The Rise of Tractable Circuits

From Cryptography to Continuous Generative Models

Robert Peharz
Graz University of Technology

Workshop on Probabilistic Circuits and Logic
Simons Institute, Berkeley, 19th Otober 2023

The Force is Strong in Tractable Circuits

Probabilistic Circuits

- representations of high-dimensional probability distributions
- probability = optimal and consistent reasoning under uncertainty ($\approx$ half way to AI)
- circuit structure enables **exact** probabilistic reasoning

Logic Circuits

- representations of large (propositional) logical formulas
- circuit structure enables **exact** logical reasoning
- symbolic language suitable for humans

Circuits as Neural Nets

- connects with machine learning

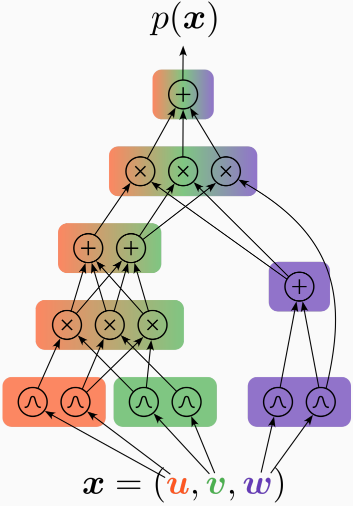
Leaves are distributions (L), internal nodes are sums (S) or products (P).

Smoothness: inputs of sum node are over same scope—means that sums are proper mixtures.

Decomposability: inputs of products are over disjoint scopes—means that products are proper factorizations.

Structured Decomposability: products over same scope factorize the same way

Determinism: at most input to each sum node is non-zero



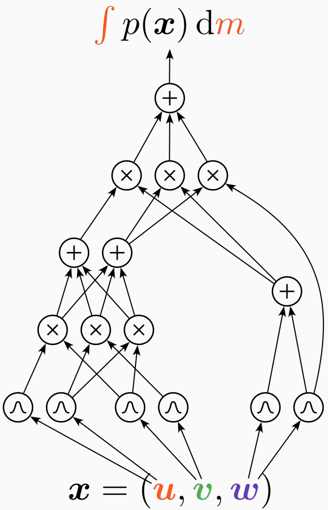
Logic vs. Probabilistic Circuits

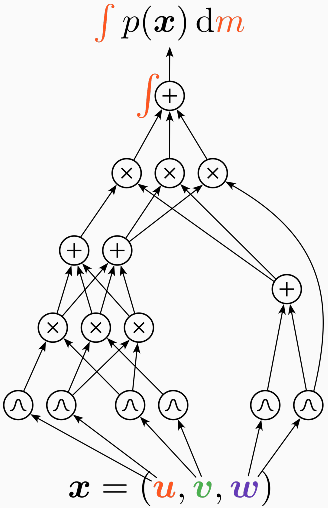
Inference in PCs

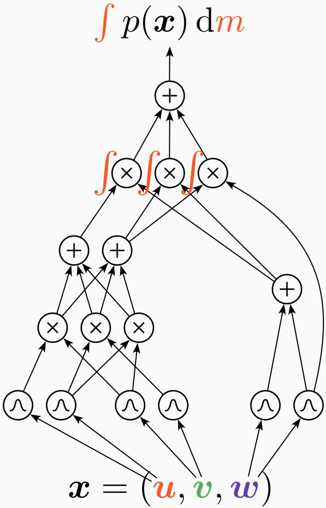
- **smoothness** and **decomposability** enable tractable **marginalization** and **conditioning**
- **determinism** enables tractable **maximization**
- **structured decomposability** (**compatibility**) enables **circuit multiplication**
- ...

Assume we want to marginalize a variable *M*, which is contained in ***U***.

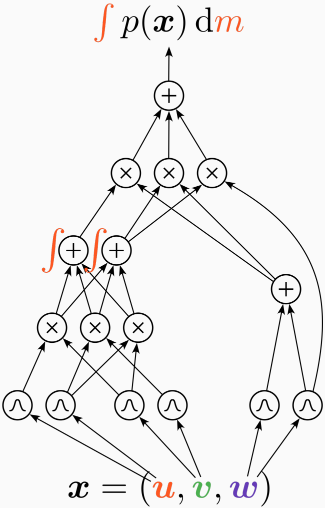
The core of probabilistic inference.

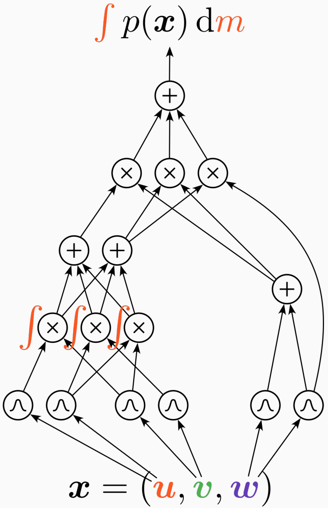



$\int p(\boldsymbol{x})\,\mathrm{d}m$
$\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$

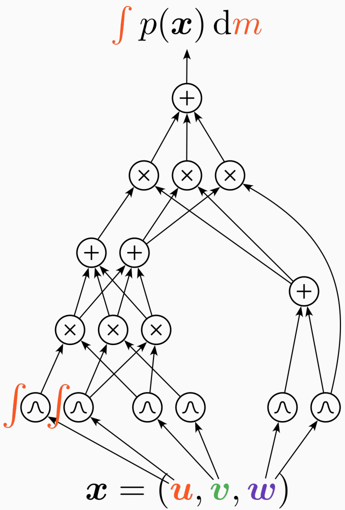

$\int p(\boldsymbol{x})\,\mathrm{d}m$
$\int$ $\int$ $\int$
$\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$

Due to decomposability, *M* appears only in one child of each product node



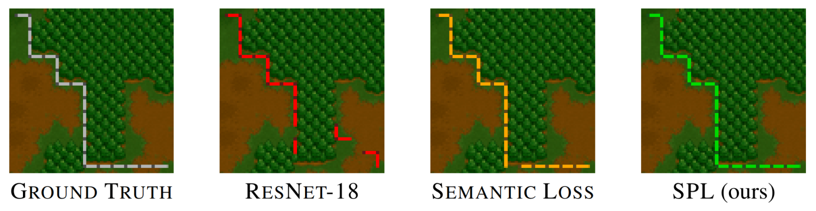

$\int p(\boldsymbol{x})\,\mathrm{d}m$
$\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$

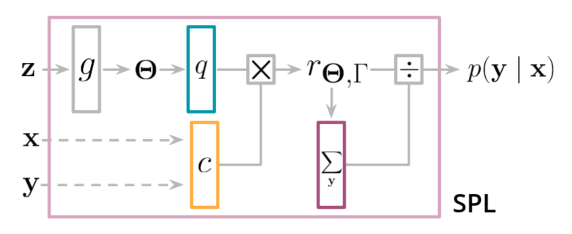
Reduces to marginalization at leaves & standard forward pass!



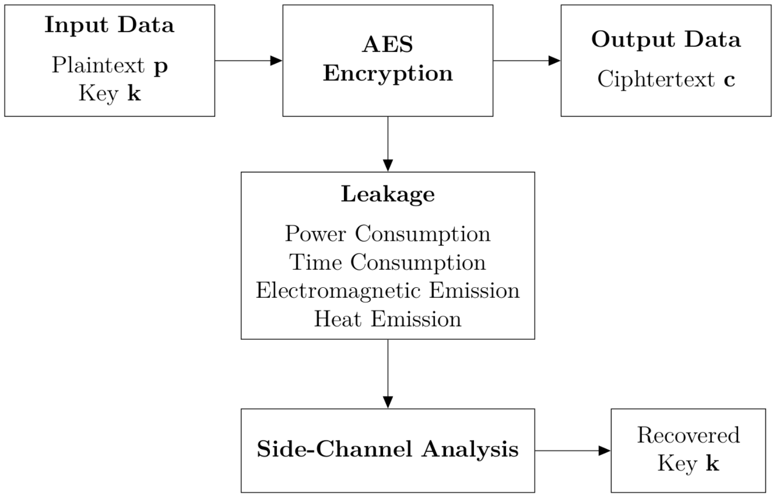
Circuits for Neuro-Symbolic AI [Ahmed et al., NeurIPS'22]

Classical neural nets don't know logical structure, so let's tell them...

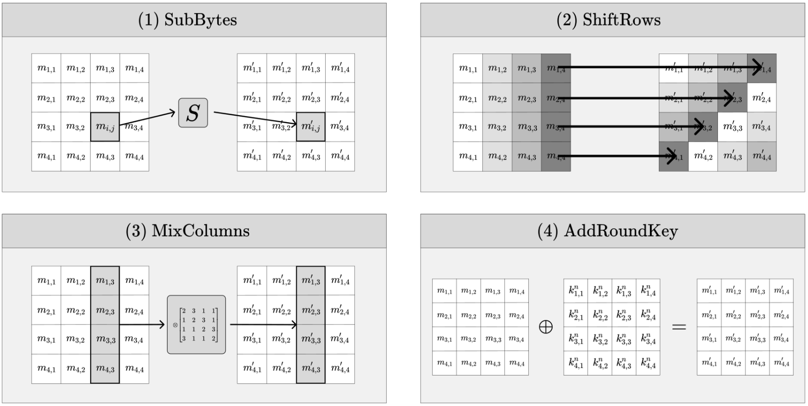




Crypto algorithms are save – unless executed on a physical device

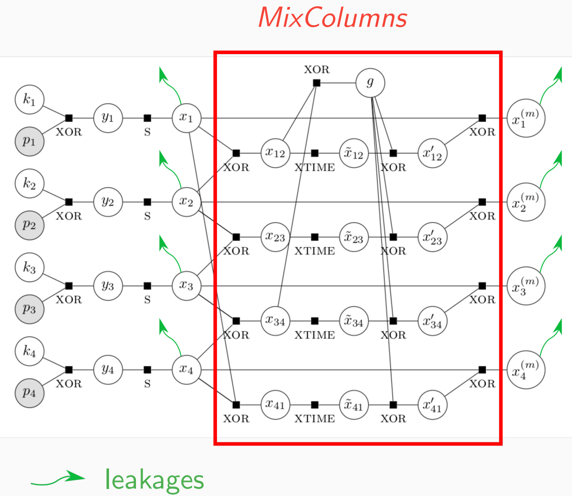


Advanced Encryption Standard (AES-128)



- uses a 128 bit **key** to convert a **plain text** into a **cypher**
- 10 rounds of *SubBytes*, *ShiftRows*, *MixColumns* (until round 9), *AddRoundKey*

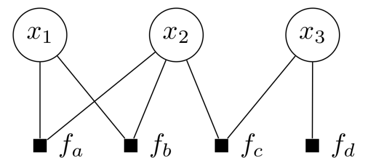
Soft Analytic Side Channel Attacks (SASCA)



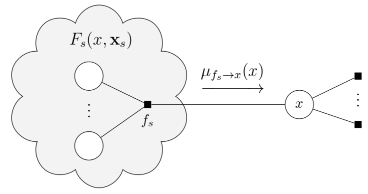
SASCA: **loopy belief propagation** to infer key k_1, k_2, k_3, k_4

surprisingly effective and state of the art!

factor graphs: represents (unnormalized) distributions as $\prod_f f(\boldsymbol{X})$

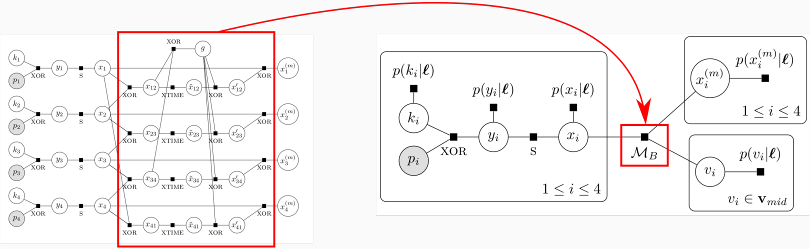


belief propagation: computes marginals via message passing



Exact on trees, but very limited guarantees on loopy graphs...

Exact SASCA with Circuits



- compile *MixColumns* to an SDD (structured decomposable)
- compile leakage distributions (256 states) to **compatible** PSDDs
- apply circuit multiplication, yielding a “big” joint over all variables
- infer key via tractable marginal query – this is still message passing, but on a high-dimensional tree (exact)

Dirty Tricks

- circuit multiplication is quadratic
- 9 circuits involved, so this didn't take off
- thus, approximate the leakage distributions
    1. **assume conditional independence**
       changed the distributions too much, led to inferior performance
    2. **sparsify**
       many values close to zero; take states corresponding to $1 - \epsilon$ of the mass, set the rest to zero, re-normalize
- with simplified leakage distributions, we indeed could perform exact inference

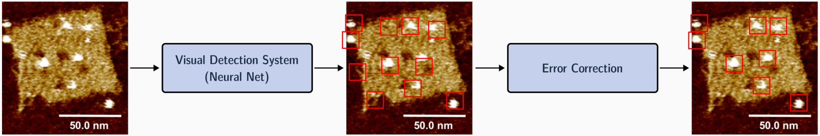
Results

Inference Method	Success Rate			
	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-8}$	$\varepsilon = 0$
Baseline	79.35%	79.57%	79.59%	**79.59**%
SASCA (3 BP iterations)	84.89%	84.88%	84.91%	**84.91**%
SASCA (50 BP iterations)	89.36%	90.45%	90.41%	**90.45**%
SASCA (100 BP iterations)	89.36%	90.27%	**90.69**%	90.52%
PSDD + MAR	93.40%	97.81%	**98.02**%	N/A
PSDD + MPE	93.67%	99.42%	**99.93**%	N/A

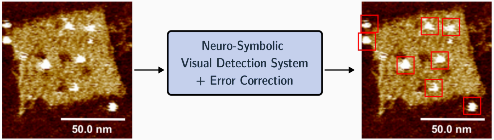
Towards DNA-based Storage

Using DNA origamis as "compact disc"

Funded by
the European Union



Integrated Pipeline



Advantages: more reliable, parameter-efficient, data-efficient

Continuous Generative Models and Probabilistic Circuits

Inference in Generative Models

Among generative models, PCs have excellent inference properties:

	GANs	VAEs	EBMs	Flows	ARMs	PCs
sampling	✓	✓	✓	✓	✓	✓
density	✗	✗	✗	✓	✓	✓
marginals	✗	✗	✗	✗	✗	✓
condition	✗	✗	✗	✗	✗	✓
moments	✗	✗	✗	✗	✗	✓
max (MAP)	✗	✗	✗	✗	✗	✓ (✗)
$\mathbb{E}$	✗	✗	✗	✗	✗	✓ (✗)

But, PCs usually have worse performance, in terms of log-likelihood, sample quality, . . .

One reason is the **tractability-expressiveness dilemma**

Another might be the “discrete nature” of PCs, while many successful generative models can be seen as **continuous mixtures**

Mixture Models

- **discrete mixtures**

$$p(\boldsymbol{x}) = \sum_{i=1}^{K} w_k \, p_i(\boldsymbol{x}) = \overbrace{\sum_{i=1}^{K} p(\boldsymbol{z} = i) \, p(\boldsymbol{x} \mid \boldsymbol{z} = i)}^{\text{latent variable interpretation}}$$

  - e.g. Gaussian mixtures, PCs
  - many tractable inference scenarios
- **continuous mixtures**

$$p(\boldsymbol{x}) = \int p(\boldsymbol{z}) \, p(\boldsymbol{x} \mid \boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}$$

  - VAEs, GANs, Flows, etc.
  - $p(\boldsymbol{z})$ usually simple, e.g. white Gaussian
  - $p(\boldsymbol{x} \mid \boldsymbol{z})$ via neural net—**continuity between $\boldsymbol{x}$ and $\boldsymbol{z}$**
  - usually intractable inference, due to high-dimensional integral

- Can we get best of both worlds?
- Continuous latent variables in PCs ("integral nodes")?
- Also, can we still have tractable inference, please?

- Can we get best of both worlds?
- Continuous latent variables in PCs ("integral nodes")?
- Also, can we still have tractable inference, please?

Arbitrary integrals are hard, but for low-dimensional $\boldsymbol{z}$

$$p(\boldsymbol{x}) = \int p(\boldsymbol{z})\, p(\boldsymbol{x} \mid \boldsymbol{z})\, \mathrm{d}\boldsymbol{z}$$

becomes "morally tractable." We might just apply good old **numerical integration**, such as **quadrature rules**:

$$\int p(\boldsymbol{z})\, p(\boldsymbol{x} \mid \boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \approx \sum_i w_{\boldsymbol{z}_i^*}\, p(\boldsymbol{z}_i^*)\, p(\boldsymbol{x} \mid \boldsymbol{z}_i^*)$$

The sum brings us back to circuit land!

Continuous Mixtures of Tractable Probabilistic Models

Alvaro H.C. Correia[1,*], Gennaro Gala[1,*],
Erik Quaeghebeur[1], Cassio de Campos[1], Robert Peharz[1,2]

- model distribution: $p(\boldsymbol{x}) = \int p(\boldsymbol{z})\, p(\boldsymbol{x} \mid \boldsymbol{z})\, \mathrm{d}\boldsymbol{z}$
- $p(\boldsymbol{z})$ is a low-dimensional white Gaussian
- $p(\boldsymbol{x} \mid \theta(\boldsymbol{z}))$ is a PC, whose parameters θ depend on $\boldsymbol{z}$ via a neural net
- we used pretty simple PC structures, such as complete factorized distributions and Chow-Liu trees

Results on 20 Binary Datasets

Dataset	BestPC	cm($\mathcal{S}_\mathsf{F}$)	cm($\mathcal{S}_\mathsf{CLT}$)	LO(cm($\mathcal{S}_\mathsf{CLT}$))	Dataset	BestPC	cm($\mathcal{S}_\mathsf{F}$)	cm($\mathcal{S}_\mathsf{CLT}$)	LO(cm($\mathcal{S}_\mathsf{CLT}$))
accid.	**-26.74**	-33.27	-28.69	-28.81	jester	-52.46	**-51.93**	-51.94	-51.94
ad	-16.07	-18.71	-14.76	**-14.42**	kdd	**-2.12**	-2.13	**-2.12**	**-2.12**
baudio	-39.77	**-39.02**	**-39.02**	-39.04	kosarek	-10.60	-10.71	-10.56	**-10.55**
bbc	-248.33	**-240.19**	-242.83	-242.79	msnbc	**-6.03**	-6.14	-6.05	-6.05
bnetflix	-56.27	-55.49	**-55.31**	-55.36	msweb	-9.73	-9.68	-9.62	**-9.60**
book	-33.83	-33.67	-33.75	**-33.55**	nltcs	**-5.99**	**-5.99**	**-5.99**	**-5.99**
c20ng	-151.47	-148.24	**-148.17**	-148.28	plants	-12.54	-12.45	**-12.26**	-12.27
cr52	-83.35	-81.52	**-81.17**	-81.31	pumbs	**-22.40**	-27.67	-23.71	-23.70
cwebkb	-151.84	-150.21	-147.77	**-147.75**	tmovie	-50.81	**-48.69**	-49.23	-49.29
dna	**-79.05**	-95.64	-84.91	-84.58	tretail	-10.84	10.85	-10.82	**-10.81**
Avg. rank	2.85	2.65	1.85	**1.75**					

Samples

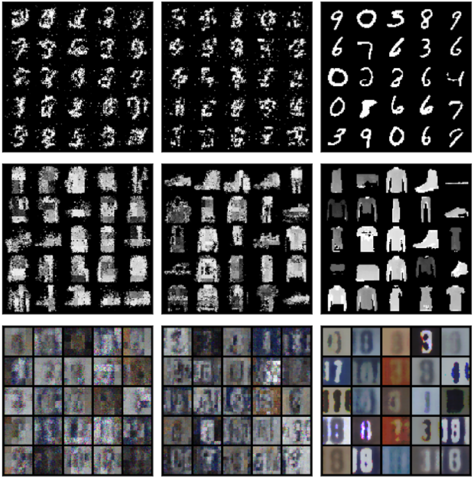
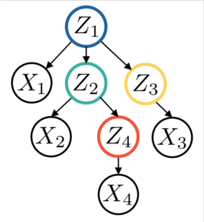
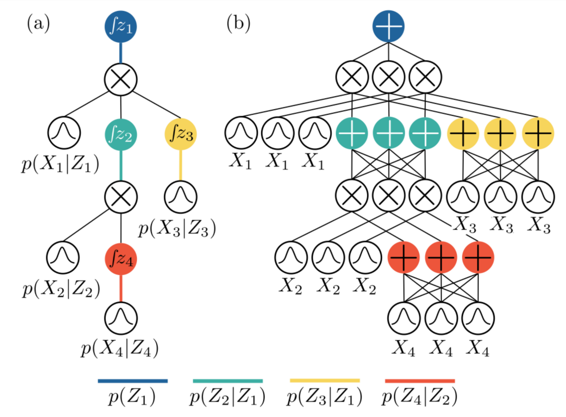


Figure 2: Samples from 'Small Einet' (left column), 'Big Einet' (middle column) and $\mathrm{cm}(\mathcal{S}_\mathsf{F})$ (right column).

Probabilistic Integral Circuits (PICs)

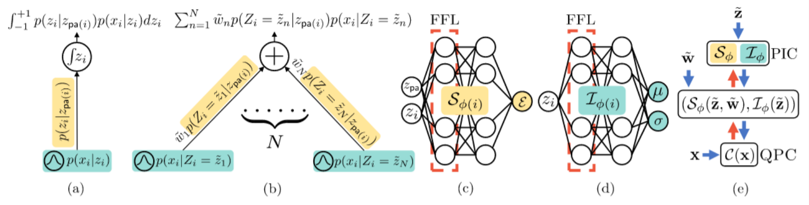
Have many integral nodes in PCs, i.e. local continuous mixtures (submitted).



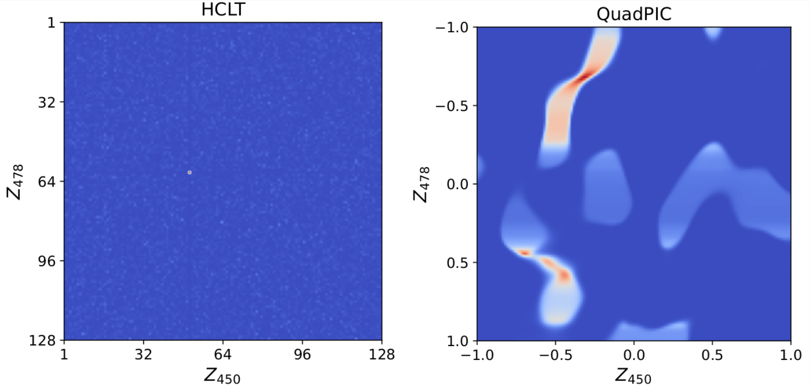


with G. Gala, C. de Campos, A. Vergari, E. Quaeghebeur

PIC Overview



Light-weight Energy-based Models



Results

	QPC	HCLT	Sp-PC	RAT	IDF	BitS	BBans	McB
MNIST	**1.18**	1.21	1.14	1.67	1.90	1.27	1.39	1.98
F-MNIST	**3.27**	3.34	3.27	4.29	3.47	3.28	3.66	3.72
EMN-MN	**1.66**	1.70	1.52	2.56	2.07	1.88	2.04	2.19
EMN-LE	**1.70**	1.75	1.58	2.73	1.95	1.84	2.26	3.12
EMN-BA	**1.73**	1.78	1.60	2.78	2.15	1.96	2.23	2.88
EMN-BY	**1.67**	1.73	1.54	2.72	1.98	1.87	2.23	3.14

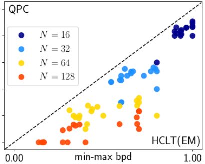


Figure 5: **QPCs systematically outperform PCs trained via EM or SGD.** Table (left): Best average test-set bpd for the MNIST-famility datasets. We compare against HCLT (Liu and Van den Broeck, 2021), SparsePC (Dang et al., 2022) RAT-SPN (Peharz et al., 2020), IDF (Hoogeboom et al., 2019), BitSwap (Kingma et al., 2019), BBans (Townsend et al., 2019) and McBits (Ruan et al., 2021). QPC results are in bold if better than HCLTs, whereas global best results are underlined. QPC and HCLT results are averaged over 5 different runs; the other results are taken from Dang et al. (2022). Scatter plot (right): bpd results for QPCs (y-axis) and HCLTs (x-axis) paired by B-N hyperparameter configuration and (min-max) normalized for every MNIST-family dataset. A similar trend occurs for binomial input units (cf. Appendix B).

Compatible vtrees