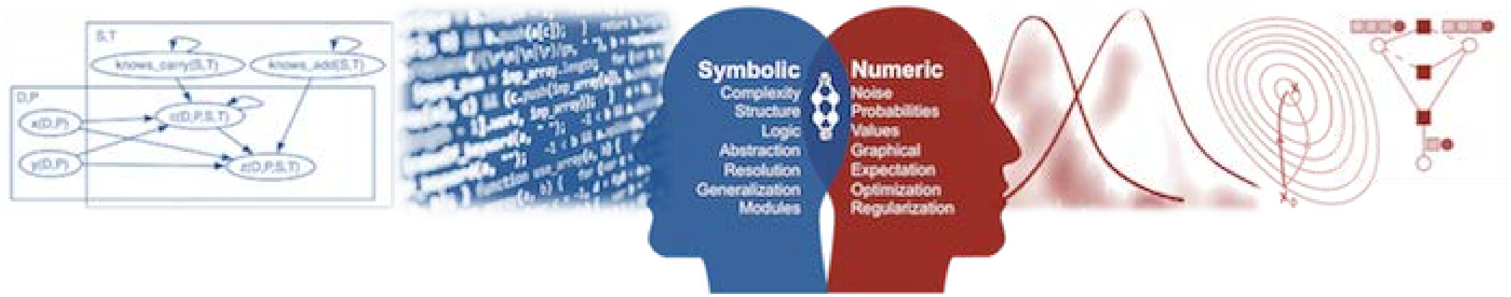
TECHNISCHE UNIVERSITÄT DARMSTADT



Fachbereich Informatik

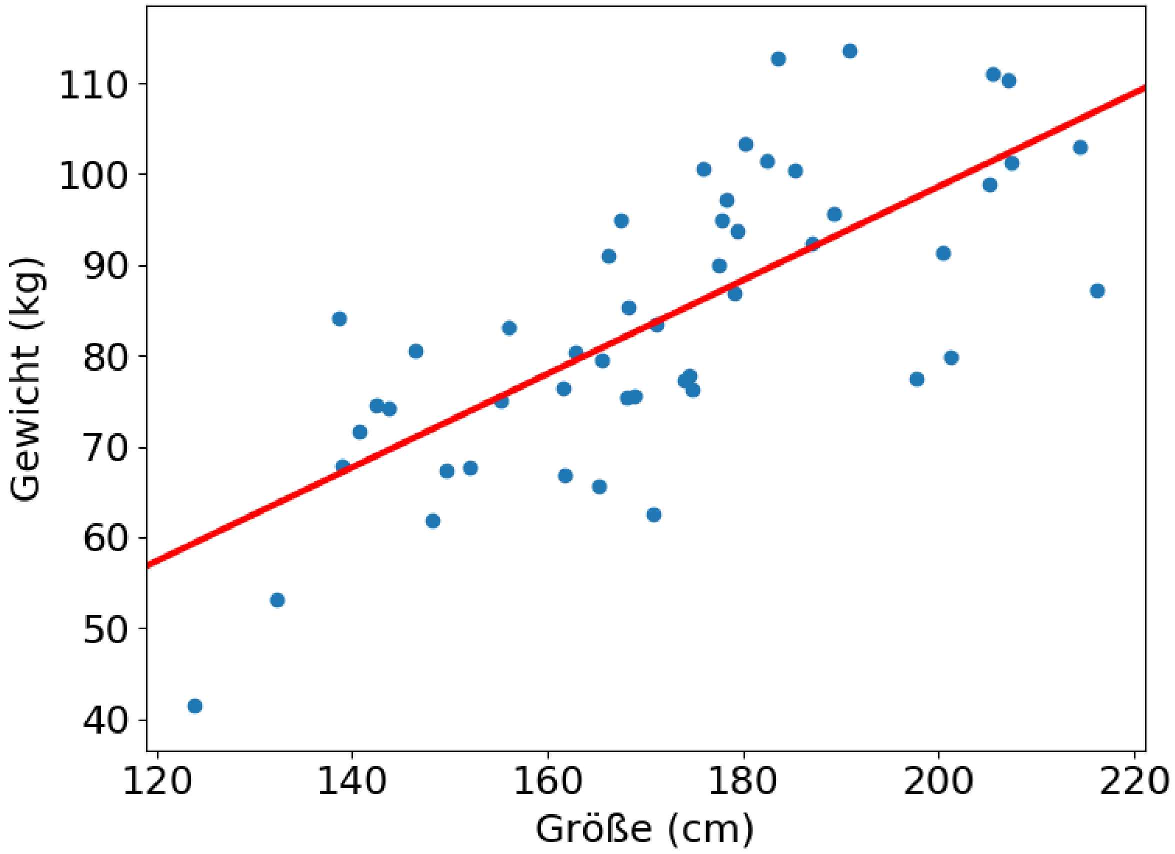
Machine Learning Applications: Generative Modelling

Karl Stelzner

Machine Learning Group, TU Darmstadt

November 15, 2019

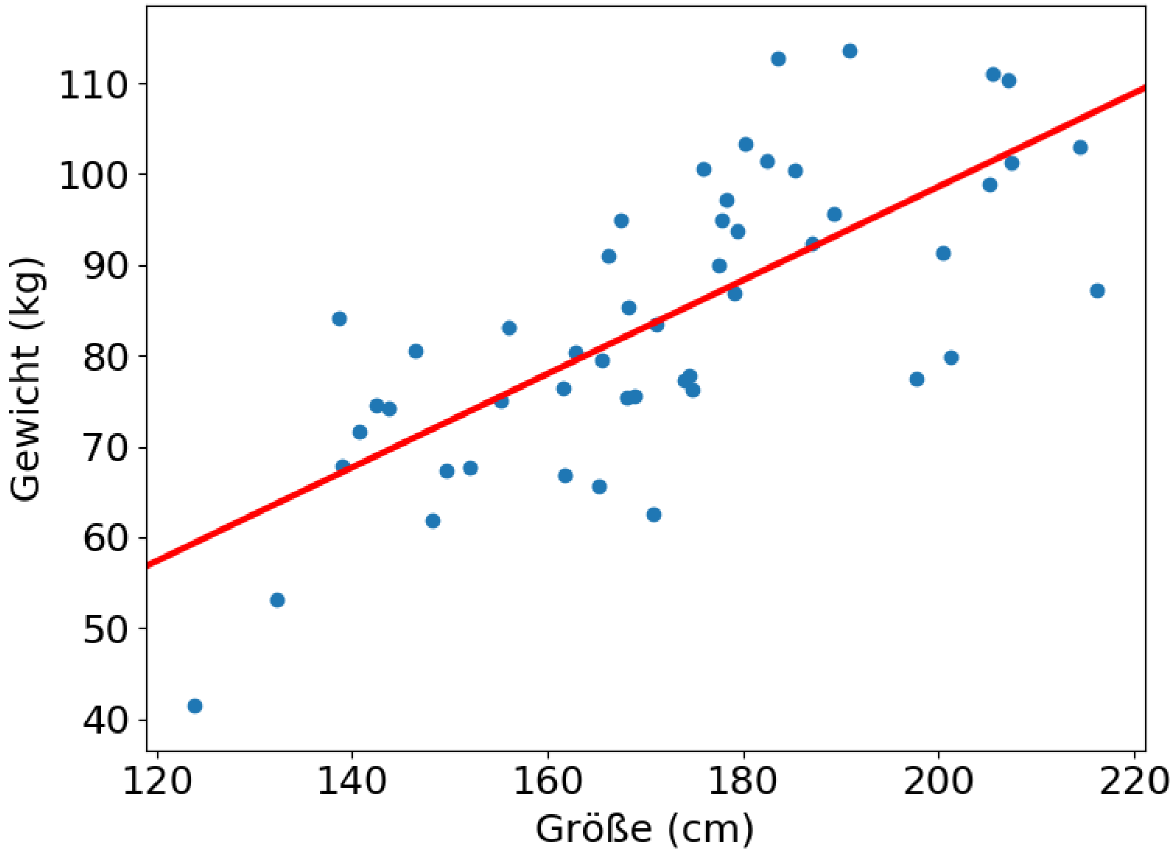
Regression is Conditional Density Estimation



- Consider a regression problem with features X and target Y
- We would like to find a predictor f(X) which minimizes an error

$$MSE = \sum_{i=1}^{N} (f(x_i) - y_i)^2$$

Regression is Conditional Density Estimation



- From a probabilistic perspective, this corresponds to estimating a conditional probability distribution p(y | x)
- For instance we can define such as distribution as a Gaussian

$$y \sim \mathcal{N}(\mu = f(x), \sigma^2)$$

Regression is Conditional Density Estimation

- The most common way to estimate a distribution is to maximize its log likelihood, i.e.

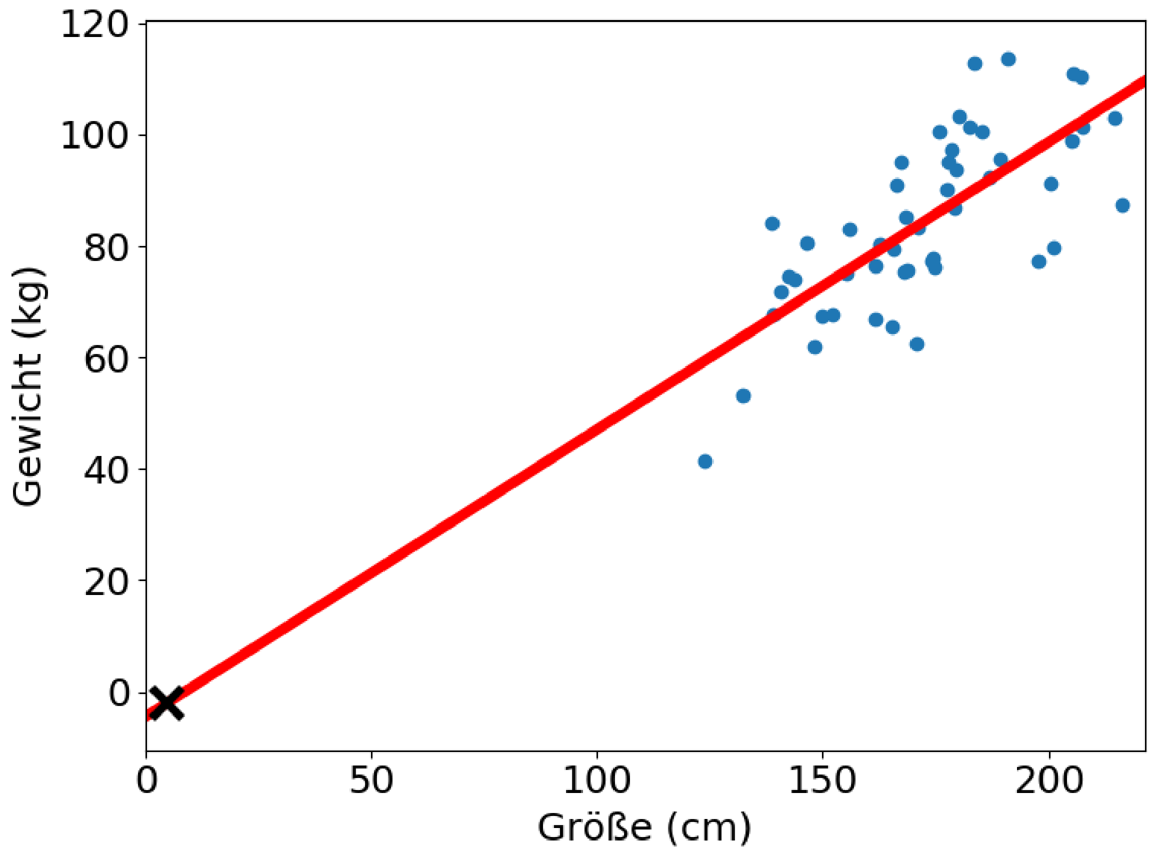
$$\mathrm{LL} = \log p(Y \mid X) = \log \prod_i p(y_i|x_i) = \sum_i \log p(y_i|x_i)$$

- For normal distributions, this likelihood is

$$LL = -n \log \sigma - \frac{n}{2} \log(2\pi) - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (\mu_i - y_i)^2 = c_1 - c_2\, MSE$$

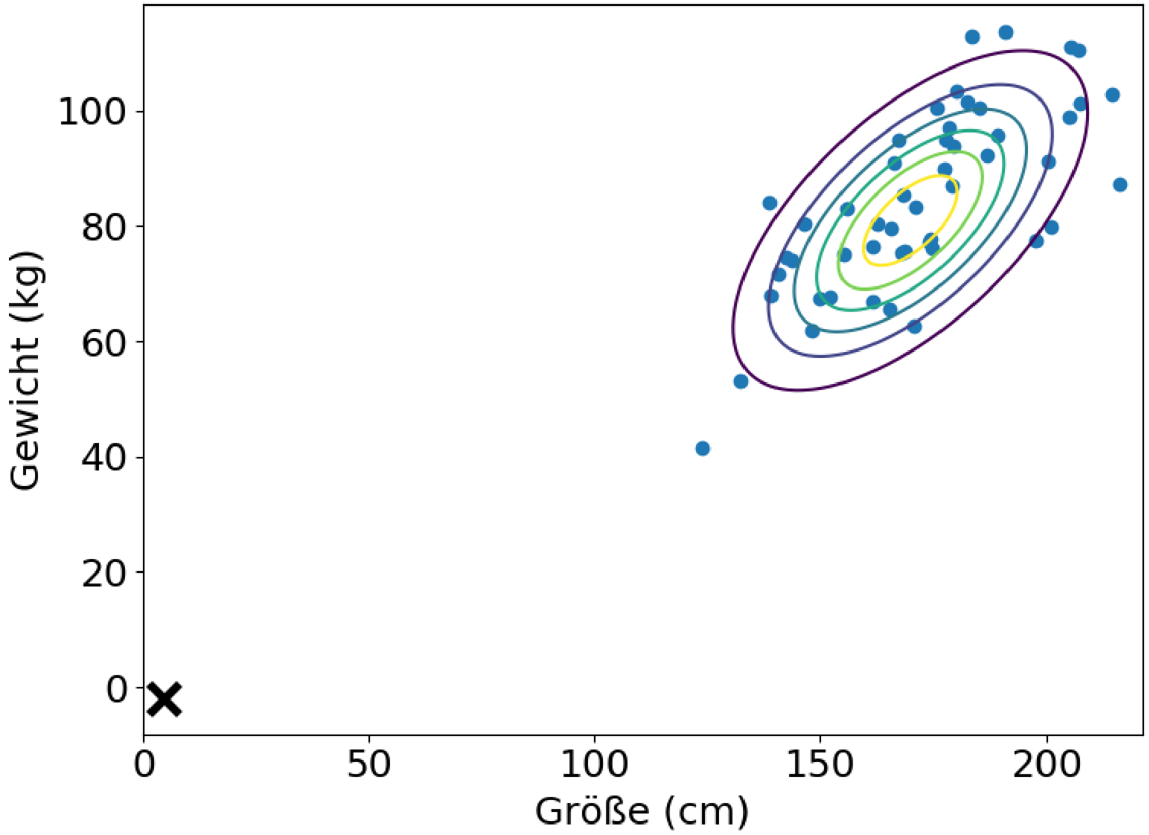
- Maximizing the LL is equivalent to minimizing the MSE!

Issue: Outlier Detection



- Regression models will give you an answer, even if the input makes no sense
- There is no good way to quantify uncertainty
- Predicting sigma does not help much

Discriminative vs. Generative Models



- Generative models aim to model the representation over the entire data, i.e. p(x, y) instead of p(y | x)
- This allows answering a variety of additional queries
- For instance, we can evaluate the input likelihood to detect outliers

$$p(x) = \int p(y,x)dy$$

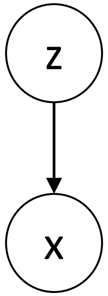
Inference Modes

- Given a joint probability $p(X_1, \ldots, X_n)$, we can, in principle, do the following things:
  - Sample new examples $\mathrm{X} \sim p(X_1, \ldots, X_n)$
  - Compute any marginal probability $\mathrm{p}(X_i) = \iint p(X_1, \ldots, X_n)$
  - Compute any conditional probability $p\left(X_i \middle| X_j\right) = \frac{p\left(X_i, X_j\right)}{p(X_j)}$
  - Find the most likely configuration given evidence $max_{X_i}\, p(X_i|X_j)$
- Another application: unsupervised learning
  - Can we learn the latent factors underlying the data, without expensive labeling?

Inference Modes: Examples

- Sample queries:
  - P(traffic = High | time=4am, location=A5)
  - $\max_{diagnosis}$ P(Condition=diagnosis| Age=80, chest_pain=True)
  - P(obstacle_type=car | Image)
- Challenge: how to represent an learn these distributions such that they are
  - expressive enough to model the data
  - allow for the computation of these queries

Latent Variable Models

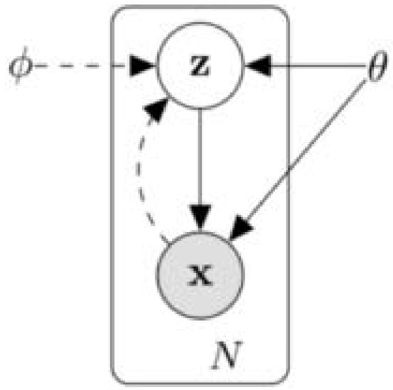


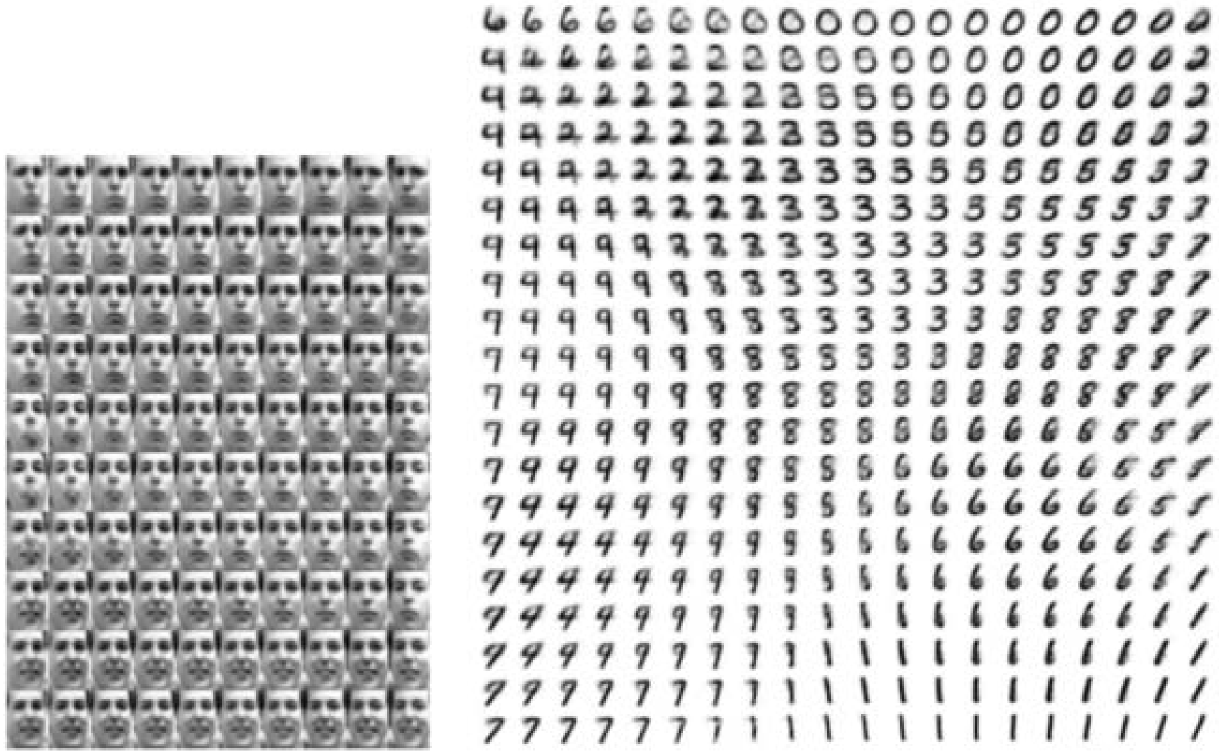
$$z \sim \mathcal{N}(0, 1)$$

$$x = f(z)$$

- Straightforward way for representing complex probability distributions p(x)
- How to learn f though?

Variational Autoencoders





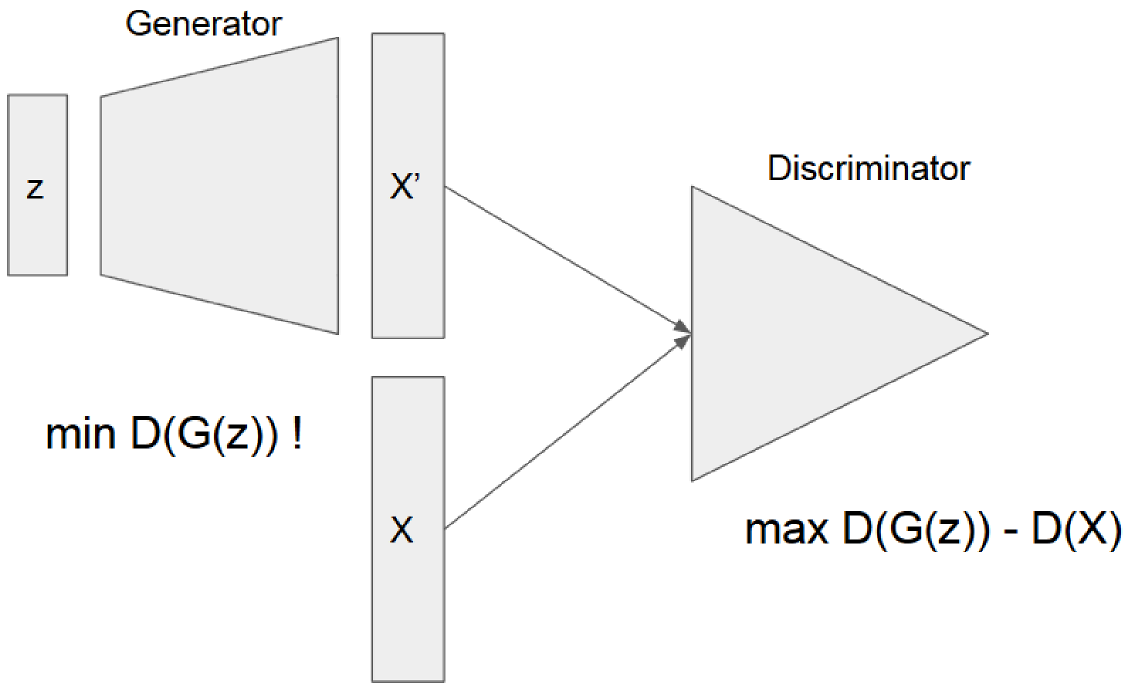
(a) Learned Frey Face manifold

(b) Learned MNIST manifold

[Kingma & Welling, 2014], [Rezende, Mohamed & Wierstra, 2014]

Generative Adversarial Nets

Setup learning as a two player game:



[Goodfellow et al., 2014]

[Karras et al., 2018]

Image Interpolation with GANs

[http://theo.io/blog/2018/11/13/biggan-interpolations/]

- Find $z_1, \ldots, z_n$ corresponding to given images $x_1, \ldots, x_n$
- Trace curve along the z's
- Draw the corresponding images

Autoregressive Models

- Assume order among variables $X_1, \dots, X_n$ and apply the chain rule
- $p(X_1, \dots, X_n) = p(X_1)p(X_2|X_1)p(X_3|X_2, X_1) \dots p(X_n|X_{n-1}, \dots, X_1)$
- Represent conditional distributions via neural nets as in regression
- Maximize likelihood directly

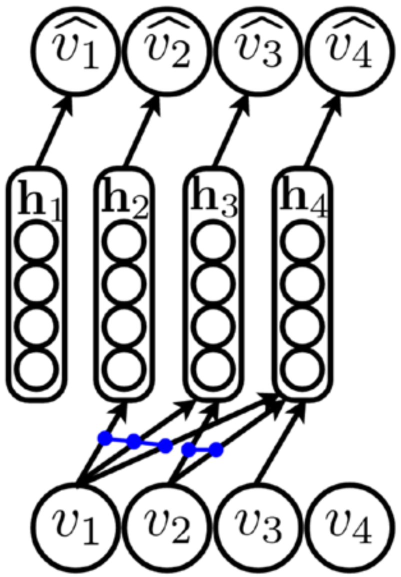
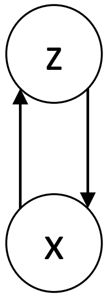


Figure 1. Image completions sampled from a PixelRNN.

[Larochelle and Murray, 2011]

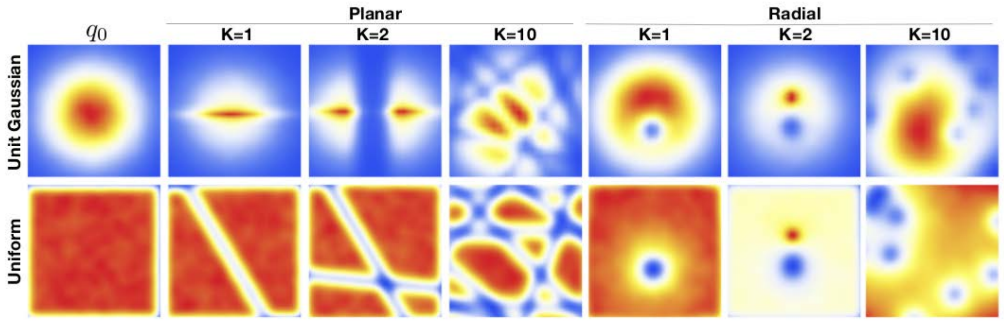
Normalizing Flows



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

$x = f(z)$

- Use bijective function f to transform random variables
- We can compute the likelihood directly via the change of variables formula



$$p(x) = p\big(Z = f^{-1}(x)\big) \left| \det \frac{\partial f(z)}{\partial z} \right|^{-1}$$

[Rezende et al., 2016]

Inference?

	GANs	**VAEs**	**NADEs**	**Flows**
Sampling	✓	✓	✓	✓
Density				
Marginals				
Conditionals				
Max (MAP)				

Inference?

	GANs	VAEs	NADEs	Flows
Sampling	✓	✓	✓	✓
Density	✗	✗ (✓)	✓	✓
Marginals				
Conditionals				
Max (MAP)				

Inference?

	GANs	VAEs	NADEs	Flows
Sampling	✓	✓	✓	✓
Density	✗	✗ (✓)	✓	✓
Marginals	✗	✗	✗ (✓)	✗ (?)
Conditionals				
Max (MAP)				

Inference?

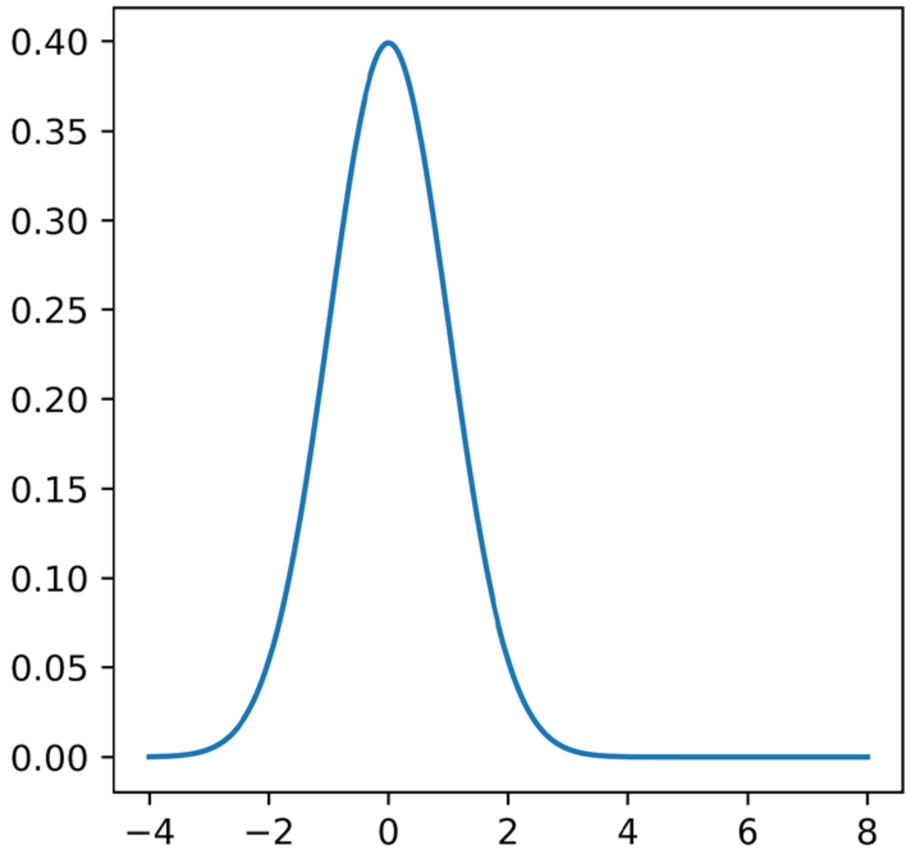
	GANs	VAEs	NADEs	Flows
Sampling	✓	✓	✓	✓
Density	✗	✗ (✓)	✓	✓
Marginals	✗	✗	✗ (✓)	✗ (?)
Conditionals	✗	✗	✗ (✓)	✗ (?)
Max (MAP)				

Inference?

	GANs	VAEs	NADEs	Flows
Sampling	✓	✓	✓	✓
Density	✗	✗ (✓)	✓	✓
Marginals	✗	✗	✗ (✓)	✗ (?)
Conditionals	✗	✗	✗ (✓)	✗ (?)
Max (MAP)	✗	✗	✗ (✓)	✗ (?)

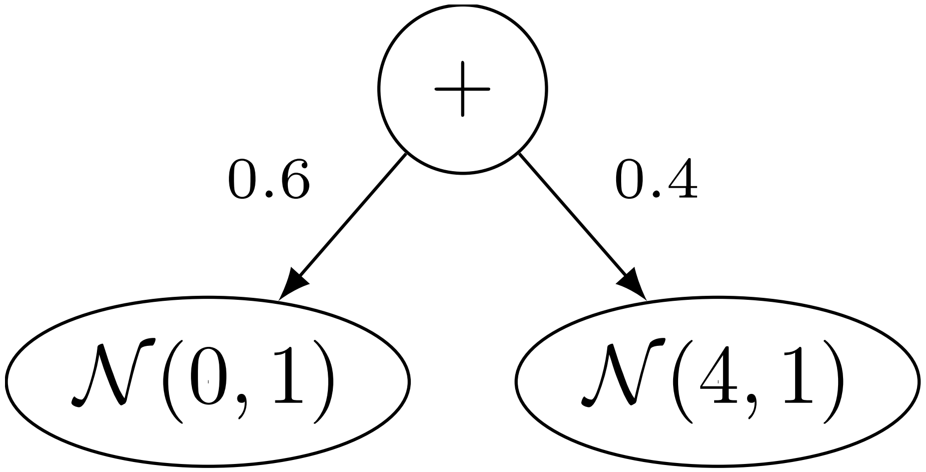
SPNs Compose Primitive Distributions…

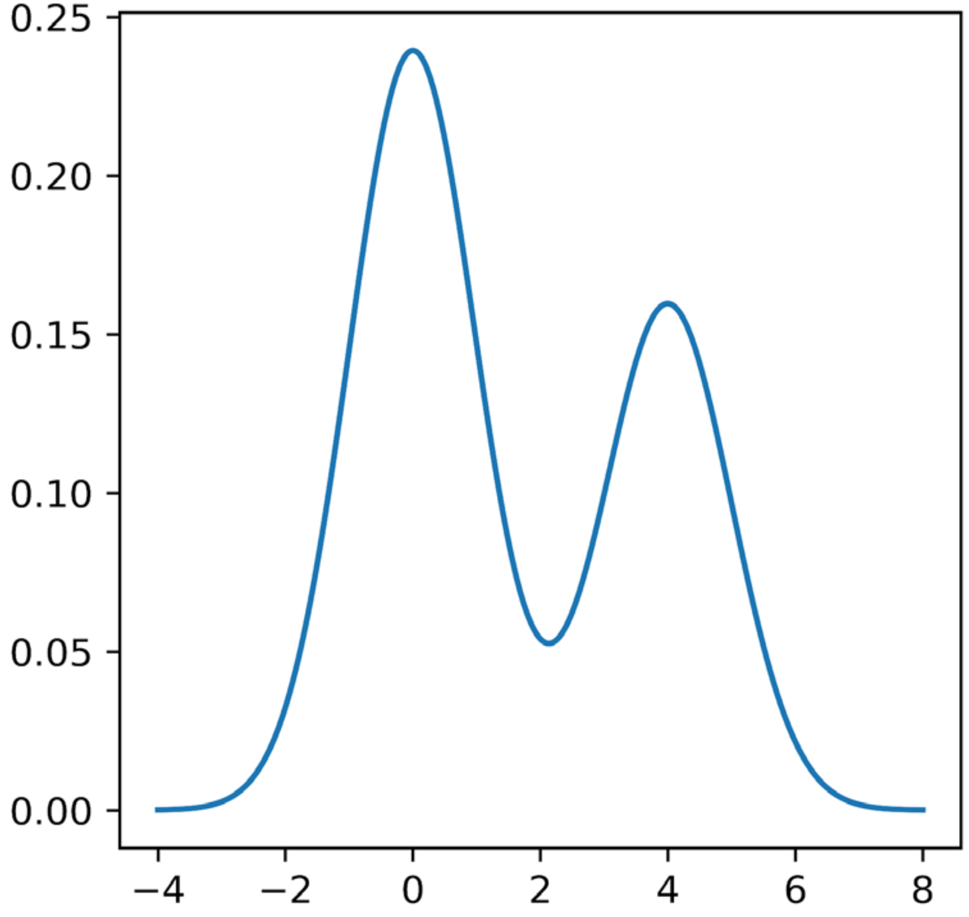
$\mathcal{N}(0,1)$



[Poon, Domingos. Sum-Product Networks: A New Deep Architecture. *UAI*, 2011]
[Adnan Darwiche. A Differential Approach to Inference in Bayesian Networks. *JACM*, 2003]

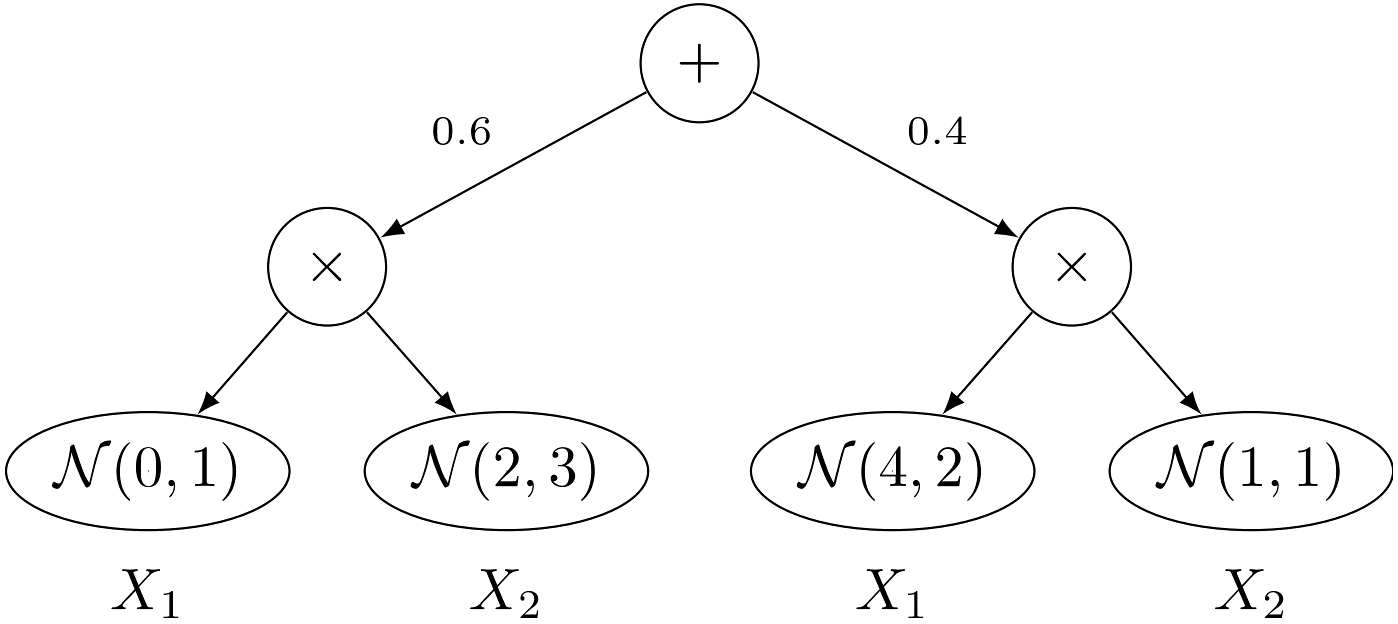
…Using Sums…

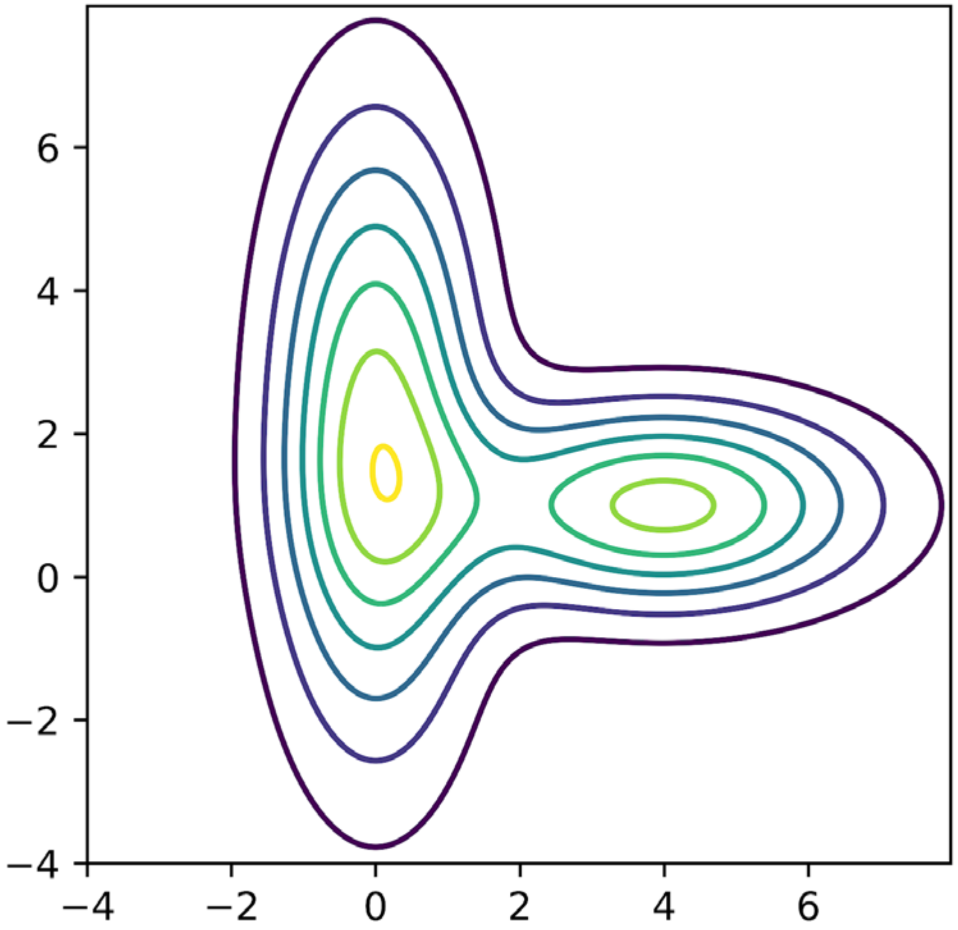




[Poon, Domingos. Sum-Product Networks: A New Deep Architecture. *UAI*, 2011]
[Adnan Darwiche. A Differential Approach to Inference in Bayesian Networks. *JACM*, 2003]

… And Products

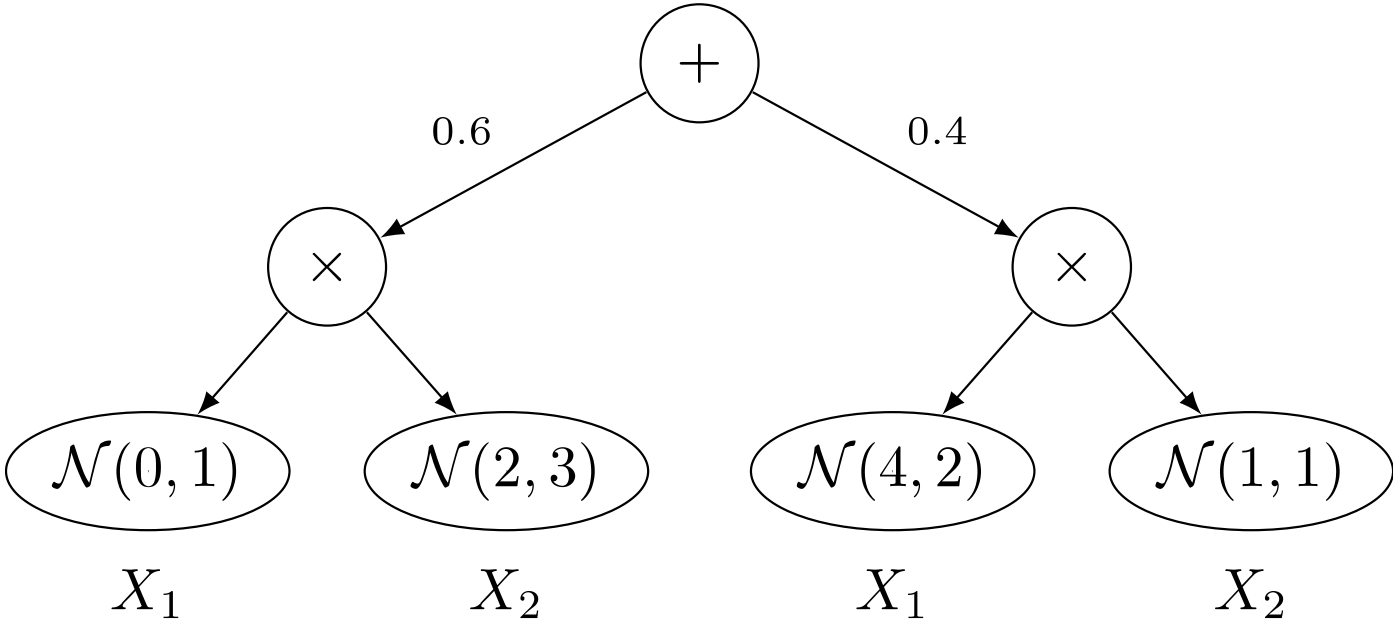


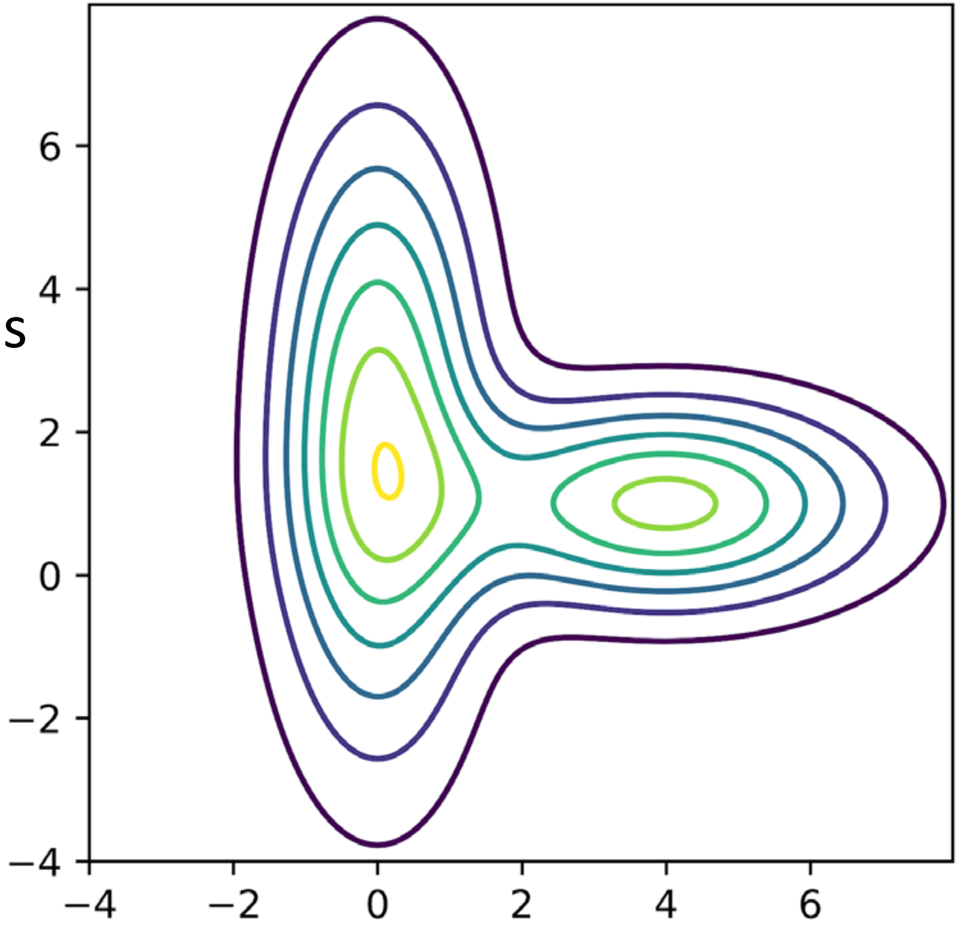


[Poon, Domingos. Sum-Product Networks: A New Deep Architecture. *UAI*, 2011]
[Adnan Darwiche. A Differential Approach to Inference in Bayesian Networks. *JACM*, 2003]

Validity

- Each node represents a distribution
- Two conditions ensure validity:
  - **Completeness**: sum children – same scope
  - **Decomposability**: product children – disjoint scopes



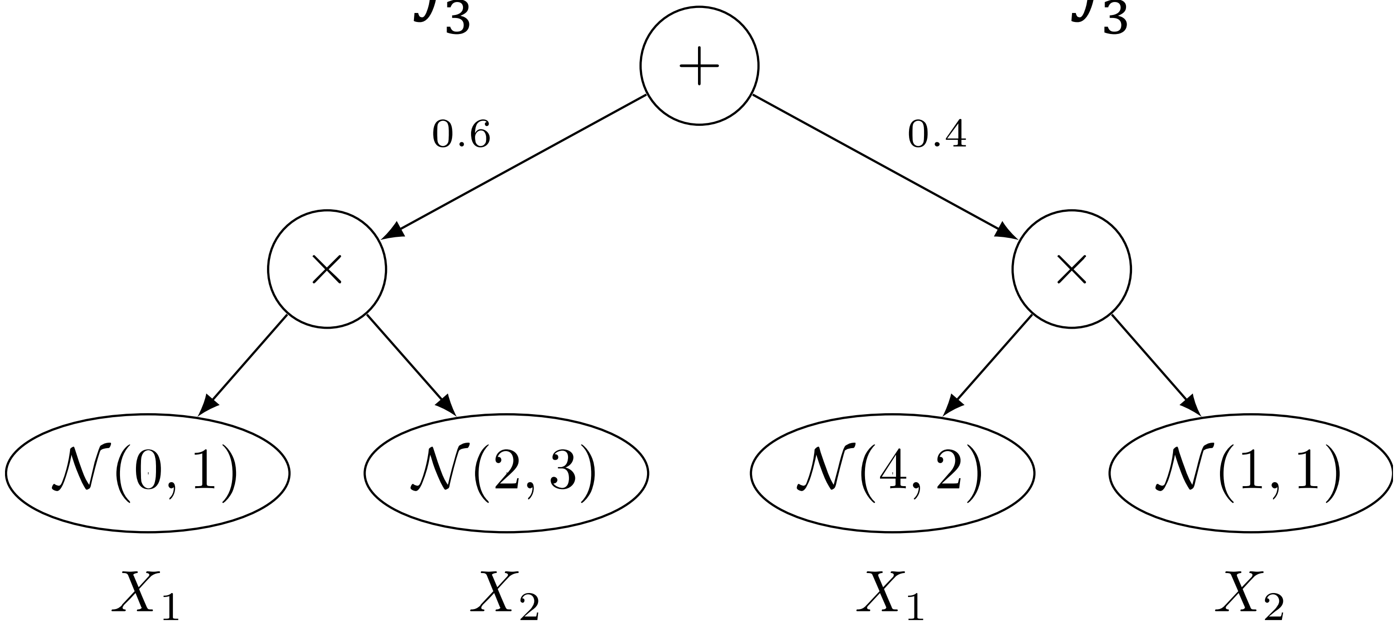


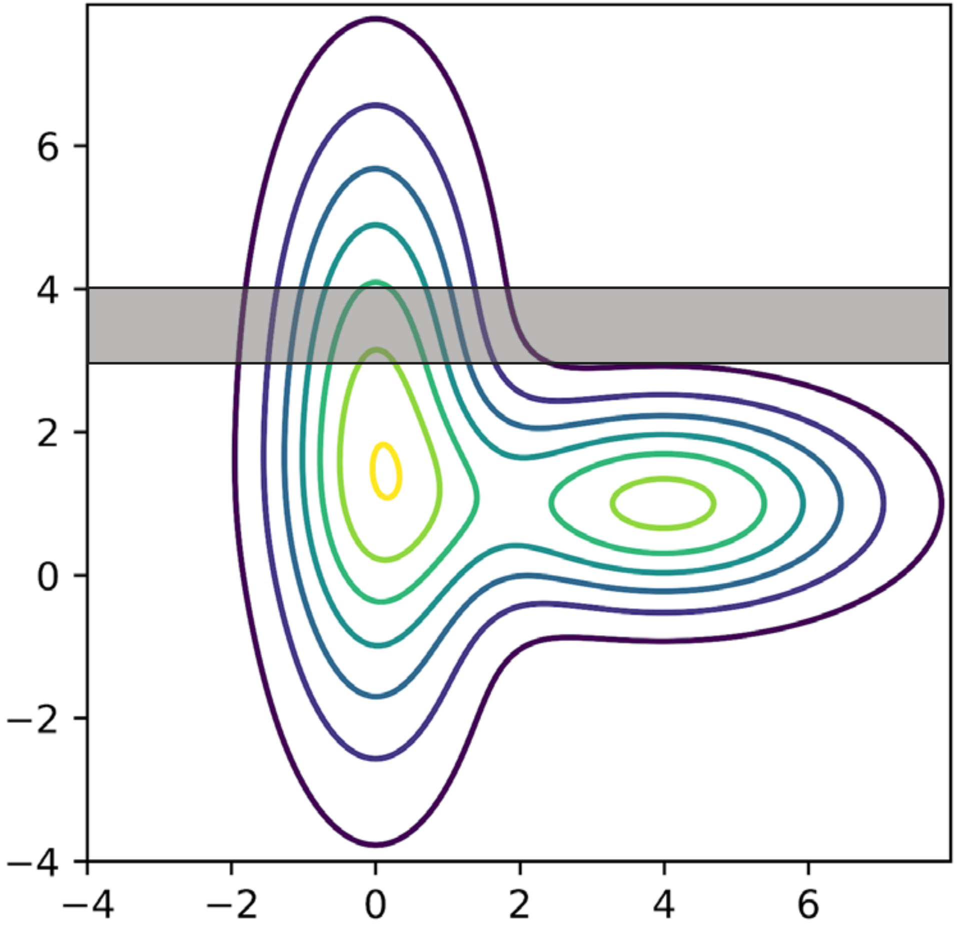
[Poon & Domingos, 2011]

Marginal Inference

$$p(3 < X_2 < 4) = \int_3^4 \int p(X_1, X_2)\, dX_1\, dX_2$$

$$= 0.6 \cdot 1 \cdot \int_3^4 N(2,3) + 0.4 \cdot 1 \cdot \int_3^4 N(1,1)$$





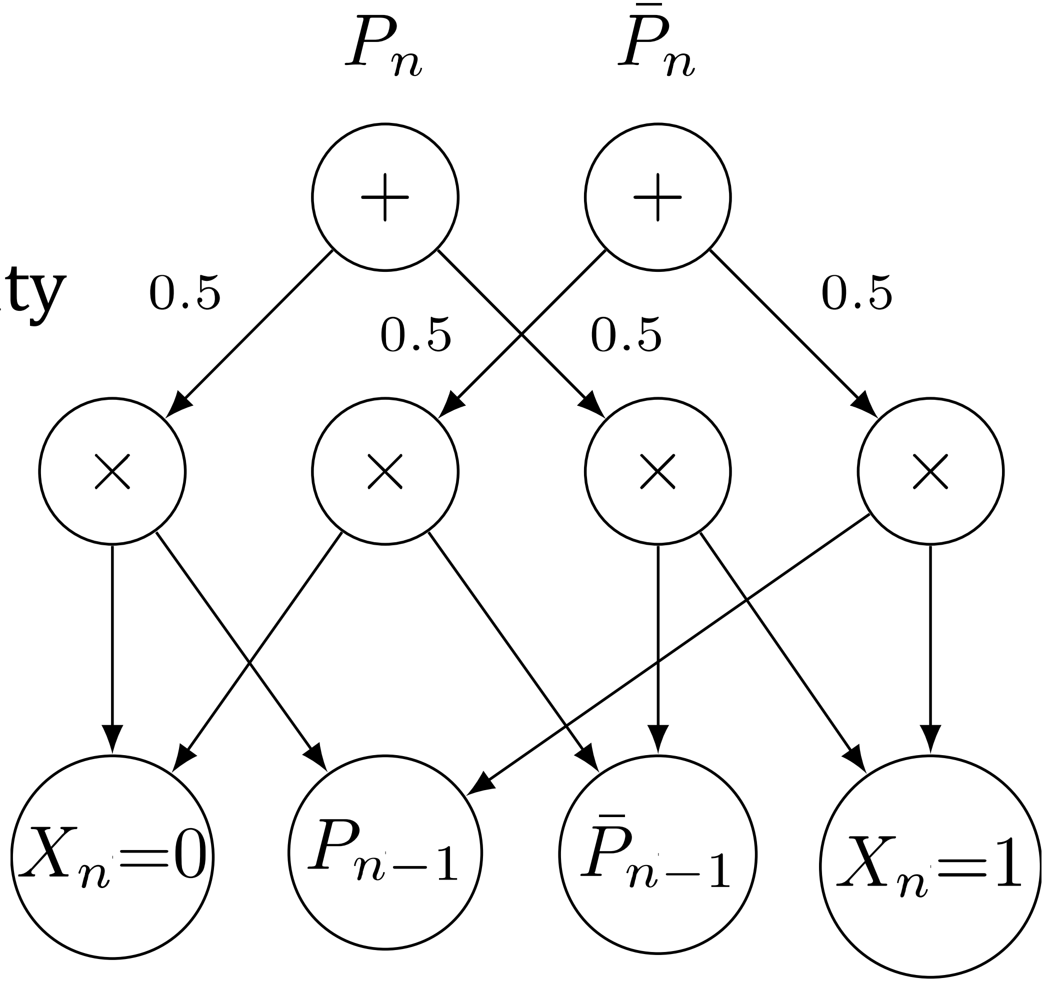
[Poon & Domingos, 2011]

SPNs vs. Mixture Models

- Consider the parity distribution

$$P_n(X_1, \dots, X_n) = \begin{cases} {}^{1}/_{Z}, & X \text{ has even parity} \\ 0 & \text{else} \end{cases}$$

- A shallow model will need an exponential number of mixture components
- Compact linear representation available at linear depth

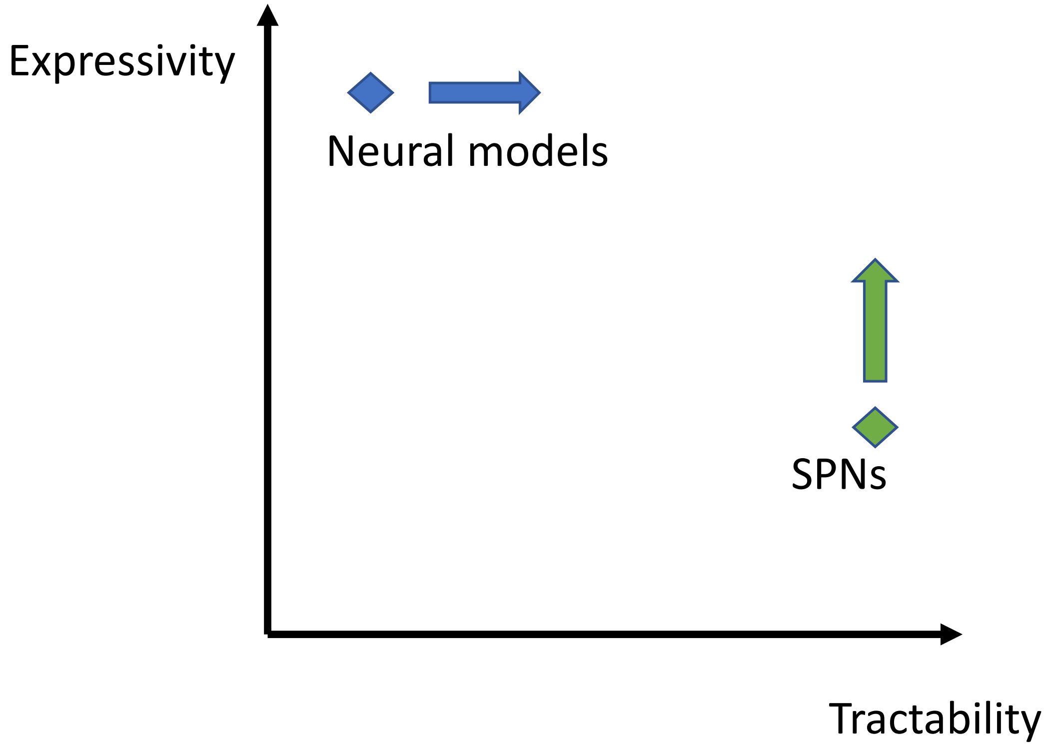


[Poon & Domingos, 2011]

Inference.

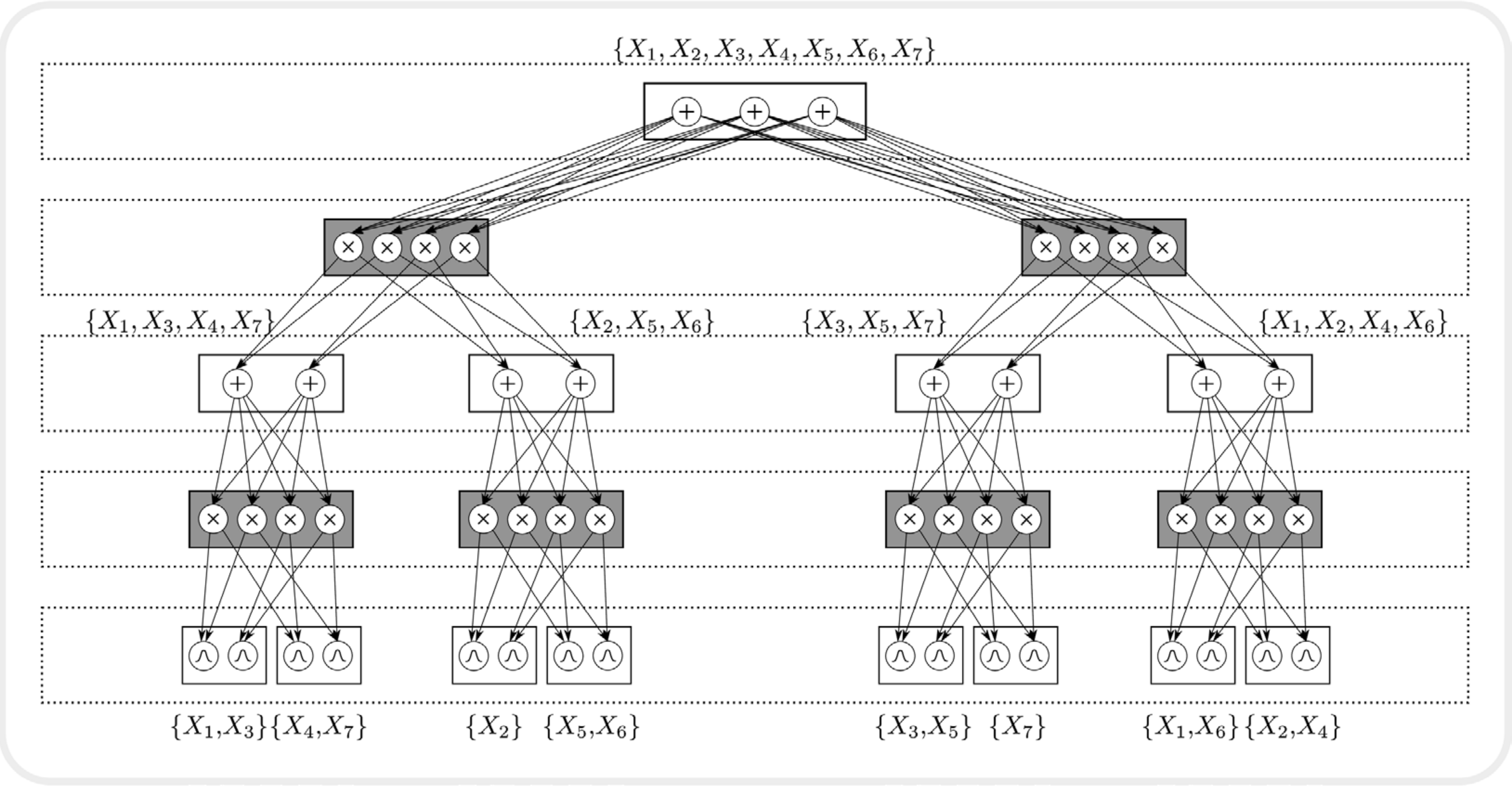
	GANs	VAEs	NADEs	Flows	SPNs
Sampling	✓	✓	✓	✓	✓
Density	✗	✗ (✓)	✓	✓	✓
Marginals	✗	✗	✗ (✓)	✗ (?)	✓
Conditionals	✗	✗	✗ (✓)	✗ (?)	✓
Max (MAP)	✗	✗	✗ (✓)	✗ (?)	✗ (✓)

Expressivity vs. Tractability



- Goal: Scale up SPNs to the size of deep learning models
- Enable tradeoffs and hybrid approaches

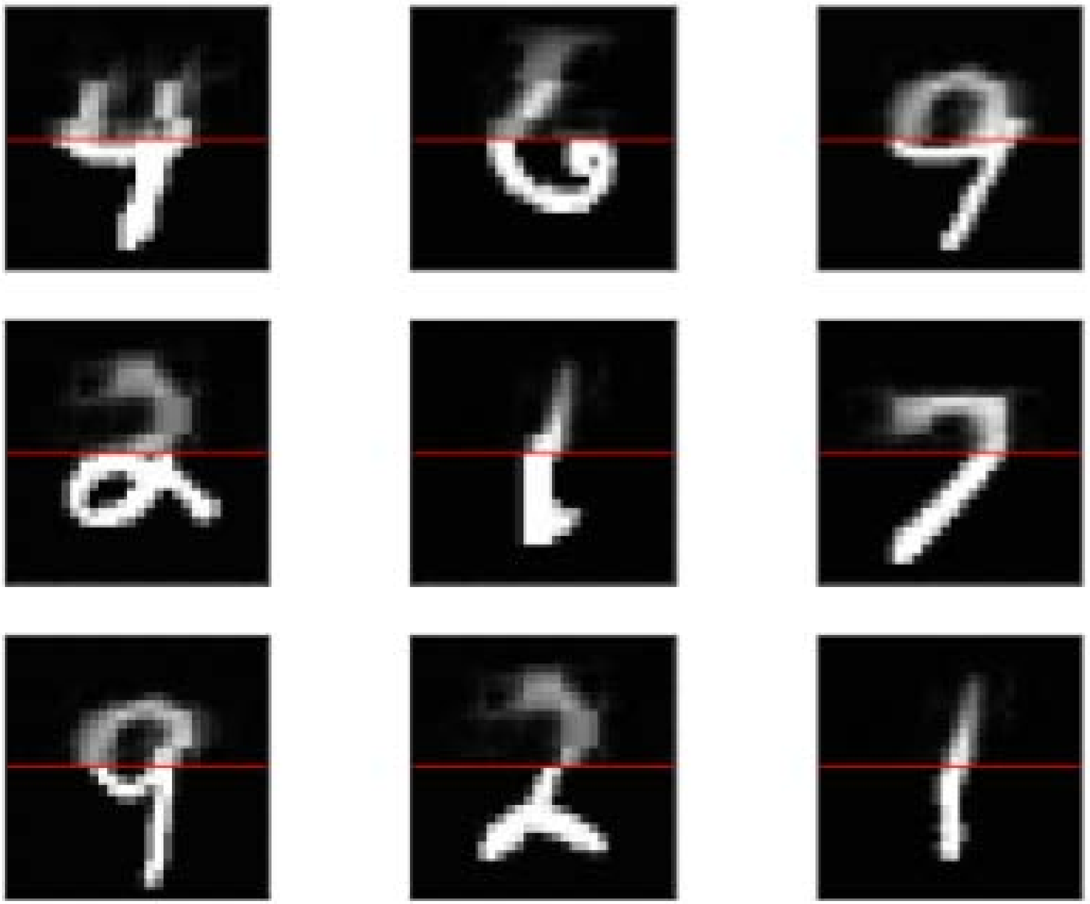
Random Sum-Product Networks [UAI'19]



Results: Classification (Test Accuracies)

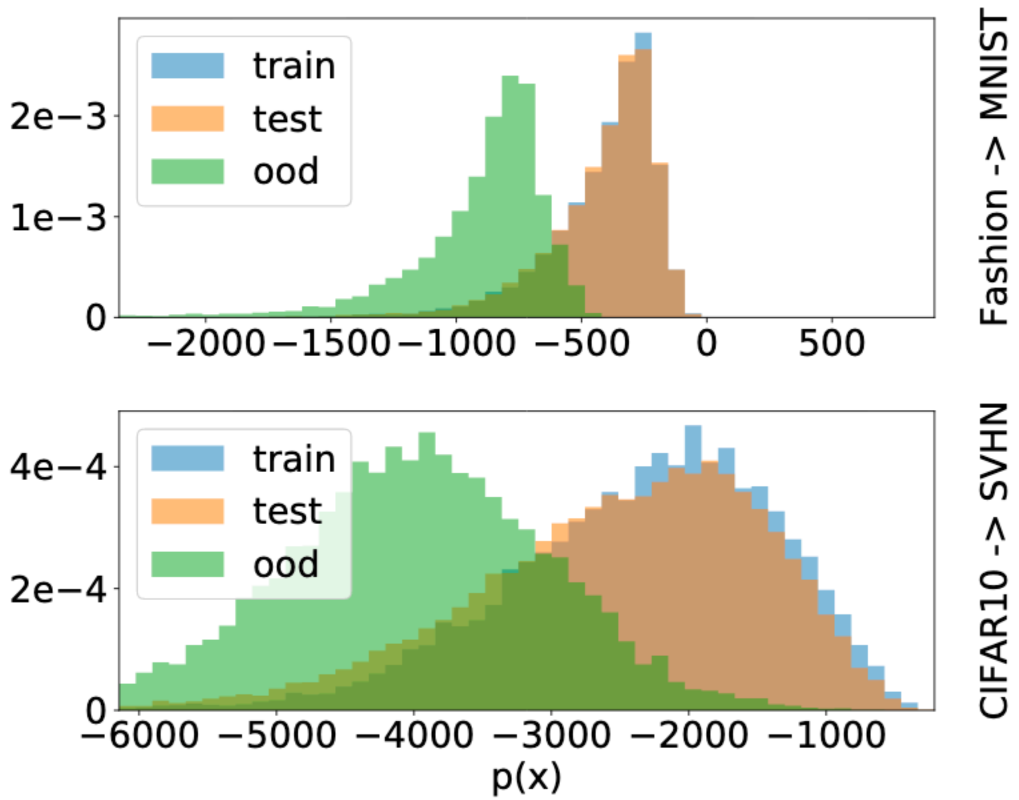
dataset	GMM	RAT-SPN	MLP	MLP+
mnist	97.37	○**98.29**	98.05	○98.52
f-mnist	88.08	89.43	**89.89**	90.63
imdb	○75.65	○**75.90**	○75.72	○75.83
theorem	○55.64	○55.47	○**57.76**	○56.21
20ng	47.61	○**48.49**	○**48.49**	○48.97
higgs	74.14	73.82	**76.36**	76.45
wine	○77.21	○77.14	○**77.83**	○79.45

Image Completion



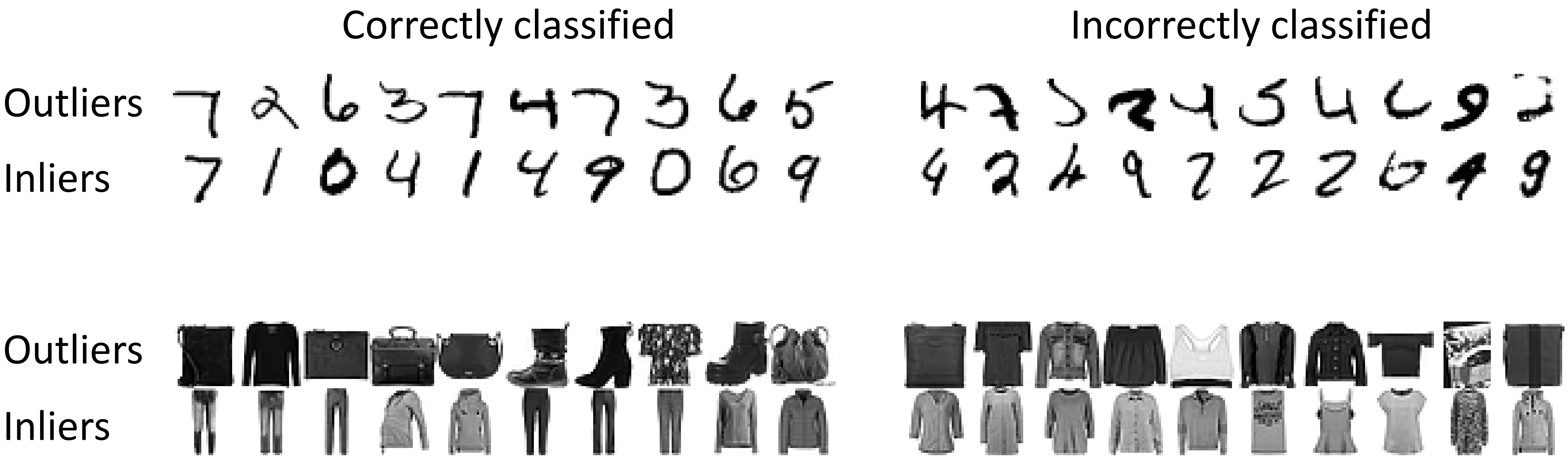
- Bottom half and label given
- Estimate top half using approximate MAP inference

Detecting Anomalies



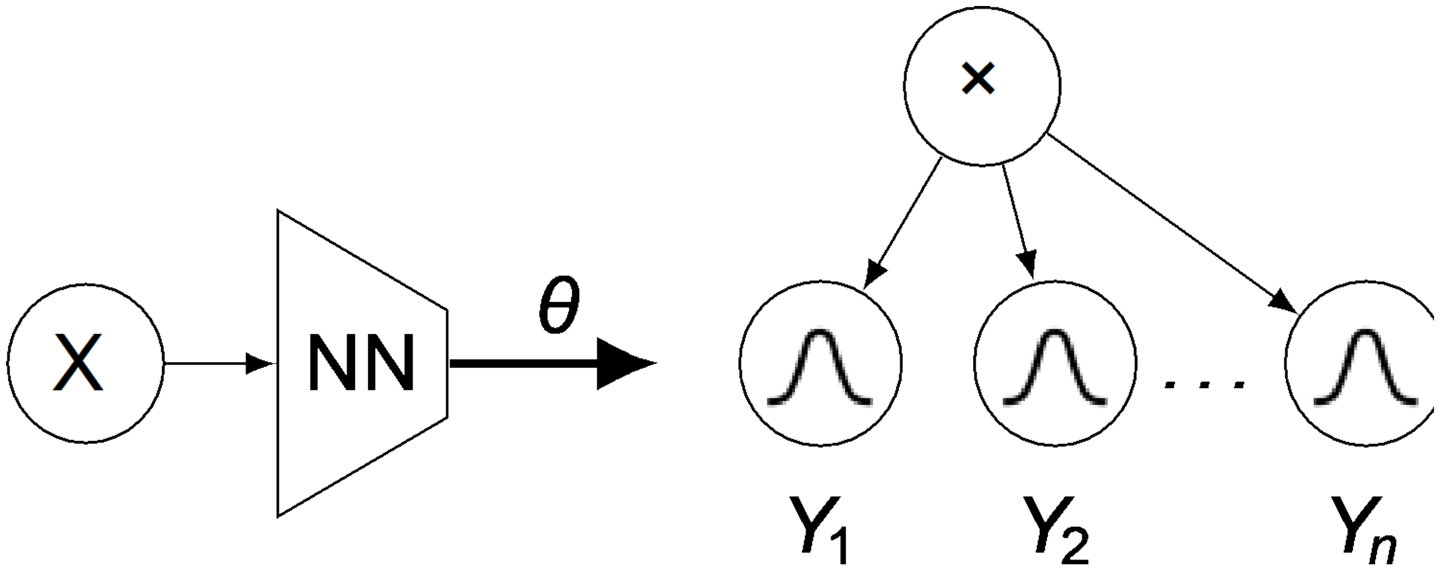
Results for Random SPNs

Inliers / Outliers



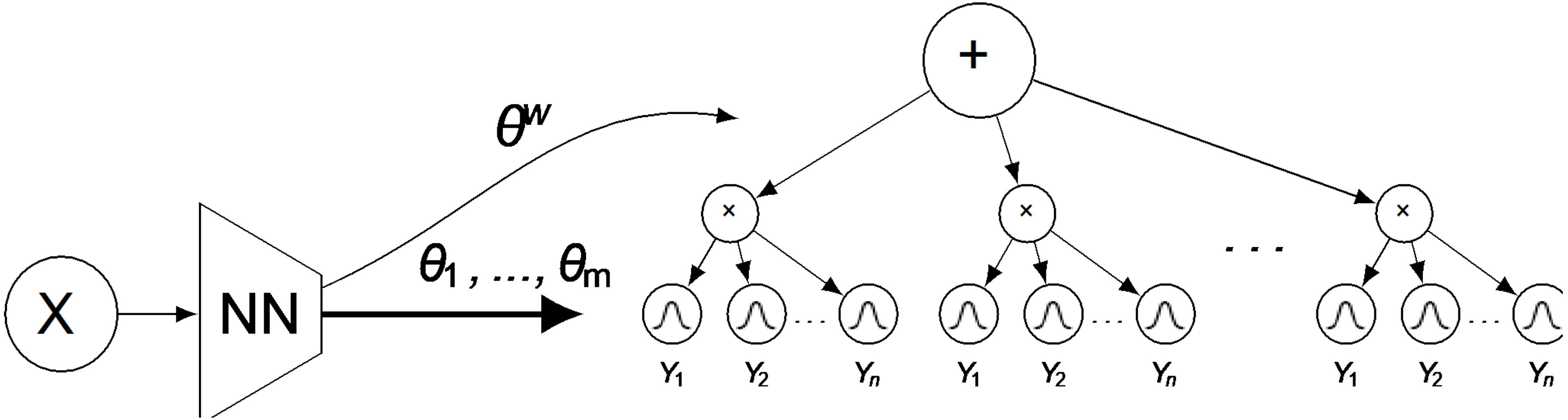
Conditional Density Estimation: Mean Field

- Estimate $P(Y_1, \dots, Y_n \mid \boldsymbol{X})$
- Common approach: $P(\boldsymbol{Y} \mid \boldsymbol{X}) = P(\boldsymbol{Y}\,;\,\boldsymbol{\theta}),\;\; \boldsymbol{\theta} = f(\boldsymbol{X})$
- Often, mean field assumption is made: $P(\boldsymbol{Y}\,;\,\boldsymbol{\theta}) = \prod P(Y_i \mid \boldsymbol{\theta})$



Conditional Mixture Models

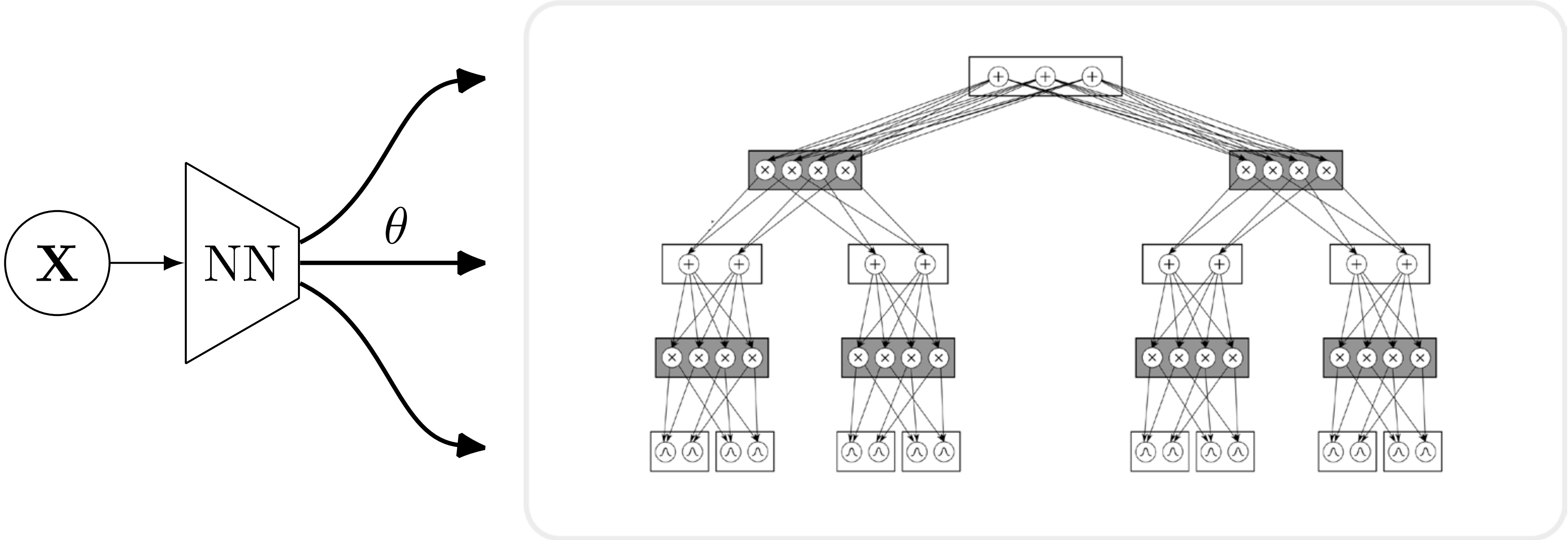
- Classic idea for more complex distributions: Output mixtures!



[Michael Jordan, Robert Jacobs. Hierarchical Mixtures of Experts and the EM Algorithm, *Neural Computation,* 1994]
[Christopher Bishop. Mixture Density Networks, *Technical Report*, 1995]

Conditional SPNs

- Predict leaf and sum weights using neural network
- Optimize conditional LL



[Shao et al., Conditional Sum-Product Networks:
Imposing Structure on Deep Probabilistic Architectures. arXiv preprint 1905.08550, 2019]

CSPN Results

- Multilabel image classification
- Same NN architecture, different conditional distributions

	CLL			ACCURACY		
	MF	MDN	CSPN	MF	MDN	CSPN
MNIST	-0.70	-0.61	**-0.54**	74.1%	76.4%	**78.4%**
FASHION	-0.95	-0.73	**-0.70**	73.4%	73.7%	**75.5%**
CELEBA	-12.1	-11.6	**-10.8**	86.6%	85.3%	**87.8%**

Thank you!