CPSC 486/586: Probabilistic Machine Learning Due: May 3rd, 2023

Final Project -Diffusion model in Causal Inference-

Lecturer: Andre Wibisono *Name: Tatsuhiro Shimizu*

1 Introduction

Causal Inference is the study of identifying the causal relationships between variables of one's interest and estimating the estimands such as Average Treatment Effect (ATE). Causal Inference is important not only because the causal relationship is the essence of science but also because we can use the consequence of it for Evidence-Based Policy Making (EBPM) beyond political parties and business decision-making. There are two mainstreams in Causal Inference: Potential Outcome framework Imbens and Rubin [2015] and Directed Acyclic Graph (DAG) framework Pearl et al. [2016]. We use the DAG framework. Since the main topic of our lecture was the diffusion model, I summarize three papers Rolland et al. [2022], Sanchez et al. [2022], Chao et al. [2023] on how to use the diffusion model for answering causal questions in the DAG framework under regular settings. Furthermore, I propose the new algorithm I created called BDCM (Backdoor Diffusion-based Causal Model) for answering the causal questions even under irregular settings. In addition, I empirically show that the BDCM works well if we normalize the output. The presentation video and slides are shown in Appendix C C and D D.

2 Setting of the Problem

To clearly state the problem we are going to answer, we need several definitions on causal inference. DAG is the main element of Pearl's framework and is defined as follows.

Definition 1 (DAG)**.** DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a pair of the set of nodes $\mathcal{V}$ and the set of edges $\mathcal{E}$ where $\mathcal{V} = \{1, \cdots, d\}$ and $\mathcal{E} = \{(i, j) \colon \exists \quad \text{edge from node i to j}\}$ DAG expresses variables by nodes and causal relationships by edges

For each node i in DAG $\mathcal{G}$, we have the corresponding variable X_i for all $i \in [d]$. Then, we have the definition of the Structural Causal Model (SCM) as follows.

Definition 2 (Structural Causal Model (SCM))**.** SCM $\mathcal{M} = (\mathcal{U}, \mathcal{V}, f)$ is the tuple of the set of exogenous variables $\mathcal{U} = \{U_1, \cdots, U_d\}$, the set of endogenous variables $\mathcal{V} = \{X_1, \cdots, X_d\}$, and the set of structural equations $f = \{f_1, \cdots, f_d\}$ such that for each $i \in [d]$, the endogenous variable satisfies $X_i = f_i(Pa(X_i), U_i)$ where $Pa(X_i)$ is the set of the parent nodes of X_i.

The example of DAG and SCM are shown in Figure 1 2.

Having DAG 1 and SCM 2, we need to have the definition of the intervention on the variable on the SCM to compare two situations in causal inference by do-operation.

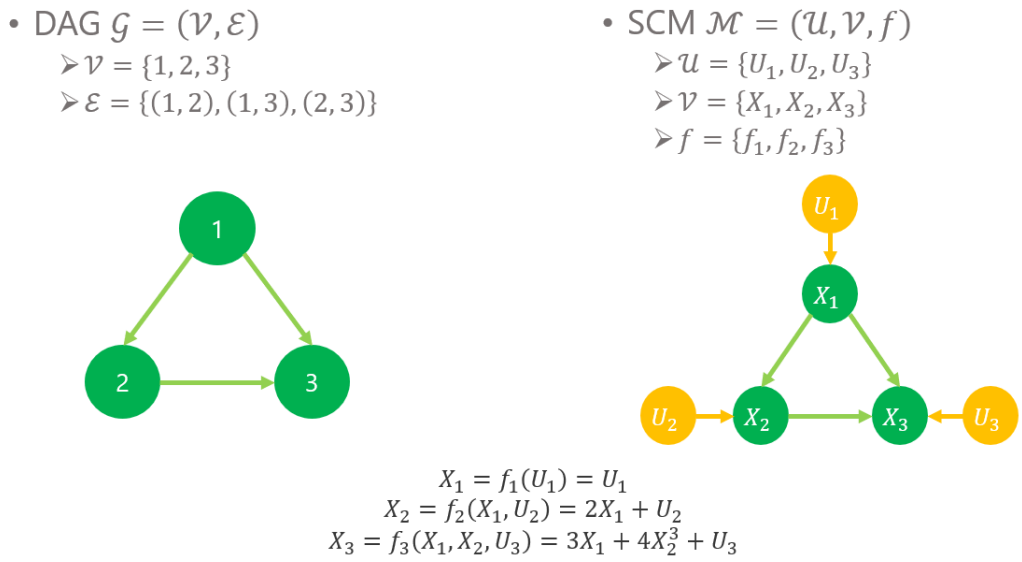

Figure 1: Example of DAG and SCM

Definition 3 (do operator)**.** For any $i \in [d]$, We define $do(X_i = x_i)$ by setting the corresponding exogenous variable to the intervened value $U_i = x_i$ and deleting all the edges coming into X_i from the endogenous variables on SCM 2

The example of do-operation is shown in Figure 2 2.

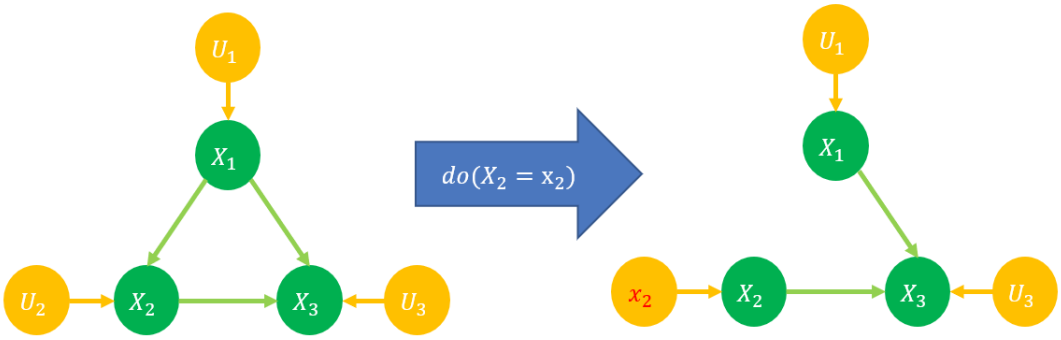

Figure 2: do operator

Then one of the main goals of the causal inference is to estimate the Average Treatment Effect (ATE) given observational data $X \in \mathbb{R}^{n \times d}$ where n is the number of samples and d is the number of nodes and we define ATE as follows.

Definition 4 (Average Treatment Effect (ATE))**.** For all $i, j \in \mathcal{V}, i \neq j$, we define the Average Treatment Effect (ATE) of the variable (cause) X_i on the variable (outcome) X_j when we compare two situations $X_i = x_i$ and $X_i = 0$ by

$$ATE(x_i, 0) := \mathbb{E}[x_j \,|\, do(X_i = x_i)] - \mathbb{E}[x_j \,|\, do(X_i = 0)] \tag{1}$$

$$= \int_{x_j} x_j \nu(X_j = x_j \,|\, do(X_i = x_i)) dx_j - \int_{x_j} x_j \nu(X_j = x_j \,|\, do(X_i = 0)) dx_j \tag{2}$$

where $\nu(X_j = x_j \,|\, do(X_i = x_i))$ is the probability density function of X_j after the surgery on the graph by do operator $do(X_i = x_i)$.

Then, our problem boils down to how to sample from $X_j = x_j \,|\, do(X_i = x_i)$ from the target distribution $\nu(X_j = x_j \,|\, do(X_i = x_i))$ given observational data $X \in \mathbb{R}^{n \times d}$.

3 Literature Review

We can split the above problem into two subproblems as follows.

Problem 5 (Causal Discovery)**.** The first subproblem is how to construct DAG given data $X \in \mathbb{R}^{n \times d}$.

Problem 6 (Sample from target)**.** The second subproblem is how to sample from the target $\nu(X \,|\, do(X_i = x_i))$ given data $X \in \mathbb{R}^{n \times d}$ and DAG where $X = (X_1, \cdots, X_d)$.

The illustration of two split problems is shown in Figure 3 3.

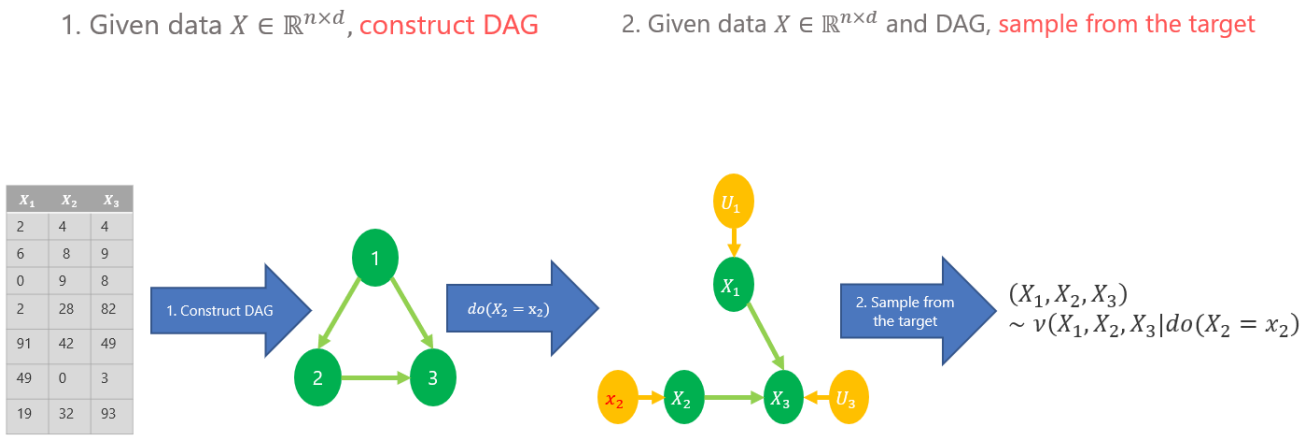

Figure 3: Two Problems

Furthermore, the causal discovery 5 is computationally intensive and NP-hard Chickering [1996], so most of the methods focus on the approximation of it. Since if we know the topological order

π of nodes in DAG, then there exists a known algorithm to construct DAG by pruning edges by selecting the feature Bühlmann et al. [2014] whose definition is shown below, we only focus on the problem of finding the topological order π given observational data.

Definition 7 (Topological Order)**.** Topological order π is a permutation of d nodes in SCM 2 such that $\pi_i < \pi_j \iff j \in De(X_i) \quad (\forall i, j \in \mathcal{V}, i \neq j)$ where $De(X_i)$ is the set of the descendant nodes of i.

Problem 8 (Discovery of Topological Order)**.** To execute the causal discovery 5, we need to find the topological order π given data $X \in \mathbb{R}^{n \times d}$.

I introduce two algorithms for the discovery of the topological order under causal sufficiency 9 and the Additive Noise Model (ANM) 10 and one diffusion-based algorithm for sampling the target distribution under causal sufficiency 9. The definitions of the two assumptions are as follows.

Assumption 9 (Causal Sufficiency)**.** Causal sufficiency means that there are no unmeasured confounders.

Assumption 10 (Additive Noise Model (ANM))**.** SCM 2 is the Additive Noise Model (ANM) if structural equations have form

$$X_i = f_i(Pa(X_i)) + U_i \quad \forall i \in [d] \tag{3}$$

3.1 Given data, construct topological order

To address the problem of the discovery of the topological order π given data, two algorithms I will present use the key idea from the following lemma on how to identify the leaf node.

Lemma 11 (Property of leaf nodes)**.** Given a nonlinear ANM 10, we have

$$\mathbb{V}_x[H_{j,j}(\log \nu(x))] = 0 \iff j \quad \text{is a leaf node} \tag{4}$$

This means that the Variance of the (j, j) element of Jacobian of the score function is 0 if and only if j is the leaf node.

3.1.1 SCORE [Rolland et al. 2022]

By using the property of leaf nodes 11, given the observational data, SCORE Rolland et al. [2022] estimates the Jacobian of the score function by score matching, finds the leaf, deletes the corresponding column of the data and iteratively continues the same procedure to construct topological order π. The concrete algorithm of SCORE is as follows 3.1.1.

The drawback of SCORE is it requires the estimation of the Hessian of log density at each step. Therefore, the complexity of the algorithm is $\mathcal{O}(dn^3)$ where n is the number of samples and d is the number of nodes.

Algorithm 1 SCORE-matching topological order search Rolland et al. [2022]

Input: observational data $X \in \mathbb{R}^{n \times d}$
Initialize the empty topological order $\pi = ()$, and nodes $= \{1, \cdots, d\}$.
for $i = 1, \cdots, d$ **do**
  Estimate the score function $\widehat{s}_{nodes} = \nabla \log \nu_{nodes}$ by using the first and second order Stein's identity
  Estimate $\widehat{V}_j = \mathbb{V}_{X_{ndoes}} \left[\frac{\partial \widehat{s}_j(X)}{\partial x_j} \right]$
  $l \leftarrow$ nodes[argmin$_j$ V_j]
  $\pi \leftarrow (l, \pi)$
  nodes $\leftarrow$ nodes$\backslash\{l\}$
  Remove l-th column of X
end for
return topological order π

3.1.2 DiffAN [Sanchez et al. 2022]

To overcome the drawback of SCORE 3.1.1, Sanchez et al. [2022] proposed DiffAN. DiffAN uses the Denoising Diffusion Probabilistic Model (DDPM) Ho et al. [2020] to estimate the score function as follows.

$$H_{i,j}(\log \nu(x)) \approx \nabla_{i,j} \epsilon_\theta(x, t) \tag{5}$$

The main idea of the algorithm is that we do not need to calculate the Hessian of log density at each iteration by updating the score function using the formula for the residue as follows.

$$\Delta_l(x, t) \approx \nabla_l \epsilon_\theta(x, t) \cdot \frac{\epsilon_\theta(x, t)}{\nabla_{l,l} \epsilon_\theta(x, t)} \tag{6}$$

Then the algorithm of DiffAN is as follows 3.1.2.

Nota that score$(-\pi)$ means that the Algorithm only cares about the outputs for the nodes not in current topological order π. The complexity of DiffAN is $\mathcal{O}(n + d^2)$ by using the mask and without the iterative estimation of the hessian of log density.

3.2 Given data and DAG, sample from the target

Using two algorithms, SCORE 3.1.1 and DiffAN 3.1.2 with some existing edge pruning algorithm Bühlmann et al. [2014], we can obtain the corresponding DAG. Then, the next problem is that given the data and DAG, how to sample from the target distribution $\nu(X \mid do(X_i = x_i))$ 6 under the causal sufficiency assumption 9. Chao et al. [2023] propose Diffusion-based Causal Model (DCM) for answering this problem. They use Denoising Diffusion Implicit Model (DDIM) Song et al. [2021]. DCM trains the diffusion model at each node. The input of the neural network for each node used for the estimation of the noise at each step includes the parent nodes so that the algorithm can have the dependencies between nodes. After the training, we can sample from the target distribution. For

Algorithm 2 DIffAN for topological order search Sanchez et al. [2022]

Input: observational data $X \in \mathbb{R}^{n\times d}$, pre-trained diffusion model ϵ_θ, and batch size k
Initialize the empty topological order $\pi = ()$, cumulative residues $\Delta_\pi = \mathbf{0}^{k\times d}$, a mask $M_\pi = \mathbf{1}^{k\times d}$, and score $= \epsilon_\theta$
while $||\pi|| \neq d$ **do**
  Randomly sample the batch of k elements $B \leftarrow_k X$
  Hide the removed leaves by mask $B \leftarrow B \circ M_\pi$
  Get the residue $\Delta_\pi \leftarrow$ Get $\Delta_\pi(\text{score}, B)$ by the sum of equation 6 over π
  Update the score by residue score $\leftarrow \text{score}(-\pi) + \Delta_\pi$
  leaf $\leftarrow \text{argmin}_{x_i\in\mathcal{X}} \mathbb{V}_B \left[\nabla_x(\text{score}(M_\pi \circ B, t))\right]$
  $\pi \leftarrow (\text{leaf}, \pi)$
  $M_{:,\text{leaf}} \leftarrow \mathbf{0}$
end while
return topological order π

the root node X_i, we just sample from the empirical distribution E_i. For the intervened node X_i, we set it to the intervened value γ_i. For other nodes X_i, we sample by the reverse diffusion process $Dec_i(Z_i, X_{Pa_i})$ using the trained neural network ϵ_θ with parent nodes X_{Pa_i} and the corresponding exogenous nodes $Z_i \sim \mathcal{N}(0, 1)$. The training process 3.2 and sampling process 3.2 are as follows.

Algorithm 3 DCM Training Chao et al. [2023]

Input: target distribution ν, scale factors $\{\alpha_t\}_{t=1}^T$, DAG $\mathcal{G}$ whose node i is represented by X_i
while not converge **do**
  Sample $X^0 \sim \nu$
  for $i = 1, \cdots, d$ **do**
    $t \sim \text{Unif}[\{1, \cdots, T\}]$
    $\epsilon \sim \mathcal{N}(0, 1)$
    Update the parameter of the node i's diffusion model ϵ_θ^i by minimization of the following loss function by Adam optimizer

$$\left\|\epsilon - \epsilon_\theta^i\left(\sqrt{\alpha_t}X_i^0 + \sqrt{1-\alpha_t}\epsilon, X_{pa_i}^0, t\right)\right\|_2^2 \tag{7}$$

  end for
end while

The example of DCM is shown in Figure 4 3.2.

4 Result of New Algorithm

One of the crucial limitations of DCM Chao et al. [2023] is that we cannot cope with the situation where there exist unmeasured confounders, which is often the case with the data collection for both

Algorithm 4 DCM Sampling Chao et al. [2023]

Input: Intervened node j with value γ_j, noise $Z_i \sim \mathcal{N}(0,1)$ for all $i \in [d]$
for $i = 1, \cdots, d$ **do**
  if i is a root node **then**
    $\widehat{X}_i \sim E_i$
  else if $i = j$ **then**
    $\widehat{X}_i \leftarrow \gamma_i$
  else
    $\widehat{X}_i \leftarrow Dec_i\left(Z_i, \widehat{X}_{Pa_i}\right)$
  end if
end for
return $\widehat{X} = \left(\widehat{X}_1, \cdots, \widehat{X}_d\right)$

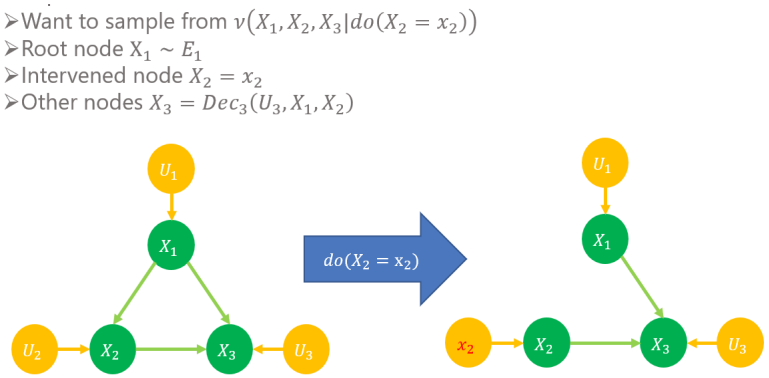

Figure 4: Example of DCM

business and social science where causal inference makes a significant contribution unless we carry out the randomized controlled trial (RCT), which entails huge expense. To overcome this problem and utilize the observational data as much as possible, I introduce the algorithm called Backdoor Diffusion-based Causal Model (BDCM).

4.1 BDCM (Backdoor Diffusion-based Causal Model)

BDCM is the combination of DCM and the backdoor criterion suggested by Pearl et al. [2016]. To propose BDCM, We define the backdoor criterion as follows.

Definition 12 (Backdoor Criterion Pearl et al. [2016])**.** A set of variables $\mathcal{B}$ satisfies backdoor criterion with respect to (X, Y) in DAG $\mathcal{G}$ if 1. No node in $\mathcal{B}$ is a descendant of X and 2. $\mathcal{B}$ blocks all paths between X (cause) and Y (outcome) which contains an arrow into X. If there exist unmeasured confounders, then the Backdoor criterion tells us which variable we should adjust for with respect to (X, Y).

Then, the idea of Backdoor DCM is that for each node X_i in SCM 2, instead of having the parents X_{Pa_i} and corresponding exogenous nodes Z_i as the input of the decoder of the diffusion model, we include the nodes which meet the backdoor criterion $X_{\mathcal{B}_i}$ and the corresponding exogenous nodes Z_i and also include the intervened node X_j if it is the child of the intervened node ($X_j \in X_{Pa_i}$)). Furthermore, we change the training process accordingly.

The training 4.1 and sampling 4.1 of BDCM are as follows.

Algorithm 5 BDCM Training

Input: target distribution ν, scale factors $\{\alpha_t\}_{t=1}^T$, DAG $\mathcal{G}$ whose node i is represented by X_i and intervened node j with intervened value γ_i

while not converge **do**

Sample $X^0 \sim \nu$

for $i = 1, \cdots, d$ **do**

$t \sim \text{Unif}[\{1, \cdots, T\}]$

$\epsilon \sim \mathcal{N}(0, 1)$

Update the parameter of the node i's diffusion model ϵ_θ^i by minimization of the following loss function depending on the nodes.

if $X_j \in X_{Pa_i}$ **then**

$$\left\| \epsilon - \epsilon_\theta^i \left(\sqrt{\alpha_t} X_i^0 + \sqrt{1-\alpha_t}\epsilon, X_{\mathcal{B}_i}^0, X_j, t \right) \right\|_2^2 \tag{8}$$

else

$$\left\| \epsilon - \epsilon_\theta^i \left(\sqrt{\alpha_t} X_i^0 + \sqrt{1-\alpha_t}\epsilon, X_{\mathcal{B}_i}^0, t \right) \right\|_2^2 \tag{9}$$

end if

end for

end while

Then, we have the following conjecture.

Conjecture 13 (potential applicability of BDCM)**.** If there exist sets of nodes that satisfy the backdoor criterion with respect to the intervened node and other nodes, then we can generalize DCM 3.2 to BDCM 4.1 to sample from the target distribution even if the causal sufficiency 9 is not satisfied.

4.2 Experiment

To show that BDCM works well where DCM cannot be used, I conduct an empirical analysis with the following example of the setting where causal sufficiency does not hold.

Example 14. The following SCM 2 $\mathcal{M} = (\mathcal{U}, \mathcal{V}, f)$ does not satisfy the causal sufficiency 9 where the set of exogenous variables is $\mathcal{U} = \{U_1, U_2, U_3, U_Z\}$ such that $U_1, U_2, U_3, U_Z \sim \mathcal{N}(0, 1)$, the set of

Algorithm 6 BDCM Sampling

Input: Intervened node j with value γ_j, noise $Z_i \sim \mathcal{N}(0,1)$ for all $i \in [d]$
for $i = 1, \cdots, d$ **do**
  if i is a root node **then**
    $\widehat{X}_i \sim E_i$
  else if $i = j$ **then**
    $\widehat{X}_i \leftarrow \gamma_i$
  else if $X_j \in X_{Pa_i}$ **then**
    $\widehat{X}_i \leftarrow Dec_i\left(Z_i, \widehat{X}_{\mathcal{B}_i}, X_j\right)$
  else
    $\widehat{X}_i \leftarrow Dec_i\left(Z_i, \widehat{X}_{\mathcal{B}_i}\right)$
  end if
end for
return $\widehat{X} = \left(\widehat{X}_1, \cdots, \widehat{X}_d\right)$

endogenous variables is $\mathcal{V} = \{X_1, X_2, X_3, Z\}$, and the set of structural equations $f = \{f_1, f_2, f_3, f_Z\}$ such that

$$Z = f_z(U_Z) = U_Z + 2 \tag{10}$$
$$X_1 = f_1(Z, U_1) = Z + U_1 + 2 \tag{11}$$
$$X_2 = f_2(Z, U_2) = Z^2 + U_2 \tag{12}$$
$$X_3 = f_3(X_1, X_2, U_3) = X_1^2 - X_1 X_2 + U_3 \tag{13}$$

where Z is the unobserved nodes. Figure 5 4.2 shows the diagram of SCM.

Then, for the example 14 where $n = 5000$ samples are observed, I use the BDCM to sample from the target distribution. One of the results is shown in Figure 6 4.2. The blue histogram is the ground truth distribution we want to sample from whereas the red histogram is the output of the BDCM. The left-hand side of the figure is before I normalize each distribution and the right-hand side shows the distributions after the normalization using each mean and standard error of the empirical distribution. This experiment shows that if we normalize them, then BDCM approximates the target well. The comprehensive result of the experiment is shown in the appendix A. Also, the Python code for the experiment is in the appendix B.

5 Discussion and Future Work

5.1 Contribution

I Combine and summarize the three main recent pieces of literature Rolland et al. [2022], Sanchez et al. [2022], Chao et al. [2023] on the usage of the diffusion model to answer the causal effect. Furthermore, I propose the new algorithm BDCM 4.1 to sample from the target where there exist

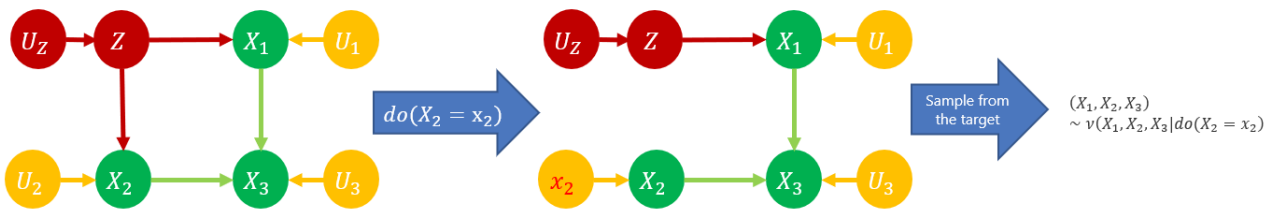

Figure 5: example of the setting where BDCM can be useful

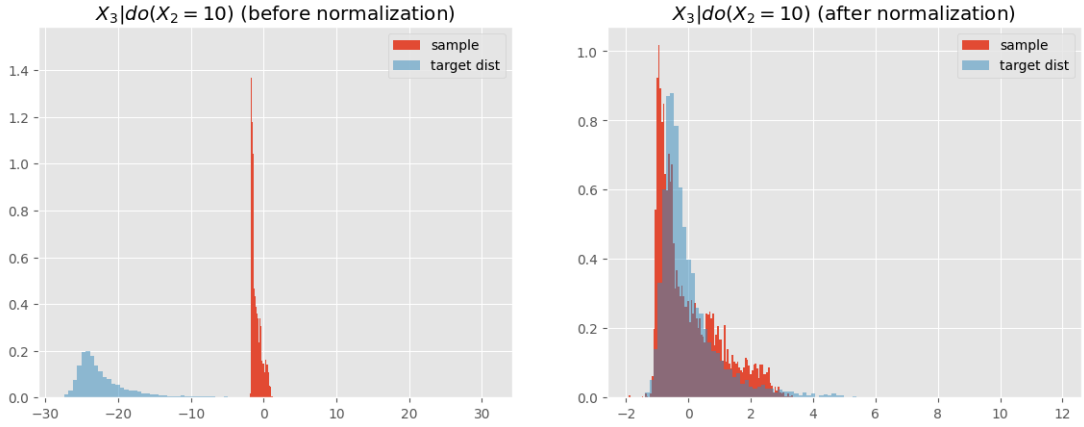

Figure 6: Result of experiment

unobserved confounders and causal sufficiency assumption 9 does not hold. Finally, I show that BDCM works at least with simple setting 14 when we normalize the outputs by experiment.

5.2 Limitation

The limitation of the paper is that BDCM works only when we normalize the target, so we need improvement so that it works without normalization.

5.3 Future Work

For future work, one of the interesting topics is to prove/disprove the convergence guarantee of BDCM. Another one is to Implement the comprehensive algorithm in Python. Also, comparing DCM and BDCM by the empirical analysis will clearly show the supremacy of BDCM over DCM. Moreover, it would be interesting to generalize BDCM by using the Front-door criterion proposed by Pearl et al. [2016] as well, which is another criterion to adjust for the nodes where there exist unobserved confounders.

References

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Science*. Cambridge University Press, 2015.

Judea Pearl, Madelyn Glymour, and Nicholas P. Jewell. *Causal Inference in Statistics: A Primer*. Wiley, 2016.

Paul Rolland, Volkan Cevher, Matthäus Kleindessner, Chris Russel, Bernhard Schölkopf, Dominik Janzing, and Francesco Locatello. Score matching enables causal discovery of nonlinear additive noise models. *arXiv*, arXiv:2203.04413 [cs.LG], 2022. doi: https://doi.org/10.48550/arXiv.2203.04413.

Pedro Sanchez, Xiao Liu, Alison Q O'Neil, and Sotirios A. Diffusion models for causal discovery via topological ordering. *ICLR 2023*, arXiv:2210.06201 [cs.LG], 2022. doi: https://doi.org/10.48550/arXiv.2210.06201.

Patrick Chao, Patrick Blöbaum, and Shiva Prasad Kasiviswanathan. Interventional and counterfactual inference with diffusion models. *arXiv*, arXiv:2302.00860 [stat.ML], 2023. doi: https://doi.org/10.48550/arXiv.2302.00860.

David Maxwell Chickering. *Learning Bayesian Networks is NP-Complete*. Springer, 1996.

Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. *Ann. Statist.*, 42(6): 2526-2556, 2014. doi: 10.1214/14-AOS1260.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv*, arXiv:2006.11239 [cs.LG], 2020. doi: https://doi.org/10.48550/arXiv.2006.11239.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR 2021*, arXiv:2010.02502 [cs.LG], 2021. doi: https://doi.org/10.48550/arXiv.2010.02502.

A the comprehensive result of the experiment

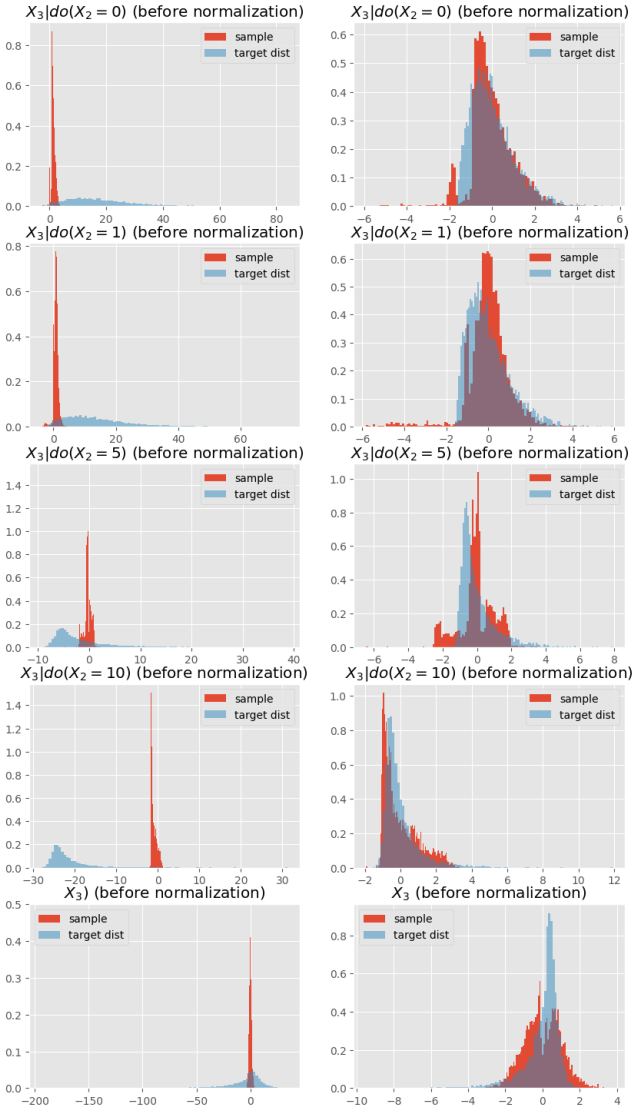

Figure 7: Result of experiment

B Python code for the experiment

GitHub tatsu432/CPSC486-Probabilistic-Machine-Learning-Backdoor-DCM

C Presentation Video

Youtube video CPSC486 Probabilistic Machine Learning Final Project

D Presentation slides

GitHub Presentation Slides