Normalizing Flows

and

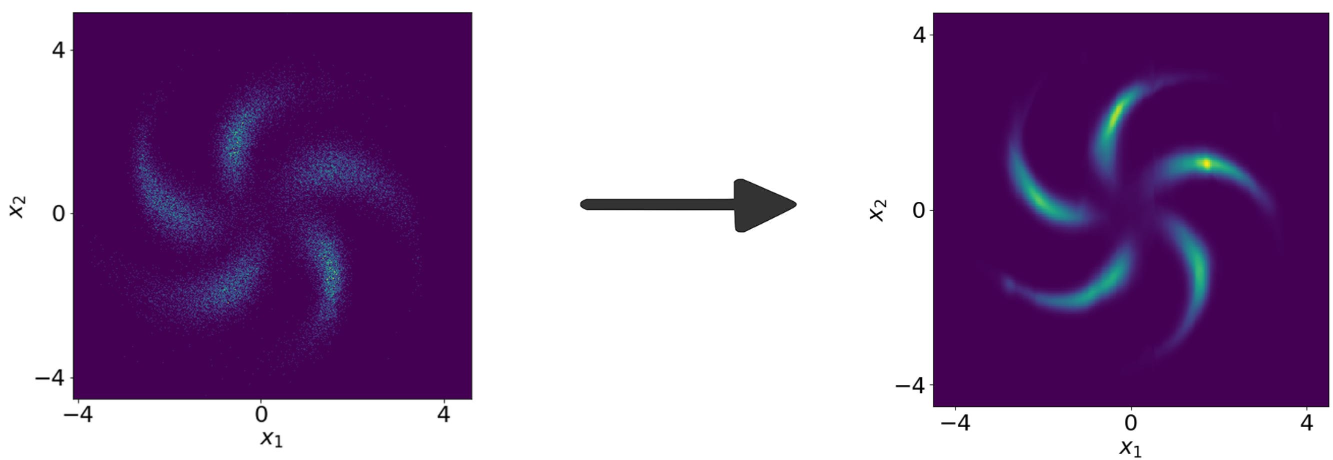
Bayesian Networks

CogSys seminar, October 2020

Antoine Wehenkel

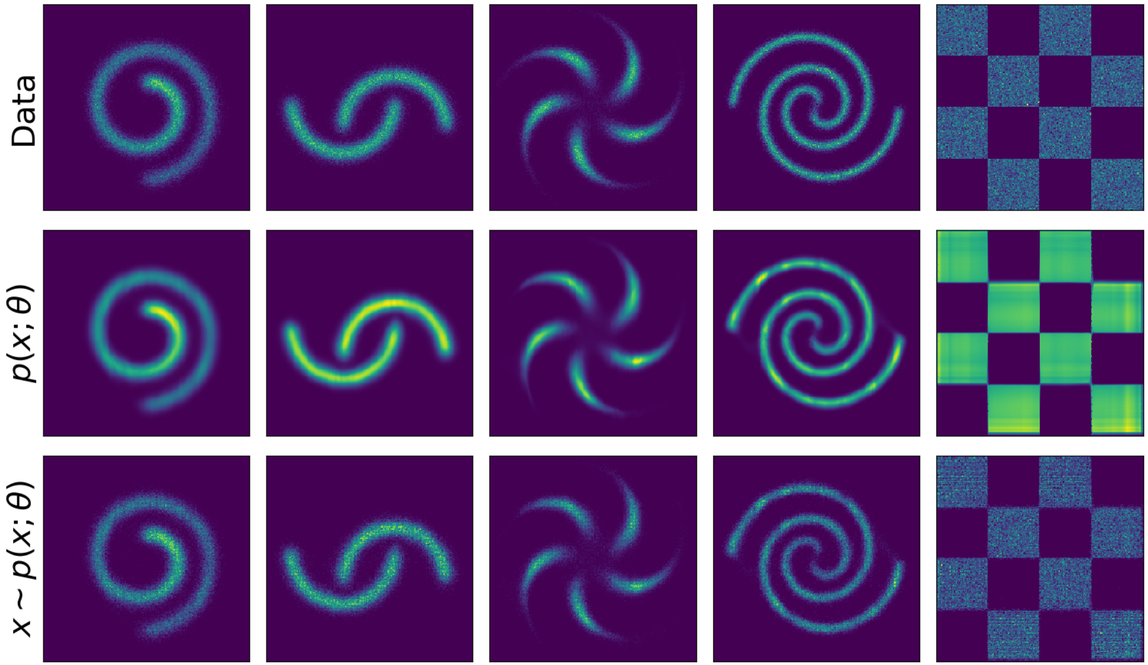
NFs pros 💪

- Access to the model's likelihood



NFs pros 💪

- Access to the model's likelihood
- Universal density estimators



NFs pros 💪

- Access to the model's likelihood
- Universal density estimators
- Good results for high dimensional data

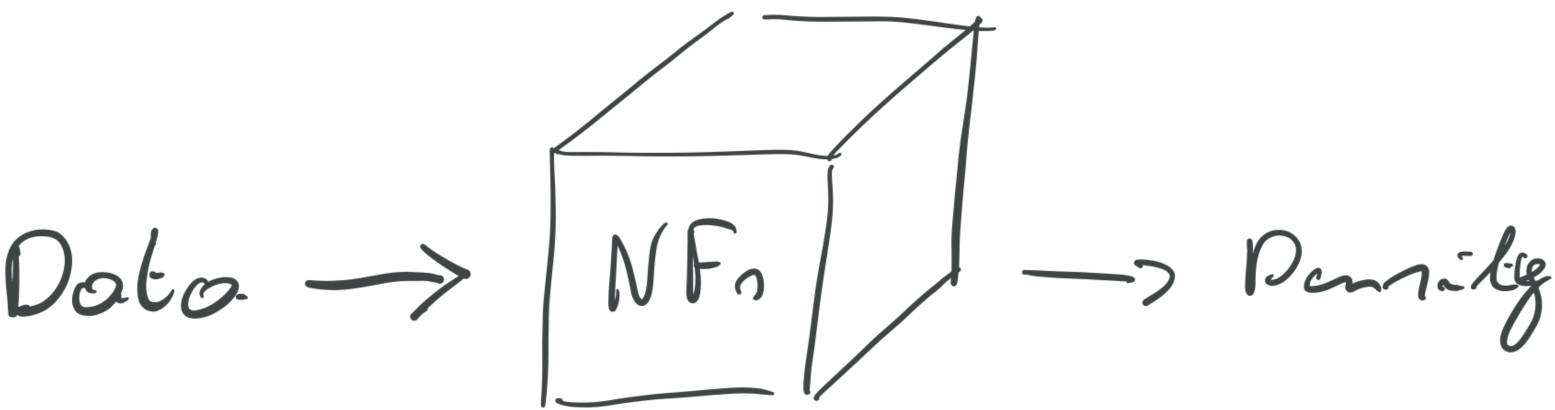
---
https://openai.com/blog/glow/

NFs cons 😈

- Arbitrary architectural choices

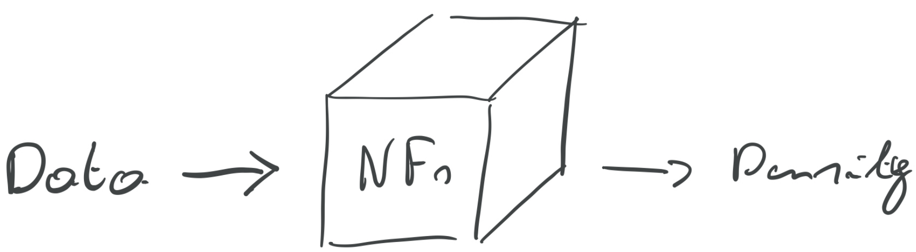
NFs cons 😈

- Arbitrary architectural choices
- Hard to interpret



NFs cons 😈

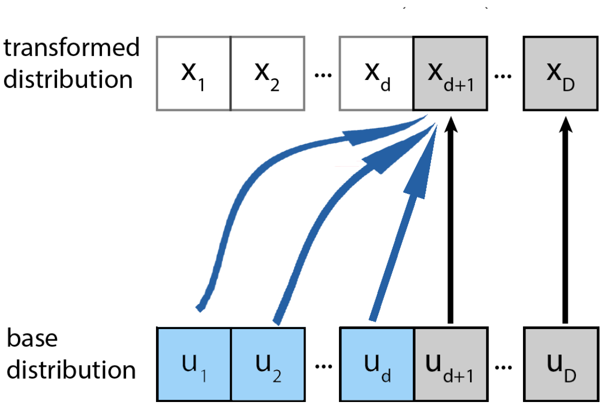
- Arbitrary architectural choices
- Hard to interpret
- Poor inductive bias



Inductive bias in NFs

How is it tackled now?

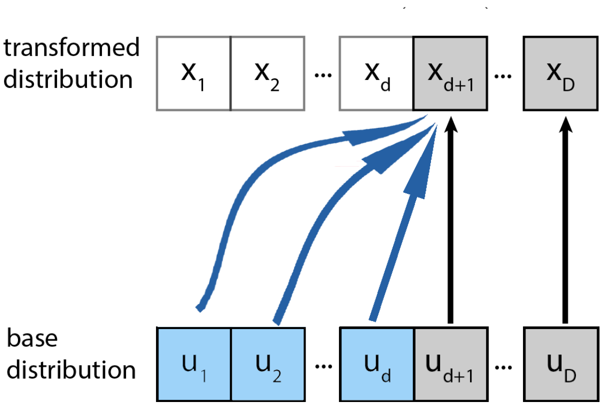
- For images:
    - Coupling layers

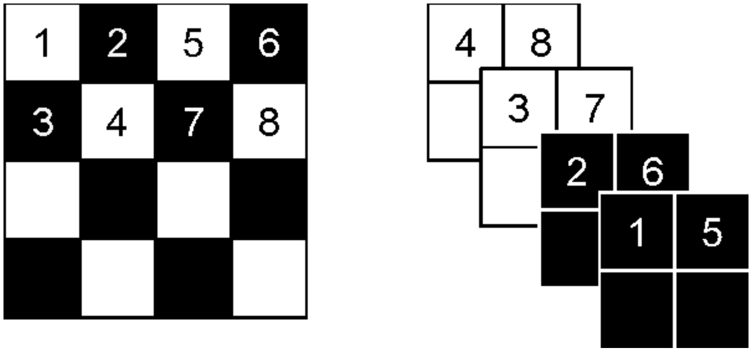


Inductive bias in NFs

How is it tackled now?

- For images:
    - Coupling layers
    - Multi-scale architectures





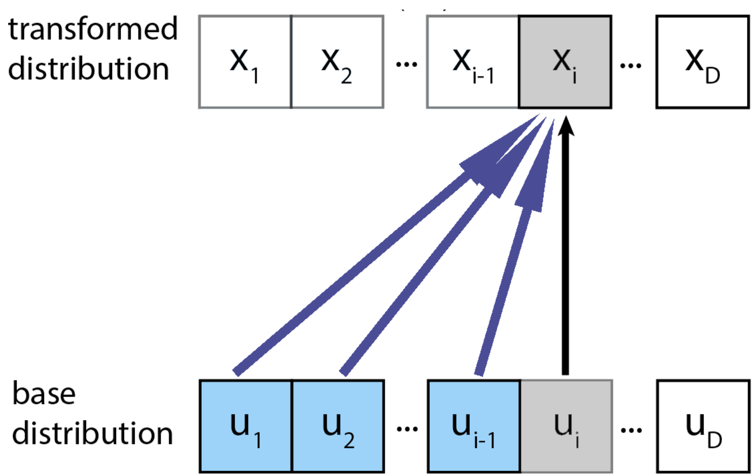
———

Dinh L, Sohl-Dickstein J, Bengio S. Density estimation using real nvp.

Inductive bias in NFs

How is it tackled now?

- For images:
  - Coupling layers
  - Multi-scale architectures
- For time series:
  - Autoregressive architectures



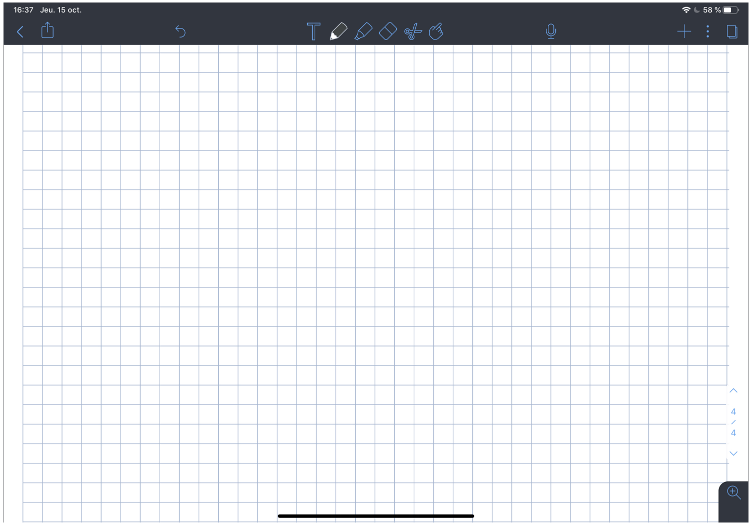
---

https://blog.evjang.com/2018/01/nf2.html

Inductive bias in NFs

How is it tackled now?

- For images:
  - Coupling layers
  - Multi-scale architectures
- For time series:
  - Autoregressive architectures
- What about tabular data or mixed data?



It is not easy to design the architecture and to understand the modeling assumptions!

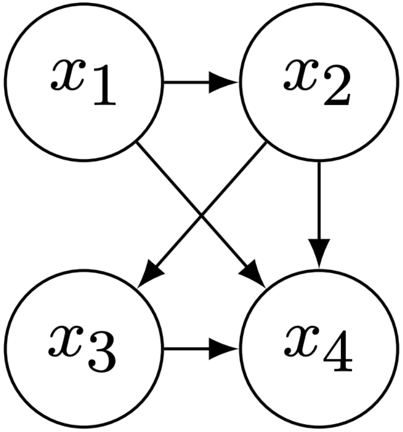
Bayesian Networks

- Probabilistic graphical models formally introduced by Judea Pearl in the 80's
- A Bayesian network is a directed acyclic graph that factorizes the model distribution as

$$p(\mathbf{x}) = \prod_{i=1}^{D} p(x_i | \mathcal{P}_i).$$

- e.g when $d = 4$:

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_1, x_2, x_3)$$



BNs: pros 💪 and cons 😈

- Good for modeling independencies and check their global impact on the modeled density 💪

BNs: pros 💪 and cons 😈

- Good for modeling independencies and check their global impact on the modeled density 💪
- Applications across science and technology 💪

BNs: pros 💪 and cons 😈

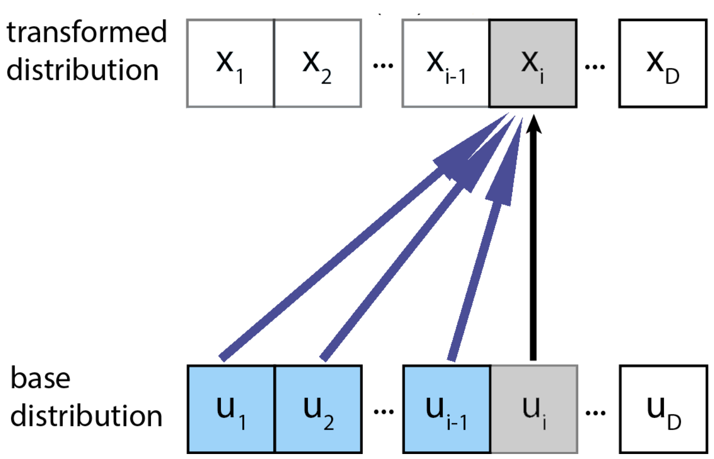
- Good for modeling independencies and check their global impact on the modeled density 💪
- Applications across science and technology 💪
- Often used with discrete or discretized data 😈

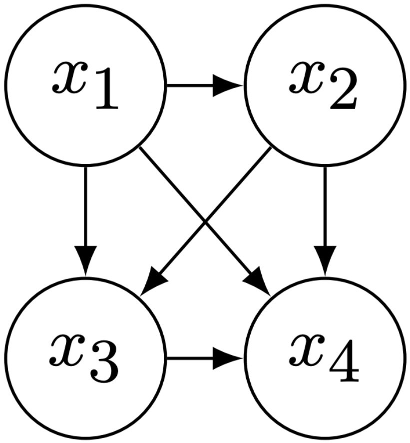
BNs: pros 💪 and cons 😈

- Good for modeling independencies and check their global impact on the modeled density 💪
- Applications across science and technology 💪
- Often used with discrete or discretized data 😈
- Outdated with respect to deep learning revolution 😈

Some NFs are BNs

Autoregressive layers





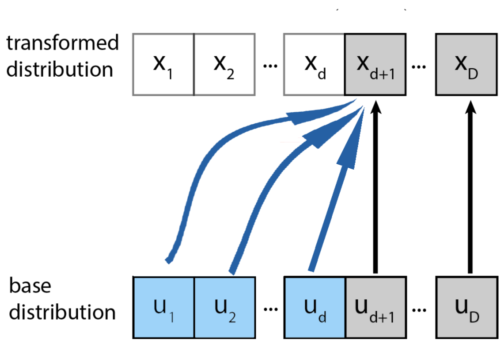
The autoregressive conditioner is defined as $\mathbf{c}^i(\mathbf{u}) = \mathbf{h}^i\left(\begin{bmatrix} u_1 & \dots & u_{i-1}\end{bmatrix}^T\right)$.

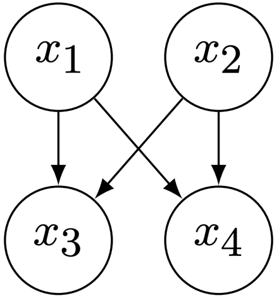
We combine the conditioner with a transformer/normalizer: $x_i = f(u_i; \mathbf{c}^i(\mathbf{u}))$.

An autoregressive density estimator learns the chain rule's factors:
$p(\mathbf{x}) = p(x_1)\Pi_{i=2}^{D} p(x_i | x_1, \dots, x_{i-1})$.

Some NFs are just BNs

Coupling layers





The coupling conditioner can be defined as $\mathbf{c}^i(\mathbf{u}) =$

- $\underline{\mathbf{h}}^i \quad \text{if} \quad i \leq d$ (a constant);
- $\mathbf{h}^i\left(\begin{bmatrix} u_1 & \ldots & u_d \end{bmatrix}^T\right)$ if $i > d$.

Coupling learns the factors of the following factorization:
$p(\mathbf{x}) = \Pi_{i=1}^{d} p(x_i) \Pi_{j=k+1}^{D} p(x_j | x_1, \ldots, x_d)$.

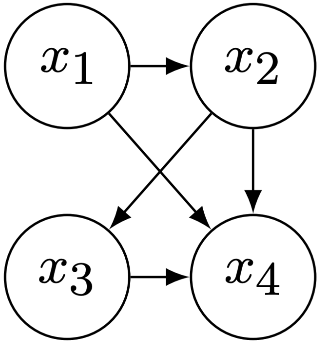
———
https://blog.evjang.com/2018/01/nf2.html

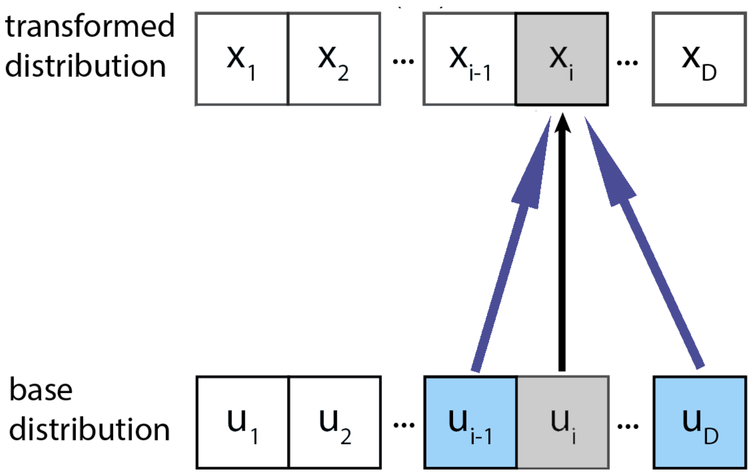
Can any BN lead to a NF layer? 💡

Can any BN lead to a NF layer? 💡

The graphical conditioner

Let $A \in \{0,1\}^D$ be the adjacency matrix of a given Bayesian network for a random vector $\mathbf{x} \in \mathbb{R}^d$. We define the graphical conditioner as: $\mathbf{c}^i(\mathbf{u}) = \mathbf{h}^i(\mathbf{u} \odot A_{i,:})$.



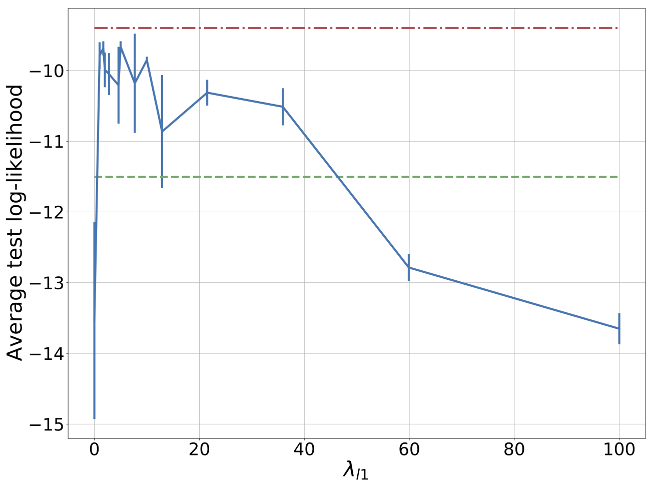


Is it useful in practice?

- It can be critical or convenient to ensure some independencies.
    - E.g. assuming independencies between gender and salary.

Is it useful in practice?

- It can be critical or convenient to respect some independencies.
  - E.g. assuming independencies between sex and salary.
- Knowing the topology helps learning good densities.



Why not learning the topology?

- Any BN corresponds to a DAG, but any DAG can be seen as the topology of a BN as well.

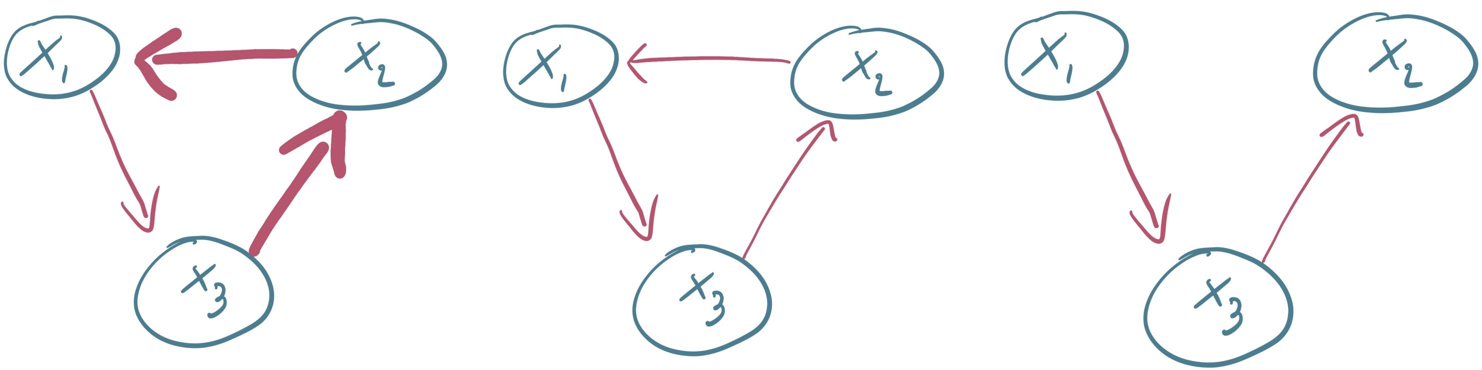
Why not learning the topology?

- Any BN corresponds to a DAG, but any DAG can be seen as the topology of a BN as well.
- We look for the DAG that maximizes the model's likelihood: $\max_{A \in \mathbb{R}^{d \times d}} F(A)$ s.t. $\mathcal{G}(A) \in$ **DAGs**.

Why not learning the topology?

- Any BN corresponds to a DAG, but any DAG can be seen as the topology of a BN as well.
- We look for the DAG that maximizes the model's likelihood: $\max_{A \in \mathbb{R}^{d \times d}} F(A)$ s.t. $\mathcal{G}(A) \in$ **DAGs**.
- We can formulate it as a continuous constraint:

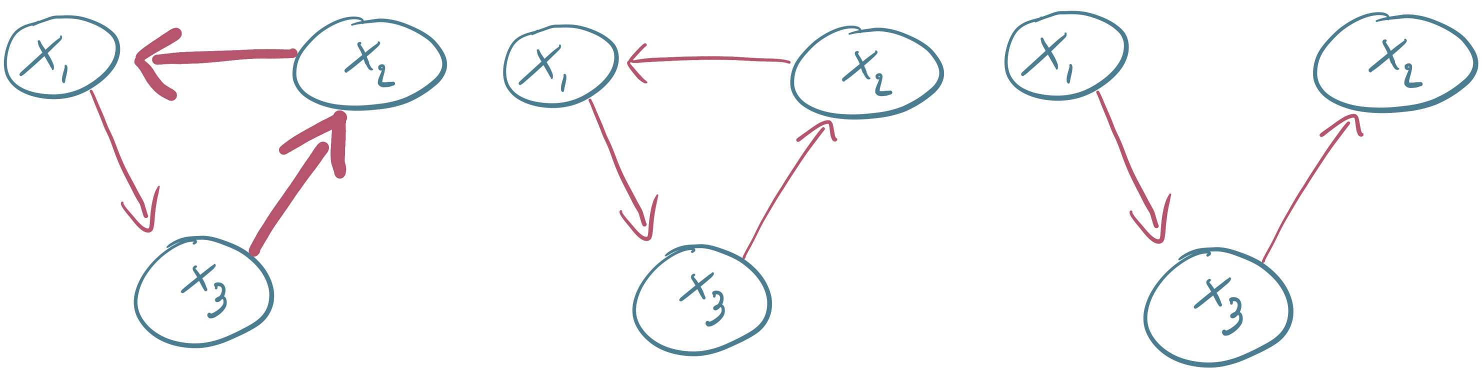
$$\max_{A \in \mathbb{R}^{d \times d}} F(A) \text{ s.t. } w(A) = 0 \text{ where } w(A) := \mathrm{Trace}\left(\sum_{i=1}^{D} A^i\right).$$



Why not learning the topology?

- Any BN corresponds to a DAG, but any DAG can be seen as the topology of a BN as well.
- We look for the DAG that maximizes the model's likelihood: $\max_{A\in\mathbb{R}^{d\times d}} F(A)$ s.t. $\mathcal{G}(A) \in$ **DAGs**.
- We can formulate it as a continuous constraint:

$$\max_{A\in\mathbb{R}^{d\times d}} F(A) \text{ s.t. } w(A) = 0 \text{ where } w(A) := \mathbf{Trace}\left(\sum_{i=1}^{D} A^i\right).$$



- We can solve the continuously constrained problem with a Lagrangian formulation!

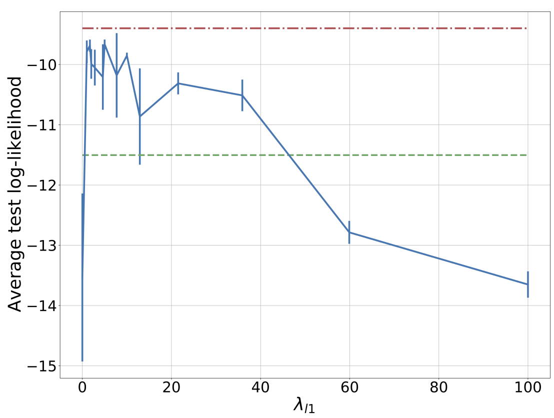
Computational cost

- Solving the sub-problems to optimality increases computational cost 😈
- As fast as autoregressive or coupling layers at inference time 💪
- The inversion of the flow will be often faster than autoregressive architectures 💪

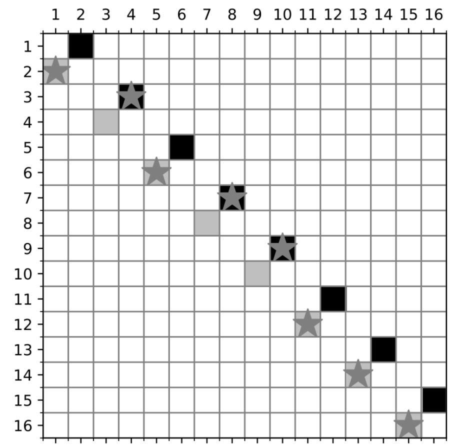
Results

Known vs Unknown Topology (Monotonic transformer)

Effect of sparsity



Topology recovered



Learning a good topology helps for density estimation.

Results

Density estimation benchmark

Dataset	POWER	GAS	HEPMASS	MINIBOONE	BSDS300
Graph.-UMNN (1)	$0.62_{\pm.04}$	$10.15_{\pm.15}$	$-14.17_{\pm.13}$	$-16.23_{\pm.52}$	$155.22_{\pm.11}$
MAF (5)	$0.14_{\pm.01}$	$9.07_{\pm.01}$	$-17.70_{\pm.01}$	$-11.75_{\pm.22}$	$155.69_{\pm.14}$
Glow* (10)	$0.42_{\pm.01}$	$12.24_{\pm.03}$	$-16.99_{\pm.02}$	$-10.55_{\pm.45}$	$156.95_{\pm.28}$
UMNN-MAF* (5)	$0.63_{\pm.01}$	$10.89_{\pm.70}$	$-13.99_{\pm.21}$	$-9.67_{\pm.13}$	$157.98_{\pm.01}$
Q-NSF* (10)	$0.66_{\pm.01}$	$12.91_{\pm.01}$	$-14.67_{\pm.02}$	$-9.72_{\pm.24}$	$157.42_{\pm.14}$
FFJORD* (5-5-10-1-2)	$0.46_{\pm.01}$	$8.59_{\pm.12}$	$-14.92_{\pm.08}$	$-10.43_{\pm.04}$	$157.40_{\pm.19}$

We may obtain density estimation results on par with the best NF architectures.

Perspectives

For graphical NFs

- Could we benefit from graphical NFs independencies with multiple steps?
- What about partial domain knowledge?
- Combine these models with causal reasoning.

More details about BNs and NFs:

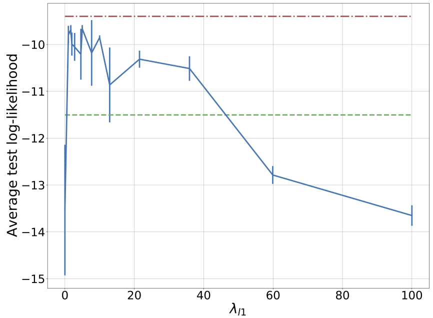
- Graphical Normalizing Flows, A. Wehenkel and G. Louppe, October 2020 - https://arxiv.org/abs/2006.02548
- You say Normalizing Flows I see Bayesian Networks, A. Wehenkel and G. Louppe, June 2020 - https://arxiv.org/abs/2006.00866

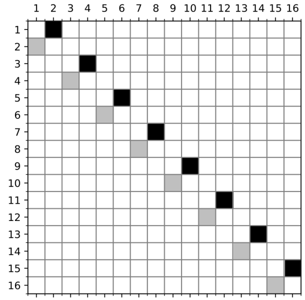
Thanks for listening

Results

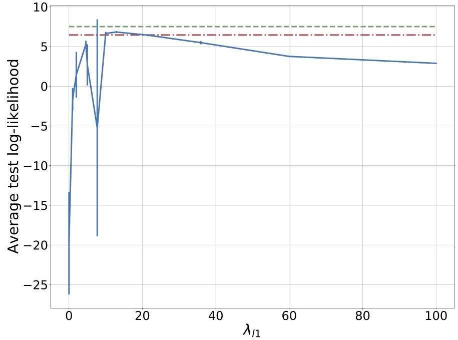
Known vs Unknown Topology (Monotonic transformer)

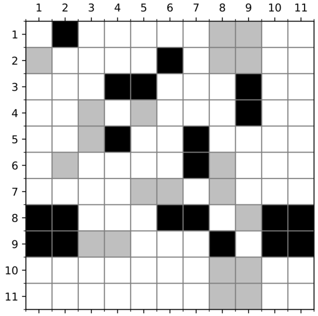
8 pairs of independent variables





Human protein dataset



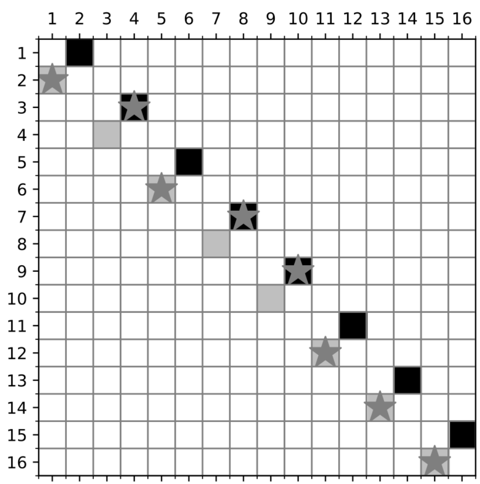


Learning a good topology helps for density estimation.

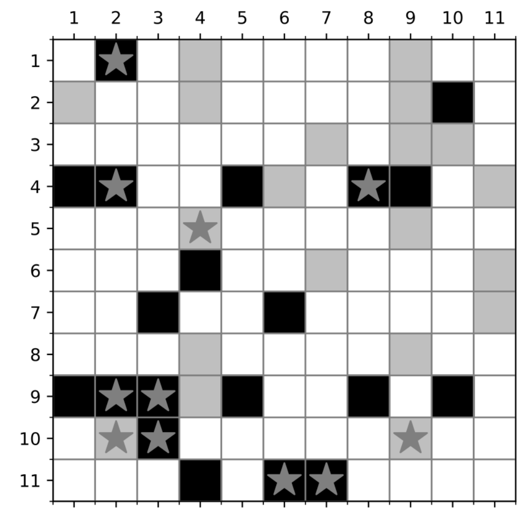
Results

Relevance of the discovered topology (Monotonic transformer)

8 pairs of independent variables



Human protein dataset



The optimization is able to remove spurious dependencies and keeps the correct ones.