# On Resource-Efficient Bayesian Network Classifiers and Deep Neural Networks


Wolfgang Roth[1], Günther Schindler[2], Holger Fröning[2], and Franz Pernkopf[1]

[1] {roth,pernkopf}@tugraz.at,
[2] {guenther.schindler,holger.froening}@ziti.uni-heidelberg.de

[1] Signal Processing and Speech Communication Laboratory, Graz University of Technology
[2] Institute of Computer Engineering, Ruprecht Karls University, Heidelberg


Der Wissenschaftsfonds.


## Summary

- **Reduce model complexity of Bayesian network classifiers**
  - **Transfer techniques from deep learning [1] to Bayesian networks**
- **Method 1: Model-size-aware TAN structure learning**
  - **Apply differentiable TAN structure learning from [2]**
    - ✓ **Structure learning using backpropagation**
  - **New extension: Trade off between accuracy and model size**
- **Method 2: Quantization-aware Training**
  - **Quantize log-probabilities (CPTs) to few bits**
  - **Apply straight-through gradient estimator**
- **Comparing Bayesian networks and deep neural networks**


## Bayesian Network (BN) Classifiers

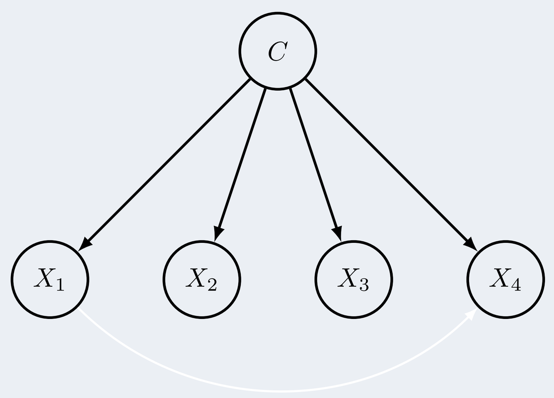
- BNs define a factorization of a joint distribution via a directed acyclic graph $\mathcal{G}$ as

$$p(\boldsymbol{X}) = \prod_{i=1}^{D} p\left(X_i \,|\, \mathrm{pa}(X_i)\right)$$

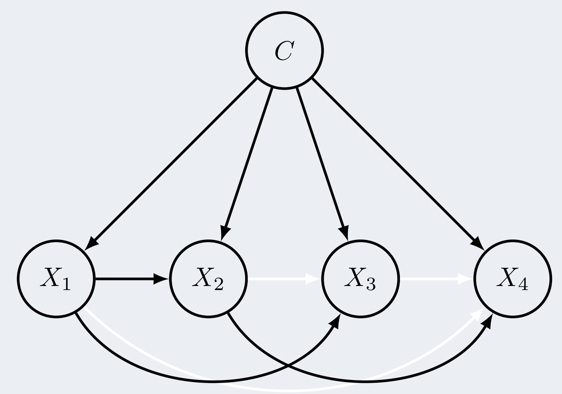
- This work: Discrete features $\boldsymbol{X}_i$
  - Parameters: Conditional probability tables (CPTs) $\boldsymbol{\theta}$
- Classification: Additional distinct class variable $\boldsymbol{C}$
  - Classify according to $\mathrm{argmax}_c \log p(\mathbf{x}, c)$
  - Classification requires only $(D+1) \cdot \#(\text{classes})$ additions in log-domain

## Naive Bayes vs. tree-augmented naive Bayes (TAN)

Naive Bayes




TAN



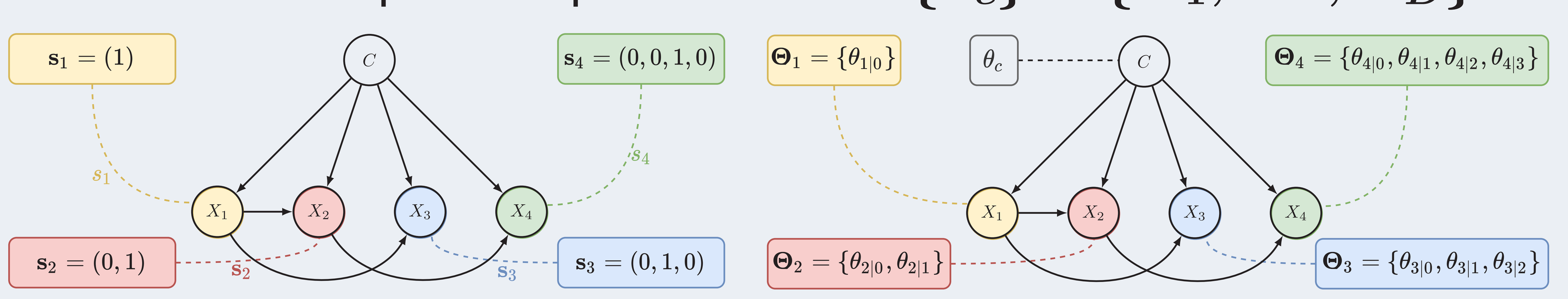

Naive Bayes:
- ✓ Few parameters
- ✓ Few operations
- ϟ Accuracy

TAN:
- ✓ Accuracy
- ✓ Few operations
- ⚠ Model size depends on structure

- Many possible TAN structures ⇒ structure learning

## Model-Size-Aware TAN Structure Learning

- Encode TAN graph $\mathcal{G}$ using one-hot vectors $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_D)$
- CPTs for all possible parents: $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_c\} \cup \{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_D\}$




- Treat parameters $\boldsymbol{\Theta}$ and structure $\mathbf{s}$ jointly

$$\log p(\boldsymbol{X}, C) = \log p_{\boldsymbol{\theta}_c}(C) + \sum_{i=1}^{D} \sum_{j=0}^{i-1} s_{i|j} \log p_{\boldsymbol{\theta}_{i|j}}(X_i \,|\, X_j, C)$$

- Continuous relaxation of one-hot $\mathbf{s}_i$ results in probabilities $\boldsymbol{\Phi}_i$
  - Probabilities $\boldsymbol{\Phi} = (\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_D)$ define a distribution over TAN graphs $\mathcal{G}$
  - Optimize expectation with respect to $\boldsymbol{\Phi}$ and take most probable graph $\mathcal{G}$

$$\mathcal{L}_{\mathrm{SL}}(\boldsymbol{\Phi}, \boldsymbol{\Theta}) = \mathbb{E}_{\mathbf{s} \sim p_{\boldsymbol{\Phi}}}\left[\mathcal{L}(\boldsymbol{\Theta}, \mathbf{s})\right] \qquad \text{(proposed in [2])}$$

  - $\mathcal{L}_{\mathrm{SL}}$ is differentiable with respect to $\boldsymbol{\Phi}$ ⇒ optimize with backpropagation
- This work: New term penalizes number of parameters

$$\mathcal{L}_{\mathrm{SL}}^{\mathrm{MS}}(\boldsymbol{\Phi}, \boldsymbol{\Theta}) = \overbrace{\mathcal{L}_{\mathrm{SL}}(\boldsymbol{\Phi}, \boldsymbol{\Theta})}^{\text{proposed in [2]}} + \overbrace{\lambda_{\mathrm{MS}}}^{\text{trade-off}} \overbrace{\mathbb{E}_{\mathbf{s} \sim p_{\boldsymbol{\Phi}}}\left[\mathcal{L}_{\mathrm{MS}}(\mathbf{s})\right]}^{\#\text{model parameters}}$$

## Quantization-Aware Training

- BNs: Quantize log-probabilities to negative fixed-point values

$$Q_{\mathrm{BN}}(\theta) = \mathrm{clip}(\mathrm{round}(\theta \cdot 2^{B_F}) \cdot 2^{-B_F}, \; -U, \; 0)$$

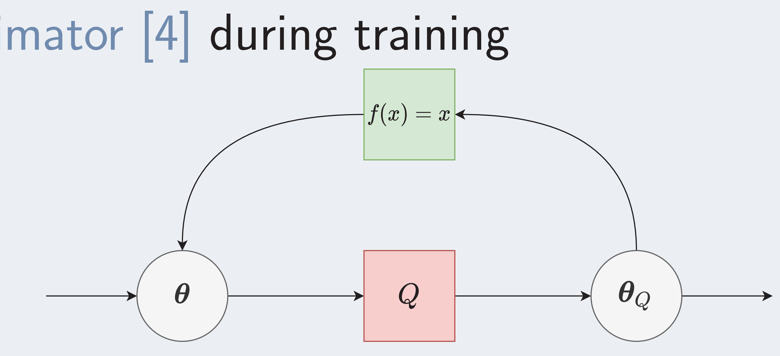
- DNNs: Quantize weights according to [3]

$$Q_{\mathrm{DNN}}(w) = Q\left(\frac{\mathrm{clip}(w, \; -1, \; 1) + 1}{2}; B\right) \cdot 2 - 1$$

$$Q(v; B) = \frac{1}{2^B - 1} \cdot \mathrm{round}((2^B - 1) \cdot v)$$

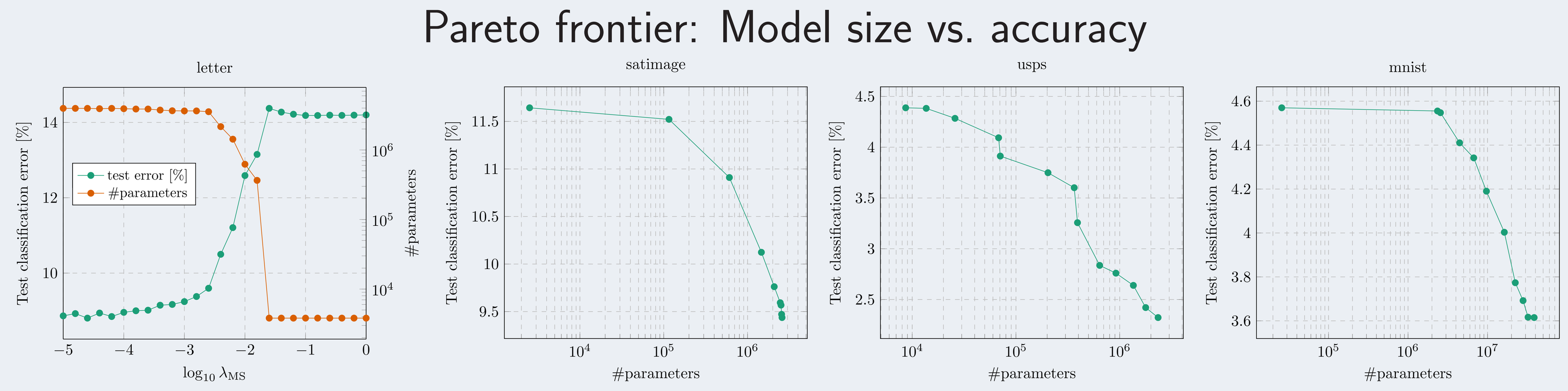
- The gradient of quantization functions is zero almost everywhere
  - Apply the straight-through gradient estimator [4] during training

At backpropagation: Pretend that the gradient of a quantization function is non-zero




## Experiments: Model-Size-Aware TAN Structure Learning

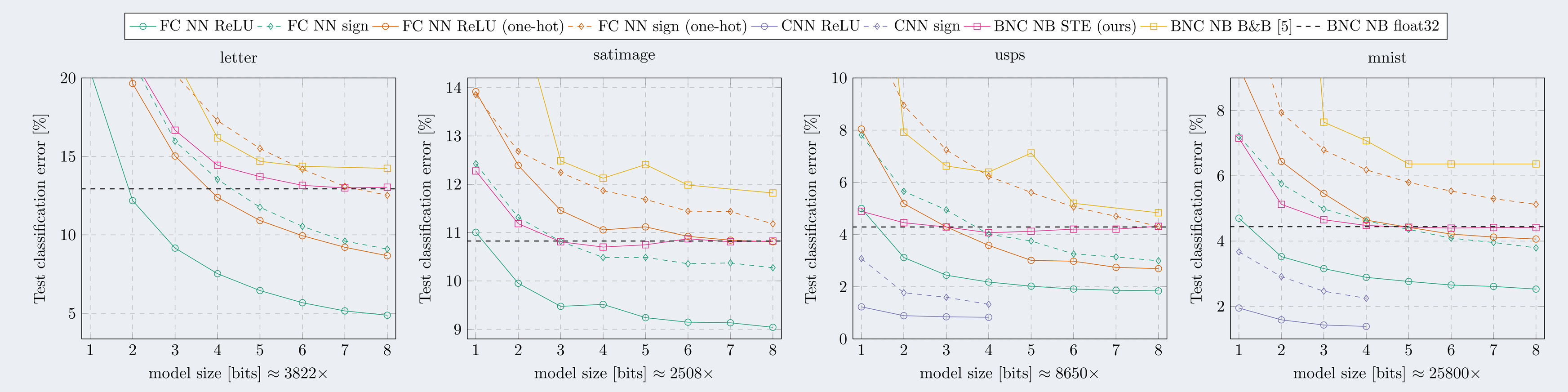
Pareto frontier: Model size vs. accuracy



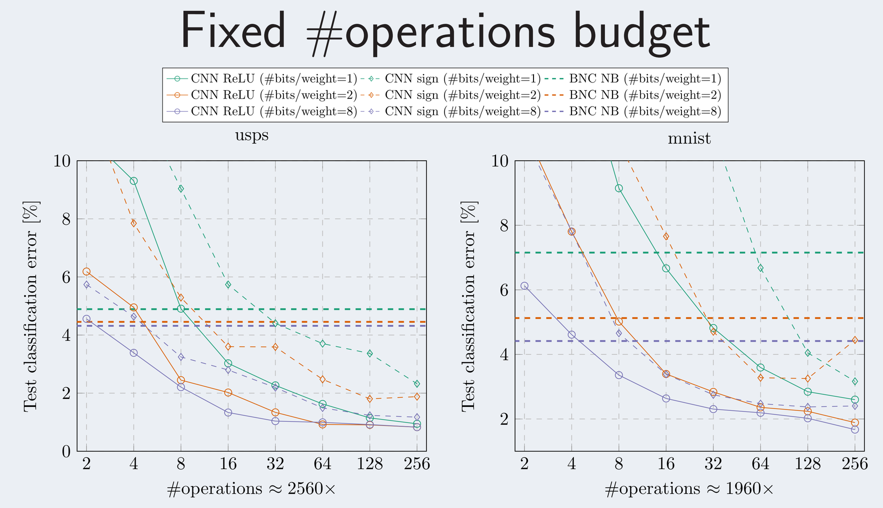

✓ Effective trade-off between accuracy and model size

## Experiments: Quantization-Aware Training

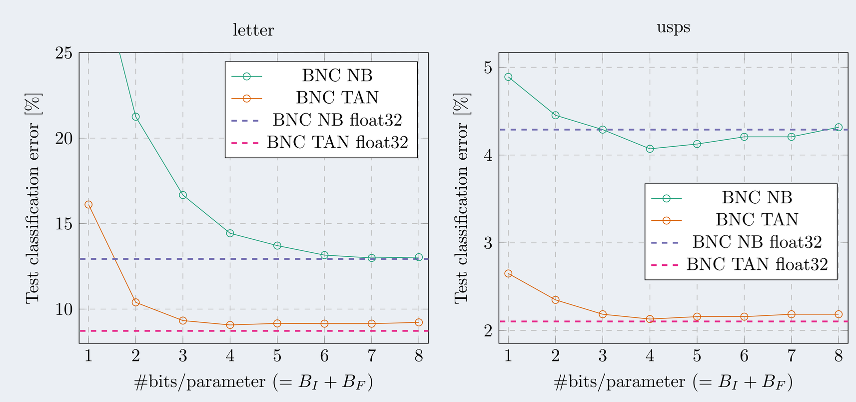
Fixed parameter memory budget



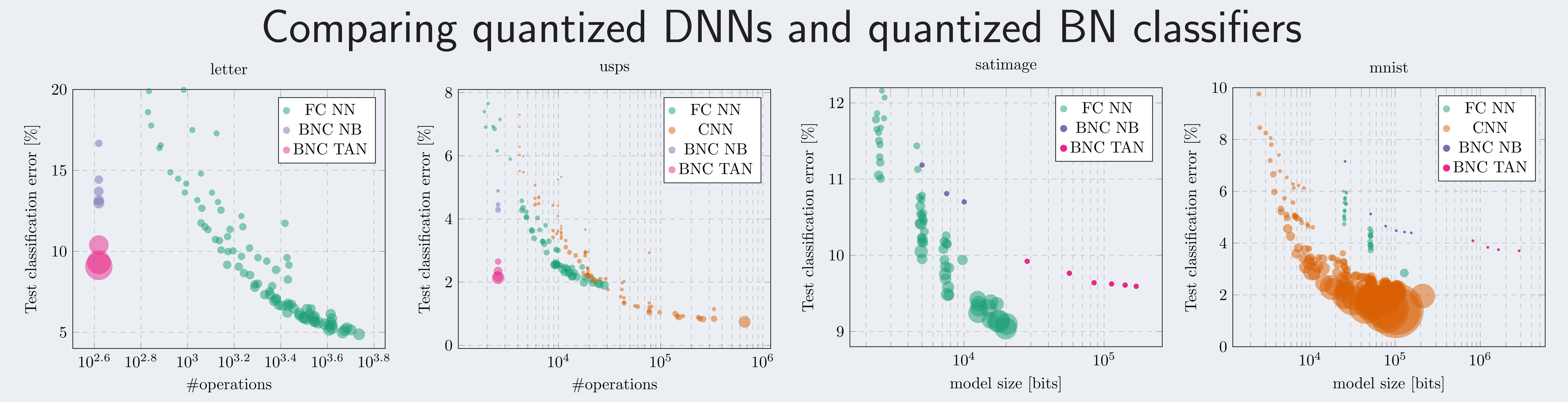

Fixed #operations budget




Naive Bayes vs TAN




Comparing quantized DNNs and quantized BN classifiers




- BNs: ✓ few operations ✓ decent accuracy
- CNNs: ✓ best accuracy ✓ few parameters ϟ many operations
- fully connected DNNs: ✓ flexible trade-offs

## References


[1] Roth et al., Resource-Efficient Neural Networks for Embedded Systems, arXiv:2001.03048

[2] Roth and Pernkopf, Differentiable TAN Structure Learning for Bayesian Network Classifiers, PGM 2020

[3] Zhou et al., DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients, arXiv:1606.06160

[4] Bengio et al., Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation, arXiv:1308.3432

[5] Tschiatschek et al., Integer Bayesian Network Classifiers, ECML 2014



Acknowledgments: This work was supported by the Austrian Science Fund (FWF) under the project number I2706-N31.