Causal normalizing flows: from theory to practice

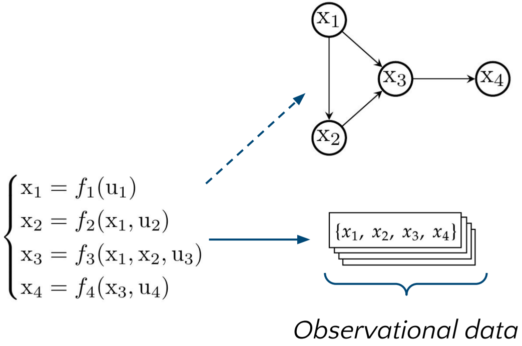
Adrián Javaloy

NeurIPS 2023 – Oral

Pablo Sánchez-Martín

Isabel Valera

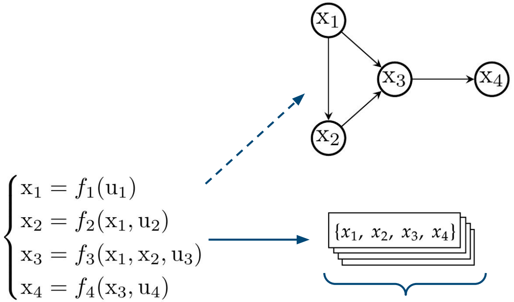
Motivation



$$\begin{cases} x_1 = f_1(u_1) \\ x_2 = f_2(x_1, u_2) \\ x_3 = f_3(x_1, x_2, u_3) \\ x_4 = f_4(x_3, u_4) \end{cases}$$

Observational data

Motivation



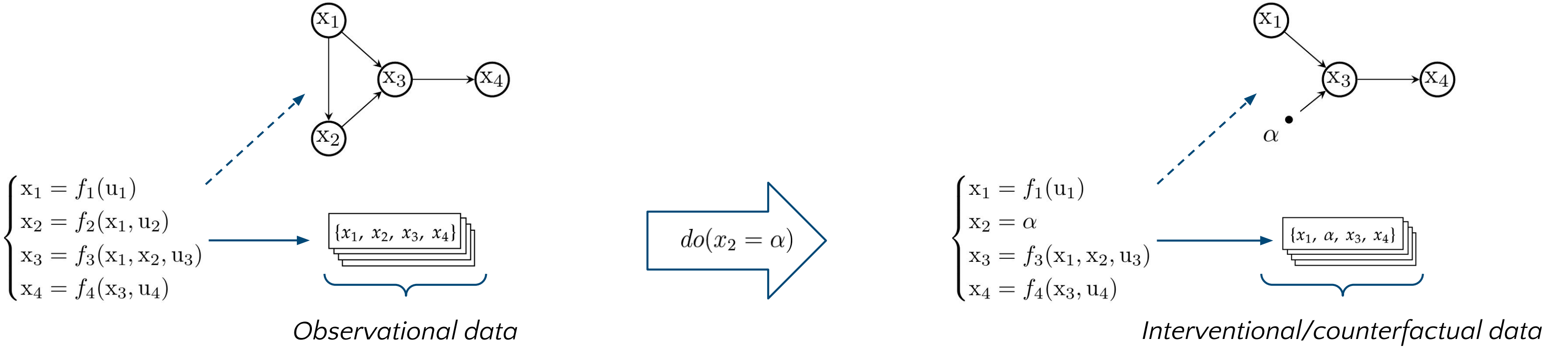
$$\begin{cases} x_1 = f_1(u_1) \\ x_2 = f_2(x_1, u_2) \\ x_3 = f_3(x_1, x_2, u_3) \\ x_4 = f_4(x_3, u_4) \end{cases}$$

$\{x_1,\ x_2,\ x_3,\ x_4\}$

Observational data

“Will I get my health insurance application approved?”

Motivation



$$\begin{cases} x_1 = f_1(u_1) \\ x_2 = f_2(x_1, u_2) \\ x_3 = f_3(x_1, x_2, u_3) \\ x_4 = f_4(x_3, u_4) \end{cases}$$

$$\underbrace{\{x_1,\ x_2,\ x_3,\ x_4\}}_{}$$

Observational data

$do(x_2 = \alpha)$

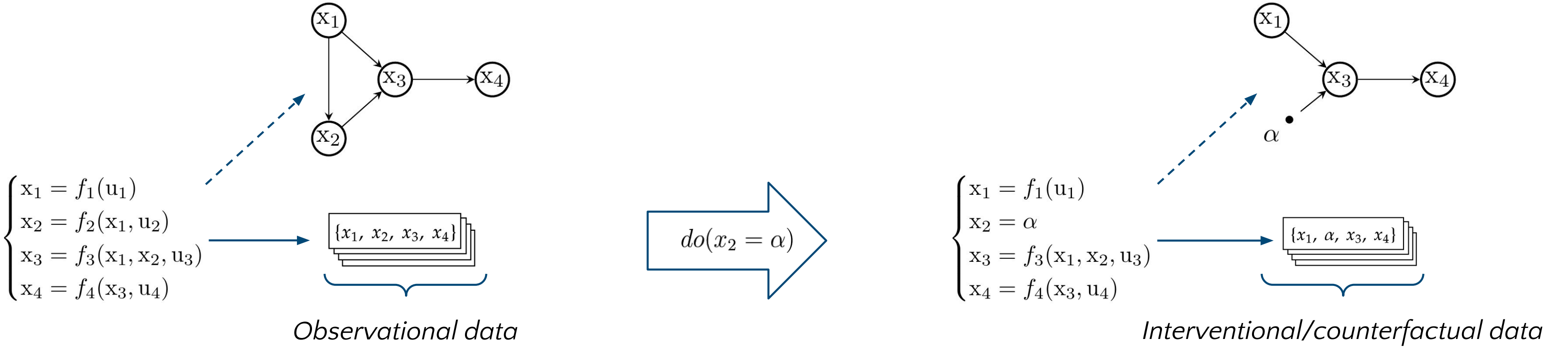
$$\begin{cases} x_1 = f_1(u_1) \\ x_2 = \alpha \\ x_3 = f_3(x_1, x_2, u_3) \\ x_4 = f_4(x_3, u_4) \end{cases}$$

$$\underbrace{\{x_1,\ \alpha,\ x_3,\ x_4\}}_{}$$

Interventional/counterfactual data

“Will I get my health insurance application approved?”

Motivation



$$\begin{cases} x_1 = f_1(u_1) \\ x_2 = f_2(x_1, u_2) \\ x_3 = f_3(x_1, x_2, u_3) \\ x_4 = f_4(x_3, u_4) \end{cases}$$

$$\underbrace{\{x_1,\ x_2,\ x_3,\ x_4\}}$$

Observational data

$$do(x_2 = \alpha)$$

$$\begin{cases} x_1 = f_1(u_1) \\ x_2 = \alpha \\ x_3 = f_3(x_1, x_2, u_3) \\ x_4 = f_4(x_3, u_4) \end{cases}$$

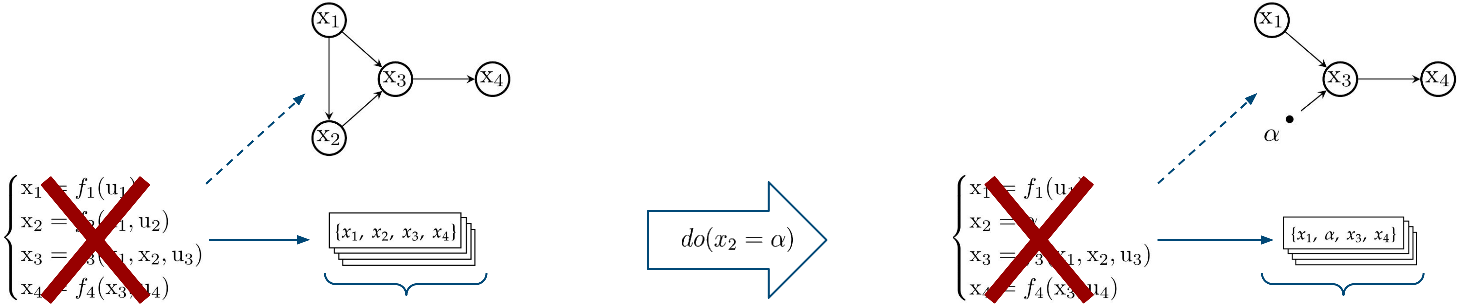
$$\underbrace{\{x_1,\ \alpha,\ x_3,\ x_4\}}$$

Interventional/counterfactual data

“Will I get my health insurance application approved?”

“I got my application rejected. Would I have gotten it if I were younger?”

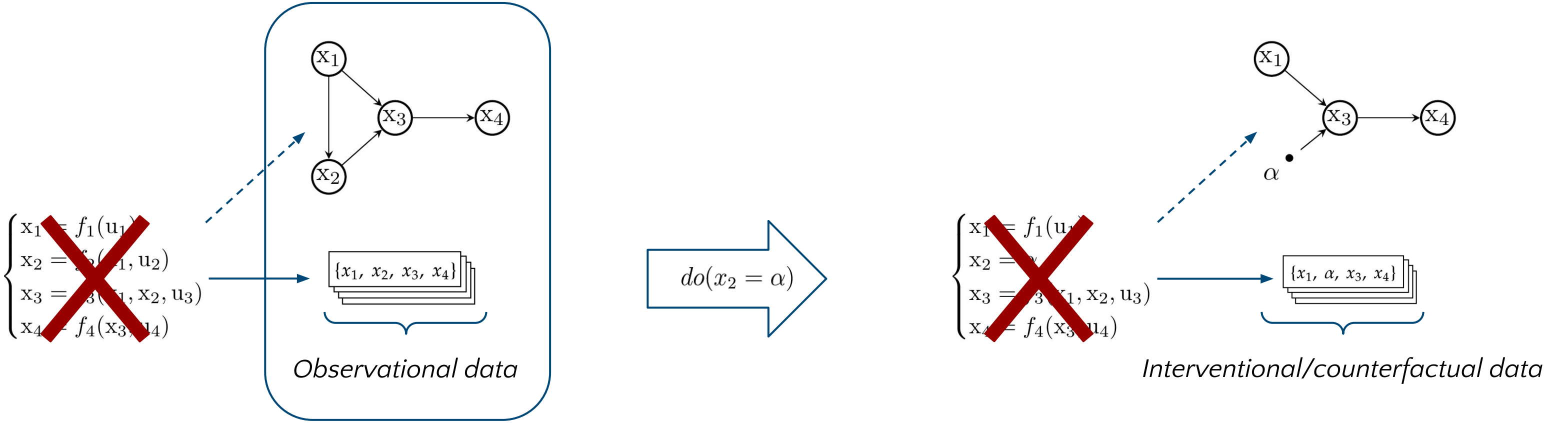
Motivation



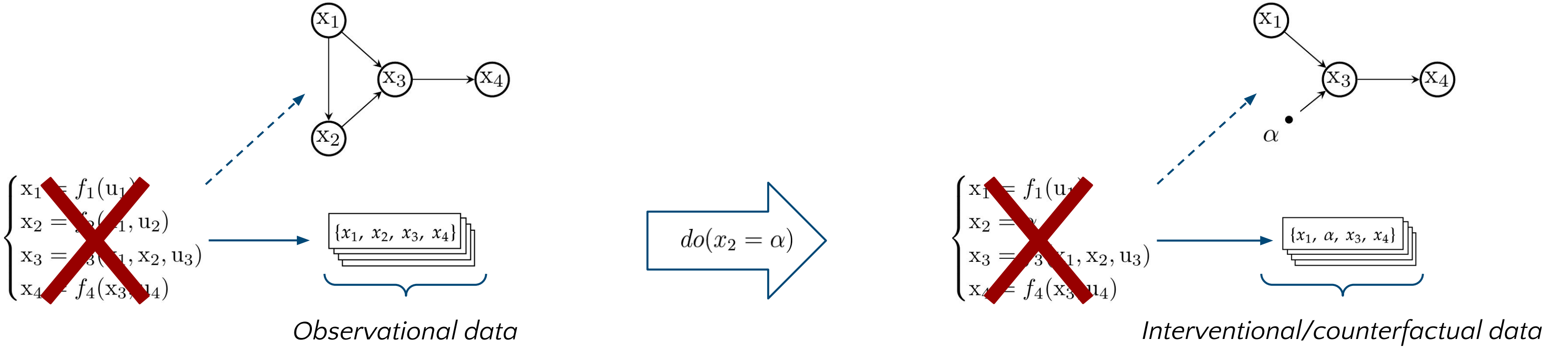
Observational data

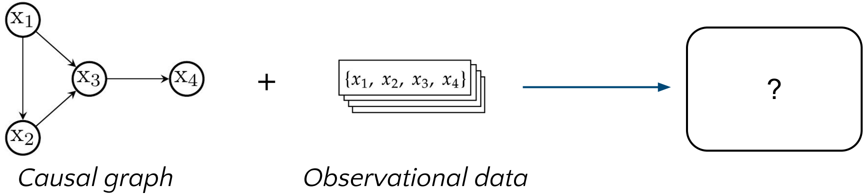
Interventional/counterfactual data

Motivation

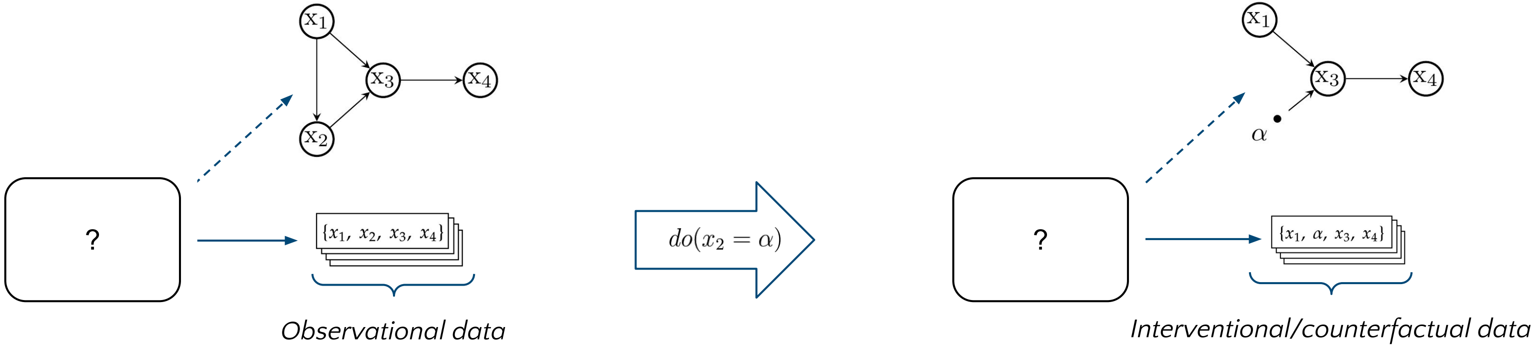

x1 = f1(u1)
x2 = f2(x1, u2)
x3 = f3(x1, x2, u3)
x4 = f4(x3, u4)
x1
x2
x3
x4
{x1, x2, x3, x4}
Observational data
do(x2 = α)
x1 = f1(u1)
x2 = α
x3 = f3(x1, x2, u3)
x4 = f4(x3, u4)
x1
α
x3
x4
{x1, α, x3, x4}
Interventional/counterfactual data

Motivation


x1
x3
x4
x2
x1 = f1(u1)
x2 = f2(x1, u2)
x3 = f3(x1, x2, u3)
x4 = f4(x3, u4)
{x1, x2, x3, x4}
Observational data
$do(x_2 = \alpha)$
x1
x3
x4
α
x1 = f1(u1)
x2 = α
x3 = f3(x1, x2, u3)
x4 = f4(x3, u4)
{x1, α, x3, x4}
Interventional/counterfactual data


x1
x3
x4
x2
Causal graph
+
{x1, x2, x3, x4}
Observational data
?

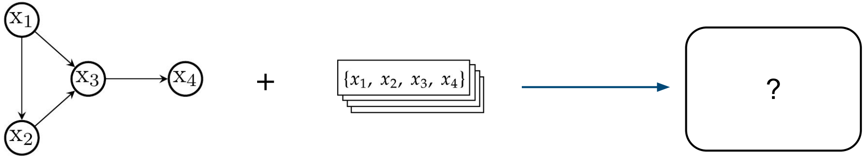
Motivation



Observational data

Interventional/counterfactual data

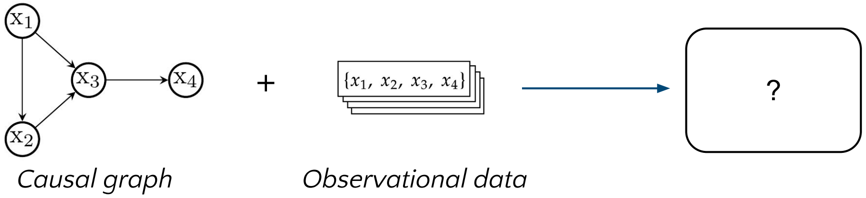
How to approach this problem…



Causal graph

Observational data

How to approach this problem…



1. Model each variable individually:
   E.g.: a linear function, spline, GP[1], NN[2], …

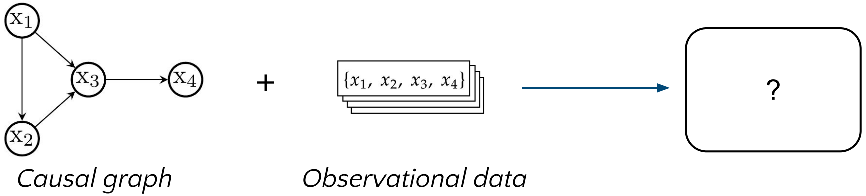
× Independent functions ✓ Straightforward

× No amortization ✓ Causally consistent

× Seq. error propagation ✓ Easy do-operator

[1] Karimi, Amir-Hossein, et al. "Algorithmic recourse under imperfect causal knowledge: a probabilistic approach." Advances in neural information processing systems 33 (2020): 265-277.
[2] Parafita, Álvaro, and Jordi Vitrià. "Estimand-Agnostic Causal Query Estimation With Deep Causal Graphs." IEEE Access 10 (2022): 71370-71386.

How to approach this problem…



1. Model each variable individually:
   E.g.: a linear function, spline, GP[1], NN[2], …

2. Model the SCM with a Deep Neural Network.
   E.g.: VACA,[1] CAREFL,[2] …

× Independent functions ✓ Straightforward

× No amortization ✓ Causally consistent

× Seq. error propagation ✓ Easy do-operator

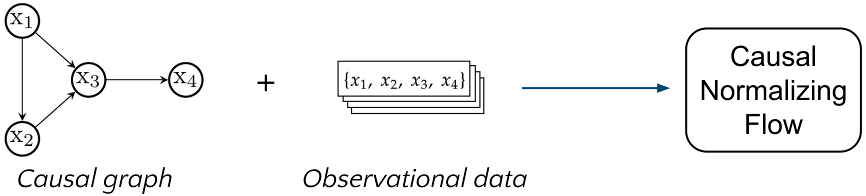
✓ Expressive × Without guarantees

✓ Parameter amortization × Complex NN training

✓ Parallel computations × Inexact do-operator

[1] Karimi, Amir-Hossein, et al. "Algorithmic recourse under imperfect causal knowledge: a probabilistic approach." Advances in neural information processing systems 33 (2020): 265-277.
[2] Parafita, Álvaro, and Jordi Vitrià. "Estimand-Agnostic Causal Query Estimation With Deep Causal Graphs." IEEE Access 10 (2022): 71370-71386.
[3] Sánchez-Martin, P., M. Rateike, and I. Valera. "VACA: Designing Variational Graph Autoencoders for Causal Queries". Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no.
[4] Khemakhem, Ilyes, et al. "Causal autoregressive flows." International conference on artificial intelligence and statistics. PMLR, 2021.

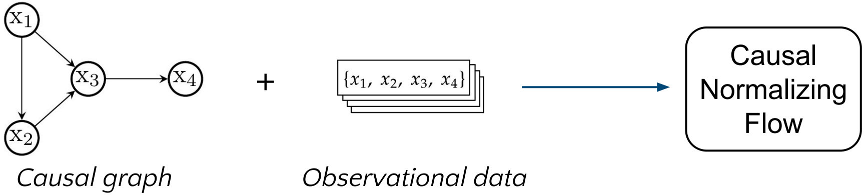
… until today



Causal Normalizing Flows:

- ✓ Straightforward
- ✓ Causally consistent
- ✓ Easy do-operator
- ✓ Expressive
- ✓ Parameter amortization
- ✓ Parallel computations

… until today

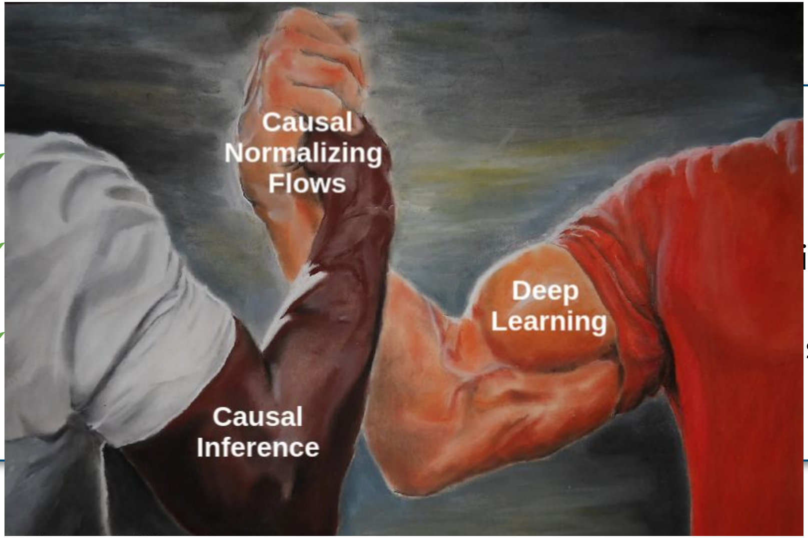


Causal Normalizing Flows:

✓

✓ ion

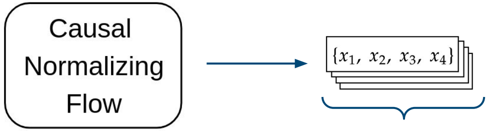
✓ s



In a nutshell

Causal Normalizing Flow

In a nutshell



Causal Normalizing Flow

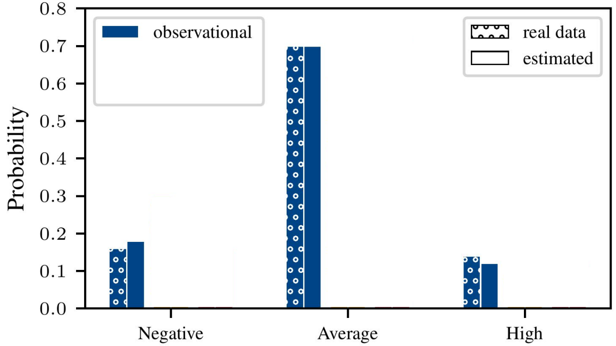
$\{x_1, x_2, x_3, x_4\}$

Observational data

Capabilities

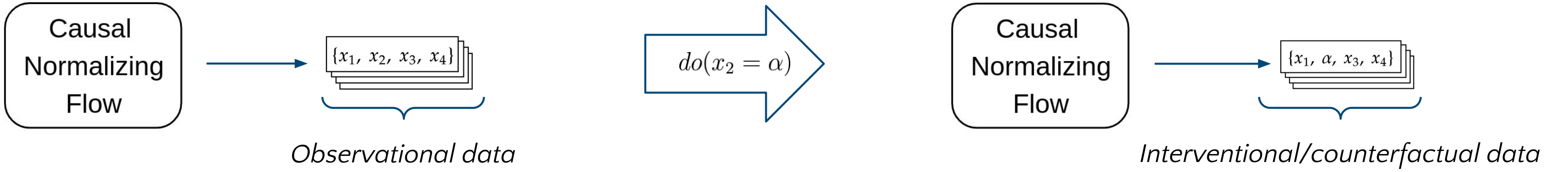
1. Generate observational data.

Objectives



German Credit - Checking account

In a nutshell



Causal Normalizing Flow → $\underbrace{\{x_1,\ x_2,\ x_3,\ x_4\}}$

Observational data

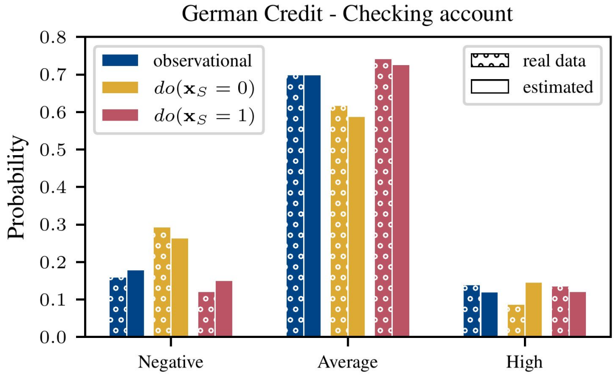
$do(x_2 = \alpha)$

Causal Normalizing Flow → $\underbrace{\{x_1,\ \alpha,\ x_3,\ x_4\}}$

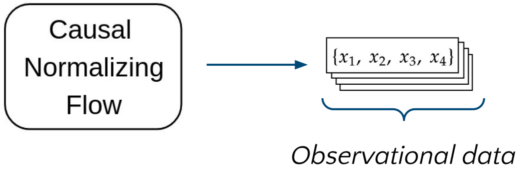
Interventional/counterfactual data

Capabilities

1. Generate observational data.
2. Generate interventional data.
3. Generate counterfactual data.



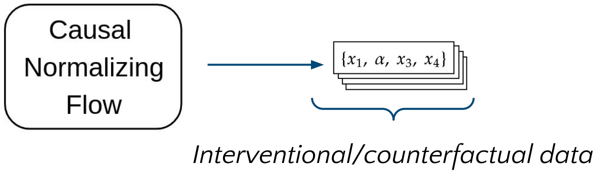
In a nutshell



Causal Normalizing Flow

$\{x_1, x_2, x_3, x_4\}$

Observational data



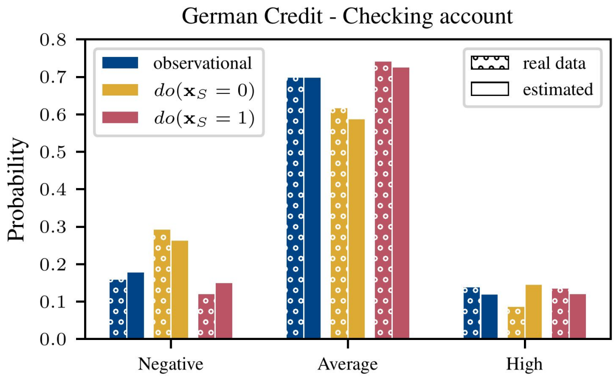
Causal Normalizing Flow

$\{x_1, \alpha, x_3, x_4\}$

Interventional/counterfactual data

Capabilities

1. Generate observational data.
2. Generate interventional data.
3. Generate counterfactual data.



In a nutshell

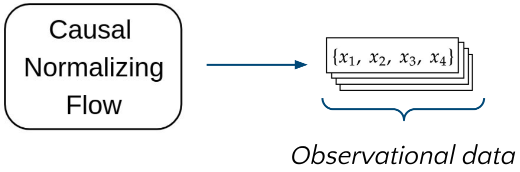
Capabilities

1. Generate observational data.
2. Generate interventional data.
3. Generate counterfactual data.

Objectives

1. Fit the observed data accurately.

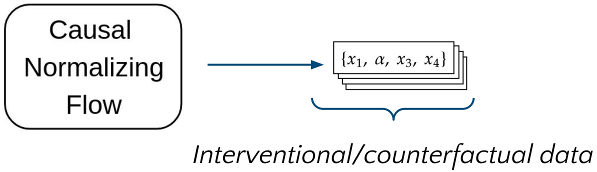
In a nutshell



Causal Normalizing Flow

$\{x_1, x_2, x_3, x_4\}$

Observational data



Causal Normalizing Flow

$\{x_1, \alpha, x_3, x_4\}$

Interventional/counterfactual data

Capabilities

1. Generate observational data.
2. Generate interventional data.
3. Generate counterfactual data.

Objectives

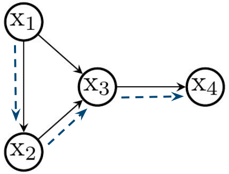
1. Fit the observed data accurately.
2. Identify the exogenous variables.
3. Ensure causal consistency wrt. the true SCM.

SCMs

$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, x_2, u_3) \\ x_4 = \tilde{f}_4(x_3, u_4) \end{cases}$$

Structural equations

induces →



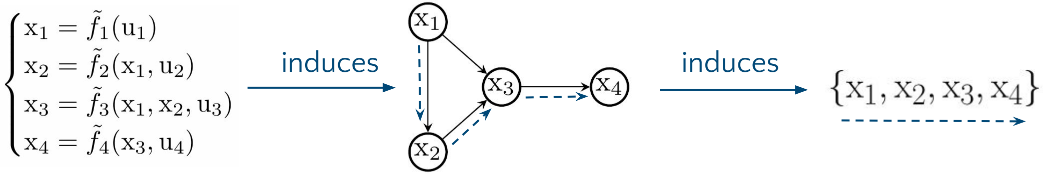
Causal graph

induces →

$$\{x_1, x_2, x_3, x_4\}$$

Causal ordering

SCMs



$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, x_2, u_3) \\ x_4 = \tilde{f}_4(x_3, u_4) \end{cases}$$

induces

induces

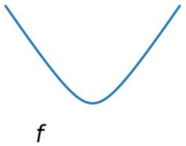
$$\{x_1, x_2, x_3, x_4\}$$

Structural equations

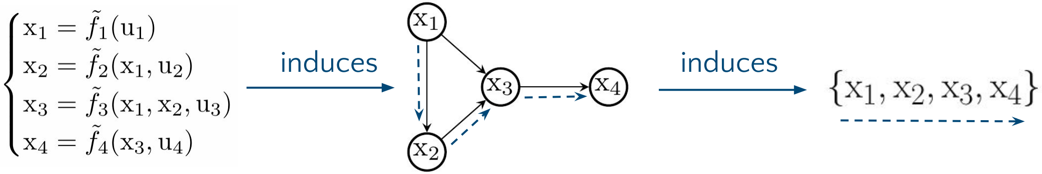
Causal graph

Causal ordering

Invertible & differentiable generators



SCMs



$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, x_2, u_3) \\ x_4 = \tilde{f}_4(x_3, u_4) \end{cases}$$

induces

induces

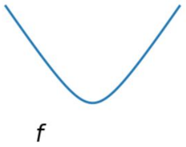
$$\{x_1, x_2, x_3, x_4\}$$

Structural equations

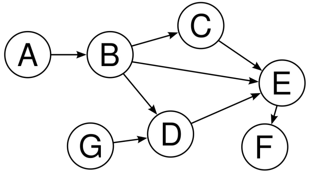
Causal graph

Causal ordering

Invertible & differentiable generators



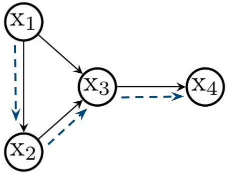
No feedback loops



SCMs

$$\begin{cases} \mathrm{x}_1 = \tilde{f}_1(\mathrm{u}_1) \\ \mathrm{x}_2 = \tilde{f}_2(\mathrm{x}_1, \mathrm{u}_2) \\ \mathrm{x}_3 = \tilde{f}_3(\mathrm{x}_1, \mathrm{x}_2, \mathrm{u}_3) \\ \mathrm{x}_4 = \tilde{f}_4(\mathrm{x}_3, \mathrm{u}_4) \end{cases}$$

induces



induces

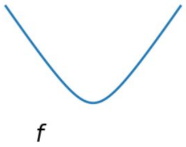
$$\underset{\dashrightarrow}{\{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \mathrm{x}_4\}}$$

Structural equations

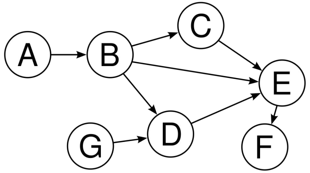
Causal graph

Causal ordering

Invertible & differentiable generators



No feedback loops



Causal sufficiency

$$p(\mathbf{u}) = \prod_i p(\mathbf{u}_i)$$

ANFs and SCMs under the same umbrella

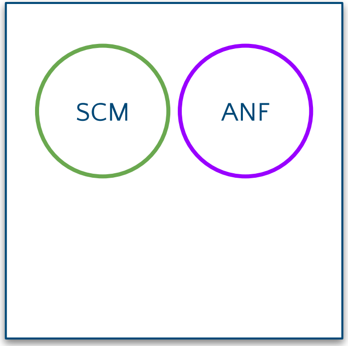
SCM→Structural Causal Model

1. Fit the observed data accurately.

ANFs and SCMs under the same umbrella

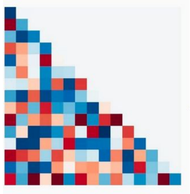
SCM→Structural Causal Model
ANF→Aut. Normalizing Flow

1. Fit the observed data accurately.



ANFs:

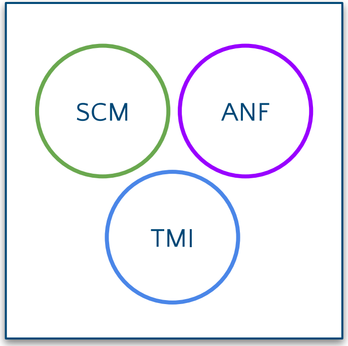
- Invertible differentiable neural networks.
- Transform random variables, $T_{\boldsymbol{\theta}}(\mathbf{x}) =: \mathbf{u} \sim P_{\mathbf{u}}$.
- Autoregressive and monotonic.



ANFs and SCMs under the same umbrella

1. Fit the observed data accurately.

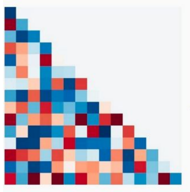
SCM→Structural Causal Model
ANF→Aut. Normalizing Flow
TMI→Triangular Monotonic Incr. Map



Triangular Monotonic Increasing (TMI) maps.

$$f(x) = \begin{bmatrix} f_1(x_1) \\ f_2(x_1, x_2) \\ \vdots \\ f_d(x_1, \ldots, x_d) \end{bmatrix}$$

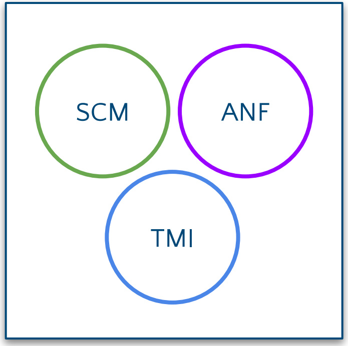
$$\partial_{\mathrm{x}_i} f_i(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_i) \geq 0$$



ANFs and SCMs under the same umbrella

SCM→Structural Causal Model
ANF→Aut. Normalizing Flow
TMI→Triangular Monotonic Incr. Map

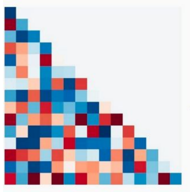
1. Fit the observed data accurately.



Triangular Monotonic Increasing (TMI) maps.

$$f(x) = \begin{bmatrix} f_1(x_1) \\ f_2(x_1, x_2) \\ \vdots \\ f_d(x_1, \ldots, x_d) \end{bmatrix}$$

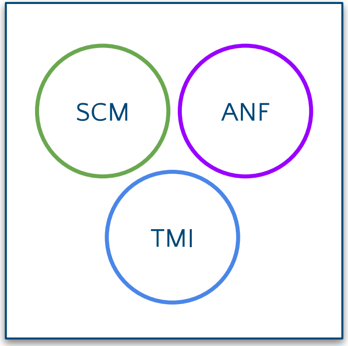
$$\partial_{\mathrm{x}_i} f_i(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_i) \geq 0$$



ANFs and SCMs under the same umbrella

1. Fit the observed data accurately.

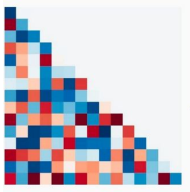
SCM→Structural Causal Model
ANF→Aut. Normalizing Flow
TMI→Triangular Monotonic Incr. Map



Triangular Monotonic Increasing (TMI) maps.

$$f(x) = \begin{bmatrix} f_1(x_1) \\ f_2(x_1, x_2) \\ \vdots \\ f_d(x_1, \dots, x_d) \end{bmatrix}$$

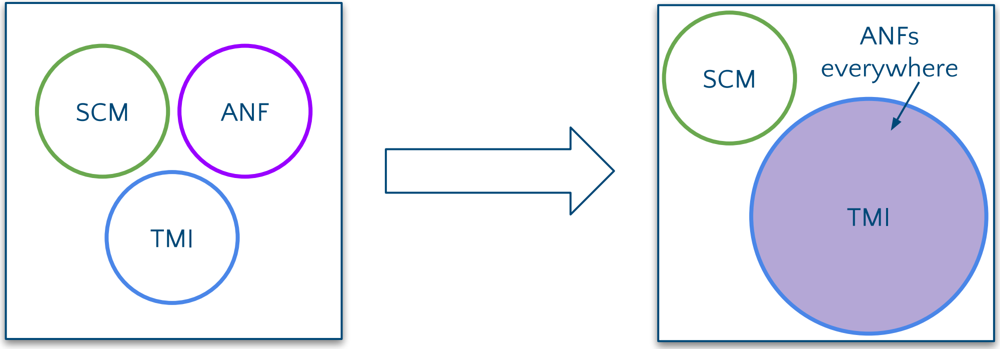
$$\partial_{\mathrm{x}_i} f_i(\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_i) \geq 0$$



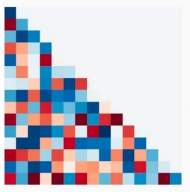
ANFs and SCMs under the same umbrella

1. Fit the observed data <u>accurately</u>.

SCM→Structural Causal Model
ANF→Aut. Normalizing Flow
TMI→Triangular Monotonic Incr. Map



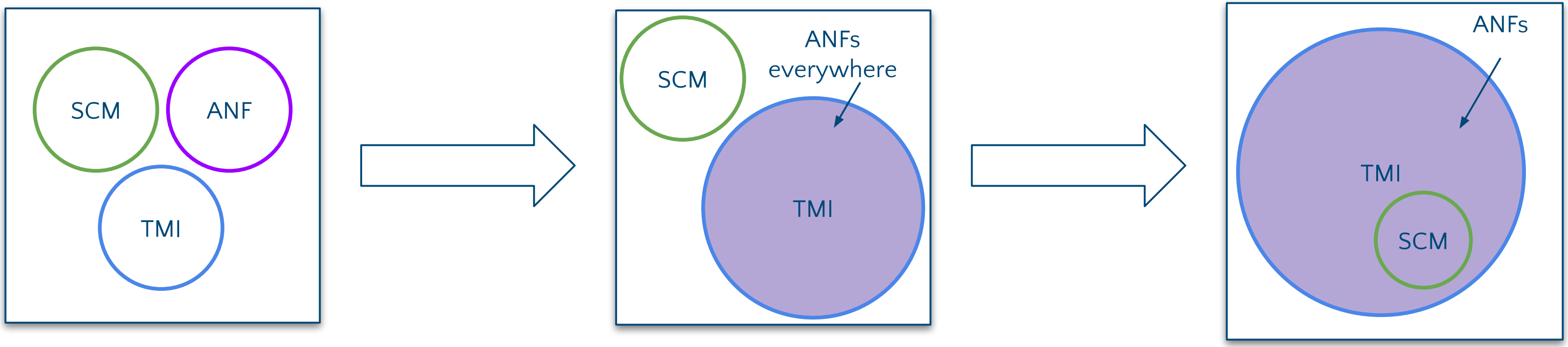
ANFs are TMI maps
and
universal approximators of any other TMI map.



ANFs and SCMs under the same umbrella

✓ 1. Fit the observed data <u>accurately</u>.

SCM→Structural Causal Model
ANF→Aut. Normalizing Flow
TMI→Triangular Monotonic Incr. Map



Structural equations
can be always
unrolled & monotonized

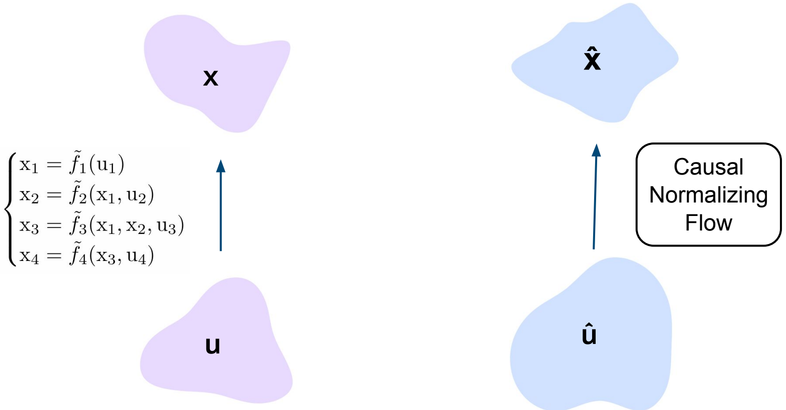
$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) = \tilde{f}_2(\tilde{f}_1(u_1), u_2) \\ x_3 = \tilde{f}_3(x_1, x_2, u_3) = \tilde{f}_3(\tilde{f}_1(u_1), \tilde{f}_2(\tilde{f}_1(u_1), u_2), u_3) \\ x_4 = \tilde{f}_4(x_3, u_4) = \tilde{f}_4(\tilde{f}_3(\tilde{f}_1(u_1), \tilde{f}_2(\tilde{f}_1(u_1), u_2), u_3), u_4) \end{cases}$$

Isolating the exogenous variables

2. Identify the exogenous variables.

$\mathcal{F} \times \mathcal{P}_{\mathbf{u}}$ – Family of TMI maps with fully-factorized distributions.

Theorem 1 (Identifiability)**.** If two elements of the family $\mathcal{F} \times \mathcal{P}_{\mathbf{u}}$ (as defined above) produce the same observational distribution, then the two data-generating processes differ by an invertible, component-wise transformation of the variables $\mathbf{u}$.



$$\begin{cases} \mathrm{x}_1 = \tilde{f}_1(\mathrm{u}_1) \\ \mathrm{x}_2 = \tilde{f}_2(\mathrm{x}_1, \mathrm{u}_2) \\ \mathrm{x}_3 = \tilde{f}_3(\mathrm{x}_1, \mathrm{x}_2, \mathrm{u}_3) \\ \mathrm{x}_4 = \tilde{f}_4(\mathrm{x}_3, \mathrm{u}_4) \end{cases}$$

Causal Normalizing Flow

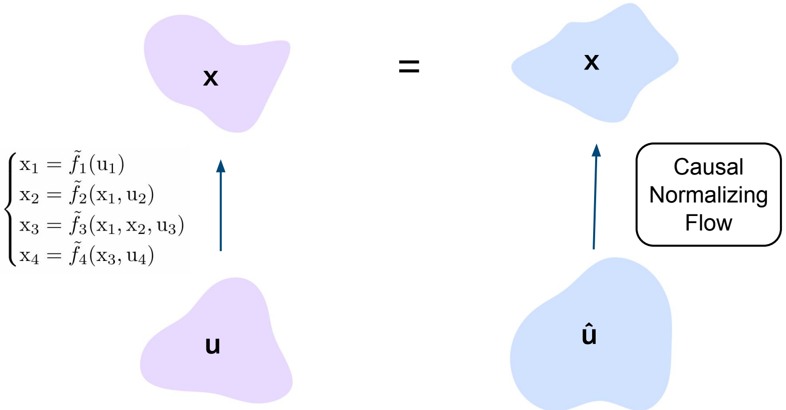
[1] Xi, Quanhan, and Benjamin Bloem-Reddy. "Indeterminacy in generative models: Characterization and strong identifiability." International Conference on Artificial Intelligence and Statistics. PMLR, 2023.

Isolating the exogenous variables

2. Identify the exogenous variables.

$\mathcal{F} \times \mathcal{P}_{\mathbf{u}}$ – Family of TMI maps with fully-factorized distributions.

Theorem 1 (Identifiability)**.** If two elements of the family $\mathcal{F} \times \mathcal{P}_{\mathbf{u}}$ (as defined above) produce the same observational distribution, then the two data-generating processes differ by an invertible, component-wise transformation of the variables $\mathbf{u}$.



$$\begin{cases} \mathrm{x}_1 = \tilde{f}_1(\mathrm{u}_1) \\ \mathrm{x}_2 = \tilde{f}_2(\mathrm{x}_1, \mathrm{u}_2) \\ \mathrm{x}_3 = \tilde{f}_3(\mathrm{x}_1, \mathrm{x}_2, \mathrm{u}_3) \\ \mathrm{x}_4 = \tilde{f}_4(\mathrm{x}_3, \mathrm{u}_4) \end{cases}$$

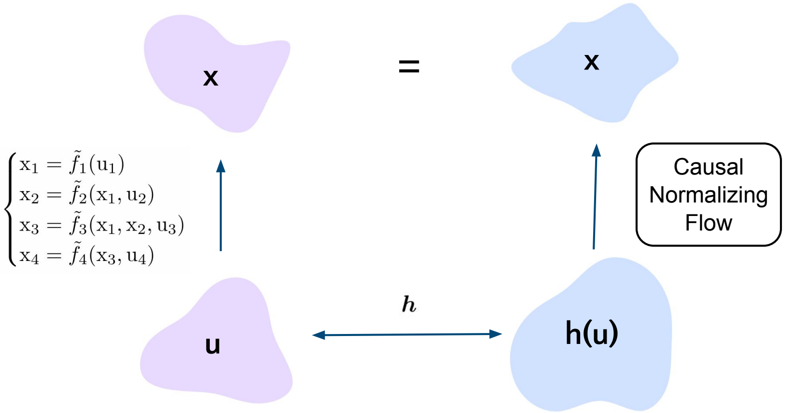
[1] Xi, Quanhan, and Benjamin Bloem-Reddy. "Indeterminacy in generative models: Characterization and strong identifiability." International Conference on Artificial Intelligence and Statistics. PMLR, 2023.

Isolating the exogenous variables

✓ 2. Identify the exogenous variables.

$\mathcal{F} \times \mathcal{P}_{\mathbf{u}}$ – Family of TMI maps with fully-factorized distributions.

Theorem 1 (Identifiability)**.** If two elements of the family $\mathcal{F} \times \mathcal{P}_{\mathbf{u}}$ (as defined above) produce the same observational distribution, then the two data-generating processes differ by an invertible, component-wise transformation of the variables $\mathbf{u}$.



$$\begin{cases} \mathrm{x}_1 = \tilde{f}_1(\mathrm{u}_1) \\ \mathrm{x}_2 = \tilde{f}_2(\mathrm{x}_1, \mathrm{u}_2) \\ \mathrm{x}_3 = \tilde{f}_3(\mathrm{x}_1, \mathrm{x}_2, \mathrm{u}_3) \\ \mathrm{x}_4 = \tilde{f}_4(\mathrm{x}_3, \mathrm{u}_4) \end{cases}$$

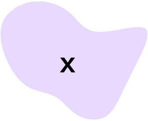
$$\boldsymbol{h}(\mathbf{u}) = (h_1(\mathrm{u}_1), h_2(\mathrm{u}_2), \ldots, h_d(\mathrm{u}_d))$$

[1] Xi, Quanhan, and Benjamin Bloem-Reddy. "Indeterminacy in generative models: Characterization and strong identifiability." International Conference on Artificial Intelligence and Statistics. PMLR, 2023.

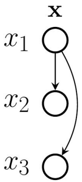
Causal consistency

3. Ensure causal consistency wrt. the true SCM.

$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, u_3) \end{cases}$$



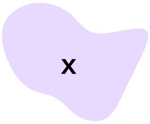
Recursive



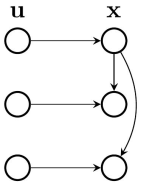
Causal consistency

3. Ensure causal consistency wrt. the true SCM.

$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, u_3) \end{cases}$$



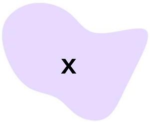
Recursive

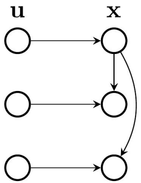


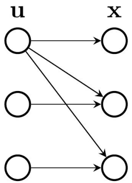
Causal consistency

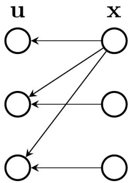
3. Ensure causal consistency wrt. the true SCM.

$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, u_3) \end{cases}$$




Recursive

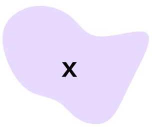

Generative


Abductive

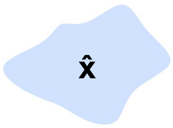
Causal consistency

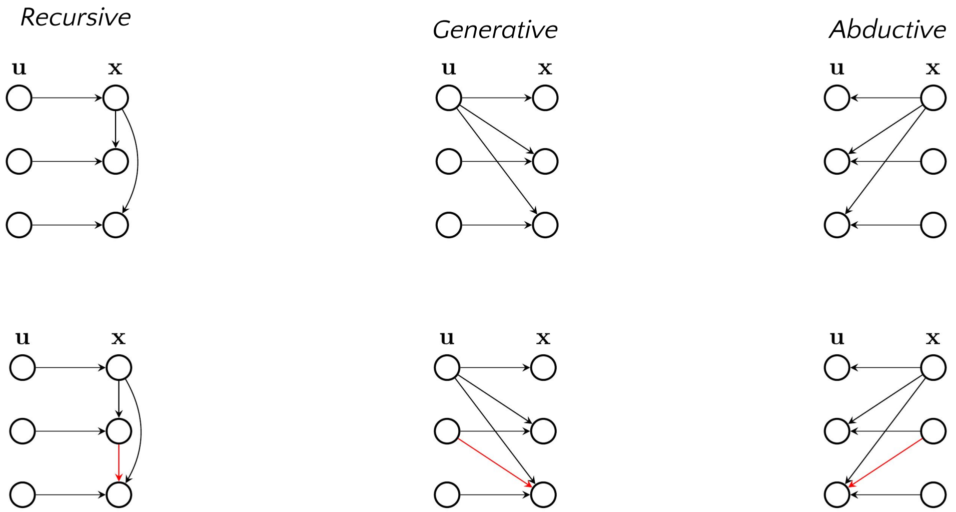
3. Ensure causal consistency wrt. the true SCM.

$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, u_3) \end{cases}$$



Causal Normalizing Flow





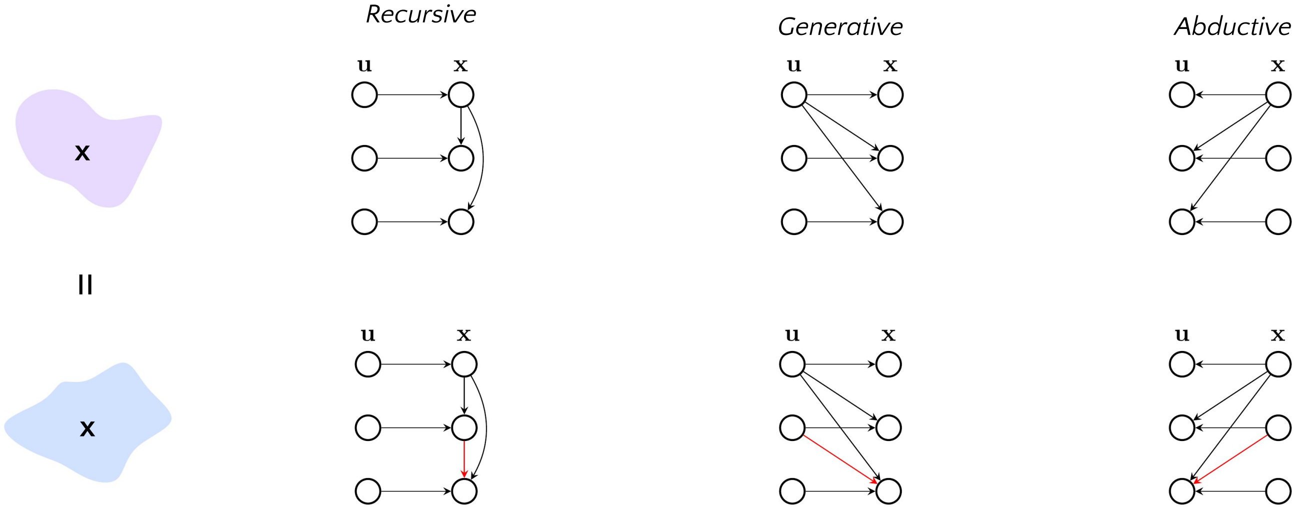
Causal consistency

3. Ensure causal consistency wrt. the true SCM.

Corollary 2 (Causal consistency)**.** If a causal NF $T_{\boldsymbol{\theta}}$ isolates the exogenous variables of an SCM $\mathcal{M}$, then $\nabla_{\mathbf{x}} T_{\boldsymbol{\theta}}(\mathbf{x}) \equiv \boldsymbol{I} - \boldsymbol{A}$ and $\nabla_{\mathbf{u}} T_{\boldsymbol{\theta}}^{-1}(\mathbf{u}) \equiv \boldsymbol{I} + \sum_{n=1}^{\operatorname{diam}(\boldsymbol{A})} \boldsymbol{A}^n$, where $\boldsymbol{A}$ is the causal adjacency matrix of $\mathcal{M}$. In other words, $T_{\boldsymbol{\theta}}$ is causally consistent with the true data-generating process, $\mathcal{M}$.

$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, u_3) \end{cases}$$

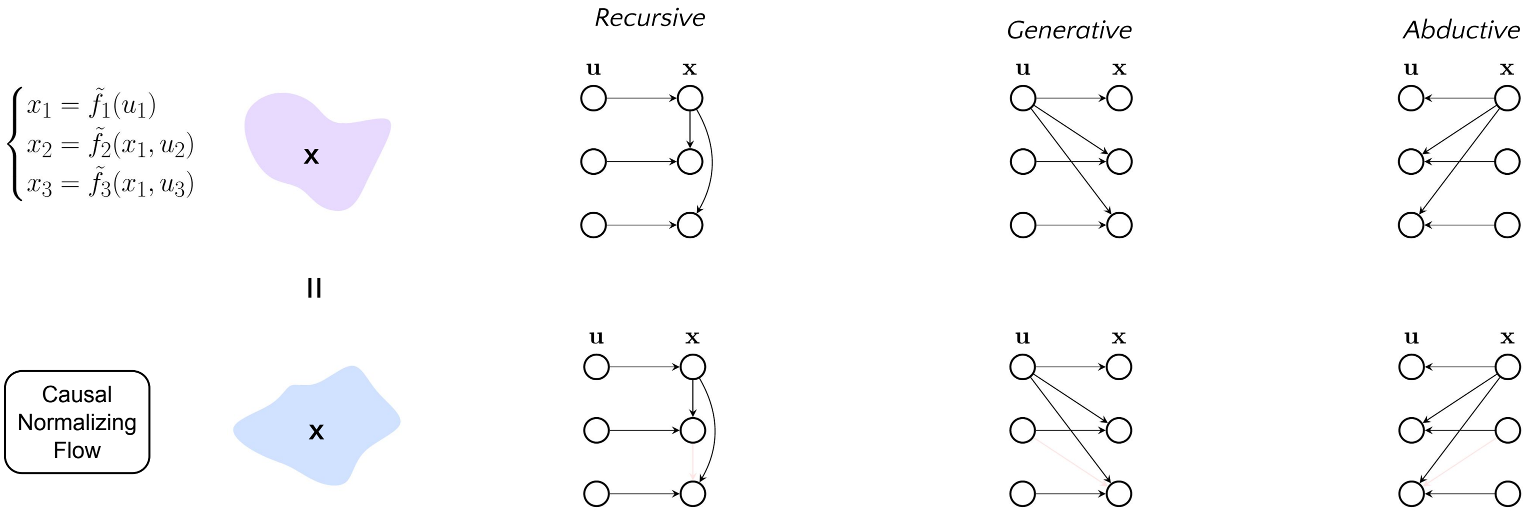
Causal Normalizing Flow



Causal consistency

3. Ensure causal consistency wrt. the true SCM.

Corollary 2 (Causal consistency)**.** If a causal NF $T_{\boldsymbol{\theta}}$ isolates the exogenous variables of an SCM $\mathcal{M}$, then $\nabla_{\mathbf{x}} T_{\boldsymbol{\theta}}(\mathbf{x}) \equiv \boldsymbol{I} - \boldsymbol{A}$ and $\nabla_{\mathbf{u}} T_{\boldsymbol{\theta}}^{-1}(\mathbf{u}) \equiv \boldsymbol{I} + \sum_{n=1}^{\operatorname{diam}(\boldsymbol{A})} \boldsymbol{A}^n$, where $\boldsymbol{A}$ is the causal adjacency matrix of $\mathcal{M}$. In other words, $T_{\boldsymbol{\theta}}$ is causally consistent with the true data-generating process, $\mathcal{M}$.



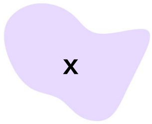
$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, u_3) \end{cases}$$

Causal consistency

3. Ensure causal consistency wrt. the true SCM.

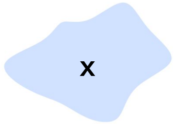
Corollary 2 (Causal consistency)**.** If a causal NF $T_{\boldsymbol{\theta}}$ isolates the exogenous variables of an SCM $\mathcal{M}$, then $\nabla_{\mathbf{x}} T_{\boldsymbol{\theta}}(\mathbf{x}) \equiv \boldsymbol{I} - \boldsymbol{A}$ and $\nabla_{\mathbf{u}} T_{\boldsymbol{\theta}}^{-1}(\mathbf{u}) \equiv \boldsymbol{I} + \sum_{n=1}^{\operatorname{diam}(\boldsymbol{A})} \boldsymbol{A}^n$, where $\boldsymbol{A}$ is the causal adjacency matrix of $\mathcal{M}$. In other words, $T_{\boldsymbol{\theta}}$ is causally consistent with the true data-generating process, $\mathcal{M}$.

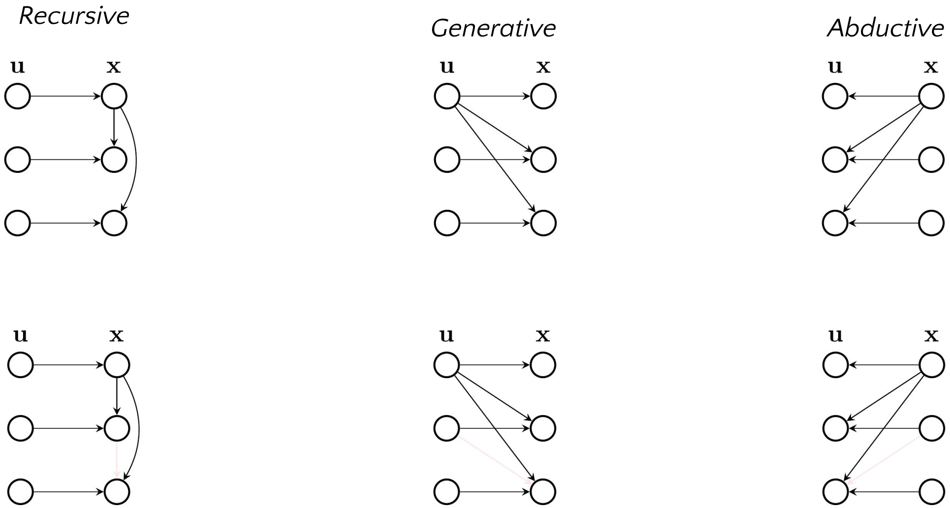
$$\begin{cases} x_1 = \tilde{f}_1(u_1) \\ x_2 = \tilde{f}_2(x_1, u_2) \\ x_3 = \tilde{f}_3(x_1, u_3) \end{cases}$$



II

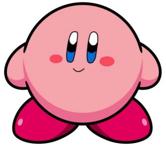
Causal Normalizing Flow

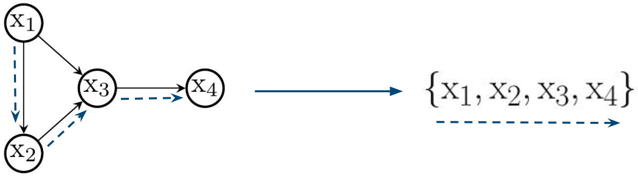




Theory vs. practice

In theory…
ANF + causal ordering is enough.





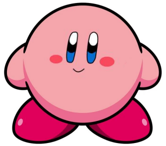
$\{x_1, x_2, x_3, x_4\}$

Causal graph

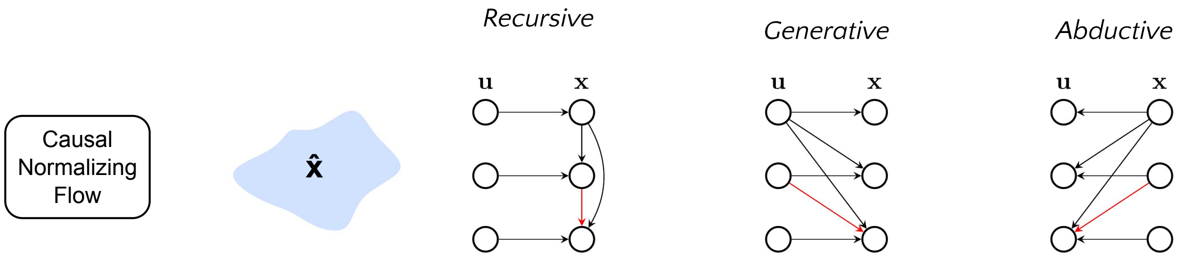
Causal ordering

Theory vs. practice

In theory...
ANF + causal ordering is enough.

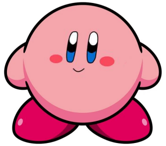


... but in practice ...
Neural networks ❤ local optima.



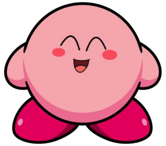
Theory vs. practice

In theory...
ANF + causal ordering is enough.



... but in practice ...
Neural networks ❤ local optima.

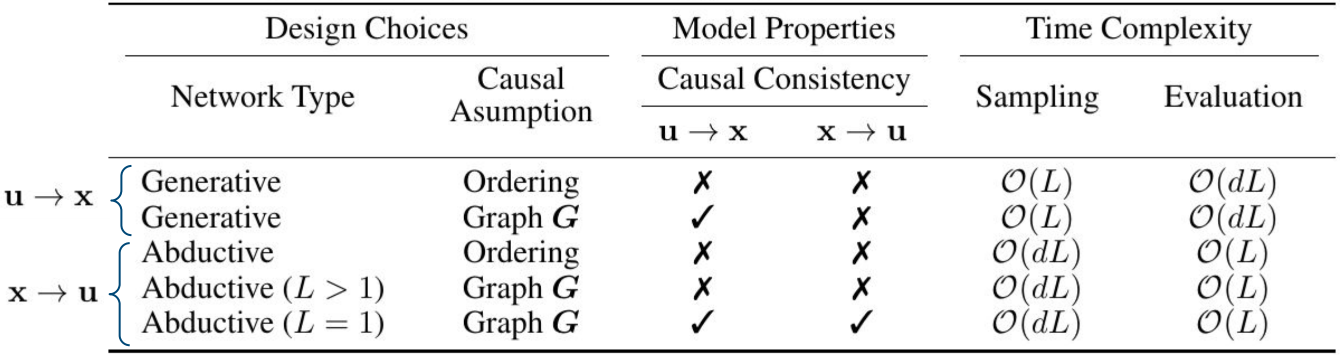
Wait!
With **G** we can design a causally consistent network!



Network design

3. Ensure causal consistency wrt. the true SCM.

Flow direction

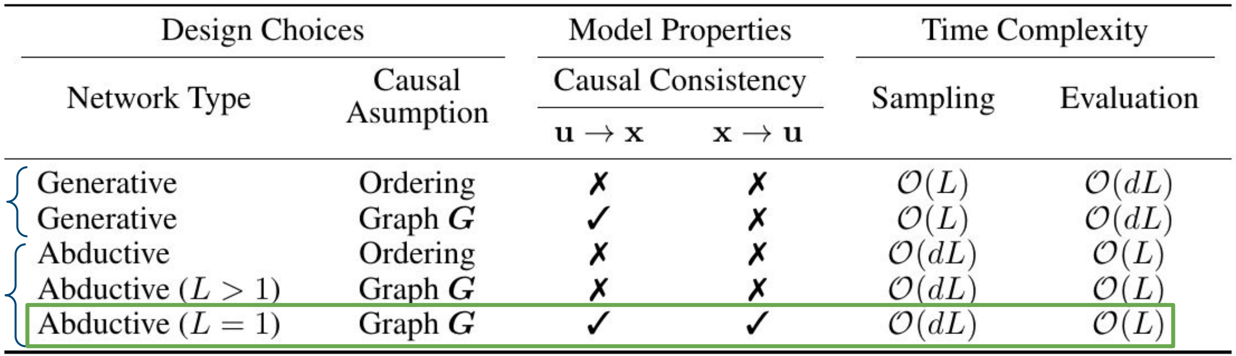


	Design Choices		Model Properties		Time Complexity	
	Network Type	Causal Asumption	Causal Consistency		Sampling	Evaluation
			$\mathbf{u} \to \mathbf{x}$	$\mathbf{x} \to \mathbf{u}$		
$\mathbf{u} \to \mathbf{x}$	Generative	Ordering	✗	✗	$\mathcal{O}(L)$	$\mathcal{O}(dL)$
	Generative	Graph $\boldsymbol{G}$	✓	✗	$\mathcal{O}(L)$	$\mathcal{O}(dL)$
$\mathbf{x} \to \mathbf{u}$	Abductive	Ordering	✗	✗	$\mathcal{O}(dL)$	$\mathcal{O}(L)$
	Abductive ($L > 1$)	Graph $\boldsymbol{G}$	✗	✗	$\mathcal{O}(dL)$	$\mathcal{O}(L)$
	Abductive ($L = 1$)	Graph $\boldsymbol{G}$	✓	✓	$\mathcal{O}(dL)$	$\mathcal{O}(L)$

Network design

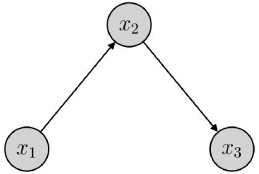
✓ 3. Ensure causal consistency wrt. the true SCM.

Flow direction

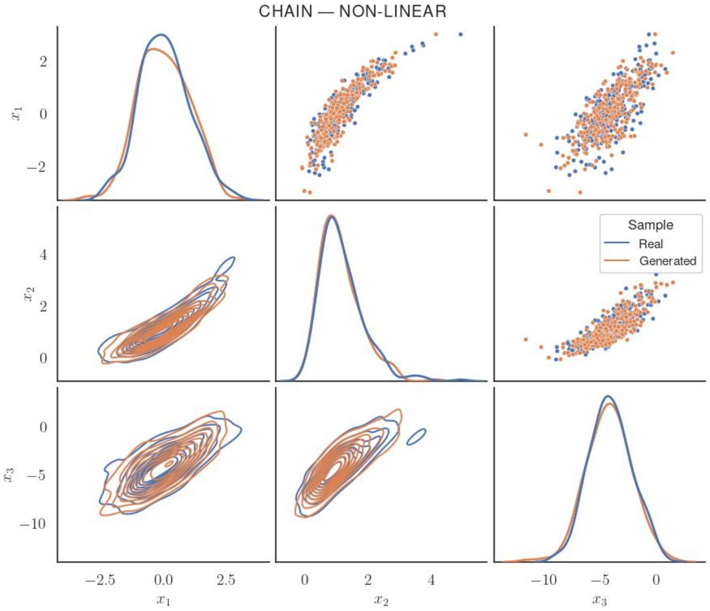


	Design Choices		Model Properties		Time Complexity	
	Network Type	Causal Asumption	Causal Consistency		Sampling	Evaluation
			$\mathbf{u} \to \mathbf{x}$	$\mathbf{x} \to \mathbf{u}$		
$\mathbf{u} \to \mathbf{x}$	Generative	Ordering	✗	✗	$\mathcal{O}(L)$	$\mathcal{O}(dL)$
	Generative	Graph $\boldsymbol{G}$	✓	✗	$\mathcal{O}(L)$	$\mathcal{O}(dL)$
$\mathbf{x} \to \mathbf{u}$	Abductive	Ordering	✗	✗	$\mathcal{O}(dL)$	$\mathcal{O}(L)$
	Abductive ($L > 1$)	Graph $\boldsymbol{G}$	✗	✗	$\mathcal{O}(dL)$	$\mathcal{O}(L)$
	Abductive ($L = 1$)	Graph $\boldsymbol{G}$	✓	✓	$\mathcal{O}(dL)$	$\mathcal{O}(L)$

Qualitative results

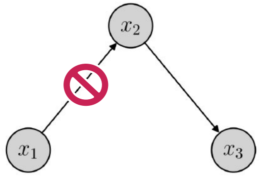


(a) 3-CHAIN

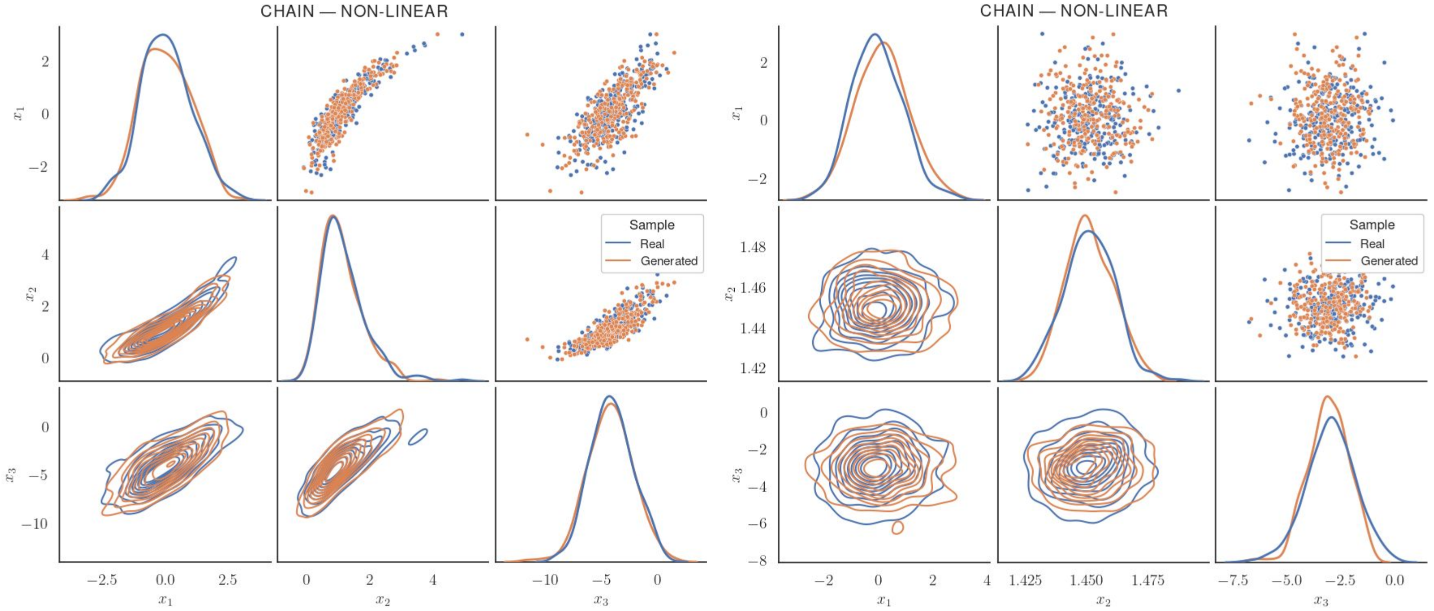


(a) Observational distribution.

Qualitative results



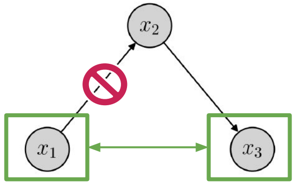
(a) 3-CHAIN



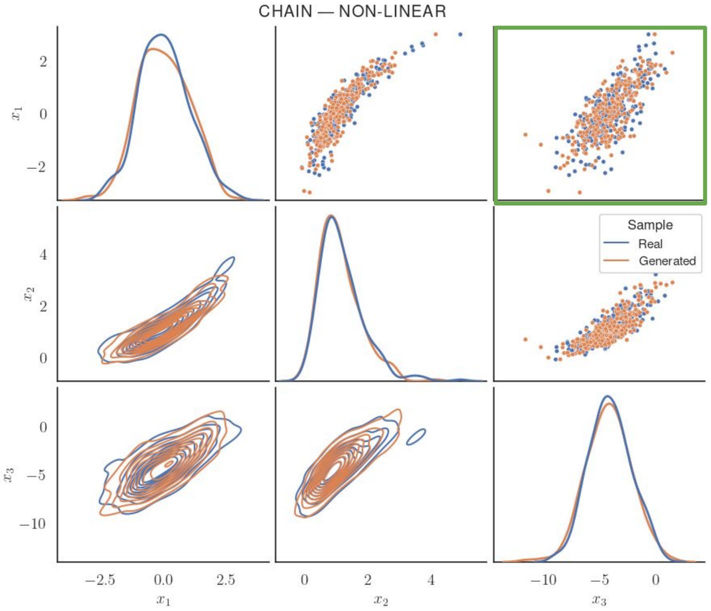
(a) Observational distribution.

(b) Interventional distribution

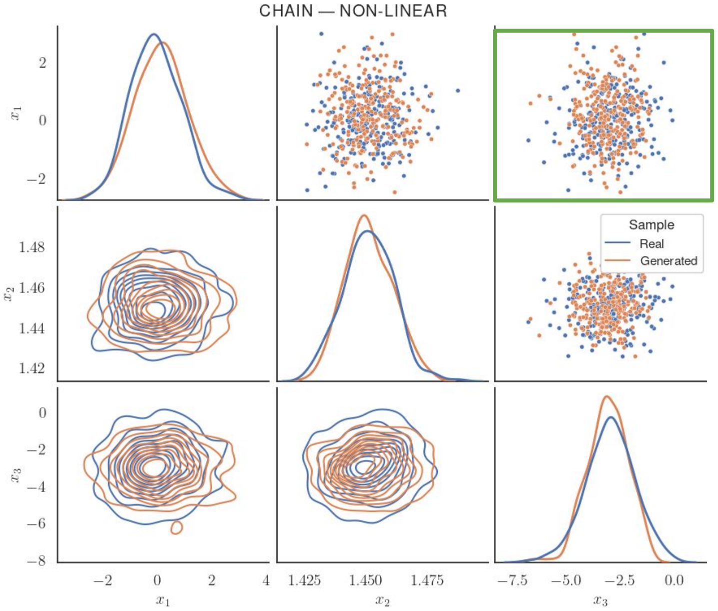
Qualitative results



(a) 3-CHAIN



(a) Observational distribution.



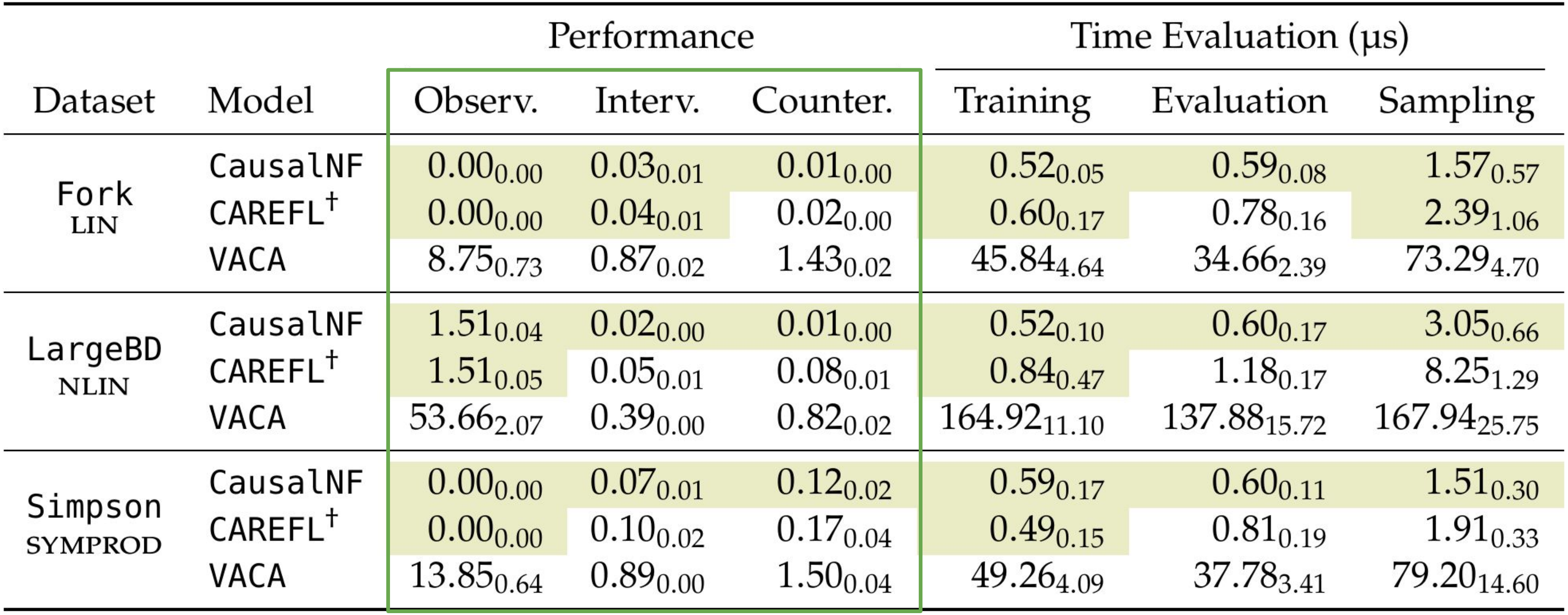
(b) Interventional distribution

Quantitative results

		Performance			Time Evaluation (µs)		
Dataset	Model	Observ.	Interv.	Counter.	Training	Evaluation	Sampling
Fork LIN	CausalNF	$0.00_{0.00}$	$0.03_{0.01}$	$0.01_{0.00}$	$0.52_{0.05}$	$0.59_{0.08}$	$1.57_{0.57}$
	CAREFL†	$0.00_{0.00}$	$0.04_{0.01}$	$0.02_{0.00}$	$0.60_{0.17}$	$0.78_{0.16}$	$2.39_{1.06}$
	VACA	$8.75_{0.73}$	$0.87_{0.02}$	$1.43_{0.02}$	$45.84_{4.64}$	$34.66_{2.39}$	$73.29_{4.70}$
LargeBD NLIN	CausalNF	$1.51_{0.04}$	$0.02_{0.00}$	$0.01_{0.00}$	$0.52_{0.10}$	$0.60_{0.17}$	$3.05_{0.66}$
	CAREFL†	$1.51_{0.05}$	$0.05_{0.01}$	$0.08_{0.01}$	$0.84_{0.47}$	$1.18_{0.17}$	$8.25_{1.29}$
	VACA	$53.66_{2.07}$	$0.39_{0.00}$	$0.82_{0.02}$	$164.92_{11.10}$	$137.88_{15.72}$	$167.94_{25.75}$
Simpson SYMPROD	CausalNF	$0.00_{0.00}$	$0.07_{0.01}$	$0.12_{0.02}$	$0.59_{0.17}$	$0.60_{0.11}$	$1.51_{0.30}$
	CAREFL†	$0.00_{0.00}$	$0.10_{0.02}$	$0.17_{0.04}$	$0.49_{0.15}$	$0.81_{0.19}$	$1.91_{0.33}$
	VACA	$13.85_{0.64}$	$0.89_{0.00}$	$1.50_{0.04}$	$49.26_{4.09}$	$37.78_{3.41}$	$79.20_{14.60}$

12 datasets in the paper!

Quantitative results



		Performance			Time Evaluation (µs)		
Dataset	Model	Observ.	Interv.	Counter.	Training	Evaluation	Sampling
`Fork` LIN	`CausalNF`	$0.00_{0.00}$	$0.03_{0.01}$	$0.01_{0.00}$	$0.52_{0.05}$	$0.59_{0.08}$	$1.57_{0.57}$
	`CAREFL`†	$0.00_{0.00}$	$0.04_{0.01}$	$0.02_{0.00}$	$0.60_{0.17}$	$0.78_{0.16}$	$2.39_{1.06}$
	`VACA`	$8.75_{0.73}$	$0.87_{0.02}$	$1.43_{0.02}$	$45.84_{4.64}$	$34.66_{2.39}$	$73.29_{4.70}$
`LargeBD` NLIN	`CausalNF`	$1.51_{0.04}$	$0.02_{0.00}$	$0.01_{0.00}$	$0.52_{0.10}$	$0.60_{0.17}$	$3.05_{0.66}$
	`CAREFL`†	$1.51_{0.05}$	$0.05_{0.01}$	$0.08_{0.01}$	$0.84_{0.47}$	$1.18_{0.17}$	$8.25_{1.29}$
	`VACA`	$53.66_{2.07}$	$0.39_{0.00}$	$0.82_{0.02}$	$164.92_{11.10}$	$137.88_{15.72}$	$167.94_{25.75}$
`Simpson` SYMPROD	`CausalNF`	$0.00_{0.00}$	$0.07_{0.01}$	$0.12_{0.02}$	$0.59_{0.17}$	$0.60_{0.11}$	$1.51_{0.30}$
	`CAREFL`†	$0.00_{0.00}$	$0.10_{0.02}$	$0.17_{0.04}$	$0.49_{0.15}$	$0.81_{0.19}$	$1.91_{0.33}$
	`VACA`	$13.85_{0.64}$	$0.89_{0.00}$	$1.50_{0.04}$	$49.26_{4.09}$	$37.78_{3.41}$	$79.20_{14.60}$

12 datasets in the paper!

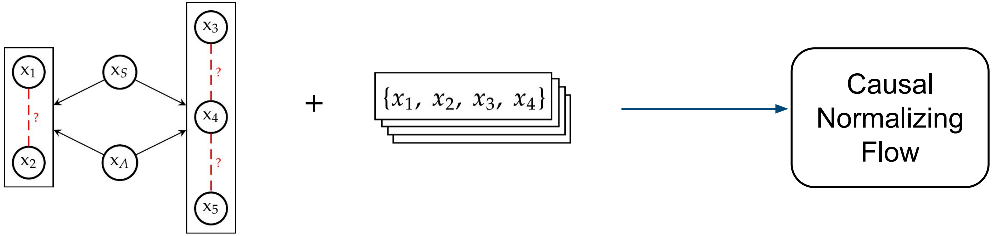
Quantitative results

		Performance			Time Evaluation (µs)		
Dataset	Model	Observ.	Interv.	Counter.	Training	Evaluation	Sampling
`Fork` LIN	`CausalNF`	$0.00_{0.00}$	$0.03_{0.01}$	$0.01_{0.00}$	$0.52_{0.05}$	$0.59_{0.08}$	$1.57_{0.57}$
	`CAREFL`†	$0.00_{0.00}$	$0.04_{0.01}$	$0.02_{0.00}$	$0.60_{0.17}$	$0.78_{0.16}$	$2.39_{1.06}$
	`VACA`	$8.75_{0.73}$	$0.87_{0.02}$	$1.43_{0.02}$	$45.84_{4.64}$	$34.66_{2.39}$	$73.29_{4.70}$
`LargeBD` NLIN	`CausalNF`	$1.51_{0.04}$	$0.02_{0.00}$	$0.01_{0.00}$	$0.52_{0.10}$	$0.60_{0.17}$	$3.05_{0.66}$
	`CAREFL`†	$1.51_{0.05}$	$0.05_{0.01}$	$0.08_{0.01}$	$0.84_{0.47}$	$1.18_{0.17}$	$8.25_{1.29}$
	`VACA`	$53.66_{2.07}$	$0.39_{0.00}$	$0.82_{0.02}$	$164.92_{11.10}$	$137.88_{15.72}$	$167.94_{25.75}$
`Simpson` SYMPROD	`CausalNF`	$0.00_{0.00}$	$0.07_{0.01}$	$0.12_{0.02}$	$0.59_{0.17}$	$0.60_{0.11}$	$1.51_{0.30}$
	`CAREFL`†	$0.00_{0.00}$	$0.10_{0.02}$	$0.17_{0.04}$	$0.49_{0.15}$	$0.81_{0.19}$	$1.91_{0.33}$
	`VACA`	$13.85_{0.64}$	$0.89_{0.00}$	$1.50_{0.04}$	$49.26_{4.09}$	$37.78_{3.41}$	$79.20_{14.60}$

12 datasets in the paper!

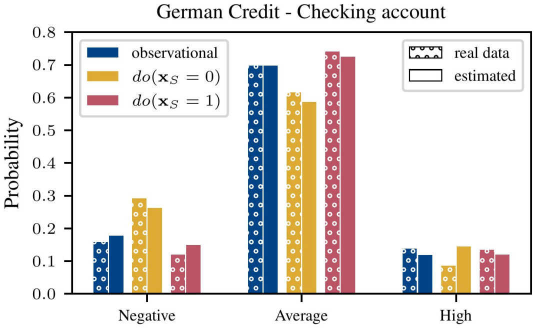
Use-case: fairness auditing and classification

German Credit

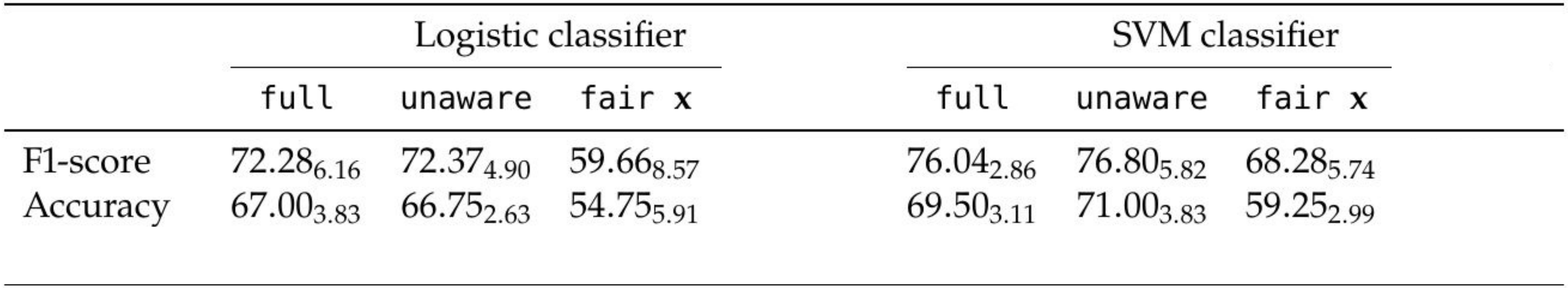


Partial
Causal graph

Mixed-typed
Observational data



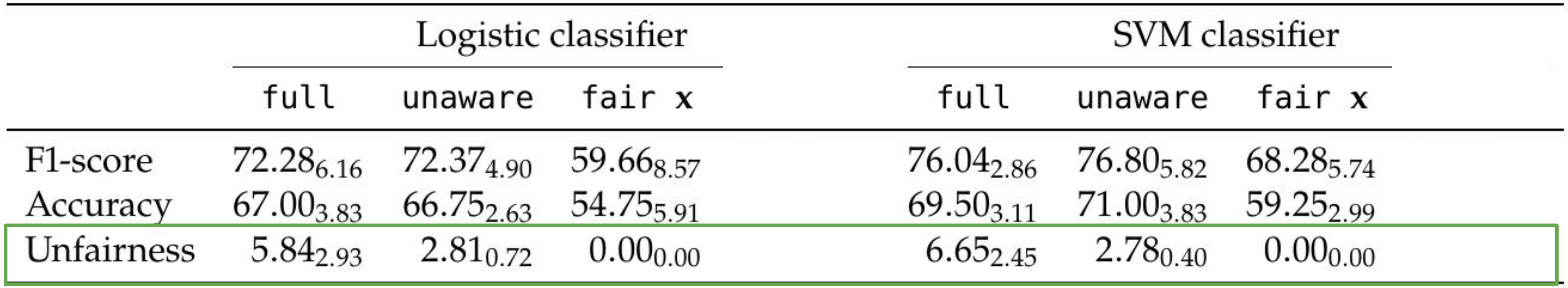
Use-case: fairness auditing and classification



	Logistic classifier			SVM classifier		
	`full`	`unaware`	`fair x`	`full`	`unaware`	`fair x`
F1-score	$72.28_{6.16}$	$72.37_{4.90}$	$59.66_{8.57}$	$76.04_{2.86}$	$76.80_{5.82}$	$68.28_{5.74}$
Accuracy	$67.00_{3.83}$	$66.75_{2.63}$	$54.75_{5.91}$	$69.50_{3.11}$	$71.00_{3.83}$	$59.25_{2.99}$

[1] Kusner, Matt J., et al. "Counterfactual fairness." Advances in neural information processing systems 30 (2017).

Use-case: fairness auditing and classification



	Logistic classifier			SVM classifier		
	`full`	`unaware`	`fair x`	`full`	`unaware`	`fair x`
F1-score	$72.28_{6.16}$	$72.37_{4.90}$	$59.66_{8.57}$	$76.04_{2.86}$	$76.80_{5.82}$	$68.28_{5.74}$
Accuracy	$67.00_{3.83}$	$66.75_{2.63}$	$54.75_{5.91}$	$69.50_{3.11}$	$71.00_{3.83}$	$59.25_{2.99}$
Unfairness	$5.84_{2.93}$	$2.81_{0.72}$	$0.00_{0.00}$	$6.65_{2.45}$	$2.78_{0.40}$	$0.00_{0.00}$

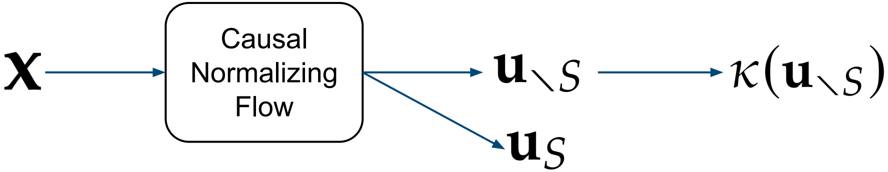
Causal Normalizing Flow →

$$\mathbb{E}_{\mathbf{x}^{\mathrm{f}}}\left[P(\kappa(\mathbf{x}^{\mathrm{cf}}) = 1 \mid do(\mathrm{x}_S = 1), \mathbf{x}^{\mathrm{f}}) - P(\kappa(\mathbf{x}^{\mathrm{cf}}) = 1 \mid do(\mathrm{x}_S = 0), \mathbf{x}^{\mathrm{f}})\right]$$

[1] Kusner, Matt J., et al. "Counterfactual fairness." Advances in neural information processing systems 30 (2017).

Use-case: fairness auditing and classification

	Logistic classifier				SVM classifier			
	`full`	`unaware`	`fair x`	`fair` **u**	`full`	`unaware`	`fair x`	`fair` **u**
F1-score	$72.28_{6.16}$	$72.37_{4.90}$	$59.66_{8.57}$	$73.08_{4.38}$	$76.04_{2.86}$	$76.80_{5.82}$	$68.28_{5.74}$	$77.39_{1.52}$
Accuracy	$67.00_{3.83}$	$66.75_{2.63}$	$54.75_{5.91}$	$66.50_{3.70}$	$69.50_{3.11}$	$71.00_{3.83}$	$59.25_{2.99}$	$69.75_{1.26}$
Unfairness	$5.84_{2.93}$	$2.81_{0.72}$	$0.00_{0.00}$	$0.00_{0.00}$	$6.65_{2.45}$	$2.78_{0.40}$	$0.00_{0.00}$	$0.00_{0.00}$



$\mathbf{x}$

Causal Normalizing Flow

$\mathbf{u}_{\setminus S}$

$\mathbf{u}_S$

$\kappa(\mathbf{u}_{\setminus S})$

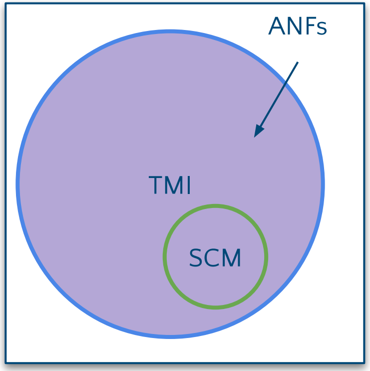
[1] Kusner, Matt J., et al. "Counterfactual fairness." Advances in neural information processing systems 30 (2017).

Concluding remarks

- Causal normalizing flows are a **natural choice** to learn SCMs.

- We provide **theoretical** results, and practical ways to:
  - **efficiently** capture a causal model, and
  - **exactly** perform causal inference.

- Lots of interesting future work! **Get in touch!**
  - Confounders?
  - Non-bijective generators?
  - Better loss functions?
  - Misspecifications?
  - Applications?



About to graduate!

Today at Poster **#822**

5:15 p.m. – 7:15 p.m

Hiring!

Questions?

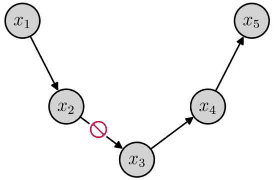
[arxiv: 2306.05415](https://arxiv.org/abs/2306.05415)

[psanch21/causal-flows](https://github.com/psanch21/causal-flows)

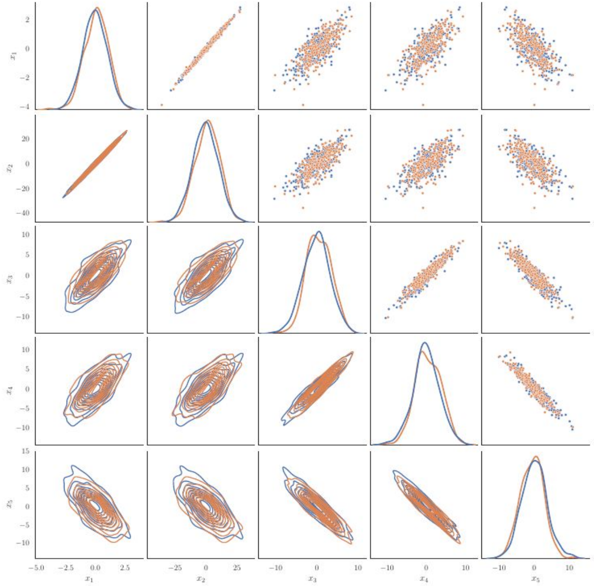
Does it work?

☑ Theoretically: App. C ⇒ Intuition: the $\mathbf{u}_i$ of the intervened value is set to cancel out the influence of its parents.

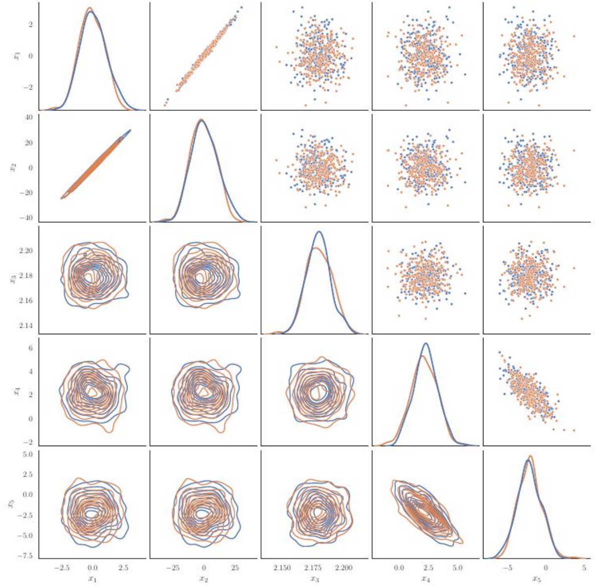
☑ Empirically:



(c) 5-CHAIN



(a) Observational distribution.



(b) Interventional distribution $do(\mathrm{x}_3 = 2.18)$.

Structural Causal Models

causal generator

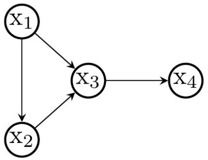
exogenous distribution

An SCM is a tuple $\mathcal{M} = (\tilde{\mathbf{f}}, P_{\mathbf{u}})$ describing a data-generating process to transform exogenous variables **u** into (observed) endogenous variables **x**.

$$\mathbf{u} \coloneqq (\mathrm{u}_1, \mathrm{u}_2, \ldots, \mathrm{u}_d) \sim P_{\mathbf{u}}\,, \qquad \mathrm{x}_i = \tilde{f}_i(\mathbf{x}_{\mathrm{pa}_i}, \mathrm{u}_i)\,, \qquad \text{for } i = 1, 2, \ldots, d\,.$$

$$\begin{cases} \mathrm{x}_1 = \tilde{f}_1(\mathrm{u}_1) \\ \mathrm{x}_2 = \tilde{f}_2(\mathrm{x}_1, \mathrm{u}_2) \\ \mathrm{x}_3 = \tilde{f}_3(\mathrm{x}_1, \mathrm{x}_2, \mathrm{u}_3) \\ \mathrm{x}_4 = \tilde{f}_4(\mathrm{x}_3, \mathrm{u}_4) \end{cases} \longrightarrow \{x_1,\ x_2,\ x_3,\ x_4\}$$

- Causal graph
- Adj. matrix
- Causal ordering



$$\boldsymbol{G} \coloneqq \nabla_{\mathbf{x}} \tilde{\mathbf{f}} \neq \mathbf{0}$$

$$\boldsymbol{G} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$$

We can use SCMs for causal inference, i.e., reason about what-if questions:
How the world would have been if X happened.

$$\begin{cases} \mathrm{x}_1 = \tilde{f}_1(\mathrm{u}_1) \\ \mathrm{x}_2 = \alpha \\ \mathrm{x}_3 = \tilde{f}_3(\mathrm{x}_1, \mathrm{x}_2, \mathrm{u}_3) \\ \mathrm{x}_4 = \tilde{f}_4(\mathrm{x}_3, \mathrm{u}_4) \end{cases} \longrightarrow \{x_1,\ \alpha,\ x_3,\ x_4\}$$

Normalizing flows

flow
(neural net)

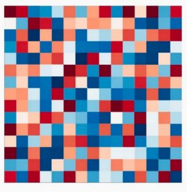
base
distribution

An NF is a tuple $(T_{\boldsymbol{\theta}}, P_{\mathbf{u}})$ that express the probability density of observed variables **x** as the transformation of base variables **u**:

$$T_{\boldsymbol{\theta}}(\mathbf{x}) =: \mathbf{u} \sim P_{\mathbf{u}} \quad \text{with log-density} \quad \log p(\mathbf{x}) = \log p(T_{\boldsymbol{\theta}}(\mathbf{x})) + \log|\det(\nabla_{\mathbf{x}} T_{\boldsymbol{\theta}}(\mathbf{x}))|$$

Learn $\boldsymbol{\theta}$ via MLE!

Jacobian



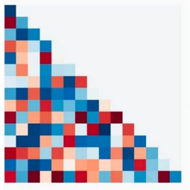
Autoregressive NFs (ANFs) model each layer of the network as:

$$\mathrm{z}_i^l := \tau_i^l(\mathrm{z}_i^{l-1}; \mathbf{h}_i^l)\,, \quad \text{where} \quad \mathbf{h}_i^l := c_i^l(\mathbf{z}_{1:i-1}^{l-1})$$

transformer
(str. monotonic)

conditioner
(only takes prev. inputs)

Jacobian



SCMs as TMI maps

We can always write an SCM as a TMI map.

1. Unroll the SCM.

☑ Acyclic

$$\begin{cases} \mathrm{x}_1 = \tilde{f}_1(\mathrm{u}_1) \\ \mathrm{x}_2 = \tilde{f}_2(\mathrm{x}_1, \mathrm{u}_2) = \tilde{f}_2(\tilde{f}_1(\mathrm{u}_1), \mathrm{u}_2) \\ \mathrm{x}_3 = \tilde{f}_3(\mathrm{x}_1, \mathrm{x}_2, \mathrm{u}_3) = \tilde{f}_3(\tilde{f}_1(\mathrm{u}_1), \tilde{f}_2(\tilde{f}_1(\mathrm{u}_1), \mathrm{u}_2), \mathrm{u}_3) \\ \mathrm{x}_4 = \tilde{f}_4(\mathrm{x}_3, \mathrm{u}_4) = \tilde{f}_4(\tilde{f}_3(\tilde{f}_1(\mathrm{u}_1), \tilde{f}_2(\tilde{f}_1(\mathrm{u}_1), \mathrm{u}_2), \mathrm{u}_3), \mathrm{u}_4) \end{cases}$$

2. “Monotonize” the SCM.

Always possible. How? Apply a Knöthe-Rosenblatt (KR) transport following the causal graph:

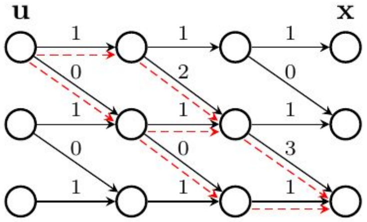
$$K_m(x_{1:m-1}, x_m) = F_\nu^{-1}\{F_\mu(x_m | x_{1:m-1}) \mid K_1(x_1), \ldots, K_{m-1}(x_{1:m-1})\}$$

If $P_\mathbf{u}$ is a standard uniform distribution ⇒ Darmois construction.

Network designs

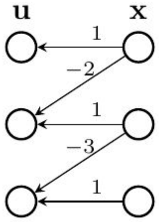
- Generative networks:
    - Defined from **u** to **x**.
    - The conditioner only takes the input according to **G**.

$$\mathrm{z}_i^{l-1} = \tau_i(\mathrm{z}_i^l; \mathbf{h}_i^{l-1}), \quad \text{where} \quad \mathbf{h}_i^{l-1} = c_i(\mathbf{z}_{\mathrm{pa}_i}^l)$$



- Abductive networks:
    - Defined from **x** to **u**.
    - The conditioner only takes the input according to **G**.

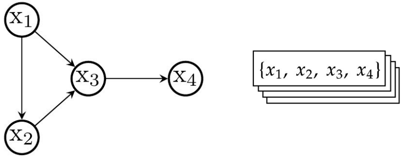
$$\mathrm{z}_i^{l} = \tau_i(\mathrm{z}_i^{l-1}; \mathbf{h}_i^{l}), \quad \text{where} \quad \mathbf{h}_i^{l} = c_i(\mathbf{z}_{\mathrm{pa}_i}^{l-1})$$



Usual implementation

The do-operator simulates an *external intervention* in the system, breaking any causal relationships going to the intervened node.

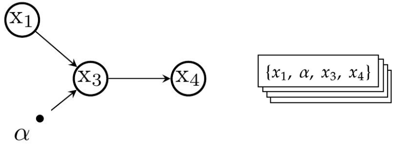
$$\begin{cases} x_1 = f_1(u_1) \\ x_2 = f_2(x_1, u_2) \\ x_3 = f_3(x_1, x_2, u_3) \\ x_4 = f_4(x_3, u_4) \end{cases}$$



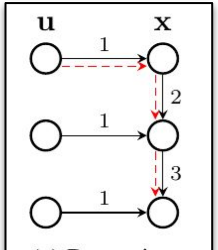
The usual implementation yields an intervened SCM with a new set of equations, $\mathcal{M}^{\mathcal{I}} = (\tilde{\mathbf{f}}^{\mathcal{I}}, P_{\mathbf{u}})$

However, it only works for the recursive formulation.

$$\begin{cases} x_1 = f_1(u_1) \\ x_2 = \alpha \\ x_3 = f_3(x_1, x_2, u_3) \\ x_4 = f_4(x_3, u_4) \end{cases}$$

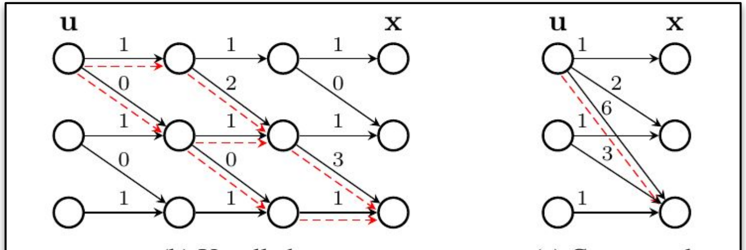


✓



(a) Recursive.

✗



(b) Unrolled. (c) Compacted.

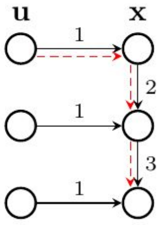
Our implementation

We propose to instead update $P_{\mathbf{u}}$ to put mass only on those values of **u** that yield the intervened value, $\mathcal{M}^{\mathcal{I}} = (\tilde{\mathbf{f}}, P_{\mathbf{u}}^{\mathcal{I}})$.

$$p^{\mathcal{I}}(\mathbf{u}) = \delta\left(\left\{\tilde{f}_i(\mathbf{x}_{\mathrm{pa}_i}, \mathrm{u}_i) = \alpha\right\}\right) \cdot \prod_{j \neq i} p_j(\mathrm{u}_j)$$

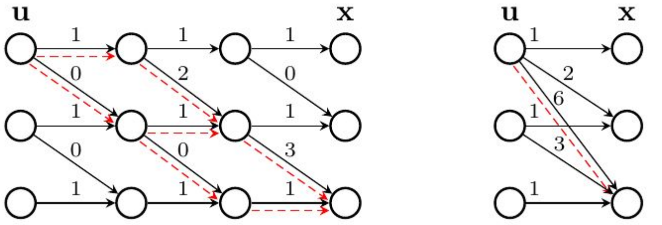
```
function SAMPLEINTERVENEDDIST($i$, $\alpha$)
    $\mathbf{u} \sim P_{\mathbf{u}}$
    $\mathbf{x} \leftarrow T_{\boldsymbol{\theta}}^{-1}(\mathbf{u})$
    $\mathrm{x}_i \leftarrow \alpha$
    $\mathrm{u}_i \leftarrow T_{\boldsymbol{\theta}}(\mathbf{x})_i$
    $\mathbf{x} \leftarrow T_{\boldsymbol{\theta}}^{-1}(\mathbf{u})$
    return $\mathbf{x}$
end function
```

✓



(a) Recursive.

✓

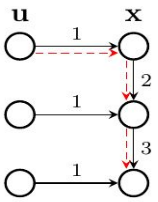


(b) Unrolled.

(c) Compacted.

The multiple representations of SCMs

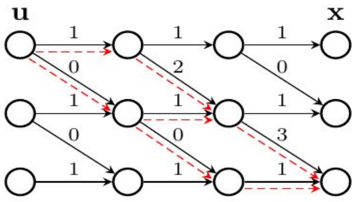
$$\mathbf{x} = \boldsymbol{G}\mathbf{x} + \boldsymbol{I}\mathbf{u}$$



(a) Recursive.

$$\begin{cases} \mathrm{x}_1 = \mathrm{u}_1 \\ \mathrm{x}_2 = 2\mathrm{x}_1 + \mathrm{u}_2 \\ \mathrm{x}_3 = 3\mathrm{x}_2 + \mathrm{u}_3 \end{cases}$$

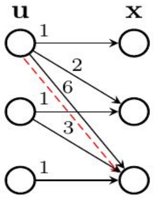
$$\mathbf{x} = \boldsymbol{G}_3(\boldsymbol{G}_2(\boldsymbol{G}_1\mathbf{u}))$$



(b) Unrolled.

$$\begin{cases} \mathrm{z}_1^1 = \mathrm{u}_1 \\ \mathrm{z}_2^1 = \mathrm{u}_2 \\ \mathrm{z}_3^1 = \mathrm{u}_3 \end{cases} \Rightarrow \begin{cases} \mathrm{z}_1^2 = \mathrm{z}_1^1 \\ \mathrm{z}_2^2 = 2\mathrm{z}_1^1 + \mathrm{z}_2^1 \\ \mathrm{z}_3^2 = \mathrm{z}_3^1 \end{cases} \Rightarrow \begin{cases} \mathrm{x}_1 = \mathrm{z}_1^2 \\ \mathrm{x}_2 = \mathrm{z}_2^2 \\ \mathrm{x}_3 = 3\mathrm{z}_2^2 + \mathrm{z}_3^2 \end{cases}$$

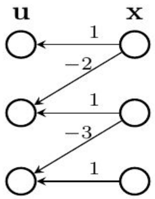
$$\mathbf{x} = (\boldsymbol{G}^2 + \boldsymbol{G} + \boldsymbol{I})\mathbf{u}$$



(c) Compacted.

$$\begin{cases} \mathrm{x}_1 = \mathrm{u}_1 \\ \mathrm{x}_2 = 2\mathrm{u}_1 + \mathrm{u}_2 \\ \mathrm{x}_3 = 6\mathrm{u}_1 + 3\mathrm{u}_2 + \mathrm{u}_3 \end{cases}$$

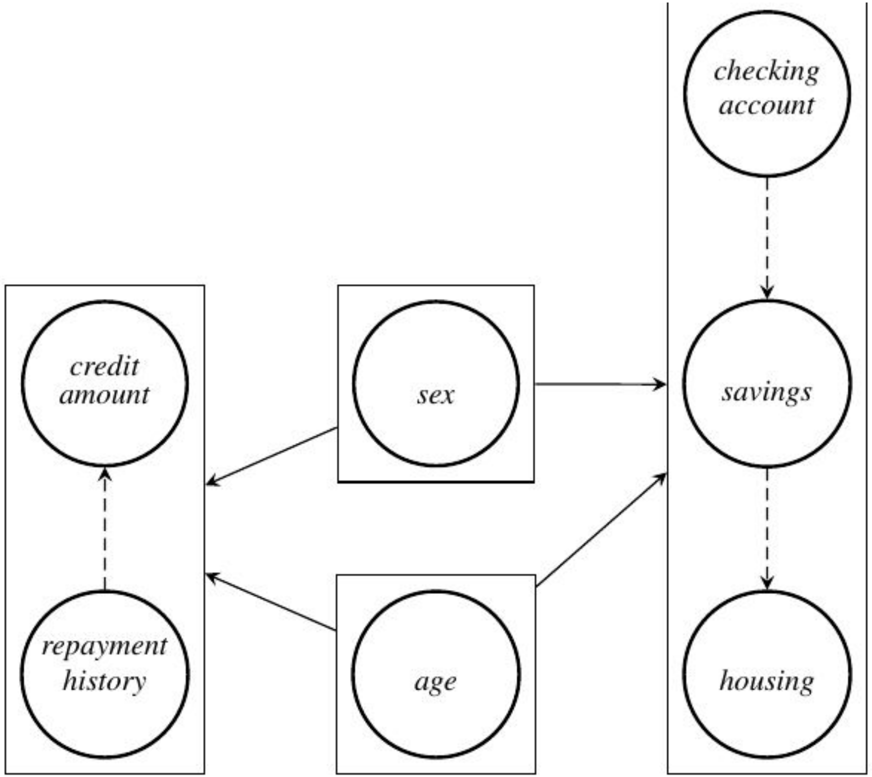
$$\mathbf{u} = (\boldsymbol{I} - \boldsymbol{G})\mathbf{x}$$



(d) Inverted.

$$\begin{cases} \mathrm{u}_1 = \mathrm{x}_1 \\ \mathrm{u}_2 = \mathrm{x}_2 - 2\mathrm{x}_1 \\ \mathrm{u}_3 = \mathrm{x}_3 - 3\mathrm{x}_2 \end{cases}$$

Extra


checking account
credit amount
sex
savings
repayment history
age
housing