# Sharing Pattern Submodels for Prediction with Missing Values

**Lena Stempfle, Ashkan Panahi, Fredrik D. Johansson**

Chalmers University of Technology
Department of Computer Science and Engineering, Gothenburg, Sweden
stempfle@chalmers.se, ashkan.panahi@chalmers.se, fredrik.johansson@chalmers.se

## Abstract

Missing values are unavoidable in many applications of machine learning and present challenges both during training and at test time. When variables are missing in recurring patterns, fitting separate pattern submodels have been proposed as a solution. However, fitting models independently does not make efficient use of all available data. Conversely, fitting a single shared model to the full data set relies on imputation which often leads to biased results when missingness depends on unobserved factors. We propose an alternative approach, called sharing pattern submodels, which i) makes predictions that are robust to missing values at test time, ii) maintains or improves the predictive power of pattern submodels, and iii) has a short description, enabling improved interpretability. Parameter sharing is enforced through sparsity-inducing regularization which we prove leads to consistent estimation. Finally, we give conditions for when a sharing model is optimal, even when both missingness and the target outcome depend on unobserved variables. Classification and regression experiments on synthetic and real-world data sets demonstrate that our models achieve a favorable tradeoff between pattern specialization and information sharing.

## 1 Introduction

Machine learning models are often used in settings where model inputs are partially missing either during training or at the time of prediction (Rubin 1976). If not handled appropriately, missing values can lead to increased bias or to models that are inapplicable in deployment without imputing the values of unobserved variables (Liu, Zachariah, and Stoica 2020; Le Morvan et al. 2020a). When missingness is dependent on unobserved factors that are related also to the prediction target, the fact that a variable is unmeasured can itself be predictive—so-called *informative missingness* (Rubin 1976; Marlin 2008). Often, imputation of missing values is insufficient, and it can be beneficial to let models make predictions based on both the partially observed data and on indicators for which variables are missing (Jones 1996; Groenwold et al. 2012). As mentioned in Le Morvan et al. (2020b), even the linear model—the simplest of all regression models—has not yet been thoroughly investigated with missing values and still reveals unexpected challenges.

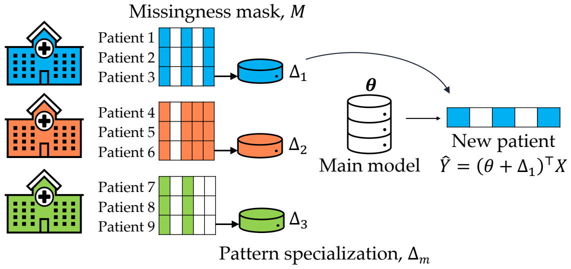

Figure 1: Coefficient sharing between a main model $\theta$ and pattern submodels for three clinics with different patterns in missing values. Without specialization, $\Delta_m$, an average prediction shared by clinics with different patterns may not lead to an optimal solution for any of them. Conversely, fitting separate models for each clinic does not use all of the available data efficiently and leads to high variance.

*Pattern missingness* emerges in data generating processes (DGPs) where there are structural reasons for which variables are measured—samples are grouped by recurring patterns of measured and missing variables (Little 1993). In Figure 1, we illustrate an example of this when observing patients from three different clinics, each systematically collecting slightly different measurements. Assume for simplicity that the pattern of missing values is unique to each clinic. In this way, a pattern-specific model is also site-specific.

*Pattern submodels* have been proposed for this setting, fitting a separate model to samples from each pattern (Mercaldo and Blume 2020; Marshall et al. 2002). This solution does not rely on imputation and can improve interpretability over black-box methods (Rudin 2019), but can suffer from high variance, especially when the number of distinct patterns is large and the number of samples for a given pattern is small. Moreover, if the fitted models differ significantly between patterns, it may be hard to compare or sanity-check their predictions. Notably, pattern submodels disregard the fact that the prediction task is shared between each pattern. However, in the context of Figure 1, using a shared model for all clinics may also be suboptimal if clinics take different measurements, or treat patients differently (high bias).

We propose the *sharing pattern submodel* (`SPSM`) in which submodels for different missingness patterns share coefficients while allowing limited specialization. This encourages efficient use of information across submodels leading to a beneficial tradeoff between predictive power and variance in the case where similar submodels are desired and sample sizes per pattern are small. Additionally, models with few and small differences between patterns are easier for domain experts to interpret.

We describe `SPSM` in Section 3, and we prove that in linear-Gaussian systems, a model which shares coefficients between patterns may be optimal—even when the prediction target depends on missing variables and on the missingness pattern (Section 4). Finally, we find in an experimental evaluation on real-world and synthetic data that `SPSM` compares favorably to baseline classifiers and regression methods, paying particular attention to how `SPSM` boosts sample efficiency and model sparsity (Section 5).

## 2 Prediction with Test-Time Missingness

Let $X = [X_1, ..., X_d]^\top$ be a vector of $d$ random variables taking values in $\mathcal{X} \subseteq \mathbb{R}^d$, and $M = [M_1, ..., M_d]^\top$ be a random missingness mask in $\mathcal{M} \subseteq \{0,1\}^d$ where $M_j = 1$ indicates that variable $X_j$ is missing. Next, let $\tilde{X} \in (\mathbb{R} \cup \{\texttt{NA}\})^d$ be the mixed observed-and-missing values of $X$ according to $M$ and define $X_{\neg M} = [X_j : M_j = 0]^\top \in \mathbb{R}^{d-\|M\|_1}$ to be the vector of *observed* covariates under $M$. The outcome of interest, $Y \in \mathbb{R}$, may depend on all of $X$, observed or missing, as well as on $M$. Let $k = |\mathcal{M}|$ denote the number of possible missingness patterns.[1] Further, assume that variables $X, M, Y$ are distributed according to a *fixed, unknown* joint distribution $p$. The assumed (causal) dependencies of the variables used, coincide most closely with *selection missingness* (Little 1993) (Figure 4 in the appendix).

Our goal is to predict $Y$ *under missingness* $M$ in $X$ using functions $f : (\mathbb{R} \cup \{\texttt{NA}\})^d \to \mathbb{R}$. We aim to minimize risk with respect to the squared loss on $p$,

$$\min_f R(f), \text{ where } R(f) \coloneqq \mathbb{E}_{\tilde{X},Y\sim p}[(f(\tilde{X}) - Y)^2]\,. \quad (1)$$

Under the assumption that $Y$ has centered, additive noise,

$$Y = g(X, M) + \epsilon \text{ where } \mathbb{E}[\epsilon] = 0, \quad (2)$$

the Bayes-optimal predictor of $Y$ is $f^* = \mathbb{E}[Y \mid X_{\neg M}, M]$. In general, observed values $X_{\neg M}$ are insufficient for predicting $Y$; $f^*$ may depend directly on the mask $M$, *even if* $Y$ *does not depend directly on* $M$ (Le Morvan et al. 2021).

A common strategy to learn $f$ is to first impute the missing values in $\tilde{X}$ and then fit a model on the observed-or-imputed covariates $X^I \in \mathbb{R}$—so-called *impute-then-regress* estimation. Even though imputation is powerful, it is not always optimal under test-time missingness (Le Morvan et al. 2021) and often assumes that data is missing at random (MAR) (Carpenter and Kenward 2012; Seaman et al. 2013).

### 2.1 Pattern Submodels

In cases where the number of distinct missingness patterns $k$ is small, it is possible to learn separate predictors $f_m$ for each pattern. This idea has been called *pattern submodels* (PSM) (Mercaldo and Blume 2020; Marshall et al. 2002), a set of models which aim to minimize the empirical risk under each missingness pattern. Let $D = \{(\tilde{x}^{(1)}, m^{(1)}, y^{(1)}), ..., (\tilde{x}^{(n)}, m^{(n)}, y^{(n)})\}$ be a data set of $n$ samples, with partially observed features $\tilde{x}^{(i)}$, corresponding to missingness patterns $m^{(i)}$, drawn independently and identically distributed from $p$. PSM may be learned by minimizing the regularized empirical risk,

$$\min_{\{f_m\}\in\mathcal{F}^k} \; \frac{1}{n}\sum_{i=1}^{n} L(f_{m^{(i)}}(\tilde{x}^{(i)}), y^{(i)}) + \sum_{m\in\mathcal{M}} \mathcal{R}(f_m) \quad (3)$$

over a suitable class of models $\mathcal{F}$ and regularization $\mathcal{R}$. Mercaldo and Blume (2020) considered linear and logistic regression models, $f_m = \sigma(\theta_m^\top x)$ with $\sigma$ either the identity or logistic function and loss $L$ chosen to match. The objective in (3) is separable in $m$ and can be solved independently for each pattern. However, this often leads to high variance in the small-sample regime since each pattern accounts for only a subset of the available samples. Without structural assumptions, the number of patterns $k$ grows exponentially with $d$ (see discussion in Section 6).

PSM allows for prediction under test-time missingness which adapts to the pattern $m$ without relying on imputation or assumptions on missingness mechanisms like MAR. However, the prediction target (and the Bayes-optimal model $f^*$) may have only a small dependence on the pattern $m$; *the optimal submodels for all* $m$ *may share significant structure*. Next, we propose estimators that exploit such structures to reduce variance and increase interpretability.

## 3 Sharing Pattern Submodels

We propose *sharing pattern submodels* (`SPSM`), linear prediction models, specialized for patterns in variable missingness, which share information during learning. Sharing is accomplished by regularizing submodels towards a main model and solving the resulting coupled optimization problem. While linear models are limited in expressive power, they are often found to be useful approximations of nonlinear functions due to their superior interpretability.

**Fitting SPSM** Let $\theta \in \mathbb{R}^d$ represent *main model* coefficients used in prediction under all missingness patterns, and define $\theta_{\neg m} = [\theta_j : m_j = 0]^\top \in \mathbb{R}^{d_m}$ to be the subset of coefficients corresponding to variables observed under $m$. To emphasize, $\theta_{\neg m}$ depends only on $m$ in selecting a subset of $\theta$—the coefficients are shared across patterns. Similarly, define $\Delta_{\neg m} \in \mathbb{R}^{d_m}$ to be *pattern-specific specialization* of these coefficients to $m$. In contrast to $\theta_{\neg m}$, the values of $\Delta_{\neg m}$ are unique to each pattern $m$. Note, a model $f_m$ depends only on the observed components of $X$. In regression tasks, we learn **sharing** pattern submodels on the form

$$f_m(x) \coloneqq (\theta_{\neg m} + \Delta_{\neg m})^\top x_{\neg m}, \;\; \text{for all } \; m \in \mathcal{M} \quad (4)$$

[1] In practical scenarios, we expect $k$ to be much smaller than the worst-case number, $2^d$.

by solving the following problem with $\lambda_m \geq 0$ and $\gamma \geq 0$,

$$\underset{\theta, \{\Delta_{\neg m}\}}{\text{minimize}} \;\; \frac{1}{n}\sum_{i=1}^{n}\left((\theta_{\neg m^{(i)}} + \Delta_{\neg m^{(i)}})^\top x^{(i)}_{\neg m^{(i)}} - y^{(i)}\right)^2$$
$$+ \frac{\gamma}{n}\|\theta\| + \sum_{m \in \mathcal{M}} \frac{\lambda_m}{n_m}\|\Delta_{\neg m}\|_1 \; . \qquad (5)$$

where $n_m$ is the number of samples of pattern $m$. $\lambda_m > 0$ and $\gamma > 0$ are regularization parameters. Intercepts (pattern-specific and shared) are left out for brevity. The optimization problem is convex, and we find optimal values for $\theta$ and $\Delta_m$ using L-BFGS-B (Byrd et al. 1995) in experiments. In classification tasks, the square loss is replaced by the logistic loss. In either case, we call the solution to (5) SPSM.

For the penalty $\|\theta\|$, we use either the $\ell_1$ or $\ell_2$ norm to tradeoff bias and variance in the main model. A high value for $\lambda_m$ regularizes the specialization of model coefficients to missingness pattern $m$ such that high $\lambda_m$ encourages smaller $\|\Delta_m\|_1$ and greater coefficient sharing. In experiments, we let $\lambda_m$ take the same value $\lambda$ for all patterns. $\ell_1$-regularization is used for $\Delta$ as we aim for a sparse solution where the majority of specialization coefficients are zero.

**Consistency** For fixed $\lambda, \gamma$, sums of the minimizers of (5), $\theta^*_{\neg m} + \Delta^*_{\neg m}$, converge to the best linear approximations of the Bayes-optimal predictors $f^*_m$ for each pattern $m$ in the large-sample limit. We state this formally and sketch a proof in Appendix A.2 using standard arguments. This result is agnostic to parameter sharing; $\Delta^*$ may not be sparse. In Section 4, we prove that, in the linear-Gaussian setting, our method also recovers the sparsity of the true process. In the large-sample limit, this may not be beneficial for variance reduction, but sparsity contributes to interpretability.

**Why is SPSM Interpretable?** Comparing pattern specializations allows domain experts to reason about how similar submodels are, and how they are affected by missing values. We argue that a set of submodels is more interpretable if specializations contain fewer non-zero coefficients, $\Delta_{\neg m}$ is sparse. Sparsity is a generally useful measure of interpretablity (Rudin 2019), since it results in only a subset of the input features affecting predictions, reducing the effective complexity of the model (Miller 1956; Cowan 2010).

# 4 Optimality of Sharing Models

In this section, we give conditions under which an optimal pattern submodel has sparse specializations (shares parameters between patterns) and when SPSM converges to such a model in the large-sample limit. We analyze DGPs where the outcome $Y$ depends linearly on *all* components of $X$ (*models* have access only the observed subset of these) and on the pattern $M$, but not on interactions between $X$ and $M$,

$$Y = \theta^\top X + \alpha_M + \epsilon \;, \text{ with } \epsilon \sim \mathcal{N}(0, \sigma_Y^2). \qquad (6)$$

Here, $\alpha_M$ is a pattern-specific intercept. Without $\alpha_M$, this is a setting often targeted by imputation methods, since the outcome is a parametric function of the full $X$. However, we know that $X$ will be partially missing also at test time, and $M$ is allowed to have arbitrary dependence on $X$. In this case, imputation need not be necessary or sufficient.

Next, we study this setting with Gaussian $X$, where we can precisely characterize optimal models and their sparsity.

## 4.1 Sparsity in Linear-Gaussian DGPs

Recall that $X_{\neg m}$ and $\theta_{\neg m}$ denote covariates and coefficients restricted to *observed* variables under pattern $m$, and define $X_m$ and $\theta_m$ analogously for missing variables. For outcomes which obey (6), the Bayes-optimal model under $m$ is

$$\mathbb{E}[Y \mid X_{\neg m}, M = m] = {\theta_{\neg m}}^\top X_{\neg m} + \xi_m \qquad (7)$$

where $\xi_m = {\theta_m}^\top \mathbb{E}_{X_m}[X_m \mid X_{\neg m}] + \alpha_m$ is the bias of the naïve prediction made using the coefficients $\theta_{\neg m}$ of the true system but restricted to observed variables. Ignoring $\xi_m$ coincides with performing prediction following zero-imputation and is biased in general. $\xi_m$ thus captures the specialization required for pattern submodels to be unbiased. For closer analysis, we study the following setting.

**Condition 1** (Linear-Gaussian DGP)**.** *Covariates* $X = [X_1, ..., X_d]^\top$ *are Gaussian,* $X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ *with mean* $\boldsymbol{\mu}$ *and covariance matrix* $\Sigma$*. The outcome* $Y$ *is linear-Gaussian as in* (6) *with parameters* $(\theta, \{\alpha_m\}, \sigma_Y)$*.* $M$ *is arbitrary.*

In line with Condition 1, let $\Sigma_{\neg m, m}$ be the submatrix of $\Sigma$ restricted to the rows corresponding to *observed* variables under $m$ and columns corresponding to variables *missing* under $m$. Define $\Sigma_{\neg m, \neg m}$ and $\Sigma_{m, \neg m}$ analogously. Throughout, we assume that $\Sigma$ is invertible so that the distribution is non-degenerate. In practice, the non-degenerate case can be handled through ridge regularization.

**Proposition 1.** *Suppose covariates* $X$ *and outcome* $Y$ *obey Condition 1 (are linear-Gaussian). Then, the Bayes-optimal predictor for an arbitrary missingness mask* $m \in \mathcal{M}$*, is*

$$f^*_m = \mathbb{E}[Y \mid X_{\neg m}, m] = (\theta_{\neg m} + \Delta_{\neg m})^\top X_{\neg m} + C_m$$

*where* $C_m \in \mathbb{R}$ *is constant with respect to* $X_{\neg m}$ *and*

$$\Delta_{\neg m} = (\Sigma^{-1}_{\neg m, \neg m})\Sigma_{\neg m, m}\theta_m \; .$$

Proposition 1 states that, for a linear-Gaussian system, the Bayes-optimal model under missingness pattern $m$ has the same form as SPSM with pattern-specific intercept, combining coefficients of a main model $\theta$ and specializations $\Delta_{\neg m}$. The result is proven in Appendix A.3.

In nonlinear DGPs, the optimal correction term $\Delta_{\neg m}$ may not be constant with respect to $X_{\neg m}$. The NeuMiss model by Le Morvan et al. (2020a) learns such corrections as functions of the input and missingness mask using deep neural networks. However, this method lacks the interpretability of sparse linear models sought here. Even in this more general case, SPSM may achieve a good bias-variance tradeoff. Indeed, we find on real-world data, which may not be linear, that SPSM is often preferable to strong nonlinear baselines.

**When is Sparsity Optimal?** Like other sparsity-inducing regularized estimators, such as LASSO (Tibshirani 1996), SPSM reduces variance by shrinking some model parameters to zero. Under appropriate conditions, when the training set grows large, we expect the learned sparsity to correspond to properties inherent to the DGP. For LASSO, this means recovering zeros in the coefficient vector of the outcome. For SPSM, objective (5) is used to learn submodels on the form $(\theta_{\neg m} + \Delta_{\neg m})^\top X_{\neg m}$ where $\theta$ is shared between patterns and $\Delta_{\neg m}$ is sparse. It is natural to ask: When can we expect the "true" or an "optimal" $\Delta_{\neg m}$ to be sparse and, if it is, when can we recover this sparsity with SPSM? Surprisingly, as we will see, the optimal specialization $\Delta_{\neg m}$ may be sparse even if $Y$ depends on *all* covariates in $X$.

Assume that Condition 1 (Linear-Gaussian DGP) holds with system parameters $(\mu, \Sigma, \theta, \{\alpha_m\}, \sigma_Y)$. We can characterize sparsity in the Bayes-optimal model $(\theta, \{\Delta_{\neg m}\})$, see Proposition 1, by the interactivity of covariates. We say that variables $X_j$ and $X_{j'}$ are non-interactive if they are statistically independent given all other covariates. As is well-known, for Gaussian $X$, $X_j$ and $X_{j'}$ are non-interactive if $S_{j,j'} = 0$, where $S = \Sigma^{-1}$ is the precision matrix.

**Proposition 2** (Sparsity in optimal model)**.** *Suppose that a covariate $j \in [d]$ is observed under pattern $m$, i.e., $m_j = 0$, and assume that $X_j$ is non-interactive with every covariate $X_{j'}$ that is missing under $m$. Then $(\Delta_{\neg m})_j = 0$.*

Proposition 2 states that the sparsity in $\Delta$ is partially determined by the covariance pattern of observed and unobserved covariates. For example, specialization is *not needed* for a variable $j$ under pattern $m$ if it is uncorrelated with all missing variables under $m$. Conversely, specialization, i.e., $(\Delta_{\neg m})_j \neq 0$, *is needed* for features $j$ that are predictive ($\theta_j \neq 0$) and redundant (replicated well by unobserved features which are also predictive). This is because in the main model, redundant variables may share the predictive burden, but when they are partitioned by missingness, they have to carry it alone. This shows that prediction with a single model and zero-imputation is sub-optimal in general.

**Consistency of SPSM** In the large-sample limit, under Condition 1, we can prove that SPSM recovers maximally sparse optimal model parameters. If the true system parameters are also sparse, SPSM learns these.

**Theorem 1.** *Suppose that Condition 1 holds with parameters $(\theta, \{\Delta_{\neg m}\})$ as in Proposition 1, such that, for each covariate $j$, the number of patterns $m$ for which $m_j = 0$ and $(\Delta_{\neg m})_j = 0$ is strictly larger than the number of patterns $m'$ for which $m'_j = 0$ and $(\Delta_{\neg m'})_j \neq 0$. Then, with $\gamma = 0$ and fixed $\lambda > 0$, the true parameters $(\theta, \{\Delta_{\neg m}\})$ are the unique solution to* (5) *in the large-sample limit, $n \to \infty$.*

*Proof sketch.* We provide a full proof in Appendix A.5. The main steps involve showing that the SPSM objective (5) is asymptotically dominated by the risk term, and the *sums* of its minimizers ($\theta^*_{\neg m}$ + $\Delta^*_{\neg m}$) coincide with optimal regression coefficients ($\hat{\theta}_{\neg m}$) fit independently for each missingness pattern $m$. For any $\lambda > 0$, regularization steers the solution towards one which is maximally sparse in $\Delta^*_{\neg m}$. □

## 4.2 Relationship to Other Methods

|  | $\gamma < \infty$ | $\gamma \to \infty$ |
|---|---|---|
| $\lambda_m \to \infty$ | Zero imputation | Constant |
| $0 < \lambda_m < \infty$ | Sharing model | Pattern submodel |
| $\lambda_m = 0$ | No sharing | Pattern submodel |

Table 1: Extreme cases and equivalences of SPSM, provided that no pattern-specific intercept is used.

For particular extreme values of the regularization parameters $\gamma, \lambda_m$, SPSM coincides with other methods (Table 1). First, the full-sharing model ($\lambda_m \to \infty, \gamma < \infty$) coincides with fitting a single model to all samples after zero-imputation. To see this, set $\Delta_{\neg m} = 0$ for all $m$ and note

$$\theta_{\neg m^{(i)}}{}^\top x^{(i)}_{\neg m^{(i)}} = \theta^\top I_0(\tilde{x}^{(i)})$$

where $I_0(\tilde{x})$ replaces missing values in $\tilde{x}$ with 0. In this setting, submodel coefficients cannot adapt to $m$. In the implementation, we allow the fitting of pattern-specific intercepts which are not regularized by $\lambda_m$. Second, ($\lambda_m < \infty, \gamma \to \infty$) corresponds with the standard PSM without parameter sharing (Mercaldo and Blume 2020) or the ExpandedLR method of (Le Morvan et al. 2020b). The precise nature of this equivalence depends on the choice of regularization.[2] In this setting, each submodel $\hat{f}_m$ is fit completely independently of every other. Finally, an SPSM model with optimal parameters $(\theta, \{\Delta_{\neg m}\})$, in the linear-Gaussian case, implicitly makes a perfect single linear imputation,

$$\mathbb{E}[X_m \mid X_{\neg m}] = X_{\neg m}\Sigma^{-1}_{\neg m,\neg m}\Sigma_{\neg m,m},$$

and applies the main model's parameters $\theta_m$ to the imputed values. If many samples are available, it may be feasible to learn the imputation directly. However, if the variables in $X_{\neg m}$ and $X_m$ are never observed together, imputation is no longer possible. In contrast, SPSM could still learn an optimal submodel for each pattern, given enough samples.

# 5 Experiments

We evaluate the proposed SPSM model[3] on simulated and on real-world data, aiming to answer two main questions: How does the accuracy of SPSM compare to baseline models, including impute-then-regress, for small and larger samples; How does sparsity in pattern specializations $\Delta$ affect performance and interpretation?

**Experimental Setup** In the SPSM algorithm, before one-hot-encoding of categorical features, all missingness patterns in the training set are identified. At test time, patterns that did not occur during training, variables are removed until the closest training pattern is recovered. Both linear and logistic variants of SPSM were trained using the L-BFGS-B solver provided as part of the SciPy Python package (Virtanen et al. 2020). Our implementation supports both $\ell_1$ and $\ell_2$-regularization of the main model parameters $\theta$ and $\ell_1$-regularization of pattern-specific deviations $\Delta$. This includes both the no-sharing pattern submodel ($\lambda_m < \infty, \gamma \to \infty$) and full-sharing model ($\lambda_m \to$

[2] Mercaldo and Blume (2020) adopted a two-stage estimation procedure, the relaxed LASSO (Meinshausen 2007).

[3] Code to reproduce experiments and the appendix are available at https://github.com/Healthy-AI/spsm.

$\infty, \gamma < \infty$) as special cases. In the experiments, $\gamma$ can take values within $[0, 0.1, 1, 5, 10, 100]$, and we used a shared $\lambda_m = \lambda \in [1, 5, 10, 100, 1000, 1e^8]$ for all patterns. Intercepts were added for both the main model and for each pattern without regularization. We do not require patterns to have a minimum sample size but support this functionality (appendix Table 8). For missingness patterns at test time that did not occur in the training data, variables were removed until the closest training pattern was recovered.

We compare linear and logistic regression models to the following baseline methods: Imputation + Ridge / logistic regression (Ridge/LR), Imputation + Multilayer perceptron (MLP) with a single hidden layer, and XGBoost (XGB), where missing values are supported by default (Chen et al. 2019). Last, we compare the Pattern Submodel (PSM) (Mercaldo and Blume 2020). Note, our implementation of PSM is based on a special case of our SPSM implementation where regularization is applied over all patterns and not in each pattern separately. Hyperparameters are based on the validation set. For imputation, we use zero ($I_0$), mean ($I_\mu$) or iterative imputation ($I_{it}$) from SciKit-Learn (Pedregosa et al. 2011b; Van Buuren 2018). XGB's handling of missing values is denoted $I_n$. Details about method implementations, hyperparameters and evaluation metrics are given in Appendix B.2.

## 5.1 Simulated Data

We use simulated data to illustrate the behavior of sharing pattern submodels and baselines in relation to Proposition 1, focusing on bias and variance. We sample $d$ input features $X$ from a multivariate Gaussian $\mathcal{N}(0, \Sigma)$ with covariance matrix $\Sigma$ specified by a cluster structure; the features are partitioned into $k$ clusters of equal size. The covariance is defined as $\Sigma_{ii} = 1$, $\Sigma_{i \neq j} = 0$ if $i, j$ are in different clusters, and $\Sigma_{i \neq j} = c$ if $i, j$ are in the same cluster, where $c$ is chosen as large as possible so that $\Sigma$ remains positive semidefinite.

Each cluster $c \in \{1, ..., k\}$ is represented in the outcome function $Y = \theta^\top X + \epsilon$ by a single feature $i(c)$, such that $\beta_{i(c)} \sim \mathcal{N}(0, 1)$ and $\theta_j = 0$ for other features. We let $\epsilon \sim \mathcal{N}(0, 1)$, independently for each sample. We consider three missingness settings: In Setting A, each variable in cluster $c$ is missing if $X_{i(c)} > -0.5$. In Setting B, each variable in cluster $c$—except one chosen uniformly at random—is missing if $X_{i(c)} > -0.5$. Both settings satisfy the conditions of Proposition 1 but are designed to violate MAR by letting the outcome variable depend directly on missing values which may not be recovered from observed ones. In Setting C, we follow missing-completely-at-random (MCAR), where variables are missing independently with probability 0.2. We generate samples with $d = 20$ and $k = 5$.

In Figure 2, we show the test set coefficient of determination ($R^2$) for Setting A. Note, that the methods which use imputation (imputation method selected based on validation error at each data set size) perform well initially but plateau quickly, indicating relatively high bias. SPSM and PSM both achieve a higher $R^2$ for the full sample. SPSM performs better than PSM for small samples indicating lower variance. The SPSM model includes 42 non-zero pattern-specific coefficients when the training set size is 0.2 and 68 with the fraction is 0.8. Results for Setting B and C are presented in Appendix C.1. Even in the MCAR setting C, PSM performs considerably worse than alternatives due to excessive variance from fitting independent pattern-specific models.

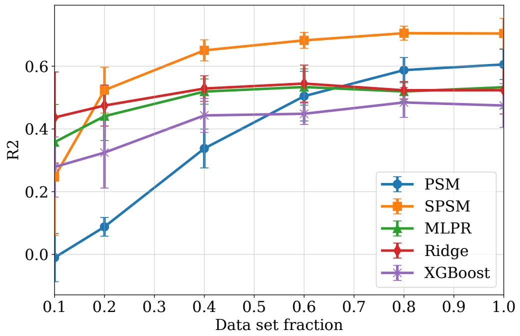

Figure 2: Performance on simulated data Setting A (higher is better). Error bars show standard deviation over 5 random data splits. The full data set has $n = 2000$ samples.

## 5.2 Real-World Tasks

We describe two health care data sets used for classification and regression. More information on the non-health related HOUSING (De Cock 2011) data is shown in Appendix C.3.

**ADNI** The data is obtained from the publicly available Alzheimer's Disease Neuroimaging Initiative (ADNI) database.[4] ADNI collects clinical data, neuroimaging and genetic data (Weiner et al. 2010). In the classification task, we predict if a patient's diagnosis will change 2 years after baseline diagnosis. The regression task aims to predict the outcome of the ADAS13 (Alzheimer's Disease Assessment Scale) (Mofrad et al. 2021) cognitive test at a 2-year follow-up based on available data at baseline.

**SUPPORT** We use data from the Study to Understand Prognoses and Preferences for Outcomes and Risks of Treatments (SUPPORT) (Knaus et al. 1995), which aims to model survival over a 180-day period in seriously ill hospitalized adults using the Physiology Score (SPS). Following Mercaldo and Blume (2020), in the regression task we predict the SPS while for the classification task, we predict if a patient's SPS is above the median; the label rate is 50/50 by definition. We mimic their MNAR setting by adding 25 units to the SPS values of subjects missing the covariate "partial pressure of oxygen in the arterial blood".

## 5.3 Results

We report the results on health care data in Table 2. For regression tasks, we provide the number of non-zero coefficients used by the linear models. In addition, we study prediction performance as a function of data set size in Figure 3 and in the appendix Figure 7. The statistical uncertainty of the average error is measured with its square root, which is a standard deviation and expressed by 95% confidence intervals over the test set. Results of HOUSING data are presented in Appendix C.3.

[4] http://adni.loni.usc.edu

| **Regression** | $R^2$ | **# Coefficients** |
|---|---|---|
| ADNI | | |
| `Ridge`, $I_\mu$ | 0.66 (0.59, 0.73) | 37 + 0 |
| `XGB`, $I_\mu$ | 0.41 (0.31, 0.50) | — |
| `MLP`, $I_0$ | 0.62 (0.55, 0.69) | — |
| `PSM` | 0.51 (0.43, 0.60) | 0 + 430 |
| `SPSM` | 0.66 (0.59, 0.73) | 37 + 21 |
| SUPPORT | | |
| `Ridge`, $I_0$ | 0.38 (0.35, 0.42) | 11 + 0 |
| `XGB`, $I_n$ | 0.30 (0.27, 0.34) | — |
| `MLP`, $I_\mu$ | 0.56 (0.53, 0.59) | — |
| `PSM` | 0.52 (0.49, 0.56) | 0 + 188 |
| `SPSM` | 0.53 (0.50, 0.56) | 11 + 91 |
| **Classification** | **AUC** | **Accuracy** |
| ADNI | | |
| `LR`, $I_0$ | 0.85 (0.80, 0.90) | 0.85 (0.74, 0.94) |
| `XGB`, $I_n$ | 0.80 (0.74, 0.86) | 0.84 (0.73, 0.94) |
| `MLP`, $I_0$ | 0.86 (0.78, 0.89) | 0.84 (0.73, 0.94) |
| `PSM` | 0.81 (0.75, 0.87) | 0.84 (0.74, 0.95) |
| `SPSM` | 0.86 (0.81, 0.90) | 0.85 (0.75, 0.96) |
| SUPPORT | | |
| `LR`, $I_0$ | 0.83 (0.81, 0.85) | 0.77 (0.74, 0.79) |
| `XGB`, $I_0$ | 0.85 (0.83, 0.87) | 0.78 (0.75, 0.81) |
| `MLP`, $I_0$ | 0.86 (0.85, 0.88) | 0.79 (0.76, 0.81) |
| `PSM` | 0.84 (0.83, 0.86) | 0.78 (0.75, 0.81) |
| `SPSM` | 0.85 (0.83, 0.86) | 0.78 (0.75, 0.80) |

Table 2: Results for ADNI and SUPPORT tasks along with the respective imputation method (see setup). We also report the number of non-zero coefficients in shared ($k$) and pattern-specific models ($l$) as $k + l$.

For ADNI regression, `SPSM` and `Ridge` are the best performing models with $R^2$ of 0.66 showing the same confidence in the prediction. Validation performance resulted in selecting $\gamma = 10.0, \lambda = 50$ for `SPSM`. With an $R^2$ score of 0.51, `PSM` seems not able to benefit from pattern-specificity in ADNI. In contrast, `SPSM` makes use of coefficient sharing which results in a significantly smaller number of coefficients compared to `PSM`. For SUPPORT regression, `PSM` achieves almost the same result as `SPSM` ($R^2$ of 0.52–0.53) with partly overlapping confidence intervals for the predictions. Although, the number of coefficients used in `SPSM` is smaller than in `PSM` due to the coefficient sharing between submodels. The best regularization parameter values for `SPSM` were $\gamma = 0.1, \lambda = 5.0$ which is lower than for ADNI, consistent with the larger data set size. The best performing model is `MLP` ($R^2$ of 0.56) for SUPPORT regression. However, the black-box nature of `MLP` is not conducive to reasoning about the influence of the missingness pattern. Mean and zero imputation have the best validation performance for `Ridge`, `XGB` and `MLP`. In summary, `SPSM` is consistently among the best-performing models in both data sets, with fairly tight confidence intervals. In ADNI classification, `SPSM`, `MLP` and `LR` achieve the highest prediction accuracy (0.84–0.85) and Area Under the ROC Curve (AUC) (0.85–0.86). All methods perform similarly well on ADNI.

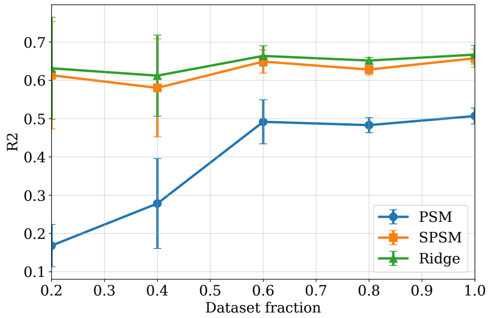

Figure 3: Performance on ADNI for the regression task. Error bars indicate standard deviation over 5 random subsamples of the data. Equal performance for `SPSM` and `Ridge` and subpar performance for `PSM` indicates that for ADNI regression, pattern specialization is mostly irrelevant.

`SPSM` selected $\gamma = 0$ and $\lambda = 1.0$ which indicates moderate coefficient sharing. For SUPPORT data, all models perform almost at the same level. `XGB` and `MLP` perform slightly better than `SPSM` ($\gamma$ = 0.1, $\lambda$ = 10.0) and `PSM`. Across ADNI and SUPPORT `LR`, `XGB` and `MLP` predominantly use zero imputation. In all tasks, `SPSM` performs comparably or favorably to all other methods. The tight confidence intervals for classification in both data sets indicate high certainty in the result averages.

**Non-Healthcare Data and Coefficient Specialization** In contrast to the previous data sets, where sharing coefficients is beneficial, we see for the HOUSING data, a large advantage from nonlinear estimation: the tree-based approach `XGB` (Table 9). It shows an $R^2$ of 0.76 and outperforms the other baseline methods for the regression task confirming the non-linearity of that data set. We also do not see the same positive effect in specializing (`PSM`, `SPSM` not better than `Ridge` with imputation). None of the missing value indicators show a significant feature importance level in `XGB` which might indicate that pattern specialization is not necessary. For results on the HOUSING data, see Appendix C.3.

**Performance with Varying Training Set Size** Figure 3 shows the test $R^2$ for linear models trained on different fractions of ADNI data. Each set was subsampled into fractions $0.2, 0.4, 0.6, 0.8, 1.0$ of the full data set. Especially for small fractions, `SPSM` benefits from coefficient sharing and lower variance data compared to `PSM`. `Ridge` with mean imputation performs comparably. A similar figure for the SUPPORT is presented in appendix Figure 7. `SPSM` and `PSM` perform equally well across the fractions, whereas `Ridge` shows high error compared to both pattern submodels.

**Pattern Specialization in `SPSM`** We inspect pattern specializations $\Delta$ for `SPSM` in the ADNI regression task with respect to interpretablity. In Table 3, we present the main model $\theta$ and pattern-specific coefficients $\Delta_4$ for pattern 4. Table 7 in the appendix shows all patterns $m$ with $\Delta_{\neg m} \neq 0$. For pattern 4, measurements of the amyloid-$\beta$ (ABETA) peptide and the proteins TAU and PTAU are missing in the

Missing features in pattern 4:
ABETA, TAU and PTAU at baseline (bl)

| **Feature** | $\Delta_4$ | $\theta$ | $\theta + \Delta_4$ |
|---|---|---|---|
| Age | -0.140 | 0.121 | -0.019 |
| FDG-PET | -0.090 | -0.039 | -0.129 |
| Whole Brain (bl) | 0.000 | -0.045 | -0.044 |
| Fusiform | 0.016 | 0.021 | 0.037 |
| ICV | 0.001 | 0.093 | 0.094 |
| Intercept | -0.10 | 0.18 | |

Table 3: Example of $\Delta_4$ for regression using `SPSM` using ADNI. `SPSM` takes $\gamma = 10$ and $\lambda = 13$ as parameters for a single seed. There are 10 missingness pattern in total, while 4 of them have non-zero coefficients for $\Delta$ and pattern-specific intercept. Coefficients are for standardized variables.

baseline diagnostics. The absence of these three features affects pattern specialization: For an imaging test FDG-PET (fluorodeoxyglucose), the magnitude of its coefficient is increased, placing heavier weight on the feature in prediction. Similarly, the coefficients for Fusiform (brain volume), and ICV (intracranial volume) increase in magnitude and predictive significance when ABETA, TAU, and PTAU are absent. In contrast, for the feature AGE, the resulting coefficient of -0.019 (compared to 0.121 in the main model) means that the predictive influence of this feature decreases under pattern 4. As Table 3 shows, `SPSM` applied to tabular data allows for short descriptions of pattern specialization, which helps construct a simple and meaningful model. We enforce sparsity in $\Delta$ to limit the number of differences between submodels, and present all features $j$ with specialized coefficients $\Delta_{\neg m}(j) \neq 0$, five in the example case. In this way, the set of submodels is more interpretable and the user, e.g., a medical staff member can be supported in decision-making. For a more detailed analysis on interpretability properties of `SPSM`, see Appendix C.4.

**Tradeoff between Interpretability and Accuracy** The interpretability-accuracy tradeoff is especially crucial for practical use of `SPSM`. The empirical results do not show any significant evidence that our proposed sparsity regularization hurts prediction accuracy (Table 2, Figure 3). Nevertheless, in a practical scenario, domain experts may choose a simpler model at a slight cost in performance. Then, we can measure the tradeoff by varying values of hyperparameters to find an adequate balance (Figure 8). The parameter selection is based on the validation set and aligns with the test set results. We see some parameter sensitivity in SUPPORT that supports sharing, but only in a moderate way.

## 6 Related Work

*Pattern-mixture missingness* refers to distributions well-described by an independent missingness component and a covariate model dependent on this pattern (Rubin 1976; Little 1993). In this work, *pattern missingness* refers to emergent patterns which may or may not depend on observed covariates (Marshall et al. 2002). Mercaldo and Blume (2020); Le Morvan et al. (2020b) and Bertsimas, Delarue, and Pauphilet (2021) define pattern submodels for flexible handling of test time missingness. The ExpandedLR method of Le Morvan et al. (2020b) represents a related method to pattern submodels. However, they neither study coefficient sharing between models nor provide a theoretical analysis of when optimal submodels have partly identical coefficients (sharing, sparsity in specialization). Marshall et al. (2002) describes the one-step sweep method using estimated coefficients and an augmented covariance matrix obtained from fully observed and incomplete data at test time. In very recent and so far unpublished work, Bertsimas, Delarue, and Pauphilet (2021) present two methods for predicting with test time missingness. First, *Affinely adaptive regression* specializes a shared model by applying a coefficient correction given by a linear function of the missingness pattern. When the number of variables $d$ is smaller than the number of patterns (which could grow as $2^d$), and the outcome is not smooth in changes to missingness mask, this may introduce significant bias. The resulting bias-variance tradeoff differs from our method, and unlike our work, is not justified by theoretical analysis. Second, *Finitely adaptive regression* starts by placing each pattern in the same model, recursively partitioning them into subsets.

Several deep learning methods which are applicable under test time missingness with or without explicitly attempting to impute missing values have been proposed (Bengio and Gingras 1995; Che et al. 2018; Le Morvan et al. 2020a,b; Nazabal et al. 2020). The NeuMiss network, discussed briefly in Section 4.1, proposes a new type of non-linearity: the multiplication by the missingness indicator (Le Morvan et al. 2020a). NeuMiss approximates the specialization term $\Delta_{\neg m}^{\top} X_{\neg m}$ (along with per-pattern biases) using a deep neural network where both covariates and missingness mask are given as input, sharing parameters across patterns. NeuMiss and Affinely adaptive regression (see above) are similar since their pattern specializations are functions of the inputs and the masks, both in contrast to SPSM. Moreover, neither method attempts to learn sparse specialization terms (e.g., no $\ell_1$ regularization of $\Delta$).

## 7 Conclusion

We have presented sharing pattern submodels (`SPSM`) for prediction with missing values at test time. We enforce parameter sharing through sparsity in pattern coefficient specializations via regularization and analyze `SPSM`'s consistency properties. We have described settings where information sharing is optimal even when the prediction target depends on missing values and the missingness pattern itself. Experimental results using synthetic and real-world data confirm that `SPSM` performs comparably or slightly better than baselines across all data sets without relying on imputation. Notably, the proposed method never performs worse than non-sharing pattern submodels as these do not use the available data efficiently. While `SPSM` is limited to learning linear models, it is not limited to learning from linear systems. An interesting direction is to identify other classes of models developed with interpretability that could benefit from this type of sharing.

# Acknowledgements

We want to thank Devdatt Dubhashi and Marine Le Morvan for their support and fruitful discussions.

This work was partly supported by WASP (Wallenberg AI, Autonomous Systems and Software Program) funded by the Knut and Alice Wallenberg foundation.

The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf

# References

Bengio, Y.; and Gingras, F. 1995. Recurrent neural networks for missing or asynchronous data. *Advances in neural information processing systems*, 8.

Bertsimas, D.; Delarue, A.; and Pauphilet, J. 2021. Prediction with Missing Data. *ArXiv*, abs/2104.03158.

Byrd, R. H.; Lu, P.; Nocedal, J.; and Zhu, C. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5): 1190–1208.

Carpenter, J.; and Kenward, M. 2012. *Multiple imputation and its application*. John Wiley & Sons.

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1): 1–12.

Chen, T.; He, T.; Benesty, M.; and Khotilovich, V. 2019. Package 'xgboost'. *R version*, 90.

Cowan, N. 2010. The magical mystery four: How is working memory capacity limited, and why? *Current directions in psychological science*, 19(1): 51–57.

Dancer, D.; and Tremayne, A. 2005. R-squared and prediction in regression with ordered quantitative response. *Journal of Applied Statistics*, 32(5): 483–493.

De Cock, D. 2011. Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project. *Journal of Statistics Education*, 19(3).

Fagerland, M. W.; Lydersen, S.; and Laake, P. 2015. Recommended confidence intervals for two independent binomial proportions. *Statistical methods in medical research*, 24(2): 224–254.

Groenwold, R. H.; White, I. R.; Donders, A. R.; Carpenter, J. R.; Altman, D. G.; and Moons, K. G. 2012. Missing covariate data in clinical research: when and when not to use the missing-indicator method for analysis. *CMAJ : Canadian Medical Association journal = journal de l'Association medicale canadienne*, 184(11): 1265–1269.

Hanley, J.; and McNeil, B. 1983. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3): 839–843.

Henelius, A.; Puolamäki, K.; and Ukkonen, A. 2017. Interpreting classifiers through attribute interactions in datasets. *arXiv preprint arXiv:1707.07576*.

Jones, M. P. 1996. Indicator and stratification methods for missing explanatory variables in multiple linear regression. *Journal of the American statistical association*, 91(433): 222–230.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Knaus, W. A.; Harrell, F. E.; Lynn, J.; Goldman, L.; Phillips, R. S.; Connors, A. F.; Dawson, N. V.; Fulkerson, W. J.; Califf, R. M.; Desbiens, N.; et al. 1995. The SUPPORT prognostic model: Objective estimates of survival for seriously ill hospitalized adults. *Annals of internal medicine*, 122(3): 191–203.

Le Morvan, M.; Josse, J.; Moreau, T.; Scornet, E.; and Varoquaux, G. 2020a. NeuMiss networks: differentiable programming for supervised learning with missing values. *arXiv:2007.01627*.

Le Morvan, M.; Josse, J.; Scornet, E.; and Varoquaux, G. 2021. What'sa good imputation to predict with missing values? *Advances in Neural Information Processing Systems*, 34.

Le Morvan, M.; Prost, N.; Josse, J.; Scornet, E.; and Varoquaux, G. 2020b. Linear predictor on linearly-generated data with missing values: non consistency and solutions. In Chiappa, S.; and Calandra, R., eds., *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, 3165–3174. PMLR.

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

Little, R. J. 1993. Pattern-mixture models for multivariate incomplete data. *Journal of the American Statistical Association*, 88(421): 125–134.

Liu, X.; Zachariah, D.; and Stoica, P. 2020. Robust Prediction When Features are Missing. *IEEE Signal Processing Letters*, 27: 720–724.

Marlin, B. M. 2008. *Missing Data Problems in Machine Learning*. Ph.D. thesis, University of Toronto.

Marshall, G.; Warner, B.; MaWhinney, S.; and Hammermeister, K. 2002. Prospective prediction in the presence of missing data. *Statistics in medicine*, 21(4): 561–570.

Meinshausen, N. 2007. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1): 374–393.

Mercaldo, S. F.; and Blume, J. D. 2020. Missing data and prediction: the pattern submodel. *Biostatistics*, 21(2): 236–252.

Miller, G. A. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2): 81.

Mofrad, S. A.; Lundervold, A. J.; Vik, A.; and Lundervold, A. S. 2021. Cognitive and MRI trajectories for prediction of Alzheimer's disease. *Scientific Reports*, 11(1): 1–10.

Nazabal, A.; Olmos, P. M.; Ghahramani, Z.; and Valera, I. 2020. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107: 107501.

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. 2011a. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

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. 2011b. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Rubin, D. B. 1976. Inference and missing data. *Biometrika*, 63(3): 581–592.

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.

Seaman, S.; Galati, J.; Jackson, D.; and Carlin, J. 2013. What is meant by ”missing at random”? *Statistical Science*, 28(2): 257–268.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1): 267–288.

Van Buuren, S. 2018. *Flexible Imputation of Missing Data (2nd ed.)*. Chapman and Hall/CRC, Boca Raton, FL.

Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3): 261–272.

Weiner, M. W.; Aisen, P. S.; Jack Jr., C. R.; Jagust, W. J.; Trojanowski, J. Q.; Shaw, L.; Saykin, A. J.; Morris, J. C.; Cairns, N.; Beckett, L. A.; Toga, A.; Green, R.; Walter, S.; Soares, H.; Snyder, P.; Siemers, E.; Potter, W.; Cole, P. E.; Schmidt, M.; and Initiative, A. D. N. 2010. The Alzheimer’s Disease Neuroimaging Initiative: Progress report and future plans. *Alzheimer’s & Dementia*, 6(3): 202–211.e7.

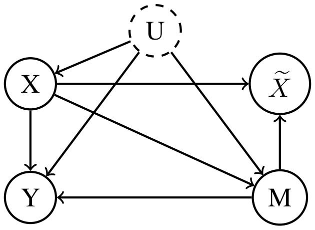

Figure 4: Directed graph showing assumed probabilistic dependencies. $\tilde{X}$ is a deterministic function of $X, M$. Unobserved variables $U$ may influence both covariates $X$, missingness $M$ and the outcome $Y$, ruling out 'missing at random' (MAR).

# A Technical appendix

## A.1 Variable dependencies

The assumed (causal) dependencies of the variables $X, M, Y$ are represented in a directed graph in Figure 4.

## A.2 Consistency in the general case

**Proposition 3.** *For each pattern $m$, the minimizers $(\theta^*, \Delta^*_{\neg m})$ of* (5) *are consistent estimators of the best linear approximation to $\mathbb{E}[Y \mid X_{\neg m}, M = m]$,*

$$\lim_{n\to\infty}(\theta^*_{\neg m} + \Delta^*_{\neg m}) = \min_{\eta} \mathbb{E}[(\eta^\top X_{\neg m} - Y)^2 \mid M = m] \; .$$

*When the true outcome is linear, $Y = \eta_{\neg m}^\top X_{\neg m} + \epsilon$ with Gaussian errors $\epsilon$, $\lim_{n\to\infty}(\theta^*_{\neg m} + \Delta^*_{\neg m}) = \eta_{\neg m}$ .*

*Proof sketch.* Minimizers $\Delta^*$ and $\theta^*$ will have bounded norm due to the quadratic form of the objectives. This, in the limit $n \to \infty$, regularization terms vanish due to normalization with $n$ and the minimizers $(\theta^*, \{\Delta^*_{\neg m}\})$ are invariant to additive transformations; with $c \in \mathbb{R}^{d_m}$, $\theta'_{\neg m} = \theta^*_{\neg m} + c$ and $\Delta'_{\neg m} = \Delta^*_{\neg m} - c$ also minimize the objective. Choosing $c = -\theta^*_{\neg m}$, we get $\theta'_{\neg m} = 0$ and the objective becomes separable in $m$. As a result, the objective can be written as $k$ standard least squares problems, one for each pattern. As is well known, for additive sub-Gaussian noise, the minimizers of these problems are consistent for the best linear approximation to the corresponding conditional mean. □

## A.3 Proof of Proposition 1

**Proposition** (Proposition 1 Restated)**.** *Suppose covariates $X$ and outcome $Y$ obey Condition 1 (are linear-Gaussian). Then, the Bayes-optimal predictor for an arbitrary missingness mask $m \in \mathcal{M}$, is*

$$f^*_m = \mathbb{E}[Y \mid X_{\neg m}, m] = (\theta_{\neg m} + \Delta_{\neg m})^\top X_{\neg m} + C_m$$

*where $C_m \in \mathbb{R}$ is constant with respect to $X_{\neg m}$ and*

$$\Delta_{\neg m} = (\Sigma^{-1}_{\neg m, \neg m})\Sigma_{\neg m, m}\theta_m \; .$$

*Proof.* By properties of the multivariate Normal distribution, we have that

$$\begin{aligned} &\mathbb{E}_{X_m}[X_m \mid X_{\neg m}] \\ &= \mathbb{E}[X_m] + \Sigma_{m,\neg m}\Sigma^{-1}_{\neg m, \neg m}(X_{\neg m} - \mathbb{E}[X_{\neg m}]) \end{aligned}$$

and as a result, following the reasoning above,

$$\begin{aligned} &\mathbb{E}[Y \mid X_{\neg m}] \\ &= (\theta_{\neg m} + (\Sigma_{m,\neg m}\Sigma^{-1}_{\neg m,\neg m})\theta_m)^\top X_{\neg m} + C_m \\ &= (\theta_{\neg m} + \Delta_m)^\top X_{\neg m} + C_m, \end{aligned}$$

where $C_m = \theta_m^\top(\mathbb{E}[X_m] - \Sigma_{m,\neg m}\Sigma^{-1}_{\neg m,\neg m}\mathbb{E}[X_{\neg m}]) + \alpha_m$, which is constant w.r.t. $X_{\neg m}$. □

## A.4 Sparsity in optimal model

**Proposition** (Proposition 2 restated)**.** *Suppose that a covariate $j \in [d]$ is observed under pattern $m$, i.e., $m_j = 0$, and assume that $X_j$ is non-interactive with every covariate $X_{j'}$ that is missing under $m$. Then $(\Delta_{\neg m})_j = 0$.*

*Proof.* Let $\bar{\theta}_{\neg m} = \theta_{\neg m} + \Delta_{\neg m}$. Recall that $S = \Sigma^{-1}$ is the precision matrix for $X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and permute the rows and columns of $\Sigma$ into observed and unobserved parts, such that, without loss of generality, we can write

$$\Sigma = \begin{bmatrix} \Sigma_{\neg m,\neg m} & \Sigma_{\neg m,m} \\ \Sigma^T_{\neg m,m} & \Sigma_{m,m} \end{bmatrix} . \tag{8}$$

Note that by the definition of $\Delta_{\neg m}$ (Proposition 1),

$$\begin{aligned} \Sigma \begin{bmatrix} \Delta_{\neg m} \\ -\bar{\theta}_{\neg m} \end{bmatrix} &= \begin{bmatrix} \Sigma_{\neg m,\neg m} & \Sigma_{\neg m,m} \\ \Sigma^T_{\neg m,m} & \Sigma_{m,m} \end{bmatrix} \begin{bmatrix} \Delta_{\neg m} \\ -\bar{\theta}_{\neg m} \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ g_m \end{bmatrix}, \end{aligned}$$

where $g_m$ is a suitable vector. Hence

$$\begin{bmatrix} \Delta_{\neg m} \\ -\bar{\theta}_{\neg m} \end{bmatrix} = \Sigma^{-1} \begin{bmatrix} 0 \\ g_m \end{bmatrix} = S \begin{bmatrix} 0 \\ g_m \end{bmatrix}$$

We conclude the result by noting that $(\Delta_{\neg m})_j$ is zero if in the $j$th row of $S$, all entries corresponding to the unobserved part is zero. □

## A.5 Consistency in linear-Gaussian DGPs

**Theorem** (Theorem 1 restated)**.** *Suppose that Condition 1 holds with parameters $(\theta, \{\Delta_{\neg m}\})$ as in Proposition 1, such that, for each covariate $j$, the number of patterns $m$ for which $m_j = 0$ and $(\Delta_{\neg m})_j = 0$ is strictly larger than the number of patterns $m'$ for which $m'_j = 0$ and $(\Delta_{\neg m'})_j \neq 0$. Then, with $\gamma = 0$ and $\lambda > 0$, the true parameters $(\theta, \{\Delta_{\neg m}\})$ are the unique solution to* (5) *in the large-sample limit, $n \to \infty$.*

*Proof.* Consider the optimization problem in eq. (5) with $\gamma = 0$. In the large-sample limit $(n \to \infty)$, minimizers of the empirical risk over $n$ samples will also minimize the expected risk and, since the outcome is linear-Gaussian, satisfy the constraint in eq. (9). Then, solving (5) is equivalent to solving the following problem:

$$\begin{aligned} \underset{\theta', \{\Delta'_{\neg m}\}}{\text{minimize}} \quad & \sum_m \|\Delta'_{\neg m}\|_1 \\ \text{subject to} \quad & \theta'_{\neg m} + \Delta'_{\neg m} = \theta_{\neg m} + \Delta_{\neg m}, \quad m \in \mathcal{M} \end{aligned} \tag{9}$$

Many parameters $(\theta', \Delta')$ can satisfy the constraint, due to translational invariance. However, for any value of $\lambda > 0$, regularization in (5) steers the solution towards the one with the smallest norm, $\|\Delta'\|_1$. The reasoning is similar to the argument in the proof of Proposition 3, adding the assumption that the true system is linear-Gaussian. Under the added assumptions of Theorem 1, we can now prove that we also get the correct decomposition.

Take a solution $\theta^*, \{\Delta^*_{\neg m}\}$ of (9). For simplicity of notation below, let vectors $\theta, \Delta_{\neg m}, \theta^*, \Delta^*_{\neg m}$ always be indexed such that the same index $j$ refers to coefficients corresponding to the same covariate $X_j$. Next, define $I_j = \{m \mid m_j = 0,\ (\Delta_{\neg m})_j = 0\}$ to be the set of patterns where covariate $j$ is observed and without specialization under the optimal model. Similarly, define $I_j^c = \{m \mid m_j = 0,\ (\Delta_{\neg m})_j \neq 0\}$ to be the set of patterns where covariate $j$ is observed and needs specialization. First, note that

$$\begin{aligned}\sum_m \|\Delta^*_{\neg m}\|_1 &= \sum_j \sum_{m \mid m_j = 0} |(\Delta^*_{\neg m})_j| \\ &= \sum_j \sum_{m \in I_j} |(\Delta^*_{\neg m})_j| + \sum_j \sum_{m \in I_j^c} |(\Delta^*_{\neg m})_j| \,.\end{aligned}$$

For $m \in I_j$, we have $\theta_j^* + (\Delta^*_{\neg m})_j = \theta_j$. Hence

$$\sum_j \sum_{m \in I_j} |(\Delta^*_{\neg m})_j| = \sum_j |I_j||\theta_j - \theta_j^*| \qquad (10)$$

For $m \in I_j^c$, we have $\theta_j^* + (\Delta^*_{\neg m})_j = \theta_j + (\Delta_{\neg m})_j$ and hence by the triangle inequality, we have

$$\begin{aligned}\sum_j \sum_{k \in I_j^c} |(\Delta_k^*)_j| \geq \sum_j \sum_{k \in I_j^c} \big(|(\Delta_k)_j| - |\theta_j - \theta_j^*|\big) = \\ \sum_m \|\Delta_{\neg m}\|_1 - \sum_j |I_j^c||\theta_j - \theta_j^*| \qquad (11)\end{aligned}$$

We conclude that

$$\begin{aligned}\sum_m \|\Delta^*_{\neg m}\|_1 &\geq \sum_m \|\Delta_{\neg m}\|_1 + \sum_j (|I_j| - |I_j^c|)|\theta_j - \theta_j^*| \\ &\geq \sum_m \|\Delta_{\neg m}\|_1\end{aligned}$$

where the last inequality is by the assumption. This provides the desired result. □

# B Experiment details

## B.1 Real world data sets

**ADNI** The compiled data set includes 1337 subjects that were preprocessed by one-hot encoding of the categorical features and standardized for the numeric features. The processed data has 37 features and 20 unique missingness patterns. The label set is quite unbalance showing 1089 patients who do not change from their baseline diagnosis, and 248 do. The regression task targets predicting the result of the cognitive test ADAS13 (Alzheimer's Disease Assessment Scale) at a 2 year follow-up (Mofrad et al. 2021) based on available data at baseline.

**SUPPORT** The data set contains 9104 subjects represented by 23 unique missingness pattern. The following 10 covariates were selected and standardized: partial pressure of oxygen in the arterial blood (pafi), mean blood pressure, white blood count, albumin, APACHE III respiration score, temperature, heart rate per minute, bilirubin, creatinine, and sodium.

## B.2 Details of the baseline methods

We compare to the following baseline methods:

**Imputation + Ridge / logistic regression** (`Ridge`/`LR`) the data is first imputed (see below) and a ridge or logistic regression is fit on the imputed data. The implementation in SciKit-Learn was used (Pedregosa et al. 2011a). The ridge coefficients are shirked by imposing a penalty on their size. They are a reduced factor of the simple linear regression coefficients and thus never attain zero values but very small values (Tibshirani 1996)

**Imputation + Multilayer perceptron** (`MLP`): The MLP estimator is based on a single hidden layer of size $\in [10, 20, 30]$ followed by a ReLu activation function and a softmax layer for classification tasks and a linear layer for regressions tasks. As input, the imputed data is concatenated with the missingness mask. The MLP is trained using ADAM (Kingma and Ba 2014), and the learning rate is initialized to constant (0.001) or adaptive. We use the implementation in SciKit-Learn (Pedregosa et al. 2011b).

**Pattern submodel** (`PSM`): For each pattern of missing data, a linear or logistic regression model is fitted, separately regularized with a $\ell_2$ penalty. Following Mercaldo and Blume (2020), for patterns with fewer than $2{*}d$ samples available, a complete-case model (CC) is used. Our implementation of `PSM` is based on a special case of our `SPSM` implementation where regularization is applied over all patterns and not in each pattern separately. To enforce fitting separated submodels for each pattern, we set $\gamma = 1e^8$ and $\lambda = 0$.

**XGBoost** (`XGB`): XGBoost is an implementation of gradient boosted decision trees. Note, XGBoost supports missing values by default (Chen et al. 2019), where branch directions for missing values are learned during training. A logistic classifier is then fit using XGBClassifier while regression tasks are trained with the XGBRegressor (Pedregosa et al. 2011b). We set the hyperparameters to 100 for the number of estimators used, and fix the learning rate to 1.0. The maximal depth of the trees is $\in [5, 10, 15]$.

Imputation methods and hyperparameters for all methods were selected based on the validation portion of random 64/16/20 training/validation/test splits. Results were averaged over five random splits of the data set. The performance metrics for classification tasks were accuracy and the Area Under the ROC Curve (AUC). For regression tasks, we use the mean squared error (MSE) and the R-square, ($R^2$) value, representing the proportion of the variance for a dependent variable that's explained by an independent variable, taking values in $[-\infty, 1]$ where negative values represent predictions worse than the mean (Dancer and Tremayne 2005). Confidence intervals at significance level $\alpha = 0.05$ are computed based on the number of test set samples. For accuracy, MSE and $R^2$ we use a Binomial proportion confidence interval (Fagerland, Lydersen, and Laake 2015) and for AUC we use the classical model of (Hanley and McNeil 1983).

**Computing Infrastructure** The computations required resources of 4 compute nodes using two Intel Xeon Gold 6130 CPUS with 32 CPU cores and 384 GiB memory (RAM). Moreover, a local disk with the type and size of SSSD 240GB with a local disk, usable area for jobs including 210 GiB was used. Inital experiments are run on a Macbook using macOS Montery with a 2,6 GHz 6-Core Intel Core i7 processor.

# C Additional experimental results

## C.1 Simulation results

Results for synthetic data with missingness Setting B (pattern-dependent) and Setting C (MCAR) can be found in Figures 5 and 6, respectively.

## C.2 Results for ADNI and SUPPORT

A figure illustrating the performance on SUPPORT with varying data set size is given in Figure 7. Table 4 presents the MSE score

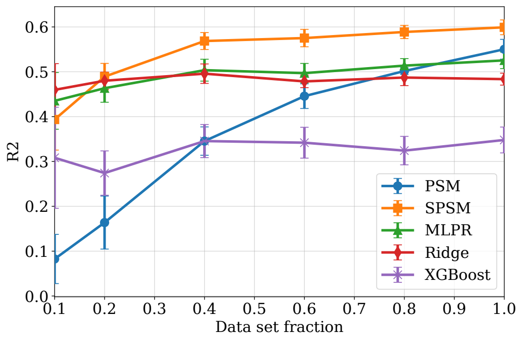

Figure 5: Performance on simulated data Setting B. Error bars indicate standard deviation over 5 random data splits. The complete data set has $n = 10000$ samples.

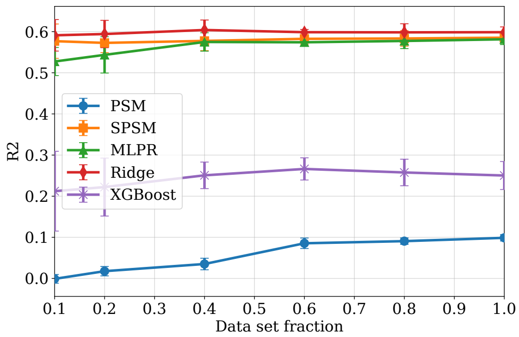

Figure 6: Performance on simulated data Setting C (MCAR). Error bars indicate standard deviation over 5 random data splits. The complete data set has $n = 10000$ samples.

| Linear Methods | MSE |
|---|---|
| *ADNI* | |
| Ridge, $I_0$ | 0.36 (0.26, 0.46) |
| XGB, $I_\mu$ | 0.60 (0.48, 0.74) |
| MLP, $I_\mu$ | 0.37 (0.27, 0.47) |
| PSM | 0.50 (0.38, 0.62) |
| SPSM | 0.35 (0.25, 0.45) |
| *SUPPORT* | |
| Ridge,$I_0$ | 0.61 (0.56, 0.66) |
| XGB, $I_\mu$ | 0.69 (0.63, 0.75) |
| MLP, $I_0$ | 0.44 (0.39, 0.48) |
| PSM | 0.47 (0.42, 0.52) |
| SPSM | 0.47 (0.42, 0.51) |

Table 4: Experimental results of regression methods for ADNI and SUPPORT data set.

as an additional performance metric for the regression tasks using ADNI and SUPPORT data. For the MAR setting in the SUPPORT data, we present the results for classification and regression tasks in Table 6 and Table 5. Moreover, the full table of pattern 4 non-zero coefficients with the corresponding missing features is displayed in Table 7.

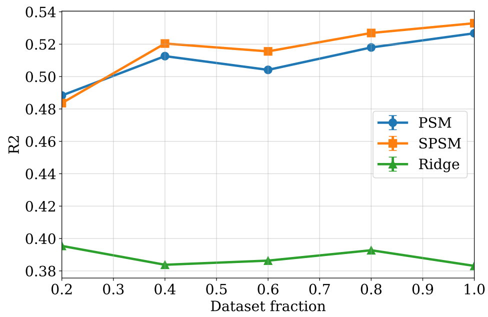

Figure 7: Performance on SUPPORT data for a regression task. Error bars indicate standard deviation over 5 random subsamples of the data.

| Regressions | $R^2$ | MSE |
|---|---|---|
| SUPPORT | | |
| Ridge, $I_0$ | 0.38 (0.34, 0.41) | 0.62 (0.57, 0.67) |
| XGB, $I_\mu$ | 0.27 (0.23, 0.31) | 0.73 (0.67, 0.78) |
| MLP, $I_0$ | 0.55 (0.52, 0.58) | 0.45 (0.40, 0.49) |
| PSM | 0.51 (0.48, 0.54) | 0.49 (0.44, 0.53) |
| SPSM | 0.52 (0.49, 0.58) | 0.47 (0.42, 0.51) |

Table 5: Experimental results of regression methods for SUPPORT data set MAR.

## C.3 HOUSING data

The Ames Housing data set (HOUSING) (De Cock 2011) was compiled by Dean De Cock for use in data science education. The data set describes the sale of individual residential property in Ames, Iowa from 2006 to 2010. The data set contains 2930 observations and a large number of explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) involved in assessing home values. In this study we used a subset of the features 27 features to describe the main characteristics of a house. Examples of features included are measurements about the land ('LotFrontage', 'LotArea', 'LotShape', 'LandContour', 'LandSlope'), the 'Neighborhood', and 'HouseStyle', when the house was build ('YearBuilt'), or remodeled ('YearRemodAdd'). Moreover, features describing the outside of the house ('RoofStyle', 'Foundation'), technical equipment ('Heating', 'CentralAir', 'Electrical', 'KitchenAbvGr', 'Functional', 'Fireplaces', 'GarageType', 'GarageCars', 'PoolArea', 'Fence', 'MiscFeature'), and information about previous house selling prices and conditions ('MoSold', 'YrSold', 'SaleType', 'SaleCondition'). The numeric features

| Classifiers | AUC | Accuracy |
|---|---|---|
| SUPPORT | | |
| LR, $I_0$ | 0.82 (0.80, 0.84) | 0.75 (0.72, 0.78) |
| XGB, $I_0$ | 0.83 (0.81, 0.85) | 0.76 (0.74, 0.78) |
| MLP, $I_0$ | 0.85 (0.84, 0.87) | 0.78 (0.76, 0.81) |
| PSM | 0.83 (0.81, 0.85) | 0.78 (0.74, 0.80) |
| SPSM | 0.83 (0.81, 0.85) | 0.76 (0.73, 0.80) |

Table 6: Experimental results of classifiers for SUPPORT data with MAR.

| No. of missingness pattern | Feature | $\Delta_m$ | $\theta$ | $\theta + \Delta_m$ |
|---|---|---|---|---|
| Missing features in pattern 0: None | | | | |
| 0 | Age | -0.038 | 0.121 | 0.082 |
| | EDUCAT | 0.014 | -0.005 | 0.009 |
| | APOE4 | 0.046 | -0.010 | 0.035 |
| | FDG | -0.032 | -0.039 | -0.071 |
| | ABETA | 0.027 | -0.000 | 0.027 |
| | LDELTOTAL | 0.051 | -0.391 | -0.340 |
| | Entorhinal | 0.007 | -0.131 | -0.124 |
| | ICV | 0.013 | 0.093 | 0.106 |
| | Diagnose MCI | 0.078 | -0.139 | -0.061 |
| | GEN Female | -0.054 | 0.003 | -0.050 |
| | GEN Male | 0.000 | 0.062 | 0.062 |
| | Not Hisp/ Latino | 0.047 | -0.114 | -0.067 |
| | Married | 0.115 | -0.159 | -0.044 |
| Missing features in pattern 1: FDG | | | | |
| 1 | Age | -0.052 | 0.121 | 0.069 |
| Missing features in pattern 4: ABETA, TAU and PTAU at baseline (bl) | | | | |
| 4 | Age | -0.140 | 0.121 | -0.019 |
| | FDG | -0.090 | -0.039 | -0.129 |
| | Whole Brain | 0.000 | -0.045 | -0.044 |
| | Fusiform | 0.016 | 0.021 | 0.037 |
| | ICV | 0.001 | 0.093 | 0.094 |
| Missing features in pattern 10: FDG, ABETA (bl), TAU (bl), PTAU (bl) | | | | |
| 10 | APOE4 | 0.038 | -0.010 | 0.027 |

Table 7: Full table showing $\Delta_m$ in the regression task using SPSM for ADNI.

| Pattern number | Number of subjects per pattern | $R^2$ |
|---|---|---|
| 0 | 119 | 0.64 (0.53, 0.75) |
| 1 | 30 | 0.30 (-0.10, 0.55) |
| 6 | 27 | 0.71 (0.50, 0.92) |
| 10 | 28 | 0.71 (0.50, 0.91) |
| others | $\leq 13$ | undefined or insignificant |

Table 8: A minimum sample size is required for SPSM to maintain predictive performance

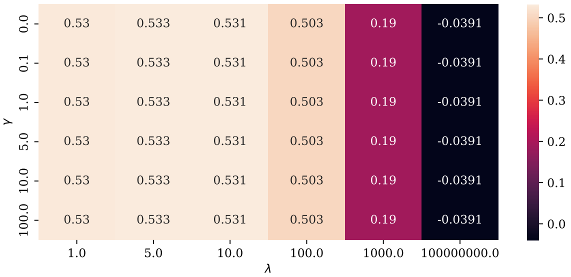

Figure 8: Heatmap visualizing the tradeoff between interpretability and prediction power including different hyperparameter values for $\gamma$ and $\lambda$, expressed by the $R^2$ using SUPPORT data. Each cell is indicating a $\gamma$,$\lambda$ combination, e.g. 1,100 represents 1 = $\gamma$ and 100 = $\lambda$.

where standardized and the categorical ones are one-hot-encoded during preprocessing. The HOUSING data set shows 15 different missingness patterns. An exploratory analysis has shown that the house sale prices are somehow skewed, which means that there is a large amount of asymmetry. The mean of the characteristics is greater than the median, showing that most houses were sold for less than the average price. In the classification predictions, we look if the sale prices for a house are above or below the median, while for regression tasks we predict the sale price for a house.

We report the results of the HOUSING data set in Table 9. In classification, on average a high performance over all models, whereas the best performing one, XGB achieves an AUC of 0.96 and an accuracy of 0.91. SPSM achieves only slightly lower prediction power of 0.95 AUC and 0.88 accuracies than XGB. While LR, XGB and MLP depend on mean or zero imputation, PSMand SPSM are able to achieve comparable results without adding bias to their prediction with high confidence on average. For the HOUSING regression, the validation power suggested $\gamma = 10$, $\lambda = 100$ for SPSM, resulting in an $R^2$ of 0.64 and an MSE of 0.39. This result is better than for PSM($R^2$ of 0.58 and MSE of 0.46) and thus demonstrates the benefit of coefficient sharing in SPSM compared to no sharing. Although Ridge, and MLP perform better the differences are only marginal to SPSM. The best performing model is the black-box method of XGB achieving an $R^2$ of 0.76 and MSE of 0.27 indicating the non-linearity of the data set.

## C.4 Analysis of interpretability

By enforcing sparsity in pattern specialization, we ensure that the resulting subset of features is reduced to relevant differences which will foster interpretability for domain experts; SPSM allows for more straight-forward reasoning about the similarity between submodels and the effects of missingness. Lipton et al. (2016) provides qualitative design criteria to address model properties and techniques thought to confer interpretability. We will show that SPSM satisfies some form of transparency by asking, i.e., *how does the model work?*. As stated in (Lipton et al. 2016), transparency is the absence of opacity or black-boxness meaning that the mechanism by which the model works is understood by a human in some way. We evaluate transparency at the level of the entire model (simulatability), at the level of the individual components (e.g., parameters) (decomposability), and at the level of the training algorithm (algorithmic transparency). First, simulatability refers to contemplating the entire model at once and is satisfied in SPSM by it's nature of a sparse linear model, as produced by lasso regression (Tibshirani 1996). Moreover, we claim that SPSM is small and simple (Rudin 2019), in that we allow a human to take the input data along with the parameters of the model and perform in a reasonable amount of time all the computations necessary to make a prediction in order to fully understand a model. The aspect of decomposabilty (Lipton et al. 2016) can be satisfied by using tabular data where features are intuitively meaningful. To that end, we use two real-world tabular data sets in the experiments and present the coefficient values for input features in Table 3. More-

**Housing**

| *Classification* | **AUC** | **Accuracy** |
|---|---|---|
| LR, $I_\mu$ | 0.96 (0.94, 0.98) | 0.90 (0.85, 0.95) |
| XGB, $I_0$ | 0.96 (0.94, 0,98) | 0.91 (0.87, 0.96) |
| MLP, $I_0$ | 0.96 (0.93, 0.98) | 0.90 (0.85, 0.94) |
| PSM | 0.93 (0.90, 0.96) | 0.88 (0.83, 0.93) |
| SPSM | 0.95 (0.92, 0.97) | 0.88 (0.83, 0.94) |
| *Regression* | $\boldsymbol{R}^2$ | **MSE** |
| Ridge, $I_\mu$ | 0.68 (0.62, 0.75) | 0.35 (0.25, 0.44) |
| XGB, $I_0$ | 0.76 (0.70, 0.81) | 0.27 (0.18,0.35) |
| MLP, $I_0$ | 0.64 (0.58, 0.71) | 0.39 (0.29, 0.49) |
| PSM | 0.58 (0.50, 0.65) | 0.46 (0.35, 0.57) |
| SPSM | 0.64 (0.57, 0.71) | 0.39 (0.29, 0.49) |

Table 9: Experimental results of classification and regression methods for HOUSING data set.

over, one can choose to display the coefficients in a standardized or non-standardized way to provide even better insights. The comprehension of the coefficients depends also on domain knowledge. Finally, algorithmic transparency is given in SPSM, since in linear models, we understand the shape of the error surface and have some confidence that training will converge to a unique solution, even for previously unseen test data. Additionally, Henelius, Puolamäki, and Ukkonen (2017) claims that knowing interactions between two or more attributes makes a model more interpretable. SPSM shows in $\theta + \Delta$ the coefficient specialization between the main model and the pattern-specific model and therefore reveals associations between attributes.