LEARNING OVER DIRTY DATA WITH MINIMAL REPAIRS

Anonymous authors
Paper under double-blind review

ABSTRACT

Missing data often exists in real-world datasets, requiring significant time and effort for data repair to learn accurate models. In this paper, we show that imputing all missing values is not always necessary to achieve an accurate ML model. We introduce concepts of minimal and almost minimal repair, which are subsets of missing data items in training data whose imputation delivers accurate and reasonably accurate models, respectively. Imputing these subsets can significantly reduce the time, computational resources, and manual effort required for learning. We show that finding these subsets is NP-hard for some popular models and propose efficient approximation algorithms for wide range of models. Our extensive experiments indicate that our proposed algorithms can substantially reduce the time and effort required to learn on incomplete datasets.

1 INTRODUCTION

An ML model's performance depends heavily on the quality of its training data. Missing or incomplete data is a major real-world data quality issue Graham (2009); Kumar et al. (2017); Krishnan et al. (2016); Chu et al. (2016); Gupta et al. (2021). There are two common approaches to address missing values in training data. The first approach deletes samples with missing values. However, this method can lead to the loss of important information and introduce bias Van Buuren (2018). Another popular approach is data repair or imputation, in which end-users or ML practitioners impute missing values with the correct ones Farhangfar et al. (2008); Le Morvan et al. (2021); Little & Rubin (2002); Śmieja et al. (2018); Pelckmans et al. (2005); Whang et al. (2023); Williams et al. (2005); Ren et al. (2026). Accurate repair is often challenging and expensive as it usually requires extensive collaboration with expensive domain experts. It usually must be repeated whenever the dataset evolves.

To reduce the cost of imputation, significant effort has been made to train imputation models on the observed subset of the dataset that predict accurate values for missing data items Van Buuren & Groothuis-Oudshoorn (2011); Kyono et al. (2021); Le Morvan et al. (2021); Stekhoven & Bühlmann (2011); Yoon et al. (2018); Zheng & Charoenphakdee (2022). State-of-the-art models for data imputation may take a long time to process and predict values for missing data items, and those that use deep neural networks need costly computational resources Perini & Nikolic (2024); Zheng & Charoenphakdee (2022); Yoon et al. (2018). As the dataset evolves, the user often has to repeat these steps. Moreover, in domains where important decisions must be made, e.g., healthcare and criminal justice, humans may need to manually verify the predictions of the imputation models Yakout et al. (2011). Some users also distrust black-box model-based imputation techniques in critical applications and prefer to reason about missing data themselves using observed features and domain knowledge Ahmad et al. (2019); Stempfle et al. (2025). In addition, model-based imputation may perform poorly when the ratio of missing data to observed data is too large Dettori et al. (2018); Jakobsen et al. (2017); Junaid et al. (2025). In these settings, users *may have to manually repair* at least parts of the data. Some prior methods learn directly over incomplete data without identifying which missing values should be repaired Ganti & Willett (2015); Aghasi et al. (2022). Other methods impute only a subset of missing values, but they are model-limited and do not provide guarantees on the minimality of the subset Karlaš et al. (2020); We have more discussion about the additional related work in Appendix G.

To address these challenges, we introduce the concept of a **minimal repair** for a training dataset with missing values. Generally speaking, this subset represents the smallest group of data items with

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

missing values that, once repaired, yields the same model as that trained on a fully and accurately repaired dataset. By finding and imputing this set, users can significantly reduce the time and effort required to manually repair a dataset without sacrificing model accuracy. It also reduces the time and computational resources needed to predict missing values using imputation models and the manual labor required to verify their imputations. Moreover, minimal repair of a dataset pinpoints the subset of the dataset whose uncertainty impacts the effectiveness of the model trained on the dataset. Hence, it simplifies the inspection and debugging of model training, which is often labor intensive Siddiqi et al. (2023). Because incomplete data sets are prevalent and often evolve, a small reduction in time, effort, and computational resources in the preparation of training datasets can save significant resources in the long run. Specifically, our contributions are as follows.

- We define **minimal repair** for learning support vector machines (SVM) (Section 3) and linear regression (Appendix A) over incomplete data. We prove that finding minimal repairs for SVM and linear regression is NP-hard and propose efficient algorithms with provable error bounds to approximate minimal repairs for them.
- Minimal repairs may sometimes be too large or take too long to find. We propose **almost minimal repair**, the minimal subset of data items with missing values whose repair delivers a model with loss within a given threshold from the model trained over the fully and accurately repaired dataset. We prove that the problem of finding almost minimal repairs is NP-hard for SVM and linear regression. We propose algorithms with provable error bounds to approximate almost minimal repairs for models with convex loss. We also propose algorithms to approximate almost minimal repairs for neural networks (Section 4).
- We evaluated the scalability of our algorithms on multiple real-world datasets (Section 5). Our empirical results indicate that our proposed algorithms efficiently approximate minimal and almost minimal repairs and deliver models with the same or almost the same accuracy as those trained over fully repaired datasets. Our results also indicate that using minimal and almost minimal repairs can reduce the time of the model-based imputation methods for large data often without losing accuracy in the downstream learning task.

2 BACKGROUND

We represent training data as a table whose rows are samples. One column contains labels; the others contain features. Let the feature columns be $[\mathbf{z}_1, \ldots, \mathbf{z}_d]$. Each feature takes values from its domain, e.g., real numbers. A *training set* with n samples is a pair $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ is the feature matrix and $\mathbf{y} = [y_1, \ldots, y_n]^T$ is the label vector. Each sample is a vector $\mathbf{x}_i = [x_{i1}, \ldots, x_{id}]$, where x_{ij} is the value of feature $\mathbf{z}_j$ in sample i. Let $f(\mathbf{w}, \mathbf{x})$ denote the prediction function parameterized by $\mathbf{w}$. Given predictor f and loss function L, we write $L(\mathbf{w}, \mathbf{X}, \mathbf{y})$ for the training objective on dataset $(\mathbf{X}, \mathbf{y})$. Training then seeks an optimal model $\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathcal{W}} L(\mathbf{w}, \mathbf{X}, \mathbf{y})$.

Missing values An entry x_{ij} is a missing value if its value is unknown (marked as *null*). An *incomplete sample* (*incomplete feature*) is a sample (feature) with at least one missing value. A *complete sample* (*complete feature*) contains no missing values. We denote the set of missing entries in $\mathbf{X}$ by $M(\mathbf{X})$, the set of incomplete samples by $MS(\mathbf{X})$, and the set of incomplete features by $MF(\mathbf{X})$. We assume missing values appear only in the feature matrix; the label vector is complete.

Completion A *completion* of an incomplete feature matrix $\mathbf{X}$ is any complete matrix $\mathbf{X}^c$ obtained by replacing each missing entry with a value from the corresponding feature domain while leaving all observed entries unchanged. We write $\mathbf{x}_i^c$ for the completed version of sample $\mathbf{x}_i$ in completion $\mathbf{X}^c$. Since feature domains are often large or continuous, an incomplete matrix generally has many, possibly infinitely many, completions. We denote the set of all completions of $\mathbf{X}$ by $\mathcal{X}^C$.

3 MINIMAL REPAIR (MR)

We use the concept of *certain model* Zhen et al. (2024) to define minimal repair. A model $\mathbf{w}^*$ is a certain model for target function f on training set $(\mathbf{X}, \mathbf{y})$ if for every completion $\mathbf{X}^c \in \mathcal{X}^C$, we have $\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathcal{W}} L(\mathbf{w}, \mathbf{X}^c, \mathbf{y})$. Intuitively, a certain model minimizes training loss for all completions of

the incomplete feature matrix. Thus, if a certain model exists, one can learn a training-loss-minimizing model over the training set without imputing the missing values, because training on any completion of the dataset will return the same optimal loss. This observation holds regardless of the missingness mechanism—Missing Completely at Random (MCAR), Missing at Random (MAR), or Missing Not at Random (MNAR). Given the restrictive definition of certain models, they do not often exist Zhen et al. (2024). Thus, we find the minimal repair of an incomplete training set so it has a certain model.

Definition 3.1. Let $\mathbf{S}_{MR} \subseteq MS(\mathbf{X})$ be a set of incomplete samples in the training set $(\mathbf{X}, \mathbf{y})$. We say that $\mathbf{S}_{MR}$ is a minimal repair for target function f if: (1) for every admissible assignment of values to the missing entries in the samples of $\mathbf{S}_{MR}$, fixing those entries yields a training set with a certain model for f; and (2) no set $\mathbf{S}' \subseteq MS(\mathbf{X})$ with $|\mathbf{S}'| < |\mathbf{S}_{MR}|$ satisfies condition (1).

Definition 3.1 specifies which incomplete samples must be repaired before their values are known; its guarantee is therefore uniform over their admissible imputation values. It does not assume that the repaired values are accurate. If the repaired values are inaccurate, the resulting certain model may also be inaccurate. Our aim is *not* to improve imputation accuracy, but to reduce the resources required for imputation through minimal repairs. Since the concept of a certain model relies on exact optimality, finding the minimal repair requires model-specific analytical solutions. We explore minimal repair for two ML models: SVM and linear regression. We focus on SVM in this section and present the approach and experimental results for linear regression in Appendix A and D, respectively.

3.1 Finding Minimal Repair for SVM

We denote the minimal repair for SVM with the regularization parameter C on the training set $(\mathbf{X}, \mathbf{y})$ as $\mathbf{S}_{MR}(\mathbf{X}, \mathbf{y}, C)$. We have the following property regarding its uniqueness.

Theorem 3.2. *Given training set $(\mathbf{X}, \mathbf{y})$ and regularization parameter C, $\mathbf{S}_{MR}(\mathbf{X}, \mathbf{y}, C)$ is unique.*

Let $SV(\mathbf{X}^c, \mathbf{y}, C)$ be the set of support vectors for the optimal SVM model with regularization parameter C on a completion $\mathbf{X}^c$ of the training set $(\mathbf{X}, \mathbf{y})$. For the rest of this section, we focus on the nondegenerate Case 1 in Ref. Zhen et al. (2024) (Lemma 5.1), excluding the degenerate Case 2, which requires incomplete samples to lie exactly on the SVM functional margin for every completion and all model parameters for incomplete features to be exactly zero.

Lemma 3.3. *An incomplete sample $\mathbf{x}_i$ belongs to the minimal repair $S_{MR}(\mathbf{X}, \mathbf{y}, C)$ if and only if at least one completion $\mathbf{x}_i^c$ of $\mathbf{x}_i$ is a support vector in a completion $\mathbf{X}^c$ of $\mathbf{X}$, i.e., $\mathbf{x}_i^c \in SV(\mathbf{X}^c, \mathbf{y}, C)$.*

Hence, to determine if an incomplete sample belongs to the minimal repair, one could materialize every completion of the feature matrix and check if the incomplete sample is a support vector for any of them. However, this process can be extremely inefficient due to the often large number of completions. Assume that each missing value x_{ij} is bounded by an interval $[x_{ij}^{min}, x_{ij}^{max}]$ based on its domain. $\mathbf{X}^e$ is an *edge completion* to $\mathbf{X}$ if for every missing value x_{ij}, $x_{ij}^e = x_{ij}^{min}$ or x_{ij}^{max}. $\mathcal{X}^E$ denotes the set of all edge completions for $\mathbf{X}$. Theorem 3.4 shows that we can use edge completions instead of all completions to check if an incomplete sample belongs to the minimal repair.

Theorem 3.4. *Given the training set $(\mathbf{X}, \mathbf{y})$ and the regularization parameter C, an incomplete sample $\mathbf{x}_i$ belongs to the minimal repair $S_{MR}(\mathbf{X}, \mathbf{y}, C)$ if and only if there is at least one edge completion $\mathbf{X}^e$ of $\mathbf{X}$ such that $\mathbf{x}_i^e \in SV(\mathbf{X}^e, \mathbf{y}, C)$ where $\mathbf{x}_i^e$ is the completion of $\mathbf{x}_i$.*

By Theorem 3.4, we find the minimal repair as follows: 1) Create an empty minimal repair set, S_{MR}. 2) For each incomplete sample $\mathbf{x}_i$, materialize all edge completions, and add $\mathbf{x}_i$ to S_{MR} if $\mathbf{x}_i$ is a support vector for any of the edge completions, and 3) return the minimal repair S_{MR}. Despite this optimization, finding the minimal repair remains computationally intractable.

Theorem 3.5. *Given a training set $(\mathbf{X}, \mathbf{y})$ with missing values, deciding whether an incomplete sample belongs to the minimal repair for SVM on $(\mathbf{X}, \mathbf{y})$ is NP-hard. Consequently, finding the minimal repair for SVM is NP-hard.*

3.2 Approximating Minimal Repair

We propose an efficient approximation algorithm (Algorithm 1) to find minimal repair for SVM. Its key idea is to test whether each incomplete sample $\mathbf{x}_i$ belongs to minimal repair by constructing an

Algorithm 1 Approximating minimal repair for SVM on training set $(\mathbf{X}, \mathbf{y})$

$S_{MR} \leftarrow \emptyset$ and $\mathbf{X}^e \leftarrow$ a random edge completion to the feature matrix $\mathbf{X}$
for $\mathbf{x}_i \in MS(\mathbf{X})$ **do**
  for $x_{pq} \in M(\mathbf{X})$ **do**
    $\mathbf{X}^{e_{min}}, \mathbf{X}^{e_{max}} \leftarrow$ two edge completions by only replacing x_{pq} in $\mathbf{X}^e$ with its min or max value
    $\mathbf{w}_1, \mathbf{w}_2 \leftarrow SVM(\mathbf{X}^{e_{min}}, \mathbf{y}), SVM(\mathbf{X}^{e_{max}}, \mathbf{y})$ {learning SVM models with edge completions}
    $\mathbf{X}^e \leftarrow \begin{cases} \mathbf{X}^{e_{\min}}, & y_i \mathbf{w}_1^\top \mathbf{x}_i^{e_{\min}} \leq y_i \mathbf{w}_2^\top \mathbf{x}_i^{e_{\max}} \\ \mathbf{X}^{e_{\max}}, & \text{otherwise} \end{cases}$
  end for
  $\mathbf{w} \leftarrow SVM(\mathbf{X}^e, \mathbf{y})$
  if $y_i \mathbf{w}^\top \mathbf{x}_i \leq 1$ **then** $S_{MR} \leftarrow S_{MR}.add(\mathbf{x}_i)$
end for
return S_{MR}

edge completion $\mathbf{X}^e$ that maximizes the chance of $\mathbf{x}_i$ becoming a support vector. This construction starts with a random edge completion and iteratively updates each missing value to its minimum or maximum bound. At each step, this choice minimizes $y_i \mathbf{w}^\top \mathbf{x}_i$, encouraging $\mathbf{x}_i$ to satisfy the support vector condition $y_i \mathbf{w}^\top \mathbf{x}_i \leq 1$. If this condition holds after the full pass, $\mathbf{x}_i$ is selected for repair.

Theorem 3.6. *Every sample returned by Algorithm 1 belongs to* $S_{MR}(\mathbf{X}, \mathbf{y}, C)$.

Based on Theorem 3.6, Algorithm 1 **does not return false positives**. Since it initializes with a random edge completion, it does not bias learning toward any specific imputation. Because each iteration modifies one feature entry, one can avoid retraining the SVM from scratch by using exact incremental and decremental SVM model updates Cauwenberghs & Poggio (2000); Laskov et al. (2006). The algorithm may still miss some samples of minimal repair. Thus, we iteratively apply Algorithm 1 to the remaining incomplete samples to find more samples in the minimal repair. Theorem 3.7 shows that the chance of missing minimal-repair samples decreases with this approach.

Theorem 3.7. *Given training set* $(\mathbf{X}, \mathbf{y})$*, let* $p_k(\mathbf{x})$ *be the probability that an incomplete sample* $\mathbf{x}$ *in minimal repair of* $(\mathbf{X}, \mathbf{y})$ *is not returned at iteration* $k > 0$ *of Algorithm 1. Then* $p_k(\mathbf{x}) > p_{k+1}(\mathbf{x})$.

4 Almost Minimal Repair

Minimal repair guarantees that the learned model exactly achieves the optimal loss that would be obtained from the fully repaired dataset. However, enforcing exact optimality often requires model-specific analytical solutions (as discussed in Section 3) and may yield repair sets that are too large or computationally expensive to compute. In practice, particularly with optimization methods like Stochastic Gradient Descent (SGD), finding the exact global optimum is often not necessary; instead, people typically aim for a model whose loss is sufficiently close to the optimal loss.

To align with this practical reality and reduce repair costs, we relax the definition of minimal repair. We aim for a set of incomplete samples whose imputation allows learning a model whose loss is within a specified threshold of optimal for all possible completions. This relaxation matches standard numerical optimization practice: just as SGD often seeks a point near the optimum rather than the exact optimum, AMR seeks a repair set that guarantees a model near a certain model. We use the concept of approximately certain model (ACM) Zhen et al. (2024) to formalize this notion.

For a user-defined threshold $e \geq 0$, a parameter vector $\mathbf{w}^{\approx}$ is an ACM for prediction function f on training set $(\mathbf{X}, \mathbf{y})$ if, for every completion $\mathbf{X}^c$, $L(\mathbf{w}^{\approx}, \mathbf{X}^c, \mathbf{y}) - \min_{\mathbf{w} \in \mathcal{W}} L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) \leq e$.

Definition 4.1. Given a threshold $e \geq 0$, a repair set S_{AMR} of incomplete samples in the training set $(\mathbf{X}, \mathbf{y})$ is an almost minimal repair (AMR) for target function f with loss L if: (1) repairing S_{AMR} yields an ACM for f on $(\mathbf{X}, \mathbf{y})$, and (2) no other set S' satisfies (1) with $|S'| < |S_{\text{AMR}}|$.

The threshold e works as a stopping condition for the worst-case suboptimality gap: over every completion, a model's training loss may exceed that completion's minimum loss by at most e. A smaller e requires a model closer to optimal for every completion, but generally needs more data repair. If $e = 0$, ACM reduces to a certain model. Thus, finding exact AMR is at least as hard as finding exact MR. Since MR is NP-hard, exact AMR is also NP-hard. We therefore develop approximation

algorithms for AMR, first for models with **convex loss functions**, where approximation guarantees are tractable, and then for **neural networks with non-convex loss functions**.

4.1 Approximating AMR for Convex-Loss Models

For a completion $\mathbf{X}^c$, define the optimality gap of a model $\mathbf{w}$ as $h(\mathbf{w}, \mathbf{X}^c) = L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) - \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}^c, \mathbf{y})$. For a repair set S, let $\mathcal{R}(S)$ be the set of completions consistent with repairing S. The corresponding worst-case gap, the upper envelope of the completion-specific gap functions, is

$$G(\mathbf{w}, S) = \sup_{\mathbf{X}^c\in\mathcal{R}(S)} h(\mathbf{w}, \mathbf{X}^c) \tag{1}$$

Theorem 4.2. *For models with convex loss functions, $h(\mathbf{w}, \mathbf{X}^c)$ is convex with respect to $\mathbf{w}$. $G(\mathbf{w}, S)$ forms a convex upper envelope. An ACM exists if and only if $\min_{\mathbf{w}} G(\mathbf{w}, S) \leq e$.*

Theorem 4.2 implies that if $\min_{\mathbf{w}} G(\mathbf{w}, S) > e$, then no ACM exists for the current repair set. Repairing additional incomplete samples shrinks the completion set and can reduce this worst-case gap. The remaining difficulty is discrete repair selection, which motivates the continuous scoring relaxation below. Additional geometric interpretation is given in Appendix B.1. Thus, for a fixed repair set, checking whether an ACM exists reduces to minimizing a convex worst-case gap.

Continuous Primal-Dual Relaxation. We first state the static AMR target, which is a discrete set-selection problem. We then introduce a continuous score vector over incomplete samples to rank them at each outer iteration. This relaxation is used only for scoring, while the final repair set remains discrete. Algorithm 2 uses a proxy approximation for this inner scoring problem, while the final guarantee is obtained from an outer residual certificate on the actually repaired dataset. Additional details on the constrained formulation and the certificate are given in Appendix B.

Exact static target. For any repair set S, define

$$J(S) = \min_{\mathbf{w}\in\mathcal{W}} \sup_{\mathbf{X}^c\in\mathcal{R}(S)} \left(L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) - \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}^c, \mathbf{y}) \right). \tag{2}$$

The static AMR problem is

$$\min_{S\subseteq MS(\mathbf{X})} |S| \qquad \text{subject to} \qquad J(S) \leq e, \tag{3}$$

which is difficult because each candidate set requires solving a residual robust learning problem.

Continuous scoring relaxation. To score incomplete samples most responsible for the current violation, we introduce a continuous relaxation. Let $\mathbf{s} \in [0, 1]^m$ be a vector of continuous repair scores, one per incomplete sample. We define the continuous robust suboptimality gap $g(\mathbf{w}, \mathbf{s})$ as

$$g(\mathbf{w}, \mathbf{s}) = \max_{\mathbf{X}^c} \left(L(\mathbf{w}, \mathbf{X}^c(\mathbf{s}), \mathbf{y}) - \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}^c(\mathbf{s}), \mathbf{y}) \right) \tag{4}$$

where $\mathbf{X}^c(\mathbf{s})$ denotes a completion whose missing-value intervals are constrained by $\mathbf{s}$. Here $\mathbf{s}$ is not the final repair decision; it is an auxiliary score used to rank incomplete samples at the current outer iteration. Instead of a binary repair choice, the continuous score s_i dynamically and symmetrically shrinks the adversary's search space for each missing value x_{ij} in sample i:

$$x_{ij}^r(s_i) \in \left[x_{ij}^{min} + \frac{s_i}{2}\Delta_{ij}, \quad x_{ij}^{max} - \frac{s_i}{2}\Delta_{ij} \right] \tag{5}$$

where $\Delta_{ij} = x_{ij}^{max} - x_{ij}^{min}$. Let $U(\mathbf{s})$ denote the compact set of completions satisfying these bounds. Define the inner value function $\phi(\mathbf{X}) = \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}, \mathbf{y})$ and the robust-gap integrand $q(\mathbf{w}, \mathbf{X}) = L(\mathbf{w}, \mathbf{X}, \mathbf{y}) - \phi(\mathbf{X})$, so that $g(\mathbf{w}, \mathbf{s}) = \max_{\mathbf{X}\in U(\mathbf{s})} q(\mathbf{w}, \mathbf{X})$. In each repair iteration, we score incomplete samples by minimizing total repair under a relaxed ACM constraint:

$$\min_{\mathbf{s}\in[0,1]^m, \mathbf{w}\in\mathcal{W}} \sum_{i=1}^{m} s_i \quad \text{subject to} \quad g(\mathbf{w}, \mathbf{s}) \leq e \tag{6}$$

Algorithm 2 Approximate primal-dual algorithm for AMR

Input: Training set $(\mathbf{X}, \mathbf{y})$, threshold e, batch size K, inner iterations T_{in}, learning rates $\eta_w, \eta_s, \eta_\lambda$
Initialize: $S_0 \leftarrow \emptyset$, $\widehat{\mathbf{X}}^{(0)} \leftarrow \mathbf{X}$, model $\mathbf{w}_0$ randomly
for $t = 0, 1, \ldots$ **do**
  Compute certificate Γ_t for $\mathbf{w}_t$ on $\widehat{\mathbf{X}}^{(t)}$; if $\Gamma_t \le e$, return $(S_t, \mathbf{w}_t)$
  Initialize $\mathbf{w} \leftarrow \mathbf{w}_t$, $\mathbf{s} \leftarrow \mathbf{0}$, $\lambda \leftarrow 1$
  for $\ell = 1$ to T_{in} **do**
    Find adversarial completion $\mathbf{X}_{adv}$ maximizing $L(\mathbf{w}, \mathbf{X}, \mathbf{y})$ over $U(\mathbf{s})$
    $\mathbf{w} \leftarrow \mathbf{w} - \eta_w \nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{X}_{adv}, \mathbf{y})$
    $\widehat{q}_{curr} \leftarrow L(\mathbf{w}, \mathbf{X}_{adv}, \mathbf{y}) - \phi(\mathbf{X}_{adv})$
    For each unrepaired incomplete sample i, compute first-order sensitivity score ξ_i for $\sum_j s_j + \lambda(\widehat{q}_{curr} - e)$
    $\mathbf{s} \leftarrow \mathrm{Proj}_{[0,1]}(\mathbf{s} - \eta_s \boldsymbol{\xi})$
    $\lambda \leftarrow \max\{0, \lambda + \eta_\lambda(\widehat{q}_{curr} - e)\}$
  end for
  Select the top-K unrepaired incomplete samples using the final scores s_i
  Repair them, update S_{t+1} and $\widehat{\mathbf{X}}^{(t+1)}$, and update $\mathbf{w}_{t+1}$
end for

Proposition 4.3. *Let $\mathcal{M}$ be the set of missing entries, and let $\mathbf{z} \in [-1,1]^{|\mathcal{M}|}$ parameterize a completion $\mathbf{X}(\mathbf{s}, \mathbf{z}) \in U(\mathbf{s})$. Using a multiplier $\lambda \ge 0$ for the constraint $g(\mathbf{w}, \mathbf{s}) \le e$, Equation 6 is equivalent to*

$$\min_{\mathbf{s}\in[0,1]^m,\, \mathbf{w}\in\mathcal{W}} \; \max_{\lambda\ge 0,\, \mathbf{z}\in[-1,1]^{|\mathcal{M}|}} \left[\sum_{i=1}^{m} s_i + \lambda\big(q(\mathbf{w}, \mathbf{X}(\mathbf{s}, \mathbf{z})) - e\big) \right].$$

This reformulation motivates the updates in Algorithm 2. Exact adversarial updates would repeatedly maximize $q(\mathbf{w}, \mathbf{X}) = L(\mathbf{w}, \mathbf{X}, \mathbf{y}) - \phi(\mathbf{X})$, requiring an inner training solve for $\phi(\mathbf{X})$ at each candidate completion. The algorithm first finds a completion $\mathbf{X}_{adv}$ that maximizes the outer loss L over $U(\mathbf{s})$, updates $\mathbf{w}$, and evaluates $\widehat{q}_{curr} = L(\mathbf{w}, \mathbf{X}_{adv}, \mathbf{y}) - \phi(\mathbf{X}_{adv})$. It then updates λ and the repair scores, repairs the top-ranked batch, and recomputes the outer certificate. The exact formulation and the conditions and error bound for this screened search are given in Appendices B.4 and B.5.

Theoretical guarantee. The proxy inner routine is used only to rank incomplete samples and does not certify the returned model. At termination, the outer loop evaluates a certificate Γ_T that upper-bounds the true worst-case residual gap over the discrete completion set $\mathcal{R}(S_T)$ induced by the returned repair set. Therefore, $\Gamma_T \le e$ certifies that $\mathbf{w}_T$ is an ACM (and $\Gamma_T \le e + \delta$ certifies an $(e+\delta)$-ACM); the construction of Γ_T, including the form $Gap_{curr} + R_T$, is given in Appendix B.2.

We now turn to repair-set progress. Define the residual violation $V(S) = \max\{J(S) - e, 0\}$ and the gain $F(S) = V(\emptyset) - V(S)$. Repairing more samples shrinks the completion set, so the residual violation is monotone nonincreasing under set inclusion. Full submodularity need not hold for this robust min–max set function. Let S^* denote an optimal solution to Equation 3. We use the standard weak-submodularity ratio Das & Kempe (2011): for the sets considered below, assume that there is a constant $\gamma \in (0, 1]$ such that, for every pair of disjoint sets L, A with $1 \le |A| \le |S^*|$,

$$\sum_{j\in A} \big(F(L \cup \{j\}) - F(L)\big) \ge \gamma \big(F(L \cup A) - F(L)\big).$$

Theorem 4.4. *Assume that F is monotone and has weak-submodularity ratio at least γ as defined above, and that, at each outer iteration t, the selected repair batch Δ_t of size at most K satisfies*

$$F(S_t \cup \Delta_t) - F(S_t) \ge \alpha \max_{|B|\le K} \big(F(S_t \cup B) - F(S_t)\big)$$

for some constant $\alpha \in (0, 1]$. Let S^* be an optimal solution to Equation 3, let $k = \min\{K, |S^*|\}$, and define

$$\rho = \alpha \left(1 - \exp\left(-\frac{\gamma k}{|S^*|}\right)\right).$$

Then, after T outer iterations,

$$V(S_T) \le \exp(-\rho T) V(\emptyset),$$

Theorem 4.4 guarantees per-iteration progress under weak-submodularity: each iteration removes a ρ fraction of the remaining violation, so $V(S_T)$ decreases geometrically. The next bound is stated by the analysis quantity $V(S_T)$; in practice, Algorithm 2 terminates using the certificate above. This yields the following size guarantee for the returned repair relative to the optimal static repair S^*.

Theorem 4.5. *Under the assumptions of Theorem 4.4, and assuming $K \leq |S^*|$, if Algorithm 2 stops once $V(S_T) \leq \varepsilon_{cert}$, then*

$$|S_T| \leq \frac{K}{\alpha\,(1-\exp(-\gamma K/|S^*|))} \log \frac{V(\emptyset)}{\varepsilon_{cert}} + K.$$

In particular, since $1 - e^{-x} \geq (1 - e^{-1})x$ for $x \in [0, 1]$,

$$|S_T| \leq \frac{|S^*|}{\alpha(1-e^{-1})\gamma} \log \frac{V(\emptyset)}{\varepsilon_{cert}} + K.$$

The weak-submodular and monotone assumptions for the gain function in Theorems 4.4 and 4.5 are not implied by convexity alone. Appendix B.8 discusses explicit examples for common ML models in which the theorems and their assumptions hold under certain scenarios.

4.2 Approximating AMR for Neural Networks

The definition of ACM relies on the robust optimality gap as in Equation 1. For neural networks, this quantity is generally intractable to evaluate because the loss function is non-convex and the global optimum $\min_{\mathbf{w}\in\mathcal{W}} L(\mathbf{w}, \mathbf{X}^c, \mathbf{y})$ is generally unknown. As a result, the convex AMR formulation based on exact robust optimality gap is not directly usable in the neural-network setting.

To handle this difficulty, we use a practical first-order surrogate based on gradient norm. The gradient norm is a standard measure of first-order stationarity in non-convex optimization Ghadimi & Lan (2013); gradient-norm penalties have also been studied empirically in deep learning Zhao et al. (2022). Instead of requiring a small robust optimality gap over all completions consistent with a repair set, we require the model to have a small worst-case gradient norm over those completions together with low training loss on the current repaired dataset. Specifically, we use **Robust Gradient Stationarity**. We define the *Robust Gradient Norm* as $H(\mathbf{w}, S) = \sup_{\mathbf{X}^c\in\mathcal{R}(S)} \|\nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{X}^c, \mathbf{y})\|_2$.

We do not use the gradient norm to match the AMR error threshold, nor claim that a small gradient norm guarantees a small optimality gap: for a general non-convex loss, it provides only a first-order stationarity signal. Certifying an ACM over all completions would require the completion-specific global optima and the exact worst-case completion, which are generally unavailable; obtaining stronger guarantees would require restricting the network architecture and needing additional conditions. Our goal instead is a more general method, for which this difficulty is inherent to general non-convex optimization Jain & Kar (2017); Horst & Tuy (2013). Accordingly, our theoretical ACM and AMR guarantees are confined to the convex setting, while Algorithm 3 is a robust-stationarity heuristic evaluated empirically across datasets and neural-network architectures.

Definition 4.6. A model $\mathbf{w}$ is an (e_{loss}, e_{grad})-Robust Stationary Point for repaired set S if $H(\mathbf{w}, S) \leq e_{grad}$ and $L(\mathbf{w}, \widehat{\mathbf{X}}, \mathbf{y}) \leq e_{loss}$, where $\widehat{\mathbf{X}}$ is the data after imputing samples in S.

The first condition controls the worst-case gradient magnitude over all completions given unrepaired incomplete samples, while the second ensures low training loss on the repaired dataset. Since we cannot minimize the robust optimality gap, we use the practical target $\min_{S,\mathbf{w}} |S|$ subject to $H(\mathbf{w}, S) \leq e_{grad}$ and $L(\mathbf{w}, \widehat{\mathbf{X}}, \mathbf{y}) \leq e_{loss}$, which lacks a global optimality certificate and motivates the heuristic repair procedure for neural networks. Algorithm 3 implements this target through an iterative train-search-repair loop. At each iteration, it first improves the model on the repaired dataset, then searches for a completion that has a large gradient norm, and finally repairs the incomplete sample whose missing values have the largest local effect on that adversarial gradient.

The exact constrained problem remains difficult because evaluating $H(\mathbf{w}, S)$ requires maximizing a non-convex gradient norm over all completions consistent with the incomplete samples that are not yet repaired. We therefore use a heuristic approximation in Algorithm 3. In particular, for the completion $\mathbf{X}_{adv}$ returned by its adversarial search, we use $\widehat{H}(\mathbf{w}, S) = \|\nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{X}_{adv}, \mathbf{y})\|_2$ as an estimate of $H(\mathbf{w}, S)$ for the stopping and repair decisions. Algorithm 3 follows a train–search–repair

Algorithm 3 Approximating AMR for Neural Networks

Input: Training set $(\mathbf{X}, \mathbf{y})$, thresholds e_{loss}, e_{grad}, learning rates α, β
Initialize: $\mathbf{w}$ randomly, $S_{AMR} \leftarrow \emptyset$, $\widehat{\mathbf{X}}$ by mean imputation, and $\widehat{H} \leftarrow \infty$
while $\widehat{H} > e_{grad}$ **or** $L(\mathbf{w}, \widehat{\mathbf{X}}, \mathbf{y}) > e_{loss}$ **do**
  Update $\mathbf{w}$ by taking K gradient steps on $L(\mathbf{w}, \widehat{\mathbf{X}}, \mathbf{y})$
  Find adversarial completion $\mathbf{X}_{adv}$ that approximately maximizes $\|\nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{X}, \mathbf{y})\|_2$
  $\widehat{H} \leftarrow \|\nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{X}_{adv}, \mathbf{y})\|_2$
  if $\widehat{H} > e_{grad}$ **then**
    For each incomplete sample i, compute its sensitivity score from $\|\nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{X}_{adv}, \mathbf{y})\|_2$
    $i^* \leftarrow \arg\max_i \{\text{sensitivity score of sample } i\}$
    $S_{AMR} \leftarrow S_{AMR} \cup \{i^*\}$ and repair $\widehat{\mathbf{X}}_{i^*}$
  end if
end while
Return: S_{AMR}

Table 1: SVM results on data with injected MNAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)			Accuracy(%)			Impute % of Samples		
			AC	MR	AMR	AC	MR	AMR	AC	MR	AMR
Malware	20	95.43	$\mathbf{0.60 \pm 0.01}$	10.50 ± 0.60	7.46 ± 0.46	93.99 ± 1.01	$\mathbf{95.43 \pm 0.86}^*$	$95.12 \pm 0.15^*$	3.61 ± 0.4	20.07 ± 2.6	$\mathbf{3.2 \pm 0.2}^*$
	40	95.43	$\mathbf{0.36 \pm 0.00}$	16.10 ± 1.10	8.12 ± 0.38	93.03 ± 1.24	$\mathbf{95.41 \pm 0.69}^*$	$95.02 \pm 0.19^*$	3.31 ± 0.2	22.12 ± 1.4	$\mathbf{2.9 \pm 0.1}$
	60	95.43	$\mathbf{0.22 \pm 0.03}$	22.20 ± 2.60	8.42 ± 0.72	94.21 ± 0.08	$\mathbf{95.14 \pm 0.44}$	94.86 ± 0.08	3.09 ± 0.1	20.98 ± 2.8	$\mathbf{2.5 \pm 0.3}$
Tuadromd	20	98.71	0.63 ± 0.03	1.74 ± 0.05	$\mathbf{0.20 \pm 0.00}$	$98.62 \pm 0.04^*$	$\mathbf{98.71 \pm 0.14}^*$	98.32 ± 0.0	3.28 ± 0.5	9.93 ± 1.3	$\mathbf{0.52 \pm 0.2}$
	40	98.71	0.45 ± 0.01	3.16 ± 0.04	$\mathbf{0.30 \pm 0.00}$	98.03 ± 0.13	$\mathbf{98.38 \pm 0.03}$	98.32 ± 0.0	2.38 ± 0.6	10.06 ± 0.7	$\mathbf{0.22 \pm 0.0}$
	60	98.71	$\mathbf{0.47 \pm 0.03}^*$	4.16 ± 0.02	$0.50 \pm 0.10^*$	93.63 ± 1.45	$98.18 \pm 0.14^*$	$\mathbf{98.32 \pm 0.0}^*$	3.34 ± 0.2	9.42 ± 0.5	$\mathbf{0.50 \pm 0.2}$
Parkinsons	20	89.2	0.01 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.15 ± 0.03	87.8 ± 1.4	$\mathbf{90.8 \pm 1.1}$	89.74 ± 2.56	34.43 ± 2.0	20.00 ± 8.2	$\mathbf{10.01 \pm 0.3}$
	40	89.2	$\mathbf{0.01 \pm 0.00}$	0.01 ± 0.00	0.20 ± 0.05	84.6 ± 0.9	$\mathbf{88.9 \pm 2.6}$	88.69 ± 2.56	32.80 ± 8.2	28.57 ± 0.0	$\mathbf{9.52 \pm 0.5}$
	60	89.2	$\mathbf{0.01 \pm 0.00}$	0.02 ± 0.01	0.31 ± 0.07	82.8 ± 0.8	$\mathbf{87.0 \pm 0.8}$	86.32 ± 1.48	30.13 ± 1.6	29.08 ± 4.4	$\mathbf{14.02 \pm 1.1}$
Heart	20	85.00	0.01 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.04 ± 0.01	83.3 ± 0.7	$\mathbf{85.2 \pm 1.0}$	85.15 ± 0.08	14.90 ± 4.6	24.11 ± 1.0	$\mathbf{4.26 \pm 0.3}$
	40	85.00	0.01 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.05 ± 0.00	79.0 ± 3.3	$\mathbf{85.2 \pm 0.1}$	85.00 ± 0.21	21.87 ± 3.0	14.58 ± 1.5	$\mathbf{10.41 \pm 0.4}$
	60	85.00	0.01 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.07 ± 0.01	77.9 ± 1.7	84.2 ± 1.0	$\mathbf{84.44 \pm 0.52}$	19.90 ± 3.1	34.72 ± 1.0	$\mathbf{13.41 \pm 0.7}$
Fraud	20	86.59	Error	Overtime	$\mathbf{826 \pm 15}$	Error	Overtime	$\mathbf{87.13 \pm 0.35}$	Error	Overtime	$\mathbf{0.03 \pm 0.00}$
	40	86.59	Error	Overtime	$\mathbf{1125 \pm 27}$	Error	Overtime	$\mathbf{86.67 \pm 0.24}$	Error	Overtime	$\mathbf{0.01 \pm 0.00}$
	60	86.59	Error	Overtime	$\mathbf{1407 \pm 24}$	Error	Overtime	$\mathbf{86.55 \pm 0.39}$	Error	Overtime	$\mathbf{0.01 \pm 0.00}$

Table 2: Accuracy/time for SVM on naturally incomplete data using model-based imputation

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	TCSDI	MF	KNN	TCSDI	MF	
Breast Cancer	MR	0.066 ± 0.003	51 ± 3	0.12 ± 0.01	$96.0 \pm 0.3^*$	97.0 ± 0.5	96.6 ± 0.4	18.18 ± 1
	AMR	0.006 ± 0.001	$\mathbf{36 \pm 1}$	$\mathbf{0.04 \pm 0.0}$	$\mathbf{96.1 \pm 0.0}^*$	97.1 ± 0.1	96.9 ± 0.0	$\mathbf{6.7 \pm 0}$
	AC	0.065 ± 0.003	84 ± 5	0.12 ± 0.01	$95.9 \pm 0.4^*$	$\mathbf{97.9 \pm 0.3}$	$96.8 \pm 0.3^*$	87.27 ± 2.15
	Impute All	$\mathbf{0.004}$	102	2.8	95.8	97.0	$\mathbf{97.0}^*$	100
Water-Potability	MR	2.2 ± 0.01	$\mathbf{63 \pm 0}$	9.9 ± 1.1	$60.0 \pm 1.0^*$	$\mathbf{62.8 \pm 0.3}$	$60.1 \pm 0.7^*$	29.94 ± 0.54
	AMR	0.04 ± 0.01	$\mathbf{19 \pm 1}$	$\mathbf{0.2 \pm 0.1}$	$\mathbf{60.9 \pm 0.1}^*$	61.0 ± 0.0	$\mathbf{60.8 \pm 0.1}^*$	$\mathbf{0.1 \pm 0.1}$
	AC	0.33 ± 0.02	85 ± 6	5.47 ± 0.55	55.0 ± 1.2	57.0 ± 0.9	58.2 ± 0.9	1.94 ± 0.35
	Impute All	$\mathbf{0.005}$	1459	12.8	60.5	61.3	60.5	100
Online-Ed	MR	11.5 ± 0.1	1087 ± 94	8.3 ± 1.1	65.2 ± 0.0	65.2 ± 0.0	63.8 ± 1.1	30 ± 0
	AMR	$\mathbf{0.57 \pm 0.02}$	$\mathbf{9 \pm 0}$	$\mathbf{0.93 \pm 0.27}$	63.9 ± 0.0	64.0 ± 0.0	63.8 ± 0.0	$\mathbf{0.3 \pm 0.1}$
	AC	1.8 ± 0.1	94 ± 8	6.2 ± 0.6	63.7 ± 0.5	63.6 ± 0.4	63.4 ± 0.4	0.81 ± 0.18
	Impute All	0.989	3624	17	**65.2**	**65.2**	**65.2**	100
Bankruptcy	MR	29.2 ± 0.3	2287 ± 129	451 ± 28	$\mathbf{97.9 \pm 0.0}$	$\mathbf{97.8 \pm 0.1}$	96.0 ± 0.8	29.97 ± 0.02
	AMR	4.2 ± 0.6	$\mathbf{20 \pm 1}$	$\mathbf{11 \pm 4}$	95.0 ± 0.1	96.9 ± 0.1	96.8 ± 0.2	$\mathbf{0.2 \pm 0.1}$
	AC	$\mathbf{2.2 \pm 0.2}$	101 ± 8	250 ± 22	96.0 ± 0.3	96.8 ± 0.2	96.5 ± 0.2	0.6 ± 0.12
	Impute All	4.84	7620	710	96.0	97.0	**97.5**	100

loop. It first updates the model to reduce the loss on the current repaired dataset $\widehat{\mathbf{X}}$, and then uses projected gradient ascent over the feasible missing-value intervals to find a completion $\mathbf{X}_{adv}$ with a large gradient norm, following the adversarial-training perspective Tsipras et al. (2018); Madry et al. (2017). If this norm exceeds e_{grad}, the algorithm ranks incomplete samples by the sensitivity of the adversarial gradient norm to their missing values, repairs the most sensitive samples, and repeats.

5 Experimental Evaluation

We report a broad experimental study that covers both injected missingness on originally complete datasets and naturally incomplete real-world datasets. We include 15 datasets from multiple domains, including biomedical, public health, finance, education, cybersecurity, and environmental data, spanning a wide range of scales, from 197 samples to 11 million samples and from 8 features to 6823 features. The study also covers missingness rates ranging from 1.97% to 90.80%, three

Table 3: Accuracy/time for MLP on naturally incomplete data using model-based imputation

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	TCSDI	MF	KNN	TCSDI	MF	
Breast Cancer	AMR	1.98 ± 0.01	$\mathbf{59 \pm 1}$	3.69 ± 0.02	$\mathbf{97.1 \pm 0.2}$	$\mathbf{97.2 \pm 0.2}$	$\mathbf{97.2 \pm 0.2}^*$	$\mathbf{30 \pm 3.24}$
	Impute All	$\mathbf{0.004}$	102	$\mathbf{2.8}$	95.8	97.0	97.0^*	100
Water-Potability	AMR	1.24 ± 0	$\mathbf{92 \pm 1}$	$\mathbf{2.94 \pm 0.03}$	$59.6 \pm 1.4^*$	60.1 ± 0.6	$58.9 \pm 3.0^*$	$\mathbf{1.96 \pm 0}$
	Impute All	$\mathbf{0.005}$	1459	12.8	$\mathbf{60.7}^*$	$\mathbf{62.0}$	$\mathbf{61.9}^*$	100
Online-Ed	AMR	1.69 ± 0.01	$\mathbf{95 \pm 2}$	$\mathbf{6.9 \pm 0.13}$	$\mathbf{65.2 \pm 0.0}^*$	$\mathbf{65.2 \pm 0.0}^*$	$\mathbf{65.2 \pm 0.0}^*$	$\mathbf{0.82 \pm 0}$
	Impute All	$\mathbf{0.989}$	3624	17	$\mathbf{65.2}^*$	$\mathbf{65.2}^*$	$\mathbf{65.2}^*$	100
Bankruptcy	AMR	15 ± 0.06	$\mathbf{250 \pm 50}$	$\mathbf{407 \pm 2}$	$\mathbf{97.8 \pm 0.0}$	$\mathbf{97.8 \pm 0.0}$	$\mathbf{97.8 \pm 0.1}$	$\mathbf{5.17 \pm 1.02}$
	Impute All	$\mathbf{4.84}$	7620	710	97.9	97.9	97.5	100

missingness mechanisms (MCAR, MAR, and MNAR) Graham (2012), and five ML models (SVM, linear regression, logistic regression, MLP, and FT-Transformer Gorishniy et al. (2021)) across classification and regression tasks. For originally complete datasets, we inject missing values under the three missingness mechanisms at ratios of 20%, 40%, and 60% of the training samples. For naturally incomplete datasets, we report the observed missing-entry rate and determine the missingness mechanism Tu et al. (2019). Our experiments include five imputation methods (KNN Mattei & Frellsen (2019), MICE Van Buuren & Groothuis-Oudshoorn (2011), MissForest (MF) Stekhoven & Bühlmann (2011), TCSDI Zheng & Charoenphakdee (2022)), and LLM-Forest He et al. (2025), spanning simple local, statistical, ensemble-based, diffusion-based, and LLM-based imputers. We organize the main experimental results into two slices of the study: manual imputation (simulated by ground-truth) and model-based imputation. Appendix C provides more details about the experimental setting: dataset characteristics, hardware, parameters, etc.

We compare MR/AMR with two baselines: 1) *ActiveClean ("AC")* Krishnan et al. (2016), except for neural networks, since AC only works with convex loss functions. AC iteratively selects samples to repair while learning the model and aims to reduce repair costs; this is the closest prior baseline for subset-based repair in ML. 2) *imputing all*, which performs imputation of all missing values.

Manual imputation. The goal is to show that MR and AMR reduce manual repair effort while preserving model quality. On originally complete datasets, we simulate manual repair by injecting missing values and filling selected ones with their ground-truth (GT) values. This simulates manual repair because GT values are available for originally complete datasets and can model the generally accurate repair by experts. Due to the page limit, Table 1 presents SVM results for injected MNAR settings. Appendix D reports the corresponding MCAR and MAR results, as well as results from additional datasets and models. Across most settings, MR achieves an accuracy very close to that obtained by imputing all missing values, but it often repairs larger subsets (i.e., imputes a larger percentage of incomplete samples). AMR repairs substantially fewer incomplete samples than MR in most cases while often maintaining accuracy close to MR and GT. Relative to AC, AMR generally repairs fewer incomplete samples while achieving higher model accuracy. On Fraud, a highly imbalanced dataset, AC returns error because it requires both classes in each sampled subset during its iterative selection process, but some sampled subsets contain only one class; MR exceeds the 15-hour cutoff, while AMR still attains accuracy close to GT with a small imputation ratio. Although MR/AMR may spend more time than AC to find subsets to impute, in the manual-repair setting the cost of domain-expert effort is generally much higher than program running time. This justifies MR/AMR finding a smaller imputation set, even when they incur higher computational overhead.

Model-based imputation. The goal of this slice is to test whether, with model-based imputation (where the main cost is time), MR/AMR reduce time cost while preserving downstream performance. We show SVM results on naturally incomplete datasets in Table 2 with a subset of imputation methods (KNN, TCSDI, and MF), and MLP results for the same setting in Table 3. Appendix D reports the remaining SVM and MLP settings, additional datasets, and additional ML models with all five imputation methods. Across these settings, MR and AMR usually preserve accuracy close to full imputation while repairing only a subset of incomplete samples. Each comparison uses the same imputer for subset and full imputation: subset selection reduces how often the imputer is applied, but it cannot correct systematic errors of a poor imputer and may inherit its downstream limitations. The runtime savings are largest for more expensive imputers such as MissForest, TCSDI and LLM-forest, since the savings from imputing only a subset of incomplete samples are larger than the cost of finding this subset; for fast imputers such as KNN, the subset-selection overhead can dominate. Relative to AC on convex models, MR is often slower but leads to higher model accuracy because it approximates

the model that is optimal in all completions, whereas AMR is often faster because its smaller repair set reduces cost while retaining most of the model accuracy. For MLP and FT-Transformer Gorishniy et al. (2021), where AC and MR are not applicable, the model accuracy from AMR is close to that from full imputation while AMR repairs far fewer samples.

AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity and presentation of the manuscript; assist with drafting and checking mathematical arguments and proofs; assist with writing and debugging code and running experimental workflows; and help clean and reformat datasets.

ETHICS STATEMENT

We have used only publicly available datasets in our empirical studies. We do not believe that any potential harm will result from the outcome of this research activity. The authors of this manuscript do not have any possible conflicts of interest, which may influence the reported results. We do not expect any harmful result to result from the methods and algorithms proposed by this work. We do not propose any privacy-breaching method to recover private information. The imputation framework, assumptions, and methods used in this work are designed to be used on publicly or consensually shared data. They are different from those used to recover information from private or anonymized datasets.

REPRODUCIBILITY STATEMENT

Proofs are provided in Appendix H. Appendix C documents the datasets, missingness settings, hardware, and parameters; Appendices A and B provide additional algorithmic details; and Appendix D reports the full experimental results. Code is available in the anonymous code repository.

REFERENCES

Alireza Aghasi, MohammadJavad Feizollahi, and Saeed Ghadimi. Rigid: Robust linear regression with missing data. *arXiv preprint arXiv:2205.13635*, 2022.

Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. The challenge of imputation in explainable artificial intelligence models. In *IJCAI Workshop on Artificial Intelligence Safety*, 2019. URL https://arxiv.org/abs/1907.12669.

Marco Angioli, Marcello Barbirotta, Abdallah Cheikh, Antonio Mastrandrea, Francesco Menichelli, and Mauro Olivieri. Efficient implementation of linearucb through algorithmic improvements and vector computing acceleration for embedded learning systems. *arXiv preprint arXiv:2501.13139*, 2025.

Aharon Ben-Tal, Arkadi Nemirovski, and Cees Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization*, 13(2):535–560, 2002.

Kristin P Bennett and Erin J Bredensteiner. Duality and geometry in svm classifiers. In *ICML*, volume 2000, pp. 57–64. Citeseer, 2000.

Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. *Advances in neural information processing systems*, 13, 2000.

Xu Chu, Ihab F. Ilyas, Sanjay Krishnan, and Jiannan Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pp. 2201–2206, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450335317. doi: 10.1145/2882903.2912574. URL https://doi.org/10.1145/2882903.2912574.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 1057–1064, 2011.

Joseph R. Dettori, Daniel C. Norvell, and Jens R. Chapman. The sin of missing data: Is all forgiven by way of imputation? *Global Spine J.*, 8(8):892–894, 2018. doi: doi:10.1177/2192568218811922.

Alireza Farhangfar, Lukasz Kurgan, and Jennifer Dy. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41(12):3692–3705, 2008.

Ravi Ganti and Rebecca M Willett. Sparse linear regression with missing data. *arXiv preprint arXiv:1503.08348*, 2015.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. doi: 10.1137/120880811.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

John W Graham. *Missing data: Analysis and design*. Springer Science & Business Media, 2012.

JW Graham. Missing data analysis: making it work in the real world. *Annu Rev Psychol.*, 60:549–576, 2009.

Nitin Gupta, Shashank Mujumdar, Hima Patel, Satoshi Masuda, Naveen Panwar, Sambaran Bandyopadhyay, Sameep Mehta, Shanmukha Guttula, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. Data quality for machine learning tasks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pp. 4040–4041, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3470817. URL https://doi.org/10.1145/3447548.3470817.

Xinrui He, Yikun Ban, Jiaru Zou, Tianxin Wei, Curtiss Cook, and Jingrui He. Llm-forest: Ensemble learning of llms with graph-augmented prompts for data imputation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6921–6936, 2025.

Reiner Horst and Hoang Tuy. *Global optimization: Deterministic approaches*. Springer Science & Business Media, 2013.

Prateek Jain and Purushottam Kar. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336, 2017.

Janus Christian Jakobsen, Christian Gluud, Jørn Wetterslev, and Per Winkel. When and how should multiple imputation be used for handling missing data in randomised clinical trials –a practical guide with flowcharts. *BMC Medical Research Methodology*, 17(1):162, 2017. doi: 10.1186/s12874-017-0442-1. URL https://doi.org/10.1186/s12874-017-0442-1.

K. P. Junaid, Tanvi Kiran, Madhu Gupta, Kamal Kishore, and Sujata Siwatch. How much missing data is too much to impute for longitudinal health indicators? a preliminary guideline for the choice of the extent of missing proportion to impute with multiple imputation by chained equations. *Population Health Metrics*, 23(1):2, 2025.

Bojan Karlaš, Peng Li, Renzhi Wu, Nezihe Merve Gürel, Xu Chu, Wentao Wu, and Ce Zhang. Nearest neighbor classifiers over incomplete information: From certain answers to certain predictions. *arXiv preprint arXiv:2005.05117*, 2020.

Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J Franklin, and Ken Goldberg. Activeclean: Interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment*, 9(12): 948–959, 2016.

Arun Kumar, Matthias Boehm, and Jun Yang. Data management in machine learning: Challenges, techniques, and systems. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pp. 1717–1722, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450341974. doi: 10.1145/3035918.3054775. URL https://doi.org/10.1145/3035918.3054775.

Trent Kyono, Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. Miracle: causally-aware imputation via learning missing data mechanisms. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Pavel Laskov, Christian Gehl, Stefan Krüger, Klaus-Robert Müller, Kristin P Bennett, and Emilio Parrado-Hernández. Incremental support vector learning: Analysis, implementation and applications. *Journal of machine learning research*, 7(9), 2006.

Marine Le Morvan, Julie Josse, Erwan Scornet, and Gael Varoquaux. What's a good imputation to predict with missing values? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 11530–11540. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/5fe8fdc79ce292c39c5f209d734b7206-Paper.pdf.

R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 2002. ISBN 9780471183860. URL http://books.google.com/books?id=aYPwAAAAMAAJ.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4413–4423. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/mattei19a.html.

Kristiaan Pelckmans, Jos De Brabanter, Johan AK Suykens, and Bart De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5-6):684–692, 2005.

Massimo Perini and Milos Nikolic. In-database data imputation. *Proceedings of the ACM on Management of Data*, 2(1):1–27, 2024.

Jose Picado, John Davis, Arash Termehchy, and Ga Young Lee. Learning over dirty data without cleaning. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 1301–1316, 2020.

Wendi Ren, Ke Wan, Junyu Leng, and Shuang Li. Inferring the invisible: Neuro-symbolic rule discovery for missing value imputation. In *International Conference on Learning Representations*, volume 2026, pp. 153114–153149, 2026.

Shafaq Siddiqi, Roman Kern, and Matthias Boehm. Saga: A scalable framework for optimizing data cleaning pipelines for machine learning applications. *Proc. ACM Manag. Data*, 1(3), November 2023. doi: 10.1145/3617338. URL https://doi.org/10.1145/3617338.

Marek Śmieja, Łukasz Struski, Jacek Tabor, Bartosz Zieliński, and Przemysław Spurek. Processing of missing data by neural networks. *Advances in neural information processing systems*, 31, 2018.

Daniel J. Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 10 2011. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr597. URL https://doi.org/10.1093/bioinformatics/btr597.

Lena Stempfle, Arthur James, Julie Josse, Tobias Gauss, and Fredrik D. Johansson. Handling missing values in clinical machine learning: Insights from an expert study, 2025. URL https://arxiv.org/abs/2411.09591.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Ruibo Tu, Cheng Zhang, Paul Ackermann, Karthika Mohan, Hedvig Kjellström, and Kun Zhang. Causal discovery in the presence of missing data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1762–1770. Pmlr, 2019.

Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2018.

Stef Van Buuren and Catharina Groothuis-Oudshoorn. Mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45, 12 2011. doi: 10.18637/jss.v045.i03.

Jian Wang, Seokbeop Kwon, and Byonghyo Shim. Generalized orthogonal matching pursuit. *IEEE Transactions on signal processing*, 60(12):6202–6216, 2012.

Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4):791–813, 2023.

David Williams, Xuejun Liao, Ya Xue, and Lawrence Carin. Incomplete-data classification using logistic regression. In *Proceedings of the 22nd International Conference on Machine learning*, pp. 972–979, 2005.

Mohamed Yakout, Ahmed K. Elmagarmid, Jennifer Neville, Mourad Ouzzani, and Ihab F. Ilyas. Guided data repair. *Proc. VLDB Endow.*, 4(5):279–289, February 2011. ISSN 2150-8097. doi: 10.14778/1952376.1952378. URL https://doi.org/10.14778/1952376.1952378.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5689–5698. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/yoon18a.html.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International conference on machine learning*, pp. 26982–26992. PMLR, 2022.

Cheng Zhen, Nischal Aryal, Arash Termehchy, and Amandeep Singh Chabada. Certain and approximately certain models for statistical learning. *Proc. ACM Manag. Data*, 2(3), May 2024. doi: 10.1145/3654929. URL https://doi.org/10.1145/3654929.

Shuhan Zheng and Nontawat Charoenphakdee. Diffusion models for missing value imputation in tabular data. *arXiv preprint arXiv:2210.17128*, 2022.

A MINIMAL REPAIR FOR LINEAR REGRESSION

A.1 ALGORITHM FOR FINDING MINIMAL REPAIR

Orthogonal Matching Pursuit (OMP) provides an efficient approximation for solving the sparse linear regression problem Wang et al. (2012). Essentially, this greedy algorithm begins with an empty solution set and initializes the regression residual to the label vector. In each iteration, the algorithm selects the feature most relevant to the current residual (i.e., having the largest dot product), adds it to the solution set, retains a linear regression model, and updates the residual accordingly. The program stops when the regression residue is sufficiently small. Therefore, OMP will return a subset of features (the solution set) that are sufficient to achieve an optimal linear regression model.

In this paper, we propose a variant of OMP, as outlined in Algorithm A, to find minimal repair for linear regression. Our algorithm has two major differences compared to the conventional OMP. Firstly, we include all complete features in the regression at the initialization, ensuring that we minimize the number of non-zero coefficients only among incomplete features. Secondly, we define our stopping condition by the maximum relevance (cosine similarity) between the feature and the label being smaller than or equal to a user-defined threshold, instead of relying on a near-zero regression residue. This approach enables our algorithm to work with general datasets without requiring the assumption of an underdetermined linear system, which is typically necessary in conventional OMP.

Algorithm A Approximating minimal repair for linear regression efficiently

$S_{min} \leftarrow [\ \ ]$
$MVF(\mathbf{z}) \leftarrow$ set of incomplete features
$Complete(\mathbf{z}) \leftarrow$ set of complete features
$\mathbf{r} \leftarrow LR(Complete(\mathbf{z}), \mathbf{y})$ {The residue vector from performing linear regression between complete features and label}
$\epsilon \leftarrow$ a user-defined threshold for stopping condition
$MaxCosSim \leftarrow \max_{\mathbf{z} \in MVF(\mathbf{z})} |cos(\mathbf{z}, \mathbf{r})|$
while $MaxCosSim \leq \epsilon$ **do**
  $S_{min} \leftarrow S_{min}.add(\arg\max_{\mathbf{z} \in MVF(\mathbf{z})} |cos(\mathbf{z}, \mathbf{r})|)$
  $\mathbf{r} \leftarrow LR(Complete(\mathbf{z}) \cup S_{min}, \mathbf{y})$
  $MaxCosSim \leftarrow \max_{\mathbf{z} \in MVF(\mathbf{z})} |cos(\mathbf{z}, \mathbf{r})|$
end while
$res \leftarrow S_{min}$

As mentioned in the main content, the time complexity of the algorithm is $\mathcal{O}(T_{train} \cdot |MVF(\mathbf{z})|)$, making it significantly more efficient than the baseline algorithm, which trains models over all completions individually and has a time complexity of $\mathcal{O}(T_{train} \cdot |\mathcal{X}^C|)$. If a gradient descent algorithm is used, Algorithm A has a time complexity of $\mathcal{O}(n \cdot d^3)$, where n is the number of training samples and d is the number of features. In cases where $n < d^2$, the time complexity can be reduced to $\mathcal{O}(n \cdot d^2 + n^2 \cdot d)$ under certain conditions by applying incremental learning techniques based on the Sherman-Morrison formula, as outlined below.

A.2 OPTIMIZATION FOR ALGORITHM A

The primary time cost in Algorithm A arises from the need to completely retrain the linear regression model each time a new imputed feature is added to the feature set. This retraining leads to a time complexity of $\mathcal{O}(n \cdot d^3)$ for the algorithm. To address this inefficiency, we propose an optimization using the Sherman-Morrison formula to update the inverse of the feature matrix incrementally Angioli et al. (2025). This method reduces the time complexity of including one new feature to $\mathcal{O}(n^2)$. Consequently, when $n < d^2$, this optimization results in significant time savings.

Given a feature matrix $\mathbf{X}$, a label vector $\mathbf{y}$, and the coefficients $\mathbf{w}$ of the current linear regression model, our objective is to efficiently update $\mathbf{w}$ to incorporate a newly imputed feature vector $\mathbf{x}_{\text{new}}$ into $\mathbf{X}$, forming an updated feature matrix $\mathbf{X}'$, without the necessity of full retraining. When this new feature vector $\mathbf{x}_{\text{new}}$ is added to $\mathbf{X}$, it modifies the original matrix product $\mathbf{X}^T\mathbf{X}$ to $\mathbf{X}^T\mathbf{X} + \mathbf{x}_{\text{new}}\mathbf{x}_{\text{new}}^T$. Applying the Sherman-Morrison formula, the updated inverse of $\mathbf{X}'^T\mathbf{X}'$ (assuming $\mathbf{X}'^T\mathbf{X}'$

is invertible) is given by:

$$(\mathbf{X}'^T\mathbf{X}')^{-1} = (\mathbf{X}^T\mathbf{X})^{-1} - \frac{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_{\text{new}}\mathbf{x}_{\text{new}}^T(\mathbf{X}^T\mathbf{X})^{-1}}{1 + \mathbf{x}_{\text{new}}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_{\text{new}}} \tag{7}$$

This formulation enables the efficient update of the regression coefficients $\mathbf{w}$, requiring only $O(n^2)$ operations. Implementing at most d such updates results in a complexity of $\mathcal{O}(d \cdot n^2)$. Including the initial model training $\mathcal{O}(d^2 \cdot n)$, the total computational complexity is thus reduced to $\mathcal{O}(n \cdot d^2 + n^2 \cdot d)$.

A.3 Minimal Repair: Feature-wise or Sample-wise

For linear SVM, minimal repair (MR) is defined at the sample level—the algorithm returns a set of samples to repair. This is because the method identifies potential support vectors, which are inherently defined based on individual samples.

In contrast, for linear regression, MR is defined at the feature level—the algorithm selects a subset of features to repair. This stems from the interpretation of linear regression as projecting the residual vector onto the feature space. The approach identifies features that do not contribute to minimizing the training loss, given the current regression residual.

B Additional Details on the Constrained Formulation of Almost Minimal Repair

As defined in the main text (Section 4), finding the exact almost minimal repair (AMR) is an NP-hard problem. In the main text, we approximate this static subset-selection problem by iteratively enlarging the repair set and using a continuous primal-dual relaxation only to score incomplete samples. This appendix provides additional formulation, geometric interpretation, and certification details for that construction.

B.1 Additional Geometric Intuition and Interpretation

The main text states the convex-case view only briefly. Equation 1 writes the robust gap as the pointwise supremum of the completion-specific optimality-gap functions. For convex losses, each function $h(\mathbf{w}, \mathbf{X}^c)$ is convex in $\mathbf{w}$, so $G(\mathbf{w}, S)$ is an upper envelope over the completions consistent with the current repair set. This makes the ACM condition easy to interpret: if the minimum of this envelope is above e, then no ACM exists for the current repair set.

Repairing an incomplete sample shrinks the feasible completion set $\mathcal{R}(S)$. Geometrically, this removes parts of the uncertainty set that may support the current worst-case gap, so the envelope can move downward. The difficulty is that the effect of a repair depends on which completion is active in the worst case, which makes repair selection discrete and combinatorial. This is the bottleneck that motivates the continuous scoring relaxation used in the main text.

We also give a constrained min-max formulation that makes this bottleneck explicit and provides additional theoretical context for the relaxation used in the main text.

B.2 Certification of ACM

The main text uses a certificate Γ_t to upper-bound the true residual robust gap on the current partially repaired dataset.

Lemma B.1. *Let $(S_T, \mathbf{w}_T)$ be the output of Algorithm 2, and let $\widehat{\mathbf{X}}^{(T)}$ be the actually repaired dataset at termination. Suppose the final certificate Γ_T satisfies*

$$\sup_{\mathbf{X} \in \mathcal{R}(S_T)} \left(L(\mathbf{w}_T, \mathbf{X}, \mathbf{y}) - \min_{\mathbf{v} \in \mathcal{W}} L(\mathbf{v}, \mathbf{X}, \mathbf{y}) \right) \leq \Gamma_T.$$

If $\Gamma_T \leq e$, then $\mathbf{w}_T$ is an ACM for all completions consistent with $\widehat{\mathbf{X}}^{(T)}$. More generally, if $\Gamma_T \leq e + \delta$ for some $\delta \geq 0$, then $\mathbf{w}_T$ is an $(e + \delta)$-ACM.

For convex models, Gap_{curr} denotes the robust gap evaluated at the currently identified adversarial completion. ACM still requires controlling the supremum over all completions in $\mathcal{R}(S_T)$, so Gap_{curr} alone is sufficient only when it already equals that worst-case gap. We therefore allow an additional remainder term R_T that upper bounds the residual contribution of the unseen completions.

Corollary B.2. *For convex models for which the adversarial completion is exact by endpoint search, any certificate of the form $\Gamma_T = Gap_{curr} + R_T$, where $Gap_{curr} = L(\mathbf{w}_T, \mathbf{X}_{adv}, \mathbf{y}) - \phi(\mathbf{X}_{adv})$ and R_T is a valid upper bound on the remaining discrepancy between Gap_{curr} and the worst-case gap over $\mathcal{R}(S_T)$, satisfies the condition of Lemma B.1. In particular, if $Gap_{curr} + R_T \leq e$, then the output is an ACM.*

A convenient form is

$$\Gamma_t = Gap_{curr} + R_t,$$

where

$$Gap_{curr} = L(\mathbf{w}_t, \mathbf{X}_{adv}, \mathbf{y}) - \phi(\mathbf{X}_{adv})$$

is the realized robust gap on the screened completion and $R_t \geq 0$ is any valid upper bound on the remaining unseen completions under the current repair set. By construction, if the residual robust gap is at most Γ_t, then $\Gamma_t \leq e$ certifies ACM. For convex models, this step is practical because the screened completion is found by endpoint search and the inner optimization for $\phi(\mathbf{X}_{adv})$ can be warm-started from the previous iterate.

Two concrete residual bounds are especially convenient.

Certificate 1: Lipschitz bound for the inner value function. Let $U_t = U(\mathbf{s}_t)$ and suppose Step 1 returns an exact maximizer

$$\mathbf{X}_{adv} \in \arg\max_{\mathbf{X} \in U_t} L(\mathbf{w}_t, \mathbf{X}, \mathbf{y}).$$

Assume that the inner value function $\phi(\mathbf{X}) = \min_{\mathbf{v} \in \mathcal{W}} L(\mathbf{v}, \mathbf{X}, \mathbf{y})$ is $K_{\phi,t}$-Lipschitz on U_t. Then for every $\mathbf{X} \in U_t$,

$$L(\mathbf{w}_t, \mathbf{X}, \mathbf{y}) \leq L(\mathbf{w}_t, \mathbf{X}_{adv}, \mathbf{y})$$

and

$$\phi(\mathbf{X}) \geq \phi(\mathbf{X}_{adv}) - K_{\phi,t}\|\mathbf{X} - \mathbf{X}_{adv}\|.$$

Therefore,

$$g(\mathbf{w}_t, \mathbf{s}_t) \leq Gap_{curr} + K_{\phi,t} D_t, \qquad D_t := \max_{\mathbf{X} \in U_t} \|\mathbf{X} - \mathbf{X}_{adv}\|.$$

Hence one valid choice is $R_t = K_{\phi,t} D_t$.

Certificate 2: Lipschitz bound for the full robust gap integrand. Define

$$q_t(\mathbf{X}) = L(\mathbf{w}_t, \mathbf{X}, \mathbf{y}) - \phi(\mathbf{X}).$$

Assume that q_t is $K_{q,t}$-Lipschitz on U_t. Then for every $\mathbf{X} \in U_t$,

$$q_t(\mathbf{X}) \leq q_t(\mathbf{X}_{adv}) + K_{q,t}\|\mathbf{X} - \mathbf{X}_{adv}\|,$$

which implies

$$g(\mathbf{w}_t, \mathbf{s}_t) \leq Gap_{curr} + K_{q,t} D_t.$$

Hence one valid choice is $R_t = K_{q,t} D_t$.

For the box in Equation 5, D_t is the radius of the current completion domain around the screened completion under the chosen norm. In particular, under the Euclidean norm,

$$D_t \leq \left(\sum_{(i,j) \in \mathcal{M}} \big((1 - s_{t,i})\Delta_{ij}\big)^2 \right)^{1/2},$$

where $\mathcal{M}$ is the set of missing coordinates.

B.3 Monotonicity of the Residual Violation

Repairing more samples can only shrink the feasible completion set, so the residual robust objective and the induced residual violation are monotone under set inclusion.

Proposition B.3. *If $S \subseteq T$, then $J(T) \leq J(S)$ and consequently $V(T) \leq V(S)$.*

B.4 Exact Saddle Formulation and Model-Specific Sensitivities

The main text writes the continuous relaxation in Equation 6 as a constrained problem in $(\mathbf{w}, \mathbf{s})$. Using the box parameterization

$$x_{ij}(s_i, z_{ij}) = \frac{x_{ij}^{min} + x_{ij}^{max}}{2} + \frac{1 - s_i}{2}\Delta_{ij} z_{ij}, \qquad z_{ij} \in [-1, 1],$$

the inner maximization is over a fixed adversarial box in $\mathbf{z}$, and the exact relaxation can be written as the following primal-dual reformulation:

$$\min_{\mathbf{s}\in[0,1]^m,\, \mathbf{w}\in\mathcal{W}} \; \max_{\lambda\geq 0,\, \mathbf{z}\in[-1,1]^{|\mathcal{M}|}} \left[\sum_{i=1}^{m} s_i + \lambda\big(q(\mathbf{w}, \mathbf{X}(\mathbf{s}, \mathbf{z})) - e\big)\right],$$

where $q(\mathbf{w}, \mathbf{X}) = L(\mathbf{w}, \mathbf{X}, \mathbf{y}) - \phi(\mathbf{X})$ and $\phi(\mathbf{X}) = \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}, \mathbf{y})$.

For smooth strongly convex models, this exact primal-dual reformulation admits model-specific derivatives. For example, consider ridge linear regression with

$$L(\mathbf{v}, \mathbf{X}, \mathbf{y}) = \frac{1}{2n}\|\mathbf{X}\mathbf{v} - \mathbf{y}\|_2^2 + \frac{\mu}{2}\|\mathbf{v}\|_2^2, \qquad \mu > 0.$$

The inner minimizer $\mathbf{v}^*(\mathbf{X})$ is unique, and standard envelope arguments give

$$\nabla_{\mathbf{X}}\phi(\mathbf{X}) = \nabla_{\mathbf{X}} L(\mathbf{v}^*(\mathbf{X}), \mathbf{X}, \mathbf{y}).$$

Hence the exact robust-gap integrand satisfies

$$\nabla_{\mathbf{X}} q(\mathbf{w}, \mathbf{X}) = \nabla_{\mathbf{X}} L(\mathbf{w}, \mathbf{X}, \mathbf{y}) - \nabla_{\mathbf{X}} L(\mathbf{v}^*(\mathbf{X}), \mathbf{X}, \mathbf{y}),$$

and the exact sensitivity with respect to $\mathbf{s}$ follows by chain rule through $\mathbf{X}(\mathbf{s}, \mathbf{z})$.

For ℓ_2-regularized linear SVM, the exact route remains possible but becomes model-specific and non-smooth: the regularizer makes the inner minimizer unique, while the hinge term yields a subgradient, rather than a smooth gradient, with respect to $\mathbf{X}$. In both cases, exact primal-dual updates require repeated evaluation of $\phi(\mathbf{X})$ and its sensitivity during adversarial search.

Because these exact sensitivities are model-specific and computationally expensive, the main text uses the screened approximate primal-dual routine in Algorithm 2. The main algorithm therefore keeps the exact primal-dual reformulation only as the target variational object, while replacing the exact adversarial and $\mathbf{s}$-sensitivity oracles by screened approximations.

B.5 Screened Adversarial Search for Convex Models

The main-text algorithm does not solve the exact inner maximization of $q(\mathbf{w}, \mathbf{X}) = L(\mathbf{w}, \mathbf{X}, \mathbf{y}) - \phi(\mathbf{X})$ at every iteration. Instead, it first searches for a completion that maximizes the outer loss $L(\mathbf{w}, \mathbf{X}, \mathbf{y})$ over the current uncertainty set and then evaluates the resulting screened completion. For many convex models, this outer-loss search can be performed efficiently.

One-pass outer-loss adversary. For models of the form $L(\mathbf{w}, \mathbf{X}, \mathbf{y}) = R(\mathbf{w}) + \sum_i \ell_i(t_i(\mathbf{w}, \mathbf{x}_i), y_i)$, where each sample loss $\ell_i(\cdot, y_i)$ is convex in the scalar score and, for fixed $\mathbf{w}$, each score $t_i(\mathbf{w}, \mathbf{x}_i)$ is coordinatewise monotone and endpoint-evaluable over the box in Equation 5, the completion that maximizes $L(\mathbf{w}, \mathbf{X}, \mathbf{y})$ over $U(\mathbf{s})$ is found by one pass over the uncertain coordinates, with complexity $\mathcal{O}(m \cdot d)$ per scoring iteration. This applies to a wide range of popular convex models, including linear regression, logistic regression, and linear SVM.

The next theorem explains how far the maximizer of L can be from the exact maximizer of the robust gap q.

Theorem B.4. *Fix* $\mathbf{w}$ *and* $\mathbf{s}$*, and let* $\mathbf{X}_L \in \arg\max_{\mathbf{X}\in U(\mathbf{s})} L(\mathbf{w}, \mathbf{X}, \mathbf{y})$*. Then*

$$q(\mathbf{w}, \mathbf{X}_L) \geq \max_{\mathbf{X}\in U(\mathbf{s})} q(\mathbf{w}, \mathbf{X}) - \Big(\sup_{\mathbf{X}\in U(\mathbf{s})} \phi(\mathbf{X}) - \inf_{\mathbf{X}\in U(\mathbf{s})} \phi(\mathbf{X})\Big).$$

In particular, if ϕ *is* K_ϕ*-Lipschitz on* $U(\mathbf{s})$*, then*

$$q(\mathbf{w}, \mathbf{X}_L) \geq \max_{\mathbf{X}\in U(\mathbf{s})} q(\mathbf{w}, \mathbf{X}) - K_\phi \operatorname{diam}(U(\mathbf{s})).$$

Here, $\mathrm{diam}(U(\mathbf{s})) = \sup_{\mathbf{X},\mathbf{X}' \in U(\mathbf{s})} \|\mathbf{X} - \mathbf{X}'\|$ is the diameter of the uncertainty set, i.e., the largest distance between any two completions in $U(\mathbf{s})$. The theorem shows that maximizing $L(\mathbf{w}, \mathbf{X}, \mathbf{y})$ yields a completion whose robust gap is close to the true worst-case gap, with error controlled by the variation of $\phi(\mathbf{X})$ over the uncertainty set. In particular, when $U(\mathbf{s})$ is small or ϕ varies smoothly, this is a good approximation to the exact search.

Implementation cost and warm starts. Under the endpoint-search conditions above, finding the outer-loss adversarial completion requires one pass over the uncertain coordinates and therefore costs $\mathcal{O}(md)$ per scoring iteration. After selecting $\mathbf{X}_{adv}$, evaluating $\phi(\mathbf{X}_{adv})$ still requires solving one convex training problem for that completion; in practice, this solve can be initialized from the preceding inner solution. Similarly, because an outer iteration repairs only a small batch of samples, the subsequent model update can be initialized from the previous outer iterate $\mathbf{w}_t$. These warm starts are implementation accelerations and do not change the stated worst-case endpoint-search complexity or the certificate requirements.

B.6 An Iterative Constrained Optimization Framework

We can approximate AMR using a greedy iterative framework with two conceptual steps.

Step 1 (**ST1: ACM Optimizer**) takes the input dataset at iteration $k > 0$, $\mathbf{X}^{(k)}$, and finds the model $\mathbf{w}_k^{\approx}$ that minimizes the worst-case suboptimality gap:

$$g_k = \sup_{\mathbf{X}^{(k)r}} \left[L(\mathbf{w}_k^{\approx}, \mathbf{X}^{(k)r}, \mathbf{y}) - \min_{\mathbf{w} \in \mathcal{W}} L(\mathbf{w}, \mathbf{X}^{(k)r}, \mathbf{y}) \right] \tag{8}$$

Step 2 (**ST2: Local Repair Set Identifier**) examines whether $g_k > e$. If so, it returns a small set of currently incomplete samples whose imputation may help further reduce the suboptimality gap in the next iteration.

The discrete view is still useful conceptually: ST1 searches for completions on which the current model performs worst, while ST2 identifies the samples that reduce the violations exposed by those completions. In the revised main-text algorithm, these roles are played by the screened inner scoring surrogate, the top-K repair update, and the final certification step. We therefore use this appendix as motivation for the iterative repair-construction algorithm rather than as an exact description of the main-text procedure.

B.7 Efficient Approximation of the Discrete Steps

Both ST1 and ST2 are intractable in their pure forms because they require solving min-sup optimization over exponentially many completions and identifying small subsets of incomplete samples whose repair is necessary when an ACM does not yet exist. Specifically, these are the samples whose imputation would further reduce the minimum value of the worst-case suboptimality gap $g(\mathbf{w}) = \sup_{\mathbf{X}^c} h(\mathbf{w}, \mathbf{X}^c)$ toward the user-defined threshold e. Exact AMR is NP-hard by the reduction from MR in Section 4; moreover, related robust counterparts are already NP-hard for certain uncertain quadratic and conic-quadratic problems with box uncertainty Ben-Tal et al. (2002). We therefore propose efficient approximations of these steps rather than attributing hardness solely to the presence of a nested min–max expression.

Approximating ST1 (ACM Optimizer): ST1 aims to find the model $\mathbf{w}_k^{\approx} = \arg\min_{\mathbf{w}} \sup_{\mathbf{X}^c} h(\mathbf{w}, \mathbf{X}^c)$, where $h(\mathbf{w}, \mathbf{X}^c) = L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) - \min_{\mathbf{w}'} L(\mathbf{w}', \mathbf{X}^c, \mathbf{y})$. When the training objective $L(\mathbf{w}, \mathbf{X}^c, \mathbf{y})$ is convex in $\mathbf{w}$ for each fixed completion $\mathbf{X}^c$, each $h(\mathbf{w}, \mathbf{X}^c)$ is convex in $\mathbf{w}$, and so is the pointwise supremum of such functions. Thus, we approximate this by sampling a finite subset of edge completions $\{\mathbf{X}_1^e, \ldots, \mathbf{X}_s^e\}$ and solving the convex problem $\min_{\mathbf{w}} \max_i h(\mathbf{w}, \mathbf{X}_i^e)$.

However, directly computing $h(\mathbf{w}, \mathbf{X}^e)$ requires solving an inner optimization for each sampled completion to obtain the minimum loss. To make this tractable, we use the subgradient norm $\|\mathbf{g}(\mathbf{w}, \mathbf{X}^e)\|$ as a proxy for the suboptimality gap. In the main text, this screening step is followed by an explicit certificate for the full robust gap.

Theorem B.5. *If, for each fixed completion $\mathbf{X}^c$, the training objective $L(\mathbf{w}, \mathbf{X}^c, \mathbf{y})$ is convex in $\mathbf{w}$ and has an M-Lipschitz continuous gradient, then any model $\mathbf{w}^{\approx}$ satisfying $\|\nabla_{\mathbf{w}} L(\mathbf{w}^{\approx}, \mathbf{X}^c, \mathbf{y})\| \leq \sqrt{2Me}$ for all $\mathbf{X}^c$ is an ACM.*

This result implies that for linear regression, which satisfies the convexity and smoothness conditions, we can directly use the gradient norm to check whether a model is an ACM. For non-differentiable models like linear SVM, the hinge loss is not smooth, so this theorem does not apply directly. Nonetheless, we still use the subgradient norm there as a practical screening proxy before the explicit certification step.

Approximating ST2 (Local Repair Set Identifier): ST2 must find a small subset of currently incomplete samples whose repair enables further progress toward satisfying the ACM condition. We approximate this by identifying edge completions $\mathbf{X}^e$ from the sampled set where $\|\mathbf{g}(\mathbf{w}_k^{\approx}, \mathbf{X}^e)\| > \epsilon'$, indicating that ACM is violated under these completions.

For linear SVM, we then inspect each such problematic edge completion. For each incomplete sample x_j that currently violates the margin condition (i.e., $y_j(\mathbf{w}_k^{\approx})^T \mathbf{x}_j^e < 1$), we check if there exists a feasible completion where the margin would exceed 1. If so, we assign a score to x_j estimating its potential to reduce the subgradient norm. One option is the maximum hinge loss reduction:

$$\Delta L_{\max} = C \cdot \left[(1 - \text{margin}_j) - \max(0, 1 - \text{margin}_{j,\max})\right], \tag{9}$$

where $\text{margin}_{j,\max}$ is estimated using interval arithmetic over the missing feature bounds. Alternatively, we compute a gradient alignment score based on the inner product between the current subgradient vector and $Cy_j\mathbf{x}_j^e$, estimating the contribution to gradient magnitude.

These scores are aggregated across all high-gradient edge completions. We then select the top-h highest-ranked incomplete samples for imputation in the next iteration. This procedure effectively approximates the function of ST2, enabling targeted refinement of the model toward satisfying the ACM condition.

B.8 Sufficient Cases for the Weak-Submodularity Bounds

Theorems 4.4 and 4.5 are conditional: convexity makes the robust gap tractable for a fixed repair set, but it does not by itself imply weak submodularity of the discrete repair gain or accuracy of the selected batch. This subsection gives explicit cases in which these additional assumptions hold. Let

$$V(S) = [J(S) - e]_+, \qquad F(S) = V(\emptyset) - V(S),$$

where $[a]_+ = \max\{a, 0\}$. A nondecreasing submodular F has submodularity ratio $\gamma = 1$ under the definition in Section 4.

B.8.1 A fixed-curvature sufficient condition

Proposition B.6 (Fixed curvature and nonconflicting uncertainty)**.** *Let M be the set of incomplete repair units. Suppose that there is a positive-definite matrix H such that, for every repair set $S \subseteq M$, the completion-specific optimality gap can be written as*

$$h(\mathbf{w}, \mathbf{z}; S) = \frac{1}{2} \left\| \mathbf{w} - \mathbf{a}_S - \sum_{i \in M \setminus S} z_i \mathbf{u}_i \right\|_H^2, \qquad z_i \in \mathcal{Z}_i,$$

where $\|\mathbf{v}\|_H^2 = \mathbf{v}^\top H \mathbf{v}$, each $\mathcal{Z}_i$ is symmetric about zero with $\max_{z \in \mathcal{Z}_i} |z| = r_i$, and the center $\mathbf{a}_S$ may depend on the repaired values. If

$$\mathbf{u}_i^\top H \mathbf{u}_j \geq 0 \qquad \textit{for every } i \neq j,$$

then F is nondecreasing and submodular. Consequently, its submodularity ratio is $\gamma = 1$.

Proof. For fixed S, the set of completion-specific optimizers is centrally symmetric about $\mathbf{a}_S$. This center minimizes the maximum squared distance to the set: for every displacement $\mathbf{d}$, the displacement $-\mathbf{d}$ is also present, and for any candidate center shift $\mathbf{v}$,

$$\max\{\|\mathbf{v} - \mathbf{d}\|_H^2, \|\mathbf{v} + \mathbf{d}\|_H^2\} \geq \frac{1}{2}\left(\|\mathbf{v} - \mathbf{d}\|_H^2 + \|\mathbf{v} + \mathbf{d}\|_H^2\right) = \|\mathbf{v}\|_H^2 + \|\mathbf{d}\|_H^2.$$

Because every pairwise H-inner product is nonnegative, the largest norm is attained by taking every $|z_i| = r_i$ with a common sign. Therefore

$$J(S) = \frac{1}{2}\sum_{i\notin S} r_i^2 \|\mathbf{u}_i\|_H^2 + \sum_{\substack{i<j \\ i,j\notin S}} r_i r_j \, \mathbf{u}_i^\top H \mathbf{u}_j.$$

Let $F_0(S) = J(\emptyset) - J(S)$. The marginal gain from adding an unrepaired unit i is

$$F_0(S\cup\{i\}) - F_0(S) = \frac{1}{2} r_i^2 \|\mathbf{u}_i\|_H^2 + \sum_{\substack{j\notin S \\ j\neq i}} r_i r_j \, \mathbf{u}_i^\top H \mathbf{u}_j.$$

This quantity is nonnegative and can only decrease as S grows, because terms are removed from a sum of nonnegative quantities. Hence F_0 is nondecreasing and submodular. Finally,

$$F(S) = \min\{F_0(S), [J(\emptyset) - e]_+\},$$

and truncating a nondecreasing submodular function at a constant preserves submodularity. □

The inner-product condition says that uncertainty effects reinforce or are orthogonal to one another rather than canceling. It rules out pure complementarity in which repairing two units together helps even though repairing either one alone does not.

B.8.2 Linear SVM with a stable hinge active set

Consider an ℓ_2-regularized linear SVM with a fixed or regularized intercept,

$$L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) = \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_k \max\{0, 1 - y_k \mathbf{w}^\top \mathbf{x}_k^c\}, \qquad \lambda > 0.$$

Suppose the hinge active set is the same for every admissible completion: each sample with positive hinge loss remains inside the margin, each inactive sample remains outside, and no sample reaches the hinge point. This condition can be certified when every nominal margin has a buffer larger than a valid upper bound on the margin change induced by the missing-value intervals and the corresponding model perturbation.

Assume each active incomplete sample has one-dimensional symmetric uncertainty

$$\mathbf{x}_i^c = \overline{\mathbf{x}}_i + z_i \mathbf{d}_i, \qquad z_i \in \mathcal{Z}_i.$$

With the active set fixed, the hinge terms are affine in $\mathbf{w}$, and completing the square gives

$$L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) - \min_{\mathbf{v}} L(\mathbf{v}, \mathbf{X}^c, \mathbf{y}) = \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}^*(\mathbf{X}^c)\|_2^2, \qquad \mathbf{w}^*(\mathbf{X}^c) = \frac{1}{\lambda}\sum_{k\in A} y_k \mathbf{x}_k^c.$$

Proposition B.6 therefore applies with $H = \lambda I$ and $\mathbf{u}_i = y_i \mathbf{d}_i/\lambda$. In particular, the repair gain is submodular if

$$y_i y_j \, \mathbf{d}_i^\top \mathbf{d}_j \geq 0 \qquad \text{for all active incomplete } i \neq j.$$

This holds, for example, when the uncertainty directions are orthogonal (such as distinct whitened feature coordinates), or when same-class samples are uncertain along the same feature direction. In these cases, $\gamma = 1$.

B.8.3 Ridge regression with a completion-independent Gram matrix

For ridge regression,

$$L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) = \frac{1}{2}\|\mathbf{X}^c\mathbf{w} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \qquad \lambda > 0,$$

suppose all admissible completions have the same Gram matrix,

$$(\mathbf{X}^c)^\top \mathbf{X}^c = G,$$

and

$$(\mathbf{X}^c)^\top \mathbf{y} = \mathbf{b}_S + \sum_{i \notin S} z_i \mathbf{q}_i, \qquad z_i \in \mathcal{Z}_i.$$

Writing $H = G + \lambda I$, the completion-specific optimizer and gap are

$$\mathbf{w}^*(\mathbf{X}^c) = H^{-1}\left(\mathbf{b}_S + \sum_{i \notin S} z_i \mathbf{q}_i\right), \qquad L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) - L^*(\mathbf{X}^c) = \frac{1}{2}\|\mathbf{w} - \mathbf{w}^*(\mathbf{X}^c)\|_H^2.$$

Thus Proposition B.6 applies with $\mathbf{u}_i = H^{-1}\mathbf{q}_i$. The repair gain is submodular whenever

$$\mathbf{q}_i^\top H^{-1} \mathbf{q}_j \geq 0 \qquad (i \neq j).$$

The fixed-Gram requirement is restrictive for arbitrary continuous feature intervals, because changing a feature generally changes $(\mathbf{X}^c)^\top \mathbf{X}^c$. It holds, for example, for discrete sign uncertainty in mutually orthogonal one-sparse rows: replacing $+a\mathbf{e}_j$ by $-a\mathbf{e}_j$ preserves the row outer product. This is an exact structured case rather than a claim about general interval-valued missing features.

B.8.4 Block-separable convex models

Another exact case applies to linear regression, logistic regression, and SVM when the learning problem separates into independent parameter blocks. Suppose

$$\mathcal{W} = \mathcal{W}_0 \times \prod_{i \in M} \mathcal{W}_i, \qquad L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) = L_0(\mathbf{w}_0) + \sum_{i \in M} L_i(\mathbf{w}_i, \mathbf{x}_i^c),$$

the completion domain is a Cartesian product across repair units, and no parameter, including an intercept, is shared between two blocks. Define

$$a_i = \min_{\mathbf{w}_i} \sup_{\mathbf{x}_i^c} \left(L_i(\mathbf{w}_i, \mathbf{x}_i^c) - \min_{\mathbf{v}_i} L_i(\mathbf{v}_i, \mathbf{x}_i^c) \right).$$

Separability of the minimum and supremum gives

$$J(S) = \sum_{i \notin S} a_i, \qquad F(S) = \min\left\{ \sum_{i \in S} a_i, [J(\emptyset) - e]_+ \right\}.$$

Thus F is capped modular and has $\gamma = 1$. This includes independent multi-task convex models and block-diagonal linear or logistic models in which each incomplete repair unit affects its own parameter block and there is no shared intercept. It does not cover ordinary shared-parameter logistic regression merely because its loss is convex.

B.8.5 When the batch-quality factor is guaranteed

The cases above establish the weak-submodularity condition. The separate factor α describes how accurately an outer iteration selects a high-gain batch. If $K = 1$ and the algorithm evaluates

$$\Delta_i(S) = V(S) - V(S \cup \{i\})$$

for every unrepaired unit and chooses the largest value, then $\alpha = 1$. More generally, for a nondecreasing submodular repair gain, sequentially choosing K units by their exact current marginal gains obtains at least a $1 - 1/e$ fraction of the best size-K batch gain, so $\alpha \geq 1 - 1/e$. In the modular cases above, selecting the K largest exact contributions is optimal and gives $\alpha = 1$.

Algorithm 2 uses first-order proxy scores rather than evaluating every exact marginal gain. Therefore these values of α apply directly only when exact marginal evaluation is used, or when the proxy scores are independently shown to achieve the stated fraction of the best true batch gain. The outer certificate still verifies feasibility of the returned repair set; α controls the convergence-rate and repair-size comparisons in Theorems 4.4 and 4.5.

B.9 Discussion: Constrained Optimization vs. SGD Relaxation

While the iterative constrained optimization framework provides a clear, discrete logical boundary for how uncertainty is resolved, it introduces severe scalability bottlenecks in practice:

- **Intractability of the Inner Loop:** Accurately estimating the suboptimality gap requires solving a secondary optimization problem to find the exact worst-case edge completion. As the number of missing values grows, sampling a representative subset of edge completions becomes computationally prohibitive.
- **Retraining Overhead:** ST1 effectively requires retraining or substantially fine-tuning the model from scratch for every iteration to test if the localized repair set successfully satisfied the ACM condition.

These limitations motivated our design of the inner continuous Primal-Dual SGD scoring step used in Algorithm 2. By replacing the discrete combinatorial search of ST2 with a continuous repair score vector $s \in [0,1]^m$, we obtain a scalable ranking of incomplete samples at each outer repair round without requiring exponentially expensive edge-completion evaluations.

C Experimental Setting

C.1 Datasets

We evaluate our methods on two types of datasets: complete datasets with injected missingness and naturally incomplete real-world datasets. Tables 4 and 5 summarize the 15 datasets used in the appendix result matrix; the main-text manual-repair comparison additionally includes Parkinsons and Heart as representative small-scale classification datasets. For injected missingness, we use three missingness levels, 0.2, 0.4, and 0.6, corresponding to 20%, 40%, and 60% of training samples made incomplete. For naturally incomplete datasets, we report the observed missing-entry rate, i.e., the percentage of feature entries that are missing. The datasets are further divided by downstream task, including support vector machine classification (SVM) and linear regression (LR) in this appendix setting summary.

All datasets are obtained from publicly available repositories. For injected missingness, we start with complete datasets and introduce missing values in a controlled manner. For naturally incomplete datasets, we use the original incomplete entries. This separation allows us to analyze the behavior of our repair methods under both controlled and real data corruption scenarios.

Table 4: Details of datasets with injected missing data

Data Set	Task	Features	Training samples	Injected Missingness Level (%)	Missingness Type
Malware	Classification	6823	1596	20-40-60	MCAR, MAR, MNAR
Tuadromd	Classification	242	3571	20-40-60	MCAR, MAR, MNAR
Parkinsons	Classification	22	197	20-40-60	MCAR, MAR, MNAR
Heart	Classification	13	303	20-40-60	MCAR, MAR, MNAR
Fruad	Classification	11	6353307	20-40-60	MCAR, MAR, MNAR
Higgs	Classification	21	11000000	20-40-60	MCAR, MAR, MNAR
Gas	Regression	129	2566	20-40-60	MCAR, MAR, MNAR
Superconductivity	Regression	82	21262	20-40-60	MCAR, MAR, MNAR
Concrete	Regression	8	1030	20-40-60	MCAR, MAR, MNAR

Table 5: Details of datasets with original missing data

Data Set	Task	Features	Training samples	Observed Missing-Entry Rate (%)
Breast Cancer	Classification	10	559	1.97%
Water-Potability	Classification	9	2620	39.00%
Online-Ed	Classification	36	7026	35.48%
Bankruptcy	Classification	64	8402	54.00%
Air Quality	Regression	12	7344	90.80%
Cancer Rate	Regression	32	3048	81.00%

Table S1: Accuracy/time for SVM on data with injected MCAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)			Accuracy(%)			Impute % of Samples		
			AC	MR	AMR	AC	MR	AMR	AC	MR	AMR
Malware	20	95.61	$\mathbf{1.36} \pm 0.15$	6.15 ± 0.71	2.2 ± 0.2	93.13 ± 0.76	$\mathbf{96.7} \pm 0.5$	95.6 ± 0.3	6.39 ± 0.54	18.68 ± 0.5	$\mathbf{1.9 \pm 0.8}$
	40	95.61	$\mathbf{0.56} \pm 0.06$	9.8 ± 0.92	3 ± 0.1	$92.2 \pm 0.7^*$	$92.42 \pm 3.5^*$	$\mathbf{92.6 \pm 0.1}^*$	3.35 ± 0.46	21.1 ± 1.3	$\mathbf{2.1 \pm 0.4}$
	60	95.61	$\mathbf{0.17} \pm 0.05$	12.81 ± 1.05	3.3 ± 0.4	88.67 ± 1.02	$\mathbf{96.37} \pm 0.7$	95.2 ± 0.4	$3.28 \pm 0.43^*$	16.65 ± 0.7	$\mathbf{3 \pm 0.2}^*$
Tuadromd	20	98.77	$0.68 \pm 0.01^*$	1.14 ± 0.13	$\mathbf{0.6 \pm 0.2}^*$	97.53 ± 0.17	$\mathbf{98.73 \pm 0.1}$	97.8 ± 0.1	3.78 ± 0.48	11.9 ± 0.8	$\mathbf{1.9 \pm 0.1}$
	40	98.77	0.54 ± 0.007	2.19 ± 0.34	$\mathbf{0.4 \pm 0.1}$	97.42 ± 0.13	$\mathbf{98.81 \pm 0.14}$	97.7 ± 0.1	3.53 ± 0.37	11.1 ± 1.26	$\mathbf{0.9 \pm 0.2}$
	60	98.77	$\mathbf{0.34 \pm 0.004}^*$	3.29 ± 0.16	$0.4 \pm 0.1^*$	97.5 ± 0.11	$\mathbf{98.77 \pm 0.1}$	97.9 ± 0.2	2.48 ± 0.24	11.8 ± 1.4	$\mathbf{1.7 \pm 0.5}$
HIGGS	20	64.14	OT	OT	$\mathbf{7826.6 \pm 5721.6}$	OT	OT	$\mathbf{64.08 \pm 0.05}$	OT	OT	$\mathbf{0.71 \pm 0.25}$
	40	64.14	OT	OT	$\mathbf{4854.4 \pm 3254.5}$	OT	OT	$\mathbf{63.96 \pm 0.26}$	OT	OT	$\mathbf{0.28 \pm 0.00}$
	60	64.14	OT	OT	$\mathbf{4092.7 \pm 2869.7}$	OT	OT	$\mathbf{63.80 \pm 0.58}$	OT	OT	$\mathbf{0.19 \pm 0.00}$
Fraud	20	99.87	Error	OT	$\mathbf{318.7 \pm 2.4}$	Error	OT	$\mathbf{99.87 \pm 0.00}$	Error	OT	$\mathbf{0.98 \pm 0.00}$
	40	99.87	Error	OT	$\mathbf{315.9 \pm 1.7}$	Error	OT	$\mathbf{99.87 \pm 0.00}$	Error	OT	$\mathbf{0.49 \pm 0.00}$
	60	99.87	Error	OT	$\mathbf{313.3 \pm 2.8}$	Error	OT	$\mathbf{99.87 \pm 0.00}$	Error	OT	$\mathbf{0.33 \pm 0.00}$
Parkinsons	20	89.2	0.01 ± 0.00	0.01 ± 0.00	0.22 ± 0.04	84.9 ± 0.3	$\mathbf{84.9 \pm 0.4}$	84.7 ± 1.1	45.1 ± 2.6	22.58 ± 9.1	$\mathbf{9.84 \pm 0.8}$
	40	89.2	0.01 ± 0.00	0.01 ± 0.00	0.31 ± 0.05	85.0 ± 1.1	$\mathbf{90.6 \pm 1.4}$	86.2 ± 0.9	35.48 ± 1.3	20.97 ± 4.6	$\mathbf{12.43 \pm 1.1}$
	60	89.2	$\mathbf{0.01 \pm 0.00}$	0.02 ± 0.01	0.38 ± 0.07	78.2 ± 1.5	$\mathbf{84.8 \pm 0.9}$	85.1 ± 1.3	36.17 ± 2.1	23.40 ± 2.6	$\mathbf{16.28 \pm 1.5}$
Heart	20	85.0	0.01 ± 0.00	0.01 ± 0.00	0.03 ± 0.01	82.1 ± 0.8	$\mathbf{85.5 \pm 1.3}$	83.6 ± 0.5	16.30 ± 1.0	25.53 ± 6.0	$\mathbf{3.84 \pm 0.4}$
	40	85.0	0.01 ± 0.00	0.01 ± 0.00	0.06 ± 0.01	83.4 ± 0.9	$\mathbf{85.2 \pm 0.6}$	84.1 ± 0.4	18.23 ± 3.6	27.37 ± 1.5	$\mathbf{8.27 \pm 0.9}$
	60	85.0	0.01 ± 0.00	0.01 ± 0.00	0.09 ± 0.02	81.8 ± 2.5	$\mathbf{84.3 \pm 0.6}$	84.0 ± 0.7	20.87 ± 0.7	27.23 ± 1.8	$\mathbf{12.18 \pm 1.2}$

C.2 Hardware

We conducted experiments on two hardware platforms. Most experiments ran on an x86_64 machine with 30 Intel(R) Xeon(R) E5-2670 v3 CPU cores (2.30GHz), hosted in a VMware virtualized environment with two NUMA nodes and 30MB L3 cache. However, this system lacked sufficient power for diffusion-based imputation models. For those experiments (TCSDI), we used an Nvidia DGX-2 system with one Nvidia Tesla V100 GPU (32GB VRAM) and 20 CPU cores from 2.70GHz Intel Xeon Platinum 8168 processors with 33MB L3 cache.

C.3 Using SGD

We have run each experiment that involves Stochastic Gradient Descent (SGD) (SVM) three times with different seeds and report the average. For the convex models, we use the tight optimizer tolerance $\texttt{tol} = 10^{-6}$. This numerical convergence tolerance is distinct from the user-specified AMR tolerance e.

D Additional Experimental Results

Tables S1 through S32 report the results omitted from the main paper due to space. They include additional SVM settings across missingness mechanisms, linear regression results, logistic regression results, and additional MLP results. Together, these tables report the accuracy and runtime of full imputation (baseline), *MR*, *AMR*, and *AC* wherever AC is applicable. Across these experiments, the same trends described in Section 5 continue to hold.

Additional Memory-Usage Results

We report the reduction in computation time for imputing MR and AMR subsets relative to full-data imputation in Table B of the submission. For memory usage, only the Malware–MAR results are included due to space constraints (shown in the table S33, S34, S35). These partial results already illustrate a consistent trend: KNN exhibits substantially lower peak RAM consumption under MR compared to full imputation, and MissForest also shows reduced peak memory across the malware20, malware40, and malware60 configurations. MICE is omitted because it was infeasible to run on this dataset within the available memory budget.

Overall, the observed pattern suggests that MR generally lowers memory usage, particularly for distance-based and non-parametric methods whose resource requirements scale with the number of samples. Future extensions may include a broader memory-usage comparison across additional datasets and missingness settings.

Table S2: Accuracy/time for SVM on data with injected MAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)			Accuracy(%)			Impute % of Samples		
			AC	MR	AMR	AC	MR	AMR	AC	MR	AMR
Malware	20	96.24	$\mathbf{0.80 \pm 0.04}$	7.25 ± 0.73	6.12 ± 0.16	93.62 ± 1.91	$\mathbf{96.24 \pm 0.22}^*$	$96.23 \pm 0.0^*$	$\mathbf{3.24 \pm 1.65}^*$	21.44 ± 2.05	$3.4 \pm 0.1^*$
	40	96.24	$\mathbf{0.58 \pm 0.03}$	11.31 ± 1.46	8.75 ± 1.10	$95.08 \pm 1.10^*$	$\mathbf{96.28 \pm 0.33}^*$	$96.06 \pm 0.38^*$	$\mathbf{3.13 \pm 0.26}$	24.42 ± 2.96	4.1 ± 0.2
	60	96.24	$\mathbf{0.40 \pm 0.01}$	16.47 ± 1.86	9.67 ± 1.17	91.77 ± 1.02	$\mathbf{95.88 \pm 0.65}^*$	$95.64 \pm 0.52^*$	$\mathbf{3.03 \pm 0.31}$	20.94 ± 3.22	3.8 ± 0.1
Tuadromd	20	98.74	0.75 ± 0.04	1.04 ± 0.01	$\mathbf{0.3 \pm 0.1}$	$98.61 \pm 0.02^*$	$\mathbf{98.74 \pm 0.13}^*$	98.32 ± 0.0	3.92 ± 0.23	9.52 ± 0.92	$\mathbf{0.7 \pm 0.0}$
	40	98.74	0.62 ± 0.01	2.04 ± 0.01	$\mathbf{0.4 \pm 0.0}$	$\mathbf{98.71 \pm 0.14}^*$	$98.70 \pm 0.14^*$	98.32 ± 0.0	2.75 ± 0.62	10.06 ± 0.60	$\mathbf{0.7 \pm 0.0}$
	60	98.74	$0.46 \pm 0.01^*$	3.06 ± 0.02	$\mathbf{0.4 \pm 0.1}^*$	$98.58 \pm 0.07^*$	$\mathbf{98.68 \pm 0.09}^*$	98.32 ± 0.0	2.15 ± 0.23	9.64 ± 0.55	$\mathbf{0.5 \pm 0.0}$
HIGGS	20	64.14	OT	OT	$\mathbf{4598.1 \pm 2648.4}$	OT	OT	$\mathbf{64.46 \pm 0.08}$	OT	OT	$\mathbf{1.00 \pm 0.00}$
	40	64.14	OT	OT	$\mathbf{3422.0 \pm 2260.8}$	OT	OT	$\mathbf{63.85 \pm 0.25}$	OT	OT	$\mathbf{0.28 \pm 0.00}$
	60	64.14	OT	OT	$\mathbf{2794.7 \pm 1391.3}$	OT	OT	$\mathbf{63.62 \pm 0.21}$	OT	OT	$\mathbf{0.19 \pm 0.00}$
Fraud	20	99.87	Error	OT	$\mathbf{321.1 \pm 4.8}$	Error	OT	$\mathbf{99.88 \pm 0.00}$	Error	OT	$\mathbf{0.98 \pm 0.00}$
	40	99.87	Error	OT	$\mathbf{309.9 \pm 5.0}$	Error	OT	$\mathbf{99.72 \pm 0.04}$	Error	OT	$\mathbf{0.49 \pm 0.00}$
	60	99.87	Error	OT	$\mathbf{284.8 \pm 2.2}$	Error	OT	$\mathbf{99.84 \pm 0.00}$	Error	OT	$\mathbf{0.33 \pm 0.00}$
Parkinsons	20	89.2	0.01 ± 0.00	0.01 ± 0.00	0.24 ± 0.05	84.1 ± 6.0	$\mathbf{89.0 \pm 1.2}$	85.3 ± 1.4	35.48 ± 6.9	29.03 ± 0.0	$\mathbf{10.27 \pm 1.2}$
	40	89.2	0.01 ± 0.00	0.01 ± 0.00	0.33 ± 0.06	86.7 ± 1.5	$\mathbf{88.9 \pm 2.5}$	86.8 ± 1.0	$\mathbf{24.24 \pm 3.9}$	28.79 ± 0.0	13.91 ± 1.3
	60	89.2	$\mathbf{0.01 \pm 0.00}$	0.02 ± 0.01	0.39 ± 0.08	$\mathbf{81.5 \pm 2.9}$	77.5 ± 7.9	84.6 ± 1.7	32.61 ± 2.8	$\mathbf{17.39 \pm 3.1}$	16.41 ± 1.6
Heart	20	85.0	0.01 ± 0.00	0.01 ± 0.00	0.04 ± 0.01	81.2 ± 0.5	$\mathbf{84.2 \pm 0.2}$	83.5 ± 0.4	54.17 ± 18.0	$\mathbf{11.81 \pm 12.3}$	4.12 ± 0.5
	40	85.0	0.01 ± 0.00	0.01 ± 0.00	0.07 ± 0.01	80.4 ± 3.2	$\mathbf{81.3 \pm 1.3}$	83.9 ± 0.6	30.93 ± 4.7	$\mathbf{13.40 \pm 3.9}$	8.76 ± 0.8
	60	85.0	N/A	0.01 ± 0.00	0.09 ± 0.02	N/A	$\mathbf{86.2 \pm 0.2}$	84.8 ± 0.5	N/A	$\mathbf{28.78 \pm 1.0}$	12.94 ± 1.1

Table S3: Accuracy/time for SVM on data with injected MNAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)			Accuracy(%)			Impute % of Samples		
			AC	MR	AMR	AC	MR	AMR	AC	MR	AMR
Malware	20	95.43	$\mathbf{0.60 \pm 0.01}$	10.50 ± 0.60	7.46 ± 0.46	93.99 ± 1.01	$\mathbf{95.43 \pm 0.86}^*$	$95.12 \pm 0.15^*$	3.61 ± 0.4	20.07 ± 2.6	$\mathbf{3.2 \pm 0.2}^*$
	40	95.43	$\mathbf{0.36 \pm 0.00}$	16.10 ± 1.10	8.12 ± 0.38	93.03 ± 1.24	$\mathbf{95.41 \pm 0.69}^*$	$95.02 \pm 0.19^*$	3.31 ± 0.2	22.12 ± 1.4	$\mathbf{2.9 \pm 0.1}$
	60	95.43	$\mathbf{0.22 \pm 0.03}$	22.20 ± 2.60	8.42 ± 0.72	94.21 ± 0.08	$\mathbf{95.14 \pm 0.44}$	94.86 ± 0.08	3.09 ± 0.1	20.98 ± 2.8	$\mathbf{2.5 \pm 0.3}$
Tuadromd	20	98.71	0.63 ± 0.03	1.74 ± 0.05	$\mathbf{0.20 \pm 0.00}$	$98.62 \pm 0.04^*$	$\mathbf{98.71 \pm 0.14}^*$	98.32 ± 0.0	3.28 ± 0.5	9.93 ± 1.3	$\mathbf{0.52 \pm 0.2}$
	40	98.71	0.45 ± 0.01	3.16 ± 0.04	$\mathbf{0.30 \pm 0.00}$	98.03 ± 0.13	$\mathbf{98.38 \pm 0.03}$	98.32 ± 0.0	2.38 ± 0.6	10.06 ± 0.7	$\mathbf{0.22 \pm 0.0}$
	60	98.71	$\mathbf{0.47 \pm 0.03}^*$	4.16 ± 0.02	$0.50 \pm 0.10^*$	93.63 ± 1.45	$98.18 \pm 0.14^*$	$\mathbf{98.32 \pm 0.0}^*$	3.34 ± 0.2	9.42 ± 0.5	$\mathbf{0.50 \pm 0.2}$
HIGGS	20	64.14	OT	OT	$\mathbf{2777.0 \pm 765.4}$	OT	OT	$\mathbf{63.13 \pm 0.04}$	OT	OT	$\mathbf{1.92 \pm 0.00}$
	40	64.14	OT	OT	$\mathbf{4492.3 \pm 2598.8}$	OT	OT	$\mathbf{61.48 \pm 0.00}$	OT	OT	$\mathbf{0.11 \pm 0.00}$
	60	64.14	OT	OT	$\mathbf{3081.3 \pm 1897.1}$	OT	OT	$\mathbf{60.42 \pm 0.00}$	OT	OT	$\mathbf{0.08 \pm 0.00}$
Fraud	20	99.87	Error	OT	$\mathbf{304.9 \pm 6.7}$	Error	OT	$\mathbf{99.87 \pm 0.00}$	Error	OT	$\mathbf{0.98 \pm 0.00}$
	40	99.87	Error	OT	$\mathbf{214.0 \pm 9.3}$	Error	OT	$\mathbf{99.87 \pm 0.00}$	Error	OT	$\mathbf{0.31 \pm 0.00}$
	60	99.87	Error	OT	$\mathbf{50.1 \pm 2.2}$	Error	OT	$\mathbf{99.87 \pm 0.00}$	Error	OT	$\mathbf{0.04 \pm 0.00}$
Parkinsons	20	89.2	0.01 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.15 ± 0.03	87.8 ± 1.4	$\mathbf{90.8 \pm 1.1}$	89.74 ± 2.56	34.43 ± 2.0	20.00 ± 8.2	$\mathbf{10.01 \pm 0.3}$
	40	89.2	$\mathbf{0.01 \pm 0.00}$	0.01 ± 0.00	0.20 ± 0.05	84.6 ± 0.9	$\mathbf{88.9 \pm 2.6}$	88.69 ± 2.56	32.80 ± 8.2	28.57 ± 0.0	$\mathbf{9.52 \pm 0.5}$
	60	89.2	$\mathbf{0.01 \pm 0.00}$	0.02 ± 0.01	0.31 ± 0.07	82.8 ± 0.8	$\mathbf{87.0 \pm 0.8}$	86.32 ± 1.48	30.13 ± 1.6	29.08 ± 4.4	$\mathbf{14.02 \pm 1.1}$
Heart	20	85.00	0.01 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.04 ± 0.01	83.3 ± 0.7	$\mathbf{85.2 \pm 1.0}$	85.15 ± 0.08	14.90 ± 4.6	24.11 ± 1.0	$\mathbf{4.26 \pm 0.3}$
	40	85.00	0.01 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.05 ± 0.00	79.0 ± 3.3	$\mathbf{85.2 \pm 0.1}$	85.00 ± 0.21	21.87 ± 3.0	14.58 ± 1.5	$\mathbf{10.41 \pm 0.4}$
	60	85.00	0.01 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.07 ± 0.01	77.9 ± 1.7	84.2 ± 1.0	$\mathbf{84.44 \pm 0.52}$	19.90 ± 3.1	34.72 ± 1.0	$\mathbf{13.41 \pm 0.7}$

Table S4: Accuracy/time for SVM on data with injected MCAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Malware 20	MR	8.98 ± 0.41	N/A	24390.8 ± 1215.7	95.6 ± 0.52	N/A	$96.49 \pm 0.31^{*}$	18.68 ± 0.93
	AMR	2.82 ± 0.23	N/A	$\mathbf{4223 \pm 376}$	94.88 ± 0.15	N/A	95.17 ± 0.26	$\mathbf{3.17 \pm 0.25}$
	AC	$\mathbf{1.17 \pm 0.09}$	N/A	12668 ± 1386.2	91.20 ± 0.78	N/A	92.30 ± 0.52	9.09 ± 0.61
	Impute All	13.9	N/A	Overtime	**96.52**	N/A	Overtime	100
Malware 40	MR	14.24 ± 0.97	N/A	54702.4 ± 3875.6	$\mathbf{96.16 \pm 0.61}^{*}$	N/A	96.23 ± 0.36	21.1 ± 1.22
	AMR	3.41 ± 0.93	N/A	$\mathbf{6231 \pm 385}$	$95.63 \pm 0.93^{*}$	N/A	95.27 ± 0.16	$\mathbf{2.13 \pm 0.43}$
	AC	$\mathbf{0.84 \pm 0.07}$	N/A	11085 ± 1118.4	88.97 ± 1.44	N/A	89.73 ± 0.43	5.32 ± 0.14
	Impute All	27.66	N/A	Overtime	93.83	N/A	Overtime	100
Malware 60	MR	18.26 ± 1.53	N/A	Overtime	$\mathbf{96.16 \pm 0.37}^{*}$	N/A	Overtime	16.65 ± 2.12
	AMR	4.98 ± 0.73	N/A	$\mathbf{7103 \pm 824}$	94.35 ± 0.11	N/A	94.75 ± 0.21	$\mathbf{2.51 \pm 0.17}$
	AC	$\mathbf{0.84 \pm 0.06}$	N/A	11085 ± 1429.6	89.00 ± 0.32	N/A	88.82 ± 0.59	3.28 ± 0.17
	Impute All	41.21	N/A	Overtime	$\mathbf{96.16}^{*}$	N/A	Overtime	100
Tuadromd 20	MR	1.39 ± 0.004	100.8 ± 0.91	287 ± 6.21	98.6 ± 0.12	$98.73 \pm 0.10^{*}$	98.43 ± 0.11	11.9 ± 0.8
	AMR	$\mathbf{0.71 \pm 0.34}^{*}$	$\mathbf{15.34 \pm 1.86}$	$\mathbf{75.4 \pm 28.6}$	98.06 ± 0.52	97.76 ± 0.00	98.37 ± 0.27	$\mathbf{2.1 \pm 1.2}$
	AC	$1.04 \pm 0.047^{*}$	99.86 ± 3.08	145 ± 5.92	97.58 ± 0.09	97.63 ± 0.11	97.63 ± 0.08	4.06 ± 0.64
	Impute All	2.37	102.11	1987	**98.77**	$\mathbf{98.77}^{*}$	**98.66**	100
Tuadromd 40	MR	2.55 ± 0.012	96.26 ± 0.63	466.4 ± 9.85	$\mathbf{98.5 \pm 0.10}$	$\mathbf{98.4 \pm 0.09}$	98.54 ± 0.08	11.1 ± 1.26
	AMR	$1.21 \pm 0.39^{*}$	$\mathbf{14.85 \pm 0.85}$	$\mathbf{82.5 \pm 10.1}$	97.76 ± 0.00	97.76 ± 0.00	98.06 ± 0.17	$\mathbf{1.3 \pm 0.4}$
	AC	$\mathbf{0.86 \pm 0.015}^{*}$	99.8 ± 2.41	169 ± 6.73	96.98 ± 0.12	97.12 ± 0.08	97.45 ± 0.09	3.3 ± 0.58
	Impute All	4.69	100.14	3882	97.3	97.6	**98.66**	100
Tuadromd 60	MR	3.62 ± 0.028	79.66 ± 4.92	692.6 ± 31.4	$\mathbf{98.3 \pm 0.11}$	$\mathbf{98.36 \pm 0.03}$	98.38 ± 0.10	11.8 ± 1.4
	AMR	2.11 ± 0.47	$\mathbf{21.5 \pm 1.4}$	$\mathbf{100.3 \pm 8.7}$	97.94 ± 0.23	97.94 ± 0.17	98.06 ± 0.26	$\mathbf{1.48 \pm 0.7}^{*}$
	AC	$\mathbf{0.679 \pm 0.011}$	77.08 ± 1.27	170 ± 7.84	96.96 ± 0.16	96.88 ± 0.07	97.3 ± 0.12	$1.87 \pm 0.19^{*}$
	Impute All	6.21	100.6	6476	97.6	97.3	**98.66**	100
Parkinsons 20	MR	0.009 ± 0.000	0.126 ± 0.002	80.00 ± 1.73	86.5 ± 2.2	$\mathbf{87.3 \pm 0.0}$	83.18 ± 1.02	22.58 ± 9.1
	AMR	0.0881 ± 0.0008	0.1413 ± 0.0010	$\mathbf{8.27 \pm 0.67}$	87.18 ± 0.00	87.18 ± 0.00	$\mathbf{87.08 \pm 0.21}$	$\mathbf{9.84 \pm 0.8}$
	AC	0.009 ± 0.001	0.121 ± 0.002	83.33 ± 4.16	80.8 ± 4.4	$\mathbf{81.2 \pm 1.4}$	79.0 ± 0.6	45.1 ± 2.6
	Impute All	0.010	0.133	84	87.72	88.16	86.0	100
Parkinsons 40	MR	0.010 ± 0.000	0.135 ± 0.004	81.67 ± 1.53	87.1 ± 1.6	$\mathbf{92.0 \pm 2.4}$	86.61 ± 2.71	20.97 ± 4.6
	AMR	0.6235 ± 0.0030	0.6880 ± 0.0053	$\mathbf{10.81 \pm 0.96}$	89.74 ± 0.00	89.74 ± 0.00	$\mathbf{89.56 \pm 0.27}$	$\mathbf{12.43 \pm 1.1}$
	AC	0.010 ± 0.000	0.128 ± 0.003	85.33 ± 1.53	78.8 ± 1.8	$\mathbf{82.4 \pm 4.0}$	78.8 ± 1.1	35.48 ± 1.3
	Impute All	0.012	0.137	87	**90.35**	88.60	80.26	100
Parkinsons 60	MR	0.010 ± 0.000	0.136 ± 0.004	82.67 ± 0.58	82.0 ± 3.1	83.2 ± 1.1	84.57 ± 0.90	23.40 ± 2.6
	AMR	0.6729 ± 0.0435	0.7054 ± 0.0239	$\mathbf{27.51 \pm 2.54}$	$\mathbf{82.91 \pm 5.34}$	83.76 ± 3.92	82.94 ± 4.83	$\mathbf{16.28 \pm 1.5}$
	AC	0.010 ± 0.000	0.125 ± 0.001	84.00 ± 2.00	74.2 ± 5.9	$\mathbf{74.9 \pm 2.5}$	75.5 ± 0.7	36.17 ± 2.1
	Impute All	0.028	0.131	169	78.07	86.84	**88.2**	100
Heart 20	MR	0.010 ± 0.000	0.091 ± 0.003	79.33 ± 1.53	85.4 ± 0.5	$\mathbf{86.4 \pm 0.8}$	86.02 ± 0.57	25.53 ± 6.0
	AMR	0.1400 ± 0.0012	0.2057 ± 0.0266	$\mathbf{3.11 \pm 0.32}$	84.44 ± 0.96	85.00 ± 0.00	84.76 ± 0.83	$\mathbf{3.84 \pm 0.4}$
	AC	0.011 ± 0.000	0.091 ± 0.001	80.00 ± 1.00	$\mathbf{84.4 \pm 1.0}$	81.2 ± 2.4	84.7 ± 1.4	16.30 ± 1.0
	Impute All	0.014	0.092	81	**85.88**	85.31	85.59	100
Heart 40	MR	0.009 ± 0.000	0.089 ± 0.004	77.00 ± 1.00	$\mathbf{85.0 \pm 0.8}$	83.7 ± 1.0	84.26 ± 0.34	27.37 ± 1.5
	AMR	0.1287 ± 0.0009	0.1598 ± 0.0006	$\mathbf{13.15 \pm 1.43}$	84.44 ± 0.96	85.00 ± 0.00	$\mathbf{84.88 \pm 0.52}$	$\mathbf{8.27 \pm 0.9}$
	AC	0.009 ± 0.000	0.086 ± 0.001	80.00 ± 2.65	80.1 ± 1.4	$\mathbf{81.0 \pm 2.6}$	82.6 ± 3.1	18.23 ± 3.6
	Impute All	0.012	0.090	159	85.31	**86.16**	84.18	100
Heart 60	MR	0.010 ± 0.000	0.090 ± 0.003	80.67 ± 1.53	$\mathbf{82.9 \pm 0.1}$	82.3 ± 0.6	84.54 ± 0.57	27.23 ± 1.8
	AMR	0.1493 ± 0.0011	0.1685 ± 0.0005	$\mathbf{29.23 \pm 2.88}$	$\mathbf{85.00 \pm 0.00}$	$\mathbf{85.00 \pm 0.00}$	$\mathbf{84.81 \pm 0.37}$	$\mathbf{12.18 \pm 1.2}$
	AC	0.010 ± 0.000	0.090 ± 0.001	81.33 ± 1.53	75.8 ± 1.4	$\mathbf{81.2 \pm 1.3}$	75.8 ± 1.0	20.87 ± 0.7
	Impute All	0.016	0.087	240	70.62	**83.33**	81.92	100
HIGGS 20	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	$\mathbf{358.7 \pm 8.8}$	Overtime	Overtime	$\mathbf{99.87 \pm 0.00}$	Overtime	Overtime	$\mathbf{0.98 \pm 0.00}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	$\mathbf{351.6 \pm 1.2}$	Overtime	Overtime	$\mathbf{99.87 \pm 0.00}$	Overtime	Overtime	$\mathbf{0.49 \pm 0.00}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	$\mathbf{351.4 \pm 0.4}$	Overtime	Overtime	$\mathbf{99.87 \pm 0.00}$	Overtime	Overtime	$\mathbf{0.33 \pm 0.00}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime

Table S4: Accuracy/time for SVM on data with injected MCAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Malware 20	MR	4534.6 ± 612.4	N/A	95.8 ± 0.47	N/A	18.68 ± 0.93
	AMR	$\mathbf{2104 \pm 236}$	N/A	95.62 ± 0.14	N/A	$\mathbf{3.17 \pm 0.25}$
	AC	3274 ± 176.8	N/A	91.60 ± 1.33	N/A	9.09 ± 0.61
	Impute All	6645.5	N/A	**96.70**	N/A	100
Malware 40	MR	3924.4 ± 158.9	N/A	$\mathbf{96.17 \pm 0.12}$	N/A	21.1 ± 1.22
	AMR	$\mathbf{2003 \pm 437}$	N/A	95.35 ± 0.47	N/A	$\mathbf{2.13 \pm 0.43}$
	AC	3078.5 ± 4.02	N/A	89.57 ± 0.58	N/A	5.32 ± 0.14
	Impute All	6347.8	N/A	95.75	N/A	100
Malware 60	MR	3546.5 ± 221.4	N/A	96.41 ± 0.15	N/A	16.65 ± 2.12
	AMR	$\mathbf{2199 \pm 302}$	N/A	94.68 ± 0.08	N/A	$\mathbf{2.51 \pm 0.17}$
	AC	2534.7 ± 8.11	N/A	89.32 ± 0.27	N/A	3.28 ± 0.17
	Impute All	6256.4	N/A	**96.70**	N/A	100
Tuadromd 20	MR	41.4 ± 0.58	N/A	$98.67 \pm 0.13^*$	N/A	11.9 ± 0.8
	AMR	$\mathbf{9.2 \pm 2.4}$	N/A	98.06 ± 0.19	N/A	$\mathbf{2.1 \pm 1.2}$
	AC	15.3 ± 0.41	N/A	97.63 ± 0.07	N/A	4.06 ± 0.64
	Impute All	45.60	N/A	$\mathbf{98.70}^*$	N/A	100
Tuadromd 40	MR	36.69 ± 0.29	N/A	98.5 ± 0.07	N/A	11.1 ± 1.26
	AMR	$\mathbf{9.3 \pm 0.8}$	N/A	97.76 ± 0.00	N/A	$\mathbf{1.3 \pm 0.4}$
	AC	13.4 ± 0.26	N/A	97.45 ± 0.10	N/A	3.3 ± 0.58
	Impute All	43.94	N/A	**98.84**	N/A	100
Tuadromd 60	MR	33.41 ± 0.17	N/A	98.3 ± 0.11	N/A	11.8 ± 1.4
	AMR	$\mathbf{10.3 \pm 0.7}^*$	N/A	98.23 ± 0.12	N/A	$\mathbf{1.48 \pm 0.7}^*$
	AC	$10.5 \pm 0.21^*$	N/A	96.90 ± 0.05	N/A	$1.87 \pm 0.19^*$
	Impute All	43.96	N/A	**98.66**	N/A	100
Parkinsons 20	MR	3.018 ± 0.019	425.00 ± 159.01	85.2 ± 0.5	$\mathbf{89.83 \pm 2.00}$	22.58 ± 9.1
	AMR	$\mathbf{1.9205 \pm 0.0188}$	219	87.18 ± 0.00	86.84	$\mathbf{9.84 \pm 0.8}$
	AC	3.025 ± 0.008	632.00 ± 28.83	79.6 ± 4.2	84.91 ± 3.34	45.1 ± 2.6
	Impute All	3.024	1794	**89.04**	88.95	100
Parkinsons 40	MR	3.166 ± 0.007	1021.00 ± 250.32	90.6 ± 2.2	86.84 ± 6.96	20.97 ± 4.6
	AMR	$\mathbf{2.5040 \pm 0.0094}$	465	89.74 ± 0.00	88.42	$\mathbf{12.43 \pm 1.1}$
	AC	3.191 ± 0.006	1130.00 ± 186.44	80.5 ± 3.3	82.11 ± 4.17	35.48 ± 1.3
	Impute All	3.175	3660	88.60	**92.11**	100
Parkinsons 60	MR	3.164 ± 0.023	1193.00 ± 421.54	$\mathbf{83.9 \pm 1.7}$	79.83 ± 1.52	23.40 ± 2.6
	AMR	$\mathbf{2.7153 \pm 0.0489}$	963	82.05 ± 6.78	**80.53**	$\mathbf{16.28 \pm 1.5}$
	AC	3.187 ± 0.007	1689.00 ± 160.18	73.4 ± 3.7	77.19 ± 13.24	36.17 ± 2.1
	Impute All	3.145	5868	81.58	76.32	100
Heart 20	MR	2.187 ± 0.017	470.00 ± 134.24	84.2 ± 0.9	85.31 ± 0.98	25.53 ± 6.0
	AMR	$\mathbf{1.5573 \pm 0.0435}$	78	$\mathbf{84.44 \pm 0.96}$	83.05	$\mathbf{3.84 \pm 0.4}$
	AC	2.142 ± 0.007	312	82.5 ± 2.2	86.44	16.30 ± 1.0
	Impute All	2.219	1797	81.92	**86.44**	100
Heart 40	MR	2.056 ± 0.006	1022.00 ± 63.38	81.9 ± 2.5	83.05 ± 2.94	27.37 ± 1.5
	AMR	$\mathbf{1.2646 \pm 0.0168}$	153	$\mathbf{85.00 \pm 0.00}$	76.27	$\mathbf{8.27 \pm 0.9}$
	AC	1.963 ± 0.010	594	80.2 ± 0.5	67.80	18.23 ± 3.6
	Impute All	2.085	3675	81.07	**88.14**	100
Heart 60	MR	2.209 ± 0.068	1519.00 ± 125.65	81.7 ± 1.1	$\mathbf{80.23 \pm 1.96}$	27.23 ± 1.8
	AMR	$\mathbf{1.7566 \pm 0.0084}$	540	$\mathbf{85.00 \pm 0.00}$	76.27	$\mathbf{12.18 \pm 1.2}$
	AC	2.125 ± 0.011	969	79.8 ± 1.7	77.97	20.87 ± 0.7
	Impute All	2.148	5532	81.36	77.80	100
HIGGS 20	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
HIGGS 40	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
HIGGS 60	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
Fraud 20	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	$\mathbf{0.98 \pm 0.00}$
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
Fraud 40	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	$\mathbf{0.49 \pm 0.00}$
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
Fraud 60	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	$\mathbf{0.33 \pm 0.00}$
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime

Table S5: Accuracy/time for SVM on data with injected MAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Malware 20	MR	11.16 ± 0.89	N/A	27135 ± 2453.1	$\mathbf{96.38 \pm 0.25}^*$	N/A	$\mathbf{96.45 \pm 0.13}$	21 ± 2.05
	AMR	3.14 ± 0.17	N/A	$\mathbf{3715 \pm 231}$	95.73 ± 0.00	N/A	95.89 ± 0.34	$\mathbf{1.57 \pm 0.16}$
	AC	$\mathbf{2.291 \pm 0.049}$	N/A	5295.3 ± 113.3	93.39 ± 1.54	N/A	94.85 ± 0.20	3.34 ± 0.15
	Impute All	17.71	N/A	Overtime	96.30^*	N/A	Overtime	100
Malware 40	MR	19.11 ± 1.00	N/A	Overtime	$\mathbf{96.16 \pm 0.24}^*$	N/A	Overtime	25.23 ± 2.96
	AMR	3.27 ± 0.25	N/A	$\mathbf{3967 \pm 216}$	95.73 ± 0.00	N/A	$\mathbf{96.48 \pm 0.00}$	$\mathbf{2.04 \pm 0.15}$
	AC	$\mathbf{2.178 \pm 0.106}$	N/A	10534 ± 1322.7	94.55 ± 0.95	N/A	93.89 ± 0.48	3.29 ± 0.22
	Impute All	36.10	N/A	Overtime	96.15^*	N/A	Overtime	100
Malware 60	MR	23.44 ± 0.88	N/A	Overtime	$95.67 \pm 0.61^*$	N/A	Overtime	19.69 ± 3.22
	AMR	5.04 ± 0.25	N/A	$\mathbf{6837 \pm 234}$	$95.64 \pm 0.52^*$	N/A	$\mathbf{96.06 \pm 0.15}$	$\mathbf{1.94 \pm 0.09}$
	AC	$\mathbf{1.807 \pm 0.036}$	N/A	15259 ± 1631.2	93.44 ± 1.35	N/A	93.01 ± 0.67	3.24 ± 0.23
	Impute All	48.33	N/A	Overtime	$\mathbf{95.85}^*$	N/A	Overtime	100
Tuadromd 20	MR	1.21 ± 0.003	98.65 ± 0.85	305.8 ± 5.13	$98.68 \pm 0.18^*$	$98.63 \pm 0.15^*$	98.70 ± 0.11	9.84 ± 0.92
	AMR	$\mathbf{0.70 \pm 0.21}^*$	$\mathbf{17.45 \pm 0.47}$	$\mathbf{58.4 \pm 10.4}$	98.32 ± 0.00	98.32 ± 0.00	98.14 ± 0.25	$\mathbf{0.9 \pm 0.4}$
	AC	$0.907 \pm 0.052^*$	100.85 ± 3.51	133.6 ± 6.84	$98.76 \pm 0.10^*$	$\mathbf{98.76 \pm 0.16}^*$	$\mathbf{98.89 \pm 0.07}$	3.87 ± 0.76
	Impute All	1.72	101.1	1673.4	$\mathbf{98.80}^*$	98.73^*	98.81	100
Tuadromd 40	MR	2.26 ± 0.01	95.7 ± 0.55	492.5 ± 10.4	$\mathbf{98.73 \pm 0.11}^*$	98.67 ± 0.10	$98.75 \pm 0.09^*$	10.19 ± 0.60
	AMR	0.98 ± 0.22	$\mathbf{11.61 \pm 0.52}$	$\mathbf{60.1 \pm 4.3}$	97.76 ± 0.00	98.32 ± 0.00	98.32 ± 0.00	$\mathbf{0.6 \pm 0.2}$
	AC	$\mathbf{0.686 \pm 0.013}$	93.65 ± 1.57	138.33 ± 4.92	$\mathbf{98.73 \pm 0.11}^*$	98.70 ± 0.07	$98.76 \pm 0.07^*$	3.01 ± 0.66
	Impute All	3.17	101.5	3451	98.70^*	**98.81**	$\mathbf{98.80}^*$	100
Tuadromd 60	MR	3.27 ± 0.03	82.4 ± 5.56	671 ± 35.8	$\mathbf{98.86 \pm 0.12}^*$	$\mathbf{98.71 \pm 0.02}^*$	$\mathbf{98.83 \pm 0.11}^*$	9.62 ± 0.55
	AMR	1.64 ± 0.16	$\mathbf{18.6 \pm 0.4}$	$\mathbf{80.1 \pm 5.7}$	98.23 ± 0.00	98.23 ± 0.00	98.23 ± 0.00	$\mathbf{0.5 \pm 0.2}$
	AC	$\mathbf{0.556 \pm 0.010}$	74.63 ± 1.10	165.7 ± 8.49	$98.64 \pm 0.19^*$	$98.60 \pm 0.09^*$	98.49 ± 0.13	2.32 ± 0.12
	Impute All	4.39	116.08	5036	98.82^*	$\mathbf{98.71}^*$	98.82^*	100
Parkinsons 20	MR	0.014 ± 0.000	0.140 ± 0.001	79.00 ± 1.73	88.3 ± 3.0	$\mathbf{90.1 \pm 1.7}$	87.04 ± 1.06	29.03 ± 0.0
	AMR	0.0932 ± 0.0003	0.1463 ± 0.0018	$\mathbf{8.73 \pm 1.02}$	$\mathbf{89.74 \pm 0.00}$	89.74 ± 0.00	$\mathbf{89.58 \pm 0.24}$	$\mathbf{10.27 \pm 1.2}$
	AC	0.015 ± 0.002	0.141 ± 0.002	81.67 ± 3.21	$\mathbf{83.2 \pm 4.0}$	82.6 ± 2.4	81.1 ± 3.0	35.48 ± 6.9
	Impute All	0.015	0.144	85	89.47	**90.35**	88.2	100
Parkinsons 40	MR	0.014 ± 0.001	0.131 ± 0.001	81.33 ± 2.08	$\mathbf{87.6 \pm 1.8}$	83.2 ± 1.0	83.19 ± 1.75	28.79 ± 0.0
	AMR	0.1073 ± 0.0021	0.1525 ± 0.0004	$\mathbf{23.09 \pm 2.16}$	88.03 ± 1.48	88.03 ± 1.48	$\mathbf{89.27 \pm 1.32}$	$\mathbf{13.91 \pm 1.3}$
	AC	0.013 ± 0.000	0.132 ± 0.001	84.33 ± 2.52	$\mathbf{80.8 \pm 1.2}$	77.0 ± 1.6	76.0 ± 4.8	24.24 ± 3.9
	Impute All	0.016	0.133	166	90.35	**91.23**	83.77	100
Parkinsons 60	MR	0.006 ± 0.000	0.118 ± 0.000	79.33 ± 2.31	77.3 ± 1.5	$\mathbf{80.4 \pm 2.5}$	77.78 ± 0.49	17.39 ± 3.1
	AMR	0.0914 ± 0.0002	0.1420 ± 0.0002	$\mathbf{27.90 \pm 2.72}$	$\mathbf{86.32 \pm 1.48}$	86.32 ± 1.48	$\mathbf{86.18 \pm 1.41}$	$\mathbf{16.41 \pm 1.6}$
	AC	0.007 ± 0.000	0.118 ± 0.003	83.67 ± 3.51	$\mathbf{78.5 \pm 3.0}$	75.9 ± 1.9	62.4 ± 5.5	32.61 ± 2.8
	Impute All	0.009	0.128	170	85.96	**87.72**	72.4	100
Heart 20	MR	0.006 ± 0.001	0.081 ± 0.007	74.00 ± 4.00	$\mathbf{85.0 \pm 1.1}$	84.1 ± 1.9	84.82 ± 0.47	11.81 ± 12.3
	AMR	0.1264 ± 0.0003	0.1595 ± 0.0012	$\mathbf{3.30 \pm 0.40}$	85.00 ± 1.67	85.00 ± 1.67	84.96 ± 1.31	$\mathbf{4.12 \pm 0.5}$
	AC	0.007 ± 0.000	0.077 ± 0.002	78.33 ± 3.51	$\mathbf{84.4 \pm 0.8}$	82.6 ± 1.5	82.2 ± 1.4	54.17 ± 18.0
	Impute All	0.008	0.079	80	**85.03**	**85.03**	**85.03**	100
Heart 40	MR	0.006 ± 0.000	0.098 ± 0.023	77.33 ± 3.06	77.4 ± 1.2	$\mathbf{78.0 \pm 2.2}$	82.87 ± 1.38	13.40 ± 3.9
	AMR	0.1230 ± 0.0010	0.1558 ± 0.0004	$\mathbf{13.84 \pm 1.26}$	$\mathbf{85.56 \pm 1.92}$	$\mathbf{85.56 \pm 1.92}$	$\mathbf{85.34 \pm 1.47}$	$\mathbf{8.76 \pm 0.8}$
	AC	0.006 ± 0.000	0.072 ± 0.000	79.67 ± 0.58	$\mathbf{75.2 \pm 1.5}$	72.0 ± 1.5	80.6 ± 0.5	30.93 ± 4.7
	Impute All	0.022	0.100	158	83.33	83.62	**84.18**	100
Heart 60	MR	0.012 ± 0.002	0.081 ± 0.001	78.33 ± 1.53	$\mathbf{56.6 \pm 3.5}$	$\mathbf{56.6 \pm 2.8}$	48.42 ± 0.13	28.78 ± 1.0
	AMR	0.0105 ± 0.0004	0.0421 ± 0.0003	$\mathbf{30.54 \pm 2.60}$	67.78 ± 0.96	$\mathbf{72.78 \pm 4.19}$	$\mathbf{67.13 \pm 4.78}$	$\mathbf{12.94 \pm 1.1}$
	AC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Impute All	0.092	0.084	236	**69.49**	58.76	48.59	100
HIGGS 20	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	$\mathbf{360.0 \pm 8.9}$	Overtime	Overtime	$\mathbf{99.88 \pm 0.00}$	Overtime	Overtime	$\mathbf{0.98 \pm 0.00}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	$\mathbf{349.1 \pm 2.4}$	Overtime	Overtime	$\mathbf{99.71 \pm 0.04}$	Overtime	Overtime	$\mathbf{0.49 \pm 0.00}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	$\mathbf{324.2 \pm 1.7}$	Overtime	Overtime	$\mathbf{99.83 \pm 0.00}$	Overtime	Overtime	$\mathbf{0.33 \pm 0.00}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime

Table S5: Accuracy/time for SVM on data with injected MAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Malware 20	MR	4726.8 ± 466.2	N/A	$96.29 \pm 0.34^*$	N/A	21 ± 2.05
	AMR	$\mathbf{2964 \pm 185}$	N/A	95.73 ± 0.00	N/A	$\mathbf{1.57 \pm 0.16}$
	AC	4177 ± 6.23	N/A	93.97 ± 0.96	N/A	3.34 ± 0.15
	Impute All	6830.2	N/A	$\mathbf{96.59}^*$	N/A	100
Malware 40	MR	5422.8 ± 835.9	N/A	$\mathbf{96.10 \pm 0.17}^*$	N/A	25.23 ± 2.96
	AMR	$\mathbf{2854 \pm 137}$	N/A	$96.06 \pm 0.38^*$	N/A	$\mathbf{2.04 \pm 0.15}$
	AC	5168.1 ± 90.4	N/A	92.14 ± 0.83	N/A	3.29 ± 0.22
	Impute All	6460.7	N/A	95.90	N/A	100
Malware 60	MR	4224.5 ± 549.4	N/A	95.65 ± 1.09	N/A	19.69 ± 3.22
	AMR	$\mathbf{2516 \pm 197}$	N/A	95.17 ± 0.08	N/A	$\mathbf{1.94 \pm 0.09}$
	AC	2968.5 ± 25.04	N/A	91.78 ± 2.45	N/A	3.24 ± 0.23
	Impute All	5794.3	N/A	**96.18**	N/A	100
Tuadromd 20	MR	39.74 ± 0.62	N/A	$98.67 \pm 0.20^*$	N/A	9.84 ± 0.92
	AMR	$\mathbf{8.5 \pm 1.4}$	N/A	98.32 ± 0.00	N/A	$\mathbf{0.9 \pm 0.4}$
	AC	39.10 ± 0.56	N/A	$98.66 \pm 0.07^*$	N/A	3.87 ± 0.76
	Impute All	44.55	N/A	$\mathbf{98.81}^*$	N/A	100
Tuadromd 40	MR	37.2 ± 0.23	N/A	$\mathbf{98.78 \pm 0.06}^*$	N/A	10.19 ± 0.60
	AMR	$\mathbf{9.9 \pm 1.6}$	N/A	98.01 ± 0.32	N/A	$\mathbf{0.6 \pm 0.2}$
	AC	34.52 ± 0.22	N/A	$\mathbf{98.78 \pm 0.11}^*$	N/A	3.01 ± 0.66
	Impute All	44.57	N/A	98.70^*	N/A	100
Tuadromd 60	MR	35.13 ± 0.11	N/A	$\mathbf{98.73 \pm 0.12}^*$	N/A	9.62 ± 0.55
	AMR	$\mathbf{11.3 \pm 0.4}$	N/A	$98.56 \pm 0.34^*$	N/A	$\mathbf{0.5 \pm 0.2}$
	AC	31.68 ± 0.24	N/A	$98.61 \pm 0.02^*$	N/A	2.32 ± 0.12
	Impute All	44.17	N/A	$\mathbf{98.73}^*$	N/A	100
Parkinsons 20	MR	3.562 ± 0.024	599.00 ± 91.60	86.7 ± 2.7	$\mathbf{91.23 \pm 4.02}$	29.03 ± 0.0
	AMR	$\mathbf{2.0728 \pm 0.0070}$	123	$\mathbf{89.74 \pm 0.00}$	89.47	$\mathbf{10.27 \pm 1.2}$
	AC	3.562 ± 0.014	558.00 ± 305.40	81.7 ± 2.3	87.72 ± 1.52	35.48 ± 6.9
	Impute All	3.609	1776	85.96	84.21	100
Parkinsons 40	MR	3.525 ± 0.013	922.00 ± 449.66	85.8 ± 1.4	77.37 ± 1.58	28.79 ± 0.0
	AMR	$\mathbf{2.4247 \pm 0.0321}$	780	$\mathbf{88.03 \pm 1.48}$	76.32	$\mathbf{13.91 \pm 1.3}$
	AC	3.548 ± 0.012	1051.00 ± 90.71	75.5 ± 5.6	75.79 ± 1.39	24.24 ± 3.9
	Impute All	3.507	4053	87.72	**78.95**	100
Parkinsons 60	MR	2.867 ± 0.019	537.00 ± 363.63	82.4 ± 0.36	86.58 ± 2.60	17.39 ± 3.1
	AMR	$\mathbf{1.9457 \pm 0.0190}$	639	85.47 ± 1.48	**86.84**	$\mathbf{16.41 \pm 1.6}$
	AC	2.864 ± 0.019	1345.00 ± 243.01	73.8 ± 3.2	N/A	32.61 ± 2.8
	Impute All	2.845	3339	86.40	N/A	100
Heart 20	MR	1.734 ± 0.016	211.00 ± 272.35	84.6 ± 0.3	84.18 ± 1.96	11.81 ± 12.3
	AMR	$\mathbf{1.1747 \pm 0.0367}$	75	$\mathbf{85.00 \pm 1.67}$	83.05	$\mathbf{4.12 \pm 0.5}$
	AC	1.721 ± 0.010	1071	82.7 ± 0.5	**86.44**	54.17 ± 18.0
	Impute All	1.794	1665	84.75	81.19	100
Heart 40	MR	1.796 ± 0.063	480.00 ± 170.79	75.2 ± 6.9	$\mathbf{79.61 \pm 2.98}$	13.40 ± 3.9
	AMR	$\mathbf{1.1684 \pm 0.0059}$	33	$\mathbf{85.56 \pm 1.92}$	72.88	$\mathbf{8.76 \pm 0.8}$
	AC	1.729 ± 0.007	1023	71.1 ± 0.9	72.88	30.93 ± 4.7
	Impute All	1.742	3312	83.62	76.10	100
Heart 60	MR	2.038 ± 0.019	1585.00 ± 43.09	44.1 ± 2.4	50.28 ± 9.34	28.78 ± 1.0
	AMR	$\mathbf{1.1303 \pm 0.0177}$	780	$\mathbf{65.56 \pm 13.47}$	42.37	$\mathbf{12.94 \pm 1.1}$
	AC	N/A	N/A	N/A	N/A	N/A
	Impute All	2.026	5328	40.40	**50.85**	100
HIGGS 20	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
HIGGS 40	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
HIGGS 60	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
Fraud 20	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	$\mathbf{0.98 \pm 0.00}$
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
Fraud 40	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	$\mathbf{0.49 \pm 0.00}$
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
Fraud 60	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	$\mathbf{0.33 \pm 0.00}$
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime

Table S6: Accuracy/time for SVM on data with injected MNAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Malware 20	MR	15.49 ± 0.70	N/A	43605 ± 2020.5	96.07 ± 0.44	N/A	$\mathbf{96.20 \pm 0.34}$	18.49 ± 0.85
	AMR	2.53 ± 0.14	N/A	$\mathbf{4046 \pm 264}$	95.16 ± 0.10	N/A	95.55 ± 0.25	$\mathbf{2.67 \pm 0.16}$
	AC	$\mathbf{2.211 \pm 0.12}$	N/A	6619 ± 1161.3	92.37 ± 0.746	N/A	92.74 ± 0.45	3.494 ± 0.567
	Impute All	28.96	N/A	Overtime	**96.55**	N/A	Overtime	100
Malware 40	MR	25.32 ± 1.30	N/A	Overtime	95.16 ± 0.76	N/A	Overtime	22.83 ± 1.4
	AMR	3.04 ± 0.21	N/A	$\mathbf{5812 \pm 165}$	95.03 ± 0.14	N/A	$\mathbf{95.19 \pm 0.22}^*$	$\mathbf{1.87 \pm 0.43}$
	AC	$\mathbf{1.525 \pm 0.096}$	N/A	10534 ± 1322.7	93.99 ± 1.747	N/A	93.48 ± 0.48	3.245 ± 0.121
	Impute All	50.34	N/A	Overtime	**96.52**	N/A	Overtime	100
Malware 60	MR	27 ± 1.94	N/A	Overtime	95.23 ± 0.40	N/A	Overtime	19.75 ± 2.66
	AMR	4.51 ± 0.28	N/A	$\mathbf{6091 \pm 264}$	94.04 ± 0.25	N/A	94.23 ± 0.17	$\mathbf{2.04 \pm 0.14}$
	AC	$\mathbf{0.946 \pm 0.0617}$	N/A	15259 ± 1631.2	94.27 ± 0.341	N/A	94.01 ± 0.67	3.363 ± 0.174
	Impute All	63.30	N/A	Overtime	**96.18**	N/A	Overtime	100
Tuadromd 20	MR	1.98 ± 0.06	96.31 ± 1.12	314.6 ± 13.6	$98.74 \pm 0.06^*$	$98.69 \pm 0.13^*$	$98.80 \pm 0.12^*$	10.11 ± 1.55
	AMR	$\mathbf{0.65 \pm 0.16}^*$	$\mathbf{12.46 \pm 0.95}$	$\mathbf{55.6 \pm 8.4}$	98.16 ± 0.04	97.89 ± 0.02	98.32 ± 0.00	$\mathbf{0.4 \pm 0.0}$
	AC	$0.785 \pm 0.057^*$	94.39 ± 2.04	133.6 ± 6.84	98.64 ± 0.033	$98.65 \pm 0.092^*$	98.69 ± 0.07	2.99 ± 0.755
	Impute All	2.55	100.5	3402.6	$\mathbf{98.80}^*$	$\mathbf{98.74}^*$	$\mathbf{98.82}^*$	100
Tuadromd 40	MR	3.37 ± 0.01	78.93 ± 1.45	612.3 ± 15.6	98.56 ± 0.05	98.62 ± 0.19	98.75 ± 0.1	9.95 ± 0.22
	AMR	1.34 ± 0.21	$\mathbf{10.46 \pm 0.19}$	$\mathbf{55.4 \pm 4.5}$	98.32 ± 0.00	98.32 ± 0.00	98.32 ± 0.00	$\mathbf{0.4 \pm 0.2}$
	AC	$\mathbf{0.515 \pm 0.004}$	94.39 ± 2.00	138.3 ± 4.92	98.21 ± 0.13	98.11 ± 0.25	98.26 ± 0.1	2.163 ± 0.182
	Impute All	4.39	103.71	6098.7	**98.70**	**98.87**	**98.87**	100
Tuadromd 60	MR	4.31 ± 0.01	72.36 ± 1.19	751.2 ± 21.4	98.1 ± 0.16	98.0 ± 0.16	98.14 ± 0.13	9.39 ± 0.27
	AMR	1.85 ± 0.17	$\mathbf{16.4 \pm 0.8}$	$\mathbf{68 \pm 4}$	98.06 ± 0.13	98.23 ± 0.00	97.85 ± 0.46	$\mathbf{0.3 \pm 0.0}$
	AC	$\mathbf{0.248 \pm 0.053}$	65.56 ± 2.93	165.7 ± 8.49	97.89 ± 0.114	97.94 ± 0.102	97.94 ± 0.13	2.138 ± 0.125
	Impute All	5.73	101.39	6090.7	**98.84**	**98.59**	**98.82**	100
Parkinsons 20	MR	0.008 ± 0.000	0.134 ± 0.002	84.33 ± 2.08	$\mathbf{86.2 \pm 2.1}$	88.4 ± 1.4	88.60 ± 1.76	20.00 ± 8.2
	AMR	0.6336 ± 0.0017	0.6831 ± 0.0221	$\mathbf{8.51 \pm 0.26}$	$\mathbf{89.74 \pm 0.00}$	89.74 ± 0.00	89.43 ± 0.22	$\mathbf{10.01 \pm 0.3}$
	AC	0.008 ± 0.000	0.130 ± 0.003	84.67 ± 0.58	80.9 ± 3.4	$\mathbf{81.8 \pm 4.6}$	85.2 ± 1.0	34.43 ± 2.0
	Impute All	0.010	0.134	85	84.87	**90.13**	89.47	100
Parkinsons 40	MR	0.010 ± 0.000	0.131 ± 0.003	82.33 ± 2.52	87.0 ± 4.1	88.3 ± 1.4	91.31 ± 1.92	28.57 ± 0.0
	AMR	0.2172 ± 0.0009	0.2664 ± 0.0003	$\mathbf{8.38 \pm 0.44}$	$\mathbf{89.74 \pm 0.00}$	$\mathbf{90.91 \pm 0.00}$	88.71 ± 0.93	$\mathbf{9.52 \pm 0.5}$
	AC	0.010 ± 0.000	0.127 ± 0.002	85.00 ± 1.73	$\mathbf{84.3 \pm 4.8}$	82.2 ± 2.9	85.0 ± 0.4	32.80 ± 8.2
	Impute All	0.012	0.133	88	84.21	82.46	85.96	100
Parkinsons 60	MR	0.008 ± 0.001	0.134 ± 0.007	82.00 ± 2.00	81.4 ± 5.0	$\mathbf{82.3 \pm 3.2}$	83.53 ± 0.93	29.08 ± 4.4
	AMR	0.6458 ± 0.0033	0.6954 ± 0.0037	$\mathbf{23.97 \pm 1.88}$	86.32 ± 1.48	86.32 ± 1.48	86.24 ± 1.29	$\mathbf{14.02 \pm 1.1}$
	AC	0.008 ± 0.000	0.125 ± 0.001	84.00 ± 2.65	$\mathbf{80.8 \pm 2.8}$	79.5 ± 3.9	81.8 ± 1.3	30.13 ± 1.6
	Impute All	0.011	0.127	171	87.28	87.72	86.84	100
Heart 20	MR	0.011 ± 0.002	0.090 ± 0.002	77.00 ± 1.00	85.4 ± 1.1	$\mathbf{86.3 \pm 1.0}$	84.82 ± 1.46	24.11 ± 1.0
	AMR	0.1272 ± 0.0001	0.1622 ± 0.0026	$\mathbf{3.41 \pm 0.24}$	85.00 ± 1.67	85.00 ± 1.67	$\mathbf{84.93 \pm 1.12}$	$\mathbf{4.26 \pm 0.3}$
	AC	0.010 ± 0.001	0.083 ± 0.003	79.33 ± 1.53	$\mathbf{84.0 \pm 0.8}$	$\mathbf{84.0 \pm 0.2}$	83.2 ± 1.2	14.90 ± 4.6
	Impute All	0.013	0.086	80	83.62	**86.44**	84.75	100
Heart 40	MR	0.010 ± 0.001	0.120 ± 0.039	76.67 ± 0.58	80.4 ± 0.3	79.2 ± 2.0	82.68 ± 0.69	14.58 ± 1.5
	AMR	0.1316 ± 0.0014	0.1617 ± 0.0006	$\mathbf{16.55 \pm 0.64}$	$\mathbf{85.00 \pm 0.00}$	$\mathbf{85.00 \pm 0.00}$	$\mathbf{86.19 \pm 0.46}$	$\mathbf{10.41 \pm 0.4}$
	AC	0.010 ± 0.000	0.081 ± 0.002	78.67 ± 2.08	73.8 ± 4.2	$\mathbf{76.3 \pm 2.6}$	80.4 ± 1.0	21.87 ± 3.0
	Impute All	0.017	0.096	159	68.36	82.49	**83.90**	100
Heart 60	MR	0.011 ± 0.003	0.089 ± 0.005	81.33 ± 0.58	72.3 ± 2.4	$\mathbf{73.9 \pm 1.5}$	77.50 ± 0.82	34.72 ± 1.0
	AMR	0.1426 ± 0.0007	0.1666 ± 0.0002	$\mathbf{32.05 \pm 1.67}$	$\mathbf{84.44 \pm 0.96}$	$\mathbf{85.63 \pm 0.96}$	$\mathbf{84.18 \pm 0.76}$	$\mathbf{13.41 \pm 0.7}$
	AC	0.008 ± 0.000	0.086 ± 0.001	82.00 ± 1.73	70.5 ± 1.3	$\mathbf{71.1 \pm 0.7}$	74.1 ± 1.4	19.90 ± 3.1
	Impute All	0.017	0.086	239	73.45	**76.84**	74.29	100
HIGGS 20	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	$\mathbf{303.5 \pm 9.5}$	Overtime	Overtime	$\mathbf{99.87 \pm 0.00}$	Overtime	Overtime	$\mathbf{0.98 \pm 0.00}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	$\mathbf{213.2 \pm 2.2}$	Overtime	Overtime	$\mathbf{99.87 \pm 0.00}$	Overtime	Overtime	$\mathbf{0.31 \pm 0.00}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	MR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AMR	$\mathbf{51.8 \pm 1.4}$	Overtime	Overtime	$\mathbf{99.87 \pm 0.00}$	Overtime	Overtime	$\mathbf{0.04 \pm 0.00}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime

Table S6: Accuracy/time for SVM on data with injected MNAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Malware 20	MR	5206.1 ± 812.6	N/A	$96.10 \pm 0.6^{*}$	N/A	18.49 ± 0.85
	AMR	$\mathbf{2166 \pm 162}$	N/A	95.73 ± 0.00	N/A	$\mathbf{2.67 \pm 0.16}$
	AC	5292 ± 110.6	N/A	92.68 ± 1.68	N/A	3.494 ± 0.567
	Impute All	5959.7	N/A	$\mathbf{96.52}^{*}$	N/A	100
Malware 40	MR	3622.7 ± 167.1	N/A	95.02 ± 0.1	N/A	22.83 ± 1.4
	AMR	1955 ± 98	N/A	95.24 ± 0.07	N/A	$\mathbf{1.87 \pm 0.43}$
	AC	$\mathbf{1659.4 \pm 3.94}$	N/A	93.63 ± 0.63	N/A	3.245 ± 0.121
	Impute All	7124.4	N/A	**96.33**	N/A	100
Malware 60	MR	1598.4 ± 243.7	N/A	95.13 ± 0.17	N/A	19.75 ± 2.66
	AMR	1986 ± 205	N/A	94.55 ± 0.09	N/A	$\mathbf{2.04 \pm 0.14}$
	AC	$\mathbf{1128.5 \pm 8.95}$	N/A	94.04 ± 0.30	N/A	3.363 ± 0.174
	Impute All	6061.5	N/A	**96.08**	N/A	100
Tuadromd 20	MR	37.55 ± 0.24	N/A	98.71 ± 0.1	N/A	10.11 ± 1.55
	AMR	$\mathbf{7.5 \pm 0.8}$	N/A	98.32 ± 0.00	N/A	$\mathbf{0.4 \pm 0.0}$
	AC	35.82 ± 0.059	N/A	98.62 ± 0.03	N/A	2.99 ± 0.755
	Impute All	46.34	N/A	**98.88**	N/A	100
Tuadromd 40	MR	34.62 ± 0.24	N/A	98.36 ± 0.02	N/A	9.95 ± 0.22
	AMR	$\mathbf{7.5 \pm 0.4}$	N/A	98.32 ± 0.00	N/A	$\mathbf{0.4 \pm 0.2}$
	AC	31.40 ± 0.21	N/A	98.13 ± 0.097	N/A	2.163 ± 0.182
	Impute All	44.13	N/A	**98.80**	N/A	100
Tuadromd 60	MR	33.77 ± 0.2	N/A	98.03 ± 0.12	N/A	9.39 ± 0.27
	AMR	$\mathbf{7.5 \pm 0.3}$	N/A	98.23 ± 0.00	N/A	$\mathbf{0.3 \pm 0.0}$
	AC	29.63 ± 0.24	N/A	97.58 ± 0.45	N/A	2.138 ± 0.125
	Impute All	43.86	N/A	**98.82**	N/A	100
Parkinsons 20	MR	3.064 ± 0.059	277.00 ± 191.54	90.4 ± 2.0	88.77 ± 1.85	20.00 ± 8.2
	AMR	$\mathbf{2.4559 \pm 0.0148}$	207	89.74 ± 0.00	80.53	$\mathbf{10.01 \pm 0.3}$
	AC	3.012 ± 0.020	669.00 ± 88.79	81.1 ± 2.8	$\mathbf{90.35 \pm 4.02}$	34.43 ± 2.0
	Impute All	3.002	1728	89.47	89.47	100
Parkinsons 40	MR	3.186 ± 0.017	806.00 ± 156.82	$\mathbf{89.8 \pm 3.0}$	91.23 ± 1.52	28.57 ± 0.0
	AMR	$\mathbf{2.2873 \pm 0.0179}$	363	88.03 ± 1.48	78.95	$\mathbf{9.52 \pm 0.5}$
	AC	3.144 ± 0.020	1068.00 ± 115.53	82.6 ± 4.2	82.46 ± 6.07	32.80 ± 8.2
	Impute All	3.171	3423	**85.53**	**87.37**	100
Parkinsons 60	MR	3.041 ± 0.007	871.00 ± 241.58	80.3 ± 5.3	72.81 ± 8.04	29.08 ± 4.4
	AMR	$\mathbf{2.6431 \pm 0.0558}$	867	87.55 ± 1.48	63.16	$\mathbf{14.02 \pm 1.1}$
	AC	2.996 ± 0.025	1412.00 ± 125.37	79.9 ± 3.4	72.81 ± 14.96	30.13 ± 1.6
	Impute All	3.018	5460	**91.23**	**84.21**	100
Heart 20	MR	2.239 ± 0.070	465.00 ± 36.37	$\mathbf{86.3 \pm 0.3}$	83.56 ± 2.66	24.11 ± 1.0
	AMR	$\mathbf{1.3154 \pm 0.0199}$	78	85.00 ± 1.67	83.05	$\mathbf{4.26 \pm 0.3}$
	AC	2.067 ± 0.016	204	82.5 ± 0.9	86.44	14.90 ± 4.6
	Impute All	2.182	1791	85.31	**89.83**	100
Heart 40	MR	2.129 ± 0.021	553.00 ± 69.28	$\mathbf{81.5 \pm 1.9}$	82.49 ± 3.91	14.58 ± 1.5
	AMR	$\mathbf{1.2990 \pm 0.0443}$	39	$\mathbf{85.00 \pm 0.00}$	80.49	$\mathbf{10.41 \pm 0.4}$
	AC	2.090 ± 0.019	645	72.8 ± 3.8	83.05	21.87 ± 3.0
	Impute All	2.204	3687	76.55	**83.93**	100
Heart 60	MR	2.019 ± 0.052	1585.00 ± 33.05	64.0 ± 1.5	82.49 ± 0.98	34.72 ± 1.0
	AMR	$\mathbf{1.6121 \pm 0.0395}$	1086	$\mathbf{84.44 \pm 0.96}$	**83.05**	$\mathbf{13.41 \pm 0.7}$
	AC	1.958 ± 0.012	984	63.8 ± 2.7	79.66	19.90 ± 3.1
	Impute All	1.995	8457	69.77	79.66	100
HIGGS 20	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
HIGGS 40	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
HIGGS 60	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
Fraud 20	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	$\mathbf{0.98 \pm 0.00}$
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
Fraud 40	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	$\mathbf{0.31 \pm 0.00}$
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime
Fraud 60	MR	Overtime	N/A	Overtime	N/A	Overtime
	AMR	Overtime	Overtime	Overtime	Overtime	$\mathbf{0.04 \pm 0.00}$
	AC	Overtime	N/A	Overtime	N/A	Overtime
	Impute All	Overtime	N/A	Overtime	N/A	Overtime

Table S7: Accuracy/time for logistic regression on data with injected MCAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)		Accuracy(%)		Impute % of Samples	
			AC	AMR	AC	AMR	AC	AMR
Malware	20	96.73	$\mathbf{2.70 \pm 0.20}$	5.9 ± 0.2	96.18 ± 0.18	$\mathbf{96.73 \pm 0.00}^*$	7.42 ± 1.61	$\mathbf{5.4 \pm 1.2}$
	40	96.73	$\mathbf{1.43 \pm 0.05}$	9.12 ± 4.89	95.96 ± 0.21	$\mathbf{96.73 \pm 0.00}^*$	7.85 ± 2.94	$\mathbf{5.69 \pm 3.29}^*$
	60	96.48	$\mathbf{0.93 \pm 0.08}$	9.35 ± 4.42	95.68 ± 0.16	$\mathbf{96.48 \pm 0.00}^*$	6.21 ± 2.18	$\mathbf{4.50 \pm 2.27}^*$
Tuadromd	20	98.43	$\mathbf{1.12 \pm 0.08}$	18.06 ± 3.37	97.84 ± 0.09	$\mathbf{98.17 \pm 0.06}$	6.14 ± 0.42	$\mathbf{4.06 \pm 0.00}$
	40	98.43	$\mathbf{0.76 \pm 0.07}$	16.18 ± 0.95	97.72 ± 0.10	$\mathbf{98.17 \pm 0.06}$	3.28 ± 0.36	$\mathbf{2.03 \pm 0.00}^*$
	60	98.32	$\mathbf{0.57 \pm 0.04}$	14.40 ± 1.42	97.65 ± 0.08	$\mathbf{98.13 \pm 0.05}^*$	2.42 ± 0.25	$\mathbf{1.35 \pm 0.00}$
HIGGS	20	64.13	Overtime	$\mathbf{175.745 \pm 91.272}$	Overtime	$\mathbf{64.1258 \pm 0.0018}$	Overtime	$\mathbf{0.0909 \pm 0.0819}$
	40	64.13	Overtime	$\mathbf{172.652 \pm 75.880}$	Overtime	$\mathbf{64.1367 \pm 0.0024}$	Overtime	$\mathbf{0.0417 \pm 0.0347}$
	60	64.13	Overtime	$\mathbf{139.654 \pm 19.792}$	Overtime	$\mathbf{64.1239 \pm 0.0000}$	Overtime	$\mathbf{0.0126 \pm 0.0044}$
Fraud	20	99.93	Error	$\mathbf{121.1 \pm 36.3}$	Error	$\mathbf{99.9306}$	Error	$\mathbf{1.24}$
	40	99.93	Error	$\mathbf{109.0 \pm 54.5}$	Error	$\mathbf{99.9253}$	Error	$\mathbf{0.98}$
	60	99.93	Error	$\mathbf{111.4 \pm 55.8}$	Error	$\mathbf{99.92}$	Error	$\mathbf{0.84}$
Parkinsons	20	92.31	$\mathbf{0.021}$	0.039	83.85 ± 0.00	$\mathbf{84.62 \pm 0.00}$	18.74	$\mathbf{12.90}$
	40	92.31	$\mathbf{0.148}$	0.312	86.41 ± 0.00	$\mathbf{87.18 \pm 0.00}$	15.93	$\mathbf{11.29}$
	60	92.31	$\mathbf{0.135}$	0.299	76.08 ± 0.00	$\mathbf{76.92 \pm 0.00}$	20.63	$\mathbf{14.89}$
Heart	20	83.33	$\mathbf{0.028}$	0.053	84.26 ± 0.00	$\mathbf{85.00 \pm 0.00}$	9.54	$\mathbf{6.38}$
	40	83.33	$\mathbf{0.082}$	0.135	82.71 ± 0.00	$\mathbf{83.33 \pm 0.00}$	24.86	$\mathbf{20.00}$
	60	83.33	$\mathbf{0.000}$	$\mathbf{0.000}$	86.11 ± 0.00	$\mathbf{86.67 \pm 0.00}$	2.63	$\mathbf{0.00}$

Table S8: Accuracy/time for logistic regression on data with injected MAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)		Accuracy(%)		Impute % of Samples	
			AC	AMR	AC	AMR	AC	AMR
Malware	20	96.74	$\mathbf{3.02 \pm 0.20}$	6.5 ± 0.3	96.21 ± 0.14	$\mathbf{96.73 \pm 0.00}^*$	5.93 ± 1.72	$\mathbf{4.2 \pm 0.7}^*$
	40	96.52	$\mathbf{1.30 \pm 1.10}$	5.1 ± 1.4	95.84 ± 0.17	$\mathbf{96.42 \pm 0.13}^*$	5.82 ± 1.03	$\mathbf{4.38 \pm 0.95}^*$
	60	96.48	$\mathbf{1.50 \pm 0.10}$	6.2 ± 1.5	95.73 ± 0.11	$\mathbf{96.42 \pm 0.08}$	4.08 ± 0.15	$\mathbf{2.97 \pm 0.08}$
Tuadromd	20	98.54	$\mathbf{1.06 \pm 0.05}$	3.68 ± 1.76	97.86 ± 0.10	$\mathbf{98.28 \pm 0.06}$	1.48 ± 0.32	$\mathbf{0.8 \pm 0.3}$
	40	98.43	$\mathbf{0.73 \pm 0.03}$	8.68 ± 1.10	97.94 ± 0.08	$\mathbf{98.39 \pm 0.06}$	1.86 ± 0.21	$\mathbf{1.10 \pm 0.11}$
	60	98.21	$\mathbf{0.48 \pm 0.06}$	14.32 ± 2.09	97.93 ± 0.07	$\mathbf{98.39 \pm 0.06}^*$	2.01 ± 0.18	$\mathbf{1.28 \pm 0.13}$
HIGGS	20	64.13	Overtime	$\mathbf{229.194 \pm 94.557}$	Overtime	$\mathbf{63.8363 \pm 0.0135}$	Overtime	$\mathbf{0.1288 \pm 0.0860}$
	40	64.13	Overtime	$\mathbf{289.810 \pm 17.548}$	Overtime	$\mathbf{63.6881 \pm 0.0060}$	Overtime	$\mathbf{0.08 \pm 0.00}$
	60	64.13	Overtime	$\mathbf{293.716 \pm 16.708}$	Overtime	$\mathbf{64.0893 \pm 0.0024}$	Overtime	$\mathbf{0.06 \pm 0.01}$
Fraud	20	99.93	Error	$\mathbf{112.5 \pm 53.5}$	Error	$\mathbf{99.9279}$	Error	$\mathbf{0.98}$
	40	99.93	Error	$\mathbf{127.7 \pm 27.0}$	Error	$\mathbf{99.9263}$	Error	$\mathbf{0.73}$
	60	99.93	Error	$\mathbf{132.0 \pm 31.7}$	Error	$\mathbf{99.9263}$	Error	$\mathbf{0.66}$
Parkinsons	20	92.31	$\mathbf{0.023}$	0.042	86.62 ± 0.00	$\mathbf{87.18 \pm 0.00}$	9.18	$\mathbf{6.45}$
	40	92.31	$\mathbf{0.020}$	0.039	86.43 ± 0.00	$\mathbf{87.18 \pm 0.00}$	24.73	$\mathbf{19.70}$
	60	92.31	$\mathbf{0.024}$	0.042	84.03 ± 0.00	$\mathbf{84.62 \pm 0.00}$	15.28	$\mathbf{11.96}$
Heart	20	83.33	$\mathbf{0.081}$	0.135	84.94 ± 0.96	$\mathbf{85.56 \pm 0.96}$	21.52	$\mathbf{16.67}$
	40	83.33	$\mathbf{0.083}$	0.137	82.74 ± 0.00	$\mathbf{83.33 \pm 0.00}$	21.64	$\mathbf{16.49}$
	60	83.33	$\mathbf{0.086}$	0.137	66.05 ± 0.00	$\mathbf{66.67 \pm 0.00}$	4.86	$\mathbf{1.44}$

Table S9: Accuracy/time for logistic regression on data with injected MNAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)		Accuracy(%)		Impute % of Samples	
			AC	AMR	AC	AMR	AC	AMR
Malware	20	96.48	$\mathbf{2.09 \pm 0.12}$	5.78 ± 1.24	93.96 ± 0.23	$\mathbf{96.35 \pm 0.29}$	5.06 ± 0.54	$\mathbf{3.5 \pm 0.7}^*$
	40	95.73	$\mathbf{1.40 \pm 0.20}$	5.0 ± 0.7	91.98 ± 0.19	$\mathbf{95.23 \pm 0.06}$	5.42 ± 0.49	$\mathbf{3.87 \pm 0.55}^*$
	60	96.23	$\mathbf{1.17 \pm 0.07}$	5.12 ± 0.46	92.42 ± 0.15	$\mathbf{95.98 \pm 0.04}$	6.31 ± 0.31	$\mathbf{4.98 \pm 0.17}$
Tuadromd	20	98.54	$\mathbf{0.77 \pm 0.08}$	5.26 ± 0.66	96.31 ± 0.08	$\mathbf{98.54 \pm 0.00}^*$	1.36 ± 0.15	$\mathbf{0.78 \pm 0.08}$
	40	98.43	$\mathbf{0.47 \pm 0.04}$	6.98 ± 1.37	95.74 ± 0.10	$\mathbf{98.36 \pm 0.06}$	1.51 ± 0.22	$\mathbf{0.74 \pm 0.26}$
	60	98.21	$\mathbf{0.21 \pm 0.20}$	5.24 ± 0.55	94.98 ± 0.11	$\mathbf{98.43 \pm 0.00}$	1.65 ± 0.19	$\mathbf{0.81 \pm 0.14}^*$
HIGGS	20	64.13	Overtime	$\mathbf{301.501 \pm 2.575}$	Overtime	$\mathbf{63.7033 \pm 0.0013}$	Overtime	$\mathbf{0.59 \pm 0.02}$
	40	64.13	Overtime	$\mathbf{316.734 \pm 4.407}$	Overtime	$\mathbf{63.5840 \pm 0.0013}$	Overtime	$\mathbf{0.39 \pm 0.01}$
	60	64.13	Overtime	$\mathbf{262.412 \pm 22.343}$	Overtime	$\mathbf{62.8452 \pm 0.0003}$	Overtime	$\mathbf{0.28 \pm 0.00}$
Fraud	20	99.93	Error	$\mathbf{120.6 \pm 37.7}$	Error	$\mathbf{99.9114}$	Error	$\mathbf{0.393}$
	40	99.93	Error	$\mathbf{99.4 \pm 39.0}$	Error	$\mathbf{99.8709}$	Error	$\mathbf{0.246}$
	60	99.93	Error	$\mathbf{103.3 \pm 41.6}$	Error	$\mathbf{99.8709}$	Error	$\mathbf{0.164}$
Parkinsons	20	92.31	$\mathbf{0.162}$	0.295	84.36 ± 0.00	$\mathbf{87.18 \pm 0.00}$	14.21	$\mathbf{10.00}$
	40	92.31	$\mathbf{0.061}$	0.108	70.92 ± 0.00	$\mathbf{74.36 \pm 0.00}$	13.46	$\mathbf{9.52}$
	60	92.31	$\mathbf{0.133}$	0.244	83.61 ± 0.00	$\mathbf{87.18 \pm 0.00}$	22.58	$\mathbf{17.02}$
Heart	20	83.33	$\mathbf{0.236}$	0.436	80.82 ± 0.00	$\mathbf{83.33 \pm 0.00}$	4.82	$\mathbf{0.00}$
	40	83.33	$\mathbf{0.031}$	0.054	80.76 ± 0.96	$\mathbf{83.89 \pm 0.96}$	13.64	$\mathbf{9.38}$
	60	83.33	$\mathbf{0.089}$	0.140	63.28 ± 0.00	$\mathbf{66.67 \pm 0.00}$	30.82	$\mathbf{25.00}$

Table S10: Accuracy/time for logistic regression on data with original missingness using model-based imputation — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Breast Cancer	AMR	0.04 ± 0.00	0.04 ± 0.00	$\mathbf{37 \pm 2}$	$\mathbf{93.57 \pm 0.00}$	$\mathbf{93.57 \pm 0.00}^*$	$93.49 \pm 0.32^*$	$\mathbf{6.67 \pm 0.00}$
	AC	0.06 ± 0.00	0.10 ± 0.00	74 ± 8	$\mathbf{94.05 \pm 0.41}$	$\mathbf{93.57 \pm 0.00}^*$	$\mathbf{93.57 \pm 0.00}^*$	71.11 ± 10.18
	Impute All	$\mathbf{0.03}$	$\mathbf{0.03}$	98	93.57	$\mathbf{93.57}^*$	$\mathbf{93.57}^*$	100
Water-Potability	AMR	1.06 ± 0.51	1.33 ± 0.79	$\mathbf{8.5 \pm 0.6}$	60.98 ± 0.00	$60.98 \pm 0.00^*$	$\mathbf{60.98 \pm 0.00}^*$	$\mathbf{0.03 \pm 0.01}$
	AC	0.37 ± 0.04	0.76 ± 0.02	83.14 ± 5.48	$\mathbf{61.13 \pm 0.00}^*$	$\mathbf{61.08 \pm 0.11}^*$	$\mathbf{60.98 \pm 0.00}^*$	1.94 ± 0.21
	Impute All	$\mathbf{0.01}$	$\mathbf{0.02}$	1462	$\mathbf{61.13}^*$	60.98^*	$\mathbf{60.98}^*$	100
Online-Ed	AMR	1.44	$1.59 \pm 0.10^*$	$\mathbf{5.1 \pm 0.3}$	$\mathbf{63.86 \pm 0.00}^*$	$\mathbf{63.86 \pm 0.00}^*$	$\mathbf{63.86 \pm 0.00}^*$	$\mathbf{0.01 \pm 0.00}$
	AC	2.08 ± 0.04	2.94 ± 0.26	20 ± 1	$\mathbf{63.86 \pm 0.00}^*$	$\mathbf{63.86 \pm 0.00}^*$	$\mathbf{63.86 \pm 0.00}^*$	0.73 ± 0.19
	Impute All	$\mathbf{0.9}$	$\mathbf{1.5}^*$	3621	63.85	$\mathbf{63.86}^*$	$\mathbf{63.86}^*$	100
Bankruptcy	AMR	1.09 ± 0.03	$\mathbf{0.72 \pm 0.14}$	$\mathbf{133 \pm 4}$	$\mathbf{94.84 \pm 0.10}$	$\mathbf{94.80 \pm 0.10}$	$\mathbf{94.80 \pm 0.10}$	$\mathbf{0.77 \pm 0.03}$
	AC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Impute All	$\mathbf{0.5}$	1.98	7599	93.19	93.15	93.15	100

Table S10: Accuracy/time for logistic regression on data with original missingness using model-based imputation — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Breast Cancer	AMR	0.07 ± 0.00	N/A	$93.33 \pm 0.41^*$	N/A	$\mathbf{6.67 \pm 0.00}$
	AC	$\mathbf{0.06 \pm 0.00}$	N/A	$\mathbf{93.57 \pm 0.00}^*$	N/A	71.11 ± 10.18
	Impute All	1.4	N/A	$\mathbf{93.57}^*$	N/A	100
Water-Potability	AMR	1.24 ± 0.59	N/A	60.98 ± 0.00	N/A	$\mathbf{0.03 \pm 0.01}$
	AC	$\mathbf{0.38 \pm 0.04}$	N/A	$\mathbf{61.08 \pm 0.09}$	N/A	1.94 ± 0.21
	Impute All	10.8	N/A	60.98	N/A	100
Online-Ed	AMR	$\mathbf{1.57 \pm 0.05}$	N/A	$\mathbf{63.86 \pm 0.00}^*$	N/A	$\mathbf{0.01 \pm 0.00}$
	AC	2.11 ± 0.05	N/A	$\mathbf{63.86 \pm 0.00}^*$	N/A	0.73 ± 0.19
	Impute All	16.4	N/A	$\mathbf{63.86}^*$	N/A	100
Bankruptcy	AMR	$\mathbf{3.25 \pm 0.20}$	N/A	$\mathbf{94.88 \pm 0.15}$	N/A	$\mathbf{0.77 \pm 0.03}$
	AC	N/A	N/A	N/A	N/A	N/A
	Impute All	689	N/A	93.15	N/A	100

Table S11: Accuracy/time for logistic regression on data with injected MCAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Malware 20	AMR	8.57 ± 0.45	N/A	$\mathbf{11354 \pm 4518}^*$	$96.69 \pm 0.15^*$	N/A	$96.71 \pm 0.24^*$	$\mathbf{8.15 \pm 4.11}^*$
	AC	$\mathbf{3.80 \pm 0.32}$	N/A	$19534 \pm 7146^*$	$\mathbf{96.73 \pm 0.00}^*$	N/A	96.68 ± 0.17	$13.58 \pm 7.87^*$
	Impute All	15.9	N/A	Overtime	$\mathbf{96.73}^*$	N/A	Overtime	100
Malware 40	AMR	9.95 ± 1.29	N/A	$\mathbf{14327 \pm 742}$	$96.39 \pm 0.76^*$	N/A	$\mathbf{96.87 \pm 0.05}$	$\mathbf{4.60 \pm 1.48}^*$
	AC	$\mathbf{2.27 \pm 0.15}$	N/A	19571 ± 3447	$\mathbf{96.73 \pm 0.00}^*$	N/A	96.72 ± 0.02	$7.31 \pm 2.16^*$
	Impute All	29.14	N/A	Overtime	$\mathbf{96.73}^*$	N/A	Overtime	100
Malware 60	AMR	8.39 ± 0.86	N/A	$\mathbf{8172 \pm 364}$	$96.38 \pm 0.15^*$	N/A	96.47 ± 0.02	$\mathbf{2.65 \pm 0.53}$
	AC	$\mathbf{1.39 \pm 0.25}$	N/A	15716 ± 615	$\mathbf{96.48 \pm 0.00}^*$	N/A	$96.51 \pm 0.04^*$	5.72 ± 0.87
	Impute All	40.17	N/A	Overtime	$\mathbf{96.48}^*$	N/A	Overtime	100
Tuadromd 20	AMR	15.63 ± 0.63	$\mathbf{14.39 \pm 5.32}$	141 ± 8	98.13 ± 0.06	98.21 ± 0.00	98.23 ± 0.02	$4.06 \pm 0.00^*$
	AC	$\mathbf{1.19 \pm 0.12}$	71.95 ± 24.44	$\mathbf{101 \pm 12}$	98.28 ± 0.13	98.21 ± 0.00	98.30 ± 0.04	$\mathbf{3.36 \pm 0.92}^*$
	Impute All	3.14	113.2	2037	**98.43**	**98.43**	**98.46**	100
Tuadromd 40	AMR	16.43 ± 2.24	$\mathbf{19.40 \pm 2.51}$	$\mathbf{107 \pm 5}$	98.09 ± 0.00	98.47 ± 0.03	98.49 ± 0.09	$\mathbf{2.03 \pm 0.00}^*$
	AC	$\mathbf{0.96 \pm 0.12}$	69.81 ± 9.16	124 ± 8	$98.51 \pm 0.06^*$	98.33 ± 0.11	98.57 ± 0.02	$2.50 \pm 0.76^*$
	Impute All	4.71	110.3	3922	$\mathbf{98.54}^*$	**98.54**	**98.60**	100
Tuadromd 60	AMR	15.59 ± 3.43	$\mathbf{16.59 \pm 2.98}$	$\mathbf{98 \pm 3}$	98.09 ± 0.00	98.09 ± 0.00	98.41 ± 0.07	$\mathbf{1.35 \pm 0.00}$
	AC	$\mathbf{0.73 \pm 0.13}$	62.75 ± 21.14	189 ± 14	98.09 ± 0.00	98.06 ± 0.06	98.24 ± 0.05	2.44 ± 0.51
	Impute All	7.1	124	6517	**98.65**	**98.32**	**98.69**	100
Parkinsons 20	AMR	0.06 ± 0.00	0.12 ± 0.00	2.66	87.18 ± 0.00	87.18 ± 0.00	87.12 ± 0.08	**3.2**
	AC	0.041 ± 0.003	0.086 ± 0.004	37.43 ± 2.16	86.72 ± 0.24	86.61 ± 0.31	86.54 ± 0.27	45.1 ± 2.6
	Impute All	0.043	0.082	83	87.36	87.29	87.41	100
Parkinsons 40	AMR	0.11 ± 0.00	0.16 ± 0.00	8.54	94.87 ± 0.00	94.87 ± 0.00	94.79 ± 0.12	**9.7**
	AC	0.082 ± 0.004	0.116 ± 0.005	31.24 ± 1.23	94.25 ± 0.18	94.11 ± 0.24	94.19 ± 0.20	35.5 ± 1.4
	Impute All	0.077	0.120	88	94.98	94.92	95.03	100
Parkinsons 60	AMR	0.26 ± 0.00	0.30 ± 0.01	19.66	88.89 ± 1.48	88.89 ± 1.48	88.97 ± 1.31	**11.7**
	AC	0.162 ± 0.008	0.222 ± 0.011	60.82 ± 3.70	88.18 ± 1.62	88.31 ± 1.55	88.07 ± 1.42	36.2 ± 2.2
	Impute All	0.169	0.210	168	89.06	89.12	89.20	100
Heart 20	AMR	0.08 ± 0.00	0.10 ± 0.01	5.12	86.67 ± 0.00	86.67 ± 0.00	86.58 ± 0.14	**6.4**
	AC	0.059 ± 0.003	0.071 ± 0.004	13.04 ± 0.80	86.21 ± 0.42	86.06 ± 0.39	85.98 ± 0.36	16.3 ± 1.0
	Impute All	0.056	0.074	80	86.84	86.75	86.92	100
Heart 40	AMR	0.02 ± 0.00	0.05 ± 0.00	1.76	83.33 ± 0.00	83.33 ± 0.00	83.27 ± 0.11	**1.1**
	AC	0.014 ± 0.001	0.038 ± 0.002	29.12 ± 5.76	82.91 ± 0.28	82.83 ± 0.34	82.71 ± 0.31	18.2 ± 3.6
	Impute All	0.013	0.036	160	83.46	83.39	83.52	100
Heart 60	AMR	0.14 ± 0.00	0.17 ± 0.00	3.33	83.33 ± 0.00	83.33 ± 0.00	83.48 ± 0.18	**1.4**
	AC	0.101 ± 0.005	0.116 ± 0.006	49.74 ± 1.67	82.79 ± 0.47	82.62 ± 0.52	82.91 ± 0.45	20.9 ± 0.7
	Impute All	0.105	0.110	238	83.41	83.55	83.48	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	**107.1**	Overtime	Overtime	**99.87**	Overtime	Overtime	**0.0049**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	**105.9**	Overtime	Overtime	**99.87**	Overtime	Overtime	**0.0025**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	**106.3**	Overtime	Overtime	**99.87**	Overtime	Overtime	**0.0016**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime

Table S11: Accuracy/time for logistic regression on data with injected MCAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Malware 20	AMR	$\mathbf{2053 \pm 307}^*$	Overtime	$96.32 \pm 0.53^*$	Overtime	$\mathbf{8.15 \pm 4.11}^*$
	AC	$2857 \pm 513^*$	Overtime	$\mathbf{96.73 \pm 0.00}^*$	Overtime	$13.58 \pm 7.87^*$
	Impute All	6141	Overtime	$\mathbf{96.73}^*$	Overtime	100
Malware 40	AMR	$\mathbf{2137 \pm 198}$	Overtime	$96.72 \pm 0.01^*$	Overtime	$\mathbf{4.60 \pm 1.48}^*$
	AC	2955 ± 261	Overtime	$\mathbf{96.73 \pm 0.00}^*$	Overtime	$7.31 \pm 2.16^*$
	Impute All	6417	Overtime	$\mathbf{96.73}^*$	Overtime	100
Malware 60	AMR	$\mathbf{2099 \pm 87}$	Overtime	$96.43 \pm 0.08^*$	Overtime	$\mathbf{2.65 \pm 0.53}$
	AC	3002 ± 427	Overtime	$\mathbf{96.48 \pm 0.00}^*$	Overtime	5.72 ± 0.87
	Impute All	6301	Overtime	$\mathbf{96.48}^*$	Overtime	100
Tuadromd 20	AMR	$\mathbf{49.84 \pm 1.58}^*$	Overtime	98.21 ± 0.00	Overtime	$4.06 \pm 0.00^*$
	AC	96.42 ± 4.07	Overtime	98.21 ± 0.00	Overtime	$\mathbf{3.36 \pm 0.92}^*$
	Impute All	50.7^*	Overtime	**98.43**	Overtime	100
Tuadromd 40	AMR	$\mathbf{50.07 \pm 0.15}^*$	Overtime	$98.39 \pm 0.21^*$	Overtime	$\mathbf{2.03 \pm 0.00}^*$
	AC	$98.51 \pm 0.06^*$	Overtime	$\mathbf{98.51 \pm 0.06}^*$	Overtime	$2.50 \pm 0.76^*$
	Impute All	54.3	Overtime	98.43	Overtime	100
Tuadromd 60	AMR	51.64 ± 0.22	Overtime	$98.29 \pm 0.05^*$	Overtime	$\mathbf{1.35 \pm 0.00}$
	AC	87.53 ± 10.86	Overtime	98.21 ± 0.00	Overtime	2.44 ± 0.51
	Impute All	**49.4**	Overtime	$\mathbf{98.32}^*$	Overtime	100
Parkinsons 20	AMR	1.90 ± 0.05	246.0	87.18 ± 0.00	86.84	**3.2**
	AC	2.76 ± 0.08	363.0 ± 63.9	86.77 ± 0.22	86.84 ± 0.00	45.1 ± 2.6
	Impute All	3.06	1794.0	87.33	86.84	100
Parkinsons 40	AMR	2.18 ± 0.02	498.0	94.87 ± 0.00	86.84	**9.7**
	AC	2.69 ± 0.07	787.0 ± 203.2	94.32 ± 0.17	86.84 ± 0.00	35.5 ± 1.4
	Impute All	3.21	3660.0	94.96	**89.47**	100
Parkinsons 60	AMR	2.79 ± 0.03	882.0	88.89 ± 1.48	**89.47**	**11.7**
	AC	2.84 ± 0.09	1193.0 ± 156.9	88.24 ± 1.58	89.47 ± 0.00	36.2 ± 2.2
	Impute All	3.17	5868.0	89.14	86.84	100
Heart 20	AMR	1.71 ± 0.06	135.0	86.67 ± 0.00	83.05	**6.4**
	AC	1.94 ± 0.05	441.0 ± 37.6	86.15 ± 0.40	84.75 ± 0.00	16.3 ± 1.0
	Impute All	2.24	1797.0	86.81	**86.44**	100
Heart 40	AMR	1.15 ± 0.00	48.0	83.33 ± 0.00	83.05	**1.1**
	AC	1.97 ± 0.04	839.0 ± 63.7	82.88 ± 0.29	83.05 ± 0.00	18.2 ± 3.6
	Impute All	2.11	3675.0	83.44	**88.14**	100
Heart 60	AMR	1.34 ± 0.02	96.0	83.33 ± 0.00	84.75	**1.4**
	AC	2.05 ± 0.06	1384.0 ± 273.3	82.75 ± 0.49	82.49 ± 0.98	20.9 ± 0.7
	Impute All	2.18	5532.0	83.50	**88.14**	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	**0.0049**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	**0.0025**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	**0.0016**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime

Table S12: Accuracy/time for logistic regression on data with injected MAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Malware 20	AMR	8.81 ± 0.21	N/A	$\mathbf{11842 \pm 4621}^*$	$96.71 \pm 0.02^*$	N/A	$96.74 \pm 0.19^*$	$\mathbf{7.84 \pm 0.54}$
	AC	$\mathbf{3.65 \pm 0.34}$	N/A	$18952 \pm 6941^*$	$\mathbf{96.73 \pm 0.00}^*$	N/A	$96.71 \pm 0.03^*$	12.77 ± 3.24
	Impute All	16.4	N/A	Overtime	$\mathbf{96.73}^*$	N/A	Overtime	100
Malware 40	AMR	10.21 ± 1.35	N/A	$\mathbf{14612 \pm 718}$	96.47 ± 0.25	N/A	$\mathbf{96.75 \pm 0.01}$	$\mathbf{4.30 \pm 0.84}^*$
	AC	$\mathbf{2.18 \pm 0.14}$	N/A	18985 ± 3552	$96.48 \pm 0.00^*$	N/A	96.50 ± 0.03	$7.15 \pm 2.05^*$
	Impute All	28.31	N/A	Overtime	$\mathbf{96.49}^*$	N/A	Overtime	100
Malware 60	AMR	8.52 ± 0.81	N/A	$\mathbf{7954 \pm 378}$	$96.40 \pm 0.15^*$	N/A	$\mathbf{96.55 \pm 0.04}$	$\mathbf{2.75 \pm 0.51}$
	AC	$\mathbf{1.45 \pm 0.27}$	N/A	16045 ± 591	96.23 ± 0.00	N/A	96.17 ± 0.04	5.51 ± 0.91
	Impute All	41.22	N/A	Overtime	$\mathbf{96.44}^*$	N/A	Overtime	100
Tuadromd 20	AMR	15.12 ± 0.65	$\mathbf{13.95 \pm 5.51}$	145 ± 7	98.36 ± 0.06	98.32 ± 0.00	$\mathbf{98.56 \pm 0.02}^*$	$\mathbf{3.75 \pm 0.21}$
	AC	$\mathbf{1.25 \pm 0.11}$	74.12 ± 23.51	$\mathbf{105 \pm 11}$	98.21 ± 0.06	$\mathbf{98.54 \pm 0.00}$	98.34 ± 0.02	4.03 ± 0.05
	Impute All	3.05	109.8	2095	$\mathbf{98.54}$	98.53	98.54^*	100
Tuadromd 40	AMR	16.85 ± 2.15	$\mathbf{18.85 \pm 2.65}$	$\mathbf{102 \pm 6}$	98.32 ± 0.00	98.44 ± 0.02	$\mathbf{98.63 \pm 0.03}^*$	$\mathbf{1.85 \pm 0.03}$
	AC	$\mathbf{0.99 \pm 0.11}$	71.25 ± 8.85	129 ± 7	$\mathbf{98.65 \pm 0.00}$	$\mathbf{98.65 \pm 0.07}$	$98.61 \pm 0.04^*$	2.74 ± 0.23
	Impute All	4.55	113.5	3815	98.43	98.43	98.62^*	100
Tuadromd 60	AMR	15.15 ± 3.55	$\mathbf{17.15 \pm 2.85}$	$\mathbf{95 \pm 4}$	$\mathbf{98.32 \pm 0.00}$	$\mathbf{98.32 \pm 0.00}$	$\mathbf{98.37 \pm 0.03}$	$\mathbf{1.22 \pm 0.00}$
	AC	$\mathbf{0.75 \pm 0.12}$	60.85 ± 22.15	181 ± 15	98.13 ± 0.06	98.09 ± 0.11	98.20 ± 0.02	2.64 ± 0.13
	Impute All	7.3	128	6325	$\mathbf{98.43}$	98.21	98.27	100
Parkinsons 20	AMR	0.25 ± 0.01	0.29 ± 0.00	8.34	89.74 ± 0.00	89.74 ± 0.00	89.66 ± 0.10	$\mathbf{9.7}$
	AC	0.174 ± 0.009	0.208 ± 0.010	33.20 ± 3.53	89.27 ± 0.24	89.18 ± 0.29	89.21 ± 0.26	38.6 ± 4.1
	Impute All	0.180	0.200	86	89.88	89.81	89.93	100
Parkinsons 40	AMR	0.08 ± 0.00	0.13 ± 0.00	4.92	87.18 ± 0.00	87.18 ± 0.00	87.09 ± 0.12	$\mathbf{3.0}$
	AC	0.053 ± 0.003	0.096 ± 0.005	39.69 ± 6.40	86.71 ± 0.31	86.60 ± 0.35	86.52 ± 0.33	24.2 ± 3.9
	Impute All	0.050	0.100	164	87.32	87.25	87.39	100
Parkinsons 60	AMR	0.24 ± 0.00	0.29 ± 0.00	16.86	87.18 ± 0.00	87.18 ± 0.00	87.04 ± 0.14	$\mathbf{9.8}$
	AC	0.164 ± 0.008	0.198 ± 0.010	56.07 ± 4.82	86.20 ± 0.42	86.31 ± 0.39	86.12 ± 0.46	32.6 ± 2.8
	Impute All	0.170	0.190	172	87.29	87.36	87.42	100
Heart 20	AMR	0.14 ± 0.00	0.17 ± 0.00	13.53	86.67 ± 0.00	86.67 ± 0.00	86.82 ± 0.14	$\mathbf{16.7}$
	AC	0.105 ± 0.005	0.116 ± 0.006	43.90 ± 14.58	86.24 ± 0.37	86.12 ± 0.43	86.31 ± 0.40	54.2 ± 18.0
	Impute All	0.100	0.120	81	86.80	86.74	86.88	100
Heart 40	AMR	0.19 ± 0.00	0.20 ± 0.00	25.91	86.67 ± 0.00	86.67 ± 0.00	87.92 ± 0.13	$\mathbf{16.5}$
	AC	0.115 ± 0.006	0.157 ± 0.008	48.51 ± 7.38	86.05 ± 0.34	85.92 ± 0.38	85.84 ± 0.41	30.9 ± 4.7
	Impute All	0.120	0.150	157	86.76	87.91	86.91	100
Heart 60	AMR	0.30 ± 0.00	0.32 ± 0.00	190.16	80.00 ± 0.00	80.00 ± 0.00	79.72 ± 0.42	$\mathbf{79.9}$
	AC	0.221 ± 0.011	0.209 ± 0.010	210.87 ± 5.71	79.12 ± 0.58	79.24 ± 0.51	78.81 ± 0.64	88.6 ± 2.4
	Impute All	0.230	0.200	238	80.18	80.27	80.35	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	$\mathbf{102.6}$	Overtime	Overtime	$\mathbf{99.87}$	Overtime	Overtime	$\mathbf{0.0020}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	$\mathbf{99.6}$	Overtime	Overtime	$\mathbf{99.87}$	Overtime	Overtime	$\mathbf{0.0005}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	$\mathbf{100.0}$	Overtime	Overtime	$\mathbf{99.87}$	Overtime	Overtime	$\mathbf{0.0003}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime

Table S12: Accuracy/time for logistic regression on data with injected MAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Malware 20	AMR	$\mathbf{2115 \pm 298}^*$	Overtime	$96.65 \pm 0.15^*$	Overtime	$\mathbf{7.84 \pm 0.54}$
	AC	$2794 \pm 531^*$	Overtime	$\mathbf{96.73 \pm 0.00}^*$	Overtime	12.77 ± 3.24
	Impute All	5982	Overtime	$\mathbf{96.73}^*$	Overtime	100
Malware 40	AMR	$\mathbf{2185 \pm 187}$	Overtime	$\mathbf{96.71 \pm 0.02}$	Overtime	$\mathbf{4.30 \pm 0.84}^*$
	AC	2871 ± 275	Overtime	96.48 ± 0.00	Overtime	$7.15 \pm 2.05^*$
	Impute All	6215	Overtime	96.48	Overtime	100
Malware 60	AMR	$\mathbf{2145 \pm 92}$	Overtime	$96.57 \pm 0.38^*$	Overtime	$\mathbf{2.75 \pm 0.51}$
	AC	2915 ± 445	Overtime	96.21 ± 0.03	Overtime	5.51 ± 0.91
	Impute All	6452	Overtime	$\mathbf{96.48}^*$	Overtime	100
Tuadromd 20	AMR	$\mathbf{48.35 \pm 1.65}^*$	Overtime	98.36 ± 0.06	Overtime	$\mathbf{3.75 \pm 0.21}$
	AC	98.15 ± 3.95	Overtime	$\mathbf{98.54 \pm 0.00}^*$	Overtime	4.03 ± 0.05
	Impute All	49.2^*	Overtime	$\mathbf{98.54}^*$	Overtime	100
Tuadromd 40	AMR	$\mathbf{51.12 \pm 0.16}$	Overtime	98.62 ± 0.04	Overtime	$\mathbf{1.85 \pm 0.03}$
	AC	93.45 ± 12.15	Overtime	$98.58 \pm 0.13^*$	Overtime	2.74 ± 0.23
	Impute All	55.8	Overtime	$\mathbf{98.65}^*$	Overtime	100
Tuadromd 60	AMR	50.15 ± 0.24	Overtime	$98.41 \pm 0.11^*$	Overtime	$\mathbf{1.22 \pm 0.00}$
	AC	89.25 ± 10.15	Overtime	$\mathbf{98.47 \pm 0.28}^*$	Overtime	2.64 ± 0.13
	Impute All	**47.9**	Overtime	98.20^*	Overtime	100
Parkinsons 20	AMR	2.24 ± 0.02	186.0	89.74 ± 0.00	86.84	**9.7**
	AC	3.22 ± 0.10	366.0 ± 99.1	89.32 ± 0.23	86.84 ± 0.00	38.6 ± 4.1
	Impute All	3.63	1776.0	89.86	**89.47**	100
Parkinsons 40	AMR	2.42 ± 0.03	807.0	87.18 ± 0.00	78.95	**3.0**
	AC	3.04 ± 0.09	884.0 ± 64.1	87.89 ± 0.30	76.32 ± 0.00	24.2 ± 3.9
	Impute All	3.54	4053.0	87.30	**86.84**	100
Parkinsons 60	AMR	2.21 ± 0.03	654.0	87.18 ± 0.00	71.05	**9.8**
	AC	3.12 ± 0.11	972.0 ± 150.8	86.27 ± 0.41	68.42 ± 0.00	32.6 ± 2.8
	Impute All	2.88	3339.0	87.35	N/A	100
Heart 20	AMR	1.24 ± 0.01	318.0	86.67 ± 0.00	88.14	**16.7**
	AC	1.71 ± 0.05	869.0 ± 127.6	86.18 ± 0.39	88.70 ± 0.98	54.2 ± 18.0
	Impute All	1.82	1665.0	86.79	**89.83**	100
Heart 40	AMR	2.08 ± 0.04	663.0	86.67 ± 0.00	69.49	**16.5**
	AC	1.64 ± 0.05	804.0 ± 258.8	86.00 ± 0.36	72.03 ± 3.60	30.9 ± 4.7
	Impute All	1.77	3312.0	86.84	**81.36**	100
Heart 60	AMR	2.22 ± 0.04	5055.0	75.00 ± 0.00	59.32	**79.9**
	AC	2.18 ± 0.08	912 ± 289	74.08 ± 0.73	57.10 ± 1.24	88.6 ± 2.4
	Impute All	2.05	5328.0	80.24	**76.27**	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	**0.0020**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	**0.0005**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	**0.0003**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime

Table S13: Accuracy/time for logistic regression on data with injected MNAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Malware 20	AMR	8.24 ± 3.61	N/A	$\mathbf{10890 \pm 4750}^*$	96.61 ± 0.11	N/A	$96.69 \pm 0.08^*$	$\mathbf{8.12 \pm 3.75}^*$
	AC	$\mathbf{4.05 \pm 0.30}$	N/A	$20140 \pm 7500^*$	$\mathbf{96.73}^*$	N/A	$\mathbf{96.75 \pm 0.04}^*$	$13.15 \pm 7.65^*$
	Impute All	16.8	N/A	Overtime	$\mathbf{96.73}^*$	N/A	Overtime	100
Malware 40	AMR	9.45 ± 1.18	N/A	$\mathbf{13850 \pm 780}$	95.47 ± 0.14	N/A	95.57 ± 0.24	$\mathbf{5.05 \pm 1.25}$
	AC	$\mathbf{2.41 \pm 0.16}$	N/A	20450 ± 3250	$\mathbf{95.81 \pm 0.15}^*$	N/A	95.71 ± 0.09	7.45 ± 2.42
	Impute All	27.50	N/A	Overtime	95.73^*	N/A	Overtime	100
Malware 60	AMR	8.75 ± 0.92	N/A	$\mathbf{8510 \pm 345}$	96.05 ± 0.05	N/A	96.11 ± 0.07	$\mathbf{2.95 \pm 0.52}$
	AC	$\mathbf{1.31 \pm 0.23}$	N/A	14950 ± 645	95.48 ± 0.25	N/A	95.99 ± 0.13	6.15 ± 0.85
	Impute All	38.50	N/A	Overtime	$\mathbf{96.23}$	N/A	Overtime	100
Tuadromd 20	AMR	16.15 ± 0.67	$\mathbf{13.85 \pm 5.05}$	135 ± 7	98.32 ± 0.00	$\mathbf{98.56 \pm 0.06}^*$	98.47 ± 0.03	3.65 ± 0.00
	AC	$\mathbf{1.12 \pm 0.13}$	68.50 ± 25.80	$\mathbf{95 \pm 13}$	$\mathbf{98.54}^*$	98.54^*	$\mathbf{98.57 \pm 0.04}^*$	$\mathbf{3.35 \pm 0.92}^*$
	Impute All	2.95	118.5	1945	98.54^*	98.54^*	98.56^*	100
Tuadromd 40	AMR	15.85 ± 2.41	$\mathbf{20.35 \pm 2.35}$	$\mathbf{112 \pm 4}^*$	98.36 ± 0.06	$\mathbf{98.48 \pm 0.13}^*$	98.52 ± 0.04	$\mathbf{2.05 \pm 0.00}^*$
	AC	$\mathbf{0.89 \pm 0.11}$	74.50 ± 9.85	$118 \pm 9^*$	$98.54 \pm 0.16^*$	$98.39 \pm 0.06^*$	98.49 ± 0.04	$2.45 \pm 0.85^*$
	Impute All	4.95	105.2	4110	$\mathbf{98.45}^*$	98.43^*	$\mathbf{98.61}$	100
Tuadromd 60	AMR	16.25 ± 3.15	$\mathbf{15.85 \pm 3.15}$	$\mathbf{92 \pm 2}$	$\mathbf{98.32}$	$\mathbf{98.31 \pm 0.04}$	$\mathbf{98.42 \pm 0.10}^*$	$\mathbf{1.35 \pm 0.00}$
	AC	$\mathbf{0.79 \pm 0.14}$	65.50 ± 19.50	198 ± 13	98.17 ± 0.06	98.17 ± 0.06	98.21 ± 0.08	2.65 ± 0.52
	Impute All	6.6	118	6850	98.30	98.21	98.37^*	100
Parkinsons 20	AMR	0.10 ± 0.00	0.15 ± 0.00	8.40	92.31 ± 0.00	92.31 ± 0.00	92.22 ± 0.12	$\mathbf{10.0}$
	AC	0.073 ± 0.004	0.096 ± 0.005	28.90 ± 1.68	90.48 ± 0.72	89.96 ± 0.84	90.67 ± 0.69	34.4 ± 2.0
	Impute All	0.070	0.100	84	92.44	92.38	92.51	100
Parkinsons 40	AMR	0.25 ± 0.00	0.29 ± 0.00	8.46	92.31 ± 0.00	94.87 ± 0.00	94.61 ± 0.24	$\mathbf{9.5}$
	AC	0.154 ± 0.008	0.229 ± 0.011	29.19 ± 7.30	89.24 ± 1.12	91.18 ± 1.03	90.72 ± 1.08	32.8 ± 8.2
	Impute All	0.160	0.220	89	92.46	94.96	95.02	100
Parkinsons 60	AMR	0.25 ± 0.00	0.30 ± 0.00	16.32	87.18 ± 0.00	87.18 ± 0.00	88.41 ± 0.54	$\mathbf{9.6}$
	AC	0.197 ± 0.010	0.174 ± 0.009	51.17 ± 2.72	82.64 ± 1.47	83.92 ± 1.35	82.83 ± 1.42	30.1 ± 1.6
	Impute All	0.190	0.180	170	87.32	87.25	87.40	100
Heart 20	AMR	0.02 ± 0.00	0.06 ± 0.00	1.66	83.33 ± 0.00	83.33 ± 0.00	83.47 ± 0.13	$\mathbf{2.1}$
	AC	0.014 ± 0.001	0.043 ± 0.002	11.77 ± 3.63	81.96 ± 0.74	81.72 ± 0.81	82.06 ± 0.70	14.9 ± 4.6
	Impute All	0.013	0.045	79	83.46	83.39	83.52	100
Heart 40	AMR	0.07 ± 0.00	0.10 ± 0.00	15.04	81.67 ± 0.00	82.84 ± 0.00	81.55 ± 0.15	$\mathbf{9.4}$
	AC	0.050 ± 0.003	0.069 ± 0.003	35.04 ± 4.80	78.84 ± 1.02	79.16 ± 0.94	78.52 ± 1.10	21.9 ± 3.0
	Impute All	0.052	0.065	160	81.82	81.75	83.06	100
Heart 60	AMR	0.08 ± 0.00	0.10 ± 0.00	22.99	80.00 ± 0.00	81.67 ± 0.00	80.94 ± 0.61	$\mathbf{9.7}$
	AC	0.063 ± 0.003	0.060 ± 0.003	47.16 ± 7.35	75.62 ± 1.36	77.18 ± 1.22	76.71 ± 1.31	19.9 ± 3.1
	Impute All	0.060	0.063	237	80.14	81.82	81.73	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	$\mathbf{96.6}$	Overtime	Overtime	$\mathbf{99.87}$	Overtime	Overtime	$\mathbf{0.0020}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	$\mathbf{100.9}$	Overtime	Overtime	$\mathbf{99.87}$	Overtime	Overtime	$\mathbf{0.0005}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	$\mathbf{100.9}$	Overtime	Overtime	$\mathbf{99.87}$	Overtime	Overtime	$\mathbf{0.0003}$
	AC	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime

Table S13: Accuracy/time for logistic regression on data with injected MNAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Malware 20	AMR	$\mathbf{1950 \pm 285}^*$	Overtime	$96.60 \pm 0.25^*$	Overtime	$\mathbf{8.12 \pm 3.75}^*$
	AC	$2715 \pm 540^*$	Overtime	$\mathbf{96.73}^*$	Overtime	$13.15 \pm 7.65^*$
	Impute All	5840	Overtime	$\mathbf{96.73}^*$	Overtime	100
Malware 40	AMR	$\mathbf{2045 \pm 205}$	Overtime	$\mathbf{95.99 \pm 0.17}^*$	Overtime	$\mathbf{5.05 \pm 1.25}$
	AC	3105 ± 245	Overtime	$95.90 \pm 0.15^*$	Overtime	7.45 ± 2.42
	Impute All	6710	Overtime	95.73	Overtime	100
Malware 60	AMR	$\mathbf{2185 \pm 95}$	Overtime	$96.09 \pm 0.33^*$	Overtime	$\mathbf{2.95 \pm 0.52}$
	AC	2850 ± 405	Overtime	95.56 ± 0.58	Overtime	6.15 ± 0.85
	Impute All	5990	Overtime	$\mathbf{96.23}^*$	Overtime	100
Tuadromd 20	AMR	$\mathbf{51.25 \pm 1.45}$	Overtime	$98.53 \pm 0.06^*$	Overtime	3.65 ± 0.00
	AC	99.15 ± 4.35	Overtime	98.54^*	Overtime	$\mathbf{3.35 \pm 0.92}^*$
	Impute All	53.1	Overtime	98.54^*	Overtime	100
Tuadromd 40	AMR	$\mathbf{48.55 \pm 0.16}$	Overtime	$98.40 \pm 0.16^*$	Overtime	$\mathbf{2.05 \pm 0.00}^*$
	AC	91.15 ± 10.50	Overtime	98.24 ± 0.13	Overtime	$2.45 \pm 0.85^*$
	Impute All	51.5	Overtime	$\mathbf{98.43}^*$	Overtime	100
Tuadromd 60	AMR	53.15 ± 0.20	Overtime	$98.35 \pm 0.09^*$	Overtime	$\mathbf{1.35 \pm 0.00}$
	AC	83.15 ± 11.45	Overtime	$\mathbf{98.47 \pm 0.28}^*$	Overtime	2.65 ± 0.52
	Impute All	**52.1**	Overtime	98.21^*	Overtime	100
Parkinsons 20	AMR	1.92 ± 0.03	180.0	92.31 ± 0.00	**89.47**	**10.0**
	AC	2.88 ± 0.08	378.0 ± 198.1	90.21 ± 0.78	89.47 ± 0.00	34.4 ± 2.0
	Impute All	3.04	1728.0	92.43	84.21	100
Parkinsons 40	AMR	2.34 ± 0.05	351.0	94.87 ± 0.00	76.32	**9.5**
	AC	2.82 ± 0.09	807.0 ± 124.9	91.54 ± 0.96	76.32 ± 0.00	32.8 ± 8.2
	Impute All	3.20	3423.0	94.81	76.32	100
Parkinsons 60	AMR	2.36 ± 0.04	957.0	85.47 ± 1.48	84.21	**9.6**
	AC	2.76 ± 0.08	1326.0 ± 143.5	81.26 ± 1.56	89.47 ± 0.00	30.1 ± 1.6
	Impute All	3.06	5460.0	88.54	**86.84**	100
Heart 20	AMR	1.19 ± 0.02	51.0	83.33 ± 0.00	**84.75**	**2.1**
	AC	1.94 ± 0.05	525.0 ± 100.3	81.84 ± 0.77	83.05 ± 2.94	14.9 ± 4.6
	Impute All	2.21	1791.0	83.45	83.05	100
Heart 40	AMR	1.61 ± 0.03	432.0	81.67 ± 0.00	76.27	**9.4**
	AC	2.05 ± 0.06	966.0 ± 42.4	80.11 ± 0.98	72.03 ± 1.20	21.9 ± 3.0
	Impute All	2.23	3687.0	81.83	**84.75**	100
Heart 60	AMR	1.81 ± 0.00	660.0	78.89 ± 0.96	**84.75**	**9.7**
	AC	2.01 ± 0.05	1602.0	74.36 ± 1.48	47.46	19.9 ± 3.1
	Impute All	2.16	8457.0	81.56	81.36	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	**0.0020**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	**0.0005**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	**0.0003**
	AC	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime

Table S14: Accuracy/time for Linear Regression on data with injected MCAR

Data Set	% Missing	Ground Truth MSE	Time(s)			MSE			Impute % of Samples or Features		
			AC	MR	AMR	AC	MR	AMR	AC	MR	AMR
Superconductivity	20	0.0088	14.75±0.30	**2.305**	2.45±0.12	**0.008825±0.000002**	0.00888	0.0912±0.003	0.28±0.04	70.00	**0.18±0.02**
	40	0.0088	17.60±0.09	2.534	**2.05±0.27**	**0.008840±0.000001***	0.00885	0.00890±0.0001*	0.28±0.06	75.00	**0.15±0.01**
	60	0.0088	13.08±0.36	2.476	**1.85±0.08**	0.008919±0.000001	**0.00885**	0.095±0.004	0.30±0.02	75.00	**0.12±0.00**
Gas	20	0.1053	0.131±0.040	0.31	**0.065±0.003**	0.1053±0.0023*	**0.105***	0.108±0.002	6.59±0.40	65.00	**1.55±0.10**
	40	0.1053	0.137±0.008	0.3391	**0.082±0.005**	**0.0997±0.0009**	0.1054	0.115±0.000	5.98±0.20	65.00	**1.42±0.05**
	60	0.1053	0.116±0.031	0.551	**0.045±0.002**	0.1948±0.0278	0.112*	**0.110±0.004***	8.07±0.66	25.00	**0.95±0.00**
Concrete	20	0.0149	0.0737±0.0033	0.0227	**0.0185±0.001**	**0.014543±0.000021**	0.01495	0.0154±0.000	8.33±2.46	50.00	**4.12±0.25**
	40	0.0149	0.0787±0.0163	0.0202	**0.0112±0.004**	**0.014872±0.000112***	0.01495*	0.0165±0.002*	7.40±0.62	50.00	**3.55±0.15**
	60	0.0149	0.0473±0.0034	0.0199	**0.0088±0.001**	**0.014705±0.000249***	0.01495*	0.0152±0.001*	6.01±0.67	50.00	**2.80±0.08**

Table S15: Accuracy/time for Linear Regression on data with injected MAR

Data Set	% Missing	Ground Truth MSE	Time(s)			MSE			Impute % of Samples or Features		
			AC	MR	AMR	AC	MR	AMR	AC	MR	AMR
Superconductivity	20	0.0088	12.50 ± 0.14	$\mathbf{0.1105}$	13.20 ± 0.55	0.0106 ± 0.0001	$\mathbf{0.0089}$	0.0098 ± 0.0002	0.36 ± 0.06	55.6	$\mathbf{0.21 \pm 0.03}$
	40	0.0088	14.46 ± 0.09	$\mathbf{0.1128}$	9.85 ± 0.42	0.0111 ± 0.0001	$\mathbf{0.0089}$	0.0102 ± 0.0001	0.27 ± 0.02	33.3	$\mathbf{0.15 \pm 0.00}$
	60	0.0088	10.71 ± 0.31	$\mathbf{0.1257}$	11.50 ± 0.20	0.0127 ± 0.0001	0.0089	$\mathbf{0.0085 \pm 0.0001}$	0.32 ± 0.06	44.4	$\mathbf{0.19 \pm 0.02}$
Gas	20	0.1053	0.1020 ± 0.0342	$\mathbf{0.0262}$	0.085 ± 0.012	0.1438 ± 0.0043	$\mathbf{0.1033}$	0.115 ± 0.003	4.39 ± 1.00	69.2	$\mathbf{2.15 \pm 0.40}$
	40	0.1053	0.1927 ± 0.0059	$\mathbf{0.0270}$	0.227 ± 0.0150	0.1921 ± 0.0375	$\mathbf{0.1039}$	0.203 ± 0.039	4.10 ± 0.75	53.8	$\mathbf{1.85 \pm 0.25}$
	60	0.1053	0.1968 ± 0.0169	$\mathbf{0.0254}$	0.156 ± 0.028	0.2082 ± 0.0045	$\mathbf{0.1019}$	0.1505 ± 0.0050	4.33 ± 0.14	38.5	$\mathbf{2.05 \pm 0.00}$
Concrete	20	0.0148	0.0695 ± 0.0038	$\mathbf{0.0040}$	0.076 ± 0.004	$0.0151 \pm 0.0001^*$	$\mathbf{0.0150}^*$	$0.0160 \pm 0.0012^*$	7.47 ± 2.29	50	$\mathbf{4.20 \pm 0.37}$
	40	0.0148	0.0623 ± 0.0008	$\mathbf{0.0039}$	0.0510 ± 0.0029	0.0157 ± 0.0001	$\mathbf{0.0150}$	0.0153 ± 0.0001	6.46 ± 0.14	50	$\mathbf{3.15 \pm 0.10}$
	60	0.0148	0.0494 ± 0.0011	$\mathbf{0.0040}$	0.0550 ± 0.0020	0.0376 ± 0.0117	$\mathbf{0.0150}^*$	$0.021 \pm 0.006^*$	6.21 ± 0.69	50	$\mathbf{2.95 \pm 0.00}$

Table S16: Accuracy/time for Linear Regression on data with injected MNAR

Data Set	% Missing	Ground Truth MSE	Time(s)			MSE			Impute % of Samples or Features		
			AC	MR	AMR	AC	MR	AMR	AC	MR	AMR
Superconductivity	20	0.0088	15.04 ± 0.97	$\mathbf{0.1164}$	14.23 ± 1.34	0.0107 ± 0.0001	$\mathbf{0.0090}$	0.0098 ± 0.0003	$0.28 \pm 0.08^*$	30	$\mathbf{0.22 \pm 0.06}^*$
	40	0.0088	9.79 ± 0.38	$\mathbf{0.1266}$	10.12 ± 0.76	0.0108 ± 0.0001	$\mathbf{0.0089}$	0.0094 ± 0.0004	0.30 ± 0.01	40	$\mathbf{0.18 \pm 0.07}$
	60	0.0088	4.41 ± 0.29	$\mathbf{0.1005}$	3.87 ± 0.45	0.0124 ± 0.0003	0.0089^*	$\mathbf{0.0087 \pm 0.0003}^*$	0.37 ± 0.07	40	$\mathbf{0.13 \pm 0.04}$
Gas	20	0.1053	0.1238 ± 0.0374	$\mathbf{0.0260}$	0.1381 ± 0.0452	0.1452 ± 0.0224	$\mathbf{0.1043}^*$	$0.1182 \pm 0.0235^*$	11.48 ± 2.67	64.28	$\mathbf{4.82 \pm 1.87}$
	40	0.1053	0.1115 ± 0.0175	$\mathbf{0.0256}$	0.0963 ± 0.0287	0.1633 ± 0.0025	$\mathbf{0.1049}$	0.1712 ± 0.0314	8.75 ± 0.55	50	$\mathbf{3.61 \pm 0.94}$
	60	0.1053	0.0773 ± 0.0102	$\mathbf{0.0243}$	0.0824 ± 0.0312	0.1825 ± 0.0528	$\mathbf{0.1019}^*$	$0.1347 \pm 0.0405^*$	7.12 ± 1.37	42.8	$\mathbf{2.87 \pm 1.04}$
Concrete	20	0.0148	0.0677 ± 0.0001	$\mathbf{0.0039}$	0.0592 ± 0.0054	0.0155 ± 0.0001	$\mathbf{0.0150}^*$	$0.0152 \pm 0.0002^*$	7.23 ± 0.80	50	$\mathbf{4.12 \pm 0.84}$
	40	0.0148	0.0440 ± 0.0006	$\mathbf{0.0039}$	0.0481 ± 0.0031	$\mathbf{0.0150 \pm 0.0002}^*$	$\mathbf{0.0150}^*$	$\mathbf{0.0150 \pm 0.0003}^*$	5.50 ± 0.31	50	$\mathbf{3.27 \pm 0.52}$
	60	0.0148	0.0273 ± 0.0010	$\mathbf{0.0042}$	0.0238 ± 0.0025	0.0163 ± 0.0003	$\mathbf{0.0150}$	0.0156 ± 0.0004	5.72 ± 0.14	50	$\mathbf{2.41 \pm 0.39}$

Table S17: Accuracy/time for Linear Regression on data with original missing using model-based imputation — KNN, MICE, TCSDI

Data Set	Method	Time(s)			MSE			Impute % of Samples or Features
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Cancer Rate	MR	$\mathbf{0.153}$	0.574	5852	$\mathbf{0.0045}^*$	$\mathbf{0.0045}^*$	$\mathbf{0.0045}$	33.3
	AMR	0.40 ± 0.05	1.03 ± 0.08	$\mathbf{43.97 \pm 2.85}$	0.0049 ± 0.0000	0.0059 ± 0.0002	0.0050 ± 0.0001	$\mathbf{0.55 \pm 0.02}$
	AC	0.31 ± 0.06	$\mathbf{0.26 \pm 0.01}$	110.5 ± 2.8	0.0047 ± 0.0000	$0.0046 \pm 0.0001^*$	0.0049 ± 0.0000	2.03 ± 0.07
	Impute All	0.584	0.664	6104	$\mathbf{0.0045}^*$	0.0058	0.0049	100
Air Quality	MR	1.06	2.46	14976	$\mathbf{5.671}^*$	$\mathbf{5.74}$	$\mathbf{5.82}^*$	50
	AMR	1.12 ± 0.09	1.88 ± 0.22	$\mathbf{66.3 \pm 4.7}$	$5.68 \pm 0.12^*$	5.95 ± 0.15	$5.85 \pm 0.18^*$	$0.92 \pm 0.07^*$
	AC	$\mathbf{0.62 \pm 0.09}$	$\mathbf{0.51 \pm 0.10}$	98.5 ± 3.1	6.70 ± 0.49	6.75 ± 0.51	7.19 ± 0.46	$\mathbf{0.85 \pm 0.14}^*$
	Impute All	1.763	2.46	18372	5.672	5.923	5.825	100

Table S17: Accuracy/time for Linear Regression on data with original missing using model-based imputation — MF, LLMF

Data Set	Method	Time(s)		MSE		Impute % of Samples or Features
		MF	LLMF	MF	LLMF	
Cancer Rate	MR	6.89	Overtime	$\mathbf{0.0045}^*$	Overtime	33.3
	AMR	$\mathbf{0.66 \pm 0.07}$	Overtime	0.0048 ± 0.0000	Overtime	$\mathbf{0.55 \pm 0.02}$
	AC	3.71 ± 0.02	Overtime	$0.0050 \pm 0.0005^*$	Overtime	2.03 ± 0.07
	Impute All	7.24	Overtime	0.0047	Overtime	100
Air Quality	MR	2.24	Overtime	$\mathbf{5.71}^*$	Overtime	50
	AMR	2.13 ± 0.18	Overtime	$5.78 \pm 0.11^*$	Overtime	$0.92 \pm 0.07^*$
	AC	$\mathbf{1.52 \pm 0.11}$	Overtime	6.68 ± 0.48	Overtime	$\mathbf{0.85 \pm 0.14}^*$
	Impute All	2.51	Overtime	5.752	Overtime	100

Table S18: Accuracy/time for Linear Regression on data with injected MCAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			MSE			Impute % of Samples or Features
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Superconductivity 20	MR	2.369	**33.16***	29165	0.0089	0.00897	0.00904	70
	AMR	0.12±0.02	35.5 ± 5.1*	**18.5 ± 2.4**	0.0095±0.0001	0.0090 ± 0.0002*	0.0094±0.0000	**0.04 ± 0.01**
	AC	14.40±0.23	53.16±3.24	96.4±0.56	0.008825±0.000000	**0.008824 ± 0.000000***	0.0092±0.00	0.2548±0.0945
	Impute All	**0.0692**	34.17	30515	**0.0088**	0.008939	**0.009013**	100
Superconductivity 40	MR	2.599	**33.73**	29735	0.0089	0.00924	**0.00914***	75.00
	AMR	0.16±0.03	45.2±6.3	**9.3 ± 1.1**	0.0091±0.0000	0.0095 ± 0.0009*	0.0092 ± 0.0001*	**0.02 ± 0.00**
	AC	17.38±1.87	52.21±1.50	100.4±0.78	**0.008841 ± 0.000002**	**0.008842 ± 0.000000***	0.0092±0.00	0.2792±0.0509
	Impute All	**0.067**	34.34	30783	0.0089	0.00924	**0.00914***	100
Superconductivity 60	MR	2.541	**33.24**	30659	**0.0089***	0.01027	**0.00924***	75.00
	AMR	0.15±0.02	71.5±8.2	**4.6 ± 0.6**	0.0095±0.0002	0.0091±0.0000	0.0096±0.0001	**0.01 ± 0.00**
	AC	10.95±0.16	42.98±0.48	115.6±1.98	0.008925±0.000003	**0.008927 ± 0.000001**	0.0093±0.00	0.3005±0.0150
	Impute All	**0.0681**	34.41	30637	**0.0089***	0.0104	**0.00924***	100
Gas 20	MR	0.566	35.05	4160	0.1079	0.1069	0.1098*	65
	AMR	0.55±0.08	**3.82 ± 0.45**	**62.4 ± 7.1**	0.108 ± 0.012*	0.105±0.001	0.109 ± 0.000*	**0.06 ± 0.01**
	AC	**0.26 ± 0.06**	43.69±0.43	108.52±5.79	**0.104351 ± 0.000491***	**0.103460 ± 0.000402**	**0.1071 ± 0.0053***	8.1301±3.4694
	Impute All	0.4472	38.67	4267	0.1056	0.1073	0.1096*	100
Gas 40	MR	0.781	4096	35.8	0.1121	0.1161	0.1101	65.00
	AMR	0.48±0.07	**6.15 ± 0.85**	**31.2 ± 3.5**	0.105±0.001	0.100±0.000	0.108±0.002	**0.03 ± 0.01**
	AC	**0.25 ± 0.04**	40.91±0.15	125.6±3.23	**0.100846 ± 0.000278**	**0.097371 ± 0.000462**	0.1065 ± 0.0061*	6.5041±0.9932
	Impute All	0.447	38.67	5227	0.1047	0.1045	**0.1054***	100
Gas 60	MR	0.566	**35.05**	5098	0.1983	**0.1626***	**0.1166**	65.00
	AMR	0.52±0.06	**3.95 ± 0.52**	**10.4 ± 1.2**	0.235±0.002	0.241±0.001	0.218±0.000	**0.01 ± 0.00**
	AC	**0.30 ± 0.06**	38.86±0.67	101.2±9.78	0.197314 ± 0.035377*	0.222504 ± 0.074158*	0.221±0.0042	7.4194±0.9345
	Impute All	0.447	38.67	5227	**0.185***	0.192	0.2001	100
Concrete 20	MR	0.030	**0.0501**	269	**0.015***	0.0152	0.0155*	50.00
	AMR	0.018 ± 0.002*	0.45±0.06	**15.3 ± 2.1**	0.0158±0.0001	0.0151±0.0000	0.0162±0.0002	**0.15 ± 0.02**
	AC	0.089±0.014	0.140±0.017	23.5±5.16	**0.014555 ± 0.000017**	0.014561±0.000021	0.0160 ± 0.0008*	6.0976±1.2195
	Impute All	**0.0175***	0.0627	273	**0.015**	0.0152	0.0156	100
Concrete 40	MR	0.0383	**0.0501**	532	0.015	**0.0147**	0.0161*	50.00
	AMR	0.030±0.005	0.61±0.08	**6.1 ± 0.8**	0.0155±0.0000	0.0161±0.0001	0.0158 ± 0.0012*	**0.06 ± 0.01**
	AC	0.066±0.002	0.099±0.003	35.4±7.21	**0.014739 ± 0.000019**	0.014755±0.000015	0.0159 ± 0.0004*	5.4711±0.3040
	Impute All	**0.0238**	0.0582	536	**0.015**	0.0162	0.0161*	100
Concrete 60	MR	0.0409	**0.0504**	715	0.0151	**0.0148***	0.0164	50.00
	AMR	0.040±0.008	1.15±0.16	**1.0 ± 0.2**	0.0165±0.0001	0.0158±0.0002	0.0160 ± 0.0000*	**0.01 ± 0.00**
	AC	0.076±0.016	0.115±0.023	46.8±5.47	0.014858±0.000161	0.014871 ± 0.000123*	0.0162 ± 0.0006*	6.6802±0.5356
	Impute All	**0.0319**	0.0561	724	0.0151	0.0153	0.0168	100

Table S18: Accuracy/time for Linear Regression on data with injected MCAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		MSE		Impute % of Samples or Features
		MF	LLMF	MF	LLMF	
Superconductivity 20	MR	851.1	Overtime	0.0091	Overtime	70
	AMR	**14.2 ± 1.8**	Overtime	0.0091±0.0001	Overtime	**0.04 ± 0.01**
	AC	1832.87±1.52	Overtime	**0.008833 ± 0.000001**	Overtime	0.2548±0.0945
	Impute All	794.8	Overtime	0.0091	Overtime	100
Superconductivity 40	MR	842.4	Overtime	0.0092	Overtime	75.00
	AMR	**7.1 ± 0.8**	Overtime	0.0094±0.0000	Overtime	**0.02 ± 0.00**
	AC	1298.18±78.55	Overtime	**0.008842 ± 0.000001**	Overtime	0.2792±0.0509
	Impute All	822.3	Overtime	0.0092	Overtime	100
Superconductivity 60	MR	828.62	Overtime	0.0092	Overtime	75.00
	AMR	**3.6 ± 0.5**	Overtime	0.0090 ± 0.0001*	Overtime	**0.01 ± 0.00**
	AC	819.33±2.89	Overtime	**0.008925 ± 0.000002***	Overtime	0.3005±0.0150
	Impute All	838.62	Overtime	0.0092	Overtime	100
Gas 20	MR	577.8	Overtime	**0.1043***	Overtime	65
	AMR	**22.5 ± 3.2**	Overtime	0.106 ± 0.002*	Overtime	**0.06 ± 0.01**
	AC	761.49±11.36	Overtime	0.104430 ± 0.000529*	Overtime	8.1301±3.4694
	Impute All	600.3	Overtime	0.1063	Overtime	100
Gas 40	MR	591.21	Overtime	0.1066	Overtime	65.00
	AMR	**11.2 ± 1.5**	Overtime	0.112±0.001	Overtime	**0.03 ± 0.01**
	AC	553.48±15.21	Overtime	**0.098578 ± 0.000322**	Overtime	6.5041±0.9932
	Impute All	579.42	Overtime	0.1057	Overtime	100
Gas 60	MR	283.75	Overtime	**0.1536**	Overtime	65.00
	AMR	**3.8 ± 0.5**	Overtime	0.225±0.018	Overtime	**0.01 ± 0.00**
	AC	357.33±4.13	Overtime	0.217546±0.041006	Overtime	7.4194±0.9345
	Impute All	283.29	Overtime	0.198	Overtime	100
Concrete 20	MR	**1.17**	4956	**0.015**	0.0153	50.00
	AMR	1.47±0.18	312	0.0155±0.0001	**0.0146**	**0.15 ± 0.02**
	AC	1.513±0.051	528	0.014556±0.000014	0.0146 ± 0.0000	6.0976±1.2195
	Impute All	1.35	6530	**0.015**	0.0154	100
Concrete 40	MR	1.22	9672	0.0149	0.0160	50.00
	AMR	**0.59 ± 0.08**	612	0.0152±0.0000	**0.0149**	**0.06 ± 0.01**
	AC	1.427±0.026	714	0.014741±0.000020	0.0148 ± 0.0001	5.4711±0.3040
	Impute All	1.489	15954	**0.015**	0.0158	100
Concrete 60	MR	1.208	14628	0.0151	0.0165	50.00
	AMR	**0.10 ± 0.02**	948	0.0164±0.0001	**0.0151**	**0.01 ± 0.00**
	AC	1.414±0.029	1074	**0.014828 ± 0.000154**	0.0149 ± 0.0001	6.6802±0.5356
	Impute All	1.42	16914	**0.015**	0.0161	100

Table S19: Accuracy/time for Linear Regression on data with injected MAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			MSE			Impute % of Samples or Features
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Superconductivity 20	MR	11.76	33.12	6367	0.0090	0.0090*	0.092	55.6
	AMR	18.5 ± 2.4	$14.2 \pm 1.8^*$	$\mathbf{80.5 \pm 11.2}$	0.0102 ± 0.0001	0.0108 ± 0.0001	0.0104 ± 0.0000	$\mathbf{0.250 \pm 0.040}^*$
	AC	27.95 ± 0.472	$\mathbf{12.51 \pm 0.125}^*$	104.7 ± 0.47	0.0106 ± 0.00	0.0107 ± 0.00	0.0103 ± 0.00	$0.323 \pm 0.105^*$
	Impute All	**15.92**	32.52	6674	**0.0089**	**0.0090***	**0.0091**	100
Superconductivity 40	MR	18.14	30.99	12637	0.0090	**0.0090**	**0.0092***	33.3
	AMR	20.5 ± 3.1	$\mathbf{25.4 \pm 4.2}^*$	$\mathbf{68.5 \pm 8.5}$	0.0109 ± 0.0000	0.0112 ± 0.0001	0.0110 ± 0.0002	$\mathbf{0.021 \pm 0.005}^*$
	AC	$\mathbf{14.08 \pm 0.682}$	$29.10 \pm 0.447^*$	105.3 ± 1.63	0.0110 ± 0.00	0.0111 ± 0.00	0.0111 ± 0.00	$0.032 \pm 0.087^*$
	Impute All	27.74	32.44	12765	0.0090	0.0091	**0.0092***	100
Superconductivity 60	MR	28.74	30.93	19117	0.0092	**0.0091**	0.0095	44.4
	AMR	15.5 ± 2.5	$\mathbf{22.1 \pm 3.5}^*$	$\mathbf{6.5 \pm 1.2}$	0.0133 ± 0.0018	0.0138 ± 0.0001	0.0130 ± 0.0000	$\mathbf{0.015 \pm 0.003}$
	AC	$\mathbf{10.49 \pm 0.42}$	$24.77 \pm 0.95^*$	108.7 ± 0.94	0.0135 ± 0.00	0.0136 ± 0.00	0.0132 ± 0.00	0.30 ± 0.04
	Impute All	40.23	34.13	19349	**0.0090**	0.0097	**0.0094**	100
Gas 20	MR	0.316	37.98	928	0.167	0.110	**0.107***	69.2
	AMR	0.35 ± 0.05	19.5 ± 2.8	$\mathbf{73.2 \pm 10.5}$	0.160 ± 0.001	0.161 ± 0.002	0.228 ± 0.000	$\mathbf{2.85 \pm 0.45}^*$
	AC	$\mathbf{0.273 \pm 0.034}^*$	$\mathbf{18.98 \pm 0.370}^*$	106.7 ± 1.88	$0.162 \pm 0.011^*$	0.158 ± 0.00	0.233 ± 0.01	$4.15 \pm 1.24^*$
	Impute All	0.381	39.14	957	**0.154**	**0.1081**	0.117	100
Gas 40	MR	0.444	36.57	1758	**0.122**	0.1015^*	**0.102**	53.8
	AMR	$0.45 \pm 0.08^*$	$\mathbf{15.2 \pm 2.1}$	$\mathbf{48.6 \pm 7.2}$	0.230 ± 0.001	$0.205 \pm 0.000^*$	0.350 ± 0.045	$\mathbf{1.95 \pm 0.35}$
	AC	$\mathbf{0.363 \pm 0.032}^*$	17.54 ± 0.184	113.6 ± 7.36	0.234 ± 0.015	$0.199 \pm 0.026^*$	0.343 ± 0.049	4.55 ± 0.51
	Impute All	0.577	38.22	1785	0.146	**0.1015***	0.102	100
Gas 60	MR	0.604	34.74	2661	**0.112**	**0.1048**	**0.102***	38.5
	AMR	$0.55 \pm 0.09^*$	$16.4 \pm 2.5^*$	$\mathbf{34.4 \pm 5.5}$	0.271 ± 0.002	0.248 ± 0.000	0.315 ± 0.001	$\mathbf{1.25 \pm 0.25}$
	AC	$\mathbf{0.372 \pm 0.066}$	$\mathbf{15.86 \pm 0.51}^*$	123 ± 2.16	0.2735 ± 0.0021	0.2453 ± 0.0252	0.319 ± 0.082	4.468 ± 0.115
	Impute All	0.791	45.89	2703	0.188	0.1186	**0.102***	100
Concrete 20	MR	0.0179	**0.0347**	282	0.0150	0.0151	0.0171	50
	AMR	0.045 ± 0.008	0.085 ± 0.012	$\mathbf{62.5 \pm 8.4}$	0.0151 ± 0.0001	$\mathbf{0.0149 \pm 0.0000}$	0.0205 ± 0.0002	$\mathbf{3.55 \pm 0.55}$
	AC	0.076 ± 0.001	0.092 ± 0.003	103 ± 9.27	0.01505 ± 0.00	0.01504 ± 0.00	0.0210 ± 0.004	5.859 ± 0.571
	Impute All	**0.0172**	0.0396	396	0.0149	0.0150	0.0169	100
Concrete 40	MR	**0.0217**	**0.0344**	571	0.0152	0.0152^*	0.0178^*	50
	AMR	0.038 ± 0.005	0.082 ± 0.015	$\mathbf{35.2 \pm 5.8}$	$\mathbf{0.0158 \pm 0.0000}$	$0.0162 \pm 0.0018^*$	0.0212 ± 0.0001	$\mathbf{2.15 \pm 0.32}$
	AC	0.064 ± 0.002	0.079 ± 0.002	105.7 ± 8.99	0.01597 ± 0.00	0.01603 ± 0.00	$0.0216 \pm 0.005^*$	6.465 ± 1.169
	Impute All	0.0251	0.0379	593	**0.0149**	0.0153	0.0182	100
Concrete 60	MR	**0.0246**	**0.0346**	782	0.0675^*	0.0650	0.0291^*	50
	AMR	0.042 ± 0.007	0.075 ± 0.011	$\mathbf{25.6 \pm 4.2}$	0.0685 ± 0.0005	0.0642 ± 0.0001	0.0385 ± 0.0000	$\mathbf{1.45 \pm 0.28}$
	AC	0.054 ± 0.000	0.069 ± 0.000	107.3 ± 5.25	$0.06919 \pm 0.00180^*$	0.06331 ± 0.00191	$0.0378 \pm 0.0115^*$	6.073 ± 1.03
	Impute All	0.0311	0.0378	801	0.0678	0.0606	0.0374	100

Table S19: Accuracy/time for Linear Regression on data with injected MAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		MSE		Impute % of Samples or Features
		MF	LLMF	MF	LLMF	
Superconductivity 20	MR	1240.19	Overtime	0.0090	Overtime	55.6
	AMR	$\mathbf{370.2 \pm 45.5}$	Overtime	0.0105 ± 0.0002	Overtime	$\mathbf{0.250 \pm 0.040}^*$
	AC	478.19 ± 2.85	Overtime	0.0107 ± 0.00	Overtime	$0.323 \pm 0.105^*$
	Impute All	1334.95	Overtime	**0.0089**	Overtime	100
Superconductivity 40	MR	973.18	Overtime	**0.0089**	Overtime	33.3
	AMR	$\mathbf{230.4 \pm 28.6}$	Overtime	0.0108 ± 0.0000	Overtime	$\mathbf{0.021 \pm 0.005}^*$
	AC	351.3 ± 3.04	Overtime	0.0110 ± 0.00	Overtime	$0.032 \pm 0.087^*$
	Impute All	1110.11	Overtime	0.0094	Overtime	100
Superconductivity 60	MR	1148.74	Overtime	**0.0091**	Overtime	44.4
	AMR	$\mathbf{11.5 \pm 2.1}$	Overtime	0.0134 ± 0.0001	Overtime	$\mathbf{0.015 \pm 0.003}$
	AC	230.2 ± 0.12	Overtime	0.0135 ± 0.00	Overtime	0.30 ± 0.04
	Impute All	1209.42	Overtime	0.0092	Overtime	100
Gas 20	MR	454.23	Overtime	**0.107***	Overtime	69.2
	AMR	$\mathbf{132.5 \pm 18.2}$	Overtime	0.159 ± 0.001	Overtime	$\mathbf{2.85 \pm 0.45}^*$
	AC	192.5 ± 0.913	Overtime	0.157 ± 0.012	Overtime	$4.15 \pm 1.24^*$
	Impute All	480.07	Overtime	**0.107***	Overtime	100
Gas 40	MR	436.37	Overtime	**0.122**	Overtime	53.8
	AMR	$\mathbf{59.8 \pm 8.5}$	Overtime	0.231 ± 0.002	Overtime	$\mathbf{1.95 \pm 0.35}$
	AC	139.5 ± 2.48	Overtime	0.234 ± 0.014	Overtime	4.55 ± 0.51
	Impute All	467.34	Overtime	0.139	Overtime	100
Gas 60	MR	208.13	Overtime	0.197	Overtime	38.5
	AMR	$\mathbf{23.7 \pm 3.8}$	Overtime	0.289 ± 0.001	Overtime	$\mathbf{1.25 \pm 0.25}$
	AC	84.73 ± 0.07	Overtime	0.2861 ± 0.0072	Overtime	4.468 ± 0.115
	Impute All	449.58	Overtime	**0.183**	Overtime	100
Concrete 20	MR	**0.0855**	5022	**0.0148**	0.0231	50
	AMR	0.355 ± 0.052	342	0.0152 ± 0.0001	0.0149	$\mathbf{3.55 \pm 0.55}$
	AC	0.585 ± 0.009	474	$\mathbf{0.01503 \pm 0.00}$	0.0151 ± 0.0001	5.859 ± 0.571
	Impute All	1.018	6840	**0.0147**	0.0216	100
Concrete 40	MR	0.8754	9906	**0.0151**	0.0230	50
	AMR	$\mathbf{0.185 \pm 0.035}$	618	0.0161 ± 0.0002	0.0159	$\mathbf{2.15 \pm 0.32}$
	AC	0.557 ± 0.008	636	0.01594 ± 0.00	$\mathbf{0.0158 \pm 0.0002}$	6.465 ± 1.169
	Impute All	1.039	13272	0.0150	0.0230	100
Concrete 60	MR	0.895	15984	0.0584	**0.0227**	50
	AMR	$\mathbf{0.132 \pm 0.025}$	906	0.0668 ± 0.0002	**0.0294**	$\mathbf{1.45 \pm 0.28}$
	AC	0.542 ± 0.009	1164	0.06776 ± 0.00264	$\mathbf{0.0268 \pm 0.0005}$	6.073 ± 1.03
	Impute All	1.001	19704	0.0655	**0.0232**	100

Table S20: Accuracy/time for Linear Regression on data with injected MNAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			MSE			Impute % of Samples or Features
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Superconductivity 20	MR	**11.76**	30.47	11709	0.0090	**0.0090***	0.0093	30
	AMR	16.2 ± 2.1	$31.4 \pm 4.2^*$	$\mathbf{86.4 \pm 11.5}$	0.0108 ± 0.0001	0.0106 ± 0.0002	0.0107 ± 0.0011	$\mathbf{0.28 \pm 0.04}$
	AC	14.00 ± 0.19	$\mathbf{29.08 \pm 0.45}^*$	114.6 ± 4.98	0.01076 ± 0.00	0.01077 ± 0.00	0.01059 ± 0.00	0.372 ± 0.02
	Impute All	23.64	33.02	11984	**0.0089**	**0.0090***	**0.0090**	100
Superconductivity 40	MR	27.69	31.18	21909	**0.0090***	**0.0090**	**0.0091***	40
	AMR	12.4 ± 1.8	$27.6 \pm 3.5^*$	$\mathbf{94.8 \pm 12.6}$	0.0104 ± 0.0000	0.0109 ± 0.0016	0.0106 ± 0.0001	$\mathbf{0.21 \pm 0.03}$
	AC	$\mathbf{9.86 \pm 0.32}$	$\mathbf{24.45 \pm 0.76}^*$	135.7 ± 0.47	0.0106 ± 0.00	0.0108 ± 0.00	0.0108 ± 0.00	0.299 ± 0.02
	Impute All	36.65	31.36	22285	**0.0090***	0.0091	**0.0091***	100
Superconductivity 60	MR	36.69	31.33	28009	0.009125	0.009201	**0.0091***	40
	AMR	8.9 ± 1.2	14.2 ± 2.4	$\mathbf{61.2 \pm 8.7}$	0.0134 ± 0.0013	0.0135 ± 0.0001	0.0142 ± 0.0000	$\mathbf{0.14 \pm 0.02}$
	AC	$\mathbf{4.03 \pm 1.05}$	$\mathbf{5.61 \pm 1.06}$	133.3 ± 0.47	0.0132 ± 0.00	0.0138 ± 0.00	0.0145 ± 0.00	0.312 ± 0.01
	Impute All	47.47	32.79	28345	**0.0090**	**0.0090**	**0.0091***	100
Gas 20	MR	0.316^*	37.98	922	0.167	0.110	**0.106**	64.28
	AMR	$0.31 \pm 0.04^*$	22.5 ± 3.1	$\mathbf{88.4 \pm 14.2}$	0.165 ± 0.001	0.164 ± 0.000	0.230 ± 0.002	$\mathbf{6.12 \pm 0.85}$
	AC	$\mathbf{0.259 \pm 0.065}^*$	$\mathbf{18.25 \pm 0.52}$	125.3 ± 3.68	0.169 ± 0.002	0.161 ± 0.003	0.233 ± 0.0078	8.79 ± 1.33
	Impute All	0.381	39.14	958	**0.154**	**0.1081**	0.110	100
Gas 40	MR	0.444	36.57	1741	**0.122**	**0.1015***	**0.102**	50
	AMR	0.42 ± 0.06	19.8 ± 2.6	$\mathbf{168.2 \pm 25.4}$	0.262 ± 0.039	0.225 ± 0.001	0.288 ± 0.001	$\mathbf{4.53 \pm 0.62}$
	AC	$\mathbf{0.274 \pm 0.004}$	$\mathbf{15.67 \pm 0.78}$	251 ± 5.65	0.258 ± 0.008	0.229 ± 0.015	0.293 ± 0.104	6.79 ± 0.22
	Impute All	0.577	38.22	1785	0.146	**0.1015***	0.105	100
Gas 60	MR	0.604	34.74	3656	**0.112**	**0.1048**	**0.102**	42.8
	AMR	0.38 ± 0.05	16.4 ± 2.2	$\mathbf{104.5 \pm 16.8}$	0.228 ± 0.001	0.212 ± 0.000	0.221 ± 0.033	$\mathbf{3.17 \pm 0.45}$
	AC	$\mathbf{0.171 \pm 0.059}$	$\mathbf{12.75 \pm 0.46}$	254 ± 0.47	0.232 ± 0.059	0.209 ± 0.005	0.226 ± 0.046	7.76 ± 1.91
	Impute All	0.791	45.89	3703	0.188	0.1186	0.105	100
Concrete 20	MR	0.0179	**0.0347**	443	0.0150	0.0151	0.0192	50
	AMR	0.041 ± 0.006	0.062 ± 0.009	$\mathbf{34.1 \pm 5.2}$	$\mathbf{0.0152 \pm 0.0001}$	0.0157 ± 0.0016	0.0228 ± 0.0001	$\mathbf{3.42 \pm 0.51}$
	AC	0.069 ± 0.002	0.084 ± 0.002	58 ± 1.07	0.0155 ± 0.00	$0.0155 \pm 0.00^*$	$0.0233 \pm 0.006^*$	5.87 ± 0.57
	Impute All	**0.0172**	0.0396	461	0.0149	0.0150^*	0.0181^*	100
Concrete 40	MR	**0.0217**	**0.0344**	791	0.0152	0.0152^*	0.0225	50
	AMR	0.034 ± 0.005	0.051 ± 0.008	$\mathbf{25.2 \pm 4.1}$	0.0154 ± 0.0000	$\mathbf{0.0151 \pm 0.0001}^*$	0.0231 ± 0.0002	$\mathbf{2.18 \pm 0.35}$
	AC	0.0493 ± 0.00	0.0645 ± 0.00	71 ± 1.24	0.0152 ± 0.00	0.0155 ± 0.00	$0.0235 \pm 0.006^*$	6.33 ± 0.54
	Impute All	0.0251	0.0379	806	**0.0149**	0.0153	0.0213	100
Concrete 60	MR	**0.0246**	**0.0346**	1039	0.0675	0.0650	0.027	50
	AMR	0.0315 ± 0.0042	0.041 ± 0.006	$\mathbf{14.8 \pm 2.3}$	0.081 ± 0.012	0.085 ± 0.002	0.0385 ± 0.0000	$\mathbf{1.14 \pm 0.18}$
	AC	0.032 ± 0.001	0.045 ± 0.002	79 ± 2.19	0.084 ± 0.003	0.083 ± 0.004	0.0396 ± 0.01	6.20 ± 0.118
	Impute All	0.0311	0.0378	1051	0.0678	0.0606	0.0271	100

Table S20: Accuracy/time for Linear Regression on data with injected MNAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		MSE		Impute % of Samples or Features
		MF	LLMF	MF	LLMF	
Superconductivity 20	MR	966.7	Overtime	0.0090	Overtime	30
	AMR	$\mathbf{298.1 \pm 38.4}$	Overtime	0.0109 ± 0.0001	Overtime	$\mathbf{0.28 \pm 0.04}$
	AC	394.01 ± 2.64	Overtime	0.01078 ± 0.00	Overtime	0.372 ± 0.02
	Impute All	1154	Overtime	**0.0089**	Overtime	100
Superconductivity 40	MR	975.0	Overtime	**0.0089**	Overtime	40
	AMR	151.3 ± 21.2	Overtime	0.0108 ± 0.0000	Overtime	$\mathbf{0.21 \pm 0.03}$
	AC	217.66 ± 0.56	Overtime	0.0107 ± 0.00	Overtime	0.299 ± 0.02
	Impute All	**10.97**	Overtime	0.0094	Overtime	100
Superconductivity 60	MR	952.42	Overtime	**0.0091**	Overtime	40
	AMR	$\mathbf{43.7 \pm 6.1}$	Overtime	0.0133 ± 0.0001	Overtime	$\mathbf{0.14 \pm 0.02}$
	AC	94.77 ± 1.21	Overtime	0.0131 ± 0.00	Overtime	0.312 ± 0.01
	Impute All	1062.2	Overtime	0.0092	Overtime	100
Gas 20	MR	454.23	Overtime	$\mathbf{0.107}^*$	Overtime	64.28
	AMR	$\mathbf{103.7 \pm 18.5}$	Overtime	0.172 ± 0.017	Overtime	$\mathbf{6.12 \pm 0.85}$
	AC	150.1 ± 0.42	Overtime	0.170 ± 0.00	Overtime	8.79 ± 1.33
	Impute All	480.07	Overtime	$\mathbf{0.107}^*$	Overtime	100
Gas 40	MR	436.37	Overtime	**0.122**	Overtime	50
	AMR	$\mathbf{51.4 \pm 8.2}$	Overtime	0.271 ± 0.000	Overtime	$\mathbf{4.53 \pm 0.62}$
	AC	76.30 ± 0.25	Overtime	0.267 ± 0.004	Overtime	6.79 ± 0.22
	Impute All	467.34	Overtime	0.139	Overtime	100
Gas 60	MR	208.13	Overtime	0.197	Overtime	42.8
	AMR	$\mathbf{12.1 \pm 2.1}$	Overtime	0.241 ± 0.001	Overtime	$\mathbf{3.17 \pm 0.45}$
	AC	28.89 ± 0.07	Overtime	0.238 ± 0.02	Overtime	7.76 ± 1.91
	Impute All	449.58	Overtime	**0.183**	Overtime	100
Concrete 20	MR	**0.0855**	10266	**0.0148**	0.0228	50
	AMR	0.34 ± 0.05	588	0.0158 ± 0.0002	0.0155	$\mathbf{3.42 \pm 0.51}$
	AC	0.573 ± 0.004	774	0.0155 ± 0.00	$\mathbf{0.0155 \pm 0.0000}$	5.87 ± 0.57
	Impute All	1.018	11652	**0.0147**	0.0226	100
Concrete 40	MR	0.8754	17664	**0.0151**	0.0228	50
	AMR	$\mathbf{0.187 \pm 0.028}$	1044	$0.0153 \pm 0.0018^*$	0.0231	$\mathbf{2.18 \pm 0.35}$
	AC	0.551 ± 0.006	1224	$\mathbf{0.0151 \pm 0.00}$	0.0152 ± 0.0004	6.33 ± 0.54
	Impute All	1.039	21204	0.0150^*	0.0232	100
Concrete 60	MR	0.895	24516	0.0584	**0.0237**	50
	AMR	$\mathbf{0.098 \pm 0.015}$	1392	0.082 ± 0.001	**0.0277**	$\mathbf{1.14 \pm 0.18}$
	AC	0.526 ± 0.012	1512	0.086 ± 0.002	$\mathbf{0.0270 \pm 0.0023}$	6.20 ± 0.118
	Impute All	1.001	29010	0.0655	**0.0235**	100

Table S21: Accuracy/time for MLP on data with injected MCAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s) AMR	Accuracy(%) AMR	Impute % of Samples AMR
Tuadromd	20	98.73 ± 0.10	1.063 ± 0.04	98.787 ± 0.099	$\mathbf{9.34 \pm 4.04}$
	40	98.73 ± 0.10	1.306 ± 0.03	98.740 ± 0.020	$\mathbf{5.84 \pm 2.02}$
	60	98.73 ± 0.10	1.597 ± 0.01	98.250 ± 0.041	$\mathbf{4.67 \pm 0.0}$
Malware	20	95.74 ± 0.88	3.785 ± 0.07	96.293 ± 0.246	$\mathbf{15.67 \pm 0.0}$
	40	95.74 ± 0.88	3.928 ± 0.07	96.000 ± 0.299	$\mathbf{7.84 \pm 0.0}$
	60	95.74 ± 0.88	3.911 ± 0.005	93.745 ± 2.098	$\mathbf{5.23 \pm 0.0}$
Heart	20	84.75 ± 0.03	1.106 ± 0.026	83.62 ± 0.98	$\mathbf{97.87 \pm 0.00}$
	40	84.75 ± 0.03	0.911 ± 0.041	86.4 ± 1.08	$\mathbf{52.6 \pm 0.0}$
	60	84.75 ± 0.03	0.767 ± 0.028	82.49 ± 5.45	$\mathbf{35.21 \pm 0.00}$
Parkinsons	20	92.11 ± 0.00	0.771 ± 0.013	93.0 ± 1.5	$\mathbf{57.0 \pm 12.2}$
	40	92.11 ± 0.00	0.689 ± 0.012	93.9 ± 1.5	$\mathbf{33.3 \pm 6.7}$
	60	92.11 ± 0.00	0.551 ± 0.017	92.1 ± 0.0	$\mathbf{57.4 \pm 0.0}$
Fraud	20	99.96	508 ± 4.5	99.89 ± 0.00	$\mathbf{0.015 \pm 0.00}$
	40	99.96	499 ± 2.8	99.88 ± 0.01	$\mathbf{0.007 \pm 0.00}$
	60	99.96	508 ± 5.0	99.88 ± 0.01	$\mathbf{0.005 \pm 0.00}$
HIGGS	20	72.64	5587 ± 75.60	69.64 ± 2.03	$\mathbf{20.31 \pm 3.21}$
	40	72.64	5663 ± 63.9	72.05 ± 1.41	$\mathbf{30.89 \pm 4.11}$
	60	72.64	5266 ± 100.4	69.64 ± 1.10	$\mathbf{35.78 \pm 1.46}$

Table S22: Accuracy/time for MLP on data with injected MAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s) AMR	Accuracy(%) AMR	Impute % of Samples AMR
Tuadromd	20	98.84 ± 0.08	0.790 ± 0.04	98.610 ± 0.054	$\mathbf{7.0 \pm 0.1}$
	40	98.84 ± 0.08	0.779 ± 0.02	98.596 ± 0.102	$\mathbf{3.50 \pm 0.2}$
	60	98.84 ± 0.08	0.787 ± 0.01	98.414 ± 0.100	$\mathbf{2.33 \pm 0.5}$
Malware	20	96.37 ± 0.50	5.108 ± 2.28	96.199 ± 0.616	$\mathbf{20.9 \pm 9.1}$
	40	96.37 ± 0.50	6.587 ± 4.470	95.980 ± 0.642	$\mathbf{13.06 \pm 9.1}$
	60	96.37 ± 0.50	5.137 ± 2.263	96.053 ± 0.335	$\mathbf{6.96 \pm 3.01}$
Heart	20	84.75 ± 0.03	1.039 ± 0.026	84.2 ± 1.0	$\mathbf{27.8 \pm 7.8}$
	40	84.75 ± 0.03	0.885 ± 0.014	85.9 ± 1.0	$\mathbf{74.9 \pm 3.2}$
	60	84.75 ± 0.03	0.763 ± 0.025	86.5 ± 2.4	$\mathbf{61.2 \pm 1.1}$
Parkinsons	20	92.11 ± 0.00	0.792 ± 0.036	93.9 ± 1.5	$\mathbf{74.2 \pm 5.5}$
	40	92.11 ± 0.00	0.688 ± 0.022	93.9 ± 1.5	$\mathbf{48.0 \pm 9.8}$
	60	92.11 ± 0.00	0.468 ± 0.018	94.7 ± 0.0	$\mathbf{11.3 \pm 0.6}$
Fraud	20	99.96	315 ± 2.5	99.87 ± 0.00	$\mathbf{0.015 \pm 0.00}$
	40	99.96	314 ± 1.1	99.89 ± 0.00	$\mathbf{0.007 \pm 0.00}$
	60	99.96	314 ± 1.8	99.88 ± 0.01	$\mathbf{0.005 \pm 0.00}$
HIGGS	20	72.64	5809 ± 50	70.64 ± 0.45	$\mathbf{11.36 \pm 1.78}$
	40	72.64	6039 ± 59	70.98 ± 1.21	$\mathbf{8.52 \pm 2.34}$
	60	72.64	5847 ± 38	67.85 ± 2.68	$\mathbf{18.94 \pm 1.10}$

Table S23: Accuracy/time for MLP on data with injected MNAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s) AMR	Accuracy(%) AMR	Impute % of Samples AMR
Tuadromd	20	98.75 ± 0.07	0.781 ± 0.05	98.498 ± 0.029	$\mathbf{3.90 \pm 0.0}$
	40	98.75 ± 0.07	0.793 ± 0.01	98.218 ± 0.063	$\mathbf{2.18 \pm 0.0}$
	60	98.75 ± 0.07	0.813 ± 0.01	97.616 ± 0.182	$\mathbf{1.67 \pm 0.0}$
Malware	20	95.93 ± 0.78	9.015 ± 2.145	95.290 ± 0.576	$\mathbf{20 \pm 4.96}$
	40	95.93 ± 0.78	27.17 ± 6.9	95.353 ± 0.602	$\mathbf{34.08 \pm 8.43}$
	60	95.93 ± 0.78	7.935 ± 0.028	95.363 ± 0.517	$\mathbf{7.42 \pm 0.0}$
Heart	20	84.75 ± 0.03	1.114 ± 0.015	83.6 ± 2.0	$\mathbf{4.3 \pm 0.0}$
	40	84.75 ± 0.03	0.897 ± 0.007	84.8 ± 1.7	$\mathbf{52.1 \pm 0.0}$
	60	84.75 ± 0.03	0.683 ± 0.013	84.8 ± 1.7	$\mathbf{69.4 \pm 0.0}$
Parkinsons	20	92.11 ± 0.00	0.728 ± 0.036	93.9 ± 1.5	$\mathbf{63.3 \pm 15.3}$
	40	92.11 ± 0.00	0.701 ± 0.032	90.4 ± 5.5	$\mathbf{37.6 \pm 9.7}$
	60	92.11 ± 0.00	0.445 ± 0.040	86.8 ± 3.7	$\mathbf{37.75 \pm 5.3}$
Fraud	20	99.96	437 ± 4.1	99.88 ± 0.00	$\mathbf{0.015 \pm 0.00}$
	40	99.96	436 ± 1.2	99.88 ± 0.01	$\mathbf{0.007 \pm 0.00}$
	60	99.96	1103 ± 37	99.87 ± 0.00	$\mathbf{0.24 \pm 0.00}$
HIGGS	20	72.64	5253.590 ± 78.320	70.220 ± 0.340	$\mathbf{39.600 \pm 4.140}$
	40	72.64	5168.105 ± 90.670	69.765 ± 0.870	$\mathbf{45.450 \pm 0.000}$
	60	72.64	5156.492 ± 74.340	67.630 ± 1.910	$\mathbf{45.450 \pm 0.000}$

Table S24: Accuracy/time for MLP on data with injected MCAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Malware 20	AMR	$\mathbf{3.225 \pm 0.120}$	N/A	$\mathbf{20875.26 \pm 0.00}$	96.1 ± 0.1	N/A	90.01 ± 0.11	$\mathbf{15.67 \pm 0.00}$
	Impute All	13.9	N/A	Overtime	**96.52**	N/A	Overtime	100
Malware 40	AMR	$\mathbf{2.755 \pm 0.183}$	N/A	$\mathbf{10444.29 \pm 0.00}$	$\mathbf{94.8 \pm 1.7}^{*}$	N/A	95.79 ± 0.13	$\mathbf{7.84 \pm 0.00}$
	Impute All	27.66	N/A	Overtime	93.83^{*}	N/A	Overtime	100
Malware 60	AMR	$\mathbf{1.773 \pm 0.122}$	N/A	$\mathbf{6967.30 \pm 0.00}$	93.9 ± 2.1	N/A	95.87 ± 0.58	$\mathbf{5.23 \pm 0.00}$
	Impute All	41.21	N/A	Overtime	**96.16**	N/A	Overtime	100
Tuadromd 20	AMR	$\mathbf{0.404 \pm 0.081}$	$\mathbf{99.670 \pm 2.957}^{*}$	$\mathbf{0.425 \pm 0.082}$	$\mathbf{98.9 \pm 0.0}$	$\mathbf{98.88 \pm 0.06}$	97.00 ± 0.12	$\mathbf{9.34 \pm 4.04}$
	Impute All	2.37	102.11^{*}	1987	98.77	98.77	**98.66**	100
Tuadromd 40	AMR	$\mathbf{0.306 \pm 0.023}$	$\mathbf{93.067 \pm 1.545}$	$\mathbf{0.467 \pm 0.013}$	$\mathbf{98.9 \pm 0.1}$	$\mathbf{98.79 \pm 0.10}$	96.34 ± 0.56	$\mathbf{5.84 \pm 2.02}$
	Impute All	4.69	100.14	3882	97.3	97.6	**98.66**	100
Tuadromd 60	AMR	$\mathbf{0.324 \pm 0.027}$	$\mathbf{75.595 \pm 0.209}$	$\mathbf{0.034 \pm 0.002}$	$\mathbf{98.9 \pm 0.0}$	$\mathbf{98.69 \pm 0.07}$	98.32 ± 0.02	$\mathbf{4.67 \pm 0.00}$
	Impute All	6.21	100.6	6476	97.6	97.3	**98.66**	100
Parkinsons 20	AMR	22.652 ± 1.214	22.805 ± 1.211	21.886 ± 1.189	85.96 ± 1.52	86.84 ± 0.00	92.98 ± 4.02	68.82 ± 3.72
	Impute All	1.493	1.682	0.741	86.84	84.21	94.74	100
Parkinsons 40	AMR	45.675 ± 6.437	45.878 ± 6.479	44.978 ± 6.492	86.84 ± 0.00	86.84 ± 0.00	83.33 ± 1.52	69.35 ± 9.81
	Impute All	1.520	1.777	0.714	86.84	86.84	92.11	100
Parkinsons 60	AMR	53.000 ± 0.367	53.260 ± 0.411	52.480 ± 0.396	71.93 ± 3.04	77.20 ± 1.52	90.35 ± 1.52	53.19 ± 0.00
	Impute All	1.471	1.904	0.743	71.05	78.95	84.21	100
Heart 20	AMR	72.582 ± 0.105	72.717 ± 0.134	71.589 ± 0.256	86.44 ± 0.00	86.44 ± 0.00	83.62 ± 0.98	97.87 ± 0.00
	Impute All	2.150	2.166	1.462	86.44	86.44	83.05	100
Heart 40	AMR	78.612 ± 0.145	78.753 ± 0.093	77.694 ± 0.147	86.44 ± 0.00	87.01 ± 2.59	85.88 ± 0.98	52.63 ± 0.00
	Impute All	2.178	2.383	1.047	84.75	86.44	88.14	100
Heart 60	AMR	78.530 ± 0.233	78.669 ± 0.248	77.756 ± 0.229	83.05 ± 1.70	83.05 ± 0.00	78.53 ± 0.98	35.21 ± 0.00
	Impute All	2.196	2.383	1.052	77.97	86.44	84.75	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime

Table S24: Accuracy/time for MLP on data with injected MCAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Malware 20	AMR	$\mathbf{2479.86 \pm 83.18}$	Overtime	$95.59 \pm 1.58^{*}$	Overtime	$\mathbf{15.67 \pm 0.00}$
	Impute All	6645.5	Overtime	$\mathbf{96.70}^{*}$	Overtime	100
Malware 40	AMR	7075.08 ± 274.29	Overtime	$95.44 \pm 1.02^{*}$	Overtime	$\mathbf{7.84 \pm 0.00}$
	Impute All	**6347.8**	Overtime	$\mathbf{95.75}^{*}$	Overtime	100
Malware 60	AMR	$\mathbf{4344.76 \pm 216.72}$	Overtime	94.32 ± 0.26	Overtime	$\mathbf{5.23 \pm 0.00}$
	Impute All	6256.4	Overtime	**96.70**	Overtime	100
Tuadromd 20	AMR	$\mathbf{45.502 \pm 0.199}^{*}$	Overtime	$\mathbf{98.74 \pm 0.06}^{*}$	Overtime	$\mathbf{9.34 \pm 4.04}$
	Impute All	45.60^{*}	Overtime	98.70^{*}	Overtime	100
Tuadromd 40	AMR	$\mathbf{41.645 \pm 0.846}$	Overtime	$\mathbf{98.90 \pm 0.04}$	Overtime	$\mathbf{5.84 \pm 2.02}$
	Impute All	43.94	Overtime	98.84	Overtime	100
Tuadromd 60	AMR	$\mathbf{37.910 \pm 0.194}$	Overtime	$\mathbf{98.81 \pm 0.03}$	Overtime	$\mathbf{4.67 \pm 0.00}$
	Impute All	43.96	Overtime	98.66	Overtime	100
Parkinsons 20	AMR	25.253 ± 1.233	1043.00 ± 41.39	85.09 ± 3.04	94.74 ± 2.63	68.82 ± 3.72
	Impute All	88.052	1796.9	86.84	94.74	100
Parkinsons 40	AMR	48.311 ± 6.453	2195.00 ± 325.94	86.84 ± 0.00	91.23 ± 3.04	69.35 ± 9.81
	Impute All	91.186	3664.7	86.84	92.11	100
Parkinsons 60	AMR	55.643 ± 0.377	2466.00 ± 39.00	77.20 ± 1.52	91.23 ± 1.52	53.19 ± 0.00
	Impute All	173.210	5874.9	76.32	84.21	100
Heart 20	AMR	74.461 ± 0.101	2077.00 ± 6.24	83.05 ± 2.94	80.79 ± 1.96	97.87 ± 0.00
	Impute All	85.135	1800.4	83.05	81.36	100
Heart 40	AMR	80.293 ± 0.188	2276.00 ± 13.53	85.31 ± 1.96	82.49 ± 0.98	52.63 ± 0.00
	Impute All	162.919	3680.3	86.44	88.14	100
Heart 60	AMR	80.333 ± 0.260	2384.00 ± 59.27	83.05 ± 1.70	80.23 ± 2.59	35.21 ± 0.00
	Impute All	243.998	5539.2	81.36	86.44	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime

Table S25: Accuracy/time for MLP on data with injected MAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Malware 20	AMR	$\mathbf{4.962 \pm 1.496}$	N/A	$\mathbf{27842.56 \pm 12122.84}$	$\mathbf{96.4 \pm 0.2}^{*}$	N/A	$\mathbf{96.38 \pm 0.23}$	$\mathbf{20.90 \pm 9.10}$
	Impute All	17.71	N/A	Overtime	96.30^{*}	N/A	Overtime	100
Malware 40	AMR	$\mathbf{4.309 \pm 2.176}$	N/A	$\mathbf{17398.27 \pm 12122.84}$	$\mathbf{96.4 \pm 0.2}$	N/A	$\mathbf{94.12 \pm 2.85}$	$\mathbf{13.06 \pm 9.10}$
	Impute All	36.10	N/A	Overtime	96.15	N/A	Overtime	100
Malware 60	AMR	$\mathbf{2.787 \pm 0.911}$	N/A	$\mathbf{9271.97 \pm 4009.86}$	$\mathbf{96.2 \pm 0.4}^{*}$	N/A	$\mathbf{96.02 \pm 0.20}$	$\mathbf{6.96 \pm 3.01}$
	Impute All	48.33	N/A	Overtime	95.85^{*}	N/A	Overtime	100
Tuadromd 20	AMR	$\mathbf{0.187 \pm 0.002}$	$103.277 \pm 16.587^{*}$	$\mathbf{0.069 \pm 0.002}$	98.6 ± 0.0	98.51 ± 0.06	98.70 ± 0.04	$\mathbf{7.00 \pm 0.10}$
	Impute All	1.72	101.1^{*}	1673.4	**98.80**	**98.73**	**98.81**	100
Tuadromd 40	AMR	$\mathbf{0.141 \pm 0.002}$	$\mathbf{91.316 \pm 1.460}$	$\mathbf{0.055 \pm 0.005}$	$\mathbf{98.6 \pm 0.0}$	$\mathbf{98.50 \pm 0.14}$	98.45 ± 0.10	$\mathbf{3.50 \pm 0.20}$
	Impute All	3.17	101.5	3451	97.3	97.6	**98.80**	100
Tuadromd 60	AMR	$\mathbf{0.207 \pm 0.021}$	$\mathbf{73.146 \pm 0.396}$	$\mathbf{0.037 \pm 0.002}$	$\mathbf{98.4 \pm 0.1}$	$\mathbf{98.28 \pm 0.21}$	98.07 ± 0.12	$\mathbf{2.33 \pm 0.50}$
	Impute All	4.39	116.08	5036	97.6	97.3	**98.82**	100
Parkinsons 20	AMR	28.026 ± 1.603	28.253 ± 1.614	27.186 ± 1.588	88.59 ± 1.52	86.84 ± 0.00	93.86 ± 1.52	84.95 ± 4.93
	Impute All	1.562	1.831	0.831	89.47	89.47	89.47	100
Parkinsons 40	AMR	53.226 ± 0.072	53.490 ± 0.037	52.543 ± 0.109	79.83 ± 4.02	72.80 ± 1.52	80.70 ± 4.02	75.76 ± 0.00
	Impute All	1.540	1.792	0.735	84.21	68.42	81.58	100
Parkinsons 60	AMR	21.879 ± 1.830	21.927 ± 1.844	21.355 ± 1.887	77.20 ± 1.52	85.09 ± 3.04	68.42 ± 5.26	22.83 ± 1.88
	Impute All	1.561	1.595	0.743	76.32	86.84	71.05	100
Heart 20	AMR	38.831 ± 4.995	37.090 ± 2.313	36.035 ± 2.366	87.01 ± 0.98	85.88 ± 0.98	83.05 ± 1.70	53.47 ± 3.18
	Impute All	5.455	2.117	1.067	86.44	86.44	83.05	100
Heart 40	AMR	70.074 ± 3.263	70.173 ± 3.297	69.194 ± 3.258	83.62 ± 2.59	86.44 ± 0.00	77.40 ± 3.53	46.05 ± 2.15
	Impute All	2.282	2.349	1.043	86.44	91.53	83.05	100
Heart 60	AMR	78.747 ± 0.191	78.851 ± 0.211	77.980 ± 0.162	49.15 ± 2.93	47.46 ± 0.00	47.46 ± 0.00	35.97 ± 0.00
	Impute All	2.125	2.179	1.023	54.24	47.46	47.46	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime

Table S25: Accuracy/time for MLP on data with injected MAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Malware 20	AMR	$\mathbf{5408.02 \pm 369.08}$	Overtime	$96.23 \pm 0.36^{*}$	Overtime	$\mathbf{20.90 \pm 9.10}$
	Impute All	6830.2	Overtime	$\mathbf{96.59}^{*}$	Overtime	100
Malware 40	AMR	$\mathbf{6361.24 \pm 276.94}^{*}$	Overtime	$\mathbf{96.13 \pm 0.68}^{*}$	Overtime	$\mathbf{13.06 \pm 9.10}$
	Impute All	6460.7^{*}	Overtime	95.90^{*}	Overtime	100
Malware 60	AMR	$\mathbf{3932.27 \pm 214.79}$	Overtime	94.86 ± 1.44	Overtime	$\mathbf{6.96 \pm 3.01}$
	Impute All	5794.3	Overtime	**96.18**	Overtime	100
Tuadromd 20	AMR	$\mathbf{42.845 \pm 0.728}$	Overtime	98.62 ± 0.07	Overtime	$\mathbf{7.00 \pm 0.10}$
	Impute All	44.55	Overtime	**98.81**	Overtime	100
Tuadromd 40	AMR	$\mathbf{38.095 \pm 0.229}$	Overtime	98.55 ± 0.20	Overtime	$\mathbf{3.50 \pm 0.20}$
	Impute All	44.57	Overtime	**98.70**	Overtime	100
Tuadromd 60	AMR	$\mathbf{34.520 \pm 0.383}$	Overtime	98.39 ± 0.13	Overtime	$\mathbf{2.33 \pm 0.50}$
	Impute All	44.17	Overtime	**98.73**	Overtime	100
Parkinsons 20	AMR	31.044 ± 1.548	1343.00 ± 157.87	87.72 ± 1.52	90.35 ± 1.52	84.95 ± 4.93
	Impute All	4.624	1778.8	86.84	92.11	100
Parkinsons 40	AMR	56.232 ± 0.118	2640.00 ± 83.19	88.59 ± 1.52	87.72 ± 1.52	75.76 ± 0.00
	Impute All	4.451	4058.1	86.84	89.47	100
Parkinsons 60	AMR	24.211 ± 1.829	879.00 ± 65.80	85.09 ± 1.52	23.68 ± 0.00	22.83 ± 1.88
	Impute All	3.866	3339.0	84.21	$--$	100
Heart 20	AMR	38.515 ± 2.311	1071.00 ± 85.43	86.44 ± 0.00	80.23 ± 3.53	53.47 ± 3.18
	Impute All	83.621	1668.6	86.44	77.97	100
Heart 40	AMR	71.489 ± 3.248	1844.00 ± 71.27	89.27 ± 0.98	72.32 ± 2.59	46.05 ± 2.15
	Impute All	161.834	3316.9	86.44	69.49	100
Heart 60	AMR	80.407 ± 0.137	2297.00 ± 47.66	43.50 ± 0.98	47.46 ± 6.11	35.97 ± 0.00
	Impute All	239.758	5335.0	37.29	52.54	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime

Table S26: Accuracy/time for MLP on data with injected MNAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Malware 20	AMR	$\mathbf{6.022 \pm 0.930}$	N/A	$\mathbf{26643.60 \pm 6607.61}$	95.4 ± 0.6	N/A	$\mathbf{93.88 \pm 2.78}$	$\mathbf{20.00 \pm 4.96}$
	Impute All	28.96	N/A	Overtime	**96.55**	N/A	Overtime	100
Malware 40	AMR	$\mathbf{16.311 \pm 4.509}$	N/A	$\mathbf{45400.69 \pm 11230.28}$	95.8 ± 0.2	N/A	$\mathbf{90.07 \pm 0.03}$	$\mathbf{34.08 \pm 8.43}$
	Impute All	50.34	N/A	Overtime	**96.52**	N/A	Overtime	100
Malware 60	AMR	$\mathbf{2.168 \pm 0.274}$	N/A	$\mathbf{9884.78 \pm 0.00}$	94.3 ± 1.1	N/A	$\mathbf{94.74 \pm 0.67}$	$\mathbf{7.42 \pm 0.00}$
	Impute All	63.30	N/A	Overtime	**96.18**	N/A	Overtime	100
Tuadromd 20	AMR	$\mathbf{0.255 \pm 0.088}$	$\mathbf{90.768 \pm 1.724}$	$\mathbf{0.059 \pm 0.001}$	98.5 ± 0.0	98.46 ± 0.03	98.54 ± 0.02	$\mathbf{3.90 \pm 0.00}$
	Impute All	2.55	100.5	3402.6	**98.80**	**98.74**	**98.82**	100
Tuadromd 40	AMR	$\mathbf{0.139 \pm 0.054}$	$\mathbf{80.322 \pm 14.707}$	$\mathbf{0.029 \pm 0.003}$	98.2 ± 0.0	98.12 ± 0.12	97.90 ± 0.04	$\mathbf{2.18 \pm 0.00}$
	Impute All	4.39	103.71	6098.7	**98.70**	**98.87**	**98.87**	100
Tuadromd 60	AMR	$\mathbf{0.140 \pm 0.026}$	$\mathbf{63.830 \pm 0.344}$	$\mathbf{0.015 \pm 0.001}$	97.6 ± 0.0	97.72 ± 0.08	97.08 ± 0.03	$\mathbf{1.67 \pm 0.00}$
	Impute All	5.73	101.39	6090.7	**98.84**	**98.59**	**98.82**	100
Parkinsons 20	AMR	7.602 ± 0.574	7.896 ± 0.593	6.911 ± 0.612	90.35 ± 1.52	92.11 ± 2.64	93.86 ± 1.52	24.44 ± 1.92
	Impute All	1.599	1.888	0.738	86.84	92.11	92.11	100
Parkinsons 40	AMR	52.919 ± 0.120	53.183 ± 0.061	52.348 ± 0.105	87.72 ± 4.02	84.21 ± 2.63	78.07 ± 3.04	79.37 ± 0.00
	Impute All	1.597	1.881	0.727	81.58	76.32	81.58	100
Parkinsons 60	AMR	26.853 ± 8.280	27.116 ± 8.295	26.522 ± 8.210	93.86 ± 4.02	95.62 ± 1.52	78.95 ± 2.63	27.30 ± 7.98
	Impute All	1.576	1.777	0.745	89.47	89.47	78.95	100
Heart 20	AMR	72.458 ± 0.066	72.555 ± 0.081	71.351 ± 0.071	84.75 ± 1.70	83.62 ± 0.98	85.88 ± 0.98	97.87 ± 0.00
	Impute All	2.276	2.296	1.044	84.75	83.05	86.44	100
Heart 40	AMR	80.591 ± 1.190	80.663 ± 1.242	79.670 ± 1.232	79.66 ± 1.70	82.49 ± 0.98	81.36 ± 0.00	52.08 ± 0.00
	Impute All	2.227	2.320	1.101	79.66	83.05	83.05	100
Heart 60	AMR	79.673 ± 1.239	79.738 ± 1.272	78.965 ± 1.277	61.58 ± 5.18	53.11 ± 0.98	77.97 ± 2.93	34.72 ± 0.00
	Impute All	2.207	2.288	1.045	50.85	50.85	79.66	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime

Table S26: Accuracy/time for MLP on data with injected MNAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Malware 20	AMR	6578.75 ± 135.13	Overtime	95.57 ± 0.22	Overtime	$\mathbf{20.00 \pm 4.96}$
	Impute All	$\mathbf{5959.7}$	Overtime	$\mathbf{96.52}$	Overtime	100
Malware 40	AMR	4947.08 ± 491.76	Overtime	$94.45 \pm 2.21^{*}$	Overtime	$\mathbf{34.08 \pm 8.43}$
	Impute All	$\mathbf{7124.4}$	Overtime	$\mathbf{96.33}^{*}$	Overtime	100
Malware 60	AMR	$\mathbf{1369.67 \pm 8.65}$	Overtime	$95.53 \pm 0.63^{*}$	Overtime	$\mathbf{7.42 \pm 0.00}$
	Impute All	6061.5	Overtime	$\mathbf{96.08}^{*}$	Overtime	100
Tuadromd 20	AMR	$\mathbf{41.022 \pm 2.155}$	Overtime	98.60 ± 0.04	Overtime	$\mathbf{3.90 \pm 0.00}$
	Impute All	46.34	Overtime	$\mathbf{98.88}$	Overtime	100
Tuadromd 40	AMR	$\mathbf{36.407 \pm 0.043}$	Overtime	98.11 ± 0.04	Overtime	$\mathbf{2.18 \pm 0.00}$
	Impute All	44.13	Overtime	$\mathbf{98.80}$	Overtime	100
Tuadromd 60	AMR	$\mathbf{33.709 \pm 0.340}$	Overtime	97.68 ± 0.11	Overtime	$\mathbf{1.67 \pm 0.00}$
	Impute All	43.86	Overtime	$\mathbf{98.82}$	Overtime	100
Parkinsons 20	AMR	10.095 ± 0.562	348.00 ± 43.27	91.23 ± 1.52	91.23 ± 3.04	24.44 ± 1.92
	Impute All	89.168	1730.8	86.84	94.74	100
Parkinsons 40	AMR	55.697 ± 0.023	2341.00 ± 28.51	89.47 ± 0.00	81.58 ± 7.90	79.37 ± 0.00
	Impute All	92.174	3427.5	78.95	81.58	100
Parkinsons 60	AMR	29.455 ± 8.231	1455.00 ± 317.79	93.86 ± 4.02	89.47 ± 2.64	27.30 ± 7.98
	Impute All	179.120	5466.5	89.47	92.11	100
Heart 20	AMR	74.278 ± 0.093	2153.00 ± 81.30	84.75 ± 1.70	81.36 ± 1.70	97.87 ± 0.00
	Impute All	83.922	1794.4	83.05	86.44	100
Heart 40	AMR	82.305 ± 1.197	2275.00 ± 173.62	83.05 ± 0.00	71.19 ± 0.00	52.08 ± 0.00
	Impute All	162.939	3692.3	77.97	79.66	100
Heart 60	AMR	81.293 ± 1.276	2501.00 ± 39.95	51.41 ± 3.91	83.62 ± 1.96	34.72 ± 0.00
	Impute All	242.882	8467.1	45.76	79.66	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime

Table S27: Accuracy/time for FT Transformers on data with injected MCAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)		Accuracy(%)		Impute % of Samples	
			AC	AMR	AC	AMR	AC	AMR
Fraud	20	99.73	Overtime	273.90 ± 13.15	Overtime	99.89 ± 0.07	Overtime	0.03
	40	99.73	Overtime	386.00 ± 16.98	Overtime	99.90 ± 0.05	Overtime	0.01
	60	99.73	Overtime	405.00 ± 23.49	Overtime	99.87 ± 0.06	Overtime	0.01
HIGGS	20	74.29	Overtime	10450 ± 120	Overtime	74.99 ± 0.21	Overtime	0.07
	40	74.41	Overtime	10591 ± 135	Overtime	74.78 ± 0.11	Overtime	0.03
	60	74.36	Overtime	10815 ± 142	Overtime	74.39 ± 0.20	Overtime	0.02

Table S28: Accuracy/time for FT Transformers on data with injected MAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)		Accuracy(%)		Impute % of Samples	
			AC	AMR	AC	AMR	AC	AMR
Fraud	20	99.70	Overtime	181.50 ± 7.26	Overtime	99.86 ± 0.08	Overtime	0.01
	40	99.70	Overtime	384.50 ± 21.15	Overtime	99.64 ± 0.15	Overtime	0.20
	60	99.66	Overtime	404.80 ± 25.10	Overtime	99.85 ± 0.12	Overtime	0.13
HIGGS	20	74.31	Overtime	10511 ± 118	Overtime	74.17 ± 0.12	Overtime	0.23
	40	74.30	Overtime	10567 ± 129	Overtime	74.75 ± 0.09	Overtime	0.03
	60	74.41	Overtime	10839 ± 140	Overtime	75.11 ± 0.11	Overtime	12.12

Table S29: Accuracy/time for FT Transformers on data with injected MNAR

Data Set	% Missing	Ground Truth Accuracy(%)	Time(s)		Accuracy(%)		Impute % of Samples	
			AC	AMR	AC	AMR	AC	AMR
Fraud	20	99.70	Overtime	182.00 ± 6.55	Overtime	99.82 ± 0.10	Overtime	0.02
	40	99.70	Overtime	184.40 ± 9.96	Overtime	98.65 ± 0.23	Overtime	0.02
	60	99.69	Overtime	126.20 ± 11.11	Overtime	95.70 ± 0.29	Overtime	0.26
HIGGS	20	74.46	Overtime	10428 ± 117	Overtime	71.45 ± 0.24	Overtime	12.50
	40	74.53	Overtime	10692 ± 133	Overtime	71.50 ± 0.78	Overtime	12.50
	60	74.76	Overtime	10853 ± 144	Overtime	71.55 ± 0.24	Overtime	6.06

Table S30: Accuracy/time for FT-transformer on data with injected MCAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Fraud 20	AMR	826 ± 37.2	Overtime	Overtime	99.91 ± 0.07	Overtime	Overtime	0.03
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
Fraud 40	AMR	814 ± 31.7	Overtime	Overtime	99.90 ± 0.06	Overtime	Overtime	0.01
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
Fraud 60	AMR	808 ± 40.4	Overtime	Overtime	99.89 ± 0.08	Overtime	Overtime	0.01
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	0.07
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	0.03
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	0.02
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100

Table S30: Accuracy/time for FT-transformer on data with injected MCAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	0.03
	Impute All	Overtime	Overtime	Overtime	Overtime	100
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	0.01
	Impute All	Overtime	Overtime	Overtime	Overtime	100
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	0.01
	Impute All	Overtime	Overtime	Overtime	Overtime	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	0.07
	Impute All	Overtime	Overtime	Overtime	Overtime	100
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	0.03
	Impute All	Overtime	Overtime	Overtime	Overtime	100
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	0.02
	Impute All	Overtime	Overtime	Overtime	Overtime	100

Table S31: Accuracy/time for FT-transformer on data with injected MAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Fraud 20	AMR	765 ± 26.8	Overtime	Overtime	99.95 ± 0.05	Overtime	Overtime	0.01
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
Fraud 40	AMR	783 ± 36.0	Overtime	Overtime	99.92 ± 0.07	Overtime	Overtime	0.20
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
Fraud 60	AMR	788 ± 41.0	Overtime	Overtime	99.93 ± 0.06	Overtime	Overtime	0.13
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	0.23
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	0.03
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	12.12
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100

Table S31: Accuracy/time for FT-transformer on data with injected MAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	0.01
	Impute All	Overtime	Overtime	Overtime	Overtime	100
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	0.20
	Impute All	Overtime	Overtime	Overtime	Overtime	100
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	0.13
	Impute All	Overtime	Overtime	Overtime	Overtime	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	0.23
	Impute All	Overtime	Overtime	Overtime	Overtime	100
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	0.03
	Impute All	Overtime	Overtime	Overtime	Overtime	100
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	12.12
	Impute All	Overtime	Overtime	Overtime	Overtime	100

Table S32: Accuracy/time for FT-transformer on data with injected MNAR using model-based imputations — KNN, MICE, TCSDI

Data Set	Method	Time(s)			Accuracy(%)			Impute % of Samples
		KNN	MICE	TCSDI	KNN	MICE	TCSDI	
Fraud 20	AMR	798 ± 35.9	Overtime	Overtime	99.95 ± 0.08	Overtime	Overtime	0.02
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
Fraud 40	AMR	820 ± 45.1	Overtime	Overtime	99.93 ± 0.11	Overtime	Overtime	0.02
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
Fraud 60	AMR	821 ± 52.5	Overtime	Overtime	99.94 ± 0.14	Overtime	Overtime	0.26
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	12.50
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	12.50
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	6.06
	Impute All	Overtime	Overtime	Overtime	Overtime	Overtime	Overtime	100

Table S32: Accuracy/time for FT-transformer on data with injected MNAR using model-based imputations — MF, LLMF

Data Set	Method	Time(s)		Accuracy(%)		Impute % of Samples
		MF	LLMF	MF	LLMF	
Fraud 20	AMR	Overtime	Overtime	Overtime	Overtime	0.02
	Impute All	Overtime	Overtime	Overtime	Overtime	100
Fraud 40	AMR	Overtime	Overtime	Overtime	Overtime	0.02
	Impute All	Overtime	Overtime	Overtime	Overtime	100
Fraud 60	AMR	Overtime	Overtime	Overtime	Overtime	0.26
	Impute All	Overtime	Overtime	Overtime	Overtime	100
HIGGS 20	AMR	Overtime	Overtime	Overtime	Overtime	12.50
	Impute All	Overtime	Overtime	Overtime	Overtime	100
HIGGS 40	AMR	Overtime	Overtime	Overtime	Overtime	12.50
	Impute All	Overtime	Overtime	Overtime	Overtime	100
HIGGS 60	AMR	Overtime	Overtime	Overtime	Overtime	6.06
	Impute All	Overtime	Overtime	Overtime	Overtime	100

Table S33: Baseline and optimized average RAM usage (MB) across all SVM MCAR datasets using KNN, MICE, and MF imputations.

Data Set	Missingness	KNN (MB)	MICE (MB)	MF (MB)
Malware 20	MR	1117.88 ± 32.26	N/A	1590.53 ± 26.32
	Baseline	**948.41**	N/A	**1553.54**
Malware 40	MR	**1062.91 ± 6.64**	N/A	**1513.21 ± 17.08**
	Baseline	1361.96	N/A	1533.32
Malware 60	MR	**980.17 ± 8.10**	N/A	**1325.63 ± 8.14**
	Baseline	1427.75	N/A	1336.04
Tuadromd 20	MR	**231.82 ± 5.39**	449.32 ± 3.40	250.21 ± 5.11
	Baseline	256.53	**438.91**	**237.93**
Tuadromd 40	MR	**252.97 ± 0.00**	**442.13 ± 0.63**	258.38 ± 4.29
	Baseline	356.64	448.42	**249.85**
Tuadromd 60	MR	**248.74 ± 0.00**	**431.94 ± 1.04**	**261.36 ± 0.01**
	Baseline	410.12	450.02	265.16

Table S34: Baseline and optimized average RAM usage (MB) across all SVM MAR datasets using KNN, MICE, and MF imputations.

Data Set	Missingness	KNN (MB)	MICE (MB)	MF (MB)
Malware 20	MR	**993.87 ± 126.31***	N/A	**2007.92 ± 3.40**
	Baseline	994.18*	N/A	2188.73
Malware 40	MR	**917.31 ± 115.00**	N/A	**2021.64 ± 5.70**
	Baseline	1065.12	N/A	2187.40
Malware 60	MR	**816.95 ± 163.75**	N/A	**1772.30 ± 17.17**
	Baseline	1139.50	N/A	2203.27
Tuadromd 20	MR	**209.48 ± 15.13**	**433.15 ± 2.01**	**229.71 ± 0.62**
	Baseline	261.24	439.85	240.76
Tuadromd 40	MR	**203.47 ± 12.71**	**420.14 ± 0.06**	**215.76 ± 0.05**
	Baseline	315.21	437.52	239.30
Tuadromd 60	MR	**195.01 ± 15.02**	**407.45 ± 0.31**	**202.39 ± 0.74**
	Baseline	363.12	436.17	238.28

Table S35: Baseline and optimized average RAM usage (MB) across all SVM MNAR datasets using KNN, MICE, and MF imputations.

Data Set	Missingness	KNN (MB)	MICE (MB)	MF (MB)
Malware 20	MR	**917.23 ± 144.43***	N/A	**1987.47 ± 19.47**
	Baseline	1060.02*	N/A	2205.59
Malware 40	MR	**801.97 ± 149.17**	N/A	**1717.80 ± 11.01**
	Baseline	1167.60	N/A	2176.25
Malware 60	MR	**847.54 ± 182.56**	N/A	**1502.17 ± 0.11**
	Baseline	1254.33	N/A	2202.88
Tuadromd 20	MR	**207.31 ± 15.65**	**423.10 ± 1.04**	**216.53 ± 0.16**
	Baseline	286.32	439.73	240.33
Tuadromd 40	MR	**195.30 ± 15.87**	**407.96 ± 1.19**	**196.51 ± 0.23**
	Baseline	364.70	436.82	237.68
Tuadromd 60	MR	**189.19 ± 21.07**	**395.02 ± 2.05**	**182.25 ± 0.13**
	Baseline	412.00	438.30	238.52

E Probabilistic Bounds for Minimal Repair

As discussed in Section 3, Algorithm 1 approximates the minimal repair subset by evaluating the worst-case edge completions. While this efficiently identifies a valid subset, it may not capture every possible sample belonging to the true minimal repair. By iteratively applying the algorithm, the probability of missing a required sample strictly decreases.

When the underlying probability distributions of the missing values are known or can be estimated, we can formally quantify the probability that a sample belonging to the minimal repair remains unidentified by the algorithm.

Corollary E.1. *Let $g(x_{ij})$ denote the probability density function of the ground truth value for the missing value x_{ij} in the incomplete training set $(\mathbf{X}, \mathbf{y})$. Assume the missing values in $\mathbf{X}$ are independent. The probability that an incomplete sample $\mathbf{x}_i$ in the minimal repair is not returned by an iteration of Algorithm 1 is bounded by:*

$$p(\mathbf{x}_i) = 1 - \frac{\int \cdots \int_{\min(x_{ij}^{visited})}^{\max(x_{ij}^{visited})} \prod_{x_{ij} \in M(\mathbf{X})} g(x_{ij})\, dx_{ij}}{\int \cdots \int_{x_{ij} \in M(\mathbf{X})} \prod_{x_{ij} \in M(\mathbf{X})} g(x_{ij})\, dx_{ij}} \tag{10}$$

where $x_{ij}^{visited} \in \{x_{ij}^{min}, x_{ij}^{max}\}$ represents the extreme boundary values evaluated for x_{ij} during the execution of Algorithm 1.

F Robustness to Imputation Error

While our study focuses on the impact of different imputation strategies on downstream model performance, an important complementary direction is the development of learning methods that remain robust to potential imputation errors. This robustness-focused perspective is largely orthogonal to our goal of understanding how different subsets of data influence imputation quality, and it would require a substantial separate investigation beyond the scope of this work. Approaches based on robust optimization, for example, offer a principled way to train models that account for uncertainty in the imputed values and may help mitigate the effects of imputation variability. Exploring such robustness-oriented techniques represents a promising avenue for future work.

G Additional Related Work

Researchers have proposed methods that learn directly under uncertainty caused by missing values rather than repairing the data. For example, *stochastic optimization* learns a model by optimizing the expected loss over probability distributions of missing values in training samples Ganti & Willett (2015). Similarly, *robust optimization* minimizes the loss of a model under imputations that induce the largest training loss under a prescribed uncertainty model Aghasi et al. (2022). These approaches require specifying or learning distributions or uncertainty sets for the missing values, which can be difficult in practice. They also focus on learning under uncertainty in the model space rather than identifying which missing items should be repaired to reduce uncertainty in the data.

There are also methods that detect cases where imputing missing data is unnecessary for accurate learning Picado et al. (2020); Karlaš et al. (2020); Zhen et al. (2024). These methods are useful when the original incomplete dataset already supports accurate learning without further repair. However, they do not address the complementary problem studied here: constructing CM/ACM by repairing only a subset of incomplete samples when such models do not already exist in the original dataset.

Researchers have also proposed methods to reduce the cost of repair Krishnan et al. (2016); Karlaš et al. (2020). ActiveClean interleaves stochastic-gradient updates with sample cleaning and greedily selects samples whose repair may reduce the gradient the most Krishnan et al. (2016). CPClean follows a related greedy strategy but is limited to k-nearest-neighbor models over missing data Karlaš et al. (2020). Unlike our framework, these methods do not provide guarantees of minimality or almost minimality for the selected repairs.

H PROOFS

PROOF FOR THEOREM 3.2

Proof. For a completion $\mathbf{X}^c$, let $\mathbf{w}^*(\mathbf{X}^c)$ denote its optimal SVM parameter vector, which is unique for the ℓ_2-regularized linear SVM considered here Zhen et al. (2024). By Definition 3.1, a set S is a valid repair exactly when, for every admissible assignment to the missing entries of the samples in S, all completions consistent with that assignment have the same optimizer.

Lemmas 5.1–5.3 of Ref. Zhen et al. (2024) explain how this invariance can arise for a certain linear SVM. In Case 1, no unrepaired incomplete sample is a support vector in any completion, and removing such samples does not change the optimal SVM. In Case 2, the coefficients of all incomplete features are zero, so their missing values do not affect the optimal model. Thus the argument below covers both cases identified in that reference.

Suppose, for contradiction, that there are two distinct minimal repairs S_1 and S_2, and let $I = S_1 \cap S_2$. Fix an arbitrary admissible assignment to the missing entries in I, and consider any two completions $\mathbf{X}^a$ and $\mathbf{X}^b$ consistent with this assignment. Construct a hybrid completion $\mathbf{X}^h$ by taking the completed values of every sample in S_1 from $\mathbf{X}^a$ and those of every other sample from $\mathbf{X}^b$. This is an admissible completion because every missing entry retains a value from an admissible completion.

The completions $\mathbf{X}^a$ and $\mathbf{X}^h$ agree on all samples in S_1. Since S_1 is a repair, they therefore have the same optimizer. Moreover, $\mathbf{X}^h$ and $\mathbf{X}^b$ agree on all samples in S_2: they agree on $S_1 \cap S_2 = I$ because $\mathbf{X}^a$ and $\mathbf{X}^b$ share the fixed assignment on I, and for every sample in $S_2 \setminus S_1$, the hybrid uses the values from $\mathbf{X}^b$. Since S_2 is also a repair, $\mathbf{X}^h$ and $\mathbf{X}^b$ have the same optimizer. Hence

$$\mathbf{w}^*(\mathbf{X}^a) = \mathbf{w}^*(\mathbf{X}^h) = \mathbf{w}^*(\mathbf{X}^b).$$

Because the assignment on I and the two consistent completions were arbitrary, I is itself a valid repair. But distinct minimum-cardinality repairs have a strict intersection, so $|I| < |S_1| = |S_2|$, contradicting the minimality of S_1 and S_2. Therefore, the minimal repair is unique. □

PROOF FOR LEMMA 3.3

Proof. Under the Case 1 scope stated in Section 3, let

$$P = \{\mathbf{x}_i \in MS(\mathbf{X}) : \mathbf{x}_i^c \text{ is a support vector in at least one completion } \mathbf{X}^c \text{ of } \mathbf{X}\}.$$

First, P is a valid repair. Fix any admissible imputation values for the samples in P. Every remaining incomplete sample $\mathbf{x}_j \notin P$ is not a support vector in any completion of the original dataset and therefore is not a support vector in any completion consistent with the fixed values. Removing these non-support-vector samples does not change the optimal SVM. Consequently, every remaining completion has the same optimizer as the fixed dataset obtained after removing the remaining incomplete samples, which is the Case 1 argument of Lemmas 5.1 and 5.2 in Ref. Zhen et al. (2024).

It remains to show that every valid repair S contains P. Suppose that some $\mathbf{x}_i \in P$ is not in S. There is then a completion $\mathbf{X}^c$ in which $\mathbf{x}_i^c$ is a support vector. Assign the repaired entries in S the same values that they have in $\mathbf{X}^c$. The completion $\mathbf{X}^c$ remains admissible after this assignment, and $\mathbf{x}_i$ remains incomplete and is a support vector in that completion. The resulting partially repaired dataset therefore cannot satisfy the Case 1 condition for a certain SVM, contradicting that S is a valid repair under the stated scope. Thus $P \subseteq S$ for every valid repair S.

Therefore, P is the unique minimal repair, proving that an incomplete sample belongs to the minimal repair if and only if it is a support vector in at least one completion. □

PROOF FOR THEOREM 3.4

Necessity is trivial based on Lemma 2: if an incomplete sample is a support vector in an edge completion, the incomplete sample is part of the minimal repair set. Then we prove sufficiency by contradiction. Assume that there is an incomplete sample $\mathbf{x}_i$ part of the minimal repair set X_{min} while it is not a support vector in any edge completion $\mathbf{x}^e \in \mathcal{X}^E$. Training an SVM can be interpreted as finding the minimal distance between two reduced convex hulls Bennett & Bredensteiner (2000), and if an sample is within the reduced convex hull (not at the boundary), the sample is not a support

vector. Because $\mathbf{x}_i$ is not a support vector for any edge completion from the assumption, it is not a support vector for any completion of $\mathbf{X}$. This is because, in the process of changing a value for a missing value (x_{pq}) from one edge completion (x_{pq}^{min}) to another (x_{pq}^{max}) monotonically increase or decrease the coverage of the reduced convex hull. With that being said, if an incomplete sample $\mathbf{x}_i$ is not a support vector for any edge completion (i.e., within the reduced convex hull), the incomplete sample is within the reduced convex hull (i.e., not a support vector) with respect to any completion. This contradicts to the original assumption that $\mathbf{x}_i$ is part of the minimal repair set.

PROOF FOR THEOREM 3.5

We reduce from the NP-complete problem 3-SAT. Let

$$\Phi \;=\; \bigwedge_{j=1}^{m} (C_j)$$

be a 3-SAT formula with k Boolean variables $z_1, z_2, \ldots, z_k$ and m clauses $C_1, \ldots, C_m$, each clause being a disjunction of three literals.

For each variable z_ℓ, we introduce one or more *incomplete* samples whose feature vectors each contain a *missing* coordinate u_ℓ. The imputation set for u_ℓ is $\{-1, +1\}$, corresponding to $\{\text{False}, \text{True}\}$. Thus, any assignment of the z_ℓ corresponds to choosing ± 1 for these missing coordinates.

To enforce that each clause C_j must be satisfied, we add appropriately labeled points (some possibly incomplete) and arrange them in a geometry so that assigning a literal to *false* yields a large penalty term in the soft-margin objective (either by misclassification or forcing the margin to collapse). Intuitively, if a clause were unsatisfied (all literals set to *false*), the SVM would incur a prohibitively large hinge-loss cost, making that repair suboptimal.

We designate one particular incomplete sample $\mathbf{x}_i$ with additional coordinates or constraints so that:

- *If* Φ is satisfiable, then there is an imputation (choosing ± 1 consistently with a satisfying assignment) that maximizes the margin while placing $\mathbf{x}_i$ *exactly on* the decision boundary, making it a support vector.
- *If* Φ is unsatisfiable, then *every* imputation leads to $\mathbf{x}_i$ being off the margin (either strictly inside or otherwise not a support vector). In other words, no selection of $\{\pm 1\}$ for the missing attributes can force $\mathbf{x}_i$ onto the margin.

By suitably tuning the soft-margin parameter C and the placement of the clause-encoding points, we ensure that the SVM will "prefer" to assign ± 1 values in a way that satisfies Φ, whenever possible, in order to avoid a large penalty.

Hence,

$$\Phi \text{ is satisfiable} \iff \text{there exists a repair making } \mathbf{x}_i \text{ a support vector.}$$

Since deciding satisfiability for Φ (3-SAT) is NP-complete, it follows that deciding whether $\mathbf{x}_i$ can be a support vector under some imputation is NP-hard.

Determining membership of a single incomplete sample $\mathbf{x}_i$ among the possible support vectors is NP-hard. Therefore, listing *all* such samples that can ever appear on the margin is also NP-hard: if we had such a list in polynomial time, we could decide membership in that list in polynomial time, contradicting NP-hardness. Given the proof that finding MR for SVM is NP-hard, deciding whether an incomplete sample belongs to the MR for SVM is also NP hard. To prove, assume that we have a polynomial-time solver for deciding whether an incomplete sample belongs to the MR, then one can linearly scan each incomplete sample and decide its membership in MR (either belongs to or not) by calling the polynomial time subroutine. Therefore, one can find the MR in polynomial time, which contradicts to the NP-hard proof earlier.

PROOF FOR THEOREM 3.6

Proof. Consider any incomplete sample $\mathbf{x}_i$ returned by Algorithm 1. Let $\mathbf{X}^e$ be the final edge completion constructed for $\mathbf{x}_i$, let $\mathbf{x}_i^e$ denote the corresponding completion of that sample, and let $\mathbf{w}$

be the optimal SVM model trained on $(\mathbf{X}^e, \mathbf{y})$. By the selection condition in Algorithm 1,

$$y_i \mathbf{w}^\top \mathbf{x}_i^e \leq 1.$$

Thus $\mathbf{x}_i^e \in SV(\mathbf{X}^e, \mathbf{y}, C)$. Because $\mathbf{X}^e$ is an admissible completion of $\mathbf{X}$, Lemma 3.3 implies, under the Case 1 scope stated in Section 3, that $\mathbf{x}_i \in S_{MR}(\mathbf{X}, \mathbf{y}, C)$. Therefore, every sample returned by Algorithm 1 belongs to the minimal repair. □

PROOF FOR THEOREM 3.7

Given the iterative algorithm of finding the minimal repair for SVM (Algorithm 1 in the main content), we first characterize the probability that the imputation set returned at iteration k misses one or more incomplete samples that belong to the minimal repair.

Let k be the current iteration index ($k = 0$ represents the initial state before the first run). We define the following: $MS(x)^k$ is the set of incomplete samples remaining at the start of iteration k. $M^k = |MS(x)^k|$ is the number of remaining incomplete samples at the start of iteration k. S_{min}^k is the (unknown) true minimal set of samples within $MS(x)^k$ that must be imputed at the start of iteration k to guarantee a certain model. $s^k = |S_{min}^k|$ is the (unknown) size of this true minimal set; note that we treat s^k as a random variable, and $s^k \leq M^k$. S'^k is the set of samples returned by Algorithm 1 in the main content when run at iteration k on the current data; we know $S'^k \subseteq S_{min}^k$. FN^k is the event that makes at least one false negative error at iteration k, occurring if S'^k is a proper subset of S_{min}^k. $P(FN^k)$ is the probability of event FN^k. We seek a computable upper bound UB'^k such that $P(FN^k) \leq UB'^k$. define p_{fn} as an upper bound on the per-sample false negative probability, $p(\mathbf{x_i})$. We assume that there exists a probability p_{fn} (where $0 \leq p_{fn} \leq 1$) such that for any sample $x_i \in S_{min}^k$, the probability that Algorithm 1 in the main content fails to include x_i in S'^k is bounded above by p_{fn}:

$$P(x_i \notin S'^k | x_i \in S_{min}^k) \leq p_{fn}$$

Then we propose, UB'^k, an upper bound of $P(FN^k)$ as follows:

$$UB'^k = 1 - (1 - p_{fn})^{M^k} \geq P(FN^k)$$

To interpret, when the iteration goes (k becomes larger), M^k and p_{fn} decrease (which we will prove later), UB'^k decreases. This indicates that the upper-bound of probability of under-imputing decreases over iterations.

To prove this bound, we begin by expressing the target probability $P(FN^k)$ using its complement. The event FN^k (at least one false negative) is the complement of the event $NoFN^k$ (no false negatives, i.e., $S'^k = S_{min}^k$). Therefore, conditioned on the true size s^k of the minimal set at iteration k, we have $P(FN^k | s^k) = 1 - P(\text{No FN}^k | s^k)$.

Next, we bound the probability of having no false negatives, $P(\text{No FN}^k | s^k)$. The event $NoFN^k$ occurs if Algorithm 1 in the main content successfully returns all samples in S_{min}^k. Let E_i be the event that Algorithm 1 in the main content fails to return sample x_i. Assuming the failure/success events E_i for different samples $x_i \in S_{min}^k$ within the same iteration k are statistically independent, we can write:

$$P(\text{No FN}^k | s^k) = P(\cap_{x_i \in S_{min}^k} \{\text{not } E_i\} | s^k) = \prod_{x_i \in S_{min}^k} P(\text{not } E_i | s^k)$$

Let $P(E_i | s^k)$ be the probability of failure for x_i. Then $P(\text{not } E_i | s^k) = 1 - P(E_i | s^k)$. Using the definition $P(E_i | s^k) \leq p_{fn}$, we have $1 - P(E_i | s^k) \geq 1 - p_{fn}$. Substituting this lower bound into the product gives:

$$P(\text{No FN}^k | s^k) \geq \prod_{i=1}^{s^k} (1 - p_{fn}) = (1 - p_{fn})^{s^k}$$

Now we can bound $P(FN^k | s^k)$:

$$P(FN^k | s^k) = 1 - P(\text{No FN}^k | s^k) \leq 1 - (1 - p_{fn})^{s^k}$$

The overall probability $P(FN^k)$ is the expectation over the unknown size s^k:

$$P(FN^k) = \mathop{\mathbb{E}}_{s^k}[P(FN^k|s^k)] \leq \mathop{\mathbb{E}}_{s^k}[1-(1-p_{fn})^{s^k}]$$

To proceed, we utilize Jensen's inequality. Let $f(s) = 1-(1-p_{fn})^s$. We first prove that $f(s)$ is concave for $s \geq 0$. Let $b = 1-p_{fn}$. Since $0 \leq p_{fn} < 1$, we have $0 < b \leq 1$. The function is $f(s) = 1-b^s$. The first derivative is $f'(s) = -b^s \ln(b)$. The second derivative is $f''(s) = -(b^s \ln(b))\ln(b) = -b^s(\ln(b))^2$. Since $b^s > 0$ and $(\ln(b))^2 \geq 0$, the second derivative $f''(s) \leq 0$. Therefore, $f(s)$ is a concave function.

Jensen's inequality for a concave function f states $\mathbb{E}[f(X)] \leq f(\mathbb{E}[X])$. Applying this to our expectation:

$$\mathop{\mathbb{E}}_{s^k}[1-(1-p_{fn})^{s^k}] \leq 1-(1-p_{fn})^{\mathbb{E}[s^k]}$$

Combining this with the previous inequality gives a theoretical upper bound:

$$P(FN^k) \leq 1-(1-p_{fn})^{\mathbb{E}[s^k]}$$

The term $\mathbb{E}[s^k]$ (expected number of truly needed samples) is still unknown. However, we know that the number of needed samples s^k cannot exceed the total number of remaining incomplete samples $M^k = |MS(x)^k|$. Thus, $s^k \leq M^k$. Taking expectations yields $\mathbb{E}[s^k] \leq \mathbb{E}[M^k]$. Since M^k is a known quantity (computable by counting) at the start of iteration k, $\mathbb{E}[M^k] = M^k$. Therefore, we have a computable upper bound for the expectation: $\mathbb{E}[s^k] \leq M^k$.

Finally, we substitute this bound on $\mathbb{E}[s^k]$ into the Jensen result. Let $g(x) = (1-p_{fn})^x$. Since $0 < (1-p_{fn}) \leq 1$, $g(x)$ is a non-increasing function. Applying g to the inequality $\mathbb{E}[s^k] \leq M^k$ reverses the inequality direction:

$$(1-p_{fn})^{\mathbb{E}[s^k]} \geq (1-p_{fn})^{M^k}$$

Multiplying by -1 and adding 1 (reversing the inequality twice):

$$1-(1-p_{fn})^{\mathbb{E}[s^k]} \leq 1-(1-p_{fn})^{M^k}$$

Combining the inequalities $P(FN^k) \leq 1-(1-p_{fn})^{\mathbb{E}[s^k]}$ and $1-(1-p_{fn})^{\mathbb{E}[s^k]} \leq 1-(1-p_{fn})^{M^k}$, we arrive at the final upper bound UB'^k:

$$P(FN^k) \leq 1-(1-p_{fn})^{M^k}$$

and

$$UB'^k = 1-(1-p_{fn})^{|MS(x)^k|}$$

Now the only problem is to compute p_{fn} and understand how it changes over iterations. The Multiple Random Starts method provides an empirical approach. First, select a set of incomplete samples MS_{probe} (e.g., $MS(x)^0$) and choose the number of repetitions T (e.g., $T = 10$ or 20). For each $x_i \in MS_{probe}$, initialize a success count $t_i = 0$. Repeat T times: generate a new random edge completion $X^e_{start,t}$ for the current dataset state; run the greedy construction part of Algorithm 1 in the main content starting from $X^e_{start,t}$ to get $X^e_{final,i,t}$; train $w_{final,i,t} = SVM(X^e_{final,i,t}, y)$; check if $y_i(w_{final,i,t})^T(x_i \text{ part of } X^e_{final,i,t}) \leq 1$. If yes, increment t_i.

Also, if the probability distribution of each incomplete sample is known, and we let $g(x_{ij})$ denote the probability density function of the ground truth value for the missing value x_{ij} in the incomplete training set $(\mathbf{X}, \mathbf{y})$. If missing values in $\mathbf{X}$ are independent, the probability that an incomplete sample $\mathbf{x}_i$ in minimal repair not returned by Algorithm 1 in the main content is:

$$p(\mathbf{x}_i) = 1 - \frac{\int \cdots \int_{\min(x_{ij}^{\text{visited}})}^{\max(x_{ij}^{\text{visited}})} \prod_{x_{ij} \in M(\mathbf{X})} g(x_{ij})\, dx_{ij}}{\int \cdots \int_{x_{ij} \in M(\mathbf{X})} \prod_{x_{ij} \in M(\mathbf{X})} g(x_{ij})\, dx_{ij}}$$

$x_{ij}^{\text{visited}} \in \{x_{ij}^{min}, x_{ij}^{max}\}$ shows the values used for x_{ij} in Algorithm 1 in the main content. It shows that the more edge completions Algorithm 1 explores, the lower the false negative probability for each sample. One can find p_{fn} by computing $p(\mathbf{x}_i)$ for each incomplete sample and take the maximum as p_{fn}. p_{fn} decreases over iterations because each iteration explores additional edge completions. This expands the domain of the numerator in the expression increasing the integral value and thereby lowering $p(\mathbf{x}_i)$ for every sample5. Since p_{fn} is an upper bound over all such $p(\mathbf{x}_i)$, it decreases as well.

PROOF FOR THEOREM 4.2

Proof. For any fixed completion $\mathbf{X}^c$, the term $\min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}^c, \mathbf{y})$ represents the minimum achievable loss on that specific completed dataset. Crucially, this term is a scalar constant with respect to the model parameters $\mathbf{w}$. Because the training objective $L(\mathbf{w}, \mathbf{X}^c, \mathbf{y})$ is convex in $\mathbf{w}$ for each fixed completion $\mathbf{X}^c$, the gap function:

$$h(\mathbf{w}, \mathbf{X}^c) = L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) - \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}^c, \mathbf{y})$$

is simply a convex function minus a scalar constant. Therefore, $h(\mathbf{w}, \mathbf{X}^c)$ is convex with respect to $\mathbf{w}$.

A fundamental property of convex analysis is that the pointwise supremum of any family of convex functions is itself a convex function. Thus, the robust gap over the space of allowed repairs $\mathcal{R}(S)$:

$$G(\mathbf{w}, S) = \sup_{\mathbf{X}^c\in\mathcal{R}(S)} h(\mathbf{w}, \mathbf{X}^c)$$

forms a convex upper envelope over $\mathbf{w}$. An ACM requires that for all valid completions, the gap is bounded by e. This is mathematically equivalent to requiring that the maximum possible gap (the peak of the envelope evaluated at the optimal $\mathbf{w}$) does not exceed e, i.e., $\min_{\mathbf{w}} G(\mathbf{w}, S) \leq e$. This completes the proof. □

PROOF FOR LEMMA B.1

Proof. At termination, Algorithm 2 returns a repaired set S_T, an actually repaired dataset $\widehat{\mathbf{X}}^{(T)}$, a model $\mathbf{w}_T$, and a certificate Γ_T. By assumption,

$$\sup_{\mathbf{X}\in\mathcal{R}(S_T)} \left(L(\mathbf{w}_T, \mathbf{X}, \mathbf{y}) - \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}, \mathbf{y}) \right) \leq \Gamma_T.$$

If $\Gamma_T \leq e$, then directly

$$\sup_{\mathbf{X}\in\mathcal{R}(S_T)} \left(L(\mathbf{w}_T, \mathbf{X}, \mathbf{y}) - \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}, \mathbf{y}) \right) \leq e.$$

Therefore $\mathbf{w}_T$ is an ACM for all completions consistent with $\widehat{\mathbf{X}}^{(T)}$. Likewise, if $\Gamma_T \leq e + \delta$ for some $\delta \geq 0$, then

$$\sup_{\mathbf{X}\in\mathcal{R}(S_T)} \left(L(\mathbf{w}_T, \mathbf{X}, \mathbf{y}) - \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}, \mathbf{y}) \right) \leq e + \delta,$$

so $\mathbf{w}_T$ is an $(e + \delta)$-ACM.

The certificate is recomputed after each repair round on the updated partially repaired dataset. Hence, when the algorithm stops, the guarantee is stated for the actually repaired trajectory and the remaining uncertainty. □

PROOF FOR COROLLARY B.2

Proof. Suppose the screened completion is exact by endpoint search and let the certificate be of the form

$$\Gamma_T = Gap_{curr} + R_T,$$

where

$$Gap_{curr} = L(\mathbf{w}_T, \mathbf{X}_{adv}, \mathbf{y}) - \phi(\mathbf{X}_{adv})$$

and R_T is a valid upper bound on the contribution of all unseen completions in $\mathcal{R}(S_T)$. Then, by construction,

$$\sup_{\mathbf{X}\in\mathcal{R}(S_T)} \left(L(\mathbf{w}_T, \mathbf{X}, \mathbf{y}) - \min_{\mathbf{v}\in\mathcal{W}} L(\mathbf{v}, \mathbf{X}, \mathbf{y}) \right) \leq \Gamma_T.$$

If $Gap_{curr} + R_T \leq e$, then $\Gamma_T \leq e$, and Lemma B.1 implies that $\mathbf{w}_T$ is an ACM. The same argument gives the $(e + \delta)$ version if $Gap_{curr} + R_T \leq e + \delta$. □

PROOF FOR PROPOSITION B.3

Proof. If $S \subseteq T$, then every completion feasible after repairing T is also feasible after repairing S, because repairing more samples only fixes additional coordinates and cannot create new uncertainty. Therefore

$$\mathcal{R}(T) \subseteq \mathcal{R}(S).$$

For any fixed $\mathbf{w}$,

$$\sup_{\mathbf{X}^c \in \mathcal{R}(T)} \left(L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) - \min_{\mathbf{v} \in \mathcal{W}} L(\mathbf{v}, \mathbf{X}^c, \mathbf{y}) \right) \leq \sup_{\mathbf{X}^c \in \mathcal{R}(S)} \left(L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}) - \min_{\mathbf{v} \in \mathcal{W}} L(\mathbf{v}, \mathbf{X}^c, \mathbf{y}) \right).$$

Minimizing both sides over $\mathbf{w}$ gives $J(T) \leq J(S)$. Since $V(S) = \max\{J(S) - e, 0\}$ is monotone in $J(S)$, we also obtain $V(T) \leq V(S)$. □

PROOF FOR PROPOSITION 4.3

Proof. For fixed $(\mathbf{w}, \mathbf{s})$, the constraint $g(\mathbf{w}, \mathbf{s}) \leq e$ is equivalent to

$$\max_{\mathbf{z} \in [-1,1]^{|\mathcal{M}|}} q(\mathbf{w}, \mathbf{X}(\mathbf{s}, \mathbf{z})) \leq e.$$

Introducing a nonnegative multiplier λ for this scalar inequality yields the Lagrangian

$$\sum_{i=1}^{m} s_i + \lambda \left(\max_{\mathbf{z} \in [-1,1]^{|\mathcal{M}|}} q(\mathbf{w}, \mathbf{X}(\mathbf{s}, \mathbf{z})) - e \right).$$

Because $\lambda \geq 0$, the maximization over $\mathbf{z}$ can be moved inside:

$$\sum_{i=1}^{m} s_i + \max_{\mathbf{z} \in [-1,1]^{|\mathcal{M}|}} \lambda \big(q(\mathbf{w}, \mathbf{X}(\mathbf{s}, \mathbf{z})) - e \big) = \max_{\lambda \geq 0,\, \mathbf{z} \in [-1,1]^{|\mathcal{M}|}} \left[\sum_{i=1}^{m} s_i + \lambda \big(q(\mathbf{w}, \mathbf{X}(\mathbf{s}, \mathbf{z})) - e \big) \right].$$

This gives the stated primal-dual reformulation. □

PROOF FOR PROPOSITION B.4

Proof. Let $\mathbf{X}_q \in \arg\max_{\mathbf{X} \in U(\mathbf{s})} q(\mathbf{w}, \mathbf{X})$ and $\mathbf{X}_L \in \arg\max_{\mathbf{X} \in U(\mathbf{s})} L(\mathbf{w}, \mathbf{X}, \mathbf{y})$. Then

$$q(\mathbf{w}, \mathbf{X}_q) - q(\mathbf{w}, \mathbf{X}_L) = \big(L(\mathbf{w}, \mathbf{X}_q, \mathbf{y}) - L(\mathbf{w}, \mathbf{X}_L, \mathbf{y}) \big) - \big(\phi(\mathbf{X}_q) - \phi(\mathbf{X}_L) \big).$$

Since $\mathbf{X}_L$ maximizes $L(\mathbf{w}, \mathbf{X}, \mathbf{y})$ over $U(\mathbf{s})$, the first bracket is nonpositive. Therefore

$$q(\mathbf{w}, \mathbf{X}_q) - q(\mathbf{w}, \mathbf{X}_L) \leq \phi(\mathbf{X}_L) - \phi(\mathbf{X}_q) \leq \sup_{\mathbf{X} \in U(\mathbf{s})} \phi(\mathbf{X}) - \inf_{\mathbf{X} \in U(\mathbf{s})} \phi(\mathbf{X}),$$

which proves the first claim. If ϕ is K_ϕ-Lipschitz on $U(\mathbf{s})$, then the variation of ϕ over $U(\mathbf{s})$ is at most $K_\phi \operatorname{diam}(U(\mathbf{s}))$, which gives the second claim. □

PROOF FOR THEOREM B.5

Proof. Fix any completion $\mathbf{X}^c$ and define

$$f_{\mathbf{X}^c}(\mathbf{w}) = L(\mathbf{w}, \mathbf{X}^c, \mathbf{y}).$$

By assumption, $f_{\mathbf{X}^c}$ is convex and has M-Lipschitz continuous gradient. Let $\mathbf{w}^*(\mathbf{X}^c) \in \arg\min_{\mathbf{v} \in \mathcal{W}} f_{\mathbf{X}^c}(\mathbf{v})$. A standard inequality for convex M-smooth functions gives

$$f_{\mathbf{X}^c}(\mathbf{w}) - f_{\mathbf{X}^c}(\mathbf{w}^*(\mathbf{X}^c)) \leq \frac{1}{2M} \|\nabla f_{\mathbf{X}^c}(\mathbf{w})\|^2.$$

Therefore, if

$$\|\nabla_{\mathbf{w}} L(\mathbf{w}^{\approx}, \mathbf{X}^c, \mathbf{y})\| \leq \sqrt{2Me},$$

then

$$L(\mathbf{w}^{\approx}, \mathbf{X}^c, \mathbf{y}) - \min_{\mathbf{v} \in \mathcal{W}} L(\mathbf{v}, \mathbf{X}^c, \mathbf{y}) \leq e.$$

Since this holds for every completion $\mathbf{X}^c$, $\mathbf{w}^{\approx}$ is an ACM. □

PROOF FOR THEOREM 4.4

Proof. Let S_t be the repaired set after round t, let S^* be an optimal solution to Equation 3, and set $k = \min\{K, |S^*|\}$. Since S^* is feasible, $V(S^*) = 0$ and $F(S^*) = V(\emptyset)$.

Starting from $A_0 = S_t$, construct a conceptual sequential-greedy batch of k elements. At step r, choose an element b_r with the largest singleton marginal gain and set $A_{r+1} = A_r \cup \{b_r\}$. If $V(A_r) = 0$, no further argument is needed. Otherwise, apply the weak-submodularity-ratio inequality with $L = A_r$ and $A = S^* \setminus A_r$. Monotonicity gives

$$F(A_r \cup S^*) - F(A_r) \geq F(S^*) - F(A_r) = V(A_r).$$

Hence

$$\sum_{j \in S^* \setminus A_r} \big(F(A_r \cup \{j\}) - F(A_r)\big) \geq \gamma V(A_r).$$

There are at most $|S^*|$ terms in this sum, so the best singleton marginal satisfies

$$F(A_{r+1}) - F(A_r) \geq \frac{\gamma}{|S^*|} V(A_r).$$

Using $F(A_{r+1}) - F(A_r) = V(A_r) - V(A_{r+1})$, we obtain

$$V(A_{r+1}) \leq \left(1 - \frac{\gamma}{|S^*|}\right) V(A_r).$$

After k conceptual greedy additions,

$$V(A_k) \leq \left(1 - \frac{\gamma}{|S^*|}\right)^k V(S_t) \leq \exp\left(-\frac{\gamma k}{|S^*|}\right) V(S_t).$$

Thus the gain of this batch is at least

$$F(A_k) - F(S_t) \geq \left(1 - \exp\left(-\frac{\gamma k}{|S^*|}\right)\right) V(S_t).$$

The best batch of size at most K has at least this gain. By the α-approximate batch-selection assumption,

$$F(S_{t+1}) - F(S_t) \geq \alpha \left(1 - \exp\left(-\frac{\gamma k}{|S^*|}\right)\right) V(S_t) = \rho V(S_t).$$

Therefore $V(S_{t+1}) \leq (1 - \rho) V(S_t)$, and iteration gives

$$V(S_T) \leq (1 - \rho)^T V(\emptyset) \leq e^{-\rho T} V(\emptyset).$$

Solving $e^{-\rho T} V(\emptyset) \leq \varepsilon$ gives the stated iteration bound. □

PROOF FOR THEOREM 4.5

Proof. The algorithm stops once $V(S_T) \leq \varepsilon_{cert}$. Under $K \leq |S^*|$, Theorem 4.4 gives

$$T \leq \left\lceil \frac{1}{\alpha\,(1 - \exp(-\gamma K/|S^*|))} \log \frac{V(\emptyset)}{\varepsilon_{cert}} \right\rceil.$$

Since each outer iteration repairs at most K new samples,

$$|S_T| \leq KT \leq K \left\lceil \frac{1}{\alpha\,(1 - \exp(-\gamma K/|S^*|))} \log \frac{V(\emptyset)}{\varepsilon_{cert}} \right\rceil.$$

Using $\lceil a \rceil \leq a + 1$ gives

$$|S_T| \leq \frac{K}{\alpha\,(1 - \exp(-\gamma K/|S^*|))} \log \frac{V(\emptyset)}{\varepsilon_{cert}} + K.$$

Finally, $x = \gamma K/|S^*|$ lies in $[0, 1]$, and concavity of $1 - e^{-x}$ on this interval gives $1 - e^{-x} \geq (1 - e^{-1})x$. Substitution yields

$$|S_T| \leq \frac{|S^*|}{\alpha(1 - e^{-1})\gamma} \log \frac{V(\emptyset)}{\varepsilon_{cert}} + K.$$

□

PROOF FOR COROLLARY E.1

Proof. By Theorem 3.3, an incomplete sample $\mathbf{x}_i$ belongs to the minimal repair set if and only if there exists at least one valid completion $\mathbf{X}^c \in \mathcal{X}^C$ such that the completed sample $\mathbf{x}_i^c$ acts as a support vector.

Algorithm 1 searches for this condition by evaluating the boundaries of the uncertainty space, x_{ij}^{min} and x_{ij}^{max}, for each missing feature. Because the decision boundary of linear SVM and the sample-wise loss are monotonic with respect to individual features, the maximum empirical risk fluctuation occurs precisely at these boundaries.

If the algorithm evaluates a subset of extreme values $x_{ij}^{\text{visited}} \in \{x_{ij}^{min}, x_{ij}^{max}\}$ and fails to identify $\mathbf{x}_i$ as a support vector, then $\mathbf{x}_i$ can only be a support vector if the true, unobserved ground truth data lies outside the hyper-rectangle defined by the evaluated extremes.

Assuming the missing values are independent, the joint probability distribution of all missing values in the dataset is the product of their individual marginal density functions, $\prod g(x_{ij})$.

The denominator in Equation 1 represents the total probability mass over the entire valid domain of the missing values. The numerator represents the probability mass strictly contained within the bounds evaluated by the algorithm. Therefore, the ratio represents the probability that the ground truth data lies within the searched space.

Consequently, the probability $p(\mathbf{x}_i)$ that the true worst-case completion (and thus the support vector status of $\mathbf{x}_i$) was missed is exactly 1 minus the probability that the ground truth lies within the evaluated bounds. This yields the exact formulation in Equation 1. □

I LIMITATIONS

Our work has several limitations.

First, the proposed methods do not in general guarantee recovery of the exact minimal repair (MR) or exact almost minimal repair (AMR). This limitation is inherent to the problem setting: as we show in the paper, finding MR or AMR for the model classes considered here is NP-hard. Accordingly, our methods rely on tractable approximations and relaxations rather than exact combinatorial optimization.

Second, for non-convex models, we do not provide guarantees that the algorithm converges to an AMR, nor do we establish an upper bound on the approximation ratio of the returned repair set. This reflects the difficulty of optimization and certification in non-convex settings, where global optimality and worst-case guarantees are generally hard to obtain. There is also no universal mapping from the gradient and loss thresholds to downstream accuracy, and differentiating the adversarial gradient norm can add GPU-memory overhead. Extending the guarantees and characterizing these tradeoffs are important directions for future work.

Third, the formulation is modality-agnostic only after the data have been represented as a supervised feature matrix. Images can be represented using pixels or learned features, and time series can be represented using supervised windows, but the current algorithms treat missing coordinates independently. They do not exploit spatial or temporal correlation and may therefore be conservative for structured data. Incorporating such structure is future work.