Generative Learning of Continuous Data by Tensor Networks

March Meeting 2023

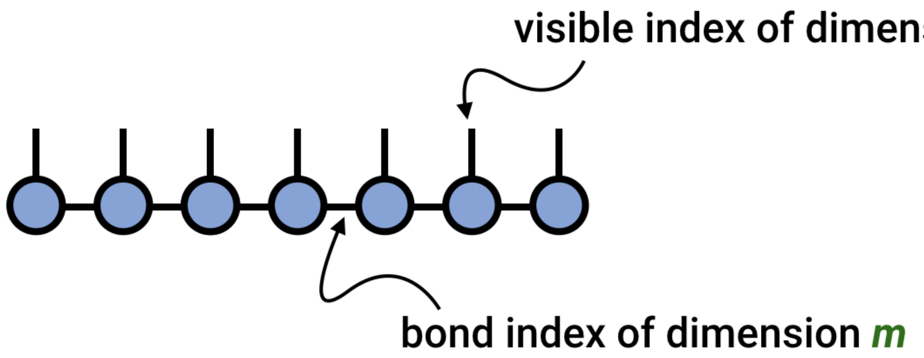
Alex Meiburg UCSB

Jing Chen, Jacob Miller, Alejandro Perdomo-Ortiz **Zapata Computing**

ZAPATA

Matrix Product States: Density Modelling

By now you've been hearing for a while about density modelling (generative modelling) with tensor networks…



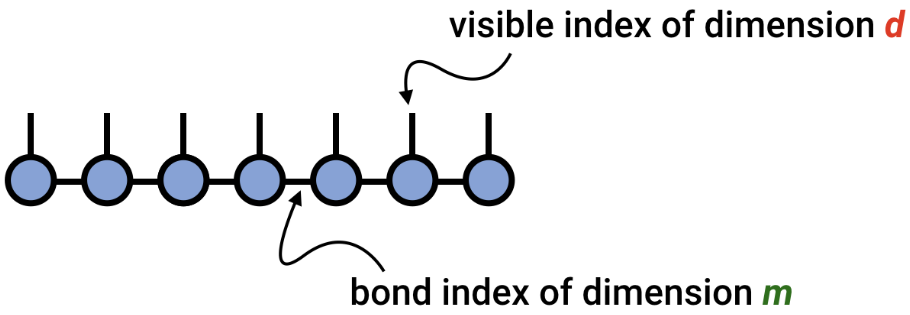
d is number of 'states' or 'values' at the site
For binary strings, d=2

Approximates some quantum state:

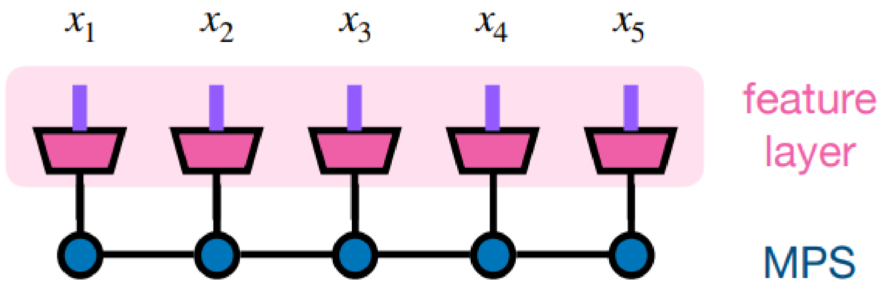
$$|\psi\rangle = \sum_b e^{i\theta_b} \sqrt{p_b}\, |b\rangle$$

ZAPATA

Bits → Real-valued distributions

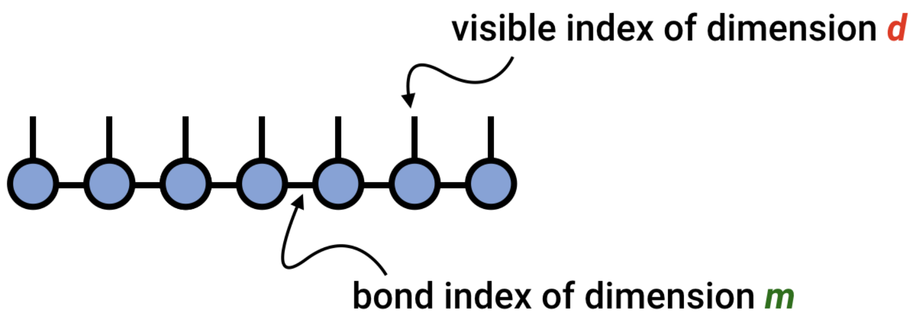


Map index *d* into Hilbert space over the reals

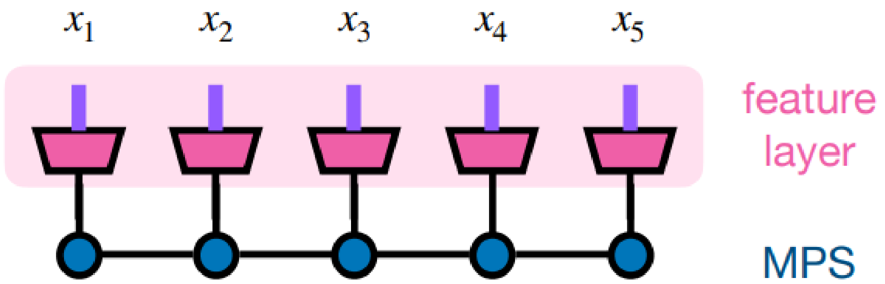


New tensor is an isometry in $\mathbb{C}^{d\times\mathbb{R}}$

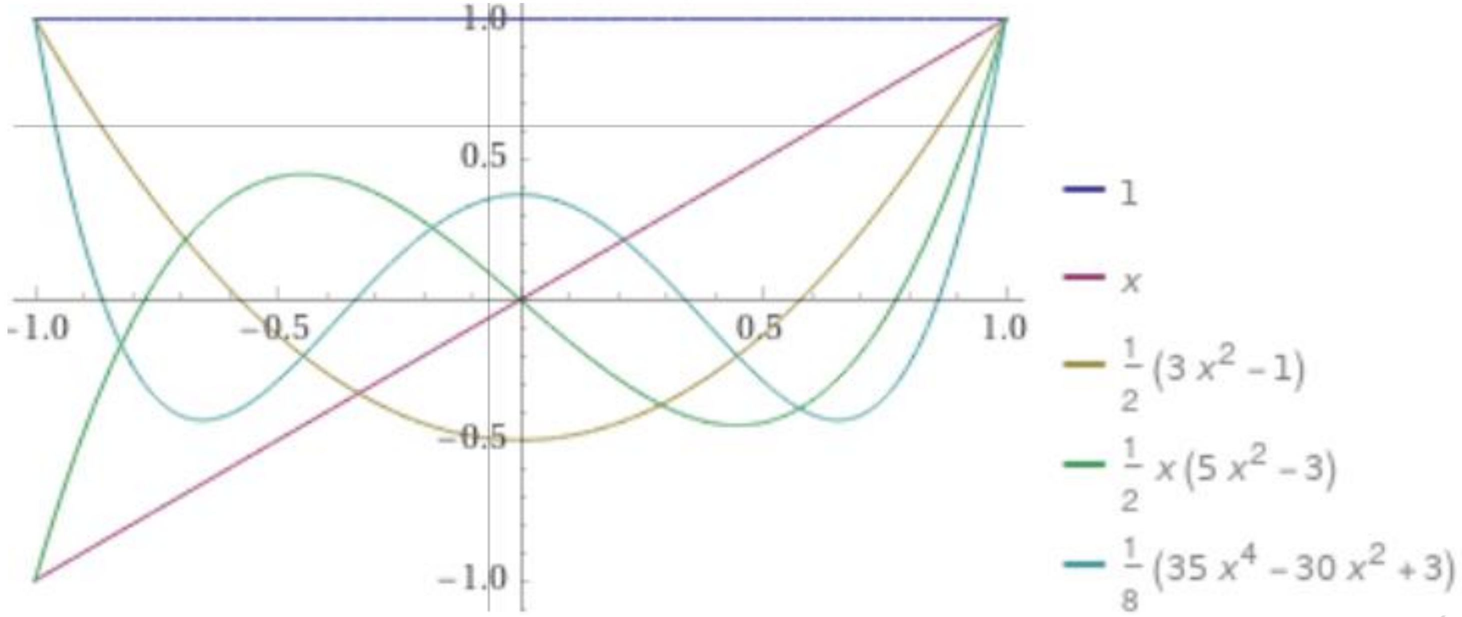
Bits → Real-valued distributions



Isometry: *d* orthogonal basis functions



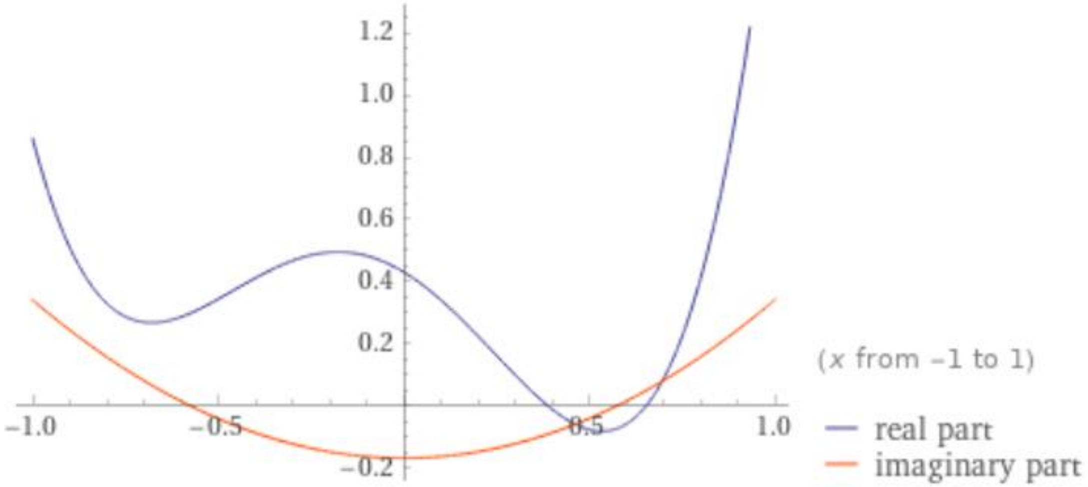
For example: polynomials on the interval [-1, 1]



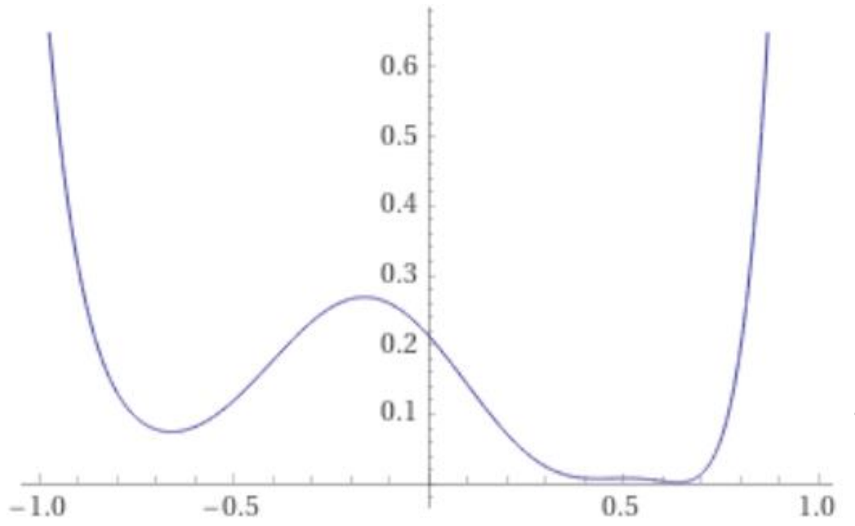
Representing real densities

- Linearly combine

with complex coefficients



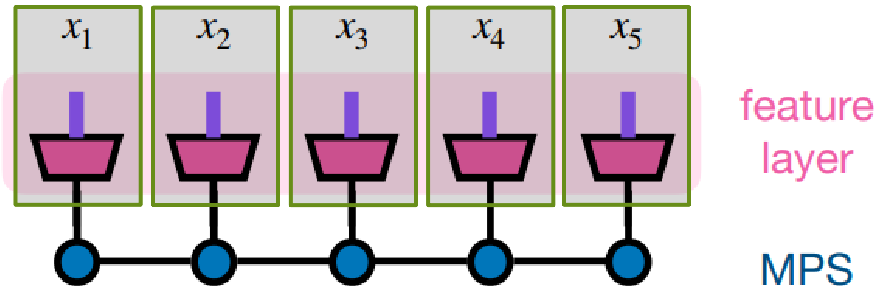
- Yields normalized density over [-1,1] (Born rule)



ZAPATA

Training the MPS

- Continuous values of a datum are fixed: push through the isometry (embed them) to *d*-dim quantum states. Then normal MPS training

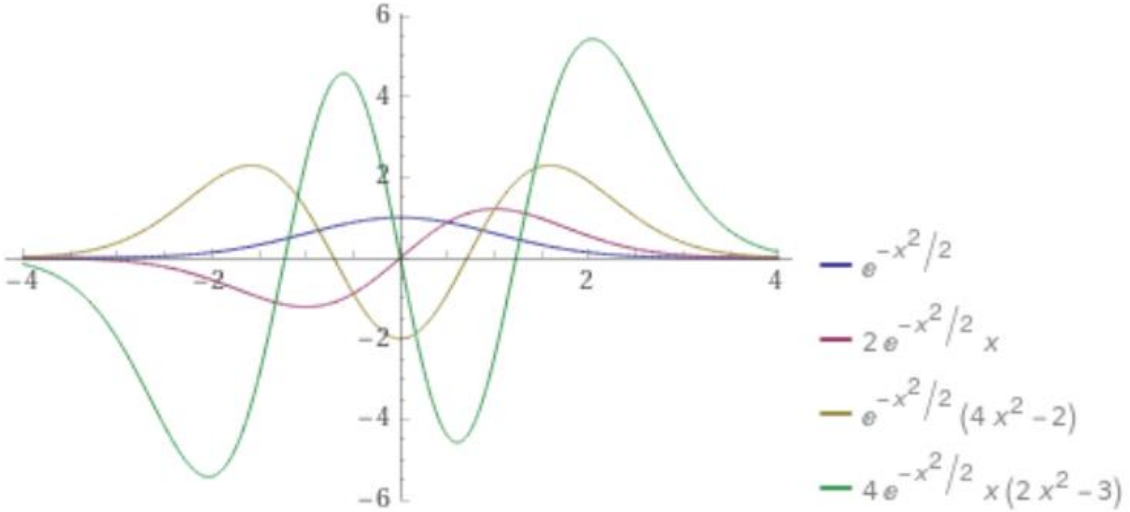


- Train to minimize negative log-likelihood (average entropy of generated samples)
- All probabilities are **densities**, so loss can be **negative**
  - e.g. uniform on [0, ½] has negative one bit of differential entropy

$$L = \frac{1}{|T|} \sum Q(\mathbf{x}) \mathbf{log}(P(\mathbf{x}))$$

Several options for basis functions ("features")

- Polynomials on fixed interval (shown before)
- Hermite functions: Gaussian × polynomial, over all of R



- Fourier series on a fixed interval
- Anything (1) easily integrable, (2) orthogonal, and ideally (3) complete

Several options for basis functions ("features")

- Anything (1) easily integrable, (2) orthogonal, and ideally (3) **complete**
- This suffices to show a universal approximation ability, for C^k functions:

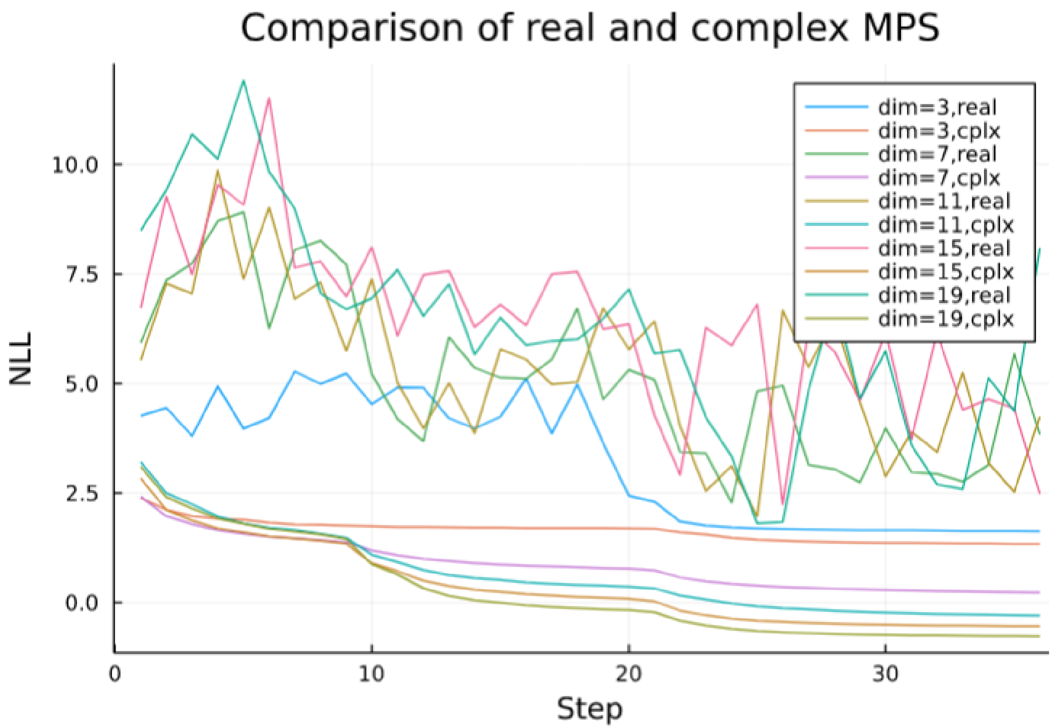
$$\mathrm{JSD}\left(p_{\mathrm{MPS}}^{(\chi,d)} \,\middle\|\, p\right) \leq O(\chi^{-\frac{k-1}{2}} + d^{-k})$$

with site dimension d and bond dimension χ.

Real vs. Complex MPS

- Real MPS: Memory compact, faster to compute with, in theory we don't *need* complex phases
- Complex MPS: Higher model capacity (per bond dimension), alters loss landscape

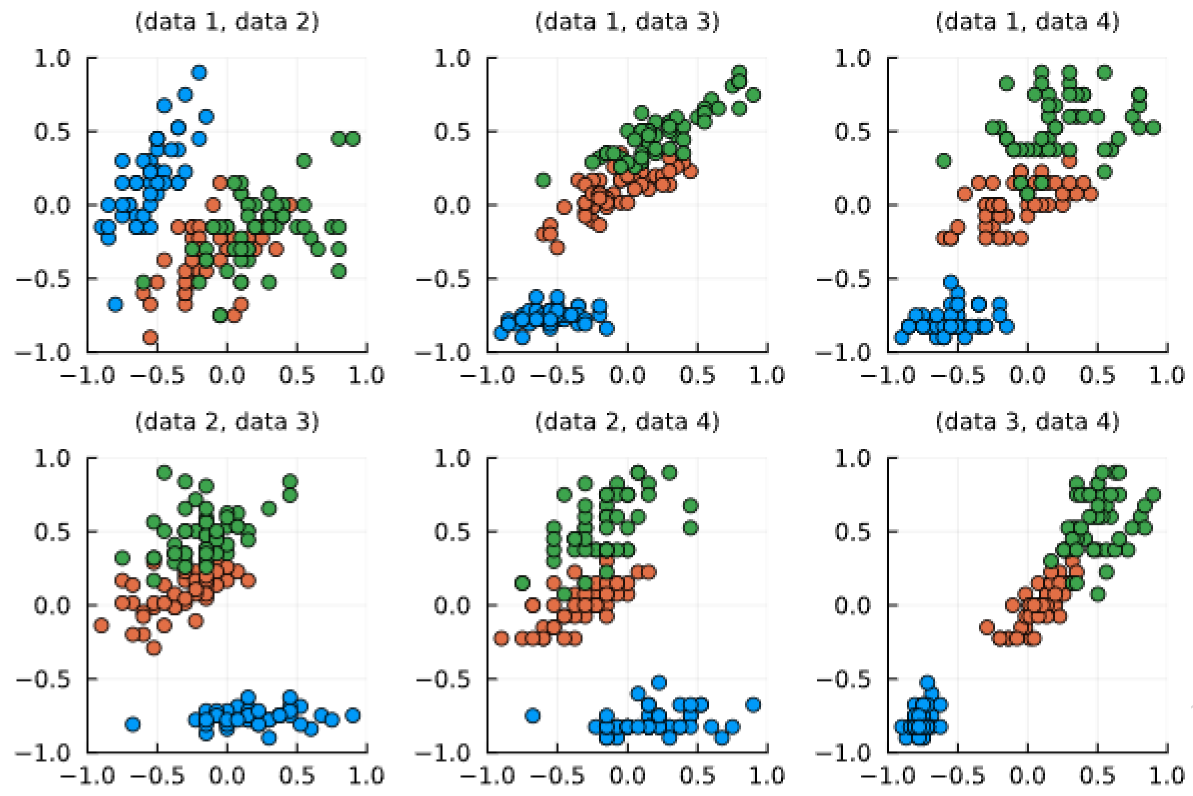
- Complex seems the better way to go!



ZAPATA

Test Problems – Iris Dataset

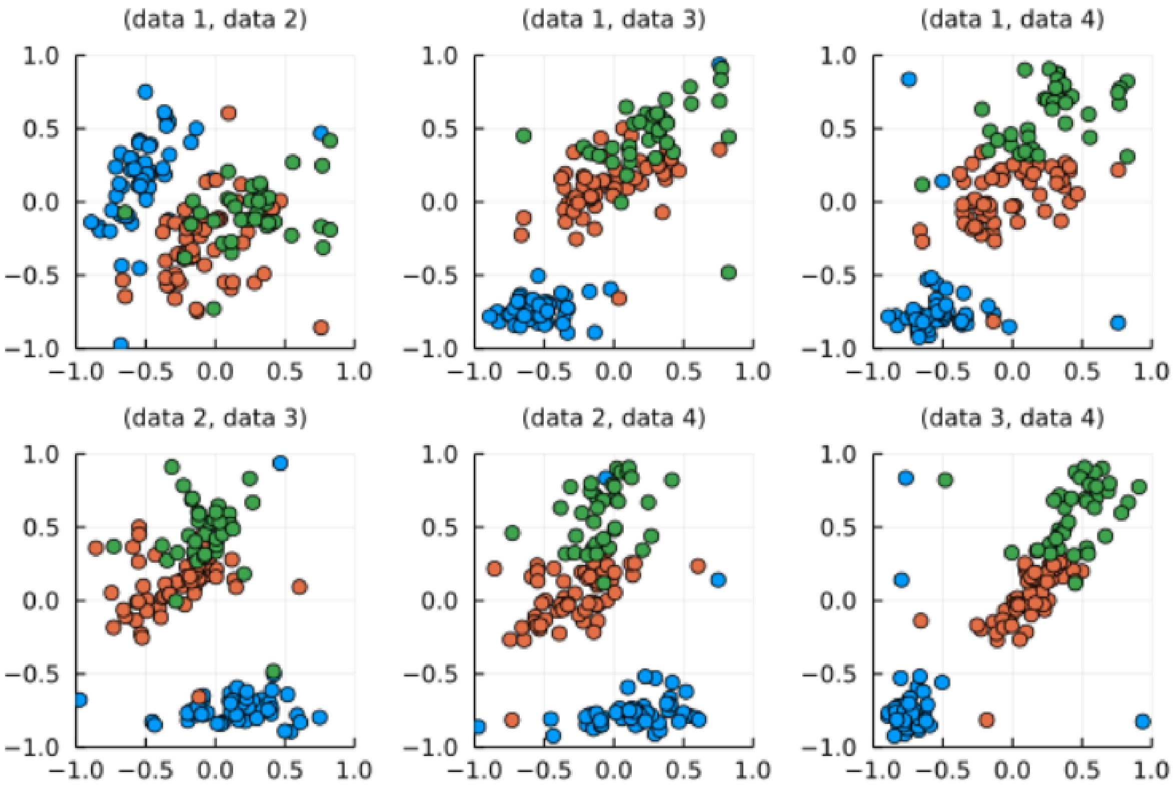
- Classic small ML dataset
- 4 continuous values, 1 class label (k=3)
- Pairwise plot of features:



ZAPATA

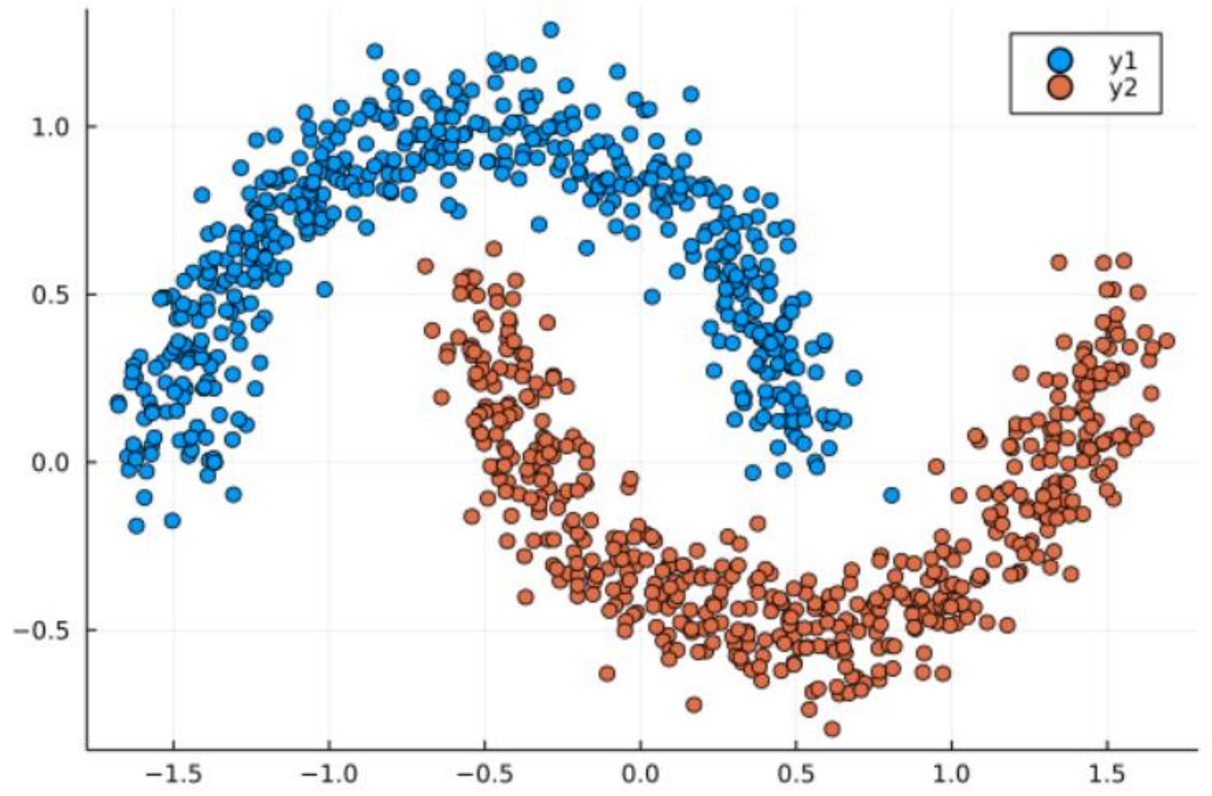
Test Problems – Iris Dataset

- We can support this mixture of continuous + discrete. Different local dimensions in the MPS. 5-site MPS
- Pairwise plot of features:



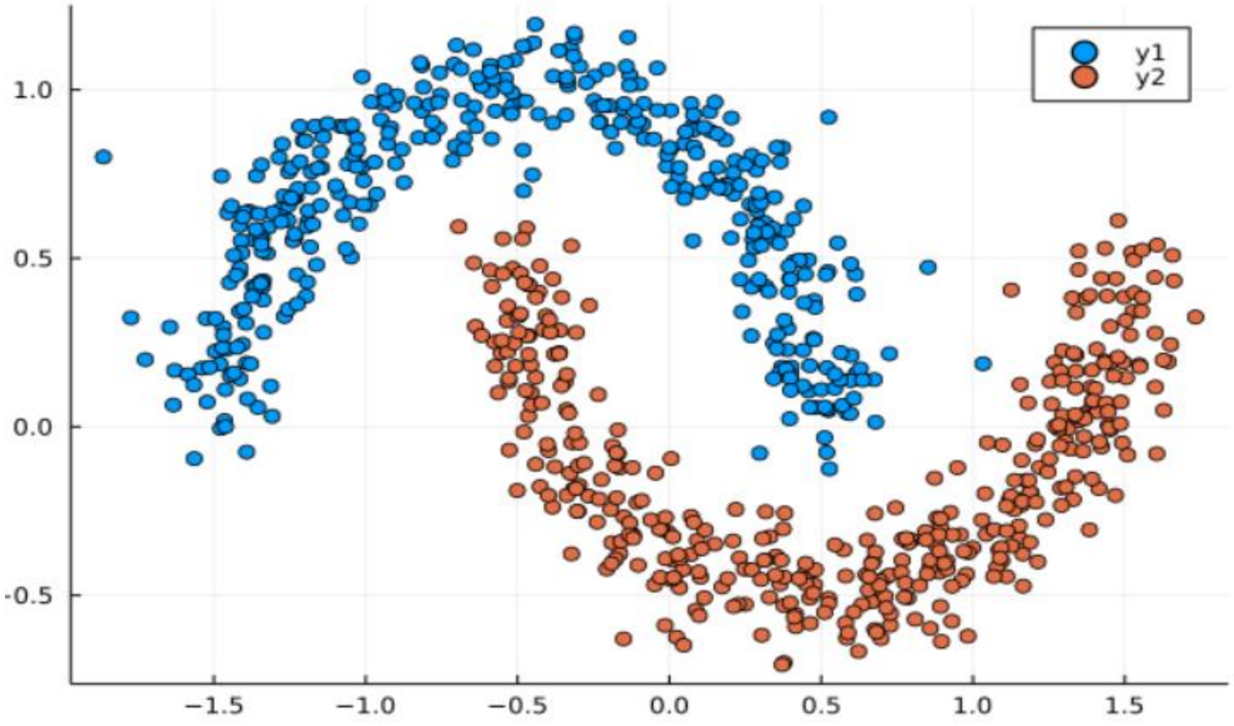
Test Problems - Moons

- Popular synthetic data for clustering problems
- Two continuous values, and one discrete class. 3-site MPS: two mapped values of dimension 10, one qubit (for class label).
- True distribution:



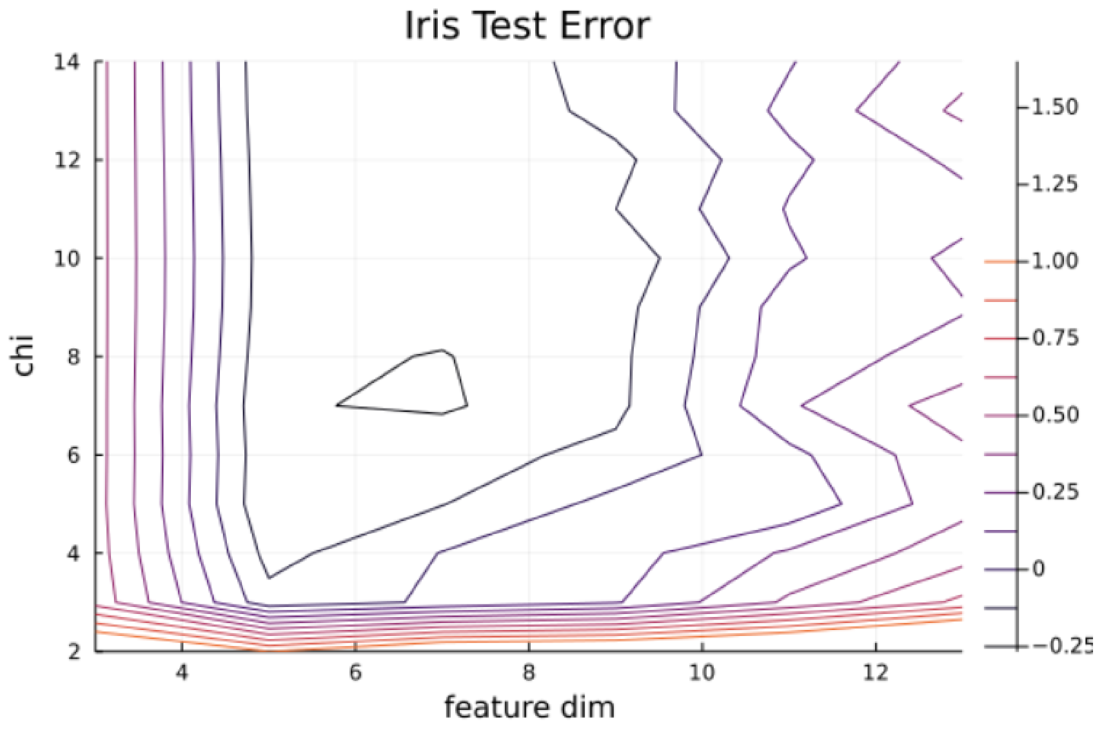
Test Problems - Moons

- Our MPS produces:



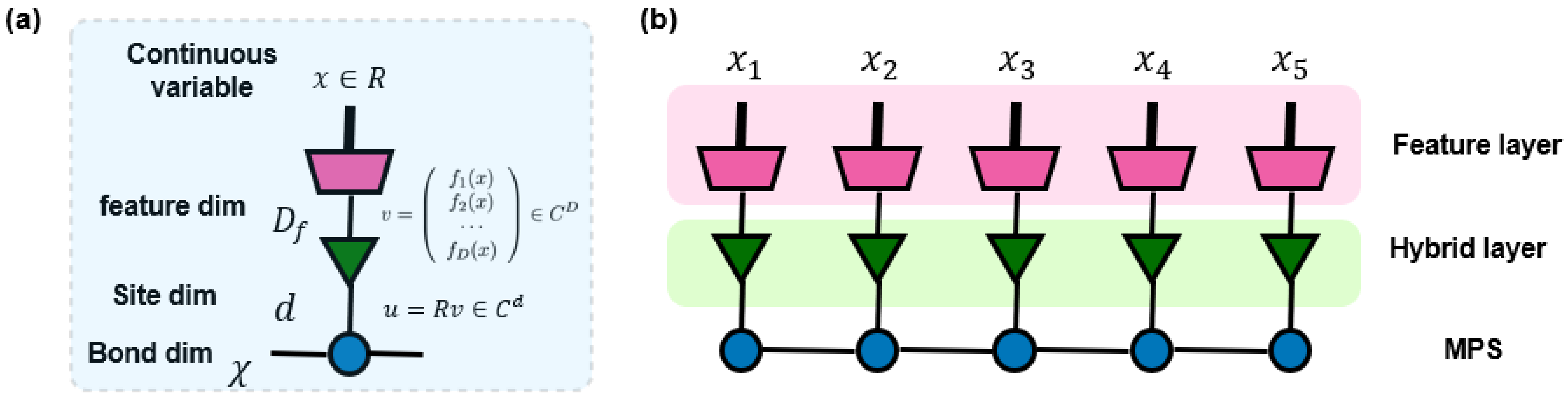
Test Problems – Iris Dataset

Generalization capability – 5-fold cross-validation



Dynamic Basis

- Getting a decent precision on the wave function requires a significant feature dimension
- An error of e requires $d \approx 1/e$ feature functions
  - Training requires SVD on matrices $d \cdot \chi$ by $d \cdot \chi$, or $O(f^3)$ time … not very favorable
- Idea: map from a lower dimension d up to D with a unitary. MPS only has dimension d.



How to Train a Basis?

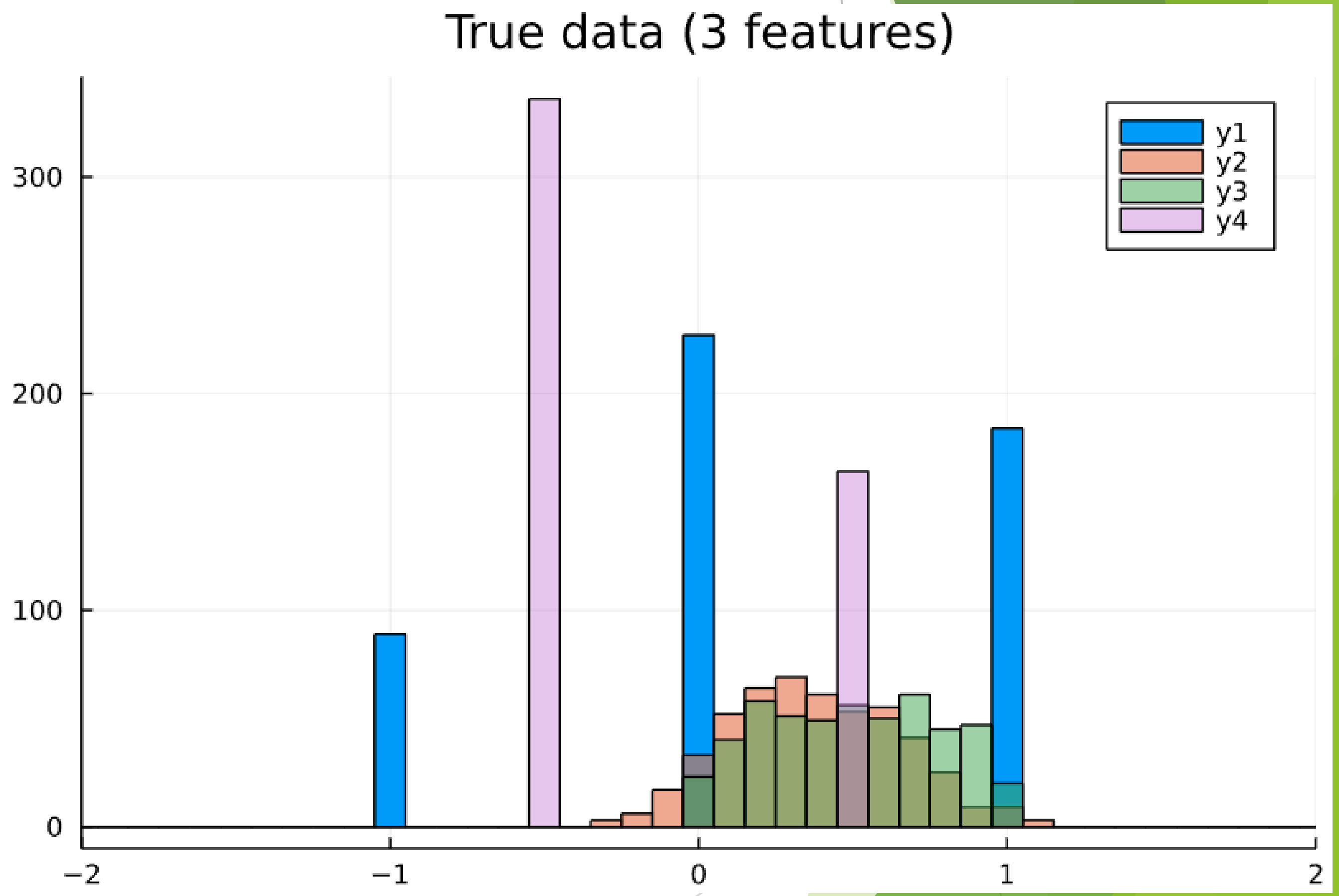
$$U_s = \operatorname{argmax} \sum_{x \in \text{training}} \log\left(|\langle x_s | U_s \, MPS \, | x_{\neg s} \rangle|\right)$$

- Related to some well-studied problems in aligning vectors
- Training loop: correct for phases and nonlinear *log* term, align the vectors using SVD, and repeat.
- Can be thought of as linearizing the NLL and then training a TNS
- Converges in a few iterations
- Alternate between optimizing basis-alignment (the green compression layer, 1-site marginals) and the MPS optimization (inter-site correlations)

ZAPATA

Test Data

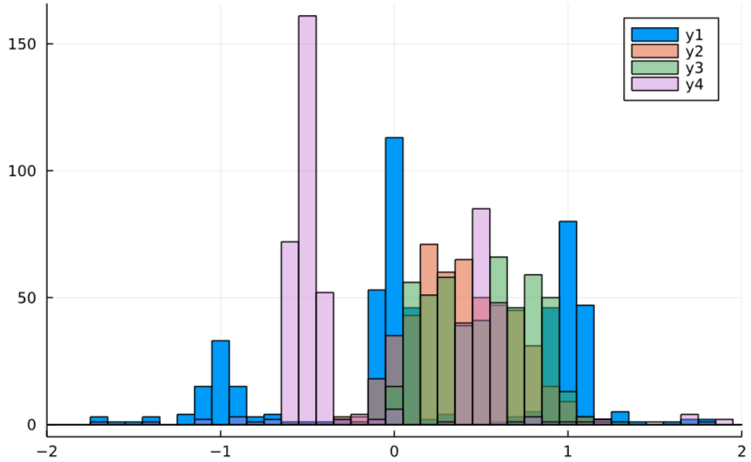
- Small synthetic dataset
- Four features
- Two features are bi- or tri-modal. Can we learn them with just *d*=3?
- For this model, each feature requires its own embedding. They can be shared



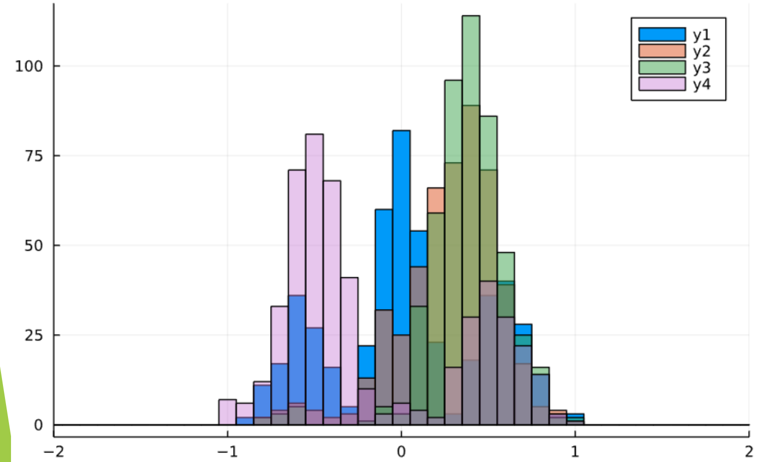
Test Data

- With a dynamic basis, mapping *c*=3 to *f*=16, the MPS can learn it pretty well again!

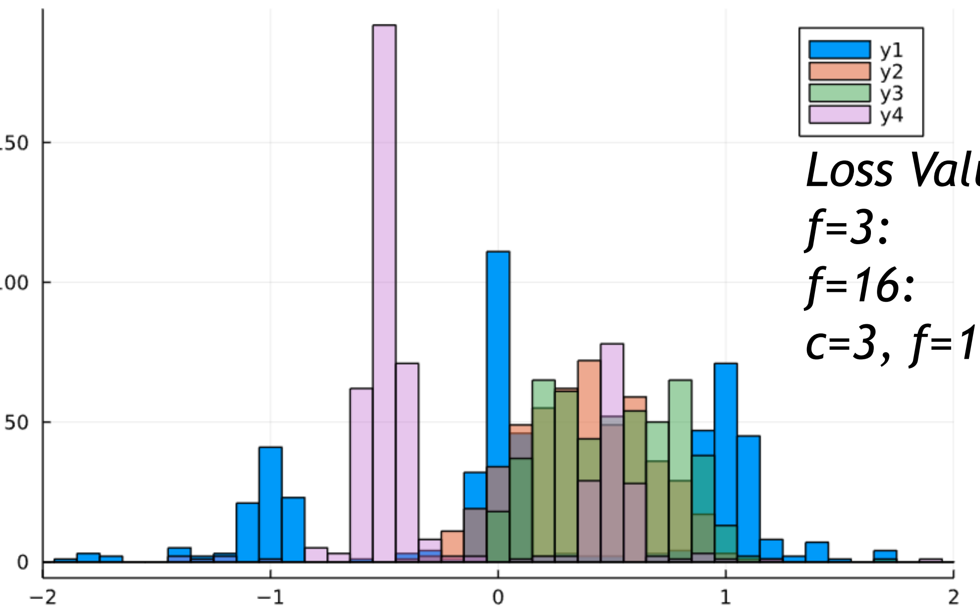
Regular MPS, Feature Dimension 16, Bond Dim 4



Regular MPS, Feature Dimension 3, Bond Dim 4



MPS with Dynamic Basis (3×16 features), Bond Dim 4



Loss Values:

f=3: +2.04

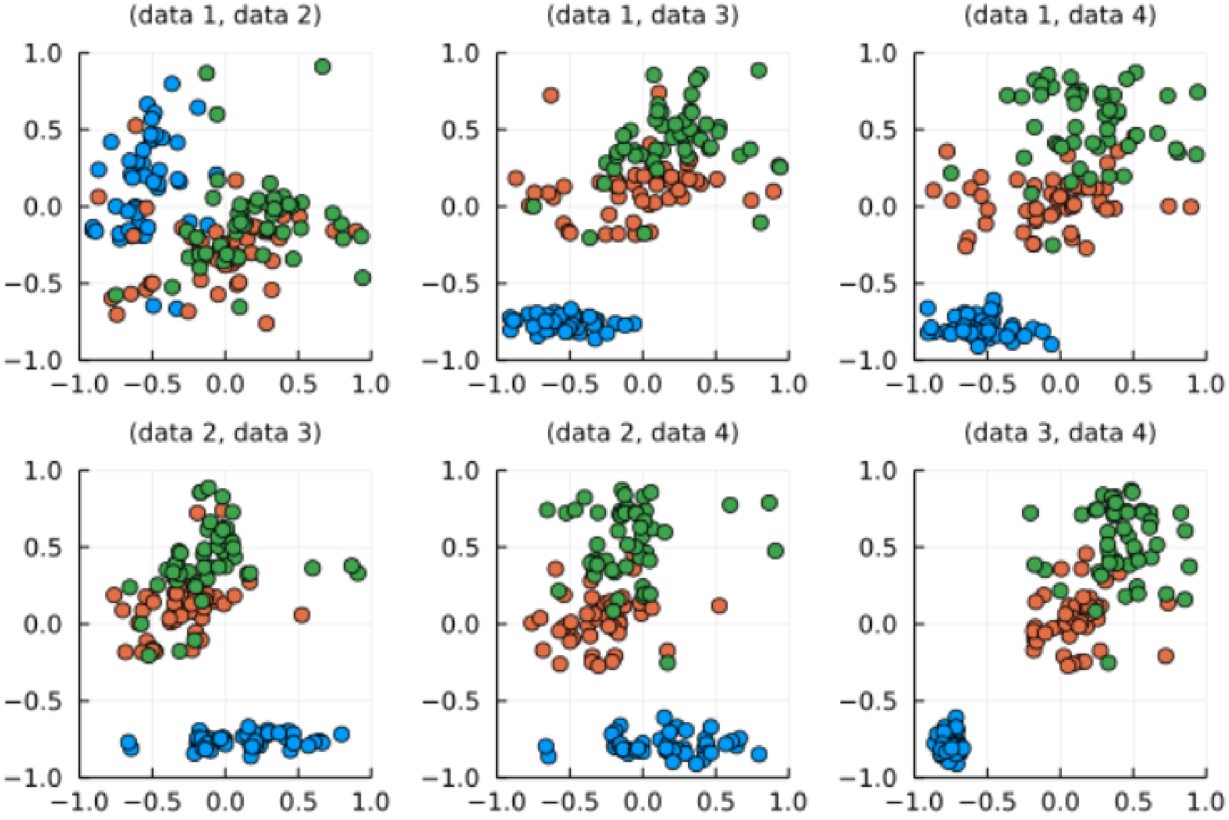
f=16: -2.17

c=3, *f*=16: -2.06

ZAPATA

Dynamic Basis - Iris

- With *d*=4 and a dynamic basis, can match the performance of *d*=8



ZAPATA

Conclusions

- MODELS DO LEARN TO GENERALIZE
- CAN BE EFFECTIVELY AND EFFICIENTLY COMPRESSED WITH A DYNAMIC BASIS
- FLEXIBLE CHOICES OF BOND DIMENSION, FEATURE TYPE, TNS TOPOLOGY
- COMPATIBLE WITH MIXED DISCRETE DATA, HETEROGENEOUS DATA TYPES AND SUPPORTS
- CAN WORK WITH ANY EXISTING MPS TRAINING STRATEGY (DMRG, SGD..)

Thank you!

… for staying until the end of the session. ☺

Paper will be on arXiv soon!

Collaborators:
Jing Chen, Jacob Miller, Alejandro Perdomo-Ortiz

ZAPATA