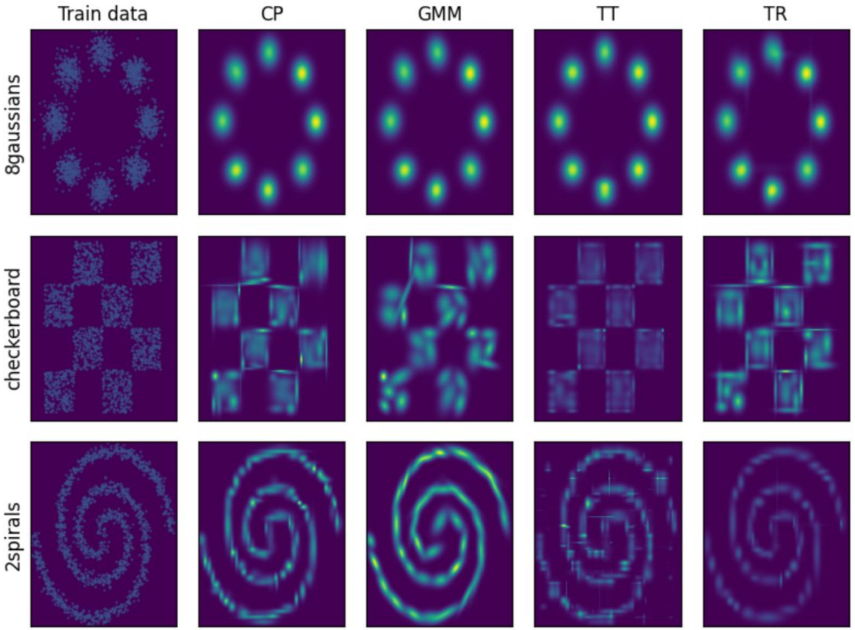
Tensor Factorized Density Estimation

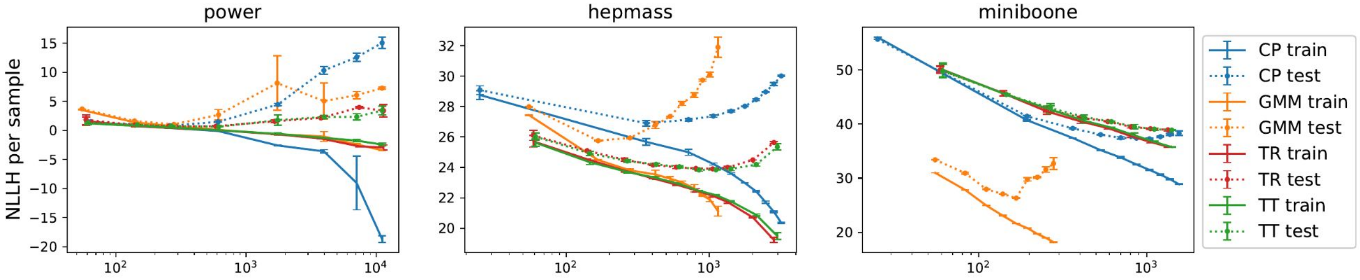
Eskild Børsting Sørensen, Johannes Emme Jørgensen,
Jonas Vestergaard Jensen, Peter Mørch Groth



Correction for **Fig. 1. Training and test performance**

Incorrect x-axis values for *hepmass* and *miniboone*:

Before:



After:

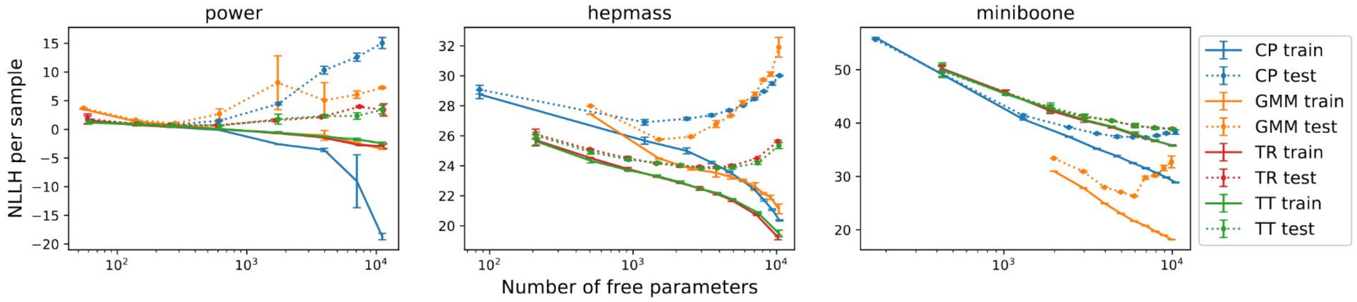


Fig. 1. **Training and test performance.** The models are evaluated at various numbers of free parameters at a sample size of 1750 for three datasets with increasing number of dimensions.

Introduction

- **Prior:** Gaussian Mixture Model (GMM) $O(KM^2)$
- **Objective:** Density estimator that scales better with high-dimensional data
- **Method:** Tensor decompositions → factorizing joint distribution
- **Evaluation:** Compare with conventional GMM and state-of-the-art neural density estimator

Tensor decompositions

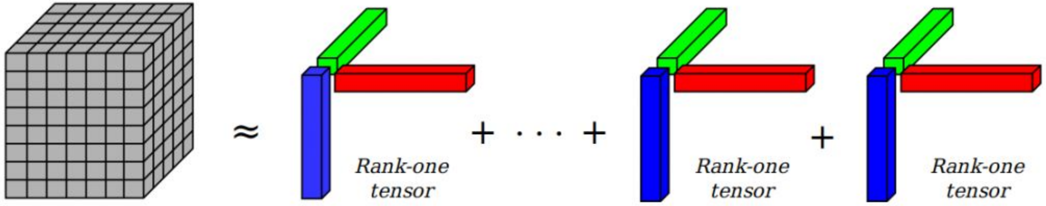
Canonical polyadic (CP) decomposition

$$A(i_1, i_2, \dots, i_M) = \sum_{\alpha}^{r} U_1(i_1, \alpha) U_2(i_2, \alpha) \cdots U_M(i_M, \alpha)$$

Factorization using CP procedure

$$p(x_1, x_2, \dots, x_M) = \sum_{k=1}^{K} p(k) \prod_{m=1}^{M} p(x_m | k)$$

$$= \sum_{k=1}^{K} w_k \prod_{m=1}^{M} \mathcal{N}(x_m | \mu_{k_m}, \sigma^2{}_{k_m})$$



[1]

[1] https://www.csee.umbc.edu/2019/06/talk-tensor-decomposition-of-nd-data-arrays-umbc/

Factorization using **Tensor Train** procedure

$$p(x_1, x_2, \dots, x_M) = \sum_{k_0,\dots,k_M}^{K_0,\dots,K_M} p(k_0) \prod_{m=1}^{M} p(x_m, k_m | k_{m-1})$$

$$= \sum_{k_0,\dots,k_M}^{K_0,\dots,K_M} w_{k_0} \prod_{m=1}^{M} w_{k_m,k_{m-1}} \mathcal{N}(x_m | \mu_{k_m,k_{m-1}}, \sigma^2{}_{k_m,k_{m-1}})$$

$$\boxed{\sim O(K^2M)}$$

Factorization using **Tensor Ring** procedure

$$p(x_1, x_2, \dots, x_M)$$

$$= \sum_{k_0,\dots,k_M}^{K_0,\dots,K_M} p(k_0) \left[\prod_{m=1}^{M-1} p(x_m, k_m | k_{m-1}) \right] p(x_M | k_{M-1}, k_0)$$

$$= \sum_{k_0,\dots,k_M}^{K_0,\dots,K_M} w_{k_0} \left[\prod_{m=1}^{M-1} w_{k_m,k_{m-1}} \mathcal{N}\left(x_m | \mu_{k_m,k_{m-1}}, \sigma^2{}_{k_m,k_{m-1}}\right) \right]$$

$$\cdot \mathcal{N}(x_M | \mu_{k_{M-1},k_0}, \sigma^2{}_{k_{M-1},k_0})$$

Performance

	power	gas	hepmass	miniboone	8gaussians	checkerboard	2spirals
FFJORD	-0.46	-8.59	**14.92**	**10.43**			
CP	-0.42	-10.30	22.76	29.23	1.47	2.02	1.85
GMM	-0.43	**-13.23**	20.78	17.84	1.47	2.04	**1.79**
TT	-0.47	-5.23	23.07	35.29	1.47	**1.93**	1.94
TR	**-0.53**	-5.33	23.06	35.27	1.48	2.01	1.85

Table 1. Test negative log likelihood per sample for the selected models. Results are averages over 3 runs.

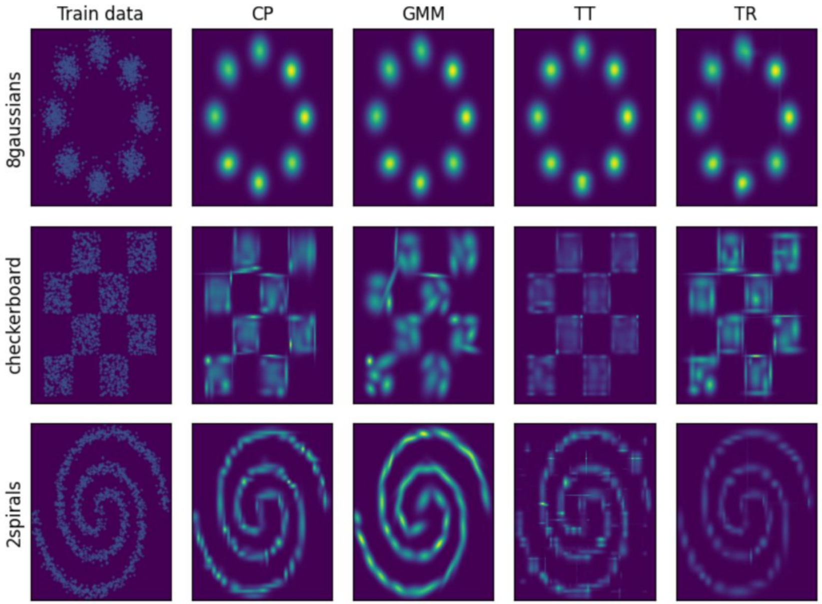


Fig. A.2. Modelling of synthetic data. The selected models qualitative performance on the 2D synthetic point data.

Structural differences in the data

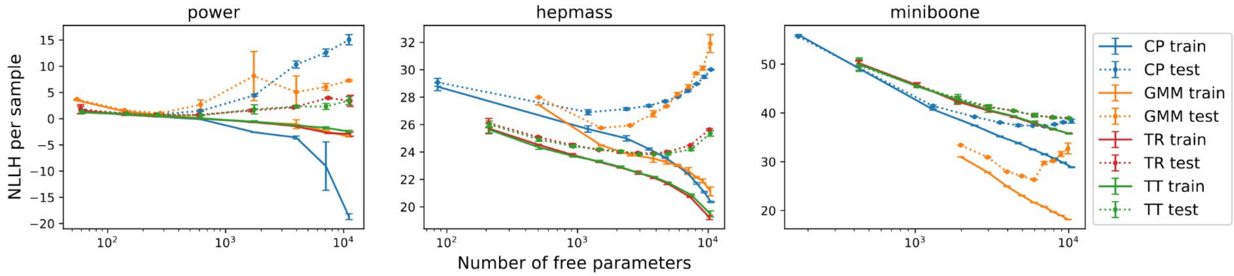
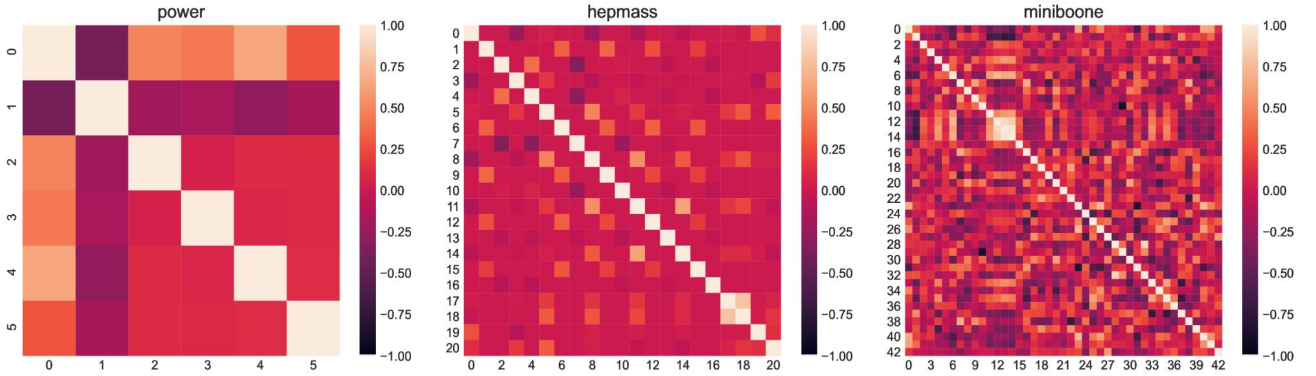


Fig. 1. Training and test performance. The models are evaluated at various numbers of free parameters at a sample size of 1750 for three datasets with increasing number of dimensions.



Regularisation in TT & TR

- **Regularisation properties:** Tensor train & Tensor ring generalizes well when increasing complexity

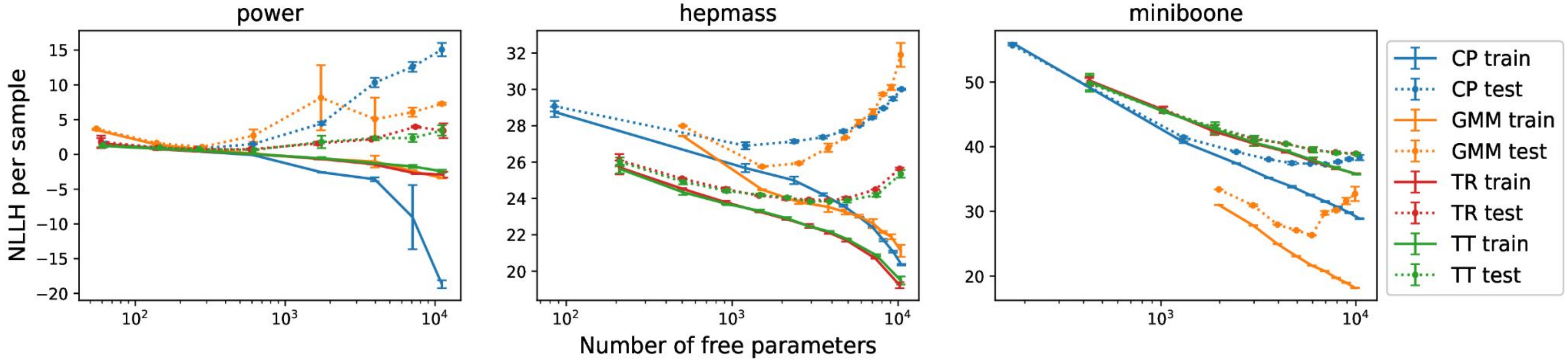


Fig. 1. Training and test performance. The models are evaluated at various numbers of free parameters at a sample size of 1750 for three datasets with increasing number of dimensions.

Ordering of variables

Ordering by maximizing correlation

- Better performance when doing early stopping.
- More overfitting

Ordering by maximizing partial correlation

- Worse performance when doing early stopping.
- Less overfitting

Regularization properties by ordering variables.

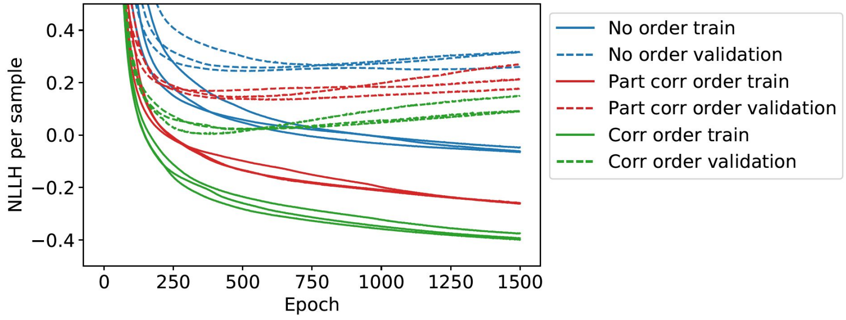


Fig. 3. Performance of TT and TR by ordering variables. The experiments were trained on the *power* dataset with 1750 subsamples and k=12.

Performance continued

	power	gas	hepmass	miniboone	8gaussians	checkerboard	2spirals
FFJORD	-0.46	-8.59	**14.92**	**10.43**			
CP	-0.42	-10.30	22.76	29.23	1.47	2.02	1.85
GMM	-0.43	**-13.23**	20.78	17.84	1.47	2.04	**1.79**
TT	-0.47	-5.23	23.07	35.29	1.47	**1.93**	1.94
TR	**-0.53**	-5.33	23.06	35.27	1.48	2.01	1.85

Table 1. Test negative log likelihood per sample for the selected models. Results are averages over 3 runs.

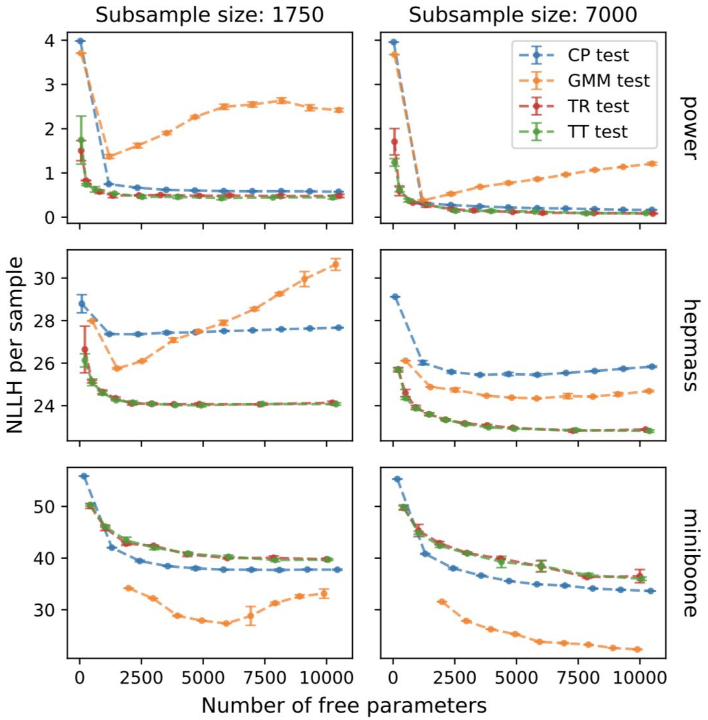
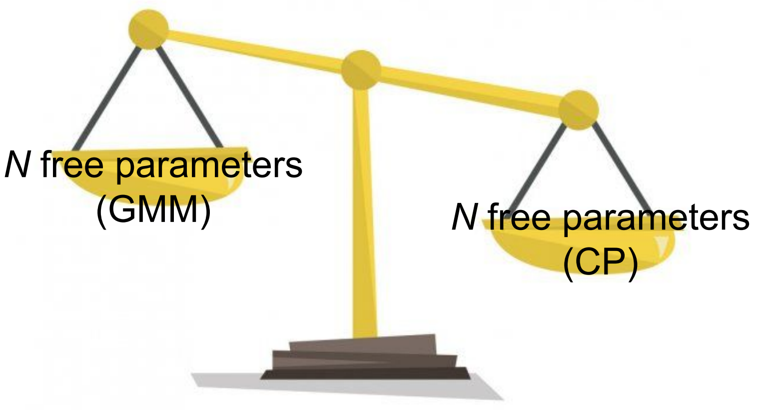


Fig. 2. **Test performance.** Comparison at various levels of model complexities.

Improvements

- **Means of comparison**
- **High-dimensional synthetic data**
- **Initialisations for TT and TR**



Conclusion

- **Tensor factorized density estimators are viable for density estimation**
- **Performance depends on dataset structure and ordering**
- **Robust when little data is available**