ACTIVE FEATURE ACQUISITION WITH INCOMPLETE TRAINING DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

In many prediction tasks, acquiring all features can be a prohibitively expensive or outright impossible task. Further, in many cases a static subset of features may not be enough to solve the problem sufficiently across various instances. Active Feature Acquisition (AFA) addresses these problems by formalizing the trade-off between feature cost and predictive performance during sequential feature selection. However, prior AFA work largely assumes access to *complete training data*, an assumption that is often violated in practice. Here we study **AFA** with **I**ncomplete **T**raining **D**ata (**AFA-ITD**), showing that under missing completely at random (MCAR) data, one-step acquisition values remain unchanged, whereas multi-step values can decrease. We analyze three approaches to learning from incomplete data: *aliasing, filtering, and generative restoration*. We show that filtering can require a number of training instances scaling exponentially with the dimension, whereas generative restoration scales exponentially with the acquisition budget. We empirically test our theory on a controlled experiment and across common AFA datasets and find that missingness mainly damages methods that exploit multi-step acquisitions and that generative restoration is able to recover lost performance in many experiments.

1 INTRODUCTION

Obtaining information in the real world is often a laborious and costly task. Medical tests require considerable time and resources, recommender systems need a continual collection of information at the price of compromising user privacy, and robotic systems expend energy on every sensor measurement (Gorry & Barnett, 1968; Jeckmans et al., 2013; Lauri et al., 2023). Also, acquiring all features before producing a prediction may be unnecessary or impractical. This has motivated *cost-sensitive learning with feature acquisition costs*, where the goal is to balance predictive performance against the cost of acquiring features. Further, the information required for an accurate prediction might differ between instances: the relevance of certain tests may only become apparent after observing the results of previous ones, with the appropriate sequence of tests differing between patients.

Active Feature Acquisition (AFA) models this problem as the sequential selection of features to acquire for each instance, balancing predictive performance against acquisition cost (Aronsson & Haghir Chehreghani, 2026; Guney et al., 2025; Norcliffe et al., 2025). AFA methods are often categorized into *myopic* and *non-myopic* approaches, depending on whether they consider the long-term consequences of acquisitions or not (Aronsson et al., 2026). However, the current AFA literature has largely assumed access to *complete training data* (Aronsson et al., 2026), which is often hard to come by in practice. For example, medical records often only contain the specific test results that were ordered for each patient; it is unrealistic to assume that every test was performed on every patient. In contrast, during medical checkups it may be possible to request any relevant test for a patient. This resulting mismatch between what is available during training and evaluation is a common challenge.

Motivated by this, we study **AFA** with **I**ncomplete **T**raining **D**ata (**AFA-ITD**), formalizing it and analyzing the effect incomplete training data has on the learning process. Our main contributions are the following.

- We formulate the **AFA-ITD** problem (Section 3.4) and characterize the difference between its training and evaluation MDPs (Section 4.1).

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.

- We analyze the effects of incomplete training data on the Bellman recursions and show that, under missing completely at random (MCAR), missingness can lower non-myopic action values even when one-step values are unchanged (Proposition 1).
- We define and distinguish three approaches to learning from incomplete training data: the *aliasing*, *filtering*, and *generative restoration* approaches, and characterize their data requirements (Theorem 1).
- We experimentally test our theory on a synthetic controlled study, and on eight common AFA datasets using six AFA baselines. In line with our theory, missingness mainly damages methods that exploit multi-step acquisitions, and generative restoration is often able to recover lost performance (Section 5).

2 Related Work

In this section, we review related work, highlighting the works closest to our problem setting.

Active Feature Acquisition. AFA methods are generally characterized along two orthogonal axes. The first axis is their *planning horizon*: myopic methods only plan for the immediate next feature according to some one-step acquisition criterion. Notable myopic AFA methods include EDDI (Ma et al., 2019), GDFS (Covert et al., 2023), and DIME (Gadgil et al., 2024). In contrast, non-myopic methods plan for the potential multi-step effect an acquisition can have, often using reinforcement learning methods or other longer-horizon supervision formulations (Dulac-Arnold et al., 2011; Shim et al., 2018; Janisch et al., 2019; Aronsson & Haghir Chehreghani, 2026). The second axis is whether they model the *acquisition dynamics*. Model-based methods estimate these conditional distributions and use them during training, whereas model-free methods train directly from experience and interaction. Hybrid methods combine both, learning from experience with an explicit acquisition model (Aronsson et al., 2026).

AFA from Incomplete Training Data. To the best of our knowledge, the AFA literature on incomplete training data remains very limited. Janisch et al. (2020) were the first to investigate the **AFA-ITD** problem setting. During training, they restrict feature-acquisition actions to available features and compare methods with mean and MICE imputation baselines. Although their approach is empirically effective, the resulting restricted Bellman problem is not formally analyzed. More recently, Kobayashi et al. (2026) studied AFA from incomplete data and characterized specific conditions under which a greedy conditional-mutual-information acquisition objective remains identifiable. von Kleist et al. (2023) instead investigated the evaluation of AFA policies from incomplete data using semi-offline estimators. Wendland et al. (2026) have also proposed miss-MDPs, a general class of POMDPs in which missingness remains part of the observation process at deployment. None of these works have formalized the **AFA-ITD** problem setting and characterized how incomplete training data affects the Bellman recursions.

Generative Models in AFA. Generative models are commonly used in AFA to approximate acquisition dynamics. Ma et al. (2019) introduced the Partial Variational Autoencoder (PVAE) architecture to estimate conditionals and to evaluate a myopic information-gain acquisition criterion. Zannone et al. (2019) later used the PVAE architecture to generate synthetic acquisition trajectories, primarily to improve data efficiency, while Li & Oliva (2021) learned a generative surrogate model to provide intermediate rewards and auxiliary information during learning. However, the benefit of generative modeling remains unclear in the literature. Schütz et al. (2026) find that the gap between model-free and model-based approaches is often smaller than previously reported on common AFA datasets using a fair benchmarking setting. In this work, we will show that using a generative model is a natural solution to incomplete training data and not only an ad-hoc extension for data efficiency.

3 Problem Formulation

In this section, we introduce and formalize the AFA optimization problem, its formulation as a (Partially Observable) Markov Decision Process ((PO)MDP), the standard taxonomy of missingness mechanisms, and formulate the AFA problem with incomplete training data.

3.1 Active Feature Acquisition (AFA)

Active Feature Acquisition (AFA) formulates an explicit trade-off mechanism between acquiring features at a cost and their (expected) predictive performance. Usually, this trade-off is formulated as

an optimization problem. We use the formulation of Aronsson et al. (2026) and only consider the *hard-budget* classification objective.

Let $x := (x_1, \ldots, x_d) \in \mathcal{X}$ be an instance with the corresponding label $y \in \mathcal{Y}$, distributed according to $(x, y) \sim p(x, y)$. For $S \subseteq [d] := \{1, \ldots, d\}$ we write x_S for the acquired values and $U := [d] \setminus S$ with x_U for the unacquired values. Taking action $a \in U$ reveals feature x_a and incurs cost c_a, and we define the accumulated cost as $c(S) := \sum_{i \in S} c_i$. Furthermore, a policy π maps the current state (x_S, S) to an unacquired feature $a \in U$ and a predictor f maps the final observation to a label prediction. Together, they define the following optimization problem:

$$\begin{aligned} &\min_{f,\pi} \mathbb{E}_{(x,y)} \mathbb{E}_{\pi}[\ell(f(x_{\pi[x]}, \pi[x]), y)] \\ &\text{subject to } c(\pi[x]) \leq b \quad \text{for all } x, \end{aligned} \tag{1}$$

where ℓ is a loss function, b is the *hard budget*, and $\pi[x] \subseteq [d]$ is the final set of acquired features for instance x under policy π. Throughout, we assume access to a well-trained predictor f, since learning the predictor is not the objective of interest in this work.

3.2 AFA as a (Partially Observable) Markov Decision Process

AFA can be modeled as a Partially Observable Markov Decision Process (POMDP) whose latent state contains the fully observed instance x and the label y, and whose observations are the features acquired so far x_S (Aronsson et al., 2026). Because the latent state does not change over time, the POMDP can be reduced to a fully observable MDP on the state $s = (x_S, S)$ (Dulac-Arnold et al., 2011). In this fully observable MDP, the acquisition dynamics can be written in terms of the individual feature probabilities

$$p(s' \mid s, a) = p(x_{S \cup \{a\}}, S \cup \{a\} \mid x_S, S, a) \tag{2}$$

$$= p(x_a \mid x_S, S) \tag{3}$$

where $s' := (x_{S \cup \{a\}}, S \cup \{a\})$. The quality of a state s and an acquisition a is captured by truncated (action-)value functions, for which the following Bellman recursions (Sutton & Barto, 2018) hold:

$$\begin{aligned} V_0(s) &= \mathbb{E}_{y|s}[-\ell(f(s), y)], \\ Q_k(s, a) &= \mathbb{E}_{x_a|s}[V_{k-1}(s')], \\ V_k(s) &= \max\left\{ V_0(s), \max_{a \in \mathcal{A}_b(S)} Q_k(s, a) \right\}. \end{aligned} \tag{4}$$

Here V_k is the optimal value when at most k further acquisitions are allowed and $\mathcal{A}_b(S) := \{a \in U : c(S) + c_a \leq b\}$. Finally, to measure the gap from an optimal policy one usually uses the *regret*. Let $\mathcal{L}(\pi) := \mathbb{E}_{(x,y)} \mathbb{E}_{\pi}[\ell(f(x_{\pi[x]}, \pi[x]), y)]$ be the expected loss, and let π^* be the optimal policy that minimizes $\mathcal{L}(\pi)$. The regret of π is then $\mathrm{Reg}(\pi) := \mathcal{L}(\pi) - \mathcal{L}(\pi^*)$.

3.3 Missingness Mechanisms

Let $m \in \{0, 1\}^d$ be a binary missingness mask of dimension d, where $m_i = 1$ indicates that feature i is missing from the training instance. The missingness mask is drawn once per instance from $p(m \mid x, y)$ and remains fixed for that instance. We write $M := \{i : m_i = 1\}$ and $\bar{M} := \{i : m_i = 0\}$ for the missing and available indices, with the corresponding feature values x_M and $x_{\bar{M}}$, respectively. Classically, Rubin's taxonomy (Rubin, 1976; Little & Rubin, 2002) classifies mechanisms as

- **M**issing **C**ompletely **A**t **R**andom (MCAR): the missingness is independent of the instance, $p(m \mid x, y) = p(m)$.
- **M**issing **A**t **R**andom (MAR): the missingness may depend on the available features, but not the missing features, $p(m \mid x, y) = p(m \mid x_{\bar{M}})$.
- **M**issing **N**ot **A**t **R**andom (MNAR): the missingness may depend on the missing feature values themselves, either through a feature's own value (self-masking MNAR) or through other unavailable features (e.g., logistic MNAR).

3.4 AFA WITH INCOMPLETE TRAINING DATA (**AFA-ITD**)

In the **AFA-ITD** problem setting, missing features cannot be part of the *training action space*: for a training instance with missingness mask m, we define the training action space as $\mathcal{A}_b^{\text{train}}(S, m) := \{a \in U : m_a = 0,\ c(S) + c_a \leq b\}$. In contrast, the evaluation action space does not have any missing features and is defined as $\mathcal{A}_b^{\text{eval}}(S) := \{a \in U : c(S) + c_a \leq b\}$. Consequently, $\mathcal{A}_b^{\text{train}}(S, m) \subseteq \mathcal{A}_b^{\text{eval}}(S)$. Thus, the training Bellman recursion can be written as

$$\begin{aligned} V_0^{\text{train}}(s, m) &= \mathbb{E}_{y|s,m}[-\ell(f(s), y)], \\ Q_k^{\text{train}}(s, m, a) &= \mathbb{E}_{x_a|s,m}\left[V_{k-1}^{\text{train}}(s', m)\right], \\ V_k^{\text{train}}(s, m) &= \max\left\{V_0^{\text{train}}(s, m), \max_{a \in \mathcal{A}_b^{\text{train}}(S,m)} Q_k^{\text{train}}(s, m, a)\right\}. \end{aligned} \tag{5}$$

These recursions hold under MCAR because the missingness mask m is independent of (x, y), leaving the conditionals unchanged. By contrast, the evaluation Bellman equations remain unchanged and are defined as in Equation (4).

4 THEORETICAL ANALYSIS

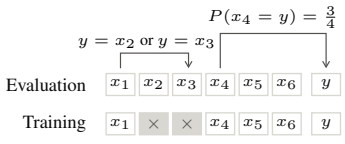
In this section, we analyze the effect incomplete training data can have on the Bellman recursions between training and evaluation. We then analyze the *aliasing* approach and how omitting the missingness mask from the state affects the learning process. Afterwards, we consider the *filtering* approach, through state augmentation with the missingness mask m, and the *generative restoration* approach. Finally, we theoretically compare their data requirements. All proofs can be found in Appendix A.

4.1 HOW THE (ACTION-)VALUE FUNCTIONS ARE AFFECTED BY INCOMPLETE TRAINING DATA

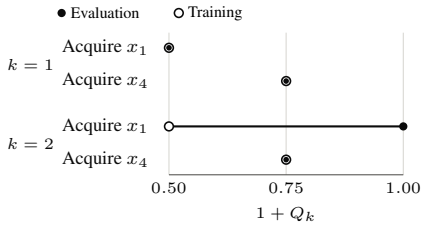
Inspecting Equation (5) and Equation (4) side by side, we see that the only difference between them is their respective action spaces. To better understand what effect this can have, we introduce a simple non-myopic AFA problem with binary features.

Example 1 (The Shortcut Problem)**.** Consider $d \geq 2b$ binary features $x_i \in \{0, 1\}$ with a hard budget of $b \geq 2$. The features consist of the *context* feature x_1, the two *branch* blocks of features $B_1 = \{2, \ldots, b\}, B_2 = \{b+1, \ldots, 2b-1\}$, with *block length* $b-1$, and the *shortcut* feature x_{2b}. The corresponding label is defined as $y = \bigoplus_{j \in B_{x_1+1}} x_j$, where $\oplus$ denotes the XOR operation. The shortcut feature x_{2b} is equal to the label y with probability $3/4$ but has a cost of b, while all other features have a cost of one. The remaining features $x_{2b+1}, \ldots, x_d$ are independent uniform *noise*.

Example 1 serves as a simple non-myopic AFA problem. The optimal policy is to always first acquire the context feature x_1, then acquire all the features in the corresponding branch block. This yields a total cost of b and an accuracy of 1. In contrast, the shortcut also costs b but only attains an accuracy of $3/4$.



(a) A realization of an instance of the shortcut problem where the branch features x_2 and x_3 are missing.



(b) The expected accuracy $(1 + Q_k)$ for the corresponding instance in Figure 1a.

Figure 1: The shortcut problem for a fixed missingness mask m where the branch features x_2 and x_3 are missing **(a)**, and the unchanged one-step and differing two-step $1 + Q_k$ optimal values for both training and evaluation **(b)**.

Figure 1a illustrates the problem for $b = 2$ and $d = 6$ for one instance where the two branch features x_2, x_3 are missing. We now analyze the effect of the mismatch between training and evaluation in Figure 1a on the (action-)value functions. For our analysis, we use the zero-one loss function $\ell(f(s), y) = \mathbf{1}\{f(s) \neq y\}$ and report the action values as $1 + Q(\dots)$, corresponding to expected accuracy under zero-one loss, for easier interpretation.

For a myopic policy, the context feature x_1 has no immediate predictive value and the best prediction after acquiring it is a guess. In contrast, the shortcut feature is preferable for myopic methods because it has immediate predictive value, predicting the label with probability $\frac{3}{4}$. Thus, we have

$$Q_1^{\text{train}}(s_0, m, a = 1) = Q_1^{\text{eval}}(s_0, a = 1) = \frac{1}{2},$$
$$Q_1^{\text{train}}(s_0, m, a = 4) = Q_1^{\text{eval}}(s_0, a = 4) = \frac{3}{4},$$

where $s_0 \coloneqq (x_\emptyset, \emptyset)$ is the initial state. In contrast, a non-myopic policy can plan for the full acquisition of both the context feature and the corresponding branch block features. However, in Figure 1a neither x_2 nor x_3 is available, which means that acquiring the context feature x_1 will yield poorer performance than the shortcut feature x_4, leaving x_1 as good as a guess. Consequently,

$$Q_2^{\text{train}}(s_0, m, a = 1) = \frac{1}{2} \neq Q_2^{\text{eval}}(s_0, a = 1) = 1,$$
$$Q_2^{\text{train}}(s_0, m, a = 4) = Q_2^{\text{eval}}(s_0, a = 4) = \frac{3}{4}.$$

Figure 1 illustrates that the restricted training action space can impact the learned action values, and the following proposition states how this holds more generally for an arbitrary planning horizon.

Proposition 1 ((Action-)values under the restricted action space)**.** *Assume that the missingness mask m is independent of (x, y) (MCAR). Then, for every state $s = (x_S, S)$ reachable under mask m and every $k \geq 0$,*

$$V_k^{\text{train}}(s, m) \leq V_k^{\text{eval}}(s),$$

with equality when $k = 0$. For every $a \in \mathcal{A}_b^{\text{train}}(S, m)$ and $k \geq 1$,

$$Q_k^{\text{train}}(s, m, a) \leq Q_k^{\text{eval}}(s, a),$$

with equality when $k = 1$.

Thus, we are left with a potential gap in the (action-)value functions that can impact performance. Hereafter, we seek to answer the following two questions: (i) what happens if we still try to learn the (action-)value functions from finite training data with varying missingness masks? and (ii) what are some possible mitigation methods that ensure we learn the correct evaluation problem?

4.2 Aliasing, Filtering, and Restoration

Proposition 1 does not guarantee that the restricted action space results in the wrong (action-)value functions, only that it is possible. However, it is clear that if we omit the missingness mask m from our state representation, we will suffer from an *aliasing* issue.

Definition 1 (Tabular Aliasing)**.** The tabular *aliasing* approach learns one (action-)value $Q_k^{\text{alias}}(s, a)$, $V_k^{\text{alias}}(s)$, or a policy $\pi^{\text{alias}}(a \mid s)$, omitting the missingness mask m entirely.

Across instances with different missingness masks m and corresponding action spaces $\mathcal{A}_b^{\text{train}}(S, m)$, the aliasing approach will assign different values to the *same* state-action pair (s, a), pooling them. We can observe this issue in Figure 1. Conditional on the context feature x_1 being observed, the relevant branch feature x_2 or x_3 is available with probability $1 - p$. The corresponding action-value for acquiring the context feature x_1 is then

$$Q_2^{\text{alias}}(s_0, a = 1) = (1 - p) \cdot 1 + p \cdot \frac{1}{2} = 1 - \frac{p}{2}. \tag{6}$$

The aliasing approach will therefore prefer the shortcut feature x_4 whenever $p > 1/2$, even though the optimal evaluation policy should always acquire the context feature x_1. This error remains even with unlimited training data.

In contrast, if we augment the state representation with the missingness mask m, we can learn separate values for different masks.

Definition 2 (Tabular Filtering)**.** The tabular *filtering* approach augments the state with the missingness mask (s, m) when learning $Q_k^{\text{filter}}(s, m, a)$, $V_k^{\text{filter}}(s, m)$, or a policy $\pi^{\text{filter}}(a \mid s, m)$.

During evaluation, the filtering approach will only use the learned values for the missingness mask $m = 0$, meaning that only *complete instances* are used for the values. Thus, provided that complete instances occur with positive probability, the filtering approach will be able to fully recover the evaluation Bellman problem with unlimited data. However, this approach can be statistically inefficient: under independent MCAR, a training instance is complete with probability $(1-p)^d$, scaling exponentially with the dimension d. Consequently, only (an expected) $n(1-p)^d$ of the n training instances contribute directly.

An alternative approach is to restore the unavailable acquisition dynamics $p(x_a \mid s, m)$ using a *generative model*, thereby recovering Equation (4) from Equation (5).

Definition 3 (Tabular Generative Restoration)**.** The tabular *generative restoration* approach learns acquisition dynamics $\hat{p}(x_a \mid s)$ and stopping values $\widehat{V}_0(s)$ from all instances with the features of s available, and uses them to restore unavailable transitions when learning $Q_k^{\text{restore}}(s, a)$, $V_k^{\text{restore}}(s)$, or a policy $\pi^{\text{restore}}(a \mid s)$.

Note that under independent MCAR, the empirical acquisition dynamics $\hat{p}(x_a \mid s)$ will be consistent because the missingness is independent of the corresponding feature and label values, i.e., $p(m \mid x, y) = p(m)$. Thus, if all features are observable with positive probability, the restoration approach can fully recover the evaluation conditionals with unlimited data, solving the exact evaluation problem (Equation (4)).

4.3 Usable Instances and Data Requirements for Filtering and Restoration

Both the filtering and the generative restoration approaches solve the correct evaluation Bellman recursions under MCAR and unlimited data. However, with a finite amount of training data, the approaches will have different numbers of *usable* training instances.

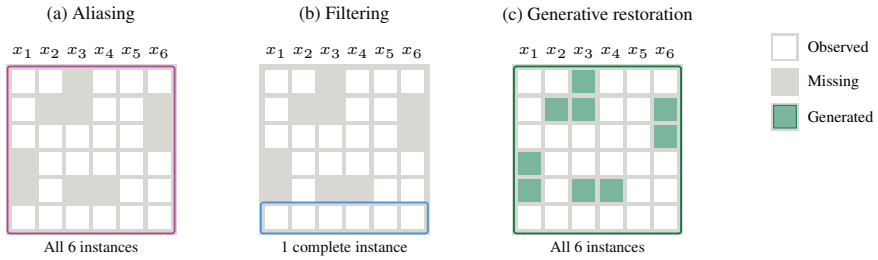


Figure 2: The *usable* training instances for each of the approaches, on Example 1 under MCAR with $p = 0.3$.

Figure 2 illustrates the difference in usable training instances for all three approaches. Both the aliasing and generative restoration approaches, the latter after restoring all missing features, can use all instances for learning, while filtering is only able to use complete instances. In the tabular setting, each approach can be viewed as solving a distinct Bellman problem: aliasing solves the *restricted* Bellman recursions (Equation (5)), generative restoration solves the approximate evaluation Bellman recursions, and filtering solves the exact evaluation Bellman recursions (Equation (4)). Since generative restoration and filtering solve the correct Bellman recursions, we are interested in their data requirements to achieve a certain regret with high probability. The following theorem quantifies this.

Theorem 1 (Data requirements under MCAR)**.** *Consider d discrete features $x_a \in \mathcal{X}_a$ with $|\mathcal{X}_a| \leq K$ for an integer $K \geq 2$, and costs $c_a \geq 1$ under the integer hard budget $1 \leq b \leq d$. Let f be a fixed predictor, independent of the n i.i.d. training instances, with bounded loss $\ell \in [0, \ell_{\max}]$. Assume MCAR, with each feature independently missing with probability $p < 1$. Let π^{filter} and π^{restore} be the exact corresponding policies obtained by their respective Bellman recursions. For every $0 < \varepsilon \leq \ell_{\max}$ and $\delta \in (0, 1)$, filtering satisfies $\text{Reg}(\pi^{\text{filter}}) \leq \varepsilon$ with probability at least $1 - \delta$*

whenever

$$n \geq C \frac{\ell_{\max}^2 K^b}{\varepsilon^2 (1-p)^d} \left(b \log(Kd) + K + \log \frac{2}{\delta} \right).$$

In contrast, generative restoration satisfies $\mathrm{Reg}(\pi^{\mathrm{restore}}) \leq \varepsilon$ *with probability at least* $1-\delta$ *whenever*

$$n \geq C \frac{\ell_{\max}^2 K^b}{\varepsilon^2 (1-p)^b} \left(b \log(Kd) + K + \log \frac{2}{\delta} \right),$$

where $C > 0$ *is a numerical constant independent of all of the above parameters.*

Thus, the filtering approach has an exponential dependency on the dimension d, whereas the generative restoration approach has an exponential dependency on the acquisition budget b. The dependency on d is necessary for filtering (Proposition 2).

5 Experiments

In this section, we solve and evaluate the synthetic controlled study of Example 1 to isolate the dimension-budget dependency in Theorem 1. Then, we benchmark *restricted training* (training in which acquisitions are restricted to the features available in each instance) against generative restoration on eight common AFA datasets with synthetic missingness, and connect the results to the theory.

5.1 Synthetic Controlled Study (Example 1)

To test the three approaches and our theory, we solve Example 1 exactly. The problem is parameterized by both the dimension d and the acquisition budget b, so we sweep the dimension over $d \in \{6, 8, 10\}$ with a fixed acquisition budget of $b = 2$ and sweep the budget over $b \in \{2, 3, 4\}$ with a fixed dimension of $d = 10$, both under MCAR with missingness rate fixed at $p = 0.6$. The approaches first estimate the acquisition dynamics $\widehat{p}(x_a \mid s)$ and the expected terminal losses $\widehat{V}_0(s)$ from the usable training instances with a fixed Bayes predictor, using a pseudocount of $1/2$ (Appendix B.2). Then, they solve their respective Bellman problem exactly with dynamic programming using these estimates. Finally, they are evaluated against the optimal oracle policy in terms of evaluation regret.

Figure 3 illustrates the myopic policy $\widehat{Q}_1$ along with full-horizon planning $\widehat{Q}_b$ on the same instances over 150 independent replicates, each with a freshly sampled training set and missingness masks, of the three approaches for Example 1. As expected from the one-step values in Section 4.1, the myopic policy prefers the shortcut and only achieves a regret of $1/4$ across all configurations. By contrast, the full-horizon filtering and restoration policies achieve zero regret given enough training data. In the dimension sweep, the filtering approach becomes more data-intensive as d increases, whereas restoration converges at the same training size across the tested dimensions. In contrast, in the budget sweep, increasing b lengthens the optimal acquisition route and consequently increases restoration's data requirements at fixed d, reflecting what is described in Theorem 1 in both sweeps. Finally, the aliasing approach converges to the shortcut and a regret of $1/4$, since the relevant branch block is available with probability $(1-p)^{b-1} < 1/2$ (Equation (6)).

5.2 AFA Methods under **AFA-ITD**

Evaluation Setup and Protocol. We use the AFA benchmark framework of Schütz et al. (2026) and adapt it to our **AFA-ITD** problem setting. We evaluate six AFA methods: the myopic DIME (Gadgil et al., 2024) and GDFS (Covert et al., 2023) methods, and the non-myopic AACO (Valancius et al., 2024), JAFA (Shim et al., 2018), OL (Kachuee et al., 2019), and ODIN (Zannone et al., 2019) methods. For each method, we train three variants: a complete-data version without any missingness, another with generative restoration applied to the training data beforehand, and a restricted training version with acquisitions restricted to available features. The non-myopic RL-based methods JAFA, OL, and ODIN are each represented by two separate methods, depending on whether they include the missingness mask in the state (e.g., OL (s, m)) or not (e.g., OL (s)), corresponding to the filtering and aliasing approaches, respectively. For restoration, we fit a label-conditioned Partial Variational Autoencoder (PVAE) (Ma et al., 2019; Zannone et al., 2019) on incomplete training data and replace only missing values with a joint sample conditioned on the observed features and label.

We use the benchmark's synthetic CUBE (Zannone et al., 2019), CUBE-NM (Schütz et al., 2026), and CUBE with non-uniform costs (CUBE-NUC) datasets. For real-world datasets, we

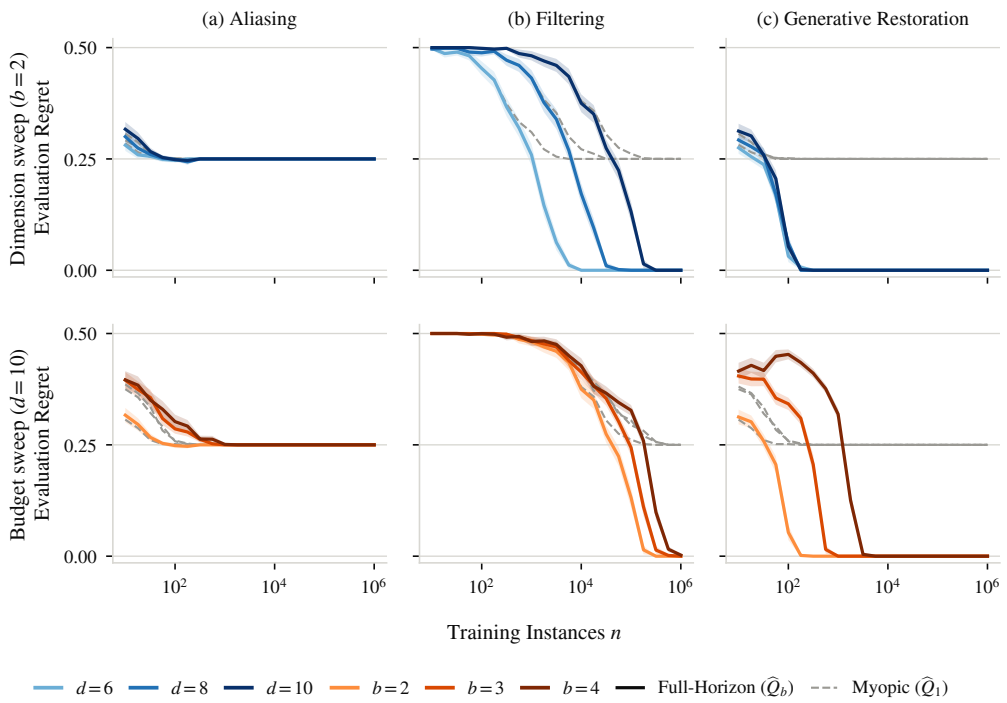


Figure 3: Evaluation regret of the myopic $\widehat{Q}_1$ (gray, dashed) and full planning-horizon $\widehat{Q}_b$ (solid) methods, across the three learning approaches swept over the dimension d (top row) and the acquisition budget b (bottom row), under MCAR at $p = 0.6$, with 95% confidence bands.

use ACTG175 (Hammer et al., 1996), Diabetes (Kachuee et al., 2019), NHANES mortality (NHANES) (Erion et al., 2022), MiniBooNE (Roe, 2005), and Heart Disease (Janosi et al., 1989). We synthetically induce MCAR, MAR, logistic MNAR, and self-masking MNAR at rates $p \in \{0.3, 0.5, 0.7\}$. We use five dataset splits, pairing restricted and restored treatments within each split. The predictor is trained before masking and held fixed across treatments, isolating the effect on acquisition. Missingness masks remain fixed across epochs and acquisition episodes; evaluation instances are fully observed. We report mean accuracy on the synthetic datasets and mean macro-F1 on the real-world datasets, at each dataset's largest predefined acquisition budget under Equation (1). RL training budgets are selected by convergence on complete data and held fixed across treatments. Exact budgets, optimization settings, missingness mechanisms, and method adaptations are given in Appendix B.1.

Figure 4 illustrates our primary findings under MCAR at $p = 0.7$. We only report results for the highest missingness rate as we observe the same patterns, but weaker, at the other missingness rates. The figure orders both methods and datasets according to their mean *missingness damage*: the difference in performance between the complete-data version (vertical line) and the restricted version (hollow circle). First, we note that the myopic method DIME is the least damaged, and the three most damaged methods are all non-myopic, consistent with Proposition 1. Second, restoration (filled circle) recovers part of the damage across a variety of methods and datasets, indicating that the advantage that Theorem 1 predicted in the tabular setting can hold true for methods using function approximation on real-world datasets. However, restoration is far from a "free lunch": in some configurations, restoration performs worse than restricted training. This is because missingness can remove redundant or noisy features, making the problem easier; restoring all features then returns the method to the harder original problem. Among the datasets, CUBE-NM is the most affected, aligning with its non-myopic structure (Schütz et al., 2026). In contrast, we note that the two least affected datasets are the real-world MiniBooNE and NHANES mortality datasets. We further investigate the structure of all eight datasets and its relation to missingness damage in Appendices B.3 and B.4,

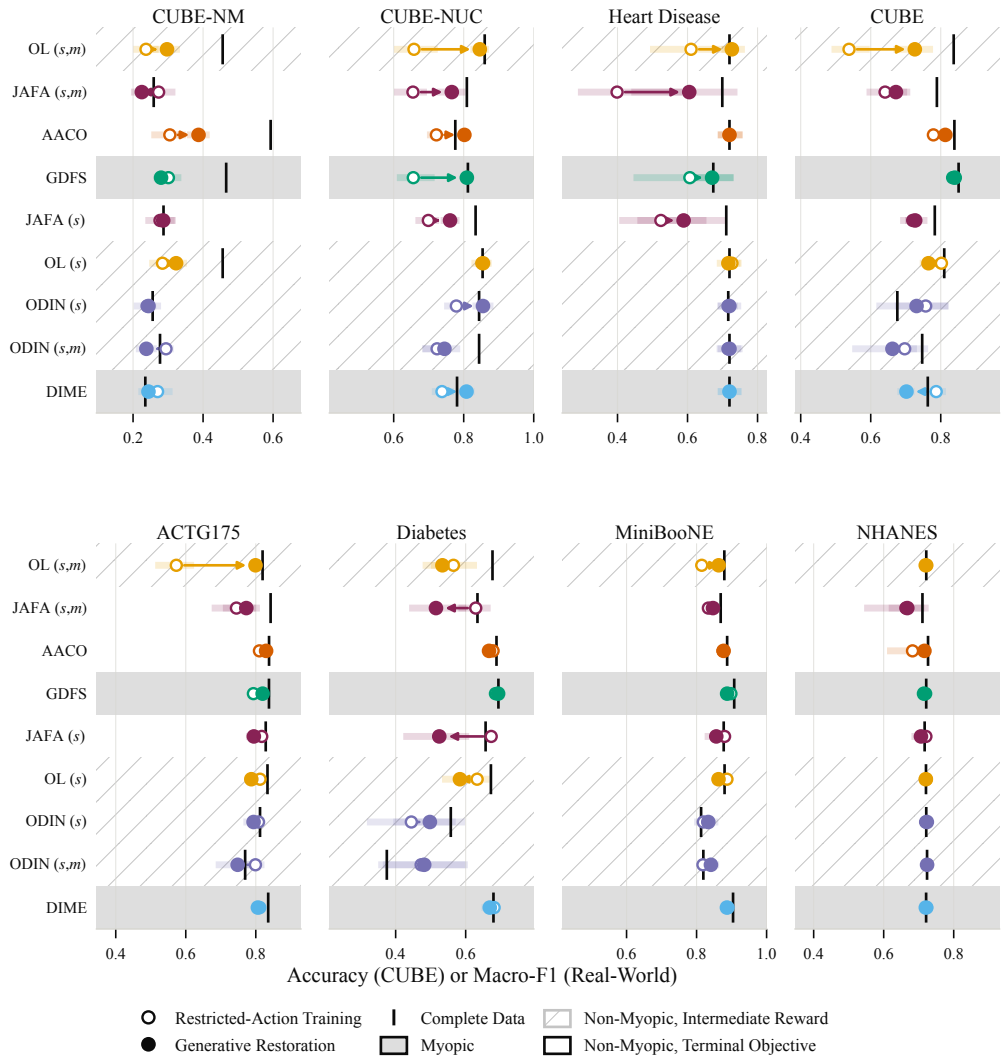


Figure 4: Restricted training vs. generative restoration under MCAR at $p = 0.7$. Arrows point from restricted training to generative restoration, and bars are 95% bootstrap intervals over five splits. Methods and datasets are ordered by descending mean missingness damage. Gray rows denote myopic methods (DIME, GDFS), hatched rows denote non-myopic methods with intermediate predictive rewards (OL, ODIN), and white rows denote non-myopic acquisition with a terminal prediction objective (AACO, JAFA).

report the computational cost of restoration in Appendix B.5, and repeat the comparison under MAR and MNAR in Appendix B.6.

6 Conclusion

We have investigated the **AFA-ITD** problem setting and found that: (i) in line with our theory (Proposition 1), incomplete training data mainly damages methods that exploit multi-step acquisitions; (ii) in the tabular setting, the data requirement of filtering grows exponentially with the dimension, whereas that of generative restoration grows only with the acquisition budget (Theorem 1); (iii) generative restoration can in practice reduce the resulting performance gap for non-myopic methods. We hope that our work opens up the development of AFA methods that leverage the theoretical framework of **AFA-ITD**. For example, by leveraging AFA learning and generative learning jointly, it might be possible to restore only the informative features and leave the uninformative ones blocked.

AI USE STATEMENT

In this work, we have used generative AI tools for: developing theoretical models or conceptual frameworks and providing feedback on research methodology or experiments. We have not used generative AI tools for: implementing the methods, supporting qualitative and thematic data analysis, assisting in the writing of proofs, proposing or refining hypotheses, providing critical ingredients for proving mathematical claims, formulating mathematical claims, generating synthetic datasets, interpreting results. Cleaning and reformatting datasets and assisting with translation are not applicable to this work. Additionally, we have used generative AI tools for: creating or modifying scientific figures or images, creating or editing software code, creating artifacts, brainstorming, searching for information, and identifying relevant literature. We have reviewed all AI-assisted work. We have checked the LLM-generated ideas, formulations, and statements to be correct or corrected them. All the code and content have been verified and tested for correctness by two authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All assumptions of our theoretical results are stated with the results, and complete proofs are given in Appendix A. The synthetic controlled study of Section 5.1 is solved exactly by dynamic programming, and its setup is described in Section 5.1. All datasets are public and are used through the `AFABench` framework (Schütz et al., 2026); the induced missingness mechanisms, restoration model, method adaptations, training budgets, and evaluation protocol are detailed in Appendix B.1. Source code, including the configurations and workflow used to produce every figure and table, is available at an anonymized repository: https://anonymous.4open.science/r/47825.

REFERENCES

Linus Aronsson and Morteza Haghir Chehreghani. Non-myopic active feature acquisition via pathwise policy gradients, 2026. URL https://arxiv.org/abs/2605.05511.

Linus Aronsson, Arman Rahbar, and Morteza Haghir Chehreghani. A survey on active feature acquisition strategies. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, 2026. URL https://arxiv.org/abs/2502.11067.

Ian Connick Covert, Wei Qiu, Mingyu Lu, Na Yoon Kim, Nathan J. White, and Su-In Lee. Learning to maximize mutual information for dynamic feature selection. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6424–6447. PMLR, 2023. URL https://proceedings.mlr.press/v202/covert23a.html.

Gabriel Dulac-Arnold, Ludovic Denoyer, Philippe Preux, and Patrick Gallinari. Datum-wise classification: A sequential approach to sparsity. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, volume 6911 of *Lecture Notes in Computer Science*, pp. 375–390. Springer, 2011. doi: 10.1007/978-3-642-23780-5_34.

Gabriel Erion, Joseph D. Janizek, Carly Hudelson, Richard B. Utarnachitt, Andrew M. McCoy, Michael R. Sayre, Nathan J. White, and Su-In Lee. A cost-aware framework for the development of AI models for healthcare applications. *Nature Biomedical Engineering*, 2022.

Soham Gadgil, Ian Connick Covert, and Su-In Lee. Estimating conditional mutual information for dynamic feature selection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oju2Qu9jvn.

G. Anthony Gorry and G. Octo Barnett. Experience with a model of sequential diagnosis. *Computers and Biomedical Research*, 1(5):490–507, 1968. doi: 10.1016/0010-4809(68)90016-5.

Osman Berke Guney, Ketan Suhaas Saichandran, Karim Elzokm, Ziming Zhang, and Vijaya B Kolachalama. Active feature acquisition via explainability-driven ranking. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20748–20765. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/guney25a.html.

Scott M. Hammer, David A. Katzenstein, Michael D. Hughes, Holly Gundacker, Robert T. Schooley, Richard H. Haubrich, W. Keith Henry, Michael M. Lederman, John P. Phair, Manette Niu, Martin S. Hirsch, and Thomas C. Merigan. AIDS clinical trials group study 175. UCI Machine Learning Repository, 1996. URL https://doi.org/10.24432/C5ZG8F.

Jaromír Janisch, Tomáš Pevný, and Viliam Lisý. Classification with costly features using deep reinforcement learning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33013959. URL https://doi.org/10.1609/aaai.v33i01.33013959.

Jaromír Janisch, Tomáš Pevný, and Viliam Lisý. Classification with costly features as a sequential decision-making problem. *Machine Learning*, 109(8):1587–1615, 2020. doi: 10.1007/s10994-020-05874-8. URL https://doi.org/10.1007/s10994-020-05874-8.

Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. Heart disease. UCI Machine Learning Repository, 1989.

Arjan J. P. Jeckmans, Michael Beye, Zekeriya Erkin, Pieter Hartel, Reginald L. Lagendijk, and Qiang Tang. Privacy in recommender systems. In *Social Media Retrieval*. Springer, 2013.

Mohammad Kachuee, Orpaz Goldstein, Kimmo Kärkkäinen, Sajad Darabi, and Majid Sarrafzadeh. Opportunistic learning: Budgeted cost-sensitive learning from data streams. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1eOHo09KX.

Yuta Kobayashi, Zilin Jing, Jiayu Yao, Hongseok Namkoong, and Shalmali Joshi. Learning-to-measure: In-context active feature acquisition. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ZumBSRsRbE.

Mikko Lauri, David Hsu, and Joni Pajarinen. Partially observable markov decision processes in robotics: A survey. *IEEE Transactions on Robotics*, 39(1):21–40, 2023.

Yang Li and Junier Oliva. Active feature acquisition with generative surrogate models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6450–6459. PMLR, 2021.

R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley Series in Probability and Mathematical Statistics. Wiley, 2002. ISBN 9780471183860.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez-Lobato, Sebastian Nowozin, and Cheng Zhang. EDDI: Efficient dynamic discovery of high-value information with partial VAE. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4234–4243. PMLR, 2019. URL https://proceedings.mlr.press/v97/ma19c.html.

Alexander Norcliffe, Changhee Lee, Fergus Imrie, Mihaela van der Schaar, and Pietro Lio. Stochastic encodings for active feature acquisition, 2025. URL https://arxiv.org/abs/2508.01957.

Byron Roe. MiniBooNE particle identification. UCI Machine Learning Repository, 2005.

Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. doi: 10.2307/2335739.

Valter Schütz, Han Wu, Reza Rezvan, Linus Aronsson, and Morteza Haghir Chehreghani. AFABench: A generic framework for benchmarking active feature acquisition. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. URL https://doi.org/10.1145/3770855.3817493.

Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Michael Valancius, Maxwell Lennon, and Junier Oliva. Acquisition conditioned oracle for nongreedy active feature acquisition. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48957–48975. PMLR, 2024. URL https://proceedings.mlr.press/v235/valancius24a.html.

Henrik von Kleist, Alireza Zamanian, Ilya Shpitser, and Narges Ahmidi. Evaluation of active feature acquisition methods for static feature settings. *arXiv preprint arXiv:2312.03619*, 2023. URL https://arxiv.org/abs/2312.03619.

Joshua Wendland, Markel Zubia, Roman Andriushchenko, Maris F. L. Galesloot, Milan Ceska, Henrik von Kleist, Thiago D. Simão, Maximilian Weininger, and Nils Jansen. Missingness-MDPs: Bridging the theory of missing data and POMDP s, 2026. URL https://arxiv.org/abs/2605.12262.

Sara Zannone, Jose Miguel Hernandez Lobato, Cheng Zhang, and Konstantina Palla. ODIN: Optimal discovery of high-value information using model-based deep reinforcement learning. In *Real-world Sequential Decision Making Workshop, ICML*, June 2019. URL https://www.microsoft.com/en-us/research/publication/odin-optimal-discovery-of-high-value-information-using-model-based-deep-reinforcement-learning/.

A PROOFS

A.1 PROOF OF PROPOSITION 1

Proof. The proof proceeds by induction on the number of remaining acquisitions k. Under MCAR, we know that $p(y \mid s, m) = p(y \mid s)$, so the stopping values must satisfy

$$\begin{aligned} V_0^{\text{train}}(s, m) &= \mathbb{E}_{y|s,m}[-\ell(f(s), y)] \\ &= \mathbb{E}_{y|s}[-\ell(f(s), y)] = V_0^{\text{eval}}(s). \end{aligned}$$

This establishes the case for $k = 0$. For the induction step, we assume that $V_{k-1}^{\text{train}}(s', m) \leq V_{k-1}^{\text{eval}}(s')$ at every successor state s' reachable under mask m. Consequently, for all training actions $a \in \mathcal{A}_b^{\text{train}}(S, m)$, we have

$$\begin{aligned} Q_k^{\text{train}}(s, m, a) &= \mathbb{E}_{x_a|s,m}\left[V_{k-1}^{\text{train}}(s', m)\right] \\ &\leq \mathbb{E}_{x_a|s}\left[V_{k-1}^{\text{eval}}(s')\right] = Q_k^{\text{eval}}(s, a), \end{aligned}$$

where the inequality step uses the induction hypothesis and MCAR. For the value function, we have

$$\begin{aligned} V_k^{\text{train}}(s, m) &= \max\left\{V_0^{\text{train}}(s, m), \max_{a \in \mathcal{A}_b^{\text{train}}(S,m)} Q_k^{\text{train}}(s, m, a)\right\} \\ &\leq \max\left\{V_0^{\text{eval}}(s), \max_{a \in \mathcal{A}_b^{\text{train}}(S,m)} Q_k^{\text{eval}}(s, a)\right\} \\ &\leq \max\left\{V_0^{\text{eval}}(s), \max_{a \in \mathcal{A}_b^{\text{eval}}(S)} Q_k^{\text{eval}}(s, a)\right\} = V_k^{\text{eval}}(s), \end{aligned}$$

where the last inequality follows from the fact that $\mathcal{A}_b^{\text{train}}(S, m) \subseteq \mathcal{A}_b^{\text{eval}}(S)$.

For $k = 1$, we further have

$$Q_1^{\text{train}}(s, m, a) = \mathbb{E}_{x_a|s,m}[V_0^{\text{train}}(s', m)] = \mathbb{E}_{x_a|s}[V_0^{\text{eval}}(s')] = Q_1^{\text{eval}}(s, a),$$

which concludes the proof. □

A.2 PROOF OF THEOREM 1

Proof. Both the filtering and restoration approaches need to estimate two things: the expected stopping values $V_0(s)$ and the acquisition dynamics $p(x_a \mid s)$ for the Bellman recursion, but they differ in how many usable training instances contribute to these estimates. The proof will proceed in the following steps:

1. We count the number of usable training instances for both approaches (Section A.2.1).
2. We count the total number of estimates each approach needs (Section A.2.2).
3. We bound the error of each estimate type (Section A.2.3).
4. Finally, we bound the total evaluation regret in terms of errors of the individual estimates (Section A.2.4).

For the bounding steps, we make use of the following two inequalities:

Multiplicative Chernoff bound Let $X = \sum_{i=1}^{n} X_i$ be a sum of independent binary random variables with mean $\mu = \mathbb{E}[X]$. For a fixed $\gamma \in (0, 1)$ the multiplicative Chernoff bound is

$$P\left(X < (1 - \gamma)\mu\right) \leq \exp\left(-\frac{\gamma^2 \mu}{2}\right).$$

Hoeffding bound Let $\bar{Z} = \frac{1}{N}\sum_{i=1}^{N} Z_i$ be an average of N independent random variables, each taking values in an interval of length R. For any $t > 0$ the one-sided Hoeffding bound is

$$P\left(\bar{Z} - \mathbb{E}[\bar{Z}] \geq t\right) \leq \exp\left(-\frac{2Nt^2}{R^2}\right)$$

with the two-sided form

$$P\left(|\bar{Z} - \mathbb{E}[\bar{Z}]| \geq t\right) \leq 2\exp\left(-\frac{2Nt^2}{R^2}\right).$$

A.2.1 Count the Number of Usable Training Instances

Let N_s^{gen} and $N_{s,a}^{\text{gen}}$ denote the number of training instances that the restoration approach can use to estimate $V_0(s)$ and $p(x_a \mid s)$, respectively. N_s^{gen} counts the number of instances that match x_S and have every feature in S available, while $N_{s,a}^{\text{gen}}$ counts instances that additionally have feature a available,

$$N_s^{\text{gen}} \coloneqq \sum_{i=1}^{n} \mathbf{1}\{x_S^{(i)} = x_S,\ S \subseteq \bar{M}^{(i)}\},$$

$$N_{s,a}^{\text{gen}} \coloneqq \sum_{i=1}^{n} \mathbf{1}\{x_S^{(i)} = x_S,\ S \cup \{a\} \subseteq \bar{M}^{(i)}\}.$$

Under MCAR, the feature values and labels are independent of availability, thus

$$\mathbb{E}[N_s^{\text{gen}}] = nP_s(1-p)^{|S|},$$
$$\mathbb{E}[N_{s,a}^{\text{gen}}] = nP_s(1-p)^{|S|+1},$$

where P_s is the probability that a random data instance agrees with the observed values x_S. Every acquisition costs at least one, which means that $|S| \leq b$ and an available next acquisition satisfies $|S| + 1 \leq b$. In contrast, the filtering approach can only use complete instances

$$\mathbb{E}[N_s^{\text{filt}}] = \mathbb{E}[N_{s,a}^{\text{filt}}] = nP_s(1-p)^d.$$

For brevity, we write the common bound for both methods

$$\mathbb{E}[N_s] \geq nqP_s,\ \mathbb{E}[N_{s,a}] \geq nqP_s,$$

where

$$q = \begin{cases} (1-p)^d, & \text{for filtering,} \\ (1-p)^b, & \text{for restoration.} \end{cases}$$

If $N_s = 0$, we set $\widehat{V}_0(s)$ to a fixed value in $[-\ell_{\max}, 0]$, and if $N_{s,a} = 0$, we set $\widehat{p}(\cdot \mid s)$ to a fixed distribution on $\mathcal{X}_a$, e.g., the uniform distribution.

A.2.2 Count the Number of Estimates

With j acquired features, there are $\binom{d}{j}$ feature sets and at most K^j possible value assignments. Each state s has one stopping-value estimate $\widehat{V}_0(s)$ and at most d acquisition dynamics to estimate $\widehat{p}(\cdot \mid s)$. The total number of estimates is therefore at most

$$(d+1)\sum_{j=0}^{b}\binom{d}{j}K^j \leq (d+1)\sum_{j=0}^{b}(Kd)^j \leq 2(d+1)(Kd)^b \leq (Kd)^{b+2},$$

and we write $J \coloneqq (Kd)^{b+2}$ for an upper bound on the number of estimates needed.

A.2.3 Bound the Error of Each Estimate

Failure possibilities and confidence allocation. For each stopping-value estimate $\widehat{V}_0(s)$, failure can occur if (i) the number of usable instances is too small, or (ii) the estimate deviates too far in either direction. For each transition-distribution estimate $\widehat{p}(\cdot \mid s)$, failure can occur if (i) the number of usable instances is too small, or (ii) for some $A \subseteq \mathcal{X}_a$, the empirical probability $\widehat{p}(A \mid s)$ deviates too far from $p(A \mid s)$.

There are at most J estimates of both types combined. Hence, there are at most J count-failure events, $2J$ stopping-value deviation events, and $2^K J$ transition-subset deviation events. Since $K \geq 2$, their total number is at most

$$J(1 + 2 + 2^K) \leq 2^{K+1}J.$$

Thus, we allocate a failure probability of $\delta/(2^{K+1}J)$ to each event and define

$$\Lambda_J \coloneqq \log\frac{2^{K+1}J}{\delta} = (b+2)\log(Kd) + (K+1)\log 2 + \log\frac{1}{\delta}.$$

Thus, whenever an individual failure probability is bounded by $\exp(-A)$, requiring $A \geq \Lambda_J$ makes it at most $\delta/(2^{K+1}J)$. Finally, after deriving the individual bounds, a union bound controls all estimates simultaneously with probability at least $1 - \delta$.

Concentration of the sample counts. The count N_s is a sum of independent binary variables, so we apply the Chernoff bound at $\gamma = 1/2$ with $\mu = \mathbb{E}[N_s] \geq nqP_s$,

$$\begin{aligned} P\left(N_s < \frac{nqP_s}{2}\right) &\leq P\left(N_s < \frac{\mathbb{E}[N_s]}{2}\right) \\ &\leq \exp\left(-\frac{\mathbb{E}[N_s]}{8}\right) \\ &\leq \exp\left(-\frac{nqP_s}{8}\right). \end{aligned}$$

Hence, whenever $nqP_s \geq 8\Lambda_J$,

$$P\left(N_s < \frac{nqP_s}{2}\right) \leq e^{-\Lambda_J}.$$

Since $\mathbb{E}[N_{s,a}] \geq nqP_s$ as well, the same bound holds for $N_{s,a}$.

Concentration of the estimates. Now suppose that $nqP_s \geq 8\Lambda_J$. On the event $N_s \geq nqP_s/2$, the two-sided Hoeffding bound gives

$$P\left(|\widehat{V}_0(s) - V_0(s)| > t,\, N_s \geq \frac{nqP_s}{2}\right) \leq 2\exp\left(-\frac{nqP_s t^2}{\ell_{\max}^2}\right).$$

Choosing

$$t = \ell_{\max}\sqrt{\frac{\Lambda_J}{nqP_s}},$$

makes this probability at most $2e^{-\Lambda_J}$. For a transition estimate, we write

$$\Delta(x_a) \coloneqq \widehat{p}(x_a \mid s) - p(x_a \mid s)$$

and define its total-variation error by

$$\mathrm{TV}(\widehat{p}, p) \coloneqq \frac{1}{2}\sum_{x_a \in \mathcal{X}_a} |\Delta(x_a)|.$$

Since $\widehat{p}(\cdot \mid s)$ and $p(\cdot \mid s)$ both sum to one, $\sum_{x_a} \Delta(x_a) = 0$. Consequently, the total positive and negative deviations are equal, and

$$\mathrm{TV}(\widehat{p}, p) = \sum_{x_a:\, \Delta(x_a) > 0} \Delta(x_a) = \max_{A \subseteq \mathcal{X}_a} \big(\widehat{p}(A \mid s) - p(A \mid s)\big).$$

Thus, it is sufficient to control the empirical probability of every subset $A \subseteq \mathcal{X}_a$. Conditional on $N_{s,a} = N$, for any fixed A we have

$$\widehat{p}(A \mid s) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{x_a^{(i)} \in A\},$$

which is an average of independent binary variables with expectation $p(A \mid s)$. The one-sided Hoeffding bound and a union bound over the at most 2^K subsets of $\mathcal{X}_a$ therefore give

$$P\left(\mathrm{TV}(\widehat{p}, p) > t \,|\, N_{s,a} = N\right) \leq 2^K \exp(-2Nt^2).$$

Hence, on the event $N_{s,a} \geq nqP_s/2$,

$$P\left(\mathrm{TV}(\widehat{p}, p) > t,\, N_{s,a} \geq \frac{nqP_s}{2}\right) \leq 2^K \exp(-nqP_s t^2).$$

Taking $t = \sqrt{\Lambda_J/(nqP_s)}$ makes this probability at most $2^K e^{-\Lambda_J}$.

A uniform bound over all states. To express both kinds of deviation error (in $\widehat{V}$ and $\widehat{p}$) on the scale of the value function, let

$$e_s \coloneqq \max\left\{|\widehat{V}_0(s) - V_0(s)|, \ell_{\max} \max_{a \in \mathcal{A}_b^{\text{eval}}(S)} \mathrm{TV}(\widehat{p}(\cdot \mid s), p(\cdot \mid s))\right\}.$$

If no acquisition is legal, only the stopping-value error is present. The concentration bound above applies when $nqP_s \geq 8\Lambda_J$. When instead $nqP_s < 8\Lambda_J$, the sample counts may be small or zero, so we make no concentration claim. However, boundedness still gives $e_s \leq \ell_{\max}$. The two cases can therefore be combined as

$$e_s \leq \ell_{\max} \min\left\{1, \sqrt{\frac{8\Lambda_J}{nqP_s}}\right\}.$$

Indeed, when $nqP_s < 8\Lambda_J$ the minimum equals 1, while when $nqP_s \geq 8\Lambda_J$ this is only a constant-factor weakening of the concentration bound above. There are at most J estimates of both types combined. A union bound over their count failures, stopping-value deviations, and transition-distribution deviations is at most

$$J(1 + 2 + 2^K)e^{-\Lambda_J} \leq 2^{K+1} J e^{-\Lambda_J} = \delta.$$

Consequently, with probability at least $1 - \delta$, the bound on e_s holds at every state with $P_s > 0$ simultaneously.

The error budget in the notation of the theorem. It remains to replace Λ_J by the quantity $\Lambda \coloneqq b \log(Kd) + K + \log \frac{2}{\delta}$ used in the theorem. Since $b \geq 1$, $K \geq 2$, and $\delta < 1$, we have

$$(b+2)\log(Kd) \leq 3b\log(Kd),\ (K+1)\log 2 \leq 2K,\ \log\frac{1}{\delta} \leq \log\frac{2}{\delta}.$$

Thus $\Lambda_J \leq 3\Lambda$ and $8\Lambda_J \leq 24\Lambda$. Since $\min\{1, \cdot\}$ is nondecreasing, we obtain

$$e_s \leq \ell_{\max} \min\left\{1, \sqrt{\frac{24\Lambda}{nqP_s}}\right\}. \tag{7}$$

A.2.4 Bound the Evaluation Regret in Terms of Errors of the Individual Estimates

How local errors affect value estimates of the initial state. Fix a deterministic policy π. Let $V_k^\pi(s)$ be its true value with at most k acquisitions remaining, and let $\widehat{V}_k^\pi(s)$ be its value using the empirical estimates. Both values lie in $[-\ell_{\max}, 0]$, since the only reward is the negative loss when the policy stops. This also holds at states without training samples because their fallback estimates (Section A.2.1) lie in the same ranges. If the policy stops at s, its value error is $|\widehat{V}_0(s) - V_0(s)| \leq e_s$. Otherwise, let a be its chosen acquisition. The current value error can be decomposed in terms of transition probability errors and next-state value errors:

$$\begin{aligned}
\widehat{V}_k^\pi(s) - V_k^\pi(s) &= \sum_{x_a \in \mathcal{X}_a} \widehat{p}(x_a \mid s)\widehat{V}_{k-1}^\pi(s') - \sum_{x_a \in \mathcal{X}_a} p(x_a \mid s) V_{k-1}^\pi(s') \\
&= \sum_{x_a \in \mathcal{X}_a} \big(\widehat{p}(x_a \mid s) - p(x_a \mid s)\big)\widehat{V}_{k-1}^\pi(s') \\
&\quad + \sum_{x_a \in \mathcal{X}_a} p(x_a \mid s)\big(\widehat{V}_{k-1}^\pi(s') - V_{k-1}^\pi(s')\big).
\end{aligned}$$

Since the successor values lie in an interval of length at most $\ell_{\max}$, the definition of total variation gives

$$\left|\sum_{x_a \in \mathcal{X}_a} \big(\widehat{p}(x_a \mid s) - p(x_a \mid s)\big)\widehat{V}_{k-1}^\pi(s')\right| \leq \ell_{\max}\,\mathrm{TV}(\widehat{p}, p) \leq e_s.$$

Indeed, subtracting the midpoint of the successor-value range from every value leaves the sum unchanged, after which the bound follows from $\mathrm{TV}(\widehat{p}, p) = \frac{1}{2}\|\widehat{p} - p\|_1$. The second sum carries

the remaining value errors forward, weighted by the true probabilities. Taking absolute values consequently gives

$$|\widehat{V}_k^\pi(s) - V_k^\pi(s)| \le e_s + \sum_{x_a \in \mathcal{X}_a} p(x_a \mid s)|\widehat{V}_{k-1}^\pi(s') - V_{k-1}^\pi(s')|. \tag{8}$$

Starting at the initial state s_0 and unrolling the recursion in Equation (8),

$$|\widehat{V}_b^\pi(s_0) - V_b^\pi(s_0)| \le \sum_{t=0}^{b} \sum_{\substack{s \text{ reached after} \\ t \text{ acquisitions}}} r_s e_s, \tag{9}$$

where r_s is the probability that π reaches s.

Summing the errors over acquisitions. After t acquisitions, a deterministic policy has observed t feature values, each with at most K possible outcomes. Its acquisition tree therefore contains at most K^t states at that depth. Reaching a state $s = (x_S, S)$ requires an instance to match x_S, so $r_s \le P_s$. Furthermore, for any instance the policy visits at most one state at depth t, and hence $\sum_{s:\ |S|=t} r_s \le 1$. The sum may be smaller than one because some instances lead to earlier stopping. For the states at depth t, these observations give

$$\begin{aligned}\sum_{s:\ |S|=t} \frac{r_s}{\sqrt{P_s}} &\le \sum_{s:\ |S|=t} \sqrt{r_s} \\ &\le \sqrt{K^t \sum_{s:\ |S|=t} r_s} \le K^{t/2},\end{aligned}$$

where the last inequality uses Cauchy–Schwarz over at most K^t states. Multiplying Equation (7) by r_s and summing gives

$$\sum_{s:|S|=t} r_s e_s \le \ell_{\max}\sqrt{\frac{24\Lambda}{nq}} \sum_{s:|S|=t} \frac{r_s}{\sqrt{P_s}} \le \ell_{\max}\sqrt{\frac{24K^t\Lambda}{nq}}.$$

Substituting into Equation (9) and summing over depths yields

$$\begin{aligned}|\widehat{V}_b^\pi - V_b^\pi| &\le \ell_{\max}\sqrt{\frac{24\Lambda}{nq}} \sum_{t=0}^{b} K^{t/2} \\ &\le 4\sqrt{24}\,\ell_{\max}\sqrt{\frac{K^b\Lambda}{nq}},\end{aligned} \tag{10}$$

where

$$\sum_{t=0}^{b} K^{t/2} = \frac{K^{(b+1)/2} - 1}{\sqrt{K} - 1} \le 4K^{b/2},$$

for $K \ge 2$.

From value error to regret. To bound the value estimation error in Equation (10) by $\varepsilon/2$, it is sufficient to require

$$nq \ge C\frac{\ell_{\max}^2 K^b \Lambda}{\varepsilon^2},$$

where $C > 0$ is a numerical constant chosen large enough to absorb the fixed constants. Under this condition,

$$|\widehat{V}_b^\pi - V_b^\pi| \le \frac{4\sqrt{24}}{\sqrt{C}}\varepsilon.$$

Choosing, for example, $C \ge 1536$ ensures that

$$|\widehat{V}_b^\pi - V_b^\pi| \le \frac{\varepsilon}{2}.$$

Both finite Bellman problems admit deterministic optimal policies. Let π^* be optimal for evaluation and let π_j be the policy returned by the chosen empirical method. Empirical optimality gives $\widehat{V}_b^{\pi_j} \geq \widehat{V}_b^{\pi^*}$. The regret can be decomposed into a sum of value estimation errors:

$$\begin{aligned}\mathrm{Reg}(\pi_j) &= V_b^{\pi^*} - V_b^{\pi_j} \\ &= \left(V_b^{\pi^*} - \widehat{V}_b^{\pi^*}\right) + \left(\widehat{V}_b^{\pi^*} - \widehat{V}_b^{\pi_j}\right) + \left(\widehat{V}_b^{\pi_j} - V_b^{\pi_j}\right) \\ &\leq |V_b^{\pi^*} - \widehat{V}_b^{\pi^*}| + |\widehat{V}_b^{\pi_j} - V_b^{\pi_j}| \\ &\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.\end{aligned}$$

The middle term is nonpositive by empirical optimality, and the remaining two terms are each at most $\varepsilon/2$ because the value-error bound holds for every deterministic policy. Finally, substituting $q = (1-p)^d$ for filtering and $q = (1-p)^b$ for restoration gives the two stated sample-size bounds. □

A.3 A Lower Bound for Filtering

Proposition 2 (Filtering requires complete instances)**.** *Consider Example 1 with $d \geq 2b+1$ under the zero-one loss, where each feature is independently missing with probability p. Let filtering use fixed fallback estimates when a state has no usable instances (Section A.2.1) and a fixed tie-breaking rule. Then, for some relabeling of the unit-cost features, with probability at least $1 - n(1-p)^d$,*

$$\mathrm{Reg}(\pi^{\mathrm{filter}}) \geq \frac{1}{4}.$$

Consequently, for $\varepsilon < 1/4$, $\mathrm{Reg}(\pi^{\mathrm{filter}}) \leq \varepsilon$ with probability at least $1-\delta$ requires $n \geq (1-\delta)(1-p)^{-d}$.

Proof. A training instance is complete with probability $(1-p)^d$, so by a union bound, no training instance is complete with probability at least $1 - n(1-p)^d$. On this event, every estimate $\widehat{V}_0(s)$ and $\widehat{p}(\cdot \mid s)$ equals its fallback, so the empirical Bellman recursion returns a fixed policy π_0 that depends only on the costs, the fallbacks, and the tie-breaking rule. Relabeling the unit-cost features leaves all three unchanged, and hence also π_0. Since the optimal policy attains accuracy 1, the regret of a policy is one minus its accuracy, and we consider the first action of π_0.

- If π_0 stops immediately, nothing is observed, and since y is uniform, its accuracy is $1/2$.
- If π_0 acquires the shortcut x_{2b}, it spends the whole budget b and its accuracy is $3/4$.
- Otherwise, π_0 acquires a unit-cost feature x_j, and since $d \geq 2b+1$, we relabel the unit-cost features so that x_j is noise. The remaining budget $b-1$ excludes the shortcut. If π_0 acquires the context x_1, at most $b-2$ features of the relevant block B_{x_1+1} remain affordable. Otherwise, π_0 observes neither x_1 nor the shortcut, so its acquisitions are independent of x_1, and it can afford at most one full block, which is the relevant one with probability at most $1/2$. If a bit of the relevant block is unobserved, it is uniform and independent of the observations, so y is uniform given the observations. Thus, the accuracy is at most $\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}$.

Hence, the regret of π_0 is at least $1/4$ in every case. Finally, if $n(1-p)^d < 1-\delta$, then $\mathrm{Reg}(\pi^{\mathrm{filter}}) \geq 1/4$ with probability greater than δ. □

In contrast, Theorem 1 guarantees the same regret for generative restoration with n scaling as $(1-p)^{-b}$.

B Experimental Details & Additional Results

B.1 Experimental Details

Benchmark and evaluation protocol. We use the AFABench benchmarking framework (Schütz et al., 2026) and inherit its dataset implementations, preprocessing, predictors, method architectures, and hard-budget evaluation protocol unless stated otherwise. We use five dataset splits for every reported setting. CUBE-NUC contains two independently noisy copies of the CUBE features, one twice as costly as the other. For each split, the predictor is trained before synthetic missingness is imposed and held fixed for all approaches.

Training missingness. For each dataset split, a fixed missingness mechanism is instantiated on the training split, and the missingness masks are fixed across epochs and AFA episodes. We implement synthetic MCAR, MAR, logistic MNAR, and self-masking MNAR at missingness rates $p \in \{0.3, 0.5, 0.7\}$. Under MCAR, features are independently unavailable with probability p. For MAR and logistic MNAR, we randomly select 30% of the features to determine the missingness probabilities of the remaining features through a logistic function. The intercepts are calibrated to match the requested missingness rate. The selected features remain available under MAR but are independently masked at the same rate under logistic MNAR. Under self-masking MNAR, the missingness probability of a feature depends on its own value. Missingness masks that would remove all acquisitions from an instance are resampled.

Generative restoration. We implement generative restoration using the PVAE architecture from Zannone et al. (2019), which is based on the Partial VAE of Ma et al. (2019). For each training–validation pair with artificially induced missingness, we fit a separate PVAE on the training split and use the validation split for model selection. During training, random input masks are applied on top of each instance's missingness, and reconstruction loss is computed only for its available features. We then condition the PVAE on each instance's available features and label, draw a single joint sample of all d features from its decoder, and replace only the unavailable values. We use latent dimension 20, batch size 128, Adam with learning rate 10^{-3}, KL weight 0.1, and sample the additional masking probability uniformly from $[0, 0.9]$. All remaining architecture and optimization settings follow `AFABench`'s ODIN configuration.

Method adaptations. For every method trained on restricted data, acquisitions unavailable in a training instance are removed from its legal action set throughout the trajectory. We evaluate ODIN in its model-free configuration by disabling the additional synthetic-data generation used by its model-based variant. For JAFA, OL, and model-free ODIN, we additionally compare two state representations. In the *aliasing* variant, the availability mask is used only to block illegal actions and is not provided to the learned decision function. In the *filtering* variant, the availability of the remaining acquisitions is included in the state: JAFA and OL learn $Q(s, m, a)$, while both the actor and critic of ODIN condition on m. The variants are otherwise identical.

For DIME and GDFS, we retain the standard AFABench architectures and objectives, but initialize unavailable acquisitions as already exhausted in their selection masks. AACO requires further adaptation because its nearest-neighbor search compares a query against incomplete training instances. For a query with acquired set S and a training instance with availability mask $\bar{m}^{(i)}$, we use the shared-support distance

$$d_i^2(s) = \frac{\sum_{j \in S} \bar{m}_j^{(i)} \left(x_j - x_j^{(i)}\right)^2}{\sum_{j \in S} \bar{m}_j^{(i)}}.$$

Training instances with no shared support are excluded. Normalizing by the shared-support size prevents instances with fewer comparable features from appearing artificially close, and candidate losses are evaluated using only features available in each neighbor.

Training, budgets, and metrics. All remaining settings follow AFABench, except for RL training lengths selected by convergence on complete data. This yielded 2,000 batches for OL and 4,000 for JAFA and model-free ODIN, with 256 frames per batch. These lengths are fixed across missingness mechanisms, rates, state representations, and restoration treatments. We report each dataset at its largest predefined hard budget: 10 for CUBE, ACTG175, and NHANES mortality; 14 for CUBE-NM; 7 for CUBE non-uniform and Heart Disease; and 15 for Diabetes and MiniBooNE. We report accuracy on the balanced synthetic datasets and macro-F1 on the real-world datasets, with means over the five dataset splits.

B.2 Pseudocount Smoothing in the Synthetic Controlled Study

The synthetic controlled study uses a pseudocount of $1/2$ for each binary outcome. For N_1 occurrences of outcome one among $N > 0$ compatible instances, the change from the empirical frequency is

$$\left| \frac{N_1 + 1/2}{N + 1} - \frac{N_1}{N} \right| = \frac{|N/2 - N_1|}{N(N+1)} \leq \frac{1}{2(N+1)}.$$

The fixed binary predictor's expected zero-one loss is either the label-one probability or its complement, so the same bound applies to the stopping-value estimate. On the count event of the proof

of Theorem 1 (Section A.2.3), $N \geq nqP_s/2$. At a state with sufficient expected support, we have $nqP_s \geq 8\Lambda_J$ and $\Lambda_J \geq \log 24 > 1$, so $\Lambda_J nqP_s > 1$ and the additional error satisfies

$$\frac{1}{2(N+1)} \leq \frac{1}{nqP_s} \leq \sqrt{\frac{\Lambda_J}{nqP_s}}.$$

Adding this to the Hoeffding deviation $\ell_{\max}\sqrt{\Lambda_J/(nqP_s)}$ of that proof at most doubles it, and $4\Lambda_J \leq 12\Lambda \leq 24\Lambda$. Thus, Equation (7) still holds. For smaller counts, including $N = 0$, the smoothed probability remains in $[0, 1]$, so the same bounded-error argument applies. The remainder of that proof is unchanged, giving the same sufficient sample-size bounds for the smoothed filtering and restoration estimates used in the synthetic controlled study.

B.3 Dataset Structure

Equation (6) shows that aliasing can converge to the wrong policy, but not that every dataset must be damaged by the restricted action space. If missingness only blocks irrelevant features, then restricting the action space does not change the optimal policy. Similarly, if several strong acquisition routes are interchangeable, blocking one route can have little effect because another route remains available. Missingness damage also depends on the acquisition structure of a dataset, for which we introduce two empirical measures. To compute both quantities, we first split the training data into a selection and a validation set. Then, we sample L feasible acquisition routes (action sequences) $R_1, \ldots, R_L$ under budget b. Let $V_{\text{sel}}(R)$ and $V_{\text{val}}(R)$ denote the static scores of route R on the selection and validation splits, respectively, and let

$$R_\star \in \operatorname*{arg\,max}_{R \in \{R_1, \ldots, R_L\}} V_{\text{sel}}(R),\ V_{\text{static}} \coloneqq V_{\text{val}}(R_\star).$$

We define *route sensitivity* as

$$\Delta_{\text{route}} \coloneqq V_{\text{static}} - \frac{1}{L}\sum_{k=1}^{L} V_{\text{val}}(R_k).$$

A large Δ_{route} indicates that the selected route outperforms a typical feasible route, while a small value indicates little separation in their validation scores. To measure whether high-performing routes use the same acquisitions, let

$$w_k \coloneqq [V_{\text{sel}}(R_k) - V_{\text{sel}}(\varnothing)]_+$$

denote the positive predictive gain over the empty route. Treating each route as a set of legal acquisition actions, we define *performance-weighted route overlap* as

$$\omega_{\text{route}} \coloneqq \frac{\sum_{1 \leq i < j \leq L} w_i w_j \dfrac{|R_i \cap R_j|}{|R_i \cup R_j|}}{\sum_{1 \leq i < j \leq L} w_i w_j}.$$

A large value indicates that predictive gain is concentrated on routes with overlapping acquisition sets, whereas a small value indicates that the gain is distributed across structurally distinct routes.

Table 1: Fixed-route structure. Each column ranks the datasets by that quantity alone, largest first, so the three rankings are independent. Mean $\pm$ standard error over five dataset splits. Static references and route weights are selected on training data; V_{static} and Δ_{route} are reported on validation data. Each dataset is at its largest evaluation budget b, which is 15 for Diabetes and MiniBooNE, 14 for CUBE-NM, 10 for ACTG175, CUBE and NHANES, 7 for CUBE-NUC and Heart disease.

V_{static}		Δ_{route}		ω_{route}	
MiniBooNE	0.894 ± 0.004	CUBE	0.245 ± 0.010	NHANES	0.669 ± 0.000
CUBE-NUC	0.847 ± 0.019	ACTG175	0.175 ± 0.012	CUBE	0.342 ± 0.000
ACTG175	0.846 ± 0.012	CUBE-NUC	0.174 ± 0.017	ACTG175	0.287 ± 0.001
CUBE	0.824 ± 0.010	Diabetes	0.166 ± 0.001	Heart disease	0.264 ± 0.003
NHANES	0.731 ± 0.005	Heart disease	0.127 ± 0.015	CUBE-NUC	0.221 ± 0.000
Heart disease	0.720 ± 0.019	CUBE-NM	0.059 ± 0.010	Diabetes	0.205 ± 0.000
Diabetes	0.679 ± 0.003	MiniBooNE	0.058 ± 0.003	MiniBooNE	0.181 ± 0.000
CUBE-NM	0.317 ± 0.020	NHANES	0.009 ± 0.002	CUBE-NM	0.164 ± 0.000

Table 1 shows V_{static}, Δ_{route}, and ω_{route} across the datasets, with additional results available in Appendix B.6.

Observations on Table 1. CUBE-NM has by far the lowest V_{static}, indicating that a static route does not yield good performance. It also has the lowest ω_{route}, so what predictive gain there is comes from structurally distinct routes. In contrast, NHANES has the largest ω_{route} and the smallest Δ_{route}, indicating that we should expect low missingness damage. We want to highlight MiniBooNE's interesting structure; it has the highest V_{static} while also having the second-smallest ω_{route} and a small Δ_{route}, indicating that there exist a few top-performing routes with minimal overlap.

B.4 Missingness Damage and Adaptive Structure

By Proposition 1, MCAR leaves one-step values unchanged and can only lower multi-step values. Thus, only methods that exploit multi-step acquisitions can have their exact values damaged. Table 2 compares each method's complete-data score with the static route V_{static} of Table 1. DIME and ODIN score below the static route on average, and on CUBE-NM, where the other methods are damaged the most, they score 0.24 and 0.27 against $V_{\text{static}} = 0.32$. In contrast, GDFS, AACO, and OL exceed the static route and are damaged. GDFS is myopic, so its exact one-step values are unchanged, but it estimates them from its own acquisition trajectories: a training instance follows the evaluation trajectory to depth t only if all t acquired features are available, which under MCAR occurs with probability $(1-p)^t$, the same factor as in Theorem 1. Restoration removes this factor, in line with its gain for GDFS. JAFA is the exception, scoring below the static route on average while being damaged on Heart Disease and CUBE-NUC. Finally, the correlation between margin and damage across method–dataset cells is moderate and driven mainly by CUBE-NM.

Table 2: Complete-data score, its margin over the static route V_{static} (Table 1), and missingness damage under MCAR at $p = 0.7$, averaged over the eight datasets and, for JAFA, OL and ODIN, over both state representations. Across method–dataset cells, Pearson $r = 0.59$ and Spearman $\rho = 0.34$ between margin and damage.

Method	Complete	Margin over V_{static}	Damage
DIME	0.705	-0.027	$+0.002$
ODIN	0.667	-0.065	$+0.009$
GDFS	0.745	$+0.013$	$+0.059$
AACO	0.758	$+0.026$	$+0.062$
JAFA	0.707	-0.026	$+0.072$
OL	0.745	$+0.013$	$+0.092$

B.5 Computational Cost of Restoration

Generative restoration introduces an additional model, but that model is trained once per dataset split and missingness setting and then reused across downstream AFA methods. Across the runs in Figure 5, the median generative-to-restricted wall-clock ratio is 1.15; 59% of pairs are within $1.25\times$ and 74% within $1.5\times$. The amortized PVAE fitting and restoration steps contribute a median of 29.3 seconds per trained method. Thus, the additional computational cost is typically modest relative to downstream AFA training, although the overhead varies substantially across methods.

B.6 Results beyond MCAR

The main text focuses on MCAR, where the theoretical comparison is sharpest. Figure 6 reports the restricted training and restoration comparison for all four mechanisms (MCAR, MAR, MNAR (logistic), and MNAR (self-masking)). We observe similar patterns as in Figure 4, namely that missingness can have little effect in some settings and substantial effect in others, and restoration does not uniformly improve every method.

To separate the amount of damage caused by missingness from the amount recovered by restoration, Figure 7 plots restoration gain against missingness damage for each method and mechanism. The fitted rays use cells with damage at least 0.01, so their slopes summarize recovery only where there is a non-negligible gap to recover.

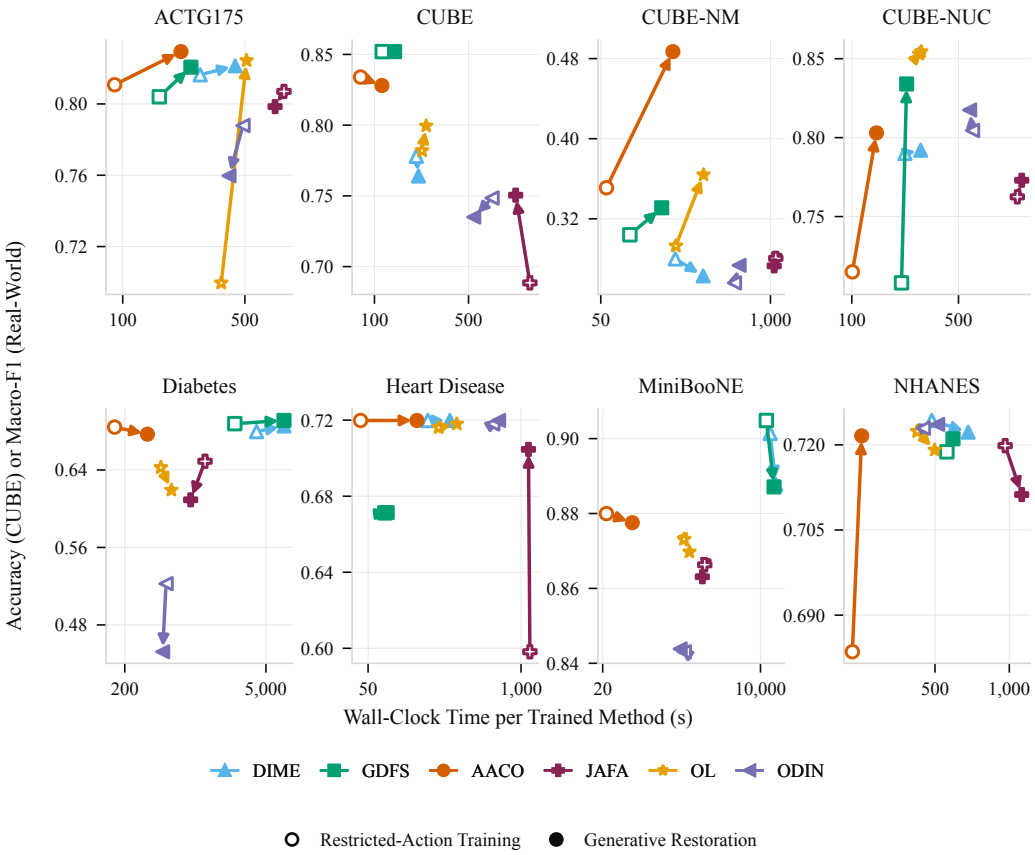


Figure 5: Predictive performance against wall-clock time per trained method. Each arrow points from restricted training to generative restoration for the same dataset and method.

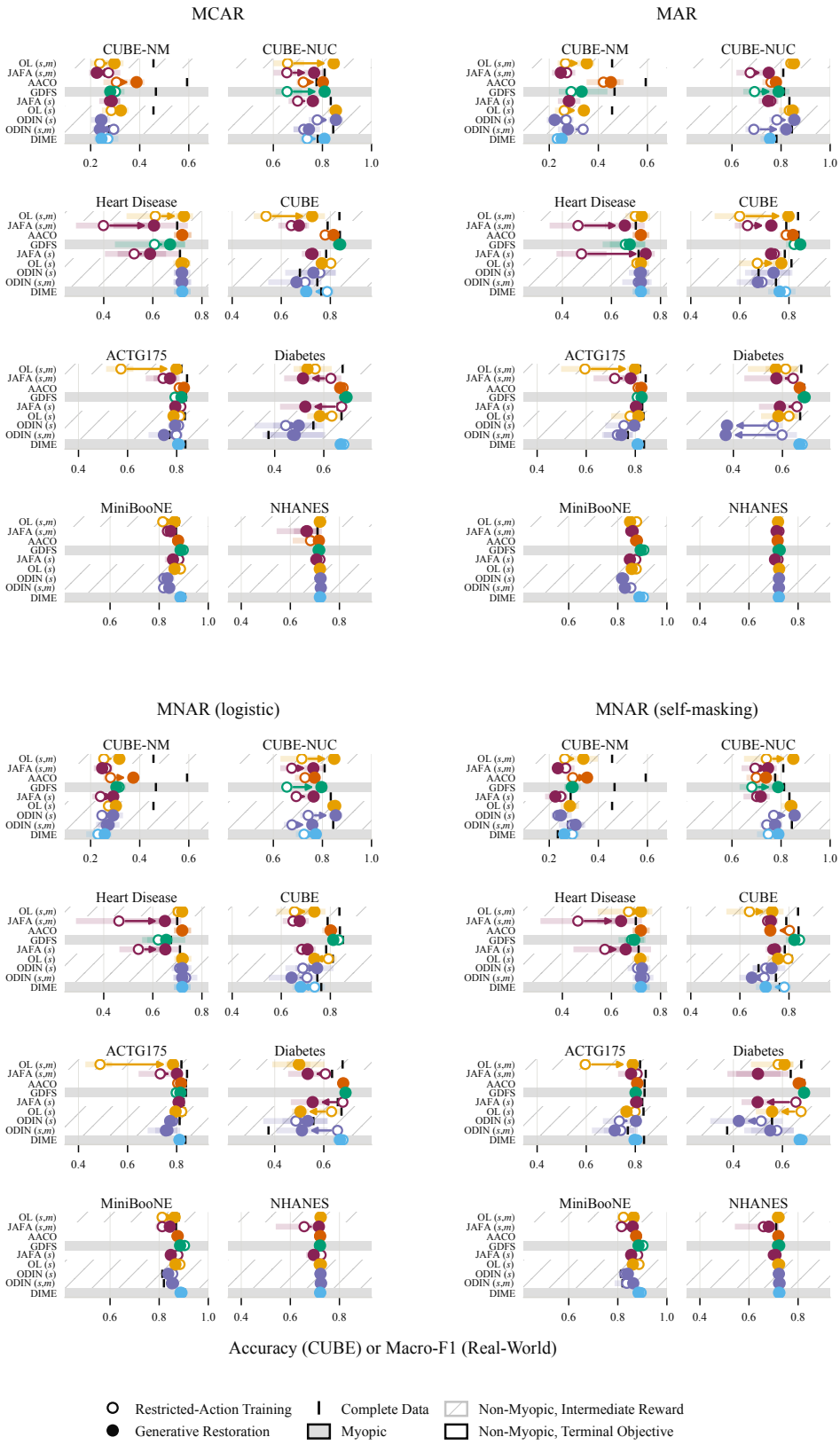


Figure 6: Restricted training against generative restoration at $p = 0.7$ under all four missingness mechanisms.

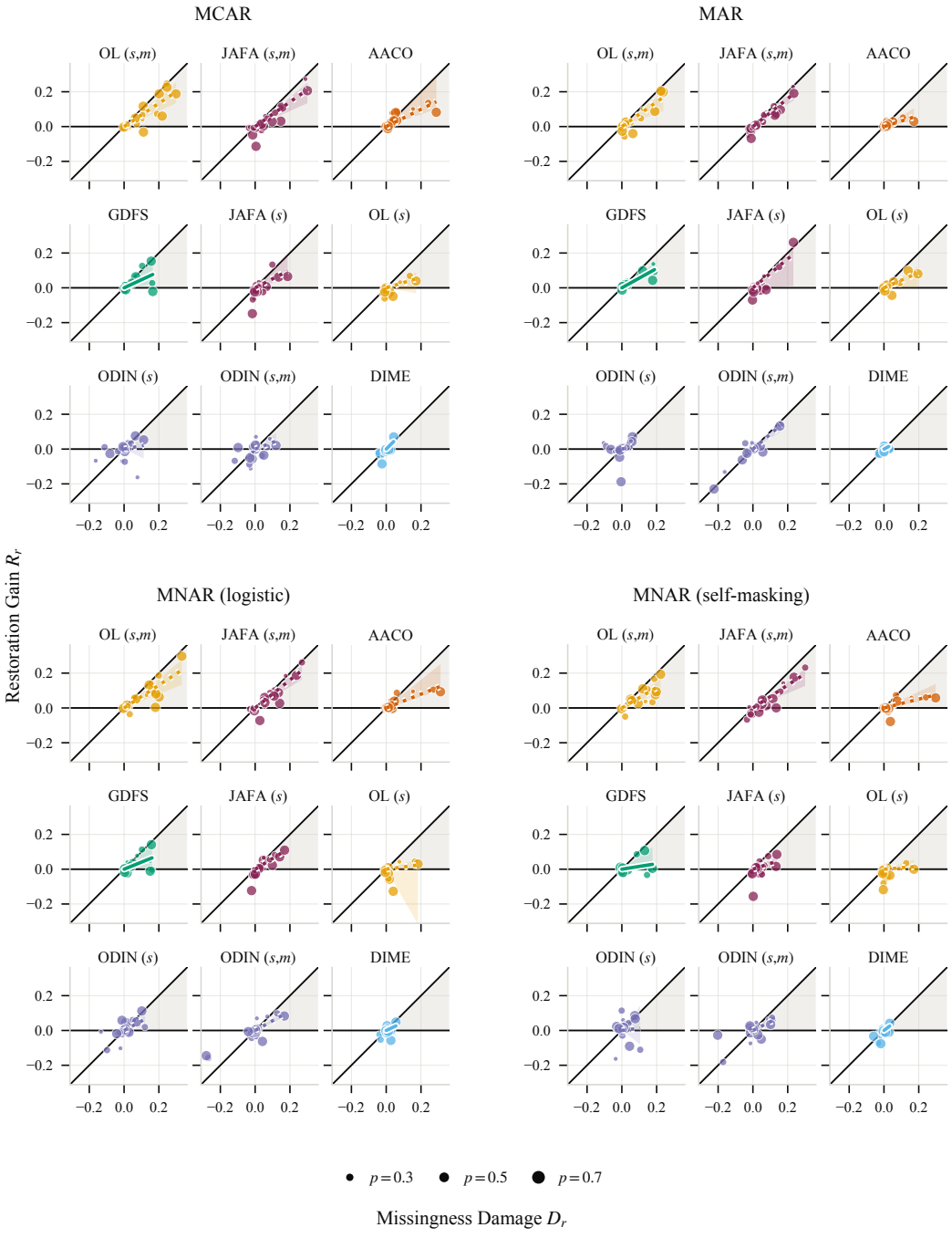


Figure 7: Restoration gain against missingness damage, one panel per method and one block per mechanism, over all eight datasets and three rates.