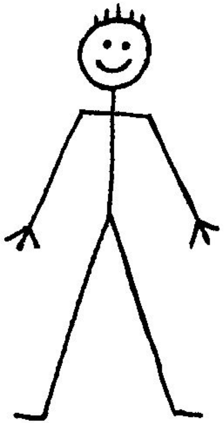
# Exploiting problem symmetry for faster quantum machine learning

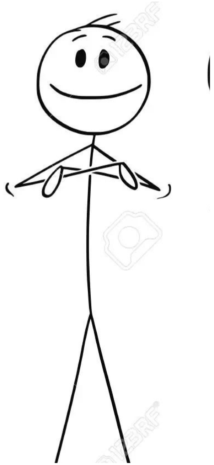
F. Mrcarica, M. Hanisch, P. Uriarte Vicani, V. Mohr

# The team:

## CQC

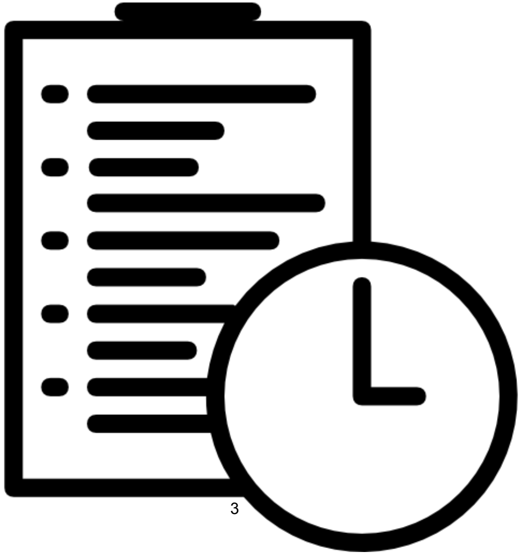
**Filip M.** **Maurice H.** **Paul U. V.** **Vinícius M.**





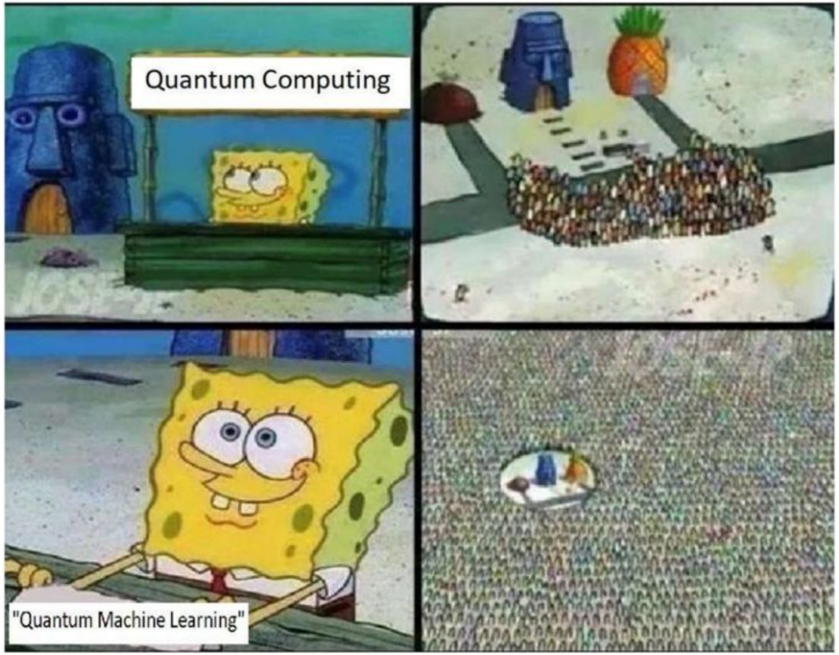
# Agenda

- Introduction/Background
- Tic-Tac-Toe problem
- Results & Layering
- Practical application: Maze
- Conclusion



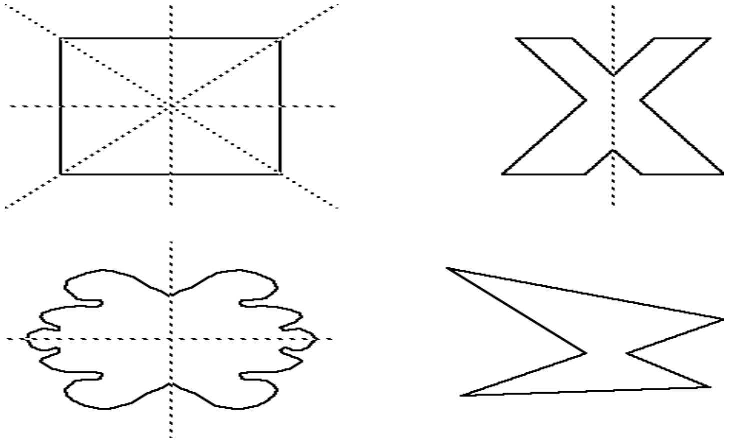
# Introduction/Background

## Quantum machine learning summarized:




Source: Arthur Pesah, https://arthurpesah.me/assets/pdf/qml-talk-vannes-2020.pdf

# Introduction/Background



**Symmetries:**

-Usual in **physics**.

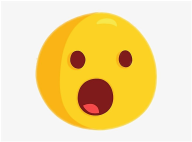
-For **ML**:

1) Reduce the number of parameters: **Expressivity** (less overfitting, still in right solution space)

2) Fewer data to achieve better validation err.: **Generalization**

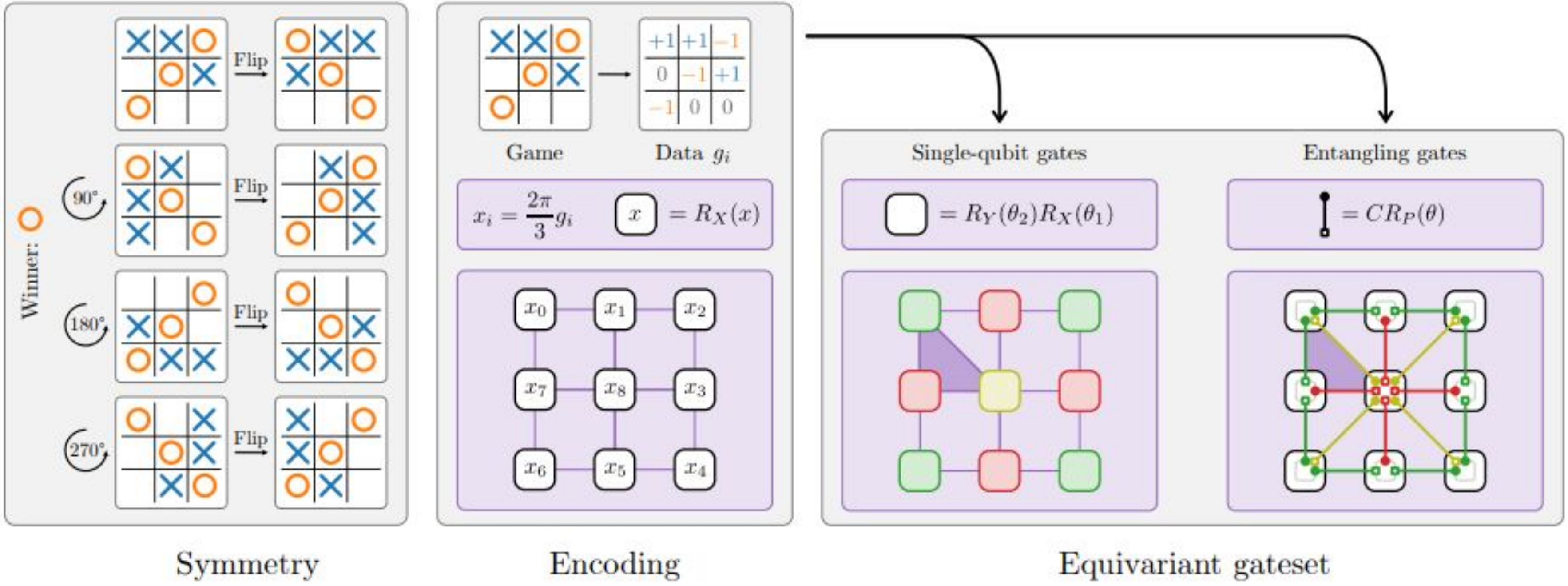
# Introduction/Background

## What we did:

- **Analyzed paper** that showed accuracy increase through use of symmetry
- **Simulated Q. ML** with our laptops 
- Showed **symmetry advantage** for completely **new**, very **generalizable** problem

-**Learned a lot!**

# Tic-Tac-Toe: Classification Problem

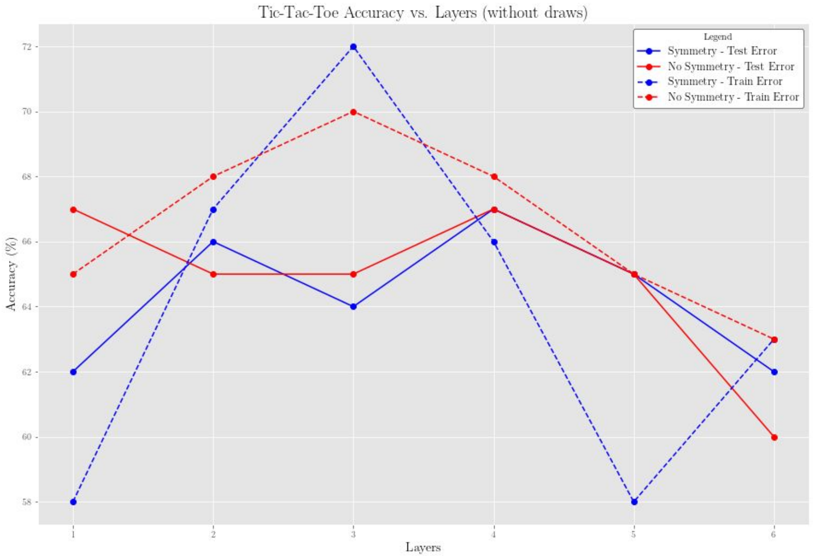


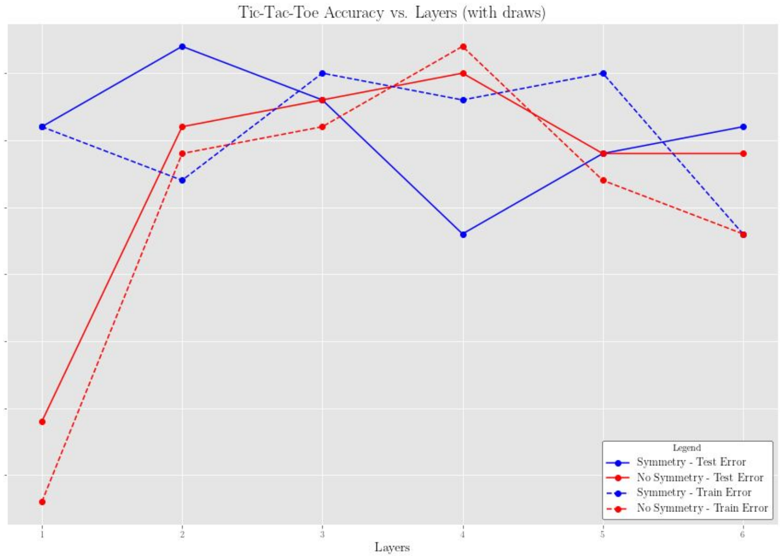
# Tic-Tac-Toe: Classification accuracy for simple optimization algorithm with one layer

| | No draws | Draws |
|---|---|---|
| Symmetries | 62 % Test / 58 % Train | 43 % Test / 43 % Train |
| No Symmetries | 67 % Test / 65 % Train | 32 % Test / 29 % Train |

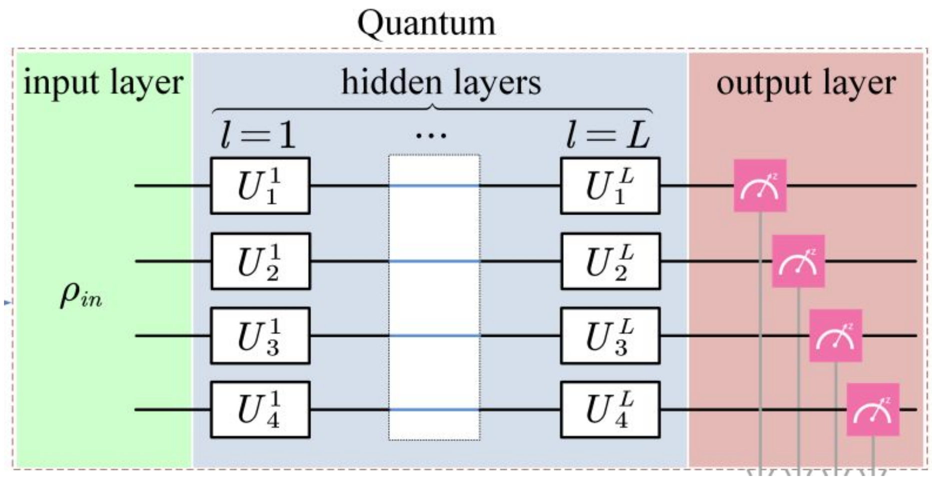
ETH zürich

# Tic-Tac-Toe: Accuracy for additional layers



# Potential Problems with layering: More Parameters to fit




- More calculations to execute
- Lose generalization
- More risk of overfitting
- More gates to execute

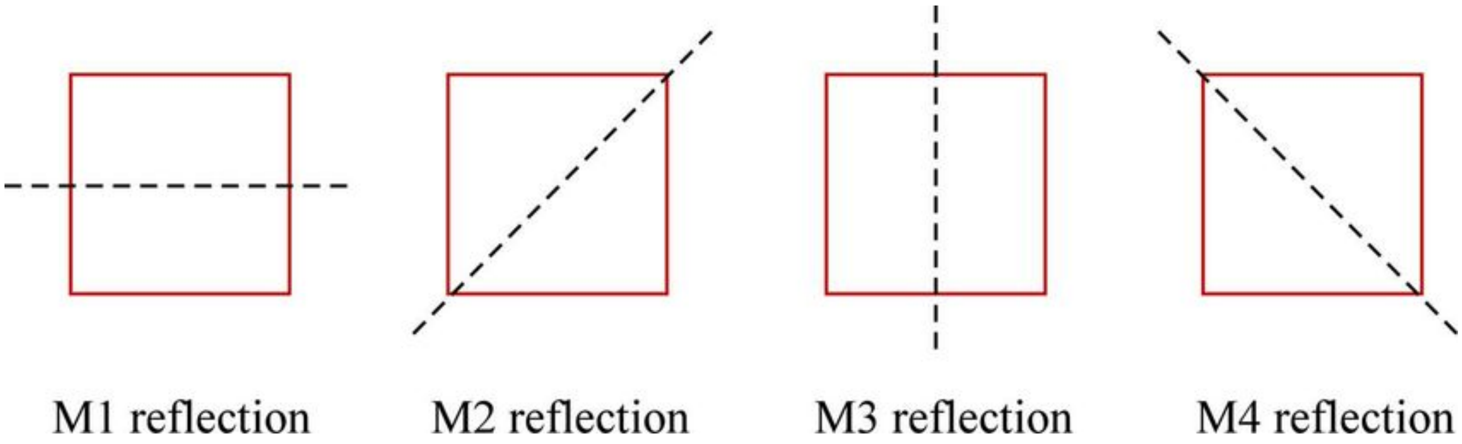
→ Can lead to more noise

ETH zürich

# Practical Application: Can you use Quantum Machine Learning to escape a maze?

# Practical Application: Can you use Quantum Machine Learning to escape a maze?

Answer: **Yes!** But using **symmetries** even better!



# Practical Application: Can you use Quantum Machine Learning to escape a maze?

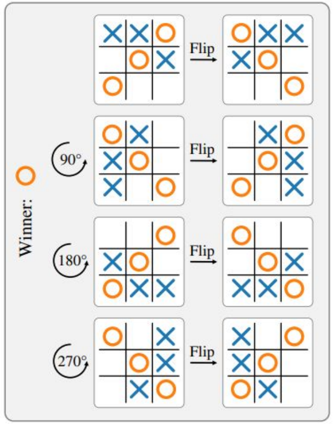
Verity AG: ETH Spin-off

## Why a maze?

- Great **generalization**

- With **same symmetry** group as TTT (Dihedral D4)

- **Scalability** (not like TTT)



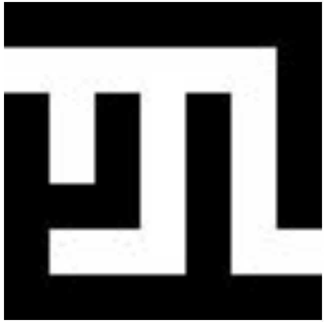

Symmetry

# Practical Application: Can you use Quantum Machine Learning to escape a maze?

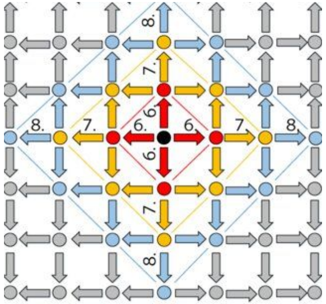
## How did we do it?

- Classical **Generation**:

All possible combinations



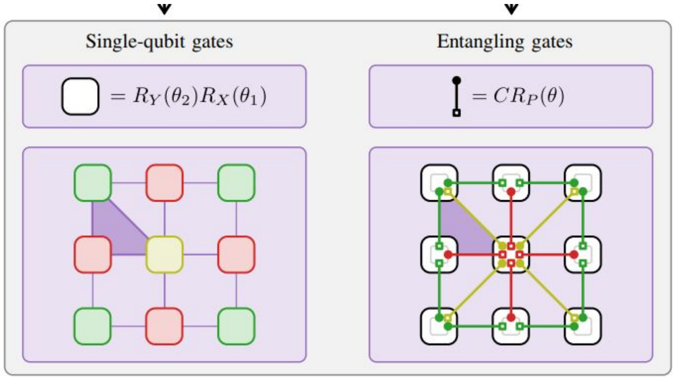
Check: Iterative steps



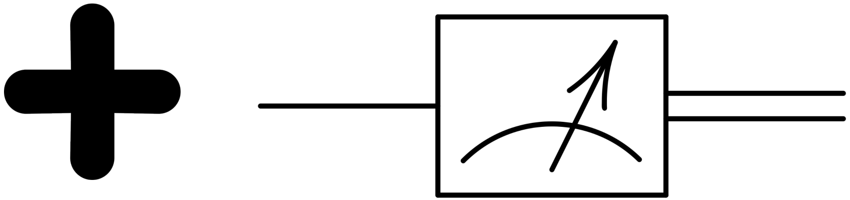
# Practical Application: Can you use Quantum Machine Learning to escape a maze?

## How did we do it?

- Quantum **Simulation**:



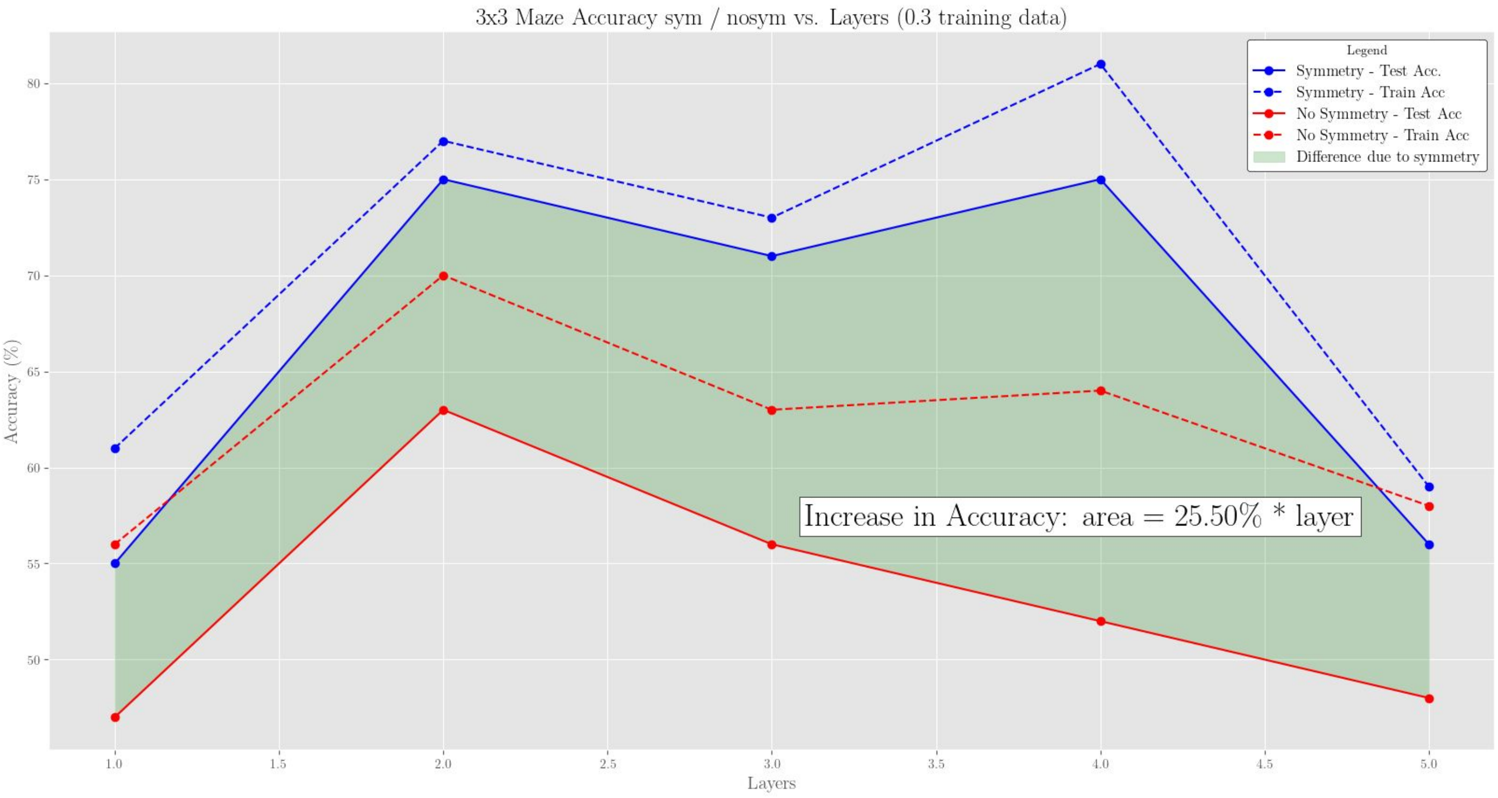

Equivariant gateset



# Practical Application: Can you use Quantum Machine Learning to escape a maze?

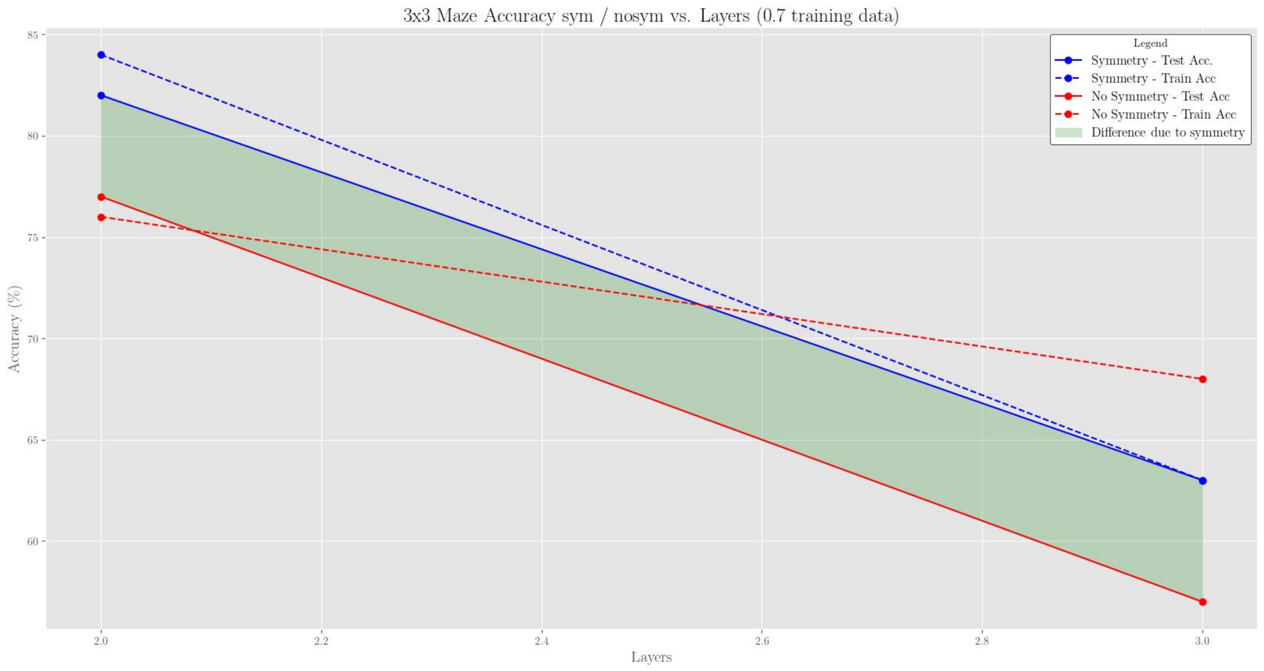
## Results!!

Significant increase in accuracy using **symmetries**

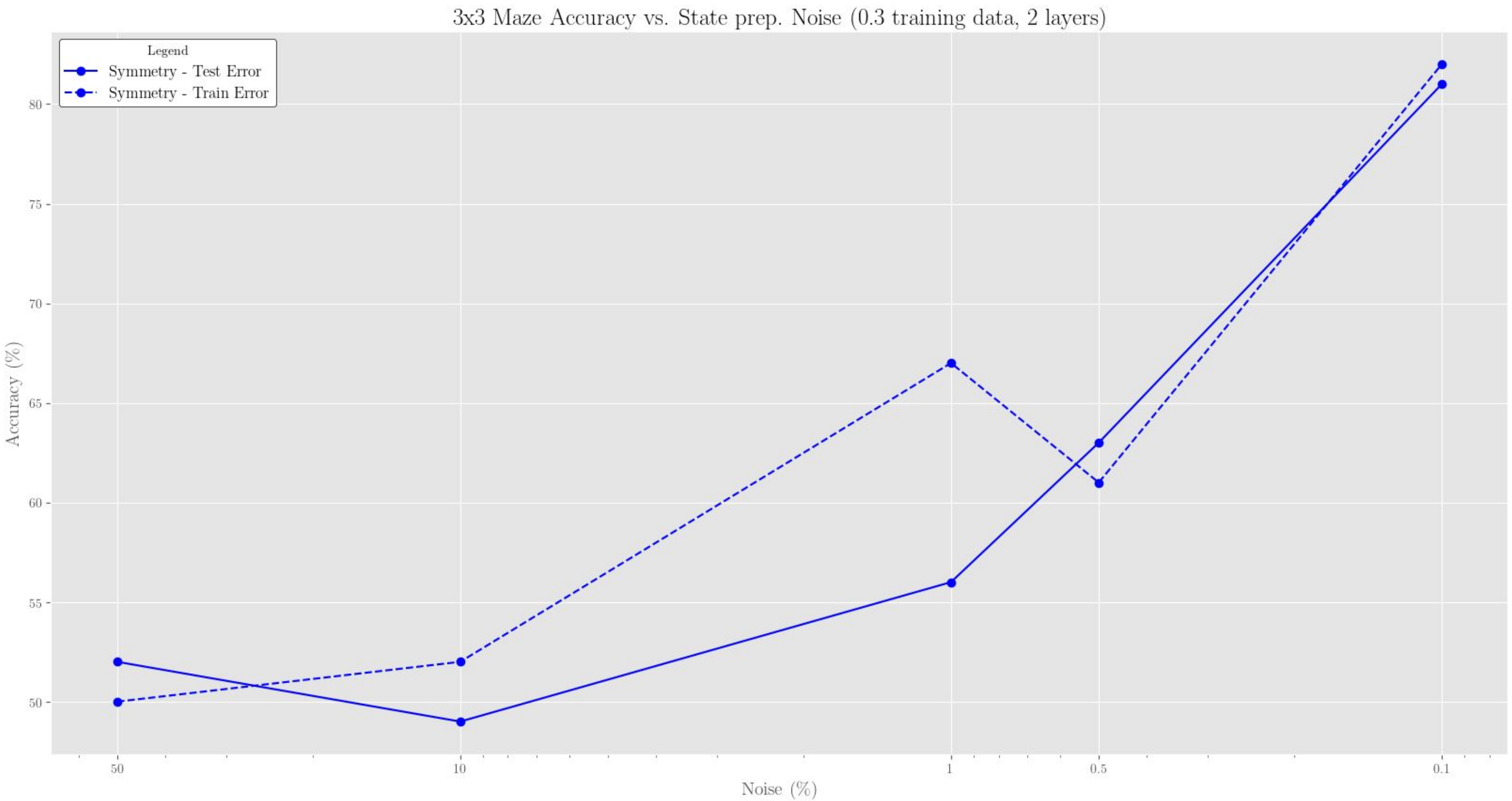


# Practical Application: Can you use Quantum Machine Learning to escape a maze?

**Results!!** (also consistent for **more training data**)



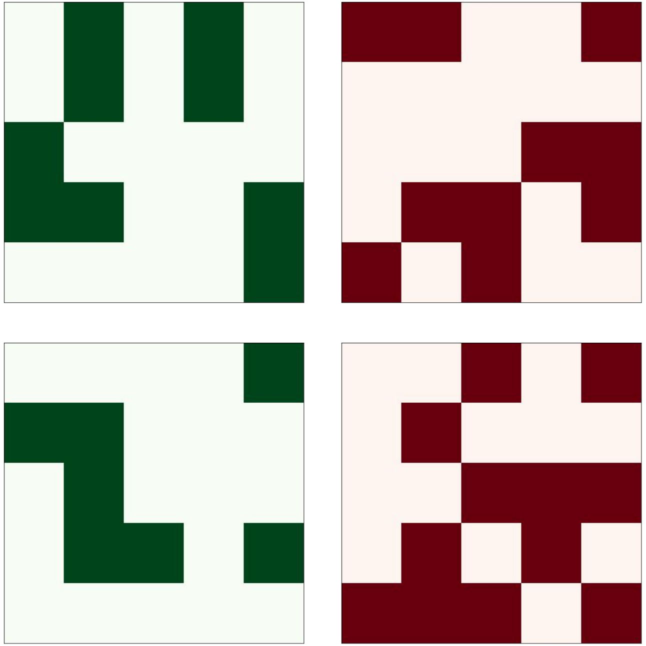
# Further steps: Noise in 3x3

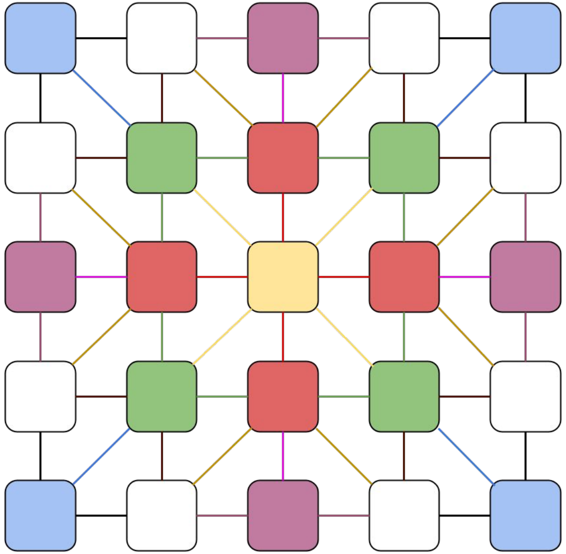


# Further steps: 5x5 Maze

(only managed to train **5/100** iterative steps in **7h30** on EULER)

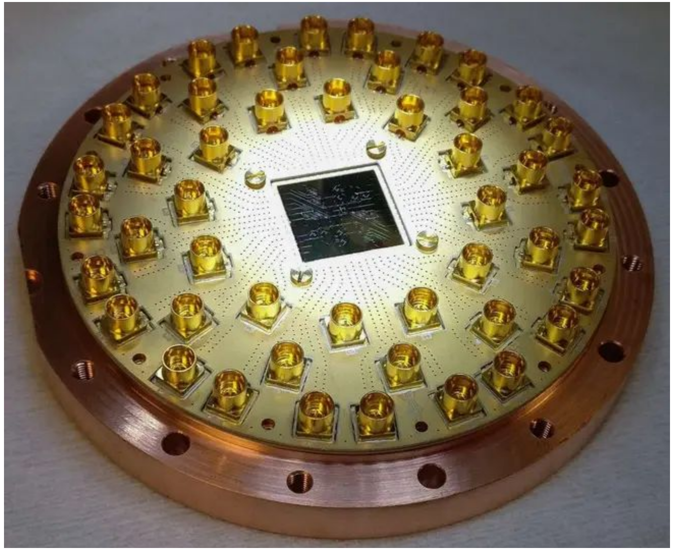
```
[mhanisc@eu-login-43 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          16197709 normal.24 script_u  mhanisc  R    7:30:52      1 eu-g5-001-2
```





# Outlook

- Introduce further noise analysis
- Generalization to real-life problems
- Look at larger quantum systems with more qubits



➔ Run the codes on an actual quantum computer

# Conclusion: What we learned

- Basics of quantum machine learning
- Importance of symmetries
- Implementing quantum neural networks
- Application to real-world problems
- Open-end: Only limiting factor was our creativity…

… and number of qubits