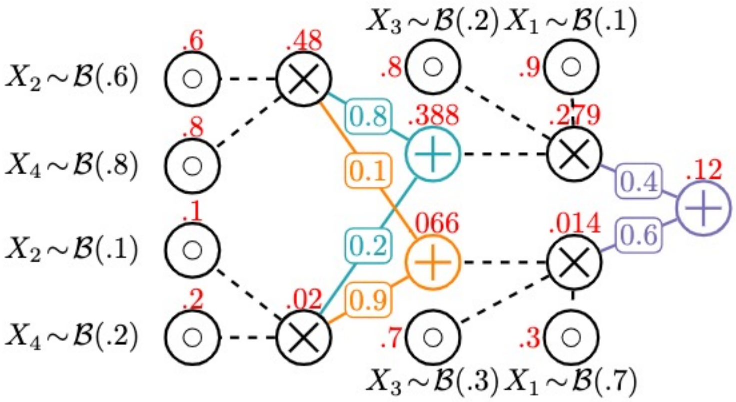
# Sparse Probabilistic Circuits via Pruning and Growing

Meihua Dang, Anji Liu, Guy Van den Broeck
Computer Science Department, University of California, Los Angeles
{mhdang, anjiliu, guyvdb}@cs.ucla.edu

NeurIPS 2022

# Probabilistic Circuits (PCs)




- PC recursively defines distributions using *sum units*, *product units*, and *univariate distributions*.

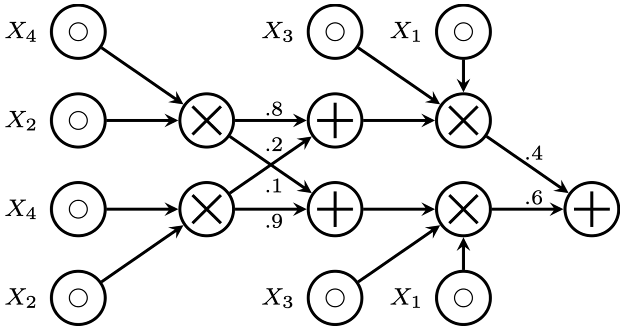
- PC is tractable probabilistic models allowing for *exact* and *efficient* computation of likelihoods and marginals.

UCLA

# PC: Evaluation and Sampling

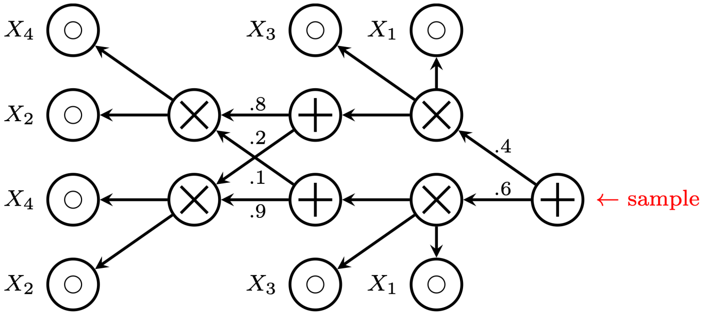
***Evaluation – Forward propagation***

- Computing likelihood for the assignment $p(\overline{x_1}, x_2, \overline{x_3}, x_4)$
- We set the values of leaf variables and then propagate the values feedforward




***Sampling – Backward propagation***

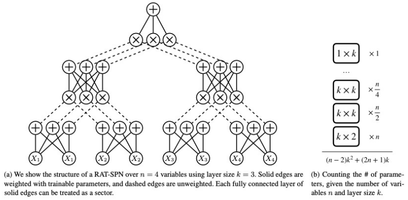
- Draw a sample from $p(X_1, X_2, X_3, X_4)$
- We start from root note and propagate backward

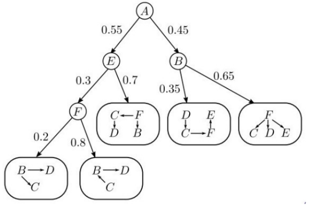


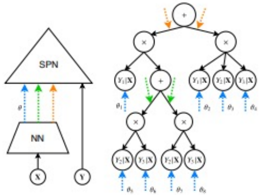

UCLA

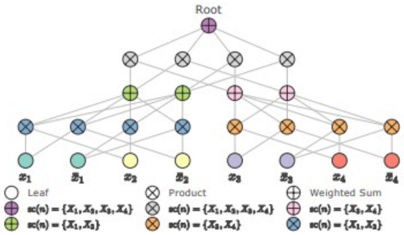
# PC: Learning

The learning of PCs aims to find good models that can better fit the data,





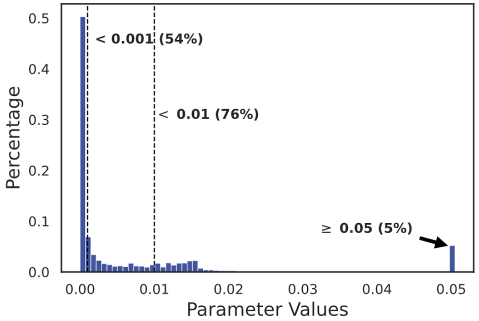

they focus on

- Finding good structures
- Adding more parameters

# Motivation

PC training performance plateaus as model size increases

- The capacity of large PCs are wasted



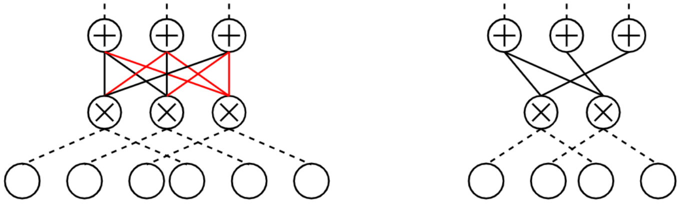

Histogram of parameter values for a sota PC with 2.18M parameters on MNIST

UCLA

# Method: Pruning

Many parameters have close-to-zero values, indicating low probabilities

→ Pruning less useful edges/parameters



(a) PC with fully connected layers (b) PC after pruning operation

*Which edges to choose?*

# Method: Pruning

***Heuristic #1: edges with smaller parameters***

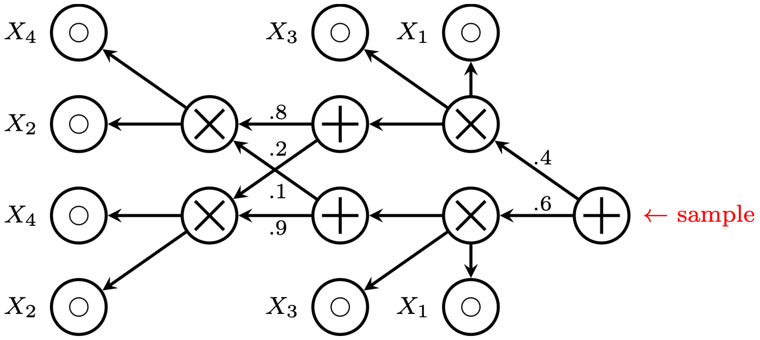
- Naïve
- Parameters only have local probability information

*Can we have more global probability information?*

# Method: Pruning

***Rethinking from PC sampling procedure***

- For each sum unit, activating one of its edge/parameter
- Keep track of the probability that each edge gets activated




***Heuristic #2: top-down probability***

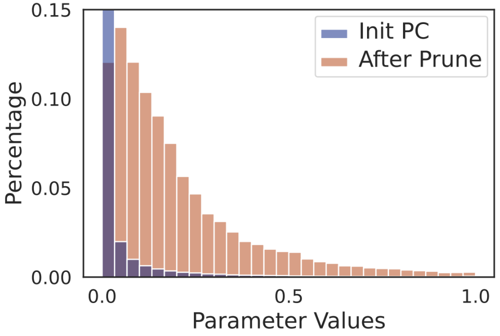
- The probability of reaching an edge in a sampling procedure

# Method: Pruning

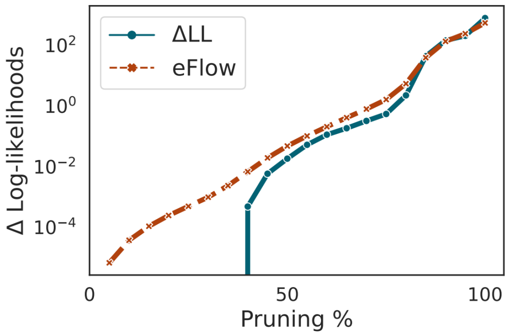
***Heuristic #3: circuit flow***

- Top-down probability conditioned on a given sample
- Aggregate circuit flows over dataset

After pruning,




param. distribution is more balanced



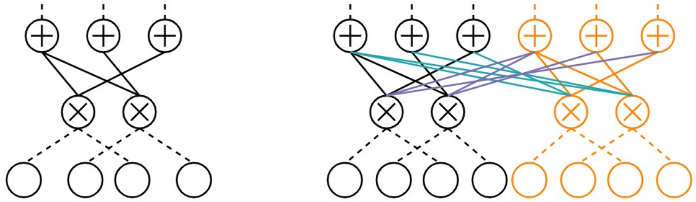

actual LL drop are close to circuit flows

UCLA

# Method: Growing

Increase PC capacity

→ Make noisy copies of existing structures and the parameters



(b) PC after pruning operation (c) PC after growing operation

- Applying pruning and growing iteratively for structure learning

# Experiments: Density Estimation

- MNIST-family image datasets

Table 1: Density estimation performance on MNIST-family datasets in test set bpd.

| Dataset | Sparse PC (ours) | HCLT | RatSPN | IDF | BitSwap | BB-ANS | McBits |
|---|---|---|---|---|---|---|---|
| MNIST | **1.14** | 1.20 | 1.67 | 1.90 | 1.27 | 1.39 | 1.98 |
| EMNIST(MNIST) | **1.52** | 1.77 | 2.56 | 2.07 | 1.88 | 2.04 | 2.19 |
| EMNIST(Letters) | **1.58** | 1.80 | 2.73 | 1.95 | 1.84 | 2.26 | 3.12 |
| EMNIST(Balanced) | **1.60** | 1.82 | 2.78 | 2.15 | 1.96 | 2.23 | 2.88 |
| EMNIST(ByClass) | **1.54** | 1.85 | 2.72 | 1.98 | 1.87 | 2.23 | 3.14 |
| FashionMNIST | **3.27** | 3.34 | 4.29 | 3.47 | 3.28 | 3.66 | 3.72 |

- Character-level language modeling

Table 2: Character-level language modeling results on Penn Tree Bank in test set bpd.

| Dataset | Sparse PC (ours) | Bipartite flow [32] | AF/SCF [36] | IAF/SCF [36] |
|---|---|---|---|---|
| Penn Tree Bank | **1.35** | 1.38 | 1.46 | 1.63 |

# Thank You

# Sparse Probabilistic Circuits via Pruning and Growing

**Meihua Dang**
CS Department
UCLA
mhdang@cs.ucla.edu

**Anji Liu**
CS Department
UCLA
liuanji@cs.ucla.edu

**Guy Van den Broeck**
CS Department
UCLA
guyvdb@cs.ucla.edu

## Abstract

Probabilistic circuits (PCs) are a tractable representation of probability distributions allowing for exact and efficient computation of likelihoods and marginals. There has been significant recent progress on improving the scale and expressiveness of PCs. However, PC training performance plateaus as model size increases. We discover that most capacity in existing large PC structures is wasted: fully-connected parameter layers are only sparsely used. We propose two operations: *pruning* and *growing*, that exploit the sparsity of PC structures. Specifically, the pruning operation removes unimportant sub-networks of the PC for model compression and comes with theoretical guarantees. The growing operation increases model capacity by increasing the dimensions of latent states. By alternatingly applying pruning and growing, we increase the capacity that is meaningfully used, allowing us to significantly scale up PC learning. Empirically, our learner achieves state-of-the-art likelihoods on MNIST-family image datasets and an Penn Tree Bank language data compared to other PC learners and less tractable deep generative models such as flow-based models and variational autoencoders (VAEs).

## 1 Introduction

Probabilistic circuits (PCs) [43, 3] are a unifying framework to abstract from a multitude of tractable probabilistic models. The key property that separates PCs from other deep generative models such as flow-based models [30] and VAEs [18] is their *tractability*. It enables them to compute various queries, including marginal probabilities, exactly and efficiently [44]. Therefore, PCs are increasingly used in inference-demanding applications such as enforcing algorithmic fairness [2, 4], making predictions under missing data [17], computing expected kernels [22], data compression [25], and anomaly detection [12].

Recent advancement of PC learning [35], regularization [39, 24] and efficient parallelism implementation [32, 29] have been pushing the

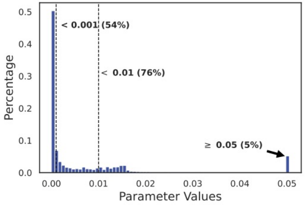



Figure 1: Histogram of parameter values for a state-of-the-art PC with 2.18M parameters on MNIST.