ROBUST SIMULATION-BASED INFERENCE UNDER MISSING DATA VIA NEURAL PROCESSES

Anonymous authors
Paper under double-blind review

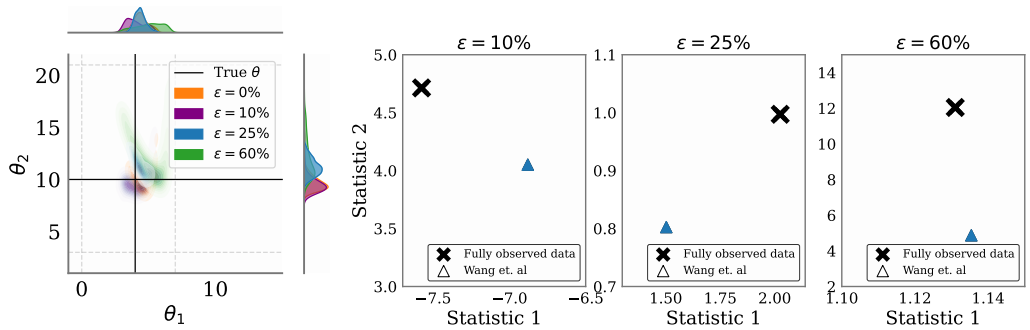
ABSTRACT

Simulation-based inference (SBI) methods typically require fully observed data to infer parameters of models with intractable likelihood functions. However, datasets often contain missing values due to incomplete observations, data corruptions (common in astrophysics), or instrument limitations (e.g., in high-energy physics applications). In such scenarios, missing data must be imputed before applying any SBI method. This work formalizes the problem of missing data in SBI and demonstrates that naive imputation methods can introduce bias into the SBI posterior. We introduce a novel method that addresses this issue by jointly learning the imputation model and the inference network within a neural posterior estimation (NPE) framework. Extensive empirical results on SBI benchmarks show that our approach provides robust inference outcomes compared to baselines, for varying levels of missing data, while being amortized.

1 INTRODUCTION

Mechanistic models for studying complex physical or biological phenomena have become indispensable tools in research fields as diverse as genetics (Riesselman et al., 2018), epidemiology (Kypraios et al., 2017), gravitational wave astronomy (Dax et al., 2021), and radio propagation (Bharti et al., 2022a). However, fitting such models to observational data can be challenging due to the intractability of their likelihood functions, which renders standard Bayesian inference methods inapplicable. Simulation-based inference (SBI) methods (Cranmer et al., 2020) tackle this issue by relying on forward simulations from the model instead of evaluating the likelihood. These simulations are then either used to train a conditional density estimator (Papamakarios and Murray, 2016; Lueckmann et al., 2017b; Greenberg et al., 2019; Papamakarios et al., 2019; Radev et al., 2020), or to measure distance with the observed data (Sisson, 2018; Briol et al., 2019; Pesonen et al., 2023) to approximately estimate the posterior distribution of the parameters of interest.

SBI methods implicitly assume that the observed data distribution is a member of the family of distributions induced by the model, or in other words, that the model is *well-specified*. However, this assumption is often violated in practice, as models of complicated real-world phenomena tend to be *misspecified* when the model is not an accurate representation of the phenomenon under study. Even if the model is well-specified, the data collection mechanism can induce missing data due to, for instance, incomplete observations (Luken et al., 2021), instrument limitations (Kasak et al., 2024), or unfavorable experimental conditions, thus hindering the application of SBI methods. Although the former problem of model misspecification has been studied in a number of works (Frazier et al., 2020; Dellaporta et al., 2022; Fujisawa et al., 2021; Bharti et al., 2022b; Ward et al., 2022; Schmitt et al., 2023; Gloeckler et al., 2023; Huang et al., 2023; Gao et al., 2023; Kelly et al., 2024), the latter problem of missing data in SBI has received relatively less attention. A notable exception is the work of Wang et al. (2024), which attempts to handle missing data by augmenting and imputing constant values (e.g., zero or sample mean) and training NPE with a binary mask indicator, but this approach can lead to biased estimates, reduced variability, and distorted relationships between variables (Graham et al., 2007). This is exemplified in Figure 1 where we investigate the impact of missing data on neural posterior estimation (NPE, Papamakarios and Murray (2016))—a popular SBI method—on a population genetics model. We observe that simply imputing the missing values with zeros or the sample mean leads to heavily biased posterior estimates. Other SBI works that address missing data include Lueckmann et al. (2017a) and Gloeckler et al. (2024).



(a) Posterior (b) Shifting statistics

Figure 1: **Effect of missing data on SBI.** (a) NPE posterior for the two-parameter Ricker model (Wood, 2010) where the method of Wang et al. (2024) (with zero augmentation) is used to handle $\varepsilon\%$ of values missing in the data. As ε increases, the NPE posteriors become increasingly biased and drift away from the true parameter value, denoted by the black lines. (b) The corresponding learned statistics for fully observed and augmented datasets. Observe that the statistics for augmented datasets shift away from the fully observed statistic value, thereby leading to a shift in the corresponding NPE posterior away from the true parameter value. Note that as we re-train the method for each missingness level, the three statistics plots should not be compared with each other.

Outside of SBI, the problem of missing data has been extensively studied (Van Buuren and Groothuis-Oudshoorn, 2011), with Rubin (1976) categorizing it into three types: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). Recent advances in machine learning have led to the development of novel methods for addressing this problem using generative adversarial networks (GANs) (Luo et al., 2018; Yoon et al., 2018; Li et al., 2019; Yoon and Sull, 2020), variational autoencoders (VAEs) (Nazabal et al., 2020; Collier et al., 2020; Mattei and Frellsen, 2019; Ipsen et al., 2020; Ghalebikesabi et al., 2021b), Gaussian processes (Casale et al., 2018; Fortuin et al., 2020; Ramchandran et al., 2021; Ong et al., 2024), and optimal transport (Muzellec et al., 2020; Zhao et al., 2023; Vo et al., 2024). These methods offer new perspectives on the problem of missing data imputation, but their application has been primarily limited to predicting missing values. Notably, they have not been developed for inference over missing values, which remains a significant challenge for SBI.

Contributions. In this paper, we introduce a novel SBI method that is robust to the shift in the posterior distribution in the presence of missing values in the data. Our method, named RISE (short for "Robust Inference under imputed SimulatEd data"), jointly performs imputation and inference by combining NPE with latent neural processes (Foong et al., 2020). Doing so allows us to learn an *amortized* model unlike other robust SBI methods in the literature, and to handle missing data under different assumptions (Little and Rubin, 2019). We summarize our main contributions below:

- We motivate the problem of missing data in SBI, demonstrating how it can induce bias in posterior estimation.
- We propose RISE, an amortized method, that jointly learns an imputation and inference model to perform inference in the presence of missing data.
- RISE outperforms competing baselines in inference and imputation tasks across varying levels of missingness, demonstrating robust performance in the presence of missing data.

2 Preliminaries

Consider a simulator-based model $p(\cdot \mid \theta)$ that takes in a parameter vector $\theta \in \Theta \subseteq \mathbb{R}^p$ and maps it to a point $\mathbf{x} = [x_1, \ldots, x_d]^\top$ in some data space $\mathcal{X} \subseteq \mathbb{R}^d$. We assume that $p(\cdot \mid \theta)$ is intractable, meaning that its associated likelihood function is unavailable and cannot be evaluated point-wise. However, generating independent and identically distributed (iid) realisations $\mathbf{x} \sim p(\cdot \mid \theta)$ for a given θ is straightforward. Given dataset $\mathbf{x}$ collected via real-world experiments from some true data-

generating process, and a prior distribution on the parameters $p(\theta)$, we are interested in approximating the posterior distribution $p(\theta \mid \mathbf{x}) \propto p(\mathbf{x} \mid \theta)p(\theta)$. This can be achieved, for instance, using the popular neural posterior estimation (NPE) framework, which we next introduce.

Neural posterior estimation. NPE (Papamakarios and Murray, 2016) involves training conditional density estimators, such as normalizing flows (Papamakarios et al., 2021), to learn a mapping from each data $\mathbf{x}$ to the posterior distribution $p(\theta \mid \mathbf{x})$. Assuming that the true posterior is a member of the family of distributions defined by the normalizing flow $q_\phi(\theta \mid \mathbf{x})$ with learnable parameters ϕ, then q_ϕ is trained by minimizing the empirical loss function

$$\ell_{\text{NPE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(\theta_i \mid \mathbf{x}_i) \approx -\mathbb{E}_{\theta\sim p(\theta)}[\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}\mid\theta)}[\log q_\phi(\theta \mid \mathbf{x})]], \tag{1}$$

using the dataset $\{(\theta_i, \mathbf{x}_i)\}_{i=1}^n$ simulated from the joint distribution $p(\theta, \mathbf{x}) = p(\mathbf{x} \mid \theta)p(\theta)$. In case that the data space $\mathcal{X}$ is high-dimensional or there are multiple observations $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}]$ for each θ, a summary function $\eta : \mathcal{X} \to \mathcal{S}$ is used to map $\mathbf{x}$ onto a low-dimensional space $\mathcal{S}$, such as a deep set (Zaheer et al., 2017). Assuming the summary function is parameterized by a neural network with ψ, the joint NPE loss over both ϕ and ψ becomes $\ell_{\text{NPE}}(\phi, \psi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(\theta_i \mid \eta_\psi(\mathbf{x}_i))$. Once both q_ϕ and η_ψ are trained, the NPE posterior $q_{\hat{\phi}}(\theta \mid \eta_{\hat{\psi}}(\tilde{\mathbf{x}}))$ for a given real data $\tilde{\mathbf{x}}$ is obtained by a simple forward pass of $\tilde{\mathbf{x}}$ through the trained networks, making NPEs amortized. We now briefly introduce some background on the missing data problem, which is the focus of this work.

Missing data background. In the context of missing data, each data sample is composed of an observed part $\mathbf{x}_{\text{obs}}$ and a missing (or unobserved) part $\mathbf{x}_{\text{mis}}$ such that $\mathbf{x} = (\mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})$. The missingness pattern for each $\mathbf{x}$ is described by a binary mask variable $\mathbf{s} \in \{0, 1\}^d$, where $s_i = 1$ if the element x_i is observed and $s_i = 0$ if x_i is missing, $i = \{1, \ldots, d\}$. The joint distribution of $\mathbf{x}$ and $\mathbf{s}$ can be factorized as $p(\mathbf{x}, \mathbf{s}) = p(\mathbf{s} \mid \mathbf{x})p(\mathbf{x})$. Assumptions on what the conditional distribution of the mask (or the missingness mechanism) depends on gives rise to three different cases (Little and Rubin, 2019): (i) missing-completely-at-random (MCAR), where $p(\mathbf{s} \mid \mathbf{x}) = p(\mathbf{s})$; (ii) missing-at-random (MAR), where $p(\mathbf{s} \mid \mathbf{x}) = p(\mathbf{s} \mid \mathbf{x}_{\text{obs}})$; and (iii) missing-not-at-random (MNAR), where $p(\mathbf{s} \mid \mathbf{x}) = p(\mathbf{s} \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})$. While the missingness mechanism can be ignored in both MCAR and MAR cases when learning $p(\mathbf{x}_{\text{obs}}, \mathbf{s})$, that is not true for MNAR, where the missingness mechanism depends on $\mathbf{x}_{\text{mis}}$ (Ipsen et al., 2020). We wish to handle all the three cases when performing SBI.

3 METHOD

We begin by analysing the problem that missing data creates for SBI in Section 3.1. We then present RISE—our method for handling missing data in SBI. Section 3.2 outlines our learning objective, and Section 3.3 describes how we parameterize the imputation model in RISE using neural processes.

3.1 MISSING DATA PROBLEM IN SBI

We assume that the simulator can faithfully replicate the true data-generating process (i.e., the simulator is well-specified), however, the data collection mechanism induces missing values in the data $\mathbf{x}$. As a result, $\mathbf{x}$ contains both observed and missing values[1], represented as $\mathbf{x} = (\mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})$. For instance, $\mathbf{x} = (0.1\ 1.2\ \text{N/A}\ 0.9)$ exemplifies a scenario where a specific observation x_i includes missing values denoted by "N/A". Naturally, SBI methods cannot operate on missing values, and so imputing $\mathbf{x}_{\text{mis}}$ is necessary before applying any SBI method on them. However, if the missing values are not imputed "correctly", then the corresponding SBI posterior becomes biased, as observed in Figure 1 for the constant imputation case. We now describe this problem mathematically.

Definition 1 (SBI posterior under true imputation)**.** *Let $p_{true}(\mathbf{x}_{mis} \mid \mathbf{x}_{obs})$ be the true predictive distribution of the missing values given the observed data. Then, the SBI posterior can be written as*

$$p_{SBI}(\theta \mid \mathbf{x}_{obs}) = \int \underbrace{p_{SBI}(\theta \mid \mathbf{x}_{obs}, \mathbf{x}_{mis})}_{\textit{Inference}} \underbrace{p_{true}(\mathbf{x}_{mis} \mid \mathbf{x}_{obs})}_{\textit{Imputation}} d\mathbf{x}_{mis}. \tag{2}$$

[1] Note that during training the $\mathbf{x}_{\text{obs,mis}}$ are partitions of simulated data, while during inference we only observe $\mathbf{y}$ from real world.

Instead of constant imputation, we now have a distribution over the missing values given $\mathbf{x}_{\text{obs}}$, and the problem of SBI under missing data is formulated as an expectation of the SBI posterior $p_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})$ with respect to $p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$, analogous to traditional (likelihood-based) Bayesian inference methods (Schafer and Schenker, 2000; Zhou and Reiter, 2010). Therefore, estimating the above expectation requires access to $p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$ (Raghunathan et al., 2001; Gelman et al., 1995), which is infeasible in most practical cases.

Definition 2 (SBI posterior under estimated imputation)**.** *Let $\hat{p}(\mathbf{x}_{mis} \mid \mathbf{x}_{obs})$ denote an estimate of the true imputation model $p_{true}(\mathbf{x}_{mis} \mid \mathbf{x}_{obs})$. Then, the corresponding SBI posterior can be written as*

$$\hat{p}_{SBI}(\theta \mid \mathbf{x}_{obs}) = \int p_{SBI}(\theta \mid \mathbf{x}_{obs}, \mathbf{x}_{mis}) \hat{p}(\mathbf{x}_{mis} \mid \mathbf{x}_{obs}) d\mathbf{x}_{mis}. \tag{3}$$

Proposition 1. *If $\hat{p}(\mathbf{x}_{mis} \mid \mathbf{x}_{obs})$ is misaligned with $p_{true}(\mathbf{x}_{mis} \mid \mathbf{x}_{obs})$, then the estimated SBI posterior $\hat{p}_{SBI}(\theta \mid \mathbf{x}_{obs})$ will be biased, i.e., $\left|\mathbb{E}_{\theta \sim p_{SBI}(\cdot \mid \mathbf{x}_{obs})}[\theta \mid \mathbf{x}_{obs}] - \mathbb{E}_{\theta \sim \hat{p}_{SBI}(\cdot \mid \mathbf{x}_{obs})}[\theta \mid \mathbf{x}_{obs}]\right| \neq 0$.*

The proof, which follows straightforwardly using Definition 1 and Definition 2, is given in Appendix A.2.1 for completeness. Proposition 1 says that the bias in the SBI posterior directly comes from the discrepancy between the true imputation model $p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$ and the estimated one $\hat{p}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$. This applies irrespective of the inference method used, and therefore, rather unsurprisingly, the key to reducing this bias is to learn the imputation model as accurately as possible. The rest of this section presents our method, named RISE, which combines the imputation task with SBI to reduce this bias.

3.2 Robust SBI under missing data

Let $p_{\text{true}}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})$ be the true posterior given both the observed data and the missing values, i.e., given $\mathbf{x} = (\mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})$. Our objective is to estimate the true posterior given only $\mathbf{x}_{\text{obs}}$. That is, we seek to approximate

$$p_{\text{true}}(\theta \mid \mathbf{x}_{\text{obs}}) \triangleq \int p_{\text{true}}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}}) p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) d\mathbf{x}_{\text{mis}} = \int p_{\text{true}}(\theta, \mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) d\mathbf{x}_{\text{mis}} .$$

We therefore introduce a family of distributions $r_\psi(\theta, \mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$ parameterized by ψ, and propose to solve the following optimization problem.

$$\arg\min_{\psi} \mathbb{E}_{\mathbf{x}_{\text{obs}} \sim p_{\text{true}}} \text{KL}\left[p_{\text{true}}(\theta, \mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) \,||\, \underbrace{r_\psi(\theta, \mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})}_{\textbf{(joint imputation and inference)}} \right] . \tag{4}$$

Solving this problem requires access to $p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$, which in most real-world scenarios, we do not have. Since samples for $\mathbf{x}_{\text{mis}}$ are required during training, we need to resort to methods such as variational approximation and expectation maximization (EM). Here, we adopt a variational approach, treating $\mathbf{x}_{\text{mis}}$ as latent variables in a probabilistic imputation setting. Specifically, the imputation network needs to estimate these latents for the inference network to map them to the output space. Both networks are tightly coupled since the distribution induced by the imputation network shapes the input of the inference network.

Mathematically, assuming access to only data samples $(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}$, we proceed to solving

$$\arg\min_{\psi} \mathbb{E}_{\mathbf{x}_{\text{obs}} \sim p_{\text{true}}} \text{KL}[p_{\text{true}}(\theta \mid \mathbf{x}_{\text{obs}}) \,||\, r_\psi(\theta \mid \mathbf{x}_{\text{obs}})] . \tag{5}$$

Our next proposition computes a variational lower bound for this objective, which we can maximize efficiently using an encoder-decoder architecture resembling variational autoencoders (VAEs).

Proposition 2 (Training objective)**.** *The objective in Equation (5) admits a variational lower bound, resulting in the following optimization problem.*

$$\begin{aligned} \hat{\phi}, \hat{\varphi} &= \arg\min_{\phi, \varphi} - \mathbb{E}_{(\mathbf{x}_{obs}, \theta) \sim p_{true}} \mathbb{E}_{\mathbf{x}_{mis} \sim p(\mathbf{x}_{mis} \mid \mathbf{x}_{obs})} \left[\log \underbrace{\hat{p}_\varphi(\mathbf{x}_{mis} \mid \mathbf{x}_{obs})}_{\textit{(imputation)}} + \log \underbrace{q_\phi(\theta \mid \mathbf{x}_{obs}, \mathbf{x}_{mis})}_{\textit{(inference)}} \right] \\ &= \arg\min_{\phi, \varphi} \ell_{RISE}(\phi, \varphi), \end{aligned} \tag{6}$$

where $\ell_{RISE}(\phi, \varphi)$ denotes the loss function for RISE.

Therefore, we can approximate the true imputation model $p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$ using a parametric neural network $\hat{p}_\varphi$, parameterized by its vector of weights and biases φ, and the SBI posterior given the full dataset $p_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{mis}}, \mathbf{x}_{\text{obs}})$ using the conditional density q_ϕ as in NPE.

The proof of Proposition 2 is outlined in Appendix A.2.2. Note that ℓ_{RISE} is a general loss which reduces to ℓ_{NPE} when there is no missing data, i.e., $\mathbf{x} = \mathbf{x}_{\text{obs}}$. In case a summary network η_κ is required before passing the data to q_ϕ, the joint loss function for RISE simply becomes

$$\ell_{\text{RISE}}(\phi, \varphi, \kappa) = -\mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}, \mathbf{x}_{\text{mis}} \sim p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})} \left[\log q_\phi(\theta \mid \eta_\kappa(\mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})) + \log \hat{p}_\varphi(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) \right].$$

The expectation in Equation (6) is taken with respect to the joint distribution of the simulator and the prior (as is standard for SBI methods), and the variational imputation distribution $p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$. Note that for simulations in our controlled experiments, we do not need to resort to the variational distribution $p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$, and can instead generate samples from $p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$ directly by first sampling $\mathbf{x}$ using the simulator, and then partitioning it into $\mathbf{x}_{\text{obs}}$ and $\mathbf{x}_{\text{mis}}$ based on the missingness assumption (i.e. creating the mask $\mathbf{s}$ under MCAR or MAR or MNAR assumption) such that $\varepsilon\%$ portion of the data is missing. The $\mathbf{x}_{\text{mis}}$ values are then used as true labels when comparing against the output of the imputation model $\hat{p}_\varphi$ during training. This allows us to amortize over instances of real data. In Section 3.3, we discuss how RISE can be used to amortize over the proportion of missing values ε in the data.

Using a latent variable representation (Kingma, 2013) for the imputation model, we factorize $\hat{p}_\varphi(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$, similar to the work by Mattei and Frellsen (2019), as,

$$\hat{p}_\varphi(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) = \int \hat{p}_\alpha(\mathbf{x}_{\text{mis}} \mid \tilde{\mathbf{z}}, \mathbf{x}_{\text{obs}}) \hat{p}_\beta(\tilde{\mathbf{z}} \mid \mathbf{x}_{\text{obs}}) d\tilde{\mathbf{z}}.$$

Here, $\tilde{\mathbf{z}} = (\mathbf{z}, \mathbf{s})$ represents both the latent variable $\mathbf{z}$ and the masking variable $\mathbf{s}$. The conditional distribution of the latent $\hat{p}_\beta(\tilde{\mathbf{z}} \mid \mathbf{x}_{\text{obs}})$ may depend on both the observed and the missing data depending on the different missingness assumptions (Little and Rubin, 2019):

- MCAR: $\hat{p}_\beta(\tilde{\mathbf{z}} \mid \mathbf{x}_{\text{obs}}) = p_{\beta_1}(\mathbf{z} \mid \mathbf{x}_{\text{obs}}) p_{\beta_2}(\mathbf{s})$
- MAR: $\hat{p}_\beta(\tilde{\mathbf{z}} \mid \mathbf{x}_{\text{obs}}) = p_{\beta_1}(\mathbf{z} \mid \mathbf{x}_{\text{obs}}) p_{\beta_2}(\mathbf{s} \mid \mathbf{x}_{\text{obs}})$
- MNAR: $\hat{p}_\beta(\tilde{\mathbf{z}} \mid \mathbf{x}_{\text{obs}}) = p_{\beta_1}(\mathbf{z} \mid \mathbf{x}_{\text{obs}}) \int p_{\beta_2}(\mathbf{s} \mid \mathbf{x}) p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) d\mathbf{x}_{\text{mis}}$

Note that for the MCAR and MAR cases, we only need the latent $\mathbf{z}$ in order to impute $\mathbf{x}_{\text{mis}}$ (Mattei and Frellsen, 2019), in which case $\tilde{\mathbf{z}} = \mathbf{z}$. However, in the MNAR case, $\tilde{\mathbf{z}} = (\mathbf{z}, \mathbf{s})$ as we will explicitly need to account for the missingness mechanism (Ipsen et al., 2020). Hereafter, we continue to denote the latent variable with $\tilde{\mathbf{z}}$ for a general formulation encompassing all the three cases. The pseudocode for training RISE is outlined in Algorithm 1.

3.3 Learning the imputation model using Neural Process

We utilise Neural Processes (NPs) (Garnelo et al., 2018) for parameterizing the imputation model $\hat{p}_\varphi(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$. NPs represent a family of neural network-based meta-learning models that combine the flexibility of deep learning with well-calibrated uncertainty estimates and a tractable training objective. These models learn a distribution over predictors given their target positions or locations, akin to a stochastic process. We refer the interested reader to Appendix A.3 for a detailed background. We employ neural processes to model the predictive density over missing values at their specific locations. Let $\mathbf{c}_{\text{mis}} = (c_{\text{mis},1}, \ldots, c_{\text{mis},k})$ and $\mathbf{c}_{\text{obs}} = (c_{\text{obs},1}, \ldots, c_{\text{obs},d-k})$ denote the location of $\mathbf{x}_{\text{mis}}$ and $\mathbf{x}_{\text{obs}}$, respectively, where k denotes the number of missing values (or the dimensionality of $\mathbf{x}_{\text{mis}}$). Furthermore, let $C = \{\mathbf{x}_{\text{obs}}, c_{\text{obs}}\}$ be the observed context set. Then, following latent neural processes (Foong et al., 2020), we can write $\hat{p}_\varphi$ as

$$\hat{p}_\varphi(\mathbf{x}_{\text{mis}} \mid c_{\text{mis}}, C) = \int \hat{p}_\alpha\big(\mathbf{x}_{\text{mis}} \mid c_{\text{mis}}, \tilde{\mathbf{z}}\big) \hat{p}_\beta(\tilde{\mathbf{z}} \mid C) d\tilde{\mathbf{z}} = \int \prod_{i=1}^{k} \hat{p}_\alpha\big(x_{\text{mis},i} \mid c_{\text{mis},i}, \tilde{\mathbf{z}}\big) \hat{p}_\beta(\tilde{\mathbf{z}} \mid C) d\tilde{\mathbf{z}}, \quad (7)$$

where $\varphi = (\alpha, \beta)$ are the trainable parameters of the imputation model.

Here we have assumed conditional independence of each $x_{\text{mis},i}$ given $c_{\text{mis},i}$ and $\tilde{\mathbf{z}}$, which allows for the joint distribution to factorize into a product of its marginals. Note that this factorization directly inherits the consistency properties from neural processes, as established by Garnelo et al. (2018); Dubois et al. (2020), ensuring a consistent distribution representation. Figure 2 showcases the associated plate diagram. To specify the model, we parameterize:

- **Encoder** $\hat{p}_\beta(\tilde{\mathbf{z}} \mid C)$, which provides a distribution over the latent variables $\tilde{\mathbf{z}}$ having observed the context set C. The encoder is parameterized to be permutation invariant to correctly treat C as a set, as required by NPs.
- **Decoder** $\hat{p}_\alpha(\mathbf{x}_{\text{mis},i} \mid c_{\text{mis},i}, \tilde{\mathbf{z}})$, which provides predictive distributions conditioned on $\tilde{\mathbf{z}}$ and the missing locations c_{mis}. In practice, it is parameterized as a Gaussian distribution, where the decoder predicts the mean and variance, denoted by α.

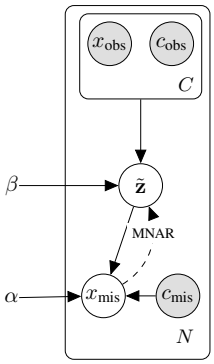


Figure 2: Plate diagram

Note that the Gaussian assumption in the decoder does not limit the expressivity of the method. In principle, it results in an infinite mixture of Gaussians (Rasmussen, 1999) in the predictive likelihood (Equation (7)), capable of representing any predictive density. While this likelihood is no longer analytically tractable, we can optimize it directly using the log-marginal predictive likelihood. Following (Foong et al., 2020), we estimate $\hat{p}_\varphi(\mathbf{x}_{\text{mis}} \mid c_{\text{mis}}, C)$ using Monte Carlo samples $\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_m \sim \hat{p}_\beta(\tilde{\mathbf{z}} \mid C)$ as

$$\log \hat{p}_\varphi(\mathbf{x}_{\text{mis}} \mid c_{\text{mis}}, C) \approx \log \left(\frac{1}{m} \sum_{j=1}^{m} \prod_{i=1}^{k} \hat{p}_\alpha\left(\mathbf{x}_{\text{mis},i} \mid c_{\text{mis},i}, \tilde{\mathbf{z}}_j\right) \right). \tag{8}$$

This can be directly used with standard optimizers (Kingma, 2014) to learn the model parameters.

As NPs are meta-learning models, we can utilize them to amortize over the proportion of missing values ε. Doing so is beneficial in cases where inference is required on multiple datasets with varying proportions of missing values, so as to avoid re-training for each ε. Assuming $p(\varepsilon)$ to be the distribution of the missingness proportion, we can consider each sample from $p(\varepsilon)$ to be one task when training RISE. Specifically, this can be done by first intitializing the parameters of RISE, and then repeating the following: (i) Sample $\varepsilon \sim p(\varepsilon)$, and (ii) Perform Steps 2-7 from Algorithm 1. We name this variant of our method as RISE-Meta. For each sample from the imputation model, we obtain a posterior distribution via the inference network, thus resulting in an ensemble of posterior distributions across all samples. In Section 5.3, we test the ability of RISE-Meta to generalize to unknown levels of missing values in the data.

Algorithm 1 RISE

Require: Simulator $p(\cdot \mid \theta)$, prior $p(\theta)$, iterations n_{iter}, missingness degree ε

1: Initialize parameters ϕ, φ of RISE
2: **for** $k = 1, \ldots, n_{\text{iter}}$ **do**
3: Sample $(\mathbf{x}, \theta) \sim p(\cdot \mid \theta)p(\theta)$
4: Create mask $\mathbf{s}$ wrt ε and MCAR/MNAR
5: Compute ℓ_{RISE} as per Equation (6)
6: $\phi, \varphi \leftarrow \texttt{optim}(\ell_{\text{RISE}}; \phi, \varphi)$
7: **end for**

4 Related Work

Missing data in SBI. Wang et al. (2024) attempt to handle missing data via data augmentation to missing values (e.g., zero or mean) and subsequently training NPE with a binary mask indicator, but this approach can lead to biased posterior estimates, as we saw in Figure 1 and Section 3.1. Wang et al. (2023; 2022) propose imputing missing values by sampling from a kernel density estimate (KDE) of the training data or using a nearest-neighbor search, and training the NPE model using augmented simulations. However, these approaches neglect the missingness mechanisms, which can distort the relationships between variables (Graham et al., 2007) and are neither scalable to higher dimensions. Lueckmann et al. (2017a) learn an imputation model agnostic of the missingness mechanism. More recently, Gloeckler et al. (2024) have proposed a transformer-based architecture for

SBI that can potentially handle conditioning on data with missing values. This method can perform arbitrary conditioning and evaluation, i.e. for a given $\mathbf{x} = [\mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}}]$, it first estimates the imputation distribution, i.e. $p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$, and then estimates the posterior distribution $p(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})$. However, it does not model the mechanism underlying the missing data and is thus not equipped to handle the MAR and MNAR settings. In contrast, RISE incorporates the missingness mechanism during its training and is therefore able to estimate the full posterior distribution, accounting for all variables.

Deep imputation methods. There is a growing body of work on imputing missing data using deep generative models. These include using GANs for missing data under MCAR assumption (Yoon et al., 2018; Li et al., 2019), and VAEs under MAR assumption (Mattei and Frellsen, 2019; Nazabal et al., 2020). Deep generative models have also been studied under MNAR assumption (Ghalebikesabi et al., 2021a; Gong et al., 2021; Ipsen et al., 2020; Ma and Zhang, 2021). We contribute to this line of work by using latent NPs to handle missing data under all the three missingness assumptions. Instead of learning an imputation model, Śmieja et al. (2018) propose replacing the neuron's response in the first hidden layer by its expected value to process missing data in neural networks.

5 Experiments

In this section, we assess the significance of RISE by empirically evaluating the following key claims. The first objective is to demonstrate that RISE yields posteriors that are robust to missing values in the data compared to NPE with constant or single imputation (see Section 5.1 and Section 5.2). Secondly, we aim to test the generalization capability of RISE-Meta in cases where the proportion of missing values in the data is not known *a priori* (Section 5.3). Thirdly, as learning the imputation model correctly is central to RISE's performance, we aim to validate that using neural processes-based imputation model used in RISE yields state-of-the-art results on imputing real-world datasets. Finally, we want to show that learning the inference and imputation model jointly, as is done in RISE, performs better than learning them separately. Experiments related to these objectives are in Section 5.4.

This section is organized as follows. We first provide results on SBI benchmarks in Sections 5.1, 5.2 and 5.3. In Section 5.4, we perform ablation studies to evaluate the imputation performance of RISE on real-world bio-activity datasets.

Performance metrics. We evaluate the accuracy of the posterior using two metrics: the maximum mean discrepancy (MMD) (Gretton et al., 2012) and the root mean squared error (RMSE). The MMD is computed between the posterior samples obtained under missing data (either using RISE or the baseline methods) and samples from a reference NPE posterior under no missing data. We use a radial basis function kernel for computing the MMD, and set its lengthscale using the median heuristic (Gretton et al., 2012) on the reference posterior samples. The RMSE is computed as $(1/N \sum_{i=1}^{N} (\theta_i - \theta_{\text{true}})^2)^{1/2}$ where $\{\theta_i\}_{i=1}^{N}$ are posterior samples, and θ_{true} is the true parameter.

Baselines. We compare RISE's performance against baselines derived from NPE (Greenberg et al., 2019) such as NPE with zero imputation (NPE-Zero) and NPE with sample mean imputation (NPE-Mean). We also include a baseline where NPE is combined with a standard feed-forward neural network (NN) for imputation, termed NPE-NN (Lueckmann et al., 2017a). Note that in NPE-NN, the neural network and NPE are trained jointly, similar to RISE. However, unlike RISE, NPE-NN performs single imputation.

Implementation. RISE is implemented in PyTorch (Paszke et al., 2019) and utilizes the same training configuration as the competing baselines, see Appendix A.4.4 for details. We take $\varepsilon = \{10\%, 25\%, 60\%\}$ to test performance from low to high missingness scenarios. We adopt the masking approach as described in Mattei and Frellsen (2019) and Ipsen et al. (2020) for MCAR and MNAR, respectively. Specifically, for MCAR we randomly mask $\varepsilon\%$ of the data, and for MNAR we use ε to compute a masking probability, which is then used to mask data according to their values. This *self-censoring* approach is described in Appendix A.4.3, and leads to a missingness proportion less than (or equal to) ε. We used a simulation budget of $n = 1000$ for all the SBI experiments, and take 1000 samples from the posterior distributions to compute the MMD. The performance is evaluated over 10 random runs. For more details, see Appendix A.4.

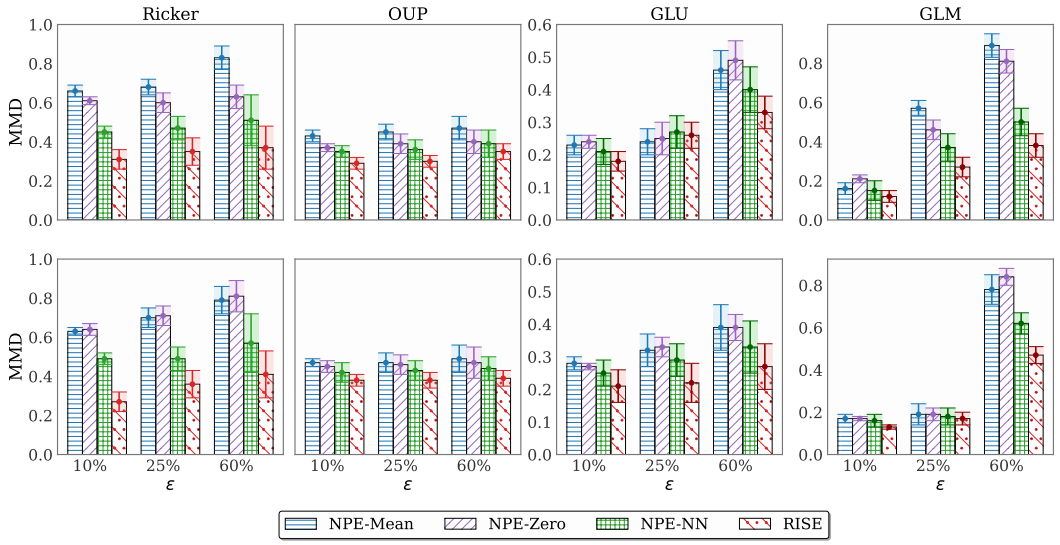


Figure 3: MMD values for RISE and baselines under MCAR (*top*) and MNAR (*bottom*) assumption with varying proportions of missing values in the data denoted by ε. RISE achieves the best performance, followed by NPE-NN, in estimating the posterior. Note that for MNAR case, the proportion of missing values is on average less than ε due to the self-censoring approach (Appendix A.4.3).

5.1 Performance on SBI benchmarks

We evaluate the performance of RISE in the presence of missing data using four common benchmark models from the SBI literature. These are (i) Ricker model: a two parameter simulator from population genetics (Wood, 2010); (ii) Ornstein-Uhlenbeck process (OUP): a two parameter stochastic differential equation model (Chen et al., 2021); (iii) Generalized Linear Model (GLM): a 10 parameter model with Bernoulli observations; and (iv) Gaussian Linear Uniform (GLU): a 10-dimensional Gaussian model with the mean vector as the parameter and a fixed covariance. The models are described in Appendix A.4.1 and the prior distributions we used are reported in Appendix A.4.2.

The results are shown in Figure 3 for MMD and Table 4 (in Appendix A.5) for RMSE at different levels of missingness ε for both MCAR and MNAR cases. We observe that RISE achieves the lowest values of MMD across all the models and types of missingness, thus outperforming the baselines in estimating the posterior distributions. As ε increases, the gap between RISE and the baselines increase, indicating that RISE is able to better handle high missingness levels in the data. As a sanity check, we also investigate the imputation capability of RISE in Figure 6, which shows that it achieves more accurate imputation results, which then naturally translates to robust posterior estimation.

5.2 Hodgkin-Huxley model

We now apply RISE on a real-world computational neuroscience simulator (Hodgkin and Huxley, 1952), namely the Hodgkin-Huxley model, which is a popular example in the SBI literature (Lueckmann et al., 2017b; Gao et al., 2023; Gloeckler et al., 2023). The aim is to infer the posterior over two parameters given the data of dimension 1200, see Appendix A.4.1 for the model description.

We set uniform priors and perform inference under different values of ε and missingness assumption, similar to Section 5.1. Figure 4 shows that RISE's posteriors are robust to increasing proportion of missing values as they stay around the true parameter value. In contrast, NPE-Zero and NPE-Mean yield posteriors that are heavily biased even under low levels of missing values.

5.3 Generalizing across unknown levels of missingness

Next, we test the generalization capability of our method to unknown levels of missing values. We perform meta-learning over different proportions of missing values ε

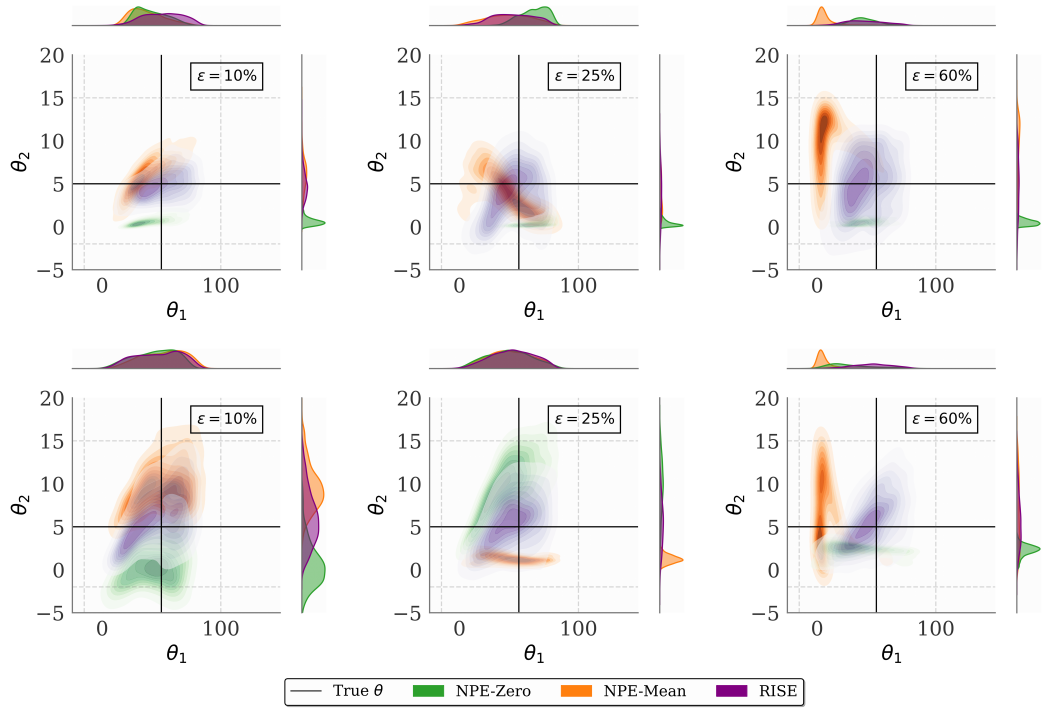


Figure 4: Posterior estimates for the Hodgkin-Huxley model under MCAR (*top row*) and MNAR (*bottom row*) with varying proportions of missing values in the data (denoted by ε). The posteriors obtained from RISE stay close to the true parameter (denoted by the black lines) for all values of ε, while those from the baseline methods move further away as ε increases.

in the dataset (termed RISE-Meta). For training RISE-Meta, we take the distribution of $p(\varepsilon)$ to be an equiprobable discrete distribution on the set $\{10\%, 25\%, 60\%\}$. We also train NPE-NN with a missingness degree of 60% as a baseline. We evaluate all the methods over 100 samples of varying missingness proportion $\varepsilon \sim \mathcal{U}([0, 1])$. Figure 5 shows the MMD and the RMSE results alongside comparisons with other baselines on the GLM and GLU tasks. We observe that RISE-Meta achieves the lowest MMD and RMSE values for both the tasks, thus demonstrating its ability to better generalize to unknown levels of missing values in the data.

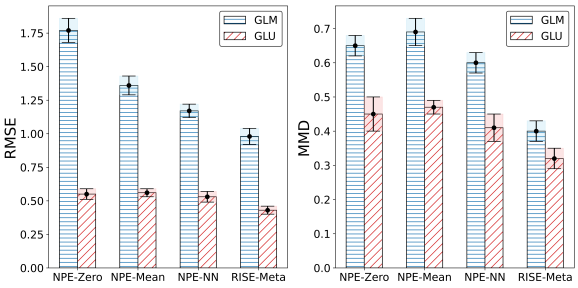


Figure 5: Meta-learning over the missingness level.

5.4 Ablation Studies

Imputation performance on real-world datasets. We now look at how the neural process-based imputation model in RISE performs on imputing real-world datasets. The task is to predict and impute bioactivity data on adrenergic receptor assays (Whitehead et al., 2019) and kinase assays (Martin et al., 2017) from the field of drug discovery. Here, the test data consists of outliers, unlike the training data, which makes imputing these datasets a challenging task that assesses RISE's generalization capabilities. We compare RISE's imputation method to other methods from this field such as QSAR (Cherkasov et al., 2014), Conduilt (Whitehead et al., 2019), and Collective Matrix Factorization (CMF) (Singh and Gordon, 2008). We also include a standard deep neural network (DNN) and a vanilla neural process as baselines. Table 1 (left) reports the coefficient of determination R^2 (Wright,

Table 1: **Ablation studies.** (*left*) R^2 scores (↑) on the bioactivity datasets. (*right*) RMSE (↓) across different proportions of missingness (ε) for comparing the effect of joint versus separate learning.

Method	Adrenergic	Kinase
QSAR	(N/A)	-0.19 ± 0.01
CMF	0.59 ± 0.02	-0.11 ± 0.01
DNN	0.60 ± 0.05	0.11 ± 0.01
NP	0.61 ± 0.03	0.17 ± 0.04
Conduilt	0.62 ± 0.04	0.22 ± 0.03
CNP	0.65 ± 0.04	0.24 ± 0.02
RISE	**0.67** ± 0.03	**0.26** ± 0.03

Missingness (ϵ)	**Method**	**GLM**	**GLU**
10%	NPE-RF-Sep	0.69 ±0.03	0.44 ±0.02
	RISE-Sep	0.67 ±0.03	0.43 ±0.02
	RISE	**0.65** ±0.04	**0.41**±0.01
25%	NPE-RF-Sep	1.02 ±0.05	0.48 ±0.02
	RISE-Sep	0.99 ±0.03	0.45 ±0.02
	RISE	**0.93** ±0.06	**0.43** ±0.02
60%	NPE-RF-Sep	1.34 ±0.10	0.64 ±0.02
	RISE-Sep	1.31 ±0.03	0.58 ±0.03
	RISE	**1.27** ±0.01	**0.56** ±0.03

1921) between the true and the predicted assays. We observe that RISE achieves state-of-the-art results in these tasks, demonstrating the efficacy of the neural processes-based imputation model.

Joint vs separate learning. Finally, our last experiment involves investigating the impact of training the imputation and the inference model in RISE jointly (as we proposed) versus separately (termed RISE-Sep). We also include another baseline termed NPE-RF-Sep where a random forest (RF) model is first used for imputation, followed by NPE. Table 1 (right) reports the RMSE values on GLM and GLU tasks for different missingness proportion ε. We observe that training the imputation and inference networks jointly yields improvement in the performance over training them separately.

6 Conclusion and Limitations

We analyzed the problem of performing SBI under missing data, and showed that inaccurately imputing the missing values leads to bias in the resulting posterior distributions. We then proposed RISE: a method that aims to reduce this bias under different notions of missingness mechanism. RISE combines the inference network of NPE with an imputation model based on neural processes to achieve robustness to missing data whilst being amortized. Additionally, RISE can be trained in a meta-learning manner over the proportion of missing values in the data, thus allowing for amortization across datasets with varying levels missingness. While RISE offers substantial advantages, there are limitations to address. RISE inherits the issues of NPE and may yield posteriors that are not well-calibrated, as found by Hermans et al. (2022). Moreover, the normality assumption in neural processes may have limited expressivity when learning a complex imputation distribution.

Reproducibility Statement

Appendix A.4 provides extensive detail about the dataset used, the method's parameterization, and training details. We have utilized open source datasets and libraries for implementation and evaluation of our method.

References

Ayush Bharti, Francois-Xavier Briol, and Troels Pedersen. A general method for calibrating stochastic radio channel models with kernels. *IEEE Transactions on Antennas and Propagation*, 70(6):3986–4001, 2022a. doi: 10.1109/tap.2021.3083761. 1

Ayush Bharti, Louis Filstroff, and Samuel Kaski. Approximate Bayesian computation with domain expert in the loop. In *International Conference on Machine Learning*, volume 162, pages 1893–1905, 2022b. 1

F-X. Briol, A. Barp, A. B. Duncan, and M. Girolami. Statistical inference for generative models with maximum mean discrepancy. *arXiv:1906.05944*, 2019. 1

Francesco Paolo Casale, Adrian Dalca, Luca Saglietti, Jennifer Listgarten, and Nicolo Fusi. Gaussian process prior variational autoencoders. *Advances in neural information processing systems*, 31, 2018. 2

Jeffrey Chan, Valerio Perrone, Jeffrey Spence, Paul Jenkins, Sara Mathieson, and Yun Song. A likelihood-free inference framework for population genetic data using exchangeable neural networks. *Advances in neural information processing systems*, 31, 2018. 16

Yanzhi Chen, Dinghuai Zhang, Michael U. Gutmann, Aaron Courville, and Zhanxing Zhu. Neural approximate sufficient statistics for implicit models. In *International Conference on Learning Representations*, 2021. 8

Artem Cherkasov, Eugene N Muratov, Denis Fourches, Alexandre Varnek, Igor I Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C Martin, Roberto Todeschini, et al. Qsar modeling: where have you been? where are you going to? *Journal of medicinal chemistry*, 57(12): 4977–5010, 2014. 9

Mark Collier, Alfredo Nazabal, and Christopher KI Williams. Vaes in the presence of missing data. *arXiv preprint arXiv:2006.05301*, 2020. 2

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. 1

Maximilian Dax, Stephen R. Green, Jonathan Gair, Jakob H. Macke, Alessandra Buonanno, and Bernhard Schölkopf. Real-time gravitational wave science with neural posterior estimation. *Physical Review Letters*, 127(24):241103, December 2021. ISSN 1079-7114. doi: 10.1103/physrevlett.127.241103. 1

Charita Dellaporta, Jeremias Knoblauch, Theodoros Damoulas, and Francois-Xavier Briol. Robust bayesian inference for simulator-based models via the mmd posterior bootstrap. In *International Conference on Artificial Intelligence and Statistics*, volume 151, pages 943–970, 2022. 1

Yann Dubois, Jonathan Gordon, and Andrew YK Foong. Neural process family. http://yanndubs.github.io/Neural-Process-Family/, September 2020. 6

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019. 24

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. *Advances in Neural Information Processing Systems*, 33:8284–8295, 2020. 2, 5, 6, 17

Vincent Fortuin, Dmitry Baranchuk, Gunnar Ratsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*, pages 1651–1661. PMLR, 2020. 2

David T. Frazier, Christian P. Robert, and Judith Rousseau. Model misspecification in approximate bayesian computation: consequences and diagnostics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(2):421–444, 2020. doi: 10.1111/rssb.12356. 1

Masahiro Fujisawa, Takeshi Teshima, Issei Sato, and Masashi Sugiyama. γ-abc: Outlier-robust approximate bayesian computation based on a robust divergence estimator. In *International Conference on Artificial Intelligence and Statistics*, volume 130, pages 1783–1791, 2021. 1

Richard Gao, Michael Deistler, and Jakob H Macke. Generalized bayesian inference for scientific simulators via amortized cost estimation. *Advances in Neural Information Processing Systems*, 36, 2023. 1, 8

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International conference on machine learning*, pages 1704–1713. PMLR, 2018. 5, 6, 17

Tomas Geffner, George Papamakarios, and Andriy Mnih. Compositional score modeling for simulation-based inference. In *International Conference on Machine Learning*, pages 11098–11116. PMLR, 2023. 16

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995. 4

Sahra Ghalebikesabi, Rob Cornish, Chris Holmes, and Luke Kelly. Deep generative missingness pattern-set mixture models. In *International conference on artificial intelligence and statistics*, pages 3727–3735. PMLR, 2021a. 7

Sahra Ghalebikesabi, Rob Cornish, Luke J. Kelly, and Chris Holmes. Deep generative pattern-set mixture models for nonignorable missingness, 2021b. 2

Manuel Gloeckler, Michael Deistler, and Jakob H Macke. Adversarial robustness of amortized bayesian inference. *arXiv preprint arXiv:2305.14984*, 2023. 1, 8

Manuel Gloeckler, Michael Deistler, Christian Dietrich Weilbach, Frank Wood, and Jakob H. Macke. All-in-one simulation-based inference. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 15735–15766. PMLR, 21–27 Jul 2024. 1, 6

Yu Gong, Hossein Hajimirsadeghi, Jiawei He, Thibaut Durand, and Greg Mori. Variational selective autoencoder: Learning from partially-observed heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 2377–2385. PMLR, 2021. 7

JW Graham, AE Olchowski, and TD Gilreath. Review: A gentle introduction to imputation of missing values. *Prev. Sci*, 8:206–213, 2007. 1, 6

David Greenberg, Marcel Nonnenmacher, and Jakob Macke. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pages 2404–2414. PMLR, 2019. 1, 7, 20

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 7

Joeri Hermans, Arnaud Delaunoy, François Rozet, Antoine Wehenkel, Volodimir Begy, and Gilles Louppe. A crisis in simulation-based inference? beware, your posterior approximations can be unfaithful. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=LHAbHkt6Aq. 10, 24

Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952. 8

Daolang Huang, Ayush Bharti, Amauri Souza, Luigi Acerbi, and Samuel Kaski. Learning robust statistics for simulation-based inference under model misspecification. *Advances in Neural Information Processing Systems*, 36, 2023. 1

Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. not-miwae: Deep generative modelling with missing not at random data. *arXiv preprint arXiv:2006.12871*, 2020. 2, 3, 5, 7, 19

Milosz Kasak, Kamil Deja, Maja Karwowska, Monika Jakubowska, Lukasz Graczykowski, and Malgorzata Janik. Machine-learning-based particle identification with missing data. *The European Physical Journal C*, 84(7):691, 2024. 1

Ryan P. Kelly, David J Nott, David Tyler Frazier, David J Warne, and Christopher Drovandi. Misspecification-robust sequential neural likelihood for simulation-based inference. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. 1

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 5

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

Theodore Kypraios, Peter Neal, and Dennis Prangle. A tutorial introduction to bayesian inference for stochastic epidemic models using approximate bayesian computation. *Mathematical Biosciences*, 287:42–53, May 2017. ISSN 0025-5564. doi: 10.1016/j.mbs.2016.07.001. 1

Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Misgan: Learning from incomplete data with generative adversarial networks. *International Conference on Learning Representations*, 2019. 2, 7

Julia Linhart, Gabriel Victorino Cardoso, Alexandre Gramfort, Sylvain Le Corff, and Pedro LC Rodrigues. Diffusion posterior sampling for simulation-based inference in tall data settings. *arXiv preprint arXiv:2404.07593*, 2024. 16

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019. 2, 3, 5

Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, 30, 2017a. 1, 6, 7

Jan-Matthis Lueckmann, Pedro J. Gonçalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H. Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems (NIPS)*, page 1289–1299, 2017b. 1, 8

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking simulation-based inference. In *International conference on artificial intelligence and statistics*, pages 343–351. PMLR, 2021. 18, 21

Kieran J Luken, Rabina Padhy, and X Rosalind Wang. Missing data imputation for galaxy redshift estimation. *arXiv preprint arXiv:2111.13806*, 2021. 1

Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems*, 31, 2018. 2

Chao Ma and Cheng Zhang. Identifiable generative models for missing not at random data imputation. *Advances in Neural Information Processing Systems*, 34:27645–27658, 2021. 7

Eric J Martin, Valery R Polyakov, Li Tian, and Rolando C Perez. Profile-qsar 2.0: kinase virtual screening accuracy comparable to four-concentration ic50s for realistically novel compounds. *Journal of chemical information and modeling*, 57(8):2077–2088, 2017. 9

Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning*, pages 4413–4423. PMLR, 2019. 2, 5, 7

Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. Missing data imputation using optimal transport, 2020. 2

Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020. 2, 7

Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013. 18

Priscilla Ong, Manuel Haussmann, and Harri Lahdesmaki. Learning high-dimensional mixed models via amortized variational inference. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024. 2, 19

George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016. 1, 3

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017. 20

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 837–848. PMLR, 2019. 1

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. 3

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 7

Henri Pesonen, Umberto Simola, Alvaro Köhn-Luque, Henri Vuollekoski, Xiaoran Lai, Arnoldo Frigessi, Samuel Kaski, David T Frazier, Worapree Maneesoonthorn, Gael M Martin, et al. Abc of the future. *International Statistical Review*, 91(2):243–268, 2023. 1

Martin Pospischil, Maria Toledo-Rodriguez, Cyril Monier, Zuzanna Piwkowska, Thierry Bal, Yves Frégnac, Henry Markram, and Alain Destexhe. Minimal hodgkin–huxley type models for different classes of cortical and thalamic neurons. *Biological cybernetics*, 99:427–441, 2008. 19

Stefan T. Radev, Ulf K. Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. Bayesflow: Learning complex stochastic models with invertible neural networks, 2020. URL https://arxiv.org/abs/2003.06281. 1

Trivellore E Raghunathan, James M Lepkowski, John Van Hoewyk, Peter Solenberger, et al. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey methodology*, 27(1):85–96, 2001. 4

Siddharth Ramchandran, Gleb Tikhonov, Kalle Kujanpää, Miika Koskinen, and Harri Lähdesmäki. Longitudinal variational autoencoder. In *International Conference on Artificial Intelligence and Statistics*, pages 3898–3906. PMLR, 2021. 2

Carl Rasmussen. The infinite gaussian mixture model. *Advances in neural information processing systems*, 12, 1999. 6

Adam J. Riesselman, John B. Ingraham, and Debora S. Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822, September 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0138-4. 1

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

Joseph L Schafer and Nathaniel Schenker. Inference with imputed conditional means. *Journal of the American Statistical Association*, 95(449):144–154, 2000. 4

Marvin Schmitt, Paul-Christian Bürkner, Ullrich Köthe, and Stefan T Radev. Detecting model misspecification in amortized bayesian inference with neural networks. In *DAGM German Conference on Pattern Recognition*, pages 541–557. Springer, 2023. 1

Maksim Sinelnikov, Manuel Haussmann, and Harri Lähdesmäki. Latent variable model for high-dimensional point process with structured missingness. *arXiv preprint arXiv:2402.05758*, 2024. 19

Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658, 2008. 9

Scott A. Sisson. *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC, 2018. 1

Marek Śmieja, Lukasz Struski, Jacek Tabor, Bartosz Zieliński, and Przemysław Spurek. Processing of missing data by neural networks. *Advances in neural information processing systems*, 31, 2018. 7

Alvaro Tejero-Cantero, Jan Boelts, Michael Deistler, Jan-Matthis Lueckmann, Conor Durkan, Pedro J. Gonçalves, David S. Greenberg, and Jakob H. Macke. sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, 2020. doi: 10.21105/joss.02505. 18

Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011. 2

Vy Vo, He Zhao, Trung Le, Edwin V Bonilla, and Dinh Phung. Optimal transport for structure learning under missing data. *arXiv preprint arXiv:2402.15255*, 2024. 2

Bingjie Wang, Joel Leja, Ashley Villar, and Joshua S Speagle. Monte carlo techniques for addressing large errors and missing data in simulation-based inference. *arXiv preprint arXiv:2211.03747*, 2022. 6

Bingjie Wang, Joel Leja, V Ashley Villar, and Joshua S Speagle. Sbi++: Flexible, ultra-fast likelihood-free inference customized for astronomical applications. *The Astrophysical Journal Letters*, 952 (1):L10, 2023. 6

Zijian Wang, Jan Hasenauer, and Yannik Schälte. Missing data in amortized simulation-based neural posterior estimation. *PLOS Computational Biology*, 20(6):e1012184, 2024. 1, 2, 6, 20, 21

Daniel Ward, Patrick Cannon, Mark Beaumont, Matteo Fasiolo, and Sebastian M Schmon. Robust neural posterior estimation and statistical model criticism. In *Advances in Neural Information Processing Systems*, 2022. 1

Thomas M Whitehead, Benedict WJ Irwin, P Hunt, Matthew D Segall, and Gareth John Conduit. Imputation of assay bioactivity data using deep learning. *Journal of chemical information and modeling*, 59(3):1197–1204, 2019. 9

Simon N. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466 (7310):1102–1104, 2010. doi: 10.1038/nature09319. 2, 8

Sewall Wright. Correlation and causation. *Journal of agricultural research*, 20(7):557, 1921. 9

Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, pages 5689–5698. PMLR, 2018. 2, 7

Seongwook Yoon and Sanghoon Sull. Gamin: Generative adversarial multiple imputation network for highly missing data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8456–8464, 2020. 2

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017. 3, 16

He Zhao, Ke Sun, Amir Dezfouli, and Edwin V Bonilla. Transformed distribution matching for missing value imputation. In *International Conference on Machine Learning*, pages 42159–42186. PMLR, 2023. 2

Xiang Zhou and Jerome P Reiter. A note on bayesian inference after multiple imputation. *The American Statistician*, 64(2):159–163, 2010. 4

A APPENDIX

In Appendix A.2, we present the proofs for Proposition 1 and Proposition 2. Appendix A.3 contains the background on neural processes, and Appendix A.4 presents the implementation details for the experiments of Section 5. Finally, Appendix A.5 contains some additional experimental results.

A.1 DISCUSSION

Handling multiple observations. Although so far we have focused on the single observation case where we have one data vector $\mathbf{x}$ for each θ, RISE can straightforwardly be extended to the multiple observations case where we obtain $\mathbf{x}^{(1:m)} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ for each θ. Then, $\mathbf{x}^{(1:m)} = (\mathbf{x}_{\text{obs}}^{(1:m)}, \mathbf{x}_{\text{mis}}^{(1:m)})$, and the objective for RISE becomes

$$\arg\min_{\phi,\varphi,\kappa} - \mathbb{E}_{(\mathbf{x}_{\text{obs}}^{(1:m)},\theta)\sim p_{\text{true}}} \mathbb{E}_{\mathbf{x}_{\text{mis}}^{(1:m)}\sim\prod_{i=1}^{m} p(\mathbf{x}_{\text{mis}}^{(i)}|\mathbf{x}_{\text{obs}}^{(i)})} \left[\frac{1}{m}\sum_{i=1}^{m} \log \underbrace{\hat{p}_\varphi(\mathbf{x}_{\text{mis}}^{(i)} \mid \mathbf{x}_{\text{obs}}^{(i)})}_{\textbf{(imputation)}} + \log \underbrace{q_\phi(\theta \mid \eta_\kappa(\mathbf{x}_{\text{obs}}^{1:m}, \mathbf{x}_{\text{mis}}^{1:m}))}_{\textbf{(inference)}} \right].$$

Note that we can summarize the data using the network η_κ (for instance, a deep set (Zaheer et al., 2017)) before passing the data into the inference network, which is standard practice when using NPE with multiple observations (Chan et al., 2018). Alternatively, one could use recent extensions based on score estimation (Geffner et al., 2023; Linhart et al., 2024) as well.

Handling model misspecification. We conjecture that replacing the inference network in RISE from the usual NPE to its robust variant such as the method of Ward et al. (2022) or Huang et al. (2023) would help in addressing model misspecification issues. It would be an interesting avenue for future research to see how to train these robust NPE methods jointly with the imputation network of RISE, and how effective such an approach is. One way is to assume a certain error model over the observed data $\mathbf{x}$ and corrupt the data $\tilde{\mathbf{x}}$ via adding a Gaussian noise, and infer the correct θ via the inference network. This can be described also via the objective as

$$\text{argmin}_{\phi,\varphi} - \mathbb{E}_{(\mathbf{x}_{\text{obs}},\theta)\sim p(\mathbf{x}_{\text{obs}},\theta),\tilde{\mathbf{x}}_{\text{obs}}\sim\mathcal{N}(\mathbf{x}_{\text{obs}},\sigma^2),\tilde{\mathbf{x}}_{\text{mis}}\sim p_{\text{true}}(\tilde{\mathbf{x}}_{\text{mis}}|\tilde{\mathbf{x}}_{\text{obs}},\theta)} \left[\log \hat{p}_\varphi(\tilde{\mathbf{x}}_{\text{mis}} \mid \tilde{\mathbf{x}}_{\text{obs}}) + \log q_\phi(\theta \mid \tilde{\mathbf{x}}_{\text{obs}}, \tilde{\mathbf{x}}_{\text{mis}})\right]. \tag{9}$$

Moreover, this can also be readily extended to incorporate prior miss-specification via similar way as,

$$\text{argmin}_{\phi,\varphi} - \mathbb{E}_{(\mathbf{x}_{\text{obs}},\theta)\sim p(\mathbf{x}_{\text{obs}},\theta),\tilde{\theta}\sim\mathcal{N}(\theta,\sigma^2),\tilde{\mathbf{x}}_{\text{mis}}\sim p_{\text{true}}(\tilde{\mathbf{x}}_{\text{mis}}|\tilde{\mathbf{x}}_{\text{obs}},\tilde{\theta})} \left[\log \hat{p}_\varphi(\tilde{\mathbf{x}}_{\text{mis}} \mid \tilde{\mathbf{x}}_{\text{obs}}) + \log q_\phi(\theta \mid \tilde{\mathbf{x}}_{\text{obs}}, \tilde{\mathbf{x}}_{\text{mis}})\right]. \tag{10}$$

A.2 PROOFS

A.2.1 PROOF FOR PROPOSITION 1

Proof. Using Equation (2) and Equation (3), the bias in the estimated posterior can be written as

$$\begin{aligned}
&\left|\mathbb{E}_{\theta\sim p_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}})}[\theta \mid \mathbf{x}_{\text{obs}}] - \mathbb{E}_{\theta\sim \hat{p}_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}})}[\theta \mid \mathbf{x}_{\text{obs}}]\right| \\
&= \int \theta p_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}}) d\theta - \int \theta \hat{p}_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}}) d\theta \\
&= \int \theta \left[p_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}}) - \hat{p}_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}})\right] d\theta \\
&= \int \theta \left[\int p_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}}) p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) d\mathbf{x}_{\text{mis}} - \int p_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}}) \hat{p}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) d\mathbf{x}_{\text{mis}}\right] d\theta \\
&= \int p_{\text{SBI}}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}}) \theta \int \left[p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) - \hat{p}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})\right] d\mathbf{x}_{\text{mis}} d\theta \\
&= \mathbb{E}_{\theta\sim p_{\text{SBI}}(\cdot \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})}\left[\theta \int \left[p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) - \hat{p}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})\right] d\mathbf{x}_{\text{mis}}\right]
\end{aligned}$$

The bias is therefore zero only when $\hat{p}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$ is aligned with $p_{\text{true}}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$, which completes the proof. □

A.2.2 Proof for Proposition 2

Proof. Recall our optimization problem from Equation (5):

$$\arg\min_{\psi} \mathbb{E}_{\mathbf{x}_{\text{obs}} \sim p_{\text{true}}} \text{KL}[p_{\text{true}}(\theta \mid \mathbf{x}_{\text{obs}}) \mid\mid r_{\psi}(\theta \mid \mathbf{x}_{\text{obs}})] \,.$$

Expanding the KL term, we note that the above is equivalent to

$$\arg\min_{\psi} \mathbb{E}_{\mathbf{x}_{\text{obs}} \sim p_{\text{true}}} \mathbb{E}_{\theta \sim p_{\text{true}}(\theta \mid \mathbf{x}_{\text{obs}})} \log \left(\frac{p_{\text{true}}(\theta \mid \mathbf{x}_{\text{obs}})}{r_{\psi}(\theta \mid \mathbf{x}_{\text{obs}})} \right) \,.$$

Since $p_{\text{true}}(\theta \mid \mathbf{x}_{\text{obs}})$ does not depend on ψ, we immediately note that the problem is equivalent to

$$\begin{aligned} &\text{argmin}_{\psi} \, \mathbb{E}_{\mathbf{x}_{\text{obs}} \sim p_{\text{true}}} \, \mathbb{E}_{p_{\text{true}}(\theta \mid \mathbf{x}_{\text{obs}})}[-\log \, r_{\psi}(\theta \mid \mathbf{x}_{\text{obs}})] \\ &= \arg\max_{\psi} \, \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \left[\log \, r_{\psi}(\theta \mid \mathbf{x}_{\text{obs}})\right] \,. \end{aligned}$$

We now obtain a lower bound for $\mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \left[\log \, r_{\psi}(\theta \mid \mathbf{x}_{\text{obs}})\right]$. Formally, we have

$$\begin{aligned} \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}}[\log r_{\psi}(\theta \mid \mathbf{x}_{\text{obs}})] &= \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \log \int r_{\psi}(\theta, \mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) d\mathbf{x}_{\text{mis}} \\ &= \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \log \int \frac{p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) r_{\psi}(\theta, \mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})}{p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})} d\mathbf{x}_{\text{mis}} \\ &\geq \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \mathbb{E}_{\mathbf{x}_{\text{mis}} \sim p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})} \left[\log \frac{r_{\psi}(\theta, \mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})}{p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})}\right] \\ &= \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \mathbb{E}_{\mathbf{x}_{\text{mis}} \sim p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})} \left[\log \frac{r_{\psi}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) r_{\psi}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})}{p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})}\right] \,, \end{aligned}$$

where we invoked the Jensen's inequality to swap the log and the conditional expectation. Splitting parameters ψ into imputation parameters φ and inference parameters ϕ, and denoting the corresponding imputation and inference networks by $\hat{p}_{\varphi}$ and q_{ϕ} respectively, we immediately get

$$\mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}}[\log r_{\phi, \varphi}(\theta \mid \mathbf{x}_{\text{obs}})] \geq \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \mathbb{E}_{\mathbf{x}_{\text{mis}} \sim p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})} \left[\log \frac{\hat{p}_{\varphi}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) q_{\phi}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})}{p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})}\right] \,.$$

Thus, we obtain the following variational objective:

$$\begin{aligned} &\text{argmax}_{\phi, \varphi} \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \mathbb{E}_{\mathbf{x}_{\text{mis}} \sim p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})} \left[\log \frac{\hat{p}_{\varphi}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) q_{\phi}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})}{p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})}\right] \\ &= \text{argmax}_{\phi, \varphi} \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \left(\mathbb{E}_{\mathbf{x}_{\text{mis}} \sim p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})} \left[\log \hat{p}_{\varphi}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}}) + \log q_{\phi}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})\right] \, + \, \mathbb{H}(p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})) \right) \\ &= \text{argmax}_{\phi, \varphi} \mathbb{E}_{(\mathbf{x}_{\text{obs}}, \theta) \sim p_{\text{true}}} \mathbb{E}_{\mathbf{x}_{\text{mis}} \sim p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})} \left[\log \underbrace{\hat{p}_{\varphi}(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})}_{\text{imputation}} + \log \underbrace{q_{\phi}(\theta \mid \mathbf{x}_{\text{obs}}, \mathbf{x}_{\text{mis}})}_{\text{inference}} \right] \,, \end{aligned}$$

since the entropy term $\mathbb{H}(p(\mathbf{x}_{\text{mis}} \mid \mathbf{x}_{\text{obs}})$ does not depend on the optimization variables ϕ and φ.

□

A.3 Neural Process

Neural Process (Garnelo et al., 2018; Foong et al., 2020) models the predictive distribution over target locations $\mathbf{x}_t$ by, (i) constructing a learnable mapping f_{γ} from the context set $(\mathbf{x}_c, \mathbf{y}_c)$ to a latent representation $\mathbf{r}$ as,

$$\mathbf{r} = f_{\gamma}(\mathbf{x}_c, \mathbf{y}_c) \tag{11}$$

and then (ii) utilizing the representation $\mathbf{r}$ to approximate the predictive distribution, given the target locations $\mathbf{x}_t$, via a learnable decoder g_ω as,

$$p(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{r}) = g_\omega(\mathbf{r}, \mathbf{x}_t) \tag{12}$$

where $\mathbf{x}_c, \mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ are the input vectors (often locations or positions) and $\mathbf{y}_c, \mathbf{y}_t \in \mathcal{Y} \subseteq \mathbb{R}^{d_y}$ be the output vectors, with $d_x, d_y \geq 1$ their dimensionality. In practice, the predictive distribution is often assumed to be gaussain as,

$$p(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{r}) = \prod_{m=1}^{M} p(\mathbf{y}_{t,m} \mid \mathbf{x}_{t,m}, \mathbf{r}) = \prod_{m=1}^{M} \mathcal{N}(\mathbf{y}_{t,m} \mid \mu_{\omega,m}, \sigma^2_{\omega,m}) \tag{13}$$

where $\mu_{\omega,m}, \sigma_{\omega,m} = g_\omega(\mathbf{r}, \mathbf{x}_{t,m})$. For a fixed context $(\mathbf{x}_c, \mathbf{y}_c)$, using Kolmogorov's extension theorem (Oksendal, 2013) the collection of these finite dimensional distributions defines a stochastic process if these are consistent under (i) permutations of any entries of $(\mathbf{x}_t, \mathbf{y}_t)$ and (ii) marginalisations of any entries of $\mathbf{y}_t$.

A.4 Implementation Details

This section is arranged as follows:

- Appendix A.4.1: Description of SBI benchmarking simulators
- Appendix A.4.2: Prior distributions used for the SBI experiments
- Appendix A.4.3: Procedure for creating the missingness mask under MCAR and MNAR
- Appendix A.4.4: Details of the neural network settings

A.4.1 Model descriptions

Ricker model simulates the temporal evolution of population size in ecological systems. In this model, the population size N_t at time t evolves as $N_{t+1} = N_t \exp(\theta_1) \exp(N_t + e_t), t = 1, \ldots, T$. The parameter $\exp(\theta_1)$ represents the growth rate, while e_t denotes independent and identically distributed Gaussian noise terms with zero mean and variance σ_e^2. The initial population size is set to $N_0 = 1$. Observations x_t are modeled as Poisson random variables with rate parameter $\theta_2 N_t$, such that $x_t \sim \text{Poiss}(\theta_2 N_t)$. For our simulations, we fixed $\sigma_e^2 = 0.09$ and focused on estimating the parameter vector $\theta = [\theta_1, \theta_2]^\top$. The prior distribution is set as a uniform distribution $\mathcal{U}([2, 8] \times [0, 20])$. We simulated the process for $T = 100$ time steps to generate sufficient data for inference, and considered a simulation budget of 1000 to create the dataset.

Ornstein-Uhlenbeck process (OUP) is a stochastic differential equation model widely used in financial mathematics and evolutionary biology. The OU process x_t is defined as,

$$x_{t+1} = x_t + \Delta x_t, \quad t = 1, \ldots, T \tag{14}$$

$$\Delta x_t = \theta_1[\exp(\theta_2) - x_t]\Delta t + 0.5w \tag{15}$$

where $T = 25$, $\Delta t = 0.2$, $x_0 = 10$ and $w \sim \mathcal{N}(0, \Delta t)$.

Generalized Linear Model (GLM). A 10 parameter Generalized Linear Model (GLM) with Bernoulli observations.

Gaussian Linear Uniform (GLU). A 10 dimensional Gaussian model, in which the parameter θ is the mean, and the covariance is fixed, with a uniform prior. We refer to Lueckmann et al. (2021); Tejero-Cantero et al. (2020) for further details on these SBI tasks.

Hodgkin Huxley Model. Hodgkin Huxley Model is a real-world computational neuroscience simulator. It describes the intricate dynamics of the generation and propagation of action potentials along neuronal membranes with the capture of the time course of membrane voltage by modeling the behavior of ion channels, particularly sodium and potassium, as well as leak currents. It consists of two parameters: $\theta_1 \equiv \bar{g}_{\text{Na}}$, and $\theta_2 \equiv \bar{g}_{\text{K}}$, which describe the density of Na and K specifically. The dynamics are parameterized as a set of differential equations,

$$C_m \frac{dV}{dt} = g_1(E - V) + \theta_1 m^3 h(E_{\text{Na}} - V) + \theta_2 n^4 h(E_{\text{K}} - V) + \bar{g}_{\text{M}} p(E_{\text{M}} - V) + I_{\text{inj}} + \sigma\eta(t)$$

$$\frac{dq}{dt} = \frac{q_\infty(V) - q}{\tau_q(V)}, \quad q \in \{m, h, n, p\}$$

Here, V represents the membrane potential, C_m the membrane capacitance, g_1 is the leak conductance, E_1 is the membrane reverse potential, θ_1, θ_2 are the densities of Na and K channel, $\bar{g}_{\text{M}}$ is the density for M channel, $E_{\text{Na,K,M}}$ are the reversal potential and $\sigma\eta(t)$ is the intrinsic neural noise. The right hand side of the voltage dynamics is composed of a leak current, a voltage-dependent Na^+, a delayed rectifier K^+, a slow voltage-dependent K^+ current responsible for spike-frequency adaptation, and an injected current I_{inj}. Channel gating variables q have dynamics fully characterized by the neuron membrane potential V, given the respective steady-state $q_\infty(V)$ and time constant $\tau_q(V)$. For more details, see Pospischil et al. (2008).

A.4.2 Prior distributions

We utilize the following prior distributions for our experiment tasks:

- Ricker: Uniform distribution $\mathcal{U}([2, 8] \times [0, 20])$
- OUP: Uniform prior $\mathcal{U}([0, 2] \times [-2, 2])$
- Hodgkin-Huxley: Uniform distribution $\mathcal{U}([10^{-4}, -0.5] \times [15.0, 100.0])$
- GLU: Uniform distribution $\mathcal{U}([-1, 1]^{10})$
- GLM: A multivariate normal $\mathcal{N}(0, (\mathbf{F}^\top \mathbf{F})^{-1})$ computed as follows,

$$\mathbf{F}_{i,i-2} = 1, \mathbf{F}_{i,i-1} = -2, \mathbf{F}_{i,i} = 1 + \sqrt{\frac{i-1}{9}}, \mathbf{F}_{i,j} = 0 \text{ otherwise}, 1 \leq i, j \leq 9 \tag{16}$$

A.4.3 Creating the missingness mask

MCAR. We followed random masking approach to simulate the MCAR scenario. For a given missingness degree ε, we randomly mask out $\varepsilon\%$ of the data sample.

MNAR. We employed the *self-masking* or *self-censoring* approach as outlined by Ipsen et al. (2020). For a given data sample $\mathbf{x} \in \mathbb{R}^d$, and following Sinelnikov et al. (2024); Ong et al. (2024), the probability of a particular data-point to be missing depends on its value. Specifically, we sample the mask s_i for i^{th} value for data sample as,

$$s_i \sim \text{Bern}(p_i), \quad p_i = \varepsilon \cdot \frac{x_i}{\max_d(\mathbf{x})} \tag{17}$$

where $0 \leq i \leq d$, $\max_d(\mathbf{x})$ represents the maximum value in the data sample and p_i is the masking probability for data-point x_i which is computed using the proportion of missing values ε.

A.4.4 Network Parametrization

Summary Networks. For the Ricker and Huxley model, the summary network is composed of 1D convolutional layers, whereas for the OUP, it is a combination of bidirectional long short-term memory (LSTM) recurrent modules and 1D convolutional layers. The dimension of the statistic space is set to four for both the models. We do not use summary networks for GLM and GLU.

Imputation Model. The parameters for the neural process-based imputation model used in RISE are given in Table 2 and Table 3.

Table 2: Default hyperparameters for imputation model $\hat{p}_\varphi$ for Ricker, OUP and Huxley model.

Module	Hyperparameter	Meaning	Value
Encoder	CNN blocks	Number of CNN layers	1
	Hidden dimension	Number of output channels of each CNN layer	64
	Kernel size	Kernel size of each convolution layer	9
	Stride	Stride of each convolution layer	1
	Padding	Padding size of each convolution layer	4
Latent	CNN blocks	Number of CNN layers	2
	Hidden dimension	Number of output channels of each CNN layer	32
	Kernel size	Kernel size of each convolution layer	3
	Stride	Stride of each convolution layer	1
	Padding	Padding size of each convolution layer	1
Decoder	CNN blocks	Number of CNN layers	[6,1]
	Hidden dimension	Number of output channels of each CNN layer	[32,2]
	Kernel size	Kernel size of each convolution layer	5
	Stride	Stride of each convolution layer	1
	Padding	Padding size of each convolution layer	2

Table 3: Default hyperparameters for imputation model $\hat{p}_\varphi$ for GLM and GLU.

Module	Hyperparameter	Meaning	Value
Encoder	MLP blocks	Number of MLP layers	[1,1]
	Hidden dimension	Number of output channels of each MLP layer	[32,64]
Latent	MLP blocks	Number of MLP layers	2
	Hidden dimension	Number of output channels of each MLP layer	32
Decoder	MLP blocks	Number of MLP layers	[6,1]
	Hidden dimension	Number of output channels of each MLP layer	[32,10]

Inference model. Our inference model implementations are based on publicly available code from the sbi library `https://github.com/mackelab/sbi`. We use the NPE-C model (Greenberg et al., 2019) with Masked Autoregressive Flow (MAF) (Papamakarios et al., 2017) as the backbone inference network, and adopt the default configuration with 20 hidden units and 5 transforms for MAF. Throughout our experiments, we maintained a consistent batch size of 50 and a fixed learning rate of 5×10^{-4}.

A.5 Additional Results

In Table 4, we report the RMSE values for the experiment on SBI benchmark simulators presented in Section 5.1. Similar to the MMD results of Figure 3, we observe that RISE yields lowest RNSE for almost all the cases, especially for Ricker and OUP where RISE beats the baselines comprehensively.

Figure 6 shows how accurate our proposed method in imputing the values of missing data simulated from the SBI benchmark models compared to the baselines. The performance is measured in terms of RMSE of the imputed values. Our method (denoted in red) performs the best in imputing the missing values, which eventually helps in accurate estimation of the posterior distribution, as we saw in Figure 3 and Table 4.

B Additional Results

B.1 Comparison with (Wang et al., 2024)

We have performed a study to compare to the methods in Wang et al. (2024), by using data augmentation with constant values and a binary mask indicator while training the NPE on GLU and GLM data sets at various levels of missingness. The table 5 presents the results, and we observe RISE performs better in across all the metrics even at higher level of missingness.

Table 4: **RMSE between predicted posterior samples and SBI posterior.**

Missingness (ϵ)	Method	Ricker		OUP		GLM		GLU	
		MCAR	MNAR	MCAR	MNAR	MCAR	MNAR	MCAR	MNAR
10%	NPE-Zero	1.17 ±0.04	1.74 ±0.05	2.95 ±0.06	0.64 ±0.03	0.78 ±0.03	0.75 ±0.04	0.44 ±0.02	0.44 ±0.03
	NPE-Mean	0.74 ±0.05	1.83 ±0.04	0.99 ±0.05	0.54 ±0.06	0.75±0.04	0.75 ±0.04	0.43 ±0.02	0.44 ±0.04
	NPE-NN	0.47 ±0.05	0.96 ±0.07	0.69 ±0.04	0.45 ±0.04	0.70 ±0.10	0.72 ±0.03	0.42 ±0.04	0.43 ±0.03
	RISE	**0.25** ±0.06	**0.33** ±0.03	**0.39** ±0.01	**0.39** ±0.03	**0.65** ±0.04	**0.59** ±0.03	**0.41**±0.01	**0.41** ±0.02
25%	NPE-Zero	1.68 ±0.04	1.65 ±0.04	2.97 ±0.08	1.14 ±0.05	1.07 ±0.04	0.77 ±0.06	0.48 ±0.03	0.47 ±0.01
	NPE-Mean	1.74 ±0.04	1.64 ±0.07	0.97 ±0.01	0.84 ±0.03	1.15 ±0.03	0.77 ±0.02	0.47 ±0.05	0.47 ±0.02
	NPE-NN	1.41 ±0.05	1.38 ±0.03	0.77 ±0.03	0.77 ±0.04	1.12 ±0.08	0.71 ±0.07	0.47 ±0.04	0.40 ±0.02
	RISE	**0.91** ±0.06	**1.06** ±0.03	**0.62** ±0.09	**0.57** ±0.07	**0.93** ±0.06	**0.69** ±0.04	**0.43** ±0.02	**0.33** ±0.04
60%	NPE-Zero	3.05 ±0.05	1.71 ±0.07	2.99 ±0.07	1.94 ±0.05	1.46 ±0.03	1.53 ±0.10	**0.52** ±0.03	0.50 ±0.02
	NPE-Mean	1.89 ±0.04	1.85 ±0.08	1.01 ±0.01	1.74 ±0.06	1.43 ±0.04	1.76 ±0.08	0.54 ±0.01	0.50 ±0.02
	NPE-NN	1.82 ±0.10	1.75 ±0.10	0.65 ±0.07	0.94 ±0.05	1.41 ±0.05	1.63 ±0.04	0.58 ±0.03	0.47 ±0.08
	RISE	**1.38** ±0.06	**1.60** ±0.09	**0.51** ±0.06	**0.67** ±0.03	**1.27** ±0.01	**1.38** ±0.04	0.56 ±0.03	**0.43** ±0.05

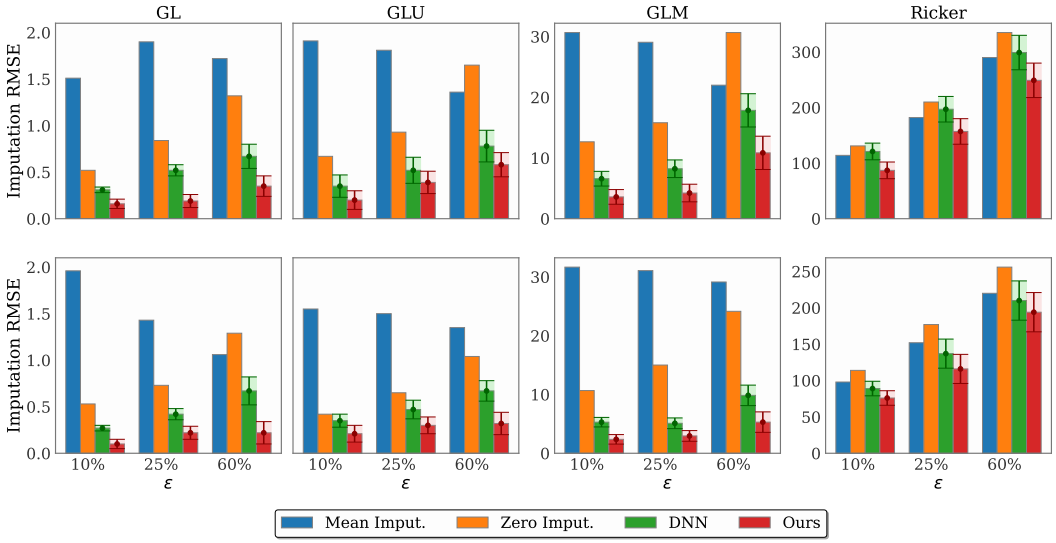


Figure 6: Imputation RMSE for MCAR (**top**) and MNAR (**bottom**) over various synthetic datasets. Here GL refers to a 10 dimension Gaussian linear model, see Lueckmann et al. (2021) for details.

Table 5: Comparisons with Wang et al. (2024).

Missingness	Method	GLU			GLM		
		RMSE	C2ST	MMD	RMSE	C2ST	MMD
10%	Wang et al. (2024)	0.47	0.87	0.28	**0.61**	0.86	0.24
	RISE	**0.41**	**0.83**	**0.18**	0.65	**0.80**	**0.12**
25%	Wang et al. (2024)	0.45	0.92	0.31	**0.86**	0.94	0.35
	RISE	**0.43**	**0.89**	**0.26**	0.93	**0.91**	**0.27**
60%	Wang et al. (2024)	0.65	0.97	0.35	1.53	0.99	**0.45**
	RISE	**0.56**	**0.93**	**0.33**	**1.27**	**0.97**	0.50

B.2 Comparison with Simformer

We have performed a study to benchmark against Simformer over MCAR and MNAR missingness mechanism. To ensure a fair comparison, we utilized the same dataset i.e Generalized Linear Model (GLM) and gaussian linear uniform (GLU) used by Simformer and the training configurations. The results (in terms of MMD and nominal log posterior probability (NLPP)) for both the MCAR and the MNAR case are shown in the table 6 and table 7 on MMD and predicted nominal log posterior probability of the nominal parameters (NLPP). For the MCAR case, both RISE and Simformer perform similarly, with RISE yielding slightly better results. The difference in performance is more

stark in the MNAR case, as expected, since Simformer does not explicitly model the missingness mechanism.

Table 6: Comparison with Simformer. (MCAR)

Missingness	Method	GLU			GLM		
		NLPP	MMD	C2ST	NLPP	MMD	C2ST
10%	Simformer	-2.45 $\pm$ 0.12	0.20 $\pm$ 0.01	0.86 $\pm$ 0.01	-6.47 $\pm$ 0.16	0.17 $\pm$ 0.02	0.84 $\pm$ 0.01
	RISE	**-2.31** $\pm$ 0.10	**0.18** $\pm$ 0.01	**0.83** $\pm$ 0.02	**-6.32** $\pm$ 0.15	**0.12** $\pm$ 0.02	**0.80** $\pm$ 0.02
25%	Simformer	**-3.65** $\pm$ 0.17	0.27 $\pm$ 0.02	0.90 $\pm$ 0.01	-7.37 $\pm$ 0.13	0.31 $\pm$ 0.03	0.92 $\pm$ 0.01
	RISE	-3.71 $\pm$ 0.11	**0.26** $\pm$ 0.01	**0.89**$\pm$ 0.02	**-7.22** $\pm$ 0.17	**0.27** $\pm$ 0.01	**0.91** $\pm$ 0.01
60%	Simformer	-6.62 $\pm$ 0.27	0.39 $\pm$ 0.03	0.95 $\pm$ 0.01	-8.93 $\pm$ 0.18	0.52 $\pm$ 0.03	0.98 $\pm$ 0.01
	RISE	**-6.21** $\pm$ 0.11	**0.33** $\pm$ 0.02	**0.93** $\pm$ 0.01	**-8.71** $\pm$ 0.14	**0.50** $\pm$ 0.01	**0.96** $\pm$ 0.01

Table 7: Comparison with Simformer. (MNAR)

Missingness	Method	GLU			GLM		
		NLPP	MMD	C2ST	NLPP	MMD	C2ST
10%	Simformer	-2.15 $\pm$ 0.10	0.18 $\pm$ 0.01	0.84 $\pm$ 0.02	-6.17 $\pm$ 0.18	0.16 $\pm$ 0.02	0.83 $\pm$ 0.01
	RISE	**-1.90** $\pm$ 0.09	**0.16** $\pm$ 0.01	**0.80** $\pm$ 0.01	**-5.82** $\pm$ 0.11	**0.13** $\pm$ 0.02	**0.79** $\pm$ 0.02
25%	Simformer	**-3.12** $\pm$ 0.12	0.25 $\pm$ 0.02	0.89 $\pm$ 0.01	-6.57 $\pm$ 0.14	0.25 $\pm$ 0.03	0.90 $\pm$ 0.01
	RISE	-3.26 $\pm$ 0.10	**0.22** $\pm$ 0.01	**0.86** $\pm$ 0.02	**-6.12** $\pm$ 0.15	**0.17** $\pm$ 0.01	**0.88** $\pm$ 0.01
60%	Simformer	-6.02 $\pm$ 0.12	0.32 $\pm$ 0.03	0.92 $\pm$ 0.01	-7.56 $\pm$ 0.15	0.50 $\pm$ 0.03	0.95 $\pm$ 0.01
	RISE	**-5.80** $\pm$ 0.13	**0.27** $\pm$ 0.04	**0.90** $\pm$ 0.01	**-7.11** $\pm$ 0.17	**0.47** $\pm$ 0.03	**0.92** $\pm$ 0.01

B.3 C2ST SCORES AND NOMINAL PROBABILITY SCORES.

We have performed a study to evaluate the C2ST scores and predicted nominal log posterior probability of the nominal parameters (NLPP) our method. The table 8 shows the C2ST scores (lower the better) and predicted nominal log posterior probability of the nominal parameters (NLPP, higher the better) for different methods over various benchmark tasks. We observe that RISE outperforms the competing baselines across all the datasets and over all metrics.

Table 8: C2ST Scores (↓) and Nominal Probability (↑) .

Missingness	**Method**	GLU		GLM		Ricker		OUP	
		C2ST	NLPP	C2ST	NLPP	C2ST	NLPP	C2ST	NLPP
10%	NPE-Zero	0.89	-2.77 $\pm$ 0.13	0.87	-6.92 $\pm$ 0.14	0.95	-5.15 $\pm$ 0.12	0.90	-2.61 $\pm$ 0.16
	NPE-Mean	0.88	-2.67 $\pm$ 0.16	0.85	-6.83 $\pm$ 0.14	0.95	-5.10 $\pm$ 0.21	0.90	-2.51 $\pm$ 0.11
	NPE-NN	0.87	-2.51 $\pm$ 0.11	0.84	-6.57$\pm$ 0.13	0.94	-4.90 $\pm$ 0.16	0.89	-2.25 $\pm$ 0.18
	RISE	**0.83**	**-2.31** $\pm$ 0.10	**0.80**	**-6.32** $\pm$ 0.15	0.90	**-4.20** $\pm$ 0.09	**0.87**	**-2.09** $\pm$ 0.11
25%	NPE-Zero	**0.88**	-4.11 $\pm$ 0.17	0.97	-8.05 $\pm$ 0.20	0.96	-5.10 $\pm$ 0.16	0.92	-2.97 $\pm$ 0.13
	NPE-Mean	0.90	-3.99 $\pm$ 0.21	0.94	-7.92 $\pm$ 0.14	0.96	-5.05 $\pm$ 0.11	0.92	-2.84 $\pm$ 0.15
	NPE-NN	0.91	-3.92 $\pm$ 0.11	0.93	-7.72 $\pm$ 0.16	0.95	-4.94$\pm$ 0.17	0.90	-2.74 $\pm$ 0.18
	RISE	0.89	**-3.71** $\pm$ 0.11	**0.91**	**-7.22** $\pm$ 0.17	**0.92**	**-4.64** $\pm$ 0.15	**0.89**	**-2.43** $\pm$ 0.15
60%	NPE-Zero	0.97	-6.98 $\pm$ 0.18	1.00	-9.63 $\pm$ 0.14	0.97	-5.17 $\pm$ 0.15	0.96	-3.07 $\pm$0.12
	NPE-Mean	0.98	-6.76 $\pm$ 0.09	0.99	-9.27 $\pm$ 0.14	0.97	-5.10 $\pm$ 0.18	0.95	-2.97 $\pm$ 0.12
	NPE-NN	0.96	-6.37 $\pm$ 0.12	0.99	-9.02 $\pm$ 0.17	0.96	-4.97 $\pm$ 0.17	0.95	-2.87 $\pm$ 0.19
	RISE	**0.93**	**-6.21** $\pm$ 0.11	**0.97**	**-8.71** $\pm$ 0.14	**0.94**	**-4.72** $\pm$ 0.17	**0.93**	**-2.52** $\pm$ 0.11

B.4 META LEARNING OVER MISSINGNESS

We have also evaluated the performance of our method in generalizing across various levels of missingness as described in section 5.3 over Ricker and OUP dataset. The table 9 displays the result and we observe that our method is able to handle and generalize over unknown levels of missingness in the data.

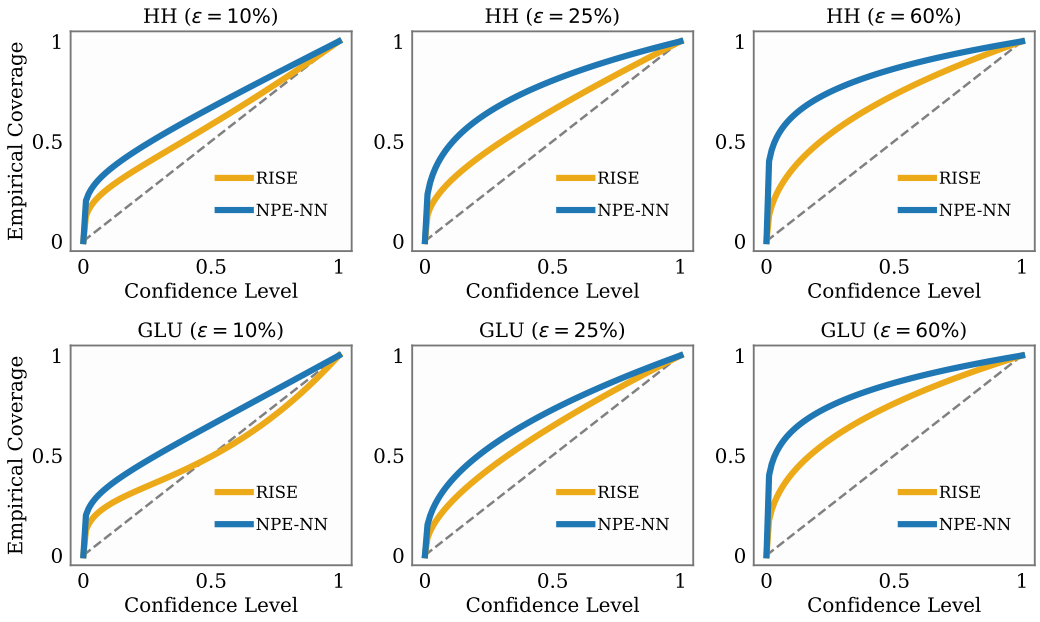


Figure 7: Expected coverage of RISE and NPE-NN for HH (top) and GLU (bottom) task over various level of missingness. The estimator becomes more conservative with increase in missingness due to the lack of information to estimate posterior and imputation distribution.

Table 9: Ablation on flow architectures (*left*) and meta learning the missingness (*right*).

Method	**C2ST**	**RMSE**	**MMD**
RISE-MAF	0.80	0.65	0.12
RISE-NSF	0.80	0.67	0.11

Method	**Ricker**		**OUP**	
	RMSE	MMD	RMSE	MMD
NPE-Zero	3.98	0.95	3.10	0.57
NPE-Mean	2.31	0.67	1.73	0.53
NPE-NN	1.97	0.51	1.32	0.50
RISE-Meta	**1.52**	**0.42**	**0.89**	**0.45**

B.5 Runtime Comparisons

We have performed an ablation study to evaluate the computational complexity of our method in comparison to standard NPE. The table 10 describes the time (in seconds) per epoch to train different models on a single V100 GPU. We observe that there is a minimal increasing in runtime due to the addition of the imputation model. The training time remains the same with respect to missingness levels over a certain data dimensionality.

Table 10: Runtime comparisons (*left*) and Simulation budget comparisons (*right*).

Method	**GLM**	**GLU**
NPE	0.12	0.10
RISE	0.18	0.16

Budget	GLU		GLM	
	C2ST	MMD	C2ST	MMD
1000	0.83	0.18	0.80	0.12
10000	0.78	0.15	0.75	0.10

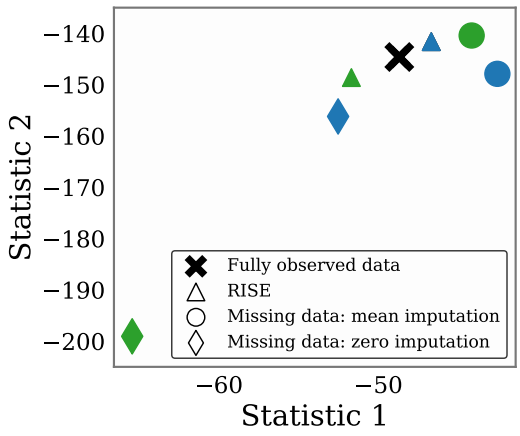
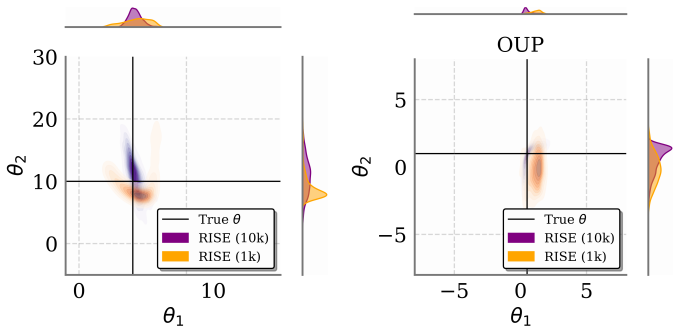


Figure 8: The learned statistics for fully observed and imputed datasets. RISE is able to reduce the shift in the summary statistic. The blue color corresponds to 10% missingness and green for 20% missingness.



(a) Ricker (b) OUP

Figure 9: Visualization of posterior estimated by RISE for Ricker and OUP for 1k and 10k simulation budget under 25% missingness level. We observe that posterior estimate becomes better with increase in simulation budget.

B.6 Performance as function of simulation budget

We have conducted a study to quantify the performance as a function of the simulation budget on GLU and GLM dataset. The table 10 shows C2ST and MMD among various simulation budgets for RISE, for 10% missingness level. As the budget increases, the performance improves.

B.7 Expected coverage plot for Hodgkin-Huxley

We compute the expected coverage (Hermans et al., 2022) of our method on various confidence levels. Figure 7 shows the expected coverage for the HH task at various levels of missingness. We observe that RISE is able to produce conservative posterior approximations and is better calibrated as compared to NPE-NN.

B.8 Additional Visualizations

B.9 Ablation on Flow architecture

We have performed an ablation study to evaluate the performance comparison among different flow architectures. We utilize NSF (Durkan et al., 2019) and MAF as competing architectures for flow

and evaluate it on GLM dataset with 10% missingness level. The table 9 showcases the result and we observe that both NSF and MAF give similar results.