Knockout: A simple way to handle missing inputs

Anonymous authors
Paper under double-blind review

Abstract

Deep learning models benefit from rich (e.g., multi-modal) input features. However, multi-modal models might be challenging to deploy, because some inputs may be missing at inference. Current popular solutions include marginalization, imputation, and training multiple models. Marginalization achieves calibrated predictions, but it is computationally expensive and only feasible for low dimensional inputs. Imputation may result in inaccurate predictions, particularly when high-dimensional data, such as images, are missing. Training multiple models, where each model is designed to handle different subsets of inputs, can work well but requires 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 (ML) and statistics, not all variables might be available for every data point. This issue, also 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 solved by marginalization, 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.

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 features (Le Morvan et al., 2020b). This can be seen as approximating the marginalization with a delta function. More sophisticated imputation methods use EM (Josse et al., 2019) or neural-based networks (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., 2020a; 2021), assume linearity of predictors (Le Morvan et al., 2020b), 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 randomly "knocked out" and replaced by constant "placeholder" values. At inference time, using the placeholder value corresponds mathematically to estimation with the appropriate marginal distribution. 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*. We demonstrate the broad applicability of Knockout in a suite of experiments using synthetic and real-world data from multiple modalities (image-based and tabular). Real world experiments include Alzheimer's forecasting, noisy label learning, multi-modal MR image segmentation, multi-modal image classification, and multi-view tree genus classification. We show the effectiveness of Knockout in handling low and high-dimensional missing inputs compared against competitive baselines.

2 Method

2.1 Background

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

A simple approach is to train a separate model for $p(Y|\boldsymbol{X}_{-\mathcal{M}})$, i.e. a model that takes only the non-missing features $\boldsymbol{X}_{-\mathcal{M}}$ 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 rewrites $p(Y|\boldsymbol{X}_{-\mathcal{M}})$ using the already available $p(Y|\boldsymbol{X})$:

$$p(Y|\boldsymbol{X}_{-\mathcal{M}}) = \int p(Y, \boldsymbol{X}_{\mathcal{M}}|\boldsymbol{X}_{-\mathcal{M}}) d\boldsymbol{X}_{\mathcal{M}} = \int p(Y|\boldsymbol{X}) p(\boldsymbol{X}_{\mathcal{M}}|\boldsymbol{X}_{-\mathcal{M}}) d\boldsymbol{X}_{\mathcal{M}}. \tag{1}$$

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

More sophisticated imputation methods capture the interdependencies between inputs (Troyanskaya et al., 2001; Stekhoven & Bühlmann, 2012), for example by explicitly modeling $p(\boldsymbol{X}_{\mathcal{M}}|\boldsymbol{X}_{-\mathcal{M}})$ by training a separate model. The point estimate $\boldsymbol{x}_{\mathcal{M}} = \arg\max_{\boldsymbol{X}_{\mathcal{M}}} p(\boldsymbol{X}_{\mathcal{M}}|\boldsymbol{X}_{-\mathcal{M}})$ can be used at inference time for the missing $\boldsymbol{X}_{\mathcal{M}}$. While properly accounting for interdependencies between inputs, this approach requires fitting a separate model for $p(\boldsymbol{X}_{\mathcal{M}}|\boldsymbol{X}_{-\mathcal{M}})$. In multiple imputation, multiple samples from $p(\boldsymbol{X}_{\mathcal{M}}|\boldsymbol{X}_{-\mathcal{M}})$ are drawn and a Monte Carlo approximation is used to estimate the integral on the RHS of Eq. 1 (Mattei & Frellsen, 2019; Kyono et al., 2021; Ipsen et al., 2022). 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 $p(Y|\boldsymbol{X})$ and all desired marginals $p(Y|\boldsymbol{X}_{-\mathcal{M}})$ 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 $\boldsymbol{M} = [M_1, M_2, \ldots, M_N] \in \{0,1\}^N$ denote a binary, induced missingness indicator vector. Let $\bar{\boldsymbol{x}} := [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N] \in \mathbb{R}^N$ denote a vector of placeholder values. Then, define $\boldsymbol{X}'(\boldsymbol{M}, \boldsymbol{X}) = \boldsymbol{M} \odot \bar{\boldsymbol{x}} + (\boldsymbol{1} - \boldsymbol{M}) \odot \boldsymbol{X}$ as augmented Knockout inputs, where $\boldsymbol{1}$ 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 $\boldsymbol{M}$ 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. $\bar{\boldsymbol{x}}_{\mathcal{M}} \notin \text{Support}(\boldsymbol{X}_{\mathcal{M}})$. Second, $\boldsymbol{M}$ must be independent of $\boldsymbol{X}$ and Y, i.e. $\boldsymbol{M} \perp\!\!\!\perp \boldsymbol{X}, Y$.[1] It follows straightforwardly that these two conditions lead to modeling the desired conditional and marginal

[1]Note it is not necessary that $M_i \perp\!\!\!\perp M_j$ for any i, j.

distributions simultaneously. First, since $\bar{\boldsymbol{x}}_{\mathcal{M}}$ is not in the support of $\boldsymbol{X}_{\mathcal{M}}$,

$$\boldsymbol{X}'_{\mathcal{M}} = \bar{\boldsymbol{x}}_{\mathcal{M}} \Leftrightarrow \boldsymbol{M}_{\mathcal{M}} = \mathbf{1}, \qquad \boldsymbol{X}'_{\mathcal{M}} \neq \bar{\boldsymbol{x}}_{\mathcal{M}} \Leftrightarrow \boldsymbol{M}_{\mathcal{M}} = \mathbf{0} \text{ and } \boldsymbol{X}'_{\mathcal{M}} = \boldsymbol{X}_{\mathcal{M}}, \tag{2}$$

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

$$p(Y|\boldsymbol{X}'_{\mathcal{M}}{=}\bar{\boldsymbol{x}}_{\mathcal{M}}, \boldsymbol{X}'_{-\mathcal{M}}{=}\boldsymbol{x}_{-\mathcal{M}}) = p(Y|\boldsymbol{M}_{\mathcal{M}}{=}\mathbf{1}, \boldsymbol{M}_{-\mathcal{M}}{=}\mathbf{0}, \boldsymbol{X}_{-\mathcal{M}}{=}\boldsymbol{x}_{-\mathcal{M}}) \tag{3}$$
$$= p(Y|\boldsymbol{X}_{-\mathcal{M}}{=}\boldsymbol{x}_{-\mathcal{M}}). \tag{4}$$

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

$$p(Y|\boldsymbol{X}'{=}\boldsymbol{x}) = p(Y|\boldsymbol{M}{=}\mathbf{0}, \boldsymbol{X}{=}\boldsymbol{x}) = p(Y|\boldsymbol{X}{=}\boldsymbol{x}), \tag{5}$$
$$p(Y|\boldsymbol{X}'{=}\bar{\boldsymbol{x}}) = p(Y|\boldsymbol{M}{=}\mathbf{1}) = p(Y). \tag{6}$$

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

$$\arg\max_{Y} p(Y|\boldsymbol{X}_{-\mathcal{M}}{=}\boldsymbol{x}_{-\mathcal{M}}) = \arg\max_{Y} p(Y|\boldsymbol{X}'_{\mathcal{M}}{=}\bar{\boldsymbol{x}}_{\mathcal{M}}, \boldsymbol{X}'_{-\mathcal{M}}{=}\boldsymbol{x}_{-\mathcal{M}}), \tag{7}$$

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

2.2.1 Knockout as an Implicit Multi-task Objective

The missingness indicator $\boldsymbol{M}$ determines how inputs are replaced with appropriate placeholder values during training. To satisfy the independence condition of $\boldsymbol{M}$ with $\boldsymbol{X}$ and $\boldsymbol{Y}$, the variables $\boldsymbol{M}$ are sampled independently from a distribution $p(\boldsymbol{M})$ 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 $\boldsymbol{M}$. Let ℓ denote the loss function to be minimized (e.g., mean-squared-error or cross-entropy loss):

$$L(\theta) = \mathbb{E}_{\boldsymbol{X}',Y}\ \ell(Y; f_\theta(\boldsymbol{X}'(\boldsymbol{M}, \boldsymbol{X})) \tag{8}$$
$$= \mathbb{E}_{\boldsymbol{X},Y}\mathbb{E}_{\boldsymbol{M}} \sum_{\boldsymbol{m}\in\boldsymbol{M}} \mathbb{I}(\boldsymbol{M}{=}\boldsymbol{m})\ \ell(Y; f_\theta(\boldsymbol{X}'(\boldsymbol{M}, \boldsymbol{X}))) \tag{9}$$
$$= \mathbb{E}_{\boldsymbol{X},Y} \sum_{\boldsymbol{m}\in\boldsymbol{M}} p(\boldsymbol{M}{=}\boldsymbol{m})\ \ell(Y; f_\theta(\boldsymbol{X}'(\boldsymbol{m}, \boldsymbol{X}))) \tag{10}$$
$$= \sum_{\boldsymbol{m}} p(\boldsymbol{M}{=}\boldsymbol{m})\ \mathbb{E}_{\boldsymbol{X},Y}\ell(Y; f_\theta(\boldsymbol{X}'(\boldsymbol{m}, \boldsymbol{X}))), \tag{11}$$

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 $p(\boldsymbol{M})$ appropriately to cover all the expected missing patterns, i.e. by sampling $\boldsymbol{m}$ during training with different weights. In the absence of such knowledge, the most general distribution for $\boldsymbol{M}$ 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 analysis assumes that the placeholder value $\bar{x}_i$ lies outside the support of X_i (see Appendix A.2 for further discussion). Previous work by Josse et al. (2019) and Bertsimas et al. (2024) has

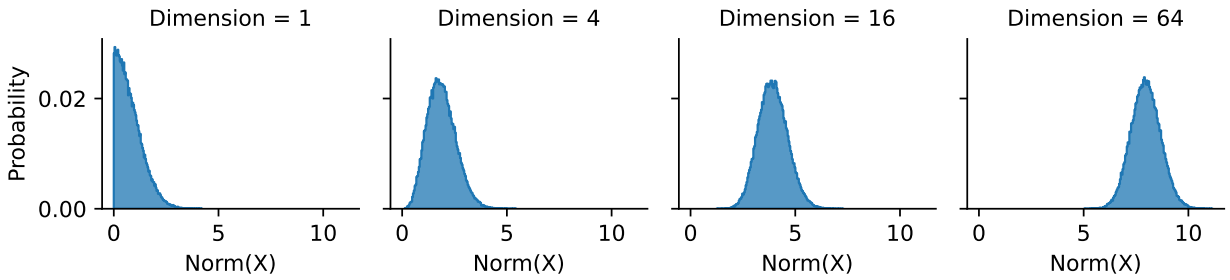

Figure 1: The region of high density of standard Gaussian shifts away from the origin as the number of dimensions increases. This motivates different choices of placeholder values at different dimensions.

shown that, with infinite data, mean imputation (for continuous variables) and out-of-support imputation (for both continuous and discrete variables) can be optimal. However, in practice, data is finite, and empirical performance becomes more important. In such cases, using out-of-support values as placeholders is often preferable and can be mathematically justified—particularly when X_i is low-dimensional. That said, for high-dimensional inputs such as vectors or images, out-of-range placeholders may lead to practical issues like unstable gradients or limited model capacity, making them less effective. In the sections that follow, we relax the out-of-support assumption and offer practical recommendations for choosing placeholder values for different types of X_i, based on these considerations.

2.3.1 Non-structured

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

- **Categorical variable**: $\bar{x}_i$ can be $N_{X_i}+1$ if X_i are integer-valued classes from 1 to N_{X_i}. If one-hot encoded, $\bar{x}_i$ can be a vector of 0s.
- **Continuous variable and Non-empty Infeasible Set**: we can scale X_i 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 variable and Empty Infeasible Set**: 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 such that $p(X_i{=}\bar{x}_i) \approx 0$. As we argue in Appendix A.1, this approach leads to an approximation of the desired marginal.

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 found that Knockout applied with an out-of-support placeholder like $\mathbf{-1}$, 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)

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

Type of X_i	*Example*	*Dimension*	*Support*	*Normalized?*	$\bar{\boldsymbol{x}}_i$
Categorical	Gender	1	$\{1, \ldots, N_{X_i}\}$	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	± 10
Structured	Images	>1000	$[a, b]$	Scale to [0, 1]	**0**
Structured	Latent vectors	>16	$(-\infty, \infty)$	Z-score	**0**

Table 2: Summary of experimental setups

Task	*Type of* X_i	*Dimension*	*Normalized?*	$\bar{\boldsymbol{x}}_i$	$\dot{\boldsymbol{x}}$
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)	256^3	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
Food Classification	Structured (latent)	768	Z-score	**0**	N/A

(also see Fig. 1). We believe this 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 $\boldsymbol{N}$ be the binary mask indicating the observed data missingness. $\boldsymbol{N}$ is different from $\boldsymbol{M}$, which denotes the missingness induced by Knockout during training. Thus, $\boldsymbol{N}$ is fixed for a data sample, while $\boldsymbol{M}$ is stochastic. Observed missingness generally falls under the following scenarios (Little & Rubin, 2019).

Missing Completely at Random (MCAR): This implies that $\boldsymbol{N} \perp\!\!\!\perp \boldsymbol{X}, Y$. Let $\boldsymbol{M}' := \boldsymbol{N} \vee \boldsymbol{M}$ be the augmented masking indicator, where $\vee$ denotes the logical OR operation. Since $\boldsymbol{N} \perp\!\!\!\perp \boldsymbol{X}, Y$ and $\boldsymbol{M} \perp\!\!\!\perp \boldsymbol{X}, Y$, so $\boldsymbol{M}' \perp\!\!\!\perp \boldsymbol{X}, Y$. Therefore, we can obtain the same result in Section 2.2 when using $\boldsymbol{M}'$ instead of $\boldsymbol{M}$ 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 $\bar{\boldsymbol{x}}$.

Missing at Random (MAR) and Missing not at Random (MNAR): This implies that $\boldsymbol{N} \not\perp\!\!\!\perp \boldsymbol{X}, Y$. 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 $\bar{\boldsymbol{x}}$ but are also outside the support of the input variables (or very unlikely values). Let the placeholders for the data missingness be $\dot{\boldsymbol{x}} \neq \bar{\boldsymbol{x}}$. 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 $\boldsymbol{M} \perp\!\!\!\perp \boldsymbol{X}, Y$.

During inference, if we know a priori that x_i of a sample is missing not at random, then we can use $\dot{x}_i$ as the placeholder. Otherwise, if we know x_i is missing completely at random, we use $\bar{x}_i$. Unless stated otherwise, Knockout always uses a different placeholder for MAR and MNAR variables.

3 Related Work

Knockout is inspired by other methods with different objectives. Dropout (Srivastava et al., 2014; Gal & Ghahramani, 2016) prevents overfitting by randomly dropping units (hidden and visible) during training,

effectively marginalizing over model parameters. During inference, this marginalization can be approximated by predicting once without dropout (Srivastava et al., 2014) or averaging multiple predictions with dropout (Gal & Ghahramani, 2016). Blankout (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 varied missing input patterns.

Imputation techniques impute missing inputs explicitly, e.g., using mean, median, or mode values. Model-based imputation methods predict missing values via 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 imputation, model-based imputation incurs significant additional computation costs, especially with high-dimensional inputs. 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 assume homogeneous data (all continuous variables or all binary variables), limiting flexibility. In contrast, Knockout uses a single classifier, avoiding the need for separate models to impute missing inputs explicitly.

Another relevant line of work is causal discovery (Spirtes 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). Similarly, methods like Selective MIM (Van Ness et al., 2023) have been proposed to enhance model performance in high-dimensional settings by selectively encoding informative missing patterns. 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. If d is the number of input variables and r is the knockout rate, we choose r such that the probability of not knocking out any variables in a mini-batch, i.e. $(1-r)^d$ is approximately 0.5. In batches with induced missingness, variables (or groups of variables in structured

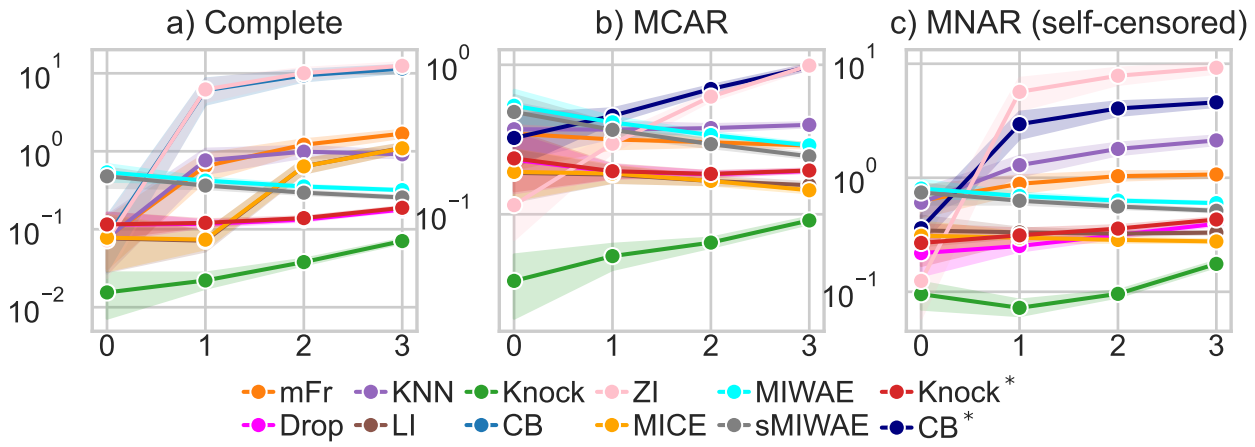

Figure 2: Test MSE evaluated against Bayes optimal prediction ($\mathbb{E}[Y|\boldsymbol{X}]$) averaged over 10 repetitions (lower is better). X axis indicates the number of missing variables at inference time. mFr: missForest, Drop: dropout, Knock: Knockout, CB: Common baseline, ZI: zero-imputation with mask, sMIWAE: supMIWAE. KNN: k-nearest-neighbor imputation, LI: linear regression imputation

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 regression simulations to predict the output Y from the input $\boldsymbol{X} \in \mathbb{R}^9$. In each simulation, we sample 30k data points in total and use 10% for training. Since the input is low-dimensional, we can additionally compare against sophisticated but computationally expensive imputation methods. The baselines for the simulation are:

- Inpute Dropout (Srivastava et al., 2014): Randomly sets a subset of input features to zero during training.
- Zero Impute with Indicator: Replaces missing values with zeros and adds a binary indicator variable to denote the presence of missing data.
- MICE (Van Buuren & Groothuis-Oudshoorn, 2011): Iteratively imputes missing values by modeling each variable with missing data as a function of other variables in a round-robin fashion.
- missForest (Stekhoven & Bühlmann, 2012): Uses random forests to iteratively predict and fill in missing values.
- MIWAE Mattei & Frellsen (2019): Imputes missing data using a VAE trained with importance-weighted bounds.
- supMIWAE (Ipsen et al., 2022): A supervised extension of MIWAE.
- KNN Impute: Fills in missing values by averaging the values of the k-nearest neighbors.
- Linear Regression Impute: Imputes missing values using predictions from a linear regression model.

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 generate training data corresponding to complete training data, MCAR training data, and MNAR training data. For MCAR data, each feature is independently missing with a probability of 10%. For MNAR data, we adopt the self-censored missing setup where a variable is missing if its value is above the variable 90th percentile. To ensure that performance comparisons reflect only differences in the training data, we keep

the test data consistent across experiments. Specifically, during testing, we apply a predefined missingness pattern (i.e., a specific set of missing variables) uniformly to all samples in the test set. We repeat this procedure exhaustively for all missingness patterns involving up to 3 missing variables. This resulted in 130 different missing patterns.

We evaluate the models' predictions against the MMSE-minimizing Bayes optimal predictions: $\mathbb{E}[Y|\boldsymbol{X}]$ (please see Appendix B.1.1 for evaluation against observations Y). Fig. 2 shows the results of 10 repetitions of this simulation. Knockout outperforms all the baselines regardless of the types of training data (complete, MCAR, or MNAR). Dropout and Knockout* achieve similar performance but both are worse than Knockout; this underscores the importance of choosing an appropriate placeholder value. We also experimented with varying the training-test split size to simulate low-data and data-rich regimes in Appendix B.1.1.

In addition, we perform an ablation to verify that choosing appropriate placeholders is critical for getting good performance (see Fig. 3). In this ablation, the value of the placeholder of Knockout is selected from the set {0, 2, 4, 6, 8, 10}. After feature normalization, the mean value of a feature is 0 and using the placeholder value of 0 resulting in bad prediction performance (higher MSE). Choosing placeholder value that is away from 0 (i.e. away from the feature mean) lowers the test MSE in both complete data (Fig. 3a) and missing data (Fig. 3b) setup. We also include an comparison aganist the Common baseline with $\bar{x}$

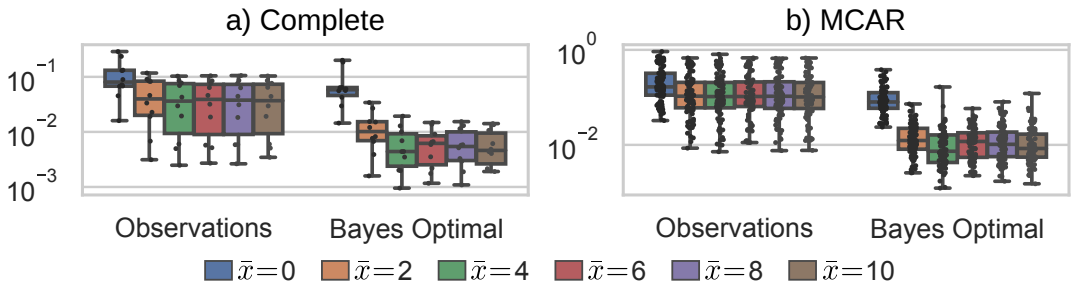

Figure 3: Lower test MSE in the regression simulation is achieved by selecting placeholder value that is far from the mean (i.e. 0). Observations: evaluate against Y. Bayes Optimal: evaluate against $\mathbb{E}[Y|\boldsymbol{X}]$.

imputation in Appendix B.1.2 to illustrate the benefit of Knockout training strategy. This baseline is similar to Knockout during inference but does not employ the augmentation strategy of Knockout during training.

To support our proposal for handling NMAR data, we include an ablation study comparing Knockout against its ablated version named Knockout-. While Knockout follows the procedure outlined in Section 2.4, Knockout- treats all observed data uniformly, without distinguishing NMAR observations. Both methods were trained using the same MNAR data detailed previously. The results of this comparison are presented in Table 3, which reports the MSE across varying levels of missingness. Knockout consistently achieves lower MSEs, demonstrating the benefit of explicitly accounting for NMAR mechanisms.

Method	**fm = 0**	**fm = 1**	**fm = 2**	**fm = 3**
CB	0.361 ± 0.183	2.953 ± 3.870	4.053 ± 5.616	4.585 ± 6.109
Knockout-	0.144 ± 0.039	0.089 ± 0.055	0.107 ± 0.071	0.163 ± 0.141
Knockout	0.095 ± 0.044	0.072 ± 0.058	0.095 ± 0.062	0.175 ± 0.156

Table 3: Knockout's vs. Knockout-'s MSE across varying numbers of missing features (**fm**).

4.2 Missing Clinical Variables in Alzheimer's Disease Forecasting

We demonstrate Knockout's ability to deal with observed missingness in a real-world clinical prediction task. Specifically, the goal is to predict progression from mild cognitive impairment to Alzheimer's Disease in the

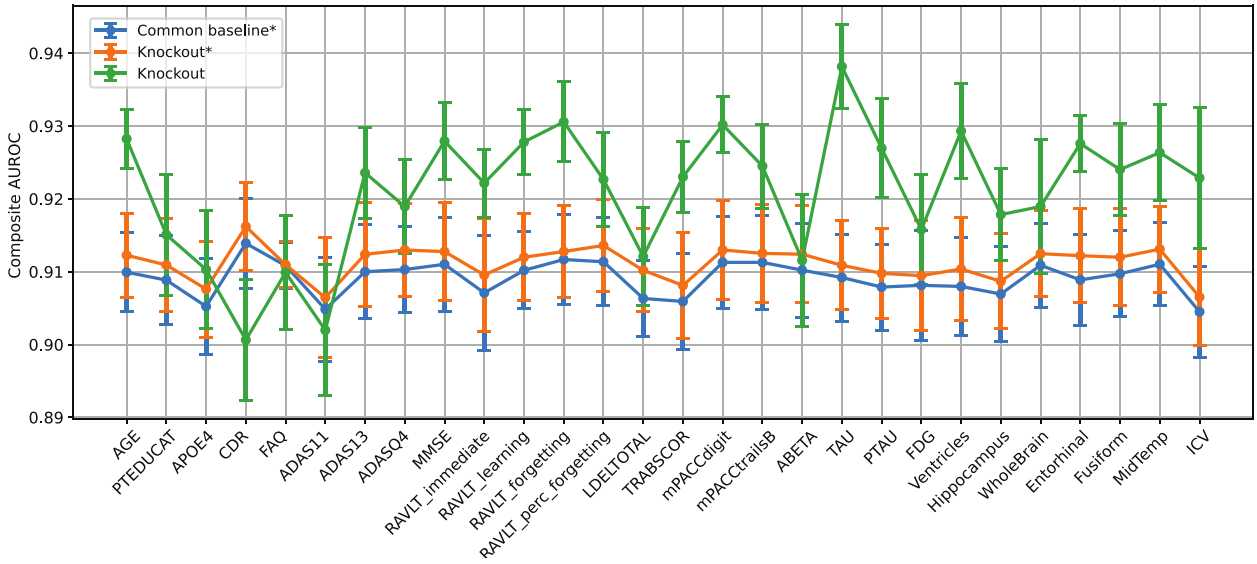

Figure 4: Composite AUROC scores obtained for the three model variants when each input feature is missing during inference (x-axis) in the Alzheimer's Disease forecasting experiment. Error bars indicate the standard error across 10 train-test splits.

next five years. We use data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (Mueller et al., 2005) and adopt the state-of-the-art model from (Karaman et al., 2022). Input features $\boldsymbol{X}$ include subject demographics, genetics, and clinical/imaging biomarkers. The target Y is a binary vector and indicates AD diagnosis in each of the five follow-up years. 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. Fig. 4 shows the composite AUROC scores when individual input features are missing during inference, averaged across 10 random train-test splits. Knockout outperforms both the common baseline and Knockout* in the vast majority of cases.

4.3 Privileged Information for Noisy Label Learning

Knockout can be used for learning with *privileged information* (PI) that is available only during training. 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. 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.5. 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 4 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. Knockout can offer competitive results when we have access to high quality PI during training.

Table 4: Average test accuracy (over 5 runs) of different methods on noisy label datasets with PI. For Quality, "High" and "Low" indicate access to sample-wise PI and batch-average PI respectively. Best results in **bold**, second-best underlined.

Datasets	Quality	No-PI	HET	SOP	Common baseline	Knockout
CIFAR-10H (Worst)	High	$51.1_{\pm 2.2}$	$50.8_{\pm 1.4}$	$51.3_{\pm 1.9}$	$55.2_{\pm 0.8}$	$\mathbf{57.4}_{\pm \mathbf{0.6}}$
CIFAR-10N (Worst)	Low	$80.6_{\pm 0.2}$	$81.9_{\pm 0.4}$	$\mathbf{85.0}_{\pm \mathbf{0.8}}$	$82.3_{\pm 0.3}$	$84.7_{\pm 0.7}$
CIFAR-100N (Fine)	Low	$60.4_{\pm 0.5}$	$60.8_{\pm 0.4}$	$61.9_{\pm 0.6}$	$60.7_{\pm 0.6}$	$\mathbf{62.1}_{\pm \mathbf{0.3}}$

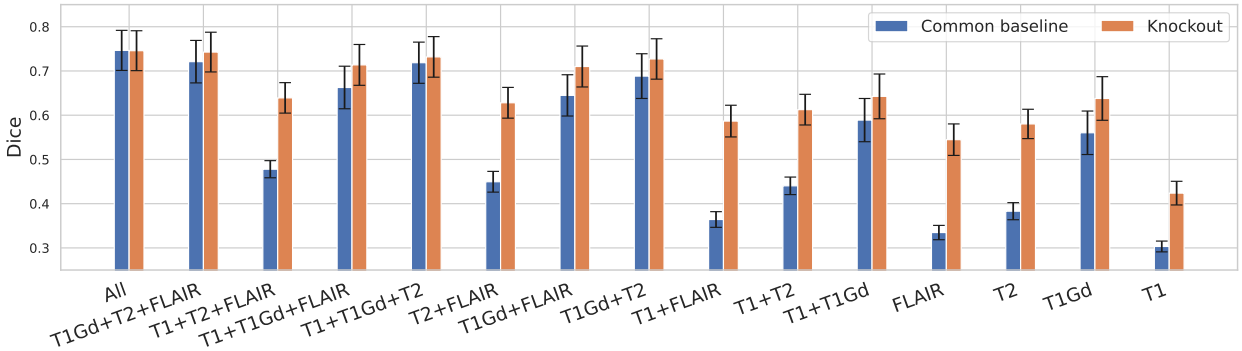

Figure 5: Dice performance of multi-modal tumor segmentation across varying missingness patterns of modality images. Knockout trained models have better Dice performance across all missingness patterns than the common baseline. Error bars depict the 95% confidence interval over test subjects.

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. 5 shows Dice scores. We observe that the Knockout-trained model has better Dice performance across all missingness patterns. When all modalities are available, Knockout and the common baseline achieve the same performance level. See Appendix A.3 for additional experiments on the effect of various placeholders.

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. 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. Table 5 shows Knockout (Structured) outperforming Knockout (Features), suggesting that matching $p(\boldsymbol{M})$ with missing patterns that we expect to see at inference can be more effective.

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

		Aerial+Street	Aerial	Street
ResNet-50	Common baseline	$0.483_{\pm 0.016}$	$\mathbf{0.312}_{\pm 0.017}$	$0.356_{\pm 0.024}$
	Knockout (Features)	$\underline{0.493}_{\pm 0.020}$	$0.284_{\pm 0.023}$	$\underline{0.381}_{\pm 0.022}$
	Knockout (Structured)	$\mathbf{0.496}_{\pm 0.016}$	$\underline{0.308}_{\pm 0.024}$	$\mathbf{0.416}_{\pm 0.014}$
ViT-B-16	Common baseline	$0.464_{\pm 0.018}$	$0.305_{\pm 0.022}$	$\underline{0.388}_{\pm 0.011}$
	Knockout (Features)	$\underline{0.473}_{\pm 0.019}$	$\underline{0.315}_{\pm 0.008}$	$0.383_{\pm 0.010}$
	Knockout (Structured)	$\mathbf{0.480}_{\pm 0.017}$	$\mathbf{0.324}_{\pm 0.019}$	$\mathbf{0.408}_{\pm 0.015}$

Table 6: F1 scores of prostate cancer dataset averaged over 5 random seeds 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	$0.21_{\pm 0.09}$	$0.37_{\pm 0.01}$	$0.28_{\pm 0.03}$	$0.32_{\pm 0.01}$	$0.18_{\pm 0.04}$	$0.33_{\pm 0.03}$	$0.30_{\pm 0.05}$
Common	$\underline{0.43}_{\pm 0.01}$	$\mathbf{0.68}_{\pm 0.02}$	$\underline{0.61}_{\pm 0.02}$	$\mathbf{0.70}_{\pm 0.01}$	$\underline{0.51}_{\pm 0.03}$	$\mathbf{0.65}_{\pm 0.01}$	$\underline{0.67}_{\pm 0.01}$
Knockout	$\mathbf{0.63}_{\pm 0.02}$	$\underline{0.60}_{\pm 0.02}$	$\mathbf{0.62}_{\pm 0.02}$	$\underline{0.67}_{\pm 0.01}$	$\mathbf{0.66}_{\pm 0.01}$	$\underline{0.65}_{\pm 0.02}$	$\mathbf{0.68}_{\pm 0.01}$

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). For further details, please see Appendix B.4. 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. In the "ensemble baseline" approach, we averaged the predicted values from the three extractors without additional training. As shown in Table 6, Knockout generally outperforms the baselines in the majority of scenarios, except for inputs with ADC. Notably, the F1 scores from the popular ensemble baseline are significantly lower than Knockout.

4.7 Missing Modalities in UPMC Food-101 Classification

To further evaluate the generalizability of our approach, we extend our experiments to the UPMC Food-101 dataset (Wang et al., 2015), a widely used benchmark for multi-modal learning with both textual and visual inputs. Our multi-modal architecture combines pretrained ResNet50 (vision) and RoBERTa-base (text) backbones. The output from ResNet50 is fed into a single linear layer to transform its dimension to match that of Roberta's output. Feature vectors from each modality are concatenated and passed through an MLP with two hidden layers of size 256 and ReLU activations. Models are trained using AdamW with a learning rate of 3×10^{-5}, for 8 epochs and a batch size of 72. We follow the standard train-test split, reserving 7,000 samples from the training set for validation.

As shown in Table 7, our multi-modal model with Knockout achieves 92.7% accuracy, outperforming the Common Baseline (91.5%) and matching the previous state-of-the-art reported by (Gallo et al., 2020). Table 7 also reports performance under unimodal ablations. Knockout exhibits greater robustness, especially in the image-only setting, improving accuracy from 47.5% to 63.2%.

Table 7: Test accuracy (%) on the Food-101 dataset for multi-modal models under three input settings: (1) both Image and Text, (2) Image only (without Text), and (3) Text only (without Image). For comparison, test accuracies of multi-modal models from previous work are also included.

Method	Image Model	Text Model	Image + Text	w/o Text	w/o Image
Wang et al. (2015)	VGG19	TF-IDF	85.1	N/A	N/A
Kiela et al. (2018)	ResNet152	FastText	90.8	N/A	N/A
Gallo et al. (2020)	Inception-v3	BERT-LSTM	92.5	N/A	N/A
Common baseline	ResNet50	RoBERTa	91.5	47.5	85.0
Knockout	ResNet50	RoBERTa	92.7	63.2	85.3

5 Discussion

We introduced Knockout, a novel, easy-to-implement strategy for handling 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 evaluation demonstrates Knockout's the versatility and robustness across diverse synthetic and real-world scenarios, consistently outperforming conventional imputation and ensemble techniques for both low and high-dimensional missing inputs. We also extend Knockout to handle observed missing values in training and present a structured version that is more effective when entire feature vectors or input modalities are missing.

Although modern multimodal models are increasingly tolerant of missing modalities, Knockout can still play a crucial role in promoting robustness and adaptability. This is especially useful during deployment, where models often face unpredictable or incomplete inputs due to sensor failures or missing data. Multimodal foundation models like LLaVA (Liu et al., 2023) and VILA (Lin et al., 2024) are generally more robust to missing modalities than earlier models, but they are not immune to failure. Their ability to handle missing inputs largely stems from pretraining on massive, diverse datasets. However, when applied to tasks with input distributions that differ significantly from their training data, these models may still struggle in the absence of a modality. Even SOTA transformer-based models have been shown to be sensitive to missing modalities (Ma et al., 2022).

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. Additionally, our method assumes model training with SGD, where samples are seen multiple times. Extending it to models that do not use SGD, such as XGBoost, is a valuable avenue. 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 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. Prevedello, Jeffrey D. Rudie, Chiharu Sako, Russell T. Shinohara, Timothy Bergquist, Rong Chai, James Eddy, Julia Elliott, Walter Reade, Thomas Schaffter, Thomas Yu, Jiaxin Zheng, Ahmed W. Moawad, Luiz Otavio Coelho, Olivia McDonnell, Elka Miller, Fanny E. Moron, Mark C. Oswood, Robert Y. Shih, Loizos Siakallis, Yulia Bronstein, James R. Mason, Anthony F. Miller, Gagandeep Choudhary, Aanchal Agarwal, Cristina H. Besada, Jamal J. Derakhshan, Mariana C. Diogo, Daniel D. Do-Dai, Luciano Farage, John L. Go, Mohiuddin Hadi, Virginia B. Hill, Michael Iv, David Joyner, Christie Lincoln, Eyal Lotan, Asako Miyakoshi, Mariana

Sanchez-Montano, Jaya Nath, Xuan V. Nguyen, Manal Nicolas-Jilwan, Johanna Ortiz Jimenez, Kerem Ozturk, Bojan D. Petrovic, Chintan Shah, Lubdha M. Shah, Manas Sharma, Onur Simsek, Achint K. Singh, Salil Soman, Volodymyr Statsevych, Brent D. Weinberg, Robert J. Young, Ichiro Ikuta, Amit K. Agarwal, Sword C. Cambron, Richard Silbergleit, Alexandru Dusoi, Alida A. Postma, Laurent Letourneau-Guillon, Gloria J. Guzman Perez-Carrillo, Atin Saha, Neetu Soni, Greg Zaharchuk, Vahe M. Zohrabian, Yingming Chen, Milos M. Cekic, Akm Rahman, Juan E. Small, Varun Sethi, Christos Davatzikos, John Mongan, Christopher Hess, Soonmee Cha, Javier Villanueva-Meyer, John B. Freymann, Justin S. Kirby, Benedikt Wiestler, Priscila Crivellaro, Rivka R. Colen, Aikaterini Kotrotsou, Daniel Marcus, Mikhail Milchenko, Arash Nazeri, Hassan Fathallah-Shaykh, Roland Wiest, Andras Jakab, Marc-Andre Weber, Abhishek Mahajan, Bjoern Menze, Adam E. Flanders, and Spyridon Bakas. The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification, 2021.

Sara Beery, Guanhang Wu, Trevor Edwards, Filip Pavetic, Bo Majewski, Shreyasee Mukherjee, Stanley Chan, John Morgan, Vivek Rathod, and Jonathan Huang. The auto arborist dataset: a large-scale benchmark for multiview urban forest monitoring under domain shift. In *CVPR*, pp. 21294–21307, 2022.

Mohamed Belghazi, Maxime Oquab, and David Lopez-Paz. Learning about an exponential amount of conditional distributions. *Advances in Neural Information Processing Systems*, 32, 2019.

Dimitris Bertsimas, Arthur Delarue, and Jean Pauphilet. Simple imputation rules for prediction with missing data: Theoretical guarantees vs. empirical performance. *TMLR*, 2024.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. In *NeurIPS*, volume 33, pp. 21865–21877, 2020.

Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

Minmin Chen, Kilian Weinberger, Fei Sha, and Yoshua Bengio. Marginalized denoising auto-encoders for nonlinear representations. In *ICML*, pp. 1476–1484. PMLR, 2014.

Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. Correlated input-dependent label noise in large-scale image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1551–1560, 2021.

Rahul S. Desikan, Florent Ségonne, Bruce Fischl, Brian T. Quinn, Bradford C. Dickerson, Deborah Blacker, Randy L. Buckner, Anders M. Dale, R. Paul Maguire, Bradley T. Hyman, Marilyn S. Albert, and Ronald J. Killiany. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *NeuroImage*, 31(3):968 – 980, 2006. ISSN 1053-8119. doi: DOI:10.1016/j.neuroimage.2006.01.021. URL `http://www.sciencedirect.com/science/article/B6WNP-4JFHF4P-1/2/0ec667d4c17eafb0a7c52fa3fd5aef1c`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Bruce Fischl, André van der Kouwe, Christophe Destrieux, Eric Halgren, Florent Ségonne, David H. Salat, Evelina Busa, Larry J. Seidman, Jill Goldstein, David Kennedy, Verne Caviness, Nikos Makris, Bruce Rosen, and Anders M. Dale. Automatically Parcellating the Human Cerebral Cortex. *Cerebral Cortex*, 14 (1):11–22, 2004. doi: 10.1093/cercor/bhg087. URL `http://cercor.oxfordjournals.org/content/14/1/11.abstract`.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *ICML*, pp. 1050–1059. PMLR, 2016.

Ignazio Gallo, Gianmarco Ria, Nicola Landro, and Riccardo La Grassa. Image and text fusion for upmc food-101 using bert and cnns. In *International Conference on Image and Vision Computing New Zealand*. IEEE, 2020.

Lovedeep Gondara and Ke Wang. Mida: Multiple imputation using denoising autoencoders. In *PAKDD*, pp. 260–272. Springer, 2018.

Elad Hazan, Roi Livni, and Yishay Mansour. Classification with low rank and missing data. In *ICML*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pp. 770–778, 2016.

Dan Hu, Han Zhang, Zhengwang Wu, Fan Wang, Li Wang, J Keith Smith, Weili Lin, Gang Li, and Dinggang Shen. Disentangled-multimodal adversarial autoencoder: Application to infant age prediction with incomplete multimodal neuroimages. *IEEE TMI*, 39(12):4137–4149, 2020.

Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. How to deal with missing data in supervised deep learning? In *ICLR*, 2022.

Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. In *ICLR*, 2019.

Hailey James, Chirag Nagpal, Katherine A Heller, and Berk Ustun. Participatory personalization in classification. In *NeurIPS*, 2023.

Julie Josse, Nicolas Prost, Erwan Scornet, and Gaël Varoquaux. On the consistency of supervised learning with missing values. *arXiv:1902.06931*, 2019.

Batuhan K. Karaman, Elizabeth C. Mormino, and Mert R. Sabuncu. Machine learning based multi-modal prediction of future decline toward alzheimer's disease: An empirical study. *PLOS ONE*, 17:e0277322, 11 2022. doi: 10.1371/journal.pone.0277322.

Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard Schölkopf, Michael Curtis Mozer, Christopher Pal, and Yoshua Bengio. Neural causal structure discovery from interventions. *TMLR*, 2023.

Douwe Kiela, Edouard Grave, Armand Joulin, and Tomas Mikolov. Efficient large-scale multi-modal classification. In *AAAI*, volume 32, 2018.

Heejong Kim, Daniel JA Margolis, Himanshu Nagar, and Mert R Sabuncu. Pulse sequence dependence of a simple and interpretable deep learning method for detection of clinically significant prostate cancer using multiparametric mri. *Academic Radiology*, 30(5):966–970, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of ICLR*, 2014.

Trent Kyono, Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. Miracle: Causally-aware imputation via learning missing data mechanisms. In *NeurIPS*, 2021.

Ranjit Lall and Thomas Robinson. The midas touch: accurate and scalable missing-data imputation with deep learning. *Political Analysis*, 30(2):179–196, 2022.

Marine Le Morvan, Julie Josse, Thomas Moreau, Erwan Scornet, and Gaël Varoquaux. Neumiss networks: differentiable programming for supervised learning with missing values. In *NeurIPS*, 2020a.

Marine Le Morvan, Nicolas Prost, Julie Josse, Erwan Scornet, and Gaël Varoquaux. Linear predictor on linearly-generated data with missing values: non consistency and solutions. In *AISTATS*. PMLR, 2020b.

Marine Le Morvan, Julie Josse, Erwan Scornet, and Gaël Varoquaux. What's a good imputation to predict with missing values? In *NeurIPS*, 2021.

Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Misgan: Learning from incomplete data with generative adversarial networks. In *ICLR*, 2019.

Yang Li, Shoaib Akbar, and Junier Oliva. Acflow: Flow models for arbitrary conditional likelihoods. In *ICML*, pp. 5831–5841. PMLR, 2020.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: On Pre-training for Visual Language Models. In *CVPR*, pp. 26689–26699, 2024.

Phillip Lippe, Taco Cohen, and Efstratios Gavves. Efficient neural causal discovery without acyclicity constraints. In *ICLR*, 2022.

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36: 34892–34916, 2023.

Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. In *International Conference on Machine Learning*, pp. 14153–14172. PMLR, 2022.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez-Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. In *ICML*, 2019.

Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu, and Xi Peng. Smil: Multimodal learning with severely missing modality. In *AAAI*, 2021.

Mengmeng Ma, Jian Ren, Long Zhao, Davide Testuggine, and Xi Peng. Are multimodal transformers robust to missing modality? In *CVPR*, 2022.

Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. Learning with marginalized corrupted features. In *ICML*, pp. 410–418. PMLR, 2013.

Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *ICML*, pp. 4413–4423. PMLR, 2019.

Susanne G. Mueller, Michael W. Weiner, Leon J. Thal, Ronald C. Petersen, Clifford R. Jack, William Jagust, John Q. Trojanowski, Arthur W. Toga, and Laurel Beckett. Ways toward an early diagnosis in alzheimer's disease: The alzheimer's disease neuroimaging initiative (adni). *Alzheimer's & Dementia*, 1:55–66, 07 2005. doi: 10.1016/j.jalz.2005.06.003.

Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. Moddrop: adaptive multi-modal gesture recognition. *IEEE TPAMI*, 38(8):1692–1706, 2015.

Guillermo Ortiz-Jimenez, Mark Collier, Anant Nawalgaria, Alexander Nicholas D'Amour, Jesse Berent, Rodolphe Jenatton, and Efi Kokiopoulou. When does privileged information explain away label noise? In *International Conference on Machine Learning*, pp. 26646–26669. PMLR, 2023.

Srinivas Parthasarathy and Shiva Sundaram. Training strategies to handle missing modalities for audio-visual expression recognition. In *International Conference on Multimodal Interaction*, 2020.

Ignacio Peis, Chao Ma, and José Miguel Hernández-Lobato. Missing data imputation and acquisition with deep hierarchical models and hamiltonian monte carlo. In *NeurIPS*, 2022.

Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9617–9626, 2019.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, pp. 5389–5400, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pp. 234–241. Springer, 2015.

Anindo Saha, Jasper Jonathan Twilt, Joeran Sander Bosma, Bram van Ginneken, Derya Yakar, Mattijs Elschot, Jeroen Veltman, Jurgen Fütterer, Maarten de Rooij, and Henkjan Huisman. The pi-cai challenge: public training and development dataset, 2022.

P Spirtes, C Glymour, R Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17 (6):520–525, 2001.

Baris Turkbey, Andrew B Rosenkrantz, Masoom A Haider, Anwar R Padhani, Geert Villeirs, Katarzyna J Macura, Clare M Tempany, Peter L Choyke, Francois Cornud, Daniel J Margolis, et al. Prostate imaging reporting and data system version 2.1: 2019 update of prostate imaging reporting and data system version 2. *European urology*, 76(3):340–351, 2019.

Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011.

Mike Van Ness, Tomas M Bosschieter, Roberto Halpin-Gregorio, and Madeleine Udell. The Missing Indicator Method: From Low to High Dimensions. In *KDD*, pp. 5004–5015, 2023.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 11(12), 2010.

Ke Wang, Guillermo Ortiz-Jimenez, Rodolphe Jenatton, Mark Collier, Efi Kokiopoulou, and Pascal Frossard. Pi-dual: Using privileged information to distinguish clean from noisy labels. *arXiv preprint arXiv:2310.06600*, 2023.

Xin Wang, Devinder Kumar, Nicolas Thome, Matthieu Cord, and Frederic Precioso. Recipe recognition with large multimodal food dataset. In *IEEE ICMEW*, 2015.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021.

Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *ICML*, pp. 5689–5698. PMLR, 2018.

Jiaxuan You, Xiaobai Ma, Yi Ding, Mykel J Kochenderfer, and Jure Leskovec. Handling missing data with graph representation learning. In *NeurIPS*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.