

# KNOCKOUT: A SIMPLE WAY TO HANDLE MISSING INPUTS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Deep learning models can extract predictive and actionable information from complex inputs. The richer the inputs, the better these models usually perform. However, models that leverage rich inputs (e.g., multi-modality) can be challenging to deploy widely, because some inputs may be missing at inference. Current popular solutions to this issue include marginalization, imputation, and training multiple models. Marginalization can obtain calibrated predictions, but it is computationally expensive and therefore only feasible for low dimensional inputs. Imputation may result in inaccurate predictions because it employs point estimates for missing variables and does not work well for high dimensional inputs (e.g., images). Training multiple models, where each model is designed to handle different subsets of inputs, can work well but requires knowing prior knowledge of missing input patterns. Furthermore, training and retaining multiple models can be costly. We propose an efficient method to learn both the conditional distribution using full inputs and the marginal distributions. Our method, Knockout, randomly replaces input features with appropriate placeholder values during training. We provide a theoretical justification for Knockout and show that it can be interpreted as an implicit marginalization strategy. We evaluate Knockout across a wide range of simulations and real-world datasets and show that it offers strong empirical performance.

## 1 INTRODUCTION

In many real-world applications of machine learning and statistics, not all variables might be available for every data point. This issue, as known as missingness, is well-studied in the literature (Little & Rubin, 2019) and common in fields like healthcare, social sciences, and environmental studies. From a Bayesian perspective, missingness can be viewed as a marginalization problem, where we would like a model to marginalize out the missing variables from the conditioning set. However, during training, we often do not know which features will be missing at inference time.

In lieu of training multiple models for every missingness pattern, a common strategy is imputation, which uses a point estimate (usually the mean or mode or a constant) to impute the missing feature (Le Morvan et al.). This can be seen as approximating the marginalization with a delta function. More sophisticated methods for handling missingness include using EM imputation (Josse et al., 2019) or neural-based imputation (Mattei & Frellsen, 2019; Ipsen et al., 2022). Although many prior methods may work well in some instances, they may not scale readily to high-dimensional inputs like images (Kyono et al., 2021; You et al., 2020), require additional networks for generation of missing variables (Ipsen et al., 2022), only apply to continuous inputs (Le Morvan et al., 2020, 2021), assume linearity of predictors (Le Morvan et al.), or make assumptions about the data distribution (Hazan et al., 2015).

In this work, we propose a simple, effective, and theoretically-justified augmentation strategy, called Knockout, for handling missing inputs. During training, features are augmented by randomly “knocking out” and replacing them with constant “placeholder” values. At inference time, using the placeholder value corresponds mathematically to estimation with the appropriate marginal distribution. In particular, we demonstrate how Knockout can be seen as implicitly maximizing the likelihood of a weighted sum of the conditional estimators and all desired marginals *in a single model*.

In a suite of experiments, we demonstrate the broad applicability of Knockout. We use both synthetic and real-world data with image-based and tabular inputs. Real world experiments include Alzheimer’s forecasting, noisy label learning, multi-modal MR image segmentation and detection, and multi-view tree genus classification. We show the effectiveness of Knockout in handling low and high-dimensional missing inputs, and compare it with appropriate baselines, including imputation and ensemble-based methods.

## 2 METHOD

### 2.1 BACKGROUND

The goal of supervised machine learning (ML) is to learn the conditional distribution  $p(Y|\mathbf{X})$  where  $Y$  is the output (predictive target) and  $\mathbf{X} \in \mathbb{R}^N$  are the vector of inputs or features. The prediction for a new sample  $\mathbf{x}$  is  $\hat{y} = \arg \max_Y p(Y|\mathbf{X} = \mathbf{x})$ . However, in many practical applications, not all features may be present for a given input. Consider the case when  $X_i$  is missing, and denote the vector of non-missing features as  $\mathbf{X}_{-i}$ . In general, multiple features may be missing at a time, and we can represent this with a missingness indicator set  $\mathcal{M}$  and corresponding non-missing features as  $\mathbf{X}_{-\mathcal{M}}$ . In this case, what we really want is  $p(Y|\mathbf{X}_{-\mathcal{M}})$ .

How can we account for missingness? A simple approach is to train a separate model for  $p(Y|\mathbf{X}_{-\mathcal{M}})$ , i.e. a model that takes only the non-missing features  $\mathbf{X}_{-\mathcal{M}}$  as inputs. However, this is expensive because a separate model is needed for each missingness pattern. Furthermore, there is no sharing of information between these separate models, even though they are theoretically related.

Another approach relies on rewriting  $p(Y|\mathbf{X}_{-\mathcal{M}})$  in terms of the already available  $p(Y|\mathbf{X})$ :

$$p(Y|\mathbf{X}_{-\mathcal{M}}) = \int_{\mathbf{X}_{\mathcal{M}}} p(Y, \mathbf{X}_{\mathcal{M}}|\mathbf{X}_{-\mathcal{M}})d\mathbf{X}_{\mathcal{M}} = \int_{\mathbf{X}_{\mathcal{M}}} p(Y|\mathbf{X})p(\mathbf{X}_{\mathcal{M}}|\mathbf{X}_{-\mathcal{M}})d\mathbf{X}_{\mathcal{M}}. \quad (1)$$

The goal now is to obtain  $p(\mathbf{X}_{\mathcal{M}}|\mathbf{X}_{-\mathcal{M}})$  and perform the integration over all possible  $\mathbf{X}_{\mathcal{M}}$ .

Imputation methods approximate Eq. (1) by replacing  $p(\mathbf{X}_{\mathcal{M}}|\mathbf{X}_{-\mathcal{M}})$  with a delta function. For example, “mean imputation” uses the mean of the missing features  $\mathbf{X}_{\mathcal{M}}$ ,  $\mathbb{E}[\mathbf{X}_{\mathcal{M}}]$ , for  $\mathbf{X}_{\mathcal{M}}$  itself. In Eq. (1) this corresponds to approximating  $p(\mathbf{X}_{\mathcal{M}}|\mathbf{X}_{-\mathcal{M}}) \approx \delta(\mathbb{E}[\mathbf{X}_{\mathcal{M}}])$ , a delta function. While convenient and commonly used, mean imputation ignores the dependency between  $\mathbf{X}_{\mathcal{M}}$  and  $\mathbf{X}_{-\mathcal{M}}$ , and does not account for any uncertainty.

More sophisticated approaches to imputation capture the interdependencies between inputs (Troyanskaya et al., 2001; Stekhoven & Bühlmann, 2012), for example by explicitly modeling  $p(\mathbf{X}_{\mathcal{M}}|\mathbf{X}_{-\mathcal{M}})$  by training a separate model. At inference time, the point estimate  $\mathbf{x}_{\mathcal{M}} = \arg \max_{\mathbf{X}_{\mathcal{M}}} p(\mathbf{X}_{\mathcal{M}}|\mathbf{X}_{-\mathcal{M}})$  can be used for the missing  $\mathbf{X}_{\mathcal{M}}$ . While properly accounting for interdependencies between inputs, this approach requires fitting a separate model for  $p(\mathbf{X}_{\mathcal{M}}|\mathbf{X}_{-\mathcal{M}})$ . In multiple imputation, multiple samples from  $p(\mathbf{X}_{\mathcal{M}}|\mathbf{X}_{-\mathcal{M}})$  are drawn and a Monte Carlo approximation is used to estimate the integral on the RHS of Eq. (1) (Kyono et al., 2021). Although this is more accurate than single imputation, it is not effective in high dimensional space.

### 2.2 KNOCKOUT

We propose a simple augmentation strategy for neural network training called Knockout that enables estimation of the conditional distribution  $p(Y|\mathbf{X})$  and all desired marginals  $p(Y|\mathbf{X}_{-\mathcal{M}})$  in a single, high capacity, nonlinear model, such as a deep neural network. During training, features are augmented by randomly “knocking out” and replacing them with constant, “placeholder” values. At inference time, using the placeholder value corresponds mathematically to estimation with the suitable marginal distribution.

Specifically, let  $\mathbf{M} = [M_1, M_2, \dots, M_N] \in \{0, 1\}^N$  denote a binary, induced missingness indicator vector. Let  $\bar{\mathbf{x}} := [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N] \in \mathbb{R}^N$  denote a vector of placeholder values. Then, define  $\mathbf{X}'(\mathbf{M}, \mathbf{X}) = \mathbf{M} \odot \bar{\mathbf{x}} + (\mathbf{1} - \mathbf{M}) \odot \mathbf{X}$  as augmented Knockout inputs, where  $\mathbf{1}$  is a vector of ones and  $\odot$  denotes element-wise multiplication. During one training iteration, a different Knockout input is used corresponding to a different randomly sampled  $\mathbf{M}$  for every data sample. The model weights are trained to minimize the loss function with respect to  $Y$ , as is done regularly.

Two mild conditions are required to ensure proper training. First, the placeholder values must be “appropriate,” as we will elaborate below. For our theoretical treatment, we will use out-of support values as appropriate; i.e.  $\bar{x}_{\mathcal{M}} \notin \text{Support}(\mathbf{X}_{\mathcal{M}})$ . Second,  $\mathbf{M}$  must be independent of  $\mathbf{X}$  and  $Y$ , i.e.  $\mathbf{M} \perp\!\!\!\perp \mathbf{X}, Y$ <sup>1</sup>. It follows straightforwardly that these two conditions lead to modeling the desired conditional and marginal distributions simultaneously. First, since  $\bar{x}_{\mathcal{M}}$  is not in the support of  $\mathbf{X}_{\mathcal{M}}$ ,

$$\mathbf{X}'_{\mathcal{M}} = \bar{x}_{\mathcal{M}} \iff \mathbf{M}_{\mathcal{M}} = \mathbf{1}, \quad \mathbf{X}'_{\mathcal{M}} \neq \bar{x}_{\mathcal{M}} \iff \mathbf{M}_{\mathcal{M}} = \mathbf{0} \text{ and } \mathbf{X}'_{\mathcal{M}} = \mathbf{X}_{\mathcal{M}}, \quad (2)$$

where  $\mathbf{0}$  and  $\mathbf{1}$  are vectors of zeros and ones of appropriate shape. Second, since  $\mathbf{M}$  is independent of  $\mathbf{X}$  and  $Y$ , it follows that imputing with the default value  $\bar{x}_{\mathcal{M}}$  is equivalent to marginalization of the missing variables defined by  $\mathcal{M}$ :

$$p(Y|\mathbf{X}'_{\mathcal{M}}=\bar{x}_{\mathcal{M}}, \mathbf{X}'_{-\mathcal{M}}=\mathbf{x}_{-\mathcal{M}}) = p(Y|\mathbf{M}_{\mathcal{M}}=\mathbf{1}, \mathbf{M}_{-\mathcal{M}}=\mathbf{0}, \mathbf{X}_{-\mathcal{M}}=\mathbf{x}_{-\mathcal{M}}) = p(Y|\mathbf{X}_{-\mathcal{M}}=\mathbf{x}_{-\mathcal{M}}). \quad (3)$$

In particular, at the two extremes, no Knockout ( $\mathbf{M} = \mathbf{0}$ ) corresponds to the original conditional distribution, and full Knockout ( $\mathbf{M} = \mathbf{1}$ ) corresponds to the full marginal:

$$p(Y|\mathbf{X}'=\mathbf{x}) = p(Y|\mathbf{M}=\mathbf{0}, \mathbf{X}=\mathbf{x}) = p(Y|\mathbf{X}=\mathbf{x}), \quad (4)$$

$$p(Y|\mathbf{X}'=\bar{\mathbf{x}}) = p(Y|\mathbf{M}=\mathbf{1}) = p(Y) \quad (5)$$

For a new test input  $\mathbf{x}$ , the prediction when  $\mathbf{x}_{\mathcal{M}}$  is missing is simply

$$\arg \max_Y p(Y|\mathbf{X}_{-\mathcal{M}}=\mathbf{x}_{-\mathcal{M}}) = \arg \max_Y p(Y|\mathbf{X}'_{\mathcal{M}}=\bar{x}_{\mathcal{M}}, \mathbf{X}'_{-\mathcal{M}}=\mathbf{x}_{-\mathcal{M}}), \quad (6)$$

i.e., the learned estimator with the augmented Knockout input.

### 2.2.1 KNOCKOUT AS AN IMPLICIT MULTI-TASK OBJECTIVE

The missingness indicator  $\mathbf{M}$  determines how inputs are replaced with appropriate placeholder values during training. To satisfy the independence condition of  $\mathbf{M}$  with  $\mathbf{X}$  and  $Y$ , the variables  $\mathbf{M}$  are sampled independently from a distribution  $p(\mathbf{M})$  during training. We show that this training strategy can be viewed as a multi-task objective (Caruana [1997]) decomposed as a weighted sum of terms, where each term is a separate marginal weighted by the distribution of  $\mathbf{M}$ . Let  $\ell$  denote the loss function to be minimized (e.g., mean-squared-error or cross-entropy loss):

$$L(\theta) = \mathbb{E}_{\mathbf{X}', Y} \ell(Y; f_{\theta}(\mathbf{X}'(\mathbf{M}, \mathbf{X}))) = \mathbb{E}_{\mathbf{X}, Y} \mathbb{E}_{\mathbf{M}} \sum_{\mathbf{m} \in \mathcal{M}} \mathbb{I}(\mathbf{M}=\mathbf{m}) \ell(Y; f_{\theta}(\mathbf{X}'(\mathbf{m}, \mathbf{X}))) \quad (7)$$

$$= \mathbb{E}_{\mathbf{X}, Y} \sum_{\mathbf{m} \in \mathcal{M}} p(\mathbf{M}=\mathbf{m}) \ell(Y; f_{\theta}(\mathbf{X}'(\mathbf{m}, \mathbf{X}))) \quad (8)$$

$$= \sum_{\mathbf{m}} p(\mathbf{M}=\mathbf{m}) \mathbb{E}_{\mathbf{X}, Y} \ell(Y; f_{\theta}(\mathbf{X}'(\mathbf{m}, \mathbf{X}))), \quad (9)$$

where  $\mathbb{I}$  is the indicator function.

If there is knowledge about the missingness patterns at inference (e.g., some  $X_i$  and  $X_j$  exhibit correlated missingness), one can design  $p(\mathbf{M})$  appropriately to cover all the expected missing patterns, i.e. by sampling  $\mathbf{m}$  during training with different weights. In the absence of such knowledge, the most general distribution for  $\mathbf{M}$  is i.i.d. Bernoulli. A common way correlated missingness arises in real-world applications is in structured inputs like latent features or images, where the entire feature vector or whole image is missing. In our experiments, we demonstrate the superiority of *structured* Knockout, over naive i.i.d. Knockout, when such correlated missingness is known a priori.

### 2.3 CHOOSING APPROPRIATE PLACEHOLDER VALUES

Our theoretical treatment assumes that the placeholder value  $\bar{x}_i$  is not in the support of  $X_i$  (see Appendix A.2 for further analysis). This is mathematically justified and works well in many cases, especially when  $X_i$  is low dimensional. However, for high dimensional inputs like vectors/images, choosing an out-of-range placeholder can be suboptimal for practical reasons such as unstable gradients and/or limited modeling capacity. In the following sections, we relax the out-of-support assumption and make some recommendations for appropriate placeholder values for various types of  $X_i$ , informed by these practical considerations.

<sup>1</sup>Note it is not necessary that  $M_i \perp\!\!\!\perp M_j$  for any  $i, j$ .

Table 1: List of different types of  $X_i$  and the recommended  $\bar{x}_i$ 

Type of $X_i$	Example	Dimension	Support	Normalized?	$\bar{x}_i$
Categorical	Gender	1	$\{1, \dots, N_{X_i}\}$	N/A	$N_{X_i} + 1$
Continuous	Test scores	1	$[a, b]$	Scale to $[0, 1]$	-1
Continuous	Temperature	1	$[a, \infty)$ or $(-\infty, b]$	Scale to $[0, \infty)$	-1
Continuous	White noise	1	$(-\infty, \infty)$	Z-score	$\pm 10$
Structured	Images	>1000	$[a, b]$	Scale to $[0, 1]$	<b>0</b>
Structured	Latent vectors	>16	$(-\infty, \infty)$	Z-score	<b>0</b>

### 2.3.1 NON-STRUCTURED

In this section, we recommend suitable placeholder values for non-structured, scalar-valued inputs.

**Categorical.** If  $X_i$  is a categorical variable with  $N_{X_i}$  integer-valued classes from 1 to  $N_{X_i}$ , then  $\bar{x}_i$  can be  $N_{X_i} + 1$ . If one-hot encoded,  $\bar{x}_i$  can be a vector of 0s.

**Continuous and Non-empty Infeasible Set.** If  $X_i$  is a continuous value within a bounded range, then we can scale the range to  $[0, 1]$  and choose  $\bar{x}_i = -1$ . More generally, if  $X_i$  has unbounded range but a non-empty infeasible set, then  $\bar{x}_i$  can be set to a value in the infeasible set. For example, if  $X_i$  only takes positive values, then we can set  $\bar{x}_i = -1$ .

**Continuous and Empty Infeasible Set.** When  $X_i$  has unbounded range and an empty infeasible set, then we suggest applying Z-score normalization and choosing  $\bar{x}_i$  such that it lies in a low probability region of the normalized  $X_i$ ,  $p(X_i = \bar{x}_i) \approx 0$ . As we argue in Appendix A.1 this approach leads to an approximation of the desired marginal.

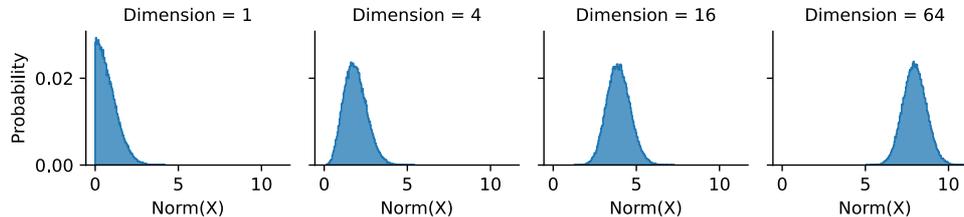


Figure 1: The region of high density of standard Gaussian shifts away from the origin as the number of dimensions increases. This motivates different choices of placeholder values at different dimensions.

Although, different distributions have different regions of low probability, we use the behavior of the Gaussian distribution as a guide to choose  $\bar{x}_i$ . Fig. 1 shows the histogram of the norm of points sampled from standard Gaussian distributions with different dimensionality. For a univariate standard Gaussian, most of the points lie close to the origin so we should choose  $\bar{x}_i$  far away from the origin. However, as the dimension increases, most of the points lie on the hyper-sphere away from the origin so we should choose  $\bar{x}_i$  to be the point at origin (i.e. a vector of zeros).

Table 1 summarizes the choices of  $\bar{x}_i$  for different types of random variables  $X_i$ .

### 2.3.2 STRUCTURED

For structured inputs like images and feature vectors, we have found that Knockout applied with an out-of-support placeholder like  $-1$ , though theoretically sound, can cause issues like unstable gradients. Therefore, we recommend an appropriate placeholder to be either the image of all 0s or the mean image. When Z-score normalization is applied, the 0 image and the mean image coincide. Theoretically, it is well known that the mean of a high dimensional random variable, such as a Gaussian, has very low probability (Vershynin, 2018) (also see Fig. 1). We believe this

Table 2: Summary of experimental setups

<i>Task</i>	<i>Type of <math>X_i</math></i>	<i>Dimension</i>	<i>Normalized?</i>	$\bar{x}_i$	$\hat{x}$
Simulations	Categorical/Continuous	1	Z-score (Cont. $X_i$ )	10	-10
Alzheimer’s Forecasting	Continuous	1	Z-score	10	-10
Privileged Information	Continuous	1	Scale to [0, 1]	-1	N/A
Tumor Segmentation	Structured (images)	256 <sup>3</sup>	Scale to [0, 1]	0	N/A
Tree Genus Classification	Structured (latent)	768 or 2048	Z-score	0	N/A
Prostate Cancer Detection	Structured (latent)	256	Z-score	0	N/A

recommendation balances the tension between ensuring an extremely-low probability placeholder with proper convergence and performance. For an empirical demonstration, see Appendix A.3.

#### 2.4 OBSERVED MISSINGNESS DURING TRAINING

The treatment above assumes complete training data, and inference-time missingness only. We now consider the situation where training data has *observed* missing inputs. Let  $N$  be the binary mask indicating the observed data missingness.  $N$  is different from  $M$ , which denotes the missingness induced by Knockout during training. Thus,  $N$  is fixed for a data sample, while  $M$  is stochastic. Observed missingness generally falls under the following scenarios (Little & Rubin 2019).

**Missing Completely at Random (MCAR):** This implies that  $N \perp\!\!\!\perp X, Y$ . Let  $M' := N \vee M$  be the augmented masking indicator, where  $\vee$  denotes the logical OR operation. Since  $N \perp\!\!\!\perp X, Y$  and  $M \perp\!\!\!\perp X, Y$ , so  $M' \perp\!\!\!\perp X, Y$ . Therefore, we can obtain the same result in Section 2.2 when using  $M'$  instead of  $M$  as the masking indicator vector. This implies that Knockout can be applied to MCAR training data simply by masking all the missing values using the same placeholders  $\bar{x}$ .

**Missing at Random (MAR) and Missing not at Random (MNAR):** This implies that  $N \not\perp\!\!\!\perp X, Y$ . Thus, we cannot replace the missing values in training data using the same placeholders. However, we can substitute these values using placeholders that are different from  $\bar{x}$  but are also outside the support of the input variables (or very unlikely values). Let the placeholders for the data missingness be  $\hat{x} \neq \bar{x}$ . During training, Knockout still randomly masks out input variables, including those that are not observed in the data. Thus, the results in Section 2.2 still hold since  $M \perp\!\!\!\perp X, Y$ .

During inference, if we know a priori that  $x_i$  of a sample is missing not at random, then we can use  $\hat{x}_i$  as the placeholder. Otherwise, if we know  $x_i$  is missing at completely random, we use  $\bar{x}_i$ .

### 3 RELATED WORK

Knockout is similar to and inspired by other methods with unrelated aims. Dropout (Srivastava et al., 2014; Gal & Ghahramani 2016) prevents overfitting by randomly dropping units (hidden and visible) during training and can be viewed as marginalizing over model parameters. During inference, marginalizing over parameters can be approximated by predicting once without dropout (Srivastava et al., 2014) or averaging multiple predictions with dropout (Gal & Ghahramani 2016). Blank-out (Maaten et al., 2013) and mDAE (Chen et al., 2014) learn to marginalize out the effects of corruption over inputs. In contrast, Knockout learns different marginals to handle different missing input patterns.

Imputation techniques impute missing inputs explicitly, for example by imputing with the mean, median, or mode. In model-based imputation, a separate model or technique first predicts the missing inputs to impute. These models include k-nearest neighbors (Troyanskaya et al., 2001), chained equations (Van Buuren & Groothuis-Oudshoorn, 2011), random forests (Stekhoven & Bühlmann, 2012), autoencoders (Gondara & Wang 2018; Ivanov et al., 2019; Lall & Robinson, 2022), GANs (Yoon et al., 2018; Li et al., 2019; Belghazi et al., 2019), or normalizing flows (Li et al., 2020). Although more accurate than simple mean/median imputation, model-based imputation incurs significant additional computation costs, especially when missing inputs are high-dimensional. In contrast, Knockout makes predictions without having to impute missing inputs explicitly. For example, some approaches (Ma et al., 2021; Peis et al., 2022) require additional training of multiple VAEs or sub-networks. Other approaches (Mattei & Frellsen, 2019; Ma et al., 2019) require training

only one VAE but they are formulated for homogeneous data (all continuous variables or all binary variables) and therefore not as flexible as Knockout. Besides, these approaches still require training two models (VAE and classifier), whereas Knockout trains only a single model (the classifier).

Another relevant line of work is causal discovery (Spirites et al., 2000), which often involves fitting a model using different subsets of available inputs and multiple distributions simultaneously (Lippe et al., 2022; James et al., 2023). To reduce computational cost, it is common to train a single model that can handle different subsets of inputs using dropout (Ke et al., 2023; Brouillard et al., 2020; Lippe et al., 2022).

Techniques like Knockout are often used in practice to train a single neural network that models multiple distributions, but are often justified empirically with little care taken in choosing placeholder values. Many works use zeros without theoretical justification (Belghazi et al., 2019; Ke et al., 2023; Brouillard et al., 2020; Lippe et al., 2022). GAIN (Yoon et al., 2018) and MisGAN (Li et al., 2019) impute using out-of-support values similar to Knockout. However, both are limited in their treatment by assuming that the supports are bounded, and do not consider categorical variables. While the approach is similar to some prior work for structural inputs (Neverova et al., 2015; Parthasarathy & Sundaram, 2020) or low-dimensional inputs (Bertsimas et al., 2024), Knockout’s theoretical backing shows that it can handle multiple data types and multiple missingness types (complete/MCAR/MAR/MNAR). Many self-supervised learning techniques can be interpreted as training to reconstruct the inputs with Knockout. In addition, Knockout can be trained with standard empirical risk minimization while some approaches need more complex optimization (Ma et al., 2021, 2022). For example, masked language modeling (Devlin et al., 2019) randomly maps tokens to an unseen “masked” token. Denoising autoencoders (Vincent et al., 2010) randomly replace image patches with black patches, which are arguably out of the support of natural images.

## 4 EXPERIMENTS

In all experiments, unless stated otherwise, we compare Knockout against a **common baseline** model trained on complete data, which, at inference time, imputes missing variables with mean (if continuous) or mode (if discrete) values. If the training is done on incomplete data with observed missing variables, imputed with mean/mode, we denote this as **common baseline\***. For most results we report a variant of Knockout but with sub-optimal placeholders (i.e. mean/mode for continuous/categorical features). We denote this variant as **Knockout\***. Note that both Knockout\* and common baseline\* use the same placeholder (mean/mode), with the only difference being that Knockout\*-trained models observe randomly knocked-out missingness *in addition to* (possible) observed missingness during training.

In all Knockout implementations, we choose random knockout rates such that, in expectation, half of the mini-batches have no induced missing variable. In batches with induced missingness, variables (or groups of variables in structured Knockout) are independently removed, with a probability equal to the knockout rate. The summary of the experimental setups are shown in Table 2.

### 4.1 SIMULATIONS

We perform simulations on both regression and classification, where the output  $Y$  needs to be predicted from some input  $X$ . In each simulation run, we sample 30k data points in total and use 10% for training. All methods use the same neural network architecture composed of a 3-layer multi-layer perceptron (MLP) with hidden layers 100 and ReLU activations. Training is done using Adam (Kingma & Ba, 2014) with learning rate  $3e-3$  for 5k steps. We restrict our focus in this section to regression results. For further experimental details and classification experiments and results, see Appendix B.1. We generate training data corresponding to complete training data, MCAR training data, and MNAR training data. For MNAR data, we adopt the self-censored missing setup where a variable is missing if its value is above the variable 90th percentile. In the regression experiments, we additionally compare against missForest (Stekhoven & Bühlmann, 2012), a competitive baseline for inference-time imputation. We also include another popular baseline (ZI) which takes zero-imputed data and a missingness indicator/mask as inputs. We tried comparing against MIRACLE (Kyono et al., 2021) but the test set size (27k) and the high number of missing patterns tested make running MIRACLE intractable.

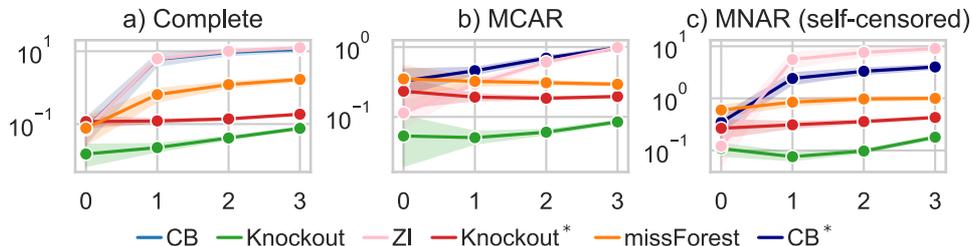


Figure 2: Test MSE evaluated against Bayes optimal prediction ( $\mathbb{E}[Y|\mathbf{X}]$ ) from 10 repetitions of the regression simulation. Lower is better. X axis indicates the number of missing variables at inference time. CB: Common baseline, ZI: zero-imputation with mask. a) Complete training data. b) Missing completely at random (MCAR) training data. c) Missing not at random (MNAR) training data

We experiment on varying the number of missing features of  $\mathbf{X} \in \mathbb{R}^9$  from 0 to 3. This resulted in 130 different missing patterns. We evaluate the models’ predictions against the MMSE-minimizing Bayes optimal predictions:  $\mathbb{E}[Y|\mathbf{X}]$ . Fig. 2 shows the results of 10 repetitions of this simulation. Both variants of Knockout outperform baselines regardless of the types of training data (complete, MCAR, or MNAR). In particular, Knockout outperforms Knockout\* in general; this underscores the importance of choosing an appropriate placeholder value.

#### 4.2 MISSING CLINICAL VARIABLES IN ALZHEIMER’S DISEASE FORECASTING

We demonstrate Knockout’s ability to manage observed missingness in a real-world clinical task: predicting the risk of progression from mild cognitive impairment (MCI) to Alzheimer’s Disease (AD) over the next five years, using data from the Alzheimer’s Disease Neuroimaging Initiative (ADNI) database (Mueller et al., 2005). Input features  $\mathbf{X}$  include subject demographic variables, genetics, cognitive assessment scores, cerebrospinal fluid (CSF) measurements, and measurements derived from magnetic resonance imaging (MRI) and positron emission tomography (PET) images. The target  $Y$  is a binary vector and indicates AD diagnosis in each of the five follow-up years. We employ the state-of-the-art model of Karaman et al. (2022). For Knockout, we use an out-of-range value of 10 for induced missingness and -10 for observed missingness, both during training and testing. Further details about the dataset and experimental setup are provided in Appendix B.2.

Figure 3 presents the average AUROC (area under the receiver operating characteristic curve) scores obtained when each input feature is missing during inference. We perform 10 random 80-20 train-test splits and calculate a Composite AUROC by averaging the AUROC scores from the five follow-up years in each split. We observe that Knockout outperforms the common baseline\* in vast majority of cases, suggesting that knocking out input during training enhances the model’s ability to handle missingness at test time. Furthermore, Knockout is largely better than Knockout\*, which underscores the importance of choosing an appropriate placeholder. We note that we present a similar analysis using the complete portion of the training dataset (i.e., with no observed missingness in training data) in Figure S5 of Appendix B.2 further demonstrating Knockout’s effectiveness.

#### 4.3 PRIVILEGED INFORMATION FOR NOISY LABEL LEARNING

In this experiment, we show that Knockout can be used for learning with *privileged information* (PI) that is available in training but absent during testing. Specifically, we evaluate this in a noisy label learning task, where the objective is to use PI, such as annotator ID or annotation time, to enhance model robustness against label noise. Due to the absence of PI in testing, existing methods (Ortiz-Jimenez et al., 2023; Wang et al., 2023) require an auxiliary classification head for PI utilization. We demonstrate that Knockout can be directly applied with a method that accepts PI as input and achieve competitive performance. We follow previous experiment setups (Wang et al., 2023) and evaluated model performance on CIFAR-10H (Peterson et al., 2019) and CIFAR-10/100N (Wei et al., 2021). These datasets involve relabeled versions of the original CIFAR. For more details, see Appendix B.6.

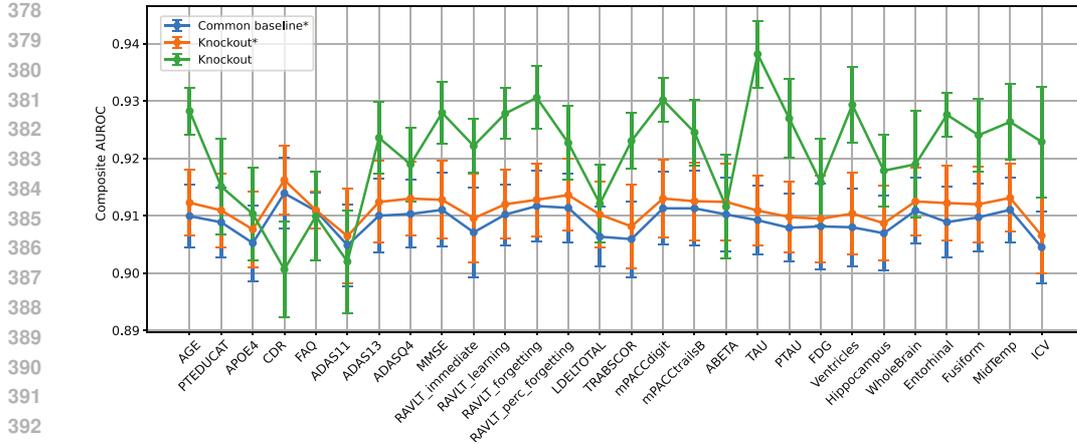


Figure 3: Composite AUROC scores obtained for the three model variants when each input feature is missing during inference (x-axis) in the Alzheimer’s Disease forecasting experiment. Displayed are averages of 10 train-test splits. Error bars indicate the standard error across these splits.

Table 3: Test accuracy of different methods on noisy label datasets with PI. We report mean and standard deviation accuracy over 5 runs. PI quality ”High” indicates a sample-wise PI is provided by the dataset. ”Low” means only batch average is provided. Best results in **bold**, second-best underlined.

Datasets	PI quality	No-PI	HET	SOP	Common baseline	Knockout
CIFAR-10H (Worst)	High	51.1±2.2	50.8±1.4	51.3±1.9	<u>55.2±0.8</u>	<b>57.4±0.6</b>
CIFAR-10N (Worst)	Low	80.6±0.2	81.9±0.4	<b>85.0±0.8</b>	82.3±0.3	<u>84.7±0.7</u>
CIFAR-100N (Fine)	Low	60.4±0.5	60.8±0.4	<u>61.9±0.6</u>	60.7±0.6	<b>62.1±0.3</b>

As a no-PI baseline, we train a Wide-ResNet-10-28 (Zagoruyko & Komodakis, 2016) model that ignores PI. We also compare against recent noisy label learning methods: HET (Collier et al., 2021) and SOP (Liu et al., 2022). We implement Knockout with a similar architecture and training scheme as the no-PI baseline, where we concatenate the PI with the image-derived features and randomly knock PI out during training. As a common baseline, we train the same architecture with complete training, but mean imputation for PI data during inference. Table 3 lists test accuracy results. For the CIFAR-10H dataset, where we have high quality PI, Knockout outperforms all baselines by a large margin, improving test accuracy by 6%. For CIFAR-10/100N datasets, where we have low quality PI during training, Knockout’s boost is more modest, performing similarly with SOP and slightly better than HET and the no-PI baseline. We conclude that Knockout can offer competitive results when we have access to high quality PI during training.

#### 4.4 MISSING IMAGES IN TUMOR SEGMENTATION

Here, we investigate the ability of Knockout to handle missingness in a high-dimensional, 3D dense image segmentation task. In particular, we experiment on a multi-modal tumor segmentation task (Baid et al., 2021), where the goal is to delineate adult brain gliomas in 3D brain MRI volumes given 4 modalities per subject: T1, T1Gd, T2, and FLAIR. We use a 3D UNet as the segmentation model (Ronneberger et al., 2015). We minimize a sum of cross-entropy loss and Dice loss with equal weighting and use Adam optimizer with a learning rate of 1e-3. See Appendix B.3 and A.3 for further details.

At inference time, we evaluate on all modality missingness patterns. Fig. 4 shows Dice scores. We observe that the Knockout-trained model has better Dice performance across all missingness

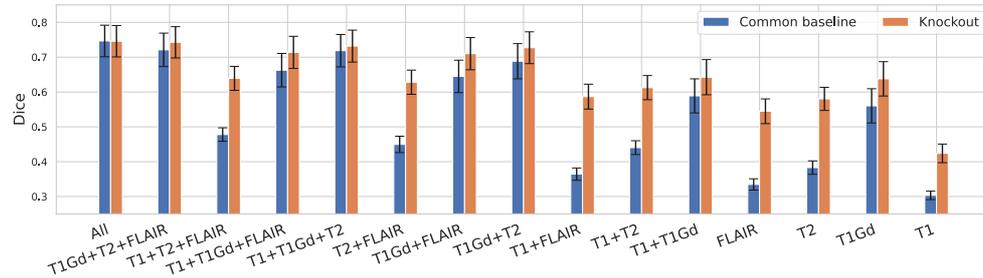


Figure 4: Dice performance of multi-modal tumor segmentation across varying missingness patterns of modality images. Knockout-trained models have better Dice performance across all missingness patterns than the common baseline. Error bars depict the 95% confidence interval over test subjects.

patterns. When all modalities are available, Knockout and the common baseline achieve the same performance level.

#### 4.5 MISSING VIEWS IN TREE GENUS CLASSIFICATION

We demonstrate Knockout’s ability to deal with missing data at the latent feature level in a classification task. The Auto Arborist dataset (Beery et al. 2022), a multi-view (street and aerial) image dataset, is used for this purpose. In this experiment, we used the top 10 genera for multi class prediction and reported results from 2 sites. A frozen ResNet-50 (He et al. 2016) and ViT-B-16 (Dosovitskiy et al. 2021) pretrained with ImageNet-v2 (Recht et al. 2019) is used as a feature extractor. The two features from street and aerial images are concatenated and were fed into 3-layer MLP with ReLU activations. We trained Knockout to randomly replace the whole latent vectors with vectors of 0s as placeholders after normalization. This variant is denoted as Knockout (Structured). We additionally trained two baselines for comparison: 1) Knockout (Features) where individual features in the latent vectors are independently replaced with placeholders, and 2) an imputation baseline, substituting latent vectors from missing views with vectors of zeros during inference. The results in Table 4 and Section S10 shows Knockout (Structured) outperforming Knockout (Features), suggesting that matching  $p(M)$  with missing patterns that we expect to see at inference can be more effective.

Table 4: F1-scores of Auto Arborist averaged over 5 random seeds (site: Columbus). Each column represents non-missing modalities at inference time. Best results in **bold**, second-best underlined.

		Aerial+Street	Aerial	Street
ResNet-50	Common baseline	0.4834 $\pm$ 0.0167	<b>0.3129</b> $\pm$ 0.0177	0.3565 $\pm$ 0.0240
	Knockout (Features)	0.4934 $\pm$ 0.0209	0.2841 $\pm$ 0.0230	<u>0.3814</u> $\pm$ 0.0221
	Knockout (Structured)	<b>0.4961</b> $\pm$ 0.0169	<u>0.3089</u> $\pm$ 0.0242	<b>0.4165</b> $\pm$ 0.0140
ViT-B-16	Common baseline	0.4649 $\pm$ 0.0183	0.3052 $\pm$ 0.0224	<u>0.3889</u> $\pm$ 0.0110
	Knockout (Features)	<u>0.4732</u> $\pm$ 0.0197	<u>0.3159</u> $\pm$ 0.0086	0.3833 $\pm$ 0.0108
	Knockout (Structured)	<b>0.4803</b> $\pm$ 0.0179	<b>0.3243</b> $\pm$ 0.0196	<b>0.4088</b> $\pm$ 0.0151

#### 4.6 MISSING MR MODALITIES IN PROSTATE CANCER DETECTION

We demonstrate structured Knockout in the context of a binary image classification task, where Knockout is applied at the latent level. The dataset consists of T2-weighted (T2w), diffusion-weighted (DWI) and apparent diffusion coefficient (ADC) MR images per subject (Saha et al. 2022). A simple “ensemble baseline” approach to address missingness is to train a separate convolutional classifier for each modality, and average the predictions of available modalities at inference time (Kim et al. 2023; Hu et al. 2020).

To train a model with latent-level structured Knockout, we use the same 3 feature extractors. Each feature extractor is trained with a different modality. The loss function is binary cross entropy loss and we use an Adam optimizer with a learning rate of  $1e-3$ . We randomly knock out each modality. In the “common baseline” approach, we trained the same architecture with complete modalities. At inference time, the latent features from missing modalities are imputed with 0s. See Appendix B.5 for more details. In the “ensemble baseline” approach, we averaged the predicted values from the three extractors without additional training.

Knockout generally outperforms the baselines in the majority of scenarios as shown in Table 5 for F1 scores and Table S11 for AUC scores, except for inputs with ADC, where the common baseline achieves the best results. Notably, the F1 scores from the popular ensemble baseline are significantly lower than Knockout.

Table 5: F1 scores of prostate cancer dataset averaged over 5 random seeds, showing performance of ensemble baseline, common baseline, and Knockout across varying missingness patterns at inference time. Each column represents non-missing modalities. Best results in **bold**, second-best underlined.

	T2	ADC	DWI	ADC +DWI	T2 +DWI	T2 +ADC	All
Ensemble	0.212 $\pm$ 0.091	0.373 $\pm$ 0.016	0.285 $\pm$ 0.032	0.327 $\pm$ 0.015	0.181 $\pm$ 0.044	0.337 $\pm$ 0.033	0.305 $\pm$ 0.050
Common	<u>0.432</u> $\pm$ 0.014	<b>0.687</b> $\pm$ 0.021	<u>0.616</u> $\pm$ 0.021	<b>0.706</b> $\pm$ 0.009	<u>0.510</u> $\pm$ 0.033	<b>0.652</b> $\pm$ 0.006	<u>0.673</u> $\pm$ 0.016
Knockout	<b>0.639</b> $\pm$ 0.023	<u>0.601</u> $\pm$ 0.019	<b>0.628</b> $\pm$ 0.025	<u>0.677</u> $\pm$ 0.016	<b>0.667</b> $\pm$ 0.010	<u>0.649</u> $\pm$ 0.023	<b>0.688</b> $\pm$ 0.014

## 5 CONCLUSION AND LIMITATIONS

We introduced Knockout, a novel, easy-to-implement strategy designed to handle missing inputs, using a mathematically principled approach. By simulating missingness during training via random “knock out” and substitution with appropriate placeholder values, our method allows a single model to learn the conditional distribution and all desired marginals. Our extensive experimental evaluation underscores the versatility and robustness of Knockout. Across diverse datasets, including both synthetic and real-world scenarios, Knockout consistently achieves competitive performance levels compared to conventional imputation and ensemble-based techniques across both low and high-dimensional missing inputs. We also extend Knockout to handle observed missing values in the training set. Our results highlight the importance of choosing the appropriate placeholder values for induced and observed missingness in training and during inference. Furthermore, we present structured version of Knockout that is more effective when entire feature vectors or input modalities might be missing.

There are several future directions for further investigation. While our paper highlights the importance of choosing an appropriate placeholder value, and there appears to be a practical tension between selecting an unlikely/infeasible value versus achieving numerical stability (e.g., avoiding exploding gradients), one can conduct a more detailed study of this to optimize the placeholder value. In our experiments, we did not compare Knockout with individual strong baseline models trained for specific missingness patterns. We considered this out of scope, as it became computationally infeasible for all the scenarios we considered. However, in practice, missingness patterns may be limited, making such an approach feasible. It remains unclear how Knockout would perform against such a strong baseline, which requires further evaluation. Another promising direction of future research is adapting Knockout to address distribution shifts in the presence of missingness. Finally, Knockout’s theoretical treatment hinges on the use of a very high capacity, non-linear model trained on very large data. In applications, where low capacity models are used and/or training data are limited, Knockout might not be as effective.

## REFERENCES

Ujjwal Baid, Satyam Ghodasara, Suyash Mohan, Michel Bilello, Evan Calabrese, Errol Colak, Keyvan Farahani, Jayashree Kalpathy-Cramer, Felipe C. Kitamura, Sarthak Pati, Luciano M.

- 540 Prevedello, Jeffrey D. Rudie, Chiharu Sako, Russell T. Shinohara, Timothy Bergquist, Rong Chai,  
541 James Eddy, Julia Elliott, Walter Reade, Thomas Schaffter, Thomas Yu, Jiabin Zheng, Ahmed W.  
542 Moawad, Luiz Otavio Coelho, Olivia McDonnell, Elka Miller, Fanny E. Moron, Mark C. Os-  
543 wood, Robert Y. Shih, Loizos Siakallis, Yulia Bronstein, James R. Mason, Anthony F. Miller,  
544 Gagandeep Choudhary, Aanchal Agarwal, Cristina H. Besada, Jamal J. Derakhshan, Mariana C.  
545 Diogo, Daniel D. Do-Dai, Luciano Farage, John L. Go, Mohiuddin Hadi, Virginia B. Hill, Michael  
546 Iv, David Joyner, Christie Lincoln, Eyal Lotan, Asako Miyakoshi, Mariana Sanchez-Montano,  
547 Jaya Nath, Xuan V. Nguyen, Manal Nicolas-Jilwan, Johanna Ortiz Jimenez, Kerem Ozturk, Bo-  
548 jan D. Petrovic, Chintan Shah, Lubdha M. Shah, Manas Sharma, Onur Simsek, Achint K. Singh,  
549 Salil Soman, Volodymyr Statsevych, Brent D. Weinberg, Robert J. Young, Ichiro Ikuta, Amit K.  
550 Agarwal, Sword C. Cambron, Richard Silbergleit, Alexandru Dusoi, Alida A. Postma, Laurent  
551 Letourneau-Guillon, Gloria J. Guzman Perez-Carrillo, Atin Saha, Neetu Soni, Greg Zaharchuk,  
552 Vahe M. Zohrabian, Yingming Chen, Milos M. Cekic, Akm Rahman, Juan E. Small, Varun Sethi,  
553 Christos Davatzikos, John Mongan, Christopher Hess, Soonmee Cha, Javier Villanueva-Meyer,  
554 John B. Freymann, Justin S. Kirby, Benedikt Wiestler, Priscila Crivellaro, Rivka R. Colen, Aika-  
555 terini Kotrotsou, Daniel Marcus, Mikhail Milchenko, Arash Nazeri, Hassan Fathallah-Shaykh,  
556 Roland Wiest, Andras Jakab, Marc-Andre Weber, Abhishek Mahajan, Bjoern Menze, Adam E.  
557 Flanders, and Spyridon Bakas. The rsna-asnr-miccai brats 2021 benchmark on brain tumor seg-  
558 mentation and radiogenomic classification, 2021.
- 559 Sara Beery, Guanhang Wu, Trevor Edwards, Filip Pavetic, Bo Majewski, Shreyasee Mukherjee,  
560 Stanley Chan, John Morgan, Vivek Rathod, and Jonathan Huang. The auto arborist dataset: a  
561 large-scale benchmark for multiview urban forest monitoring under domain shift. In *CVPR*, pp.  
562 21294–21307, 2022.
- 563 Mohamed Belghazi, Maxime Oquab, and David Lopez-Paz. Learning about an exponential amount  
564 of conditional distributions. *Advances in Neural Information Processing Systems*, 32, 2019.
- 565 Dimitris Bertsimas, Arthur Delarue, and Jean Pauphilet. Simple imputation rules for prediction with  
566 missing data: Theoretical guarantees vs. empirical performance. *TMLR*, 2024.
- 567 Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre  
568 Drouin. Differentiable causal discovery from interventional data. volume 33, pp. 21865–21877,  
569 2020.
- 570 Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.
- 571 Minmin Chen, Kilian Weinberger, Fei Sha, and Yoshua Bengio. Marginalized denoising auto-  
572 encoders for nonlinear representations. In *ICML*, pp. 1476–1484. PMLR, 2014.
- 573 Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. Correlated  
574 input-dependent label noise in large-scale image classification. In *Proceedings of the IEEE/CVF  
575 conference on computer vision and pattern recognition*, pp. 1551–1560, 2021.
- 576 Rahul S. Desikan, Florent Ségonne, Bruce Fischl, Brian T. Quinn, Bradford C. Dickerson,  
577 Deborah Blacker, Randy L. Buckner, Anders M. Dale, R. Paul Maguire, Bradley T.  
578 Hyman, Marilyn S. Albert, and Ronald J. Killiany. An automated labeling system  
579 for subdividing the human cerebral cortex on mri scans into gyral based regions of inter-  
580 est. *NeuroImage*, 31(3):968 – 980, 2006. ISSN 1053-8119. doi: DOI:10.  
581 1016/j.neuroimage.2006.01.021. URL [http://www.sciencedirect.com/science/  
582 article/B6WNP-4JFHF4P-1/2/0ec667d4c17eafb0a7c52fa3fd5aef1c](http://www.sciencedirect.com/science/article/B6WNP-4JFHF4P-1/2/0ec667d4c17eafb0a7c52fa3fd5aef1c)
- 583 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep  
584 bidirectional transformers for language understanding. In *NAACL*, pp. 4171–4186, 2019.
- 585 Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas  
586 Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An  
587 image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- 588 Bruce Fischl, André van der Kouwe, Christophe Destrieux, Eric Halgren, Florent Ségonne, David H.  
589 Salat, Evelina Busa, Larry J. Seidman, Jill Goldstein, David Kennedy, Verne Caviness, Nikos  
590 Makris, Bruce Rosen, and Anders M. Dale. Automatically Parcellating the Human Cerebral  
591

- 594       Cortex. *Cerebral Cortex*, 14(1):11–22, 2004. doi: 10.1093/cercor/bhg087. URL [http://](http://cercor.oxfordjournals.org/content/14/1/11.abstract)  
595       [cercor.oxfordjournals.org/content/14/1/11.abstract](http://cercor.oxfordjournals.org/content/14/1/11.abstract).  
596
- 597       Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model  
598       Uncertainty in Deep Learning. In *ICML*, pp. 1050–1059. PMLR, 2016.
- 599       Lovedeep Gondara and Ke Wang. Mida: Multiple imputation using denoising autoencoders. In  
600       *PAKDD*, pp. 260–272. Springer, 2018.
- 601
- 602       Elad Hazan, Roi Livni, and Yishay Mansour. Classification with low rank and missing data. In  
603       *ICML*, 2015.
- 604       Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recog-  
605       nition. In *Proceedings of CVPR*, pp. 770–778, 2016.
- 606
- 607       Dan Hu, Han Zhang, Zhengwang Wu, Fan Wang, Li Wang, J Keith Smith, Weili Lin, Gang Li,  
608       and Dinggang Shen. Disentangled-multimodal adversarial autoencoder: Application to infant age  
609       prediction with incomplete multimodal neuroimages. *IEEE TMI*, 39(12):4137–4149, 2020.
- 610       Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. How to deal with missing data in  
611       supervised deep learning? In *ICLR*, 2022.
- 612
- 613       Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary condi-  
614       tioning. In *ICLR*, 2019.
- 615       Hailey James, Chirag Nagpal, Katherine A Heller, and Berk Ustun. Participatory personalization in  
616       classification. In *NeurIPS*, 2023.
- 617
- 618       Julie Josse, Nicolas Prost, Erwan Scornet, and Gaël Varoquaux. On the consistency of supervised  
619       learning with missing values. *arXiv:1902.06931*, 2019.
- 620       Batuhan K. Karaman, Elizabeth C. Mormino, and Mert R. Sabuncu. Machine learning based multi-  
621       modal prediction of future decline toward alzheimer’s disease: An empirical study. *PLOS ONE*,  
622       17:e0277322, 11 2022. doi: 10.1371/journal.pone.0277322.
- 623
- 624       Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard  
625       Schölkopf, Michael Curtis Mozer, Christopher Pal, and Yoshua Bengio. Neural causal structure  
626       discovery from interventions. *TMLR*, 2023.
- 627       Heejong Kim, Daniel JA Margolis, Himanshu Nagar, and Mert R Sabuncu. Pulse sequence depen-  
628       dence of a simple and interpretable deep learning method for detection of clinically significant  
629       prostate cancer using multiparametric mri. *Academic Radiology*, 30(5):966–970, 2023.
- 630       Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings*  
631       *of ICLR*, 2014.
- 632
- 633       Trent Kyono, Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. Miracle: Causally-aware  
634       imputation via learning missing data mechanisms. In *NeurIPS*, 2021.
- 635       Ranjit Lall and Thomas Robinson. The midas touch: accurate and scalable missing-data imputation  
636       with deep learning. *Political Analysis*, 30(2):179–196, 2022.
- 637
- 638       Marine Le Morvan, Nicolas Prost, Julie Josse, Erwan Scornet, and Gaël Varoquaux. Linear predictor  
639       on linearly-generated data with missing values: non consistency and solutions. In *AISTATS*.  
640       PMLR.
- 641       Marine Le Morvan, Julie Josse, Thomas Moreau, Erwan Scornet, and Gaël Varoquaux. Neumiss  
642       networks: differentiable programming for supervised learning with missing values. In *NeurIPS*,  
643       2020.
- 644       Marine Le Morvan, Julie Josse, Erwan Scornet, and Gaël Varoquaux. What’s a good imputation to  
645       predict with missing values? In *NeurIPS*, 2021.
- 646
- 647       Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Misgan: Learning from incomplete data  
      with generative adversarial networks. In *ICLR*, 2019.

- 648 Yang Li, Shoaib Akbar, and Junier Oliva. Acflow: Flow models for arbitrary conditional likelihoods.  
649 In *ICML*, pp. 5831–5841. PMLR, 2020.
- 650  
651 Phillip Lippe, Taco Cohen, and Efstratios Gavves. Efficient neural causal discovery without acyclic-  
652 ity constraints. In *ICLR*, 2022.
- 653 Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John  
654 Wiley & Sons, 2019.
- 655 Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-  
656 parameterization. In *International Conference on Machine Learning*, pp. 14153–14172. PMLR,  
657 2022.
- 658  
659 Chao Ma, Sebastian Tschatschek, Konstantina Palla, Jose Miguel Hernandez-Lobato, Sebastian  
660 Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with  
661 partial vae. In *ICML*, 2019.
- 662 Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu, and Xi Peng. Smil: Multimodal  
663 learning with severely missing modality. In *AAAI*, 2021.
- 664  
665 Mengmeng Ma, Jian Ren, Long Zhao, Davide Testuggine, and Xi Peng. Are multimodal transform-  
666 ers robust to missing modality? In *CVPR*, 2022.
- 667 Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. Learning with marginalized  
668 corrupted features. In *ICML*, pp. 410–418. PMLR, 2013.
- 669  
670 Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of  
671 incomplete data sets. In *ICML*, pp. 4413–4423. PMLR, 2019.
- 672  
673 Susanne G. Mueller, Michael W. Weiner, Leon J. Thal, Ronald C. Petersen, Clifford R. Jack, William  
674 Jagust, John Q. Trojanowski, Arthur W. Toga, and Laurel Beckett. Ways toward an early diagnosis  
675 in alzheimer’s disease: The alzheimer’s disease neuroimaging initiative (adni). *Alzheimer’s &*  
676 *Dementia*, 1:55–66, 07 2005. doi: 10.1016/j.jalz.2005.06.003.
- 677  
678 Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. Moddrop: adaptive multi-  
679 modal gesture recognition. *IEEE TPAMI*, 38(8):1692–1706, 2015.
- 680  
681 Guillermo Ortiz-Jimenez, Mark Collier, Anant Nawalgaria, Alexander Nicholas D’Amour, Jesse  
682 Berent, Rodolphe Jenatton, and Efi Kokiopoulou. When does privileged information explain  
683 away label noise? In *International Conference on Machine Learning*, pp. 26646–26669. PMLR,  
684 2023.
- 685  
686 Srinivas Parthasarathy and Shiva Sundaram. Training strategies to handle missing modalities for  
687 audio-visual expression recognition. In *International Conference on Multimodal Interaction*,  
688 2020.
- 689  
690 Ignacio Peis, Chao Ma, and José Miguel Hernández-Lobato. Missing data imputation and acquisi-  
691 tion with deep hierarchical models and hamiltonian monte carlo. In *NeurIPS*, 2022.
- 692  
693 Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human  
694 uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF international  
695 conference on computer vision*, pp. 9617–9626, 2019.
- 696  
697 Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishal Shankar. Do imagenet classifiers  
698 generalize to imagenet? In *ICML*, pp. 5389–5400, 2019.
- 699  
700 Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomed-  
701 ical image segmentation. In *MICCAI*, pp. 234–241. Springer, 2015.
- 702  
703 Anindo Saha, Jasper Jonathan Twilt, Joeran Sander Bosma, Bram van Ginneken, Derya Yakar, Mat-  
704 tijns Elschot, Jeroen Veltman, Jurgen Fütterer, Maarten de Rooij, and Henkjan Huisman. The pi-cai  
705 challenge: public training and development dataset, 2022.
- 706  
707 P Spirtes, C Glymour, R Scheines, and David Heckerman. *Causation, prediction, and search*. MIT  
708 press, 2000.

- 702 Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov.  
703 Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958,  
704 2014.
- 705  
706 Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for  
707 mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.
- 708 Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani,  
709 David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays.  
710 *Bioinformatics*, 17(6):520–525, 2001.
- 711  
712 Baris Turkbey, Andrew B Rosenkrantz, Masoom A Haider, Anwar R Padhani, Geert Villeirs,  
713 Katarzyna J Macura, Clare M Tempny, Peter L Choyke, Francois Cornud, Daniel J Margolis,  
714 et al. Prostate imaging reporting and data system version 2.1: 2019 update of prostate imaging  
715 reporting and data system version 2. *European urology*, 76(3):340–351, 2019.
- 716 Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equa-  
717 tions in r. *Journal of statistical software*, 45:1–67, 2011.
- 718  
719 Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*,  
720 volume 47. Cambridge university press, 2018.
- 721 Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and  
722 Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network  
723 with a local denoising criterion. *JMLR*, 11(12), 2010.
- 724  
725 Ke Wang, Guillermo Ortiz-Jimenez, Rodolphe Jenatton, Mark Collier, Efi Kokiopoulou, and Pascal  
726 Frossard. Pi-dual: Using privileged information to distinguish clean from noisy labels. *arXiv*  
727 *preprint arXiv:2310.06600*, 2023.
- 728 Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learn-  
729 ing with noisy labels revisited: A study using real-world human annotations. *arXiv preprint*  
730 *arXiv:2110.12088*, 2021.
- 731  
732 Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative  
733 adversarial nets. In *ICML*, pp. 5689–5698. PMLR, 2018.
- 734  
735 Jiaxuan You, Xiaobai Ma, Yi Ding, Mykel J Kochenderfer, and Jure Leskovec. Handling missing  
736 data with graph representation learning. In *NeurIPS*, 2020.
- 737  
738 Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint*  
739 *arXiv:1605.07146*, 2016.
- 740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755