LEARNING CAUSAL NORMALIZING FLOWS FROM INCOMPLETE DATA VIA OBSERVED-DATA LIKELIHOOD

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal Normalizing Flows (CNFs) enable causal inference from observational data given the causal structure, but they assume fully observed training data. We introduce MissCNF, which trains CNFs directly on incomplete data by maximizing the marginal likelihood of each partially observed sample, without discarding rows or constructing a completed dataset. Thanks to the causal structure encoded in the autoregressive factorization of CNFs, only missing variables in the ancestral closure of the observed set are integrated out, while the others are dropped without computation. We further establish the conditions under which MissCNF recovers the true joint distribution, and introduce *causal-family positivity*, where identification is possible even when no record in the dataset is ever complete. We compare MissCNF with two common strategies for handling missing data: listwise deletion and impute-then-fit pipelines. Across eight synthetic causal benchmarks, three missingness mechanisms, and missing rates up to 90%, MissCNF achieves the lowest KL divergence in 23 of 24 nonlinear MCAR and MAR settings and in all nonlinear MNAR settings, as well as the lowest counterfactual error in 20 of 24 settings. On linear SCMs, where linear imputation performs best, MissCNF ranks in the top two in 22 of 24 settings.

1 INTRODUCTION

Randomized controlled trials remain the reference standard for estimating causal effects, but they are expensive, slow, and in many settings impossible to run. This has motivated a large body of work on causal inference from observational data, including Deep Structural Causal Models (Pawlowski et al., 2020), Variational Graph Autoencoders (Sánchez-Martin et al., 2022), and Diffusion-based approaches (Chao et al., 2024). Causal Normalizing Flows (CNFs) (Javaloy et al., 2023) parameterize the structural equations with an invertible flow masked according to the causal graph. CNFs have two key advantages: they yield exact likelihoods rather than bounds, and their density factorizes along the graph, so a model fitted to observational data answers interventional and counterfactual queries directly.

However, like many machine learning methods, CNFs were developed assuming complete training data. This is a significant limitation in domains such as healthcare, where missing data is unavoidable. One option is to discard incomplete records, also known as *listwise deletion*, which wastes data and, more importantly, introduces bias whenever the probability of being fully observed depends on the values themselves, which Missing At Random (MAR) does not rule out. The other common option is first to fit an imputation model, then fit the causal model on its output as if it were the true data (Parra et al., 2026). Treating imputed values as observed data discards the uncertainty in the imputation stage and lets any error there propagate into the causal modelling stage.

In this work, we introduce **MissCNF**, which trains CNFs directly on missing data, without discarding rows or constructing any completed dataset, by computing the marginal likelihood of an arbitrary observed subset of variables. The proposed method can be traced back to *Full Information Maximum Likelihood* (FIML), originally developed for the multivariate normal distribution (Anderson, 1957) and linear-Gaussian Structural Equation Models (Lee, 1986; Muthén et al., 1987), in which the marginal likelihood is available in closed form. In our setting, the Structural Causal Models represented by CNFs are not necessarily linear and Gaussian, so no closed form is available. We instead compute the required marginal likelihood by a Monte Carlo estimator that exploits the Directed

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.

Acyclic Graph (DAG) factorization: it samples only missing variables in the ancestral closure of that observed subset, while variables outside the ancestral closure are marginalized without simulation.

We then study what can be learned from the resulting observed-data likelihood. Under Missing At Random (MAR), the missingness mechanism is ignorable for likelihood-based inference (Rubin, 1976), but ignorability alone does not guarantee that the complete-data distribution is identified: if a set of variables is never jointly observed in some region of their support, the observed data carry no information about their joint distribution there. Recovering the complete-data distribution therefore requires some form of support-overlap assumption relating what is missing to what is observed (Näf et al., 2026). Exploiting the causal structure a CNF encodes, we show that identifying the complete joint distribution requires only a *causal-family positivity* condition that holds even if no row is ever fully observed: every variable must be jointly observed with its causal parents in some pattern, with positive probability for almost every value in the relevant support. We further show that MissCNF already recovers, as a byproduct, every conditional an imputation model would need to target. Identification under MNAR requires additional assumptions and is left to future work, though our experiments on eight synthetic causal graphs, across three missingness mechanisms and multiple missingness rates, show MissCNF still performs well in different regimes, including MNAR.

Our contributions are as follows: (1) We derive a marginal-likelihood objective that trains CNFs directly from incomplete data by integrating only over missing ancestral variables (Section 3.1); (2) Under MAR, we separate likelihood ignorability from identification and establish a *causal-family positivity* condition that can be strictly weaker than *full-pattern positivity* and is enough to recover the full joint even when no row is fully observed (Section 3.2); (3) We characterize the population target of impute-then-fit pipelines and show that MissCNF recovers every such conditional as a byproduct (Section 3.3); (4) We empirically validate the method on eight synthetic SCMs, showing consistent improvements in density estimation, counterfactual prediction, and average treatment effect estimation over baselines across missingness mechanisms and rates (Section 4).

2 Background

Structural causal models. A *Structural Causal Model* (SCM) (Pearl, 2009) $\mathcal{M} = (\mathbf{f}, P_{\mathbf{u}})$ describes a data-generating process that transforms d mutually independent exogenous noise variables $\mathbf{u} = \{u_i\}_{i=1}^{d} \sim P_{\mathbf{u}}$ into d observed endogenous variables $\mathbf{x}$ through structural equations $\mathbf{f} = \{f_i\}_{i=1}^{d}$:

$$x_i = f_i\big(\mathbf{x}_{\mathrm{pa}(i)}, u_i\big), \qquad i = 1, \dots, d,$$

where $\mathrm{pa}(i)$ are the parents of i. Each SCM induces a causal graph $G = (V, E)$ on $V = \{1, \dots, d\}$, with $(j, i) \in E$ if $j \in \mathrm{pa}(i)$; we fix a topological order $\prec$ compatible with G throughout, so $j \in \mathrm{pa}(i) \Rightarrow j \prec i$.

Causal normalizing flows. A *Causal Normalizing Flow* (CNF) (Javaloy et al., 2023) instantiates an SCM $\mathcal{M}_\theta = (T_\theta, P_{\mathbf{u}})$ by representing the structural equations implicitly through a masked autoregressive normalizing flow T_θ (Germain et al., 2015; Papamakarios et al., 2017). A CNF takes the causal graph G as given and assumes *causal sufficiency* (Spirtes et al., 1993) such that there is no unobserved confounding. We inherit these assumptions unchanged in this work. We adopt the abductive direction with a single layer as it is causally consistent in both directions between $\mathbf{u}$ and $\mathbf{x}$, without requiring additional layers or regularization (Javaloy et al., 2023). Let h_i be a conditioner masked according to G, taking only the parents of i as input, and $\tau(\cdot\,; \eta)$ be a transformer strictly monotonic and continuously differentiable in its first argument for every conditioner output η. Each coordinate is then given by

$$u_i = \tau\big(x_i\,;\, h_i(\mathbf{x}_{\mathrm{pa}(i)})\big), \qquad x_i = \tau^{-1}\big(u_i\,;\, h_i(\mathbf{x}_{\mathrm{pa}(i)})\big), \qquad i = 1, \dots, d,$$

Since the u_i are mutually independent, $P_{\mathbf{u}}$ factorizes as $\prod_i p(u_i)$, and the resulting density $p_\theta(\mathbf{x}) \equiv p_\theta(x_1, \dots, x_d)$ factorizes according to G:

$$p_\theta(x_1, \dots, x_d) = \prod_{i=1}^{d} p_\theta(x_i \mid \mathbf{x}_{\mathrm{pa}(i)}), \qquad p_\theta(x_i \mid \mathbf{x}_{\mathrm{pa}(i)}) = p(u_i) \left| \frac{\partial \tau}{\partial x_i}(x_i; h_i(\mathbf{x}_{\mathrm{pa}(i)})) \right|.$$

For any index subset $O \subseteq V$ we write $p_{\theta,O}(\mathbf{x}_O)$ for the corresponding marginal of p_θ. Javaloy et al. (2023) show that, given only the causal ordering, the family of CNFs is identifiable: any two

CNFs matching the true observational distribution recover the same exogenous variables up to an invertible, component-wise transformation of variables $\mathbf{u}$. Consequently, fitting T_θ to observational data by maximum likelihood, $\max_\theta \log p_\theta(\mathbf{x})$, recovers the true SCM in this sense. This objective, however, presumes each $\mathbf{x}$ is fully observed, which might not always happen in many applications.

Missing mechanisms. Let $\mathcal{R} \in \{0,1\}^d$ be a random missingness mask and $\mathbf{X} \in \mathbb{R}^d$ be a random sample, with $\mathcal{R}_i = 1$ indicating that variable X_i is observed. For a realization R of $\mathcal{R}$, we write $O(R) := \{i : R_i = 1\}$ and $M(R) := V \setminus O(R)$ as the observed and missing index sets. We drop R when it is clear from context, so that $\mathbf{x}_O$ and $\mathbf{x}_M$ denote the observed and missing coordinates of a realization $\mathbf{x}$. Because R, O, and M map bijectively to one another, we use them interchangeably and refer to this shared representation as a ***pattern***. The missingness mechanism is the conditional distribution $p_{\text{miss}}(R \mid \mathbf{x})$, which can be one of the following (Rubin, 1976):

- *Missing Completely At Random* (MCAR): $p_{\text{miss}}(R \mid \mathbf{x}) = p_{\text{miss}}(R)$, i.e., the mask is independent of $\mathbf{x}$ entirely.
- *Missing At Random* (MAR): $p_{\text{miss}}(R \mid \mathbf{x}) = p_{\text{miss}}(R \mid \mathbf{x}_O)$, i.e., the mask may depend only on the observed values. MCAR is the special case of MAR.
- *Missing Not At Random* (MNAR): $p_{\text{miss}}(R \mid \mathbf{x})$ depends on $\mathbf{x}_M$ as well.

We use $p_\theta(\mathbf{x})$ and $p^*(\mathbf{x})$ to denote the full joint densities of the fitted CNF and the true SCM, respectively, while $p_{\theta,O}(\mathbf{x}_O)$ and $p^*_O(\mathbf{x}_O)$ denote their marginals over a subset $O \subseteq V$. Together $\mathcal{M}^*$ and p_{miss} induce a joint density over $(\mathbf{X}, \mathcal{R})$ via $p_{\text{joint}}(\mathbf{x}, R) := p^*(\mathbf{x})\, p_{\text{miss}}(R \mid \mathbf{x})$. Because complete rows are generally not observed, the distribution that a sample is actually drawn from is not p_{joint} but its marginal over the unobserved coordinates. The *observed-data* density is thus $p_{\text{obs}}(\mathbf{x}_O, R) := \int p_{\text{joint}}(\mathbf{x}_O, \mathbf{x}_M, R)\, d\mathbf{x}_M$.

3 METHOD

We aim to fit a CNF $p_\theta(\mathbf{x})$ over all variables from partially observed data. We first introduce MissCNF and derive its training objective, then study the corresponding population objective under different missingness assumptions and establish conditions for recovering the complete-data distribution. Finally, we compare MissCNF with alternative approaches for learning from incomplete data.

3.1 MISSCNF

For an observed/missing split $O \sqcup M = V$ of a given sample, the full-data likelihood $p_\theta(\mathbf{x})$ cannot be evaluated because $\mathbf{x}_M$ is unobserved. The sample therefore contributes the marginal likelihood

$$p_{\theta,O}(\mathbf{x}_O) = \int p_\theta(x_1, \ldots, x_d)\, d\mathbf{x}_M.$$

Let $A := \text{anc}(O)$ be O with all its ancestors, and $M' := A \setminus O$ the missing nodes inside that set (Figure 1a). Any node outside A is a descendant-or-unrelated node with respect to O (*barren*). Barren nodes vanish from the calculation, since integrating a barren node out of the joint density yields 1. The only remaining integral is over M', the missing ancestors. Rather than integrating over $\mathbf{x}_{M'}$ directly, we integrate over its noise $\mathbf{u}_{M'}$. This is a reparameterization: the noise terms are independent draws from the base distribution, which is easy to sample, and each x_i for i in M' is obtained by pushing u_i through the flow given its (already computed) parents (Figure 1b). We formally state this in Proposition 1.

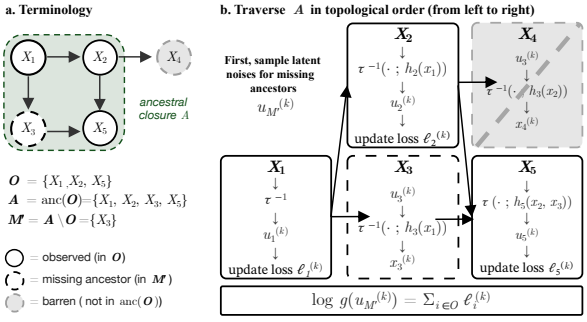


Figure 1: (a) With $O = \{1, 2, 5\}$ observed, A is $\{1, 2, 3, 5\}$. X_3 is a missing ancestor and must be integrated out. X_4 is barren and is dropped without integration. (b) One MC sample of Algorithm 1. Nodes in A are processed in topological order. Observed nodes are mapped forward to their noise and added to the log-likelihood. The missing X_3 is generated from the noise $u_3^{(k)}$ and passed on to its children. X_4 is skipped.

Proposition 1. *The computation of $p_{\theta,O}(\mathbf{x}_O)$ involves only the variables in the ancestral closure $A = \mathrm{anc}(O)$; equivalently, only the missing ancestral coordinates $M' = A \setminus O$ need to be integrated out:*

$$p_{\theta,O}(\mathbf{x}_O) = \mathbb{E}_{\mathbf{u}_{M'} \sim \prod_{i \in M'} p(u_i)}\big[g(\mathbf{u}_{M'})\big], \qquad g(\mathbf{u}_{M'}; \mathbf{x}_O) := \prod_{i \in O} p(u_i) \left| \frac{\partial \tau}{\partial x_i} \right|,$$

where, for each $i \in A$ processed in topological order, $u_i = \tau(x_i; h_i(\mathbf{x}_{\mathrm{pa}(i)}))$ if $i \in O$, and $x_i = \tau^{-1}(u_i; h_i(\mathbf{x}_{\mathrm{pa}(i)}))$ if $i \in M'$. (Proof in Appendix A.1.)

Algorithm 1 Computing $\hat{p}_{\theta,O}(\mathbf{x}_O)$ for a masked autoregressive CNF

Require: Observed values $\mathbf{x}_O$; ancestral set $A = \mathrm{anc}(O)$, missing-within-A set $M' = A \setminus O$; topological order $\prec$ restricted to A; number of samples K

Draw $\mathbf{u}_{M'}^{(k)} \sim \prod_{i \in M'} p(u_i)$ i.i.d. for $k = 1, \ldots, K$
for $k = 1, \ldots, K$ **do**
  for $i \in A$ in topological order $\prec$ **do** ▷ parents of i already resolved
    if $i \in O$ **then**
      $u_i \leftarrow \tau(x_i\,;\, h_i(\mathbf{x}_{\mathrm{pa}(i)}))$ ▷ observed: evaluate forward
      $\ell_i^{(k)} \leftarrow \log p(u_i) + \log |\partial \tau / \partial x_i|$
    else ▷ $i \in M'$: missing, already sampled as $u_i = u_i^{(k)}$
      $x_i \leftarrow \tau^{-1}(u_i^{(k)}\,;\, h_i(\mathbf{x}_{\mathrm{pa}(i)}))$ ▷ invert to propagate to children
    end if
  end for
  $\log g(\mathbf{u}_{M'}^{(k)}; \mathbf{x}_O) \leftarrow \sum_{i \in O} \ell_i^{(k)}$
end for
return $\log \hat{p}_{\theta,O}(\mathbf{x}_O) \leftarrow \mathrm{logsumexp}_{n=1}^{K} \log g(\mathbf{u}_{M'}^{(k)}) - \log K$

We estimate the expectation by Monte Carlo (MC) estimation. Algorithm 1 gives the full procedure: draw K samples of $\mathbf{u}_{M'}$ from the base distribution, propagate each through the flow in topological order to obtain g, and average in log-space. If no missing node is an ancestor of an observed node, M' is empty, and $K = 0$ samples are needed. As we use an abductive autoregressive CNF with an $L = 1$ layer, as in Javaloy et al. (2023), the full-density evaluation costs $\mathcal{O}(L)$, while sampling requires $\mathcal{O}(dL)$ due to autoregressive inversion (Javaloy et al., 2023; Papamakarios et al., 2021). Computing the marginal density in the algorithm requires K MC samples; therefore, it gives $\mathcal{O}(KdL)$ total computational work.

Given this recipe for computing $\hat{p}_{\theta,O}(\mathbf{x}_O)$ for any observed subset O, we can now train a CNF directly on incomplete data by maximizing the likelihood it assigns to whatever each row actually shows. Given a training set of n i.i.d. samples $D^{(j)} = (\mathbf{x}_{O_j}^{(j)}, O_j)$, we define the training objective as $\mathcal{L}_{n,K}(\theta) := -\frac{1}{n} \sum_{i=1}^{n} \log \hat{p}_{\theta,O_j}\big(\mathbf{x}_{O_j}^{(j)}\big)$. As both the number of MC samples K in Algorithm 1 and the training set size n grow, $\hat{p}_{\theta,O} \to p_{\theta,O}$ and the empirical loss converges to the population risk

$$\mathcal{L}(\theta) = \mathbb{E}_{(\mathbf{x}_{O(\mathcal{R})}, \mathcal{R}) \sim p_{\mathrm{obs}}}\big[-\log p_{\theta,O(\mathcal{R})}(\mathbf{X}_{O(\mathcal{R})})\big]\,.$$

The objective uses only the observed entries of each row, so it can be applied to any missingness pattern. Whether its minimizer recovers p^*, however, depends on the missingness mechanism, which we study in Section 3.2.

3.2 MAR: Ignorability and Identification

The population risk in Section 3.1 uses only the model marginal $p_{\theta,O}(\mathbf{x}_O)$, rather than explicitly modeling the missingness pattern R. We first show that, under MAR, this is equivalent to optimizing the joint likelihood of the observed values and missingness pattern, up to a term independent of θ.

For any candidate complete-data model p_θ, let q_θ denote the distribution over $(\mathbf{X}_{O(\mathcal{R})}, \mathcal{R})$ obtained by applying the fixed missingness mechanism p_{miss}. Lemma 1 shows that MAR factorizes this observed-data density into the model marginal $p_{\theta,O}(\mathbf{x}_O)$ and a missingness term independent of θ. This is the classical likelihood-ignorability property under MAR (Rubin, 1976).

Lemma 1 (MAR pull-out identity, for any candidate θ)**.** *Fix a pattern $O \subseteq V$ and let $R = \mathbf{1}_O$. For* any *θ, define*

$$q_\theta(\mathbf{x}_O, R) := \int p_\theta(\mathbf{x}_O, \mathbf{x}_M)\, p_{\text{miss}}(R \mid \mathbf{x}_O, \mathbf{x}_M)\, d\mathbf{x}_M,$$

the observed-data density that would arise if $\mathbf{x}$ were generated by the candidate model p_θ while R is generated by the true, fixed mechanism p_{miss}. Under MAR, q_θ simplifies for every *θ:*

$$q_\theta(\mathbf{x}_O, R) = p_{\theta,O}(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O).$$

In particular, if the model is realizable ($p_{\theta_0} = p^*$ for some θ_0), then $p_{obs}(\mathbf{x}_O, R) = q_{\theta_0}(\mathbf{x}_O, R) = p^*_O(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O)$. (Proof in Appendix A.2.)

Since $\log q_\theta(\mathbf{x}_O, R) = \log p_{\theta,O}(\mathbf{x}_O) + \log p_{\text{miss}}(R \mid \mathbf{x}_O)$ and the second term does not depend on θ, minimizing $\mathcal{L}$ is equivalent to maximum likelihood on the observed values and patterns,

$$\arg\min_\theta \mathcal{L}(\theta) = \arg\min_\theta \mathbb{E}_{p_{\text{obs}}}\big[-\log q_\theta(\mathbf{X}_{O(\mathcal{R})}, \mathcal{R})\big],$$

so p_{miss} can be dropped from the objective without changing the minimizer.

Theorem 1 (MAR identification of the population risk, on the identified region)**.** *Suppose MAR holds, and the CNF family is realizable ($p_{\theta_0} = p^*$ for some θ_0). Define $Z_O := \{\mathbf{x}_O : p_{\text{miss}}(\mathbf{1}_O \mid \mathbf{x}_O) = 0\}$ the set of values of $\mathbf{x}_O$ that pattern O never reveals. Then θ_0 is a global minimizer of $\mathcal{L}(\theta)$, and* every *global minimizer θ^* satisfies*

$$p_{\theta^*,O}(\mathbf{x}_O) = p^*_O(\mathbf{x}_O) \quad \textit{for } p^*_O\textit{-almost every } \mathbf{x}_O \notin Z_O, \quad \textit{simultaneously for every } O.$$

(Proof in Appendix A.2.)

Theorem 1 says that the trained model reproduces the true distribution of each observed pattern, but only where that pattern actually reveals data. It does not tell us what the model does on values a pattern never shows. To allow the model to recover the true distribution, we need one of the following conditions to hold.

Definition 1 (Positivity conditions)**.** *We say* full-pattern positivity *holds if $p^*(Z_V) = 0$, i.e. some rows are fully observed and $p_{\text{miss}}(\mathbf{1}_V \mid \mathbf{X} = \mathbf{x}) > 0$ for p^*-almost every $\mathbf{x}$. We say* causal-family positivity *holds if, for every node $i \in V$, there exists a pattern O with* causal-family *of $i = \{i\} \cup \text{pa}(i) \subseteq O$, and $p^*_O(Z_O) = 0$, i.e. $p_{\text{miss}}(\mathbf{1}_O \mid \mathbf{X}_O = \mathbf{x}_O) > 0$ for p^*-almost every $\mathbf{x}_O$ (as formalized in Corollary 1).*

Two routes to full joint recovery. Theorem 1 identifies each marginal $p_{\theta,O}(\mathbf{x}_O)$ outside its unidentified region Z_O, pattern by pattern. Taking $O = V$ gives the direct route to obtain $p_{\theta^*} = p^*$, full-pattern positivity: every $\mathbf{x}$ has some chance of being fully observed. The second route exploits the CNF's causal structure. Because the same model marginal over any subset is shared across every observation pattern containing that subset, a causal family may be identified through one pattern even if another pattern provides no support there. Corollary 1 shows that positivity need only be checked locally, once per node against its own causal family, rather than globally against the full vector $\mathbf{x}$.

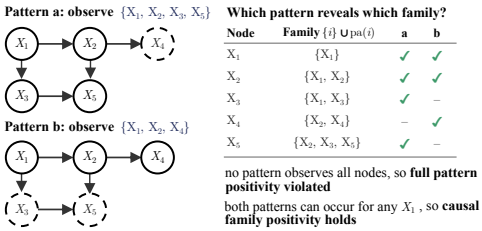


Figure 2: **Left:** the two observed patterns. **Right:** the causal family $\{i\} \cup \text{pa}(i)$ of each node and the patterns that contain it. No pattern observes all of V, but every family is observed by at least one pattern.

Figure 2 gives an example. Only two patterns occur: pattern a observes $\{X_1, X_2, X_3, X_5\}$ and pattern b observes $\{X_1, X_2, X_4\}$. No row is ever complete, so full-pattern positivity is violated. Suppose X_1 is always observed and the pattern is chosen with probability $\text{sigmoid}(x_1)$ for a and $1 - \text{sigmoid}(x_1)$ for b. The mechanism then depends only on X_1, which both patterns observe, so it is MAR. Since both probabilities are positive for every x_1, $Z_{O_a} = Z_{O_b} = \emptyset$. Each causal family is

contained in at least one pattern (right panel), so causal-family positivity holds, and the joint is still recovered.

Corollary 1 (Joint recovery via causal-family positivity, without requiring full-pattern positivity in general)**.** *Suppose p_{θ^*} and p^* both factorize according to G (Section 2). Under causal-family positivity (Definition 1) and $p^*_{\mathrm{pa}(i)}(\mathbf{x}_{\mathrm{pa}(i)}) > 0$ on the relevant domain for each i, every global minimizer θ^* of $\mathcal{L}$ satisfies $p_{\theta^*}(\mathbf{x}) = p^*(\mathbf{x})$ for p^*-almost every $\mathbf{x}$. (Proof in Appendix A.2.)*

Missing Not At Random (MNAR). Under MNAR, the missingness probability may depend on values that are themselves unobserved, so the MAR factorization used in Theorem 1 no longer holds, and the distribution of the observed entries alone does not generally identify $p^*(\mathbf{x})$. Nevertheless, our estimator remains well-defined under MNAR and continues to optimize its marginal-likelihood objective based on $p_{\theta,O}(\mathbf{x}_O)$, without constructing any completed dataset. This does not remove MNAR-induced selection bias, but it avoids compounding it with a second, imputation-based stage. We evaluate the method empirically under MNAR in Section 4.3.2, while leaving formal identification under additional MNAR assumptions to future work.

3.3 Comparison to Alternative Estimators

Listwise deletion. Listwise deletion discards every row with at least one missing coordinate and fits the model only on complete cases. It therefore targets

$$p^*(\mathbf{x} \mid \mathcal{R} = \mathbf{1}_V) \propto p^*(\mathbf{x})\, p_{\mathrm{miss}}(\mathbf{1}_V \mid \mathbf{x})$$

rather than $p^*(\mathbf{x})$ itself. Under MCAR, $p_{\mathrm{miss}}(\mathbf{1}_V \mid \mathbf{x})$ is constant, and the two coincide; under general MAR, the complete-case probability may vary with $\mathbf{x}$, so the complete-case distribution is generally reweighted. Moreover, listwise deletion requires complete samples, whereas Corollary 1 shows that MissCNF can recover the full joint even when no row is fully observed under the causal-family positivity (Corollary 1).

Imputation-then-fit Approaches A common alternative is impute-then-fit: an imputation model $q(\mathbf{x}_M \mid \mathbf{x}_O, R)$ first completes the missing coordinates, after which a downstream model is fit to the completed data. The completed rows follow

$$p_{\mathrm{comp}}(\mathbf{x}) = \sum_R p_{\mathrm{obs}}(\mathbf{x}_O, R)\, q(\mathbf{x}_M \mid \mathbf{x}_O, R)\,,$$

so maximum likelihood on the completed data targets p_{comp}, not directly p^*. Under MAR, the oracle conditional $q(\mathbf{x}_M \mid \mathbf{x}_O, R) = p^*(\mathbf{x}_M \mid \mathbf{x}_O)$ is sufficient for $p_{\mathrm{comp}} = p^*$; otherwise, the downstream target generally inherits errors from the imputation model.

MissCNF avoids the intermediate model entirely: under causal-family positivity, it recovers the full joint and therefore every conditional required for imputation, $p_{\theta^*}(\mathbf{x}_B \mid \mathbf{x}_A) = p^*(\mathbf{x}_B \mid \mathbf{x}_A)$, for any partition $V = A \sqcup B$, wherever the conditional is defined (Corollary 2, Appendix A.3). Issues specific to single and multiple imputation, including uncertainty understatement, model pooling, and compatibility, are discussed in Appendix A.4.

4 Experiments

4.1 Setup

We evaluate on eight synthetic SCMs used in prior work (Sánchez-Martin et al., 2022; Javaloy et al., 2023; Chao et al., 2024): Chain, Triangle, Collider, and Fork, each with a linear and a nonlinear version (Appendix B.1). Following Javaloy et al. (2023), we use 20,000 training, 2,500 validation, and 2,500 test samples for each dataset.

We consider three missingness mechanisms, **MCAR**, **MAR**, and **self-masking MNAR** based on previous work (Muzellec et al., 2020). For MCAR and MAR, we use missingness rates of 30%, 60%, and 90%. For MNAR, we use 30% and 60%. The missingness is only applied to the training

and validation sets. The root nodes are never masked to make sure MAR always has observed conditioning variables. The exception is Experiment 3, where we use a MAR variant to test causal-family coverage. Under MCAR, each variable is masked independently. Under MAR and self-masking MNAR, the masking probability of a non-root node is the sigmoid of a linear function of its conditions, which are its root parents (or of all roots if it has none) for MAR and the variable's own value for self-masking MNAR. Full definitions of the missingness mechanisms are given in Appendix B.2.

We compare MissCNF against **listwise deletion** and impute-then-fit pipelines using five non-causal imputers: **Mean imputation**, **MissForest** (Stekhoven & Bühlmann, 2012), **MICE** with single imputation (van Buuren & Groothuis-Oudshoorn, 2011), and the more recent methods **KPI** (Wang et al., 2025) and **DiffPuter** (Zhang et al., 2025), which report state-of-the-art imputation accuracy against other deep imputers such as GAIN (Yoon et al., 2018), MIWAE (Mattei & Frellsen, 2019), and ReMasker (Du et al., 2024). We also use one causal imputer, MIRACLE (Kyono et al., 2021). While MIRACLE learns the causal graph itself, we give it the true graph instead and call this variant **Oracle-graph MIRACLE (O-MIRACLE)**. We additionally train the same CNF on complete data as a reference. All methods use the same downstream causal normalizing flow and training budget; baseline and implementation details are provided in Appendices B.3 and B.4.

Each experiment is repeated over five random seeds affecting the data, missingness mask, and model initialization. Following Javaloy et al. (2023), we evaluate on the complete test set using symmetric **KL** divergence for observational fit, $\mathbf{RMSE_{ATE}}$ for interventional accuracy, and $\mathbf{RMSE_{CF}}$ for counterfactual prediction. More details are given in Appendix B.5. We report KL as the main metric: under the identifiability conditions of Javaloy et al. (2023), a causal NF that matches the observational distribution also recovers the interventional and counterfactual distributions. Nevertheless, finite-sample differences in KL need not translate directly into lower ATE or CF error. We therefore report KL in the main text and provide the full results in Appendix C.

4.2 Main result: Performance under MCAR and MAR missingness

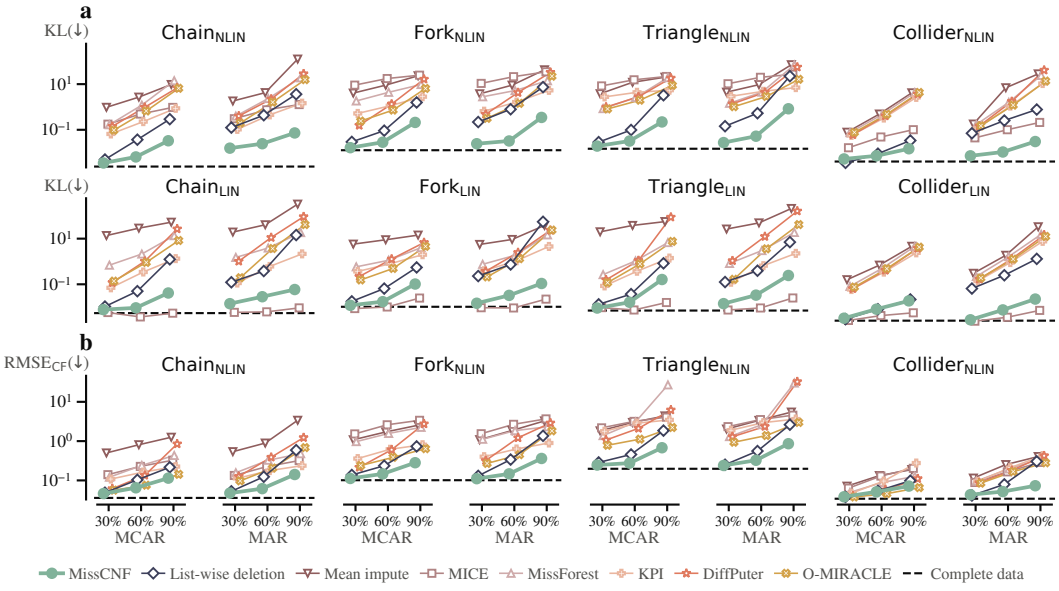


Figure 3: Results under MCAR and MAR at 30%, 60%, and 90% missing rates (mean ± std over 5 seeds; lower is better). (a) Symmetric KL divergence to the ground-truth distribution on all SCMs. (b) $\mathrm{RMSE_{CF}}$ on the nonlinear SCMs. The dashed line marks the model trained on complete data.

We first analyse model identifiability under MCAR and MAR missingness and report the KL in Figure 3a. For nonlinear SCMs, MissCNF attains the lowest symmetric KL divergence in 23 of 24 settings of tasks, mechanisms and missing rates. In contrast, on linear SCMs, MICE achieves the lowest KL in 23 of 24 settings, whereas MissCNF remains competitive, ranking among the top two in 22 of 24 settings This advantage is not limited to comparisons with non-causal imputers: even

O-MIRACLE does not outperform MissCNF overall. At the same time, the strong performance of MICE on the linear benchmarks shows that impute-then-fit can be highly effective when the underlying mechanisms are simple. We investigate this dependence on nonlinearity more directly in Appendix D.1 and confirm that the performance of MICE decreased rapidly while MissCNF remains stable as the degree of nonlinearity increases. The results also illustrate the sensitivity of listwise deletion to the missingness mechanism. Across all tasks and missing rates, its average KL deteriorates by $14.0\times$ from MCAR to MAR, compared with a $2.0\times$ increase for MissCNF. This is consistent with complete-case analysis becoming biased when the probability that a row is fully observed depends on observed variables (Section 3.3). Figure 3b reports RMSE_{CF} on the nonlinear SCMs. The results follow the KL results: MissCNF has the lowest error in 20 of 24 settings. Results on linear SCMs and for RMSE_{ATE} are in Appendix C.1. They show the same pattern, but the gap between methods is smaller for the ATE.

4.3 Robustness to Weaker Recovery Conditions

In this section, we test the recovery conditions (full-pattern positivity, causal-family positivity) developed in Section 3.2. Then we move to MNAR missingness, where the observed-data distribution no longer generally identifies the full-data distribution.

4.3.1 From Causal-Family Positivity to Non-Identification

In contrast to the previous experiment, where full-pattern positivity holds, we first violate it by retaining causal-family positivity in the absence of any fully observed samples. We use Chain and Fork SCMs with two observation patterns O_1 and O_2 in each that share an always-observed node x_{cut} (Appendix B.2.2). Which pattern is observed depends only on x_{cut}, so the mechanism is MAR. Under *Family positivity*, the pattern probability (O_1 vs. O_2) is a sigmoid of the standardized x_{cut}, so both patterns occur on the whole support, every causal family is covered, and p^* is identifiable. Under *Family positivity violated*, we additionally never observe O_1 when x_{cut} is below its median τ, so p^* is not identifiable on $Z_{O_1} = \{\mathbf{x} : x_{\text{cut}} < \tau\}$. As a *full positivity* reference, we use the previous MAR mechanism at a 50% missing rate, which matches the number of observed entries. Figure 4 reports the KL; full results are in Appendix C.2.1.

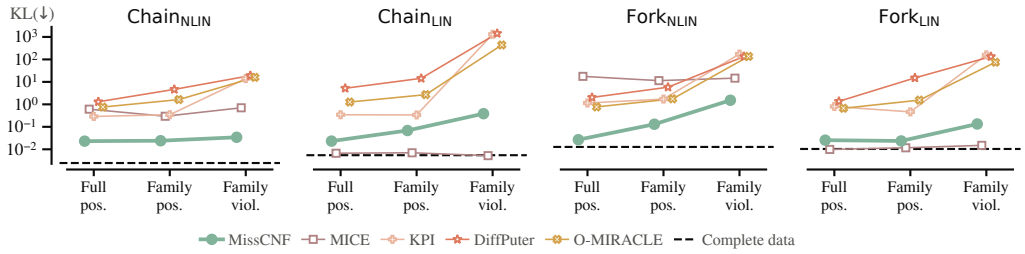


Figure 4: KL to the ground-truth distribution on Chain and Fork graphs (linear and nonlinear) under three regimes: *Full positivity*, *Family positivity*, and *Family positivity violated*. The dashed line marks the model trained on complete data.

Under *Family positivity*, MissCNF matches the full-positivity reference on $\text{Chain}_{\text{NLIN}}$ (0.024 vs. 0.023) and Fork_{LIN} (0.023 vs. 0.025). On $\text{Chain}_{\text{LIN}}$ and $\text{Fork}_{\text{NLIN}}$, its KL increases (0.068 vs. 0.023 and 0.13 vs. 0.027), but it stays below KPI, DiffPuter, and O-MIRACLE on every task. The gap can be explained by how much of the marginal likelihood requires Monte Carlo marginalization. In Chain, coverage holds requires $p(x_2, x_3)$, which integrates over x_1, on half of the data. The full-positivity reference requires $p(x_1, x_3)$, which integrates over x_2, only on a quarter of the data.

Under *Family positivity violated*, the KL of MissCNF rises overall as expected. We further locate the error in Figure 7 (Appendix C.2.1) and confirm the error lies inside Z_{O_1} and stays close to zero outside it, which matches our claim in Theorem 1.

4.3.2 MNAR Missingness

Finally, we consider self-masking MNAR missingness at missing rates of 30% and 60%, where the probability that a variable is missing may depend on its own value, making the full-data distribu-

Method	$\text{Chain}_{\text{LIN}}$	Fork_{LIN}	$\text{Collider}_{\text{LIN}}$	$\text{Triangle}_{\text{LIN}}$	$\text{Chain}_{\text{NLIN}}$	$\text{Fork}_{\text{NLIN}}$	$\text{Collider}_{\text{NLIN}}$	$\text{Triangle}_{\text{NLIN}}$
MICE	**0.058**±0.046	**0.030**±0.010	**0.110**±0.080	**0.014**±0.010	0.931±0.279	19.116±0.236	0.251±0.146	16.060±0.589
KPI	0.685±0.220	0.997±0.158	0.731±0.349	0.650±0.268	0.687±0.301	1.574±0.136	0.785±0.399	5.596±0.778
O-MIRACLE	3.808±4.190	1.011±0.386	0.956±0.399	3.936±4.829	2.749±2.780	1.700±0.423	0.921±0.372	3.839±2.393
MissCNF	0.083±0.052	0.048±0.025	0.117±0.094	0.031±0.016	**0.121**±0.042	**0.066**±0.030	**0.177**±0.127	**0.288**±0.280

Table 1: KL to the ground-truth distribution under self-masking MNAR at 60% missingness. Mean ± std over 5 seeds. Lower is better. Bold = best, underline = second-best per task.

tion not generally identified from the observed data. Table 1 reports KL at 60%; full results are in Appendix C.2.2. Despite the absence of a general identification guarantee, the linear–nonlinear crossover observed under MCAR and MAR remains clear. On the nonlinear SCMs, MissCNF achieves the lowest KL in all 8 combinations of task and missing rate. On the linear SCMs, MICE is best in all 8 settings, and MissCNF is second-best in each. MICE degrades sharply on nonlinear Fork and Triangle (KL above 16 at 60%), consistent with the misspecification failure of non-causal-aware imputers. The experiment shows that MissCNF remains empirically robust to these MNAR perturbations although no guarantee holds in this regime.

4.4 Monte Carlo Approximation and Computational Cost

We study the effect of the number of Monte Carlo samples used to estimate the observed-data marginal likelihood on $\text{Triangle}_{\text{NLIN}}$ under MAR and a missing rate of 60%. The running time is nearly constant up to $K = 128$, where KL plateaus, and grows linearly in K beyond that (Figure 5, complete results in Appendix C.3). The linear growth in K is consistent with the $\mathcal{O}(KdL)$ per-step cost of Section 3. Peak memory reaches 10 GB at $d = 64$, which fits well on a single GPU.

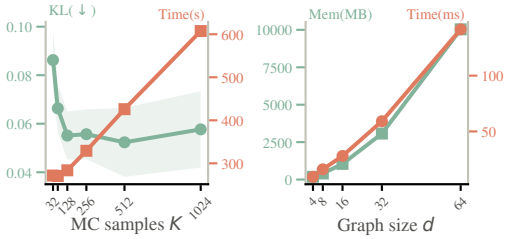


Figure 5: Scaling of MC samples and compute.

5 Conclusions and Limitations

We introduced MissCNF, which trains causal normalizing flows directly on incomplete data. Unlike listwise deletion and impute-then-fit pipelines, MissCNF maximizes the observed-data likelihood of every sample without constructing any completed dataset. This can be done thanks to the autoregressive structure of CNFs: missing variables are integrated out by Monte Carlo, and only those in the ancestral closure of the observed set require sampling at all.

We analyse two concepts which are often conflated: ignorability and identifiability. Ignorability under MAR establishes that the observed-data likelihood is the right objective to maximize. The central contribution of our work is establishing the conditions for identifiability, which is where causal structure pays off. We show that recovering the full joint requires only that each variable be observed across its support together with its parents in some pattern, a condition that can hold even when no record in the dataset is ever complete. Where this condition fails, our theory identifies which region is unrecoverable.

Empirically, MissCNF performs best where causal normalizing flows are motivated in the first place. On nonlinear SCMs it consistently achieves the lowest symmetric KL divergence, including under self-masking MNAR, where no identification guarantee applies. On linear SCMs, however, a strong imputer can recover the relevant conditionals accurately. Finally, because the objective relies only on the autoregressive structure, our method can be easily extended to CNF extensions such as DeCaFlow (Almodóvar et al., 2025), which accounts for latent confounding, and TSCNF (Yeom et al., 2026), which handles longitudinal data.

Limitations. We evaluate on synthetic data since the metrics require a known ground-truth joint distribution and known interventional and counterfactual outcomes. Validation on real-world data will be an important next step. The Monte Carlo likelihood estimate becomes expensive as the number of missing ancestral variables grows. Finally, MissCNF inherits the assumptions of CNFs: a known causal graph and no unobserved confounding. Combining MissCNF with DeCaFlow is a natural way to remove the second assumption, and we leave this to future work.

6 AI DISCLOSURE

In this work, we used generative AI tools to formalize mathematical claims, contribute ingredients to proofs, assist in writing proofs, implement the method and experiments, and summarize experimental results. Given our initial algorithm, an LLM formalized why missing values can be integrated out and barren nodes can be ignored, which became Proposition 1. Given our initial proof of Theorem 1 with Z_O oversight, it identified the unidentifiable region Z_O, over which no claim can be made. It also helped write the proofs more formally. For implementation, it wrote code for our method, the baselines, and the experiments. We verified, cleaned up, and restructured this code manually. We did not use generative AI tools to generate synthetic datasets or to clean and reformat data. All SCMs are taken from prior work. Translation and qualitative data analysis do not apply to this work. Additionally, we used generative AI tools to plot figures, to search for and summarize related literature, and to improve the paper's readability. Given our idea and method, an LLM checked whether similar work existed and identified the literature on Full Information Maximum Likelihood. We also used it to find classical missing-data literature and state-of-the-art imputation methods. We read and checked all the literature and verified that there were no hallucinations and the summaries were correct. Given our initial draft, it gave feedback on wording and structure. We have reviewed all AI-assisted work. All code was verified and restructured, and all proofs were checked by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Alejandro Almodóvar, Adrián Javaloy, Juan Parras, Santiago Zazo, and Isabel Valera. Decaflow: A deconfounding causal generative model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=mOYGK7Hw9Y. (Page 9)

T. W. Anderson. Maximum likelihood estimates for a multivariate normal distribution when some observations are missing. *Journal of the American Statistical Association*, 52(278):200–203, 1957. ISSN 01621459. URL http://www.jstor.org/stable/2280845. (Page 1)

Jonathan W Bartlett, Shaun R Seaman, Ian R White, James R Carpenter, and Alzheimer's Disease Neuroimaging Initiative*. Multiple imputation of covariates by fully conditional specification: accommodating the substantive model. *Statistical methods in medical research*, 24(4): 462–487, 2015. (Pages 18 and 19)

Stef Buuren. *Flexible Imputation of Missing Data, Second Edition*. Chapman & Hall/CRC, 2nd edition, 07 2018. ISBN 9780429492259. doi: 10.1201/9780429492259. (Page 18)

Patrick Chao, Patrick Blöbaum, Sapan Kirit Patel, and Shiva Kasiviswanathan. Modeling causal mechanisms with diffusion models for interventional and counterfactual queries. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=EDHQDsqiSe. (Pages 1 and 6)

Tianyu Du, Luca Melis, and Ting Wang. Remasker: Imputing tabular data with masked autoencoding. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KI9NqjLVDT. (Page 7)

Tomas Geffner, Javier Antoran, Adam Foster, Wenbo Gong, Chao Ma, Emre Kiciman, Amit Sharma, Angus Lamb, Martin Kukla, Nick Pawlowski, Agrin Hilmkil, Joel Jennings, Meyer Scetbon, Miltiadis Allamanis, and Cheng Zhang. Deep end-to-end causal inference. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=e6sqttxEGX. (Page 21)

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 881–889, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/germain15.html. (Page 2)

Daniel Jarrett, Bogdan C Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. HyperImpute: Generalized iterative imputation with automatic model selection. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9916–9937. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/jarrett22a.html. (Pages 22 and 23)

Adrián Javaloy, Pablo Sanchez Martin, and Isabel Valera. Causal normalizing flows: from theory to practice. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=QIFoCI7ca1. (Pages 1, 2, 4, 6, 7, 19, 20, and 23)

Trent Kyono, Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. MIRACLE: Causally-aware imputation via learning missing data mechanisms. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=GzeqcAUFGl0. (Pages 7 and 22)

Sik-Yum Lee. Estimation for structural equation models with missing data. *Psychometrika*, 51(1): 93–99, 1986. doi: 10.1007/BF02294002. (Page 1)

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. (Page 7)

Xiao-Li Meng. Multiple-Imputation Inferences with Uncongenial Sources of Input. *Statistical Science*, 9(4):538 – 558, 1994. doi: 10.1214/ss/1177010269. URL https://doi.org/10.1214/ss/1177010269. (Page 19)

Bengt Muthén, David Kaplan, and Michael Hollis. On structural equation modeling with data that are not missing completely at random. *Psychometrika*, 52(3):431–462, 1987. doi: 10.1007/BF02294365. (Page 1)

Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. Missing data imputation using optimal transport. In *International Conference on Machine Learning*, pp. 7130–7140. PMLR, 2020. (Page 6)

Jeffrey Näf, Erwan Scornet, and Julie Josse. What is a good imputation under mar missingness?, 2026. URL https://arxiv.org/abs/2403.19196. (Page 2)

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf. (Page 2)

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL http://jmlr.org/papers/v22/19-1028.html. (Page 4)

Olivia Jullian Parra, Sara Zoccheddu, David Catalan Cerezo, Tom Forzy, Franziska Ulrich, William Sutcliffe, Jakob Martin Burgstaller, Oliver Senn, Patrick Owen, and Nicola Serra. Joint treatment effect estimation from incomplete healthcare data: Temporal causal normalizing flows with llm-driven evolutionary mnar imputation, 2026. URL https://arxiv.org/abs/2605.05125. (Page 1)

Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 857–869. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0987b8b338d6c90bbedd8631bc499221-Paper.pdf. (Page 1)

Judea Pearl. *The Logic of Structure-Based Counterfactuals*, pp. 201–258. Cambridge University Press, 2009. (Page 2)

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. (Page 23)

Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. (Pages 2, 3, and 4)

Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley, 1987. (Page 18)

Pablo Sánchez-Martin, Miriam Rateike, and Isabel Valera. VACA: Designing variational graph autoencoders for causal queries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8159–8168. AAAI Press, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/20789. (Pages 1, 6, and 19)

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Springer-Verlag, New York, 1993. (Page 2)

Daniel J. Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 01 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr597. URL https://doi.org/10.1093/bioinformatics/btr597. (Pages 7 and 22)

Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, 2011. doi: 10.18637/jss.v045.i03. URL https://www.jstatsoft.org/index.php/jss/article/view/v045i03. (Pages 7 and 22)

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2. (Page 21)

Hao Wang, zhengnan li, Zhichao Chen, Xu Chen, Shuting He, Guangyi Liu, Haoxuan Li, and Zhouchen Lin. Iterative missing data imputation with model form adaptation and non-missing feature supervision. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=L84DdFuvwV. (Pages 7 and 22)

Yoonseok Yeom, Jonghwan Kim, Taehui Yun, Juhyun Lyu, Jung-Hee Kim, Sangmin Lee, Jinseok Yang, Hyemin Jung, Woohyung Lim, and Sanghack Lee. Estimating interventional outcomes over time with causal normalizing flow. In *Forty-Second Annual Conference on Uncertainty in Artificial Intelligence*, 2026. URL https://openreview.net/forum?id=EuwsW9vL8S. (Page 9)

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. (Page 7)

Hengrui Zhang, Liancheng Fang, Qitian Wu, and Philip S. Yu. Diffputer: An EM-driven diffusion model for missing data imputation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3fl1SENSYO. (Pages 7, 22, and 23)

LEARNING CAUSAL NORMALIZING FLOWS FROM INCOMPLETE DATA VIA OBSERVED-DATA LIKELIHOOD

ADDITIONAL MATERIAL

A Proofs **15**

A.1 Marginal Likelihood and Training Objective 15

A.2 MAR: Ignorability and Identification 16

A.3 Recovery of Imputation Conditionals 18

A.4 Discussion of Imputation-Based Alternatives 18

B Experimental details **19**

B.1 Datasets 19

B.2 Missingness mechanisms 21

B.3 Imputation Baselines 22

B.4 Implementation 23

B.5 Metrics 23

C Full results **26**

C.1 Main result: Performance under standard missingness (Section 4.2) 26

C.2 Robustness to Weaker Recovery Conditions (Section 4.3) 29

C.3 Monte Carlo Approximation and Computational Cost (Section 4.4) 31

D Additional experiments **31**

D.1 Controlled Nonlinearity Study (mentioned in Section 4.2) 31

A PROOFS

A.1 MARGINAL LIKELIHOOD AND TRAINING OBJECTIVE

Proposition 1 The computation of $p_{\theta,O}(\mathbf{x}_O)$ involves only the variables in the ancestral closure $A = \mathrm{anc}(O)$; equivalently, only the missing ancestral coordinates $M' = A \setminus O$ need to be integrated out:

$$p_{\theta,O}(\mathbf{x}_O) = \mathbb{E}_{\mathbf{u}_{M'} \sim \prod_{i \in M'} p(u_i)}\big[g(\mathbf{u}_{M'})\big], \qquad g(\mathbf{u}_{M'}; \mathbf{x}_O) := \prod_{i \in O} p(u_i) \left| \frac{\partial \tau}{\partial x_i} \right|,$$

where, for each $i \in A$ processed in topological order, $u_i = \tau(x_i; h_i(\mathbf{x}_{\mathrm{pa}(i)}))$ if $i \in O$, and $x_i = \tau^{-1}(u_i; h_i(\mathbf{x}_{\mathrm{pa}(i)}))$ if $i \in M'$.

Proof of Proposition 1. Let $B := V \setminus A$ denote the barren nodes. We first show $\int \prod_{i \in B} p_\theta(x_i \mid \mathbf{x}_{\mathrm{pa}(i)})\, d\mathbf{x}_B = 1$ for any fixed $\mathbf{x}_{V \setminus B}$.

No node in B has a child in A: if $i \in B$ had a child $j \in A = \mathrm{anc}(O)$, then i would itself be an ancestor of some node in O, so $i \in A$, contradicting $i \in B$. Hence every child of a node in B is itself in B. Processing B in reverse topological order (children before parents), when we reach node $i \in B$, every factor $p_\theta(x_j \mid \mathbf{x}_{\mathrm{pa}(j)})$ where i is a parent of j has already been integrated out, so x_i appears only in its own factor:

$$\int p_\theta(x_i \mid \mathbf{x}_{\mathrm{pa}(i)})\, dx_i = \int p(u_i)\, du_i = 1,$$

regardless of $\mathbf{x}_{\mathrm{pa}(i)}$. Repeating over all of B removes the whole product.

Splitting the product over $V = A \sqcup B$ and integrating out x_B first (the integrand is a nonnegative density, so the order of integration is free),

$$\begin{aligned} p_{\theta,O}(\mathbf{x}_O) &= \int \left[\prod_{i \in A} p_\theta(x_i \mid \mathbf{x}_{\mathrm{pa}(i)})\right] \left(\int \prod_{i \in B} p_\theta(x_i \mid \mathbf{x}_{\mathrm{pa}(i)})\, d\mathbf{x}_B\right) d\mathbf{x}_{M'} \\ &= \int \prod_{i \in A} p_\theta(x_i \mid \mathbf{x}_{\mathrm{pa}(i)})\, d\mathbf{x}_{M'}. \end{aligned}$$

It remains to rewrite the integral over $\mathbf{x}_{M'}$ as an expectation over $\mathbf{u}_{M'}$. Since $\mathrm{pa}(i) \subseteq A$ for every $i \in A$, we process A in a topological order $\prec$, so that when node i is reached, every parent $\mathbf{x}_{\mathrm{pa}(i)}$ is already resolved either as observed data, or as a value already substituted via an earlier node's u_j, then:

- If $i \in M'$, x_i is an integration variable; since $\tau(\cdot\,; h_i(\mathbf{x}_{\mathrm{pa}(i)}))$ is a bijection, the change of variables $x_i = \tau^{-1}(u_i; h_i(\mathbf{x}_{\mathrm{pa}(i)}))$ cancels the density and measure Jacobians exactly, $p_\theta(x_i \mid \mathbf{x}_{\mathrm{pa}(i)})\, dx_i = p(u_i)\, du_i$.
- Otherwise, $i \in O$, x_i is instead fixed data, so we simply re-express its density via the forward change-of-variables formula, $p_\theta(x_i \mid \mathbf{x}_{\mathrm{pa}(i)}) = p(u_i)\,|\partial\tau/\partial x_i|$.

Applying this to every $i \in A$ in topological order and reassembling the product gives

$$p_{\theta,O}(\mathbf{x}_O) = \int \underbrace{\prod_{i \in O} p(u_i) \left| \frac{\partial \tau}{\partial x_i} \right|}_{g(\mathbf{u}_{M'}; \mathbf{x}_O)} \prod_{i \in M'} p(u_i)\, du_i = \mathbb{E}_{\mathbf{u}_{M'} \sim \prod_{i \in M'} p(u_i)}\big[g(\mathbf{u}_{M'}; \mathbf{x}_O)\big]. \quad \square$$

A.2 MAR: IGNORABILITY AND IDENTIFICATION

Lemma 1 Fix a pattern $O \subseteq V$ and let $R = \mathbf{1}_O$. For *any* θ, define

$$q_\theta(\mathbf{x}_O, R) := \int p_\theta(\mathbf{x}_O, \mathbf{x}_M)\, p_{\text{miss}}(R \mid \mathbf{x}_O, \mathbf{x}_M)\, d\mathbf{x}_M,$$

the observed-data density that would arise if $\mathbf{x}$ were generated by the candidate model p_θ while R is generated by the true, fixed mechanism p_{miss}. Under MAR, q_θ simplifies for *every* θ:

$$q_\theta(\mathbf{x}_O, R) = p_{\theta,O}(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O).$$

In particular, if the model is realizable ($p_{\theta_0} = p^*$ for some θ_0), then $p_{obs}(\mathbf{x}_O, R) = q_{\theta_0}(\mathbf{x}_O, R) = p^*_O(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O)$.

Proof of Lemma 1. Under MAR, $p_{\text{miss}}(R \mid \mathbf{x}) = p_{\text{miss}}(R \mid \mathbf{x}_O)$ does not depend on the variable of integration $\mathbf{x}_M$, so it may be pulled outside the integral defining q_θ:

$$\begin{aligned} q_\theta(\mathbf{x}_O, R) &= \int p_\theta(\mathbf{x}_O, \mathbf{x}_M)\, p_{\text{miss}}(R \mid \mathbf{x}_O)\, d\mathbf{x}_M \\ &= p_{\text{miss}}(R \mid \mathbf{x}_O) \int p_\theta(\mathbf{x}_O, \mathbf{x}_M)\, d\mathbf{x}_M \\ &= p_{\text{miss}}(R \mid \mathbf{x}_O)\, p_{\theta,O}(\mathbf{x}_O). \end{aligned}$$

For any missingness mechanism, substituting $p_\theta = p^*$ into the definition of q_θ gives

$$\begin{aligned} q_\theta(\mathbf{x}_O, R) &= \int p^*(\mathbf{x}_O, \mathbf{x}_M)\, p_{\text{miss}}(R \mid \mathbf{x}_O, \mathbf{x}_M)\, d\mathbf{x}_M \\ &= \int p_{joint}(\mathbf{x}_O, \mathbf{x}_M, R)\, d\mathbf{x}_M \\ &= p_{obs}(\mathbf{x}_O, R). \end{aligned}$$

matching the definition of p_{obs} in Section 2.

Combining both parts at $\theta = \theta_0$ with $p_{\theta_0} = p^*$ gives $q_{\theta_0}(\mathbf{x}_O, R) = p_{obs}(\mathbf{x}_O, R)$ and $q_{\theta_0}(\mathbf{x}_O, R) = p^*_O(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O)$, so $p_{obs}(\mathbf{x}_O, R) = p^*_O(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O)$. □

Theorem 1 Suppose MAR holds, and the CNF family is realizable ($p_{\theta_0} = p^*$ for some θ_0). Define $Z_O := \{\mathbf{x}_O : p_{\text{miss}}(\mathbf{1}_O \mid \mathbf{x}_O) = 0\}$ the set of values of $\mathbf{x}_O$ that pattern O never reveals. Then θ_0 is a global minimizer of $\mathcal{L}(\theta)$, and *every* global minimizer θ^* satisfies

$$p_{\theta^*,O}(\mathbf{x}_O) = p^*_O(\mathbf{x}_O) \quad \text{for } p^*_O\text{-almost every } \mathbf{x}_O \notin Z_O, \quad \text{simultaneously for every } O.$$

Proof of Theorem 1. By definition, we have $\mathcal{L}(\theta) = \mathbb{E}_{(\mathbf{x}_{O(\mathcal{R})},\mathcal{R})\sim p_{obs}}\big[-\log p_{\theta,O(\mathcal{R})}(\mathbf{X}_{O(\mathcal{R})})\big]$. Moreover, by Lemma 1, $p_{obs}(\mathbf{x}_O, R) = p^*_O(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O)$, so wherever $p_{obs}(\mathbf{x}_O, R) > 0$ we necessarily have $p_{\text{miss}}(R \mid \mathbf{x}_O) > 0$; thus $\log p_{\text{miss}}(R \mid \mathbf{x}_O)$ is finite p_{obs}-almost every $(\mathbf{x}_O, R)$. Therefore, for any θ, add and subtract $\log p_{\text{miss}}(R \mid \mathbf{x}_O)$ inside the expectation defining $\mathcal{L}(\theta) -$

$\mathcal{L}(\theta_0)$:

$$\begin{aligned}\mathcal{L}(\theta) - \mathcal{L}(\theta_0) &= \mathbb{E}_{p_{obs}}\Big[\log \frac{p_{\theta_0,O}(\mathbf{x}_O)}{p_{\theta,O}(\mathbf{x}_O)}\Big] \\ &= \mathbb{E}_{p_{obs}}\Big[\log \frac{p_{\theta_0,O}(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O)}{p_{\theta,O}(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O)}\Big] \\ &= \mathbb{E}_{p_{obs}}\Big[\log \frac{q_{\theta_0}(\mathbf{x}_O, R)}{q_{\theta}(\mathbf{x}_O, R)}\Big] \\ &= \mathbb{E}_{p_{obs}}\Big[\log \frac{p_{obs}(\mathbf{x}_O, R)}{q_{\theta}(\mathbf{x}_O, R)}\Big] \\ &= \mathrm{KL}\big(p_{obs} \,\big\|\, q_\theta\big) \;\geq\; 0,\end{aligned}$$

using $q_{\theta_0} = p_{obs}$ by Lemma 1 in the second-to-last step. Since $\mathcal{L}(\theta) \geq \mathcal{L}(\theta_0)$ for every θ, θ_0 is a global minimizer. Equality $\mathcal{L}(\theta^*) = \mathcal{L}(\theta_0)$ holds iff the KL term vanishes, i.e. $q_{\theta^*}(\mathbf{x}_O, R) = p_{obs}(\mathbf{x}_O, R)$ for p_{obs}-almost every $(\mathbf{x}_O, R)$, i.e.

$$p_{\theta^*,O}(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O) = p^*_O(\mathbf{x}_O)\, p_{\text{miss}}(R \mid \mathbf{x}_O) \qquad \text{for } p_{obs}\text{-almost every } (\mathbf{x}_O, R).$$

Wherever $p_{obs}(\mathbf{x}_O, R) > 0$ we have $p_{\text{miss}}(R \mid \mathbf{x}_O) > 0$, so dividing gives $p_{\theta^*,O}(\mathbf{x}_O) = p^*_O(\mathbf{x}_O)$ for p_{obs}-almost every $(\mathbf{x}_O, R)$, hence for p^*_O-almost every $\mathbf{x}_O$. Since Z_O is exactly the complement of $\{\mathbf{x}_O : p_{\text{miss}}(\mathbf{1}_O \mid \mathbf{x}_O) > 0\}$, this is equivalent to $p_{\theta^*,O}(\mathbf{x}_O) = p^*_O(\mathbf{x}_O)$ for p^*_O-almost every $\mathbf{x}_O \notin Z_O$, simultaneously for every O.

□

Corollary 1 Suppose p_{θ^*} and p^* both factorize according to G (Section 2). Under causal-family overlap (Definition 1) and $p^*_{\text{pa}(i)}(\mathbf{x}_{\text{pa}(i)}) > 0$ on the relevant domain for each i, every global minimizer θ^* of $\mathcal{L}$ satisfies $p_{\theta^*}(\mathbf{x}) = p^*(\mathbf{x})$ for p^*-almost every $\mathbf{x}$.

Proof of Corollary 1. Fix i and let $F_i := \{i\} \cup \text{pa}(i) \subseteq o_i$. By Theorem 1, $p_{\theta^*,o_i}(\mathbf{x}_{o_i}) = p^*_{o_i}(\mathbf{x}_{o_i})$ for $p^*_{o_i}$-almost every $\mathbf{x}_{o_i} \notin Z_{o_i}$; by hypothesis $p^*_{o_i}(Z_{o_i}) = 0$, so this equality holds for $p^*_{o_i}$-almost every $\mathbf{x}_{o_i}$. Both sides are nonnegative densities in $\mathbf{x}_{o_i}$, so marginalizing over $\mathbf{x}_{o_i \setminus F_i}$ preserves the almost every equality, giving

$$p_{\theta^*,F_i}(\mathbf{x}_{F_i}) = p^*_{F_i}(\mathbf{x}_{F_i}) \qquad \text{for } p^*_{F_i}\text{-almost every } \mathbf{x}_{F_i}.$$

Marginalizing once more over x_i gives

$$p_{\theta^*,\text{pa}(i)}(\mathbf{x}_{\text{pa}(i)}) = p^*_{\text{pa}(i)}(\mathbf{x}_{\text{pa}(i)}) \qquad \text{for } p^*_{\text{pa}(i)}\text{-almost every } \mathbf{x}_{\text{pa}(i)}.$$

Hence, wherever $p^*_{\text{pa}(i)}(\mathbf{x}_{\text{pa}(i)}) > 0$,

$$p_{\theta^*}(x_i \mid \mathbf{x}_{\text{pa}(i)}) = \frac{p_{\theta^*,F_i}(\mathbf{x}_{F_i})}{p_{\theta^*,\text{pa}(i)}(\mathbf{x}_{\text{pa}(i)})} = \frac{p^*_{F_i}(\mathbf{x}_{F_i})}{p^*_{\text{pa}(i)}(\mathbf{x}_{\text{pa}(i)})} = p^*(x_i \mid \mathbf{x}_{\text{pa}(i)})$$

for $p^*_{F_i}$-almost every $\mathbf{x}_{F_i}$. Let $N_i \subseteq \mathcal{X}_{F_i}$ denote the exceptional set, where $\mathcal{X}_{F_i}$ denote the sample space of $\mathbf{X}_{F_i}$, so $p^*_{F_i}(N_i) = 0$.

To combine these d statements for d variables, lift each N_i to the full sample space:

$$\widetilde{N}_i := \{\mathbf{x} \in \mathcal{X} : \mathbf{x}_{F_i} \in N_i\}.$$

Since $p^*_{F_i}(N_i) = 0$, the definition of the marginal gives $p^*(\widetilde{N}_i) = p^*_{F_i}(N_i) = 0$. As V is finite, $\widetilde{N} := \bigcup_{i=1}^d \widetilde{N}_i$ is also p^*-null. Off $\widetilde{N}$, every conditional equality holds simultaneously, so, using that both p_{θ^*} and p^* factorize according to G,

$$p_{\theta^*}(\mathbf{x}) = \prod_{i=1}^d p_{\theta^*}(x_i \mid \mathbf{x}_{\text{pa}(i)}) = \prod_{i=1}^d p^*(x_i \mid \mathbf{x}_{\text{pa}(i)}) = p^*(\mathbf{x})$$

for p^*-almost every $\mathbf{x}$. □

A.3 Recovery of Imputation Conditionals

Corollary 2 (Recovery of arbitrary imputation conditionals)**.** *Suppose the hypotheses of Corollary 1 hold. Then every global minimizer θ^* of $\mathcal{L}$ satisfies*

$$p_{\theta^*}(\mathbf{x}_B \mid \mathbf{x}_A) = p^*(\mathbf{x}_B \mid \mathbf{x}_A)$$

for any partition $V = A \sqcup B$, for p^*_A-almost every $\mathbf{x}_A$ for which the conditional is defined.

Proof of Corollary 2. By Corollary 1,

$$p_{\theta^*}(\mathbf{x}) = p^*(\mathbf{x})$$

for p^*-almost every $\mathbf{x}$. Fix any partition $V = A \sqcup B$. Marginalizing over $\mathbf{x}_B$ gives

$$p_{\theta^*,A}(\mathbf{x}_A) = p^*_A(\mathbf{x}_A)$$

for p^*_A-almost every $\mathbf{x}_A$. Wherever this common marginal is positive,

$$p_{\theta^*}(\mathbf{x}_B \mid \mathbf{x}_A) = \frac{p_{\theta^*}(\mathbf{x}_A, \mathbf{x}_B)}{p_{\theta^*,A}(\mathbf{x}_A)} = \frac{p^*(\mathbf{x}_A, \mathbf{x}_B)}{p^*_A(\mathbf{x}_A)} = p^*(\mathbf{x}_B \mid \mathbf{x}_A).$$

□

A.4 Discussion of Imputation-Based Alternatives

Single imputation and Multiple imputation. Single imputation replaces a conditional distribution by a single point estimate. Even the oracle conditional mean $\mu(\mathbf{x}_O) = \mathbb{E}_{p^*}[\mathbf{x}_M \mid \mathbf{x}_O]$ loses variation. The completed data can have the correct conditional mean and still the wrong joint distribution. This is a classical motivation for multiple imputation (Rubin, 1987).

A stochastic imputer such as MICE or a joint generative imputer can generate $m > 1$ completed datasets. Classical multiple imputation then fits the downstream analysis separately to each completed dataset and combines the resulting estimands and uncertainty estimates using Rubin-style rules (Rubin, 1987). In our setting, however, the downstream object is an entire generative model rather than a finite-dimensional estimand. Generating m completed datasets would therefore produce

$$p_{\theta^{(1)}}, \ldots, p_{\theta^{(m)}}.$$

To the best of our knowledge, there is no standard Rubin-style rule for combining these learned densities or their parameters. A natural predictive construction is the equally weighted mixture

$$\bar{p}(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^{m} p_{\theta^{(j)}}(\mathbf{x}),$$

equivalently sampling an index J uniformly and then sampling from $p_{\theta^{(J)}}$. This is a valid mixture distribution, but it is not the classical Rubin combination procedure and requires training, storing, and evaluating m downstream CNFs. Our main experiments therefore use $m = 1$ for all imputation baselines We leave the consideration of multiple imputation to future work.

Compatibility of fully conditional specification. The form of the imputation model matters independently of whether one or multiple completed datasets are generated. Fully conditional specification methods such as MICE specify a collection of conditional models, one for each partially observed variable. These conditionals need not, in general, correspond to any single coherent joint distribution; this is the classical compatibility problem for chained-equation imputation (Bartlett et al., 2015). When compatibility does hold, the resulting Gibbs-style procedure can often be interpreted as targeting a joint law (Buuren, 2018). Joint-model imputers, including diffusion-based or other generative imputers, do not suffer from this particular FCS incompatibility because they parameterize a joint distribution directly.

Congeniality between imputation and analysis models. When imputation and downstream learning remain two separately fitted stages, classical multiple-imputation theory distinguishes whether the imputation and analysis procedures are *congenial*: roughly, whether their probabilistic assumptions and target quantities can be embedded in a common coherent model (Meng, 1994; Bartlett et al., 2015). Lack of congeniality can invalidate downstream inference even when each stage appears reasonable in isolation. We do not rely on congeniality as a universal failure theorem for imputation methods; rather, it is an additional requirement introduced by fitting q and the downstream model separately.

B Experimental details

B.1 Datasets

B.1.1 Main experiment

We test our method on eight SCMs: four graphs (Chain, Triangle, Collider, and Fork), each in a linear and a nonlinear version. We take all SCMs from Javaloy et al. (2023) except $\text{Collider}_{\text{NLIN}}$, which they do not include. Therefore, we take it from Sánchez-Martin et al. (2022). All exogenous variables follow $\mathcal{N}(0, 1)$. Each SCM comes with a list of intervention variables, which is used to compute the metrics (Appendix B.5). We present their formulations below and their causal graphs in Figure 6.

$\text{Chain}_{\text{LIN}}$:

$$\begin{aligned} f_1(u_1) &= u_1 \\ f_2(x_1, u_2) &= 10 \cdot x_1 - u_2 \\ f_3(x_2, u_3) &= 0.25 \cdot x_2 + 2 \cdot u_3 \end{aligned}$$

$\text{Chain}_{\text{NLIN}}$:

$$\begin{aligned} f_1(u_1) &= u_1 \\ f_2(x_1, u_2) &= e^{x_1/2} + u_2/4 \\ f_3(x_2, u_3) &= \frac{(x_2 - 5)^3}{15} + u_3 \end{aligned}$$

$\text{Collider}_{\text{LIN}}$:

$$\begin{aligned} f_1(u_1) &= u_1 \\ f_2(u_2) &= 2 - u_2 \\ f_3(x_1, x_2, u_3) &= 0.25 \cdot x_2 - 0.5 \cdot x_1 + 0.5 \cdot u_3 \end{aligned}$$

$\text{Collider}_{\text{NLIN}}$:

$$\begin{aligned} f_1(u_1) &= u_1 \\ f_2(u_2) &= u_2 \\ f_3(x_1, x_2, u_3) &= 0.25 \cdot x_2^2 + 0.05 \cdot x_1 + 0.5 \cdot u_3 \end{aligned}$$

Fork_{LIN}:

$$\begin{aligned} f_1(u_1) &= u_1 \\ f_2(u_2) &= 2 - u_2 \\ f_3(x_1, x_2, u_3) &= 0.25 \cdot x_2 - 1.5 \cdot x_1 + 0.5 \cdot u_3 \\ f_4(x_3, u_4) &= x_3 + 0.25 \cdot u_4 \end{aligned}$$

Fork$_{\text{NLIN}}$:

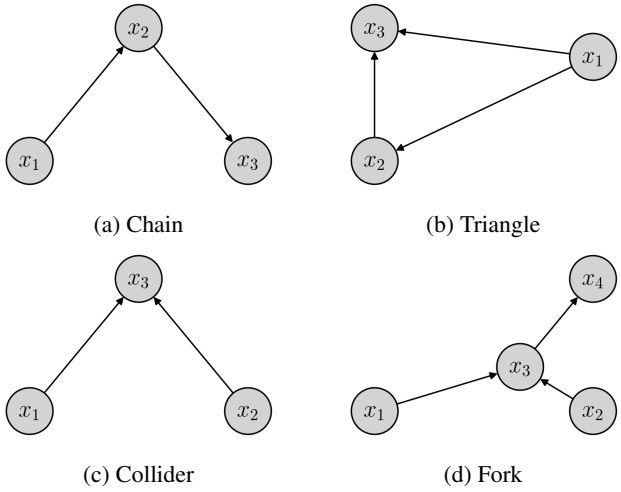
$$
\begin{aligned}
f_1(u_1) &= u_1 \\
f_2(u_2) &= u_2 \\
f_3(x_1, x_2, u_3) &= \frac{4}{1 + e^{-x_1 - x_2}} - x_2^2 + 0.5 \cdot u_3 \\
f_4(x_3, u_4) &= \frac{20}{1 + e^{0.5 \cdot x_3^2 - x_3}} + u_4
\end{aligned}
$$

Triangle$_{\text{LIN}}$:

$$
\begin{aligned}
f_1(u_1) &= u_1 + 1 \\
f_2(x_1, u_2) &= 10 \cdot x_1 - u_2 \\
f_3(x_1, x_2, u_3) &= 0.5 \cdot x_2 + x_1 + u_3
\end{aligned}
$$

Triangle$_{\text{NLIN}}$:

$$
\begin{aligned}
f_1(u_1) &= u_1 + 1 \\
f_2(x_1, u_2) &= 2 \cdot x_1^2 + u_2 \\
f_3(x_1, x_2, u_3) &= \frac{20}{1 + e^{-x_2^2 + x_1}} + u_3
\end{aligned}
$$



(a) Chain (b) Triangle

(c) Collider (d) Fork

Figure 6: Causal graph of the different SCMs considered in Section 4 (Javaloy et al., 2023).

.

B.1.2 Controlled nonlinearity study

We modify Chain and Collider SCMs by introducing α to control the nonlinearity of the SCMs:

Chain$_{\alpha}$:

$$
\begin{aligned}
f_1(u_1) &= u_1 \\
f_2(x_1, u_2) &= 0.5x_1 + 0.5u_2 \\
f_3(x_2, u_3) &= h_\alpha(x_2) + 0.5u_3,
\end{aligned}
$$

Collider$_{\alpha}$:

$$
\begin{aligned}
f_1(u_1) &= u_1 \\
f_2(u_2) &= u_2 \\
f_3(x_1, x_2, u_3) &= 0.5h_\alpha(x_1) + 0.5h_\alpha(x_2) + 0.5u_3,
\end{aligned}
$$

where

$$h_\alpha(x) = \frac{(1-\alpha)x + \alpha(x^2-1)/\sqrt{2}}{\sqrt{(1-\alpha)^2+\alpha^2}}, \qquad \alpha \in [0,1].$$

At $\alpha = 0$, $h_\alpha(x) = x$ and the SCMs are exactly linear, while at $\alpha = 1$, $h_\alpha(x) = (x^2-1)/\sqrt{2}$ and the relationship is purely quadratic. We evaluate $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$. The results of Chain_α and Collider_α are in Appendix D.1.

B.1.3 Larger graph

We test scalability on Chain_d, an extension of the Chain graph to d nodes. The SCM has the following form:

Chain_d:

$$f_1(u_1) = u_1$$
$$f_{i+1}(x_i, u_i) = g_i(x_i) + u_i \quad \text{for} \quad i \in \{i, ..., d-1\}$$

where we design g_i as a random spline function, following Geffner et al. (2024). We place 9 equally spaced knots on $[-3, 3]$ (8 intervals), sample the knot values from $\mathcal{N}(0, 1)$, and fit a natural cubic spline g_i using Scipy (Virtanen et al., 2020). Input x_i is clipped to $[-3, 3]$ to avoid extrapolation and

$$g_i(x_i) = g_i(\text{clip}(x_i)).$$

B.2 Missingness mechanisms

All the missing mask is generated based on complete data without any missing entries.

B.2.1 Main experiment (Full-pattern positivity)

Root variables are always observed. Therefore, missingness is applied only to each non-root variable x_i with a target rate $r \in \{30\%, 60\% 90\%\}$ for MCAR, MAR, and $r \in \{30\%, 60\%\}$ for Self-masking MNAR.

MCAR Each entry of x_i is missing independently with probability $p_{\text{miss}}(x_i) = r$.

MAR The missingness of x_i depends on a conditioning set $\text{cond}(i)$, which contains the non-root parents of x_i. If x_i has none, $\text{cond}(i)$ contains all other non-root variables. Therefore, $\text{cond}(i)$ is always observed, which satisfies the definition of MAR. We set

$$p_{\text{miss}}(x_i) = \sigma\Big(\sum_{j \in \text{cond}(i)} w_j x_j + b_i\Big),$$

where σ is the sigmoid function and $w_j \sim \mathcal{U}[0.1, 1.1]$. We find b_i by bisection so that the mean of $p_{\text{miss}}(x_i)$ over the data is within 10^{-3} of r. Therefore, the missing probability of each non-root node is independent.

Self-masking MNAR The missingness of x_i depends on its own value:

$$p_{\text{miss}}(x_i) = \sigma(w_i x_i + b_i), \tag{1}$$

with $w_i \sim \mathcal{U}[0.1, 1.1]$ and b_i found by bisection as above.

B.2.2 Pattern-based MAR

For the experiment in Section 4.3, we use Chain and Fork with two observation patterns, o_1 and o_2. Each sample is observed under exactly one pattern. Let x_{cut} be the node shared by both patterns, and let $\bar{x}_{\text{cut}} = (x_{\text{cut}} - \mu)/s$, where μ and s are the mean and standard deviation of x_{cut} under p^*.

- **Chain** ($x_1 \to x_2 \to x_3$): $o_1 = \{x_1, x_2\}$, $o_2 = \{x_2, x_3\}$, and $x_{\text{cut}} = x_2$.
- **Fork** ($x_1, x_2 \to x_3 \to x_4$): $o_1 = \{x_1, x_2, x_3\}$, $o_2 = \{x_3, x_4\}$, and $x_{\text{cut}} = x_3$.

Coverage holds. We set $p(o_1 \mid \mathbf{x}) = \sigma(\bar{x}_{\text{cut}})$ and $p(o_2 \mid \mathbf{x}) = 1 - \sigma(\bar{x}_{\text{cut}})$. Since σ takes values in $(0, 1)$, both patterns have positive probability at every value of x_{cut}. In Chain, the causal families $\{x_1\}$ and $\{x_1, x_2\}$ are contained in o_1, and $\{x_2, x_3\}$ equals o_2. In Fork, the families $\{x_1\}$, $\{x_2\}$ and $\{x_1, x_2, x_3\}$ are contained in o_1, and $\{x_3, x_4\}$ equals o_2. Causal-family coverage therefore holds, and p^* is identifiable.

Coverage violated. Let τ be the median of x_{cut}. We set

$$p(o_1 \mid \mathbf{x}) = \sigma(\bar{x}_{\text{cut}})\, \mathbf{1}\{x_{\text{cut}} \geq \tau\}, \qquad p(o_2 \mid \mathbf{x}) = 1 - p(o_1 \mid \mathbf{x}).$$

Pattern o_1 is never observed on $Z_{o_1} = \{\mathbf{x} : x_{\text{cut}} < \tau\}$, and $p^*(Z_{o_1}) = 1/2$. The families contained only in o_1 lose coverage on this region.

B.2.3 Computational scalability

For Chain$_d$, we use an alternating mask over consecutive nodes. Each sample follows one of two patterns, chosen with probability 0.5: [missing, observed, missing, observed, ...] or [observed, missing, observed, missing, ...].

B.3 Imputation Baselines

Mean imputation replaces each missing value with the empirical mean of its variable over the observed entries.

MICE (van Buuren & Groothuis-Oudshoorn, 2011) imputes each incomplete variable with a regression model conditioned on the other variables. It cycles through the variables for several rounds and draws each missing value from the model's predictive distribution instead of using the point prediction. We use the implementation in HyperImpute (Jarrett et al., 2022), where each per-variable model is a Bayesian ridge regression, and average the imputations over several runs.

MissForest (Stekhoven & Bühlmann, 2012) also cycles through the variables like MICE, but uses random forests. It first fills missing values with the column means. Then, for each variable with missing values, it trains a random forest on the rows where that variable is observed, using the other variables as inputs, and predicts the missing entries. It repeats until the change between successive imputed datasets stops decreasing. Unlike MICE, it uses the forest's prediction directly with no random draw, so it returns a single imputation.

DiffPuter (Zhang et al., 2025) follows the EM algorithm and alternates between an M-step and an E-step. In the M-step, it trains a diffusion model on the current completed data to learn $p(x)$. In the E-step, it samples x_{mis} given x_{obs} from the learned model: the observed entries are replaced by noised versions of their values, and the reverse process is run on the missing entries. The average of $N = 10$ samples becomes the new imputation.

KPI (Wang et al., 2025) imputes iteratively with kernel ridge regression using a weighted mix of Gaussian kernels with different bandwidths. In each step, it picks a random feature as the target and uses the others as inputs. It takes two batches of rows, fits the kernel ridge regression on the first batch in closed form, and uses it to predict the target on the second batch. The prediction error is then used to update, by gradient descent, both the imputed values (treated as learnable parameters) and the kernel mixing weights.

MIRACLE (Kyono et al., 2021) refines an initial imputation iteratively. Given the current completed data, a network with two sub-networks outputs a refined imputation ($\mathbb{R}^d \to \mathbb{R}^d$) and the missingness probabilities ($\mathbb{R}^d \to [0, 1]^{d_S}$), where d is the number of variables and d_S is the number of variables with at least one missing value. The m-graph over the variables and the missingness indicators is encoded in the input-layer weights. Besides reconstructing the observed features and predicting the missingness indicators, the network is trained with an acyclicity penalty on this graph and a moment regularizer. The regularizer matches an inverse-propensity-weighted estimate of each feature's mean, computed with the predicted missingness probabilities, to the mean of the imputation network's predictions.

B.4 Implementation

B.4.1 Imputation baselines

Mean, MICE, and MissForest. We use the corresponding plugins of the `hyperimpute` library (Jarrett et al., 2022). Mean imputation uses `SimpleImputer` from `scikit-learn` (Pedregosa et al., 2011). MICE uses `IterativeImputer` from `scikit-learn` with a `BayesianRidge` estimator and posterior sampling (`max_iter` $= 100$, one imputation, mean initialization, ascending imputation order). MissForest uses random forest regressors (`n_estimators` $= 10$, `max_iter` $= 100$).

DiffPuter. We use the authors' code at `https://github.com/hengruizhang98/DiffPuter` and follow their EM training loop and configuration: hidden width 1024, 50 diffusion steps, 20 RePaint resampling steps, batch size 4096, learning rate 10^{-4}, and up to 10,001 epochs per M-step with early stopping (patience 500) and a plateau learning-rate schedule. However, we use 5 EM rounds and 10 imputation trials instead of 10 and 20, to reduce computational cost. Zhang et al. (2025) report in their ablation that 5 rounds and 10 trials are sufficient.

KPI. We use the authors' code at `https://github.com/FMLYD/kpi` with a configuration based on their default settings and running scripts: 500 epochs, 2 batch pairs per step, batch size 128, early-stopping patience 10, ridge $\lambda = 1$, kernel bandwidths $\{0.01, 0.1, 5, 10, 100, 1000, 10^4\}$, and learning rate 10^{-2}.

O-MIRACLE. We use the MIRACLE authors' code at `https://github.com/vanderschaarlab/MIRACLE`. MIRACLE normally learns a causal graph over an augmented node set (the features plus one missingness indicator per partially observed feature), using a causal regularizer. Instead, we provide the true structure through its per-node input mask and call it **O-MIRACLE** to make the difference clear. The parents of each feature come from the SCM adjacency, and the parents of each indicator come from the conditioning set of the true missingness mechanism (Appendix B.2). We drop the causal regularizer and leave all other components unchanged. This gives O-MIRACLE information of both the causal graph and missing mechanisms. The remaining settings follow the authors' example script: hidden width 32, `reg_lambda` $=$ `reg_m` $= 0.1$, window 10, 400 refinement steps, learning rate 10^{-4}, and batch size 32.

B.4.2 Causal normalizing flows

We use the causal normalizing flow implementation from the `causal-flows` library (Javaloy et al., 2023), available at `https://github.com/adrianjav/causal-flows`. The flow has a single masked autoregressive layer with an affine transform per dimension. A hyper-network maps x to a shift μ_i and a log-scale s_i for every dimension i, and the flow computes

$$u_i = \mu_i + \exp(s_i)\, x_i.$$

The hyper-network is a masked MLP with input size d, two hidden layers of 64 units with ReLU activations, and output size $2d$. Its masks follow the graph adjacency, so μ_i and s_i depend only on the parents of x_i. The base distribution is a standard normal. The number of autoregressive passes is set to the diameter of the causal graph, which is the smallest number consistent with its depth. All of these are the library defaults. We train with AdamW (learning rate 10^{-3}, weight decay 0), batch size 4096, and 1000 epochs. The learning rate follows `ReduceLROnPlateau` (mode `min`, factor 0.95, patience 60). Both the optimizer and the schedule follow the authors' implementation at `https://github.com/psanch21/causal-flows`. We use $N = 512$ Monte Carlo samples in all our experiments of MissCNF, unless stated otherwise. Each run uses a single NVIDIA RTX 4090 GPU (24 GB).

B.5 Metrics

We use the evaluation metrics of Javaloy et al. (2023), and add a conditional KL distance for the experiment in Section 4.3.1. All metrics are computed against the true SCM, and the test data are complete with n samples.

B.5.1 SYMMETRIC KULLBACK–LEIBLER DIVERGENCE

We report the symmetric **KL** divergence (KL)

$$D_J(p^*, p_\theta) = \mathrm{KL}(p^* \parallel p_\theta) + \mathrm{KL}(p_\theta \parallel p^*).$$

Both p^* and p_θ have closed-form densities, so we estimate each term by Monte Carlo without importance weighting:

$$\mathrm{KL}(p^* \parallel p_\theta) \approx \frac{1}{n}\sum_{i=1}^{n}\big[\log p^*(x_i) - \log p_\theta(x_i)\big], \qquad x_i \sim p^* \text{ (test set)},$$

$$\mathrm{KL}(p_\theta \parallel p^*) \approx \frac{1}{n}\sum_{i=1}^{n}\big[\log p_\theta(\tilde{x}_i) - \log p^*(\tilde{x}_i)\big], \qquad \tilde{x}_i \sim p_\theta \text{ (learned flow)}.$$

Each of the four log-density terms is clamped from below at -10^4 to avoid infinite or NaN values.

B.5.2 AVERAGE TREATMENT EFFECT (ATE)

Each SCM has a set of intervention variables:

- Chain: $\{x_1, x_2\}$
- Triangle: $\{x_1, x_2\}$
- Collider: $\{x_2\}$
- Fork: $\{x_2, x_3\}$

For each intervention variable x_i, we draw 5000 fresh samples from the true SCM and compute the 25th, 50th, and 75th percentiles p_{25}, p_{50}, p_{75} of x_i. This gives three pairs (a, b): (p_{25}, p_{50}), (p_{25}, p_{75}), and (p_{50}, p_{75}). For each pair, we draw 10,000 samples under $\mathrm{do}(x_i{=}a)$ and 10,000 under $\mathrm{do}(x_i{=}b)$, and estimate the ATE by Monte Carlo:

$$\widehat{\mathrm{ATE}}_{p^*} = \mathbb{E}\big[X \mid \mathrm{do}(x_i{=}b)\big] - \mathbb{E}\big[X \mid \mathrm{do}(x_i{=}a)\big].$$

$\widehat{\mathrm{ATE}}_{p_\theta}$ is computed in the same way from the learned flow. Both are vectors in $\mathbb{R}^d$. The error for one pair is

$$\textbf{RMSE}_{\textbf{ATE}}(i, a, b) = \big\|\widehat{\mathrm{ATE}}_{p^*} - \widehat{\mathrm{ATE}}_{p_\theta}\big\|_2,$$

and we report its average over all variables and pairs.

B.5.3 COUNTERFACTUAL (CF) PREDICTION

For each intervention variable x_i and each value $a \in \{p_{25}, p_{50}, p_{75}\}$ of its percentiles, we compute the counterfactual of every test sample under $\mathrm{do}(x_i{=}a)$, once with the true SCM and once with the flow and $X^{\mathrm{cf}}_{p_\theta}$. The error is

$$\textbf{RMSE}_{\textbf{CF}}(i, a) = \frac{1}{n}\sum_{k=1}^{n}\big\|\mathbf{x}^{\mathrm{cf}}_{p^*,k} - \mathbf{x}^{\mathrm{cf}}_{p_\theta,k}\big\|_2,$$

where $\mathbf{x}^{\mathrm{cf}}_{\cdot,k}$ is the counterfactual of the k-th of $n = 2500$ test samples. We report the average over all variables and values of a.

B.5.4 LOCAL KL ERROR

To locate where along x_{cut} the model error lies, we partition the range of x_{cut} into J bins $B_1, \ldots, B_J$ and write $\ell(\mathbf{x}) = \log p^*(\mathbf{x}) - \log p_\theta(\mathbf{x})$. The local error in bin B_j is the mean of ℓ over samples that fall in the bin,

$$D_j = \mathbb{E}_{p^*}\left[\ell(\mathbf{X}) \mid x_{\text{cut}} \in B_j\right].$$

Unlike a contribution to the global KL, D_j is not weighted by the probability of the bin, so bins with little mass, such as the tails of x_{cut}, are not shrunk toward zero. Let P_j and Q_j be the probabilities of B_j under p^* and p_θ, and let p_j^* and $p_{\theta,j}$ be the two distributions conditioned on B_j. Since $p_j^*(\mathbf{x}) = p^*(\mathbf{x})/P_j$ and $p_{\theta,j}(\mathbf{x}) = p_\theta(\mathbf{x})/Q_j$ on B_j,

$$D_j = \mathrm{KL}\big(p_j^* \,\|\, p_{\theta,j}\big) + \log \frac{P_j}{Q_j}.$$

The first term measures how well the model fits the shape of the distribution inside the bin, and the second whether it assigns the bin the right total mass. D_j can be negative, but only when p_θ assigns more mass to B_j than p^* does. A correct model gives $D_j = 0$ in every bin, so values below zero do not indicate a better fit. Weighting by bin probability recovers the global KL, $\sum_j P_j D_j = \mathrm{KL}(p^* \,\|\, p_\theta)$, when the bins cover the whole range.

We use $J = 16$ bins of equal width between the 1% and 99% quantiles of x_{cut} in the test set. We estimate D_j by $\hat{D}_j = \frac{1}{m_j} \sum_{\mathbf{x} \in B_j} \ell(\mathbf{x})$, where the sum is over the m_j complete test samples with $x_{\text{cut}} \in B_j$. Since m_j is smaller near the edges of the range, $\hat{D}_j$ is noisier there.

C FULL RESULTS

C.1 MAIN RESULT: PERFORMANCE UNDER STANDARD MISSINGNESS (SECTION 4.2)

Method	$\text{Chain}_{\text{LIN}}$	Fork_{LIN}	$\text{Collider}_{\text{LIN}}$	$\text{Triangle}_{\text{LIN}}$	$\text{Chain}_{\text{NLIN}}$	$\text{Fork}_{\text{NLIN}}$	$\text{Collider}_{\text{NLIN}}$	$\text{Triangle}_{\text{NLIN}}$
Full data (reference)	*0.005±0.003*	*0.010±0.004*	*0.003±0.002*	*0.007±0.005*	*0.002±0.002*	*0.013±0.003*	*0.004±0.001*	*0.015±0.007*
MCAR, 30%								
Listwise deletion	0.011±0.005	0.018±0.006	0.003±0.003	0.013±0.007	0.005±0.004	0.029±0.004	**0.004**±0.003	0.029±0.014
Mean	13.642±0.373	5.750±0.176	0.158±0.011	19.878±0.948	0.963±0.041	4.051±0.132	0.078±0.013	3.793±0.071
MissForest	0.688±0.075	0.582±0.091	0.081±0.008	0.266±0.032	0.166±0.010	1.830±0.127	0.075±0.013	0.845±0.062
MICE	**0.006**±0.005	**0.009**±0.002	**0.003**±0.002	0.010±0.007	0.177±0.018	9.056±0.181	0.016±0.002	8.357±0.287
KPI	0.070±0.006	0.365±0.016	0.059±0.008	0.085±0.008	0.066±0.008	0.520±0.026	0.057±0.008	2.765±0.073
DiffPuter	0.119±0.009	0.218±0.036	0.064±0.011	0.156±0.010	0.141±0.012	0.156±0.029	0.061±0.012	0.987±0.080
O-MIRACLE	0.138±0.008	0.155±0.014	0.076±0.013	0.118±0.006	0.096±0.004	0.241±0.022	0.077±0.013	0.820±0.090
MissCNF (ours)	0.008±0.004	0.012±0.005	0.003±0.002	**0.009**±0.005	**0.004**±0.001	**0.016**±0.006	0.005±0.001	**0.019**±0.006
MCAR, 60%								
Listwise deletion	0.050±0.042	0.065±0.021	0.008±0.004	0.037±0.014	0.037±0.033	0.092±0.012	0.009±0.003	0.098±0.009
Mean	28.783±0.474	8.912±0.060	0.689±0.056	37.877±2.459	2.696±0.069	9.027±0.224	0.490±0.044	12.014±3.386
MissForest	2.116±0.197	1.127±0.155	0.505±0.041	1.057±0.105	1.204±0.086	4.383±0.167	0.473±0.047	2.586±0.301
MICE	**0.004**±0.003	**0.010**±0.004	**0.004**±0.002	**0.007**±0.005	0.488±0.027	17.612±0.368	0.048±0.004	15.482±0.472
KPI	0.345±0.009	0.794±0.029	0.342±0.031	0.373±0.012	0.228±0.017	1.129±0.034	0.327±0.045	4.388±0.185
DiffPuter	1.024±0.082	1.270±0.088	0.410±0.063	1.097±0.132	0.966±0.135	1.333±0.227	0.410±0.059	2.784±0.318
O-MIRACLE	0.908±0.034	0.494±0.020	0.485±0.056	0.781±0.027	0.666±0.019	0.729±0.022	0.478±0.047	1.979±0.121
MissCNF (ours)	0.009±0.006	0.017±0.006	0.008±0.004	0.016±0.006	**0.006**±0.005	**0.029**±0.006	**0.007**±0.001	**0.032**±0.003
MCAR, 90%								
Listwise deletion	1.242±1.693	0.548±0.145	0.022±0.004	0.828±0.614	0.292±0.056	1.576±0.998	0.034±0.006	3.262±1.489
Mean	52.425±1.445	14.273±0.372	4.619±0.393	55.761±1.344	9.648±1.007	23.242±2.710	4.156±0.278	20.136±1.173
MissForest	13.484±1.503	4.037±0.367	4.592±0.274	6.727±0.704	14.147±3.702	9.635±0.375	4.283±0.383	19.151±7.998
MICE	**0.005**±0.003	**0.025**±0.012	**0.006**±0.005	**0.016**±0.011	0.956±0.125	24.716±0.285	0.101±0.002	21.659±0.612
KPI	1.331±0.067	1.976±0.134	2.389±0.143	1.408±0.040	0.872±0.067	2.814±0.353	2.622±0.222	5.428±0.266
DiffPuter	26.614±3.665	6.734±0.501	3.751±0.213	87.161±61.871	6.873±0.802	16.467±1.562	3.674±0.325	18.378±7.317
O-MIRACLE	8.285±0.371	4.581±0.111	4.245±0.426	7.567±0.480	6.569±0.088	6.460±0.553	4.311±0.419	8.825±0.811
MissCNF (ours)	0.042±0.020	0.103±0.084	0.019±0.009	0.162±0.034	**0.033**±0.003	**0.210**±0.181	**0.015**±0.001	**0.221**±0.034
MAR, 30%								
Listwise deletion	0.122±0.021	0.229±0.095	0.065±0.032	0.126±0.026	0.124±0.024	0.219±0.087	0.070±0.038	0.144±0.024
Mean	19.358±4.200	5.570±0.396	0.303±0.095	26.278±4.850	1.823±0.433	3.769±0.187	0.178±0.082	4.448±0.340
MissForest	1.506±0.729	0.752±0.105	0.220±0.101	0.819±0.472	0.397±0.182	2.791±0.328	0.170±0.080	1.447±0.327
MICE	**0.006**±0.004	**0.009**±0.004	**0.002**±0.002	**0.007**±0.004	0.306±0.121	10.785±0.873	0.044±0.011	10.417±1.726
KPI	0.114±0.029	0.391±0.037	0.139±0.055	0.123±0.023	0.104±0.036	0.653±0.062	0.118±0.041	2.938±0.115
DiffPuter	1.020±0.966	0.368±0.140	0.168±0.077	1.073±0.923	0.401±0.241	0.474±0.207	0.147±0.071	1.371±0.205
O-MIRACLE	0.194±0.020	0.211±0.041	0.189±0.086	0.166±0.019	0.196±0.042	0.318±0.051	0.150±0.059	1.027±0.155
MissCNF (ours)	0.014±0.008	0.015±0.008	0.003±0.002	0.014±0.007	**0.016**±0.021	**0.025**±0.010	**0.007**±0.002	**0.028**±0.010
MAR, 60%								
Listwise deletion	0.381±0.065	0.739±0.298	0.249±0.117	0.393±0.072	0.435±0.127	0.793±0.293	0.254±0.140	0.532±0.166
Mean	40.447±7.807	9.197±0.272	1.863±1.027	48.858±7.987	4.240±0.891	9.013±0.570	6.925±13.396	9.477±0.569
MissForest	3.883±1.328	1.861±0.509	1.641±0.952	2.997±1.325	2.343±0.874	5.227±0.566	1.592±1.571	4.450±0.820
MICE	**0.006**±0.003	**0.009**±0.002	**0.004**±0.004	**0.009**±0.006	0.751±0.256	21.217±1.883	0.101±0.026	19.927±2.473
KPI	0.604±0.270	1.075±0.235	0.897±0.413	0.655±0.285	0.450±0.201	1.642±0.435	1.090±0.833	4.855±0.529
DiffPuter	11.181±6.655	2.458±0.906	1.296±0.687	12.377±9.267	2.212±1.015	4.268±2.237	1.738±2.116	4.350±1.234
O-MIRACLE	3.690±4.407	1.326±0.772	1.263±0.671	3.578±4.510	1.613±0.954	1.435±0.726	1.198±0.688	2.878±0.615
MissCNF (ours)	0.029±0.017	0.033±0.013	0.008±0.002	0.034±0.016	**0.025**±0.022	**0.032**±0.011	**0.011**±0.001	**0.052**±0.032
MAR, 90%								
Listwise deletion	14.365±10.153	54.749±105.659	1.300±1.779	7.028±4.523	3.622±2.310	7.401±2.831	0.772±0.678	22.316±27.533
Mean	324.773±337.569	37.922±28.441	33.047±37.771	209.427±259.684	125.221±194.035	42.405±19.754	28.089±31.147	70.806±94.904
MissForest	18.373±3.435	14.614±9.701	15.488±11.749	18.650±9.691	18.934±1.970	14.088±4.933	24.085±21.509	69.188±27.696
MICE	**0.009**±0.009	**0.022**±0.010	**0.007**±0.005	**0.025**±0.023	1.272±0.352	34.704±5.341	0.210±0.064	27.042±4.654
KPI	2.153±0.658	4.550±2.090	7.601±5.810	2.262±0.652	1.463±0.377	5.347±2.216	10.629±5.841	7.058±0.758
DiffPuter	91.241±71.097	23.082±16.464	14.365±10.648	161.602±128.402	28.363±26.479	32.768±21.681	40.917±61.437	55.624±43.512
O-MIRACLE	41.293±28.772	23.727±20.749	12.479±9.174	41.247±27.602	15.721±5.929	22.174±9.637	13.622±8.549	16.275±6.147
MissCNF (ours)	0.059±0.019	0.108±0.052	0.023±0.005	0.243±0.132	**0.073**±0.027	**0.348**±0.114	**0.031**±0.017	**0.834**±0.989

Table 2: KL across all 8 SCMs, both missingness mechanisms and three missingness rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (mechanism × rate); the full-data reference row (model trained with no missingness) is excluded from best/second-best marking.

Method	$\text{Chain}_{\text{LIN}}$	Fork_{LIN}	$\text{Collider}_{\text{LIN}}$	$\text{Triangle}_{\text{LIN}}$	$\text{Chain}_{\text{NLIN}}$	$\text{Fork}_{\text{NLIN}}$	$\text{Collider}_{\text{NLIN}}$	$\text{Triangle}_{\text{NLIN}}$
Full data (reference)	*0.054±0.014*	*0.033±0.004*	*0.011±0.001*	*0.350±0.074*	*0.027±0.008*	*0.063±0.014*	*0.018±0.006*	*0.145±0.048*
MCAR, 30%								
Listwise deletion	0.089±0.035	0.035±0.006	0.013±0.002	0.333±0.052	0.040±0.011	0.083±0.030	**0.019**±0.007	0.219±0.091
Mean	1.778±0.043	0.299±0.010	0.067±0.004	3.809±0.311	0.599±0.016	0.772±0.048	0.027±0.006	1.354±0.189
MissForest	0.407±0.090	0.057±0.012	0.012±0.002	0.373±0.289	0.077±0.021	0.580±0.172	0.028±0.009	1.130±0.473
MICE	**0.070**±0.008	0.032±0.004	0.012±0.003	0.372±0.046	0.122±0.005	1.187±0.037	0.026±0.007	2.184±0.062
KPI	0.086±0.018	0.043±0.006	0.014±0.005	0.924±0.039	0.082±0.009	0.125±0.042	0.024±0.011	1.503±0.249
DiffPuter	0.078±0.027	**0.031**±0.004	0.013±0.003	**0.243**±0.143	0.045±0.014	0.111±0.036	0.020±0.009	0.643±0.373
O-MIRACLE	0.074±0.020	0.035±0.005	0.012±0.003	0.387±0.068	**0.033**±0.006	0.075±0.021	0.020±0.007	0.560±0.120
MissCNF (ours)	0.093±0.018	0.033±0.003	**0.012**±0.002	0.343±0.044	0.041±0.015	**0.069**±0.015	0.022±0.007	**0.164**±0.057
MCAR, 60%								
Listwise deletion	0.171±0.023	0.038±0.013	0.015±0.004	0.359±0.211	0.095±0.043	0.131±0.076	0.031±0.011	0.236±0.102
Mean	3.486±0.036	0.556±0.015	0.132±0.006	5.000±0.620	1.091±0.012	1.383±0.034	0.047±0.004	2.998±0.133
MissForest	0.369±0.159	0.083±0.019	0.027±0.011	4.374±4.999	0.164±0.030	0.826±0.142	0.049±0.010	1.953±1.493
MICE	**0.060**±0.024	0.033±0.004	**0.011**±0.004	**0.299**±0.104	0.190±0.023	1.826±0.053	0.045±0.006	3.039±0.035
KPI	0.111±0.024	0.051±0.015	0.018±0.007	1.518±0.048	0.125±0.031	0.212±0.034	0.033±0.010	2.625±0.112
DiffPuter	0.374±0.103	0.043±0.011	0.016±0.003	0.789±0.377	0.088±0.022	0.342±0.064	0.028±0.009	1.028±0.463
O-MIRACLE	0.067±0.012	0.035±0.006	0.013±0.005	0.403±0.118	**0.042**±0.012	0.111±0.017	**0.025**±0.015	0.816±0.159
MissCNF (ours)	0.093±0.022	**0.033**±0.004	0.013±0.004	0.419±0.092	0.051±0.019	**0.093**±0.026	0.031±0.010	**0.178**±0.057
MCAR, 90%								
Listwise deletion	0.534±0.269	0.079±0.011	0.031±0.016	1.283±0.424	0.139±0.052	0.421±0.097	0.064±0.025	1.083±0.281
Mean	5.118±0.032	0.816±0.004	0.204±0.002	6.840±0.057	1.465±0.013	1.956±0.044	0.069±0.002	4.469±0.032
MissForest	0.701±0.120	0.090±0.024	0.050±0.019	7.394±8.690	0.339±0.071	2.383±0.358	0.068±0.005	6.561±10.536
MICE	**0.083**±0.045	**0.037**±0.005	**0.015**±0.007	**0.458**±0.192	0.278±0.061	2.173±0.120	0.079±0.006	3.806±0.130
KPI	0.209±0.096	0.060±0.016	0.043±0.016	2.089±0.135	0.156±0.075	0.291±0.075	0.098±0.035	3.247±0.095
DiffPuter	2.837±0.355	0.178±0.042	0.126±0.015	52.412±63.952	0.523±0.073	1.544±0.069	0.040±0.009	2.638±0.899
O-MIRACLE	0.295±0.124	0.056±0.015	0.023±0.012	1.576±0.262	0.093±0.025	0.272±0.082	**0.037**±0.026	1.625±0.273
MissCNF (ours)	0.177±0.078	0.045±0.011	0.024±0.008	1.572±0.128	**0.088**±0.023	**0.230**±0.056	0.047±0.018	**0.404**±0.112
MAR, 30%								
Listwise deletion	0.075±0.013	0.033±0.004	0.014±0.002	0.354±0.109	0.039±0.013	0.287±0.108	0.019±0.010	0.494±0.119
Mean	1.836±0.047	0.253±0.028	0.052±0.010	3.779±0.392	0.586±0.026	0.910±0.034	0.024±0.006	1.754±0.113
MissForest	0.343±0.101	0.051±0.015	0.015±0.003	0.412±0.156	0.091±0.018	0.857±0.080	0.019±0.007	0.913±0.057
MICE	**0.053**±0.012	**0.032**±0.003	0.012±0.003	**0.300**±0.102	0.050±0.014	1.234±0.040	0.031±0.007	2.298±0.072
KPI	0.071±0.022	0.039±0.007	0.013±0.005	0.903±0.063	0.100±0.016	0.145±0.020	0.024±0.011	1.641±0.159
DiffPuter	0.089±0.040	0.038±0.014	0.017±0.008	0.374±0.329	0.057±0.018	0.157±0.063	0.023±0.010	0.551±0.134
O-MIRACLE	0.053±0.018	0.033±0.003	0.012±0.002	0.345±0.103	**0.033**±0.009	0.078±0.011	0.018±0.004	0.599±0.082
MissCNF (ours)	0.069±0.020	0.035±0.002	**0.012**±0.002	0.367±0.102	0.037±0.010	**0.062**±0.014	**0.016**±0.004	**0.153**±0.029
MAR, 60%								
Listwise deletion	0.152±0.071	0.044±0.012	0.015±0.007	0.436±0.210	0.074±0.028	0.528±0.213	0.031±0.019	0.881±0.180
Mean	3.768±0.216	0.517±0.030	0.133±0.010	5.242±0.532	1.117±0.015	1.373±0.043	0.047±0.011	3.135±0.345
MissForest	0.468±0.207	0.069±0.016	0.024±0.004	1.470±1.210	0.187±0.050	1.300±0.292	0.043±0.009	1.866±1.183
MICE	**0.063**±0.027	**0.035**±0.007	**0.011**±0.002	**0.297**±0.062	0.081±0.026	2.240±0.176	0.094±0.026	3.429±0.121
KPI	0.105±0.042	0.060±0.012	0.017±0.005	1.489±0.056	0.124±0.017	0.207±0.016	0.039±0.015	2.658±0.151
DiffPuter	1.078±0.469	0.114±0.062	0.015±0.005	2.010±1.467	0.255±0.140	0.718±0.264	0.022±0.010	1.604±0.175
O-MIRACLE	0.102±0.068	0.040±0.001	0.016±0.005	0.451±0.186	**0.034**±0.012	0.189±0.100	**0.018**±0.010	0.853±0.052
MissCNF (ours)	0.093±0.034	0.037±0.007	0.014±0.005	0.492±0.345	0.045±0.010	**0.083**±0.024	0.027±0.013	**0.190**±0.043
MAR, 90%								
Listwise deletion	0.704±0.571	0.087±0.047	0.032±0.014	1.340±0.510	0.242±0.050	1.299±0.449	0.046±0.020	1.871±0.691
Mean	5.333±0.136	0.811±0.023	0.206±0.005	7.013±0.103	1.471±0.017	1.803±0.032	0.075±0.003	4.602±0.055
MissForest	0.831±0.248	0.133±0.036	0.048±0.028	18.145±35.762	0.311±0.080	1.879±0.321	0.067±0.007	18.476±29.456
MICE	**0.095**±0.043	**0.041**±0.004	**0.013**±0.003	**0.444**±0.270	0.207±0.037	3.106±0.244	0.218±0.066	4.427±0.417
KPI	0.243±0.085	0.075±0.035	0.031±0.012	2.005±0.160	0.152±0.101	0.323±0.088	0.074±0.020	3.339±0.170
DiffPuter	4.345±0.619	0.342±0.228	0.117±0.021	12.606±12.593	0.878±0.104	1.673±0.273	0.068±0.018	4.628±0.731
O-MIRACLE	1.899±0.804	0.205±0.141	0.038±0.029	3.368±0.801	0.420±0.180	1.638±0.478	0.041±0.030	2.550±0.427
MissCNF (ours)	0.157±0.126	0.046±0.012	0.028±0.011	1.557±0.487	**0.124**±0.088	**0.175**±0.121	**0.040**±0.019	**0.460**±0.058

Table 3: RMSE_{ATE} across all 8 SCMs, both missingness mechanisms and three missingness rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (mechanism × rate); the full-data reference row (model trained with no missingness) is excluded from best/second-best marking.

Method	$\text{Chain}_{\text{LIN}}$	Fork_{LIN}	$\text{Collider}_{\text{LIN}}$	$\text{Triangle}_{\text{LIN}}$	$\text{Chain}_{\text{NLIN}}$	$\text{Fork}_{\text{NLIN}}$	$\text{Collider}_{\text{NLIN}}$	$\text{Triangle}_{\text{NLIN}}$
Full data (reference)	*0.067±0.012*	*0.022±0.002*	*0.016±0.001*	*0.268±0.037*	*0.036±0.002*	*0.101±0.011*	*0.034±0.002*	*0.198±0.037*
MCAR, 30%								
Listwise deletion	0.091±0.017	0.028±0.003	0.019±0.001	0.287±0.036	0.049±0.006	0.137±0.013	0.038±0.005	0.287±0.053
Mean	1.372±0.037	0.218±0.011	0.066±0.003	3.381±0.242	0.500±0.011	1.082±0.021	0.071±0.008	1.825±0.064
MissForest	0.390±0.044	0.073±0.006	0.029±0.005	0.456±0.202	0.121±0.014	0.971±0.086	0.050±0.009	1.381±0.307
MICE	**0.075**±0.010	0.022±0.001	**0.018**±0.003	**0.283**±0.027	0.140±0.004	1.526±0.038	0.063±0.010	2.178±0.042
KPI	0.087±0.017	0.033±0.004	0.026±0.006	0.911±0.043	0.108±0.008	0.360±0.019	0.047±0.008	1.789±0.102
DiffPuter	0.087±0.018	**0.022**±0.003	0.019±0.002	0.386±0.175	0.061±0.006	0.215±0.023	0.040±0.005	1.067±0.211
O-MIRACLE	0.077±0.014	0.022±0.002	0.018±0.002	0.296±0.022	0.057±0.005	0.239±0.008	**0.038**±0.004	0.784±0.056
MissCNF (ours)	0.094±0.014	0.023±0.002	0.020±0.001	0.310±0.054	**0.046**±0.008	**0.112**±0.014	0.038±0.003	**0.246**±0.043
MCAR, 60%								
Listwise deletion	0.192±0.029	0.051±0.007	0.026±0.005	0.354±0.109	0.103±0.025	0.236±0.038	0.054±0.007	0.457±0.053
Mean	2.406±0.032	0.389±0.031	0.113±0.002	4.084±0.268	0.808±0.010	1.710±0.035	0.133±0.008	2.985±0.095
MissForest	0.584±0.021	0.114±0.009	0.050±0.004	2.583±2.614	0.241±0.029	1.568±0.046	0.089±0.014	3.070±1.692
MICE	**0.072**±0.008	**0.022**±0.002	**0.019**±0.004	**0.250**±0.056	0.227±0.012	2.605±0.038	0.129±0.010	3.180±0.052
KPI	0.122±0.015	0.049±0.004	0.048±0.005	1.557±0.048	0.163±0.009	0.620±0.014	0.096±0.017	2.865±0.069
DiffPuter	0.341±0.053	0.043±0.004	0.026±0.004	1.054±0.309	0.134±0.016	0.603±0.030	0.057±0.014	2.110±0.372
O-MIRACLE	0.087±0.010	0.023±0.002	0.020±0.003	0.324±0.088	0.076±0.003	0.386±0.012	**0.046**±0.007	1.123±0.054
MissCNF (ours)	0.108±0.015	0.030±0.002	0.025±0.006	0.371±0.072	**0.065**±0.014	**0.146**±0.005	0.051±0.006	**0.275**±0.045
MCAR, 90%								
Listwise deletion	0.586±0.204	0.125±0.029	0.044±0.006	1.236±0.336	0.216±0.050	0.733±0.266	0.098±0.015	1.851±0.423
Mean	3.545±0.124	0.645±0.022	0.165±0.004	5.155±0.214	1.256±0.067	2.539±0.173	0.188±0.007	4.380±0.067
MissForest	0.893±0.076	0.160±0.006	0.078±0.013	4.031±3.657	0.431±0.084	2.240±0.126	0.126±0.011	27.034±52.249
MICE	**0.087**±0.032	**0.027**±0.006	**0.019**±0.005	**0.422**±0.122	0.326±0.040	3.386±0.046	0.203±0.006	4.069±0.069
KPI	0.245±0.028	0.083±0.006	0.123±0.011	2.298±0.068	0.225±0.034	0.809±0.048	0.277±0.017	3.466±0.064
DiffPuter	2.352±0.265	0.270±0.046	0.113±0.010	39.859±38.052	0.850±0.079	2.745±0.266	0.111±0.021	6.204±1.746
O-MIRACLE	0.297±0.086	0.042±0.007	0.033±0.006	1.522±0.298	0.143±0.010	0.642±0.069	**0.066**±0.004	2.219±0.229
MissCNF (ours)	0.206±0.028	0.051±0.003	0.040±0.006	1.633±0.120	**0.114**±0.012	**0.282**±0.045	0.072±0.011	**0.678**±0.159
MAR, 30%								
Listwise deletion	0.098±0.015	0.032±0.005	0.021±0.002	0.307±0.085	0.053±0.008	0.130±0.016	**0.041**±0.006	0.251±0.017
Mean	1.323±0.014	0.255±0.030	0.091±0.020	3.170±0.303	0.538±0.034	1.087±0.025	0.115±0.031	2.025±0.064
MissForest	0.474±0.029	0.085±0.006	0.055±0.018	0.482±0.072	0.157±0.011	1.115±0.065	0.096±0.029	1.302±0.057
MICE	**0.064**±0.005	**0.021**±0.001	**0.017**±0.002	**0.279**±0.067	0.134±0.011	1.543±0.034	0.089±0.015	2.338±0.067
KPI	0.123±0.022	0.047±0.008	0.049±0.014	0.946±0.077	0.113±0.008	0.402±0.018	0.095±0.016	1.895±0.056
DiffPuter	0.224±0.043	0.054±0.022	0.047±0.016	0.576±0.167	0.126±0.025	0.318±0.048	0.087±0.033	1.293±0.470
O-MIRACLE	0.154±0.044	0.040±0.012	0.048±0.014	0.321±0.064	0.097±0.021	0.271±0.026	0.085±0.030	0.937±0.130
MissCNF (ours)	0.082±0.005	0.023±0.001	0.019±0.002	0.327±0.083	**0.047**±0.007	**0.111**±0.009	0.043±0.005	**0.240**±0.062
MAR, 60%								
Listwise deletion	0.222±0.047	0.071±0.009	0.039±0.007	0.410±0.122	0.124±0.030	0.339±0.095	0.079±0.012	0.559±0.064
Mean	2.474±0.082	0.461±0.054	0.187±0.035	4.356±0.559	0.893±0.065	1.865±0.093	0.253±0.131	3.368±0.128
MissForest	0.799±0.147	0.145±0.019	0.113±0.035	1.277±0.592	0.326±0.032	1.853±0.156	0.196±0.086	2.624±0.613
MICE	**0.072**±0.012	**0.022**±0.002	**0.018**±0.004	**0.254**±0.060	0.224±0.024	2.677±0.065	0.165±0.026	3.532±0.083
KPI	0.185±0.028	0.081±0.018	0.096±0.024	1.570±0.045	0.178±0.010	0.631±0.033	0.210±0.060	2.958±0.097
DiffPuter	0.907±0.225	0.181±0.060	0.098±0.034	2.924±2.596	0.388±0.131	1.215±0.366	0.180±0.086	2.357±0.268
O-MIRACLE	0.345±0.084	0.083±0.035	0.088±0.029	0.585±0.287	0.206±0.042	0.451±0.085	0.167±0.068	1.387±0.141
MissCNF (ours)	0.111±0.009	0.033±0.006	0.026±0.005	0.465±0.284	**0.062**±0.012	**0.147**±0.004	**0.052**±0.006	**0.325**±0.055
MAR, 90%								
Listwise deletion	2.604±1.905	0.343±0.339	0.134±0.063	1.714±0.555	0.590±0.175	1.364±0.153	0.305±0.128	2.619±0.498
Mean	10.793±10.221	0.915±0.207	0.297±0.079	12.767±16.181	3.394±4.000	3.271±0.465	0.405±0.179	5.483±1.581
MissForest	1.160±0.220	0.231±0.049	0.160±0.038	12.064±21.237	0.478±0.038	2.316±0.174	0.357±0.150	30.084±23.658
MICE	**0.089**±0.017	**0.026**±0.002	**0.018**±0.004	**0.475**±0.219	0.320±0.034	3.679±0.093	0.287±0.049	4.571±0.210
KPI	0.287±0.038	0.131±0.022	0.178±0.030	2.138±0.097	0.237±0.039	0.896±0.087	0.428±0.101	3.646±0.052
DiffPuter	3.786±1.512	0.593±0.273	0.212±0.059	10.625±6.849	1.227±0.412	2.891±0.440	0.424±0.286	32.592±43.503
O-MIRACLE	1.567±0.632	0.324±0.233	0.139±0.055	2.872±0.645	0.684±0.150	1.823±0.387	0.281±0.109	2.987±0.368
MissCNF (ours)	0.206±0.046	0.058±0.008	0.041±0.007	1.683±0.478	**0.141**±0.044	**0.362**±0.065	**0.073**±0.012	**0.860**±0.253

Table 4: RMSE_{CF} across all 8 SCMs, both missingness mechanisms and three missingness rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (mechanism × rate); the full-data reference row (model trained with no missingness) is excluded from best/second-best marking.

C.2 Robustness to Weaker Recovery Conditions (Section 4.3)

C.2.1 From Causal-Family Coverage to Non-Identification (Section 4.3.1)

Task	Setting	MissCNF	MICE	O-MIRACLE	KPI	DiffPuter
$\text{Chain}_{\text{LIN}}$	Full pos.	0.023 ± 0.010	$\mathbf{0.007 \pm 0.005}$	1.27 ± 1.13	0.345 ± 0.102	5.20 ± 3.81
	Cov. hold	0.068 ± 0.023	$\mathbf{0.007 \pm 0.005}$	2.70 ± 0.113	0.338 ± 0.026	14.4 ± 0.438
	Cov. violated	0.385 ± 0.116	$\mathbf{0.005 \pm 0.004}$	439 ± 277	1309 ± 1082	1434 ± 116
$\text{Chain}_{\text{NLIN}}$	Full pos.	$\mathbf{0.023 \pm 0.026}$	0.610 ± 0.213	0.759 ± 0.284	0.290 ± 0.129	1.30 ± 0.688
	Cov. hold	$\mathbf{0.024 \pm 0.014}$	0.296 ± 0.021	1.63 ± 0.381	0.345 ± 0.047	4.64 ± 4.71
	Cov. violated	$\mathbf{0.034 \pm 0.017}$	0.703 ± 0.042	15.9 ± 0.530	14.0 ± 2.70	19.0 ± 1.61
Fork_{LIN}	Full pos.	0.025 ± 0.010	$\mathbf{0.010 \pm 0.004}$	0.670 ± 0.305	0.802 ± 0.114	1.36 ± 0.758
	Cov. hold	0.023 ± 0.007	$\mathbf{0.012 \pm 0.003}$	1.54 ± 0.081	0.463 ± 0.023	15.0 ± 1.12
	Cov. violated	0.134 ± 0.041	$\mathbf{0.015 \pm 0.005}$	74.9 ± 11.8	158 ± 53.5	134 ± 79.7
$\text{Fork}_{\text{NLIN}}$	Full pos.	$\mathbf{0.027 \pm 0.008}$	17.6 ± 1.38	0.769 ± 0.240	1.15 ± 0.130	2.03 ± 0.981
	Cov. hold	$\mathbf{0.130 \pm 0.071}$	11.4 ± 0.355	1.76 ± 0.050	1.72 ± 0.444	5.80 ± 0.184
	Cov. violated	$\mathbf{1.52 \pm 1.25}$	14.7 ± 1.20	136 ± 15.8	173 ± 66.9	135 ± 15.3

Table 5: KL across all 4 SCMs under 3 settings. Values are mean $\pm$ std over seeds. Lower is better. Bold = best, within each row.

Task	Setting	MissCNF	MICE	O-MIRACLE	KPI	DiffPuter
$\text{Chain}_{\text{LIN}}$	Full pos.	0.078 ± 0.035	$\mathbf{0.072 \pm 0.021}$	0.083 ± 0.029	0.102 ± 0.036	0.504 ± 0.234
	Cov. hold	0.117 ± 0.055	$\mathbf{0.057 \pm 0.030}$	0.111 ± 0.044	0.148 ± 0.065	1.44 ± 0.196
	Cov. violated	0.157 ± 0.057	$\mathbf{0.064 \pm 0.026}$	2.01 ± 0.096	29.1 ± 43.7	3.51 ± 0.248
$\text{Chain}_{\text{NLIN}}$	Full pos.	0.041 ± 0.011	0.069 ± 0.007	$\mathbf{0.034 \pm 0.006}$	0.134 ± 0.019	0.165 ± 0.197
	Cov. hold	$\mathbf{0.041 \pm 0.011}$	0.062 ± 0.013	0.067 ± 0.014	0.079 ± 0.005	0.537 ± 0.017
	Cov. violated	$\mathbf{0.042 \pm 0.018}$	0.176 ± 0.017	0.470 ± 0.021	0.468 ± 0.020	0.985 ± 0.028
Fork_{LIN}	Full pos.	0.037 ± 0.002	0.035 ± 0.004	$\mathbf{0.034 \pm 0.005}$	0.052 ± 0.007	0.073 ± 0.042
	Cov. hold	0.034 ± 0.002	0.034 ± 0.004	0.052 ± 0.008	$\mathbf{0.021 \pm 0.003}$	0.584 ± 0.049
	Cov. violated	0.042 ± 0.007	$\mathbf{0.032 \pm 0.003}$	0.051 ± 0.010	0.084 ± 0.018	0.526 ± 0.042
$\text{Fork}_{\text{NLIN}}$	Full pos.	$\mathbf{0.068 \pm 0.014}$	1.94 ± 0.094	0.120 ± 0.050	0.211 ± 0.014	0.580 ± 0.218
	Cov. hold	$\mathbf{0.063 \pm 0.018}$	1.97 ± 0.093	0.429 ± 0.040	0.686 ± 0.099	0.899 ± 0.046
	Cov. violated	0.366 ± 0.195	1.49 ± 0.072	0.731 ± 0.030	$\mathbf{0.345 \pm 0.077}$	0.438 ± 0.078

Table 6: RMSE_{ATE} across all 4 SCMs under 3 settings. Values are mean $\pm$ std over seeds. Lower is better. Bold = best, within each row.

Task	Setting	MissCNF	MICE	O-MIRACLE	KPI	DiffPuter
$\text{Chain}_{\text{LIN}}$	Full pos.	0.104 ± 0.006	$\mathbf{0.075 \pm 0.007}$	0.264 ± 0.058	0.166 ± 0.028	0.569 ± 0.129
	Cov. hold	0.133 ± 0.029	$\mathbf{0.068 \pm 0.010}$	0.472 ± 0.009	0.277 ± 0.014	1.40 ± 0.071
	Cov. violated	0.200 ± 0.030	$\mathbf{0.076 \pm 0.011}$	5.69 ± 1.62	33.4 ± 18.8	92.3 ± 71.5
$\text{Chain}_{\text{NLIN}}$	Full pos.	$\mathbf{0.057 \pm 0.004}$	0.207 ± 0.017	0.163 ± 0.033	0.163 ± 0.008	0.267 ± 0.129
	Cov. hold	$\mathbf{0.057 \pm 0.010}$	0.204 ± 0.016	0.258 ± 0.012	0.159 ± 0.012	0.707 ± 0.086
	Cov. violated	$\mathbf{0.058 \pm 0.011}$	0.296 ± 0.033	0.965 ± 0.034	0.813 ± 0.034	1.52 ± 0.135
Fork_{LIN}	Full pos.	0.029 ± 0.003	$\mathbf{0.022 \pm 0.002}$	0.063 ± 0.023	0.069 ± 0.016	0.121 ± 0.045
	Cov. hold	0.030 ± 0.003	$\mathbf{0.022 \pm 0.002}$	0.068 ± 0.006	0.062 ± 0.003	0.369 ± 0.011
	Cov. violated	0.064 ± 0.005	$\mathbf{0.020 \pm 0.003}$	0.180 ± 0.015	0.184 ± 0.044	0.478 ± 0.053
$\text{Fork}_{\text{NLIN}}$	Full pos.	$\mathbf{0.129 \pm 0.010}$	2.32 ± 0.038	0.366 ± 0.040	0.562 ± 0.030	0.824 ± 0.248
	Cov. hold	$\mathbf{0.153 \pm 0.017}$	1.97 ± 0.040	0.264 ± 0.018	0.424 ± 0.102	1.49 ± 0.011
	Cov. violated	$\mathbf{0.407 \pm 0.156}$	2.05 ± 0.067	1.93 ± 0.022	2.04 ± 0.193	2.39 ± 0.067

Table 7: RMSE_{CF} across all 4 SCMs under 3 settings. Values are mean $\pm$ std over seeds. Lower is better. Bold = best, within each row.

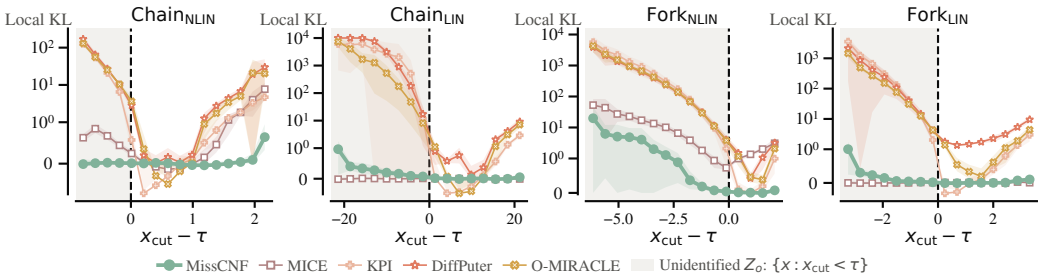


Figure 7: Local KL error (defined in Appendix B.5.4) as a function of the distance to the identification boundary. The dashed line denotes the boundary $x_{\text{rel}} = 0$. Negative values correspond to the revealed side, while positive values lie inside the non-identified region Z_o. Curves show mean ± std over seeds.

C.2.2 MNAR Missingness (Section 4.3.2)

Method	Chain$_{\text{LIN}}$	Fork$_{\text{LIN}}$	Collider$_{\text{LIN}}$	Triangle$_{\text{LIN}}$	Chain$_{\text{NLIN}}$	Fork$_{\text{NLIN}}$	Collider$_{\text{NLIN}}$	Triangle$_{\text{NLIN}}$
MNAR (self-masking), 30%								
MICE	**0.020**±0.012	**0.013**±0.004	**0.035**±0.026	**0.008**±0.006	0.446±0.204	9.367±0.115	0.087±0.050	9.131±0.391
KPI	0.132±0.026	0.383±0.031	0.131±0.056	0.121±0.024	0.261±0.149	0.678±0.047	0.149±0.070	2.951±0.308
O-MIRACLE	0.210±0.030	0.204±0.063	0.152±0.062	0.168±0.025	0.312±0.167	0.426±0.070	0.171±0.075	0.921±0.212
MissCNF (ours)	0.028±0.014	0.021±0.014	0.037±0.025	0.014±0.008	**0.054**±0.033	**0.022**±0.005	**0.060**±0.040	**0.078**±0.043
MNAR (self-masking), 60%								
MICE	**0.058**±0.046	**0.030**±0.010	**0.110**±0.080	**0.014**±0.010	0.931±0.279	19.116±0.236	0.251±0.146	16.060±0.589
KPI	0.685±0.220	0.997±0.158	0.731±0.349	0.650±0.268	0.687±0.301	1.574±0.136	0.785±0.399	5.596±0.778
O-MIRACLE	3.808±4.190	1.011±0.386	0.956±0.399	3.936±4.829	2.749±2.780	1.700±0.423	0.921±0.372	3.839±2.393
MissCNF (ours)	0.083±0.052	0.048±0.025	0.117±0.094	0.031±0.016	**0.121**±0.042	**0.066**±0.030	**0.177**±0.127	**0.288**±0.280

Table 8: KL across all 8 SCMs under self-masking MNAR at two missing rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (missing rate).

Method	Chain$_{\text{LIN}}$	Fork$_{\text{LIN}}$	Collider$_{\text{LIN}}$	Triangle$_{\text{LIN}}$	Chain$_{\text{NLIN}}$	Fork$_{\text{NLIN}}$	Collider$_{\text{NLIN}}$	Triangle$_{\text{NLIN}}$
MNAR (self-masking), 30%								
MICE	0.086±0.033	**0.034**±0.003	**0.014**±0.006	**0.312**±0.076	0.077±0.040	1.385±0.032	0.029±0.006	1.698±0.101
KPI	0.100±0.045	0.046±0.005	0.015±0.005	0.872±0.058	0.113±0.014	0.192±0.081	**0.024**±0.008	1.304±0.274
O-MIRACLE	**0.083**±0.045	0.035±0.004	0.014±0.006	0.349±0.097	0.054±0.024	0.163±0.062	0.025±0.004	0.477±0.116
MissCNF (ours)	0.086±0.037	0.035±0.004	0.015±0.006	0.359±0.117	**0.049**±0.015	**0.074**±0.032	0.025±0.012	**0.216**±0.092
MNAR (self-masking), 60%								
MICE	**0.098**±0.070	**0.035**±0.004	0.018±0.007	**0.241**±0.090	0.097±0.025	2.068±0.078	0.049±0.007	2.185±0.100
KPI	0.125±0.048	0.055±0.005	0.022±0.014	1.490±0.033	0.137±0.018	0.286±0.108	0.033±0.013	2.833±0.146
O-MIRACLE	0.140±0.062	0.046±0.010	**0.016**±0.010	0.498±0.223	**0.052**±0.026	0.417±0.196	0.027±0.006	0.770±0.203
MissCNF (ours)	0.106±0.064	0.042±0.013	0.019±0.016	0.440±0.296	0.076±0.026	**0.136**±0.033	**0.025**±0.010	**0.283**±0.073

Table 9: RMSE$_{\text{ATE}}$ across all 8 SCMs under self-masking MNAR at two missing rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (missing rate).

Method	$\text{Chain}_{\text{LIN}}$	Fork_{LIN}	$\text{Collider}_{\text{LIN}}$	$\text{Triangle}_{\text{LIN}}$	$\text{Chain}_{\text{NLIN}}$	$\text{Fork}_{\text{NLIN}}$	$\text{Collider}_{\text{NLIN}}$	$\text{Triangle}_{\text{NLIN}}$
MNAR (self-masking), 30%								
MICE	**0.090**±0.033	**0.022**±0.001	**0.019**±0.004	**0.268**±0.053	0.149±0.009	1.654±0.027	0.091±0.015	1.928±0.057
KPI	0.129±0.040	0.039±0.001	0.031±0.007	0.906±0.082	0.132±0.008	0.414±0.037	0.066±0.020	1.634±0.078
O-MIRACLE	0.146±0.048	0.029±0.004	0.025±0.004	0.330±0.056	0.104±0.019	0.325±0.031	0.053±0.011	0.920±0.190
MissCNF (ours)	0.101±0.034	0.025±0.001	0.023±0.003	0.328±0.096	**0.059**±0.003	**0.113**±0.013	**0.045**±0.009	**0.317**±0.086
MNAR (self-masking), 60%								
MICE	**0.107**±0.058	**0.022**±0.002	**0.022**±0.003	**0.240**±0.072	0.242±0.020	2.738±0.040	0.149±0.017	2.866±0.042
KPI	0.193±0.052	0.055±0.003	0.062±0.009	1.564±0.032	0.199±0.022	0.634±0.049	0.140±0.029	2.829±0.161
O-MIRACLE	0.326±0.081	0.043±0.010	0.039±0.014	0.632±0.307	0.200±0.033	0.567±0.069	0.075±0.030	1.314±0.120
MissCNF (ours)	0.134±0.051	0.032±0.003	0.031±0.011	0.437±0.244	**0.082**±0.015	**0.178**±0.032	**0.052**±0.010	**0.390**±0.054

Table 10: RMSE_{CF} across all 8 SCMs under self-masking MNAR at two missing rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (missing rate).

C.3 Monte Carlo Approximation and Computational Cost (Section 4.4)

	$\text{Fork}_{\text{NLIN}}$				$\text{Triangle}_{\text{NLIN}}$			
K	KL	RMSE_{ATE}	RMSE_{CF}	Time (s)	KL	RMSE_{ATE}	RMSE_{CF}	Time (s)
32	0.054±0.015	0.078±0.026	0.170±0.012	272±3	0.086±0.028	0.248±0.118	0.433±0.102	272±6
64	0.041±0.015	0.085±0.028	0.157±0.006	273±4	0.066±0.023	0.190±0.086	0.343±0.073	270±4
128	0.036±0.015	0.083±0.024	0.154±0.006	284±4	0.055±0.022	0.251±0.093	0.374±0.116	284±4
256	0.032±0.012	0.076±0.024	0.149±0.005	328±4	0.056±0.023	0.193±0.053	0.322±0.019	329±4
512	0.032±0.011	0.083±0.024	0.147±0.004	416±4	0.052±0.032	0.190±0.043	0.325±0.055	426±13
1024	0.034±0.011	0.082±0.021	0.150±0.005	602±21	0.058±0.035	0.181±0.059	0.328±0.045	607±9

Table 11: Monte Carlo ablation on $\text{Fork}_{\text{NLIN}}$ and $\text{Triangle}_{\text{NLIN}}$. K is the number of Monte Carlo samples used to estimate the observed-data marginal likelihood. Values are mean ± std over seeds. Time is total training time in seconds.

D Additional Experiments

D.1 Controlled Nonlinearity Study (mentioned in Section 4.2)

We construct a controlled SCM in which the degree of nonlinearity can be varied continuously while keeping the causal graph and noise distribution fixed. We use the Chain_α and Collider_α defined in Appendix B.1.2 where α controls the nonlinearity of the SCM. At $\alpha = 0$, the SCM is exactly linear, while at $\alpha = 1$, it is purely quadratic. We evaluate $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$. We consider MCAR and MAR with missing rates of 30% and 60%, using the same training setup and baselines as in the previous experiment, except we only run for 3 seeds. We exclude Mean imputation and MissForest due to their poor performance.

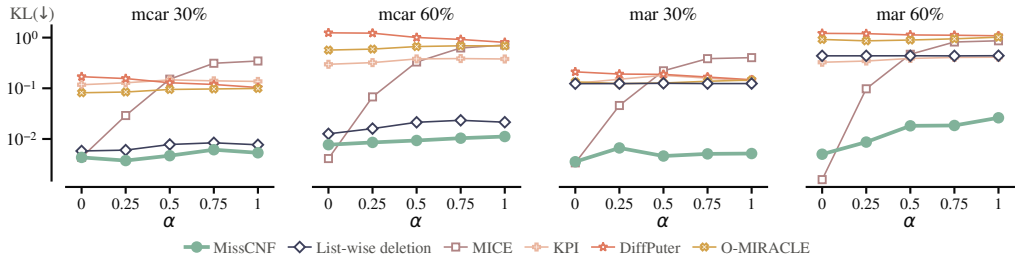


Figure 8: KL to the ground-truth distribution as α increases, on Chain_α under MCAR and MAR at 30% and 60% missing rates (mean ± std over seeds). α interpolates the $X_2 \to X_3$ mechanism from purely linear ($\alpha = 0$) to purely quadratic ($\alpha = 1$).

Figure 8 shows the KL divergence, while full metrics are reported in Tables 12, 13 and 14 (Appendix D.1). At $\alpha = 0$ (fully linear), MICE matches or outperforms MissCNF. Once nonlinearity

is introduced, its KL divergence increases rapidly, whereas MissCNF remains stable. Under MAR with 30% missingness, the ratio between the KL divergence of MICE and MissCNF increases from $0.95\times$ at $\alpha = 0$ to $6.9\times$, $48.0\times$, $75.6\times$, and $78.0\times$ at $\alpha = 0.25, 0.5, 0.75$, and 1, respectively. The same qualitative behavior is observed under MCAR and at the higher missing rate. This controlled experiment therefore further confirms the linear–nonlinear behaviour in the previous benchmark. The same crossover appears in ATE and RMSE_{CF}, though the gap is smaller (up to $3.2\times$ and $6.9\times$, respectively, at $\alpha = 1$) since these estimands depend only on specific functionals of the joint distribution rather than the full quantity. The remaining baselines (KPI, DiffPuter, O-MIRACLE) make no linearity assumption, and their KL divergence stays comparatively flat across α, albeit at a worse level. We show full results in the tables below, along with a similar study on Collider_α.

Method	$\alpha = 0.00$	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
MCAR, 30%					
Listwise deletion	0.006±0.002	0.006±0.001	0.008±0.002	0.008±0.003	0.008±0.001
MICE	0.004±0.001	0.029±0.004	0.151±0.013	0.312±0.021	0.345±0.021
KPI	0.118±0.012	0.128±0.014	0.146±0.014	0.140±0.014	0.137±0.016
DiffPuter	0.169±0.007	0.156±0.006	0.129±0.020	0.119±0.015	0.104±0.012
O-MIRACLE	0.082±0.004	0.084±0.006	0.095±0.011	0.097±0.011	0.099±0.008
MissCNF (ours)	**0.004**±0.001	**0.004**±0.001	**0.005**±0.002	**0.006**±0.003	**0.005**±0.002
MCAR, 60%					
Listwise deletion	0.013±0.001	0.016±0.001	0.021±0.004	0.023±0.002	0.021±0.004
MICE	**0.004**±0.002	0.068±0.003	0.330±0.020	0.624±0.023	0.714±0.020
KPI	0.297±0.025	0.321±0.022	0.380±0.027	0.386±0.016	0.378±0.009
DiffPuter	1.242±0.148	1.224±0.096	1.006±0.024	0.920±0.096	0.806±0.043
O-MIRACLE	0.567±0.015	0.593±0.016	0.666±0.041	0.689±0.042	0.686±0.026
MissCNF (ours)	0.008±0.002	**0.009**±0.002	**0.009**±0.002	**0.010**±0.002	**0.011**±0.002
MAR, 30%					
Listwise deletion	0.123±0.023	0.124±0.026	0.125±0.024	0.124±0.025	0.124±0.025
MICE	**0.003**±0.002	0.046±0.003	0.221±0.006	0.384±0.006	0.403±0.014
KPI	0.125±0.004	0.151±0.004	0.180±0.006	0.158±0.009	0.140±0.008
DiffPuter	0.211±0.043	0.191±0.047	0.188±0.044	0.167±0.028	0.147±0.031
O-MIRACLE	0.134±0.032	0.124±0.035	0.127±0.034	0.138±0.039	0.145±0.036
MissCNF (ours)	0.004±0.001	**0.007**±0.002	**0.005**±0.001	**0.005**±0.002	**0.005**±0.001
MAR, 60%					
Listwise deletion	0.437±0.166	0.437±0.167	0.438±0.166	0.436±0.168	0.439±0.166
MICE	**0.002**±0.001	0.098±0.007	0.469±0.030	0.819±0.054	0.865±0.053
KPI	0.326±0.009	0.345±0.012	0.388±0.016	0.406±0.024	0.411±0.031
DiffPuter	1.212±0.135	1.197±0.125	1.129±0.257	1.114±0.194	1.085±0.116
O-MIRACLE	0.920±0.148	0.863±0.156	0.896±0.169	0.946±0.194	1.020±0.178
MissCNF (ours)	0.005±0.002	**0.009**±0.002	**0.018**±0.015	**0.018**±0.011	**0.026**±0.022

Table 12: KL vs. nonlinearity strength α on the Chain_α graph, both missingness mechanisms and two missing rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (mechanism × rate).

Method	$\alpha = 0.00$	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
MCAR, 30%					
Listwise deletion	0.027±0.006	0.027±0.008	0.025±0.011	0.022±0.009	0.023±0.006
MICE	**0.021**±0.006	0.020±0.003	0.030±0.007	0.041±0.006	0.048±0.007
KPI	0.086±0.016	0.083±0.011	0.064±0.006	0.038±0.008	0.029±0.008
DiffPuter	0.039±0.013	0.038±0.011	0.042±0.015	0.038±0.009	0.037±0.006
O-MIRACLE	0.021±0.005	**0.020**±0.005	**0.019**±0.005	**0.018**±0.005	**0.019**±0.003
MissCNF (ours)	0.030±0.007	0.028±0.007	0.026±0.010	0.027±0.008	0.027±0.006
MCAR, 60%					
Listwise deletion	0.044±0.002	0.045±0.001	0.047±0.003	0.050±0.007	0.049±0.006
MICE	0.023±0.004	0.029±0.007	0.046±0.004	0.061±0.003	0.070±0.002
KPI	0.109±0.024	0.105±0.021	0.080±0.015	0.045±0.013	0.036±0.008
DiffPuter	0.050±0.018	0.057±0.028	0.063±0.011	0.071±0.022	0.063±0.007
O-MIRACLE	**0.017**±0.001	**0.019**±0.001	**0.019**±0.004	**0.018**±0.003	**0.018**±0.004
MissCNF (ours)	0.034±0.005	0.035±0.005	0.029±0.008	0.029±0.006	0.026±0.006
MAR, 30%					
Listwise deletion	0.021±0.007	**0.020**±0.004	0.020±0.002	0.020±0.004	0.020±0.005
MICE	0.026±0.005	0.023±0.010	0.037±0.010	0.055±0.011	0.053±0.013
KPI	0.084±0.014	0.075±0.013	0.055±0.015	0.035±0.010	0.033±0.010
DiffPuter	0.043±0.003	0.048±0.019	0.071±0.040	0.069±0.022	0.067±0.021
O-MIRACLE	**0.019**±0.004	0.020±0.007	0.021±0.008	0.021±0.008	0.020±0.007
MissCNF (ours)	0.022±0.006	0.024±0.009	**0.017**±0.003	**0.016**±0.002	**0.016**±0.001
MAR, 60%					
Listwise deletion	0.041±0.007	0.040±0.009	0.038±0.009	0.037±0.005	0.032±0.007
MICE	**0.016**±0.006	0.040±0.002	0.097±0.014	0.137±0.018	0.139±0.014
KPI	0.110±0.010	0.102±0.013	0.074±0.011	0.046±0.013	0.046±0.006
DiffPuter	0.094±0.047	0.102±0.069	0.113±0.069	0.094±0.067	0.110±0.057
O-MIRACLE	0.020±0.008	**0.022**±0.010	**0.022**±0.008	**0.021**±0.005	**0.022**±0.006
MissCNF (ours)	0.027±0.012	0.029±0.016	0.032±0.008	0.028±0.006	0.025±0.007

Table 13: RMSE_{ATE} vs. nonlinearity strength α on the Chain_α graph, both missingness mechanisms and two missing rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (mechanism × rate).

Method	$\alpha = 0.00$	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
MCAR, 30%					
Listwise deletion	0.024±0.002	**0.026**±0.002	**0.027**±0.004	**0.027**±0.004	**0.026**±0.003
MICE	**0.020**±0.003	0.041±0.001	0.085±0.002	0.119±0.002	0.124±0.002
KPI	0.085±0.009	0.083±0.005	0.083±0.002	0.080±0.003	0.077±0.006
DiffPuter	0.041±0.010	0.042±0.007	0.046±0.003	0.042±0.007	0.041±0.006
O-MIRACLE	0.023±0.001	0.029±0.002	0.038±0.003	0.033±0.002	0.034±0.006
MissCNF (ours)	0.027±0.005	0.027±0.003	0.028±0.006	0.029±0.005	0.030±0.004
MCAR, 60%					
Listwise deletion	0.044±0.004	0.047±0.002	0.052±0.002	0.052±0.006	0.053±0.004
MICE	**0.022**±0.002	0.069±0.003	0.148±0.005	0.203±0.005	0.216±0.005
KPI	0.125±0.015	0.121±0.012	0.122±0.009	0.120±0.012	0.116±0.012
DiffPuter	0.081±0.007	0.089±0.007	0.097±0.006	0.097±0.018	0.084±0.006
O-MIRACLE	0.023±0.005	0.039±0.006	0.058±0.003	0.054±0.004	0.053±0.008
MissCNF (ours)	0.035±0.008	**0.037**±0.007	**0.037**±0.009	**0.037**±0.005	**0.034**±0.007
MAR, 30%					
Listwise deletion	0.023±0.002	**0.024**±0.002	0.025±0.003	0.024±0.004	0.025±0.003
MICE	**0.022**±0.002	0.045±0.004	0.091±0.007	0.125±0.007	0.130±0.007
KPI	0.088±0.008	0.088±0.007	0.089±0.010	0.082±0.009	0.080±0.007
DiffPuter	0.061±0.015	0.063±0.020	0.084±0.031	0.082±0.022	0.076±0.023
O-MIRACLE	0.050±0.007	0.046±0.015	0.045±0.022	0.048±0.021	0.053±0.013
MissCNF (ours)	0.022±0.004	0.025±0.003	**0.021**±0.002	**0.022**±0.002	**0.021**±0.002
MAR, 60%					
Listwise deletion	0.065±0.023	0.066±0.025	0.068±0.028	0.069±0.030	0.066±0.026
MICE	**0.016**±0.001	0.080±0.002	0.179±0.008	0.248±0.011	0.261±0.009
KPI	0.126±0.009	0.123±0.009	0.122±0.010	0.123±0.011	0.123±0.008
DiffPuter	0.182±0.041	0.190±0.058	0.202±0.050	0.175±0.048	0.170±0.032
O-MIRACLE	0.106±0.007	0.086±0.015	0.076±0.027	0.088±0.024	0.107±0.016
MissCNF (ours)	0.028±0.004	**0.029**±0.006	**0.036**±0.005	**0.035**±0.004	**0.038**±0.012

Table 14: RMSE_{CF} vs. nonlinearity strength α on the Chain_α graph, both missingness mechanisms and two missing rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (mechanism × rate).

Method	$\alpha = 0.00$	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
MCAR, 30%					
Listwise deletion	0.004±0.004	**0.007**±0.003	**0.009**±0.002	**0.010**±0.001	**0.008**±0.002
MICE	**0.003**±0.003	0.029±0.004	0.168±0.010	0.349±0.014	0.398±0.016
KPI	0.055±0.010	0.056±0.012	0.059±0.010	0.061±0.007	0.065±0.010
DiffPuter	0.057±0.007	0.060±0.011	0.068±0.008	0.068±0.010	837.056±1871.559
O-MIRACLE	0.071±0.010	0.073±0.011	0.075±0.010	0.079±0.011	0.079±0.008
MissCNF (ours)	0.004±0.003	0.007±0.004	0.011±0.007	0.010±0.003	0.009±0.001
MCAR, 60%					
Listwise deletion	0.007±0.003	0.010±0.004	0.013±0.003	**0.015**±0.004	0.016±0.003
MICE	**0.005**±0.003	0.085±0.009	0.457±0.020	0.875±0.031	0.983±0.032
KPI	0.321±0.025	0.328±0.035	0.330±0.031	0.340±0.050	0.334±0.041
DiffPuter	0.399±0.032	0.402±0.028	0.414±0.053	0.418±0.039	0.417±0.037
O-MIRACLE	0.472±0.043	0.475±0.048	0.481±0.052	0.489±0.049	0.485±0.055
MissCNF (ours)	0.007±0.005	**0.010**±0.005	**0.013**±0.005	0.015±0.004	**0.013**±0.005
MAR, 30%					
Listwise deletion	0.068±0.038	0.071±0.036	0.077±0.039	0.078±0.038	0.080±0.040
MICE	**0.005**±0.004	0.076±0.021	0.334±0.077	0.617±0.117	0.698±0.142
KPI	0.118±0.049	0.130±0.054	0.159±0.067	0.210±0.124	0.176±0.083
DiffPuter	0.198±0.134	0.233±0.162	0.268±0.160	0.234±0.123	0.226±0.136
O-MIRACLE	0.151±0.059	0.149±0.058	0.153±0.057	0.156±0.053	0.155±0.053
MissCNF (ours)	0.005±0.003	**0.010**±0.004	**0.010**±0.002	**0.013**±0.005	**0.010**±0.004
MAR, 60%					
Listwise deletion	0.258±0.130	0.262±0.131	0.261±0.140	0.264±0.137	0.268±0.133
MICE	**0.005**±0.002	0.175±0.058	0.787±0.203	1.383±0.317	1.538±0.337
KPI	0.810±0.456	1.080±0.945	1.305±1.366	1.414±1.666	1.657±2.176
DiffPuter	1.725±1.762	1.329±0.788	1.548±1.276	2.516±3.531	1.015±0.361
O-MIRACLE	1.680±1.784	1.663±1.704	1.527±1.370	1.434±1.194	1.421±1.134
MissCNF (ours)	0.010±0.003	**0.017**±0.011	**0.014**±0.005	**0.017**±0.003	**0.018**±0.006

Table 15: KL vs. nonlinearity strength α on the Collider$_\alpha$ graph, both missingness mechanisms and two missing rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (mechanism × rate).

Method	$\alpha = 0.00$	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
MCAR, 30%					
Listwise deletion	0.017±0.005	0.015±0.004	0.015±0.003	0.015±0.003	0.015±0.003
MICE	0.016±0.003	0.015±0.001	0.020±0.003	0.022±0.005	0.026±0.003
KPI	0.015±0.005	0.014±0.005	0.014±0.005	0.016±0.004	0.014±0.004
DiffPuter	0.017±0.005	0.015±0.006	0.015±0.006	0.020±0.009	0.531±1.153
O-MIRACLE	**0.015**±0.003	**0.013**±0.003	**0.013**±0.003	**0.013**±0.004	0.013±0.003
MissCNF (ours)	0.017±0.003	0.013±0.003	0.015±0.005	0.013±0.003	**0.013**±0.004
MCAR, 60%					
Listwise deletion	0.019±0.004	0.018±0.005	0.019±0.006	0.020±0.006	0.018±0.004
MICE	**0.015**±0.003	0.023±0.007	0.042±0.007	0.054±0.005	0.056±0.004
KPI	0.018±0.005	0.020±0.007	0.020±0.006	0.020±0.005	0.019±0.007
DiffPuter	0.046±0.011	0.036±0.012	0.026±0.005	0.020±0.008	**0.015**±0.007
O-MIRACLE	0.015±0.006	**0.016**±0.007	**0.015**±0.007	0.017±0.007	0.015±0.005
MissCNF (ours)	0.019±0.005	0.018±0.006	0.018±0.006	**0.016**±0.004	0.015±0.005
MAR, 30%					
Listwise deletion	0.015±0.003	**0.012**±0.002	0.015±0.005	0.016±0.006	0.016±0.006
MICE	0.016±0.003	0.022±0.005	0.032±0.003	0.038±0.006	0.038±0.009
KPI	0.014±0.003	0.014±0.003	0.015±0.005	0.015±0.005	0.014±0.005
DiffPuter	0.028±0.020	0.030±0.018	0.027±0.011	0.018±0.007	0.018±0.008
O-MIRACLE	**0.014**±0.002	0.013±0.001	0.013±0.003	0.013±0.003	0.013±0.003
MissCNF (ours)	0.016±0.003	0.013±0.004	**0.012**±0.003	**0.013**±0.001	**0.012**±0.002
MAR, 60%					
Listwise deletion	0.018±0.010	0.016±0.008	0.021±0.013	0.020±0.012	0.021±0.012
MICE	**0.016**±0.004	0.047±0.010	0.087±0.027	0.115±0.037	0.116±0.040
KPI	0.023±0.006	0.022±0.007	0.019±0.007	0.018±0.009	0.019±0.011
DiffPuter	0.105±0.051	0.096±0.046	0.072±0.035	0.030±0.019	0.022±0.006
O-MIRACLE	0.024±0.019	0.023±0.012	0.023±0.009	0.021±0.008	0.021±0.008
MissCNF (ours)	0.019±0.003	**0.013**±0.002	**0.016**±0.006	**0.017**±0.007	**0.016**±0.006

Table 16: RMSE$_{\text{ATE}}$ vs. nonlinearity strength α on the Collider$_\alpha$ graph, both missingness mechanisms and two missing rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (mechanism × rate).

Method	$\alpha = 0.00$	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
MCAR, 30%					
Listwise deletion	0.020±0.003	0.025±0.003	0.028±0.002	0.029±0.002	0.029±0.002
MICE	**0.019**±0.002	0.033±0.004	0.064±0.005	0.080±0.005	0.086±0.003
KPI	0.026±0.004	0.028±0.004	0.030±0.004	0.031±0.001	0.031±0.006
DiffPuter	0.021±0.003	0.024±0.001	0.028±0.004	0.031±0.003	0.422±0.873
O-MIRACLE	0.020±0.002	**0.023**±0.003	**0.025**±0.002	**0.026**±0.003	**0.026**±0.003
MissCNF (ours)	0.020±0.001	0.024±0.001	0.030±0.004	0.029±0.002	0.029±0.004
MCAR, 60%					
Listwise deletion	0.026±0.003	0.031±0.002	0.034±0.004	0.036±0.005	0.036±0.004
MICE	**0.020**±0.001	0.061±0.003	0.123±0.007	0.161±0.008	0.168±0.008
KPI	0.045±0.004	0.049±0.004	0.046±0.007	0.045±0.004	0.044±0.002
DiffPuter	0.041±0.005	0.039±0.007	0.037±0.006	0.046±0.005	0.042±0.006
O-MIRACLE	0.022±0.004	**0.026**±0.003	**0.026**±0.003	**0.031**±0.004	**0.030**±0.005
MissCNF (ours)	0.025±0.003	0.028±0.002	0.031±0.004	0.032±0.002	0.032±0.004
MAR, 30%					
Listwise deletion	0.021±0.002	**0.024**±0.002	0.027±0.003	0.029±0.003	0.031±0.002
MICE	**0.020**±0.002	0.050±0.006	0.088±0.014	0.109±0.018	0.115±0.019
KPI	0.045±0.012	0.048±0.012	0.051±0.012	0.052±0.013	0.050±0.011
DiffPuter	0.059±0.026	0.062±0.027	0.066±0.026	0.061±0.019	0.060±0.021
O-MIRACLE	0.044±0.014	0.046±0.014	0.047±0.013	0.048±0.014	0.048±0.013
MissCNF (ours)	0.022±0.003	0.025±0.002	**0.027**±0.004	**0.028**±0.004	**0.030**±0.003
MAR, 60%					
Listwise deletion	0.036±0.002	0.038±0.003	0.039±0.006	0.039±0.006	0.040±0.002
MICE	**0.018**±0.002	0.080±0.012	0.165±0.027	0.212±0.039	0.221±0.042
KPI	0.095±0.028	0.099±0.034	0.097±0.037	0.094±0.038	0.096±0.042
DiffPuter	0.155±0.071	0.147±0.053	0.139±0.061	0.138±0.069	0.110±0.027
O-MIRACLE	0.092±0.050	0.095±0.049	0.095±0.044	0.093±0.041	0.093±0.039
MissCNF (ours)	0.029±0.006	**0.032**±0.005	**0.032**±0.005	**0.037**±0.004	**0.036**±0.003

Table 17: RMSE_{CF} vs. nonlinearity strength α on the Collider_α graph, both missingness mechanisms and two missing rates. Values are mean ± std over seeds. Lower is better. Bold = best, underline = second-best, within each row-block (mechanism × rate).