Deeply Learned Generalized Linear Models with Missing Data

David K. Lim*, Naim U. Rashid*, Junier B. Oliva**, and Joseph G. Ibrahim*

*Department of Biostatistics, University of North Carolina at Chapel Hill, Chapel Hill, NC; **Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC

ABSTRACT
Deep Learning (DL) methods have dramatically increased in popularity in recent years, with significant growth in their application to various supervised learning problems. However, the greater prevalence and complexity of missing data in such datasets present significant challenges for DL methods. Here, we provide a formal treatment of missing data in the context of deeply learned generalized linear models, a supervised DL architecture for regression and classification problems. We propose a new architecture, dlglm, that is one of the first to be able to flexibly account for both ignorable and non-ignorable patterns of missingness in input features and response at training time. We demonstrate through statistical simulation that our method outperforms existing approaches for supervised learning tasks in the presence of missing not at random (MNAR) missingness. We conclude with a case study of the Bank Marketing dataset from the UCI Machine Learning Repository, in which we predict whether clients subscribed to a product based on phone survey data. Supplementary materials for this article are available online.

1. Introduction
Deep Learning (DL) methods have been increasingly used in an array of supervised learning problems in various fields, for example, in the biomedical sciences (Razzak, Naz, and Zaib 2017; Lopez et al. 2018). While a number of deep learning architectures have been proposed for supervised learning, the feed forward neural network (FFNN) is commonly used in most architectures. In a FFNN, sequential nonlinear transformations are applied to the values of the input layer. Each value in the subsequent layer of the FFNN is computed by applying a nonlinear (or “activation”) function to the linear transformation of the values in the previous layer, outputting a complex nonlinear transformation of the input (Svozil, Kvasnicka, and Pospichal 1997). For example, a FFNN architecture called the deeply learned GLM (Tran et al. 2019) has been applied in the context of supervised learning to describe nonlinear relationships between the covariates and the response. Due to the large number of parameters, so optimization is often done via stochastic gradient descent (Guo and Gelfand 1990).

However, the common presence of missing data in datasets can hinder the training and generalizability of supervised deep learning methods (Wells et al. 2013), where missingness can occur both in the input features and the response variable. Missingness has commonly been categorized into three mechanisms: Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR) (Rubin 1976). While a number of methods have been proposed in the statistical literature to address MNAR missingness in the regression setting (Ibrahim et al. 2005), such methods often cannot take into account complex relationships between predictors and response and are not scalable to higher dimensions (Chen et al. 2019), or have been specifically designed for unsupervised learning tasks (Lim et al. 2021). Supervised deep learning is one way to address capture complex relationships between predictors and response in a scalable manner (Kingma and Welling 2019), however, it is unclear how best to account for more complex forms of missingness, such as MAR or MNAR missingness, in this setting.

There have been some recent attempts to perform prediction using deep learning in the presence of missing features (Ipsen, Mattei, and Frellsen 2021), but such methods typically assume either MCAR or MAR missingness. Commonly used missing data methods in supervised deep learning applications, such as mean imputation or complete case analysis, have historically yielded biased results (Ibrahim and Molenberghs 2009). Multiple Imputation by Chained Equations (mice) has also been widely employed to account for missing data in a supervised learning setting. However, mice is unable to apply a trained imputation model to handle missingness that may exist at test time (Hoogland et al. 2020). In addition, multiple imputation-based methods may not be feasible to apply when the downstream model is computationally intensive, such as in the setting of training a deep learning neural network, since one must train the model separately for each imputed dataset. Lastly, existing approaches to handle MAR or MCAR missingness when training deep learning models for supervised learning tasks are currently limited, and have not been sufficiently explored in the literature.

To address these issues, we present dlglm: a deep generalized linear model (GLM) for probabilistic supervised learning in the presence of missing input features and/or response across a variety of missingness patterns. Our proposed method uses

CONTACT David K. Lim david.lim@alumni.unc.edu Department of Biostatistics, University of North Carolina at Chapel Hill, Chapel Hill, NC.
variational inference to learn approximate posterior distributions for the missing variables, and replaces missing entries with samples from these distributions during maximization. In this way, \( dlglm \) can perform supervised learning in the presence of missingness in both the features and the response of interest. We also incorporate a model for the missingness, which can take into account MNAR patterns of missingness, even at training time. Through neural networks, \( dlglm \) is able to model complex nonlinear relationships between the input features and the response, and is scalable to large quantities and dimensionalities of data. Prediction can be done seamlessly on fully or partially observed samples using the trained model, without requiring separate imputation of the missing values.

2. Methods

Here we first discuss the formulation of the generalized linear model (GLM) in Section 2.1, and then introduce the deeply learned GLM in Section 2.2. We then discuss missingness in the context of GLMs in Section 2.3, and lastly propose a novel deep learning architecture \( dlglm \) in Section 2.4 to fit deeply learned GLMs in the presence of missingness.

2.1. Generalized Linear Models (GLMs)

Let \( X \) be the \( n \times p \) matrix of covariates (input features) with observation vectors \( x_i \), where each corresponding entry \( x_{ij} \) denotes the value of the \( i \)th observation of the \( j \)th feature for \( i = 1, \ldots, n \) and \( j = 1, \ldots, p \). Also, let \( Y = \{y_1, \ldots, y_n\} \) be the vector of univariate responses where \( y_i \) is the response pertaining to the \( i \)th observation. We note that \( y_i \) may also be assumed to be multivariate; however, we focus specifically on the case of univariate response to simplify the discussion, and discuss extensions to the setting of multivariate response in Section 4. Then, denote \( \eta = X\beta \), where \( \beta \) is a vector of regression coefficients and \( \eta \) is the linear predictor. Also define \( \mu = \{\mu_1, \ldots, \mu_n\} \) with \( \mu_i = E(y_i|x_i) \) and link function \( g(\cdot) \) such that \( g(\mu_i) = \eta_i = x_i\beta \). We assume that the conditional distribution \( p(y_i|x_i) \) is a member of the exponential family of distributions (McCullagh and Nelder 2019), such that \( p(y_i|x_i) \) can be written as

\[
p(y_i|x_i) = \exp \left[ \frac{y_i(\Theta_i - b(\Theta_i))}{a(\alpha)} + c(y_i, \alpha) \right],
\]

with canonical parameter \( \Theta_i \), dispersion parameter \( \alpha \), and some functions \( a(\cdot), b(\cdot), \) and \( c(\cdot) \). Here, we further assume \( g(\cdot) \) is a canonical link function such that \( g(\mu_i) = \Theta_i \). With the appropriate specification of the canonical link \( g(\cdot) \) and variance function \( V_\alpha(\cdot) \), we obtain the formulation of a GLM.

GLMs were first motivated by the limitations of the traditional linear model, which imposed strict assumptions of linearity between \( \mu \) and \( X \) and of normality of errors with fixed variance. GLMs instead use specific link and variance functions, allowing for model fitting on types of response data that may violate these assumptions, such as count or categorical outcomes, without having to rely on heuristic transformations of the data (Nelder and Wedderburn 1972). Typically, GLMs are estimated by using iteratively re-weighted least squares in lower dimensions (Holland and Welsch 1977), with extensions to the higher dimensional case via penalized likelihood (Friedman, Hastie, and Tibshirani 2010).

2.2. Deeply Learned GLMs

The traditional GLM assumes \( g(\mu_i) \) is a linear function of \( x_i \), that is, \( g(\mu_i) = x_i\beta \). In many modern applications, one may wish to model \( g(\mu_i) \) as a nonlinear function of \( x_i \) or capture complex interactions between features to predict response (Qi and Wu 2003). In such cases, we may generalize the GLM to a deeply learned GLM (Tran et al. 2019) with the following expression:

\[
g(\mu_i) = h_{\pi}(x_i)\beta, \quad \text{where } h_{\pi}(\cdot) \text{ denotes the output of a series of nonlinear transformations applied to the input } X \text{ by a neural network, with weights and bias parameters denoted by } \pi.
\]

In addition, \( \eta_i \) can alternatively be expressed \( \eta_i = s_{\pi,\beta}(x_i) \), where \( s_{\pi,\beta}(\cdot) \) is a neural network where \( \beta \) denotes the weights and bias associated with the output (last) layer of \( s_{\pi,\beta}(\cdot) \). This formulation allows for the traditional interpretation of \( \beta \) as the coefficients pertaining to a transformed version of the input covariates. Figure 1 shows an illustration of this architecture.

Let \( n_{HL} \) denote the number of hidden layers in \( s_{\pi,\beta}(\cdot) \). We note that if \( n_{HL} = 0 \), then \( h_{\pi}(x_i) = x_i \) and \( s_{\pi,\beta}(x_i) = x_i\beta \), reducing to the traditional GLM. Deeply learned GLMs and other neural networks are often maximized using stochastic gradient descent (Bottou 2012). Details of this algorithm can be found in Appendix A1 of the supplementary materials.

2.3. Missingness in GLMs

Many modern datasets often contain complex forms of missingness (Ghorbani and Zou 2018). In GLMs, missingness can exist in either \( X \) or \( Y \). Therefore, we specify three cases of missingness in this context: missing covariates with fully observed response (Case \( x \)), missing response with fully observed covariates (Case \( y \)), and missing covariates and missing response (Case \( xy \)). Define \( R = (R^X, R^Y) \) as the “missingness mask,” which denotes the missingness of \( \{X, Y\} \), such that \( R^X \) and \( R^Y \) have the same
dimension as \( X \) and \( Y \), respectively, and a value of 1 in \( R \) denotes that the corresponding entry in \( [X, Y] \) is observed, while a value of 0 denotes that it is unobserved. Additionally, let \( R = \{ r_1, \ldots, r_n \} \) with \( r_i = \{ r_{i1}, r_{i2} \} = \{ r_{i1}^X, r_{i2}^X, r_{i1}^Y, r_{i2}^Y \} \), \( R^X = \{ r_{i1}^X, \ldots, r_{i1}^Y \} \), and \( R^Y = \{ r_{i1}^Y, \ldots, r_{i2}^Y \} \) with elements \( r_{ij}^X \) and \( r_{ij}^Y \) pertaining to the missingness of the \( i \)th observation of \( X \) and \( Y \), respectively. Then, \( X \) and \( Y \) can be factored into the unobserved and observed entries \( [X^m, X^o] \) and \( [Y^m, Y^o] \), respectively, such that \( X^m = \{ X : R^X = 0 \} \) with \( X^m = \{ x_i : r_{i1}^X = 0 \} \), \( X^o = \{ X : R^X = 1 \} \) with \( x_i = \{ x_i : r_{i1}^X = 1 \} \), and \( Y^m = \{ Y : R^Y = 0 \} \) and \( Y^o = \{ Y : R^Y = 1 \} \), with \( y_i = \{ y_i : r_{i1}^Y = 0 \} \) and \( y_i^o = \{ y_i : r_{i1}^Y = 1 \} \).

Missingness was classified into three primary mechanisms in the seminal work by Little and Rubin (2002): missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). They satisfy the following relations:

- **MCAR:** \( p(r_i|x_i, y_i, \phi) = p(r_i|\phi) \)
- **MAR:** \( p(r_i|x_i, y_i, \phi) = p(r_i|x_i^m, y_i^m, y_i, \phi) \)
- **MNAR:** \( p(r_i|x_i, y_i, \phi) = p(r_i|x_i^m, y_i^m, y_i, \phi) \).

Here, \( \phi \) denotes the collection of parameters for the model of the missingness mask \( r_i \). In the presence of missingness, the marginal log-likelihood can generally be written as

\[
\log p_{\theta, \beta, \pi, \psi}(X, Y, O) = \log \int \int p_{\theta, \beta, \pi, \psi}(X | Y) p(Y | X) dX^m dY^m
\]

where \( \psi \) is a set of parameters associated with the covariate distribution \( p_{\theta}(X) \). We factor \( p_{\theta, \beta, \pi, \psi}(X | Y, R) \) using the selection model factorization (Diggle and Kenward 1994).

Under MNAR, it is not possible to remove \( p_{\theta}(R | X, Y) \) from the integral, since \( R \) can depend on \( [X^m, Y^m] \). Therefore, MNAR missingness is said to be non-ignorable, because it requires specification of the so-called "missingness model" \( p(r_i|x_i, y_i, \phi) \) (Stubbendick and Ibrahim 2003). There are a number of ways to specify this model. For example, Diggle and Kenward (1994) proposes a binomial model for the missing data mechanism, which can be written in this setting as

\[
p(R | X, Y, \phi_{jm}) = \prod_{i=1}^{n} \prod_{jm=1}^{p_{miss}} \left[ p(r_{jm} = 1 | x_i, y_i, \phi_{jm}) \right]^{r_{jm}} \times \left[ 1 - p(r_{jm} = 1 | x_i, y_i, \phi_{jm}) \right]^{1-r_{jm}},
\]

where \( j_m = 1, \ldots, p_{miss} \) indexes the \( p_{miss} \) features in \( [X, Y] \) that contain missingness. Here \( p_{miss} = P_{miss}^X + P_{miss}^Y \), where \( P_{miss}^X \) is the total number of features containing missingness in \( X \), and \( P_{miss}^Y \) is 1 if \( Y \) contains missingness (0 otherwise). Also, \( \phi_{jm} \) is the set of coefficients pertaining to the missingness model of the \( j_m \) missing variable, and \( p(r_{jm} = 1 | x_i, y_i, \phi_{jm}) \) can be modeled straightforwardly by a logistic regression model, such that

\[
\logit(p(r_{jm} = 1 | x_i, y_i, \phi_{jm})) = \phi_{0jm} + \gamma_i \phi_{1jm} + x_i^T \phi_{2jm} + x_i^m \phi_{3jm},
\]

where \( \phi_{0jm} \) is the intercept of the \( j_m \)th missingness model, \( \phi_{1jm} \) is the coefficient pertaining to the response variable \( Y \), and \( \phi_{2jm} = (\phi_{2jm,1}, \ldots, \phi_{2jm,p_{obs}} X^T) \) and \( \phi_{3jm} = (\phi_{3jm,1}, \ldots, \phi_{3jm,p_{miss}} x_i^m) \) are the sets of coefficients of the \( j_m \)th variable’s missingness model pertaining to the effects of the observed and missing features on the missingness, respectively, with \( p_{obs}^X \) and \( p_{miss}^X \) denoting the number of completely-observed and partially observed features in \( X \), respectively. Note that this model assumes independence of \( R \) across the \( p_{miss} \) missing features, such that the missingness of each variable is conditionally independent of whether any other variable has been observed, which may or may not be realistic in some settings (Ibrahim et al. 2005).

When missingness is assumed to be MAR or MCAR, the marginal log-likelihood can be factored as

\[
\log p_{\theta, \beta, \pi, \psi}(X, Y, O) = \log p_{\theta, \beta, \pi, \psi}(X^m, Y^o) + \log p_{\theta}(R^o | X^o, Y^o). \quad (2)
\]

In this case, the quantity \( \log p_{\theta}(R^o | X^o, Y^o) \) need not be specified, since it is independent from the parameters of interest pertaining to \( p_{\theta, \beta, \pi, \psi}(X^o, Y^o) \). Therefore, MAR or MCAR missingness is often referred to as "ignorable" missingness. Equation (1) can then be expressed as

\[
\log p_{\theta, \beta, \pi, \psi}(X^o, Y^o) = \log \int \int p_{\theta, \beta, \pi, \psi}(Y | X) p(Y | X) dX^m dY^m.
\]

### 2.4. Deeply Learned GLM with Missingness (dlglm)

In this section, we propose an algorithm for training deeply learned GLMs in the presence of MCAR, MAR, and MNAR missingness. Before discussing this model, we first discuss the specification of the so-called covariate distribution \( p_{\theta}(X) \) introduced in (1) and (2), which is critical for maximizing the marginal log-likelihood in either setting. In Sections 2.4.1–2.4.2, we discuss two different models for \( p_{\theta}(X) \), and then in Section 2.4.3 we propose a novel method to handle missingness using a deeply learned GLM architecture with an Importance-Weighted Autoencoder (IWAE) covariate structure. To simplify the discussion, we narrow the scope of our discussion to the Case \( x \) setting, where only \( X \) contains missingness, but note that the proposed methodology naturally extends to Case \( y \) and Case \( xy \) settings as well.

#### 2.4.1. Modeling \( p_{\theta}(X) \) with Known Distribution

Given (1), we must model \( X \) with some assumed covariate distribution \( p_{\theta}(X) \). Care must be taken in specifying this distribution, as improper specification may reduce the accuracy of estimation of the parameters of interest \( \beta \) (Lipsitz and Ibrahim 1996). For example, we may assume \( p_{\theta}(X) \) follows some known multivariate distribution such as the multivariate normal distribution, where \( X \sim N_p(\mu, \Sigma) \) and \( \psi = \{ \mu, \Sigma \} \). Here, \( \psi \) can be optimized jointly with the rest of the parameters \( \{ \alpha, \beta, \pi, \phi \} \) that are involved in the marginal log-likelihood. However, this assumption may not be applicable in many instances such as in the case when \( X \) contains mixed data types, where both continuous and discrete features may be correlated and a joint distribution may be difficult to specify in closed form. In certain cases, it may be beneficial to model \( p_{\theta}(X) \) flexibly, such that no strong prior assumptions need to be made on the form of this distribution. To address this, a sequence of 1-D conditionals have previously been proposed to model the covariate distribution (Lipsitz and Ibrahim 1996), but such a model may be computationally intractable when the number of covariates is very large.
Once an explicit form for the covariate distribution is specified, one aims to maximize the marginal log-likelihood, as introduced in (1) in Section 2.3. However, due to the integral involved, this quantity is often intractable and is difficult to maximize directly, so a lower bound of the marginal log-likelihood is often maximized instead. The derivation of this lower bound can be found in Appendix A2 of the supplementary materials.

### 2.4.2. Modeling \( p_\psi(X) \) with Variational and Importance-Weighted Autoencoders

Alternatively, one can approximately learn \( p_\psi(X) \) from the training data by using an IWAE neural network architecture. In this section, we first introduce a general form of the variational autoencoder (VAE) and IWAE in the case of completely-observed data \( X \). Then, in Section 2.4.3, we apply the IWAE variational structure to the deeply learned GLM setting and show how this representation naturally extends to the case where MCAR, MAR, or MNAR missingness is observed in \( X \) when training deeply learned GLMs.

First, let \( Z \) be an \( n \times d \) matrix, such that \( Z = (z_1, \ldots, z_i) \) and \( z_i \) is a latent vector of length \( d \) pertaining to the \( i \)-th sample latent variable, and let \( Z \) represent a lower-dimensional representation or subspace of \( X \). It is common practice to tune the value of \( d \) as a hyperparameter by choosing the optimal integer value that best fits the data, as measured by some objective function. In a VAE, we assume \( x_1, \ldots, x_n \) are iid samples from a multivariate p.d.f or "generative model" \( p_\psi(X|Z) \) with accompanying parameters \( \psi \) that describes how \( X \) is generated from the lower dimensional space \( Z \). In this manner, a VAE aims to learn accurate representations of high-dimensional data, and may be used to generate synthetic data with similar qualities as the training data. These aspects are also aided through the use of embedded deep learning neural networks, for example within \( p_\psi(X|Z) \), which also facilitates its applicability to larger dimensions and complex datasets.

In a VAE with completely observed training data, one aims to maximize the marginal log-likelihood as \( \log p_\psi(X) = \log \int p_\psi(X,Z)dZ = \log \int p_\psi(X|Z)p(Z)dZ. \) However, this quantity is also often intractable and difficult to maximize directly. Therefore, VAE’s alternatively optimize an objective function called the “Evidence Lower Bound” (ELBO), which lower bounds \( \log p_\psi(X) \) and has the following form (Kingma and Welling 2013):

\[
\mathcal{L}^{\text{ELBO}}(\theta, \psi) = \mathbb{E}_{Z \sim q_\theta(Z|X)} \left[ \log \frac{p_\psi(X|Z)p(Z)}{q_\theta(Z|X)} \right]
\]

Here, \( \mathcal{L}^{\text{ELBO}}(\theta, \psi) \) denotes the ELBO such that \( \mathcal{L}^{\text{ELBO}}(\theta, \psi) \leq \log p_\psi(X). \) Also let \( \mathcal{L}^{\text{ELBO}}_K(\theta, \psi) \) denote the empirical approximation to (3) computed by Monte Carlo integration, such that \( \mathcal{L}^{\text{ELBO}}(\theta, \psi) \approx \mathcal{L}^{\text{ELBO}}_K(\theta, \psi) \) and \( \tilde{Z}_1, \ldots, \tilde{Z}_K \) are \( K \) samples drawn from \( q_\theta(Z|X) \), the variational approximation of the true but intractable posterior \( p_\theta(Z|X) \), also called the “recognition model”. Furthermore, denote \( f_\psi(Z) \) and \( g_\psi(X) \) as the decoder and encoder feed forward neural networks of the VAE, where \( \psi \) and \( \theta \) are the sets of weights and biases pertaining to each of these neural networks, respectively. Given \( Z, f_\psi(Z) \) outputs the distributional parameters pertaining to \( p_\psi(X|Z) \).

In variational inference, \( q_\theta(Z|X) \) is constrained to be from a class of simple distributions, or "variational family", to obtain the best candidate from within that class to approximate \( p_\psi(X|Z) \). Variational inference is usually used in tandem with amortization of the parameters where the neural network parameters are shared across observations (Gershman and Goodman 2014), allowing for stochastic gradient descent (SGD) to be used for optimization of (4) (Kingma and Welling 2019). In practice, both \( q_\theta(Z|X) \) and \( p(Z) \) are typically assumed to have simple forms, such as multivariate Gaussians with diagonal covariance structures, and \( q_\theta(Z|X) \) is commonly assumed to be factorizable, such that \( q_\theta(Z|X) = \prod_{i=1}^n q_\theta(z_i|x_i) \) (Kingma and Welling 2019). Although one can specify a class of more complicated distributions for \( q_\theta(Z|X) \) as long as they are reparameterizable (Li, Akbar, and Oliva 2020; Strauss and Oliva 2021, 2022), the multivariate Gaussian with diagonal covariance structure is most often used, following works by Burda, Grosse, and Salakhutdinov (2015) and Kingma and Welling (2013), due to the convenience in sampling and computation (Kingma and Welling 2019).

Let \( (\theta^{(t)}, \psi^{(t)}) \) be the estimates of \( (\theta, \psi) \) at update (or iteration) \( t \). For \( t = 0 \), these values are often initialized to small values centered around 0, although other initialization schemes may be used (Saxe, McClelland, and Ganguli 2014; Murphy 2016). Each subsequent update \( t \geq 1 \) consists of two general steps to maximize \( \mathcal{L}(\theta, \psi) \). First, \( K \) samples are drawn from \( q_\theta(Z|X) \) to compute the quantity in (4), conditional on \( \theta \) and \( \psi \), similar to importance sampling. Then, the so-called "reparameterization trick" is used to facilitate the calculation of gradients of this approximation to obtain \( (\hat{\theta}^{t+1}, \hat{\psi}^{t+1}) \) using stochastic gradient descent (Kingma and Welling 2013). The networks \( f_\psi(Z) \) and \( g_\psi(X) \) also allow the VAE to capture complex and nonlinear relationships between features in outputting the distributional parameters for the generative and recognition models, respectively. This procedure may be repeated for a fixed number of iterations, or may be terminated early due to pre-specified convergence criteria (Prechelt 1998). Kingma and Welling (2013) provides additional details on the maximization procedure for VAEs.

The IWAE (Burda, Grosse, and Salakhutdinov 2015) is a generalization of the standard VAE. Both the VAE and IWAE estimate \( \log p_\psi(X) \) by drawing samples of latent variables to estimate an expectation. However, while the VAE uses \( p_\psi(X|Z)/q_\theta(Z|X) \) as the importance weights in deriving the ELBO, the IWAE uses the average of \( K \) importance weights in the integrand for a tighter lower bound of the marginal log-likelihood (Burda, Grosse, and Salakhutdinov 2015). The resulting IWAE bound, corresponding to the ELBO in (3), can be written as

\[
\mathcal{L}^{\text{IWAE}}_K(\theta, \psi) = \mathbb{E}_{Z \sim q_\theta(Z|X)} \left[ \frac{1}{K} \sum_{k=1}^K \frac{p_\psi(X|Z_k)p(Z_k)}{q_\theta(Z_k|X)} \right]
\]

(5)

\[
\hat{\mathcal{L}}^{\text{IWAE}}_K(\theta, \psi) = \log \left[ \frac{1}{K} \sum_{k=1}^K \frac{p_\psi(X|Z_k)p(Z_k)}{q_\theta(Z_k|X)} \right]
\]

(6)
Importantly, although K samples are drawn from \( q(Z|X) \) to estimate the lower bound for both the VAE and IWAE, a VAE assumes a single latent variable \( Z \) that is sampled K times, whereas an IWAE assumes \( Z_1, \ldots, Z_K \) are iid latent variables, and each variable is sampled once from \( q(Z|X) \). Typically, just one sample is drawn for each latent variable to estimate the ELBO and IWAE bound. If \( K = 1 \), \( L_{K}^{IWA} = L_{K}^{VAE} \), and the IWAE corresponds exactly to the standard VAE. For \( K > 1 \), Burda, Grosse, and Salakhutdinov (2015) showed that \( \log p(X) \leq L_{K}^{IWA} \leq L_{K}^{VAE} \), such that \( L_{K}^{IWA} \to \log p(X) \) as \( K \to \infty \) if \( \log p(X) \) is bounded. Thus, the IWAE bound more closely approximates the true marginal log-likelihood when \( K > 1 \) (Cremer, Morris, and Duvenaud 2017), but the computational burden is increased due to the increased number of samples. A visualization of the workflow for an IWAE can be found in Appendix A3 of the supplementary materials.

### 2.4.3. dlglm: Modeling X in the Presence of Missingness

Now, we extend the above framework to the deeply learned GLM framework, where features within \( X \) are partially observed during training. We formally introduce the dlglm model to handle MNAR missingness in the context of deeply learned GLMs, as well as a variant of dlglm to specifically handle MCAR and MAR missingness.

Let us define \( q_\theta(X, X^m) \) as the variational joint posterior pertaining to \((Z, X^m)\). Then, we can factor this variational joint posterior as \( q_\theta(X, X^m) = q_\theta_1(Z|X^o)q_\theta_2(X^m|Z, X^o, R, Y) \). Here, for \( k = 1, \ldots, K \), we assume \( Z_k \sim q_\theta(Z|X) \) similar to an IWAE, and additionally assume \( X^m_k \sim q_\theta_2(X^m|Z, X^o, R, Y) \), where each \( X^m_k \) has dimensionality \( p_{X}^m \). Hence, assume that \( \mathbf{Y} \) is generated by \( X \) and thus it is redundant to utilize \( \mathbf{Y} \) in the part of the variational joint posterior pertaining to \( Z \). Empirically, we observed that including \( \mathbf{Y} \) in the conditional, such that \( q(Z|X^o, Y) \), did not have a significant impact. Additionally, we note that the form \( q_\theta_2(X^m|Z, X^o, R, Y) \) includes \( \mathbf{Y} \), allowing for more accurate imputation of missing values; however, we remove this term in the conditional in the context of prediction, in order to predict \( \mathbf{Y} \) in an unbiased manner.

We then use the class of factored variational posteriors, such that \( q_\theta(X, X^m) = \prod_{i=1}^{n} q_\theta(z_i, x^m_i) \) and \( q_\theta(z_i, x^m_i) = q_\theta_1(z_i|x^o_i)q_\theta_2(x^m_i|z_i,x^o_i,r^*_i) \), with \( \theta = \{\theta_1, \theta_2\} \). Then, denoting \( z_{ik} \) and \( x^m_{ik} \) as the \( i \)th observation vectors of \( Z_k \) and \( X^m_k \), respectively, we have \( z_{ik}, \ldots, z_{ik} \sim q_\theta_1(z_i|x^o_i) \) and \( x^m_{ik}, \ldots, x^m_{ik} \sim q_\theta_2(x^m_i|z_i,x^o_i,r^*_i) \). In this case, the lower bound, which we call the “dlglm bound,” can be derived as follows:

\[
\log p_{\theta, \beta, \pi, \psi, \phi}(X^y, Y, R^*) = \sum_{i=1}^{n} \log p_{\theta, \beta, \pi, \psi, \phi}(x^y_i, y_i, r^*_i)
= \sum_{i=1}^{n} \log \left[ \int p_{\theta, \beta, \pi, \psi, \phi}(x^y_i, x^m_i, y_i, r^*_i, z_i) dx^m_i \right]
= \sum_{i=1}^{n} \log \mathbb{E}_{z_{ik}, x^m_{ik} \sim q_\theta(z_i, x^m_i)} \left[ \frac{1}{K} \sum_{k=1}^{K} p_{\theta, \beta, \pi, \psi, \phi}(x^m_i|z_{ik}, r^*_i, x^o_i) q_\theta(z_{ik}, x^m_i) \right]
\geq \sum_{i=1}^{n} \mathbb{E}_{z_{ik}, x^m_{ik} \sim q_\theta(z_i, x^m_i)} \log \left[ \frac{1}{K} \sum_{k=1}^{K} p_{\theta, \beta, \pi, \psi, \phi}(x^m_i|z_{ik}, r^*_i, x^o_i) q_\theta(z_{ik}, x^m_i) \right]
= \ell_{K}^{dlglm}.
\]

Here, \( \psi, \beta, \pi, \phi, \theta \) are the weights and biases associated with the neural networks that output the parameters of the distributions that are involved, \( \alpha \) is the dispersion parameter associated with the variance function of \( Y \), and \( z_{ik} \) and \( x^m_{ik} \) are the samples drawn from \( q_\theta_1(z_i|x^o_i) \) and \( q_\theta_2(x^m_i|z_i, x^o_i, r^*_i) \), respectively.

As discussed in Section 2.3, we use the selection model factorization of the complete data log-likelihood, such that \( p_{\alpha, \beta, \pi, \psi, \phi}(x^o_i, x^m_i, y_i, r^*_i, z_i) = p_{\alpha, \beta, \pi}(y_i|x_i) p_{\psi}(x|z_i)p(z_i)p(r|X, y) \). As before, we can remove \( y_i \) from \( p_{\psi}(r^*_i|x_i, y) \) for unbiased prediction. Then, applying this factorization to (7), we obtain the form of the estimate of the “dlglm bound,” where the integral is estimated via Monte Carlo integration:

\[
\hat{L}_{K}^{dlglm} = \sum_{i=1}^{n} \log \left[ \frac{1}{K} \sum_{k=1}^{K} p_{\alpha, \beta, \pi}(y_i|x^o_i) p_{\psi}(x|z_{ik}) p(z_{ik}) p(r^*_i|x_i, y_i) \right].
\]

We see that this quantity closely resembles the lower bound of an IWAE, and, similar to traditional VAEs, we use neural networks \( f_{\theta}(z_i), g_{\phi}(z^o_i), g_{\phi}(z_i, x^o_i, r^*_i), s_{\beta, \pi}, \psi(x), \text{ and } h_{\phi}(x_i) \) to learn the values of the parameters of \( p_{\psi}(x|z_i), q_{\theta_1}(z_i|x^o_i), q_{\theta_2}(x^m_i|z_i, x^o_i, r^*_i), p_{\alpha, \beta, \pi}(y_i|x_i) \), and \( p_{\psi}(r^*_i|x_i) \). The associated weights and biases of the neural networks \( \{\beta, \pi, \psi, \phi, \theta\} \) as well as the dispersion parameter \( \alpha \) are updated using stochastic gradient descent via the ADAM optimizer (Kingma and Ba 2014). Importantly, we call the neural network denoted by \( h_{\phi}(x_i) \) the “missingness network.” The inclusion of this network allows us to learn a model for the missingness mechanism, which is essential for accurate analysis in the presence of MNAR or non-ignorable missingness. The architecture of dlglm can be found in Figure 2. A pseudo-algorithm of dlglm can be found in Appendix A4 of the supplementary materials. We limited our discussion in this article to Case x, where missingness exists only in \( X \) but not in \( Y \); however, the lower bound for dlglm can similarly be derived for the more general Case xy as well, and this derivation can be found in Appendix A5 of the supplementary materials.

We can obtain a variant of this method, which we call ignorably missing dlglm (\( idlglm \)), by assuming independence between \( X^m \) and \( R \) by omitting \( r^*_i \) from (7), and removing \( p_{\psi}(r^*_i|x_i^m, x^o_i, r^*_i) \) and letting \( p_{\psi}(x^m_i|z_{ik}, x^o_i, r^*_i) \rightarrow p_{\psi}(x^m_i|z_{ik}, x^o_i) \) in Equation 8. Whereas dlglm is better suited to handle MNAR, idlglm may be more appropriate for the MCAR or MAR settings, where a missingness model need not be specified.

In this article, we are primarily interested in supervised learning. However, following training, dlglm and idlglm can also perform imputation as in the unsupervised learning architecture for handling missingness proposed by Lim et al. (2021), although such imputation is not necessary for training, coefficient estimation, or prediction. The single imputation procedure, and additional computational details of dlglm and idlglm can be found in Appendix A6 and A1 of the supplementary materials, respectively.

A recently published method by Ipsen, Mattei, and Frellesen (2021) performs unsupervised learning by similarly learning a missingness model in their neural network framework to handle...
MNAR missingness. However, they assume that \( p(X^m|Z) \approx q(X^m|Z, X^o, R) \). This may be an oversimplification, as in the MNAR case, \( R \) cannot be assumed to be independent of \( X^o \). Recent work by Ma and Zhang (2021) similarly performs unsupervised learning under MNAR missingness, including an auxiliary fully-observed variable to guarantee identifiability. However, they also make the same simplifying assumption as Ipsen, Mattei, and Frellsen (2021), which may not hold in MNAR. In addition, both methods are designed for imputation, rather than supervised tasks, and extending these methods for such tasks may be non-trivial, especially for computationally intensive models. Issues of identifiability in missing data applications often lead to issues of convergence during model training (Beesley, Taylor, and Little 2019). We note that although deriving the identifiability of \( dlglm \) is not focal point of this article, we consistently observed convergence in training the \( dlglm \) architecture in various simulations and real data settings.

3. Numerical Examples

In this section, we evaluate the performance of \( dlglm \) and \( idlglm \) to analyze each method’s performance in imputation, coefficient estimation, and prediction tasks on simulated datasets under MCAR, MAR, and MNAR missingness in Section 3.1. We also compare our methods to two commonly used approaches for modeling missing data in the supervised setting, mean imputation and the \( mice \) method for multiple imputation (Van Buuren and Groothuis-Oudshoorn 2011). We also compared performance in simulated data with two deep learning methods that were recently published \( miwae \) (Mattei and Frellsen 2019) and \( notmiwae \) (Ipsen, Mattei, and Frellsen 2021). To account for potential non-linearity and complex relationships between features, in Section 3.2, we mask completely-observed datasets obtained from the UCI Machine Learning Repository with varying mechanisms of missingness on the predictors. Finally, in Section 3.3, we perform prediction on the Bank Marketing dataset, which inherently contains missingness in the predictors.

In all simulated and real data analyses, we tuned a variety of hyperparameters for deep learning methods, including the number of hidden layers, the dimensionality of the latent variable \( Z \), and the number of nodes per hidden layer. For \( dlglm \), we additionally tuned the number of hidden layers in the missingness network separately, allowing the network to accurately capture potentially complex nonlinear relationships in the missingness model.

A grid-search approach was used for training based upon discrete pre-specified values, selecting the optimal combination of hyperparameters using the lower bound computed on a held out validation set. The selected hyperparameters for the simulated datasets, as well as the UCI and Bank Marketing datasets are listed in Appendix B1 of the supplementary materials.

3.1. Simulated Data

3.1.1. Simulation Setup

We first used completely synthetic data to evaluate the performance of each. Here, \( X \) is generated such that \( X = normalize(ZW + B) + B_0 \), where \( normalize(\cdot) \) takes an input matrix and standardizes each column to mean 0 and standard deviation 1, and \( W \) and \( B \) are matrices of dimensions \( d \times p \) and \( n \times p \), respectively, \( Z \sim N_d(0, I) \), and \( W_{ij} \sim N(0, 0.5) \) and \( B_{ij} \sim N(0, 1) \) for \( i = 1, \ldots , n, p = 1, \ldots , p \), and \( j = 1, \ldots , d \), and \( B_0 = 2 \) is fixed. We also generated a binary response variable \( Y \) such that \( logit(Pr(Y = 1|X)) = \beta_0 + \beta X \), where \( \beta \) are drawn randomly from \( \{-\frac{1}{2}, \frac{1}{2}\} \), and \( \beta_0 \) is chosen such that approximately half of the sample are in either class. Values of \( Y \) are drawn from Bernoulli(\( Pr(Y = 1|X) \)).

We then simulate the missingness mask matrix \( R_X \) such that 50% of features in \( X \) are partially observed, and 30% of the observations for each of these features are missing. We generate \( r_{ij} \) from the Bernoulli distribution with probability equal to \( p(r_{ij} = 1) = 1|X_i, y_i, \phi \), such that \( logit(p(r_{ij} = 1) = 1|X_i, y_i, \phi) = \phi_0 + \phi_1 y_i + \phi_2 x_{i1} + \phi_3 x_{i2} \), where \( j = 1, \ldots , p_X \) index the missing features, \( \phi_1 \) is the coefficient pertaining to the response, \( \phi_2 = \{\phi_{21}, \ldots , \phi_{2p_X}\} \) are the coefficients pertaining to the observed features, and \( \phi_3 = \{\phi_{31}, \ldots , \phi_{3p_{mis}}\} \) are those pertaining to the missing features, where \( p_X = r_{obs} + p_{mis} \) are the total number of features that are observed and missing, respectively, with \( p_{mis} = floor(0.5 \times p) \) and \( p_{obs} = p - p_{mis} \). Here, we fixed \( \phi_1 = 0 \), and drew nonzero values of \( \{\phi_2, \phi_3\} \) from the
log-normal distribution with mean $\mu_\phi = 5$, with log standard deviation $\sigma_\phi = 0.2$.

To evaluate the impact of the misspecification of the missingness mechanism on model performance, $r_{jm}$ was simulated under each mechanism as follows: (a) MCAR: $\{\phi_1, \phi_2, \phi_3\} = 0$ (b) MAR: Same as MCAR except $\phi_2j_m \neq 0$ for one completely-observed feature $j_m$ (c) MNAR: Same as MCAR except $\phi_3j_m \neq 0$ for one missing feature $j_m$. In this way, for each MAR or MNAR feature, the missingness is dependent on just one feature.

In each case, we used $\phi_0$ to control for an expected rate of missingness of 30% in each partially observed feature. We note that for each of these simulations, we use all features in $X$ as well as the response $Y$ as input into dlglm’s missingness network, although only one feature is involved under the true missingness model. Additionally, we searched for the optimal variational distributions of $q_{\phi}(Z|X^o)$ and $q_{\ell}(X^m|Z, X^o, R)$ from a class of Gaussian distributions with diagonal covariance structures, as discussed in Section 2.4.2. We fixed $K = 5$ during training, and increased $K$ to 500 at test time.

We vary $n$ and $d$ such that $n = \{10,000, 100,000\}$ and $d = \{2, 8\}$, and fix $p = 50$. We simulated 5 datasets per simulation condition, spanning various missingness mechanisms and values for $\{n, d\}$. We fix the values of $\beta$ at 0.25 for each feature, and adjusted $\beta_0$ to ensure equal proportions for the binary class response $Y$. For each simulation case, we partitioned the data into training, validation, and test sets with ratios of 8:1:1. For mice imputation, we averaged across 500 multiply-imputed datasets to obtain a single imputed dataset. We note that we generated $Y$ by a linear transformation of $X$ in these simulations in order to facilitate fair comparisons with mice, which cannot account for nonlinear relationships between the features and the response. Because no hyperparameter tuning is required, the validation set is not utilized for mice and mean imputation.

We measured the performance of each method with respect to three different tasks: imputation of missing values, coefficient estimation, and prediction. Imputation performance was measured with respect to the truth on a single imputed dataset by mean, dlglm and idlglm imputation, and on an average of multiply-imputed datasets by mice. Coefficient estimation for mean, miwae, notmiwae, and mice were based on downstream fitted GLM(s) on these imputed dataset(s), where estimates were pooled using Rubin’s rules (Rubin 2004) for mice. For dlglm and idlglm, we estimated the coefficients by the weights and bias $\beta$ of the last layer of the $s_{8,n}(\cdot)$ trained neural network. Here, we fixed the number of hidden layers in $s_{8,n}(\cdot)$ to 0 to allow for direct comparison with the other methods. A more complex prediction model via a neural network can be learned by simply incorporating additional hidden layers in $s_{8,n}(\cdot)$. We note that dlglm and idlglm can estimate $\beta$ without having to perform multiple imputation and downstream modeling unlike mice, where fitting complex methods such as neural networks each of the multiply imputed datasets separately may be computationally prohibitive.

After obtaining the coefficient estimates and trained models, we performed prediction on the test set in two ways: (a) using the incomplete (predI) test set, where the true values of $X^m$ are not known at prediction time, and (b) using the complete (predC) test set, where the true simulated values of $X^m$ are known at prediction time. These two ways reflect the two realistic cases in which (a) missingness is present during training time but complete data is available at prediction time, and (b) missingness is present during both training and prediction time. For predI, miwae, notmiwae, mice and mean imputation require an additional imputation step on the test set before predicting $Y$; for dlglm and idlglm, we simply input the incomplete test set into the trained model without needing to separately impute the test set, and we predict using the trained model. That is, miwae, notmiwae, mice and mean imputation cannot generalize the trained model to impute the test set, dlglm and idlglm provide a seamless framework to utilize the already-trained model to impute and predict on a held-out test set. For predC, we use the underlying true values of $X^m$ to predict on the test dataset.

Imputation error was measured by the average L1 distance between true and imputed masked values in $X$. Letting $\hat{X}^m$ denote the imputed masked values of the true $X^m$ values of the missing entries, we denote the average L1 distance is simply $\frac{\sum_{i=1}^{N_{miss}} |\hat{x}_{im} - x_{im}|}{N_{miss}}$, where $N_{miss}$ is the total number of missing entries in the dataset. Performance in coefficient estimation was measured by the average percent bias (PB) of the coefficient estimates compared to the truth, averaged across the $p$ features, that is, $PB = 100 \times \frac{1}{p} \sum_{j=1}^{p} \left| \frac{\hat{\beta}_j - \beta_j}{\beta_j} \right|$. Finally, predC and predI prediction error was measured by the average L1 distance between predicted and true values of the probabilities of class membership $\Pr(Y = 1|X)$ in the test set.

In order to assess the sensitivity of the performance of these methods to the specification of the missingness model used to synthetically mask the data, we also repeated the analyses on data with missingness mask simulated by the following: $\logit[p(r_{jm} = 1|x, y, \phi)] = \phi_0 + \phi_1 \log(x^o_i + \min(x^o_i)) + \phi_2 \log(x^o_i + \min(x^m_i))$, such that for the MAR and MNAR missingness cases, the missingness was dependent on the log of one of the completely or partially observed features. We denote this set of simulation conditions the “nonlinear missingness” case, where the missingness was simulated from the specified nonlinear logistic regression model. We show the results of this analysis in Appendix B2 of the supplementary materials.

### 3.1.2. Simulation Results

Figures 3 and 4 illustrate the simulation results pertaining to imputation accuracy, coefficient estimation, and prediction accuracy for the condition $p = 50$. We see that across all combinations of $\{n, d\}$ and mechanisms of missingness, mean imputation consistently performs poorly in imputation, coefficient estimation, and prediction, while mice and idlglm perform comparably in these metrics. Also, we note that under MNAR missingness, dlglm generally yields the lowest imputation and prediction error, as well as percent bias across all simulation cases. Under MAR missingness, dlglm performs comparably to idlglm and mice. This shows the ability of dlglm to learn an accurate model of the missingness, even under severe over-parametrization of the missingness model (model need not be
Figure 3. Simulation results with $n = 10,000$ and $p = 50$, varying $d = 2$ (top 4) and $d = 8$ (bottom 4). In each quadrant, we measure imputation accuracy by the average L1 distance between imputed versus true values in $X$ (top-left), coefficient estimation accuracy by the average percent bias (PB) of the estimates $\hat{\beta}$ compared to the truth (top-right), and prediction accuracy by the average L1 distance between the predicted and true probabilities of class 1 membership of $Y$ using the true unmasked test set (predC, bottom-left) and the incomplete test set (predI, bottom-right). In predI, we first impute missing values of the test data for mean, miwae, notmiwae, and mice imputation, and we input the incomplete test set as-is for dlglm and idlglm.

specified for ignorable missingness). However, due to the complexity of the model, we see that dlglm does generally perform poorly compared to idlglm and mice under MCAR missingness, when $n = 10,000$, although it still performs comparably to other
Figure 4. Simulation results with $n = 100,000$ and $p = 50$, varying $d = 2$ (top 4) and $d = 8$ (bottom 4). In each quadrant, we measure imputation accuracy by the average L1 distance between imputed versus true values in $X$ (top-left), coefficient estimation accuracy by the average percent bias (PB) of the estimates $\hat{\beta}$ compared to the truth (top-right), and prediction accuracy by the average L1 distance between the predicted and true probabilities of class 1 membership of $Y$ using the true unmasked test set (predC, bottom-left) and the incomplete test set (predI, bottom-right). In predI, we first impute missing values of the test data for mean, miwae, notmiwae, and mice imputation, and we input the incomplete test set as is for dlglm and idglm.

methods when the sample size is very large ($n = 100,000$). As one may expect, prediction performance using the incomplete data (predI) was poorer than prediction performance using the complete data (predC) for all methods.
We additionally show results pertaining to \( p = 25 \) in Appendix B2 of the supplementary materials. We similarly found that \( \text{dlglm} \) performed best under MNAR missingness, and comparatively to \( \text{idlgm} \), \( \text{mice} \), \( \text{miwae} \), and \( \text{notmiwae} \) under MCAR and MAR missingness.

### 3.2. Real Data with Simulated Missingness

Next, we analyzed three completely observed, large datasets from the UCI Machine Learning Repository (Dua and Graff 2017) that contained a specific response variable of interest, in order to preserve non-linearity and interactions between observed features. Unlike the simulated datasets, these UCI datasets don't follow a specific distribution that may be leveraged to inform a supervised learning method. The DRYBEAN dataset contains 16 features describing 13,611 images of dry beans taken with a high-resolution camera, and the response variable of interest was the type of dry bean each image represents, with seven different possible types of beans. The LETTER dataset contains 16 attributes of 20,000 black-and-white pixel images, each displaying one English letter (A to Z). Finally, the SHUTTLE dataset contains nine numerical attributes pertaining to 58,000 shuttle stat logs (observations), which are classified into seven different categories. Due to a low sample size in four of the seven categories, we pre-filtered the observations pertaining to these categories out of the dataset, and the resulting dataset contained 57,756 observations of three categories. In each of these datasets, the response variable was categorical with greater than two levels. Additional information regarding these datasets, and how to obtain the raw data files can be found in Appendix C of the supplementary materials.

We then simulated the missingness mask \( R^X \) with MCAR, MAR, and MNAR patterns of missingness in the manner described in Section 3.1.1. We split the samples in each dataset by a similar 8:1:1 ratio of training/validation/test samples. In the test set samples, we then imputed the missing values and predicted the response variables with each method in a manner similar to Section 3.1.1. For \( \text{dlglm} \) and \( \text{idlgm} \), we account for potential nonlinear relationships between the covariates and response by allowing the number of hidden layers in \( s_{\beta,n}(-) \) to be nonzero in hyperparameter tuning. Then, we compared imputation and prediction accuracy on each dataset, under each mechanism of missingness. Since the underlying true probabilities of class membership were unavailable, we measured prediction accuracy by the Cohen's kappa metric on the complete (kappaC) and incomplete (kappal) test set, with predicted class determined by the maximum predicted probability of membership. This metric measures how accurately a categorical variable was predicted, with a value of \(-1\) indicating worst possible performance, and a value of \(1\) indicating perfect concordance with the truth.

Results from the imputation and prediction analyses on these datasets can be found in Figure 5. We found that, as in the simulations, mean imputation performed most poorly in both imputation and downstream prediction, while \( \text{dlglm} \) tended to perform best in the MNAR cases, and performed comparatively to \( \text{mice} \) and \( \text{idlgm} \) under the MCAR and MAR cases. This further validates our claims under a more realistic setting, where the true data generation mechanism may be unknown. We also see that under both MCAR and MAR missingness, \( \text{mice} \) performed worse than \( \text{idlgm} \) in prediction on the LETTER and SHUTTLE datasets. The \( \text{mice} \) model has been known to break down under nonlinear relationships between the features (Van Buuren 2018), as may be the case in real-world datasets like the ones being examined. Using neural networks to model the data generation process allows \( \text{idlgm} \) to better model potential nonlinear relationships between features, allowing for more accurate prediction.

Interestingly, all of the algorithms performed similarly in prediction on the DRYBEAN dataset. We found that this dataset contained extremely high levels of correlation between the variables (see Web Appendix C of the supplementary materials). When features containing missingness are highly correlated to other fully-observed features, such missingness may not truly reflect the MNAR scenario (Hapfelmeier et al. 2012). This is because there exist fully-observed features that are highly correlated with the missing features, and ignorably missing data methods like \( \text{idlgm} \) may gather information about the missing entries from these correlated, fully observed features without having to explicitly model the mechanism of missingness. Still, \( \text{idlgm} \) and \( \text{dlglm} \) imputed missing entries much more accurately than \( \text{mean} \) and \( \text{mice} \) under MAR and MNAR. Interestingly, we also see that \( \text{idlgm} \) performed similarly to \( \text{dlglm} \) under MNAR in this dataset.

We additionally performed similar analyses on five other smaller UCI datasets, and these results can be found in Appendix D of the supplementary materials. We found that under small sample size settings, performance via \( \text{dlglm} \) was more variable under MNAR. We suggest use of \( \text{dlglm} \) when the data contains at least 10,000 samples, as the model may be too complex to be accurately trained under smaller sample sizes.

### 3.3. Bank Marketing Dataset

Finally, we performed prediction on the Bank Marketing dataset from the UCI Machine Learning Repository. This dataset contained 41,188 observations of 20 different attributes that were obtained based on direct phone calls from a Portuguese banking institution as part of a promotion campaign for a term deposit subscription (Moro, Cortez, and Rita 2014). The response variable of interest was a fully-observed binary measure of whether the client subscribed a term deposit. Of the 20 attributes, we removed 1 attribute as directed from the manual due to perfect correlation with the response variable, and removed three other attributes that were deemed irrelevant to the prediction task: month of contact, day of contact, and communication type (cell phone or telephone).

Missingness was present in 8 of the 16 attributes: type of job, marital status, level of education, whether the client had a credit in default, whether the client had a housing loan, whether the client had a personal loan, number of days since the client was contacted in a previous campaign, and outcome of the previous campaign. The remaining 8 attributes were fully-observed: age of client, number of contacts during this campaign, number of contacts before this campaign, employment variation rate, consumer price index, consumer confidence index, euribor 3 month rate, and employee number. The global rate of missingness was about 13.3%. The response variable of interest was collected by additional follow-up calls to confirm whether the client...
Figure 5. Imputation (top row) and prediction results from predC (middle row) and predI (bottom row) from comparative methods run on 3 large datasets from the UCI Machine Learning Repository: DRYBEAN, LETTER, and SHUTTLE (columns, left to right). Imputation error was measured by the average L1 distance between true and imputed entries, with lower values indicating better performance, and prediction performance was measured by the Cohen’s kappa metric for both predC (kappaC) and predI (kappaI), with higher values indicating better performance.
Table 1. Results from prediction analyses on the Bank Marketing dataset from the UCI Machine Learning Repository.

|          | AUC  | PPV  | kappa | F1   |
|----------|------|------|-------|------|
| dlglm    | 0.778| 0.475| 0.397 | 0.470|
| dlglm_y | 0.880| 0.481| 0.445 | 0.516|
| idglm    | 0.791| 0.475| 0.407 | 0.460|
| rowcolor | 0.779| 0.446| 0.411 | 0.488|
| mean     | 0.769| 0.448| 0.385 | 0.46  |
| mice     | 0.771| 0.455| 0.396 | 0.471|

NOTE: We measured concordance between the true and predicted binary response by four metrics: Area Under ROC Curve (AUC), Positive Predictivity (PPV), Cohen's kappa (kappa), and F1 score. For dlglm and idglm, Y was included in the input for neural networks \( g_y(\cdot) \) and \( h_\theta(\cdot) \), in order to allow for more accurate imputed values. The highlighted values are the maximum values for each of the diagnostic measures.

subscribed to the product. Additional information regarding the bank marketing dataset, and how to obtain the raw data files can be found in Appendix E of the supplementary materials.

This type of dataset reflects the most realistic situation in practice, where missingness exists in a dataset and one has no prior knowledge of either the relationships between the features and the response, or the underlying mechanism of the missingness. We divided the dataset into the 8:1:1 training, validation, and test set ratio, and performed prediction as before. Because neither the data nor the missingness was simulated, we compared just the predl prediction performance across the methods.

In order to more deeply dive into this real data example, we assessed the prediction performance for dlglm and idglm in the context of prediction (excluding Y from neural networks \( g_y(\cdot) \) and \( h_\theta(\cdot) \)) and imputation (including Y, denoted by dlglm_y and idglm_y).

Table 1 shows the results from these prediction analyses. We measured prediction performance of the binary response variable by 4 metrics: Area Under the ROC Curve (AUC), Positive Predictivity (PPV), Cohen's kappa (kappa), and the F1 metric. The formulas for PPV and F1 metrics are given in Appendix F of the supplementary materials. For each metric, a larger value represents greater concordance between the true and predicted response. We see that although dlglm_y yielded a significantly greater performance in prediction via all metrics, dlglm does not significantly outperform idglm. The similar performance between dlglm and idglm may indicate that the real mechanism of missingness in this data may not be MNAR, although this claim is not testable in practice.

Additionally, the trained dlglm model chose 0 hidden layers in the \( s_{\beta,\pi}(X) \) neural network in the optimal model, such that \( s_{\beta,\pi}(X) \rightarrow s_{\beta}(X) \). Therefore, the weights of that neural network \( \beta \) coincide exactly with the coefficient estimates of a classic generalized linear model, that is, logit\( [\text{Pr}(Y = 1)] = X\beta \). The features in the dataset with the largest effects on the probability of a client subscribing to a term deposit were employment variation rate (0.538), age of client (0.508), and whether the client had a personal loan (-0.402). Specifically, a client was more likely to subscribe if the company experienced higher levels of variation in employment and if the client were older, while a client was less likely to subscribe if they had a personal loan.

4. Discussion

In this article, we introduced a novel deep learning method called Deeply learned Generalized Linear Model with Missing Data (dlglm), which is able to perform coefficient estimation and prediction in the presence of missing not at random (MNAR) data. dlglm uses a deep learning neural network architecture to model the generation of the data matrix \( X \), as well as the relationships between the response variable \( Y \) and \( X \) and between the missingness mask \( R \) and \( X \). In this way, we are able to (a) generalize the traditional GLM to account for complex nonlinear interactions between the features, and (b) account for ignorable and non-ignorable forms of missingness in the data. We also showed through simulations and real data analyses that dlglm performs better in coefficient estimation and prediction in the presence of MNAR missingness than other impute-then-regress methods, like mean and mice imputation. Furthermore, we found that dlglm was generally robust to the mechanism of missingness, performing comparably well to mice and idglm under MCAR and MAR settings. Still, it is recommended to utilize idglm when assuming the missingness is ignorable, given that the missingness model that is learned in dlglm is not necessary in this setting.

Supervised learning algorithms such as dlglm and idglm can be particularly useful in analyzing real-life data in the presence of missingness. In reality, the mechanism underlying missing values cannot be explicitly known or tested, but dlglm may allow flexibility to evaluate multiple assumptions regarding the missingness mechanism. Furthermore, whereas impute-then-regress methods may typically require fully observed observations at test time for prediction, dlglm and idglm can predict the response of interest using partially observed observations. This provides a convenient workflow, where a user need not separately re-impute the prediction set at test time.

In this article, we focused specifically on the case of univariate response \( Y \). dlglm and idglm can be generalized to the multivariate \( Y \) case by (a) including \( Y \) in the existing IWAE structure and (b) expanding the neural network \( s_{\beta,\pi}(x_i) \) to account for all \( q \) responses in \( Y \), and utilizing samples of \( Z \) as additional input into this network such that \( s_{\beta,\pi}(x_i) \rightarrow s_{\beta,\pi}(x_i, z) \). By doing this, we account for multivariate \( Y \), outputting additional parameters pertaining to the newly-specified distribution of \( p_{\theta,\pi}(y_i|x_i,z_i) \) and modeling correlation of \( Y \) by the learned latent structure. We leave this as an extension of our method.

Supplementary Materials

Supplementary Materials: Additional details of the dlglm algorithm and the datasets used in this article. (pdf)

R-package for dlglm: R-package dlglm containing code to perform the diagnostic methods described in the article. The package can be downloaded from https://github.com/DavidKLim/dlglm (website)

R Paper repo for reproducibility: Github repository to replicate all analyses from this article can be found here: https://github.com/DavidKLim/dlglm_Paper (website)

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

The authors gratefully acknowledge NIH grants U01-CA274298, P50-CA257911, P50-CA058223, T32-CA106209, 1R01AA02687901A1, and
References

Beesley, L. J., Taylor, J. M., and Little, R. J. (2019), “Sequential Imputation for Models with Latent Variables Assuming Latent Ignorability,” Australian & New Zealand Journal of Statistics, 61, 213–235. [643]

Bottou, L. (2012), “Stochastic Gradient Descent Tricks,” in Beesley, L. J., Taylor, J. M., and Little, R. J. (2019), “Sequential Imputation for Missing Data in Supervised Deep Learning,” in Advances in Neural Information Processing Systems, 34, 27645–27658. [643]

Cremer, C., Morris, Q., and Duvenaud, D. (2017), “Reinterpreting Importance-Weighted Autoencoders,” arXiv e-prints p. arXiv:1704.02916. [642]

Dormehl, L. (2019), “What is An Artificial Neural Network? Here's Everything You Need to Know,” available at https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/ [639]

Du, D., and Graff, C. (2017), “UCI Machine Learning Repository,” available at http://archive.ics.uci.edu/ml/ [647]

Gershman, S. J., and Goodman, N. D. (2014), “Amortized Inference in Probabilistic Reasoning,” in CoGSci. [641]

Holland, P. W., and Welsch, R. E. (1977), “Robust Regression Using Iteratively Reweighted Least-Squares,” Communications in Statistics - Theory and Methods, 6, 813–827. [639]

Hoogland, J., Barzveled, M., Debray, T. P. A., Reitsma, J. B., Verstraalen, T. E., Dijkgraaf, M. G. W., and Zwinderman, A. H. (2020), “Handling Missing Predictor Values When Validating and Applying a Prediction Model to New Patients,” Statistics in Medicine, 39, 3591–3607. [638]

Ibrahim, J. G., Chen, M.-H., Lipsitz, S. R., and Herring, A. H. (2005), “Missing-Data Methods for Generalized Linear Models,” Journal of the American Statistical Association, 100, 332–346. [638,640]

Ibrahim, J. G., and Molenberghs, G. (2009), “Missing Data Methods in Longitudinal Studies: A Review,” TEST, 18, 1–43. [638]

Ipsen, N. B., Mattei, P.-A., and Freiljen, J. (2021), “How to Deal with Missing Data in Supervised Deep Learning?” in International Conference on Learning Representations. [638,642,643]

Kingma, D. P., and Ba, J. (2014), “Adam: A Method for Stochastic Optimization,” arXiv e-prints p. arXiv:1412.6980. [642]

Kingma, D. P., and Welling, M. (2013), “Auto-Encoding Variational Bayes,” arXiv e-prints p. arXiv:1312.6114. [641]

Li, Y., Akbar, S., and Oliva, J. (2020), “Aclflow: Flow Models for Arbitrary Conditional Likelihoods,” in International Conference on Machine Learning, PMLR, pp. 5831–5841. [641]

Lim, D. K., Rashid, N. U., Oliva, J. B., and Ibrahim, J. G. (2021), “Handling Non-ignorably Missing Features in Electronic Health Records Data Using Importance-Weighted Autoencoders,” arXiv e-prints p. arXiv:2101.07357. [638,642]

Lipsitz, S. R., and Ibrahim, J. G. (1996), “A Conditional Model for Incomplete Covariates in Parametric Regression Models,” Biometrika, 83, 916–922. [640]

Little, R. J. A., and Rubin, D. B. (2002), Statistical Analysis with Missing Data, Hoboken, NJ: Wiley. [640]

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018), “Deep Generative Modeling for Single-Cell Transcriptionomics,” Nature Methods, 15, 1053–1058. [638]

Ma, C., and Zhang, C. (2021), “Identifiable Generative Models for Missing not at Random Data Imputation,” Advances in Neural Information Processing Systems, 34, 27645–27658. [643]

Matei, P.-A., and Freiljen, J. (2019), “MFWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets,” in Proceedings of the 36th International Conference on Machine Learning, Vol. 97 of Proceedings of Machine Learning Research, eds. K. Chaudhuri and R. Salakhutdinov, pp. 4413–4423, PMLR, Long Beach, California, USA. [639]

McCullagh, P., and Nelder, J. A. (2019), Generalized Linear Models, Boca Raton, FL: Routledge. [639]

Moro, S., Cortez, P., and Rita, P. (2014), “A Data-Driven Approach to Predict the Success of Bank Telemarketing,” Decision Support Systems, 62, 22–31. [647]

Murphy, J. (2016), An Overview of Convolutional Neural Network Architectures for Deep Learning, pp. 1–22, Plymouth, MA: Microway Inc. [641]

Nelder, J. A., and Wedderburn, R. W. M. (1972), “Generalized Linear Models,” Journal of the Royal Statistical Society, Series A, 135, 370–384. [639]

Prechtl, L. (1998), “Early Stopping-but When?” in Neural Networks: Tricks of the Trade, eds. G. Montavon, G. B. Orr, and K.-R. Müller, pp. 1–22, Berlin: Springer. [641]

Qi, M., and Wu, Y. (2003), “Nonlinear Prediction of Exchange Rates with Monetary Fundamentals,” Journal of Empirical Finance, 10, 623–640. [640]

Razzak, M. I., Naz, S., and Zaib, A. (2017), “Deep Learning for Medical Image Processing: Overview, Challenges and the Future,” in Classification in BioApps: Automation of Decision Making, Lecture Notes in Computational Vision and Biomechanics, eds. N. Dey, A. S. Ashour, and S. Borra, pp. 323–350, Cham: Springer. [638]

Rubin, D. B. (1976), “Inference and Missing Data,” Biometrika, 63, 581–592. [638]

——— (2004), Multiple Imputation for Nonresponse in Surveys (Vol. 81), Hoboken, NJ: Wiley. [644]

Saxe, A. M., Mcclelland, J. L., and Ganguli, S. (2014), “Exact Solutions to the Nonlinear Dynamics of Learning in Deep Linear Neural Network,” in International Conference on Learning Representations. [641]

Strauss, R., and Oliva, J. B. (2021), “Arbitrary Conditional Distributions with Energy,” Advances in Neural Information Processing Systems, 34, 752–763. [641]

——— (2022), “Posterior Matching for Arbitrary Conditioning,” in Advances in Neural Information Processing Systems (Vol. 35), pp. 18088–18099. [641]

Stubbendick, A. L., and Ibrahim, J. G. (2003), “Maximum Likelihood Methods for Nonignorable Missing Responses and Covariates in Random Effects Models,” Biometrics, 59, 1140–1150. [640]

Svozil, D., Kvasnicka, V., and Pospichal, J. (1997), “Introduction to Multi-Layer Feed-Forward Neural Networks,” Chemometrics and Intelligent Laboratory Systems, 39, 43–62. [638]

Tran, M.-N., Nguyen, N., Nott, D., and Kohn, R. (2019), “Bayesian Deep Net GLM and GLMM,” Journal of Computational and Graphical Statistics, 29, 97–113. [638,639]

Van Buuren, S. (2018), Flexible Imputation of Missing Data, Boca Raton, FL: CRC Press. [647]

Van Buuren, S., and Groothuis-Oudshoorn, K. (2011), “mice: Multivariate Imputation by Chained Equations in R,” Journal of Statistical Software, 45, 1–67. [643]

Wells, B. J., Nowacki, A. S., Chagin, K., and Kattan, M. W. (2013), “Strategies for Handling Missing Data in Electronic Health Record Derived Data,” eGEMs (Generating Evidence &: Methods to Improve Patient Outcomes), 1, 7. [638]