Exploiting Missing Data Remediation Strategies Using Adversarial Missingness Attacks

Deniz Koyuncu[1*], **Alex Gittens**[1], **Bülent Yener**[1*], **Moti Yung**[2,3]
[1]Rensselaer Polytechnic Institute
[2]Google LLC
[3]Columbia University
deniz_kync@icloud.com

Abstract

Adversarial Missingness (AM) attacks aim to manipulate model fitting by carefully engineering a *missing* data problem to achieve a specific malicious objective. AM attacks are significantly different from prior data poisoning attacks in that no malicious data inserted and no data is maliciously perturbed. Current AM attacks are feasible only under the assumption that the modeler (victim) uses full-information maximum likelihood methods to handle missingness. This work aims to remedy this limitation of AM attacks; in the approach taken here, the adversary achieves their goal by solving a bi-level optimization problem to engineer the adversarial missingness mechanism, where the lower level problem incorporates a differentiable approximation of the targeted missingness remediation technique. As instantiations of this framework, AM attacks are provided for three popular techniques: (i) complete case analysis, (ii) mean imputation, and (iii) regression-based imputation for general *empirical risk minimization* (ERM) problems. Experiments on real-world data show that AM attacks are successful with modest levels of missingness (less than 20%). Furthermore, we show on the real-world *Twins* dataset that AM attacks can manipulate the estimated average treatment effect (ATE) as an instance of the general ERM problems: the adversary succeeds in not only reversing the sign, but also in substantially inflating the ATE values from a true value of -1.61% to a manipulated one as high as 10%. These experimental results hold when the ATE is calculated using multiple regression-based estimators with different architectures, even when the adversary is restricted to modifying only a subset of the training data. The goals of this work are to: (i) establish the vulnerability to AM attacks of a significantly wider class of missingness remediation strategies than established in prior work, and (ii) brings the AM threat model to the attention of the community, as there are currently no defense strategies for these attacks.

Code — https://github.com/cruyffturn/AM-AAAI26
Extended version — https://arxiv.org/abs/2409.04407

1 Introduction

Missing data is ubiquitous in real-world datasets, and recent machine learning work has renewed focus on handling it. Meanwhile, adversarial and data poisoning attacks have gained attention, but the notion that missingness itself could be adversarial remains largely unexplored. Existing poisoning models typically assume the attacker perturbs or injects data, whereas adversarial missingness (AM) harms by *selectively omitting data*— a distinct and orthogonal attack vector. Standard defenses (e.g., data sanitization and outlier detection, statistically robust training) do not apply, as AM involves no modification of the observed data and can mimic naturally occurring missing-not-at-random (MNAR) patterns.

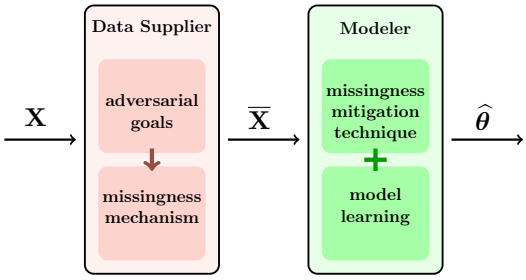

Figure 1: The AM threat model: the adversary adds missingness to the initial dataset $\boldsymbol{X}$ to obtain $\bar{\boldsymbol{X}}$, and the modeler mitigates the missingness to learn a model $\hat{\boldsymbol{\theta}}$. Because the adversary carefully engineered the missingness mechanism with a malicious goal in mind, $\hat{\boldsymbol{\theta}}$ is steered to minimize the adversarial objective $g(\cdot; \boldsymbol{X})$.

AM attacks, first explored in prior work (Koyuncu et al. 2023, 2024), model a setting where an adversary (as data supplier) omits a subset of the training data before it is processed by a modeler using a missingness remediation technique (e.g., imputation); see Figure 1 for a visualization of the threat model. Existing AM attacks (Koyuncu et al. 2023, 2024) have very narrow applicability, as they were introduced specifically to manipulate the learning of Gaussian Structural Causal Models when Full Information Maximum Likelihood (FIML) is used by the modeler as the missing data remediation strategy. FIML makes assumptions on the form of the joint distribution of the features, X, and the response, Y; this is appropriate when learning causal models,

[*]This work was done in part while the author was visiting Google LLC, New York, NY.

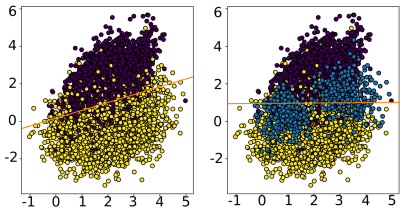

Figure 2: Example of manipulating a logistic regression model for a classification problem. By omitting the x-coordinate of 8.4% of the samples (colored blue), the adversary rotates the optimal decision boundary (left figure) to a horizontal line (right figure) under mean imputation. The classification accuracy decreased by 0.4% but with high confidence the modeler asserts that the x variable has a coefficient close to zero (p value=0.688).

but is overly restrictive in most other learning settings. For example, to apply previous AM attacks to the setting of discriminative learning, one must assume that Y and the features of X are jointly Gaussian. Thus these attacks are not applicable when Y is discrete (as in logistic regression); or when any of the features of X are categorical, have multiple modes, are always positive, and so on.

This paper introduces a general framework for AM attacks on empirical risk minimization (ERM) tasks with differentiable losses, without assuming specific distributions or model architectures. We instantiate this framework for widely-used remediation strategies: complete case analysis, mean imputation, and regression-based imputation, framing the attacker's objective as a bi-level optimization problem using differentiable proxies. Figure 2 shows an example of an AM attack on logistic regression with mean imputation developed using this framework— a learning setting that cannot be addressed using previous AM frameworks.

Our contributions are: (i) a general bi-level framework for AM attacks on differentiable ERM problems; (ii) differentiable approximations of several popular missingness mitigation techniques; (iii) empirical results showing successful manipulation of model behavior—including feature importance and treatment effects—using real-world data sets. Notably, our attacks often transfer across models and mitigation strategies.

Our empirical examples target tabular data sets, as missingness is natural in such data sets. We defer extensions to other modalities (e.g., deep nets, images) and efficiency improvements to future work. Our results expose a vulnerability in standard ML pipelines that handle missing data, motivating the need for defenses against AM attacks.

Notation The random vector corresponding to the training data is $X = (X_1, \ldots, X_d)$, while a realization of that random vector (i.e. an instance of training data) is denoted by $\mathbf{x} = (x_1, \ldots, x_d)$; the rows of the matrix $\boldsymbol{X} \in \mathbb{R}^{N \times d}$ constitute the training data $\{\mathbf{x}_i\}_{i=1}^N$. Similarly, the masking matrix $\boldsymbol{R} \in \{0,1\}^{N \times d}$ introduced in the next section comprises rows of realizations $\mathbf{r}_i$ of instances of the random vector $R = (R_1, \ldots, R_d)$. Probability density functions (pdf) parameterized by a parameter $\boldsymbol{\theta}$ are denoted by $\mathrm{p}(\mathbf{x}; \boldsymbol{\theta})$ (which random variable is under consideration will be clear from context), and similarly, parameterized conditional pdfs are denoted using notation like $\mathrm{p}_{R|X}(\mathbf{r} \mid \mathbf{x}; \boldsymbol{\phi})$.

2 Problem Formulation

In AM attacks, the adversary is restricted to hiding existing entries. We indicate the entries of $\boldsymbol{X}$ to be made missing with the binary masking matrix $\boldsymbol{R} \in \{0,1\}^{N \times d}$. After selecting the masking matrix, the adversary hides the corresponding entries of the original matrix to obtain the matrix $\bar{\boldsymbol{X}}$ of partially observed data: $\bar{\boldsymbol{X}}_{i,j} = \text{NA}$ if $\boldsymbol{R}_{i,j} = 0$ while $\bar{\boldsymbol{X}}_{i,j} = \boldsymbol{X}_{i,j}$ if $\boldsymbol{R}_{i,j} = 1$.

The modeler uses the resulting partially observed dataset $\bar{\boldsymbol{X}}$ to learn a model by minimizing an objective function f over a set of feasible parameters Θ:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \Theta} f(\boldsymbol{\theta}; \bar{\boldsymbol{X}}) \tag{1}$$

The modeler's objective f accomplishes both the missingness mitigation and the learning of the model. For example, (Koyuncu et al. 2024) focuses on the case where the modeler fits a Gaussian generative model using the observed portion of the dataset, $\bar{\boldsymbol{X}}_{\text{obs}}$, and the objective is $f(\boldsymbol{\theta}; \bar{\boldsymbol{X}}) = -\log \mathrm{p}(\bar{\boldsymbol{X}}_{\text{obs}}; \boldsymbol{\theta})$. In the more general learning setup considered in this paper, f comprises the combination of a missingness mitigation technique (complete case analysis, mean imputation, or conditional mean imputation) with an arbitrary differentiable empirical risk minimization objective.

The model resulting from training, $\hat{\boldsymbol{\theta}}$, depends on which entries were observed. The goal of the adversary is to steer $\hat{\boldsymbol{\theta}}$ towards an adversarial model $\boldsymbol{\theta}_\alpha$ selected to achieve a specific adversarial objective, while introducing a minimal amount of missingness. The adversary may also select $\boldsymbol{\theta}_\alpha$ to limit the occurrence of auditable outcomes such as a low predictive performance or unintended differences between the models learned with and without adversarial manipulation. The central challenge in accomplishing the adversary's goal lies in effectively searching through the set of possible masking strategies, which has size $2^{N \times d}$.

To bypass this combinatorial search problem, we replace the selection of the deterministic masking matrix $\boldsymbol{R}$ with the selection of a *missingness mechanism* $\mathrm{p}_{R|X}$ that the adversary then uses to probabilistically generate missingness masks for the individual rows of $\boldsymbol{X}$. Given the ith observation, $\mathbf{x}^{(i)}$, the adversary samples the ith row of the masking matrix, $\mathbf{r}^{(i)}$, proportional to $\mathrm{p}_{R|X}(\cdot \mid \mathbf{x}^{(i)})$. We learn the adversarial missingness mechanism (AM mechanism) by solving a bi-level optimization problem in which the lower level problem corresponds to the modeler's objective and the upper level problem models the adversary's intent:

$$\begin{aligned} \min_{\mathrm{p}_{R|\boldsymbol{X}}} \; & g(\tilde{\boldsymbol{\theta}}; \boldsymbol{X}) + \lambda \cdot \Omega(\mathrm{p}_{R|X}; \boldsymbol{X}) \\ \text{s.t.} \quad & \tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \Theta} \tilde{f}(\boldsymbol{\theta}, \mathrm{p}_{R|X}; \boldsymbol{X}) \end{aligned} \tag{2}$$

The parameter learned in the upper level is the AM mechanism $\mathrm{p}_{R|X}$; given this AM mechanism, the lower level problem returns an approximation $\tilde{\boldsymbol{\theta}}$ to the model that the modeler would learn using the combination of their missingness

mitigation and learning objective when given training data that was masked using $\mathrm{p}_{R|X}$.

In the upper-level problem, the objective $g(\tilde{\boldsymbol{\theta}}; \boldsymbol{X})$ captures the adversary's intent. For example, given a specific adversarial model $\boldsymbol{\theta}_\alpha$, the adversary may take $g(\tilde{\boldsymbol{\theta}}; \boldsymbol{X}) = \|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}_\alpha\|_2^2$; if the goal were instead to cause a linear model to make specific predictions on each datapoint, an appropriate choice would be $g(\tilde{\boldsymbol{\theta}}; \boldsymbol{X}) = \|\boldsymbol{X}\tilde{\boldsymbol{\theta}} - \mathbf{y}_{\text{target}}\|_2^2$. The regularization term Ω ensures that the learned missingness mechanism has a low missingness rate. See Section 4 for the specific form of Ω, and Section 5 for two examples of g tailored respectively to reducing variable importance and manipulating average treatment effect estimation. Tuning λ balances between achieving the adversarial objective and lowering the missingness rate.

In general the modeler's objective f is *not* differentiable with respect to the missingness mechanism, so to facilitate bi-level learning with gradient methods, the lower-level problem simulates the modeler's training process using an approximation $\tilde{f}$ to the modeler's objective function f that *is* differentiable with respect to the missingness mechanism.

(Koyuncu et al. 2023, 2024) pioneered the use of missingness mechanisms to avoid the combinatorial search problem; these works developed AM attacks on Gaussian causal models learned using FIML. Our innovations are: (i) the introduction of a bilevel formulation for learning the missingness mechanism, as this allows the attacking of arbitrary ERM learning with differentiable objectives, and (ii) the development of effective differentiable proxies $\tilde{f}$ for common missingness remediation strategies used by modelers.

3 Differentiable Proxy Objectives for Missing Data Remediation

In this section we provide one of the major contribution of this work: novel differentiable proxies $\tilde{f}$ for complete-case analysis (CCA) and mean imputation; using similar ideas, a proxy for conditional mean imputation (using linear regression) is developed in Appendix B. These proxies allow the adversary to use (equation 2) to learn adversarial missingness mechanisms targeting these missingness mitigation techniques.

To design differentiable approximations to the modeler's objective, we assume that the modeler's goal, given a completely observed data set $\boldsymbol{X}$, is to solve the empirical risk minimization problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{i=1}^{N} J(\mathbf{x}^{(i)}; \boldsymbol{\theta}) \tag{3}$$

for a specified score function. For instance, if the modeler's goal is to fit a logistic regression model on data $\mathbf{x}^i = (\boldsymbol{\omega}_i, y_i)$, then $J(\mathbf{x}^i; \boldsymbol{\theta}) = -\log(1 + \exp(-y_i \boldsymbol{\theta}^T \boldsymbol{\omega}_i))$.

From here on, to simplify the presentation we assume that the missingness mechanism $\mathrm{p}_{R|X}$ is completely determined by a parameter vector $\boldsymbol{\phi}$, so that "differentiability with respect to the missingness mechanism" means differentiability with respect to $\boldsymbol{\phi}$ and learning $\mathrm{p}_{R|X}$ reduces to a finite-dimensional optimization problem.

3.1 Complete-Case Analysis

In CCA (Little and Rubin 2002), the modeler discards all rows with missing entries before proceeding with their analysis. Denoting the set of completely observed rows by $\mathcal{S} = \{i : \mathbf{r}^{(i)} = \mathbf{1}\}$, the modeler's objective (equation 1) is to learn $\hat{\boldsymbol{\theta}}$ to minimize the average score of the completely observed examples:

$$f(\boldsymbol{\theta}; \bar{\boldsymbol{X}}) = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} J(\mathbf{x}^{(i)}; \boldsymbol{\theta}) \tag{4}$$

The random set $\mathcal{S}$ that the modeler has to work with is determined by the adversary's missingness mechanism. The missing data masks are *sampled* from the AM mechanism, so the objective function (equation 4) is not differentiable with respect to $\boldsymbol{\phi}$, and cannot be used in the bi-level formulation used to learn the AM mechanism.

However, because the training samples are i.i.d., as the number of samples goes to infinity, the weak law of large numbers asserts that, for any $\boldsymbol{\theta}$, the modeler's objective (equation 4) converges in probability to the expected score of $\boldsymbol{\theta}$ conditioned on all entries in X being observed.

$$\begin{aligned} f(\boldsymbol{\theta}; \bar{\boldsymbol{X}}) &\xrightarrow{P} \mathbb{E}[J(X; \boldsymbol{\theta}) \mid R = \mathbf{1}] \\ &= \frac{1}{\mathbb{P}_{R;\boldsymbol{\phi}}(\mathbf{1})} \int J(X; \boldsymbol{\theta}) \mathrm{p}_{R|X}(\mathbf{1} \mid \mathbf{x}; \boldsymbol{\phi}) \mathrm{p}_X(\mathbf{x})\, d\mathbf{x}. \end{aligned} \tag{5}$$

The equality follows from an application of Bayes' Theorem.

Equation 5 clarifies that, under CCA, the asymptotic impact of the missingness mechanism is to weigh the score function by the missingness mechanism. Larger weights are given to observations that are likely to be completely observed under the missingness mechanism. The asymptotic objective in (equation 5) is differentiable with respect to the missingness mechanism, but cannot be evaluated on finite training data.

To obtain the final proxy objective that is both differentiable with respect to the missingness mechanism and can be evaluated on finite training data, we take $\tilde{f}$ to be the approximation of the expectation in (equation 5) on the training data:

$$\tilde{f}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}) = \frac{1}{\mathbb{P}_{R;\boldsymbol{\phi}}(\mathbf{1})} \frac{1}{N} \sum_{i=1}^{N} \mathrm{p}_{R|X}(\mathbf{1} \mid \mathbf{x}^{(i)}; \boldsymbol{\phi}) J(\mathbf{x}^{(i)}; \boldsymbol{\theta}).$$

Directly choosing an adversarial mask matrix $\boldsymbol{R}$ to attack a CCA modeler would require searching over the 2^N possible subsets of observed rows; while the bi-level probabilistic formulation of AM attacks on CCA modelers using this proxy objective involves optimizing over an inner objective function that is computable in time linear in the number of training observations.

3.2 Missing Data Imputation

When the modeler uses imputation, the missing entries in data matrix $\bar{\boldsymbol{X}}$ are imputed to obtain a completed matrix, which we denote by $\hat{\boldsymbol{X}}$. As a popular example, consider

mean imputation; here, the missing entries in the jth column are replaced with the average of the observed entries in that column, denoted by $\hat{\boldsymbol{\mu}}_j \in \mathbb{R}$. Consequently, $\hat{\boldsymbol{X}}_{i,j} = \hat{\boldsymbol{\mu}}_j$ if $\mathbf{r}_{i,j} = 0$ while $\hat{\boldsymbol{X}}_{i,j} = \boldsymbol{X}_{i,j}$ if $\mathbf{r}_{i,j} = 1$. After the imputation, the modeler uses the resulting complete dataset to train a model, so the objective of a modeler using mean imputation is given by

$$f(\boldsymbol{\theta}; \bar{\boldsymbol{X}}) = \frac{1}{N} \sum_{i=1}^{N} J(\hat{\mathbf{x}}^{(i)}; \boldsymbol{\theta}). \tag{6}$$

Unlike CCA, in general, imputation introduces dependence between the initially independent observations. Consequently, the weak law of large numbers cannot be readily used to obtain a differentiable proxy function for (equation 6).

To find such a proxy function, we initially assume the imputation model is not learned from data, but instead is fixed *a priori*. In this case, the rows of $\hat{\boldsymbol{X}}$ are i.i.d., and a similar argument to before gives the asymptotic behavior

$$\begin{aligned} f(\boldsymbol{\theta}; \bar{\boldsymbol{X}}) &\xrightarrow{P} \mathbb{E}_{X,R,\hat{X}}[J(\hat{X}; \boldsymbol{\theta})] \\ &= \mathbb{E}_X \left[\sum_{\mathbf{r}} \mathrm{p}_{R|X}(\mathbf{r} \mid X; \boldsymbol{\phi}) \mathbb{E}_{\hat{X}|X,R=\mathbf{r}}[J(\hat{X}; \boldsymbol{\theta})] \right], \end{aligned} \tag{7}$$

where $\mathbf{r} \in \{0,1\}^d$ varies over the missingness masks that have nonzero probability under the missingness mechanism $\mathrm{p}_{R|X}$, and the conditional pdf $\mathrm{p}_{\hat{X}|X,R}$ denotes the fixed imputation model that takes in a row of incompletely observed entries and imputes the missing entries. This gives us a differentiable approximation with respect to $\boldsymbol{\phi}$. To evaluate it with finite-data, we empirically approximate the two remaining expectations:

$$\tilde{f}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{\mathbf{r} \neq \mathbf{0}} \mathrm{p}_{R|X}(\mathbf{r} \mid \mathbf{x}^{(i)}; \boldsymbol{\phi}) J(\hat{\mathbf{x}}^{(i,\mathbf{r})}; \boldsymbol{\theta}), \tag{8}$$

where $\hat{\mathbf{x}}^{(i,\mathbf{r})}$ is sampled[1] proportionally to the fixed imputation model $\mathrm{p}_{\hat{X}|X,R}(\cdot \mid \mathbf{x}^{(i)}, \mathbf{r})$

In the worst case, when the missingness mechanism allows masking each d feature of X, evaluating this proxy objective requires $N2^d$ summations; this is a significant reduction in complexity compared to directly looking for an adversarial mask, which searches over a search space of size $2^{N \times d}$. However, we reached (equation 8) by assuming that the imputation mechanism is fixed before seeing the data $\overline{\boldsymbol{X}}$. In practice, the imputation depends on the observed data, and thus on the missingness mechanism.

To capture that dependence, we propose to use the asymptotic forms of the imputation methods to derive expressions for $\hat{\mathbf{x}}^{(i,\mathbf{r})}$ that are differentiable with respect to the missingness mechanism; then we use (equation 8) with these $\hat{\mathbf{x}}^{(i,\mathbf{r})}$. In the subsequent part, we derive the asymptotic form of the commonly used mean imputation. The asymptotic form of linear regression-based imputation is provided in the Appendix B.

[1]For simplicity, a single sample is used to approximate the inner-most expectation; more could be employed.

Mean Imputation By the weak law of large numbers, the imputed value of an unobserved entry in the jth column converges to the conditional mean of the jth variable given that this variable is observed, i.e. $\mathbb{E}[X_j \mid R_j = 1]$. This conditional expectation can be expressed in terms of the missingness mechanism as follows:

$$\mathbb{E}[X_j \,|\, R_j = 1] = \frac{1}{\mathbb{P}_{R_j;\boldsymbol{\phi}}(1)} \mathbb{E}\left[X_j \sum_{\forall \mathbf{r}, \mathbf{r}_j = 1} \mathrm{p}_{R|X}(\mathbf{r} \mid X; \boldsymbol{\phi}) \right]. \tag{9}$$

See Proposition 1 in the Appendix for a proof.

The resulting imputed vector of the ith sample with missingness mask $\mathbf{r}$ is denoted by $\hat{\mathbf{x}}^{(i,\mathbf{r})} \in \mathbb{R}^d$, and its elements satisfy:

$$\hat{\mathbf{x}}_j^{(i,\mathbf{r})} = \mathbf{r}_j \mathbf{x}_j^{(i)} + (1 - \mathbf{r}_j) \hat{\boldsymbol{\mu}}_j(\boldsymbol{\phi}). \tag{10}$$

Here, $\hat{\boldsymbol{\mu}}_j(\boldsymbol{\phi})$ denotes a finite data approximation of the conditional expectation in (equation 9). To compute $\hat{\boldsymbol{\mu}}_j(\boldsymbol{\phi})$, first we empirically approximate the marginal probability of observing the jth feature under the AM mechanism as

$$\pi_j(\boldsymbol{\phi}) := \frac{1}{N} \sum_{i=1}^{N} \sum_{\forall \mathbf{r}, \mathbf{r}_j = 1} \mathrm{p}_{R|X}(\mathbf{r} \mid \mathbf{x}^{(i)}; \boldsymbol{\phi}). \tag{11}$$

Next, we empirically approximate the expectation in (equation 9) as

$$\hat{\boldsymbol{\mu}}_j(\boldsymbol{\phi}) := \frac{1}{N \pi_j(\boldsymbol{\phi})} \sum_{i=1}^{N} \mathbf{x}_j^{(i)} \sum_{\forall \mathbf{r}, \mathbf{r}_j = 1} \mathrm{p}_{R|X}(\mathbf{r} \mid \mathbf{x}^{(i)}; \boldsymbol{\phi}).$$

As $\hat{\mathbf{x}}^{(i,\mathbf{r})}$ is now differentiable with respect to $\boldsymbol{\phi}$, using a sample from (equation 10) in the proxy objective $\tilde{f}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X})$ from (equation 8) gives a proxy objective function for modelers using mean imputation that is differentiable with respect to $\boldsymbol{\phi}$.

4 Solving the Bi-level Problem

The previous section derived differentiable formulations of the inner problems in our bi-level formulation (equation 2) of AM attacks, when the modeler uses CCA, mean imputation, and regression-based imputation. In this section, we discuss the solution of the bi-level problem. For notational brevity we drop the dependence on the dataset $\boldsymbol{X}$ throughout this section. We take $\Omega(\phi; \boldsymbol{X})$ to be the empirical approximation of the expected fraction of missing data, so that the upper level objective is

$$\ell(\boldsymbol{\phi}, \boldsymbol{\theta}) := g(\tilde{\boldsymbol{\theta}}; \boldsymbol{X}) + \frac{\lambda}{N} \sum_{i=1}^{N} \mathbb{E}\left[\frac{|\{j \mid R_j = 0\}|}{d} \,\middle|\, X = \mathbf{x}^{(i)} \right].$$

The corresponding bi-level optimization problem is

$$\min_{\boldsymbol{\phi}} \ell(\boldsymbol{\phi}, \tilde{\boldsymbol{\theta}}(\boldsymbol{\phi})), \text{ s.t. } \tilde{\boldsymbol{\theta}}(\boldsymbol{\phi}) = \arg\max_{\boldsymbol{\theta} \in \Theta} \tilde{f}(\boldsymbol{\theta}, \boldsymbol{\phi}) \tag{12}$$

We use gradient descent on the upper level problem (the adversary's objective) to learn $\boldsymbol{\phi}$, the parameters of the missingness mechanism; computation of the gradient with respect to $\boldsymbol{\phi}$ using the implicit function theorem is standard,

and is described in Appendix C.3. We call the resulting algorithm the Bi-level Formulation for Learning AM Mechanisms (BLAMM). A listing is provided as Algorithm 2 in Appendix C.3. Solvers that are more scalable (Zhang et al. 2023) or more suitable to complex optimization problems such as deep learning (Shen et al. 2024; Hao, Ji, and Liu 2023) exist and can replace the exact solver we used.

Computation of the gradient requires solving the lower-level problem at each iteration. In Section 5, the runtime of the iterated least squares solver we used for the lower-level problems scales linearly with the number of samples in the dataset. In datasets that do not fit into memory, stochastic lower-level solvers can be used. The lower-level problems use objectives that sum over all possible missingness masks, incurring exponential growth with the number of masked variables; in practice, we found masking at most two carefully selected features was sufficient.

We consider two methods of parameterizing the missingness mechanism, $\mathrm{p}_{R|X}(\mathbf{r}|\mathbf{x}, \boldsymbol{\phi})$. The first, proposed in (Koyuncu et al. 2024), uses a neural network to determine the probability of observing the masked features $\mathcal{M}$ for a given instance $\mathbf{x}$. Specifically, the neural network outputs a probability distribution over the $2^{|\mathcal{M}|}$ feasible missingness patterns– those patterns $\mathbf{r}$ in which all non-masked features and a subset of the masked features are observed.

The second method parameterizes the missingness mechanism on a per data-point basis: a distinct parameter vector $\boldsymbol{\phi}^{(i)} \in \mathbb{R}^{2^{|\mathcal{M}|}}$ is learned for each data point $\mathbf{x}^{(i)}$ and the softmax function is applied to $\boldsymbol{\phi}^{(i)}$ to obtain the probabilities of each feasible missingness pattern specific to that data point. Further details are provided in Appendix C.4.

5 Experiments

Our experimental study focuses on tabular data which is of interest for many applications (medical EHR, product reviews, insurance and census data, etc.).

The effectiveness of AM attacks are evaluated on two tasks: (i) manipulating the p-values of feature coefficients in linear and logistic regression models, and (ii) manipulating average treatment effect estimation using regression estimators. The attacks are successful even when: (i) the information available to adversary is limited (e.g. the missingness mitigation technique used by the modeler is unknown), and (ii) the percentage of the training data that can be modified is limited.

5.1 Manipulating p-values of Features

Linear and logistic regression models are the most popular instance of generalized linear models (GLMs). GLMs are learned by maximizing appropriate log-likelihood functions and are widely used in data analysis, so their vulnerabilities to AM attacks has real-world implications. In our notation, the negative log-likelihood function of the GLM corresponds to the score function $J(\mathbf{x}; \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ are the coefficients of the model and $\mathbf{x}$ contains both the features and the response variable.

In our experiments the adversary aims to make the modeler statistically confident that the coefficient of a target variable is zero, $\boldsymbol{\theta}_t = 0$. The AM attack is deemed successful if the average p-value of the target coefficient is greater than 0.05, indicating that the modeler fails to reject the null hypothesis that $\boldsymbol{\theta}_t = 0$. To minimize the change in the predictive accuracy under the AM attack, the remaining coefficients of the adversarial target $\boldsymbol{\theta}_\alpha$ are selected by finding the closest GLM to the underlying data, subject to the condition $\boldsymbol{\theta}_{\alpha,t} = 0$; that is, they are determined using constrained maximum likelihood estimation (see Appendix C.2 for details).

We used two classification (wine-quality, german-credit) and two regression (ca-housing, diabetes) datasets (see Table 3 in the Appendix) to test our attacks. In each dataset, we selected one highly statistically significant feature, as identified using a GLM learned on the complete data, as the target coefficient. In all experiments, the AM mechanism is parameterized using a one-hidden layer neural network with 100 neurons in the hidden layer, and the masking set is restricted to the target feature, $\mathcal{M} = \{t\}$. In training the AM mechanism, the adversarial objective g is taken to be the empirical approximation to the KL-divergence between the learned GLM and the adversarial GLM (see Appendix C.2 for its formulation).

The resulting missingness rates of the target variable ranged from 4.5%-18.1% in the four datasets, except for the CCA attack on the ca-housing dataset (see Table 5 in the Appendix). On this dataset, BLAMM for the CCA attack converged to masking 40.2% entries of the target feature.

To compare the learned AM mechanism with a reasonable baseline, we defined a missing completely at random (MCAR) missingness mechanism with (asymptotically) the same amount of missing data in the same features, given by $\mathrm{p}_R(\mathbf{r}) = N^{-1} \sum_{i=1}^{N} \mathbb{P}_{R|X}(\mathbf{r} \mid \mathbf{x}^{(i)}; \boldsymbol{\phi})$. We sampled 20 masking matrices $\boldsymbol{R}$ from both the learned and the MCAR missingness mechanism. Given the partially observed dataset, the modeler first applied either one of CCA, mean imputation, or the MICE algorithm (r-package "mice" v 3.14.0 (van Buuren and Groothuis-Oudshoorn 2011)) to fill in the missing data. See Appendix D.1 for details. Next, using the resulting data set, the modeler estimated the coefficients of the models and their corresponding p-values. Additional experiments regarding the tailored attack for linear regression imputation are provided in the Appendix D.1.

When the attack type matched the modeler's type, we observed that the learned adversary in all cases successfully made the target variable insignificant (see Table 1, Table 6 in the Appendix D for regression-based imputation attacks, and Table 7 in the Appendix D for diabetes dataset). In a stark contrast, MCAR was unsuccessful in *all* cases. When there was a mismatch between the attack and modeler type, mean imputation attacks showed limited generalization while CCA generally successfully manipulated the coefficients of the modeler using mean imputation. The MICE algorithm was the most robust imputation strategy but failed in preventing the target variables being insignificant against the CCA attacks in the German-credit and diabetes datasets.

Data Valuation as a Defense Data valuation methods assign utility scores to training examples using a clean vali-

Modeler/Attacker		mean/mean	mean/cca	cca/mean	cca/cca	mice/mean	mice/cca
ca-housing	BLAMM	**0.01±0.0** (✓)	**0.01±0.0** (✓)	**1.12±0.0**	**0.06±0.0** (✓)	**0.69±0.0**	**0.27±0.0**
	MCAR	0.76±0.0	0.46±0.0	1.16±0.0	1.17±0.0	1.16±0.0	1.16±0.0
wine-quality	BLAMM	**0.01±0.0** (✓)	**0.04±0.0** (✓)	**0.71±0.0**	**0.04±0.0** (✓)	**0.54±0.0**	**0.20±0.0**
	MCAR	0.69±0.0	0.47±0.0	0.87±0.0	0.86±0.0	0.78±0.0	0.68±0.0
german-credit	BLAMM	**0.01±0.0** (✓)	**0.01±0.0** (✓)	0.38±0.0 (✓)	**0.09±0.0** (✓)	**0.03±0.0** (✓)	**0.01±0.0** (✓)
	MCAR	0.10±0.0	0.10±0.0	**0.16±0.0**	0.24±0.1	0.10±0.0	0.11±0.0

Table 1: The average (over 20 trials) normalized ℓ_1 norm of the difference between the modeler-estimated coefficients and the adversarial coefficients, i.e. $||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_\alpha||_1/||\boldsymbol{\theta}_\alpha||_1$ and its standard deviation (denoted as $\pm$). If the target coefficient resulting from the attack is insignificant on average (i.e. average p-value > 0.05, Table 8 in the Appendix D), this is indicated using a ✓.

dation set. A common defense against data poisoning (Just et al. 2023) discards the lowest-utility examples, up to a preset budget, before training. Since no existing defense methods exist for the AM threat model, we tested this strategy against our AM attacks by discarding samples after imputation. Due to space constraints, the results are provided in Appendix D.1.

5.2 Manipulating ATE under Partial Data Access

As an additional demonstration of the potential of AM attacks we explore their efficacy in manipulating the estimation of average treatment effects (ATEs). The ATE quantifies the causal effect of changing a treatment variable W on an outcome variable Y. When W is binary-valued, the ATE measures the expected difference in the outcome when W is set to 1 versus when W is set to 0. Formally, using the do-calculus (Pearl 2000), the ATE is expressed as $\tau = \mathbb{E}[Y \mid \text{do}(W = 1)] - \mathbb{E}[Y \mid \text{do}(W = 0)]$. For example, if W indicates whether a person received a flu shot and Y indicates whether they caught the flu, the ATE measures the expected change in a person's chance of getting the flu if they were vaccinated compared to if they were unvaccinated.

Various methods have been proposed to address the challenge of ATE estimation. In this section we focus on the popular class of *regression estimators*, which rely on learning the function $\mu_w(x) = \mathbb{E}[Y \mid X = x, \text{do}(W = w)]$ under assumptions such as *unconfoundedness* and *overlap* (Imbens 2004). When these assumptions hold, $\mu_w(x)$ can be estimated using the conditional expectation $\mathbb{E}[Y \mid X = x, W = w]$. The ATE is then estimated by computing the average difference of the predicted outcomes, $\hat{\tau} = \mathbb{E}[\hat{\mu}_1(x) - \hat{\mu}_0(x)]$.

Simple regression estimators use linear models for $\hat{\mu}_w(x)$ by treating W as an additional covariate (Imbens 2004). More recently, neural network models have been proposed. For instance, T-Net trains separate MLPs for each treatment group using the data subsets where $W = w$ (Curth and van der Schaar 2021). TARNet improves upon this by learning a joint representation layer trained using all samples which is used as input to treatment level-specific hypothesis layers trained on the corresponding subsets of the data (Shalit, Johansson, and Sontag 2017). Other nonparametric methods include Causal Forest (CF) (Wager and and Athey 2018), a random forest–based approach.

We evaluated AM attacks in this setting using the Twins dataset (Louizos et al. 2017), which studies how birth weight affects infant mortality. The dataset consists of 11,400 twin pairs with birth weights under 2 kg and includes 30 covariates. In each pair, the heavier twin is assigned as treated and the lighter as control. The observed mortality rates are 17.69% (control) and 16.08% (treatment), yielding a ground truth ATE of -1.61%, indicating slightly higher mortality among lighter twins. As is standard practice (Curth and van der Schaar 2021; Curth et al. 2021), we construct a realistic observation dataset from the ground truth set by including only one of each twin pair, randomly, in the dataset. The dataset is split into training (50%) and testing (50%) sets for model fitting and evaluation. We used the python package "CATENets" (Curth et al. 2021) (v0.2.4) and R package "grf" (v2.4.0) (Athey, Tibshirani, and Wager 2019) for the implementation of non-linear ATE estimators.

In our experiments, the adversary is limited to introducing missing values in covariates (excluding the treatment variable W), while the modeler uses either mean or MICE imputation to impute the missing variables. We also tested a doubly robust estimation procedure introduced in (Mayer et al. 2020) introduced specifically for handling missing data in covariates. The adversary's objective is to manipulate the estimated ATE to be 10%—a drastic shift of approximately 700%, which falsely suggests that heavier babies have substantially higher mortality.

To implement the BLAMM attack, we used a logistic regression model (conditioning on both X and W) as $\hat{\mu}_w(x)$ in the lower-level problem. The upper-level objective minimizes the absolute error between the target ATE and the estimated ATE, i.e., $g(\tilde{\boldsymbol{\theta}}; \boldsymbol{X}) = |\hat{\tau} - 0.10|$ (See Appendix C.2 for its derivation). We introduced missingness in two covariates: gestat (gestational age) and wtgain (weight gain during pregnancy), and parameterized the missingness mechanism using the per-data-point setup. As an additional baseline to the MCAR missingness introduced earlier, we tested an MNAR mechanism that uses the masked variables value through a logistic function to determine the probability its observed (Muzellec et al. 2020).

Partial Data Access: In practice, datasets can be aggregated from multiple sources. Therefore, it is of interest to consider settings where an adversary only controls a subset of the full dataset. We model this scenario by assuming that the aggregated dataset contains N rows, of which only

	Access: 100%		Access: 75%		Access: 50%		Access: 25%	
	BLAMM	MCAR	BLAMM	MCAR	BLAMM	MCAR	BLAMM	MCAR
TARnet + mean	**10.52±0.9**	-1.0±2.2	**8.49±1.7**	-0.38±1.9	**7.26±2.3**	0.04±0.8	**1.62±2.0**	-0.42±1.8
Tnet + mean	**10.42±0.9**	-2.6±3.1	**7.44±1.4**	-1.72±3.2	**3.67±1.4**	-4.7±1.2	**-1.5±0.1**	-3.98±0.9
linear + mean	**9.86±0.1**	-1.45±0.2	**10.1±0.0**	-1.56±0.2	**6.29±0.0**	-1.65±0.1	**2.26±0.0**	-1.33±0.2
CF + mean	**7.65±0.5**	-1.44±0.2	**3.32±0.1**	-1.45±0.3	**3.01±0.0**	-1.3±0.1	**1.54±0.0**	-1.25±0.1

Table 2: BLAMM attacks trained with a linear proxy are effective against non-linear models. The average ATE (%) of regression estimators, $\hat{\tau}$ (over 5 trials). "Access:" indicates the percentage of rows manipulatable. Bold highlights the missingness closer to the target ATE of 10%. See Appendix Tables 13, 14,15 for the additional baselines described in the text.

N_0 rows are provided or influenced by the adversary. We consider four data access regimes where the adversary controls 25%, 50%, 75%, or 100% of the data (i.e., $N_0/N \in \{0.25, 0.5, 0.75, 1.0\}$). We assume this proportion is known and use it to adapt the BLAMM algorithm accordingly. Under partial data access, the effective missingness mechanism becomes a mixture: rows controlled by the adversary follow the AM-induced missingness pattern, while the remaining rows follow a fully observed (non-adversarial) pattern. This is incorporated into BLAMM as a convex combination of the adversarial and identity mechanisms (see Appendix C.4 for details).

Table 2 presents the results across different access regimes. The expected fraction of missing values in the masked covariates (measured over the full dataset) was 7.2%, 11.1%, 24.3%, and 12.5% for adversary access levels of 100%, 75%, 50%, and 25%, respectively. Despite reduced access, BLAMM consistently succeeded in misleading various regression-based ATE estimators, producing inflated estimates with the incorrect sign. We found that attacks targeting a mean imputation proxy still inflated the ATE, even when the modeler used more sophisticated methods like MICE imputation or the MIA technique for the CF algorithm (Appendix Table 13). In contrast, under the baseline MCAR and MNAR attacks, the estimated ATE remained close to the ground truth of -1.61% (Table 2, Appendix Tables 14,15).

These results suggest that AM mechanisms, even when optimized against a simple logistic regression model, can generalize across model classes and remain effective under realistic constraints on adversarial data access.

6 Relevant Work

Bi-level optimization has previously been used to develop insertion-based data poisoning attacks (Jagielski et al. 2018), but our formulation differs as it is for the threat model of adversarial missingness, so both the upper and lower objectives are incomparable.

There is little prior work on omission-based attacks. (Koyuncu et al. 2023, 2024) are the most relevant, as they develop attacks under the same threat model, but the applicability of their AM attacks is severely limited, as detailed in the introduction. (Barash et al. 2020) considers removing complete examples from a dataset, which can be considered an attack on CCA; their approach requires combinatorial optimization, as opposed to our differentiable formulation. (Cheng and Ge 2018) shows a semi-random adversary that, by selectively revealing the true values of initially missing entries, can invalidate the assumptions of non-convex matrix completion and introduce spurious local minimas.

Another remediation strategy is to jointly model the partially observed variables and the underlying missingness mechanism. Recent work (Ipsen, Mattei, and Frellsen 2021; Ma and Zhang 2021; Ghalebikesabi et al. 2021) considers learning deep generative models to impute the missing entries. Such approaches make restrictions on the missingness mechanism to ensure identifiability: (Ipsen, Mattei, and Frellsen 2021; Ma and Zhang 2021) assumes the missing value indicators are conditionally independent given the complete observations, and (Ghalebikesabi et al. 2021) assumes independence of the observed and missing variables given the missingness pattern. While we expect such methods to show robustness to our attack, their assumptions can limit their success. Further, recent surveys of missing data imputation methods suggest that traditional imputation algorithms, including the MICE algorithm, can outperform deep-learning based approaches (Sun et al. 2023; Wang et al. 2022). Our AM attack showed success against the MICE algorithm's implementation in the popular "mice" R-package.

7 Conclusion

This work introduces a general and effective framework for adversarial missingness attacks targeting widely used missing data remediation techniques. Our results show that moderate levels of adversarially introduced missingness can suppress feature significance and reverse treatment effect estimates even when the adversary can adversarily modify only a subset of the training data. These findings raise the need to reassess the security implications of current methods for learning with incomplete data.

The core of our approach is a flexible bi-level optimization strategy for constructing AM attacks, applicable to both supervised and unsupervised learning and to a wide range of objective functions. Our framework currently assumes knowledge of the model class and missingness handling method used by the modeler, although there is some evidence that attacks designed for one mitigation mechanism are effective upon others. Extending this framework to settings involving other remediation techniques—including multiple imputation, k-nearest neighbor imputation, and generative models—is a promising direction for future work.

References

Athey, S.; Tibshirani, J.; and Wager, S. 2019. Generalized Random Forests. *The Annals of Statistics*, 47(2): 1148–1178.

Barash, G.; Shehory, O.; Kraus, S.; and Farchi, E. 2020. Learner-Independent Targeted Data Omission Attacks. In Shehory, O.; Farchi, E.; and Barash, G., eds., *Engineering Dependable and Secure Machine Learning Systems*, volume 1272, 23–41. Cham: Springer International Publishing. ISBN 978-3-030-62143-8 978-3-030-62144-5.

Cheng, Y.; and Ge, R. 2018. Non-convex matrix completion against a semi-random adversary. In *Conference On Learning Theory*, 1362–1394. PMLR.

Curth, A.; Svensson, D.; Weatherall, J.; and van der Schaar, M. 2021. Really Doing Great at Estimating CATE? A Critical Look at ML Benchmarking Practices in Treatment Effect Estimation. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 1.

Curth, A.; and van der Schaar, M. 2021. Nonparametric Estimation of Heterogeneous Treatment Effects: From Theory to Learning Algorithms. arXiv:2101.10943.

Curth, A.; and van der Schaar, M. 2021. On Inductive Biases for Heterogeneous Treatment Effect Estimation. In *Advances in Neural Information Processing Systems*, volume 34, 15883–15894. Curran Associates, Inc.

Ghalebikesabi, S.; Cornish, R.; Holmes, C.; and Kelly, L. 2021. Deep Generative Missingness Pattern-Set Mixture Models. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, 3727–3735. PMLR.

Hao, J.; Ji, K.; and Liu, M. 2023. Bilevel Coreset Selection in Continual Learning: A New Formulation and Algorithm. In *Thirty-Seventh Conference on Neural Information Processing Systems*.

Imbens, G. W. 2004. Nonparametric Estimation of Average Treatment Effects under Exogeneity: A Review. *The Review of Economics and Statistics*, 86(1): 4–29.

Ipsen, N. B.; Mattei, P.-A.; and Frellsen, J. 2021. Not-MIWAE: Deep Generative Modelling with Missing Not at Random Data. *arXiv:2006.12871 [cs, stat]*.

Jagielski, M.; Oprea, A.; Biggio, B.; Liu, C.; Nita-Rotaru, C.; and Li, B. 2018. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, 19–35. San Francisco, CA: IEEE. ISBN 978-1-5386-4353-2.

Just, H. A.; Kang, F.; Wang, J. T.; Zeng, Y.; Ko, M.; Jin, M.; and Jia, R. 2023. Lava: Data Valuation without Pre-Specified Learning Algorithms. *arXiv preprint arXiv:2305.00054*.

Koyuncu, D.; Gittens, A.; Yener, B.; and Yung, M. 2023. Deception by Omission: Using Adversarial Missingness to Poison Causal Structure Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, 1164–1175. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701030.

Koyuncu, D.; Gittens, A.; Yener, B.; and Yung, M. 2024. Adversarial missingness attacks on causal structure learning. *ACM Transactions on Intelligent Systems and Technology*, 15(6): 1–60.

Little, R. J. A.; and Rubin, D. B. 2002. *Statistical Analysis with Missing Data*. Wiley Series in Probability and Statistics. Hoboken: Wiley, second edition edition. ISBN 978-1-118-62586-6 978-1-119-01356-3.

Louizos, C.; Shalit, U.; Mooij, J.; Sontag, D.; Zemel, R.; and Welling, M. 2017. Causal Effect Inference with Deep Latent-Variable Models. arXiv:1705.08821.

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.

Mayer, I.; Sverdrup, E.; Gauss, T.; Moyer, J.-D.; Wager, S.; and Josse, J. 2020. Doubly Robust Treatment Effect Estimation with Missing Attributes. *The Annals of Applied Statistics*, 14(3): 1409–1431.

Muzellec, B.; Josse, J.; Boyer, C.; and Cuturi, M. 2020. Missing Data Imputation Using Optimal Transport. In *Proceedings of the 37th International Conference on Machine Learning*, 7130–7140. PMLR.

Pearl, J. 2000. *Causality: Models, Reasoning, and Inference*. New York, NY, USA: Cambridge University Press, 2 edition. ISBN 0-521-77362-8.

Shalit, U.; Johansson, F. D.; and Sontag, D. 2017. Estimating Individual Treatment Effect: Generalization Bounds and Algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, 3076–3085. PMLR.

Shen, H.; Chen, P.-Y.; Das, P.; and Chen, T. 2024. SEAL: Safety-enhanced Aligned LLM Fine-tuning via Bilevel Data Selection. arXiv:2410.07471.

Sun, Y.; Li, J.; Xu, Y.; Zhang, T.; and Wang, X. 2023. Deep Learning versus Conventional Methods for Missing Data Imputation: A Review and Comparative Study. *Expert Systems with Applications*, 227: 120201.

van Buuren, S.; and Groothuis-Oudshoorn, K. 2011. Mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45: 1–67.

Wager, S.; and and Athey, S. 2018. Estimation and Inference of Heterogeneous Treatment Effects Using Random Forests. *Journal of the American Statistical Association*, 113(523): 1228–1242.

Wang, Z.; Akande, O.; Poulos, J.; and Li, F. 2022. Are Deep Learning Models Superior for Missing Data Imputation in Surveys? Evidence from an Empirical Comparison. *Survey Methodology*, (12).

Zhang, Y.; Khanduri, P.; Tsaknakis, I.; Yao, Y.; Hong, M.; and Liu, S. 2023. An Introduction to Bi-level Optimization: Foundations and Applications in Signal Processing and Machine Learning. arXiv:2308.00788.