Probabilistic Circuits That Know What They Don't Know

TECHNISCHE UNIVERSITÄT DARMSTADT hessian.AI

Fabrizio Ventola*[1] Steven Braun*[1] Zhongjie Yu[1] Martin Mundt[1,2] Kristian Kersting[1,2,3,4]

[1]Dep. of CS, TU Darmstadt [2]hessian.AI [3]DFKI [4]Centre for Cognitive Science, TU Darmstadt

KompAKI DFKI

Overview

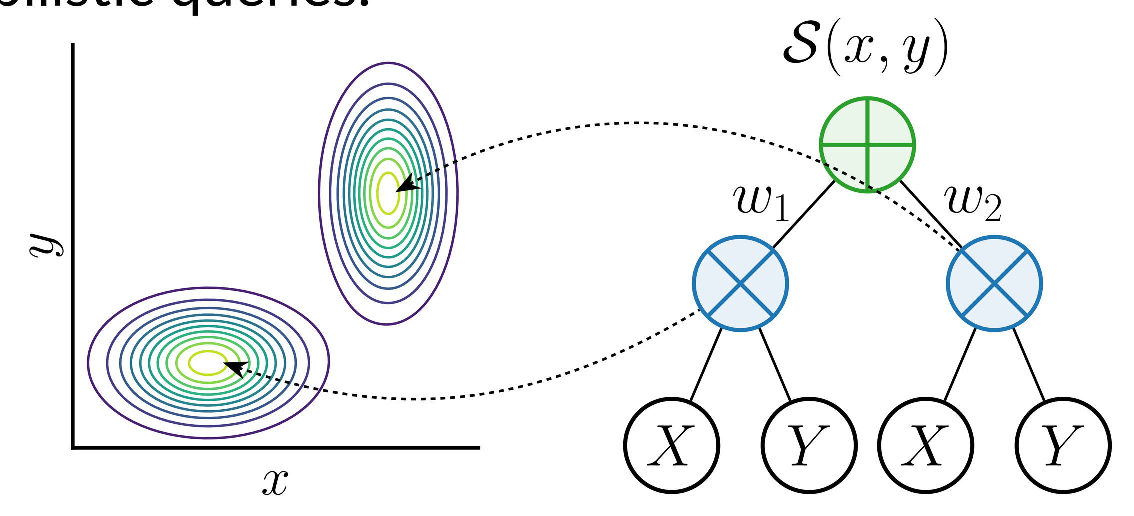
Problem: Probabilistic Circuits (PCs) are *overconfident*!
Solution: *Tractable Dropout Inference (TDI)*

- Provides **model uncertainty quantification** for PCs, enhancing **robustness** to perturbations, corruptions, and OOD data, all in an **efficient, single forward pass**.
- Enables integration of prior knowledge about epistemic and aleatoric uncertainty and the potential for incorporating uncertainty directly into training.

Research Questions: Are PCs overconfident? Is TDI a valuable model uncertainty quantification method that makes PCs more robust?

Probabilistic Circuits

Expressive probabilistic models, model joint distributions, enable exact, tractable inference for various probabilistic queries.



Sum Nodes: Convex combination of input nodes.
Product Nodes: Product of input nodes.
Leaf Nodes: Tractable, normalized PDFs/PMFs.

Paper **Code**

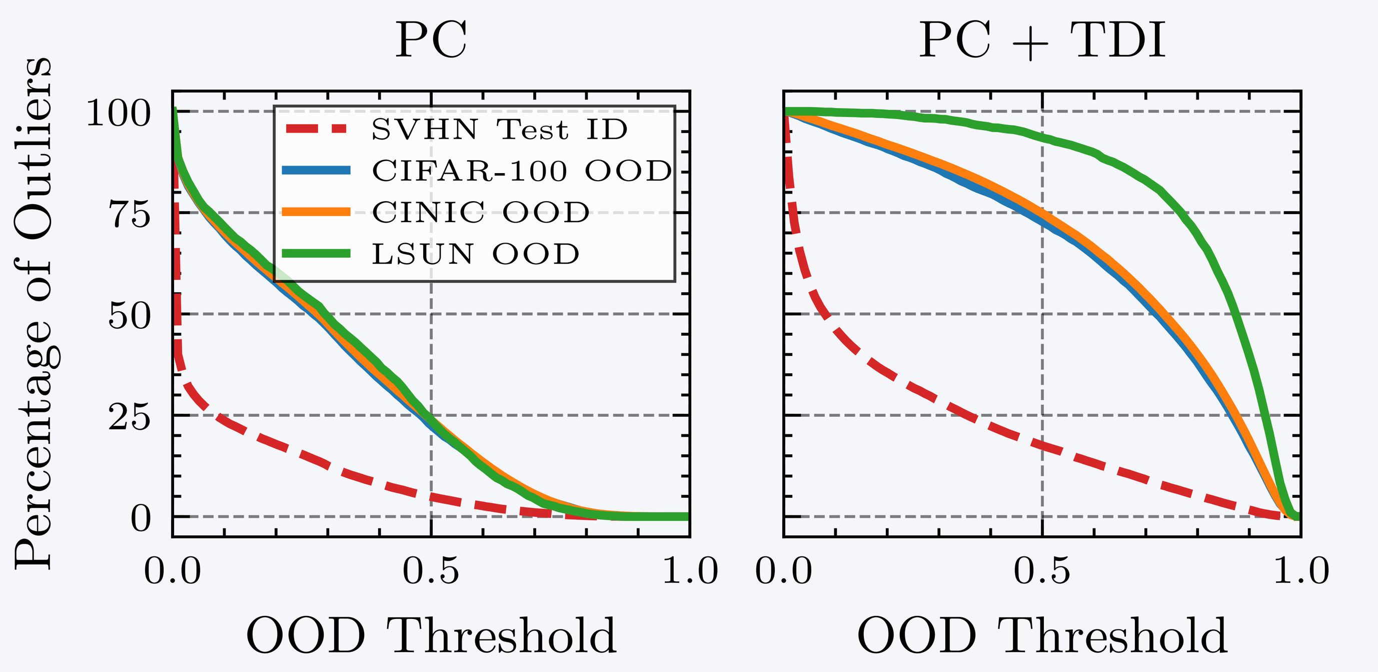
{ventola, steven.braun}@cs.tu-darmstadt.de, *equal contribution

Challenge: Probabilistic Circuits are Overconfident!

... when classifying

- **out-of-distribution** data
- **perturbed** data
- **corrupted** data

just like other (generative) models, despite assumed calibrated.



Solution: Uncertainty via *Tractable Dropout Inference*

Goal: **Quantify model uncertainty** to know what the model "*does not know*".

Inspiration: Monte Carlo dropout (MCD) → uncertainty by *multiple* stochastic model evaluations.

TDI: Leverage PCs' clear probabilistic semantics to provide a **closed-form** approximation to **MCD** in PCs.

Tractable Dropout Inference

1. View sum nodes as linear combinations of input RVs and *dropout* RVs: $\mathsf{S} = \sum_i \delta_i w_i \mathsf{P}_i$, with $\delta_i \sim \mathrm{Bern}(q)$.

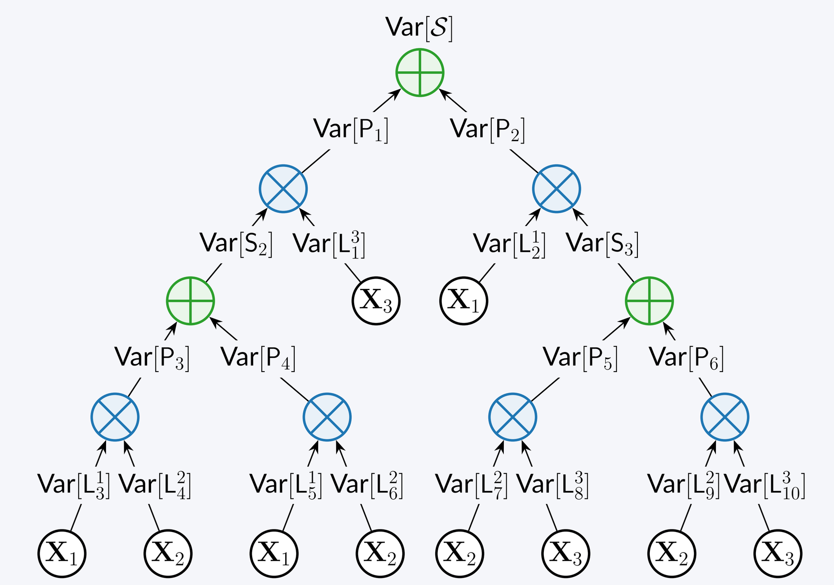
2. Derive closed-form expr. of *expectation*, *variance*, and *covariance* for nodes as a function of their inputs.

$$\mathbb{E}[\mathsf{S}] = q \textstyle\sum_i w_i \mathbb{E}[\mathsf{N}_i] \qquad \mathbb{E}[\mathsf{P}] = \prod_i \mathbb{E}[\mathsf{N}_i]$$

$$\mathrm{Var}[\mathsf{S}] = q \textstyle\sum_i w_i^2 (\mathrm{Var}[\mathsf{N}_i] + p\mathbb{E}[\mathsf{N}_i]^2) + q^2 \sum_{i \neq j} w_i w_j \mathrm{Cov}[\mathsf{N}_i, \mathsf{N}_j] \qquad \mathrm{Var}[\mathsf{P}] = \prod_i (\mathrm{Var}[\mathsf{N}_i] + \mathbb{E}[\mathsf{N}_i]^2) - \prod_i \mathbb{E}[\mathsf{N}_i]^2$$

$$\mathrm{Cov}[\mathsf{S}^A, \mathsf{S}^B] = q^2 \textstyle\sum_i w_i^A \sum_j w_j^B \mathrm{Cov}[\mathsf{N}_i^A, \mathsf{N}_j^B] \qquad \mathrm{Cov}[\mathsf{P}^A, \mathsf{P}^B] = \mathbb{E}[\prod_i \mathsf{N}_i^A \prod_j \mathsf{N}_j^B] - \prod_i \mathbb{E}[\mathsf{N}_i^A] \prod_j \mathbb{E}[\mathsf{N}_j^B]$$

3. Propagate exp., var., and cov. in a bottom-up pass from leaves to the root node.

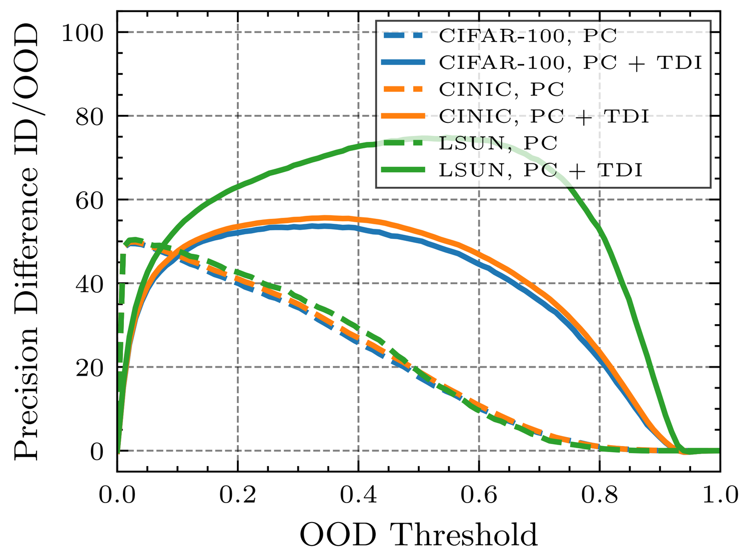


Efficiency: TDI provides **tractable** uncertainty quantification in a **single forward pass**.

Robustness:

1. detects OOD data and adequately balances ID vs. OOD accuracy
2. better detects distribution shifts while retaining accuracy
3. is more robust against challenging image corruptions

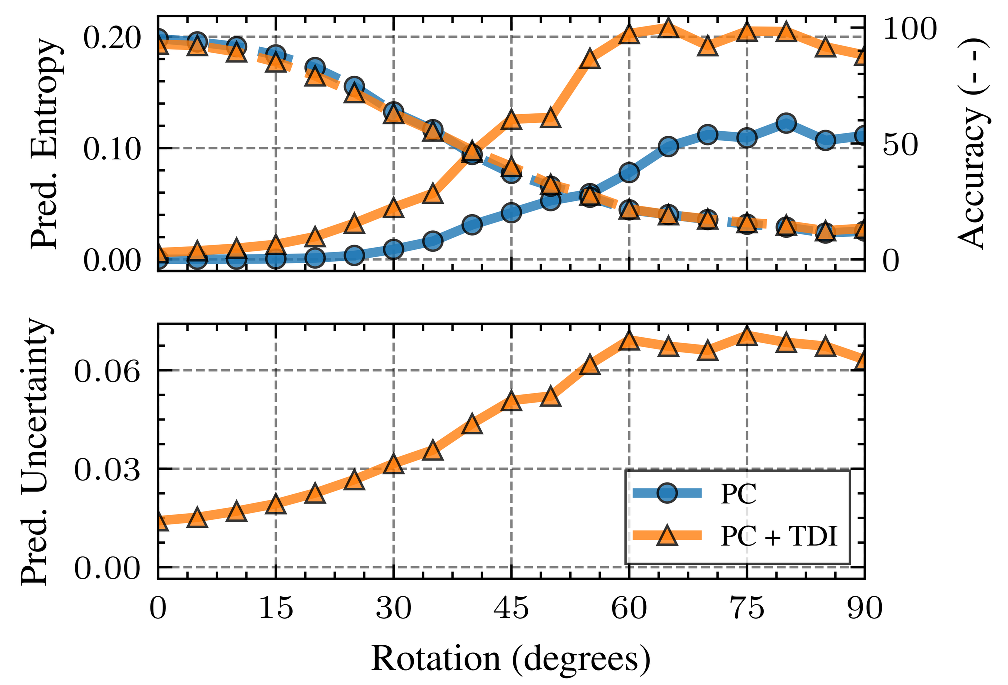
1st Scenario: Out-of-distribution data



AUC (↑)	CIFAR	CINIC	LSUN
PC	29.3	29.9	30.3
PC + TDI	**64.6**	**66.1**	**81.8**
PC + MCD	68.5	70.0	84.9
MLP	56.0	58.1	55.9
MLP + MCD	80.8	95.9	80.5
LeNet	17.1	15.6	2.6
LeNet + MCD	43.4	92.9	29.5

MCD needs 100 evaluations, TDI requires only a single pass!

2nd Scenario: Perturbed data



3rd Scenario: Corrupted data