# Data Re-Uploading on Qudits


Noah L. Wach[1], Manuel S. Rudolph[1,2], Sebastian Schmitt[3], Fred Jendrzejewski[1]

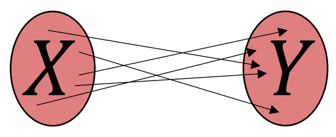
[1]Kirchhoff-Institut für Physik, Universität Heidelberg, Im Neuenheimer Feld 227, 69120 Heidelberg, Germany
[2]Zapata Computing Canada Inc., 325 Front St W, Toronto, ON, M5V 2Y1, Canada
[3]Honda Research Institute Europe GmbH, Carl-Legien-Str. 30, 63073 Offenbach, Germany
noah.wach@stud.uni-heidelberg.de


UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

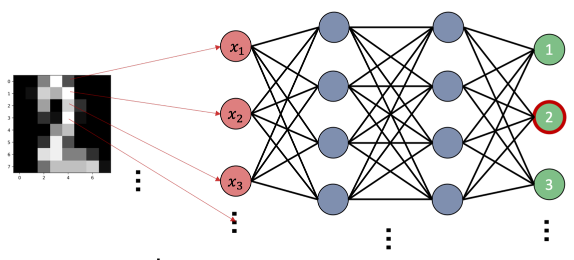
KIRCHHOFF-INSTITUT FÜR PHYSIK

SynQS

## Quantum Algorithm - Data Re-Uploading



- Belongs to the family of supervised machine learning algorithm
- Objective: Find map between data $X$ and labels $Y$
- Quantum algorithm consists of multiple layers (similarly to classical neural net)



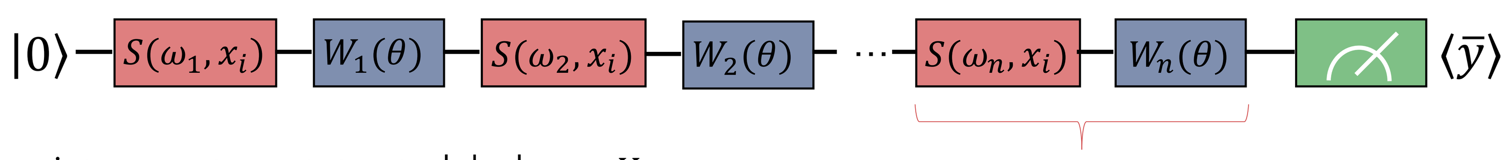
Each layer contains two steps:

1. $\hat{S}(\omega_i x_i)$ Encode datapoint $x_i$ with weight $\omega_i$ using rotations
   - Use alternating $\mathcal{R}_x(x)$ and $\mathcal{R}_z(x)$ rotations
2. $\hat{W}(\boldsymbol{\theta})$ Manipulate quantum state using trainable parameters $\boldsymbol{\theta}$ to end up in desired state $|y\rangle$
   - Structure: $\mathcal{R}_x(\theta)\,\mathcal{R}_z(\theta)\,\mathcal{R}_x(\theta)\,\mathcal{R}_{zz}(\theta)$

- Through repeated application of layers algorithm gains expressivity

This results in:

$$|\psi(\boldsymbol{x_i}, \boldsymbol{\theta}, \omega)\rangle = \hat{S}(\omega_1 \boldsymbol{x_i})\hat{W}_1(\boldsymbol{\theta}) \dots \hat{S}(\omega_n \boldsymbol{x_i})\hat{W}_n(\boldsymbol{\theta})|\psi_0\rangle = \hat{U}(\boldsymbol{x_i}, \boldsymbol{\theta}, \omega)|\psi_0\rangle$$

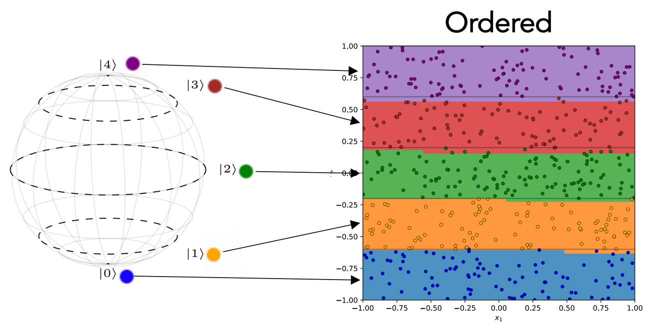



Layer $n$ with one encoding and one training block

- Associate quantum states to labels $y \in Y$
- Address $d$-class classification problem with one qudit
- **Goal**: Maximize overlap $P$ with desired state $|\boldsymbol{y}\rangle$
- $P(y|\boldsymbol{x_i}, \boldsymbol{\theta}, \omega) = |\langle y|\psi(\boldsymbol{x_i}, \boldsymbol{\theta}, \omega)\rangle|^2$

## Simple Classification Tasks

Goal: Classify points in 2D plane into one of $n$ classes depending on their $x$-$y$ position

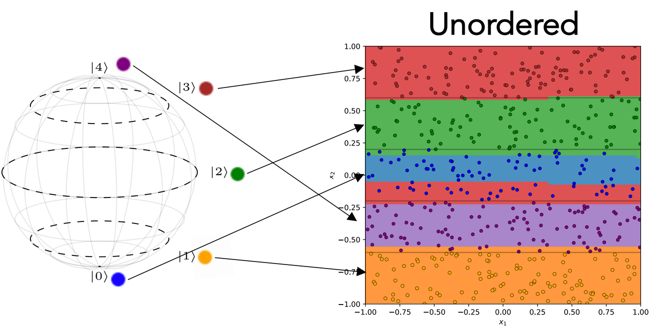



### Two Cases:

1. Ordered stripes

   Order of stripes corresponds to order on qudit

   → 3 layers are sufficient to achieve accuracies above 95%




2. Unordered stripes

   Order of stripes does not correspond to order on qudit

   → Over 6 layers are needed to achieve accuracies above 80%

Algorithm tries to suppress in between classes
→Succeeds with more layers

Model has **bias** towards data matching order of labels on qudit

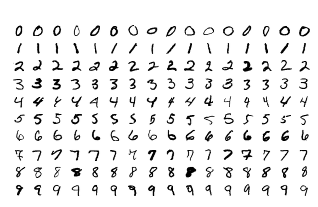
**Bias can be reduced by applying more layers**

## Classifying MNIST

### How does the algorithm perform on a real-world dataset?

Learn the parts of the **MNIST** dataset (digits 0, 1, 2)

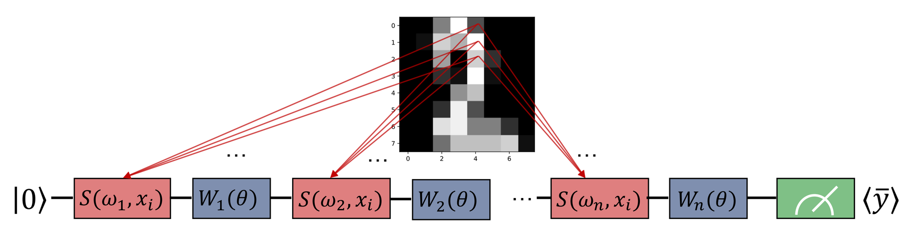
Qudit state corresponds to label ($|0\rangle \hat{=} 0, \dots$), dimension of qudit $d = 3 \rightarrow \ell = 1$



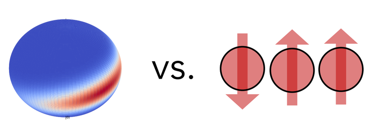
LeCun, Yann and Cortes, Corinna. "MNIST handwritten digit database." (2010)

**Setup:**

1. Reduce dataset to 3 dimensions using PCA
2. Use model with 2 layers
3. Train model over 50 epochs
4. Test on unseen data



**Preliminary Results:**



Accuracy on exact simulator: 97% vs. Accuracy on IBMQ hardware: 97%

For small $d$, hardware not a limiting factor → Further expand model on entire dataset

## Conclusion & Outlook

- It is possible to learn on real quantum hardware
- Qudit approach improves stability of algorithm → performs well on current NISQ devices

Future Goals:

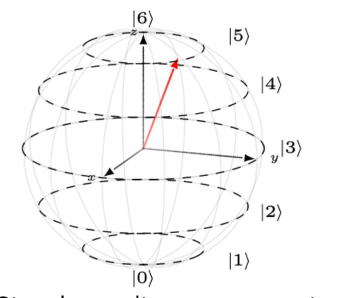
- Expand Model onto entire MNIST dataset
- Improve reliability of quantum algorithm

## Qudits

**General description:**

- $d$-level generalization of qubit
- Represented by vector in the d-dimensional Hilbertspace:

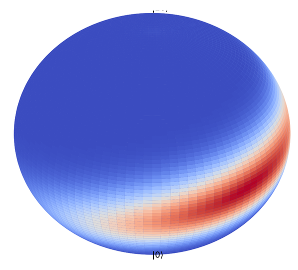
$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle + \dots c_{d-1}|d-1\rangle$$



Simple qudit representation

- Can be represented on a Bloch sphere via Husimi-Q distribution (quasi probability distribution)
- Qudits are controlled via $d$-dimensional angular momentum operators of total angular momentum $\ell = \frac{d-1}{2}$, and angular momentum quantum number $m = z - \frac{d-1}{2} \in \{-\ell \ \dots, \ell\}$

$$z \in \{0, \dots, d-1\}$$



Husimi-Q distribution

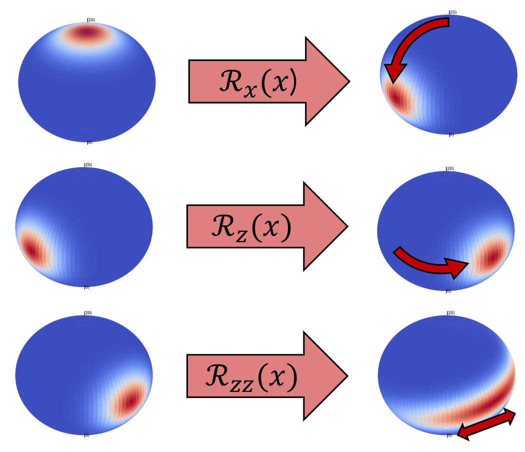
*Operations*

$$L_+|m,l\rangle = \sqrt{(\ell - m)(\ell + m + 1)}|\ell, m+1\rangle$$
$$L_-|m,l\rangle = \sqrt{(\ell + m)(\ell - m + 1)}|\ell, m-1\rangle$$

- Achieve universality using three operations:

1. X-Rotation $L_x = \frac{1}{2}(L_+ + L_-)$
2. Z-Rotation $L_z|\ell, m\rangle = m|\ell, m\rangle$
3. One-axis twisting $L_z^2$

Generate rotations through time evolution of operators $\mathcal{R}_i(x) = e^{-ix\hat{L}_i}$



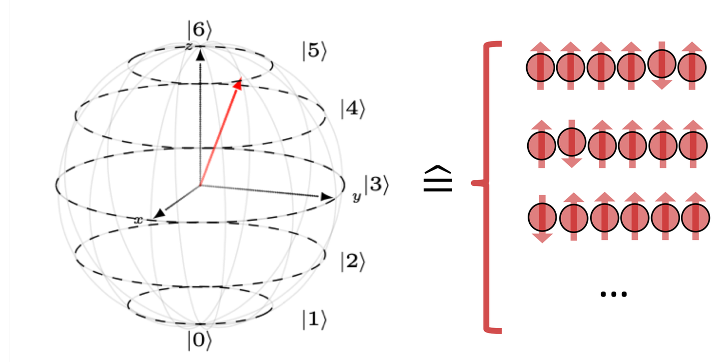

Connection to qubits:

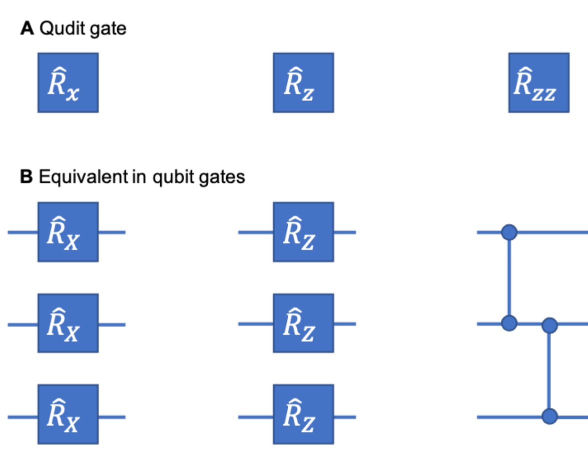
- Running the data re-uploading algorithm on qubit based NISQ devices

→Represent one qudit using multiple qubits

Example:

$|5\rangle \hat{=}$





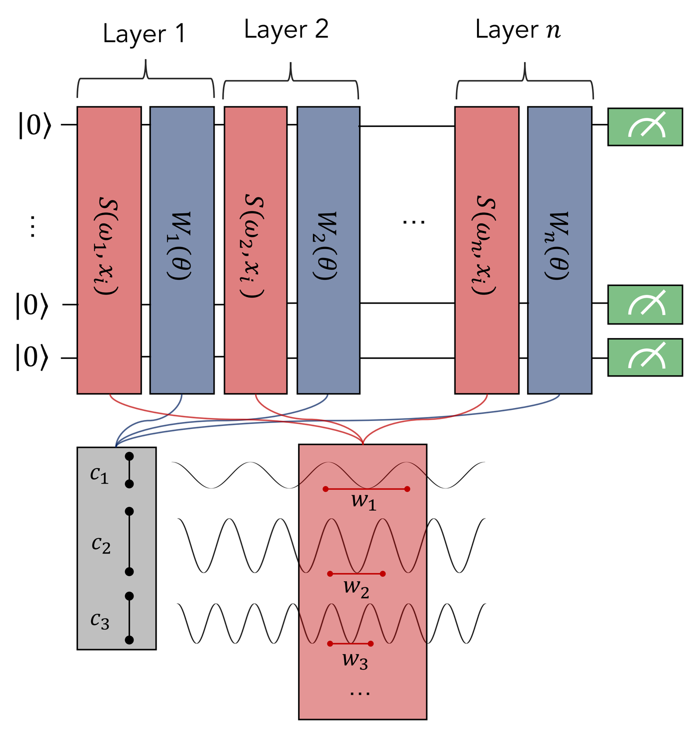

**Can we construct a quantum machine learning algorithm using qudits?**

## Expressivity of Data Re-Uploading Circuits

- Express quantum model as partial Fourier series
- Repeated encoding and train blocks increases expressivity

In our case: Number of Fourier coefficients $= 2 \cdot L \cdot \ell$

with $L$ as number of layers and qudit spinlength $\ell$




Two approaches:

1. Train and evaluate using exact state vector representation

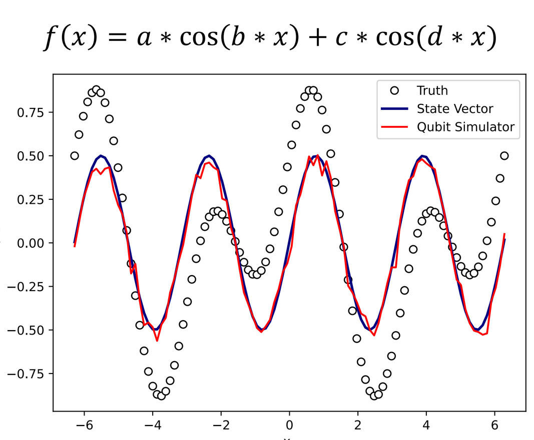
   → Exact, no noise, can be simulated easily

2. Train and evaluate using IBMQ qubit simulator

   → Noisy, is evaluated statistically using shots, can also be run on real hardware

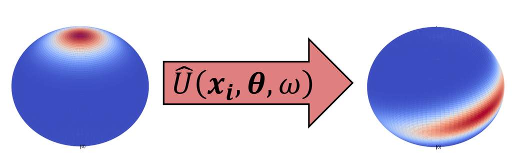
### Learn exact Fourier coefficients

$f(x) = a * \cos(b * x) + c * \cos(d * x)$

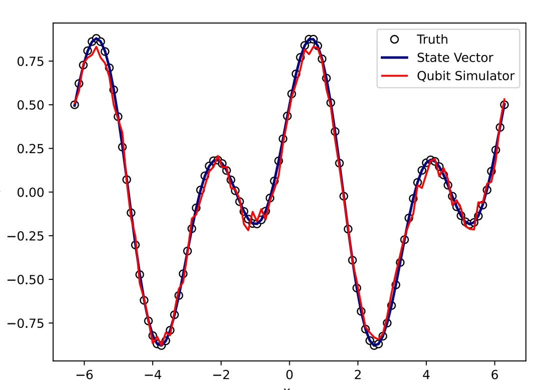



Model with $\ell = 1$ and $L = 1$

→ not enough Fourier coefficients for approximation




$\rightarrow \langle \hat{L}_z \rangle \hat{=}$ prediction




Model with $\ell = 1$ and $L = 2$

→ successful approximation

Qubit simulator inaccuracies are due to the shot noise

**Model can learn any number of Fourier coefficients of dataset with sufficiently many layers**

Honda Research Institute EU

Emmy Noether-Programm DFG