# Quantum Information Sciences

17. September, 2024; rev. 15. Januar 2026

Linda Riesen (rieselin)

# 1 Vorlesung Introduction

## 1.1 Begriffe

- Bit: Data Structure, two state system, two voltage levels
- Qubit (Quantum bit): a qubit can be in a superposition (a combination) of 0 and 1
  - Mathematically, we represent a qubit as a vector in a two-dimensional complex (= can have irrell numbers) vector space (Hilbert space). $|q\rangle = c_0|0\rangle + c_1|1\rangle$ mit [$c_i \in \mathbb{C}$ und $|c_0|^2 + |c_1|^2 = 1$]
  - Qubit nur proper"wenn: $|\,|q\rangle\,|^2 = 1$
  - Physically: A spin-1/2 system e.g. Electron (spin = abstract property)
  - Superposition vs. pure basis state is a matter of perspective (basis choice). The physical reality is just the quantum state vector; “superposition” is about how you write it down.
- The scalar product is only symmetric under complex conjugation.
- Base states are not linear combinations of other bases states!
- Three qubits constitute 8 base states. For each base state, we have complex coefficient which is defined by two real parameters. Because of the normalization, we have 7 free parameters.
- Quantum Register (Coupled Qubits): Qubits can be grouped into a register, this can (unlike classical register) represent a mixture of base states
- Power by Linerarity: Quantum computing is powerful because unitary operations are linear. This means they apply to all components of a superposition at the same time, giving exponential parallelism (one operation acts on $2^n$ states for $n$ qubits), while still keeping the result structured and interpretable.
- Verwirrung mit $H$: kann sein: Hamiltonian operator, Hermitian Conjucation, Hadamard gate
- Harmitian = Self-Adjoint Operator: $(Af, g) = (f, Ag)$ in Matrices: $A = A^H <=> A_{ij} = \bar{A}_{ji}$ If H is self-adjoint operator:
  - the eigenvalues $E_r$ of $H$ are all real numbers
  - the eigenfunctions $\varphi_r$ of $H$ form an orthogonal basis.
  - if $H$ does not depend on time (no variable $t$), the eigenfunctions $\varphi_r$ are also independent of time.
  - the eigenfunction can be normalized $(\varphi_r, \varphi_r) = \delta_{rs}$
  - the eigenfunctions and values depend on $V(x, y, z)$.

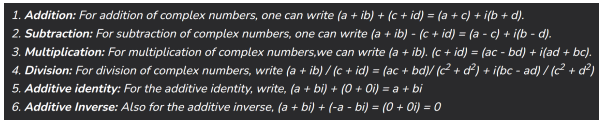



Abbildung 1: Complex Numbers Calculation Rules

Quantum Composer Thing
Complex Number Calculator
Matrix Calculator

### 1.1.1 Operation 1: Quantum Mechanics: Measurements

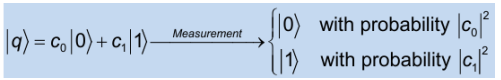



Abbildung 2: Quantum Mechanics: Measurements

**Stochastic and Irreversible**

- Before measurement: $|q\rangle$ is a linear combination of $|0\rangle$ and $|1\rangle$
- After the measurement: $|q\rangle$ is either in state $|0\rangle$ or $|1\rangle$

$$U^\dagger U = I \tag{1}$$

Abbildung 3: † = Conjugate Transpose = Transponieren und alle complexen zahlen complex conjugate of each entry (if $z = a + bi$, then $\bar{z} = a - bi$)

  - After a measurement, we have a state, which is always normalized.

- The probability of ending in $|0\rangle$ is equal to $|c_0|^2$ and in $|1\rangle$ to $|c_1|^2$

#### 1.1.2 Operation 2: Unitary Transformation

**Deterministic and Reversible**

- Reversible because: follows from schroedingers equation, and observation and conservation laws (probability, energy) (=> otherwise information would be able to be destroyed)

Unitary = preserve length (norm)

- A unitary transformation rotates or reflects a quantum state $|\psi\rangle$ without changing its length (so probabilities remain valid).

- The condition

  ensures normalization.

- Any unitary can be generated from some Hermitian Hamiltonian $H$ via

$$U = e^{-iH}.$$

### 1.2 Quantum State Notation and Vector Representation

Quantum states are written using **Dirac notation** (bra-ket notation). The basic building blocks are:

#### 1.2.1 Single Qubit States

A qubit has two basis states:

- $|0\rangle$ represents the quantum state żero"

- $|1\rangle$ represents the quantum state öne"

- $|+\rangle$ represents superposition of $|0\rangle$ and $|1\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$

- $|-\rangle$ represents superposition of $|0\rangle$ and $|1\rangle = H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$

These correspond to column vectors:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \tag{2}$$

#### 1.2.2 Multi-Qubit States

Multiple qubits are combined using the **tensor product** operator $\otimes$. For a two-qubit system, we have four basis states:

$$|0\rangle_2 = |0\rangle \otimes |0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3}$$

$$|1\rangle_2 = |0\rangle \otimes |1\rangle = |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{4}$$

$$|2\rangle_2 = |1\rangle \otimes |0\rangle = |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{5}$$

$$|3\rangle_2 = |1\rangle \otimes |1\rangle = |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{6}$$

**Notation:** In $|k\rangle_n$, the subscript $n$ indicates an $n$-qubit system, and $k$ is the decimal representation of the binary state. For example, $|2\rangle_2$ corresponds to binary 10.

### 1.2.3 General n-Qubit Systems

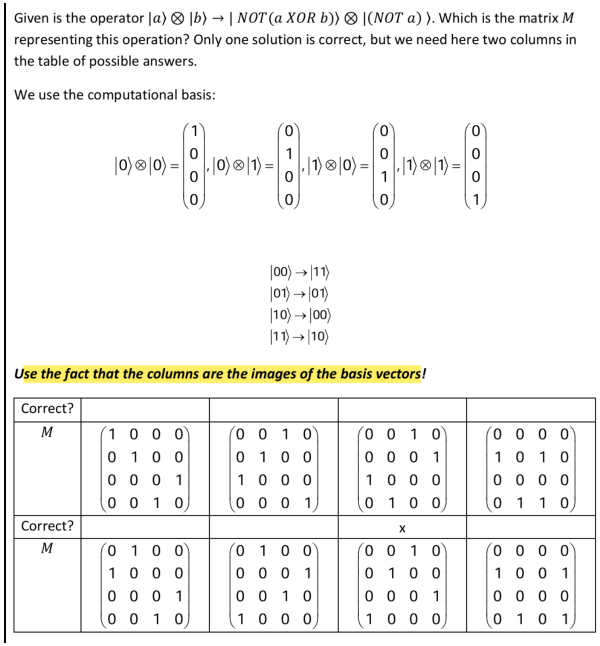
For an $n$-qubit system:

- There are $2^n$ basis states labeled $|0\rangle_n, |1\rangle_n, \ldots, |2^n - 1\rangle_n$
- Each state is represented as a $2^n$-dimensional column vector
- The label $k$ in $|k\rangle_n$ corresponds to the binary representation of the qubits

**Example:** For a 3-qubit system ($n = 3$), we have 8 basis states where $|5\rangle_3 = |101\rangle$ is an 8-dimensional vector with a 1 in the 6th position (counting from 1).

### 1.2.4 Vector Interpretation

Each entry in the state vector indicates the amplitude for that basis state. For pure basis states, exactly one entry is 1 and all others are 0. Quantum operators (represented as matrices) act on these state vectors through standard matrix multiplication.



Given is the operator $|a\rangle \otimes |b\rangle \rightarrow |\ NOT(a\ XOR\ b)\rangle \otimes |(NOT\ a)\ \rangle$. Which is the matrix $M$ representing this operation? Only one solution is correct, but we need here two columns in the table of possible answers.

We use the computational basis:

$$|0\rangle\otimes|0\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |0\rangle\otimes|1\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, |1\rangle\otimes|0\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, |1\rangle\otimes|1\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

$|00\rangle \rightarrow |11\rangle$
$|01\rangle \rightarrow |01\rangle$
$|10\rangle \rightarrow |00\rangle$
$|11\rangle \rightarrow |10\rangle$

***Use the fact that the columns are the images of the basis vectors!***

| Correct? | | | | |
|---|---|---|---|---|
| $M$ | $\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$ | $\begin{pmatrix}0&0&1&0\\0&1&0&0\\1&0&0&0\\0&0&0&1\end{pmatrix}$ | $\begin{pmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{pmatrix}$ | $\begin{pmatrix}0&0&0&0\\1&0&1&0\\0&0&0&0\\0&1&1&0\end{pmatrix}$ |
| Correct? | | | x | |
| $M$ | $\begin{pmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$ | $\begin{pmatrix}0&1&0&0\\0&0&0&1\\0&0&1&0\\1&0&0&0\end{pmatrix}$ | $\begin{pmatrix}0&0&1&0\\0&1&0&0\\0&0&0&1\\1&0&0&0\end{pmatrix}$ | $\begin{pmatrix}0&0&0&0\\1&0&0&1\\0&0&0&0\\0&1&0&1\end{pmatrix}$ |

Abbildung 4: What is the operator

**Columns of Operator are images of basis vector**

### 1.2.5 Notation

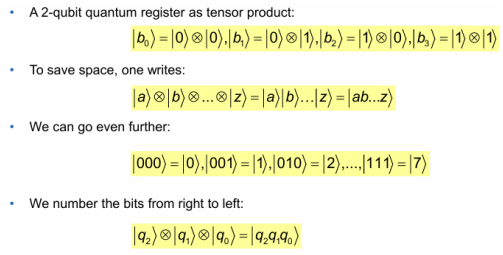



Abbildung 5: Notation (Tensorproducts)

## 1.3 Quantum Computing: Gate ( = unitary operation) – Based and Annealing

Dargestellt als circuit aber ist serie von transformations on input state (= temporal order)

1. We start with a quantum register in a product state (no entanglement).
2. We perform a series of unitary transformation. Thereby, entangled states can occur.
3. The computation ends with a measurement.

**Fact: 2 Qubits langet** Any multi-qubit unitary can be decomposed into a sequence of gates that act on just one or two qubits at a time.

### 1.3.1 Types of Gates: (general 1- qubit gate)

- **1-Qubit Gate** $U(\theta, \phi, \lambda) = \begin{pmatrix} cos(\frac{\theta}{2}) & -e^{i\lambda} sin(\frac{\theta}{2}) \\ e^{i\lambda} sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)} cos(\frac{\theta}{2}) \end{pmatrix}$
- Phase Gates:

$$P(\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}, \quad S = P\left(\tfrac{\pi}{2}\right) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}, \quad T = P\left(\tfrac{\pi}{4}\right) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

$$P^H(\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\varphi} \end{pmatrix}, \quad S^H = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\pi/2} \end{pmatrix}, \quad T^H = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{pmatrix}$$

- Rotation:

$$R_x(\theta) = e^{-iX\frac{\theta}{2}} = \begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$$

$$R_y(\theta) = e^{-iY\frac{\theta}{2}} = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$$

$$R_z(\theta) = e^{-iZ\frac{\theta}{2}} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$$

- Identity Gate: $I = I^H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- **X,Y,Z Gates (aka Pauli Gates)** Rotation around x-axis / NOT-gate$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ Rotation around y-axis $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$Rotation around z-axis$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
- **Controlled X Gate (aka Gate CX, aka CNOT):** The CNOT gate looks at the first qubit. If that qubit is $|1\rangle$, it flips the second qubit. If the first qubit is $|0\rangle$, it leaves the second qubit alone. Because qubits can be in superposition, this rule applies in parallel, creating entanglement.
- **Hadamard Gate: "Hadamarization"** produces perfect mixture of all base states: $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

### 1.3.2 All possible input states at once => $2^n$ results?

Even though you had all $f(x)$ values encoded in amplitudes, you can only extract one value per measurement. To benefit, we need algorithms that amplify the amplitude of the useful answers while suppressing the useless ones (e.g., Grover's algorithm, Shor's algorithm).

# 2 Qubits and Quantum Gates

## 2.1 Qubits

As a datastructure, qubit = two dimensional vector mit length = 1, mit components in $\mathbb{C}$

$$|q\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = c_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = c_0|0\rangle + c_1|1\rangle \tag{7}$$

$$|c_0|^2 + |c_1|^2 = 1 \tag{8}$$

## 2.2 Spin 1/2

Electron has property: position, momentum & **spin** (= abstract proprerty)
A particle with spin ½ is described by a normalized vector in a two - dimensional complex vector space.

- Spin can look üpwardsör "downwards"
- in math representation this is done in 2 base vectors that are orthogonal

## 2.3 Unitary Matrices

### 2.3.1 Begriffe

- Transponieren (transpose): Matrix an diagonale spiegeln
- konjugieren (conjugate): alle elmente von Matrix complex konjugieren (dh. vorzeichen vom imaginärteil umdrehen)
- Hermitian of a Matrix = adjoint matrix = adjunkt = Matrix Transpoinert + Konjugiert
  - A Hermitian matrix has real numbers on its diagonal and the off diagonal elements must be the complex conjugates of each other with respect to reflection on the diagonal.
- Matrix U ist unitary wenn: $U^H U = UU^H = 1$

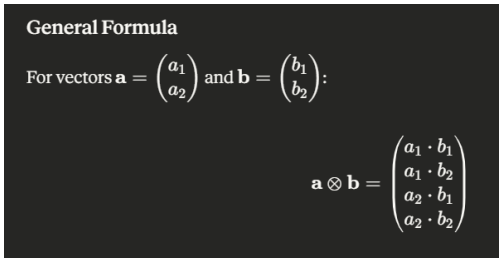



Abbildung 6: Example Tensor Product (multiply + stack)

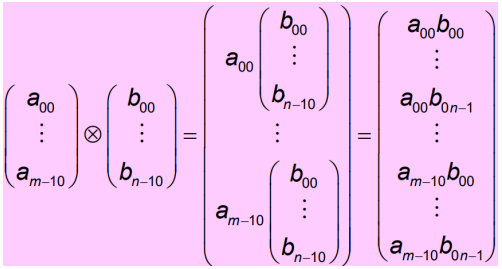



Abbildung 7: Tensorproduct of Vectors

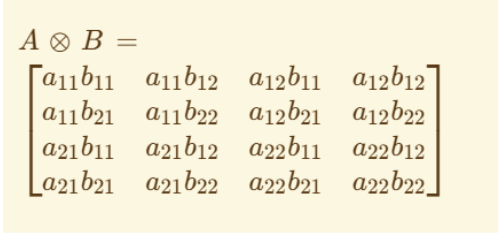



Abbildung 8: Tensorproduct of Matrixes

Tensor Product Calculator Since the columns are the images of the unit vectors, we can determine the matrix shape of the tensor product.

### 2.3.2 Hilbertspaces

The state $|\Psi\rangle$ of a quantum register composed from $2^n$ qubits is a vector in a $2^n$- dimensional complex vector space

$$|\Psi\rangle = \sum_{k=0}^{2^n-1} c_k|b_k\rangle, \quad |b_k\rangle = k - \text{the base Vector} \tag{1}$$

$$|\Phi\rangle = \sum_{k=0}^{2^n-1} d_k|b_k\rangle = \sum_{k=0}^{2^n-1} c_k|b_k\rangle \tag{2}$$

$$\langle\Phi|\Psi\rangle = \sum_{k=0}^{2^n-1} \bar{d_k}|c_k \tag{3}$$

$$\langle\Psi|\Psi\rangle = \sum_{k=0}^{2^n-1} \bar{c_k}|c_k = \sum_{k=0}^{2^n-1} |c_k|^2 = ||\Psi\rangle^2| \tag{4}$$

$$\langle\Phi|\Psi\rangle = \overline{\langle\Phi|\Psi\rangle} \tag{5}$$

## 2.4 Entanglement

Entangled states of a two-qubit register are states that CANNOT be understood as the product of two individual qubits:

- One Qubit is defined by 3 free real numbers => the state of $|\Psi\rangle$ is defind by 3 real parameters
- But the tenor product of two qubits is defined by 7 free real parameters
- A general state of a $n$-qubit register is determined by $2^{n+1} - 1$ parameters (Note: We have $2^n$ complex parameters, means $2^{n+1}$ real numbers).

The state of a quantum register cannot, in general, be understood as a combination of individual qubits, means as a tensor product of individual qubits

## 2.5 Quantum Gates and Quantum Circuits

- Quantum gate: Physical realization of a unitary operation.
- All quantum computations can be realized by a combination of Toffoli and Hadamard gates.
- Quantum circuit: A sequential series of gates.
- Input in Quantum circuits is considered a not yet entangled tensor product of individual qubit states
- **Circuit Matrix Multiplication Order**: Gates are applied right-to-left in matrix notation (rightmost gate acts first)

### 2.5.1 Proving Circuit Identities

To prove circuit identities, translate the circuit into matrices and multiply.

**Example**: Prove that $H^{\otimes 2}CX_{01}H^{\otimes 2} = CX_{10}$

**Step 1**: Calculate $H^{\otimes 2}$ (done earlier):

$$H^{\otimes 2} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

**Step 2**: Write $CX_{01}$ matrix:

$$CX_{01} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

**Step 3**: Multiply $H^{\otimes 2}CX_{01}H^{\otimes 2}$ (use software for efficiency)

**Step 4**: Verify result equals $CX_{10}$:

$$CX_{10} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

**Key insight**: The circuit identity shows that applying Hadamard gates to both qubits before and after a CNOT with control on qubit 0 is equivalent to a CNOT with control on qubit 1.

### 2.5.2 2-Qubit Gates

Can be represented by 4x4 matrices

**Types of Gates (2-Qubit)**

- Controlled Gates: An 1-qubit U gate is only applied if another qubit is in the state $|1\rangle$: E.g.: CNOT
  - In general, $CU_{ij}$ means a controlled gate, with qubit $i$ as control bit and $j$ the target bit. The target bit is subject to the 1-bit gate $U$ if the control bit is 1
  - **CNOT/CX Matrix** ($CX_{01}$ - qubit 0 controls, qubit 1 target):

$$CX_{01} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

- Toffoli Gates: Aka: CCNOT Gate: Perform AND, OR, NOT gates (emulate all classical computational processes) $\text{TOFF} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

# 3 Quiskit

- Pauli Gates: qc.x(0) / qc.y(0) / qc.z(0) (applied on qubit 0)
- Create Superpositions with Hadamerd Gate: gc.h(0)

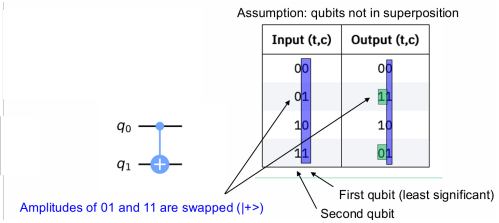



Abbildung 9: CNOT Gate Table

This is because:

- The Hadamard gate created a superposition of $|0\rangle$ and $|1\rangle$ on q0.
- The CNOT gate **entangles** the two qubits: it correlates q1 with q0.
- So when q0 is $|0\rangle$, q1 stays $|0\rangle \rightarrow |00\rangle$
- When q0 is $|1\rangle$, q1 flips to $|1\rangle \rightarrow |11\rangle$

This results in a **maximally entangled state** known as a **Bell state**:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Abbildung 10: Bell State Result

## 3.1 Multi-Qubit Gates Create Entanglement

### 3.1.1 Without Superposition

CNOT-gate (Performs X-gate (not on second qubit (target), if state of first qubit (control) is 1: qc.cx(0,1)

### 3.1.2 With Superposition

Apply Hadamard gate to q = Superposition => Apply CNOT = Entangled state ("Bell state")

- 50%probability of being measured in state |00> or |11>
- 0% chance of being measured in states |01> or |10>

# 4 Quiskit 2

## 4.1 Grover's Algorithm (Unstructured Search)

Search (does array contain) => worst case O(n)
Grovers Algorithm $O(\sqrt{N})$

1. Apply Hadamar Gate to all Possibilities:

$$H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in 0,1^n} |x\rangle = |\phi\rangle$$

   - Makes amplitude of all item in array to possibility

2. Repeat $\sim \sqrt{N}$ times

   (a) **Search & Reflect** Flip Amplitude of searched item (I = Einheitsmatrix)

$$U_s = I|s\rangle\langle s|$$

   (b) **Diffusion** Operator (makes probability of flipped item higher so it will get returned when measured)

$$\text{Apply } U_s = 2|\phi\rangle\langle\phi| - I$$

3. Measure

$$\begin{aligned} f(v1,v2,v3) = {} & (\neg v1 \vee \neg v2 \vee \neg v3) \wedge \\ & (v1 \vee \neg v2 \vee v3) \wedge \\ & (v1 \vee v2 \vee \neg v3) \wedge \\ & (v1 \vee \neg v2 \vee \neg v3) \wedge \\ & (\neg v1 \vee v2 \vee v3) \end{aligned}$$

Abbildung 12: Satisfiability Problem

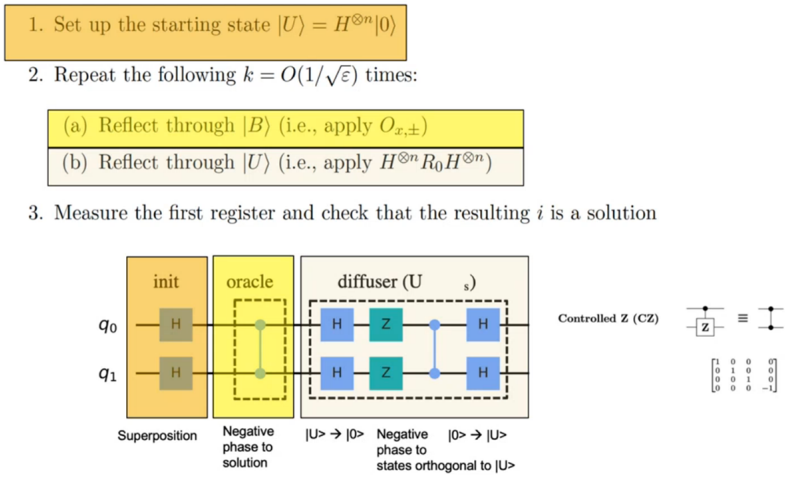



Abbildung 11: Grover's Algorithm

Cannot really be used for search algorithm (as the item needs to be marked to be found with it)
But can be used to solve a problem like this

# 5 Quantum Error Correction, Quantum Cryptography

## 5.1 No-Cloning Theorem

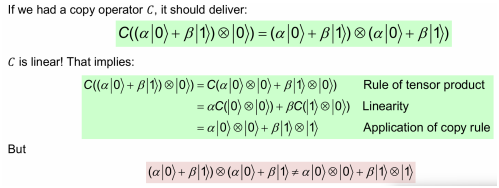



Abbildung 13: You can't copy a qubit

- Quantum Backup not possible
- Data cannot be stolen
- Bits can't be duplicated for error correcting codes
  - redundancy bits cannot be produced
  - qubits can't be measured without destroying them
  - infinite number of possible errors (bit flip, phase flip, phase change)
    - all possible fault are a linear combination of bit flips (X), phase flips (Z) and a combination of phase and bit flips (XZ = -ZX).

## 5.2 Error Correction

**Syndrome determination** = measuring auxiliary qubits to extract information about errors in a quantum code without disturbing the logical qubit. It tells you what error occurred but not what the quantum information is.

### 5.2.1 Partial Bitflip Correction

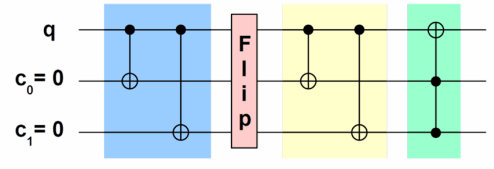



Abbildung 14: Bitflip Correction

Creates entanglement with control bits which is used after to correct flipped bits while de-entangle-ing
Correction Bits are not restored (okay because not used any further

### 5.2.2 Protection against Phase Flips

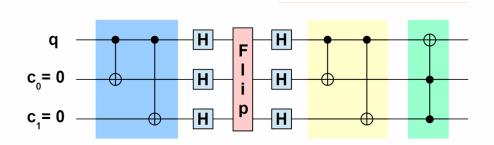



Abbildung 15: Phase Flip Protection

Works only if error indeed a phase flip and not a bit flip because hadamard gate transforms the problem into a bitflip

### 5.2.3 Shor's 9 Qubit Error Correction

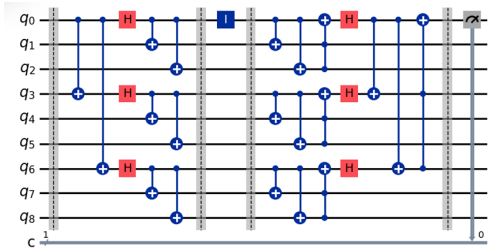



Abbildung 16: Shor's Error Correction (uses 8 qubits to protect 1 qubit)

Can be done more efficiently but harder to understand

## 5.3 Quantum Cryptography

Quantum cryptography deals with ways to exchange keys in such a way that you can detect with high probably whether somebody eavesdropped this exchange.

There are two basic facts we use:

- If one measures a state, the state is altered in a non-reversible, stochastic manner.
- You can't copy a quantum state (no – cloning theorem)

**Polarization (in classical physics):**

- Filter Parallel: everything gets through
- Filter Horizontal: nothing gets through
- Filter Diagonal: some pass through

**Polarization (in QM)**

- Photon passes with probability: $\cos^2(\Theta)$
- (Filter acts as measurement)

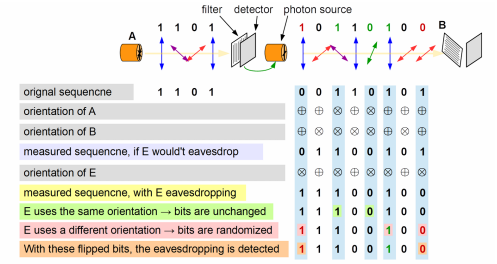



Abbildung 17: Secured Exchange of Key

Base orientation of both sender and reciever are chosen randomly
Compare Bases publicly (sender tells which they sent in the same state as reciever) Compare some of the bits that should be correct publicly => if intercepted needed to be measured and the interceptor would need to know in which direction to sent out the bit

# 6 Quantum Mechanics

In quantum mechanics (QM), a particle is represented by a wave function:

$$\Psi(x, y, z, t) \in \mathbb{C}$$

The wave function is a complex-valued function of space and time.

## 6.1 What does the wave function describe?

- The wave function of a particle cannot be observed directly.

$$P(C) = |\Psi(x,y,z,t)|^2 dV$$
$$P(x_1,x_2,y_1,y_2,z_1,z_2,t) = \int_{x_1}^{x_2}\int_{y_1}^{y_2}\int_{z_1}^{z_2} |\Psi(x,y,z,t)|^2 dxdydz$$

Abbildung 18: probability that the particle is located in the cube given by [x, x + dx; y, y + dy; z, z + dz]

## 6.2 Spin

- wave Function is usually not in Spin-0 as above but in vector form ($\Psi$ = complex-valued vector with n components)
- Number of components = spin(s) of the Particle

$$s = \frac{n-1}{2}$$
$$s_i = 2q_i - 1$$

## 6.3 What can we do with the wave function

- The mathematical object is the solution of a certain equation, the so-called Schrödinger equation (SE). This mathematical object is well understood.
- The wave function is a physical entity. The physical reality of the wave function is not fully understood, especially with regard to measurements (see later).
- Physical Quantities are determined with operators

**wave function**

$$\lambda_{dB}\frac{2\pi h}{mv}$$

- $h = 6.626 \times 10^{-34}$ J·s (Planck's constant)
- $m = 9.109 \times 10^{-31}$ kg (electron mass)

### 6.3.1 Operators

Natural / Well Guessed leads to these:

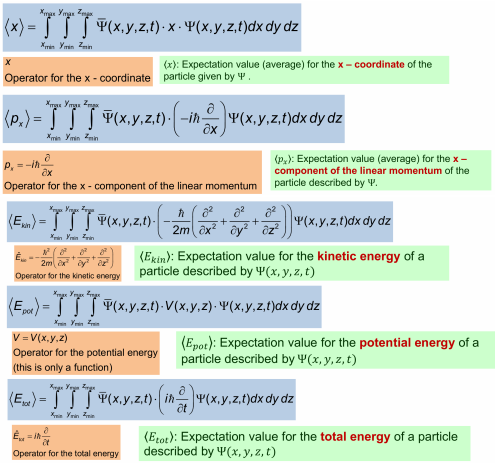



Abbildung 19: Operators

### 6.3.2 Normalization Condition (Special Operator)

$$\int_{x_{min}}^{x_{max}}\int_{y_{min}}^{y_{max}}\int_{z_{min}}^{z_{max}} \bar{\Psi}(x,y,z,t) * \Psi(x,y,z,t)dxdydz = 1$$

## 6.4 Schrödingers Equation

- Goal: Determine temporal rate of change of data structure (Diff function)
- Idea: Apply principle of energy conservation (kin + pot = total)

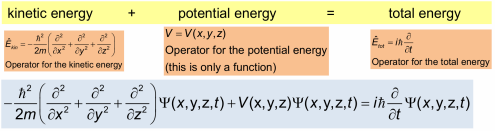



Abbildung 20: Ausgangslage Schrödinger

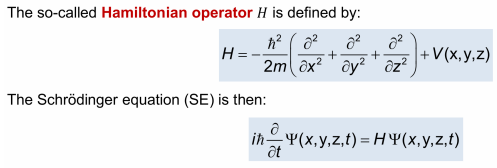



Abbildung 21: Hamiltonian Operator and Schrödinger's Equation

**Facts about SE solution**

- The Schrödinger equation (SE) is easily solvable for an eigenfunction (if $H$ does not depend on time)
- The SE is linear The sum of the solutions is again a solution.
- The eigenfunctions form an orthonormal basis => an initial wave function $\Psi(x, t = 0)$ written as a sum of eigenfunctions
- If the wave function of an object (e.g. a particle) is given by an eigenfunction$\varphi_r$ of $H$, then the eigenvalue $E_r$ gives the energy of the object.

**Key Principle**

In quantum mechanics, when a state is written as a superposition of energy eigenstates:

$$\Psi(x,t) = \sum_r c_r \varphi_r(x,t)$$

The probability of measuring energy $E_r$ is given by:

$$P(E_r) = |c_r|^2$$

provided the wavefunction is normalized ($\sum |c_r|^2 = 1$).

**Finding** $P(E_2)$

From the given wavefunction:

$$\Psi(x,t) = (0.45 - 0.41i)\varphi_0 + (0.32 + 0.52i)\varphi_1 + (0.11 + 0.37i)\varphi_2 + 0.33\varphi_3$$

The coefficient for $\varphi_2$ is:

$$c_2 = 0.11 + 0.37i$$

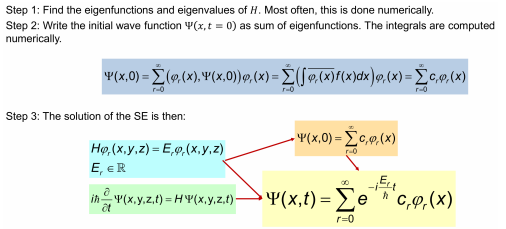



Abbildung 22: Solution of Schrödinger Equation

**Calculating** $|c_2|^2$

$$|c_2|^2 = (0.11)^2 + (0.37)^2$$

$$|c_2|^2 = 0.0121 + 0.1369$$

$$|c_2|^2 = 0.1490$$

**Solution of SE**

## 6.5 Lessons learnt

- Free Particle: (Particle without potential) can be described with wave function
- Particle behaves like wave => double slit experiments, can be used with optics (photography)
- compute wave length of objects based on its mass
- Localized signal has a spectrum => the more localized the broader its spectrum
- The more you localize a particle in real space, the broader its Fourier – spectrum. In quantum mechanics, the frequencies of the Fourier – transform are proportional to linear momentum, therefore the more you localize a particle, the less defined is its momentum!
- Particle in a box:

  - Enclosingaparticle in a box imposes some constraints on its wave function:
    * Inside thebox, we have a free particle => The wave function is given by a true wave.
    * Outside the box, the wave function must be zero, because there, the potential energy of the particle would be infinite (the potential is infinite!).
    * The wavefunction must be continuous.
  - The wavesmust fitïnto a box. =>this restricts the allowed wave lengths.
  - wavelengthsarerelatedto momentum and momentum is related to energy.
  - only discrete energies are possible.

# 7 Quantum Annealing and Variational Quantum Approaches

## 7.1 Begriffe

- NP-Complete: problems that can be solved in polynomial time by a nondeterministic machine and can be verified in polynomial time by a deterministic Machine
  - if you can solve one NP-complete problem, you can solve all NP-hard problems
- NP-Hard: hard to solve, easy to verify (if it is solution)
- SSP (Subset-Sum-Problem): given array of numbers and a sum, does there exist a subset of numbers in the array that sum up to the given sum (NP-Complete Problem) => can be formulated as an optimization problem

## 7.2 Energy

### 7.2.1 Energy Grundlagen:

- Systems and their environments contain energy
- The energy is available in packets, so-called energy quanta.
- A system can release these energy quanta into its environment.
- Likewise, a system can absorb energy quanta from the environment.
- To put it loosely: The ënergy densityïn the system corresponds to the temperature.

### 7.2.2 System Dynamics: Energy

- System state can be described by parameter x, energy content of system = f(x)
- minimal energy bei T = 0 = ground state, all energy of system will be released into environment
- T > 0: energy will fluctuate around a minimum until temperature is too high and system disintegrates
- System can have multiple energy minima => fluctuations can lead to transition from one minimum to another (local minimum = metastable, global minimum = stable)
- often fluctuations can be neglected and system assumed at state of minimal energy

### 7.2.3 Finding Minima

- can be easy => just go down
- can be hard => to leave local minimum needs to at least get energy $\Delta E$

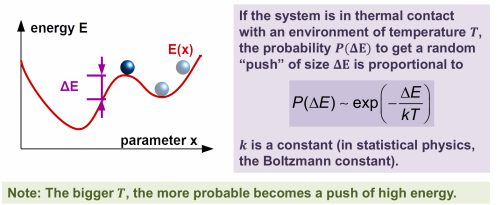



Abbildung 23: Leaving a local Minimum

- Fitness Landscapes
  - If the global minimum is not significantly lower than the local minima and in addition, if the energy barriers between minima are high, simulated annealing either gets stuck or jumps around more or less randomly.
  - If the landscape is very shallow, except for one deep hole, a so called golf court landscape, finding the global minimum is also difficult.

**Tricks to find Minima**

- Start at high temperature => find the valley
- Lower the temperature => simulate annealing

**Classical vs Quantum Annealing**

- In classical annealing, you can't jump over a hill if you don't have enough energy
- In quantum mechanics, you just have to wait long enough and eventually. the system will "tunnel" through the energy barrier.
- in Quantum Annealing we construct the ground state with $H_0$

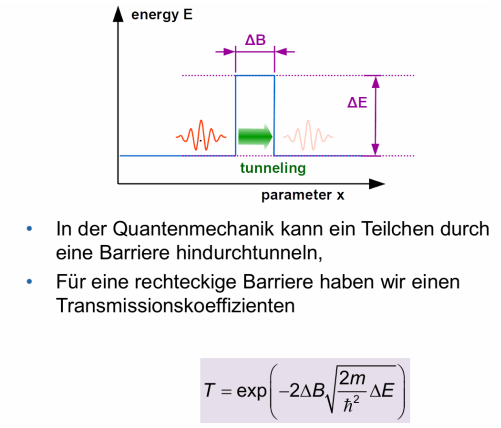



Abbildung 24: Tunnel Effect, Transmission Coefficient

## 7.3 Ising - Models and QUBO

Energy function of interacting Spins to find ground state: Spin Variables $S_i$, constants $h_i$ and $K_{ij}$

$$E = -\sum_i h_i S_i - \sum_{i,j} K_{ij} S_i S_j$$

- after measurement $S_i$ can be 1 or -1
- Simplifications made in Model:
  - system of spins which can attain only two values: up and down.
  - only neighbouring spins interact.
- Model can be used to:
  - study the emergence of so called «magnetic domains» by energy minimization
  - It does not give proper quantitative answers, but reproduces surprisingly well the qualitative behaviour of more complex systems.

### 7.3.1 QUBO Model

Change to measured qubit variables ($q \in \{0, 1\}$:

$$q_i = \frac{1 + S_i}{2}, 2q_i - 1 = S_i \tag{1}$$

$$E_{\text{QUBO}} = c + \sum_i a_i q_i + \sum_{i,j} b_{ij} q_i q_j \tag{2}$$

**Berechung:** $s = 2q - 1$ => einsetzen und ausrechnen.

**Facts about QUBO**

- Finding minimum of QUBO = NP-hard
- One can couple spins. With according hardware, parameters $h_i$ and $K_{ij}$ can be chosen

- Energy of spin system = $E_{\text{Ising}}$ = minimal = ground state
- All NP-Hard problems can be formulated as Ising Energy Minima Problems

## 7.4 Quantum Annealing

- Start with a standard Hamiltonian $H_0$, which has known and easy to produce ground state.
- The energy function we want to encode leads to a Hamiltonian $H_{QC}$
- The Hamiltonian of the quantum computer is then implemented by a process in which we start with a well-known system in which we can prepare the ground state. Over a time $T$, we switch off $H_0$ and switch on $H_{QC}$. This can e.g. be realized by switching on an external field.
- We present Adiabatic Quantum Optimization (AQO). AQO does not take into account
- the effects of temperature, means fluctuations. Somewhat simplified, AQO with temperature is called quantum annealing.

$$H = (1 - \frac{t}{T})H_0 + \frac{t}{T}H_{QC}$$

**The Adiabatic Theorem** The adiabatic theorem of quantum mechanics states that a quantum mechanical system remains in a good approximation in an eigenstate (especially the ground state!) if the associated Hamiltonian operator explicitly depends on time, but changes only slowly. The temporal change is based on parameters specified outside the system, e.g. magnetic or electric fields or geometric quantities.
Slow: means depends on energy gap between groundstate and first excited state $\Delta E$

### 7.4.1 Can we beat NP

Probably not. QA is still exponential but it may be that $\alpha$ and $\beta$ are smaller in QA than in classical computation

$$T = O(\exp(\alpha N^{\beta})$$

## 7.5 Evolutionary Optimization

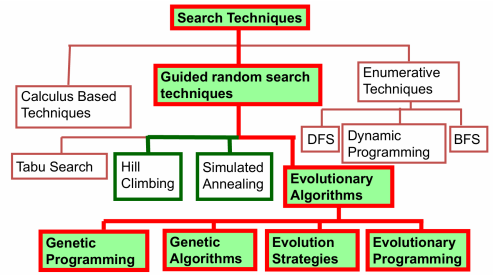



Abbildung 25: Classes of Search Techniques

Artificial evolution combines both approaches: The structural unboundedness of evolution with the potential of human foresight.

**Four Types of Evolutionary Algorithms**

1. Evolution strategy (ES): Genotype is a real valued vector, mutation rates are permanently adapted.
2. Genetic algorithms (GA): Genotype is a string of numbers. The most popular form of EA. Mutation rates are fixed.
3. Evolutionary programming (EP): Genotype represents a parameterized program. These parameters are evolved.
4. Genetic Programming (GP): Genotype represents a program in form of a tree. This tree is evolved. (Modern developments blur this distinction)

# 8 Quantum Machine Learning

## 8.1 Grundlagen

Support Vector Machine: Separate Groups => if not linearly separable add a dimension (transform into higher dimension = called ploynominal expansion (= expensive calculation))

kernels (kernel functions) are the solution to this dilemma. They allow to efficiently work in higher-dimensional spaces without going through the transformations explicitly.
Kernel = use dot-product of input features directly

## 8.2 Different Kernel Functions

- Polynominal: $K_{poly}(x_1, x_2) = (< x_1, x_2 > +1)^d$
- Radial basis Function: $K_{RBF}(x_1, x_2) = e^{-\frac{1}{2}||x_1 - x_2||^2}$

## 8.3 Quantum Kernel

Replace classical kernel with quantum kernel (can only do better if hard to estimate classically, also no guarantee of advantage

## 8.4 Quantum Classification

**Most general Ansatz: quantum variational circuit**

$$W(\theta)U_{\Phi(\vec{x})}|0\rangle$$

- Convert classical data to quantum data using quantum feature maps:
  - Using a quantum circuit $U_{\Phi(\vec{x})}|0\rangle$
  - Where psi could be any classical function applied to data x
- Process the data using parameterized quantum circuit $W(\theta)$
- Measure the data and return classical values -1 or 1

### 8.4.1 Quantum Feature Map $V(\Phi(\vec{x}))$

- Maps classical data to quantum data
- Is a parameterized quantum circuit
- Can leverage quantum advantage

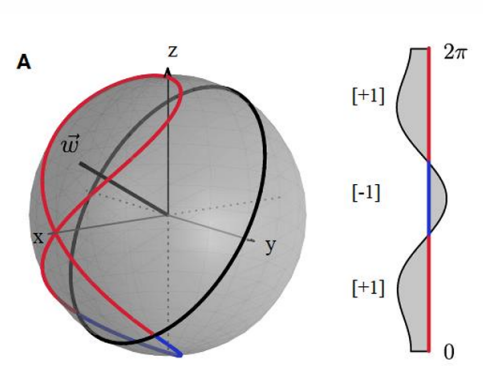



Abbildung 26: Quantum Feature Map Example

### 8.4.2 Quantum Kernel

Every computation in QC is already a tensor product (as everything is a vector) => therefore this is a very natural way for ML
With feature map $\Phi(\vec{x})$: and can be represented as matrix

$$K_{ij} = |\langle \Phi^t(\vec{x}_j)|\Phi(\vec{x}_i)\rangle|^2$$

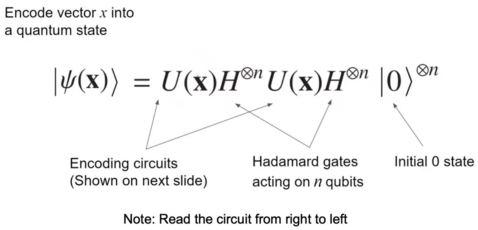



Abbildung 27: Quantum Support Vector Machine

## 8.5 Different Types of Feature Maps

### 8.5.1 ZFeatureMap

Each qubit represents one feature, no interaction between features, no entanglement, 1 Unitary operation

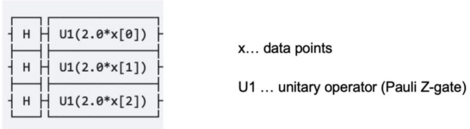



Abbildung 28: ZFeature Map

### 8.5.2 ZZFeatureMap

Second order Pauli-Z-evaluation circuit, each qubit represents 1 feature, allows interactions between features, linear entaglement,
$\Phi$ is classical non-linear function with default: $\Phi(x) = x$ and $\Phi(x, y) = (\pi - x)(\pi - y)$

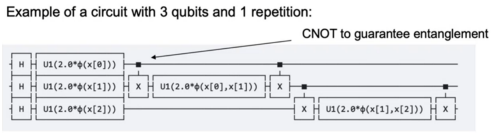



Abbildung 29: ZZFeature Map

### 8.5.3 PauliFeatureMap

Pauli expansion circuit, can be created with different gates, each qubit represents one feature, allows interactions between features, enables enganglement

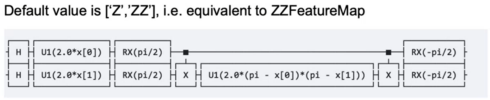



Abbildung 30: Pauli Feature Map

## 8.6 For Implementation in qiskit

p.25-28 in Lect. 8 slides

# 9 Quantum Machine Learning Part2

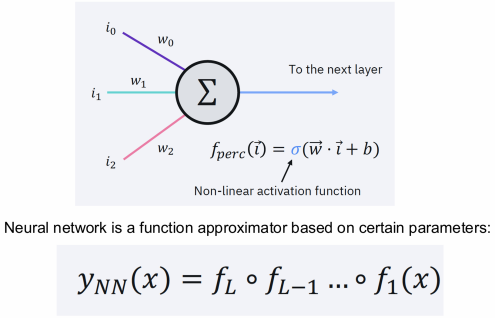



Abbildung 31: Perceptron (the simplest neural network)

Neural Networks = basically a vector product (which can be very nicely done with quantum circuits)

## 9.1 Approach 1

- Use a classical neural network
- Replace some parts with a quantum circuit:
  - Similar to Quantum SVM
  - Quantum hidden layers are quantum circuits

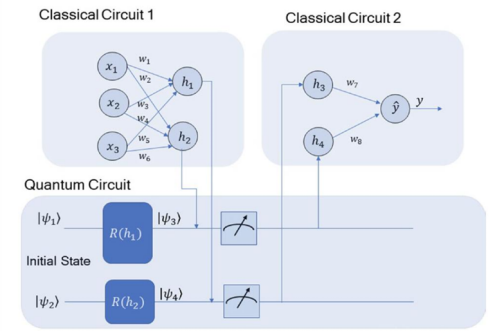



Abbildung 32: Hybrid Quantum-Classical Neural Network

**Steps:**

- Classical part:
  - Input: $[X_1, X_2, X_3]^T$
  - Convert input to the hidden layer activations $[h_1, h_2]^T$ by the classical circuit 1

$$h_1 = \sigma(x_1 w_1 + x_2 w_2 + x_3 w_5) \quad (1)$$
$$h_2 = \sigma(x_1 w_1 + x_2 w_4 + x_3 w_6) \quad (2)$$

  [mit $\sigma()$ = sigmoid activation function]

- Quantum Part:
  - Input: hidden activations $h_1$ and $h_2$
  - Act as rotational angle parameters for the gates $R_1$ and $R_2$ n quantum circuit
  - They change the initial states $|\Psi_1\rangle$ and $|\Psi_2\rangle$:

$$|\Psi_3\rangle = R(h_1)|\Psi_1\rangle \quad (1)$$
$$|\Psi_4\rangle = R(h_2)|\Psi_2\rangle \quad (2)$$

  - Output: $\hat{y} = \sigma(h_3 w_7 + h_4 w_8)$

## 9.2 Approach 2

- The whole network is implemented as a parametrized quantum circuit
- QNN with i layers: $U(\theta) = U_l(\theta_l)U_{l-1}(\theta_{l-1}...U_1(\theta_1)$
  - U ... unitary transformation
  - $\theta = [\theta_L, \theta_{L-1}, ..., \theta_1]^T$ set of parameters for the QNN

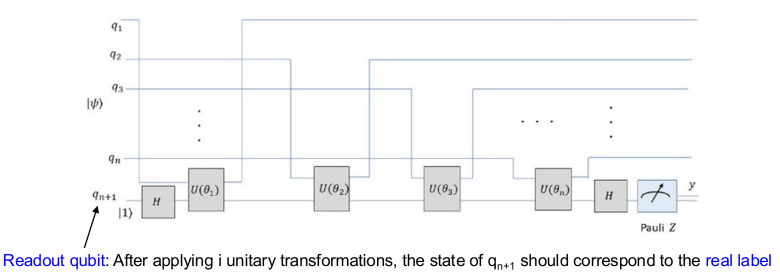



Abbildung 33: Approach 2

## 9.3 What is a good quantum circuit

- It needs to generalize well
- Two measures:
  - Expressibility: Ability to generate states in Hilbert space with good coverage (aka how many states on the sphere)
  - Entanglement: Ability to entangle states

## 9.4 Encoding Data (aka how to represent data as quantum states)

### 9.4.1 Basis encoding

Classical N-bit string is represented by specific computational basis state (binary encoding: x=2: 2 bit: 10 or quantum state: |10>)

- Entire dataset consisting of M data points is represented as a superposition over all basis states (set of states = $\mathcal{X}$:

$$|\mathcal{X}\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} |x^m\rangle$$

- $|x^{(m)}\rangle$ is a quantum state

### 9.4.2 Amplitude Encoding

- Data is encoded into amplitude of a quantum state

- All data points are encoded into one amplitude vector

$$\sum_{i=1}^{N} \left|x_i^{(1)}\right|^2 = 4^2 + 8^2 + 5^2 = 105 = |\alpha|^2 \rightarrow \alpha = \sqrt{105}$$

Abbildung 34: Encoding of example dataset: $\mathcal{X}_{ex}, \vec{x}^{(1)} = (4, 5, 8)$

### 9.4.3 Angle Encoding

Data is encoded in rotation angles, each data point is encoded separately
E.g. data point $x = (x_1, ..., x_N)$ is encoded as:

$$|x\rangle = \otimes_{i=1}^{N} \cos(x_i)|0\rangle + \sin(x_i)|1\rangle$$

### 9.4.4 Arbitrary Encoding

Example: Encode data point x = [0.1, 0.2, 0.3, 0.4, …. 1.1, 1.2]

Encode with 12 features

```
x = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2]
encode = circuit.bind_parameters(x)
encode.decompose().draw()
```

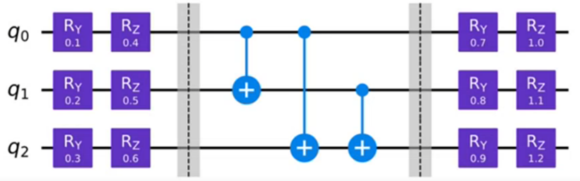



Abbildung 35: Arbitrary Encoding Example

### 9.4.5 Comparison of Encodings

Generally: Depends on data and goal

| | **Basis Encoding** | **Amplitude Encoding** |
|---|---|---|
| #qubits | $N$ | $\log_2(N)$ |
| Complexity | Low | High |

Tabelle 1: N = number of features (basic vs. amplitude encoding)

## 9.5 Quantum Neural Networks

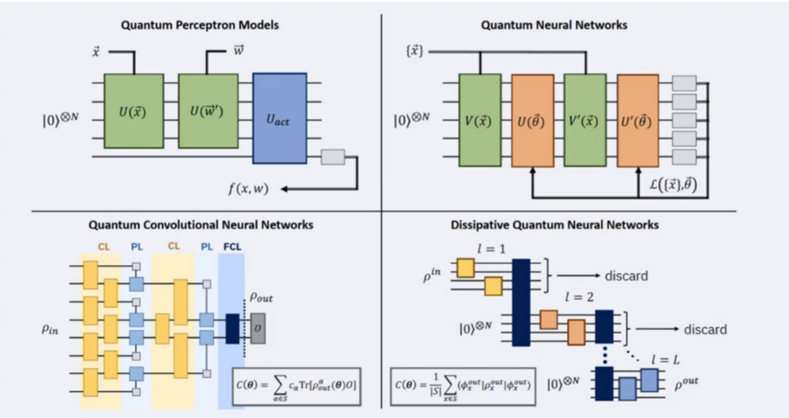



Abbildung 36: Übersicht Neural Networks

### 9.5.1 Quantum Perceptrons

- NN performs non-linear transformations
- but quantum circuits perform linear trnasformations (unitary operators)

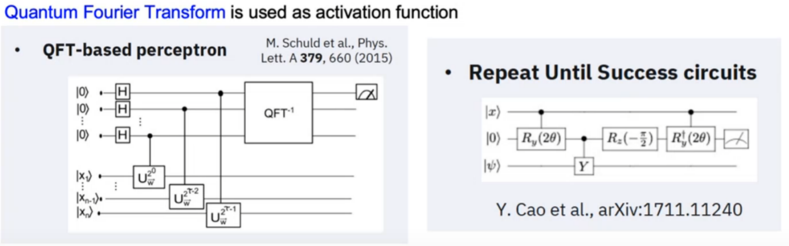



Abbildung 37: Quantum Fourier Transform

$$U_{perceptron} = \sum_{\alpha} |\alpha\rangle\langle\alpha| \otimes U(\alpha)$$

[$\alpha$ = input qubits, $U$ = Unitary Operator]

### 9.5.2 Quantum Neural Networks

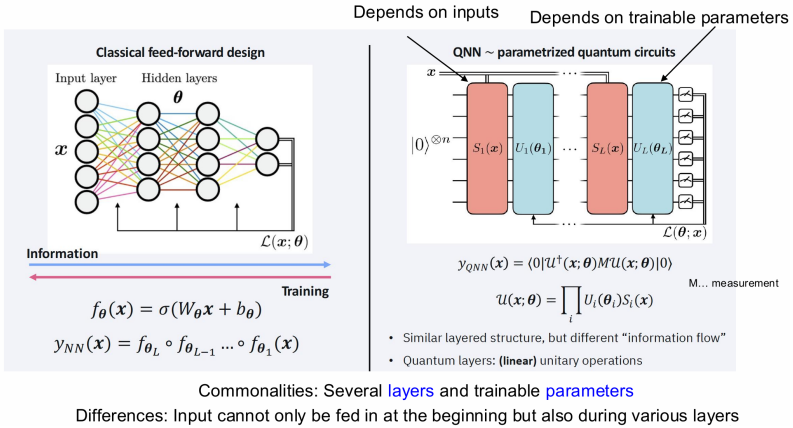



Abbildung 38: Quantum Neural Networks

- Because copying not possible => reupload input data several times
- Encoding has to be repeated many times to increase the frequency of the "signal"
- Spectrum size K grows linearly with number of repetitions L

**Begriff: Convolution** Used in classical NN: don't use fully connected layers, use convolutions (each convolution analyses subpart of whole image) = reduced feature dimensions

- makes images more manageable for successive layers
- makes network resilient to small changes
- avoids overfitting

### 9.5.3 Direct Quantum Implementation of CNN

- Advantages: Speedup (better linear algebra manipulations), leverage quantum superposition)

- Disadvantages: data encoding bottelneck (QRAM required), non-trivial subroutines required (=> error prone)

### 9.5.4 Quanvolutional Implementation of CNN

Extend classicla CNNs with quantum kernels as new transformation/filter in convolution layers, uses quantum feature map (similar to Quantum SVM), can be directly integrated in existing classical architecture, requires shallow circuits, does not require QRAM

- Fewer parameters required than full QNN (=> easier to train)
- no vanishing gradient effect (=> local optimization instead of global)

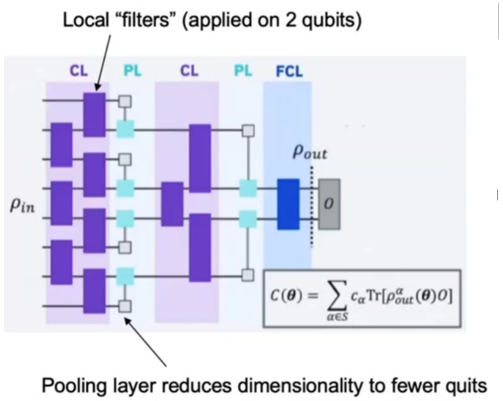



Abbildung 39: Quantum Convolutional Neural Network

### 9.5.5 Dissipative Quantum Neural Networks

collection of quantum perceptrons, learns an unknown quantum transformation, feed forward architecture, backprogagation-like training can be performed, dissipative (discarding of layers of qubits ater layer x has interacted with layyer x+i

| Aspect | Quantum Machine Learning (QML) | Quantum Optimization (QO) |
|---|---|---|
| Analogy | Learning to be an artist | Being a coloring police officer |
| Goal | Learn patterns from many examples | Find the best solution for one problem |
| Cost Function | "How wrong are my predictions?" | "How many rules are broken?" |
| Output | A smart model that can handle new problems | The actual solution to this specific problem |
| When to Use | You have lots of similar problems | You have one important problem to solve |

Abbildung 41: ML or Optimization

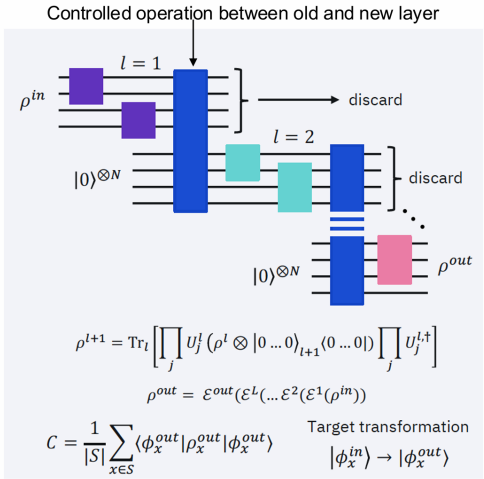



Abbildung 40: Dissipative Quantum Neural Networks

# 10 Quantum Machine Learning vs Quantum Optimization

## 10.1 Hamiltonian = Quantum Cost Function

A Hamiltonian is a rule that tells a quantum system:

- which states are good (low score / low energy)
- which states are bad (high score / high energy)

A cost Hamiltonian $H_C$ is a Hermitian operator diagonal in the computational basis where each basis state $|z\rangle$ (a bitstring gets an energy equal to the classical cost C(z): $H_C = \sum_z C(z)|z\rangle\langle z|$: So measuring $\langle z|H_C|z\rangle$ returns the classical cost for the bitstring $z$. In implementations we usually express $H_C$ as sums of Pauli operators (mostly Z and Identities), because those are native to qubit hardware.

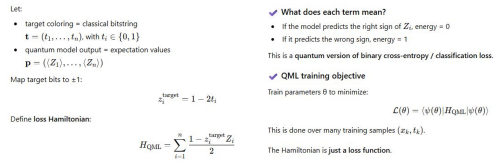



Abbildung 42: QML: Prediction Error Hamiltonian

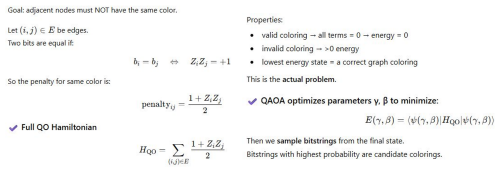



Abbildung 43: QO: Constraint Violation Hamiltonian

Gradients are the game changers for complicated quantum models, especially if you want to then connect it to a classical network There are many ways to introduce nonlinearity in quantum neural network - e.g. the simplest is a combination of a specific measurement and a subsequent classical conditional operation (like if measurement_result > threshold: apply_X_gate()) can play a functionally similar role as ReLU

# 11 Advanced Quantum Machine Learning

## 11.1 Dimensionality Reduction in QML

Besides classical-to-quantum feature maps, dimensionality reduction plays a major role in preparing data for quantum models. Three main pipelines exist:

1. **Classical non-ML dimensionality reduction + QML**
   Use classical preprocessing techniques (e.g., resizing, PCA-like transforms) before feeding data into a quantum or hybrid model.

2. **Learned classical dimensionality reduction + QML**
   Classical autoencoders compress high-dimensional data; the compressed representation is then processed by a classical or quantum model.

3. **Learned quantum dimensionality reduction**
   Quantum autoencoders reduce dimensionality directly on quantum states, enabling efficient quantum-native representations.

## 11.2 Advanced Quantum Feature Map Techniques

Beyond standard Z-, ZZ-, and Pauli-feature maps:

- Re-uploading data multiple times into the *same qubit wire* enables encoding of high-dimensional inputs with few qubits.

- Convolution-inspired quantum encoders (“quanvolution”) allow localized processing of data patches.

- Quantum feature maps can be designed to encode extremely large feature vectors by iterative or layered embedding.

## 11.3 Quantum Generative Adversarial Networks (QGANs)

GAN concepts can be extended to the quantum domain:

- **Classical GAN recap:** a generator and discriminator compete against each other to model complex distributions.

- **Quantum GANs:** replace generator, discriminator, or both with parameterized quantum circuits.

- **Patch-based QGANs:** process local regions (patches) of the data with quantum circuits to simplify training.

### 11.4 Optimization Strategies: Gradient vs. Non-Gradient

Hybrid quantum-classical models allow both gradient-based and gradient-free optimization:

- **Hybrid gradients:** use parameter-shift rules to compute gradients from quantum circuits.
- **Gradient-free methods:** useful when gradients are noisy or expensive (e.g., evolutionary strategies, SPSA).
- QML models often require long optimization times due to noise and non-convex loss landscapes.

### 11.5 Quantum Reinforcement Learning (QRL)

QRL integrates quantum circuits into reinforcement learning workflows:

- Parameterized quantum circuits can serve as Q-function approximators.
- QML visualization tools (e.g., QML playgrounds) help explore circuit behavior.

Quantum advantages stem from:

- **Superposition:** explore multiple actions simultaneously.
- **Entanglement:** encode correlations between states and actions.
- **Interference:** amplify good policies and suppress poor ones.

### 11.6 Mixed States and Density Matrices

While previous sections focus on pure states, QML also uses mixed-state formulations:

- A mixed state is represented by a density matrix $\rho$ instead of a state vector $|\psi\rangle$.
- Mixed states appear naturally due to noise, partial tracing, and ancilla usage.
- Qiskit and other frameworks allow explicit density-matrix simulation for QML experiments.

### 11.7 Nonlinearity via Ancilla Qubits

Because quantum operations are linear, nonlinearity must be introduced indirectly:

- Using additional **ancilla qubits** and tracing them out produces effective nonlinear transformations.
- Partial measurements on subsystems lead to conditional dynamics resembling activation functions.
- Ancillas can also provide workspace for reversible subroutines and noise-injection mechanisms.

### 11.8 Other Uses of Ancilla Qubits

Beyond nonlinearity:

- Ancillas allow modular circuit construction.
- They enable error detection and syndrome extraction.
- They help implement more expressive quantum channels and controlled operations.

### 11.9 Modelling Complex Distributions with QML

Quantum circuits can represent probability distributions that are hard to model classically. This includes:

- Nontrivial or multimodal distributions.
- Toy examples demonstrating expressiveness of small quantum models.

## 12 Quantum Weirdness & Physical Aspects of Quantum Information

### 12.1 Non-Locality (Phyisics, Math, Space)

2 entangled bits influence eachother even when space-wise separated = non-locality: that causes measurement at one location to influence measurement at

another location = states are correlated entanglement is independent from physical location even if the non-local states are measured locally measurements can disentangle non-local states into just correlated states

Ist ausgabe wirklich zufall oder durch "verborgene variabel (hidden variables) bereits vorgegeben die dann normal local sind (= assumption of realsim / locality assumption)? => hidden variables disproved by bell theorem

**(ARe) Assumption of Realism:** Assumption of realism: The value of $v(A)$ always exists and represents a real property of $|\Psi\rangle$ independent of whether we measure it or not. $v(A)$ takes one of the eigenvalues of the operator $A$. After a measurement $M_{A,Q}$, the system is in the eigenstate that belongs to the measures eigenvalue.

**(ALo) Assumption of locality:** The value of a variable, or the state of a system, is only influenced by its immediate environment

**(AQc) Assumption of quantum compatibility:** If we assume realism ARe and run an experiment several times, the results are $v(A)$ which are distributed according to the rules of quantum mechanics.

**(AFc) Assumption of function composition:** We assume compatible self-adjoint operators $A$ and $B$ ($[A, B] = AB - BA = 0$). => both operators have the same eigenvectors, both of which both can be measured (no problems with the uncertainty principle).

**(ANc) Assumption of non-contextuality:** Given are two operators $A$ and $B$ with $AB = BA$. => This means: the two operators $A, B$ have the same eigenstates $|\Phi_i\rangle$ (but not necessarily the same eigenvalues). The of a system $|\Psi\rangle = \sum a_i|\Phi_i\rangle$ using a measuring device $M_{A,Q}$ leads to a (by hidden variables) determined $|\Phi_i\rangle$ regardless of whether a measurement $M_{B,Q}$ has been or is being carried out.

### 12.1.1 CHSH Ungleichung (=> später vergallemeinert in Bells Ungleichungen)

**Experimenteller Aufbau:** Alice und Bob führen jeweils zwei Experimente durch (klassisch oder quantenmechanisch), wobei Alices Experimente nicht notwendigerweise dieselben wie Bobs sind.

- Alices Experimente: $A_0$ und $A_1$ mit Resultaten $a_0, a_1 \in \{-1, +1\}$
- Bobs Experimente: $B_0$ und $B_1$ mit Resultaten $b_0, b_1 \in \{-1, +1\}$
- Die Experimente werden nach einer zufälligen Wahl an mehreren Testsystemen durchgeführt

**Definition der CHSH-Größe:**

$$S = a_0b_0 + a_0b_1 + a_1b_0 - a_1b_1 \tag{3}$$

**Realismus-Annahme:** Wir nehmen Realismus an (à la Einstein). Das bedeutet, dass alle Teile von $S$ existieren, unabhängig davon, ob sie gemessen werden (klassisch natürlich).

**Umformung von $S$:**

$$S = a_0b_0 + a_0b_1 + a_1b_0 - a_1b_1 = (a_0 + a_1)b_0 + (a_0 - a_1)b_1 \tag{4}$$

Da $a_i = \pm 1$ gilt:

- Entweder ($a_0 - a_1 = 0$ & $a_0 + a_1 = \pm 2$) oder ($a_0 - a_1 = \pm 2$ & $a_0 + a_1 = 0$)
- Wegen $b_i = \pm 1$ folgt: $S = \pm 2$

**Erwartungswert:** Alice und Bob bestimmen nie $S$ selbst, sondern messen nur $a_0$ oder $a_1$ bzw. $b_0$ oder $b_1$.

Unter der Realismus-Annahme ist der Erwartungswert von $S$ (über viele Messungen):

$$\langle S\rangle = \langle a_0b_0\rangle + \langle a_0b_1\rangle + \langle a_1b_0\rangle - \langle a_1b_1\rangle \tag{5}$$

Der Erwartungswert einer Summe ist die Summe der Erwartungswerte.

**CHSH-Ungleichung:** Da $S = \pm 2$ experimentell bestimmt werden kann, folgt:

$$\boxed{\langle S\rangle \leq 2} \tag{6}$$

**Wichtig:** Diese Beziehung gilt für alle Arten von Experimenten, quantenmechanische und klassische!

**Hinweis zum Realismus:** Hier haben wir den Realismus eingeschmuggelt. Wir sind nie berechtigt, den tatsächlichen Wert von $S$ zu messen, da wir nie alle Komponenten gleichzeitig messen, obwohl wir annehmen, dass die gemessenen Werte existieren, unabhängig davon, ob wir sie messen. Dies ist nur erlaubt, weil wir annehmen, dass die gemessenen Werte existieren, unabhängig davon, ob wir sie messen!

=> Widerspricht relativity theory

### 12.1.2 Bells Theorem

laut Quantenmechanik ist die Kopplung verschränkter Teilchen stärker, als durch irgendeine lokal-realistische Theorie erklärbar wäre, auch wenn diese auf verborgene Variablen zurückgreift. Bell betrachtete dazu verschiedene Messungen an verschränkten Bits. Er konnte zeigen, dass die Ergebnisse in jeder möglichen lokal-realistischen Theorie eine Ungleichung erfüllen müssen, welche jedoch nach den Voraussagen der Quantenmechanik verletzt würde. Also könnte experimentell überprüft werden,ob diese Ungleichung verletzt wird.

**Conclusion:** Local realism is incompatible with the laws of quantum mechanics. The latter are well established experimentally, so we conclude that the former is not valid. There is now also experimental evidence for this conclusion (see Nobel Prize 2022). The debate about what should replace local realism (should we abandon locality or realism, or both?) is still ongoing, and there is no definitive answer.

## 12.2 Definitions

**Projection** A projection operator$P_v$ is an operator that transforms a vector $x$ into the vector that results from the projection of $x$ on $v$

- Projectors are operators with orthogonal eigenvectors and the eigenvalues 0 and 1.
- Projectors are self-adjoint operators

**Identity Matrix** The identity matrix 1 equals the sum of all projectors on the eigenstates $|\Phi_i\rangle$

## 12.3 Kochen-Specker theorem

Using the orthonormal basis $\{|\Phi_i\rangle\}$, we can construct $n$ projectors $P_{\Phi_i} = |\Phi_i\rangle\langle\Phi_i|$:

- The projector $P_{\Phi_i}$ has the eigenvalues $\lambda_j = \delta_{ij}$.
- Projectors on orthonormal states commute $[P_{\Phi_i}, P_{\Phi_j}] = 0$
- This means that all projectors have the same eigenvectors and has $P_{\Phi_i}$ the eigenvalues 0, except for the ith eigenvalue, which is equal to 1.

**Step 1: We define a set of 18 four-dimensional vectors:**

$$w_0 = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}, w_1 = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, w_2 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\\0\\0\end{pmatrix}, w_3 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\\0\\0\end{pmatrix}, w_4 = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, w_5 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\\0\end{pmatrix}, w_6 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\-1\\0\end{pmatrix}, w_7 = \frac{1}{2}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix}, w_8 = \frac{1}{2}\begin{pmatrix}1\\-1\\-1\\1\end{pmatrix},$$
$$w_9 = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\1\\1\end{pmatrix}, w_{10} = \frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}, w_{11} = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\-1\end{pmatrix}, w_{12} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}, w_{13} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\-1\end{pmatrix}, w_{14} = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\-1\\0\end{pmatrix}, w_{15} = \frac{1}{2}\begin{pmatrix}1\\1\\-1\\1\end{pmatrix}, w_{16} = \frac{1}{2}\begin{pmatrix}1\\1\\1\\-1\end{pmatrix}, w_{17} = \frac{1}{2}\begin{pmatrix}-1\\1\\1\\1\end{pmatrix}$$

Abbildung 44: Step 1

**Step 2: From these 18 vectors we form 9 basic sets (check it, in a base the vectors are orthogonal to each other):**

| | set 0 | set 1 | set 2 | set 3 | set 4 | set 5 | set 6 | set 7 | set 8 |
|---|---|---|---|---|---|---|---|---|---|
| phi_0 | w0 | w0 | w7 | w7 | w1 | w8 | w15 | w15 | w16 |
| phi_1 | w1 | w4 | w8 | w10 | w4 | w10 | w16 | w17 | w17 |
| phi_2 | w2 | w5 | w2 | w6 | w12 | w13 | w3 | w5 | w12 |
| phi_3 | w3 | w6 | w9 | w11 | w13 | w14 | w9 | w11 | w14 |

Abbildung 45: Step 2

Step 3:

- We have 9 different base sets $\{|\phi_{k,i}\rangle\}, k = 0, \ldots 8, i = 0, \ldots, 3$
- For all $k = 0, \ldots, 8$ we have

$$1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \sum_{i=0}^{3} P_{\phi_{k,i}}$$

- **Because of ARe, $v(1) = 1$, because all intrinsic values of the identity matrix are equal to one.**
- **Because of AFc, $v(1) = v\left(\sum P_{\phi_{k,i}}\right) = \sum v(P_{\phi_{k,i}})$**
- **Because of ANc, $v(P_{\phi_{k,i}})$ is either zero or one for all $P_{\phi_{k,i}}$. The value is defined regardless of whether we are taking measurements.**
- **➔ $v(1) = \sum v(P_{\phi_{k,i}}) = 1$ and $v\left(P_{\phi_{k,i}}\right) = 1, 0$ implies that for each $k$ there is one index $i$ with $v\left(P_{\phi_{k,i}}\right) = 1$ and for all other values of the index $i$, we have $v\left(P_{\phi_{k,j\neq i}}\right) = 0$ !**
- **Less confusing: We add up four numbers (each equal to one or zero) and get a sum of one. Therefore, one of these numbers is equal to one, the others are equal to zero.**

Abbildung 46: Step 3

Step 4:

- With $(v(1) = 1)$ && $(v(1) = \sum v(P_{\phi_{k,i}}))$ && $(v\left(P_{\phi_{k,i}}\right) \in \{0,1\})$ ➔ in each column, one entry must be one, all others are zero.

| | set 0 | set 1 | set 2 | set 3 | set 4 | set 5 | set 6 | set 7 | set 8 |
|---|---|---|---|---|---|---|---|---|---|
| phi_0 | w0 | w0 | w7 | w7 | w1 | w8 | w15 | w15 | w16 |
| phi_1 | w1 | w4 | w8 | w10 | w4 | w10 | w16 | w17 | w17 |
| phi_2 | w2 | w5 | w2 | w6 | w12 | w13 | w3 | w5 | w12 |
| phi_3 | w3 | w6 | w9 | w11 | w13 | w14 | w9 | w11 | w14 |

- ➔ We have nine columns, we need to have nine entries with a 1, all the others are zero. ➔ **The number of ones in this table is odd.**
- **Non−contextuality means that a certain measurement result $v(P_{\phi_{k,i}})$ exists independently, no matter in which combination of base vectors we study it!**
- Each base vector appears twice in the table, as indicated by the color scheme. Because of **ANc** (non-contextuality) and **ARe** (realism), we need to have an **even number of ones** in the table.
- One of the assumptions **ARe, ANc or AFc** must be wrong!

Abbildung 47: Step 4

**Cabello's proof of the Kochen – Specker-theorem**

## 12.4 Quantum Teleportation

### 12.4.1 Preliminaries

1. **Shared Entanglement:** Alice and Bob begin with an entangled pair of qubits. Alice can manipulate only her qubit (A), and Bob can manipulate only his (B). Their spatial separation ensures that neither can directly influence the other's qubit.

2. **Physical Realization:** Entanglement may be implemented using pairs of photons. Although mathematically well understood, the physical mechanism underlying entanglement correlations remains conceptually puzzling.

3. **Alice's Unknown Qubit:** Alice possesses an additional qubit in the unknown state
$$|\Phi\rangle = \alpha|0\rangle + \beta|1\rangle.$$
Her goal is to transmit this state to Bob without physically sending the qubit.

### 12.4.2 Step 1 — CNOT Operation

Alice applies a CNOT gate with her unknown qubit as the control and her half of the entangled pair as the target. This produces a non-local, three-qubit entangled state involving:

(Alice's unknown qubit) (Alice's entangled qubit) (Bob's entangled qubit).

The information in $\alpha$ and $\beta$ becomes distributed across the global system.

### 12.4.3 Step 2 — Hadamard Operation

Alice applies a Hadamard gate to her unknown qubit. After expanding and rearranging the resulting state, one can rewrite it so that Bob's qubit appears to contain $\alpha$ and $\beta$, but transformed by different Pauli operators depending on the measurement outcome. This rearrangement is purely algebraic—nothing physical has traveled to Bob.

### 12.4.4 Step 3 — Alice's Measurement

Alice measures her two qubits (the original unknown qubit and her part of the entangled pair). There are four possible outcomes:

$$00,\ 01,\ 10,\ 11,$$

each occurring with probability $1/4$. Each result leaves Bob's qubit in a related but not yet correct version of the original state $|\Phi\rangle$.

### 12.4.5 Step 4 — Classical Communication

Alice sends Bob the two classical bits corresponding to her measurement result. This communication must occur over a classical channel, ensuring that teleportation does not exceed the speed of light.

### 12.4.6 Step 5 — Bob's Correction

Upon receiving Alice's result, Bob applies one of four possible single-qubit corrections:

| Alice's result | Bob applies |
|---|---|
| 00 | $I$ |
| 01 | $X$ |
| 10 | $Z$ |
| 11 | $XZ$ |

After this correction, Bob's qubit becomes exactly

$$|\Phi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

Alice no longer retains any information about the original state because her qubits were measured in Step 3.

### 12.4.7 Conclusion

The state has been transferred from Alice to Bob without copying, consistent with the no-cloning theorem. Teleportation relies on shared entanglement and classical communication, and it succeeds without Alice ever learning the values of $\alpha$ or $\beta$.

## 12.5 Summary of the GHZ Argument

### 12.5.1 Introduction

The Greenberger–Horne–Zeilinger (GHZ) framework provides a stronger and more direct contradiction between quantum mechanics and the assumptions of *local realism* than Bell's original inequality. Local realism consists of two assumptions:

- **Realism (ARe):** Physical quantities possess definite values prior to and independent of measurement.
- **Locality (ALo):** A measurement on one particle cannot influence variables associated with a distant particle.

We analyze a three-qubit state closely related to the standard GHZ state:

$$|\Psi\rangle = \frac{1}{2}\big(|000\rangle - |011\rangle - |101\rangle - |110\rangle\big).$$

### 12.5.2 Measurement of the GHZ State

The three qubits are separated so that their measurements cannot influence one another. We denote the measurement outcomes on qubits $0$, $1$, and $2$ by $x_0$, $x_1$, and $x_2$. Quantum mechanics predicts the following outcomes, each with probability $1/4$:

$$\begin{aligned}
000 &\Rightarrow (x_0, x_1, x_2) = (0, 0, 0),\\
011 &\Rightarrow (x_0, x_1, x_2) = (0, 1, 1),\\
101 &\Rightarrow (x_0, x_1, x_2) = (1, 0, 1),\\
110 &\Rightarrow (x_0, x_1, x_2) = (1, 1, 0).
\end{aligned}$$

From these results, one obtains the constraint

$$x_0 \oplus x_1 \oplus x_2 = 0.$$

### 12.5.3 Applying Hadamard Transformations

Now consider a second scenario where, after separation, we apply Hadamard gates to qubits 1 and 2. The Hadamard operation is given by

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

After applying $H_1$ and $H_2$ to the GHZ state, we obtain the new state

$$H_1 H_2 |\Psi\rangle = \frac{1}{2}\left(|010\rangle + |001\rangle + |111\rangle + |100\rangle\right).$$

If we denote the measurement results after this transformation by $(x_0, h_1, h_2)$, then quantum mechanics predicts:

$$\begin{aligned} 010 &\Rightarrow (x_0, h_1, h_2) = (0, 1, 0), \\ 001 &\Rightarrow (x_0, h_1, h_2) = (0, 0, 1), \\ 111 &\Rightarrow (x_0, h_1, h_2) = (1, 1, 1), \\ 100 &\Rightarrow (x_0, h_1, h_2) = (1, 0, 0). \end{aligned}$$

Thus we obtain a new constraint:

$$x_0 \oplus h_1 \oplus h_2 = 1.$$

#### 12.5.4 Local Realism and Additional Hadamard Configurations

Because of realism (ARe), the six variables

$$x_0,\ x_1,\ x_2,\ h_0,\ h_1,\ h_2$$

must have predefined, context-independent values, even if we do not measure all of them.

Because of locality (ALo), measuring or applying a Hadamard gate on one qubit cannot change the value of a variable associated with a distant qubit.

Applying the same analysis to all combinations of qubits that may be Hadamard-transformed yields the following system of equalities (all modulo 2):

$$\begin{aligned} x_0 \oplus x_1 \oplus x_2 &= 0, \\ x_0 \oplus h_1 \oplus h_2 &= 1, \\ h_0 \oplus x_1 \oplus h_2 &= 1, \\ h_0 \oplus h_1 \oplus x_2 &= 1. \end{aligned}$$

#### 12.5.5 Deriving the Contradiction

If local realism is correct, all four equations must simultaneously hold for fixed values of the variables. Adding (modulo 2) all four constraints yields:

$$(x_0 \oplus x_0) \oplus (x_1 \oplus h_1) \oplus (x_2 \oplus h_2) \oplus (h_0 \oplus h_0) = 0 \oplus 1 \oplus 1 \oplus 1,$$

i.e.

$$0 = 1 \pmod 2,$$

a clear contradiction.

#### 12.5.6 Conclusion

The contradiction shows that the assumptions of *local realism* cannot reproduce the predictions of quantum mechanics for GHZ states. This result strengthens Bell's theorem by providing a deterministic, inequality-free demonstration that local hidden variable theories are incompatible with quantum phenomena.

# 13 Quantum Complexity (Quantum Information Sciences)

## 13.1 Begriffe

- **Complexity Theory:** Classifying problems based on their computational difficulty.
- **Difficulty**: is measured in terms of computational resources required.
- Various **types of resources** are considered:
  - **Time**: number of "computational state" transitions
  - **Space**: maximum size of the "computation state"
  - **Descriptive complexity**: size of a smallest model-instance (program) solving the problem
- **Computational Problem**: is a (binary) classification task, i.e., a task to decide whether any given string belongs to a certain **Yes** set or a certain **No** set.
  - The Yes and No sets characterize the problem; thus, a problem is defined by these two sets.
- A **promise problem** is a pair (A,B), consisting of two disjoint subsets of 0,1*.

- **Complexity Class**: a set of computational problems (p.p's) solvable by a certain class of algorithms, usually defined by a model of computation along with restrictions on the resources (e.g., time, space (memory)) that may be used.
- **a problem's complexity** is determined by its membership in various complexity classes.

- **Input (Promise):** A binary string that is guaranteed to belong to $A \cup B$.
- **Expected Output:**
  - Output **Yes** if the input string belongs to $A$.
  - Output **No** if the input string belongs to $B$.

Abbildung 48: Solving Promise Problems (Inputs not in $A \cup B$ are outside of scope and without guarantee about output)

### Promise Problems

### Measuring Complexity

- with Turing Machines: (ëasyto measure)
  - Time: maximum number of transitions used for inputs of length n
  - Space: Maximum number of tape cells utilized for inputs of length n
- with (Quantum-) Circuits (not as straightforward)
  - The processing time of a circuit is more difficult to measure.
  - Circuits are non-uniform models of computation, meaning individual circuits are designed for specific input sizes and cannot process inputs of arbitrary length.
    - The circuit size complexity defines complexity classes of problems that can be solved by circuit families of bounded size.
    - The circuit depth complexity defines complexity classes of problems that can be solved by circuit families of bounded depth (= max possible length of a path in the circuit).

### Circuit Family

- **Circuit Family** To solve (all instances of) a promise problem, we need not one circuit but a family of circuits C1,C2,...

Definition

- Every family $C_1, C_2, \ldots$ of circuits induces a function[1]

$$f : \{0,1\}^* \to \{0,1\}$$
$$f(x) := C_{|x|}(x)$$

- We say that the family $C_1, C_2, \ldots$ solves a promise problem if the induced function solves the promise problem.

Abbildung 49: Circuit Family Definition

- **Uniform Circuit Family** a circuit family C1,C2,... where the function n => Cn is computable by a Turing machine.
- **polynomial-time uniform family of circuits** is a uniform circuit family where there exists a Turing machine that, given the input $1^n$ (i.e., n in unary), outputs a description of Cn in time polynomial in n.

### Turing Machines vs Classical Boolean Circuits

- If $t(n) \geq n$, then

$$\mathrm{TIME}(t(n)) \subseteq \mathrm{SIZE}(t(n) \cdot \log t(n)).$$

- If a problem is in $\mathrm{TIME}(p(n))$ for some polynomial $p$, then it can be solved by a **polynomial-size uniform family of Boolean circuits**.
- This follows by **unrolling the Turing machine computation**: a TM running in $p(l)$ steps on inputs of length $l$ produces a circuit of size $O(p(l) \cdot \log l)$.
- Conversely, if a problem can be solved by a polynomial-size uniform Boolean circuit family, then it is in $\mathrm{TIME}(p(n))$ for some polynomial $p$.

- A Turing machine can simulate each circuit **gate-by-gate**, where each gate takes $O(1)$ time.
- The circuit descriptions can be generated by a Turing machine in polynomial time (uniformity requirement).

## 13.2 Complexity Classes

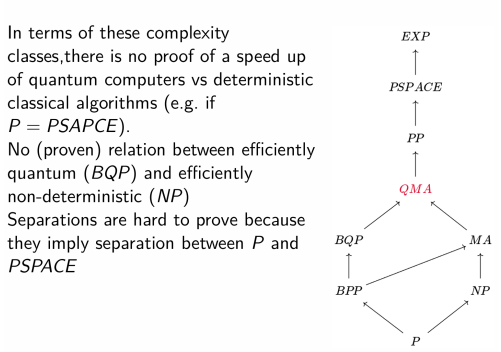



Abbildung 50: Complexity Classes Vergleich

### 13.2.1 Complexity Class P

- The complexity class P (f̃easible/tractable on classical hardware)
- Set of all problems which are solvable in polynomial time by a deterministic Turing machine.
- For an input x, the computation (asymptotically) ends after p(|x|) transitions (for some polynomial p)
- Polynomials are very robust and closed under many operations => do not depend on details of underlying model (as long as it's deterministic

### 13.2.2 Complexity Class EXP

- Set of all problems which are solvable in exponential time by a deterministic Turing machine.
- For an input x, the computation (asymptotically) ends after $2^{p(|x|)}$ transitions (for some polynomial p).
- $P \subsetneq EXP$ ({(T,x)| T accepts after at most $2^{|x|}$ steps} $\in$ EXP $\setminus$ P).

### 13.2.3 Complexity Class PSPACE

- Set of all problems which are decidable by a deterministic Turing machine that uses polynomial amount of space (on the tape).
- For an input x the number of cells used on the tape is bound by $p(|x|)$ for some polynomial p.
- $P \subseteq PSPACE$: To mutate a certain number of cells on the tape at least the same numbers of steps are required (XTime $\subseteq$ XSpace).
- $PSPACE \subseteq EXP$: if a deterministic Turing machine only touches $|p(n)|$ many cells in any computation that involves more steps than $|Q| \cdot |\sum|^{p(n)} \cdot p(n)$, then the computation never stops.

### 13.2.4 Complexity Class BPP (Bounded-error Probabilistic Polynomial time)

- A probabilistic Turing machine randomly executes one of two transition functions at every transition. (Turing Machine runs in polynomial time)
- Such a machine solves a promise problem with a certain probability of correctness.
- For all relevant inputs x:
  - If $x \in A$, then P[T(x) = Yes] $\geq \frac{3}{4}$. (or any constant $> \frac{1}{2}$ and probability can be reduced via repetition)
  - If $x \in B$, then P[T(x) = No] $\geq \frac{3}{4}$.

### 13.2.5 Complexity Class PP (Probabilistic Polynomial time)

consists of promise problems (A,B) for which there exists a probabilistic Turing machine T such that: T runs in polynomial time. (T is nondeterministic)

- If $x \in A$, then P[T(x) = Yes] $> \frac{1}{2}$.
- If $x \in B$, then P[T(x) = No] $> \frac{1}{2}$.

### 13.2.6 Complexity Class BQP (Bounded Error, Quantum, Polynomial)

Problems that can be solved efficiently (e2e in polynomial time) on a quantum computer
consists of promise problems (A,B) for which there exists a polynomial time uniform family $Q = Q_i | i \in \mathbb{N}$ of quantum circuits such that: For all relevant inputs x:

- If $x \in A$, then P[T(x) = Yes] $\geq \frac{3}{4}$. (again $\frac{3}{4}$ can be stipulated with other suitable value)
- If $x \in B$, then P[T(x) = No] $\geq \frac{3}{4}$.

### 13.2.7 Complexity Class NP

A problem A is in NP, if a nondeterministic Turing machine solves the problem in polynomial time.
A problem (A,B) is in NP, if there is a polynomial p and a language $L \in P$, such that:

- $w \in A \Rightarrow \exists x \left(|x| \leq p(|w|) \wedge (x, w) \in L\right).$
- $w \in B \Rightarrow \forall x \left(|x| \leq p(|w|) \Rightarrow (x, w) \notin L\right).$

The string x can be seen as a “proof” or “certificate” for the fact that $w \in A$.

### 13.2.8 Complexity Class MA (Merlin-Arthur)

Probabilistic version of NP; verification is a polynomial time probabilistic computation.
if there is a polynomial p and a polynomial-time probabilistic algorithm T such that:

- If $w \in A$, then $\exists x \left(|x| \leq p(|w|) \wedge \Pr[T(x, w) = \text{Yes}] \geq \frac{3}{4}\right).$
- If $w \in B$, then $\forall x \left(|x| \leq p(|w|) \Rightarrow \Pr[T(x, w) = \text{No}] \geq \frac{3}{4}\right).$

### 13.2.9 Complexity Class QMA

Verification is a polynomial time quantum computation, Certificate is polynomial-size quantum state
A problem $(A, B)$ is in QMA if there exist a polynomial $p$ and a polynomial-time uniform family of quantum circuits $Q = \{Q_i \mid i \in \mathbb{N}\}$ such that:

- If $w \in A$, then there exists a quantum state $x$ on $p(|w|)$ qubits with $\Pr[Q(x, w) = 1] \geq \frac{3}{4}$.
- If $w \in B$, then for every quantum state $x$ on $p(|w|)$ qubits it holds that $\Pr[Q(x, w) = 0] \geq \frac{3}{4}$.