDA methods, in contrast, have neither access to source data nor target labels, and often proceed by learning pseudo-labels for the target data. A comprehensive survey of SFDA approaches is given in Li et al. (2024). These surveyed strategies are predominantly non-model based, purely empirical, and lack a unified underlying framework. Moreover, those that focus on fine-tuning pre-trained neural network models on a target data set require the source model to be a neural network, and often fail to provide crucial uncertainty quantification.

Recently, there have been efforts to investigate theoretical properties of transfer learning approaches. For instance, a learning method based on LASSO for high-dimensional penalized linear regression is considered in Li et al. (2022), while diminishing the effect of *negative transfer*. Negative transfer occurs when including source data negatively impacts the performance on target data. In a similar setting, asymptotically valid confidence intervals for generalized linear model (GLM) parameters in high-dimensional transfer learning problems are established in Tian and Feng (2022). This technique is adapted to a more complicated federated transfer learning setting in Li et al. (2023). A parameter is defined in Cai and Wei (2021) to calculate an “effective sample size” to quantify total amount of information that can be transferred when the source and target conditional distributions differ. This approach is extended in Reeve et al. (2021), where assumptions are relaxed on the relationship between the source and target conditional distributions. Hector and Martin (2024) propose and study the inferential properties of an information-driven shrinkage estimator that is robust to heterogeneity between source and target feature-to-label mappings but assumes this mapping is of the same parametric

form. These methods offer more mathematically rigorous motivations, but are restrictive in their modeling options. Such restrictions are eliminated in our proposed framework.

2.3 RECaST Framework

Our transfer learning problem is defined by the following four assumptions: (i) there is a well-developed structural component of the prediction model for the source domain denoted by $f(\boldsymbol{\theta}_S, \mathbf{x}_S)$ which represents the relationship between the features and parameters; (ii) there exist ample source data for estimating the parameter(s) $\boldsymbol{\theta}_S$; (iii) $\mathcal{X}_S = \mathcal{X}_T$, and the structural component of the target prediction model, denoted by $g(\boldsymbol{\theta}_T, \mathbf{x}_T)$, is believed to be *similar* to $f(\boldsymbol{\theta}_S, \mathbf{x}_T)$; and (iv) there does not exist sufficient target data for reliably estimating the parameter(s) $\boldsymbol{\theta}_T$. We hereafter refer to $f(\boldsymbol{\theta}_S, \mathbf{x}_S)$ and $g(\boldsymbol{\theta}_T, \mathbf{x}_T)$ as *structural components* of their respective models. The notion of *similarity* will be defined in the construction of our RECaST framework for transfer learning, presented next.

Denote the forward data-generating representations of $P(y_S | \mathbf{x}_S)$ and $P(y_T | \mathbf{x}_T)$, respectively, by

$$\begin{aligned} Y_S &= h\{f(\boldsymbol{\theta}_S, \mathbf{x}_S), U_S\} \text{ and} \\ Y_T &= h\{g(\boldsymbol{\theta}_T, \mathbf{x}_T), U_T\}, \end{aligned} \tag{2.1}$$

where $\mathcal{X}_S = \mathcal{X}_T = \mathbb{R}^p$, and U_T and U_S are independent and identically distributed auxiliary random variables. We give two examples of the h function (for continuous and binary response examples), but the h function is much more general. It is to be understood as any scalar-valued function that relates the covariates to the auxiliary random variable in the fashion of a data generating equation. In fact, in the case of a continuous random variable, the h function can be taken to be the inverse cumulative distribution function, by the probability integral transform. For example, if

$$\begin{aligned} f(\boldsymbol{\theta}_S, \mathbf{x}_S) &= \mathbf{x}_S^\top \boldsymbol{\theta}_S, \\ h(\mathbf{x}_S^\top \boldsymbol{\theta}_S, U_S) &= \mathbf{x}_S^\top \boldsymbol{\theta}_S + U_S, \text{ and} \\ U_S &\sim \mathcal{N}(0, 1), \end{aligned}$$

then $Y_S \sim \mathcal{N}(\mathbf{x}_S^\top \boldsymbol{\theta}_S, 1)$. Or in the case of binary outcome data, for example, if

$$\begin{aligned} f(\boldsymbol{\theta}_S, \mathbf{x}_S) &= \text{expit}(\mathbf{x}_S^\top \boldsymbol{\theta}_S), \\ h(\mathbf{x}_S^\top \boldsymbol{\theta}_S, U_S) &= \mathbf{1}\{U_S < \text{expit}(\mathbf{x}_S^\top \boldsymbol{\theta}_S)\}, \text{ and} \\ U_S &\sim \text{Uniform}(0, 1), \end{aligned}$$

then $Y_S \sim \text{Bernoulli}\{\text{expit}(\mathbf{x}_S^\top \boldsymbol{\theta}_S)\}$, where $\text{expit}(z) := e^z / (1 + e^z)$. The *similarity* between the source and target that makes this a formulation of a transfer learning problem is determined by how well the structural component $f(\boldsymbol{\theta}_S, \mathbf{x}_T)$ of the source model approximates the structural component $g(\boldsymbol{\theta}_T, \mathbf{x}_T)$ of the target model.

Accordingly, transfer learning should be effective if $\beta := g(\boldsymbol{\theta}_T, \mathbf{x}_T) / f(\boldsymbol{\theta}_S, \mathbf{x}_T) \approx 1$, and sufficient source data is available for reliable estimation of $\boldsymbol{\theta}_S$; in fact, the source and target models are identical if $\beta = 1$. Assuming $f(\boldsymbol{\theta}_S, \mathbf{x}_T) \neq 0$ *almost surely* (a.s.), it follows a.s. that

$$Y_{T,i} = h\{\beta_i \cdot f(\boldsymbol{\theta}_S, \mathbf{x}_{T,i}), U_{T,i}\}, \quad (2.2)$$

for $i \in \{1, \dots, n_T\}$, where $Y_{T,1}, \dots, Y_{T,n_T}$ is an independent sample of n_T target labels with associated features $\mathbf{x}_{T,1}, \dots, \mathbf{x}_{T,n_T}$, and $\beta_i := g(\boldsymbol{\theta}_T, \mathbf{x}_{T,i}) / f(\boldsymbol{\theta}_S, \mathbf{x}_{T,i})$. The identity given by Equation (2.2) is further motivated by the fact that, for first-order approximations of the source and target models, if we assume $\mathbf{x}_{T,1}, \dots, \mathbf{x}_{T,n_T} \stackrel{\text{iid}}{\sim} \mathcal{N}_p(\mathbf{0}, \mathbf{I}_p)$, then by Lemma 1 (a well-known result for which we provide a proof in Appendix A.1, for convenience), $\beta_i = (\mathbf{x}_{T,i}^\top \boldsymbol{\theta}_T) / (\mathbf{x}_{T,i}^\top \boldsymbol{\theta}_S) \sim \text{Cauchy}(\delta, \gamma)$, with

$$\begin{aligned} \delta &= \frac{\boldsymbol{\theta}_T^\top \boldsymbol{\theta}_S}{\|\boldsymbol{\theta}_S\|^2}, \text{ and} \\ \gamma &= \frac{1}{\|\boldsymbol{\theta}_S\|^2} \sqrt{\|\boldsymbol{\theta}_S\|^2 \|\boldsymbol{\theta}_T\|^2 - (\boldsymbol{\theta}_T^\top \boldsymbol{\theta}_S)^2}. \end{aligned}$$

Lemma 1. *For any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$, if $\mathbf{x} \sim \mathcal{N}_p(\mathbf{0}, \mathbf{I}_p)$ then $(\mathbf{x}^\top \mathbf{a}) / (\mathbf{x}^\top \mathbf{b}) \sim \text{Cauchy}(\delta, \gamma)$, with $\delta = \mathbf{a}^\top \mathbf{b} / \|\mathbf{b}\|^2$ and $\gamma = \|\mathbf{b}\|^{-2} \sqrt{\|\mathbf{b}\|^2 \|\mathbf{a}\|^2 - (\mathbf{a}^\top \mathbf{b})^2}$.*

That being so, while Equation (2.2) is motivated by a first-order approximation, $f(\boldsymbol{\theta}_S, \mathbf{x}_S)$ and $g(\boldsymbol{\theta}_T, \mathbf{x}_T)$ need *not* share the same structure to implement the RECaST framework described by Equation (2.2). In fact, Equation (2.2) does not make any account of $g(\boldsymbol{\theta}_T, \mathbf{x}_T)$; it only assumes that the source model and parameters are available with the target data.

In practice, we assume without loss of generality that features have been centered and scaled to have mean zero and unit variance. Central limit theory supports the Gaussian approximation for more complex, nonlinear models (i.e., for the large p scenarios that characterize modern

ML approaches). Specifically, appealing to the Lyapunov or Lindeberg central limit theorem gives Gaussian approximations for the distributions of $\mathbf{x}_{T,i}^\top \boldsymbol{\theta}_S / \|\boldsymbol{\theta}_S\|^2$ and $\mathbf{x}_{T,i}^\top \boldsymbol{\theta}_T / \|\boldsymbol{\theta}_T\|^2$. For more general assumptions on f and g , first-order approximations motivate $f(\boldsymbol{\theta}_S, \mathbf{x}_{T,i}) \approx \mathbf{x}_{T,i}^\top \boldsymbol{\theta}_S$ and $g(\boldsymbol{\theta}_T, \mathbf{x}_{T,i}) \approx \mathbf{x}_{T,i}^\top \boldsymbol{\theta}_T$. The edge case with $\gamma \rightarrow \infty$ describes a situation in which there is no link between the source and target domains. Assuming $\gamma < \infty$, the RECaST model specified by Equation (2.2) with random effect $\beta_i \sim \text{Cauchy}(\delta, \gamma)$ fully characterizes the *similarity* between the source and target domains. In addition to being the exact distribution in the linear model case with Gaussian features, the Cauchy distribution also provides benefit through its heavy tails. This attribute allows β_i to capture large disparities between source and target data sets, improving the frequentist coverage of resulting prediction sets.

Estimating parameters of Cauchy distributions is a notoriously difficult problem since the heavy tails allow outlying events to happen with relatively high probability (Schuster 2012). Some estimation procedures focus on estimating solely the location parameter (Zhang 2010) or the scale parameter (Kravchuk and Pollett 2012), but rarely both. Fegyverneki (2013) explores the trade-off between using simple robust estimators, for both parameters, which are less asymptotically efficient than the maximum likelihood estimators. Recently, limit theorems are established in Akaoka et al. (2022) for quasi-arithmetic means for point estimation in cases where the strong law of large numbers fails, such as with Cauchy random variables. The fact that the Cauchy distribution appears in our work speaks to the difficulty of a transfer learning problem.

There are three primary advantages of our RECaST transfer learning model formulation in Equation (2.2) with random effect $\beta_i \sim \text{Cauchy}(\delta, \gamma)$. First, regardless of the complexity of the source model (e.g., $f(\boldsymbol{\theta}_S, \cdot)$ could represent a DNN with millions of parameters in $\boldsymbol{\theta}_S$ trained on extensive source data), RECaST only ever requires estimation of the parameters δ and γ , and perhaps a scale parameter associated with $U_{T,i}$ through $h(\cdot, U_{T,i})$. Existing transfer learning methods require either estimation of $\boldsymbol{\theta}_T$ (often via fine-tuning from an estimate of $\boldsymbol{\theta}_S$) or learning of $n_T + n_S$ weights for pooling the source and target data, where n_S is the number of source training labels. The scalability of our approach cannot be overstated. Second, RECaST needs no source data, only requiring the estimated source parameters $\hat{\boldsymbol{\theta}}_S$. Such a feature is vital in applications such as with medical data where privacy constraints place legal and ethical barriers to accessing certain data sets. Third, RECaST naturally facilitates uncertainty quantification of target label predictions via the construction of prediction sets. The following two sections propose a Bayesian framework for estimation of the posterior predictive distribution of target labels in the continuous and binary response settings, respectively.

2.4 Continuous Response Data

2.4.1 Model and Estimation

Assume that $Y_{S,1}, \dots, Y_{S,n_S}$ and $Y_{T,1}, \dots, Y_{T,n_T}$ are mutually independent, continuous random variables generated according to source and target models, respectively, as expressed in Equation (2.1). Also assume that an estimator $f(\hat{\boldsymbol{\theta}}_S, \mathbf{x})$ is available for any feature vector $\mathbf{x} \in \mathcal{X}_S = \mathcal{X}_T$, where $\hat{\boldsymbol{\theta}}_S$ is an estimator of $\boldsymbol{\theta}_S$ based on $Y_{S,1}, \dots, Y_{S,n_S}$. In the continuous response setting, a natural choice for the h function in the RECaST model, defined by Equation (2.2), is the Gaussian innovation formulation,

$$Y_{T,i} = \beta_i \cdot f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i}) + \sigma \cdot U_{T,i},$$

independently for $i \in \{1, \dots, n_T\}$, where $U_{T,i} \sim \mathcal{N}(0, 1)$, $\sigma > 0$ is a scaling parameter to be learned from the target data, and $\beta_i \sim \text{Cauchy}(\delta, \gamma)$.

We specify a canonical prior on (δ, γ, σ) as

$$\pi(\delta, \gamma, \sigma) = \mathcal{N}(\delta \mid 1, \sigma_\delta^2) \cdot \log \mathcal{N}(\gamma \mid a, b) \cdot \log \mathcal{N}(\sigma \mid c, d).$$

The prior distributions are standard for shape and scale parameters. The hyperparameters for σ can be chosen based on prior information about the target domain. The hyperparameters for δ and γ can be chosen based on prior information about similarity between the source and target data. If the domains are known to be very similar, then the prior on δ may be centered near 1 with a small variance and the prior on γ may be chosen to have a mode near 0 with a small variance. This will result in a prior favoring δ and γ values that encourage $\beta = g(\boldsymbol{\theta}_T, \mathbf{x}_T) / f(\boldsymbol{\theta}_S, \mathbf{x}_T)$ values of 1, which indicates a similar source and target. In practice, to demonstrate the robustness of the RECaST framework and to cover a broad range of transfer learning settings, we choose hyperparameter values that induce diffuse priors. See Appendix A.3 for more details.

A posterior distribution of the parameters (δ, γ, σ) can be expressed as

$$\begin{aligned}
& \pi(\delta, \gamma, \sigma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S) \\
&= \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} \pi(\delta, \gamma, \sigma, \beta_1, \dots, \beta_{n_T} \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S) d\beta_1 \dots d\beta_{n_T} \\
&\propto \pi(\delta, \gamma, \sigma) \cdot \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} \prod_{i=1}^{n_T} [\mathcal{N}\{y_{T,i} \mid \beta_i f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i}), \sigma^2\} \cdot \text{Cauchy}(\beta_i \mid \delta, \gamma)] d\beta_1 \dots d\beta_{n_T} \\
&= \pi(\delta, \gamma, \sigma) \cdot \prod_{i=1}^{n_T} \int_{\mathbb{R}} \mathcal{N}\{y_{T,i} \mid \beta_i f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i}), \sigma^2\} \cdot \text{Cauchy}(\beta_i \mid \delta, \gamma) d\beta_i \\
&= \pi(\delta, \gamma, \sigma) \cdot \prod_{i=1}^{n_T} \int_{\mathbb{R}} \mathcal{N}\{\beta_i f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i}) \mid y_{T,i}, \sigma^2\} \cdot \text{Cauchy}(\beta_i \mid \delta, \gamma) d\beta_i \\
&= \pi(\delta, \gamma, \sigma) \cdot \prod_{i=1}^{n_T} \int_{\mathbb{R}} \mathcal{N}\left\{\beta_i \mid \frac{y_{T,i}}{f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})}, \frac{\sigma^2}{f^2(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})}\right\} \cdot \frac{\text{Cauchy}(\beta_i \mid \delta, \gamma)}{|f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})|} d\beta_i, \tag{2.3}
\end{aligned}$$

where the univariate integrals in the last expression can be evaluated numerically. Next, the posterior predictive distribution of the label \tilde{Y}_T associated with some new target feature vector $\tilde{\mathbf{x}}_T$ can be derived as the marginal distribution of

$$\begin{aligned}
& \pi(\tilde{y}_T, \tilde{\boldsymbol{\beta}}, \sigma, \delta, \gamma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S) \\
&= \mathcal{N}\{\tilde{y}_T \mid \tilde{\boldsymbol{\beta}} f(\hat{\boldsymbol{\theta}}_S, \tilde{\mathbf{x}}_T), \sigma^2\} \cdot \pi(\tilde{\boldsymbol{\beta}}, \sigma, \delta, \gamma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S) \\
&= \mathcal{N}\{\tilde{y}_T \mid \tilde{\boldsymbol{\beta}} f(\hat{\boldsymbol{\theta}}_S, \tilde{\mathbf{x}}_T), \sigma^2\} \cdot \text{Cauchy}(\tilde{\boldsymbol{\beta}} \mid \delta, \gamma) \cdot \pi(\delta, \gamma, \sigma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S). \tag{2.4}
\end{aligned}$$

2.4.2 Remarks on Implementation

To estimate the posterior distribution given in Equation (2.3), we implement a random walk Metropolis-Hastings algorithm, numerically solving the univariate integrals with the Julia package QuadGK (Johnson 2013). Furthermore, by expressing these integrals as expectations with respect to a Gaussian distribution (i.e., the final expression in Equation (2.3)), we show that they are numerically equivalent to definite integrals from -39 to 39 . See Appendix A.2 for the mathematical details of this bound. This substantially reduces the computational overhead for the numerical integration.

We detail our implementation of the Metropolis-Hastings algorithm in Appendix A.3. The chosen number of iterations and length of the burn-in period can be adjusted based on computational resources. Because Metropolis-Hastings evaluates the likelihood for all target data points for each iteration, the computational complexity is $\mathcal{O}(n_T \cdot n_{\text{iterations}})$, with $n_{\text{iterations}}$ the number of Metropolis-Hastings iterations. The fact that n_T is assumed to be small for trans-

for learning problems mitigates concerns about scalability. Posterior predictive credible sets can be constructed as usual in Bayesian inference, from the highest posterior density regions calculated via the empirical quantiles of the sampled posterior predictive values.

In Algorithm 1, we propose a procedure for drawing samples from the posterior predictive distribution described by Equation (2.4). Again take $\tilde{\mathbf{x}}_T$ to be the feature vector for a new target data point with label \tilde{Y}_T . With the learned posterior distribution of (δ, γ, σ) , we are able to sample from the posterior predictive distribution of \tilde{Y}_T . We first sample n_{post} (δ, γ, σ) triplets from the posterior distribution. For *each* of these triplets, we sample n_β β 's from a Cauchy distribution with location and scale parameters corresponding to the δ and γ sampled from the posterior. Finally, for *each* sampled β we sample n_Y \tilde{y}_T 's from the normal distribution with mean and variance determined by $\tilde{\mathbf{x}}_T$, the sampled β , and the sampled σ . This gives a total of $n_{\text{post}} \cdot n_\beta \cdot n_Y$ samples from the posterior predictive distribution for each new target observation. These samples are used to construct the posterior predictive credible sets as described in Algorithm 1 with a computational complexity of $\mathcal{O}(n_{\text{post}} \cdot n_\beta \cdot n_Y)$. We discuss our choices for these parameters in Appendix A.3. We showcase the effectiveness of these proposed computational strategies in a variety of simulation scenarios in Section 2.6.2.

Algorithm 1 RECaST posterior predictive sampling: continuous response data

Input: $\tilde{\mathbf{x}}_T$, samples from $\pi(\delta, \gamma, \sigma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S)$, and sample sizes n_{post} , n_β , and n_Y

Output: A sample of values from $\pi(\tilde{y}_T \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S)$

```

for  $i \leftarrow 1$  to  $n_{\text{post}}$  do
     $\delta, \gamma, \sigma \leftarrow \text{random}\{\pi(\delta, \gamma, \sigma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S)\}$ 
    for  $j \leftarrow 1$  to  $n_\beta$  do
         $\tilde{\beta} \leftarrow \text{random}\{\text{Cauchy}(\delta, \gamma)\}$ 
        for  $k \leftarrow 1$  to  $n_Y$  do
             $\tilde{Y}_T \leftarrow \text{random}[\mathcal{N}\{\tilde{\beta}f(\hat{\boldsymbol{\theta}}_S, \tilde{\mathbf{x}}_T), \sigma^2\}]$ 
        end for
    end for
end for

```

2.4.3 Theoretical Guarantees

In this section, we establish the asymptotic validity of our proposed posterior predictive credible sets in the case of linear source and target models with independent Gaussian innovations. Here, asymptotic validity means that the empirical coverage of a $1 - \alpha$ level prediction credible

set attains $1 - \alpha$ level coverage, asymptotically in n_T , as described by the result of Theorem 3. Our mathematical proof of this result and of all supporting results are organized in Appendix A.1.

Suppose that $Y_{S,j}$ follows a Gaussian distribution centered at $\mathbf{x}_{S,j}^\top \boldsymbol{\theta}_S$, independently for $j \in \{1, \dots, n_S\}$. In the class of transfer learning problems we consider, it is assumed that consistent or meaningful estimators are available for all source model parameters, and that ample data/resources are available for estimating them. Accordingly, assume that n_S is sufficiently large such that $\boldsymbol{\theta}_S$ is regarded as known. Next, assume that $Y_{T,1}, \dots, Y_{T,n_T} \stackrel{iid}{\sim} \mathcal{N}(\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_T, \sigma^2)$, for some feature vector $\tilde{\mathbf{x}} \in \mathcal{X}_T = \mathcal{X}_S$, and $\boldsymbol{\theta}_T$ unknown. Leveraging the RECaST transfer learning framework, the likelihood function of (δ, γ) can be expressed as

$$L(\delta, \gamma \mid y_{T,1}, \dots, y_{T,n_T}, \beta_1, \dots, \beta_{n_T}) = \prod_{i=1}^{n_T} \left[\mathcal{N}\{y_{T,i} \mid \beta_i \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S, \sigma^2\} \cdot \text{Cauchy}(\beta_i \mid \delta, \gamma) \right]. \quad (2.5)$$

We investigate the asymptotic coverage of prediction sets constructed from the RECaST posterior predictive distribution with plugin maximum likelihood estimators (MLEs) $\hat{\delta}$ and $\hat{\gamma}$ for δ and γ , respectively:

$$\pi(\tilde{y}_T, \tilde{\boldsymbol{\beta}} \mid y_1, \dots, y_{n_T}) = \mathcal{N}(\tilde{y}_T \mid \tilde{\boldsymbol{\beta}} \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S, \sigma^2) \cdot \text{Cauchy}(\tilde{\boldsymbol{\beta}} \mid \hat{\delta}, |\hat{\gamma}|).$$

This is the same as considering maximum a posteriori (MAP) estimators for δ and γ with a flat prior $\pi(\delta, \gamma) \propto 1$, and the choice of prior is not so meaningful in the $n_T \rightarrow \infty$ setting. Recall that in the RECaST framework the $\beta_1, \dots, \beta_{n_T}$ that appear in the likelihood function in Equation (2.5) are iid $\text{Cauchy}(\delta, \gamma)$ random effects. Nonetheless, we demonstrate with Lemma 2 that the MLEs $\hat{\delta}$ and $\hat{\gamma}$ converge in probability to fixed points such that

$$\pi(\tilde{Y}_T, \tilde{\boldsymbol{\beta}} \mid y_1, \dots, y_{n_T}) \approx \mathcal{N}(\tilde{Y}_T \mid \tilde{\boldsymbol{\beta}} \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S, \sigma^2) \cdot \mathbf{1} \left\{ \tilde{\boldsymbol{\beta}} = \frac{\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_T}{\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S} \right\} = \mathcal{N}(\tilde{Y}_T \mid \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_T, \sigma^2),$$

as desired. This fact leads to our main theoretical result, Theorem 3, which establishes the asymptotic validity of $1 - \alpha$ level RECaST prediction sets of the form $[a_{n_T}^\alpha, b_{n_T}^\alpha]$, with

$$\begin{aligned} a_{n_T}^\alpha &:= \Phi^{-1}(\alpha/2) \cdot \sigma + \tilde{\boldsymbol{\beta}} \cdot \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S \quad \text{and} \\ b_{n_T}^\alpha &:= \Phi^{-1}(1 - \alpha/2) \cdot \sigma + \tilde{\boldsymbol{\beta}} \cdot \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S, \end{aligned}$$

for any $\alpha \in (0, 1)$ and $\tilde{\boldsymbol{\beta}} \sim \text{Cauchy}(\hat{\delta}, |\hat{\gamma}|)$.

Lemma 2. Assuming $Y_{T,1}, \dots, Y_{T,n_T} \stackrel{iid}{\sim} \mathcal{N}(\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_T, \sigma^2)$ and $\beta_1, \dots, \beta_{n_T} \stackrel{iid}{\sim} \text{Cauchy}(\delta, \gamma)$, indepen-

dently, the MLEs of δ and γ for Equation (2.5) satisfy

$$\widehat{\delta} \longrightarrow \frac{\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T}{\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S} \quad \text{and} \quad \widehat{\gamma} \longrightarrow 0$$

in probability as $n_T \rightarrow \infty$.

Theorem 3. Assume that $\widetilde{Y}_T \sim \mathcal{N}(\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T, \sigma^2)$. Then, for any $\alpha \in (0, 1)$,

$$P\left(\widetilde{Y}_T \in [a_{n_T}^\alpha, b_{n_T}^\alpha]\right) = \int_{a_{n_T}^\alpha}^{b_{n_T}^\alpha} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(\widetilde{y}_T - \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T)^2} d\widetilde{y}_T \longrightarrow 1 - \alpha$$

in probability as $n_T \rightarrow \infty$.

In Section 2.6, we provide empirical evidence that RECaST achieves near nominal coverage even in more practical, small n_T settings, trained on target data that arise from both linear and non-linear models. In the empirical investigations in Section 2.6, we relax the assumptions of known σ and the availability of repeated samples from a fixed feature vector $\widetilde{\mathbf{x}}$.

2.5 Binary Response Data

2.5.1 Model and Estimation

Assume that $Y_{S,1}, \dots, Y_{S,n_S}$ and $Y_{T,1}, \dots, Y_{T,n_T}$ are mutually independent, Bernoulli random variables generated according to source and target models, respectively, as expressed in Equation (2.1). Also assume that an estimator $f(\widehat{\boldsymbol{\theta}}_S, \mathbf{x})$ is available for any feature vector $\mathbf{x} \in \mathcal{X}_S = \mathcal{X}_T$, where $\widehat{\boldsymbol{\theta}}_S$ is an estimator of $\boldsymbol{\theta}_S$ based on $Y_{S,1}, \dots, Y_{S,n_S}$. In the binary response setting, a natural choice for the h function in the RECaST model, defined by Equation (2.2), is the logistic model formulation,

$$Y_{T,i} = \mathbf{1}[U_{T,i} < \text{expit}\{\beta_i \cdot f(\widehat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})\}],$$

with $U_{T,i} \sim \text{Uniform}(0, 1)$ independently for $i \in \{1, \dots, n_T\}$ and $\beta_i \sim \text{Cauchy}(\delta, \gamma)$.

As in the continuous setting, the RECaST posterior distribution of the parameters can be

constructed as

$$\begin{aligned}
& \pi(\delta, \gamma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S) \\
&= \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} \pi(\delta, \gamma, \beta_1, \dots, \beta_{n_T} \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S) d\beta_1 \dots d\beta_{n_T} \\
&\propto \pi(\delta, \gamma) \cdot \prod_{i=1}^{n_T} \int_{\mathbb{R}} \text{Bernoulli}[y_{T,i} \mid \text{expit}\{\beta_i f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})\}] \cdot \text{Cauchy}(\beta_i \mid \delta, \gamma) d\beta_i,
\end{aligned}$$

and the posterior predictive distribution of the label \tilde{Y}_T associated with some new target feature vector $\tilde{\mathbf{x}}_T$ can be derived as the marginal distribution of

$$\begin{aligned}
& \pi(\tilde{y}_T, \tilde{\beta}, \delta, \gamma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S) \\
&= \text{Bernoulli}[\tilde{y}_T \mid \text{expit}\{\tilde{\beta} f(\hat{\boldsymbol{\theta}}_S, \tilde{\mathbf{x}}_T)\}] \cdot \text{Cauchy}(\tilde{\beta} \mid \delta, \gamma) \cdot \pi(\delta, \gamma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S). \quad (2.6)
\end{aligned}$$

We specify a canonical prior on (δ, γ) as

$$\pi(\delta, \gamma) = \mathcal{N}(\delta \mid 1, \sigma_\delta^2) \cdot \log \mathcal{N}(\gamma \mid a, b),$$

with diffuse choices of the hyperparameters σ_δ, a, b . A similar description to that in Section 3.1 of the choice of priors holds here.

A $1 - \alpha$ level RECaST prediction credible set, denoted $\Gamma_{n_T}^\alpha$, for binary response values is constructed as

$$\Gamma_{n_T}^\alpha = \begin{cases} \{0\}, & \text{if } \tilde{p} < 1 - \tilde{p} \text{ and } 1 - \alpha \leq 1 - \tilde{p} \\ \{1\}, & \text{if } 1 - \tilde{p} \leq \tilde{p} \text{ and } 1 - \alpha \leq \tilde{p} \\ \{0, 1\}, & \text{else,} \end{cases} \quad (2.7)$$

where $\tilde{p} := \pi(\tilde{y}_T = 1 \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S)$.

2.5.2 Remarks on Implementation

The RECaST transfer learning computations in the binary response setting follow analogously to those described in Section 2.4.2. For completeness, Algorithm 2 specifies the procedure we propose for drawing samples from the posterior predictive distribution described by Equation (2.6).

Algorithm 2 RECaST posterior predictive sampling: binary response data

Input: $\tilde{\mathbf{x}}_T$, samples from $\pi(\delta, \gamma \mid y_{T,1}, \dots, y_{T,n_T}, \widehat{\boldsymbol{\theta}}_S)$, and sample sizes n_{post} , n_β , and n_Y

Output: A sample of values from $\pi(\tilde{y} \mid y_{T,1}, \dots, y_{T,n_T}, \widehat{\boldsymbol{\theta}}_S)$

```

for  $i \leftarrow 1$  to  $n_{\text{post}}$  do
   $\delta, \gamma \leftarrow \text{random}\{\pi(\delta, \gamma \mid y_{T,1}, \dots, y_{T,n_T}, \widehat{\boldsymbol{\theta}}_S)\}$ 
  for  $j \leftarrow 1$  to  $n_\beta$  do
     $\tilde{\beta} \leftarrow \text{random}\{\text{Cauchy}(\delta, \gamma)\}$ 
    for  $k \leftarrow 1$  to  $n_Y$  do
       $\tilde{Y}_T \leftarrow \text{random}(\text{Bernoulli}[\text{expit}\{\tilde{\beta} f(\widehat{\boldsymbol{\theta}}_S, \tilde{\mathbf{x}}_T)\}])$ 
    end for
  end for
end for

```

2.6 Simulation Study

2.6.1 Objectives and Setup

In this section, we examine the finite sample performance of RECaST through simulations on synthetic data. We consider continuous and binary responses with source models corresponding to linear (RECaST LM) and logistic (RECaST GLM) regression, respectively, as well as a DNN (RECaST DNN) source model for both response types. We assess the empirical coverage with respect to the nominal coverage level of the prediction sets. If the method is calibrated, the empirical coverage will match the nominal significance level. We use the terms *empirical coverage* and *observed coverage* interchangeably.

We generate the synthetic data from linear and logistic regressions with source parameter vector $\boldsymbol{\theta}_S$ and target parameter vector $\boldsymbol{\theta}_T$, with $p = 50$ features (including an intercept). The features are generated from the standard Gaussian distribution, $\mathbf{x}_{S,i}, \mathbf{x}_{T,j} \sim \mathcal{N}_{p-1}(\mathbf{0}, \mathbf{I}_{p-1})$. We fix the source data generating parameters $\boldsymbol{\theta}_S$. The source data generating parameters are set to $\boldsymbol{\theta}_S = (-\mathbf{a}, \mathbf{b})$ where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{25}$ have components independently sampled from $\text{Uniform}(0.75, 5)$ and then fixed for all simulations. The *similarity* of source and target domains is controlled by choosing the value of $\sigma_{\text{TL}} > 0$ in constructing $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}_p(\mathbf{0}, \sigma_{\text{TL}}^2 \mathbf{I}_p)$. We consider values of $\sigma_{\text{TL}}^2 \in \{0, 0.25, 1, 4\}$. Setting $\sigma_{\text{TL}}^2 = 0$ corresponds to $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S$, i.e., no difference between the source and target distributions. Since the source parameters lie within $[-5, -0.75] \cup [0.75, 5]$, a variance of $\sigma_{\text{TL}}^2 = 4$ allows for significant differences between $\boldsymbol{\theta}_T$ and $\boldsymbol{\theta}_S$. We fix the source sample size at $n_S = 1000$, and vary the target sample size n_T to examine performance when $p < n_T$ ($n_T = 100, 250$), p is near n_T ($n_T = 40, 60$), and $p > n_T$ ($n_T = 20$). We simulate 300 source

and target data sets for each of these 20 combinations of σ_{TL}^2 and n_T values, and implement the estimation procedures described in Sections 2.4.2 and 2.5.2. See Appendix A.3 for additional details about the specifics of our implementations.

We compare to a linear model baseline (LM) which is built only on the target data. Another baseline for comparison is constructed from training a DNN on the target data, without any transfer learning, and we compare RECaST to other state-of-the-art transfer learning approaches. We build a DNN on the source data and fine-tune the last layer on the target data (Unfreeze DNN); this is often referred to as *freezing* the weights of the source DNN and *unfreezing* the last layer. See Appendix A.4 for details on this procedure. Other state-of-the-art transfer learning approaches that we compare RECaST to include TransRF (Gu et al. 2022), a source-free method that adapts a random forest model built in the source domain to target data, and glmtrans (Tian and Feng 2022), which is based on penalized GLMs and designed to mitigate the impact of negative transfer. Unlike RECaST and TransRF, glmtrans requires the source data to be available during the training of the model. In the continuous setting, we compare to the source-free methods outlined by Tripuraneni et al. (2021). We compare to both their first order method (MTL FO) and their method of moments approach (MTL MoM). Note that while this method does not require the source data when learning the target model, it does require that the source parameters were learned following their formulation whereas RECaST is agnostic to the choice of source model. In the binary setting, we compare RECaST to the regularized logistic regression (Wiens) approach of Wiens et al. (2014). This approach uses the combined source and target EHR data to build a regularized model for disease prediction – similar to the real data application we consider in Section 2.7, but with the disadvantage that Wiens requires access to the source data (while RECaST does not). In the binary setting, we also compare RECaST to the adversarial transfer learning approach WDGRL (Shen et al. 2018), which also requires access to the source data.

Throughout this section, all DNN training proceeds by setting aside a portion of the training data to be used as a calibration data set. The final DNN parameters are chosen from the epoch with the minimum calibration loss to improve generalizability to out-of-sample test sets. Additional details/specifications for our DNN training procedures are provided in Appendix A.4.

2.6.2 Continuous Response Results

Table 2.1 and Figure 2.1 summarize the performance of the prediction uncertainty quantification provided by our RECaST framework implementations. Table 2.1 presents the empirical coverage for 95% nominal level prediction sets for each simulation setting. Recall that the em-

pirical coverage should ideally match the nominal significance for a given level; an empirical coverage greater than the nominal coverage level corresponds to a conservative interval estimate. RECaST methods consistently provide empirical coverage at or slightly above nominal levels, supporting the use of RECaST for inference on out-of-sample target domain predictions. Additionally, Figure 2.1 plots empirical versus nominal coverage for the $\sigma_{\text{TL}}^2 = 0.25$, $n_T = 100$ and $\sigma_{\text{TL}}^2 = 4$, $n_T = 20$ settings at a grid of nominal levels. The empirical coverages consistently achieve the associated nominal levels or are slightly conservative.

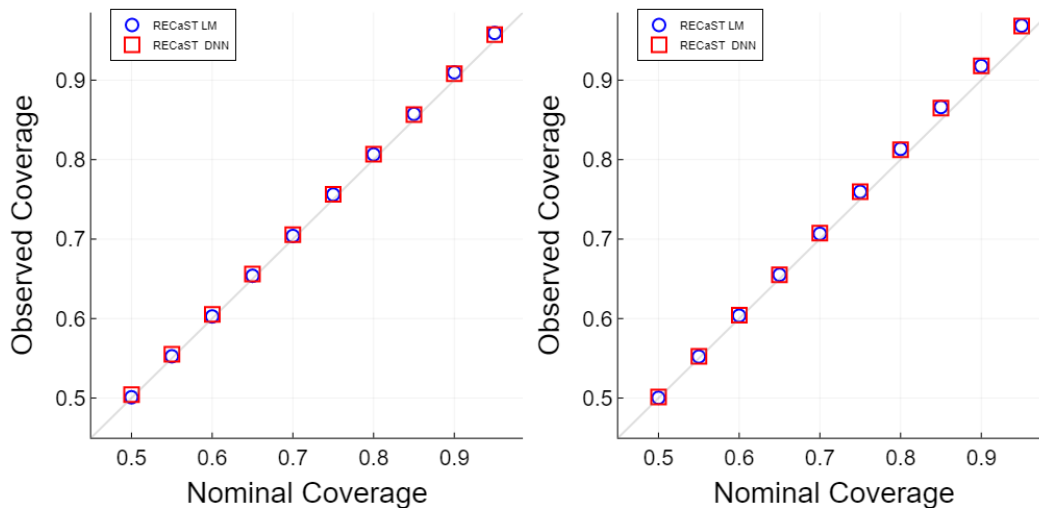


Figure 2.1: Reliability curves of the nominal coverage versus the empirical coverage, averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations. The left panel shows an easy setting: $n_T = 100$ and $\sigma_{\text{TL}}^2 = 0.25$. The right panel shows a difficult setting: $n_T = 20$ and $\sigma_{\text{TL}}^2 = 4$.

Out-of-sample root mean squared errors (RMSEs) for all methods, averaged over 300 source and target data sets are presented in Table 2.2. The LM provides the best prediction when the sample size is large since in this case it correctly specifies the data generating model and has enough data to estimate the parameters. There is a large decrease in performance, noted by the increase in RMSE, when $n_T < p$ and a generalized inverse has to be used for parameter estimation. As expected, the performance of DNN deteriorates as the target sample size decreases. Note that the baseline DNN is overparameterized, which leads to it having higher RMSEs than the baseline LM.

Interestingly, the RECaST RMSE values remain consistent for each value of σ_{TL}^2 , regardless of sample size, suggesting that RECaST is appropriate even when the target sample size is so small as to preclude a target-only analysis. Meanwhile, Unfreeze DNN exhibits an increase

in RMSE for each value of σ_{TL}^2 as n_T decreases. As source and target become more dissimilar, both Unfreeze DNN and RECaST exhibit similar increases in RMSE. In fact, with $n_T = 250$ and $\sigma_{\text{TL}}^2 = 4$, the target-only DNN outperforms both RECaST methods. This setting is the most prone to negative transfer: the target sample size is large enough to learn meaningful DNN parameters, *and* the source and target data distributions differ greatly, making transfer difficult. We see this phenomenon with the target only LM as well; with a sample size of $n_T = 40$, both RECaST methods outperform the LM except for when the source and target are most dissimilar. When $n_T = 20$, the RECaST methods outperform the LM in all settings. This highlights a situation where transfer learning is necessary because the target domain lacks sufficient data to efficiently estimate the target parameters, even with a correctly specified model.

Table 2.1: Empirical coverage (standard error) at the 95% nominal level, averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	σ_{TL}^2	RECaST LM	RECaST DNN
250	0	96(1.8)	94(1.9)
	0.25	95(1.9)	95(1.9)
	1	95(1.9)	95(1.8)
	4	95(2.0)	95(1.9)
100	0	96(1.8)	94(2.1)
	0.25	96(1.8)	96(2.0)
	1	96(1.8)	96(1.8)
	4	96(1.8)	96(1.9)
60	0	97(2.2)	94(2.6)
	0.25	96(1.9)	96(1.8)
	1	96(1.8)	96(1.8)
	4	96(1.8)	96(1.8)
40	0	97(2.1)	94(3.3)
	0.25	96(2.4)	96(2.4)
	1	96(2.6)	96(2.5)
	4	96(2.8)	96(2.8)
20	0	98(1.8)	95(3.0)
	0.25	97(2.6)	97(2.8)
	1	97(2.6)	97(2.8)
	4	97(2.7)	97(2.9)

The MTL FO and MTL MoM both see increases in RMSE as the source and target become more dissimilar and see a larger increase in RMSE as the target sample size decreases. Interestingly, in this simulation these two methods have the same performance when there are more target sample points than there are features. While in some settings with larger target sample sizes the Unfreeze DNN slightly outperforms RECaST, it has larger standard errors and fails to provide uncertainty quantification. We find that TransRF sometimes performs well but with high RMSE variance. We were not able to evaluate TransRF when the target sample size was 20 as the software gave NA values instead of predictions without an accompanying error message. While glmtrans sometimes has smaller RMSE than RECaST, recall that it requires access to the source data and that only RECaST provides uncertainty quantification for predictions.

Table 2.2: Out-of-sample RMSE (standard error) averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations.

n_T	σ_{TL}^2	LM	DNN	RECaST LM	RECaST DNN	Unfreeze DNN	TransRF	glmtrans	MTL FO	MTL MoM
250	0	0.57(0.03)	2.8(0.38)	0.52(0.027)	1.2(0.090)	0.58(0.038)	14(1.5)	0.56(0.026)	1.9(0.07)	1.9(0.07)
	0.25	0.57(0.03)	2.9(0.37)	3.6(0.43)	3.8(0.4)	2.8(0.42)	14(1.4)	0.56(0.027)	1.9(0.5)	1.9(0.5)
	1	0.57(0.03)	3.1(0.43)	7.1(0.86)	7.2(0.84)	5.5(0.90)	14(1.5)	0.56(0.027)	2.0(1.0)	2.0(1.0)
	4	0.57(0.03)	3.7(0.52)	14(1.7)	14(1.7)	11(1.8)	17(2.6)	0.56(0.027)	2.5(1.7)	2.5(1.7)
100	0	0.71(0.07)	8.9(1.6)	0.52(0.022)	1.2(0.095)	0.81(0.095)	22(12)	0.69(0.047)	2.4(0.2)	2.4(0.2)
	0.25	0.71(0.06)	9.1(1.3)	3.6(0.42)	3.8(0.40)	3.2(0.57)	28(75)	0.73(0.068)	2.4(0.7)	2.4(0.7)
	1	0.71(0.06)	9.4(1.3)	7.1(0.85)	7.2(0.83)	6.3(1.1)	23(16)	0.74(0.075)	2.5(1.2)	2.5(1.2)
	4	0.71(0.06)	11(1.52)	14(1.7)	14(1.7)	13(2.1)	31(34)	0.74(0.073)	3.1(2.1)	3.1(2.1)
60	0	1.3(0.27)	14(2.5)	0.52(0.025)	1.2(0.11)	1.5(0.29)	46(73)	0.75(0.05)	4.0(0.87)	4.0(0.88)
	0.25	1.3(0.23)	13(1.6)	3.6(0.43)	3.8(0.42)	3.7(0.78)	48(99)	1.2(0.21)	4.0(1.2)	4.0(1.2)
	1	1.3(0.23)	14(1.8)	7.1(0.87)	7.2(0.86)	6.8(1.3)	54(85)	1.7(0.38)	4.2(2.1)	4.2(2.1)
	4	1.3(0.23)	16(2.6)	14(1.8)	14(1.8)	13(2.0)	51(59)	3.0(0.85)	5.1(3.5)	5.1(3.5)
40	0	10(2.1)	17(2.6)	0.52(0.024)	1.2(0.089)	1.8(0.61)	74(64)	0.78(0.063)	11(2.1)	10(1.9)
	0.25	11(1.8)	17(2.4)	3.6(0.41)	3.8(0.40)	4.1(1.1)	62(74)	2.5(0.51)	11(2.1)	10(1.9)
	1	11(2.0)	18(2.5)	7.2(0.83)	7.3(0.83)	7.6(2.2)	69(110)	4.7(1.1)	11(2.4)	11(2.2)
	4	12(2.0)	20(2.9)	14(1.7)	14(1.7)	14(3.0)	150(540)	8.9(2.3)	13(3.0)	13(2.9)
20	0	18(1.8)	21(1.8)	0.54(0.03)	1.2(0.11)	2.5(2.2)	-	0.81(0.078)	18(2.0)	17(1.6)
	0.25	18(1.8)	21(1.8)	3.7(0.44)	3.9(0.42)	4.7(2.5)	-	3.4(0.39)	18(1.9)	18(1.7)
	1	18(1.9)	22(2.0)	7.3(0.90)	7.4(0.90)	8.5(3.5)	-	6.7(0.8)	19(2.1)	18(1.9)
	4	21(2.4)	24(2.7)	15(1.8)	15(1.8)	16(4.0)	-	6.7(0.8)	21(2.5)	20(2.4)

2.6.3 Binary Response Results

Table 2.3 shows that RECaST procedures, again, provide near nominal coverages with low standard errors across sample sizes in the binary response setting.

Table 2.3: Empirical coverage (standard error) at the 95% nominal level, averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	σ_{TL}^2	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	0	84(9.5)	95(0.78)	96(0.090)	100(0)	91(8.4)	89(12)	98(4.4)	95(4.4)
	0.25	89(7.5)	95(0.82)	96(0.13)	100(0)	93(6.8)	88(11)	98(2.6)	94(4.8)
	1	91(6.2)	95(0.65)	96(0.14)	100(0)	95(4.5)	87(9.5)	98(3.6)	86(6.0)
	4	93(6.4)	95(0.40)	95(0.39)	99(1.5)	95(4.8)	86(11)	97(4.1)	75(7.6)
100	0	80(12)	96(1.1)	96(0.25)	100(0)	90(8.7)	68(22)	96(6.4)	96(3.4)
	0.25	83(11)	95(1.3)	96(0.27)	100(0)	92(7.8)	78(3.0)	94(6.2)	93(4.2)
	1	88(8.2)	95(1.2)	96(0.35)	100(4.6)	94(5.9)	69(18)	94(8.5)	89(5.7)
	4	92(6.2)	95(0.81)	95(0.90)	95(13)	94(10.0)	49(20)	83(4.6)	89(2.3)
60	0	80(13)	95(1.2)	96(0.54)	100(0.0)	92(6.4)	65(19)	93(9.3)	95(3.7)
	0.25	77(17)	95(1.3)	96(0.67)	100(0.0)	91(8.1)	64(22)	88(12)	95(3.1)
	1	80(21)	95(1.1)	95(0.867)	100(0.49)	94(6.1)	63(19)	88(15)	90(4.7)
	4	84(15)	95(0.59)	95(1.0)	96.8(4.7)	93(8.7)	58(19)	89(11)	80(6.9)
40	0	68(23)	95(1.6)	96(0.86)	100(0.0)	89(11)	60(20)	88(14)	95(5.5)
	0.25	72(20)	95(1.6)	96(0.99)	100(0.0)	90(7.9)	55(25)	81(16)	95(4.1)
	1	76(19)	94(1.5)	95(1.2)	100(0.53)	89(8.7)	59(19)	85(14)	89(5.9)
	4	77(25)	94(1.4)	94(1.1)	97(3.2)	90(7.2)	63(20)	78(14)	78(7.4)
20	0	67(22)	95(1.1)	96(0.78)	100(0.0)	85(17)	-	86(15)	-
	0.25	75(16)	95(1.1)	95(0.98)	100(0.0)	86(14)	-	68(15)	-
	1	75(16)	95(0.84)	95(1.1)	100(0.55)	86(17)	-	63(18)	-
	4	72(13)	95(0.47)	94(0.99)	98(1.5)	80(18)	-	66(17)	-

Compared to the other approaches, RECaST provides substantial inferential advantages that are robust to small target sample sizes and large dissimilarity between source and target. Recall from Equation (2.7) that prediction sets in the binary response setting are determined entirely by the Bernoulli probability of observing label 1. Thus, we can construct prediction sets for the DNN, Unfreeze DNN, and Wiens methods, as well. When a method fails to discriminate between the two labels at level $1 - \alpha$ (e.g., when the Bernoulli probability of success and failure are *both below* $1 - \alpha$), then the prediction set must include both labels to attain the $1 - \alpha$ level. In such cases, as observed for the Wiens method in various settings in Table 2.3, the prediction set achieves 100% empirical coverage, but is unhelpful for prediction.

Table 2.4 provides the area under the receiver operator characteristic curve (AUC) for all methods and simulation settings. In all settings except one, RECaST DNN outperforms all other methods. We see similar patterns here as in the continuous setting. The RECaST models consistently report the highest AUC, with low standard errors across sample sizes. In contrast, the AUC of DNN and Unfreeze DNN drastically declines as n_T decreases. As expected, the AUC of all transfer learning methods decreases as the difficulty of the problem increases with larger values of σ_{TL}^2 . RECaST DNN and WDGRL frequently outperform other methods; however, WDGRL requires access to the source data, an important limitation that is unrealistic in many

applications. WDGRL crashed with a sample size of $n_T = 20$, so we are unable to evaluate its performance in these settings.

The benefits to coverage properties and predictive performance of the RECaST method are especially important in the binary response case. This demonstrates that RECaST can be used even when the linearity assumption of Lemma 1 is violated.

Table 2.4: Out-of-sample AUC (standard error) averaged over 300 source and target data sets for each setting; the out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	σ_{1L}^2	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	0	95(1.7)	98(2.1)	98(0.61)	80(3.5)	97(1.2)	66(13)	97(2.0)	97(0.95)
	0.25	95(1.6)	97(2.3)	98(0.89)	80(3.9)	97(1.2)	69(10)	96(1.6)	97(1.3)
	1	94(1.5)	93(3.8)	96(1.5)	79(3.9)	95(1.7)	66(11)	97(1.8)	95(1.5)
	4	95(1.7)	84(5.5)	89(2.8)	76(4.0)	89(3.0)	67(12)	97(1.6)	88(3.2)
100	0	85(7.9)	96(2.2)	98(0.64)	81(4.2)	96(2.2)	47(19)	87(9.3)	98(0.66)
	0.25	83(9.6)	95(2.7)	97(1.0)	80(4.1)	95(1.9)	18(14)	81(5.3)	97(1.0)
	1	84(8.4)	92(3.8)	95(1.4)	79(4.4)	93(2.4)	48(20)	81(5.5)	95(1.4)
	4	82(11)	83(4.7)	89(3.1)	74(4.3)	87(4.8)	49(20)	83(4.6)	89(2.3)
60	0	72(13)	96(1.9)	98(1.0)	80(4.3)	94(5.2)	49(20)	83(4.6)	89(2.3)
	0.25	74(11)	94(2.5)	97(1.4)	80(4.3)	94(2.34)	36(21)	74(5.1)	97(0.78)
	1	75(10)	90(3.5)	95(1.7)	78(4.1)	91(6.0)	33(20)	74(5.3)	95(1.9)
	4	72(11)	83(4.0)	89(3.3)	73(4.7)	84(8.0)	29(18)	75(5.6)	89(2.4)
40	0	68(11)	96(1.6)	98(1.1)	80(3.8)	94(4.5)	27(16)	83(16)	97(1.1)
	0.25	68(11)	94(2.2)	97(1.3)	80(4.0)	92(6.7)	19(15)	67(4.9)	97(1.2)
	1	65(12)	90(3.0)	95(1.9)	78(3.9)	89(7.9)	32(18)	69(5.7)	95(1.7)
	4	67(12)	82(4.1)	89(3.5)	74(4.2)	80(12)	31(18)	69(4.9)	89(3.2)
20	0	60(8.7)	96(1.7)	97(1.4)	80(3.9)	89(10)	-	81(18)	-
	0.25	61(9.1)	94(2.1)	97(1.9)	79(4.2)	87(13)	-	62(5.1)	-
	1	60(9.5)	90(2.7)	94(2.5)	77(4.5)	83(14)	-	60(5.0)	-
	4	62(8.2)	82(3.5)	88(3.0)	72(5.0)	77(11)	-	63(5.0)	-

2.6.4 Robustness of RECaST

Over-Parameterized RECaST DNN

In all previous simulations, the true data generating mechanisms are linear or logistic models. To test the robustness of RECaST, we now consider a more complex case where data are generated from neural networks. We generate data from a neural network with a densely connected input layer of size $\ell_1 = (p, 10)$ and then pass through a ReLU activation function to an output layer of size $\ell_2 = (10, 1)$, where there are $p = 50$ features generated as described in Section 2.6.1. For the binary response data, we append a sigmoid activation function to the end of the output layer. While the source and target data generating networks share architectures, we consider two relationships between the source and target neural network parameters.

In our first set of simulations, as in Section 2.6.1, we take the parameters of the source neural network to be $\boldsymbol{\theta}_S \stackrel{\text{iid}}{\sim} U(-1, 1)$ and define the parameters of the target neural network as $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}_{p \times 10 + 10 \times 1}(\mathbf{0}, 0.025\mathbf{I})$. For a continuous outcome, Table 2.5 shows that the RECaST DNN methods have the lowest RMSE for all sample sizes. Both RECaST methods outperform the target-only DNN across all settings, even when the target sample size is large ($n_T = 250$). The glmtrans method performs similarly to RECaST LM but worse than RECaST DNN. For all sample sizes, the RECaST framework produces wide posterior predictive intervals with 100% observed coverage for the 95% nominal confidence level – see Table A.1 in Appendix A.5. This greater than nominal coverage demonstrates RECaST will be conservative but reliable. Indeed, the observed over-coverage is safer than narrower intervals centered around incorrect values with below nominal coverage. For a binary outcome, Table 2.6 reveals that both RECaST methods outperform the target-only DNN for all sample sizes. This shows robustness to negative transfer. The performance of the RECaST methods is stable across target sample sizes in this setting, with stable AUCs and standard errors, whereas other methods degrade in performance as the target sample size decreases. Table 2.7 shows the empirical coverages of each method at the 75% nominal level. Only the RECaST GLM, RECaST DNN, and Wiens methods provide conservative coverage values for all sample sizes whereas the other methods tend to under-cover the true labels as the target sample size decreases.

Table 2.5: Out of sample RMSE (standard error) averaged over 300 source and target data sets when the generating models are neural networks and the target model parameters are generated as $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.025\mathbf{I})$. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	LM	DNN	RECaST LM	RECaST DNN	Unfreeze DNN	TransRF	glmtrans	MTL FO	MTL MoM
250	2.9(0.18)	3.1(0.22)	2.9(0.16)	2.1(0.15)	3(0.22)	3.9(0.53)	2.7(0.15)	3.1(0.21)	3.1(0.2)
100	3.8(0.36)	4.2(0.51)	2.9(0.16)	2.1(0.15)	3.3(0.37)	5.5(2.6)	2.7(0.18)	3.8(0.37)	3.9(0.38)
60	6.7(1.6)	5.1(0.56)	2.9(0.16)	2.1(0.15)	3.7(0.65)	12(10)	2.8(0.19)	6.3(1.3)	6.4(1.4)
40	6.1(1.1)	5.4(0.56)	2.9(0.16)	2.1(0.15)	4(0.67)	170(560)	2.8(0.18)	7.4(1.4)	6.8(1.5)
20	4.8(0.38)	5.8(0.44)	3(0.21)	2.2(0.17)	4.6(0.93)	-	2.9(0.26)	7.2(0.83)	4.9(0.42)

Table 2.6: Out-of-sample AUC (standard error) averaged over 300 source and target data sets when the generating models are neural networks and the target model parameters are generated as $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.025\mathbf{I})$. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	85(3.3)	92(1.6)	91(1.9)	75(4)	89(2.8)	64(13)	86(2.5)	89(1.7)
100	73(9.5)	92(1.6)	91(1.8)	75(3.5)	85(5.9)	46(18)	76(4.4)	89(1.7)
60	66(9.7)	92(1.7)	91(2)	75(4.4)	81(9)	29(22)	71(6.9)	89(2.2)
40	62(9)	92(1.8)	91(1.8)	76.0(3.5)	78(11)	24(16)	67(8)	89(1.9)
20	58(7.9)	92(1.8)	91(2)	76.0(3.6)	72(14)	-	56(6.4)	-

Orthogonal Source and Target Data Generating Model Parameters

In our second set of simulations, we set the source and target weight matrices to be orthogonal, i.e., $\boldsymbol{\theta}_S^\top \boldsymbol{\theta}_T = \mathbf{0}$. For a continuous outcome, Table 2.8 shows that RECaST again provides consistent predictive performance across target sample sizes. For small sample sizes, both RECaST methods outperform the target-only LM and DNN. The unfreeze DNN and glmtrans also perform well, but we mention again that they do not provide uncertainty quantification of predictions. Table A.2 in Appendix A.5 shows that RECaST provides conservative coverage intervals which, again, is a safe feature in this difficult transfer learning setting. For a binary outcome, Table 2.9 shows that RECaST again outperforms the target-only DNN in realistic settings where the target sample size is small. The RECaST methods have consistent AUCs across target sample sizes whereas other methods deteriorate as the sample size decreases. Table 2.10 shows that only RECaST GLM, RECaST DNN, and the Wiens method provide conservative uncertainty quantification for all target sample sizes at the 75% nominal level.

Overall, the results presented in this section show that RECaST is robust to negative transfer under more complex data generating mechanisms. In all cases, the RECaST methods outperformed the target-only DNN while boasting conservative predictive coverage intervals when the target sample size is small. In Appendix A.6 we explore other relationships between the source and target data when the data generating mechanism is a (generalized) linear model. These include orthogonality of source and target parameters and the target data having more features than the source.

Table 2.7: Empirical coverage (standard error) at the 75% nominal level for a binary response, averaged over 300 source and target data sets when the generating models are neural networks and the target model parameters are generated as $\theta_T = \theta_S + \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, 0.025\mathbf{I})$. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	68(13)	98(2)	97(4.1)	88(6.6)	71(13)	72(12)	80(11)	70(7.9)
100	63(12)	95(6.1)	94(6.7)	86(7.5)	70(13)	58(15)	72(12)	64(16)
60	59(12)	90(14)	91(10)	88(7.6)	66(14)	53(26)	75(16)	69(12)
40	58(12)	87(15)	87(13)	88(5.2)	63(15)	61(13)	69(17)	67(15)
20	55(12)	81(17)	81(18)	89(6.6)	59(14)	-	57(17)	-

Table 2.8: Out of sample RMSE (standard error) averaged over 300 source and target data sets when the source and target neural network weight matrices are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	LM	DNN	RECaST LM	RECaST DNN	Unfreeze DNN	TransRF	glmtrans	MTL FO	MTL MoM
250	0.93(0.43)	1.1(0.19)	2.9(0.91)	3(0.98)	1.6(0.45)	3.5(1.1)	1(0.41)	1.3(0.69)	1.3(0.69)
100	1.2(0.56)	2.3(0.51)	2.8(0.89)	3(0.97)	1.8(0.52)	5.9(3.2)	1.3(0.52)	1.6(0.86)	1.6(0.86)
60	2.2(1.1)	3.2(0.88)	2.8(0.89)	3(0.96)	2.1(0.62)	7.7(6.3)	1.9(0.7)	2.6(1.4)	2.7(1.3)
40	2.8(0.94)	3.8(0.97)	2.8(0.89)	3(0.97)	2.4(0.8)	21(24)	2.3(0.61)	4.5(1.3)	3.4(1.4)
20	3.7(1.1)	4.5(1.1)	2.9(0.91)	3.1(1)	2.9(0.89)	-	2.6(0.83)	6.6(1.1)	3.8(1.1)

2.7 eICU Data

The eICU Collaborative Research Database (Pollard et al. 2018) is a publicly available database of ICU encounters across multiple hospitals in the United States, making it well-suited for imitating transfer learning settings using real data. In the spirit of the transfer learning application in Wiens et al. (2014), we focus on correctly diagnosing physiological shock for newly admitted ICU patients. We define a binary response variable as the indicator of the event that a patient experienced shock upon ICU admission, using a combination of Internal Classification of Diseases 10 (ICD-10) codes: R57 Shock, not elsewhere classified; R58 Hemorrhage, not elsewhere classified; or R65.21 Severe sepsis with septic shock. Features are limited to baseline variables measured at admission. While the simulations of Section 2.6 explicitly link the source and target data through the data generation process, the similarity between source and targets defined in our eICU data application is unknown.

We consider 19 features including patient demographics, Acute Physiology Score III variables, and Glasgow Coma Scale test. Descriptions of these features can be found in Table A.11 in Appendix A.8. The data consist of measurements on 45,945 patients across 156 unique hospitals. Only 700 of these patients were diagnosed with shock upon admission. No individual hospital had enough positive cases to be reliably used as a source data set. To curate a balanced data set, we take all 700 patients with shock and randomly sample an additional 700 patients with

Table 2.9: Out-of-sample AUC (standard error) averaged over 300 source and target data sets when the source and target neural network weight matrices are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	94(1.8)	76(17)	87(11)	72(6.7)	90(6.3)	69(9.5)	95(2.1)	89(11)
100	84(7.2)	77(17)	87(11)	67(7.4)	87(7.9)	43(16)	79(5.6)	87(11)
60	74(10)	78(16.0)	87(11)	64(9)	83(11)	33(19)	66(11)	83(14)
40	68(10)	79(15)	87(11)	64(9.7)	81(13)	24(17)	61(8.4)	87(12)
20	60(9.6)	83(12)	87(12)	65(11)	77(14)	-	55(5.7)	-

Table 2.10: Empirical coverage (standard error) at the 75% nominal level for a binary response, averaged over 300 source and target data sets when the source and target neural network weight matrices are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	68(18)	100(0)	98.0(1.8)	77(29)	70(14)	73(9.5)	79(9.7)	65(19)
100	65(14)	100(0)	99.0(2.2)	80(25)	69(15)	58(13)	72(10)	64(17)
60	63(12)	88(3.5)	94.0(6.2)	80(23)	67(15)	57(16)	68(16)	53(17)
40	60(13)	89(9.9)	89(15)	75(20)	63(15)	61(20)	63(18)	67(15)
20	56(12)	81(15)	81(17)	78(22)	61(15)	-	56(15)	-

no shock. Next, 80% of the hospitals associated with our sampled 1,400 patients are randomly selected to define the ‘source hospital’. The source data set consists of all ICU encounters at the ‘source hospital’. Of the remaining 20% of hospitals, half are randomly assigned to the ‘target training hospital’, and the other half define a ‘target testing hospital’. Notice that this procedure splits hospitals rather than patients; the source data set may not consist of 80% of patients. The target training and target testing data sets typically contain 80 to 130 patients each.

We repeat the described sampling procedure 300 times, to imitate 300 transfer learning scenarios from real data. A logistic regression model and a DNN model are trained on each of the 300 source data sets, and all previously considered binary response transfer learning methods are implemented on the target data sets. To boost the performance of the source DNN model, the architecture of the DNN is chosen from a set of candidate architectures by maximizing AUC, averaged over 100 of the source data sets; additional details are provided in Appendix A.4. In Figure 2.2, we report the empirical coverage and AUC.

Because the real data generating model is unknown we consider two additional target-only models to test for negative transfer. We compare to a GLM and a Gaussian process (GP) trained only on the target data. In this setting, they perform worse than all of the transfer learning methods, with the GLM achieving an AUC of 0.606 and the GP achieving an AUC of 0.512. Plots for the TransRF, glmtrans, and WDGRL methods can be found in Appendix

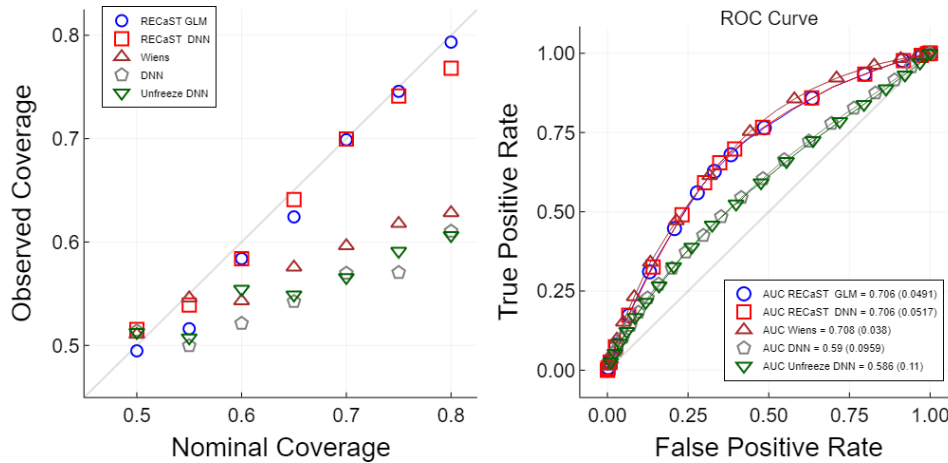


Figure 2.2: The left panel displays the reliability curve of the nominal versus empirical out-of-sample coverage of prediction sets averaged over 300 target-testing data sets; the right panel reports the out-of-sample receiver operating characteristic (ROC) curve averaged pointwise over 300 target-testing data sets. The legend also reports the AUC (standard error) averaged over the same 300 target-testing data sets. Note that we cut the reliability curve at a nominal coverage of 0.8 because there are very few observations with higher coverage, undermining the reliability of coverage estimation at higher nominal levels.

A.7. The AUCs of glmtrans and WDGRL were 0.68 and 0.708, respectively. RECaST has similar predictive performance to Wiens and WDGRL but without requiring access to the source data, and it outperforms the DNN and Unfreeze DNN approaches. Pairing RECaST with either the logistic regression or DNN source models produced near optimal average AUC, with respect to the average AUC values of 0.704 and 0.708, respectively, for the source logistic regression model and source DNN model. Figure 2.2 also demonstrates that RECaST generally produces prediction sets that achieve their nominal level of coverage for target test response values, even for non-linear models with non-Gaussian data, whereas the other approaches do not.

In addition to splitting the data into source and target by hospital, we explore making this division based on other features. First, we take the target data to be all patients aged 51 and under. This split resulted in approximately 20% of the patients in the target data and 80% in the source data. Second, we take the target data to be all patients aged 55 and under. This age was chosen because 20% of the patients *that experienced shock* are aged 55 and under. Third, we take the target data to be all female patients, which account for about 45% of the data. This more even split between source and target will be a good test for negative transfer. Finally, we take the target data to be all patients who are not Caucasian, corresponding to roughly 20% of the data.

Table 2.11 shows the average AUC, AUC standard error, and average empirical coverage at the 80% nominal level summarized over 300 target training and testing data sets. While the standard errors are large, we see that the average AUC of RECaST is larger than that of the

target-only methods in all but one setting. The only instance in which RECaST has smaller AUC is when the target data consist of the female patients. This may be due to the similar sample sizes between the source and target for this particular setting, as we demonstrated in the synthetic data simulations that RECaST is most advantageous when the target sample size is small. The RECaST AUCs are within a standard error of Wiens, glmtrans, and WDGRL, but RECaST does not require access to the source data. We see that the empirical coverages for the RECaST method are near the 80% nominal value; the Wiens and glmtrans methods are more conservative when the data are split by age. The TransRF method reports coverage lower than the 80% nominal level in all settings. This analysis demonstrates a general use case for RECaST as a clinical tool across a broad range of scenarios.

Table 2.11: Out-of-sample AUC (standard error) [empirical coverage at the 80% nominal level] averaged over 300 target training and testing data sets for each target data setting of the eICU data. All reported values are multiplied by 100.

	Age \leq 51	Age \leq 55	Female	Non-Caucasian
Target only GLM	71(6.9) [0.74]	71(6.0) [0.73]	70(4.2) [0.74]	67(6.4) [0.72]
Target only GP	66(16) [0.78]	67(13) [0.73]	64(11) [0.67]	61(11) [0.74]
Target only DNN	69(8.1) [0.72]	69(7.0) [0.69]	68(5.2) [0.70]	67(8.4) [0.69]
RECaST GLM	73(6.8) [0.78]	72(6.2) [0.78]	69(4.5) [0.77]	71(6.0) [0.84]
RECaST DNN	73(6.8) [0.82]	72(6.1) [0.78]	69(4.5) [0.79]	71(6.1) [0.83]
Unfreeze DNN	69(9.0) [0.75]	69(7.8) [0.73]	68(5.2) [0.72]	66(8.4) [0.70]
Wiens	73(6.5) [0.85]	73(5.7) [0.87]	70(4.5) [0.79]	71(6.6) [0.87]
glmtrans	71(7.1) [0.85]	71(5.5) [0.84]	70(4.7) [0.76]	66(7.2) [0.82]
TransRF	54(14) [0.69]	59(13) [0.71]	66(7.6) [0.72]	56(12) [0.69]
WDGRL	72(7.2) [0.76]	71(6.7) [0.73]	70(3.8) [0.74]	73(6.4) [0.80]

2.8 Concluding Remarks

The RECaST framework is adaptable to virtually any source model that makes predictions, and can accommodate both continuous and binary responses. The source data themselves are not required, which is a significant advantage when legal or ethical barriers to access of source data sets exist, e.g., due to privacy concerns. Unlike other transfer learning methods, RECaST always provides uncertainty quantification through prediction sets. Our conclusions are supported by both theoretical justifications and performance in simulation studies on synthetic and real data using linear and two-layer neural network source models.

The RECaST framework may be extended in several directions to accommodate the com-

plexity of EHR data. Broadening RECaST to handle differing feature spaces between source and target hospitals would allow for it to be applied in more general settings. As EHR databases are updated, it would be useful to perform online transfer learning. Patient clinical notes are also frequently available in EHR data and have been used by other transfer learning approaches (e.g., Si and Roberts 2020). However, transfer learning approaches that combine quantitative and text features to create a unified patient representation are currently lacking. Another promising direction is to study RECaST framework formulations for multi-class classification. One such formulation would be to specify the h function in Equation (2.2) as

$$h\{f(\boldsymbol{\theta}_s, \mathbf{x}_s), U_s\} = \sum_{k=1}^K k \cdot \mathbf{1}[U_s \in \Delta_k\{f(\boldsymbol{\theta}_s, \mathbf{x}_s)\}],$$

where K is the number of classes and $U_s \sim \text{Uniform}(\Delta)$ with $\Delta_1, \dots, \Delta_K$ – all functions of $f(\boldsymbol{\theta}_s, \mathbf{x}_s)$ – being triangular regions that form a partition of the simplex Δ over the multi-class outcome space (e.g., see, Jacob et al. 2021; Williams 2021).

CHAPTER

3

MULTIVARIATE AND ONLINE TRANSFER LEARNING WITH UNCERTAINTY QUANTIFICATION

3.1 Introduction

The field of transfer learning has developed to solve machine learning problems where data or computational resources may be limited. Transfer learning leverages information from *source* domains where data are plentiful to aid in learning about related, resource scarce *target* domains. Importantly, while these domains are similar they are not necessarily the same. If the domains are different, transfer learning methods may incur *negative transfer* where the inclusion of unrelated source information hinders the learning of the target task.

The Random Effect Calibration of Source to Target (RECaST) framework proposed in Hickey et al. (2024) is a Bayesian transfer learning method. The RECaST framework represents the similarity between the source and target as a Cauchy-distributed random effect. The posterior distribution of the random effect parameters is learned on the target data using a model built in the source domain. This learned posterior distribution is used to construct predictions

and posterior predictive credible sets for unseen target data – providing both prediction and uncertainty quantification without necessitating the access to the source data.

The HealthPartners Institute at Minneapolis, Minnesota has collected a longitudinal, observational data set of their participants through routine dental checkups as described in Guan et al. (2020). The data consider demographic and dental features along with two outcomes that are indicators of periodontal disease. Certain racial groups are much less represented than others in the data. Ignoring this imbalance and using all of the data jointly to build a single model could result in poor predictive performance and uncertainty quantification on underrepresented populations. We address this disparity by sharing information across related groups with multivariate online RECaST. The proposed methods enable us to model both outcomes jointly and to create better models for underrepresented racial groups. While transfer learning methods to model online or multivariate data exist, no others provide uncertainty quantification on the predictions.

We extend the RECaST framework in two ways. First, we model multivariate outcomes whereas the original framework was limited to univariate outcomes. We provide two different multivariate random effect distributions to model the similarity between source and target. One uses a multivariate Cauchy distribution, a natural extension from the univariate Cauchy. The other incorporates the univariate Cauchy marginal distributions of the original RECaST framework with a multivariate Gaussian copula. Second, we propose an online method to share information between sequential target data sets while mitigating negative transfer. As we demonstrate in the data analysis, this online analysis can also be applied to data that are collected concurrently but modeled sequentially.

The newly proposed methods retain many of the advantages of the original RECaST framework. Data privacy is maintained between all data sets. Only the source model, not the source data, is required when fitting the posterior distribution for the target data. When considering multiple target data sets, only the learned posterior distribution from previous targets are shared and not the data themselves. This property is especially important when working with sensitive data such as medical information. Additionally, the framework remains agnostic to the specification of the source model, such that it can correspond to a parametric generalized linear model or a highly flexible neural network.

The RECaST methods are evaluated on both predictive performance and posterior predictive coverage. We consider synthetic simulation studies that vary sample size, outcome correlation, and similarity between source and target data sets. Changes in these parameters represent different challenges of real data sets and enables us to isolate the effects that each parameter has on predictive performance and uncertainty quantification. Negative transfer is examined throughout the studies.

The remainder of this paper is organized as follows. In Section 3.2 we discuss related work in transfer learning. We provide a summary of the original RECaST framework that this work builds upon in Section 3.3. We extend this framework in Section 3.4 to model multivariate outcomes, providing two natural modeling choices. In Section 3.5, we develop an online extension to handle sequential target data sets. We test these methods in Section 3.6 through extensive simulation studies. Section 3.7 analyzes the periodontal disease outcomes. Section 3.8 concludes. Further mathematical and computational details can be found in the Appendix.

3.2 Related Work

Pan and Yang (2010), Lu et al. (2015), Weiss et al. (2016), Dube et al. (2020), Zhuang et al. (2021) are general survey papers on transfer learning. See Suder et al. (2023) for a specialized survey of Bayesian transfer learning methods.

Using transfer learning for underrepresented demographic groups can provide significant improvement when compared to building models where demographic differences are ignored. TransRF is a random forest based approach to predict breast cancer using European women as the source population and patients with African and South Asian ancestry as the target population (Gu et al. 2022). In Hong et al. (2024), a LASSO based federated transfer learning method is proposed to address demographic and outcome class imbalance for stroke risk prediction. In Li et al. (2023), ideas from sparse high-dimensional regression are used to construct polygenic risk scores for Type II diabetes for patients in underrepresented racial groups using genome-wide association studies. The STRIFLE method builds upon this by including nuisance parameters which provide robustness to negative transfer (Tianxi Cai and Liu 2024).

Multivariate outcomes are common in transfer learning, especially for computer vision tasks. A deep convolutional neural network built to classify images on a fixed set of outcome classes is generalized to predict outcomes in other domains in Donahue et al. (2014b). In Maddox et al. (2021), a method is proposed to handle posterior sampling for a large number outcomes which incurs expensive sampling because of a large covariance matrix in . In Singh et al. (2023), the standard approach of pre-training a neural network in the source domain and fine-tuning in the target domain is expanded to include an additional pre-pre-training step to improve foundational models. Sequential improvement through online image data sets is considered in Yu et al. (2024). These methods focus exclusively on prediction while multivariate RECaST also provides valuable uncertainty quantification.

Multitask learning uses information from a source domain to improve performance on multiple target domains (Caruana 1997). In Bakker and Heskes (2003), a Bayesian neural network method is proposed that fits the output network weights with task-specific data, but

uses data from all domains to learn shared parameters for the rest of the network. A multitask Bayesian optimization for Gaussian process models that focuses on efficiently handling the exploitation-exploration trade-off of parameter optimization to jointly minimize error over all tasks is developed in Swersky et al. (2013). These methods generally optimize over all of the target tasks jointly. In contrast, the proposed online RECaST method learns a posterior distribution specific to each target task.

Online and online-batch transfer learning methods focus on updating model parameters as data sets arrive sequentially. This is in contrast to batch learning which requires all of the data to be present before training can occur. Opper and Winther (1999) develop an early approach to updating a Bayesian posterior distribution as new data points arrive, boasting asymptotic efficiency with Gaussian families. An online Bayesian method that combines models from multiple data sets is proposed by Chen et al. (2001). Wu et al. (2017b) perform online transfer learning by weighing models built on multiple source domains in an ensemble in combination with a target domain classifier to improve prediction in the target domain. In Wu et al. (2023), performance bounds are calculated based on source and target sample sizes to choose informative Bayesian priors when the target domain data arrive in one batch, online, or in sequential batches. In Patel et al. (2023), a federated, online approach is proposed to a sequential decision making framework where communication between sites is limited. A weighting scheme is developed for online data across multiple sites to control the trade-off between training time and bias in Marfoq et al. (2023).

Some Bayesian transfer learning methods focus on learning an informative prior to improve prediction on a target task. Raina et al. (2006b) propose a method to learn a covariance matrix from source text classification tasks which is used as a prior on the covariance matrix in the target domain. In Kapoor et al. (2021), a continual learning approach to Bayesian inference is developed that updates a posterior distribution as new tasks arrive using a Gaussian process. A tuning parameter that adapts the posterior learned on a source task to be used as the prior on a target task is introduced in Shwartz-Ziv et al. (2022). This is similar to the proposed online RECaST with some key differences: online RECaST uses posterior information from previous *target* data sets in the prior distribution and introduces a weighting parameter that is *learned* rather than tuned. In Abba et al. (2023b), a penalized complexity prior based on the Kullback–Leibler divergence between the source and target models is developed. A shrinkage estimator is used in the prior when the difference between the source and target tasks is sparse in Abba et al. (2024). The online RECaST method that we propose uses a prior informed by previous target data sets while mitigating negative transfer.

3.3 Summary of the RECaST Framework

We begin with a summary of the RECaST framework from Hickey et al. (2024). RECaST is a Bayesian transfer learning method for source and target data sets that share the same outcome and feature spaces but may have differences in feature-to-outcome mappings. It is scalable, requiring estimation of only 2-3 parameters, which is especially important when there are minimal target data. RECaST maintains data privacy, requiring only the fitted source model to be shared and not the source data. The estimation of a Bayesian posterior distribution naturally begets uncertainty quantification through posterior predictive credible intervals.

Take Y_S to be an observation from the source data set with corresponding features \mathbf{x}_S and Y_T to be an observation from the target data set with features \mathbf{x}_T . Denote the forward data-generating mechanism of the feature-to-outcome mappings $P_S(y_S | \mathbf{x}_S)$ and $P_T(y_T | \mathbf{x}_T)$, respectively, by

$$Y_S = h\{f(\boldsymbol{\theta}_S, \mathbf{x}_S), U_S\} \quad \text{and} \quad Y_T = h\{g(\boldsymbol{\theta}_T, \mathbf{x}_T), U_T\},$$

where $f(\boldsymbol{\theta}_S, \mathbf{x}_S)$ and $g(\boldsymbol{\theta}_T, \mathbf{x}_T)$ are the *structural components* that relate an observation's features \mathbf{x} to the data generating parameters $\boldsymbol{\theta}$. The h function relates these structural components to U_S and U_T , which are independent auxiliary random variables. For example,

$$\begin{aligned} f(\boldsymbol{\theta}_S, \mathbf{x}_S) &= \mathbf{x}_S^\top \boldsymbol{\theta}_S, \\ h(\mathbf{x}_S^\top \boldsymbol{\theta}_S, U_S) &= f(\boldsymbol{\theta}_S, \mathbf{x}_S) + U_S, \quad \text{and} \\ U_S &\sim \mathcal{N}(0, 1), \end{aligned}$$

then $Y_S \sim \mathcal{N}(\mathbf{x}_S^\top \boldsymbol{\theta}_S, 1)$. With a small target sample size it may be infeasible to estimate $g(\boldsymbol{\theta}_T, \mathbf{x}_T)$ making transfer learning necessary to model the target task.

The RECaST framework defines $\beta := g(\boldsymbol{\theta}_T, \mathbf{x}_T)/f(\boldsymbol{\theta}_S, \mathbf{x}_T)$ which intuitively represents the *similarity* between the source and target data generating mechanisms. If the source and target are generated in a similar way, then $\beta \approx 1$. With this new β term, the target generating mechanism can be expressed as

$$\begin{aligned} Y_{T,i} &= h\{g(\boldsymbol{\theta}_T, \mathbf{x}_{T,i}), U_{T,i}\} \\ &= h\left\{\frac{f(\boldsymbol{\theta}_S, \mathbf{x}_{T,i})}{f(\boldsymbol{\theta}_S, \mathbf{x}_{T,i})} \cdot g(\boldsymbol{\theta}_T, \mathbf{x}_{T,i}), U_{T,i}\right\} \\ &= h\{\beta_i \cdot f(\boldsymbol{\theta}_S, \mathbf{x}_{T,i}), U_{T,i}\} \end{aligned}$$

for $i \in \{1, \dots, n_T\}$, where $Y_{T,1}, \dots, Y_{T,n_T}$ is an independent sample of n_T target outcomes with

associated features $\mathbf{x}_{T,1}, \dots, \mathbf{x}_{T,n_T}$ and $\beta_i = g(\boldsymbol{\theta}_T, \mathbf{x}_{T,i})/f(\boldsymbol{\theta}_S, \mathbf{x}_{T,i})$. Notice that now the model for $Y_{T,i}$ depends on $f(\boldsymbol{\theta}_S, \mathbf{x}_{T,i})$, which is assumed to have a reliable estimate from the source, and β_i . The reliance on estimating $g(\boldsymbol{\theta}_T, \mathbf{x}_{T,i})$ has been replaced with β_i which is modeled as a random effect.

Lemma 1 of Hickey et al. (2024) states in the canonical case where $\mathbf{x}_{T,1}, \dots, \mathbf{x}_{T,n_T} \stackrel{\text{iid}}{\sim} \mathcal{N}_p(\mathbf{0}, \mathbf{I}_p)$, and $g(\boldsymbol{\theta}, \mathbf{x}) = f(\boldsymbol{\theta}, \mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}$ that $\beta_i \sim \text{Cauchy}(\delta, \gamma)$ with

$$\delta = \frac{\boldsymbol{\theta}_T^\top \boldsymbol{\theta}_S}{\|\boldsymbol{\theta}_S\|^2} \quad \text{and} \quad \gamma = \frac{1}{\|\boldsymbol{\theta}_S\|^2} \sqrt{\|\boldsymbol{\theta}_S\|^2 \|\boldsymbol{\theta}_T\|^2 - (\boldsymbol{\theta}_T^\top \boldsymbol{\theta}_S)^2}. \quad (3.1)$$

In this case, the distribution of β_i exactly follows a Cauchy distribution. In fact, the heavy tails of the Cauchy distribution also make it practically useful for situations where those canonical assumptions are violated. Since β_i characterizes the similarity between the source and the target domains, these heavy tails allow the random effect to capture large disparities between the source and target. RECaST models the β_i terms as random effects following a $\text{Cauchy}(\delta, \gamma)$ distribution with shared parameters δ and γ . The joint posterior distribution of (δ, γ) is learned using the target data and the pre-fitted source model. RECaST parameters δ and γ can be learned using standard Markov chain Monte Carlo methods even outside of the canonical case. The Bayesian posterior predictive distribution can be sampled to generate empirical credible intervals for uncertainty quantification.

3.4 Multivariate RECaST

3.4.1 Data Generating Mechanism

We now extend the RECaST framework to model multivariate outcomes. Take $\mathbf{y} = [y_1, \dots, y_m]^\top$ to be a vector of m outcomes, $\mathbf{x} = [x_1, \dots, x_p]$ to be the corresponding vector of p features, and $\boldsymbol{\Theta}$ to be the generating parameters. The forward data-generating mechanisms $P_S(\mathbf{y}_S | \mathbf{x}_S)$ and $P_T(\mathbf{y}_T | \mathbf{x}_T)$ can be represented, respectively, by

$$\mathbf{Y}_S = h\{f(\boldsymbol{\Theta}_S, \mathbf{x}_S), \mathbf{U}_S\} \quad \text{and} \quad \mathbf{Y}_T = h\{g(\boldsymbol{\Theta}_T, \mathbf{x}_T), \mathbf{U}_T\},$$

where \mathbf{U}_S and \mathbf{U}_T are independent auxiliary random variables with dimension $m \times m$.

As before, the transfer learning question arises when there is not enough target data to reliably estimate $g(\boldsymbol{\Theta}_T, \mathbf{x}_T)$. To capture the similarity between the source and target domains we define $\beta_{i,j} := g(\boldsymbol{\Theta}_{T,j}, \mathbf{x}_{T,i})/f(\boldsymbol{\Theta}_{S,j}, \mathbf{x}_{T,i})$ for $i = 1, \dots, n_T$ and $j = 1, \dots, m$. In words, for each target observation i , there is a $\beta_{i,j}$ ratio for each outcome that models the relation between

the source and the target outcome. Take $\boldsymbol{\beta}_i = [\beta_{i,1}, \dots, \beta_{i,m}]^\top$ to be the vector of ratios and $\text{diag}(\boldsymbol{\beta})$ to be the matrix with elements of $\boldsymbol{\beta}$ on the diagonal and 0's elsewhere. With this, we now express the target generating mechanism as

$$\begin{aligned} \mathbf{Y}_{T,i} &= h\{g(\boldsymbol{\Theta}_T, \mathbf{x}_{T,i}), \mathbf{U}_{T,i}\} \\ &= h\{\text{diag}(\boldsymbol{\beta}_i)f(\boldsymbol{\Theta}_S, \mathbf{x}_{T,i}), \mathbf{U}_{T,i}\}. \end{aligned}$$

Thus, $\mathbf{Y}_{T,i}$ is no longer expressed as a function of $g(\boldsymbol{\Theta}_T, \mathbf{x}_{T,i})$, which cannot be reliably estimated. The function $g(\boldsymbol{\Theta}_T, \mathbf{x}_{T,i})$ is replaced by $f(\boldsymbol{\Theta}_S, \mathbf{x}_{T,i})$, which we assume can be reliably estimated from the source domain, and $\boldsymbol{\beta}_i$, the random effect for individual i , $i = 1, \dots, n_T$. In addition to removing the dependence on $g(\boldsymbol{\Theta}_T, \mathbf{x}_{T,i})$, considering The relationship between the outcomes in the target domain will be estimated through the parameters of $\mathbf{U}_{T,i}$.

Considering ratios of different outcomes, such as $g(\boldsymbol{\Theta}_{T,j}, \mathbf{x}_{T,i})/f(\boldsymbol{\Theta}_{S,k}, \mathbf{x}_{T,i})$ for $j \neq k$, would result in a non-diagonal matrix multiplied by $f(\boldsymbol{\Theta}_S, \mathbf{x}_{T,i})$ and thus would not remove the dependence on $g(\boldsymbol{\Theta}_T, \mathbf{x}_{T,i})$. Additionally, considering only ratios makes the $\beta_{i,j}$ terms interpretable: if the source and target are generated from the same distribution, then $\boldsymbol{\beta}_i = \mathbf{1}_m$, a vector of 1's. This mirrors $\beta = 1$ in univariate RECaST described in Section 3.3. It is unclear what the expected behavior would be if a ratio of different outcomes were considered.

Unlike univariate RECaST, the multivariate random effect $\boldsymbol{\beta}_i$ does not always follow a multivariate Cauchy distribution, even in the canonical case. A vector of random variables follows a multivariate Cauchy distribution if and only if every linear combination of the components follows a univariate Cauchy distribution. Because of the covariance between the elements of $\boldsymbol{\beta}_i$, this is not always the case (Pillai 2016; Pillai and Meng 2016). Thus, we provide two choices of random effect distributions for $\boldsymbol{\beta}_i$. The first is a multivariate Cauchy distribution. While this may not always be the exact distribution of $\boldsymbol{\beta}_i$, it is a natural extension of the univariate model. In practice, the heavy tails are, again, beneficial for capturing the relationship between source and target even if the underlying distributions differ. Second, we propose a copula-based approach that leverages the fact that, in the canonical case, the marginal distributions of the elements of $\boldsymbol{\beta}_i$ are known to be univariate Cauchy. A multivariate Gaussian copula is used to model the dependence structure. In practice, these models can be used regardless of the form of the source model $f(\boldsymbol{\Theta}_S, \mathbf{x})$.

3.4.2 Multivariate Cauchy

Univariate RECaST models the scalar β term with a Cauchy distribution. This follows from β representing the ratio of two normally distributed random variables in the canonical case.

The natural extension for a multi-dimensional outcome is to model $\boldsymbol{\beta}$ with an m -dimensional multivariate Cauchy distribution: $\boldsymbol{\beta}_i \sim \text{Cauchy}_m(\boldsymbol{\delta}, \boldsymbol{\Gamma})$. Here $\boldsymbol{\delta} \in \mathbb{R}^{m \times 1}$ is the location vector and $\boldsymbol{\Gamma}$ is an $m \times m$ positive definite scale matrix, $i \in \{1, \dots, n_T\}$. If the source and target come from similar distributions we expect the elements of $\boldsymbol{\delta}$ to be close to 1. The $\boldsymbol{\Gamma}$ matrix captures the dependence between the elements of $\boldsymbol{\beta}_i$. For continuous responses $\mathbf{y} \in \mathbb{R}^{m \times 1}$, a natural choice for the h function is the Gaussian innovation function,

$$\mathbf{y}_{T,i} = \text{diag}(\boldsymbol{\beta}_i) f(\widehat{\boldsymbol{\Theta}}_S, \mathbf{x}_{T,i}) + \mathbf{U}_{T,i},$$

where $\mathbf{U}_{T,i} \sim \mathcal{N}_m(0, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma}$ is an unknown $m \times m$ covariance matrix, and $\widehat{\boldsymbol{\Theta}}_S$ are parameter estimates from the source model. Denoting $\pi(\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma})$ as a prior density on $(\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma})$, the posterior distribution of $(\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma})$ is

$$\begin{aligned} p(\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma} | \mathbf{y}_{T,1}, \dots, \mathbf{y}_{T,n_T}, \widehat{\boldsymbol{\Theta}}_S) \\ \propto \pi(\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}) \cdot \prod_{i=1}^{n_T} \int_{\mathbb{R}^m} \mathcal{N}_m\{\mathbf{y}_{T,i} | \text{diag}(\boldsymbol{\beta}_i) f(\widehat{\boldsymbol{\Theta}}_S, \mathbf{x}_{T,i}), \boldsymbol{\Sigma}\} \cdot \text{Cauchy}_m(\boldsymbol{\beta}_i | \boldsymbol{\delta}, \boldsymbol{\Gamma}) d\boldsymbol{\beta}_i. \end{aligned} \quad (3.2)$$

Since $\boldsymbol{\delta}$ is a real valued location vector, $\boldsymbol{\Gamma}$ is a scale matrix, and $\boldsymbol{\Sigma}$ is a covariance matrix, a canonical choice of prior distribution is

$$\pi(\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}) = \mathcal{N}_m(\boldsymbol{\delta} | \mathbf{1}_m, \boldsymbol{\Sigma}_\delta) \cdot \text{IW}_m(\boldsymbol{\Gamma} | \boldsymbol{\Psi}_\Gamma, \nu_\Gamma) \cdot \text{IW}_m(\boldsymbol{\Sigma} | \boldsymbol{\Psi}_\Sigma, \nu_\Sigma),$$

where IW is the inverse-Wishart distribution. The hyperparameters for $\boldsymbol{\Sigma}$ can be chosen based on prior information known about the covariance of the outcomes in the target domain. The hyperparameters for $\boldsymbol{\delta}$ and $\boldsymbol{\Gamma}$ can be chosen based on the relationship between the source and target domains; if they are known to be similar then a small covariance for $\boldsymbol{\delta}$ and a mean near the identity for $\boldsymbol{\Gamma}$ will result in a prior that favors $\boldsymbol{\beta}$ being near $\mathbf{1}_m$. In practice, we choose hyperparameters that are diffuse to demonstrate that RECaST can still perform well on unseen target test data without prior information.

The posterior distribution is estimated using the random walk Metropolis-Hastings algorithm using the `nimble` package for R (de Valpine et al. 2017). The total number of iterations $n_{\text{iterations}}$ and number of burn-in steps can be chosen based on available computational resources at the target site or stopped early based on convergence. The computational complexity is $\mathcal{O}(n_T \cdot n_{\text{iterations}})$. This alleviates concerns about scalability since n_T is assumed to be small for transfer learning problems. For this choice of random effect distribution, a Gibbs sampler could also be used; the details for this, including the full conditional distributions, are outlined in Appendix A.9. Once the posterior distribution is estimated, the posterior predictive distribution

is constructed as follows.

Take $\tilde{\mathbf{y}}_T$ to be the outcome vector of a newly observed target observation with corresponding features $\tilde{\mathbf{x}}_T$. The posterior predictive distribution of $\tilde{\mathbf{y}}_T$ is the marginal distribution of

$$\begin{aligned} & \pi(\tilde{\mathbf{y}}_T, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma} \mid \mathbf{y}_{T,1}, \dots, \mathbf{y}_{T,n_T}, \hat{\boldsymbol{\Theta}}_S) \\ &= \mathcal{N}_m\{\tilde{\mathbf{y}}_T \mid \text{diag}(\tilde{\boldsymbol{\beta}})f(\hat{\boldsymbol{\Theta}}_S, \tilde{\mathbf{x}}_T), \boldsymbol{\Sigma}\} \cdot \text{Cauchy}_m(\tilde{\mathbf{B}} \mid \boldsymbol{\delta}, \boldsymbol{\Gamma}) \cdot p(\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma} \mid \mathbf{y}_{T,1}, \dots, \mathbf{y}_{T,n_T}). \end{aligned}$$

Algorithm 3 describes a procedure for drawing posterior predictive samples from this distribution. This procedure gives a sample of outcome vectors for the newly observed features $\tilde{\mathbf{x}}$ of size $n_{\text{post}} \cdot n_{\beta} \cdot n_Y$. Correspondingly, the computational complexity is $\mathcal{O}(n_{\text{post}} \cdot n_{\beta} \cdot n_Y)$; the sampling parameters can be adjusted based on available computing resources. Given a nominal coverage level α , we create elliptical posterior predictive credible sets from these samples. For each sampled outcome, we compute the Mahalanobis distance to the mean of the samples; denote this set of Mahalanobis distances as **MH**. The outcome is covered if the Mahalanobis distance between the true value $\tilde{\mathbf{y}}_T$ and the mean of the sampled outcomes is within the inner $1 - \alpha$ percentile interval of **MH**.

Algorithm 3 Multivariate Cauchy RECaST Posterior Predictive Sampling

Input: $\tilde{\mathbf{x}}_T$, an estimated posterior $p(\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma} \mid \mathbf{y}_{T,1}, \dots, \mathbf{y}_{T,n_T})$, and sample sizes n_{post} , n_{β} , and n_Y

Output: A sample of values from $\pi(\tilde{\mathbf{y}}_T \mid \mathbf{y}_{T,1}, \dots, \mathbf{y}_{T,n_T}, \hat{\boldsymbol{\Theta}}_S)$

```

for  $i \leftarrow 1$  to  $n_{\text{post}}$  do
   $\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma} \leftarrow \text{random}\{p(\boldsymbol{\delta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma} \mid \mathbf{y}_{T,1}, \dots, \mathbf{y}_{T,n_T}, \hat{\boldsymbol{\Theta}}_S)\}$ 
  for  $j \leftarrow 1$  to  $n_{\beta}$  do
     $\tilde{\boldsymbol{\beta}} \leftarrow \text{random}\{\text{Cauchy}_m(\boldsymbol{\delta}, \boldsymbol{\Gamma})\}$ 
    for  $k \leftarrow 1$  to  $n_Y$  do
       $\tilde{\mathbf{Y}}_T \leftarrow \text{random}[\mathcal{N}_m\{\tilde{\mathbf{y}}_T \mid \text{diag}(\tilde{\boldsymbol{\beta}})f(\hat{\boldsymbol{\Theta}}_S, \tilde{\mathbf{x}}_T), \boldsymbol{\Sigma}\}]$ 
    end for
  end for
end for
end for

```

3.4.3 Multivariate Normal Copula with Cauchy Marginals

Copulas are a modeling approach that construct a joint distribution from marginal distributions (Nelsen 2006). From univariate RECaST we know that the ratios $\beta_j = (\boldsymbol{\Theta}_{T,j} \mathbf{x}_T) / (\boldsymbol{\Theta}_{S,j} \mathbf{x}_T) \sim \text{Cauchy}(\delta_j, \gamma_j)$ in the canonical case where $j = 1, \dots, m$ indexes the outcomes. That is, the marginal distributions of the elements of $\boldsymbol{\beta}$ are known to be univariate Cauchy with their own

center and scale parameters δ_j and γ_j , respectively. As with univariate RECaST, this canonical linear form is used to motivate the choice of parametric family but is not required to apply RECaST. Define $u_j := F_{\beta_j}(\beta_j)$ where $F_{\beta_j}(\cdot)$ is the cumulative distribution function of the Cauchy distribution associated with β_j . By the probability integral transform, $u_{i,j} \sim U(0, 1)$. We use a Gaussian copula centered at $\mathbf{0}_m$ with correlation matrix \mathbf{R} to capture the dependence between the β_j 's. The joint distribution function of the β_j 's is given by

$$\begin{aligned} & \pi(\beta_1, \dots, \beta_m \mid \mathbf{R}, \delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m) \\ &= \prod_{j=1}^m f_{\beta_j}(\beta_j) \cdot c(u_1, \dots, u_m \mid \mathbf{R}, \delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m) \\ &= \prod_{j=1}^m [f_{\beta_j}\{F_{\beta_j}^{-1}(u_j)\}] \cdot \mathcal{N}_m\left(\left[\Phi^{-1}\{F_{\beta_1}(\beta_1)\} \quad \dots \quad \Phi^{-1}\{F_{\beta_m}(\beta_m)\}\right] \mid \mathbf{0}_m, \mathbf{R}\right), \end{aligned}$$

where $\Phi^{-1}(\cdot)$ is the inverse of the standard Gaussian cumulative distribution function. The unknown parameters are the center parameters for the Cauchy marginal distributions $\delta_1, \dots, \delta_m$, the scale parameters for the Cauchy marginal distributions $\gamma_1, \dots, \gamma_m$, the correlation matrix for the multivariate Gaussian copula over the random effects \mathbf{R} , and the covariance matrix over the target outcomes Σ . The posterior distribution for these unknown parameters is

$$\begin{aligned} & p(\delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \Sigma \mid \mathbf{y}_1, \dots, \mathbf{y}_{n_T}, \widehat{\Theta}_S) \\ & \propto \pi(\delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \Sigma) \cdot \\ & \quad \prod_{i=1}^{n_T} \left\{ \int_{\mathbb{R}} \dots \int_{\mathbb{R}} \pi(\mathbf{y}_i \mid \beta_{i,1}, \dots, \beta_{i,m}, \Sigma, \widehat{\Theta}_S) \cdot \right. \\ & \quad \left. \pi(\beta_{i,1}, \dots, \beta_{i,m} \mid \delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}) d\beta_{i,1} \dots d\beta_{i,m} \right\}. \end{aligned} \quad (3.3)$$

As before, the posterior is estimated using the Metropolis-Hastings algorithm. The integrals of this posterior distribution can also be calculated with respect to the $u_{i,j}$; this may have computational advantages since the integrals are on the bounded interval $[0, 1]$. The conversion of these integrals is provided in Appendix A.10. A canonical choice of prior for the unknown parameters is

$$\begin{aligned} & \pi(\delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \Sigma) \\ &= \text{IW}_m(\mathbf{R} \mid \Psi_{\mathbf{R}}, \nu_{\mathbf{R}}) \cdot \text{IW}_m(\Sigma \mid \Psi_{\Sigma}, \nu_{\Sigma}) \cdot \prod_{j=1}^m \mathcal{N}(\delta_j \mid \mu_{\delta_j}, \sigma_{\delta_j}^2) \cdot \text{IG}(\gamma_j \mid a_{\gamma_j}, b_{\gamma_j}), \end{aligned}$$

where IG is the inverse-gamma distribution.

The posterior predictive distribution for a new target observation $\tilde{\mathbf{y}}_T$ with associated features $\tilde{\mathbf{x}}_T$ is the marginal distribution of

$$\begin{aligned} & \pi(\tilde{\mathbf{y}}_T, \tilde{\beta}_1, \dots, \tilde{\beta}_m, \delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \Sigma | \mathbf{y}_1, \dots, \mathbf{y}_{n_T}, \hat{\Theta}_S) \\ &= \pi(\tilde{\mathbf{y}}_T | \tilde{\beta}_1, \dots, \tilde{\beta}_m, \Sigma) \cdot \pi(\tilde{\beta}_1, \dots, \tilde{\beta}_m | \delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}) \cdot \\ & \quad \pi(\delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \Sigma | \mathbf{y}_1, \dots, \mathbf{y}_{n_T}, \hat{\Theta}_S). \end{aligned}$$

Algorithm 4 describes how to sample outcomes from this posterior distribution. These samples can be used to create posterior predictive credible intervals and calculate empirical coverages as discussed in the previous section.

Algorithm 4 Multivariate Copula RECaST Posterior Predictive Sampling

Input: $\tilde{\mathbf{x}}_T$, an estimated posterior $p(\delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \Sigma | \mathbf{y}_1, \dots, \mathbf{y}_{n_T}, \hat{\Theta}_S)$, and sample sizes n_{post} , n_β , and n_Y

Output: A sample of values from $\pi(\tilde{\mathbf{y}}_T | \mathbf{y}_{T,1}, \dots, \mathbf{y}_{T,n_T}, \hat{\Theta}_S)$

for $i \leftarrow 1$ to n_{post} **do**

$\delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \Sigma \leftarrow \text{random}\{p(\delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \Sigma | \mathbf{y}_1, \dots, \mathbf{y}_{n_T}, \hat{\Theta}_S)\}$

for $j \leftarrow 1$ to n_β **do**

$\tilde{\beta}_1 \leftarrow \text{random}\{\text{Cauchy}(\delta_1, \gamma_1)\}$

\vdots

$\tilde{\beta}_m \leftarrow \text{random}\{\text{Cauchy}(\delta_m, \gamma_m)\}$

for $k \leftarrow 1$ to n_Y **do**

$\tilde{Y}_T \leftarrow \text{random}[\mathcal{N}_m\{\tilde{\mathbf{y}}_T | \tilde{\mathbf{B}}f(\hat{\Theta}_S, \tilde{\mathbf{x}}_T), \Sigma\}]$

end for

end for

end for

3.5 Online RECaST

We further build upon the RECaST framework to model multiple sequential target data sets. We address the problem of online transfer learning, with multiple target domains, each of which may be lacking in sample size or computational resources. As with previous RECaST methods, we operate under the assumption that the data are private and cannot be shared across sites, even between targets. We propose a sequential approach that includes a RECaST model built in one target domain to inform a new target domain.

We motivate the problem with an example. A hospital has data from the past 10 years to build a model that reliably diagnoses whether a patient has the flu – this will act as the source. Now the hospital wants to use that model on incoming patients for this year’s flu season – this target data set is T_1 . The influenza virus itself, however, may have changed or evolved and the previously built model may not be directly applicable. So a RECaST transfer learning model is used to calibrate the source model to this year’s target data. The following year’s flu season will see more patients – this target data set is T_2 . The online RECaST method uses the source model and the T_1 model to build a model on the T_2 patients. It is possible, however, that the new flu strain is very different between T_1 and T_2 and that including this extra information would be harmful. This would be an example of negative transfer. Our online method incorporates source and previous target data sets while reducing the influence of negative transfer.

The online RECaST method can be applied to the univariate and multivariate outcome cases. Take $\mathbf{y}_{T_1,1}, \dots, \mathbf{y}_{T_1,n_{T_1}}$ to be a sample of n_{T_1} outcomes from the first target, T_1 , and $\mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2,n_{T_2}}$ to be a sample of n_{T_2} outcomes from the second target, T_2 , with associated features $\mathbf{x}_{T_2,1}, \dots, \mathbf{x}_{T_2,n_{T_2}}$. Take Ω to be the unknown parameters: in the multivariate Cauchy model $\Omega = \{\delta, \Gamma, \Sigma\}$ and in the multivariate Gaussian copula model $\Omega = \{\delta_1, \dots, \delta_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \Sigma\}$. Let $f(\widehat{\Theta}_S, \mathbf{x})$ be the fitted source model and $p(\Omega | \mathbf{y}_{T_1,1}, \dots, \mathbf{y}_{T_1,n_{T_1}}, \widehat{\Theta}_S)$ be the learned RECaST posterior distribution for T_1 given by either Equation (3.2) or Equation (3.3). Only the learned posterior distribution on T_1 is shared and not the data themselves. In transferring from S and T_1 to T_2 , we construct a new prior distribution on Ω given by

$$\pi_*(\Omega, \alpha) = \pi(\alpha) \left\{ \alpha \cdot p(\Omega | \mathbf{y}_{T_1,1}, \dots, \mathbf{y}_{T_1,n_{T_1}}, \widehat{\Theta}_S) + (1 - \alpha) \cdot \pi(\Omega) \right\}.$$

where $\alpha \in [0, 1]$ is a weight parameter to be estimated. The prior distribution is a convex combination of the RECaST posterior distribution learned from T_1 and another prior $\pi(\Omega)$ that is uninformed by T_1 . The weight parameter α controls the degree to which each prior influences the posterior. An additional prior $\pi(\alpha)$ is included as a prior on the weight parameter. A larger α value indicates an increase in the weight of the posterior from T_1 ; this is the behavior we would expect the model to learn if T_1 is informative for T_2 . If $\alpha = 0$ no information from the T_1 is used; then this reverts to the same posterior as single target RECaST and avoids negative transfer. The posterior distribution of (Ω, α) is

$$p(\Omega, \alpha | \mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2,n_{T_2}}, \widehat{\Theta}_S) \propto \pi_*(\Omega, \alpha) \cdot \prod_{i=1}^{n_{T_2}} \int_{\mathbb{R}^m} p(\mathbf{y}_{T_2,i} | \text{diag}(\beta_i), \Omega, \widehat{\Theta}_S) \cdot p(\beta_i | \Omega) d\beta_i.$$

As this new parameter α is only present in the prior, the posterior predictive sampling algorithms for $\tilde{\mathbf{y}}_{T_2}$ are unchanged.

The marginal mean of α gives insight into the amount of information shared from T_1 . First, the marginal posterior distribution of α is

$$\begin{aligned} p(\alpha | \mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2, n_{T_2}}, \widehat{\Theta}_S) \\ &= \int p(\boldsymbol{\Omega}, \alpha | \mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2, n_{T_2}}, \widehat{\Theta}_S) d\boldsymbol{\Omega} \\ &\propto \pi(\alpha) \cdot \left\{ \alpha \cdot \left[\int p(\boldsymbol{\Omega} | \mathbf{y}_{T_1,1}, \dots, \mathbf{y}_{T_1, n_{T_1}}, \widehat{\Theta}_S) \cdot p(\mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2, n_{T_2}} | \boldsymbol{\Omega}, \widehat{\Theta}_S) d\boldsymbol{\Omega} \right] + \right. \\ &\quad \left. (1 - \alpha) \cdot \left[\int \pi(\boldsymbol{\Omega}) \cdot p(\mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2, n_{T_2}} | \boldsymbol{\Omega}, \widehat{\Theta}_S) d\boldsymbol{\Omega} \right] \right\}. \end{aligned}$$

The first integral contains information from the T_1 posterior distribution whereas the second does not. As they are constant with respect to α , take the first integral to be k_1 and the second to be k_2 . The marginal mean of α is

$$\mathbb{E}(\alpha | \mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2, n_{T_2}}) = \frac{k_1 \{ \mathbb{E}_{\pi(\alpha)}(\alpha)^2 + \text{Var}_{\pi(\alpha)}(\alpha) \} + k_2 \{ \mathbb{E}_{\pi(\alpha)}(\alpha) - \mathbb{E}_{\pi(\alpha)}(\alpha)^2 - \text{Var}_{\pi(\alpha)}(\alpha) \}}{k_1 \mathbb{E}_{\pi(\alpha)}(\alpha) + k_2 \{ 1 - \mathbb{E}_{\pi(\alpha)}(\alpha) \}},$$

where $\mathbb{E}_{\pi(\alpha)}(\alpha)$ and $\text{Var}_{\pi(\alpha)}(\alpha)$ are the expected value and variance of α with respect to its prior distribution, respectively. From this, we can determine the behavior based on the choice of prior on α . A natural, uninformative choice of prior for α is the Uniform(0, 1), which gives the posterior mean

$$\mathbb{E}(\alpha | \mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2, n_{T_2}}) = \frac{\frac{2}{3}k_1 + \frac{1}{3}k_2}{k_1 + k_2}.$$

While the Metropolis-Hastings algorithm used to estimate the posterior distribution of $(\boldsymbol{\Omega}, \alpha)$ will explore the entire parameter space, this choice of prior gives a maximum marginal mean of α of $2/3$ if the T_1 data are informative to learning the posterior on T_2 or a minimum of $1/3$ if the T_1 data are not informative. The value of the posterior mean of α is thus determined by both the similarity between T_1 and T_2 and the prior $\pi(\alpha)$. A prior such as Beta($1/a$, $1/a$) for $a > 1$ will give posterior means closer to 0 and 1 for large values of a . This may be preferable in practice if the T_1 information is very likely (or unlikely) to be informative for T_2 .

This idea naturally extends to more target data sets as well. If there are ℓ target data sets, we take $\boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_\ell\}$ to be a weight vector where $\alpha_k \geq 0$ for $k = 1, \dots, \ell$ and $\sum_{k=1}^{\ell} \alpha_k = 1$. The

online posterior distribution for T_ℓ is given by

$$\begin{aligned}
& p(\boldsymbol{\Omega}, \boldsymbol{\alpha} \mid \mathbf{y}_{T_\ell,1}, \dots, \mathbf{y}_{T_\ell, n_{T_\ell}}, \widehat{\boldsymbol{\Theta}}_S) \\
& \propto \pi(\boldsymbol{\alpha}) \left\{ \alpha_\ell \cdot \pi(\boldsymbol{\Omega}) + \sum_{k=1}^{\ell-1} \alpha_k \cdot p(\boldsymbol{\Omega} \mid \mathbf{y}_{T_k,1}, \dots, \mathbf{y}_{T_k, n_{T_k}}, \widehat{\boldsymbol{\Theta}}_S) \right\} \\
& \prod_{i=1}^{n_{T_\ell}} \int_{\mathbb{R}^m} p(\mathbf{y}_{T_\ell, i} \mid \boldsymbol{\beta}_i, \widehat{\boldsymbol{\Theta}}_S) \cdot p(\boldsymbol{\beta}_i \mid \boldsymbol{\Omega}) d\boldsymbol{\beta}_i,
\end{aligned}$$

where $\pi(\boldsymbol{\alpha})$ is now a multivariate prior on $\boldsymbol{\alpha}$, for example a Dirichlet distribution. An analogous derivation for the posterior means of the elements of $\boldsymbol{\alpha}$ for the multivariate prior is provided in Appendix A.11.

3.6 Simulation Study

3.6.1 Overview

In this section we examine the performance of the multivariate and online RECaST methods through simulations on synthetic data. We consider two methods for comparison. The first is a group ridge regression model that is built only on the target data set implemented with the `glmnet` package in R (Friedman et al. 2010). The second is the univariate RECaST method where a posterior is fit on each outcome separately. This will serve as validation to determine whether the multivariate RECaST methods improve predictive performance and uncertainty quantification.

The source model for the RECaST Cauchy and RECaST copula methods is a group ridge regression fit to the source data. The Mahalanobis distance $d(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) = \sqrt{(\tilde{\mathbf{y}} - \hat{\mathbf{y}})^\top \mathbf{S}^{-1} (\tilde{\mathbf{y}} - \hat{\mathbf{y}})}$ measures the distance between the test values $\tilde{\mathbf{y}}$ and the predicted values $\hat{\mathbf{y}}$, where \mathbf{S} is the empirical covariance of the test set. The test sets are of size 100. The posterior predictive sampling values are $n_{\text{post}} = 50$, $n_B = 50$, and $n_Y = 20$, and the posterior predictive sample is of size 50,000 for each test observation. This sample is used to calculate the empirical coverage and the Mahalanobis distance. Each simulation setting is repeated over 100 target test data sets. The Mahalanobis distances and empirical coverage values reported are averages over the 100 target test data sets.

The source sample size is large at $n_S = 1,000$. The data are generated from a multivariate linear regression with fixed source data generating parameters $\boldsymbol{\Theta}_S$ with $p = 50$ features (including an intercept) and $m = 2$ responses, $\mathbf{Y}_S \sim \mathcal{N}_p(\boldsymbol{\Theta}_S \mathbf{x}_S, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma}$ is the covariance between the responses. The features are generated from the standard Gaussian distribution,

$$\mathbf{x}_S \sim \mathcal{N}_{p-1}(\mathbf{0}, \mathbf{I}_{p-1}).$$

3.6.2 Multivariate Single Target Simulations

While the source sample size $n_S = 1,000$ is fixed, the target sample sizes vary: $n_T \in \{20, 50, 100\}$. With the data generating mechanism of linear regression, the ridge baseline model built only on the target data should perform well with 100 target data points. This provides a good test for negative transfer since the ridge baseline should have enough data to reliably estimate the true data generating parameters. With a sample size of 50, there are as many data points in the target as there are features and for a sample size of 20, there are fewer data points than features. With such little data, it is expected that target only models would perform poorly and that the inclusion of source information could be beneficial. We also vary the covariance between the outcomes Σ used to generate data: no covariance $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, positive covariance $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$, and negative covariance $\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$.

We first consider an additive relationship between the source and target data generating parameters. The source parameters for the first response $\Theta_{S,1}$, are fixed between 0.5 and 5 and the parameters for the second response $\Theta_{S,2}$, are fixed between -0.5 and -5 . The target data generating parameters are constructed by adding uniformly distributed noise to Θ_S . The target parameters for the first response are generated as $\Theta_{T,1} = \Theta_{S,1} + U(0, a)$ and the parameters for the second response are generated as $\Theta_{T,2} = \Theta_{S,2} + U(b, 0)$. This controls the similarity between the source and the target data: as a increases or b decreases, the target data generating parameters can grow further from the source parameters. We consider values of $(a, b) \in \{(0.5, -0.5), (1, -1), (2, -2)\}$.

Tables 3.1, 3.2, and 3.3 summarize the performance results stratified by Σ . RECaST using a multivariate Cauchy and a multivariate Gaussian copula is presented as MV Cauchy and MV Copula, respectively. In all cases, the multivariate RECaST methods have better predictive performance than the univariate method, demonstrating the benefit of considering the outcomes jointly. The multivariate methods also boast lower standard errors than univariate RECaST. This shows that the multivariate methods have more concentrated predictions. Across every scenario, both multivariate RECaST methods outperform the baseline ridge when $n_T = 20$ or 50 as shown by the lower Mahalanobis distances. As expected, as the target sample size increases the baseline ridge method improves. In fact, ridge has lower Mahalanobis distances when $n_T = 100$, indicating negative transfer. Even with a large sample size, when the source and target data sets are similar, $(a, b) = (0.5, -0.5)$, the Mahalanobis distances for the ridge and RECaST methods are close (in some cases less than a standard error apart) indicating less negative transfer. In contrast, when the sample size is large ($n_T = 100$) and the source and target are the

most dissimilar, $(a, b) = (2, -2)$, negative transfer is most apparent.

There is slight improvement in performance when the covariance between the outcomes is negative compared to when they are independent shown by the decrease in Mahalanobis distances. There is a similar slight decrease in performance when the responses are positively correlated. For every scenario, the multivariate RECaST methods achieve excellent empirical coverage of 96% or 97% at the 95% nominal level. The marginal coverages from univariate RECaST are often 100, which is a result of the entire outcome space being in the credible set. In these cases the credible set is uninformative at the 95% level.

Table 3.1: Performance metrics for $\Theta_{T,1,\cdot} = \Theta_{S,1,\cdot} + U(0, a)$ and $\Theta_{T,2,\cdot} = \Theta_{S,2,\cdot} + U(b, 0)$ with $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The reported values are averaged over 100 source and target data sets: average Mahalanobis distance (standard error of Mahalanobis distances) [empirical coverage at the 95% nominal level]. For univariate RECaST the marginal coverages are reported.

a, b	n_T	Ridge	Univariate	MV Cauchy	MV Copula
0.5, -0.5	20	1.2 (0.23)	1.1 (0.19) [100, 100]	0.18 (0.032) [97]	0.18 (0.032) [97]
	50	0.48 (0.09)	1.1 (0.14) [100, 100]	0.17 (0.022) [97]	0.17 (0.022) [97]
	100	0.21 (0.02)	1 (0.1) [99, 99]	0.16 (0.016) [97]	0.16 (0.015) [97]
1, -1	20	1.2 (0.24)	1.1 (0.2) [99, 100]	0.29 (0.051) [97]	0.29 (0.052) [97]
	50	0.47 (0.09)	1.1 (0.15) [100, 100]	0.27 (0.039) [97]	0.27 (0.039) [97]
	100	0.2 (0.02)	1 (0.11) [100, 100]	0.27 (0.032) [97]	0.27 (0.031) [97]
2, -2	20	1.2 (0.25)	1.1 (0.21) [99, 100]	0.5 (0.096) [97]	0.49 (0.09) [97]
	50	0.44 (0.09)	1.1 (0.15) [100, 100]	0.48 (0.078) [97]	0.47 (0.078) [97]
	100	0.18 (0.02)	1.1 (0.11) [100, 100]	0.47 (0.067) [97]	0.46 (0.065) [97]

Table 3.2: Performance metrics for $\Theta_{T,1,\cdot} = \Theta_{S,1,\cdot} + U(0, a)$ and $\Theta_{T,2,\cdot} = \Theta_{S,2,\cdot} + U(b, 0)$ with $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$. The reported values are averaged over 100 source and target data sets: average Mahalanobis distance (standard error of Mahalanobis distances) [empirical coverage at the 95% nominal level]. For univariate RECaST the marginal coverages are reported.

a, b	n_T	Ridge	Univariate	MV Cauchy	MV Copula
0.5, -0.5	20	1.2 (0.23)	1.1 (0.19) [100, 100]	0.19 (0.035) [97]	0.19 (0.035) [97]
	50	0.42 (0.09)	1.1 (0.14) [99, 100]	0.18 (0.023) [97]	0.18 (0.023) [97]
	100	0.16 (0.02)	1 (0.1) [99, 99]	0.18 (0.017) [97]	0.18 (0.017) [97]
1, -1	20	1.2 (0.24)	1.1 (0.2) [99, 100]	0.3 (0.052) [97]	0.29 (0.051) [97]
	50	0.42 (0.09)	1.1 (0.14) [100, 100]	0.28 (0.04) [97]	0.28 (0.039) [97]
	100	0.15 (0.02)	1 (0.1) [100, 100]	0.27 (0.032) [96]	0.27 (0.031) [96]
2, -2	20	1.2 (0.25)	1.1 (0.19) [100, 100]	0.5 (0.095) [96]	0.5 (0.095) [97]
	50	0.4 (0.09)	1.1 (0.14) [100, 100]	0.48 (0.078) [97]	0.48 (0.078) [97]
	100	0.14 (0.02)	1.1 (0.11) [100, 100]	0.47 (0.067) [97]	0.47 (0.067) [97]

Table 3.3: Performance metrics for $\Theta_{T,1,\cdot} = \Theta_{S,1,\cdot} + U(0, a)$ and $\Theta_{T,2,\cdot} = \Theta_{S,2,\cdot} + U(b, 0)$ with $\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$. The reported values are averaged over 100 source and target data sets: average Mahalanobis distance (standard error of Mahalanobis distances) [empirical coverage at the 95% nominal level]. For univariate RECaST the marginal coverages are reported.

a, b	n_T	Ridge	Univariate	MV Cauchy	MV Copula
0.5, -0.5	20	1.2 (0.23)	1.1 (0.19) [100, 100]	0.17 (0.03) [97]	0.16 (0.03) [97]
	50	0.54 (0.09)	1.1 (0.14) [99, 100]	0.16 (0.02) [97]	0.15 (0.019) [97]
	100	0.24 (0.03)	1 (0.1) [99, 99]	0.15 (0.014) [97]	0.15 (0.014) [97]
1, -1	20	1.2 (0.24)	1.1 (0.2) [99, 99]	0.28 (0.051) [97]	0.28 (0.052) [97]
	50	0.525 (0.09)	1.1 (0.15) [100, 100]	0.27 (0.038) [97]	0.26 (0.037) [97]
	100	0.24 (0.03)	1 (0.11) [100, 100]	0.26 (0.032) [97]	0.26 (0.031) [97]
2, -2	20	1.2 (0.25)	1.1 (0.2) [99, 99]	0.46 (0.067) [97]	0.46 (0.065) [97]
	50	0.48 (0.09)	1.1 (0.15) [100, 100]	0.47 (0.078) [97]	0.47 (0.077) [97]
	100	0.21 (0.03)	1.1 (0.11) [100, 100]	0.46 (0.067) [97]	0.46 (0.065) [97]

We next consider a multiplicative relationship between the source and target data generating parameters. Our source parameters remain the same as above, but now the target

parameters are constructed as $\Theta_T = c \cdot \Theta_S$ with $c \in \{0.5, 2\}$. The data are generated as above. Table 3.4 summarizes the results. As before, the baseline ridge regression performance improves as the sample size increases; however, notice that for $n_T = 100$ both multivariate RECaST methods have lower Mahalanobis distances for both $c = 0.5$ and $c = 2$ than the baseline ridge. In this setting, the multivariate RECaST methods are robust to negative transfer. They also outperform univariate RECaST. Here we see a greater improvement in predictive performance for both multivariate RECaST methods when the results are positively correlated than in the previous simulation. Both methods again provide excellent coverage slightly above the 95% nominal level.

Table 3.4: Performance metrics for $\Theta_T = c \cdot \Theta_S$. The reported values are averaged over 100 source and target data sets: average Mahalanobis distance (standard error of Mahalanobis distances) [empirical coverage at the 95% nominal level]. For univariate RECaST, the marginal coverages are reported.

Σ	c	n_T	Ridge	Univariate	MV Cauchy	MV Copula
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	0.5	20	1.2 (0.23)	1.1 (0.21) [100, 100]	0.23 (0.046) [97]	0.22 (0.045) [97]
		50	0.74 (0.12)	1 (0.13) [99, 99]	0.21 (0.022) [97]	0.21 (0.022) [97]
		100	0.39 (0.04)	1 (0.1) [98, 99]	0.2 (0.02) [97]	0.2 (0.019) [97]
	2	20	1.2 (0.24)	1 (0.2) [100, 100]	0.06 (0.012) [97]	0.06 (0.011) [97]
		50	0.36 (0.08)	1 (0.13) [99, 99]	0.06 (0.0061) [97]	0.06 (0.0061) [97]
		100	0.11 (0.012)	1 (0.1) [98, 99]	0.06 (0.0053) [96]	0.06 (0.0053) [97]
$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$	0.5	20	1.2 (0.24)	1 (0.21) [100, 100]	0.25 (0.051) [97]	0.25 (0.053) [97]
		50	0.6 (0.1)	1 (0.13) [99, 99]	0.23 (0.026) [97]	0.23 (0.026) [97]
		100	0.3 (0.03)	1 (0.1) [98, 99]	0.23 (0.024) [97]	0.23 (0.024) [97]
	2	20	1.2 (0.24)	1 (0.2) [100, 100]	0.07 (0.013) [97]	0.07 (0.0073) [97]
		50	0.34 (0.08)	1 (0.13) [99, 99]	0.07 (0.0073) [97]	0.07 (0.0073) [97]
		100	0.09 (0.01)	1 (0.1) [98, 99]	0.07 (0.0064) [97]	0.07 (0.0065) [97]

3.6.3 Multivariate Online Simulations

We now consider two target data sets, T_1 and T_2 . The data are again generated from multivariate linear regressions with $p = 50$ features (including an intercept) and $m = 2$ outcomes. The source data generating parameters are fixed within the intervals $\Theta_{S,1.} \in [2, 2.5]$ and $\Theta_{S,2.} \in [-2.5, -2]$. The source sample size is $n_S = 1,000$. The target T_1 data generating parameters Θ_{T_1} are also fixed within the intervals $\Theta_{T_1,1.} \in [6.5, 7]$ and $\Theta_{T_1,2.} \in [-7, -6.5]$. We take the target T_1 sample

size to be large with $n_{T_1} = 1,000$. Thus, this analysis isolates the effects of the online results on target T_2 .

As before, we use sample sizes of $n_{T_2} \in \{20, 50, 100\}$. We test the same covariance structures as in Section 3.6.2 with the responses either having no covariance, positive covariance, or negative covariance. The data generating parameters for the domain of interest, T_2 , are based on the other target data set T_1 . We take $\Theta_{T_2,1,\cdot} = \Theta_{T_1,1,\cdot} + U(0, a)$ and $\Theta_{T_2,2,\cdot} = \Theta_{T_1,2,\cdot} + U(b, 0)$. We consider two scenarios, the first with $a = 0$ and $b = 0$ and the second with $a = 3$ and $b = -3$. The first corresponds to the T_1 and T_2 data coming from close underlying distributions. In this case, we expect the multivariate, online methods to perform well because the posterior distribution from T_1 should be informative. In the second scenario, the target data sets are not coming from close distributions as the data generating parameters could be substantially different.

We compare the multivariate RECaST methods to a baseline group ridge model fit only on the T_2 data and to univariate RECaST. For all RECaST methods, we use a group ridge model fit on the source data as the source model. The multivariate RECaST methods we consider in this analysis are:

- multivariate RECaST using the multivariate Cauchy distribution learned on T_2 (MV Cauchy),
- multivariate, online RECaST using a multivariate Cauchy distribution learned on T_2 with a prior informed by the posterior fit on T_1 (MV-On Cauchy),
- multivariate RECaST using the multivariate Normal copula learned on T_2 (MV Copula),
- multivariate, online RECaST using a multivariate Normal copula learned on T_2 with a prior informed by the posterior fit on T_1 (MV-On Copula).

A standard uniform distribution is used as the prior on the weight parameter α .

Tables 3.5, 3.6, and 3.7 summarize the predictive performance results for the multivariate, online simulations stratified by Σ . All of the multivariate RECaST methods have better predictive performance univariate RECaST. They also had lower Mahalanobis distances than the ridge baseline when $n_{T_2} = 20$. This is also the case when $n_{T_2} = 50$ except when there is negative covariance between the outcomes and Θ_{T_2} if far from Θ_{T_1} . When $n_{T_2} = 100$ the effects of negative transfer are clear as the baseline ridge method outperforms all RECaST methods.

We do not see any effects of negative transfer between T_1 and T_2 , even when Θ_{T_2} if far from Θ_{T_1} . This is evident by comparing the MV Cauchy and MV-On Cauchy Mahalanobis distances and the MV Copula and MV-On Copula Mahalanobis distances. In both cases the MV-On method performs as well or better, indicated by an equal or smaller Mahalanobis distance, than

the corresponding offline method. This is even the case when Θ_{T_2} is far from Θ_{T_1} . This lends merit to the idea of using a convex combination of priors to mitigate negative transfer. The MV-On Cauchy has the best performance shown by the lowest Mahalanobis distances. This difference between methods is most apparent when the T_2 sample sizes are small; this is where transfer learning would be the most necessary as there is the least amount of information provided in the target domain of interest.

Tables 3.8, 3.9, and 3.10 present the empirical coverage values at the 95% nominal level for the RECaST methods, stratified by Σ . In all cases, the MV Cauchy, MV-On Cauchy, and MV Copula RECaST methods achieve empirical coverage of at least 95%. This is true even when the Θ_{T_2} is far from Θ_{T_1} . The MV-On Copula method under-covers slightly when sample sizes are larger, but achieves nominal coverage when transfer learning is most necessary, when $n_{T_2} = 20$. In most cases, univariate RECaST provides marginal coverages of 100, indicating that the credible sets are not helpful at the 95% nominal level.

For the online methods, Tables 3.11, 3.12, and 3.13 summarize the average estimated α weights, stratified by Σ . For the multivariate Cauchy method, the posterior mean of α is centered around 0.67 when Θ_{T_2} is close to Θ_{T_1} . Since a uniform distribution was used as the prior for α , this is the expected mean when the posterior for T_1 is more useful than an uninformative prior. When Θ_{T_2} is far from Θ_{T_1} , more information is shared when sample sizes are small. As n_{T_2} increases, the information from T_1 does not need to be relied upon as much and the posterior mean of α decreases. The copula-based method tends to borrow the same amount of information from T_1 in all settings except when there is negative correlation between the outcomes.

Table 3.5: Mahalanobis distances for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The reported values are: Average Mahalanobis distance (standard error of Mahalanobis distances)

a, b	n_{T_2}	Ridge	Univariate	MV Cauchy	MV-On Cauchy	MV Copula	MV-On Copula
0, 0	20	1.2 (0.23)	1 (0.20)	0.67 (0.16)	0.62 (0.11)	0.65 (0.13)	0.64 (0.12)
	50	0.94 (0.15)	0.96 (0.14)	0.61 (0.08)	0.59 (0.08)	0.59 (0.08)	0.59 (0.08)
	100	0.52 (0.06)	0.92 (0.10)	0.58 (0.07)	0.58 (0.07)	0.61 (0.18)	0.61 (0.19)
3, -3	20	1.2 (0.26)	12 (2.6)	0.97 (0.22)	0.93 (0.18)	1.04 (0.26)	1.03 (0.25)
	50	0.46 (0.1)	11 (1.7)	0.87 (0.11)	0.87 (0.11)	0.86 (0.11)	0.86 (0.11)
	100	0.18 (0.03)	11 (1.3)	0.83 (0.1)	0.83 (0.1)	0.83 (0.1)	0.83 (0.1)

Table 3.6: Mahalanobis distances for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$. The reported values are: Average Mahalanobis distance (standard error of Mahalanobis distances)

a, b	n_{T_2}	Ridge	Univariate	MV Cauchy	MV-On Cauchy	MV Copula	MV-On Copula
0, 0	20	1.3 (0.22)	1 (0.19)	0.71 (0.13)	0.67 (0.11)	0.7 (0.12)	0.7 (0.13)
	50	1.1 (0.16)	0.95 (0.14)	0.67 (0.1)	0.66 (0.09)	0.66 (0.09)	0.66 (0.09)
	100	0.62 (0.07)	0.93 (0.11)	0.64 (0.07)	0.63 (0.07)	0.64 (0.07)	0.64 (0.08)
3, -3	20	1.2 (0.26)	12 (2.5)	0.98 (0.23)	0.93 (0.19)	1.02 (0.24)	1.02 (0.23)
	50	0.51 (0.1)	11 (1.6)	0.86 (0.11)	0.86 (0.12)	0.86 (0.11)	0.86 (0.11)
	100	0.22 (0.03)	11 (1.3)	0.84 (0.1)	0.84 (0.1)	0.83 (0.1)	0.83 (0.1)

Table 3.7: Mahalanobis distances for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$. The reported values are: Average Mahalanobis distance (standard error of Mahalanobis distances)

a, b	n_{T_2}	Ridge	Univariate	MV Cauchy	MV-On Cauchy	MV Copula	MV-On Copula
0, 0	20	1.2 (0.24)	1 (0.20)	0.61 (0.13)	0.56 (0.1)	0.58 (0.12)	0.58 (0.12)
	50	0.76 (0.13)	0.95 (0.14)	0.53 (0.07)	0.53 (0.07)	0.53 (0.07)	0.52 (0.07)
	100	0.4 (0.04)	0.92 (0.1)	0.53 (0.05)	0.52 (0.05)	0.53 (0.09)	0.51 (0.11)
3, -3	20	1.2 (0.26)	12 (2.6)	0.98 (0.22)	0.92 (0.18)	1.04 (0.27)	1.04 (0.27)
	50	0.4 (0.1)	11 (1.7)	0.87 (0.12)	0.87 (0.12)	0.86 (0.11)	0.86 (0.11)
	100	0.14 (0.02)	11 (1.3)	0.83 (0.1)	0.83 (0.1)	0.83 (0.1)	0.83 (0.1)

Table 3.8: Empirical coverage values at the 95% nominal level for multivariate, online RECaST and empirical coverage values averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. For univariate RECaST the marginal coverages are reported.

a, b	n_{T_2}	Univariate	MV Cauchy	MV-On Cauchy	MV Copula	MV-On Copula
0, 0	20	100, 100	97	97	97	96
	50	100, 100	96	97	97	94
	100	100, 100	96	97	97	93
3, -3	20	96, 95	96	96	97	95
	50	99, 100	96	97	97	94
	100	100, 100	97	97	97	93

Table 3.9: Empirical coverage values at the 95% nominal level for multivariate, online RECaST and empirical coverage values averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$. For univariate RECaST the marginal coverages are reported.

a, b	n_{T_2}	Univariate	MV Cauchy	MV-On Cauchy	MV Copula	MV-On Copula
0, 0	20	100, 100	97	96	96	97
	50	100, 100	96	97	97	94
	100	100, 100	96	96	97	94
3, -3	20	96, 95	95	96	97	95
	50	99, 100	96	97	97	94
	100	100, 100	97	97	97	93

Table 3.10: Empirical coverage values at the 95% nominal level for multivariate, online RECaST and empirical coverage values averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$. For univariate RECaST the marginal coverages are reported.

a, b	n_{T_2}	Univariate	MV Cauchy	MV-On Cauchy	MV Copula	MV-On Copula
0, 0	20	100, 100	96	96	97	96
	50	100, 100	96	97	97	95
	100	100, 100	97	97	97	94
3, -3	20	95, 95	96	96	97	95
	50	99, 100	96	96	97	94
	100	100, 100	97	97	97	93

Table 3.11: Posterior means of α for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$: average posterior mean (standard deviation of the posterior means).

a, b	n_{T_2}	MV-On Cauchy	MV-On Copula
0, 0	20	0.67 (0.0028)	0.46 (0.15)
	50	0.67 (0.0027)	0.47 (0.15)
	100	0.67 (0.0031)	0.45 (0.15)
3, -3	20	0.62 (0.12)	0.47 (0.15)
	50	0.41 (0.14)	0.47 (0.15)
	100	0.34 (0.057)	0.46 (0.15)

Table 3.12: Posterior means of α for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$: average posterior mean (standard deviation of the posterior means).

a, b	n_{T_2}	MV-On Cauchy	MV-On Copula
0, 0	20	0.67 (0.0027)	0.51 (0.15)
	50	0.67 (0.0026)	0.47 (0.15)
	100	0.67 (0.0032)	0.47 (0.15)
3, -3	20	0.59 (0.14)	0.47 (0.15)
	50	0.38 (0.12)	0.47 (0.14)
	100	0.35 (0.066)	0.45 (0.14)

Table 3.13: Posterior means of α for multivariate, online RECaST averaged over 100 source and target data sets for $\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$: average posterior mean (standard deviation of the posterior means).

a, b	n_{T_2}	MV-On Cauchy	MV-On Copula
0, 0	20	0.67 (0.0027)	0.34 (0.11)
	50	0.67 (0.0029)	0.37 (0.1)
	100	0.67 (0.0029)	0.39 (0.12)
3, -3	20	0.64 (0.091)	0.38 (0.11)
	50	0.43 (0.15)	0.41 (0.13)
	100	0.36 (0.096)	0.39 (0.12)

3.7 Dental Data Analysis

Difficult measurements need to be taken at each tooth to diagnose periodontitis. A reliable model for these outcomes would help alleviate the burden on dentists and provide additional information to inform their decision making.

Participants included in the data set had at least 8 years of continuous dental insurance during the study period. We consider two periodontal measurements measured in millimeters (mm): clinical attachment level (CAL) and pocked depth (PD). PD is the “distance from the gingival margin to the base of the gingival sulcus or periodontal pocket” and CAL is the “distance from the cemento-enamel junction (or another definite chosen landmark) to the base of the sulcus or periodontal pocket” (Page and Eke 2007). These are both important measurements in diagnosing periodontitis. A general categorization of periodontitis is: *slight* = 1 to 2 mm of CAL, *moderate* = 3 to 4 mm of CAL, and *severe* ≥ 5 mm of CAL (AAP 2015). Figure 3.1 shows the skewness in the outcome measurements; CAL has a range of [0, 7.75] and PD has a range of [0, 6.2]. These measurements are *whole-mouth averages*, averaged over all of each participant’s teeth.

Our goal is to be able to predict CAL and PD based on a number of demographic, general health, and insurance variables. These outcomes are time consuming and challenging to measure; being able to predict them would aid in predicting gum disease. To achieve this we take the information of each participant at their first visit. There are 23,529 participants considered. There is a large disparity in the racial groups represented in the study. Of those participants 21,029 are White, 1,297 are Black or African American, 1,050 are Asian, and 153 are Native American or Alaskan Native. Table 3.14 shows the breakdown of the measured features and outcomes by demographic group. As we saw in the literature in Section 3.2, using models

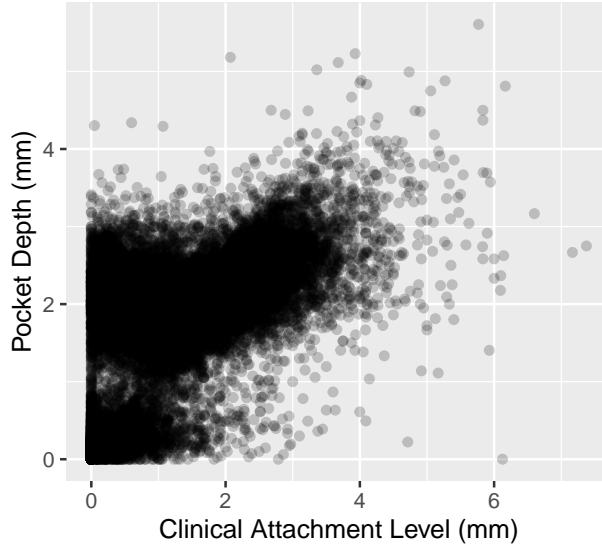


Figure 3.1: The outcome measurements for each participant’s first visit.

built on a majority population to make predictions on underrepresented groups can produce poor results. To address this, we use these groups to split the data set into source (White) and three target (Black or African American, Asian, and Native American or Alaska Native) data sets.

We perform 10-fold cross validation to split the data into training and testing for the target data set of interest T_2 . The reported metrics are those calculated on the held out testing data while the training data are used to learn the RECaST posterior distribution or the baseline ridge parameter estimates. We use 1 fold as the training and the remaining 9 as testing. This process is repeated 10 times, until all of the target data have been used for training exactly once. Note that the T_1 data is not split and all of it is used to learn the posterior distribution for T_1 . The posterior predictive sampling values are set as $n_{\text{post}} = 50$, $n_B = 100$, and $n_Y = 20$. We perform three analyses corresponding to different target data sets to explore the relationship between racial groups. This will help leverage the most similar data to build reliable target models.

Table 3.14: HP Data feature and outcome summaries stratified by racial group. For age, CAL, and PD the 25th, 50th, and 75th percentiles are presented.

	White	Black or African American	Asian	Native American or Alaska Native
<i>n</i>	21,029	1,297	1,050	153
Age [25%, 50%, 75%]	[46, 55, 67]	[37, 45, 54]	[36, 44.5, 57]	[41, 51, 59]
Female	12,577(59.8%)	744(57.4%)	617(58.8%)	102(66.7%)
Diabetes				
Not indicated	19,460(92.5%)	1,148(88.5%)	959(91.3%)	134(87.6%)
Type 1	147(0.7%)	17(1.3%)	7(0.7%)	2(1.3%)
Type 2	1,422(6.8%)	132(10.2%)	84(8%)	17(11.1%)
Tobacco Use				
Never	16,909(80.4%)	1,062(81.9%)	946(90.1%)	114(74.5%)
Former	2,200(10.5%)	89(6.9%)	37(3.5%)	21(13.7%)
Current	1,920(9.1%)	146(11.3%)	67(6.4%)	18(11.8%)
Brushing				
Not indicated	4,277(20.3%)	231(17.8%)	184(17.5%)	28(18.3%)
Daily	16,635(79.1%)	1,057(81.5%)	863(82.2%)	124(81%)
Weekly	96(0.5%)	8(0.6%)	3(0.3%)	1(0.7%)
Less than weekly	21(0.1%)	1(0.1%)	0(0%)	0(0%)
Flossing				
Not indicated	6,672(31.7%)	466(35.9%)	331(31.5%)	49(32%)
Daily	7,321(34.8%)	404(31.1%)	434(41.3%)	50(32.7%)
Weekly	4,583(21.8%)	229(17.7%)	174(16.6%)	39(25.5%)
Less than Weekly	2,453(11.7%)	198(15.3%)	111(10.6%)	15(9.8%)
Insurance				
Commercial (C)	16,186(77%)	1029(79.3%)	806(76.8%)	125(81.7%)
Government Subsidized (G)	2,594(12.3%)	232(17.9%)	213(20.3%)	22(14.4%)
Private Pay (P)	67(0.3%)	0(0%)	3(0.3%)	1(0.7%)
C + G	312(1.5%)	13(1%)	1(0.1%)	1(0.7%)
C + P	34(0.2%)	1(0.1%)	0(0%)	0(0%)
G + P	2,594(12.3%)	232(17.9%)	213(20.3%)	22(14.4%)
All	10(<0.1%)	0(0%)	1(0.1%)	0(0%)
CAL [25%, 50%, 75%]	[0.611, 1.518, 2.086]	[0.625, 1.766, 2.375]	[1, 1.821, 2.378]	[0.406, 1.413, 1.981]
PD [25%, 50%, 75%]	[1.552, 1.984, 2.302]	[2.032, 2.379, 2.745]	[1.763, 2.135, 2.458]	[1.625, 2.069, 2.431]

The first analysis uses the data from the Black or African American participants as T_1 and the Asian participants as the domain of interest T_2 . Based on the cross validation scheme, this gives a training size of 105 for each fold; there are many more training data points than features. Table 3.15 reports the predictive and coverage results. We see that all of the multivariate transfer learning methods boast lower Mahalanobis distances when compared to the target only ridge baseline. Univariate RECaST performs worse than both the baseline and the other transfer learning methods. This again demonstrates the importance of considering the outcomes jointly. All of the multivariate methods achieve desirable empirical coverage values slightly above the nominal 95% level. Univariate RECaST also achieves near nominal marginal coverage on the PD outcome, but an empirical coverage of 100% on CAL indicating that at the 95% nominal level the credible set contains the entire outcome space.

Interestingly, the online methods have posterior means of α centered on opposite ends of the spectrum. Since a uniform prior was used for α , an average posterior mean of 0.67 corresponds to the online multivariate Cauchy method putting significant weight on the prior informed by the model built using the Black or African American participants. An average posterior mean of 0.33 for the online copula-based method shows that it minimized the use of

Table 3.15: Performance metrics averaged over 10-fold cross validation. We report the joint empirical coverage at the 95% nominal level; for univariate RECaST the marginal coverages are reported. For the online methods we report the posterior mean of α averaged across the fold. The source population is the White participants. The target of interest is the Asian participants where 10% of the data was used for training in each cross validation fold ($n_{T_2} = 105$). For the online methods, the informative prior is created using the corresponding multivariate RECaST method with the Black or African American participants.

Model	Mahalanobis Distance	95% Coverage	$\mathbb{E}(\alpha \mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2,n_{T_2}})$
Target Ridge	1.22 (0.07)		
Univariate	1.54 (0.11)	100, 97	
MV Cauchy	1.21 (0.06)	98	
MV-On Cauchy	1.21 (0.07)	98	0.67
MV Copula	1.22 (0.07)	97	
MV-On Copula	1.2 (0.06)	97	0.33

information from T_1 .

The next analysis uses the data from the Black or African American participants as T_1 and the Native American or Alaska Native participants as the domain of interest T_2 with a training size of 15 for each fold. Table 3.16 reports the predictive and coverage results where we see similar results as before. All of the multivariate RECaST methods outperform the ridge regression with smaller Mahalanobis distances. In this case, the MV-On Cauchy method has the best overall predictive performance. Again, all of the multivariate methods achieve desired slightly conservative joint coverage. Neither of the marginal coverages from univariate RECaST provide information at the 95% level.

Table 3.16: Performance metrics averaged over 10-fold cross validation. We report the joint empirical coverage at the 95% nominal level; for univariate RECaST the marginal coverages are reported. For the online methods we report the posterior mean of α averaged across the fold. The source population is the White participants. The target of interest is the Native American and Alaska Native participants where 10% of the data was used for training in each cross validation fold ($n_{T_2} = 15$). For the online methods, the informative prior is created using the corresponding multivariate RECaST method with the Black or African American participants.

Model	Mahalanobis Distance	95% Coverage	$\mathbb{E}(\alpha \mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2, n_{T_2}})$
Target Ridge	1.61 (0.64)		
Univariate	1.92 (0.35)	100, 100	
MV Cauchy	1.51 (0.25)	97	
MV-On Cauchy	1.46 (0.25)	98	0.67
MV Copula	1.5 (0.26)	97	
MV-On Copula	1.5 (0.26)	97	0.36

Finally, we use the data from the Asian participants as T_1 and the Native American or Alaskan Native participants as the domain of interest T_2 . Table 3.17 reports the predictive and coverage results. Again all multivariate RECaST methods provide better predictive performance than the baseline ridge. This data setting shows the biggest difference between the baseline and transfer learning methods with the MV-On Cauchy method again performing the best. Empirical joint 95% coverage is achieved by all of the multivariate RECaST methods; univariate RECaST again fails to provide useful uncertainty quantification for both CAL and PD.

Table 3.17: Performance metrics averaged over 10-fold cross validation. We report the joint empirical coverage at the 95% nominal level; for univariate RECaST the marginal coverages are reported. For the online methods we report the posterior mean of α averaged across the fold. The source population is the White participants. The target of interest is the Native American and Alaska Native participants where 10% of the data was used for training in each cross validation fold ($n_{T_2} = 15$). For the online methods, the informative prior is created using the corresponding multivariate RECaST method with the Asian participants.

Model	Mahalanobis Distance	95% Coverage	$\mathbb{E}(\alpha \mathbf{y}_{T_2,1}, \dots, \mathbf{y}_{T_2, n_{T_2}})$
Target Ridge	1.61 (0.64)		
Univariate	1.96 (0.79)	100, 100	
MV Cauchy	1.55 (0.58)	97	
MV-On Cauchy	1.47 (0.61)	97	0.67
MV Copula	1.5 (0.6)	97	
MV-On Copula	1.5 (0.6)	97	0.34

To summarize, all of the multivariate RECaST methods improve joint prediction of CAL and PD in underrepresented populations. The biggest improvements are seen on the Native American and Alaska Native population, which is the least represented population in the data. The online approaches give the most accurate predictions, demonstrating the importance of including related target data. All of the multivariate RECaST methods provided excellent joint coverage properties, which could provide clinicians both improved predictions and credible sets.

3.8 Concluding Remarks

Future work for this method includes considering multiple measurements per participant. Some dental data sets provide CAL and PD measurements for all of the teeth in a participant's mouth rather than a single averaged outcome (Guan et al. 2020). This could further expand into including modeling longitudinal data with multiple follow-up visits for each participant. With the inclusion of other dental measurements, the spatial relationship between teeth can also be used (Jhuang et al. 2020). Additionally, this multivariate framework can be applied to multiclass outcomes that are common for image classification (Deng et al. 2009).

CHAPTER

4

ADAPTIVE DISCRETIZATION FOR EVENT PREDICTION (ADEPT)

4.1 Introduction

Time to event modeling, also called survival analysis, is ubiquitous throughout clinical medicine as well as in many other fields concerned with predicting risk of events of interest (*e.g.*, clinical outcomes) based on available features (*e.g.*, patient characteristics). Traditional approaches include the well-known Cox proportional hazards (Cox-PH) model (Cox 1972), in which features modulate a baseline hazard rate; and the accelerated failure time (AFT) model (Wei 1992) model, in which features accelerate or decelerate a learned, parametric event time density.

Recently developed methods have focused on *a*) allowing effects of features on the hazard rate or event time density to be non-linear and flexible (Katzman et al. 2018; Ranganath et al. 2016; Kvamme et al. 2019; Miscouridou et al. 2018); and *b*) also allowing greater flexibility in the form of the event time density itself via approaches that *discretize* time, then predict the probability of event occurrence in each resulting *time interval* (Yu et al. 2011; Lee et al. 2018; Ren et al. 2019; Tjandra et al. 2021; Engelhard and Henao 2022).

The prognostic information provided by these models often has direct and significant impact on stakeholder decision-making. In a clinical setting, for example, information about

risk within a particular time interval might influence providers' or patients' decisions about whether to pursue treatment, or which specific treatment to pursue. It is therefore critical not only that predictions are accurate, but also that they are easily interpretable by stakeholders who wish to integrate them in decision-making. The predictions of a Cox-PH model might be presented to stakeholders as relative hazards, for instance, whereas it is natural to present the predictions of more recent models as the probability of event occurrence in a time interval of interest.

Importantly, however, decisions about these intervals made during model development – in other words, choices about the number and placement of *cut points* used to discretize the event time space – can have substantial impact on interpretability as well as performance. Equipped with unlimited data, we might use a large number of cut points to divide the timeline into tiny intervals; this would then allow us to summarize risk over an arbitrary time period of interest by combining predictions across all the intervals that comprise that period. However, the amount of data required to accurately estimate risk in each interval increases as the number of intervals increases, making this approach impractical even for large observational datasets. Equipped with unlimited time, on the other hand, we might present risk in a format most relevant to a particular patient, or to the decision at hand. Again, however, practical considerations typically require us to instead summarize risk over a consistent, limited number of time-frames (*e.g.*, 10-year risk, 5-year risk). In some cases a particular discretization is most actionable given the clinical context, but in others the choice is arbitrary, and it would be preferable to identify a discretization that facilitates more accurate prediction.

To illustrate the problem more concretely, consider the following example from the maternal health setting, which partly motivated this work. Patients with preeclampsia and gestational hypertension have substantially increased risk of postpartum cardiovascular events (Meng et al. 2022), but this risk can be mitigated by regular monitoring (*e.g.*, increased visits) of high-risk patients in the months after delivery. When developing a monitoring strategy, it is important to determine not only (a) which patients are at highest risk, but also (b) how long monitoring should take place; yet we have limited data available for learning because the outcome rates are low.

Our goal, therefore, is to develop a principled, data-driven approach to answer both of these questions. Specifically, we wish to develop a method that providers can use to identify time intervals that are optimal when *understanding* risk, for example to design an intervention or monitoring strategy, as well as when *reporting* risk to patients. At the same time, we wish to retain the substantial advantages and flexibility of other recently developed approaches, including their lack of strong parametric assumptions about the form of the event density. To solve this problem, we develop Adaptive Discretization for Event Prediction (ADEPT).

We begin by recasting learning from *discrete* survival times as learning from *continuous* survival times under the assumption that the density is piecewise constant; and then formulate a smooth relaxation of this piecewise constant density that allows cut points (*i.e.*, interval boundaries) to be learned by gradient-based optimization methods. We then present our learning procedure and results of experiments with two simulated and three real datasets – including a newly harmonized stroke risk prediction dataset that pools data across three large cohorts – that illustrate the effectiveness and potential clinical relevance of ADEPT.

Our performance evaluation focuses on comparing our method to its state of the art alternative, namely, discrete-time, neural network-based risk prediction over fixed-length intervals.

In summary, our contributions are as follows:

- Present ADEPT, a novel model and associated learning procedure to learn an optimal event time partition from data rather than fixing it *a priori*.
- Present simulation results illustrating effective learning of cut points that are consistent with the true, underlying generative model.
- Demonstrate improved prediction performance across three real datasets, including two clinical datasets.
- Identify clinically meaningful risk cut points illustrating the potential of the approach to provide improved prognostic information.

4.2 Methods

4.2.1 Setup and Notation

Consider a time-to-event outcome where each observation is represented by the triplet $\{\mathbf{X}, Y, S\}$, where $\mathbf{X} \in \mathcal{X} \in \mathbb{R}^p$ is a p -dimensional feature vector, $Y \in (0, T_{\max}]$ is an observed event time over a finite time horizon, and $S \in \{0, 1\}$ indicates whether Y is a right-censoring time ($S = 0$) or an event time ($S = 1$). The observed time Y is the minimum of the event time T and the right-censoring time U , *i.e.*, $Y = \min(T, U)$, and $S = \mathbb{I}(T < U)$, where the indicator function $\mathbb{I}(\cdot)$ is 1 when the argument is true and 0 otherwise.

We consider possible sequences of M *cut points* $C = \{c_j\}_{j=1}^M$, where $0 = c_0 < c_1 < \dots < c_M < c_{M+1} = T_{\max}$, that partition the event time space, $(0, T_{\max}]$, into the intervals I_1, \dots, I_{M+1} , where $I_j = (c_{j-1}, c_j]$. Figure 4.1 provides an example of the event time space partitioned into four intervals: $I_1 = (0, c_1]$, $I_2 = (c_1, c_2]$, $I_3 = (c_2, c_3]$, and $I_4 = (c_3, T_{\max}]$. Given such a partition, we

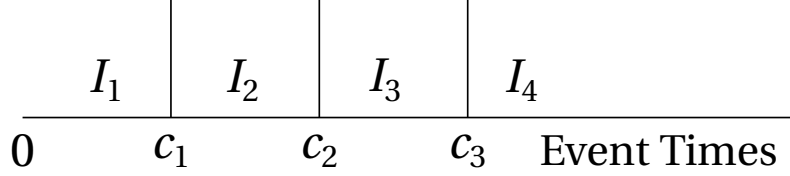


Figure 4.1: The event time space partitioned by three cut points into four intervals.

introduce an auxiliary random variable $Z \in \{1, \dots, M+1\}$ that indicates which interval contains T , *i.e.*, $Z = j \iff t \in I_j$.

4.2.2 Piecewise Constant Density

We begin by considering learning with fixed cut points, which is currently the predominant approach. For example, Lee et al. (2018) and other recently-developed methods (Ren et al. 2019; Tjandra et al. 2021; Engelhard and Henao 2022) use fixed cut points to discretize time in order to avoid placing restrictive, parametric assumptions on the form of the event time density. Instead, the density is restricted to be piecewise constant according to the intervals defined by the cut points. The cut points themselves might be evenly spaced in time, or alternatively they might be evenly spaced across the observed or estimated event time distribution, *e.g.*, via empirical quantiles. The goal of learning is then to estimate $P(Z|\mathbf{X})$, the conditional probability that T will fall in each of the pre-defined intervals, rather than $p(T|\mathbf{X})$, the conditional density of T . Typically T is discretized to Z *a priori*.

However, it is not possible to *learn* the cut points C with this approach, because Z depends on C in addition to T . To see this, consider the value of Z associated with an observed time $t \in (0, T_{\max}]$ under the binary partition defined by the single cut point c_1 . If we choose $c_1 \geq t$, then $t \in (0, c_1]$, therefore $Z = 1$; but for $c_1 < t$, we have $t \in (c_1, T_{\max}]$, therefore $Z = 2$.

To circumvent this limitation, we note that estimating $P(Z|\mathbf{X})$ is equivalent to estimating $p(T|\mathbf{X})$ with the following piecewise constant model, which supposes $p(T|\mathbf{X})$ has uniform density over each interval I_j :

$$\hat{p}(t|\mathbf{x}) = \sum_{j=1}^{M+1} p_{\phi}(z_j|\mathbf{x}) \frac{\mathbb{I}_{I_j}(t)}{|I_j|}, \quad (4.1)$$

where $\mathbb{I}_{I_j}(\cdot)$ is the indicator function associated with the interval I_j , and ϕ parameterizes our model of $P(Z|\mathbf{X})$. Importantly, we must normalize by $|I_j|$, the length of I_j , to ensure $\int_{(0, T_{\max})} \hat{p}(t|\mathbf{x}) = 1$ and $\int_{I_j} \hat{p}(t|\mathbf{x}) = p_{\phi}(z_j|\mathbf{x})$. As a potential drawback of this approach, we note that whereas standard discrete-time approaches are well suited to handle outlying event times, in principle this normalization term could cause the loss to become numerically unstable in

the case of extreme event time outliers.

4.2.3 Smooth Relaxation of Piecewise Density

The parameters ϕ of our model for Z can be learned directly from equation (4.1). However, our goal is to learn not only ϕ but also C , the specific partition that allows our model to best approximate $p(T|\mathbf{X})$ across a given dataset. Unfortunately, (4.1) cannot be optimized with respect to C via gradient-based methods. This is because the indicator function $\mathbb{I}_{I_j}(\cdot)$ implicitly depends on C , and is discontinuous whenever a cut point is equal to an observed event time.

To illustrate, consider learning a single cut point c_1 while holding the parameters ϕ fixed. For small ε such that $0 < \varepsilon < t$, where t is an observed event time associated with covariates \mathbf{x} , suppose the cut point $c_1 = t + \varepsilon$ is just after the observed event time. In this case, we have $t \in I_1$, therefore $\mathbb{I}_{I_1}(t) = 1$ and $\mathbb{I}_{I_2}(t) = 0$, and consequently $\hat{p}(t|\mathbf{x}) = p_\phi(z_1|\mathbf{x})/|I_1|$. On the other hand, suppose the cut point $c_1 = t - \varepsilon$ is just before the observed event time. In this case, we have $t \in I_2$, therefore $\mathbb{I}_{I_1}(t) = 0$ and $\mathbb{I}_{I_2}(t) = 1$, and consequently, $\hat{p}(t|\mathbf{x}) = p_\phi(z_2|\mathbf{x})/|I_2|$. Thus, for any non-trivial model p_ϕ for which $p_\phi(z_1|\mathbf{x}) \neq p_\phi(z_2|\mathbf{x})$, equation (4.1) is discontinuous at $c_1 = t$. This argument readily generalizes to all cut points.

To smooth this discontinuity and allow gradient-based optimization, we replace the indicator function $\mathbb{I}_{I_j}(t)$ in (4.1) with the smooth approximation $\sigma((t - c_{j-1})/\tau) * \sigma((c_j - t)/\tau)$, where $\sigma(z) = (1 + e^{-z})^{-1}$ is the sigmoid function. The temperature τ is a hyperparameter of the model that should be tuned based on the scale of the observed event times.

This results in the following relaxed model:

$$\hat{p}(t|\mathbf{x}) = \sum_{j=1}^{M+1} p_\phi(z_j|\mathbf{x}) \frac{\sigma(\frac{t-c_{j-1}}{\tau})\sigma(\frac{c_j-t}{\tau})}{|I_j|}, \quad (4.2)$$

which is approximately piecewise constant for $\tau \ll T_{\max}$, yet differentiable everywhere with respect to C and thus suitable for gradient-based optimization.

4.2.4 Learning Procedure

Under the common assumption of non-informative right-censoring, we may ignore the censoring density and optimize $\hat{p}(y, s|\mathbf{x}; \theta)$, where $\theta = \{\phi, C\}$, over the observed data $\mathcal{D} = \{\mathbf{x}_i, y_i, s_i\}_{i=1}^N$ as follows:

$$\theta = \arg \max_{\theta} \sum_i^N [s_i \log \hat{p}(t_i|\mathbf{x}; \theta) + (1 - s_i) \log \hat{P}(t_i > y_i|\mathbf{x}_i; \theta)], \quad (4.3)$$

where $\hat{P}(t_i > y_i | \mathbf{x}_i; \theta) = 1 - \int_0^T \hat{p}(\tau | \mathbf{x}_i; \theta)$ is the survival function associated with $\hat{p}(t_i | \mathbf{x}_i; \theta)$ for observation i .

However, optimizing equation (4.2) alone can result in degenerate solutions in which cut points become arbitrarily close together or even coincide. In the extreme case, it is possible to have $I_j = (0, T_{\max}]$ for a particular $j \in \{1, M + 1\}$, resulting in the trivial model in which $p_\phi(z | \mathbf{x})$ places all mass on z_j .

It is therefore critical to balance optimizing equation (4.2) versus ensuring that $p_\phi(z | \mathbf{x})$ is non-trivial. We accomplish this by incorporating a regularization term, $H(p_\phi(z | \mathbf{x}))$, with associated hyperparameter λ_1 in our optimization procedure. We use a scaled Beta(1.5, 1.5) distribution on each cut point. For example, suppose there are three cut points $c_1 < c_2^* < c_3$ where c_2^* is the newly proposed value for the middle cut point c_2 . We scale the value of the cut point to find its location relative to the cut points near it: $c_{2,\text{scaled}}^* = (c_2^* - c_1) / (c_3 - c_1)$. The final regularization value is the PDF value of $c_{2,\text{scaled}}^*$ evaluated over a Beta(1.5, 1.5) distribution. This regularization term encourages cut points to be near the center of their two surrounding cut points.

We may then optimize θ over \mathcal{D} by choosing $\theta = \arg \min_{\theta} \sum_{\mathcal{D}} \mathcal{L}(\theta)$, where $\mathcal{L}(\theta)$ is defined as follows:

$$\mathcal{L}(\theta) = -\log \hat{p}(y, s | \mathbf{x}; \theta) - \lambda_1 H(p_\phi). \quad (4.4)$$

Here the first term is the negative log likelihood in (4.3) and the second is our beta distribution regularizer. Our learning procedure then becomes:

$$\theta = \arg \min_{\theta} \sum_{\mathcal{D}} \mathcal{L}(\theta) + \lambda_2 R(\theta), \quad (4.5)$$

where we have included an additional regularization term $R(\cdot)$ (e.g., L_2 -regularization) along with an associated hyperparameter λ_2 to control for overfitting.

4.3 Implementation Details

4.3.1 Baseline Model: Discrete-Time Neural Network

We compare ADEPT to a discrete-time neural network baseline that is identical to the proposed model, except the cut points (and corresponding intervals) are initialized based on the observed outcomes and remain fixed when learning the classifier. This approach, hereafter called the *DTNN*, was popularized by DeepHit (Lee et al. 2018) and is currently the predominant approach. However, to isolate the effect of learning the partition we do not include the ranking loss used

in DeepHit . Our method and the DTNN have similar computational complexity, which is dominated by the computation of gradients with respect to model parameters θ rather than the cut points C . In this work, we instantiate p_θ as neural network, but our approach is flexible to the model choice; thus it can be changed based on application if, for example, interpretability is more important than predictive performance.

We initialize the DTNN model's cut points to be evenly spaced on the percentiles of the empirical Kaplan-Meier curve of the observed outcomes; for example, if there are three cut points then they would be placed at the time points associated with the 25th, 50th, and 75th percentiles on the estimated Kaplan-Meier curve. With these cut points fixed, we then build a model predicting the probability that the patient will experience the outcome in each interval. Note that this differs from ADEPT in which we also consider the cut points themselves as parameters. The DTNN classification model learns the probability of each observation being in each of the pre-defined intervals. In the notation of Section 4.2.2, the DTNN approach learns only the model parameters, ϕ_{baseline} , whereas ADEPT learns both model parameters ϕ and the cut points C . Importantly, we search the same grid of hyperparameters for the DTNN model as for ADEPT.

Due to its popularity, we also include a comparison to DeepSurv (Katzman et al. 2018). However, understanding differences between DeepSurv and other approaches is challenging due to their stark differences, notably the assumption of proportional hazards, which can either improve or worsen performance depending on the degree to which this assumption holds. Moreover, because DeepSurv does not incorporate discretization, it will have fewer comparative performance metrics.

4.3.2 Performance Quantification

With simulated data we were able to judge the correctness of the estimated cut points by their proximity to the true cut points used in the data generation mechanism. For the real data we do not know the true cut point values and thus need other metrics to judge performance. While we focus on several metrics quantifying predictive performance, clinical collaboration is necessary to determine which metrics are most relevant in a particular clinical context, including when implementing treatment decisions.

Time-Dependent Concordance Index (CI) Since we consider a time-to-event outcome with censored observations rather than a regression or classification outcome, standard metrics such as root mean square error and area under the receiver operating characteristic are insufficient to capture prediction performance. Initially developed by Harrell Jr et al. (1984), the concordance index (CI) measures how well predicted event times match the order of the

true event times. However, both ADEPT and the DTNN predict discrete interval membership rather than continuous event times, and the ordering of predicted risk can change over time. To properly account for these characteristics, we use a discrete-time implementation of the time-dependent concordance index developed by Antolini et al. (2005). This metric compares model-predicted risk at observed failure times to the model-predicted risks *at that time* for other individuals known to have later failure times. Pairs of individuals are only considered if (a) both failure times are known (neither are censored), or (b) one failure time is known to have occurred before the censoring time of the other.

AUC at last cut point The Area Under the Receiver Operating Characteristic Curve (AUC) is a common metric to evaluate predictive performance for a binary outcome. To adapt this to our method, we focus on the AUC at the last cut point. That is, we are interested in determining if the methods are able to predict whether an event happens before or after the final cut point. This is especially relevant for data sets with high amounts of censoring at the end of the study. The cases are all observations that experienced an event prior to the final cut point and the controls are all observations with an observed time (either an event or censored) after the final cut point. Notice that observations that are censored prior to the last cut point are omitted from this metric.

Integrated Brier Score (IBS) The Brier Score evaluated at time a chosen time t is the mean squared difference between the model-predicted cumulative event probability at time t and the true, binary, observation of whether the even occurred by t . The Integrated Brier Score improves upon this by integrating over all times $t \in \{t_{\min}, t_{\max}\}$ (Graf et al. 1999). An IBS of 0 indicates that the model was able to perfectly predict outcomes.

Calibration slope and intercept We also consider the calibration slope and intercept as described by Crowson et al. (2016), which quantify the degree to which model-predicted probabilities accurately estimate true event probabilities, as determined based on observed event rates. A well calibrated model will have a calibration slope near 1 and a calibration intercept near 0.

It is important to note that DeepSurv does not discretize the event time space; because of this, only the AUC and IBS metrics are included. We cannot compare the discrete-time concordance index to the standard concordance index because there are inevitably more ties with the discrete-time approach.

4.3.3 Hyperparameter Tuning

ADEPT is flexible, allowing for any number of cut points. In our simulation examples we will know exactly how many cut points were used to generate the data; however, this is not the case for the real data experiments. Thus, we use 3, 5, and 10 cut points. We use a two layer neural

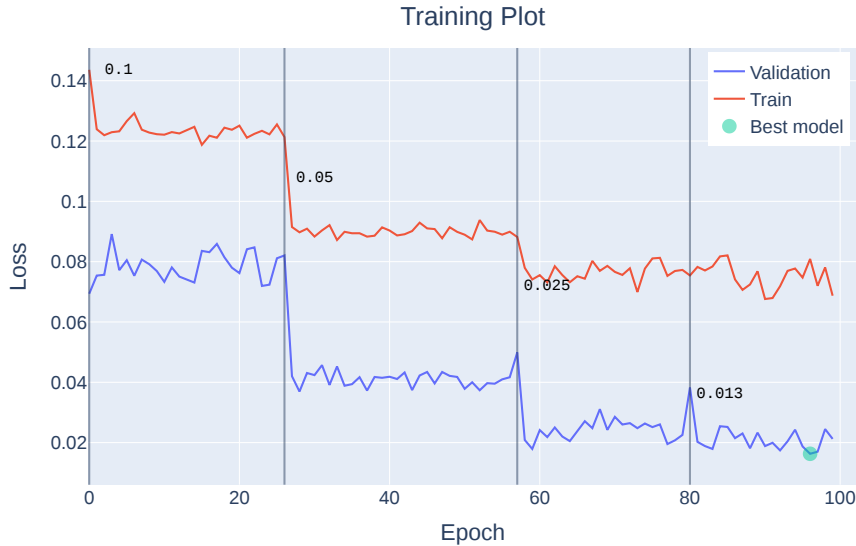


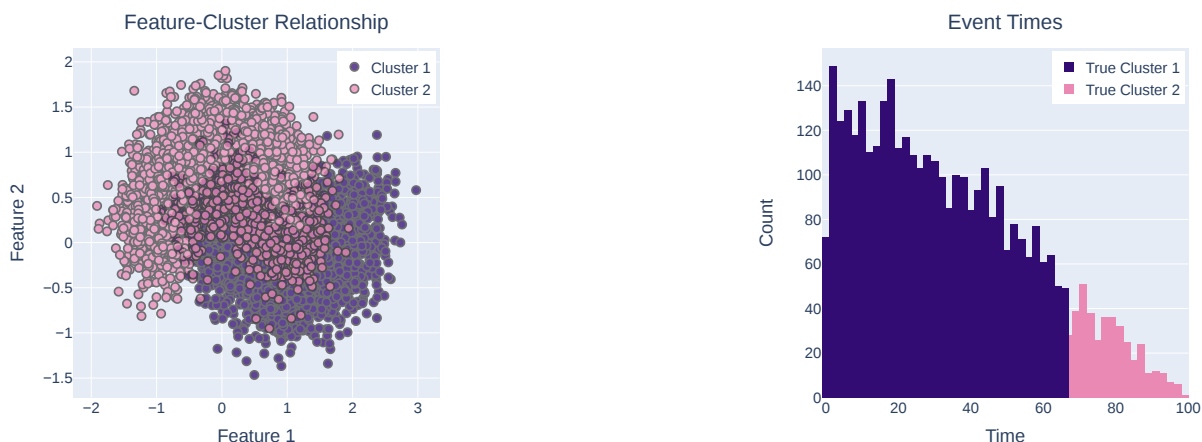
Figure 4.2: A training plot of the training and validation loss at each epoch for the two interval simulation example. The vertical lines represent drops in sigmoid temperature τ and the accompanying new value of τ .

network as our predictive model. The first layer has input dimension p based on the feature dimension of the data and output dimension h , for which we explore values of 32, 128, and 512. This is then connected by a Rectified Linear Unit activation function to another layer with input dimension h and output dimension. These networks are optimized using Adam (Kingma and Ba 2014) with a learning rate of 0.01 and weight decay values between 0.0001 and 0.1. We vary the strength of the regularization on the cut points λ_1 from values in the range of 0.1 to 20 and use a mini-batch size of 64 for the training data.

During training, we initially set the sigmoid temperature used in our smooth approximation (see Equation (4.2)) to a value $\tau = 0.1$, then lower it when the loss stops changing significantly between epochs. Lowering the temperature reduces the degree of smoothing and sharpens the boundaries between intervals defined by each cut point. Figure 4.2 shows an example training plot where the temperature drops after multiple epochs with no improvement in the validation loss. It is clear that this drop then leads to an improvement in both training and validation loss.

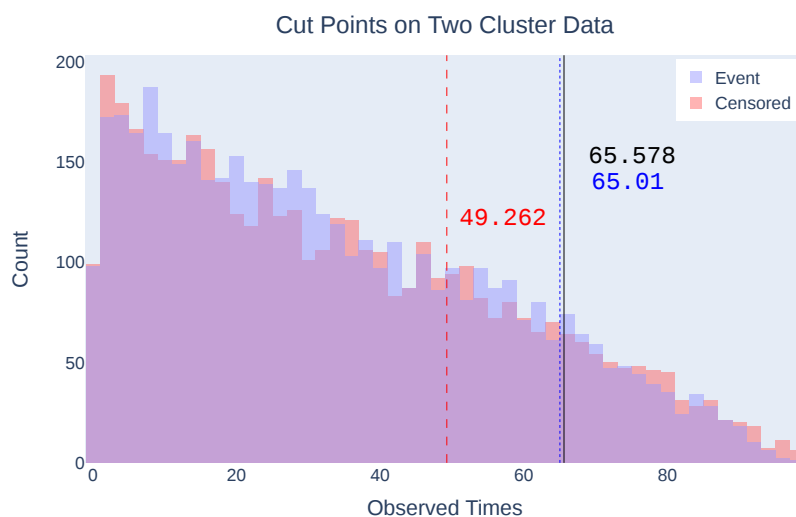
We perform a grid search over the hyper parameters, testing every combination of output dimension, weight decay, and regularization strength. The evaluation process to compare hyperparameters is described in Section 4.3.2. We train each network for 250 epochs.

To evaluate performance we perform five-fold cross validation. We randomly partition the data into training (75%), validation (15%), and test (10%) sets. For each set of hyperparameters



(a) The noisy, nonlinear feature-cluster relationship.

(b) The event times colored by true cluster.



(c) The cut points for ADEPT with $n = 7,500$ (black, solid), ADEPT with $n = 300$ (blue, dotted), and DTNN (red, dashed).

Figure 4.3: The event times and observed times of the two interval data. The true cut point is at time 67.

we perform this partition five times, using the training sets for learning the model parameters. We then calculate average performance metrics on the out of sample validation sets. Only the model with the best average validation set performance is then applied to the corresponding, yet unseen, test sets. We report the average and standard deviations of the performance metrics calculated across the folds on the test sets. Through this general cross-

Table 4.1: Performance metrics for synthetic data. We report average metrics across 5-fold cross validation with standard errors in parentheses. In bold are the highest CI values for each setting.

Intervals	ADEPT CI	DTNN CI
Two ($n = 7,500$)	0.947 (0.001)	0.797 (0.002)
Two ($n = 300$)	0.813 (0.039)	0.756 (0.016)
Four ($n = 7,500$)	0.980 (0.012)	0.937 (0.007)
Four ($n = 300$)	0.964 (0.013)	0.931 (0.013)

validation strategy, we are able to find the hyperparameter setting that performs the best on out of sample data from the hyperparameters tested. We report the mean and standard deviation of each metric across the folds.

We perform the same parameter search and evaluation to find the best DTNN model as described in Section 4.3.1. We compare the metrics of the best ADEPT model to that of the best baseline models. We report the performance metrics all both methods calculated on the unseen test set.

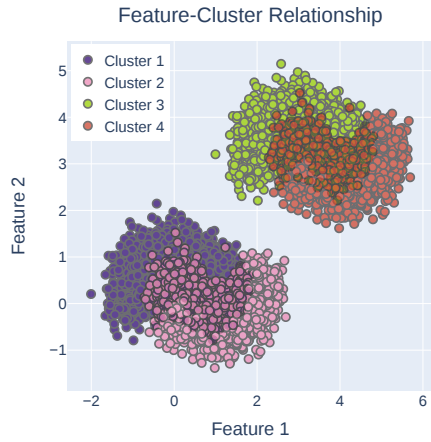
4.4 Simulation Examples

4.4.1 Learning Two Intervals

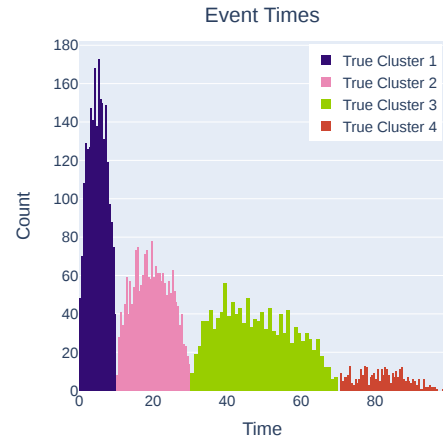
We start with the simple case of data generated from two clusters with uniform censoring. Cluster membership is generated using the `make_moons` function `sklearn` Python package to get a noisy, nonlinear relationship between $p = 2$ features (Pedregosa et al. 2011). Figure 4.3a shows the feature-cluster relationship; each cluster has 5,000 observations for a total of $n = 7,500$ observations. These clusters are used to generate the event times.

Event times in Cluster 1 are generated uniformly on the interval $(0, 67]$ and event times in Cluster 2 are uniformly on the interval $(67, 100)$. Censoring times are then generated uniformly throughout the entirety of $(0, 100)$. Note that while the censoring and event times are both uniformly distributed, the observed times are the minimum of the two and therefore not uniformly distributed. These observed times are shown in Figure 4.3b. Because these intervals are determined by the relationship between the covariates, this simulates data generated with a true cut point at time 67.

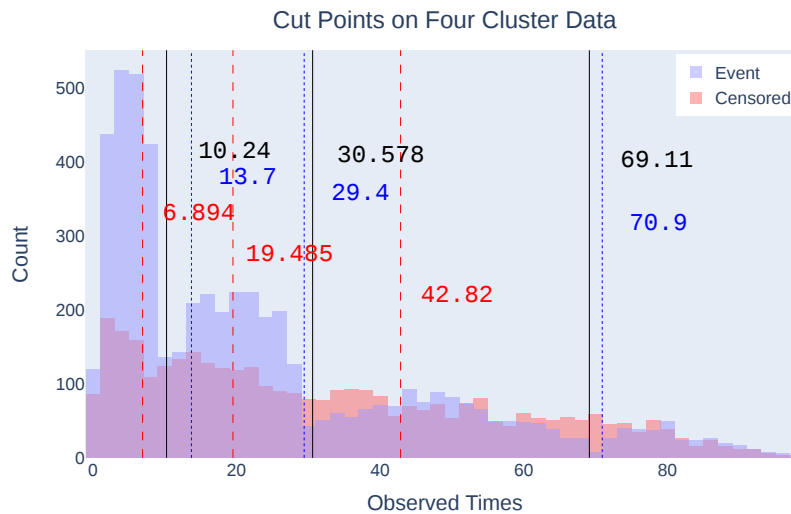
Figure 4.3c shows out of sample test set along with the DTNN cut point in red at $t = 49.3$ and the ADEPT learned cut point in black at $t = 65.6$. Knowing that the true cut point is at time 67 demonstrates the efficacy of our method. Even with a starting point far from the true cut



(a) The noisy, nonlinear feature-cluster relationship.



(b) The event times colored by true cluster.



(c) The cut points for ADEPT with $n = 7,500$ (black, solid), ADEPT with $n = 300$ (blue, dotted), and DTNN (red, dashed).

Figure 4.4: Event times and observed times of the four interval data. The true cut points are at 10, 30, and 70.

points, we are able to recover the true cut point. Table 4.1 reports the performance metrics, showing a large gain in CI.

In this simple example, many combinations of hyperparameters were able to recover the true cut point; reported are the results from using a small neural network with $h = 32$ with Adam weight decay of 0 and a regularization strength of $\lambda = 1$.

To demonstrate that our method works in limited data settings, we repeat this experiment randomly selecting only $n = 300$ training data points. Despite this data limitation, the blue lines in Figure 4.3c shows that ADEPT was still able to recover the true cut point. Table 4.1 shows that ADEPT still outperforms the DTNN model in prediction.

4.4.2 Learning Four Intervals

With confidence in ADEPT’s ability to learn a single cut point when it is present in the data generation, we expand to learning three true cut points. Again we use the `make_moons` function to generate noisy, nonlinear relationships between $p = 2$ features, however now for four separate clusters as shown in Figure 4.4a; each cluster has 2,500 observations for a total of $n = 7,500$ observations. Figure 4.4b shows the how these clusters are used to generate event times. Event times are generated using a $\text{Beta}(1.5, 1.5)$ distribution which are then scaled to be in the appropriate interval based on the observation’s cluster. The first cluster has event times on the interval $(0, 10]$, the second on the interval $(10, 30]$, the third on the interval $(30, 70]$, and the fourth on the interval $(70, 100)$. This corresponds to the true but points being at $t = 10, 30, 70$. We again apply uniform censoring times to all observations. Note that with uniform censoring, there are particularly few uncensored observations for events in the last interval. This makes learning the final cut point more difficult.

The black lines in Figure 4.4c shows that ADEPT was able to successfully recover all three cut points despite the challenges due to censoring. Table 4.1 shows that the ADEPT’s learned intervals provide an increase in CI over the DTNN. Since we know the data generating mechanism, it is intuitive for this simulation example that including more than 3 cut points leads to worse performance as introducing more would overparametrize the model. The results in the next section suggest that it is beneficial to consider models with fewer cut points even when the generating mechanism is unknown.

We repeat this simulation with only $n = 300$ data points. With the uniform censoring, this results in even fewer observed events in the final cluster. As in the two interval case, ADEPT is still able to recover the true cut points, shown in Figure 4.4c, and outperform the DTNN baseline model in predictive performance, shown in Table 4.1.

4.5 Data Analysis

4.5.1 Real-World Data Sources

We apply our method to three real-world data sources of varying sizes.

German Breast Cancer Study Group (GBSG) The GBSG data set is a publicly available data set introduced by Schumacher et al. (1994). It is a multicenter clinical trial which includes $n = 686$ patients with $p = 8$ features. The endpoint of recurrence free survival occurred for 299 (43.6%) patients.

Assay of Serum Free Light Chain (FL Chain) The FL Chain data set is a publicly available data set introduced by Dispenzieri et al. (2012) studies the relationship between nonclonal serum immunoglobulin free light chains and mortality. We examine the data for the $n = 6,524$ patients that had no missing data with $p = 8$ features. The endpoint of death occurred for 1,962 (30.1%) of these patients.

Pooled Stroke Risk Cohorts This is a combined dataset consisting of the Framingham Offspring Study (Feinleib et al. 1975) ($n_1 = 8,348$), The Atherosclerosis Risk in Communities Study (Investigators 1989) ($n_2 = 23,158$), and the Multi-Ethnic Study of Atherosclerosis ($n_3 = 6,390$) (Bild et al. 2002). Data harmonization procedures and characteristics of the dataset have previously been described by Hong et al. (2023). We consider a total of $n = 35,450$ data points of which 1,221 (3.44%) experience a stroke. There are $p = 69$ features that include cardiovascular medical history, demographic indicators, and diet information.

4.5.2 Results

Figure 4.5 shows the best ADEPT learned cut points for each data set compared to the DTNN. Note Figure 4.5c is a histogram of the proportion of observations rather than raw counts because of the high amount of censoring. Table 4.2 shows the performance metrics for all real-world data sets. The reported metrics are calculated on the held-out test sets not used for training or model validation.

Two interesting trends strongly support the benefits of ADEPT. First, for all data sets, the CI was the highest for the models that used only 3 cut points and tended to decrease as more cut points were added. Additionally, for all numbers of cut points, the predictive performance in both CI and AUC for the learned cut point model was higher than the DTNN model.

Notice that the greatest improvement in CI was observed for the GBSG data set, which has

the fewest observations among all data sets. This underscores the importance of ADEPT. For small data sets with a limited number of outcomes, it is necessary to limit the number of cut points, but performance can be improved by optimizing their locations. Notice that for the FL Chain data set, the DTNN model achieved its highest CI with 10 cut points, but this was still lower than the performance of ADEPT using only 3 learned cut points. Interestingly, the Stroke data set, which had the most data points and the highest outcome imbalance, also had a higher CI and AUC with 10 cut points than with 5. Similar to the other data sets, it achieved its highest CI and AUC using 3 cut points.

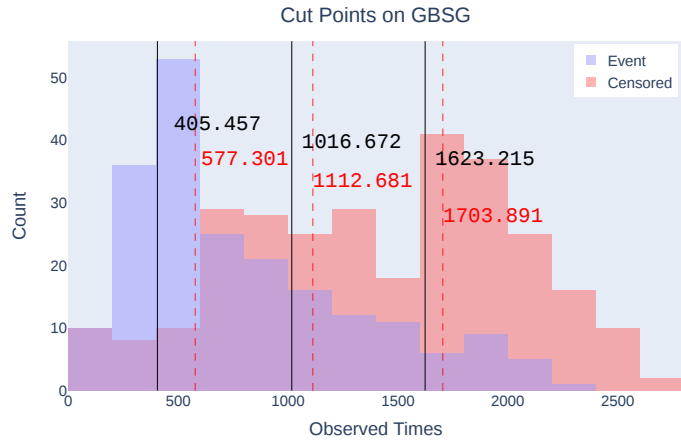
A model that is well calibrated has a calibration slope near 1 and a calibration intercept near 0. While the DTNN model had slightly better calibration slopes for the GBSG and FL Chain data sets for 3 cut points, ADEPT was better calibrated in nearly every setting with more cut points demonstrating model robustness.

While ADEPT was able to outperform DTNN in IBS for any given number of cut points, the continuous prediction of DeepSurv had the lowest overall IBS for each data set.

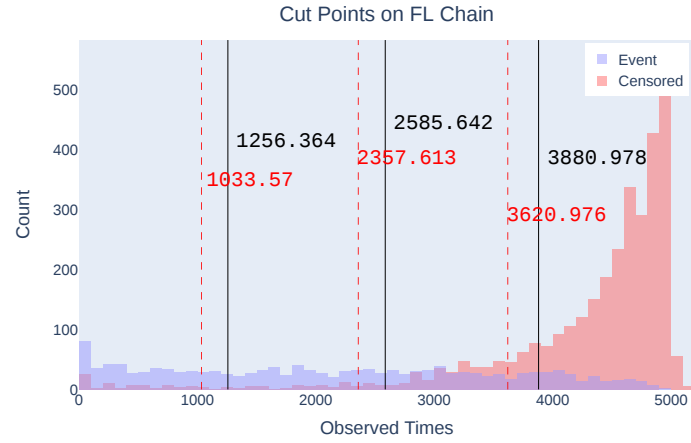
4.6 Conclusion

Herein we develop ADEPT, a flexible method to learn an optimal partitioning of the event time space that does not place strong assumptions on the form of the event density. Our approach is designed for clinical applications in which it is advantageous to learn, from data, a time discretization that facilitates more accurate prediction. The simulated examples demonstrated the ability of our method to recover cut points when they are truly present in the data generation mechanism. Moreover, results on real data show that the approach improves prediction performance over otherwise equivalent, state of the art models that use a fixed discretization scheme.

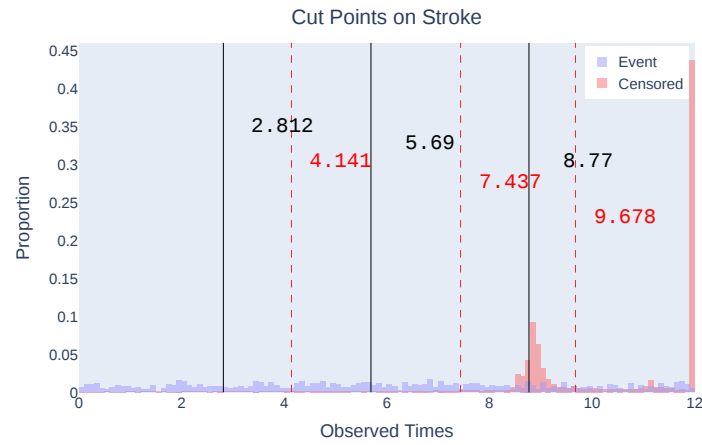
Our approach can be extended in several ways. In future work, we will consider a similar approach to learn separating hyperplanes in higher dimensional output spaces. The method can also be extended to sequential or time series data by using an appropriate encoder (*e.g.*, a recurrent neural network). Another interesting extension motivated by the real-world data analysis would be to learn the number of cut points from the data instead of fixing it *a priori*.



(a) Cut points on the GBSG data.



(b) Cut points on the FL Chain data.



(c) Cut points on the Stroke data.

Figure 4.5: The DTNN (red, dashed) and ADEPT learned (black, solid) cut points.

Table 4.2: Test-set performance metrics for real-world data averaged across 5-fold cross validation with standard errors in parentheses. In bold are the highest CI, highest AUC, and lowest IBS models for each data set.

	3 Cut Points	5 Cut Points	10 Cut Points
GBSG			
ADEPT CI	0.744 (0.015)	0.68 (0.018)	0.671 (0.024)
DTNN CI	0.681 (0.027)	0.651 (0.059)	0.619 (0.065)
ADEPT AUC	0.804 (0.021)	0.801 (0.02)	0.822 (0.024)
DTNN AUC	0.800 (0.03)	0.750 (0.034)	0.807 (0.016)
DeepSurv AUC	0.795 (0.054)	—	—
ADEPT IBS	0.180 (0.005)	0.189 (0.012)	0.173 (0.006)
DTNN IBS	0.187 (0.006)	0.198 (0.008)	0.194 (0.005)
DeepSurv IBS	0.165 (0.007)	—	—
ADEPT Calibration Slope	0.813 (0.089)	0.995 (0.112)	0.793 (0.165)
DTNN Calibration Slope	1.00 (0.091)	1.855 (0.154)	1.397 (0.254)
ADEPT Calibration Intercept	0.178 (0.048)	0.130 (0.042)	0.142 (0.06)
DTNN Calibration Intercept	0.129 (0.057)	-0.285 (0.084)	-0.215 (0.129)
FL Chain			
ADEPT CI	0.798 (0.003)	0.793 (0.003)	0.787 (0.004)
DTNN CI	0.763 (0.007)	0.772 (0.004)	0.774 (0.012)
ADEPT AUC	0.806 (0.004)	0.81 (0.004)	0.834 (0.002)
DTNN AUC	0.788 (0.008)	0.809 (0.004)	0.828 (0.004)
DeepSurv AUC	0.831 (0.002)	—	—
ADEPT IBS	0.194 (0.005)	0.163 (0.006)	0.146 (0.005)
DTNN IBS	0.191 (0.008)	0.153 (0.004)	0.13 (0.002)
DeepSurv IBS	0.099 (0.001)		
ADEPT Calibration Slope	1.199 (0.103)	1.182 (0.075)	1.057 (0.098)
DTNN Calibration Slope	1.005 (0.04)	0.999 (0.039)	0.898 (0.076)
ADEPT Calibration Intercept	0.056 (0.007)	0.050 (0.01)	0.085 (0.018)
DTNN Calibration Intercept	0.102 (0.008)	0.100 (0.009)	0.116 (0.026)
Stroke			
ADEPT CI	0.789 (0.014)	0.747 (0.02)	0.765 (0.006)
DTNN CI	0.778 (0.01)	0.739 (0.017)	0.758 (0.022)
ADEPT AUC	0.766 (0.011)	0.701 (0.03)	0.723 (0.013)
DTNN AUC	0.743 (0.01)	0.681 (0.031)	0.713 (0.021)
DeepSurv AUC	0.705 (0.015)		
ADEPT IBS	0.027 (0.002)	0.029 (0.002)	0.03 (0.002)
DTNN IBS	0.032 (0.003)	0.033 (0.002)	0.032 (0.002)
DeepSurv IBS	0.019 (0.001)		
ADEPT Calibration Slope	0.783 (0.013)	1.117 (0.195)	1.270 (0.321)
DTNN Calibration Slope	1.098 (0.321)	1.370 (0.248)	1.295 (0.376)
ADEPT Calibration Intercept	0.019 (0.002)	0.022 (0.003)	0.025 (0.002)
DTNN Calibration Intercept	0.022 (0.003)	0.019 (0.002)	0.025 (0.002)

REFERENCES

- AAP (2015). American academy of periodontology task force report on the update to the 1999 classification of periodontal diseases and conditions. *Journal of Periodontology*, 86(7):835–838.
- Abba, M. A., Williams, J. P., and Reich, B. J. (2023a). A penalized complexity prior for deep Bayesian transfer learning with application to materials informatics. *Annals of Applied Statistics*, 17(4):3241 – 3256.
- Abba, M. A., Williams, J. P., and Reich, B. J. (2023b). A penalized complexity prior for deep Bayesian transfer learning with application to materials informatics. *Annals of Applied Statistics*, 17(4):3241 – 3256.
- Abba, M. A., Williams, J. P., and Reich, B. J. (2024). A Bayesian shrinkage estimator for transfer learning. *arXiv preprint arXiv:2403.17321*.
- Ahishakiye, E., Van Gijzen, M. B., Tumwiine, J., Wario, R., and Obungoloch, J. (2021). A survey on deep learning in medical image reconstruction. *Intelligent Medicine*, 1(3):118–127.
- Akaoka, Y., Okamura, K., and Otobe, Y. (2022). Limit theorems for quasi-arithmetic means of random variables with applications to point estimations for the Cauchy distribution. *Brazilian Journal of Probability and Statistics*, 36(2):385 – 407.
- Antolini, L., Boracchi, P., and Biganzoli, E. (2005). A time-dependent discrimination index for survival data. *Statistics in Medicine*, 24(24):3927–3944.
- Bakker, B. and Heskes, T. (2003). Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4(May):83–99.
- Baxter, J. (1998). *Theoretical Models of Learning to Learn*, pages 71–94. Springer US, Boston, MA.
- Bild, D. E., Bluemke, D. A., Burke, G. L., Detrano, R., Diez Roux, A. V., Folsom, A. R., Greenland, P., Jacobs Jr, D. R., Kronmal, R., Liu, K., et al. (2002). Multi-ethnic study of atherosclerosis: objectives and design. *American Journal of Epidemiology*, 156(9):871–881.
- Bueno, A., Benítez, C., De Angelis, S., Díaz Moreno, A., and Ibáñez, J. M. (2020). Volcano-seismic transfer learning and uncertainty quantification with Bayesian neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 58(2):892–902.
- Cai, T. T. and Wei, H. (2021). Transfer learning for nonparametric classification: Minimax rate and adaptive classifier. *Annals of Statistics*, 49(1):100 – 128.
- Caruana, R. (1997). Multitask learning. *Machine learning*, 28:41–75.
- Chandra, R. and Kapoor, A. (2020). Bayesian neural multi-source transfer learning. *Neurocomputing*, 378:54–64.

- Chen, R., Sivakumar, K., and Kargupta, H. (2001). An approach to online bayesian learning from multiple data streams. In *Proceedings of Workshop on Mobile and Distributed Data Mining, PKDD*, volume 1, pages 31–45. Citeseer.
- Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202.
- Crowson, C. S., Atkinson, E. J., and Therneau, T. M. (2016). Assessing calibration of prognostic risk scores. *Statistical Methods in Medical Research*, 25(4):1692–1706.
- Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. (2007). Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, pages 193–200, New York, NY, USA. Association for Computing Machinery.
- de Valpine, P., Turek, D., Paciorek, C. J., Anderson-Bergman, C., Lang, D. T., and Bodik, R. (2017). Programming with models: writing statistical algorithms for general model structures with nimble. *Journal of Computational and Graphical Statistics*, 26(2):403–413.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- Desautels, T., Calvert, J., Hoffman, J., Mao, Q., Jay, M., Fletcher, G., Barton, C., Chettipally, U., Kerem, Y., and Das, R. (2017). Using transfer learning for improved mortality prediction in a data-scarce hospital setting. *Biomedical Informatics Insights*, 9.
- Dispenzieri, A., Katzmann, J. A., Kyle, R. A., Larson, D. R., Therneau, T. M., Colby, C. L., Clark, R. J., Mead, G. P., Kumar, S., Melton III, L. J., et al. (2012). Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population. In *Mayo Clinic Proceedings*, volume 87, pages 517–523. Elsevier.
- Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014a). Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 647–655, Beijing, China. PMLR.
- Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014b). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR.
- Dube, P., Bhattacharjee, B., Petit-Bois, E., and Hill, M. (2020). *Improving Transferability of Deep Neural Networks*, pages 51–64. Springer International Publishing, Cham.
- Eicker, F. (1985). Sums of independent squared Cauchy variables grow quadratically: Applications. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 47(1):133–140.
- Engelhard, M. and Henao, R. (2022). Disentangling whether from when in a neural mixture cure model for failure time data. In *International Conference on Artificial Intelligence and Statistics*, pages 9571–9581. PMLR.

- Fegyverneki, S. (2013). A simple robust estimation for parameters of cauchy distribution. *Miskolc Math. Notes*, 14(3):887–892.
- Feinleib, M., Kannel, W. B., Garrison, R. J., McNamara, P. M., and Castelli, W. P. (1975). The framingham offspring study. design and preliminary data. *Preventive Medicine*, 4(4):518–525.
- Freund, Y. and Schapire, R. (1999). A short introduction to boosting. *Japanese Society For Artificial Intelligence*, 14(771-780):1612.
- Friedman, J. H., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.
- Gao, Y. and Cui, Y. (2021). Multi-ethnic survival analysis: Transfer learning with Cox neural networks. In *Survival Prediction-Algorithms, Challenges and Applications*, pages 252–257. PMLR.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.
- Gong, J. J., Sundt, T. M., Rawn, J. D., and Guttag, J. V. (2015). Instance weighting for patient-specific risk stratification models. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15*, pages 369–378, New York, NY, USA. Association for Computing Machinery.
- Goussies, N. A., Ubalde, S., and Mejail, M. (2014). Transfer learning decision forests for gesture recognition. *Journal of Machine Learning Research*, 15(113):3847–3870.
- Graf, E., Schmoor, C., Sauerbrei, W., and Schumacher, M. (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, 18(17-18):2529–2545.
- Gu, T., Han, Y., and Duan, R. (2022). A transfer learning approach based on random forest with application to breast cancer prediction in underrepresented populations. In *Pacific Symposium on Biocomputing 2023*, pages 186–197. World Scientific.
- Gu, T., Lee, P. H., and Duan, R. (2023). Commute: Communication-efficient transfer learning for multi-site risk prediction. *Journal of Biomedical Informatics*, 137:104243.
- Guan, Q., Reich, B. J., Laber, E. B., and Bandyopadhyay, D. (2020). Bayesian nonparametric policy search with application to periodontal recall intervals. *Journal of the American Statistical Association*, 115(531):1066–1078.
- Harrell Jr, F. E., Lee, K. L., Califf, R. M., Pryor, D. B., and Rosati, R. A. (1984). Regression modelling strategies for improved prognostic prediction. *Statistics in Medicine*, 3(2):143–152.
- Hector, E. C. and Martin, R. (2024). Turning the information-sharing dial: Efficient inference from different data sources. *Electronic Journal of Statistics*, 18(2):2974 – 3020.

- Hickey, J., Williams, J. P., and Hector, E. C. (2024). Transfer learning with uncertainty quantification: Random effect calibration of source to target (recast). *Journal of Machine Learning Research*, 25(338):1–40.
- Hinkley, D. V. (1969). On the ratio of two correlated normal random variables. *Biometrika*, 56(3):635–639.
- Hong, C., Liu, M., Wojdyla, D. M., Hickey, J., Pencina, M., and Henao, R. (2024). Trans-balance: Reducing demographic disparity for prediction models in the presence of class imbalance. *Journal of Biomedical Informatics*, 149:104532.
- Hong, C., Pencina, M. J., Wojdyla, D. M., Hall, J. L., Judd, S. E., Cary, M., Engelhard, M. M., Berchuck, S., Xian, Y., D’Agostino, R., et al. (2023). Predictive accuracy of stroke risk prediction models across black and white race, sex, and age groups. *JAMA*, 329(4):306–317.
- Investigators, A. (1989). The atherosclerosis risk in community (aric) study: design and objectives. *American Journal of Epidemiology*, 129(4):687–702.
- Jacob, P. E., Gong, R., Edlefsen, P. T., and Dempster, A. P. (2021). A Gibbs sampler for a class of random convex polytopes. *Journal of the American Statistical Association*, 116(535):1181–1192. PMID: 35340357.
- Jhuang, A.-T., Fuentes, M., Bandyopadhyay, D., and Reich, B. J. (2020). Spatiotemporal signal detection using continuous shrinkage priors. *Statistics in medicine*, 39(13):1817–1832.
- Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning, ICML ’99*, page 200–209, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Johnson, S. G. (2013). QuadGK.jl: Gauss–Kronrod integration in Julia. <https://github.com/JuliaMath/QuadGK.jl>.
- Kapoor, S., Karaletsos, T., and Bui, T. D. (2021). Variational auto-regressive Gaussian processes for continual learning. In *International Conference on Machine Learning*, pages 5290–5300. PMLR.
- Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. (2018). Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1):24.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kravchuk, O. Y. and Pollett, P. K. (2012). Hodges-Lehmann scale estimator for Cauchy distribution. *Communications in Statistics - Theory and Methods*, 41(20):3621–3632.
- Kvamme, H., Borgan, Ø., and Scheel, I. (2019). Time-to-event prediction with neural networks and cox regression. *Journal of Machine Learning Research*, 20(129):1–30.

- Lee, C., Zame, W. R., Yoon, J., and van der Schaar, M. (2018). Deephit: A deep learning approach to survival analysis with competing risks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Lee, G., Rubinfeld, I., and Syed, Z. (2012). Adapting surgical models to individual hospitals using transfer learning. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 57–63.
- Li, J., Yu, Z., Du, Z., Zhu, L., and Shen, H. T. (2024). A comprehensive survey on source-free domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–22.
- Li, S., Cai, T., and Duan, R. (2023). Targeting underrepresented populations in precision medicine: A federated transfer learning approach. *Annals of Applied Statistics*, 17(4):2970–2992.
- Li, S., Cai, T. T., and Li, H. (2022). Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 84(1):149—173.
- Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., and Zhang, G. (2015). Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems*, 80:14–23.
- Maddox, W. J., Balandat, M., Wilson, A. G., and Bakshy, E. (2021). Bayesian optimization with high-dimensional outputs. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 19274–19287. Curran Associates, Inc.
- Marfoq, O., Neglia, G., Kameni, L., and Vidal, R. (2023). Federated learning for data streams. In Ruiz, F., Dy, J., and van de Meent, J.-W., editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 8889–8924. PMLR.
- Maurer, A., Pontil, M., and Romera-Paredes, B. (2015). The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17.
- Meng, M.-L., Frere, Z., Fuller, M., Li, Y.-J., Habib, A. S., Federspiel, J. J., Wheeler, S. M., Gilner, J. B., Shah, S. H., Ohnuma, T., et al. (2022). Maternal cardiovascular morbidity events following preeclampsia: A retrospective cohort study. *Anesthesia & Analgesia*, pages 10–1213.
- Miscouridou, X., Perotte, A., Elhadad, N., and Ranganath, R. (2018). Deep survival analysis: Nonparametrics and missingness. In *Machine Learning for Healthcare Conference*, page 244–256. PMLR.
- Myrick, K., McNeal, M., and Yin, X. (2022). National electronic health records survey. *National Center for Health Statistics*.
- Nelsen, R. B. (2006). *An introduction to copulas*. Springer.

- of the National Coordinator for Health Information Technology, O. (2017). Non-federal acute care hospital electronic health record adoption.
- Opper, M. and Winther, O. (1999). A bayesian approach to on-line learning. In *On-line learning in neural networks*, pages 363–378. Cambridge University Press.
- Page, R. C. and Eke, P. I. (2007). Case definitions for use in population-based surveillance of periodontitis. *Journal of periodontology*, 78:1387–1399.
- Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.
- Patel, K. K., Wang, L., Saha, A., and Srebro, N. (2023). Federated online and bandit convex optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 27439–27460. PMLR.
- Paul, R., Hawkins, S. H., Balagurunathan, Y., Schabath, M. B., Gillies, R. J., Hall, L. O., and Goldgof, D. B. (2016). Deep feature transfer learning in combination with traditional features predicts survival among patients with lung adenocarcinoma. *Tomography*, 2:388 – 395.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Pillai, N. S. (2016). Ratios and cauchy distribution. *arXiv preprint arXiv:1602.08181*.
- Pillai, N. S. and Meng, X.-L. (2016). An unexpected encounter with Cauchy and Lévy. *Annals of Statistics*, 44(5):2089 – 2097.
- Pollard, T. J., Johnson, A. E., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. (2018). The eICU collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1):1–13.
- Raghu, M., Zhang, C., Kleinberg, J., and Bengio, S. (2019). Transfusion: Understanding transfer learning for medical imaging. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Raina, R., Ng, A. Y., and Koller, D. (2006a). Constructing informative priors using transfer learning. In *Proceedings of the 23rd International Conference on Machine Learning, ICML ’06*, pages 713–720, New York, NY, USA. Association for Computing Machinery.
- Raina, R., Ng, A. Y., and Koller, D. (2006b). Constructing informative priors using transfer learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 713–720.
- Ranganath, R., Perotte, A., Elhadad, N., and Blei, D. (2016). Deep survival analysis. *arXiv preprint arXiv:1608.02158*.

- Reeve, H. W. J., Cannings, T. I., and Samworth, R. J. (2021). Adaptive transfer learning. *Annals of Statistics*, 49(6):3618 – 3649.
- Ren, K., Qin, J., Zheng, L., Yang, Z., Zhang, W., Qiu, L., and Yu, Y. (2019). Deep recurrent survival analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4798–4805.
- Roy, S., Trapp, M., Pilzer, A., Kannala, J., Sebe, N., Ricci, E., and Solin, A. (2022). Uncertainty-guided source-free domain adaptation. In Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T., editors, *Computer Vision – ECCV 2022*, pages 537–555, Cham. Springer Nature Switzerland.
- Schumacher, M., Bastert, G., Bojar, H., Hübner, K., Olschewski, M., Sauerbrei, W., Schmoor, C., Beyerle, C., Neumann, R. L., and Rauschecker, H. F. (1994). Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. german breast cancer study group. *Journal of Clinical Oncology*, 12(10):2086–2093. PMID: 7931478.
- Schuster, S. (2012). Parameter estimation for the cauchy distribution. In *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 350–353.
- Shao, S., McAleer, S., Yan, R., and Baldi, P. (2019). Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, 15(4):2446–2455.
- Shen, J., Qu, Y., Zhang, W., and Yu, Y. (2018). Wasserstein distance guided representation learning for domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Shickel, B., Davoudi, A., Ozrazgat-Baslanti, T., Ruppert, M., Bihorac, A., and Rashidi, P. (2021). Deep multi-modal transfer learning for augmented patient acuity assessment in the intelligent icu. *Frontiers in Digital Health*, 3.
- Shickel, B., Tighe, P. J., Bihorac, A., and Rashidi, P. (2017). Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE journal of biomedical and health informatics*, 22(5):1589–1604.
- Shwartz-Ziv, R., Goldblum, M., Souri, H., Kapoor, S., Zhu, C., LeCun, Y., and Wilson, A. G. (2022). Pre-train your loss: Easy Bayesian transfer learning with informative priors. *Advances in Neural Information Processing Systems*, 35:27706–27715.
- Si, Y. and Roberts, K. (2020). Patient representation transfer learning from clinical notes based on hierarchical attention network. *AMIA Summits on Translational Science Proceedings*, 2020:597.
- Singh, M., Duval, Q., Alwala, K. V., Fan, H., Aggarwal, V., Adcock, A., Joulin, A., Dollar, P., Feichtenhofer, C., Girshick, R., Girdhar, R., and Misra, I. (2023). The effectiveness of mae pre-pretraining for billion-scale pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5484–5494.

- Stojanov, P., Gong, M., Carbonell, J., and Zhang, K. (2019). Low-dimensional density ratio estimation for covariate shift correction. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 3449–3458. PMLR.
- Suder, P. M., Xu, J., and Dunson, D. B. (2023). Bayesian transfer learning. *arXiv preprint arXiv:2312.13484*.
- Swersky, K., Snoek, J., and Adams, R. P. (2013). Multi-task bayesian optimization. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.
- Tian, Y. and Feng, Y. (2022). Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, 0(0):1–14.
- Tianxi Cai, M. L. and Liu, M. (2024). Semi-supervised triply robust inductive transfer learning. *Journal of the American Statistical Association*.
- Tjandra, D., He, Y., and Wiens, J. (2021). A hierarchical approach to multi-event survival analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 591–599.
- Tripuraneni, N., Jin, C., and Jordan, M. (2021). Provable meta-learning of linear representations. In *International Conference on Machine Learning*, pages 10434–10443. PMLR.
- Vapnik, V. (2009). Transductive inference and semi-supervised learning. In *Semi-Supervised Learning*. IEEE.
- Wei, L.-J. (1992). The accelerated failure time model: a useful alternative to the cox regression model in survival analysis. *Statistics in Medicine*, 11(14-15):1871–1879.
- Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):9.
- Wiens, J., Gutttag, J., and Horvitz, E. (2014). A study in transfer learning: leveraging data from multiple hospitals to enhance hospital-specific predictions. *Journal of the American Medical Informatics Association*, 21(4):699–706.
- Williams, J. P. (2021). Discussion of “a Gibbs sampler for a class of random convex polytopes”. *Journal of the American Statistical Association*, 116(535):1198–1200.
- Wohlert, J., Munk, A., Sengupta, S., and Laumann, F. (2018). Bayesian transfer learning for deep networks. *viXra*.
- Wu, Q., Wu, H., Zhou, X., Tan, M., Xu, Y., Yan, Y., and Hao, T. (2017a). Online transfer learning with multiple homogeneous or heterogeneous sources. *IEEE Transactions on Knowledge and Data Engineering*, 29(7):1494–1507.
- Wu, Q., Zhou, X., Yan, Y., Wu, H., and Min, H. (2017b). Online transfer learning by leveraging multiple source domains. *Knowledge and Information Systems*, 52:687–707.

- Wu, X., Manton, J. H., Aickelin, U., and Zhu, J. (2023). A Bayesian approach to (online) transfer learning: Theory and algorithms. *Artificial Intelligence*, 324:103991.
- Yang, H., Jiao, S., and Sun, P. (2020). Bayesian-convolutional neural network model transfer learning for image detection of concrete water-binder ratio. *IEEE Access*, 8:35350–35367.
- Yu, C.-N., Greiner, R., Lin, H.-C., and Baracos, V. (2011). Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.
- Yu, J., Zhuge, Y., Zhang, L., Hu, P., Wang, D., Lu, H., and He, Y. (2024). Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23219–23230.
- Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. (2017). Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP.
- Zhang, J. (2010). A highly efficient L-estimator for the location parameter of the Cauchy distribution. *Computational statistics*, 25(1):97–105.
- Zhao, P., Hoi, S. C., Wang, J., and Li, B. (2014). Online transfer learning. *Artificial Intelligence*, 216:76–102.
- Zhou, C., Zhang, J., Liu, J., Zhang, C., Shi, G., and Hu, J. (2020). Bayesian transfer learning for object detection in optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(11):7705–7719.
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2021). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

APPENDIX

APPENDIX

A

CHAPTER 2 SUPPLEMENTARY MATERIAL

A.1 Proofs

Proof of Lemma 1. It is well-established (see, e.g., Hinkley 1969) that if $V \sim \mathcal{N}(0, \sigma_V^2)$ and $W \sim \mathcal{N}(0, \sigma_W^2)$ with correlation coefficient ρ , then

$$\frac{V}{W} \sim \text{Cauchy}\left(\frac{\rho\sigma_V}{\sigma_W}, \frac{\sigma_V}{\sigma_W}\sqrt{1-\rho^2}\right). \quad (\text{A.1})$$

Accordingly, since

$$\begin{bmatrix} \mathbf{a}^\top \\ \mathbf{b}^\top \end{bmatrix} \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{a}^\top \mathbf{a} & \mathbf{a}^\top \mathbf{b} \\ \mathbf{b}^\top \mathbf{a} & \mathbf{b}^\top \mathbf{b} \end{bmatrix}\right),$$

it follows that $\mathbf{x}^\top \mathbf{a} \sim \mathcal{N}(0, \mathbf{a}^\top \mathbf{a})$, $\mathbf{x}^\top \mathbf{b} \sim \mathcal{N}(0, \mathbf{b}^\top \mathbf{b})$, and $\rho = (\mathbf{a}^\top \mathbf{b})/(\|\mathbf{b}\| \|\mathbf{a}\|)$. The result follows from Equation (A.1) by taking $V = \mathbf{x}^\top \mathbf{a}$ and $W = \mathbf{x}^\top \mathbf{b}$. \square

Before proceeding directly to the proof of Lemma 2, the following necessary supporting result is stated and proved.

Lemma 4. *The MLEs of γ and δ for Equation (2.5), respectively, are*

$$\hat{\gamma} = \frac{\sum_{i=1}^{n_T} (v_i - \bar{v})(y_i - \bar{y}_T)}{\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S \sum_{i=1}^{n_T} (v_i - \bar{v})^2} \quad \text{and}$$

$$\hat{\delta} = \frac{\bar{y}_T}{\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S} - \bar{v} \cdot \hat{\gamma},$$

where $v_i = (\beta_i - \delta)/\gamma$ for $i \in \{1, \dots, n_T\}$, $\bar{v} := \sum_{i=1}^{n_T} v_i/n_T$ and $\bar{y}_T := \sum_{i=1}^{n_T} y_{T,i}/n_T$.

Proof of Lemma 4. After the change of variables $v_i = (\beta_i - \delta)/\gamma$ for $i \in \{1, \dots, n_T\}$, the likelihood function in Equation (2.5) takes the form

$$\prod_{i=1}^{n_T} \left[\mathcal{N}\{y_{T,i} \mid (\gamma v_i + \delta)\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S, \sigma^2\} \cdot \text{Cauchy}(v_i \mid 0, 1) \right].$$

Taking partial derivatives with respect to δ and γ gives the first-order conditions

$$\sum_{i=1}^{n_T} \left\{ \frac{y_{T,i}}{\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S} - \gamma v_i - \delta \right\} = 0$$

$$\sum_{i=1}^{n_T} \left\{ \frac{y_{T,i}}{\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S} - \gamma v_i - \delta \right\} v_i = 0.$$

Solving this system yields the MLEs in Lemma 4. □

Proof of Lemma 2. With the assumptions that $Y_{T,1}, \dots, Y_{T,n_T} \stackrel{\text{iid}}{\sim} \mathcal{N}(\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_T, \sigma^2)$ independent of $V_1, \dots, V_{n_T} \stackrel{\text{iid}}{\sim} \text{Cauchy}(0, 1)$, first, define the following notations:

$$\mathbf{Y} := \begin{pmatrix} Y_{T,1} \\ \vdots \\ Y_{T,n_T} \end{pmatrix}, \quad \bar{\mathbf{Y}} := \bar{Y}_T \cdot \mathbf{1}_{n_T}, \quad \bar{Y}_T := \frac{1}{n_T} \sum_{i=1}^{n_T} Y_{T,i},$$

and

$$\mathbf{V} := \begin{pmatrix} V_1 \\ \vdots \\ V_{n_T} \end{pmatrix}, \quad \bar{\mathbf{V}} := \bar{V} \cdot \mathbf{1}_{n_T}, \quad \bar{V} := \frac{1}{n_T} \sum_{i=1}^{n_T} V_i,$$

where $\mathbf{1}_{n_T}$ is an n_T -dimensional column vector with every component having value 1.

By the Cauchy-Schwarz inequality,

$$|\hat{\gamma}| = \frac{\left| \sum_{i=1}^{n_T} (V_i - \bar{V})(Y_i - \bar{Y}_T) \right|}{\left| \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S \sum_{i=1}^{n_T} (V_i - \bar{V})^2 \right|} \leq \frac{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_2 \|\mathbf{V} - \bar{\mathbf{V}}\|_2}{\left| \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S \right| \|\mathbf{V} - \bar{\mathbf{V}}\|_2^2} = \frac{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_2}{\left| \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S \right| \|\mathbf{V} - \bar{\mathbf{V}}\|_2},$$

where $\|\cdot\|_2$ is the Euclidean norm. We first need to establish the fact that square-root sums of independent, centered, and squared Cauchy random variables grow in value at the rate of at least $n_T^{\alpha+\frac{1}{2}}$ for any $\alpha \in (0, 1/2)$. Accordingly, for any $\varepsilon > 0$ and any $\alpha \in (0, 1/2)$,

$$\begin{aligned}
P\left(\|\mathbf{V} - \bar{\mathbf{V}}\|_2 < n_T^{\alpha+\frac{1}{2}} \varepsilon^{-1}\right) &= P\left(\|\mathbf{V} - \bar{\mathbf{V}}\|_2^2 < n_T^{2\alpha+1} \varepsilon^{-2}\right) \\
&= P\left(\sum_{i=1}^{n_T} V_i^2 - n_T \bar{V}^2 < n_T^{2\alpha+1} \varepsilon^{-2}\right) \\
&\leq P\left(\sum_{i=1}^{n_T} V_i^2 - n_T^{1+\alpha} < n_T^{2\alpha+1} \varepsilon^{-2}\right) + P\left(-n_T \bar{V}^2 < -n_T^{1+\alpha}\right) \\
&= P\left(\sum_{i=1}^{n_T} V_i^2 < n_T^{2\alpha+1} \varepsilon^{-2} + n_T^{1+\alpha}\right) + P\left(|\bar{V}| > n_T^{\frac{\alpha}{2}}\right) \\
&\leq P\left(\sum_{i=1}^{n_T} V_i^2 < n_T^{2\alpha+1} \{\varepsilon^{-2} + 1\}\right) + 2F_V\left(-n_T^{\alpha/2}\right), \tag{A.2}
\end{aligned}$$

where $F_V(\cdot)$ is the Cauchy(0, 1) distribution function. The first term vanishes for any $\alpha \in (0, 1/2)$ as $n_T \rightarrow \infty$ by Lemma 2.1 in Eicker (1985), and the second term vanishes as $n_T \rightarrow \infty$ by the definition of a distribution function.

Next, in order to show the convergence of both MLEs, we need that $n_T^{\alpha/2} \hat{\gamma} \rightarrow 0$ in probability as $n_T \rightarrow \infty$. Our argument goes as follows. For any $\varepsilon > 0$ and any $\alpha \in (0, 1/2)$,

$$\begin{aligned}
P\left(|\hat{\gamma}| > n_T^{-\alpha/2} \varepsilon\right) &\leq P\left(\frac{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_2}{|\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S| \|\mathbf{V} - \bar{\mathbf{V}}\|_2} > \frac{n_T^{(1+\alpha)/2} \varepsilon}{n_T^{(1+\alpha)/2} n_T^{\alpha/2}}\right) \\
&\leq P\left(\frac{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_2}{|\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S|} > n_T^{(1+\alpha)/2}\right) + P\left(\frac{1}{\|\mathbf{V} - \bar{\mathbf{V}}\|_2} > \frac{\varepsilon}{n_T^{\alpha+1/2}}\right) \\
&= P\left(\frac{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_2^2}{\sigma^2} > \frac{|\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S|^2}{\sigma^2} n_T^{1+\alpha}\right) + P\left(\|\mathbf{V} - \bar{\mathbf{V}}\|_2 < n_T^{\alpha+\frac{1}{2}} \varepsilon^{-1}\right).
\end{aligned}$$

Denoting $S := \|\mathbf{Y} - \bar{\mathbf{Y}}\|_2^2 / \sigma^2 \sim \chi_{n_T-1}^2$, and applying the Chernoff bound to the first quantity in the last expression gives, for any $t < 1/2$,

$$P\left(S > \frac{|\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S|^2}{\sigma^2} n_T^{1+\alpha}\right) \leq (1-2t)^{-(n_T-1)/2} \exp\left\{-\frac{t|\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S|^2}{\sigma^2} n_T^{1+\alpha}\right\}.$$

Choosing $t = 1/4$ yields the bound

$$P\left(|\hat{\gamma}| > n_T^{-\alpha/2} \varepsilon\right) \leq e^{-n_T^{1+\alpha} \frac{1}{2} \left(\frac{1}{2\sigma^2} |\tilde{\mathbf{x}}^\top \boldsymbol{\theta}_S|^2 - n_T^{-\alpha} + n_T^{-1-\alpha}\right)} + P\left(\|\mathbf{V} - \bar{\mathbf{V}}\|_2 < n_T^{\alpha+\frac{1}{2}} \varepsilon^{-1}\right).$$

Thus, by Equation (A.2), it follows that $n_T^{\alpha/2}\widehat{\gamma} \rightarrow 0$ in probability as $n_T \rightarrow \infty$. This fact implies that $\widehat{\gamma} \rightarrow 0$ in probability as $n_T \rightarrow \infty$, and is needed to prove the asymptotic convergence of $\widehat{\delta}$, next.

Since $Y_{T,1}, \dots, Y_{T,n_T} \stackrel{\text{iid}}{\sim} \mathcal{N}(\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T, \sigma^2)$, it follows that $\overline{Y}_T = \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T + \sigma n_T^{-\frac{1}{2}} U$, where $U \sim \mathcal{N}(0, 1)$. That being so, for any $\varepsilon > 0$ and any $\alpha \in (0, 1/2)$,

$$\begin{aligned} P\left(\left|\widehat{\delta} - \frac{\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T}{\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S}\right| > \varepsilon\right) &= P\left\{\left|\frac{1}{\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S}(\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T + \sigma n_T^{-\frac{1}{2}} U) - \overline{V}\widehat{\gamma} - \frac{\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T}{\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S}\right| > \varepsilon\right\} \\ &= P\left(\left|\frac{\sigma}{\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S} n_T^{-\frac{1}{2}} U - \overline{V}\widehat{\gamma}\right| > \varepsilon\right) \\ &\leq P\left(\left|\frac{\sigma}{\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S} n_T^{-\frac{1}{2}} U\right| > \frac{\varepsilon}{2}\right) + P(|\overline{V}\widehat{\gamma}| > \frac{\varepsilon}{2}) \\ &= 2\Phi\left\{-n_T^{\frac{1}{2}} \cdot \varepsilon |\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S| / (2\sigma)\right\} + P(|\overline{V}| > n_T^{\alpha/2}/2) + P(|\widehat{\gamma}| > n_T^{-\alpha/2}\varepsilon) \\ &= 2\Phi\left\{-n_T^{\frac{1}{2}} \cdot \varepsilon |\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S| / (2\sigma)\right\} + 2F_V\left(-n_T^{\alpha/2}/2\right) + P(|\widehat{\gamma}| > n_T^{-\alpha/2}\varepsilon), \end{aligned}$$

where $\Phi(\cdot)$ is the standard Gaussian distribution function. The first two terms in the last expression vanish by the definition of a distribution function, and the third term vanishes by the same because we previously established that $n_T^{\alpha/2}\widehat{\gamma} \rightarrow 0$ in probability as $n_T \rightarrow \infty$. Hence, $\widehat{\delta} \rightarrow \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T / (\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S)$ in probability as $n_T \rightarrow \infty$. \square

Proof of Theorem 3. Our argument begins with direct evaluation of the probability that $\widetilde{Y}_T \sim \mathcal{N}(\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T, \sigma^2)$ is contained in the interval $[a_{n_T}^\alpha, b_{n_T}^\alpha]$, and it finishes by applying the result of Lemma 2.

$$\begin{aligned} P\left(\widetilde{Y}_T \in [a_{n_T}^\alpha, b_{n_T}^\alpha]\right) &= \int_{a_{n_T}^\alpha}^{b_{n_T}^\alpha} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(\widetilde{y}_T - \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T)^2} d\widetilde{y}_T \\ &= \Phi\left(\frac{b_{n_T}^\alpha - \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T}{\sigma}\right) - \Phi\left(\frac{a_{n_T}^\alpha - \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T}{\sigma}\right), \end{aligned}$$

where $\Phi(\cdot)$ is the standard Gaussian distribution function. We will first demonstrate that $\Phi(W) \rightarrow 1 - \alpha/2$, with

$$\begin{aligned} W &:= \frac{b_{n_T}^\alpha - \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T}{\sigma} \\ &= \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) + \frac{1}{\sigma}(\widetilde{\boldsymbol{\beta}} \cdot \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S - \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T) \\ &\sim \text{Cauchy}\left\{\Phi^{-1}\left(1 - \frac{\alpha}{2}\right) + \frac{1}{\sigma}(\widehat{\boldsymbol{\delta}} \cdot \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S - \widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_T), \left|\frac{\widehat{\gamma}}{\sigma}\widetilde{\mathbf{x}}^\top \boldsymbol{\theta}_S\right|\right\} \end{aligned}$$

since $\tilde{\beta} \sim \text{Cauchy}(\widehat{\delta}, |\widehat{\gamma}|)$.

For any $\epsilon > 0$,

$$\begin{aligned} P\left(|\Phi(W) - (1 - \alpha/2)| < \epsilon\right) &= P\left(1 - \alpha/2 - \epsilon < \Phi(W) < 1 - \alpha/2 + \epsilon\right) \\ &= P\left\{\Phi^{-1}(1 - \alpha/2 - \epsilon) < W < \Phi^{-1}(1 - \alpha/2 + \epsilon)\right\} \\ &= F_W\left\{\Phi^{-1}(1 - \alpha/2 + \epsilon)\right\} - F_W\left\{\Phi^{-1}(1 - \alpha/2 - \epsilon)\right\}, \end{aligned}$$

where $F_W(\cdot)$ is the Cauchy distribution function associated with W . Then,

$$F_W\left\{\Phi^{-1}(1 - \alpha/2 + \epsilon)\right\} = \frac{1}{2} + \frac{1}{\pi} \arctan\left\{\frac{c_1 - (\widehat{\delta} \cdot \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_s - \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_T)/\sigma}{|\widehat{\gamma} \cdot \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_s|/\sigma}\right\},$$

with $c_1 := \Phi^{-1}(1 - \alpha/2 + \epsilon) - \Phi^{-1}(1 - \alpha/2) > 0$, and similarly,

$$F_W\left\{\Phi^{-1}(1 - \alpha/2 - \epsilon)\right\} = \frac{1}{2} + \frac{1}{\pi} \arctan\left\{\frac{c_2 - (\widehat{\delta} \cdot \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_s - \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_T)/\sigma}{|\widehat{\gamma} \cdot \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_s|/\sigma}\right\},$$

with $c_2 := \Phi^{-1}(1 - \alpha/2 - \epsilon) - \Phi^{-1}(1 - \alpha/2) < 0$. Accordingly, it follows by Lemma 2 that

$$F_W\left\{\Phi^{-1}(1 - \alpha/2 + \epsilon)\right\} \longrightarrow 1 \quad \text{and} \quad F_W\left\{\Phi^{-1}(1 - \alpha/2 - \epsilon)\right\} \longrightarrow 0$$

in probability as $n_T \rightarrow \infty$, and so

$$\Phi\left(\frac{b_{n_T}^\alpha - \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_T}{\sigma}\right) = \Phi(W) \longrightarrow 1 - \alpha/2$$

in probability as $n_T \rightarrow \infty$. A similar argument shows that

$$\Phi\left(\frac{a_{n_T}^\alpha - \tilde{\mathbf{x}}^\top \boldsymbol{\theta}_T}{\sigma}\right) \longrightarrow \alpha/2,$$

in probability as $n_T \rightarrow \infty$, concluding the proof. □

A.2 Bounding Continuous Integral

Recall the posterior distribution of the calibration parameters for the continuous response setting,

$$\begin{aligned} & \pi(\delta, \gamma, \sigma \mid y_{T,1}, \dots, y_{T,n_T}, \hat{\boldsymbol{\theta}}_S) \\ &= \pi(\delta, \gamma, \sigma) \cdot \prod_{i=1}^{n_T} \int_{\mathbb{R}} \frac{\text{Cauchy}(\beta_i \mid \delta, \gamma)}{|f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})|} \cdot \mathcal{N}\left\{\beta_i \mid \frac{y_{T,i}}{f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})}, \frac{\sigma^2}{f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})^2}\right\} d\beta_i. \end{aligned}$$

Calculating this posterior requires the evaluation of n_T integrals over \mathbb{R} . For computational efficiency, we estimate the posterior by integrating over closed intervals. The incurred numerical error can be tuned to be lower than computer precision.

Performing the substitution $u_i = \{\beta_i - y_{T,i}/f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})\}/\{\sigma/|f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})|\}$ re-expresses the i th integral as

$$\begin{aligned} & \int_{\mathbb{R}} \frac{\mathcal{N}(u_i \mid 0, 1)}{\sigma} \cdot \text{Cauchy}\left[u_i \mid \frac{|f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})|}{\sigma} \left\{ \delta - \frac{y_{T,i}}{f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})} \right\}, \frac{|f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})|\gamma}{\sigma}\right] du_i \\ & \leq \frac{1}{\sigma} \left(\int_{s_1}^{s_2} \mathcal{N}(u_i \mid 0, 1) \cdot \text{Cauchy}\left[u_i \mid \frac{|f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})|}{\sigma} \left\{ \delta - \frac{y_{T,i}}{f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})} \right\}, \frac{|f(\hat{\boldsymbol{\theta}}_S, \mathbf{x}_{T,i})|\gamma}{\sigma}\right] du_i \right. \\ & \quad \left. + \phi(s_1) + \phi(s_2) \right), \end{aligned}$$

for any s_1 and s_2 satisfying $s_1 < 0 < s_2$, where $\phi(\cdot)$ is the standard Gaussian density function. Then choose s_1 and s_2 so that $\phi(s_1) + \phi(s_2)$ is as small as desired. For example, we set $s_1 = -39$ and $s_2 = 39$, giving $\phi(s_1)$ and $\phi(s_2)$ numerically equal to zero in the base Julia software (for comparison, $\phi(38) = 1.097 \times 10^{-314}$).

A.3 MCMC Implementation Details

Sections 2.4.2 and 2.5.2 detail the procedure for sampling from the posterior predictive distribution of a new observation. RECaST first estimates the joint posterior density of the re-calibration parameters (δ, γ, σ) in the linear model and (δ, γ) in the logistic model. We specify disperse priors $\delta \sim \mathcal{N}(1, 400)$, $\log(\gamma) \sim \mathcal{N}(0, 9)$, and in the continuous setting $\log(\sigma^2) \sim \mathcal{N}(0, 9)$. We run the Metropolis-Hastings estimation algorithm of the posterior distribution for 100,000 iterations with the initial 20,000 iterations used as a burn-in period to tune the proposal variance. The parameters from the final 50,000 iterations are used as the posterior distribution. Finally, $n_{\text{post}} = 300$ equally spaced triplets/pairs of this distribution are taken as a posterior sample to

be used in the posterior predictive estimation, which we denote by $\{\delta_i, \gamma_i, \sigma_i\}_{i=1}^{300}$ and $\{\delta_i, \gamma_i\}_{i=1}^{300}$ in the linear and logistic models respectively. For each triplet/pair, a sample of $n_\beta = 300$ β 's are taken from the Cauchy distribution, each used to generate $n_Y = 300$ samples from the posterior predictive distribution. This gives $300 \times 300 \times 300 = 27,000,000$ posterior predictive observations for each out-of-sample test point, $(Y_{T,\text{test}}, \tilde{\mathbf{x}}_T)$.

A.4 Neural Network Training Procedure

The following procedure is used to train all neural networks considered: the source DNN, the DNN trained only on target data, and the Unfreeze DNN.

We initialize the weights using Xavier initialization (Glorot and Bengio 2010). The network is trained for 2500 epochs using the ADAM optimizer and an MSE loss. A portion of the training data is set aside as an out-of-sample calibration set during training. At each epoch, the training and calibration loss are tracked. The final parameterization used is taken from the epoch with the lowest calibration loss to avoid overfitting.

The candidate architectures ranged from networks with 316 parameters to 11,641 parameters with varied number of layers, layer sizes, activation functions, and dropout proportions. The architecture described below was chosen as it had the best test set AUC on the eICU data of all considered architectures. We use a two layer neural network with layer sizes $\ell_1 = (p, 25)$ and $\ell_2 = (25, 1)$. These layers are connected with a Rectified Linear Unit (ReLU) activation function. In the binary response setting, the output of ℓ_2 is converted to a probability through a softmax activation function. For consistency, this architecture is also used for the simulated data analysis in Section 2.6.

The source neural network for RECaST learns parameters in both layers using only source data. The DNN network learns parameters in both layers using only the target data. The Unfreeze DNN network learns parameters in both layers first using only the source data. Then, the target data are processed through the same neural network, re-training parameters in the second layer and leaving the first layer unchanged from the values learned on the source data set.

Table A.1: Empirical coverage (standard error) at the 95% nominal level for a continuous response, averaged over 300 source and target data sets when the target model parameters are generated as $\boldsymbol{\theta}_T = \boldsymbol{\theta}_S + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}_{p \times 10 + 10 \times 1}(\mathbf{0}, 0.025\mathbf{I})$. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	RECaST LM	RECaST DNN
250	100(0.4)	100(0.28)
100	100(0.33)	100(0.33)
60	100(0.35)	100(0.32)
40	100(0.29)	100(0.32)
20	100(0.37)	100(0.37)

Table A.2: Empirical coverage (standard error) at the 95% nominal level for a continuous response, averaged over 300 source and target data sets when the source and target neural network weight matrices are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	RECaST LM	RECaST DNN
250	100(0.53)	100(0.24)
100	100(0.42)	100(0.68)
60	100(0.36)	100(0.3)
40	100(0.3)	100(0.93)
20	100(0.26)	100(0.25)

A.5 Additional Tables for Section 2.6.4

A.6 Additional Robustness Results

Here, we consider the case where $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_T$ are orthogonal. We take $\boldsymbol{\theta}_S$ to be the same $p = 50$ feature vector as in Section 2.6.1 and take $\boldsymbol{\theta}_T$ to be a vector in the null space of $\boldsymbol{\theta}_S$. The data are otherwise generated following Section 2.6.1 from a linear or logistic regression with a fixed source sample size of $n_S = 1000$ and a varying target sample size of $n_T \in \{20, 40, 60, 100, 250\}$.

In the continuous outcome case, we compare RECaST to the DNN, Unfreeze DNN, TransRF and glmtrans approaches. In the binary outcome case, we compare RECaST to the target-only DNN, Unfreeze DNN, TransRF, glmtrans, WDGRL and Wiens methods. Table A.3 shows the predictive performance of RECaST for the orthogonally misaligned source and target setting with a continuous response.

Table A.3: Out of sample RMSE (standard error) averaged over 300 source and target data sets when the source data generating parameters are orthogonal to the target data generating parameters. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	LM	DNN	RECaST LM	RECaST DNN	Unfreeze DNN	TransRF	glmtrans	MTL FO	MTL MoM
250	0.56(0.03)	0.85(0.061)	1.1(0.051)	1.1(0.051)	0.92(0.082)	0.63(0.066)	0.54(0.025)	0.6(0.03)	0.6(0.03)
100	0.71(0.06)	1.1(0.12)	1.1(0.050)	1.1(0.051)	1.1(0.19)	0.97(1.1)	0.55(0.031)	0.76(0.06)	0.76(0.06)
60	1.3(0.28)	1.3(0.13)	1.1(0.05)	1.1(0.051)	1.2(0.28)	1.8(2.0)	0.57(0.043)	1.3(0.26)	1.3(0.26)
40	1.2(0.22)	1.3(0.13)	1.1(0.056)	1.1(0.056)	1.5(0.40)	3.0(6.2)	0.6(0.066)	3.2(0.71)	1.4(0.29)
20	1.0(0.07)	1.4(0.14)	1.2(0.078)	1.2(0.073)	1.9(0.59)	-	0.66(0.08)	5.5(0.81)	1.0(0.07)

When the target data are plentiful ($n_T = 100, 250$) the RMSE for the LM built solely on the target data outperforms the RECaST methods. This aligns with previous results where there is a large amount of data and a large discrepancy between the source and target (i.e., when transfer learning is not appropriate). As the number of target data points decreases, RECaST outperforms target-only DNN. These results further demonstrate the robustness of RECaST to negative transfer. Notice that glmtrans is also robust; in each of these scenarios glmtrans opted to not use the source data. The MTL MoM method also provides good predictive performance for all sample sizes without requiring access to the source data. Similar to the previous robustness tests, Table A.4 shows that RECaST provides conservative predictive intervals resulting in over-coverage at the 95% level.

Table A.4: Empirical coverage (standard error) at the 95% nominal level, averaged over 300 source and target data sets when the source data generating parameters are orthogonal to the target data generating parameters. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	RECaST LM	RECaST DNN
250	100(0)	100(0)
100	100(0)	100(0)
60	100(0)	100(0)
40	100(0)	100(0)
20	100(0.024)	100(0.024)

For a binary response, Table A.5 shows the difficulty of this problems. All methods have very low AUCs, including the target-only DNN. For large sample sizes, glmtrans performs relatively well, again due to its ability to ignore the source data entirely and because the model matches the data generating mechanism. Table A.6 shows that all methods have predictive coverages

with very high standard errors, again displaying the difficulty of this problem.

Next we consider a setting in which the source feature space is a subset of the target feature space: $\mathcal{X}_S \subsetneq \mathcal{X}_T$. We assign 12 features to the true target data \mathbf{X}_T but only 9 features to the true source data \mathbf{X}_S . The parameters are generated as $\boldsymbol{\theta}_T = (-\mathbf{a}, \mathbf{b})$ where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^6$ have components independently sampled from $\text{Uniform}(0.75, 5)$, and $\boldsymbol{\theta}_S = [\theta_{T,1}, \dots, \theta_{T,9}]$, the first nine components of $\boldsymbol{\theta}_T$. The responses, \mathbf{Y}_S and \mathbf{Y}_T , are generated via linear or logistic regression with their respective feature vectors.

Table A.7 shows that for a continuous response, every method has similar predictive performance when the target sample size is large. As the target sample size decreases, RECaST and glmtrans have the best performance, maintaining a stable RMSE value and outperforming the target-only DNN. This shows that RECaST is robust to negative transfer in this setting.

Table A.5: Out-of-sample AUC (standard error) averaged over 300 source and target data sets when the source and target model parameter vectors are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	59(5.6)	51(3.1)	50(3.7)	49(3.6)	55(4.7)	57(9.1)	72(3.5)	49(3.7)
100	54(5.3)	50(3.5)	49(3.4)	45(3.7)	53(4.7)	37(14)	69(7.1)	50(3.5)
60	52(5.3)	50(3.8)	50(4.2)	43(3.4)	51(4.7)	25(16)	61(10)	48(3.6)
40	52(4.9)	50(3.5)	50(3.8)	42(2.9)	52(4)	23(16)	59(9.1)	49(4)
20	52(4.8)	50(3.5)	50(3.4)	41(4.1)	51(4.5)	-	56(9.5)	-

Table A.6: Empirical coverage (standard error) at the 75% nominal level, averaged over 300 source and target data sets when the source and target model parameter vectors are orthogonal. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	59(11)	100(0)	100(0)	53(10)	55(10)	61(13)	81(11)	52(17)
100	55(11)	83(29)	100(0)	52(11)	52(11)	54(13)	77(15)	47(18)
60	50(11)	89(19)	51(10)	48(11)	49(12)	52(15)	74(19)	47(17)
40	52(11)	45(16)	77(27)	47(11)	52(11)	52(27)	67(19)	48(10)
20	51(11)	60(27)	57(25)	51(12)	51(14)	-	59(13)	-

Table A.8 shows RECaST again provides conservative predictive intervals at the 95% level. We see similar results for a binary response outcome in Table A.9. Both RECaST methods have stable AUCs as the target sample size decreases, outperforming the target-only DNN and the other transfer learning methods. WDGRL and glmtrans also perform well for larger sample sizes, but both require access to the source data while training. Table A.10 shows that the RECaST and Wiens methods again provide conservative predictive coverage for all target sample sizes. WDGRL and glmtrans under-cover in some scenarios.

Table A.7: The reported values are: average out-of-sample RMSE (standard deviation). These summaries are over all 300 different source and target data sets for each target sample size when the target data had more features than the source.

n_T	LM	DNN	RECaST LM	RECaST DNN	Unfreeze DNN	TransRF	glmtrans	MTL FO	MTL MoM
250	5.5(1.3)	5.8(1.4)	5.4(1.3)	5.4(1.3)	5.52(1.36)	6.5(1.5)	5.3(1.2)	6.3(1.2)	6.3(1.2)
100	5.7(1.3)	6.3(1.5)	5.4(1.3)	5.4(1.3)	5.68(1.35)	11.0(14.0)	5.4(1.2)	6.5(1.2)	6.5(1.2)
60	5.8(1.4)	6.8(1.7)	5.4(1.3)	5.4(1.3)	5.94(1.53)	14.0(14.0)	5.3(1.2)	6.7(1.3)	6.7(1.3)
40	6.2(1.5)	7.2(1.8)	5.4(1.4)	5.4(1.4)	6.06(1.76)	30.0(43.0)	5.4(1.3)	6.9(1.4)	6.9(1.4)
20	7.3(2.1)	8.2(1.9)	5.5(1.3)	5.5(1.4)	6.63(1.93)	-	5.6(1.5)	8.1(1.9)	8.1(1.9)

Table A.8: Empirical coverage (standard error) at the 95% nominal level, averaged over 300 source and target data sets when the target data had more features than the source. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	RECaST LM	RECaST DNN
250	100(0)	100(0)
100	100(0)	100(0)
60	100(0)	100(0)
40	100(0)	100(0)
20	100(0.022)	100(0.2)

Table A.9: Out-of-sample AUC (standard error) averaged over 300 source and target data sets when the target data had more features than the source. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	86(5.3)	89(4.5)	89(4.5)	74(7.9)	89(3.9)	60(16)	90(5.9)	89(6.5)
100	83(10)	89(4.4)	89(4.3)	75 (8.0)	84(10)	46(18)	88(4.1)	89(4.2)
60	75(13)	89(4.2)	89(4)	74(7.2)	79(12)	38(21)	86(3.8)	90(3.4)
40	73(11)	89(4.3)	89(4.3)	73(7.3)	78(14)	32(12)	82(9.2)	89(4.0)
20	75(7.9)	88(4.5)	88(4.3)	74(6.6)	81(9.6)	-	69(13)	-

Table A.10: Empirical coverage (standard error) at the 75% nominal level, averaged over 300 source and target data sets when the target data had more features than the source. The out-of-sample test sets each contain 250 observations. All reported values are multiplied by 100.

n_T	DNN	RECaST GLM	RECaST DNN	Wiens	Unfreeze DNN	TransRF	glmtrans	WDGRL
250	70(14)	100(0)	98(1.6)	87(11)	75(13)	73(14)	71(12)	63(15)
100	65(7.6)	96(0)	95(7.3)	85(8.9)	73(12)	64(15)	72(11)	64(13)
60	67(16)	89(6.5)	90(12)	86(8.2)	70(14)	58(22)	79(13)	69(9.9)
40	60(11)	86(16)	88(11)	86(8.2)	72(13)	53(17)	74(15)	63(17)
20	59(12)	82(17)	80(19)	86(9.8)	67(19)	-	70(16)	-

A.7 Comparative eICU Results

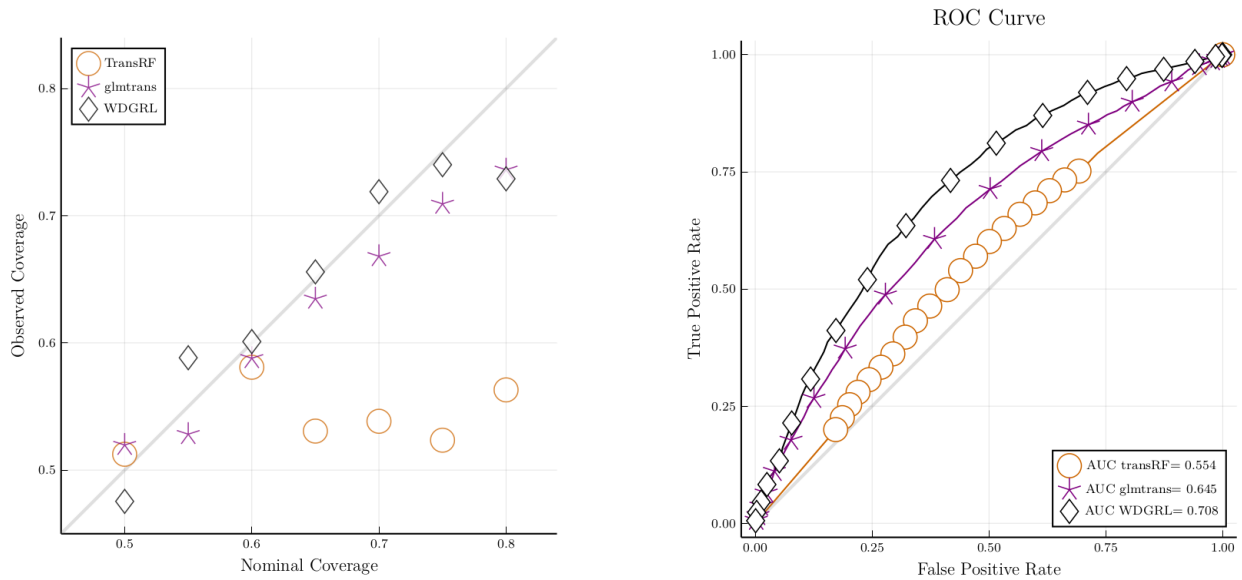


Figure A.1: Results for TransRF, glmtrans, and WDGRL on the eICU data set. The left panel displays the reliability curve of the nominal versus empirical out-of-sample coverage of prediction sets averaged over 300 target-testing data sets; the right panel reports the out-of-sample receiver operating characteristic (ROC) curve averaged pointwise over 300 target-testing data sets. The legend also reports the AUC (standard error) averaged over the same 300 target-testing data sets. Note that we cut the reliability curve at a nominal coverage of 0.8 because there are very few observations with higher coverage, undermining the reliability of coverage estimation at higher nominal levels.

A.8 eICU Feature Descriptions

Table A.11: Descriptions of the features from the eICU Collaborative Research Database used in the shock data analysis.

Variable	Description
Age	age in years
Gender	gender as either Male, Female, Unknown or Other
Ethnicity	ethnicity as either Asian, Caucasian, African American, Native American, Hispanic or Other/Unknown
Weight	weight upon admission
Temperature	worst temperature measured from a midpoint of 38°C
White blood cell count	worst white blood cell count from a midpoint of 11,500 white blood cells per microliter
Respiratory rate	worst respiratory rate from a midpoint of 19 breaths per minute
Heart rate	worst heart rate from a midpoint of 75 beats per minute
Hematocrit level	worst hematocrit from a midpoint of 45.5%
Creatinine level	worst serum creatinine from a midpoint of 1.0 milligrams per deciliter
Glucose level	worst glucose from a midpoint of 130 milligrams per deciliter
Oxygen saturation	oxygen saturation in the blood measured by a pulse oximeter
Dialysis	an indicator reporting if the patient is on dialysis
Intubated	an indicator reporting if the patient was intubated during the worst measurement of their arterial blood gas
Ventilated	binary an indicator reporting if the patient was ventilated during the measurement worst respiratory rate
Eye	eye score ranging from 1 to 4 on the Glasgow Coma Scale
Motor	motor score ranging from 1 to 6 on the Glasgow Coma Scale
Verbal	verbal score ranging from 1 to 3 on the Glasgow Coma Scale

A.9 Multivariate Cauchy Gibbs Sampler

For univariate RECaST, the following is the equivalent of putting a location-scale t -distribution prior on the mean of a Gaussian random variable

$$\begin{aligned} \mathbf{y}_1, \dots, \mathbf{y}_{n_T} \mid \boldsymbol{\mu}, \sigma^2, \sigma_0^2 &\sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2) \\ \boldsymbol{\mu} \mid \sigma_0^2 &\sim \mathcal{N}(\boldsymbol{\delta}, \sigma_0^2) \\ \sigma_0^2 &\sim \text{IG}\left(\frac{\nu}{2}, \frac{\nu \cdot \gamma^2}{2}\right). \end{aligned}$$

Then take priors

$$\begin{aligned} \boldsymbol{\delta} &\sim \mathcal{N}(\boldsymbol{\mu}_\delta, \sigma_\delta^2) \\ \gamma &\sim \text{IG}(a_\gamma, b_\gamma). \end{aligned}$$

For the multivariate t -distribution, we have the following hierarchical representation:

$$\begin{aligned} \mathbf{Y}_{T,1}, \dots, \mathbf{Y}_{T,n_T} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma} &\sim \mathcal{N}_m(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \boldsymbol{\mu} \mid u, \boldsymbol{\delta}, \boldsymbol{\Gamma} &\sim \mathcal{N}_m(\boldsymbol{\delta}, u^{-1}\boldsymbol{\Gamma}) \\ u &\sim \text{Gamma}\left(\frac{\nu}{2}, \frac{\nu}{2}\right) \end{aligned}$$

with priors

$$\begin{aligned} \boldsymbol{\Sigma} &\sim \text{IW}_m(\boldsymbol{\Psi}_\Sigma, \nu_\Sigma) \\ \boldsymbol{\delta} &\sim \mathcal{N}_m(\boldsymbol{\mu}_\delta, \boldsymbol{\Sigma}_\delta) \\ \boldsymbol{\Gamma} &\sim \text{IW}_m(\boldsymbol{\Psi}_\Gamma, \nu_\Gamma), \end{aligned}$$

where ν is the degrees of freedom for the t -distribution prior on $\boldsymbol{\mu}$. The following full conditional distributions can be used for Gibbs sampling

$$\begin{aligned} \boldsymbol{\mu} \mid \mathbf{Y}_{T,1}, \dots, \mathbf{Y}_{T,n_T}, \boldsymbol{\Sigma}, \boldsymbol{\delta}, \boldsymbol{\Gamma}, u &\sim \mathcal{N}\left(\left[u\boldsymbol{\Gamma}^{-1} + n_T\boldsymbol{\Sigma}^{-1}\right]^{-1}\left[u\boldsymbol{\Gamma}^{-1}\boldsymbol{\delta} + n_T\boldsymbol{\Sigma}^{-1}\bar{\mathbf{y}}_T\right], \left[u\boldsymbol{\Gamma}^{-1} + n_T\boldsymbol{\Sigma}^{-1}\right]^{-1}\right) \\ \boldsymbol{\Sigma} \mid \mathbf{Y}_{T,1}, \dots, \mathbf{Y}_{T,n_T}, \boldsymbol{\mu}, \boldsymbol{\delta}, \boldsymbol{\Gamma}, u &\sim \text{IW}\left(n_T + \nu_\Sigma, \boldsymbol{\Psi}_\Sigma + \sum_{i=1}^{n_T} [\mathbf{y}_{T,i} - \boldsymbol{\mu}][\mathbf{y}_{T,i} - \boldsymbol{\mu}]^\top\right) \\ \boldsymbol{\delta} \mid \mathbf{Y}_{T,1}, \dots, \mathbf{Y}_{T,n_T}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Gamma}, u &\sim \mathcal{N}_m\left(\left[\boldsymbol{\Sigma}_\delta^{-1} + u\boldsymbol{\Gamma}^{-1}\right]^{-1}\left[\boldsymbol{\Sigma}_\delta^{-1}\boldsymbol{\mu}_\delta + u\boldsymbol{\Gamma}^{-1}\boldsymbol{\mu}\right], \left[\boldsymbol{\Sigma}_\delta^{-1} + u\boldsymbol{\Gamma}^{-1}\right]^{-1}\right) \\ \boldsymbol{\Gamma} \mid \mathbf{Y}_{T,1}, \dots, \mathbf{Y}_{T,n_T}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\delta}, u &\sim \text{IW}_m\left(\nu_\Gamma - 1, u(\boldsymbol{\mu} - \boldsymbol{\delta})(\boldsymbol{\mu} - \boldsymbol{\delta})^\top + \boldsymbol{\Psi}_\Gamma\right) \\ u \mid \mathbf{Y}_{T,1}, \dots, \mathbf{Y}_{T,n_T}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\delta}, \boldsymbol{\Gamma} &\sim \text{Gamma}\left(\frac{\nu + m}{2}, \frac{1}{2}[\boldsymbol{\mu} - \boldsymbol{\delta}]\boldsymbol{\Gamma}^{-1}[\boldsymbol{\mu} - \boldsymbol{\delta}] + \frac{\nu}{2}\right). \end{aligned}$$

Setting $\nu = 1$ corresponds to setting a multivariate Cauchy prior on $\boldsymbol{\mu}$.

A.10 Multivariate Gaussian Copula Finite Integrals

$$\begin{aligned}
& \pi(\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \boldsymbol{\Sigma} \mid y_{1,1}, \dots, y_{n_T,m}, \widehat{\boldsymbol{\Theta}}_S) \\
&= \int_{\mathbb{R}} \dots \int_{\mathbb{R}} \pi(\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \boldsymbol{\Sigma}, \boldsymbol{\beta}_{1,1}, \dots, \boldsymbol{\beta}_{n_T,m} \mid y_{1,1}, \dots, y_{n_T,m}, \widehat{\boldsymbol{\Theta}}_S) d\boldsymbol{\beta}_{1,1} \dots d\boldsymbol{\beta}_{n_T,m} \\
&\propto \pi(\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_m, \gamma_1, \dots, \gamma_m, \mathbf{R}, \boldsymbol{\Sigma}) \cdot \\
&\quad \prod_{i=1}^{n_T} \int_0^1 \dots \int_0^1 \pi(y_{i,1}, \dots, y_{i,m} \mid F_{\beta_{i,1}}^{-1}(\mathbf{u}_{i,1}), \dots, F_{\beta_{i,m}}^{-1}(\mathbf{u}_{i,m}), \boldsymbol{\Sigma}, \widehat{\boldsymbol{\Theta}}_S) \cdot \\
&\quad c(\mathbf{u}_{i,1}, \dots, \mathbf{u}_{i,m} \mid \boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_m, \gamma_1, \dots, \gamma_m, \mathbf{R}) d\mathbf{u}_{i,1} \dots d\mathbf{u}_{i,m}.
\end{aligned}$$

A.11 Online Mean Derivation

When there are ℓ target data sets, the marginal posterior of $\boldsymbol{\alpha}$ is

$$p(\boldsymbol{\alpha} \mid \mathbf{y}_{T_\ell,1}, \dots, \mathbf{y}_{T_\ell, n_{T_\ell}}) \propto \pi(\boldsymbol{\alpha}) \cdot \left[\alpha_\ell \cdot k_\ell + \sum_{i=1}^{\ell-1} \alpha_i \cdot k_i \right].$$

Where $\pi(\boldsymbol{\alpha})$ is the prior on the vector $\boldsymbol{\alpha}$, α_ℓ is the weight associated with the uninformative prior term k_ℓ and α_i and k_i are the weights associated and posterior distribution terms associated with T_i , respectively, for $i \in 1, \dots, \ell - 1$. The normalizing constant is

$$\begin{aligned}
1 &= c \cdot \int_{\alpha_1} \dots \int_{\alpha_{\ell-1}} \pi(\boldsymbol{\alpha}) \cdot \left\{ \left(1 - \sum_{i=1}^{\ell-1} \alpha_i \right) k_\ell + \sum_{i=1}^{\ell-1} \alpha_i k_i \right\} d\alpha_{\ell-1} \dots d\alpha_1 \\
&= c \cdot \left\{ k_\ell \int_{\alpha_1} \dots \int_{\alpha_{\ell-1}} \pi(\boldsymbol{\alpha}) d\alpha_{\ell-1} \dots d\alpha_1 + (k_i - k_\ell) \int_{\alpha_1} \dots \int_{\alpha_{\ell-1}} \pi(\boldsymbol{\alpha}) \cdot \alpha_i d\alpha_{\ell-1} \dots d\alpha_1 \right\} \\
&= c \cdot \left\{ k_\ell + \sum_{i=1}^{\ell-1} (k_i - k_\ell) \mathbb{E}_{\pi(\boldsymbol{\alpha})}(\alpha_i) \right\} \\
c &= \left\{ k_\ell + \sum_{i=1}^{\ell-1} (k_i - k_\ell) \mathbb{E}_{\pi(\boldsymbol{\alpha})}(\alpha_i) \right\}^{-1}.
\end{aligned}$$

The posterior mean of α_j is

$$\begin{aligned}
\mathbb{E}(\alpha_j | \mathbf{y}_{T_\ell, 1}, \dots, \mathbf{y}_{T_\ell, n_{T_\ell}}) &= c \cdot \int_{\alpha_1} \cdots \int_{\alpha_{\ell-1}} \alpha_j \cdot \pi(\boldsymbol{\alpha}) \cdot \left[\left(1 - \sum_{i=1}^{\ell-1} \alpha_i\right) k_\ell + \sum_{i=1}^{\ell-1} \alpha_i k_i \right] d\alpha_{\ell-1} \dots d\alpha_1 \\
&= c \cdot \left\{ k_\ell \int_{\alpha_1} \cdots \int_{\alpha_{\ell-1}} \alpha_j \cdot \pi(\boldsymbol{\alpha}) d\alpha_{\ell-1} \dots d\alpha_1 + \right. \\
&\quad (k_j - k_\ell) \int_{\alpha_1} \cdots \int_{\alpha_{\ell-1}} \alpha_j^2 \cdot \pi(\boldsymbol{\alpha}) d\alpha_{\ell-1} \dots d\alpha_1 + \\
&\quad \left. \sum_{i=1, i \neq j}^{\ell-1} (k_i - k_\ell) \int_{\alpha_1} \cdots \int_{\alpha_{\ell-1}} \alpha_j \cdot \alpha_i \cdot \pi(\boldsymbol{\alpha}) \cdot d\alpha_{\ell-1} \dots d\alpha_1 \right\} \\
&= c \cdot \left[k_\ell \cdot \mathbb{E}_{\pi(\boldsymbol{\alpha})}(\alpha_j) + (k_j - k_\ell) \left\{ \mathbb{E}_{\pi(\boldsymbol{\alpha})}(\alpha_j)^2 + \text{Var}_{\pi(\boldsymbol{\alpha})}(\alpha_j) \right\} + \right. \\
&\quad \left. \sum_{i=1, i \neq j}^{\ell-1} (k_i - k_j) \left\{ \text{Cov}_{\pi(\boldsymbol{\alpha})}(\alpha_i, \alpha_j) + \mathbb{E}_{\pi(\boldsymbol{\alpha})}(\alpha_i) \mathbb{E}_{\pi(\boldsymbol{\alpha})}(\alpha_j) \right\} \right].
\end{aligned}$$