Stochastic Neural Networks for Causal Inference with Missing Confounders

Anonymous authors
Paper under double-blind review

Abstract

One of the major challenges in causal inference with observational data is handling missing confounders. Latent variable modeling offers a valid framework to address this challenge, but existing approaches within this framework often suffer from consistency issues in causal effect estimation and are difficult to extend to more complex application scenarios. To bridge this gap, we propose a new latent variable modeling approach, *Confounder Imputation with Stochastic Neural Networks* (CI-StoNet). The CI-StoNet utilizes a stochastic neural network to jointly model the outcome function and the missing confounders, and employs an adaptive stochastic gradient Hamiltonian Monte Carlo (SGHMC) algorithm to impute the missing confounders and train the neural networks simultaneously. Under mild conditions, we show that the causal effect remains identifiable through CI-StoNet, even though the missing confounders are non-identifiable – these confounders can only be identified up to an unknown loss-invariant transformation due to the non-identifiability inherent in neural network models. The CI-StoNet provides state-of-the-art performance on benchmarks for causal effect estimation and showcases its adaptability to proxy variable and multiple-cause scenarios. This new approach also serves as a versatile tool for modeling various causal relationships, leveraging the flexibility of stochastic neural networks in natural process modeling.

1 Introduction

Causal inference from observational studies is a topic of significant interest in fields such as genetics, economics, and social science. Under the potential outcome framework (Rubin, 1974), a fundamental condition for identifying causal effects is the strong ignorability condition (Rosenbaum & Rubin, 1983):

$$\boldsymbol{A} \perp\!\!\!\perp \{\boldsymbol{Y}(\boldsymbol{a}) : \boldsymbol{a} \in \mathcal{A}\} \mid \boldsymbol{Z}, \tag{1}$$

where $\boldsymbol{A}$ denotes the treatment variable taking values in the space $\mathcal{A}$, $\boldsymbol{Y}(\cdot)$ denotes the outcome function, and $\boldsymbol{Z}$ denotes confounders. A confounder refers to a variable that influences both the treatment and the outcome. For the strong ignorability condition to hold, all confounders must be observed. However, this requirement is rarely satisfied in observational data, leading to potentially severe bias in causal effect estimation.

One strategy to address the issue of missing confounders is to model them as latent variables. Wang & Blei (2018) proposed using a latent factor model to obtain a latent representation for multiple causes, enabling the capture of multiple-cause confounders under the assumption that no single-cause confounder exists. Kallus et al. (2018) tackled this problem under a proxy variable setting by leveraging the low-rank components of the proxy variables, obtained through matrix factorization, as an approximation to the true confounders. Louizos et al. (2017) also addressed the issue with proxy variables and introduced the causal effect variational autoencoder (CEVAE) to infer missing confounders from the observational distribution of the proxy, treatment, and outcome. These works represent significant advancements in causal inference using observational data; however, they have notable limitations. For instance, Imai & Jiang (2019) pointed out that Wang & Blei (2018) essentially models the substitute confounder as a deterministic function of treatments, leading it to converge to a function

of the observed treatments rather than the true confounder. Kallus et al. (2018) focuses primarily on the linear regression setting, limiting its applicability to nonlinear models unless many proxies are available for a small number of latent variables. Rissanen & Marttinen (2021) examined the consistency of the causal effect estimator in Louizos et al. (2017) and showed that it fails to correctly estimate cause effects when the latent variable is misspecified or the data distribution is overly complex.

In this paper, we propose a latent variable imputation approach to address the issue of missing confounders. This new approach is built on the stochastic neural network (StoNet) (Sun & Liang, 2022; Liang et al., 2022) and sparse deep learning theory (Sun et al., 2022), effectively overcoming the limitations of existing approaches. The core idea involves modeling the causal directed acyclic graph (causal DAG), which forms the foundation for causal inference, using a StoNet and impute the missing confounders according to the conditional distribution formed by the StoNet. This makes causal effect identifiable by leveraging StoNet's universal approximation capability, its inherent Markovian structure, and its parameter estimation consistency in a sparse learning mode. The StoNet is trained using an adaptive stochastic gradient MCMC algorithm (Liang et al., 2022; Deng et al., 2019), which allows for the simultaneous imputation of missing confounders and estimation of sparse StoNet parameters. We refer to the proposed approach as *Confounder Imputation with Stochastic Neural Networks* (CI-StoNet). In summary, it offers the following advantages in addressing the missing confounder issue:

(i) **Accurate Causal Effect Estimation:** This property is supported by StoNet's inherent ability to handle missing data, the consistency of sparse deep learning, and the convergence guarantee offered by the adaptive stochastic gradient MCMC algorithm. Under mild conditions, we show that the causal effect remains identifiable through CI-StoNet, even though the missing confounders can only be identified up to some unknown loss-invariant transformations (due to non-identifiability of neural network models).

(ii) **Complex nonlinear modeling.** CI-StoNet inherits the universal-approximation property of deep neural networks (DNNs), enabling effective modeling of complex nonlinear relationships across diverse applications.

(iii) **Structural flexibility.** The Markovian architecture of CI-StoNet provides structural flexibility for representing diverse dependency patterns in causal DAGs. It supports localized updates to each DNN module, promoting modular design and easy adaptation to varying causal relationships.

2 CI-StoNet for Missing Confounders

2.1 The CI-StoNet Approach

This section introduces the CI-StoNet approach or, more generally, a deep learning framework for performing causal inference in presence of missing confounders.

Consider the scenario of simple confounding, as depicted by Figure 1, which involves treatment $\boldsymbol{A} \in \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m\}$, missing/latent confounders $\boldsymbol{Z}$, and an outcome $\boldsymbol{Y}$. The corresponding model is given by

$$\begin{aligned} \boldsymbol{A} &= g_1(\boldsymbol{Z}, \boldsymbol{e}_a), \\ \boldsymbol{Y} &= g_2(\boldsymbol{Z}, \boldsymbol{A}) + \boldsymbol{e}_y, \end{aligned} \tag{2}$$

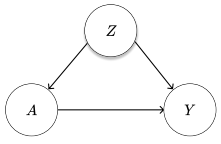

Figure 1: simple confounding

where $g_1(\cdot)$ and $g_2(\cdot)$ are unknown functions that can be nonlinear and highly complex, and $\boldsymbol{e}_a$ and $\boldsymbol{e}_y$ are random errors. In this paper, we assume $\boldsymbol{e}_y \sim N(0, \sigma_y^2 I_{d_y})$, where d_y denotes the dimension of $\boldsymbol{Y}$. There is flexibility in specifying the distribution of $\boldsymbol{e}_a$. If each component of $\boldsymbol{A}$ takes values in the binary space $\{0, 1\}$, then $\boldsymbol{e}_a$ follows a Logistic distribution. If $\boldsymbol{A}$ is continuous, then $\boldsymbol{e}_a$ can be assumed to follow a Gaussian distribution or any other continuous distribution. Furthermore, if $\boldsymbol{A}$ is mixed, the distribution of each component of $\boldsymbol{e}_a$ can be specified accordingly. This scenario has included multiple causes considered in Wang & Blei (2018), where $\boldsymbol{a}_i$ is a multi-dimensional vector,

as a special case. Since $\boldsymbol{Z}$ is missing, we impute it from the conditional distribution:

$$\pi(\boldsymbol{Z}|\boldsymbol{A},\boldsymbol{Y}) \propto \pi(\boldsymbol{Z})\pi(\boldsymbol{A}|\boldsymbol{Z})\pi(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{A}) \propto \pi(\boldsymbol{Z}|\boldsymbol{A})\pi(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{A}), \tag{3}$$

where $\pi(\cdot)$ denotes a distribution or conditional distribution in the appropriate context.

The latter part of Eq. (3) suggests that, mathematically, $\boldsymbol{A}$, $\boldsymbol{Z}$, and $\boldsymbol{Y}$ can be interpreted as the exogenous input, latent state and output of a stochastic model. Motivated by this view, we propose to perform the imputation using a CI-StoNet (see Figure 2 for its structure), formulated as:

$$\begin{aligned} \boldsymbol{Z} &= \mu_1(\boldsymbol{A},\boldsymbol{\theta}_1) + \boldsymbol{e}_z, \\ \boldsymbol{Y} &= \mu_2(\boldsymbol{Z},\boldsymbol{A},\boldsymbol{\theta}_2) + \boldsymbol{e}_y, \end{aligned} \tag{4}$$

where $\mu_1(\cdot)$ and $\mu_2(\cdot)$ are two neural network functions, parameterized by $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively; $\boldsymbol{e}_z \sim N(\boldsymbol{0}, \sigma_z^2 I_{d_z})$; and $\boldsymbol{e}_y$ is as defined in (2). The two neural networks are interconnected through the latent variable $\boldsymbol{Z}$. Additionally, we impose the following assumptions on the models (2) and (4):

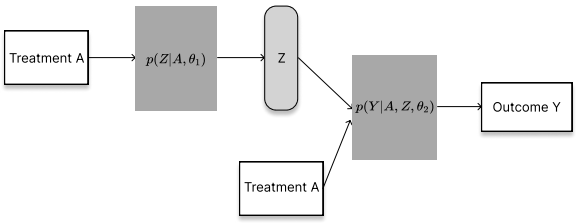

Figure 2: Diagram of CI-StoNet under simple confounding, where white rectangles represent variables from observed data; light-grey rounded-rectangles represent latent variable to impute; and dark-grey rectangles represent neural network modules to learn respective conditional distributions.

Assumption 1. *(i)* $\boldsymbol{e}_a \perp\!\!\!\perp \boldsymbol{e}_y$, $\boldsymbol{e}_a \perp\!\!\!\perp \boldsymbol{Z}$, $\boldsymbol{e}_y \perp\!\!\!\perp (\boldsymbol{Z},\boldsymbol{A})$*; (ii) there exist sparse DNNs* $\mu_1(\cdot)$ *and* $\mu_2(\cdot)$ *such that (4) holds,* $\boldsymbol{e}_z \sim N(\boldsymbol{0}, \sigma_z^2 I_{d_z})$, $\boldsymbol{e}_y \sim N(0, \sigma_y^2 I_{d_y})$, $\boldsymbol{e}_z \perp\!\!\!\perp \boldsymbol{e}_y$, *and* $\boldsymbol{e}_z \perp\!\!\!\perp \boldsymbol{A}$*;*

Part (i) of Assumption 1 ensures that strong ignorability (1) holds. Part (ii) assumes that the true model is a sparse StoNet (as $\boldsymbol{Z}$ is random), which greatly facilitates the subsequent theoretical studies. Otherwise, the DNN approximation errors in relation to model (2) would need to be considered in the subsequent analysis. Refer to Remark 2 (in Section 2.2) regarding the expressivity of sparse DNNs.

Notably, the functional expression in (4) does not imply a causal mechanism $\boldsymbol{A} \to \boldsymbol{Z}$. For example, rain ($\boldsymbol{Z}$) causes a wetland ($\boldsymbol{A}$), but a wetland does not cause rain. Similarly, $\boldsymbol{Z}$ cannot be interpreted as a mediator due to the nonexistence of a causal mechanism $\boldsymbol{A} \to \boldsymbol{Z}$, although (4) has a mathematical structure similar to mediation models. For the time being, we assume that there is no mediator in the causal pathway between the treatment $\boldsymbol{A}$ and the outcome $\boldsymbol{Y}$, thereby ruling out any potential misinterpretation for the role of $\boldsymbol{Z}$. However, if a mediator does exist, issues related to the total causal effect estimation and the interpretation of $\boldsymbol{Z}$ will be addressed at the end of Section 2.2.

Under the missing data framework, the CI-StoNet can be trained by solving the following equation, which represents a Bayesian version of Fisher's identity (Song et al., 2020):

$$\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}|\boldsymbol{A},\boldsymbol{Y}) = \int \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}|\boldsymbol{Z},\boldsymbol{A},\boldsymbol{Y}) \pi(\boldsymbol{Z}|\boldsymbol{A},\boldsymbol{Y},\boldsymbol{\theta}) d\boldsymbol{Z}, \tag{5}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$, $\boldsymbol{Z}$ is missing, $\pi(\boldsymbol{\theta}|\boldsymbol{Z},\boldsymbol{A},\boldsymbol{Y}) \propto \pi(\boldsymbol{\theta}_1)\pi(\boldsymbol{\theta}_2)\pi(\boldsymbol{Z}|\boldsymbol{A},\boldsymbol{\theta}_1)\pi(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{A},\boldsymbol{\theta}_2)$, $\pi(\boldsymbol{Z}|\boldsymbol{A},\boldsymbol{Y},\boldsymbol{\theta}) \propto \pi(\boldsymbol{Z}|\boldsymbol{A},\boldsymbol{\theta}_1)\pi(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{A},\boldsymbol{\theta}_2)$, and $\pi(\boldsymbol{\theta}_1)$ and $\pi(\boldsymbol{\theta}_2)$ denote the prior distributions imposed on $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively. In this paper, we assume that the components of $\boldsymbol{\theta}$ are *a priori* independent and are subject to the following mixture Gaussian prior (Sun et al., 2022):

$$\pi(\boldsymbol{\theta}) = \prod_{i=1}^{K_n} (1-\lambda_n)\phi(\theta_i/\sigma_0) + \lambda_n \phi(\theta_i/\sigma_1), \tag{6}$$

where λ_n is the mixture proportion, K_n is the total number of connections in the StoNet (i.e., the dimension of $\boldsymbol{\theta}$), $\phi(\cdot)$ represents the density function of the standard normal distribution, and σ_0 and σ_1 are the standard deviations of the two Gaussian components, respectively.

The identity (5) further suggests that the target equation

$$\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}|\boldsymbol{A},\boldsymbol{Y}) = 0, \tag{7}$$

can be solved using an adaptive stochastic gradient MCMC algorithm, which iteratively alternates between latent variable imputation and parameter updates. In this paper, we employ the adaptive stochastic gradient Hamiltonian Monte Carlo (SGHMC) (Liang et al., 2022), as given in Algorithm 1, to solve equation (7).

Algorithm 1: Adaptive SGHMC

0. Set the prior hyperparameters: λ_n, σ_0, and σ_1.
1. (*Latent variable imputation*) Simulate $\boldsymbol{Z}$ from $\pi(\boldsymbol{Z}|\boldsymbol{A},\boldsymbol{Y},\boldsymbol{\theta})$ via Hamiltonian Monte Carlo updates:

$$\begin{aligned}\mathbf{v}^{(k+1)} &= (1-\epsilon_{k+1}\eta)\mathbf{v}^{(k)} + \epsilon_{k+1}\nabla_{\boldsymbol{Z}}\log\pi(\boldsymbol{Z}^{(k)}|\boldsymbol{A},\boldsymbol{\theta}_1^{(k)}) + \epsilon_{k+1}\nabla_{\boldsymbol{Z}}\log\pi(\boldsymbol{Y}|\boldsymbol{Z}^{(k)},\boldsymbol{A},\boldsymbol{\theta}_2^{(k)})) \\ &\quad + \sqrt{2\epsilon_{k+1}\eta}\mathbf{e}^{(k+1)}, \\ \mathbf{Z}^{(k+1)} &= \mathbf{Z}^{(k)} + \epsilon_t\mathbf{v}^{(k)},\end{aligned}$$

   where $\boldsymbol{e}^{(k+1)} \sim N(0, I_{d_z})$, d_z is the dimension of $\boldsymbol{Z}$, and ϵ_{k+1} is the learning rate.
2. (*Parameter update*) Given $\boldsymbol{Z}^{(k+1)}$, update $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ separately:

$$\begin{aligned}\boldsymbol{\theta}_1^{(k+1)} &= \boldsymbol{\theta}_1^{(k)} + \gamma_{k+1}\nabla_{\boldsymbol{\theta}_1}\log\pi(\boldsymbol{Z}^{(k+1)}|\boldsymbol{A},\boldsymbol{\theta}_1^{(k)}) + \gamma_{k+1}\nabla_{\boldsymbol{\theta}_1}\log\pi(\boldsymbol{\theta}_1^{(k)}), \\ \boldsymbol{\theta}_2^{(k+1)} &= \boldsymbol{\theta}_2^{(k)} + \gamma_{k+1}\nabla_{\boldsymbol{\theta}_2}\log\pi(\boldsymbol{Y}|\boldsymbol{Z}^{(k+1)},\boldsymbol{A},\boldsymbol{\theta}_2^{(k)}) + \gamma_{k+1}\nabla_{\boldsymbol{\theta}_2}\log\pi(\boldsymbol{\theta}_2^{(k)}).\end{aligned}$$

In model (4), both σ_z and σ_y are scalar. They can be treated as hyperparameters to specify in simulations, while having minimal impact on the downstream inference. Notably, σ_z is essentially non-identifiable in model (4), due to the universal approximation property of neural networks. In the inference stage, see equation (9), we provide a Bayesian estimator for σ_z^2 to facilitate imputation of missing confounders. Specifically, we impose an inverse gamma prior $\sigma_z^2 \sim \text{InvGamma}(\alpha,\beta)$, leading to the Bayesian estimator:

$$\hat{\sigma}_z^2 = \frac{\beta + \frac{1}{2}\sum_{j=1}^{n}(z_j - \mu_1(\boldsymbol{A}_j,\boldsymbol{\theta}_1))^2}{\frac{n}{2}+\alpha-1}, \tag{8}$$

where we set $\alpha = \beta = 1$ for a flat prior, and $\boldsymbol{A}_j$ denotes the value of $\boldsymbol{A}$ in sample j. In simulations, its value can also be updated as in (8) along with iterations, while having minimal impact on the performance of the algorithm.

To enable causal inference, we introduce the following additional assumptions, which are standard conditions for causal effect identification:

Assumption 2. *1.* **Stable unit treatment value assumption (SUTVA)***: the potential outcome of one subject are independent of the assigned treatment of another subject; that is, there is no interference between subjects and there is only a single version of each assigned treatment.*

2. **Overlap***: The substitute confounder $\boldsymbol{Z}$ satisfies the overlap condition: $p(A \in \mathcal{A}|\boldsymbol{Z}) > 0$ for all sets $\mathcal{A}$ with positive measure, i.e., $p(\mathcal{A}) > 0$.*

Under Assumptions 1 and 2, the causal effect can be estimated in the following procedure:

Causal effect estimation. After Algorithm 1 converges, with learned parameters $\hat{\boldsymbol{\theta}}_1^*$ and $\hat{\boldsymbol{\theta}}_2^*$, draw $\mathcal{M}$ samples $\{\boldsymbol{z}^{(l)}\}_{l=1}^{\mathcal{M}}$ from $\pi(\boldsymbol{z} \mid \boldsymbol{a};\hat{\boldsymbol{\theta}}_1^*)$. The expected outcome $\mathbb{E}\{Y(\boldsymbol{a}) \mid \boldsymbol{\theta}^*\} = \int \mu_2(\boldsymbol{z},\boldsymbol{a};\boldsymbol{\theta}_2^*)\,\pi(\boldsymbol{z} \mid \boldsymbol{a};\boldsymbol{\theta}_1^*)\,d\boldsymbol{z}$ is then approximated by the Monte Carlo average

$$\widehat{\mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}^*)} = \frac{1}{\mathcal{M}}\sum_{l=1}^{\mathcal{M}}\mu_2(\boldsymbol{z}^{(l)},\boldsymbol{a},\hat{\boldsymbol{\theta}}_2^*). \tag{9}$$

The direct causal effect for a binary treatment can be estimated by

$$\hat{\tau} := \widehat{\mathbb{E}(Y(\boldsymbol{a}_1)|\hat{\boldsymbol{\theta}}^*)} - \widehat{\mathbb{E}(Y(\boldsymbol{a}_0)|\hat{\boldsymbol{\theta}}^*)}, \tag{10}$$

where $\boldsymbol{a}_0$ and $\boldsymbol{a}_1$ denote the control and treatment, respectively. In the case of multiple causes (Wang & Blei, 2018), where each $\boldsymbol{a} = (a_1, \ldots, a_m)^T$ is a continuous multi-dimensional vector, one might be interested in estimating the causal effect of each individual cause. In this case, the marginal causal effect of a_j can be estimated as:

$$\hat{\tau}_{a_j} = \frac{1}{\mathcal{M}} \sum_{l=1}^{\mathcal{M}} \frac{\partial \mu_2(\boldsymbol{z}^{(l)}, \boldsymbol{a}, \hat{\boldsymbol{\theta}}_2^*)}{\partial a_j}, \tag{11}$$

analogous to the estimator given in (9).

2.2 Theoretical Guarantees

The consistency of the estimator (9) can be established through several steps, with all proofs deferred to Appendix A3. First, we show that the estimator $\boldsymbol{\theta}^{(k)} = \{\boldsymbol{\theta}_1^{(k)}, \boldsymbol{\theta}_2^{(k)}\}$ obtained from Algorithm 1 converges in probability to a solution of (7), denoted by $\hat{\boldsymbol{\theta}}_n^*$ (see Lemma 1) A discussion on how to address the non-uniqueness of $\hat{\boldsymbol{\theta}}_n^*$ is followed. Next, we show that $\hat{\boldsymbol{\theta}}_n^*$ is a consistent estimator of $\boldsymbol{\theta}^*$, the true parameter vector of the sparse StoNet defined in (4) (see Theorem 1). Building on this result, we establish the consistency of the estimator (9) (see Theorem 2). These results are presented in the following.

Lemma 1. *(Theorem S1, Liang et al. (2022)) Suppose Assumptions A3-A5 (given in Supplement A3) hold. For Algorithm 1, if we set $\epsilon_k = C_\epsilon/(c_e + k^\alpha)$ and $\gamma_k = C_\gamma/(c_g + k^\alpha)$ for some constants $\alpha \in (0, 1)$, $C_\epsilon > 0$, $C_\gamma > 0$, $c_e \geq 0$ and $c_g \geq 0$, then there exists an iteration k_0 and a constant $\Lambda_0 > 0$ such that for any $k > k_0$,*

$$\mathbb{E}(\|\boldsymbol{\theta}^{(k)} - \hat{\boldsymbol{\theta}}_n^*\|^2) \leq \Lambda_0 \gamma_k, \tag{12}$$

where $\hat{\boldsymbol{\theta}}_n^*$ denotes a solution to Eq. (7), i.e., $\hat{\boldsymbol{\theta}}_n^* \in \mathcal{L} = \{\boldsymbol{\theta} : \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}|\boldsymbol{A}, \boldsymbol{Y}) = 0\}$; and $\Lambda_0 = \Lambda_0' + 6\sqrt{6} C_{\boldsymbol{\theta}}^{1/2}((3M^2 + \varsigma_2)C_{\boldsymbol{Z}} + 3M^2 C_{\boldsymbol{\theta}} + 3B^2 + \varsigma_2^2)^{1/2}$ for some positive constants Λ_0', $C_{\boldsymbol{\theta}}$, and $C_{\boldsymbol{Z}}$.

Refer to Lemma S1 of Liang et al. (2022) for the derivation of the constants $C_{\boldsymbol{\theta}}$ and $C_{\boldsymbol{Z}}$, which indicate the dependence of the convergence of $\boldsymbol{\theta}^{(k)}$ on the structure of the StoNet (4). As a consequence of the l_2-convergence (12), we immediately have $\|\boldsymbol{\theta}^{(k)} - \hat{\boldsymbol{\theta}}_n^*\| \xrightarrow{p} 0$ as $k \to \infty$, where $\xrightarrow{p}$ denotes convergence in probability.

Remark 1. *For neural networks, it is known that their loss function is invariant under certain transformations of the connection weights, such as reordering hidden neurons within a layer or jointly changing the signs or scales of specific weights and biases, refer to, e.g., Liang et al. (2018b) and Sun et al. (2022) for detailed discussions. As a result, the solution $\hat{\boldsymbol{\theta}}_n^*$ is not unique, and all such solutions can be viewed as belonging to an equivalence class of unique solutions, defined by loss-invariant transformations. This equivalence class forms a reduced representation of the parameter space, where each member corresponds to a distinct network (i.e., not transformable into another via loss-invariant operations) and may have a different loss value. The consistency results established in this paper apply specifically to this reduced space of neural networks.*

Theorem 1. *Suppose the regularity conditions give in Lemma A1 and Assumptions A6-A7 (given in Supplement A3) hold. Additionally, assume that the dimension of $\boldsymbol{\theta}$, denoted by K_n, increases with n in a polynomial rate $K_n = O(n^\zeta)$ for some constant $\zeta > 1$, while the true StoNet is sparse with the number of nonzero connections $m_n \prec \frac{n}{c \log(K_n/n)}$ for some constant $c > 1$. Set the hyper-parameters of the prior (6) to satisfy the conditions:*

$$(\frac{n}{K_n})^c \prec \lambda_n \prec \frac{n}{K_n}, \quad \sigma_1 = O(1), \quad (\frac{n}{K_n})^c \prec \sigma_0 \prec \min\left\{1 - \frac{n}{K_n}, \frac{\delta_n}{\sqrt{c\log(K_n) - (c-1)\log(n)}}\right\}. \tag{13}$$

Then $\|\hat{\boldsymbol{\theta}}_n^* - \boldsymbol{\theta}^*\| \xrightarrow{p} 0$ holds as $n \to \infty$, where $\boldsymbol{\theta}^*$ denotes the true parameter of the StoNet (4), and $\hat{\boldsymbol{\theta}}_n^*$ is up to a loss-invariant transformation.

Remark 2. *In Theorem 1, we assume that the true sparse StoNet is of size $m_n = o(n)$. This assumption can be justified based on the theory established in Bölcskei et al. (2019), Schmidt-Hieber (2017), and Petersen & Voigtlaender (2018), where it is shown that a DNN of this size has been large enough to approximate many classes of functions, including affine, piecewise smooth, and α-Hölder smooth functions. See Sun et al. (2022) for discussions on this issue. Additionally, Sun et al. (2022) showed that a sparse neural network of this size has been large enough to achieve the desired function approximation and posterior consistency, with the mixture Gaussian prior (6), as the sample size n becomes large. Our theory allows K_n to increase polynomially with n, which is typically satisfied by deep neural networks.*

Theorem 2. *Suppose Assumptions 1-2 and the conditions in Lemma 1 and Theorem 1 hold. Then*

$$\|\widehat{\mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}_n^*)} - \mathbb{E}(Y(\boldsymbol{a})|\boldsymbol{\theta}^*)\| \xrightarrow{p} 0, \quad \text{as } \mathcal{M} \to \infty \text{ and } n \to \infty.$$

Remark 3. *As shown in the proof of Theorem 2, the consistency of the estimator (9) arises from the existence of the true sparse StoNet as well as the consistency of $\hat{\boldsymbol{\theta}}_n^*$. It is important to note that, due to the non-uniqueness of $\hat{\boldsymbol{\theta}}_n^*$ as discussed in Remark 1, the imputed latent confounders may differ from their true values. However, $\pi(\boldsymbol{z}|\boldsymbol{A}, \hat{\boldsymbol{\theta}}_1^*)$ still serves as a consistent estimator (in terms of the density function) of $\pi(\boldsymbol{z}|\boldsymbol{A}, \boldsymbol{\theta}_1^*)$, up to a loss-invariant transformation of $\hat{\boldsymbol{\theta}}_n^*$. Nevertheless, this does not affect the consistency of the estimator (9), which is a remarkable property.*

Notably, due to the universal approximation power of DNNs, CI-StoNet can capture all confounders, including both multiple-cause and single-cause confounders. This brings less restrictions on the confounding structure compared to the models considered in Wang & Blei (2018). CI-StoNet also differs from the variational autoencoder approach proposed in Louizos et al. (2017). When large neural networks are used to fit the functions $\mu_1(\cdot)$ and $\mu_2(\cdot)$, the extracted latent variable may fail to capture the information encoded in the observed data. CI-StoNet addresses this issue by leveraging its parameter estimation consistency, which is achieved through sparse deep learning under a Bayesian setting.

Additionally, we note that the latent variable imputed in step (i) of Algorithm 1 cannot be used for causal effect estimation, as it may contain information related to colliders. Figure 3(a) illustrates this concept, where the collider variable $\boldsymbol{C}$ is influenced by both $\boldsymbol{A}$ and $\boldsymbol{Y}$. In step (i), we impute $\boldsymbol{Z}$ conditioned on both $\boldsymbol{A}$ and $\boldsymbol{Y}$. If a collider variable exists, the imputed latent variable may introduce spurious associations between $\boldsymbol{A}$ and $\boldsymbol{Y}$, potentially biasing the causal effect estimation. To mitigate this issue, we specifically impute the latent variables from $\pi(\boldsymbol{Z}|\boldsymbol{A}, \hat{\boldsymbol{\theta}}_1^*)$, ensuring that any collider-related information is excluded from the analysis.

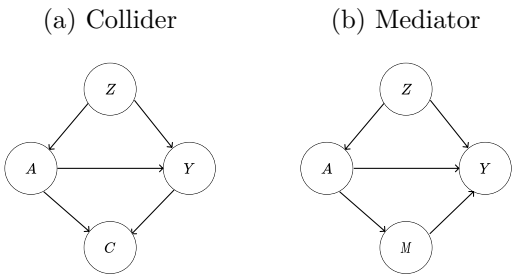

Figure 3: Other examples of causal structures: (a) existence of colliders, represented by C ; (b) existence of mediators, represented by M.

In Section 2.1, we assumed the absence of mediators to enable a clear interpretation of $\boldsymbol{Z}$ as a latent confounder. However, if a mediator $\boldsymbol{M}$ does exist, as illustrated in Figure 3(b), the imputed latent variable $\boldsymbol{Z}$ may inadvertently encapsulate information related to $\boldsymbol{M}$. Mathematically, the conditional distribution can be expressed as:

$$\pi(\boldsymbol{Y} \mid \boldsymbol{A}, \boldsymbol{Z}) = \int \pi(\boldsymbol{Y} \mid \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{M})\, \pi(\boldsymbol{M} \mid \boldsymbol{A})\, d\boldsymbol{M},$$

which indicates that, without observing $\boldsymbol{M}$, its effect will be absorbed into $\boldsymbol{Z}$, making them statistically indistinguishable within the CI-StoNet framework. In this case, (10) serves as an estimator for the *total causal effect* of $\boldsymbol{A}$ on $\boldsymbol{Y}$, while $\boldsymbol{Z}$ acts as a latent adjustment variable that facilitates estimation of the *total causal effect*. Although this precludes pathway-specific interpretations, it does not invalidate estimation of the total causal effect. If, however,

the mediator $\boldsymbol{M}$ is known from domain knowledge or experimental design and there is no unmeasured confounding between $\boldsymbol{A}$ and $\boldsymbol{M}$, or between $\boldsymbol{M}$ and $\boldsymbol{Y}$, then the front-door criterion (Pearl, 2009) can be applied. In this case, $\boldsymbol{M}$ can be included as part of the latent confounder layer in the CI-StoNet to enable identification of the direct causal effect via front-door adjustment.

2.3 A Simulation Study

As a concept-proof example, we evaluated CI-StoNet using a simulation study with a nonlinear data-generating process for $\boldsymbol{A}$ and $\boldsymbol{Y}$ under both separable and non-separable confounding scenarios. We generate the latent confounders $Z_1, \ldots, Z_6$ as independent standard Gaussian random variables, and then draw $A_1, \ldots, A_9$ independently from the distribution, using inverse CDF:

$$p(a_i|\boldsymbol{Z}) = \frac{\text{expit}(\xi(\boldsymbol{Z})a_i)}{\int_{-1}^{1} \text{expit}(\xi(\boldsymbol{Z})a_i)} 1_{\{-1 \le a_i \le 1\}}, \quad i = 1, \ldots, 9,$$

where $\text{expit}(\xi(\boldsymbol{Z})a_i) = \frac{\exp\{\xi(\boldsymbol{Z})a_i\}}{1+\exp\{\xi(\boldsymbol{Z})a_i\}}$, and $\xi(\boldsymbol{Z}) = \sum_{i=1}^{2} \beta_i \sin z_i + \sum_{j=3}^{4} \beta_j \cos z_j + \sum_{k=5}^{6} \frac{1}{1+\exp\{-\beta_k z_k + 0.5\}}$. We set $f_1(\boldsymbol{A}) = \boldsymbol{\theta}^T \boldsymbol{A}^{\otimes 2}$ and $f_2(\boldsymbol{A}) = \sum_{i<j} a_i a_j$, where $\boldsymbol{A}^{\otimes 2}$ represent an element-wise square operation, and generate Y in two settings: (i) *Separable confounding.* the treatment and confounder impact the outcome separately: $Y = f_1(\boldsymbol{A}) - \theta_0 f_2(\boldsymbol{A}) + \xi(\boldsymbol{Z}) + \epsilon$, where $\theta_0 \sim U(-1, 1)$ and $\epsilon \sim N(0, 1)$. (ii) *Non-separable confounding.* there exists interaction between the treatment and confounder: $Y = f_1(\boldsymbol{A}) - \xi(\boldsymbol{Z}) f_2(\boldsymbol{A}) + \xi(\boldsymbol{Z}) + \epsilon$, where $\epsilon \sim N(0, 1)$.

For each setting, the experiment was conducted on 10 simulated datasets, each comprising 1000 training samples, 500 validation samples, and 500 test samples. The marginal treatment effects were calculated using the test set. Figure S1 compares the true and estimated marginal treatment effects across the 10 datasets. The plots show that most of the estimated marginal effects lie within half a standard deviation of the true marginal effects, indicating that CI-StoNet is able to estimate the marginal effect of each treatment with small bias.

3 Causal StoNet for Proxy Variables

For some problems, it is possible to obtain proxies for a missing confounder, which may be noisy or provide only partial measurements of the missing confounder. Conditioning on these proxy variables helps control, though not fully eliminate, the confounding bias. It is natural to incorporate the proxy variable into the model as a substitute for the missing confounder. Kuroki & Pearl (2014) introduces conditions to use proxies effectively even when the exact distribution of the measurement errors is unknown. Specifically, it proposes to use matrix adjustments to estimate causal effects when external information is available, while eigen-decomposition methods can be used to identify causal effects under specific assumptions about the proxies when external information is absent. Tchetgen et al. (2020) and Miao et al. (2018) proposed the proximal causal inference framework. They demonstrated that, under mild conditions, the identification of causal effects with missing confounders is possible, provided that two types of proxy variables can be measured: one serving as a treatment confounding proxy, and the other as an outcome confounding proxy. Louizos et al. (2017) proposed an algorithm based on a variational autoencoder to model causal relationships using a single proxy.

Consider the causal structure with a single proxy, as depicted in Figure 4(a). This causal structure suggests that $\boldsymbol{Z}$ can be imputed based on the following conditional distribution:

$$\pi(\boldsymbol{Z}|\boldsymbol{A}, \boldsymbol{Y}, \boldsymbol{X}) \propto \pi(\boldsymbol{Z})\pi(\boldsymbol{X}|\boldsymbol{Z})\pi(\boldsymbol{A}|\boldsymbol{Z})\pi(\boldsymbol{Y}|\boldsymbol{Z}, \boldsymbol{A}) \propto \pi(\boldsymbol{Z}|\boldsymbol{X})\pi(\boldsymbol{A}|\boldsymbol{Z})\pi(\boldsymbol{Y}|\boldsymbol{Z}, \boldsymbol{A}). \tag{14}$$

The decomposition of the conditional distribution (14) further suggests the StoNet structure:

$$\boldsymbol{Z} = \mu_1(\boldsymbol{X}, \boldsymbol{\theta}_1) + \boldsymbol{e}_z, \quad \boldsymbol{A} = \mu_2(\boldsymbol{Z}, \boldsymbol{\theta}_2, \boldsymbol{e}_a), \quad \boldsymbol{Y} = \mu_3(\boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{\theta}_3) + \boldsymbol{e}_y, \tag{15}$$

where $\boldsymbol{e}_z$ and $\boldsymbol{e}_y$ are Gaussian random errors, while the form of $\boldsymbol{e}_a$ can be determined according to the types of treatments. These random errors are mutually independent and

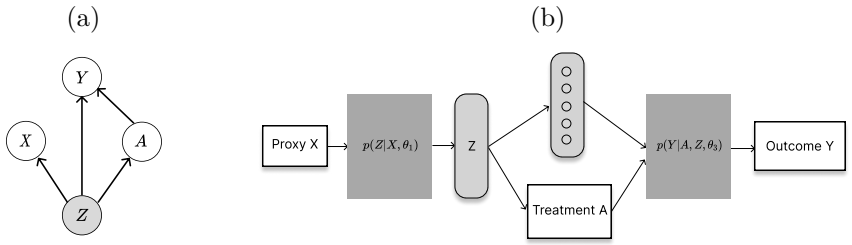

Figure 4: (a) Causal DAG: without dependence on the proxy; (b) Diagram of CI-StoNet under the proxy setting: white rectangles represent variables from observed data; light-grey rounded-rectangles represent hidden neurons; dark-grey rectangles represent network modules to learn respective conditional distributions.

are also independent of $\boldsymbol{X}$. Figure 4(b) illustrates the corresponding CI-StoNet structure. Alternatively, we can consider the following StoNet model:

$$\boldsymbol{Z} = \mu_1(\boldsymbol{X}, \boldsymbol{\theta}_1) + \boldsymbol{e}_z, \quad \boldsymbol{A} = \mu_2(\boldsymbol{Z}, \boldsymbol{\theta}_2) + \boldsymbol{e}_a, \quad \boldsymbol{Y} = \mu_3(\boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{\theta}_3) + \boldsymbol{e}_y, \tag{16}$$

where $\boldsymbol{e}_z$, $\boldsymbol{e}_a$ and $\boldsymbol{e}_y$ are Gaussian random errors. Notably, the model (15) and the model (16) are asymptotically equivalent, even when $\boldsymbol{A}$ is a binary vector. In the binary case, their equivalence is supported by the result that, as shown in Liang (2003) and Duda et al. (2001), $\mu_2(\boldsymbol{Z}, \boldsymbol{\theta}_2)$ converges to the probability function $P(\boldsymbol{A} = \boldsymbol{1}|\boldsymbol{Z}, \boldsymbol{\theta}_2)$ as $n \to \infty$.

In this paper, we adopt the model (16) for computational simplicity. A gradient equation analogous to (7) can be constructed for the model. An adaptive SGHMC algorithm, similar to Algorithm 1, can be employed for its solution. Let $\{\boldsymbol{z}^{(l)} : l = 1, 2, \ldots, \mathcal{M}\}$ denote the samples simulated from $\pi(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{a}, \hat{\boldsymbol{\theta}}_1^*)$. Then the expected outcome function $\mathbb{E}(Y(\boldsymbol{a})|\boldsymbol{x})$ can be estimated by the Monte Carlo average as

$$\widehat{\mathbb{E}(Y(\boldsymbol{a})|\boldsymbol{x}, \hat{\boldsymbol{\theta}}_3^*)} = \frac{1}{\mathcal{M}} \sum_{l=1}^{\mathcal{M}} \mu_3(\boldsymbol{z}^{(l)}, \boldsymbol{a}, \hat{\boldsymbol{\theta}}_3^*). \tag{17}$$

In the case that the treatment $\boldsymbol{a}$ is continuous and multi-dimensional, the marginal causal effect can be estimated as in (11).

3.1 Numerical Experiments

For simplicity, we consider a single binary treatment in our experiments. CI-StoNet is compared with the following baselines:

(i) Designed for average treatment effect (ATE): double selection estimator (**DSE**)(Belloni et al., 2014), approximate residual balancing estimator (**ARBE**) (Athey et al., 2018), targeted maximum likelihood estimator (**TMLE**) (van der Laan & Rubin, 2006), and deep orthogonal networks for unconfounded treatments (**DONUT**) (Hatt & Feuerriegel, 2021).

(ii) Designed for heterogeneous treatment effect: **X-learner** (Künzel et al., 2017), **Dragonnet**(Shi et al., 2019), causal multi-task deep ensemble (**CMDE**) (Jiang et al., 2023)), causal multi-task gaussian processes (**CMGP** (Alaa & van der Schaar, 2017)), causal effect variational autoencoder (**CEVAE**) (Louizos et al., 2017), generative adversarial networks (**GANITE**) (Yoon et al., 2018), and counterfactual regression net (**CFRNet** (Shalit et al., 2017)). For the baselines in part (ii), we use the code of Jiang et al. (2023) at GitHub.

For performance evaluation, we consider two metrics: (i) estimation accuracy of ATE, which is measured by the mean absolute error (MAE) of the ATE estimates; and (ii) estimation accuracy of CATE, which is measured by precision in estimation of heterogeneous effect (PEHE).

3.1.1 Simulated Examples

This example is designed to compare methods on problems with nonlinear treatment effect and nonlinear outcome function. We generated 10 datasets using the procedure as described in Section A1.2, with each dataset consisting of 2000 training samples, 500 validation samples, and 500 test samples. Table 1 shows that CI-StoNet provides accurate estimates for the heterogeneous treatment effect and outperforms the baselines.

Table 1: Comparison of different methods for estimation of heterogeneous treatment effects with proxy variables, where PEHE was computed over 10 datasets, 'In-sample PEHE' was computed with training and validation samples, and 'Out-of-sample PEHE' was computed with test samples.

	In-Sample PEHE	Out-of-Sample PEHE
CI-StoNet	**0.3614(0.0328)**	**0.3731(0.0350)**
CMDE	0.9019(0.0746)	0.9059 (0.0699)
CMGP	1.8823(0.0836)	2.2116 (0.1682)
CEVAE	0.6190(0.0350)	0.6246 (0.0384)
Ganite	1.2099(0.0558)	1.1797 (0.0499)
X-learner-RF	0.8308(0.0200)	1.4272 (0.0132)
X-learner-Bart	0.6489(0.0168)	0.6570 (0.0151)
CFRNet-Wass	1.7127(0.1668)	1.7258 (0.1667)
CFRNet-MMD	2.0238(0.0537)	2.0250 (0.0582)
DragonNet	0.4217(0.0356)	0.4305 (0.0361)

3.1.2 Benchmark Datasets

We evaluated CI-StoNet on some benchmark datasets, including the Twins dataset and 10 datasets from Atlantic Causal Inference Conference (ACIC) 2019 Data Challenge. The results reported in Section A1.3 indicate that CI-StoNet outperforms the baselines.

4 Conclusion

By integrating StoNets with adaptive stochastic gradient MCMC, this paper presents a practical, flexible, and theoretically rigorous framework for addressing the issue of missing confounders in causal inference from observational data. Specifically, the proposed CI-StoNet approach utilizes StoNet to model the dependence structure in the underlying causal DAG while estimating its parameters using adaptive stochastic gradient MCMC algorithms. The validity of this approach is supported by the convergence theory of adaptive stochastic gradient MCMC and the consistency theory of sparse StoNets, even though the missing confounders can only be identified up to an unknown loss-invariant transformation (due to the non-identifiability of neural network models). Furthermore, we have demonstrated that CI-StoNet can effectively handle causal inference problems that involve multiple causes or proxy variables, showcasing its broad applicability.

Despite its advantages, this study has some limitations. First, the structure and parameter estimation of CI-StoNet rely on the correct identification of the underlying causal DAG. For instance, in the case of multiple treatments, if an unknown mediator exists, CI-StoNet may inadvertently incorporate mediator information into the learned substitute confounder. This occurs because the model only considers the joint distribution of $(\boldsymbol{A}, \boldsymbol{Y}, \boldsymbol{Z})$ when defining the dependence structure in the causal DAG. Including post-treatment variables such as mediators can introduce bias into causal effect estimation. However, if a mediator is correctly identified in the causal DAG, the structure of CI-StoNet can be adjusted to accommodate its information appropriately. Another limitation is that the current version of CI-StoNet does not explicitly quantify the causal effect uncertainty. This limitation, however, can be addressed by extending existing methods for uncertainty quantification. For example, instead of using a standard DNN module, the original version of the StoNet (Liang et al., 2022) can be employed to model each conditional distribution for the treatment and confounder. In this setup, uncertainty quantification can be achieved based on the properties of StoNet, as detailed in Liang et al. (2022).

Finally, the Markovian structure of CI-StoNet affords substantial flexibility for modeling a wide range of causal structures. Section A2 extends CI-StoNet to two proxy-variable settings: (i) outcome depending on the proxy and (ii) treatment depending on the proxy. See that section for details.

References

Ahmed M. Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. *NIPS*, 2017. URL https://api.semanticscholar.org/CorpusID:15589829.

Susan Athey, Guido W. Imbens, and Stefan Wager. Approximate residual balancing: Debiased inference of average treatment effects in high dimensions. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 80(4):597–623, 2018.

Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. Inference on treatment effects after selection amongst high-dimensional controls. *Review of Economic Studies*, 81: 608–650, 2014.

Helmut Bölcskei, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. Optimal approximation with sparsely connected deep neural networks. *CoRR*, abs/1705.01714, 2019.

Wei Deng, Xiao Zhang, Faming Liang, and Guang Lin. An adaptive empirical bayesian method for sparse deep learning. *Advances in neural information processing systems*, 2019: 5563–5573, 2019.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification, 2nd Edition*. Wiley, New York, 2001. URL https://api.semanticscholar.org/CorpusID:361680.

Tobias Hatt and Stefan Feuerriegel. Estimating average treatment effects via orthogonal regularization. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 680–689, 2021.

Kosuke Imai and Zhichao Jiang. Comment: The challenges of multiple causes. *Journal of the American Statistical Association*, 2019.

Ziyang Jiang, Zhuoran Hou, Yiling Liu, Yiman Ren, Keyu Li, and David Carlson. Estimating causal effects using a multi-task deep ensemble. In *Proceedings of the 40 th International Conference on Machine Learning (ICML)*, PMLR, pp. 680–689, 2023.

Nathan Kallus, Xiaojie Mao, and Madeleine Udell. Causal inference with noisy and missing covariates via matrix factorization. *Advances in Neural Information Processing Systems*, 2018.

Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences of the United States of America*, 116:4156 – 4165, 2017. URL https://api.semanticscholar.org/CorpusID:73455742.

Manabu Kuroki and Judea Pearl. Measurement bias and effect restoration in causal inference. *Biometrika*, 101, 2014.

F. Liang, B. Jia, J. Xue, Q. Li, and Y. Luo. An imputation-regularized optimization algorithm for high-dimensional missing data problems and beyond. *Journal of the Royal Statistical Society, Series B*, 80(5):899–926, 2018a.

F. Liang, Q. Li, and L. Zhou. Bayesian neural networks for selection of drug sensitive genes. *Journal of the American Statistical Association*, 113:955–972, 2018b.

Faming Liang. An effective bayesian neural network classifier with a comparison study to support vector machine. *Neural Computation*, 15:1959–1989, 2003. URL https://api.semanticscholar.org/CorpusID:14168018.

Siqi Liang, Yan Sun, and Faming Liang. Nonlinear sufficient dimension reduction with a stochastic neural network. *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022.

Christos Louizos, Uri Shalit, Joris M. Mooij, David A. Sontag, Richard S. Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:260564.

Wang Miao, Zhi Geng, and Eric J. Tchetgen Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 105, 2018.

S.F. Nielsen. The stochastic em algorithm: Estimation and asymptotic results. *Bernoulli*, 6: 457–489, 2000.

Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 01 2009. doi: 10.1214/09-SS057.

Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018.

Severi Rissanen and Pekka Marttinen. A critical look at the consistency of causal estimation with deep latent variable models. *Advances in Neural Information Processing Systems*, 2021.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70:41–55, 1983. URL `https://api.semanticscholar.org/CorpusID:49190930`.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66:688–701, 1974. URL `https://api.semanticscholar.org/CorpusID:52832751`.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *arXiv:1708.06633*, 2017.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34 th International Conference on Machine Learning (ICML)*, volume 6 of *PMLR*, pp. 4709–4718, International Convention Centre, Sydney, Australia, 06–11 Aug 2017.

Claudia Shi, David M. Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. In *NeurIPS*, 2019.

Q. Song, Y. Sun, M. Ye, and F. Liang. Extended stochastic gradient mcmc for large-scale bayesian variable selection. *Biometrika*, 107(4):997–1004, 2020.

Y. Sun, Q. Song, and F. Liang. Consistent sparse deep learning: Theory and computation. *Journal of the American Statistical Association*, 117(540):1981–1995, 2022.

Yan Sun and Faming Liang. A kernel-expanded stochastic neural network. *Journal of the Royal Statistical Society Series B*, 84(2):547–578, 2022.

Eric J Tchetgen Tchetgen, Andrew Ying, Yifan Cui, Xu Shi, and Wang Miao. An introduction to proximal causal learning. *arxiv preprint, arXiv:2009.10982*, 2020.

Mark J. van der Laan and Daniel Rubin. Targeted maximum likelihood learning. *The International Journal of Biostatistics*, 2(1):11, 2006.

Yixin Wang and David M. Blei. The blessings of multiple causes. *Journal of the American Statistical Association*, 114:1574 – 1596, 2018. URL `https://api.semanticscholar.org/CorpusID:21694910`.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018. URL `https://api.semanticscholar.org/CorpusID:65516833`.

A Appendix

A1 Supplementary Examples

A1.1 Figures for the Simulation Study in Section 2.3

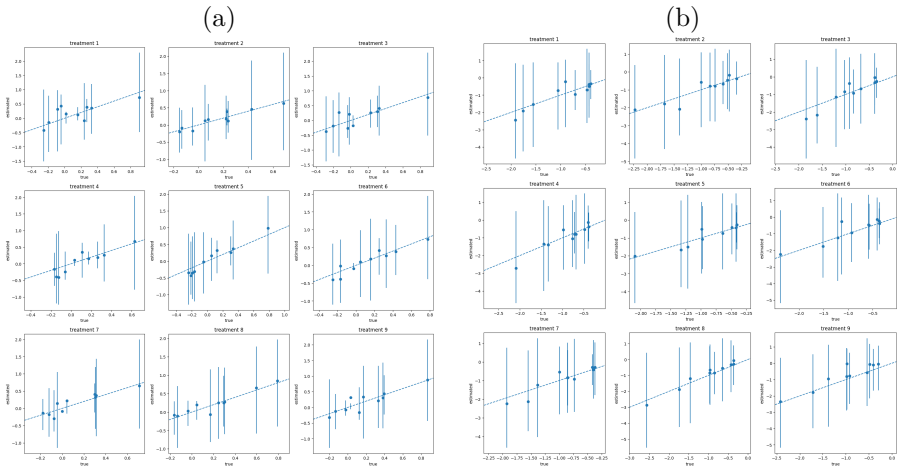

Figure S1: CI-StoNet Results for the simulation study: (a) separable confounding; (b) non-separable confounding, where the point shows the true marginal effect, and the error bar represents one standard error of the marginal effect estimator.

A1.2 Simulated Examples

The simulated examples are used to compare the performance of different methods for problems with nonlinear treatment effect and nonlinear outcome function. We generated ten datasets using the following procedure, with each dataset comprising 2000 training samples, 500 validation samples, and 500 test samples.

1. Generate the confounder $\boldsymbol{z}_i = (z_{i,1}, \cdots, z_{i,5})$ independently from $N(0,1)$.
2. Generate $\gamma_i, r_{i,1}, \cdots, r_{i,100}$ independently from $N(\mu_i, 1)$ truncated in the interval $[-10, 10]$, with $\mu_i = \frac{1}{5}\sum_{k=1}^{5} z_{i,k}$. Set the proxy variable $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,100})$, with $x_{i,j} = \frac{\gamma_i + r_{i,j}}{\sqrt{2}}$, where $\boldsymbol{x}$ and $\boldsymbol{z}$ are dependent through μ.
3. The propensity score $p(\boldsymbol{z}_i) = \frac{1}{4}(1 + \beta_{2,4}(\frac{1}{3}(\Phi(z_{i,1}) + \Phi(z_{i,3}) + \Phi(z_{i,5}))))$, where $\beta_{2,4}$ is the CDF of the beta distribution with shape parameters (2, 4), and Φ denotes the CDF of the standard normal distribution. This ensures that $p(\boldsymbol{z}_i) \in [0.25, 0.5]$, thereby providing sufficient overlap. Treatment A_i is hence generated from a Bernoulli distribution with the success probability $p(\boldsymbol{z}_i)$. Resampling from the treatment and control groups has been performed for ensuring that the dataset contains balanced samples for treatment group and control group.
4. To simulate the outcome, we set

$$y_i = c(\boldsymbol{z}_i) + (\tau + \eta(\boldsymbol{z}_i))A_i + \sigma_y e_i,$$
$$c(\boldsymbol{z}_i) = \frac{5z_{i3}}{1 + z_{i4}^2} + 2z_{i5},$$

where $\eta(\boldsymbol{z}_i) = f(z_{i1})f(z_{i2}) - E(f(z_{i1})f(z_{i2}))$ and $f(w) = \frac{2}{1+\exp(-w+0.5))}$. That is, we set the treatment effect $\tau(\boldsymbol{z}_i) = \tau + \eta(\boldsymbol{z}_i)$, which is homogeneous for different individuals. We generated the samples under the setting $\tau = 3$, $\sigma_y = 0.25$, and $e_i \sim N(0,1)$.

A1.3 Benchmark Datasets

We compare the performance of the proposed method on some benchmark datasets, including the Twins dataset and 10 datasets from Atlantic Causal Inference Conference (ACIC) 2019 Data Challenge.

ACIC 2019 Datasets. We first worked on 10 ACIC 2019 datasets. This experiment focuses on comparing CI-StoNet with the baselines designed for ATE estimation. The results are summarized in Table S1, which indicates that CI-StoNet outperforms the baselines.

Table S1: ATE estimation across 10 ACIC 2019 datasets, where the number in the parentheses represents the standard deviation of the MAE.

Method	In-Sample	Out-of-Sample
CI-StoNet	**0.0669 (0.0166)**	**0.0709 (0.0133)**
DSE	0.0776 (0.0193)	0.1632 (0.0251)
ARBE	0.0729 (0.0166)	0.1335 (0.0179)
TMLE(Lasso)	0.0869 (0.0164)	0.0867 (0.0165)
TMLE(ensemble)	0.1140 (0.0394)	0.1316 (0.0429)
DONUT	0.5294 (0.2640)	0.5290(0.2642)

Twins Data. We analyzed a real-world dataset of twin births from 1989 to 1991 in the United States. The treatment variable is binary, with '1' denoting the heavier twin at birth. The dataset contains 46 variables that include clinical information and socioeconomic status of parents, and we regard them as proxy variables for latent confounders. The outcome variable is binary, with '1' indicating twin mortality within the first year. We regard each twin-pair's records as potential outcomes, allowing us to find the true ATE. After data pre-processing, we obtained a dataset with 4,821 samples. In this final dataset, mortality rates for lighter and heavier twins are 16.9% and 14.42%, respectively, resulting in a true ATE of -2.48%.

We conducted the experiment in three-fold cross validation, where we partitioned the dataset into three subsets, trained the model using two subsets and estimated the ATE using the remaining one. Table S2 (left panel) reports the averaged ATE over three folds and the standard deviation of the average. CI-StoNet yields a more stable ATE estimate (in RMSE) compared to the baseline methods.

Table S2: Comparison of different methods in average treatment effect (ATE) estimation for Twins data, where the number in the parentheses represents the standard deviation of the absolute error of ATE, and RMSE denotes the root mean squared error.

Methods	With confounder *gestat10*		Missing confounder *gestat10*	
	Absolute Error of ATE	RMSE	Absolute Error of ATE	RMSE
CI-StoNet	0.0099(0.0089)	**0.0133**	**0.0135(0.0071)**	**0.0153**
DSE	0.0157(0.0176)	0.0236	0.0211(0.0193)	0.0286
ARBE	0.0152 (0.0201)	0.0252	0.0168(0.0257)	0.0307
TMLE(Lasso)	0.0855 (0.0599)	0.1044	0.0932(0.0791)	0.1222
TMLE(ensemble)	0.1042 (0.0779)	0.1301	0.1238(0.0607)	0.1379
DONUT	0.0490 (0.0128)	0.0506	0.0490(0.0124)	0.0505
CMDE	0.0108(0.0905)	0.0911	0.0635(0.0905)	0.1106
CEVAE	0.0249(**0.0002**)	0.0249	0.0327(0.0633)	0.0712
Ganite	0.3519 (0.1533)	0.3838	0.4198(0.2278)	0.4776
X-learner-RF	**0.0056** (0.0257)	0.0252	0.0157(0.0257)	0.0301
X-learner-Bart	0.0194 (0.0192)	0.0273	0.0251(0.0312)	0.0400
CFRNet-Wass	0.0189 (0.0425)	0.0465	0.0211(0.0254)	0.0330
CFRNet-MMD	0.0439 (0.0146)	0.0463	0.0619(0.0158)	0.0639

Finally, to provide more convincing evidence that the proposed method performs well when confounders are missing, we conducted an experiment where a significant confounder, *gestat10* (gestational age), is intentionally omitted. In preprocessing the dataset, we followed Louizos et al. (2017) to focus on the same-sex twin pairs with birth weights less than 2 kg, and used the variable *gestat10* to generate "pseudo treatment assignments". Since *gestat10* is also an important factor for newborn mortality, it serves as a significant confounder. We removed *gestat10* from the dataset. The results in Table S2 (right panel) show that CI-StoNet exhibits robust performance in presence of missing confounders. In this scenario, it outperforms all baselines in both the absolute error of ATE and RMSE, indicating the superiority of CI-StoNet gained from latent confounder imputation.

A2 Extension to other Causal Structures

The Markovian structure embedded in CI-StoNet provides it with great flexibility to model a wide range of causal structures. In this section, we extend CI-StoNet to handle other causal structures involving proxy variables. Specifically, we consider two scenarios: the outcome depending on the proxy, and the treatment depending on the proxy.

A2.1 Outcome Depending on Proxy

When outcome depends on proxy, see Figure S2(a), the imputation of $\boldsymbol{Z}$ is based on the following decomposition:

$$\pi(\boldsymbol{Z}|\boldsymbol{A},\boldsymbol{Y},\boldsymbol{X}) \propto \pi(\boldsymbol{Z})\pi(\boldsymbol{X}|\boldsymbol{Z})\pi(\boldsymbol{A}|\boldsymbol{Z})\pi(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{A},\boldsymbol{X}) \propto \pi(\boldsymbol{Z}|\boldsymbol{X})\pi(\boldsymbol{A}|\boldsymbol{Z})\pi(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{A},\boldsymbol{X}).$$

Accordingly, the structure of the CI-StoNet can be arranged as follows:

$$\begin{aligned} \boldsymbol{Z} &= \mu_1(\boldsymbol{X},\boldsymbol{\theta}_1) + \boldsymbol{e}_z, \\ \boldsymbol{A} &= \mu_2(\boldsymbol{Z},\boldsymbol{\theta}_2) + \boldsymbol{e}_a, \\ \boldsymbol{Y} &= \mu_3(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{A},\boldsymbol{\theta}_3) + \boldsymbol{e}_y, \end{aligned} \tag{A1}$$

where $\boldsymbol{e}_z$, $\boldsymbol{e}_a$, and $\boldsymbol{e}_y$ denote Gaussian random errors. The corresponding diagram is shown in Figure S2(b).

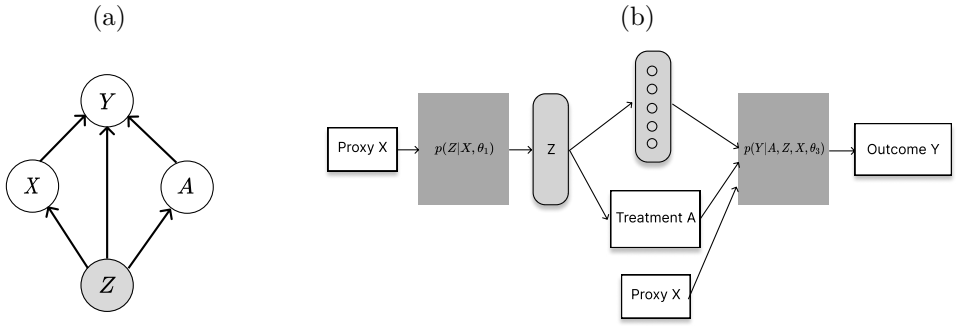

Figure S2: (a) Causal structure: outcome depends on the proxy; and (b) CI-StoNet structure for the scenario where outcome depends on the proxy.

A2.2 Treatment Depending on Proxy

When the treatment depends on the proxy, see Figure S3(a), the imputation of $\boldsymbol{Z}$ is based on the decomposition:

$$\pi(\boldsymbol{Z}|\boldsymbol{A},\boldsymbol{Y},\boldsymbol{X}) \propto \pi(\boldsymbol{Z})\pi(\boldsymbol{X}|\boldsymbol{Z})\pi(\boldsymbol{A}|\boldsymbol{Z},\boldsymbol{X})\pi(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{A}) \propto \pi(\boldsymbol{Z}|\boldsymbol{X})\pi(\boldsymbol{A}|\boldsymbol{Z},\boldsymbol{X})\pi(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{A}).$$

The structure of the CI-StoNet can be arranged as follows:

$$\begin{aligned} \boldsymbol{Z} &= \mu_1(\boldsymbol{X},\boldsymbol{\theta}_1) + \boldsymbol{e}_z, \\ \boldsymbol{A} &= \mu_2(\boldsymbol{Z},\boldsymbol{X},\boldsymbol{\theta}_2) + \boldsymbol{e}_a, \\ \boldsymbol{Y} &= \mu_3(\boldsymbol{Z},\boldsymbol{A},\boldsymbol{\theta}_3) + \boldsymbol{e}_y, \end{aligned} \tag{A2}$$

where $\boldsymbol{e}_z$, $\boldsymbol{e}_a$, and $\boldsymbol{e}_y$ denote Gaussian random errors. The corresponding diagram is shown in Figure S3(b).

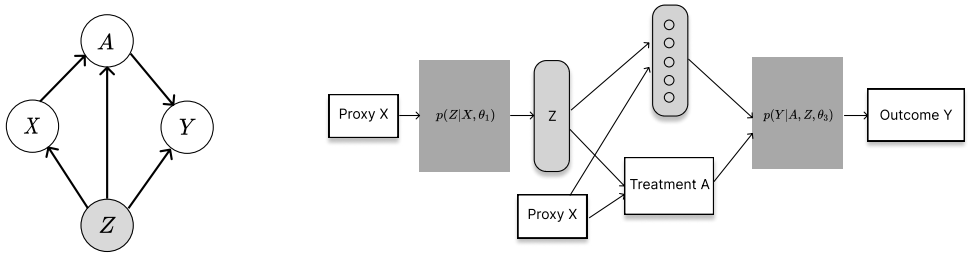

Figure S3: (a) Causal structure and (b) CI-StoNet structure for the scenario where treatment depends on the proxy.

Both models can be trained using an adaptive stochastic gradient MCMC algorithm, and the corresponding causal effects can be estimated based on the imputed confounders from $\pi(\boldsymbol{z}|\boldsymbol{X}, \hat{\boldsymbol{\theta}}_1^*)$.

For causal structures shown in Figures S2(a) and S3(a), X is the proxy variable, Z represents the missing confounder, and A and Y represents the treatment variable and outcome variable, respectively. The white nodes represent observed variables, while the light-grey node represent the unobserved variables. For the CI-StoNet structures shown in Figures S2(b) and S3(b), white rectangles represent variables from observed data; light-grey rounded-rectangles represent hidden neurons; and dark-grey rectangles represent network modules to learn respective conditional distributions.

A3 Theoretical Proofs

A3.1 Convergence of $\boldsymbol{\theta}^{(k)}$

To train the CI-StoNet using the IRO algorithm, it requires that the full dataset is used at each iteration, making the algorithm difficult to scale up to large-scale neural networks. In contrast, the adaptive SGHMC algorithm can use mini-batch data in parameter updating. As shown in Liang et al. (2022), the adaptive SGHMC algorithm solves equation (7) under the following conditions.

Notations: We let $\boldsymbol{D}$ denote a dataset of n observations, and let D_i denote the i-th observation of $\boldsymbol{D}$. For StoNet, D_i has included both the input and output variables of the observation. For the CI-StoNet, D_i includes the treatment and outcome, i.e., $D_i = \{\boldsymbol{A}_i, \boldsymbol{Y}_i\}$. For simplicity of notation, we re-denote the latent variable corresponding to D_i by Z_i, and denote by $f_{D_i}(Z_i, \boldsymbol{\theta}) = -\log \pi(Z_i|D_i, \boldsymbol{\theta})$ the negative log-density function of Z_i. Let $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_n)$, let $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)$ be a realization of $\boldsymbol{Z}$, let $F_{\boldsymbol{D}}(\boldsymbol{Z}, \boldsymbol{\theta}) = \sum_{i=1}^n f_{D_i}(Z_i, \boldsymbol{\theta})$, and let $H(\boldsymbol{Z}, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{Z}|\boldsymbol{A}, \boldsymbol{\theta})$. To study the convergence of the adaptive SGHMC algorithm presented in Algorithm 1, we make the following assumptions:

Assumption A3. *(i) (Boundedness) The function $F_{\boldsymbol{D}}(\cdot, \cdot)$ takes nonnegative real values, and there exist constants $A, B \geq 0$, such that $|F_{\boldsymbol{D}}(0, \boldsymbol{\theta}^*)| \leq A$, $\|\nabla_{\boldsymbol{Z}} F_{\boldsymbol{D}}(0, \boldsymbol{\theta}^*)\| \leq B$, $\|\nabla_{\boldsymbol{\theta}} F_{\boldsymbol{D}}(0, \boldsymbol{\theta}^*)\| \leq B$, and $\|H(0, \boldsymbol{\theta}^*)\| \leq B$.*

(ii) (Smoothness) $F_{\boldsymbol{D}}(\cdot,\cdot)$ is M-smooth and $H(\cdot,\cdot)$ is M-Lipschitz: there exists some constant $M>0$ such that for any $\boldsymbol{Z},\boldsymbol{Z}'\in\mathbb{R}^{d_z}$ and any $\boldsymbol{\theta},\boldsymbol{\theta}'\in\Theta$,

$$\begin{aligned}&\|\nabla_{\boldsymbol{Z}}F_{\boldsymbol{D}}(\boldsymbol{Z},\boldsymbol{\theta})-\nabla_{\boldsymbol{Z}}F_{\boldsymbol{D}}(\boldsymbol{Z}',\boldsymbol{\theta}')\|\le M\|\boldsymbol{Z}-\boldsymbol{Z}'\|+M\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|,\\&\|\nabla_{\boldsymbol{\theta}}F_{\boldsymbol{D}}(\boldsymbol{Z},\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}F_{\boldsymbol{D}}(\boldsymbol{Z}',\boldsymbol{\theta}')\|\le M\|\boldsymbol{Z}-\boldsymbol{Z}'\|+M\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|,\\&\|H(\boldsymbol{Z},\boldsymbol{\theta})-H(\boldsymbol{Z}',\boldsymbol{\theta}')\|\le M\|\boldsymbol{Z}-\boldsymbol{Z}'\|+M\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|.\end{aligned}$$

(iii) (Dissipativity) For any $\boldsymbol{\theta}\in\Theta$, the function $F_{\boldsymbol{D}}(\cdot,\boldsymbol{\theta}^*)$ is (m,b)-dissipative: there exist some constants $m>\frac{1}{2}$ and $b\ge 0$ such that $\langle\boldsymbol{Z},\nabla_{\boldsymbol{Z}}F_{\boldsymbol{D}}(\boldsymbol{Z},\boldsymbol{\theta}^*)\rangle\ge m\|\boldsymbol{Z}\|^2-b$.

(iv) (Gradient noise) There exists a constant $\varsigma\in[0,1)$ such that for any $\boldsymbol{Z}$ and $\boldsymbol{\theta}$, $\mathbb{E}\|\nabla_{\boldsymbol{Z}}\hat{F}_{\boldsymbol{D}}(\boldsymbol{Z},\boldsymbol{\theta})-\nabla_{\boldsymbol{Z}}F_{\boldsymbol{D}}(\boldsymbol{Z},\boldsymbol{\theta})\|^2\le 2\varsigma(M^2\|\boldsymbol{Z}\|^2+M^2\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\|^2+B^2)$.

Assumption A4. *The step size $\{\gamma_k\}_{k\in\mathbb{N}}$ is a positive decreasing sequence such that $\gamma_k\to 0$ and $\sum_{k=1}^{\infty}\gamma_k=\infty$. In addition, let $h(\boldsymbol{\theta})=\mathbb{E}(H(\boldsymbol{Z},\boldsymbol{\theta}))$, then there exists $\delta>0$ such that for any $\boldsymbol{\theta}\in\Theta$, $\langle\boldsymbol{\theta}-\boldsymbol{\theta}^*,h(\boldsymbol{\theta})\rangle\ge\delta\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\|^2$, and $\liminf_{k\to\infty}2\delta\frac{\gamma_k}{\gamma_{k+1}}+\frac{\gamma_{k+1}-\gamma_k}{\gamma_{k+1}^2}>0$.*

Assumption A5. *(Solution of Poisson equation) For any $\boldsymbol{\theta}\in\Theta$, $\boldsymbol{z}\in\mathfrak{Z}$, and a function $V(\boldsymbol{z})=1+\|\boldsymbol{z}\|$, there exists a function $\mu_{\boldsymbol{\theta}}$ on $\mathfrak{Z}$ that solves the Poisson equation $\mu_{\boldsymbol{\theta}}(\boldsymbol{z})-\mathcal{T}_{\boldsymbol{\theta}}\mu_{\boldsymbol{\theta}}(\boldsymbol{z})=H(\boldsymbol{\theta},\boldsymbol{z})-h(\boldsymbol{\theta})$, where $\mathcal{T}_{\boldsymbol{\theta}}$ denotes a probability transition kernel with $\mathcal{T}_{\boldsymbol{\theta}}\mu_{\boldsymbol{\theta}}(\boldsymbol{z})=\int_{\mathfrak{Z}}\mu_{\boldsymbol{\theta}}(\boldsymbol{z}')\mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{z}')d\boldsymbol{z}'$, such that*

$$H(\boldsymbol{\theta}_k,\boldsymbol{z}_{k+1})=h(\boldsymbol{\theta}_k)+\mu_{\boldsymbol{\theta}_k}(\boldsymbol{z}_{k+1})-\mathcal{T}_{\boldsymbol{\theta}_k}\mu_{\boldsymbol{\theta}_k}(\boldsymbol{z}_{k+1}),\quad k=1,2,\ldots. \tag{A3}$$

Moreover, for all $\boldsymbol{\theta},\boldsymbol{\theta}'\in\Theta$ and $\boldsymbol{z}\in\mathfrak{Z}$, we have $\|\mu_{\boldsymbol{\theta}}(\boldsymbol{z})-\mu_{\boldsymbol{\theta}'}(\boldsymbol{z})\|\le\varsigma_1\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|V(\boldsymbol{z})$ and $\|\mu_{\boldsymbol{\theta}}(\boldsymbol{z})\|\le\varsigma_2 V(\boldsymbol{z})$ for some constants $\varsigma_1>0$ and $\varsigma_2>0$.

Proof of Lemma 1

Proof. Lemma 1 is a restatement of Theorem S1 of Liang et al. (2022), and its proof is thus omitted. □

A3.2 Consistency of $\hat{\boldsymbol{\theta}}_n^*$

A3.2.1 Consistency of the IRO Algorithm

The IRO Algorithm The IRO algorithm (Liang et al., 2018a) starts with an initial weight setting $\hat{\boldsymbol{\theta}}^{(0)}=(\hat{\boldsymbol{\theta}}_1^{(0)},\hat{\boldsymbol{\theta}}_2^{(0)})$ and then iterates between the imputation of latent confounders and regularized optimization for parameter updating:

- **Imputation:** simulate $\boldsymbol{z}_i^{(t+1)}$ from the predictive distribution:
$$\pi(\boldsymbol{z}_i\mid y_i,\boldsymbol{a}_i,\hat{\boldsymbol{\theta}}^{(t)},\boldsymbol{\sigma}_{CI}^2)\propto\pi(\boldsymbol{z}_i\mid\boldsymbol{a}_i,\hat{\boldsymbol{\theta}}_1^{(t)},\sigma_z^2)\pi(y_i\mid\boldsymbol{z}_i,\boldsymbol{a}_i,\hat{\boldsymbol{\theta}}_2^{(t)},\sigma_y^2)$$
where t indexes iterations, and $\boldsymbol{\sigma}_{CI}^2=(\sigma_z^2,\sigma_y^2)$.
- **Regularized optimization:** Given the pseudo-complete data $\{(y_i,\boldsymbol{z}_i^{(t+1)},\boldsymbol{a}_i):i=1,2,\ldots,n\}$, update $\hat{\boldsymbol{\theta}}^{(t+1)}$ by maximizing the penalized log-likelihood function as follows:
$$\hat{\boldsymbol{\theta}}^{(t+1)}=\arg\max_{\boldsymbol{\theta}}\Big\{\frac{1}{n}\sum_{i=1}^{n}\log\pi(y_i,\boldsymbol{z}_i^{(t+1)}|\boldsymbol{a}_i,\boldsymbol{\theta},\boldsymbol{\sigma}_{CI}^2)-\frac{1}{n}\log P_{\lambda_n}(\boldsymbol{\theta})\Big\}. \tag{A4}$$

The penalty function $\frac{1}{n}\log P_{\lambda_n}(\boldsymbol{\theta})$ satisfies some conditions (see Assumption A8) such that $\hat{\boldsymbol{\theta}}^{(t+1)}$ forms a consistent estimator, uniformly over iterations, for the working parameter

$$\begin{aligned}\boldsymbol{\theta}_*^{(t+1)}&=\arg\max_{\boldsymbol{\theta}}\mathbb{E}_{\hat{\boldsymbol{\theta}}^{(t)}}\log\pi(y,\boldsymbol{z}|\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\sigma}_{CI}^2)\\&=\arg\max_{\boldsymbol{\theta}}\int\log\pi(y,\boldsymbol{z}|\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\sigma}_{CI}^2)\pi(\boldsymbol{z}\mid y,\boldsymbol{a},\hat{\boldsymbol{\theta}}^{(t)},\sigma_z^2)\pi(y\mid\boldsymbol{a},\boldsymbol{\theta}^*,\sigma_y^2)d\boldsymbol{z}dy,\end{aligned} \tag{A5}$$

where $\boldsymbol{\theta}^*$ denotes the true parameter value of the CI-StoNet model.

Consistency of Parameter Estimation The main proof for the consistency of parameter estimation is built on the theoretical framework developed in Liang et al. (2018a). Let $\tilde{\boldsymbol{x}} = (\boldsymbol{A}, \boldsymbol{Y}, \boldsymbol{Z})$ be the complete data, which is a collection of observed variable and latent variables. Define

$$G_n(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(t)}) = \int \log \pi(y, \boldsymbol{z} | \boldsymbol{a}, \boldsymbol{\theta}, \boldsymbol{\sigma}_{CI}^2) \pi(\boldsymbol{z} \mid y, \boldsymbol{a}, \hat{\boldsymbol{\theta}}^{(t)}, \sigma_z^2) \pi(y \mid \boldsymbol{a}, \boldsymbol{\theta}^*, \sigma_y^2) d\boldsymbol{z} dy,$$

$$\hat{G}_n(\boldsymbol{\theta} \mid \tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}^{(t)}) = \frac{1}{n} \sum_{i=1}^{n} \log \pi(y_i, \boldsymbol{z}_i | \boldsymbol{a}_i, \boldsymbol{\theta}, \boldsymbol{\sigma}_{CI}^2), \quad \boldsymbol{z}_i \sim \pi(\boldsymbol{z} | y_i, \boldsymbol{a}_i, \hat{\boldsymbol{\theta}}^{(t)}, \sigma_z^2),$$

Lemma A1. *(Theorem 1; Liang et al. (2018a)) Let T denote the total number of iterations of the IRO algorithm. Under mild regularity conditions (See Assumptions 1-3 in Liang et al. (2018a)), the following uniform law of large numbers holds for any T, with $\log(T) = o(n)$:*

$$\sup_{\hat{\boldsymbol{\theta}}^{(t)} \in \boldsymbol{\theta}^T} \sup_{\boldsymbol{\theta} \in \Theta} \left| \hat{G}_n(\boldsymbol{\theta} \mid \tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}^{(t)}) - G_n(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(t)}) \right| \xrightarrow{p} 0, \tag{A6}$$

as the sample size $n \to \infty$.

Assumption A6. *For each $t = 1, 2, \ldots, T$, $G_n(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(t)})$ has a unique maximum (up to loss-invariant transformations) at $\boldsymbol{\theta}_*^{(t)}$; for any $\epsilon > 0$, $\sup_{\boldsymbol{\theta} \in \Theta \setminus B_t(\epsilon)} G_n(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(t)})$ exists, where $B_t(\epsilon) = \{\boldsymbol{\theta} \in \Theta : \|\boldsymbol{\theta} - \boldsymbol{\theta}_*^{(t)}\| < \epsilon\}$. Let $\delta_t = G_n(\boldsymbol{\theta}_*^{(t)} \mid \hat{\boldsymbol{\theta}}^{(t)}) - \sup_{\boldsymbol{\theta} \in \Theta \setminus B_t(\epsilon)} G_n(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(t)})$, $\delta = \min_{t \in \{1,2,\ldots,T\}} \delta_t > 0$ holds.*

Assumption A6 restricts the shape of $G_n(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}^{(t)})$ around the global maximizer, ensuring that it is neither discontinuous nor too flat. Given the nonidentifiability of neural network models, Assumption A6 implicitly assumes that each $\boldsymbol{\theta}$ is unique up to loss-invariant transformations, such as reordering the hidden neurons within the same layer or simultaneously altering the signs or scales of certain weights and biases, see e.g., Liang et al. (2018b) and Sun et al. (2022) for further discussions. Alternatively, the optimal solutions can be considered as belonging to an equivalence class, subject to appropriate loss-invariant transformations, with the uniqueness assumption applying to this equivalence class.

Furthermore, consider the mapping $M(\boldsymbol{\theta})$ defined by

$$M(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}'} \mathbb{E}_{\boldsymbol{\theta}} \log \pi(\boldsymbol{Y}, \boldsymbol{Z} | \boldsymbol{a}, \boldsymbol{\theta}', \boldsymbol{\sigma}_{CI}^2).$$

As argued in Liang et al. (2018a) and Nielsen (2000), it is reasonable to assume that the mapping is a contraction, as a recursive application of the mapping, i.e., setting

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \boldsymbol{\theta}_*^{(t+1)} = M(\hat{\boldsymbol{\theta}}^{(t)}),$$

leads to a monotone increase of the target expectations $\mathbb{E}_{\hat{\boldsymbol{\theta}}^{(t)}} \log \pi(\boldsymbol{Y}, \boldsymbol{Z} | \boldsymbol{a}, \boldsymbol{\theta}, \boldsymbol{\sigma}_{CI}^2)$ for $t = 1, 2, \ldots$.

Assumption A7. *The mapping $M(\boldsymbol{\theta})$ is differentiable. Let $\rho_n(\boldsymbol{\theta})$ be the largest singular value of $\partial M(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}$. There exists a number $\rho^* < 1$ such that $\rho_n(\boldsymbol{\theta}) \leq \rho^*$ for all $\boldsymbol{\theta} \in \Theta$ for sufficiently large n and almost every observed sequence of $(\boldsymbol{A}, \boldsymbol{Y})$.*

Assumption A8. *The penalty function $\frac{1}{n} \log P_{\lambda_n}(\boldsymbol{\theta})$ converges to 0 uniformly over the set $\{\boldsymbol{\theta}_*^{(t)} : t = 1, 2, \ldots, T\}$ as $n \to \infty$, where λ_n is a regularization parameter and its value can depend on the sample size n.*

Lemma A2. *(Theorem 4; Liang et al. (2018a)) Suppose the conditions of Lemma A1, Assumptions A6-A8 hold, and $\sup_{n,t} \mathbb{E} \|\hat{\boldsymbol{\theta}}_n^{(t)}\| < \infty$ hold. Then for sufficiently large t and almost every $(\boldsymbol{A}, \boldsymbol{Y})$-sequence, $\|\hat{\boldsymbol{\theta}}_n^{(t)} - \boldsymbol{\theta}^*\| \xrightarrow{p} 0$, as $n \to \infty$.*

A3.2.2 Verification of Assumption A8

To verify Assumption A8, we prove the following lemma.

Lemma A3. *Let $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_{K_n})^T$. Suppose that all components of $\boldsymbol{\theta}$ are a priori independent and they are subject to the following mixture Gaussian prior (6). Suppose $K_n \succ n$, $\boldsymbol{\theta}$ is sparse at a level of $m_n \prec \frac{n}{c\log(K_n/n)}$ for some constant $c > 1$, and $\min\{|\theta_i| : \theta_i \neq 0, i = 1, 2, \ldots, K_n\} > \delta_n$ for some constant $\delta_n = o(1)$. If we set $\sigma_1 = O(1)$ and set (λ_n, σ_0) to satisfy the conditions:*

$$\begin{aligned} &(\frac{n}{K_n})^c \prec \lambda_n \prec \frac{n}{K_n}, \\ &(\frac{n}{K_n})^c \prec \sigma_0 \prec \min\left\{1 - \frac{n}{K_n}, \frac{\delta_n}{\sqrt{c\log(K_n) - (c-1)\log(n)}}\right\}, \end{aligned} \tag{A7}$$

then the following result holds:

$$\frac{1}{n}\left|\log \pi(\boldsymbol{\theta}) + K_n \log\left(\sqrt{2\pi}\sigma_0\right)\right| \to 0, \quad \text{as } n \to \infty. \tag{A8}$$

Proof. A straightforward calculation shows that

$$\begin{aligned} |\log \pi(\boldsymbol{\theta}) + K_n \log(\sigma_0)| \lesssim K_n|\log(1-\lambda_n)| + (K_n - m_n)\frac{\sigma_0\lambda_n}{\sigma_1(1-\lambda_n)} + m_n\left|\log\left(\frac{\sigma_0\lambda_n}{1-\lambda_n}\right)\right| \\ - m_n\frac{\delta_n^2}{2\sigma_1^2} + \frac{m_n(1-\lambda_n)\sigma_1}{\lambda_n\sigma_0}e^{-\frac{\delta_n^2}{2}(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2})}. \end{aligned}$$

To ensure $K_n|\log(1-\lambda)| \prec n$, we set

$$\lambda_n \prec 1 - e^{-n/K_n} \asymp \frac{n}{K_n}. \tag{A9}$$

To ensure $m_n\left|\log\left(\frac{\sigma_0\lambda_n}{1-\lambda_n}\right)\right| \prec n$, we set

$$\sigma_0 \succ (\frac{n}{K_n})^c \succ e^{-n/m_n}, \quad \lambda_n \succ (\frac{n}{K_n})^c \succ e^{-n/m_n}. \tag{A10}$$

To ensure $(K_n - m_n)\frac{\sigma_0\lambda_n}{\sigma_1(1-\lambda_n)} \prec n$, we set

$$\sigma_0 \prec 1 - \frac{n}{K_n} \prec \frac{n}{K_n}\frac{(1-\lambda_n)}{\lambda_n}. \tag{A11}$$

To ensure $\frac{m_n(1-\lambda_n)\sigma_1}{\lambda_n\sigma_0}e^{-\frac{\delta_n^2}{2}(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2})} \prec n$, we set

$$\sigma_0 \prec \frac{\delta_n}{\sqrt{c\log(K_n) - (c-1)\log(n)}} \prec \frac{\delta_n}{\sqrt{|\log(n\lambda_n/m_n))|}}. \tag{A12}$$

Since $\delta_n \prec o(1)$ and $m_n \prec n$, we have $m_n\frac{\delta_n^2}{2\sigma_1^2} \prec n$.

As a summary of (A9)-(A12), we can set (λ_n, σ_0) as stated in (A7), which ensures (A8) holds. □

A3.2.3 Proof of Theorem 1

Proof. Since $\hat{\boldsymbol{\theta}}_n^*$ is a solution to equation (7), it serves as the maximum *a posteriori* (MAP) estimator of $\boldsymbol{\theta}$ with respect to the incomplete data (by treating $\boldsymbol{Z}$ as missing). By Lemma A2, we immediately have its consistency with respect to $\boldsymbol{\theta}^*$, i.e.,

$$\|\hat{\boldsymbol{\theta}}_n^* - \boldsymbol{\theta}^*\| \xrightarrow{p} 0, \quad \text{as } n \to \infty. \tag{A13}$$

Among the conditions of Lemma A2, we only need to verify Assumption A8, since the others are generally satisfied. Recall that we adopt the mixture Gaussian prior (6) in computing the MAP of $\boldsymbol{\theta}$. By Lemma A3, Assumption A8 is satisfied. This concludes the proof. □

A3.3 Proof of Theorem 2

Proof. Consider the joint density function:

$$\pi(\boldsymbol{Z}, \boldsymbol{Y}|\boldsymbol{A}, \boldsymbol{\theta}^*) = \pi(\boldsymbol{Z}|\boldsymbol{A}, \boldsymbol{\theta}_1^*)\pi(\boldsymbol{Y}|\boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{\theta}_2^*),$$

under the assumption that the true model is a sparse StoNet (4) parameterized by $\boldsymbol{\theta}^*$. Then we have

$$\mathbb{E}(Y(\boldsymbol{a})|\boldsymbol{\theta}^*) = \int \boldsymbol{y}\pi(\boldsymbol{z}|\boldsymbol{A}, \boldsymbol{\theta}_1^*)\pi(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{a}, \boldsymbol{\theta}_2^*)d\boldsymbol{z}d\boldsymbol{y} = \int \mu_2(\boldsymbol{z}, \boldsymbol{a}, \boldsymbol{\theta}_2^*)\pi(\boldsymbol{z}|\boldsymbol{a}, \boldsymbol{\theta}_1^*)d\boldsymbol{z}.$$

Let $\boldsymbol{z}^{(l)}$, for $l = 1, 2, \ldots, \mathcal{M}$, denote $\mathcal{M}$ independent samples drawn from $\pi(\boldsymbol{z}|\boldsymbol{a}, \hat{\boldsymbol{\theta}}_1^*)$. Let

$$\widehat{\mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}_n^*)} = \frac{1}{\mathcal{M}}\sum_{l=1}^{\mathcal{M}} \mu_2(\boldsymbol{z}^{(l)}, \boldsymbol{a}, \hat{\boldsymbol{\theta}}_2^*).$$

By the standard property of Monte Carlo averages, we have

$$\|\widehat{\mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}_n^*)} - \mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}_n^*)\| \xrightarrow{p} 0, \quad \text{as } \mathcal{M} \to \infty. \tag{A14}$$

On the other hand, by the consistency of $\hat{\boldsymbol{\theta}}_n^* = (\hat{\boldsymbol{\theta}}_1^*, \hat{\boldsymbol{\theta}}_2^*)$ (with respect to $\boldsymbol{\theta}^*$) as established in Lemma 1, we have

$$\|\mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}_n^*) - \mathbb{E}(Y(\boldsymbol{a})|\boldsymbol{\theta}^*)\| \xrightarrow{p} 0, \quad \text{as } n \to \infty, \tag{A15}$$

since $\mu_2(\cdot)$ is continuous respect to the parameters (as assumed for the neural network model).

Combining the convergence results in (A14) and (A15), we have

$$\|\widehat{\mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}_n^*)} - \mathbb{E}(Y(\boldsymbol{a})|\boldsymbol{\theta}^*)\| \le \|\widehat{\mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}_n^*)} - \mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}^*)\| + \|\mathbb{E}(Y(\boldsymbol{a})|\hat{\boldsymbol{\theta}}^*) - \mathbb{E}(Y(\boldsymbol{a})|\boldsymbol{\theta}^*)\| \xrightarrow{p} 0,$$

as $n \to \infty$ and $\mathcal{M} \to \infty$. This concludes the proof. □

A4 Experimental Settings

A4.1 Simulated Examples

A4.1.1 Missing confounders

For case with missing confounders, the hidden layers of the network consists of two modules. The first module takes the treatment variables as input and imputes the latent confounder, and the second module takes the concatenated vector of the imputed confounder and the treatment as input to model the outcome. For separable confounding and non-separable scenario, the first module contains two layers with size 32 and 6, and the second layer contains two layers with size 8 and 4. The variance of the noise term $\boldsymbol{e}_z$ and $\boldsymbol{e}_y$ in (4) are set as 10^{-5} and 10^{-3}, respectively. The training consists of three stages - pre-training, training, and finetuning after pruning, with epochs being 100, 500, and 100, respectively. The network is trained like a plain vanilla DNN for pre-training and training, but the decay of imputation learning rate ϵ_k and network parameter learning rate γ_k only starts at training. After training, the network is pruned and refined during the fine-tuning stage with smaller learning rate. Finetuning stage is usually optional and doesn't have dramatic improvement to the overall performance.

The initial imputation learning rate ϵ is set at 5×10^{-4} for non-separable confounding and 10^{-3} for separable confounding, and decays with $\epsilon_k = \frac{\epsilon_k}{1+\epsilon_k \times k^{0.95}}$. The initial parameter learning rate γ is set as 5×10^{-7} and 5×10^{-6}, for the first module and the second module, respectively, and decays with $\gamma_k = \frac{\gamma_k}{1+\gamma_k \times k^{0.7}}$. For the mixture Gaussian prior 6, $\lambda_n = 10^{-6}$, $\sigma_0^2 = 10^{-4}$, and $\sigma_1^2 = 10^{-1}$.

A4.1.2 Proxy Variable

For case with proxy variable, the hidden layers of the network consists of three modules. The first module takes the proxy variables as input and imputes the latent confounder, the second module takes the concatenated vector of the imputed confounder as input to model the treatment variable, and the third module takes the treatment variable as input and model the outcome. The first module contains two layers with size 64 and 32, the second layer contains one layer with size 16, and the third layer contains one layer with size 8.

The variance of the noise term $\boldsymbol{e}_z$, $\boldsymbol{e}_a$, and $\boldsymbol{e}_y$ in (16) are set as 10^{-5}, 10^{-4}, and 10^{-3}, respectively. The training consists of three stages - pre-training, training, and finetuning after pruning, with epochs being 50, 100, and 50, respectively.

The initial imputation learning rate are $\epsilon_1 = 10^{-3}$ and $\epsilon_2 = 10^{-4}$, and decays with $\epsilon_k = \frac{\epsilon_k}{1+\epsilon_k \times k^{0.8}}$. The initial parameter learning rates are set as $\gamma_1 = 5 \times 10^{-6}$, $\gamma_2 = 5 \times 10^{-5}$, and $\gamma_3 = 5 \times 10^{-7}$, for three modules, respectively, and decays with $\gamma_k = \frac{\gamma_k}{1+\gamma_k \times k^{0.6}}$. For the mixture Gaussian prior (6), $\lambda_n = 10^{-6}$, $\sigma_0^2 = 10^{-4}$, and $\sigma_1^2 = 10^{-2}$.

A4.2 Benchmark Dataset

The network structures for benchmark dataset is similar to proxy variable.

A4.2.1 ACIC

The first module contains two layers with size 64 and 32, the second layer contains one layer with size 16, and the third layer contains one layer with size 8.

The variance of the noise term $\boldsymbol{e}_z$, $\boldsymbol{e}_a$, and $\boldsymbol{e}_y$ in (16) are set as 10^{-5}, 10^{-4}, and 10^{-3}, respectively. The training consists of three stages - pre-training, training, and finetuning after pruning, with epochs being 50, 100, and 50, respectively.

The initial imputation learning rate are $\epsilon_1 = 5 \times 10^{-3}$ and $\epsilon_2 = 5 \times 10^{-4}$, and decays with $\epsilon_k = \frac{\epsilon_k}{1+\epsilon_k \times k^{0.8}}$. The initial parameter learning rates are set as $\gamma_1 = 10^{-6}$, $\gamma_2 = 10^{-5}$, and $\gamma_3 = 10^{-7}$, for three modules, respectively, and decays with $\gamma_k = \frac{\gamma_k}{1+\gamma_k \times k^{0.6}}$. For the mixture Gaussian prior (6), $\lambda_n = 10^{-6}$, $\sigma_0^2 = 2 \times 10^{-4}$, and $\sigma_1^2 = 10^{-2}$.

A4.2.2 Twins

The first module contains two layers with size 64 and 32, the second layer contains one layer with size 16, and the third layer contains one layer with size 8.

The variance of the noise term $\boldsymbol{e}_z$, $\boldsymbol{e}_a$, and $\boldsymbol{e}_y$ in (16) are set as 10^{-3}, 10^{-5}, and 10^{-7}, respectively. The training consists of three stages - pre-training, training, and finetuning after pruning, with epochs being 100, 1000, and 200, respectively.

The initial imputation learning rate are $\epsilon_1 = 3 \times 10^{-3}$ and $\epsilon_2 = 5 \times 10^{-5}$, and decays with $\epsilon_k = \frac{\epsilon_k}{1+\epsilon_k \times k^{0.8}}$. The initial parameter learning rates are set as $\gamma_1 = 10^{-3}$, $\gamma_2 = 10^{-5}$, and $\gamma_3 = 10^{-10}$, for three modules, respectively, and decays with $\gamma_k = \frac{\gamma_k}{1+\gamma_k \times k^{0.95}}$. For the mixture Gaussian prior (6), $\lambda_n = 10^{-6}$, $\sigma_0^2 = 2 \times 10^{-5}$, and $\sigma_1^2 = 10^{-2}$.