

Interpretable (not just posthoc-explainable) medical claims modeling for discharge placement to reduce preventable all-cause readmissions or death

Joshua C. Chang,^{1, 2} Ted L. Chang,^{1, 2} Carson C. Chow,^{3, 1} Rohit Mahajan,^{1, 2} Sonya Mahajan,^{1, 2} Joe Maisog,⁴ Shashaank Vattikuti,¹ Hongjing Xia,^{1, 2}

¹Mederrata Research, United States of America ²Sound Prediction Inc., United States of America

³NIH NIDDK, United States of America ⁴Blue Health Intelligence, United States of America

Abstract

We developed an inherently interpretable multilevel Bayesian framework for representing variation in regression coefficients that mimics the piecewise linearity of ReLU-activated deep neural networks. We used the framework to formulate a survival model for using medical claims to predict hospital readmission and death that focuses on discharge placement, adjusting for confounding in estimating causal local average treatment effects. We trained the model on a 5% sample of Medicare beneficiaries from 2008 and 2011, based on their 2009–2011 inpatient episodes, and then tested the model on 2012 episodes. The model scored an AUROC of approximately 0.76 on predicting all-cause readmissions (defined using official CMS methodology) or death within 30-days of discharge, being competitive against XGBoost and a Bayesian deep neural network, demonstrating that one need not sacrifice interpretability for accuracy. Crucially, as a regression model, we provide what blackboxes cannot – the exact gold-standard global interpretation of the model, identifying relative risk factors and quantifying the effect of discharge placement. We also show that the posthoc explainer SHAP fails to provide accurate explanations.

Introduction

Preventable readmission after hospital discharge is costly. In 2011, for adult 30-day all cause hospital readmission in the United States, the cost was about \$41.3 billion [Hines et al. 2014]. To improve outcomes, Medicare, through its Hospital Readmissions Reduction Program (HRRP) [McIlvannan, Eapen, and Allen 2015], penalizes providers for readmissions that occur within the 30-days after discharge; penalties have spurred interest in interventions surrounding transitions of care including discharge planning services such as hand-offs to less-intensive healthcare institutions. Population-level medical claims data make it possible to assess the efficacy of these interventions retroactively. This manuscript focuses on the problem of deciding discharge placement for individuals in order to prevent readmission or death.

Readmission models: A recent review [Huang et al. 2021] surveyed properties of readmission models in the literature. By and large, they found no model type to perform

consistently better than others, though several studies have reported marginal improvements using either XGBoost or neural networks over logistic regression [Shameer et al. 2016, Jamei et al. 2017, Allam et al. 2019, Liu et al. 2020, Min, Yu, and Wang 2019, Futoma, Morris, and Lucas 2015, Larsson et al. 2021]. Generally, the literature has focused on 30-day readmissions, though nuances in how readmission is defined complicate direct performance comparisons. Models based on medical claims data typically achieved AUROC of approximately 0.70 for predicting their version of all cause 30-day readmission.

Another factor that complicates the direct comparison of modeling efforts is differences in datasets – and hence the underlying patient populations. We are aware of two readmission studies performed on datasets identical to ours. Lah lou et al. [2021] created an attention-based neural network for predicting admissions after discharge within 30-days and reported an AUROC value of 0.81, however, they did not distinguish between transfers, planned admissions, and acute admissions in their outcome label so they solve a different problem that is of less practical utility. More related to our work, MacKay et al. [2021] developed XGBoost models for predicting a set of adverse events, reporting an AUROC of 0.73 for all-cause readmission prediction.

Yet, having a high AUROC is insufficient for making a model useful. A model needs transparency in order to assess its validity. Most studies surveyed were aware of the importance of model interpretability, claiming their blackbox model to be interpretable. Yet, most studies that claim interpretability for their blackbox solutions only offer “posthoc explainability,” a catch-all phrase for narratives generated in order to promote a sense that a model is interpretable when it is not.

Interpretability: The goal of interpretable modeling is to produce predictions that an end-user can understand [Rudin 2019, 2014], which is a prerequisite for making a prediction actionable. One necessary yet insufficient aspect of intrinsic model interpretability is feature attribution. Blackbox models do not admit feature attributions without the use of unreliable approximations. Conversely, feature attribution is exact in regression models, and each model coefficient has the unequivocal interpretation as the conditional expected change of the response corresponding to a given unit of change in the predictor, while fixing the other pre-

dictors. For this reason, even ignoring attributes other than feature attribution, a significant disconnect separates blackbox models from inherently interpretable models. Fig 1 is a representation of the spectrum of interpretability focusing on structured data problems in healthcare.

Computational interpretability is a necessary yet insufficient attribute for prediction comprehensibility. ReLU-activated neural networks, matrix composition methods like principle components analysis (PCA), and large multiple regression models are computationally interpretable, whereas Deep Learning (DL) models in-general and ensemble trees methods like XGBoost are not.

However, knowing how a prediction is computed from individual features does not automatically make the prediction comprehensible – it may still difficult to understand how a model behaves as there is a limit to the capacity of information that humans can process simultaneously [Miller 1956]. Sudjianto and Zhang [2021] note that additivity, sparsity, linearity, smoothness, monotonicity, and visualizability are some attributes of interpretable models that are also comprehensible.

The highest bar for interpretability is for a model to be mechanistically meaningful. These models often leverage domain knowledge and are capable of providing deep and robust insights. They also often justify causal interpretations [Peters, Bauer, and Pfister 2020]. Even if one can truly understand a model, one often cannot act on it. To be directly actionable, a model also needs to adjust for biases in the data so that its prediction of the effects of interventions can be interpreted causally [Pearl 2009]. Yet, independent of causal validity, predictive model interpretability is still important because it allows practitioners to better understand the risks and biases of a given model.

Posthoc explainable-AI (xAI): Posthoc xAI is a set of techniques to market blackbox models as interpretable (Fig. 1). The most popular xAI methods (LIME [Ribeiro, Singh, and Guestrin 2016] and SHAP [Lipovetsky and Conklin 2001, Datta, Sen, and Zick 2016]), use approximations [Lundberg and Lee 2017, Aas, Jullum, and Løland 2021] to provide narratives of feature importance within a prediction. Other methods such as attention [Niu, Zhong, and Yu 2021] build an explanation mechanism as a module within a blackbox model in order to more-easily compute them [Jain and Wallace 2019, Zhou et al. 2022]. Narratives, convincing as they might seem, are not necessarily true. In fact, researchers have shown [Laugel et al. 2019, Kumar et al. 2020, Slack et al. 2020, Alvarez-Melis and Jaakkola 2018, Zhou et al. 2022] that these methods provide imprecise and unreliable explanations of models, and often disagree. Aptly, Krishna et al. [2022] coined this the “disagreement problem” with posthoc interpretability and conducted a survey of real world data scientists finding no consistent or principled method to handle these inconsistencies. As Rudin [2019] notes, “an explanation model that is correct 90% of the time is wrong 10% of the time.” Despite marketing claims, xAI does not carry blackbox models across even a very minimal bar of requirements for interpretability. If an explanation is not true to one’s model, any sense that the model is comprehensible is based on faulty information.

Blackbox models: Methods such as Deep Learning (DL) and ensemble boosted trees (XGBoost, LightGBM, others) can model nonlinearities. When copious training data is available, these methods yield models that are more expressive than traditional generalized linear models. Most-generally, blackbox models like DL and ensemble trees are nonlinear kernel machines (function interpolations) [Domingos 2020]. The convoluted nature of their interpolations makes these models uninterpretable. Massive investment exists in these models because of their predictive performance and low effort. This existing investment, the challenge of creating truly interpretable models, and a myth that blackboxes perform better than interpretable models [Rudin 2019], incentivize the marketing of posthoc-xAI as an alternative to interpretable modeling. In finance, a similarly high-stakes domain, there has been wide resistance to blackbox modeling, formalized recently in model risk management guidelines published by The Office of the Comptroller of the Currency (OCC) [2021]. We should also be wary of the use of these models in healthcare, where the risk to patients requires truly trustworthy solutions.

Blackboxes provide clues on how to extend traditional linear models. DL is the application of artificial neural networks (ANNs) to prediction problems. ANNs consist of sequences (or more generally of graphs) of successive affine matrix arithmetic operations, sandwiched between activation functions. In general, these methods are blackboxes, with the exception of ReLU-activated neural networks (ReLU-nets for short). Examining ReLU-nets elucidates the nature of how DL captures nonlinearities. ReLU-nets use the activation function $\text{ReLU}(x) = 0$ if $x \leq 0$ or x otherwise. In these models, ReLU is independently applied to each matrix coordinate after each successive matrix operation. The output of the function is nonzero if and only if a linear combination of the elements computed by the prior layer are positive. Hence, ReLU defines an inequality over quantities within the model – applied to each coordinate within each layer, ReLU defines recursive sets of inequalities. These inequalities collectively segment the training data into disjoint regions. In sum, ReLU-nets are composed of regionally-disjoint generalized linear models – each of which is interpreted in the same manner as linear regression. Hence, ReLU-nets are computationally interpretable. The salient nonlinearity of these models is locality. To interpret a specific prediction given by these models, one needs to map the input to a particular linear submodel. Then, conditional on this mapping, a ReLU net is locally a simple generalized multiple linear regression model. Observing this fact, Sudjianto et al. [2020] provides a tool for exactly interpreting trained ReLU neural networks, by unwrapping the cascades of inequalities. In this manuscript we mimic this property of ReLU-nets within a well-controlled multilevel Bayesian regression framework in order to gain expressiveness while prioritizing interpretability.

Methods

We generalize the classic readmission problem of within 30 days of discharge, to the likelihood of readmission at any arbitrary day after discharge. To this end, our objective is

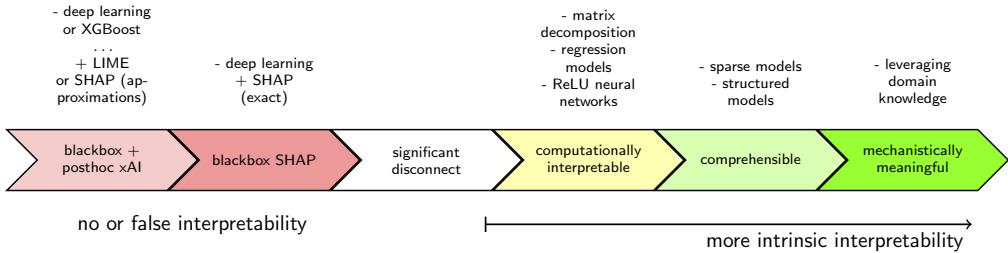


Figure 1: **Model interpretability** lies along a spectrum with a clear chasm existing between intrinsically interpretable models others. More-interpretable models are more trustworthy and insightful. Explanations for models without intrinsic interpretability rely on unreliable approximations to their computations.

to characterize the statistics of the inter-inpatient wait time T_n . Additionally, we focus on identifying the effects of discharge placement, representing the choices symbolically as I_n , ranked in terms of health acuity: (0) discharge home, (1) discharge home with home health, (2) discharge to skilled nursing, (3) intermediate care/critical access, (4) long term care, (5) other less-acute inpatient. The issue that complicates the estimation of discharge placement effects is unobserved confounding – providers use the patient’s health status in order to decide placement. To resolve the treatment assignment bias, we model the joint outcomes

$$\begin{aligned} T_n &\sim f(T_n | \mathbf{x}_n, \mathbf{I}_n, \boldsymbol{\alpha}_n, \boldsymbol{\beta}_n, \boldsymbol{\gamma}_n) \\ I_n &\sim g(I_n | \mathbf{x}_n, \boldsymbol{\nu}_n, \boldsymbol{\xi}_n), \end{aligned} \quad (1)$$

where $\mathbf{x}_n \in \mathbb{R}^p$ is a covariate vector and we explicitly adjust for assignment bias. Note that we distinguish between the scalar-valued I_n , which corresponds to the list of interventions above, and the vector valued \mathbf{I}_n which we will explain later in this manuscript. For the sake of interpretability, we formulate f and g in Eq. 1 as hierarchical multilevel Bayesian generalized linear regression models. However, to increase expressivity by introducing the type of nonlinearity seen in ReLU-nets, we allow all of the model parameters $\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n, \boldsymbol{\gamma}_n, \boldsymbol{\nu}_n, \boldsymbol{\xi}_n$ to vary locally [Hastie and Tibshirani 1993, Fan and Zhang 2008, Li, Li, and Feng 2021] across regions defined by \mathbf{x}_n , in ways that comport with domain knowledge.

Data Preprocessing: The available dataset, the CMS Limited Dataset (CMS LDS), consists of a national 5% beneficiary sample of Medicare FFS Part A and B claims from 2008 to 2012. The 2008 claims had only quarter date specificity so we used them solely to fill out the medical history for 2009 inpatient stays, by assuming that each 2008 claim fell in the middle of its given quarter. We trained the readmission models on 2009 – 2011 admissions, and evaluated the models on 2012 admissions.

After grouping claims into coherent episodes, based on date, provider, and patient overlap, we filtered for inpatient-specific episodes with certain characteristics to use as index admissions. We retained only episodes where the patient had a continuous prior year of Part A/B enrollment. We also excluded episodes from consideration as index episodes if they did not correspond to discharges to less-intensive care (excluding death and most inpatient-to-inpatient transfers). Ad-

ditionally, we used the official CMS methodology for determining whether each episode is a planned admission, acute admission, or potentially planned admission [CMS 2015]. For each episode we then computed the waiting time to either the next unplanned acute episode or death, or until censorship due to the end of the observation window. In the end, the training dataset consisted of approximately 1.2 million inpatient episodes, of which approximately 17% were followed by an unplanned acute inpatient episode or death within 30 days. The histogram of the wait times is presented in Fig. 2.

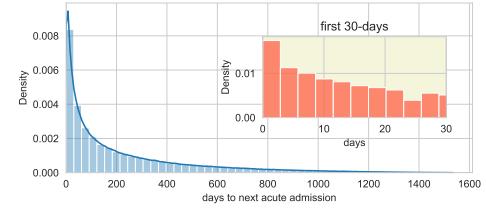


Figure 2: **Histogram of the wait time after discharge to the next acute admission or death.**

For each episode, we collected all billing codes, creating lists of concurrent procedure and diagnostic codes. Additionally, we collected the preceding four quarters of history for each episode, aggregating billing codes on a lagged quarterly basis.

Feature engineering: Medical claims data consists of series of billing codes in several dialects (ICD9/10, HCPCS, RUG, HIPPS, etc). We down-sampled diagnostic (Dx) and procedure (Tx) codes, from their original dialects to multilevel Clinical Classification Software (CCS) codes [AHRQ 2022]. CCS codes are clinically curated hierarchical categories that are more tractable for analysis and interpretation. Mapping to CCS drastically removes redundancy in the vocabulary of the dataset and helps to extract the health-specific information in billing codes buried noisy reimbursement-specific details.

We used AHRQ Healthcare Cost and Utilization Project (HCUP) databases in order to tag codes for comorbidities, chronic conditions, surgical flags, utilization flags, and procedure flags. Included within skilled nursing facility (SNF) and home health (HH) claim codes are also activities of

daily living (ADL) assessments. We converted these codes to ADL scores, where higher scores correspond to lower functional ability. We also incorporated CMS’s risk adjustment methodology, hierarchical condition categories (HCC), as model predictors. The CMS LDS contains beneficiary county codes that we used to incorporate the urban rural index and social economic scale as model features. Together with beneficiary race information and Medicaid state buy-in, these variables allowed for some measure of social determinants of health.

We encoded CCS and other code mappings into numerical vectors by counting the incidences of each permissible code. In the case of CCS, which is multilevel, we truncated codes at each of the first two levels and counted at each level. Altogether, the numerically encoded derived features constituted a vector of size $p = 1072$, which encompassed both concurrent episode codes and the past four quarters of history, where CCS was truncated to the first level for history.

Feature quantization: To improve model interpretability, we made an effort to place all model parameters (log hazard ratios) on the same scale so that the magnitudes of the parameters are directly comparable. In examining our derived data features, we came to find that they were predominantly sparse and heavy tailed. This finding, and our desire to optimize model interpretability, led us to quantize all numerical variables so that the input variables into the model are entirely binary. To this end, we first computed the percentiles for each feature across the entire dataset. Then we re-coded each quantity into a series of binary variables corresponding to inequalities, where the cutoffs were determined by examining each variables at a set of quantiles and eliminating duplicate values. Generally, we retained only the quantized features in specifying the models except when otherwise specified. The total size of the feature vector after dropping all original non-quantized numerical features and all constant features expanded to $p = 3143$.

Survival Modeling: For flexibly modeling the wait time distribution f , we use the piecewise exponential survival regression model (PEM) [Friedman 1982]. PEMs are defined by specifying the time-dependent hazard $\lambda(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ using a piecewise constant function, where the hazard changes across breakpoints that define disjoint time intervals. The probability density function for the PEM follows $f(t) = \lambda(t)e^{-\int_0^t \lambda(u)du}$. In this manuscript we set the breakpoints between time intervals at 1 week, 4 weeks, and 9 weeks after discharge. For each episode n , we can estimate a wait time distribution by estimating the log-hazard within each time interval i ,

$$\log \lambda_{ni} = \alpha_{ni} + \beta'_{ni} \mathbf{x}_n + \gamma'_{ni} \mathbf{I}_n, \quad (2)$$

where we allow the model parameters to vary across the data regionally, in a manner that emulates the type of nonlinearity seen in ReLU-nets. In Eq. 2 we separate out the discharge placement effects (γ_n) from other effects (β_n). Doing so makes it easier to structure the model for causally interpreting the discharge assignment effects. We incorporate domain knowledge by acuity-ordering the interventions, enforcing monotonicity of intervention effect by constraining the last five coefficients of ξ_n to non-positivity.

Causal inference: We model the discharge placement process g using an ordinal logistic regression model, where

$$\begin{aligned} I_n | \mathbf{p}_n &\sim \text{Categorical}(p_{n0}, \dots, p_{n5}) \\ p_{nk} | \mathbf{x}_n, \boldsymbol{\nu}_n, \boldsymbol{\xi}_n &= \Pr(I_n \geq k | \mathbf{x}_n, \boldsymbol{\nu}_n, \boldsymbol{\xi}_n) \\ &\quad - \Pr(I_n \geq k + 1 | \mathbf{x}_n, \boldsymbol{\nu}_n, \boldsymbol{\xi}_n) \\ \Pr(I_n \geq k | \mathbf{x}_n, \boldsymbol{\nu}_n, \boldsymbol{\xi}_n) &= \text{logit}^{-1}(\nu_{nk} + \xi'_n \mathbf{x}_n), \end{aligned} \quad (3)$$

where ξ_n are slopes corresponding to episode n and $\boldsymbol{\nu}_n = [\nu_{n1}, \dots, \nu_{n5}]$ are intercepts under the constraints $\nu_{nk} < \nu_{n,k+1}, \forall k, n$. The predictions given by this model then feed back into the prediction of the wait time through a slope term for each element of the covariate vector $\mathbf{I}_n = [\Pr(I_n \geq 1 | \dots), \dots, \Pr(I_n \geq 5 | \dots), 1_{I_n \geq 1}, \dots, 1_{I_n \geq 5}]$. Utilizing the discharge placement probabilities as model covariates adjusts for the confounding bias caused by the selection process, in a manner analogous to incorporating the local treatment probability as a covariate [Bafumi and Gelman 2007]. Additionally, directly modeling the treatment effects within a multilevel model allows us to infer locally-varying treatment effects, partially pooled for stable inference in regions where the data is sparse [Gelman 2006, Feller and Gelman 2015].

Parameter decomposition: The piecewise linear nature of ReLU-nets, and the observation that neural networks produce learned data representations [Goodfellow, Bengio, and Courville 2016], suggest that an approach to mimicking their expressivity within regression models is to allow slopes (and intercepts) to vary across regions of the data. We do so by expressing each of these regionally-varying parameters using an additive decomposition.

First, for delineating regions in data space (corresponding to cohorts), we project portions of the input data to lower dimensions through unsupervised methods. In Chang et al. [2020], the authors make a connection between sparse probabilistic matrix factorization and probabilistic autoencoders. We use this approach to develop a low-dimensional representation of the portions of the input covariate vector that pertain to the lagged quarterly history. Then, we compute the statistics of the learned representation in the training data and develop for each dimension a set of cut-offs to use for bucketization. This procedure puts each inpatient episode into a specific cohort, represented by a location within a multidimensional lattice, based on medical history. Specifically, we used a single cut-off (the median) for each of five dimensions (Fig. 3), creating a set of $2^5 = 32$ groups based on history. By design, the rules governing the group assignment can be easily converted to a set of inequalities over sparse subsets of the original data features. Additionally, we included interactions between the history groups with other discrete attributes such as the major diagnostic category (MDC), complication or comorbidity (CC) or a major complication or comorbidity (MCC), and race, to create high dimensional discrete lattices where the cells define coarse interaction cohorts in the data. When partitioning data by a high-order interaction, a big data problem quickly becomes many small data problems – divide-and-conquer approaches can suffer from overfitting. To combat this issue, we developed a multiscale modeling approach where higher-order interactions are regularized by partially pooling their effects into related lower-order interactions. Specifically, given a multidimensional lattice that represents all cohorts for which

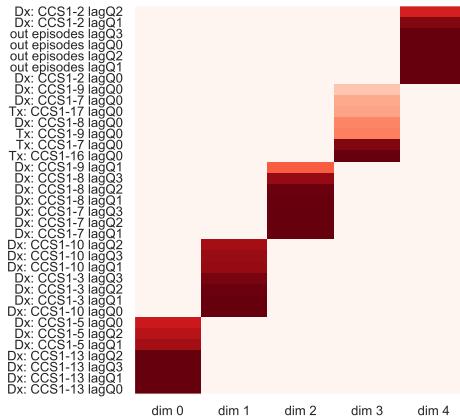


Figure 3: History encoding weights for inpatient episodes
where the top seven variables for each dimension of a five dimensional factorization are shown. The features consist of multilevel CCS counts of diagnoses and procedures, as well as counts of the number of episodes, on a quarter-lagged basis. An episodes history representation is found by linear combination of history count features of the given weights.

the parameter will vary, we assign for each parameter a value within the lattice by decomposing the value into the form

$$\theta^{(\kappa)} = \overbrace{\theta^{(*,*,\dots,*)}}^{\text{zero order}} + \overbrace{\theta^{(\kappa_1,*,\dots,*)} + \theta^{(*,\kappa_2,*,\dots,*)} + \dots}^{\text{first order}} + \overbrace{\theta^{(\kappa_1,\kappa_2,*,\dots,*)}}^{\text{second order}} + \theta^{(\kappa_1,\kappa_2,\kappa_3,*,\dots,*)} + \dots + \text{H.O.T.}, \quad (4)$$

where $\kappa = (\kappa_1, \kappa_2, \dots, \kappa_D)$ is a D dimensional multi-index. In practice, we truncate the maximum order of terms in this decomposition due to memory constraints. More details on the exact decompositions that we used for our model parameters can be found in the Supplemental Materials.

Regularization: By design, the parameter decomposition method inherently regularizes by partial pooling [Gelman 2006]. Additionally, we used weakly informative priors on the component tensors in these decompositions in order to encourage shrinkage at higher orders. For the regression coefficients, we utilized the horseshoe prior for local-global shrinkage [Ghosh and Doshi-Velez 2017, Bhadra et al. 2019, Polson and Scott 2011, van Erp, Oberski, and Mulder 2019]. Please see the Supplemental Materials for more details on the model specification.

Implementation: We used Tensorflow Probability [Dillon et al. 2017], developing a set of libraries for managing the parameter decompositions that is publicly available at [github:mederrata/bayesianquilts](https://github.com/mederrata/bayesianquilts).

We trained our model using minibatch mean-field stochastic ADVI, using batch sizes of 10^4 , and a parameter sample size of 8 for approximating the variational loss function. We utilized the Adam optimizer with a starting learning rate of 0.0015, embedded within a lookahead optimizer [Zhang et al. 2019] for stability. Each epoch where the mean batch loss did not decrease, we set the learning rate to decay by 10%. Training was set to conclude if there was

Model	Interpretability	AUROC	AUPRC
XGBoost	None	0.758	0.497
ReLU-BNN	Computationally	0.757	0.500
LR w/o quantization	Comprehensibly	0.731	0.381
LR	Comprehensibly	0.755	0.481
PEM	Mechanistically	0.757	0.497

Table 1: 30-day unplanned readmission or death classification metrics for evaluated models: XGBoost, Sparse logistic regression (LR), Bayesian neural network (BNN), our Piecewise exponential model (PEM). Quantization refers to the histogram-based bucketization of real-valued features. Area under the receiver operator curve (AUROC) and area under the precision-recall curve (AUPRC) computed on held-out 2012 inpatient episodes. Models trained on 2009-2011 episodes. Interpretability judged according to Fig. 1.

no improvement for 5 epochs, or if we reached 100 epochs, whichever came sooner. More information on the training is present in the Supplemental Materials. We used scikit-learn 1.1.1 for fitting baseline logistic regression models, and XGBoost 1.6.1 for fitting a reference blackbox model for comparison. We implemented a horseshoe Bayesian convolutional neural network with ReLU activation using TFP, where we used a single hidden layer of size one-fifth the input layer. For computing global SHAP values, we used regression-based KernelSHAP [Covert and Lee 2021]. All computation was performed using the Pittsburgh Supercomputing Center’s Bridges2 resources. We utilized extreme memory (EM) nodes for preprocessing, and Bridges2-GPU-AI for training.

Results

Prediction Accuracy: Table. 1 shows the classification accuracy of our model in predicting readmissions or death within the first 30 days, benchmarked against predictions given by alternative models trained on the same dataset. The standard deviation in both the AUROC and AUPRC measures, as determined using bootstrap, was approximately 0.003. Quantization of the data features improved the accuracy of logistic regression to nearly match that of XGBoost on this dataset as measured by AUROC. Hence, we used quantization for features in both the Bayesian neural network (BNN) and piecewise exponential (PEM) models. The Bayesian neural network we developed utilizes sparsity-inducing horseshoe priors [Carvalho, Polson, and Scott 2010] on the weights and biases, which has been shown to improve model performance [Bhadra et al. 2019].

Interpretation: In addition to being competitive with blackbox methods in terms of prediction accuracy, our model, as a generalized linear survival regression model, is easily interpretable. To be specific, our model is a generalized linear survival model where the coefficients vary. The value of each coefficient is the logarithm of a hazard ratio corresponding to the effect of a given feature, for a given data cohort, for a given time period. Log hazards greater than zero correspond to increased probability of event (readmission or death). Here, we provide select portions of the *ground-truth* global interpretation of the model, found by simply reading

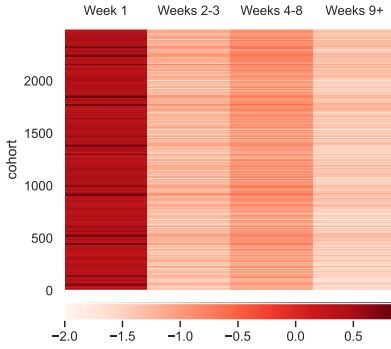


Figure 4: Mean baseline log-hazards by week for each episode interaction cohort defined within the model. Larger log-hazards corresponds to more readmission risk. Personalized values of α_n specific to each episode are found by mapping an episode into its cohort grouping.

off the values of the regression coefficients. Please see the Supplemental Materials for a more-complete accounting of the model. This type of exposition is impossible with black-box models without relying on unreliable approximations.

Time-dependent risk factors: The model segments the data based on covariability and assigns for each predictor a cohort-level effect within each time interval. The effects within the model are hazard ratios, which describe the instantaneous relative risk associated with a predictor relative to a baseline. In most cases, the baseline refers to a typical or normal value of a variable. Membership to cohorts also itself is associated with a baseline risk – baseline log-hazards are presented in Fig. 4 for the 12480 episode cohort types defined within the decomposition for the parameter vector α_n in Eq. 2. Larger values of the hazard imply higher probability of event (readmission or death). There exists variability in the hazards across cohorts (rows), though the most obvious change is in time. Generally, the hazard is greatest in the first week after discharge. This finding implies that patients are more vulnerable in the first week than afterwards – keeping a patient out of the hospital within the first week reduces the overall risk that they will die or be readmitted. For this reason, we will focus on understanding the model’s predictions of the first-week risk.

The 40 most-impactful first-week factors are shown in Fig. 5, where the parameters have been decomposed in order to control for racial biases. The most-predictive single feature was length of stay. Lengths of stay less than a full day had a relative log hazard ratio of 0.97 (95% CI: 0.96 – 0.98) (note LOS<1 day was the reference group and so is the converse of LOS ≥ 1 days shown in Fig. 5). Having an acute primary diagnosis code, at least one inpatient stay in the previous quarter (lagQ0, within 90 days of admit), and discharge against medical advice were also strong predictors associated with increased risk of readmission or death. Patients who received skilled nursing care in the quarter preceding an inpatient episode, who had a Resource Utilization Group (RUG) Activities of Daily Living (ADL) score of at least 6.125 tended to have a lower risk of readmission in the

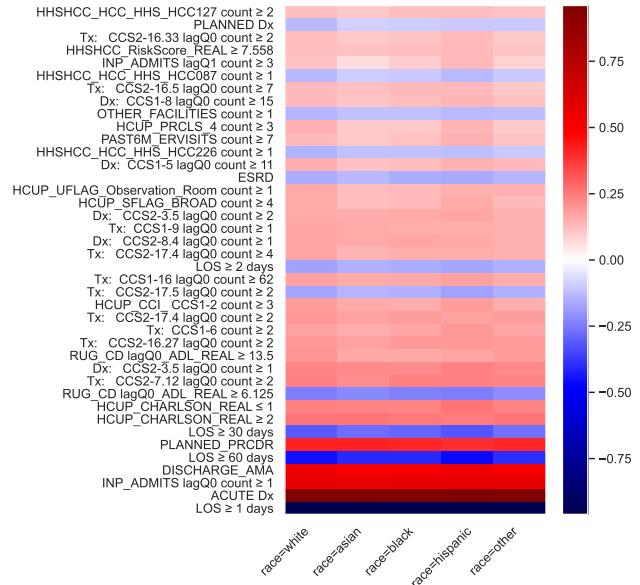


Figure 5: The 40 predictors with the largest absolute coefficients in the first week (through day 7) after readmission. All predictors are binary and all parameters are additive log hazard ratios. Higher (red) corresponds to larger hazards and greater readmission risk.

first week than otherwise, however, the risk increased for quarter-lagged ADL scores of at least 13.5.

Discharge placement effects: In Fig. 6, we show the cohort-wise causally-adjusted mean local average treatment effects of discharge to each of the given care settings as well as the local standard deviation in the effect. Focusing on the effect of discharging to skilled nursing care, the effects were greatest for episodes graded by DRG code as having either a complication or comorbidity (CC) or a major complication or comorbidity (MCC). In particular, CC/MCC episodes with a major diagnostic code of 2 (Diseases and Disorders of the Eye), 14 (Pregnancy, Childbirth And Puerperium), and 22 (Burns) have the greatest response to discharge to skilled nursing.

Posthoc-xAI (SHAP) misleads: Knowing what the model is doing in exact terms, let us see how posthoc-xAI thinks the model is working. In Fig. 7 we display the most important model features as determined by magnitude of global SHAP values in the prediction of readmission or death within the first 30 days. SHAP is computationally costly to approximate – the details of our SHAP computation are available in the Supplemental Materials. The four most-influential features according to the explainer are specific CCS classes of treatments and diagnoses in the recent quarterly history. Comparing these results to the parameter values of Fig. 5, it is evident that the feature sets disagree. Nor do the values in Fig. 7 align with parameter values for later weeks (see Supplemental Materials). This finding is unsurprising; SHAP has been consistently shown to fail to recover ground-truth interpretations [Kumar et al. 2020, Bordt et al. 2022],

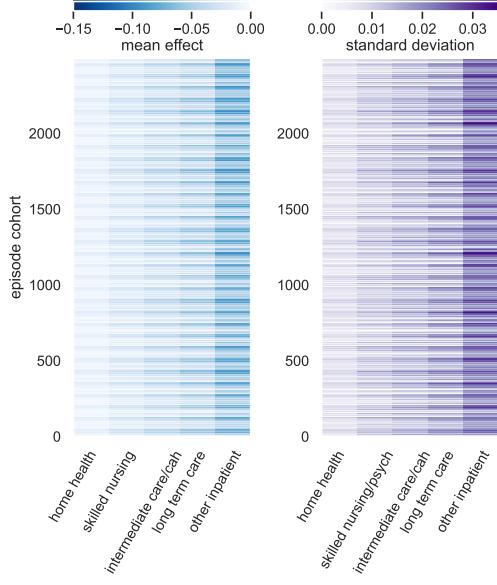


Figure 6: First-week effects of discharge placement: Mean (left) and standard deviation (right) by cohort (row) of the five placement interventions assessed, in increasing order of implied acuity. Effect is difference in log-hazard relative to a normal discharge (home).

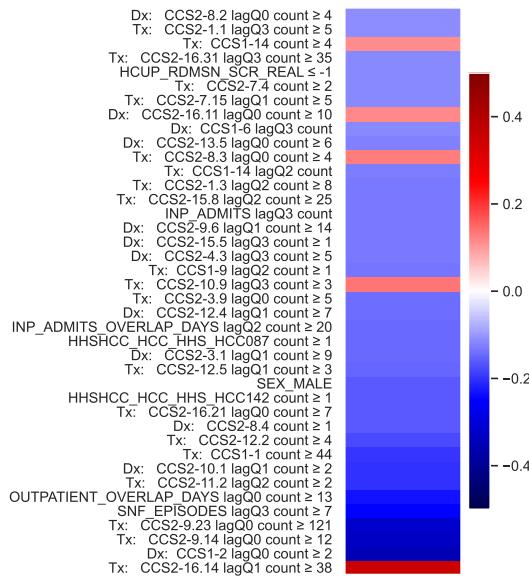


Figure 7: Shapley values: Top 40 absolute feature weights for prediction of 30 day readmission using our survival model, where we have the ground truth explanation (See Figs 3–6, and the supplement for how the features actually are incorporated into the model. SHAP fails to identify the features a model is using whenever features are correlated.

in problems where the predictors are correlated. SHAP fundamentally does not answer the question of what a given model is doing in order to reach a prediction. Furthermore, feature importance is not grounded in any relevant units and also does not speak to relevant interactions that are captured in a model. We criticize SHAP because it is one of the most popular posthoc-xAI techniques, however, similar arguments hold for other techniques [Rudin 2019, Babic et al. 2021, Zhou et al. 2022].

Discussion

We presented a method for mimicking ReLU-nets within inherently interpretable multilevel Bayesian models. We applied this methodology to the prediction of hospital readmissions or death after discharge, and to the causal inference of the effects of discharge assignments.

Accuracy without blackboxes: We demonstrated how we were able to perform like blackboxes, without sacrificing interpretability. We accomplished this feat through two classes of methods: First, our novel modeling framework allowed us some fine-grain resolution in looking at the differential effects of the predictors in data subgroups. Additionally, it helped regularize the inference of local average treatment effects for choosing discharge placement. Second, we performed layers of feature engineering. The first layer was an extraction of medically-relevant information from the raw billing that gave us attributes such as chronic diseases, comorbidities, and ADL function. Second, we reduced noise in the raw coding by mapping to the clinically-relevant CCS system. These two steps were sufficient for our logistic regression model to match the performance of an XGBoost model in the literature based on the same dataset [MacKay et al. 2021]. Finally, we performed feature quantization based on the per-feature statistics. Quantization led to a big performance increase in logistic regression and also in the neural network for a given model size. We took these lessons and used them in defining our interpretable survival model.

Posthoc xAI is inherently untrustworthy: Our model, being inherently interpretable, admits an unequivocal ground-truth explanation. Hence, it is a good test case for testing the accuracy of posthoc explainers. We tested SHAP on our model; it failed in coming close to the ground-truth. This finding is consistent with other literature that has looked critically at SHAP and other xAI tools.

While posthoc-xAI does not make blackboxes interpretable, interpretability is not always unnecessary. Quantifying sample average treatment effects and making predictions does not require interpretable modeling [Hill 2011], or even necessarily models at all [Ding and Miratrix 2017]. Blackbox methods offer good performance with minimal thoughtfulness. For these reasons, blackbox methods remain inherently useful – so long as one does not whitewash them with false explainability.

Limitations: Our modeling approach has downsides. Numerical stability generally requires the use of double precision floating point. The lattice-based parameter decomposition is memory-intensive which in some applications may severely limit expressivity.

Acknowledgments

We thank the Innovation Center of the Center for Medicare and Medicaid services for providing access to the CMS Limited Dataset through DUA LDSS-2019-54177. We also thank Dr. Pei-Shu Ho for help in understanding Medicare billing data. CCC is supported by the Intramural Research Program of the NIH, NIDDK. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) [Towns et al. Sept.-Oct. 2014], which is supported by National Science Foundation grant number ACI-1548562 through allocation TG-DMS190042.

References

- Aas, K.; Jullum, M.; and Løland, A. 2021. Explaining Individual Predictions When Features Are Dependent: More Accurate Approximations to Shapley Values. *Artificial Intelligence*, 298: 103502.
- AHRQ. 2022. HCUP-US Tools & Software Page. <https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccsfactsheet.jsp>.
- Allam, A.; Nagy, M.; Thoma, G.; and Krauthammer, M. 2019. Neural Networks versus Logistic Regression for 30 Days All-Cause Readmission Prediction. *Scientific Reports*, 9(1): 9277.
- Alvarez-Melis, D.; and Jaakkola, T. S. 2018. On the Robustness of Interpretability Methods. *arXiv:1806.08049 [cs, stat]*.
- Babic, B.; Gerke, S.; Evgeniou, T.; and Cohen, I. G. 2021. Beware Explanations from AI in Health Care. *Science*, 373(6552): 284–286.
- Bafumi, J.; and Gelman, A. 2007. Fitting Multilevel Models When Predictors and Group Effects Correlate.
- Bhadra, A.; Datta, J.; Li, Y.; and Polson, N. G. 2019. Horseshoe Regularization for Machine Learning in Complex and Deep Models.
- Bordt, S.; Finck, M.; Raidl, E.; and von Luxburg, U. 2022. Post-Hoc Explanations Fail to Achieve Their Purpose in Adversarial Contexts. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 891–905.
- Carvalho, C. M.; Polson, N. G.; and Scott, J. G. 2010. The Horseshoe Estimator for Sparse Signals. *Biometrika*, 97(2): 465–480.
- Chang, J. C.; Fletcher, P.; Han, J.; Chang, T. L.; Vattikuti, S.; Desmet, B.; Zirikly, A.; and Chow, C. C. 2020. Sparse Encoding for More-Interpretable Feature-Selecting Representations in Probabilistic Matrix Factorization. In *International Conference on Learning Representations*.
- CMS. 2015. 2015 Measure Information About the 30-Day All-Cause Hospital Readmission Measure, Calculated for the Value-Based Payment Modifier Program — Guidance Portal. <https://www.hhs.gov/guidance/document/2015-measure-information-about-30-day-all-cause-hospital-readmission-measure-calculated>.
- Covert, I.; and Lee, S.-I. 2021. Improving KernelSHAP: Practical Shapley Value Estimation via Linear Regression. *arXiv:2012.01536*.
- Datta, A.; Sen, S.; and Zick, Y. 2016. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *2016 IEEE Symposium on Security and Privacy (SP)*, 598–617.
- Dillon, J. V.; Langmore, I.; Tran, D.; Brevdo, E.; Vasudevan, S.; Moore, D.; Patton, B.; Alemi, A.; Hoffman, M.; and Sauvage, R. A. 2017. TensorFlow Distributions. *arXiv:1711.10604 [cs, stat]*.
- Ding, P.; and Miratrix, L. W. 2017. Model-Free Causal Inference of Binary Experimental Data. *arXiv:1705.08526*.
- Domingos, P. 2020. Every Model Learned by Gradient Descent Is Approximately a Kernel Machine. *arXiv:2012.00152 [cs, stat]*.
- Fan, J.; and Zhang, W. 2008. Statistical Methods with Varying Coefficient Models. *Statistics and its interface*, 1(1): 179–195.
- Feller, A.; and Gelman, A. 2015. Hierarchical Models for Causal Effects. In *Emerging Trends in the Social and Behavioral Sciences*, 1–16. John Wiley & Sons, Ltd. ISBN 978-1-118-90077-2.
- Friedman, M. 1982. Piecewise Exponential Models for Survival Data with Covariates. *The Annals of Statistics*, 10(1): 101–113.
- Futoma, J.; Morris, J.; and Lucas, J. 2015. A Comparison of Models for Predicting Early Hospital Readmissions. *Journal of Biomedical Informatics*, 56: 229–238.
- Gelman, A. 2006. Multilevel (Hierarchical) Modeling: What It Can and Cannot Do. *Technometrics*, 48(3): 432–435.
- Ghosh, S.; and Doshi-Velez, F. 2017. Model Selection in Bayesian Neural Networks via Horseshoe Priors. *arXiv:1705.10388 [stat]*.
- Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. ISBN 978-0-262-33737-3.
- Hastie, T.; and Tibshirani, R. 1993. Varying-Coefficient Models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(4): 757–796.
- Hill, J. L. 2011. Bayesian Nonparametric Modeling for Causal Inference. *Journal of Computational and Graphical Statistics*, 20(1): 217–240.
- Hines, A.; Barrett, M.; Jiang, J.; and Steiner, C. 2014. Conditions With the Largest Number of Adult Hospital Readmissions by Payer, 2011. Technical Report #172.
- Huang, Y.; Talwar, A.; Chatterjee, S.; and Aparasu, R. R. 2021. Application of Machine Learning in Predicting Hospital Readmissions: A Scoping Review of the Literature. *BMC Medical Research Methodology*, 21(1): 96.
- Jain, S.; and Wallace, B. C. 2019. Attention Is Not Explanation. *arXiv:1902.10186*.
- Jamei, M.; Nisnevich, A.; Wetchler, E.; Sudat, S.; and Liu, E. 2017. Predicting All-Cause Risk of 30-Day Hospital Readmission Using Artificial Neural Networks. *PLoS ONE*, 12(7).
- Krishna, S.; Han, T.; Gu, A.; Pombra, J.; Jabbari, S.; Wu, S.; and Lakkaraju, H. 2022. The Disagreement Problem in Explainable Machine Learning: A Practitioner’s Perspective.

- Kumar, I. E.; Venkatasubramanian, S.; Scheidegger, C.; and Friedler, S. 2020. Problems with Shapley-value-based Explanations as Feature Importance Measures. *arXiv:2002.11097 [cs, stat]*.
- Lahlou, C.; Crayton, A.; Trier, C.; and Willett, E. 2021. Explainable Health Risk Predictor with Transformer-based Medicare Claim Encoder. *arXiv:2105.09428*.
- Larsson, A.; Berg, J.; Gellerfors, M.; and Gerdin Wärnberg, M. 2021. The Advanced Machine Learner XGBoost Did Not Reduce Prehospital Trauma Mistriage Compared with Logistic Regression: A Simulation Study. *BMC Medical Informatics and Decision Making*, 21(1): 192.
- Laugel, T.; Lesot, M.-J.; Marsala, C.; Renard, X.; and Detyniecki, M. 2019. The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations. *arXiv:1907.09294*.
- Li, F.; Li, Y.; and Feng, S. 2021. Estimation for Varying Coefficient Models with Hierarchical Structure. *Mathematics*, 9(2): 132.
- Lipovetsky, S.; and Conklin, M. 2001. Analysis of Regression in Game Theory Approach. *Applied Stochastic Models in Business and Industry*, 17(4): 319–330.
- Liu, W.; Stansbury, C.; Singh, K.; Ryan, A. M.; Sukul, D.; Mahmoudi, E.; Waljee, A.; Zhu, J.; and Nallamothu, B. K. 2020. Predicting 30-Day Hospital Readmissions Using Artificial Neural Networks with Medical Code Embedding. *PLoS ONE*, 15(4).
- Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- MacKay, E. J.; Stubna, M. D.; Chivers, C.; Draugelis, M. E.; Hanson, W. J.; Desai, N. D.; and Groeneveld, P. W. 2021. Application of Machine Learning Approaches to Administrative Claims Data to Predict Clinical Outcomes in Medical and Surgical Patient Populations. *PLOS ONE*, 16(6): e0252585.
- McIlvennan, C. K.; Eapen, Z. J.; and Allen, L. A. 2015. Hospital Readmissions Reduction Program. *Circulation*, 131(20): 1796–1803.
- Miller, G. A. 1956. The Magical Number Seven plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological review*.
- Min, X.; Yu, B.; and Wang, F. 2019. Predictive Modeling of the Hospital Readmission Risk from Patients' Claims Data Using Machine Learning: A Case Study on COPD. *Scientific Reports*, 9(1): 2362.
- Niu, Z.; Zhong, G.; and Yu, H. 2021. A Review on the Attention Mechanism of Deep Learning. *Neurocomputing*, 452: 48–62.
- Pearl, J. 2009. Causal Inference in Statistics: An Overview. *Statistics Surveys*, 3(none): 96–146.
- Peters, J.; Bauer, S.; and Pfister, N. 2020. Causal Models for Dynamical Systems. *arXiv:2001.06208*.
- Polson, N. G.; and Scott, J. G. 2011. *Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction* *. Oxford University Press. ISBN 978-0-19-173192-1.
- Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv:1602.04938 [cs, stat]*.
- Rudin, C. 2014. Algorithms for Interpretable Machine Learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 1519. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-2956-9.
- Rudin, C. 2019. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence*, 1(5): 206–215.
- Shameer, K.; Johnson, K. W.; Yahi, A.; Miotto, R.; Li, L.; Ricks, D.; Jebakaran, J.; Kovatch, P.; Sengupta, P. P.; Gelijns, S.; Moskovitz, A.; Darrow, B.; David, D. L.; Kasarskis, A.; Tatonetti, N. P.; Pinney, S.; and Dudley, J. T. 2016. Predictive Modeling of Hospital Readmission Rates Using Electronic Medical Record-Wide Machine Learning: A Case-Study Using Mount Sinai Heart Failure Cohort. In *Bio-computing 2017*, 276–287. WORLD SCIENTIFIC. ISBN 978-981-320-780-6.
- Slack, D.; Hilgard, S.; Jia, E.; Singh, S.; and Lakkaraju, H. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post Hoc Explanation Methods. *arXiv:1911.02508 [cs, stat]*.
- Sudjianto, A.; Knauth, W.; Singh, R.; Yang, Z.; and Zhang, A. 2020. Unwrapping The Black Box of Deep ReLU Networks: Interpretability, Diagnostics, and Simplification. *arXiv:2011.04041 [cs, stat]*.
- Sudjianto, A.; and Zhang, A. 2021. Designing Inherently Interpretable Machine Learning Models. *arXiv:2111.01743*.
- The Office of the Comptroller of the Currency (OCC). 2021. Comptroller's Handbook: Model Risk Management. In *Comptroller's Handbook*, Safety and Soundness.
- Towns, J.; Cockerill, T.; Dahan, M.; Foster, I.; Gaither, K.; Grimshaw, A.; Hazlewood, V.; Lathrop, S.; Lifka, D.; Peterson, G. D.; Roskies, R.; Scott, J. R.; and Wilkins-Diehr, N. Sept.-Oct. 2014. XSEDE: Accelerating Scientific Discovery. *Computing in Science & Engineering*, 16(5): 62–74.
- van Erp, S.; Oberski, D. L.; and Mulder, J. 2019. Shrinkage Priors for Bayesian Penalized Regression. *Journal of Mathematical Psychology*, 89: 31–50.
- Zhang, M. R.; Lucas, J.; Hinton, G.; and Ba, J. 2019. Look-ahead Optimizer: k Steps Forward, 1 Step Back.
- Zhou, Y.; Booth, S.; Ribeiro, M. T.; and Shah, J. 2022. Do Feature Attribution Methods Correctly Attribute Features? *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9): 9623–9633.

Supplementary Methods

Medicare data preprocessing

Here we describe some details on the choices we made in preprocessing that will help make our work reproducible. Kyle Barron’s <https://kylebarron.dev/medicare-documentation/Medicare Documentation repository> of Medicare data documentation is an excellent resource for acquainting oneself with this standardized dataset. Our first steps in processing the CMS LDS were to merge the files, originally organized by year, into long tables for each claim type. In the process, we renamed pre-2011 columns in the dataset to match 2011+ plus columns where-ever they differed. We will refer to the dataset using 2011 and beyond column names.

Episode Grouping The CMS LDS consists of records organized into claims. Multiple claims can constitute a single period or episode of service. We determined episodes of the following types:

1. inpatient (inp)
2. skilled nursing facility (snf)
3. hospice (hosp)
4. outpatient (out, car)

For determining episodes, we grouped claims of each of the given types by person, and sorted by either the admission date (for inp, snf, hosp), or the claim through-date for (out, car).

Then for inp, snf, hosp, we merged successive claims into running episodes if they overlapped temporally, if the provider was the same and the intermediate discharge code indicates that the individual was not otherwise discharged home in between (we allow for distinct episodes with zero days of wait if a patient is discharged home and returns on the same day).

For out and car, we did the same merging with all claim types together, relaxing the need for the provider to match in an episode. Then we filtered for out/car episodes that did not overlap with inp, snf, hosp episodes – we determined these to be true outpatient episodes.

Then, for out and inp episodes, we determined if they corresponded to emergency department visits by looking for corresponding revenue center codes.

Model Specification

Parameter	Decomposition	Max order
α	MDC \times Hx \times CC/MCC	2
β	race	1
γ	MDC \times Hx \times CC/MCC	2
ν	MDC \times Hx \times CC/MCC	2

Table 2: **Specific decompositions used per parameter to define cohorts**, where major diagnostic category (MDC) is of size 26, history (Hx) is of size 2^5 , corresponding to low/high in each of the five dimensions, CC/MCC is of size 3, and race is of size 5.

The specific decompositions that we used for each of the model terms are displayed in Fig. 2. For the missing parameter ξ , the results in this manuscript are all determined using $\xi = 0$.

The python package `bayesianquilts`, with demonstration available at github.com/mederrata/bayesianquilts contains utilities for managing decompositions such as these.

We used a regularized horseshoe prior in order to encourage β to be sparse. Specifically, we applied an independent horseshoe prior to this parameter within every model cohort.

The individual components of each of the parameter decompositions were all modeled using Gaussian weakly-informative priors (with a default scale of 5 for the zero-order terms in the expansion). We helped encourage shrinkage by having the scale of these priors decay for higher order terms in the decomposition. For the results in the paper, we used a decay factor of 0.1 per each order.

Training We utilize TFP’s ADVI routines, which utilize stochastic sampling in computation of the ELBO. For this reason, it is not uncommon for specific parameter combinations to be in highly improbable locations – which can trigger underflows. To avoid instabilities, re adjust the likelihood on a per-observation level, first computing the minimum finite value of the log likelihood and then setting any divergent values to the minimum finite value minus a fixed offset of 100. We use the soft-plus function as a default bijector for any parameters that are supposed to be non-negative.

SHAP KernelSHAP, the general all-purpose model-agnostic implementation of SHAP, is very resource intensive, so we had to tune it in order to run it. First, we used the regression-based version of KernelSHAP, found in the python package `shapreg` [Covert and Lee 2021], which is not as resource intensive.

Second, although we had a very powerful computational resource at our disposal (Pittsburgh Supercomputer Center Bridges 2-AI with 512GB RAM), we had to restrict the input data size to 5k random training examples. Otherwise, we found that the system would run out of memory, causing the application to segmentation fault.

Finally, in order to get around an error involving a singular matrix, we regularized the linear algebra problem embedded within the algorithm, adding a fixed small constant of 10^{-8} to the diagonal of the linear transformation matrix (see <https://github.com/iancovert/shapley-regression/compare/master..joshchang:shapley-regression:mederrata>) for the exact modification made.

We were able to run `shapreg` with `n_samples=2400` and `batch_size=24` on our resource in approximately 5 hours. Due to the memory requirement issues with our large dataset, we are not able to scale this result to more data. SHAP is known to be computationally expensive, particularly for large datasets and a large number of features (see github issues <https://github.com/slundberg/shap/issues/10531053> <https://github.com/slundberg/shap/issues/14951495>), and its very computation is inherently based on approximations.

We believe that our computation of SHAP for our model is a reasonable representation of how well-approximated it is in practice, on a real problem and on a real dataset with a large number of predictors.

Our main point in the main text is a reiteration of the well-known fact that SHAP feature importance is not guaranteed to match what a model is doing in practice when the features in the training data are correlated – as would be true in most real-world problems. Examining Fig. 7 in the context of Fig. S6, one sees that the most-important SHAP features are not themselves very important in the model.

Supplementary Results

Here are results omitted from the main text for space constraints.

History representation

We utilized sparse probabilistic matrix factorization in order to obtain a low-dimension representation of personal medical history for the year prior to each episode. The encodings given by the model (Fig. S1 is an expanded version of Fig. 3 from the main text) specify linear combinations of the original data features that define a representation of an episode's history. The representations then can be constituted into a predictive distribution for the original features by transformation against a decoding matrix (Fig. S2). Note that this method finds a subset of the input features that can be used to predict the value of all features.

Random slopes Although we do not use this terminology in the main text, in the language of hierarchical mixed effects models the parameters β , ξ in the model are random slopes. In the main text we presented the week 1 slopes in Fig. 5. In Fig. S3, we present the components of β of the largest magnitudes, across all time intervals. As we noted in the main text, length of stay being at least 1 day, or conversely, being less than a full day, was the most impact predictor of early readmission. However, the effect disappears after one week. Long length of stay (greater than 30 days) appeared to follow the same trend, with those having a length of stay of at least a month having a lower readmission risk in the first week after discharge, but not reduced risk after the first week. Generally, the magnitude of the slopes tended to increase over time, with a few exceptions.

Random intercepts The parameter α from Eq. 2 is specific to each cohort in the model – it is a random intercept in hierarchical mixed effects modeling terminology. We presented the posterior mean for this parameter in Fig. 4, interpreting this quantity as a cohort-specific baseline survival.

Causal inference In our model we adjust for treatment selection bias by incorporating estimates of the treatment probabilities as covariates. The ordinal logistic regression intercepts are provided in Fig. S4.

The first five components in the parameter γ adjust for the selection bias present in claims. We present our cohort-specific estimates of γ in Fig. S5. We present the full time-course of discharge placement effects in Fig. S6. Largely, the discharge placement affects appear to strengthen from week

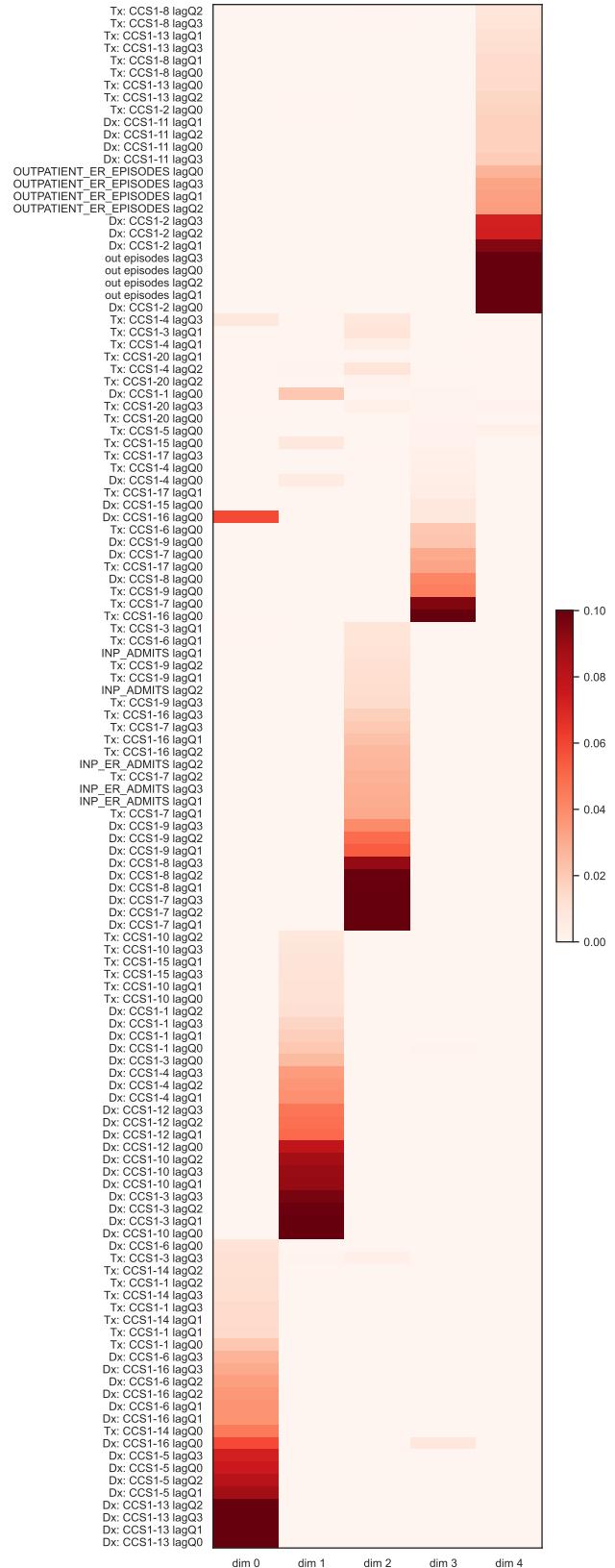


Figure S1: Extended version of Fig. 3 with up to 25 features per dimension.

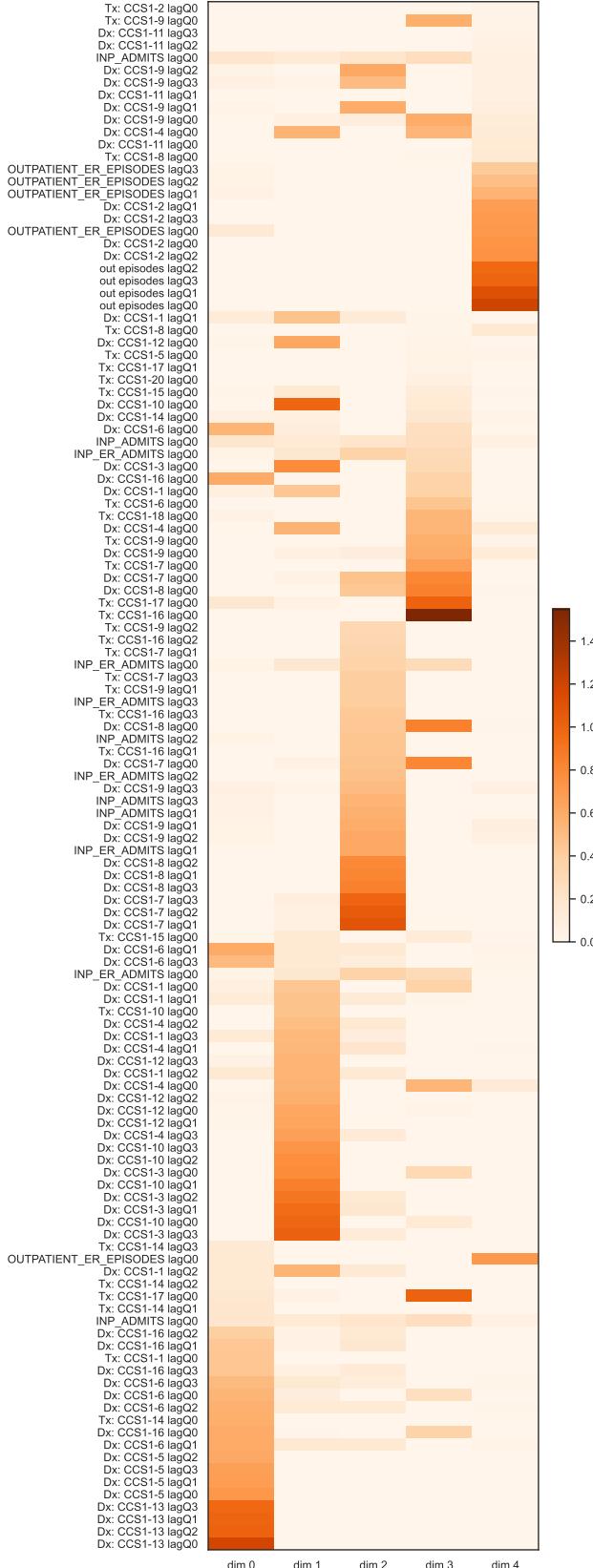


Figure S2: Decoding matrix corresponding to the encoding model of Fig. S1 showing up 25 features per dimension

1 to weeks 2/3 before weakening from week four onwards. The discharge placement bias effects have more cohort-level variability after the first week. In Fig. S7, we zoom in on the effects for the cohorts that benefit the most from the discharge placement interventions. A key advantage of this form of modeling against even the computationally interpretable ReLU-nets is the ability to perform mesoscopic cohort-level inference and interpretation. Cohort-level information facilitates making a model actionable since actions can be applied to subgroups all at once.

Understanding the correlations implied by the decomposition

Let

$$\theta^{(\kappa)} = \sum_{o=0}^O \theta_o^{(\kappa)}, \quad (5)$$

where $\mathbb{E}(\theta_0^{(\kappa)}) = \mu_o^{(\kappa)}$, $\forall o \leq O$. Suppose that for two multi-indices κ_1, κ_2 , that

- $\theta_o^{(\kappa_1)} = \theta_o^{(\kappa_2)}$, $\forall o \leq U < O$
- $\text{Cov}(\theta_j^{(\kappa_1)}, \theta_k^{(\kappa_2)}) = 0$ if $j \neq k$
- $\text{Var}(\theta_o^{(\kappa)}) = \sigma_o^2$

Then,

$$\begin{aligned} \text{Cov}(\theta^{(\kappa_1)}, \theta^{(\kappa_2)}) &= \sum_j \sum_k \text{Cov}(\theta_j^{(\kappa_1)}, \theta_k^{(\kappa_2)}) \\ &= \left(\sum_{j=0}^U \sum_{k=0}^U + \sum_{j=U+1}^O \sum_{k=0}^U + \sum_{j=0}^U \sum_{k=U+1}^O \right) \text{Cov}(\theta_j^{(\kappa_1)}, \theta_k^{(\kappa_2)}) \\ &= \sum_{o=0}^U \sigma_o^2 + \sum_{o=U+1}^O \text{Cov}(\theta_o^{(\kappa_1)}, \theta_o^{(\kappa_2)}). \end{aligned} \quad (6)$$

So,

$$\begin{aligned} \rho(\theta^{(\kappa_1)}, \theta^{(\kappa_2)}) &= \frac{\sum_{o=0}^U \sigma_o^2 + \sum_{o=U+1}^O \text{Cov}(\theta_o^{(\kappa_1)}, \theta_o^{(\kappa_2)})}{\sum_{o=0}^O \sigma_o^2} \\ &= \frac{\sum_{o=0}^U \sigma_o^2 + \sum_{o=U+1}^O \rho_o \sigma_o^2}{\sum_{o=0}^O \sigma_o^2}. \end{aligned} \quad (7)$$

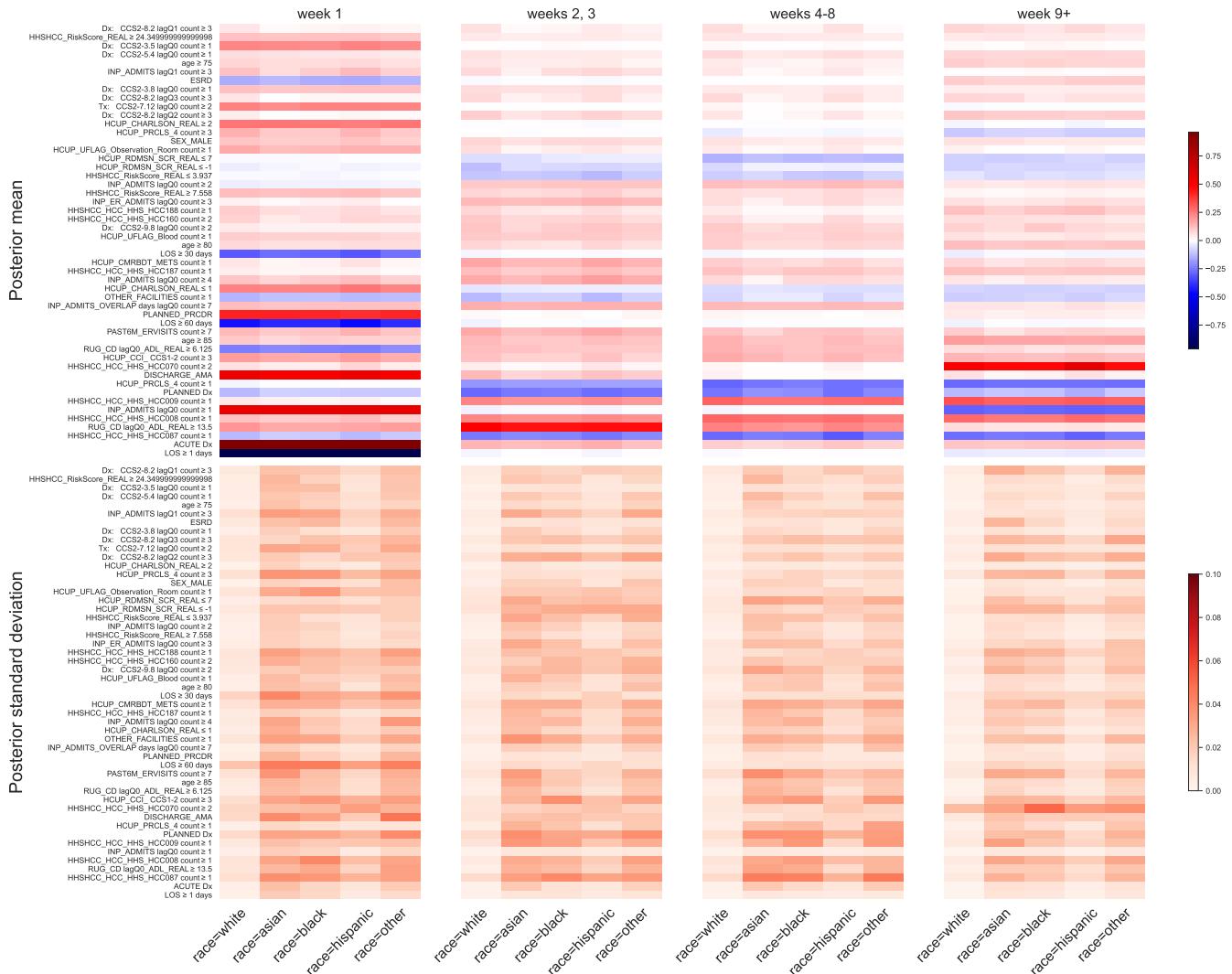


Figure S3: The 50 most influential regressors β (posterior mean, standard deviation) tracked through all time intervals. A more-comprehensive version of this figure can be found in our other supplemental file.

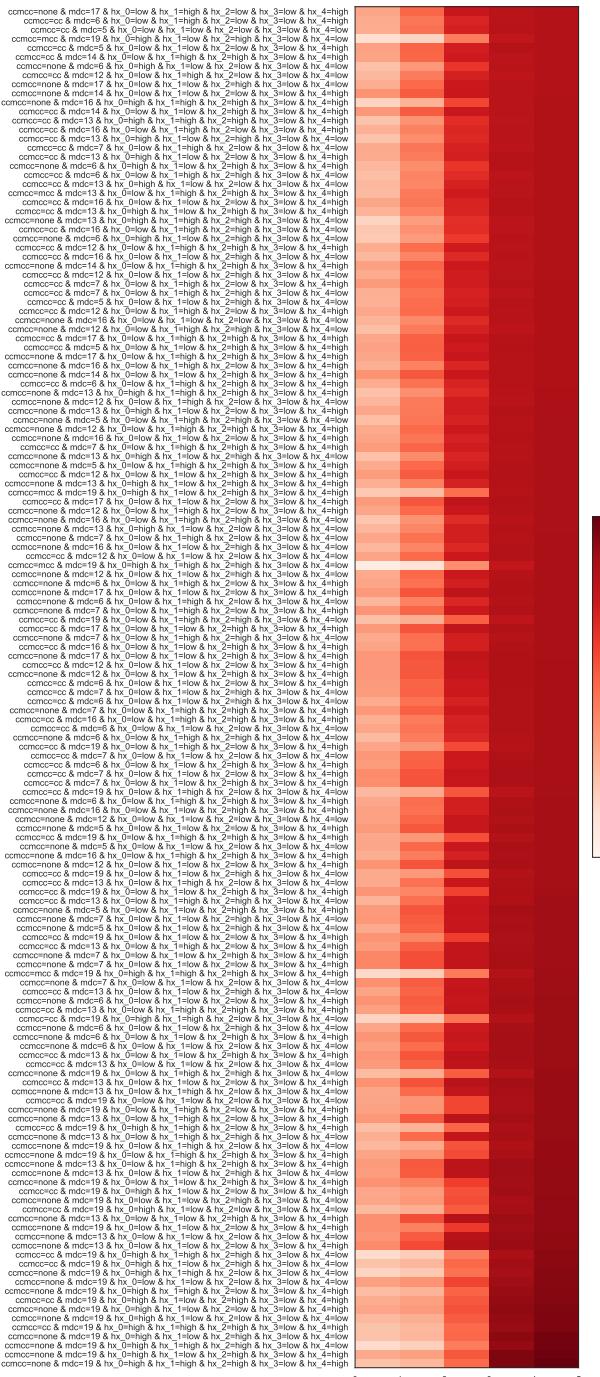


Figure S4: Cohort-wise ordinal intercept terms ν for the prediction of the distribution of discharge assignments.

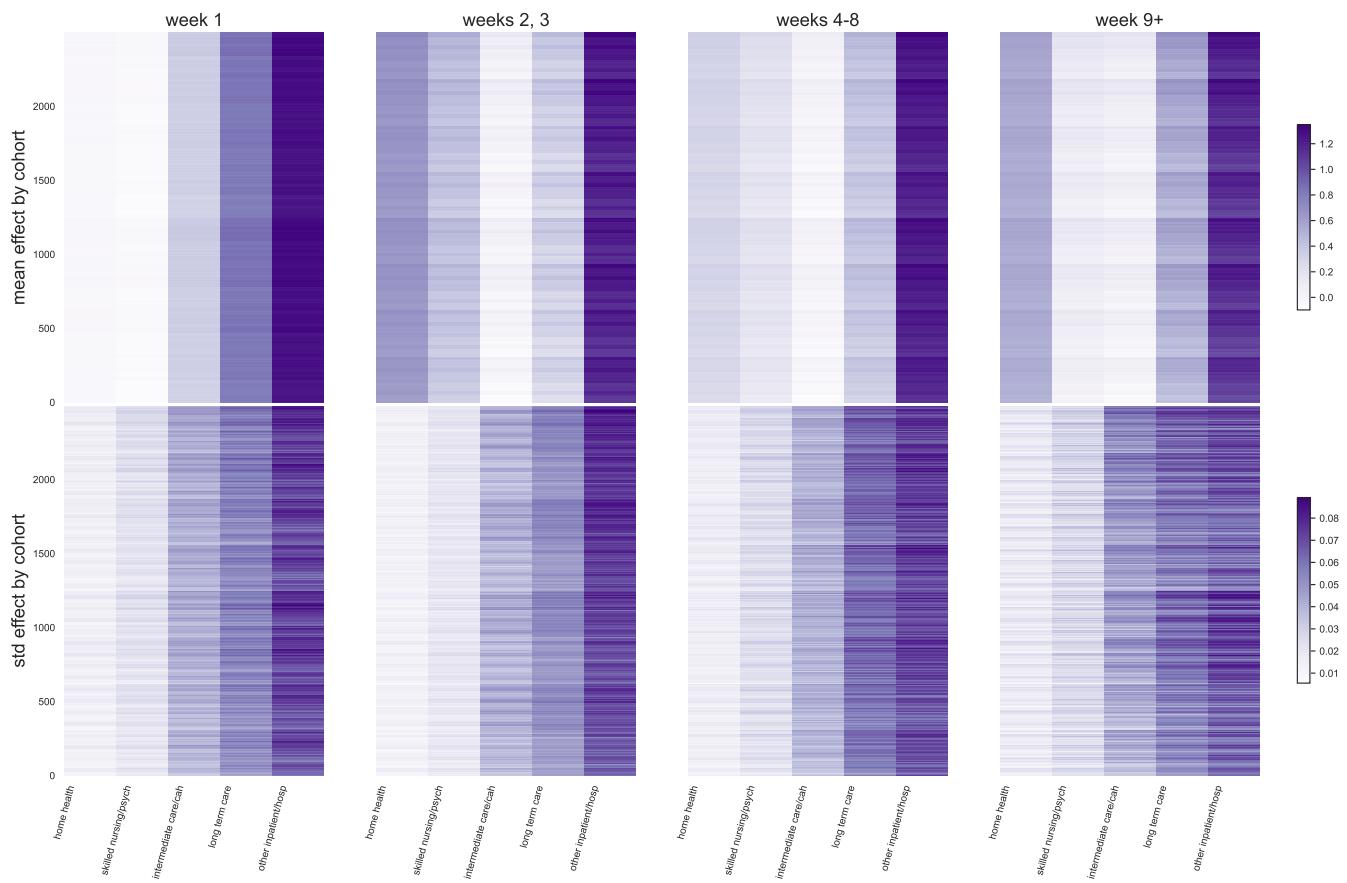


Figure S5: Discharge assignment adjustment terms corresponding to first five terms of γ (posterior mean, standard deviation) in terms of log hazard ratio under the log-additive effects model of Eq. 2

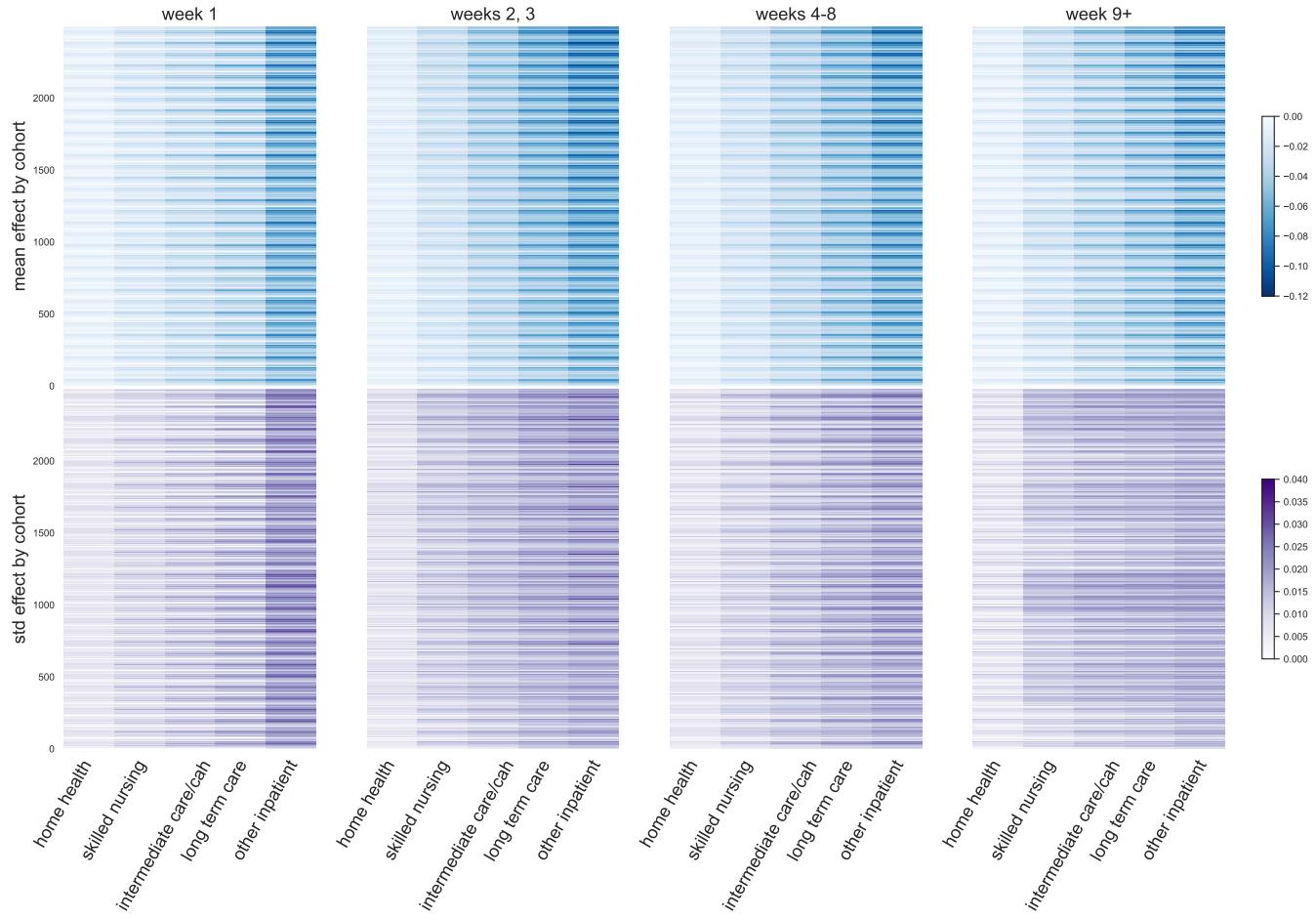


Figure S6: **Discharge placement effects γ (posterior mean, standard deviation)** in terms of log hazard ratio under the log-additive effects model of Eq. 2

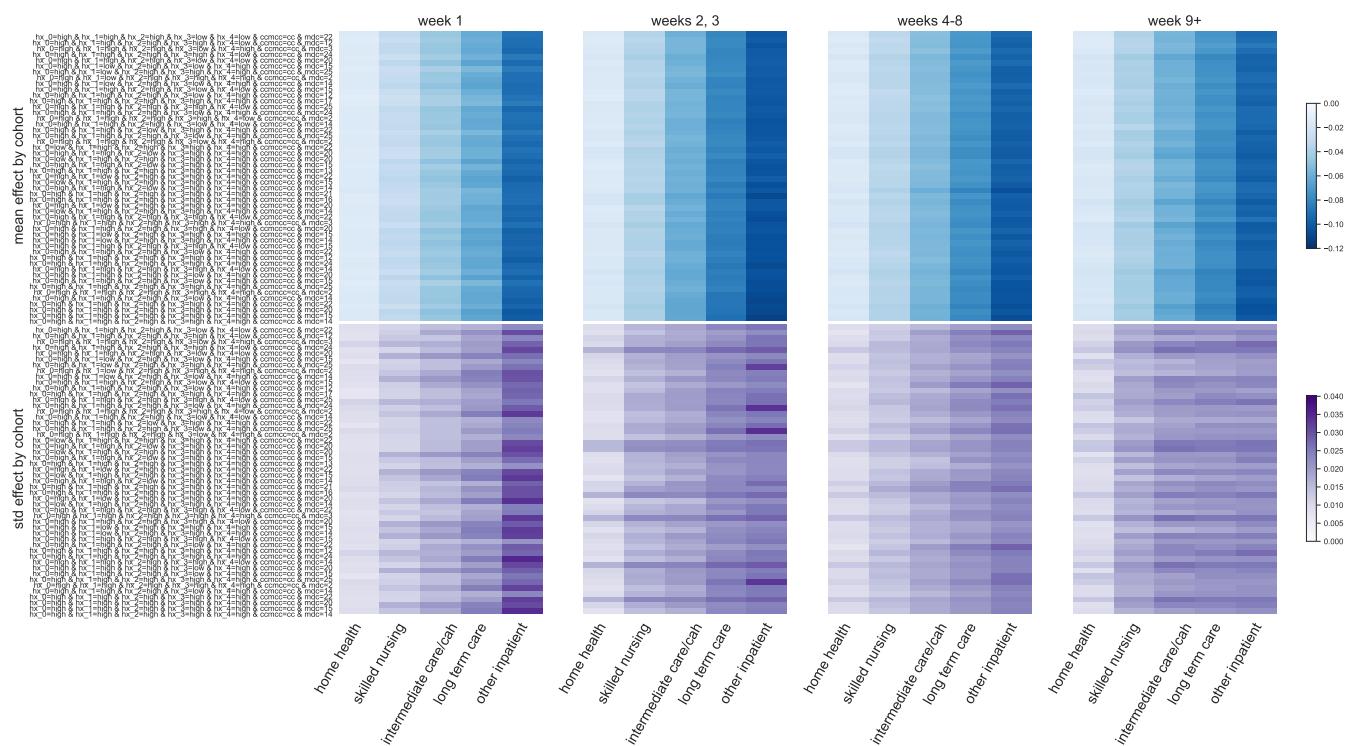


Figure S7: **Discharge placement effects for select cohorts** with the largest mean discharge placement effects.