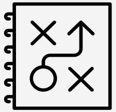
Symmetry-invariant Quantum Machine Learning Force Fields

Isabel Nha Minh Le[1,2,3], Oriel Kiss[4,5], Julian Schuhmacher[1], Ivano Tavernelli[1], Francesco Tacchino[1]. arXiv:2311.11362.

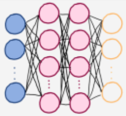
[1]IBM Quantum, IBM Research Europe – Zurich
[2]Institute for Quantum Information, RWTH Aachen University
[3]Department of Computer Science, Technical University of Munich
[4]European Organization for Nuclear Research (CERN)
[5]Department of Nuclear and Particle Physics, University of Geneva

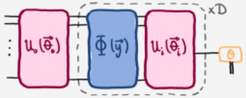
Today’s agenda

Challenge and existing tools



Molecular dynamics

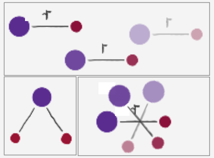




Classical and quantum tools

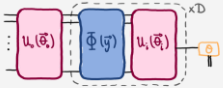
Symmetry-invariant Quantum Machine Learning Force Fields

Symmetries and QML



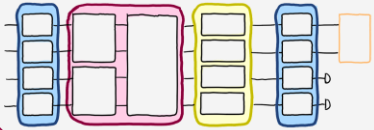
Relevant symmetries

Inclusion within existing tools



Symmetry-invariant QML

Results



Symmetry-invariant QML force fields for LiH, H2O, and H2O dimer

Summary

Summary of results and outcome of project

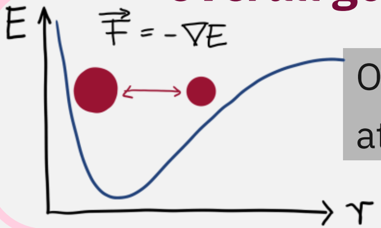
Challenge: Molecular Dynamics

Goal: Find dynamics of atoms in a chemical system

Approach: Numerical integration of $\vec{F} = m\ddot{\vec{r}}$

Requirement: Knowledge of $\vec{F}$, e.g., via potential energy surface (PES)

Overall goal



Obtain PES/ atomic forces

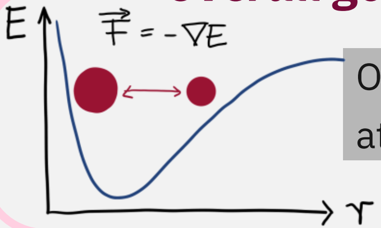
Challenge: Molecular Dynamics

Goal: Find dynamics of atoms in a chemical system

Approach: Numerical integration of $\vec{F} = m\ddot{\vec{r}}$

Requirement: Knowledge of $\vec{F}$, e.g., via potential energy surface (PES)

Overall goal



Obtain PES/ atomic forces

Existing Tools: Classical [1]

Accuracy ↑ / Comp. efficiency ↓

Ab initio (e.g. DFT) = quantum mech. description of atomic interaction; computationally very costy

Machine Learning FF = learning FF from reference data; no further approximations

Force Field (FF) = analytic empirical description of PES; classical treatment of atomic interaction; approximation-based

[1] Unke (2021)

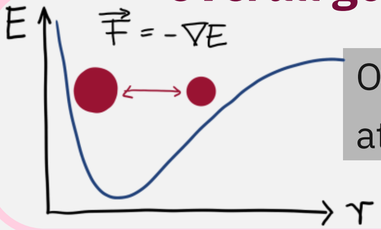
Challenge: Molecular Dynamics

Goal: Find dynamics of atoms in a chemical system

Approach: Numerical integration of $\vec{F} = m\ddot{\vec{r}}$

Requirement: Knowledge of $\vec{F}$, e.g., via potential energy surface (PES)

Overall goal



Obtain PES/ atomic forces

Existing Tools: Classical [1]

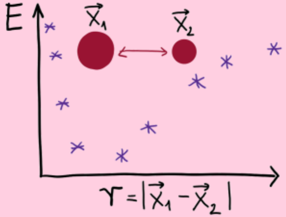
Accuracy ↑ / Comp. efficiency ↓

Ab initio (e.g. DFT) = quantum mech. description of atomic interaction; computationally very costy

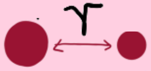
Machine Learning FF = learning FF from reference data; no further approximations

Force Field (FF) = analytic empirical description of PES; classical treatment of atomic interaction; approximation-based

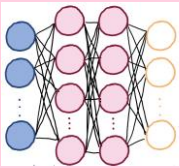
Reference data: $\{(\text{atomic config.} \mid \text{PES})\}$



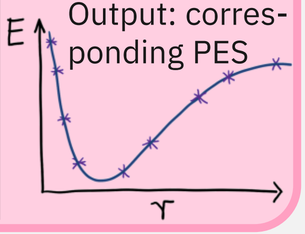
Input: atomic configuration



Regression



Output: corres-ponding PES



[1] Unke (2021)

Overall goal

Obtain PES/ atomic forces

Existing Tools: Classical

Accuracy ↑

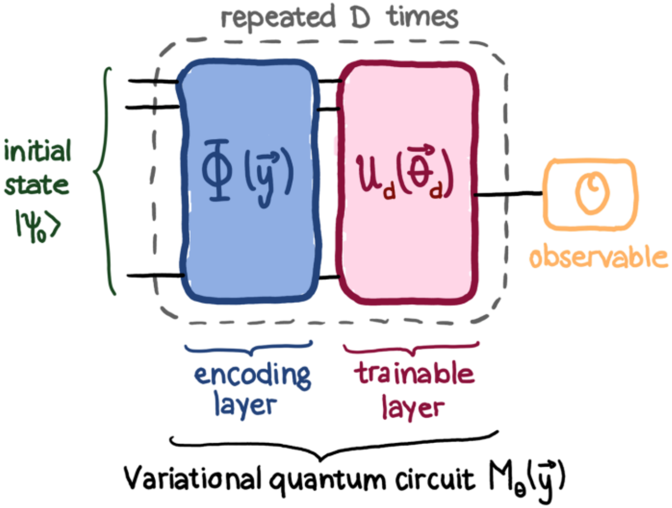
Ab initio (e.g. DFT) = computationally very costy

ML-FF = learning FF from reference data

Force Field (FF) = anal. empirical description of PES

Comp. eff. ↓

Existing Tools: Quantum [2]



[2] Kiss (2022) [3] Schuld (2021)

Overall goal

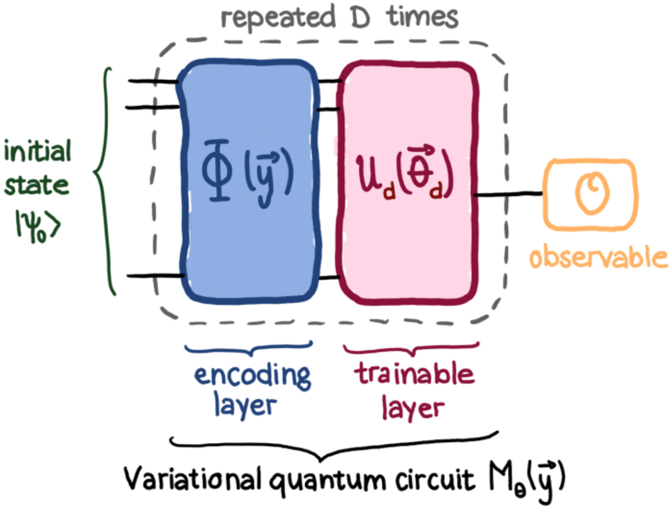
Obtain PES/ atomic forces

Existing Tools: Classical

Accuracy ↑ | Comp. eff. ↓

- **Ab initio** (e.g. DFT) = computationally very costy
- **ML-FF** = learning FF from reference data
- **Force Field (FF)** = anal. empirical description of PES

Existing Tools: Quantum [2]



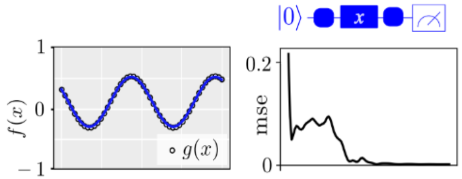
Quantum learning model

$$f_\theta(\vec{y}) = \langle \psi_0 | M_\theta^\dagger(\vec{y}) \, O \, M_\theta(\vec{y}) | \psi_0 \rangle$$

Quantum re-uploading models = Fourier sums [3]

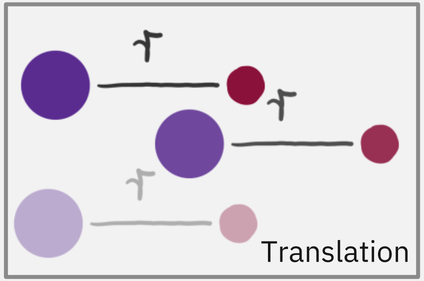
$$f(x) = \sum_{\omega \in \Omega} c_\omega e^{i\omega x}$$

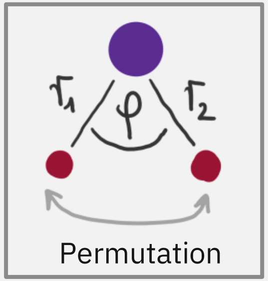
Ω dependent on D

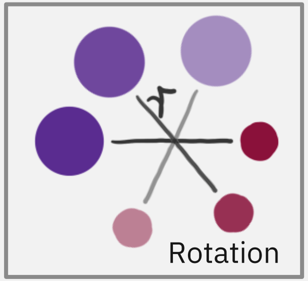


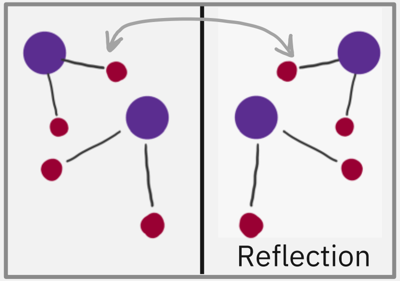
[2] Kiss (2022) [3] Schuld (2021)

Symmetries: Some relevant examples

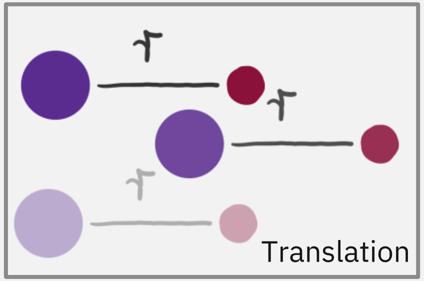


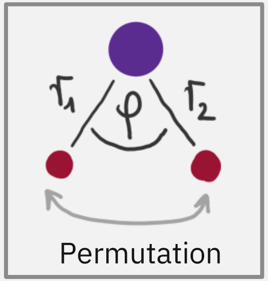


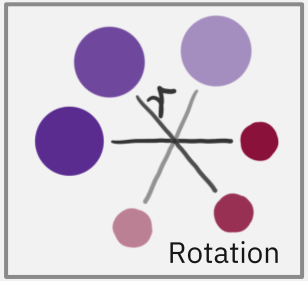


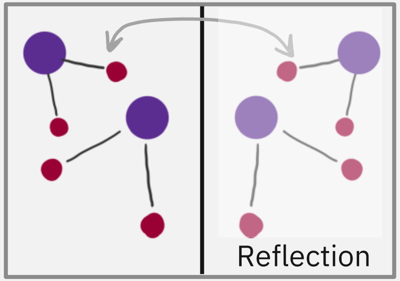


Symmetries: Some relevant examples









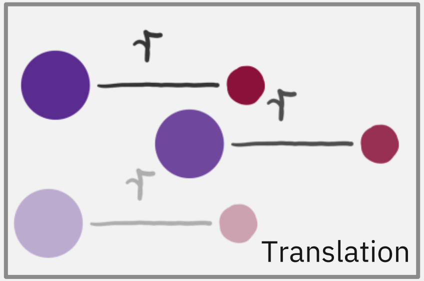
Symmetries: Inclusion within existing tools

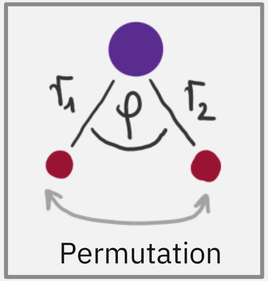
Internal coordinates

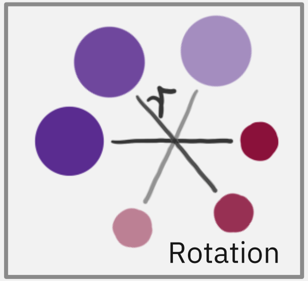
Symmetry functions [4] „Descriptor" of local chemical environment of atom

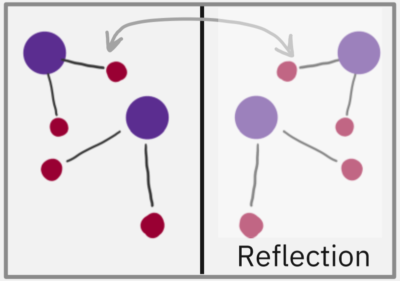
[4] Behler and Parinello (2007)

Symmetries: Some relevant examples



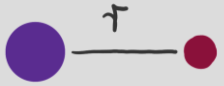


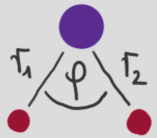




Symmetries: Inclusion within existing tools

Internal coordinates





Symmetry functions [4] — „Descriptor" of local chemical environment of atom

Symmetry incorporation by preprocessed input data

Challenges:

- Unique configuration – energy map
- Complete configuration – energy map
- Back transformation might amplify errors

[4] Behler and Parinello (2007)

Before: Input symmetry invariant representations of atomic configurations

Existing Tools: Classical

Accuracy ↑ / Comp. eff. ↓

Ab initio (e.g. DFT) = computationally very costy

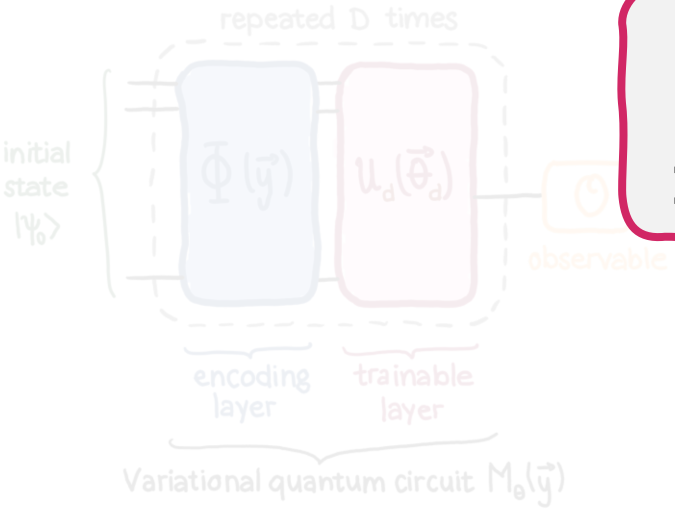
ML-FF = learning FF from reference data

Force Field (FF) = analytic empirical description of PES

Overall goal

Quantum model to obtain PES/ atomic forces

Existing Tools: Quantum



Invariance in

↓

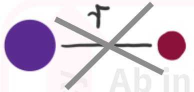
Invariance out

Quantum re-uploading models = Fourier sums

$$f(x) = \sum_{\omega \in \Omega} c_\omega e^{i\omega x}$$

Ω dependent on D

Before: Input symmetry invariant representations of atomic configurations



Now: Input atomic configurations as Cartesian coordinates

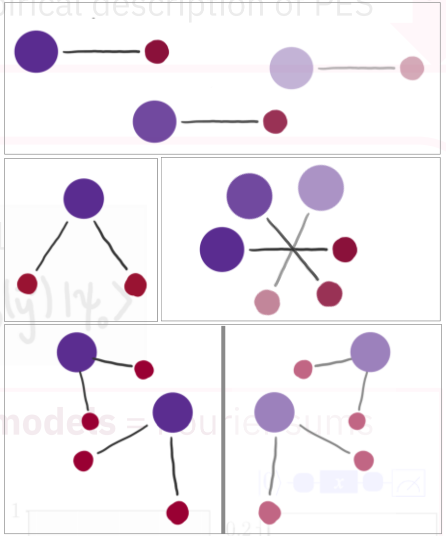
"Raw data"

~~**Invariance**~~ **in**

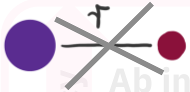
↓

Invariance out ?

Can we find an invariant quantum learning model?



Before: Input symmetry invariant representations of atomic configurations



Now: Input atomic configurations as Cartesian coordinates

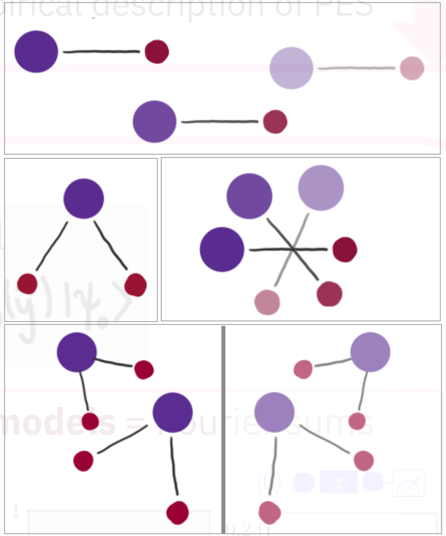
"Raw data"

~~**Invariance**~~ **in**

↓

Invariance out?

Can we find an invariant quantum learning model?



Advantages:

Reasonable inductive bias	Better trainability
Better generalization	No postprocessing

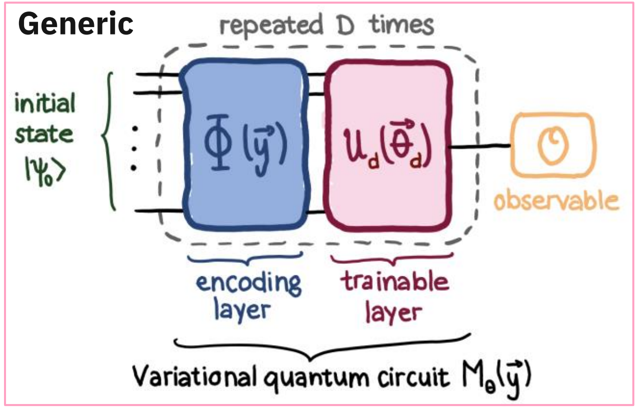
"Raw data"

~~Invariance~~ in

↓

Invariance out !

[4] Meyer (2022)



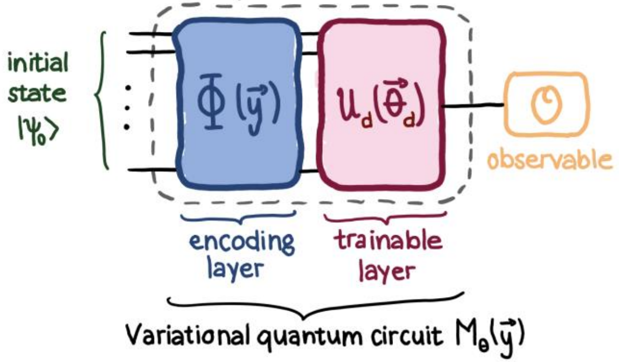
"Raw data"

~~Invariance~~ in

Invariance out !

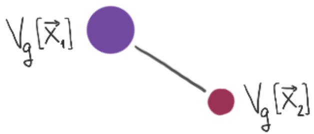
[4] Meyer (2022)

Generic



Invariant quantum learning model

$$f_\Theta(\vec{x}) = f_\Theta(V_g[\vec{x}])$$



V_g representation on data level

"Raw data"

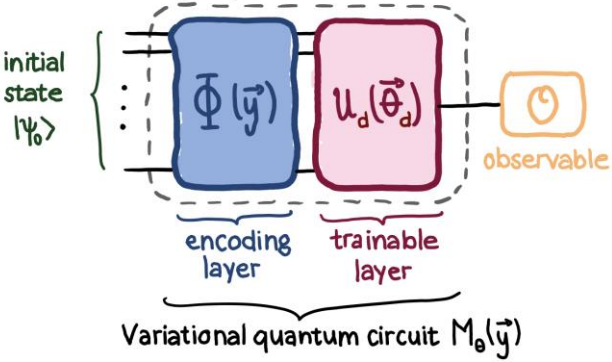
~~**Invariance**~~ **in**

↓

Invariance out !

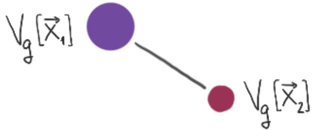
[4] Meyer (2022)

Generic

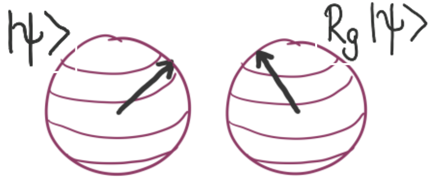


Invariant quantum learning model

$$f_\Theta(\vec{x}) = f_\Theta(V_g[\vec{x}])$$



V_g representation on data level



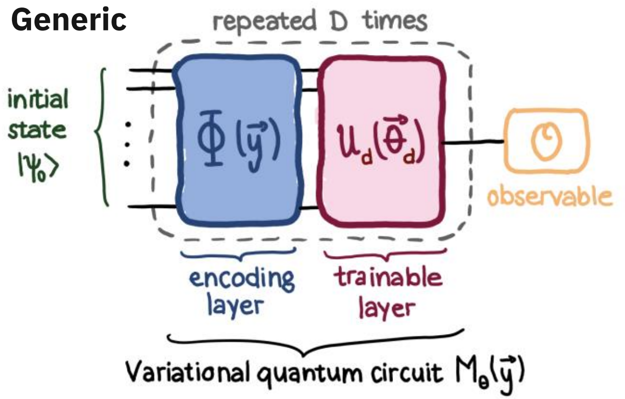
R_g unitary representation on Hilbert space

"Raw data"

~~Invariance~~ in

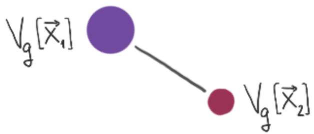
Invariance out!

[4] Meyer (2022)



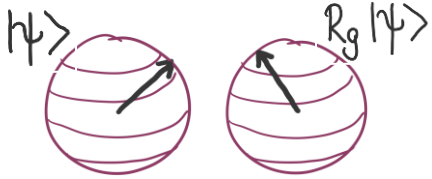
Invariant quantum learning model

$$f_\Theta(\vec{x}) = f_\Theta(V_g[\vec{x}])$$

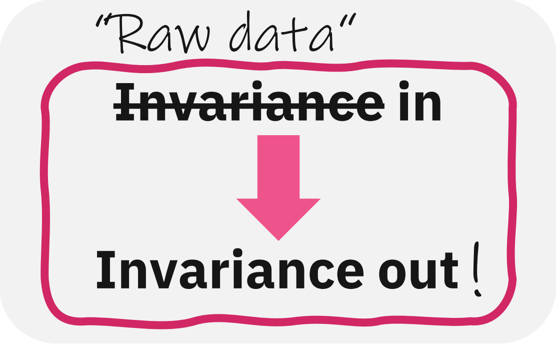


V_g representation on data level

IBM **Quantum**



R_g unitary representation on Hilbert space



Equivariant re-uploading circuit

$$M_\Theta(V_g[\vec{x}]) = R_g M_\Theta(\vec{x}) R_g^\dagger$$

Push the symmetry action from data level to qubit level

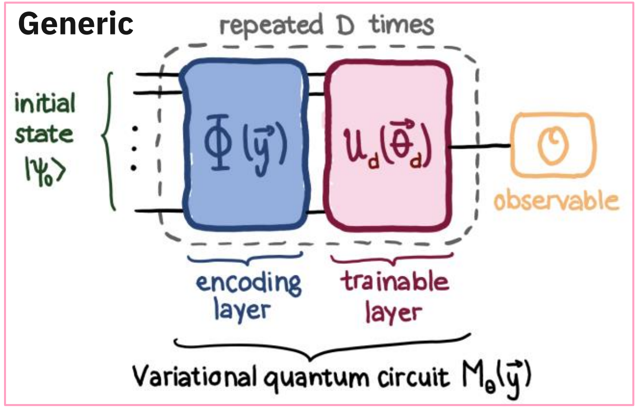
Equivariant encoding

$$\Phi(V_g[\vec{x}]) = R_g \Phi(\vec{x}) R_g^\dagger$$

Equivariant trainable

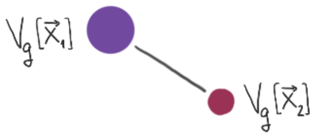
$$[\mathcal{U}_d(\vec{\theta}_d), R_g] = 0$$

[4] Meyer (2022)



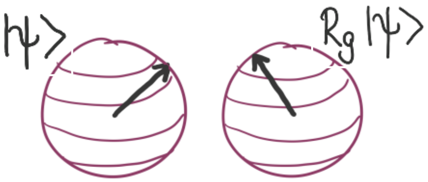
Invariant quantum learning model

$$f_\Theta(\vec{x}) = f_\Theta(V_g[\vec{x}])$$



V_g representation on data level

IBM **Quantum**



R_g unitary representation on Hilbert space

"Raw data"

~~**Invariance**~~ **in**

↓

Invariance out !

Equivariant re-uploading circuit

$$M_\Theta(V_g[\vec{x}]) = R_g M_\Theta(\vec{x}) R_g^\dagger$$

Push the symmetry action from data level to qubit level

Invariant observable: $R_g \mathcal{O} R_g^\dagger = \mathcal{O}$

Invariant initial state: $R_g|\psi_0\rangle = |\psi_0\rangle$

Let the initial state and observable absorb the symmetry action

Equivariant encoding

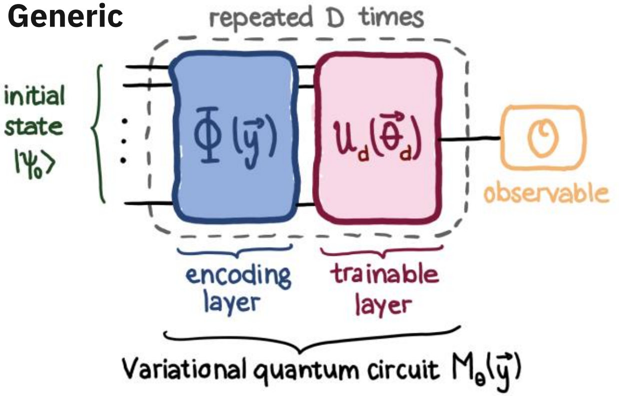
$$\Phi(V_g[\vec{x}]) = R_g \Phi(\vec{x}) R_g^\dagger$$

Equivariant trainable

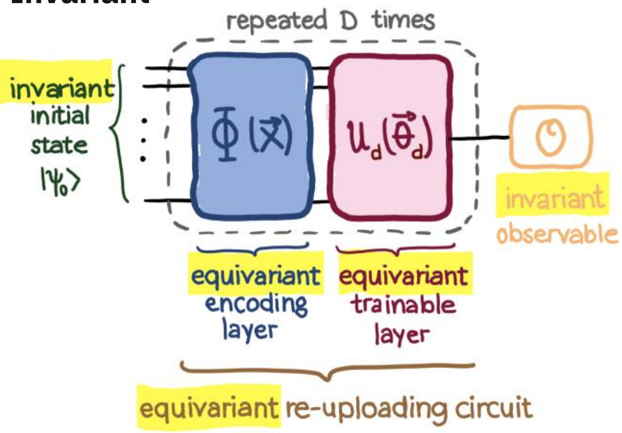
$$[\mathcal{U}_d(\vec{\theta}_d), R_g] = 0$$

[4] Meyer (2022)

Generic

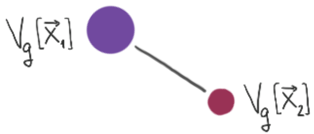


Invariant



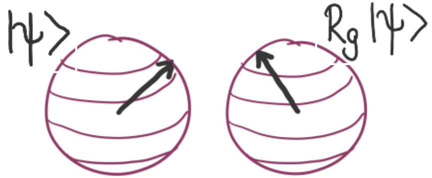
Invariant quantum learning model

$$f_\Theta(\vec{x}) = f_\Theta(V_g[\vec{x}])$$



V_g representation on data level

IBM **Quantum**



R_g unitary representation on Hilbert space

"Raw data"

~~**Invariance**~~ **in**

Invariance out!

Equivariant re-uploading circuit

$$M_\Theta(V_g[\vec{x}]) = R_g M_\Theta(\vec{x}) R_g^\dagger$$

Push the symmetry action from data level to qubit level

Invariant observable: $R_g \mathcal{O} R_g^\dagger = \mathcal{O}$

Invariant initial state: $R_g|\psi_0\rangle = |\psi_0\rangle$

Let the initial state and observable absorb the symmetry action

Equivariant encoding

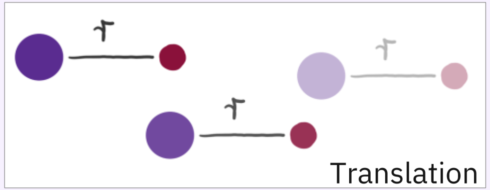
$$\Phi(V_g[\vec{x}]) = R_g \Phi(\vec{x}) R_g^\dagger$$

Equivariant trainable

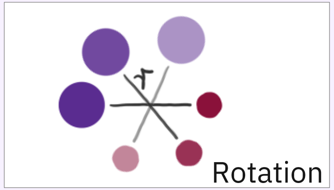
$$[\mathcal{U}_d(\vec{\theta}_d), R_g] = 0$$

[4] Meyer (2022)

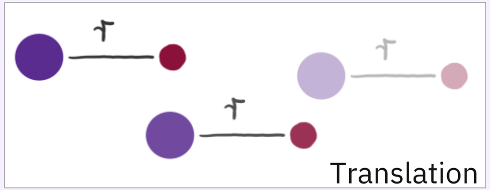
Diatomic case: lithium hydride (LiH)



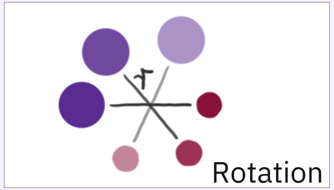
Relevant symmetries



Diatomic case: lithium hydride (LiH)

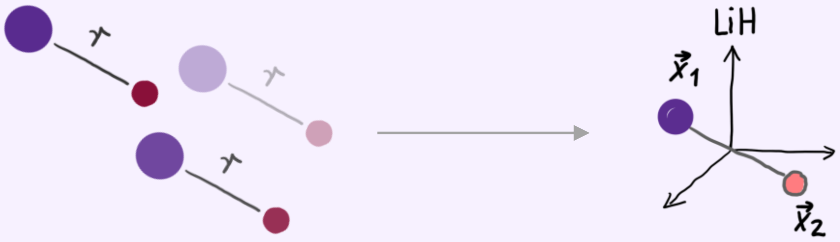


Relevant symmetries



Translational symmetry

Input data: $\vec{x}_1, \vec{x}_2 \in \mathbb{R}^3$ but translational invariant



Diatomic case (LiH): rotational invariance

*Rotational representation on **data** level*

$$r(\psi, \theta, \phi) = r_z(\psi) r_x(\theta) r_z(\phi)$$

Rotation matrix in $\mathbb{R}^3$

*Rotational representation on **qubit** level*

$$U(\psi, \theta, \phi) = R_Z(\psi) R_X(\theta) R_Z(\phi)$$

Single-qubit rotation

Diatomic case (LiH): rotational invariance

*Rotational representation on **data** level*

$$r(\psi, \theta, \phi) = r_z(\psi) r_x(\theta) r_z(\phi)$$

Rotation matrix in $\mathbb{R}^3$

*Rotational representation on **qubit** level*

$$U(\psi, \theta, \phi) = R_Z(\psi) R_X(\theta) R_Z(\phi)$$

Single-qubit rotation

Rotationally invariant quantum state

Invariant two-qubit state: $|S\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$

Invariant four-qubit state: $|\psi_0\rangle = |S\rangle|S\rangle$

Let the initial state absorb the symmetry action

Diatomic case (LiH): rotational invariance

Rotational representation on ***data*** *level*

$$r(\psi, \theta, \phi) = r_z(\psi) r_x(\theta) r_z(\phi)$$

Rotation matrix in $\mathbb{R}^3$

Rotational representation on ***qubit*** *level*

$$U(\psi, \theta, \phi) = R_Z(\psi) R_X(\theta) R_Z(\phi)$$

Single-qubit rotation

Rotationally invariant quantum state

Invariant two-qubit state: $|S\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$

Invariant four-qubit state: $|\psi_0\rangle = |S\rangle|S\rangle$

Let the initial state absorb the symmetry action

Rotationally equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

Diatomic case (LiH): rotational invariance

*Rotational representation on **data** level*

$$r(\psi,\theta,\phi) = r_z(\psi)r_x(\theta)r_z(\phi)$$

Rotation matrix in $\mathbb{R}^3$

*Rotational representation on **qubit** level*

$$U(\psi,\theta,\phi) = R_Z(\psi)R_X(\theta)R_Z(\phi)$$

Single-qubit rotation

Rotationally invariant quantum state

Invariant two-qubit state: $|S\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$

Invariant four-qubit state: $|\psi_0\rangle = |S\rangle|S\rangle$

Let the initial state absorb the symmetry action

Rotationally equivariant data embedding

How to encode one data point $\vec{x} = (x,y,z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

Equivariance:

$$\Phi(r(\psi,\theta,\phi)\vec{x}) = U(\psi,\theta,\phi)\Phi(\vec{x})U(\psi,\theta,\phi)^\dagger$$

Push the symmetry action from data level to qubit level

Diatomic case (LiH): rotational invariance

IBM **Quantum**

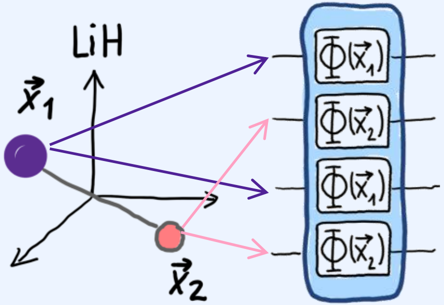
Rotationally equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode two atoms $\vec{x}_1, \vec{x}_2$ in four qubits:

$$\Phi(\vec{x}_1, \vec{x}_2) = \Phi^{(1)}(\vec{x}_1)\Phi^{(2)}(\vec{x}_2)\Phi^{(3)}(\vec{x}_1)\Phi^{(4)}(\vec{x}_2)$$



Push the symmetry action from data level to qubit level

Diatomic case (LiH): rotational invariance

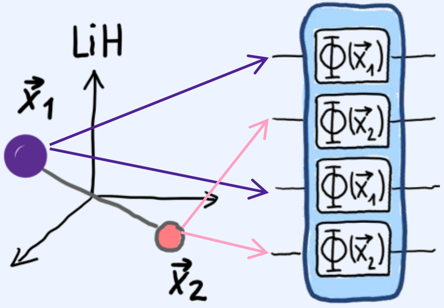
Rotationally equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode two atoms $\vec{x}_1, \vec{x}_2$ in four qubits:

$$\Phi(\vec{x}_1, \vec{x}_2) = \Phi^{(1)}(\vec{x}_1)\Phi^{(2)}(\vec{x}_2)\Phi^{(3)}(\vec{x}_1)\Phi^{(4)}(\vec{x}_2)$$



Push the symmetry action from data level to qubit level

Rotationally invariant operator (Heisenberg interaction)

$$H^{(i,j)}(J) = -J\left(X^{(i)}X^{(j)} + Y^{(i)}Y^{(j)} + Z^{(i)}Z^{(j)}\right), \qquad J \in \mathbb{R}$$

Diatomic case (LiH): rotational invariance

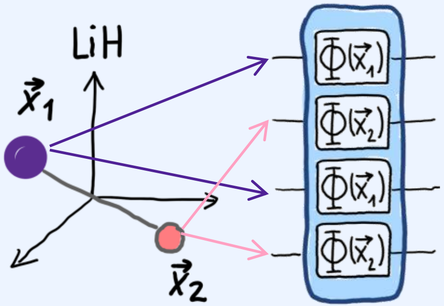
Rotationally equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode two atoms $\vec{x}_1, \vec{x}_2$ in four qubits:

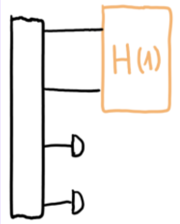
$$\Phi(\vec{x}_1, \vec{x}_2) = \Phi^{(1)}(\vec{x}_1)\Phi^{(2)}(\vec{x}_2)\Phi^{(3)}(\vec{x}_1)\Phi^{(4)}(\vec{x}_2)$$



Push the symmetry action from data level to qubit level

Rotationally invariant operator (Heisenberg interaction)

$$H^{(i,j)}(J) = -J\left(X^{(i)}X^{(j)} + Y^{(i)}Y^{(j)} + Z^{(i)}Z^{(j)}\right), \qquad J \in \mathbb{R}$$



Invariant observable

$$\mathcal{O} = X^{(0)}X^{(1)} + Y^{(0)}Y^{(1)} + Z^{(0)}Z^{(1)}$$

Diatomic case (LiH): rotational invariance

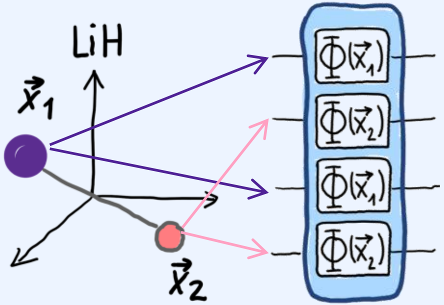
Rotationally equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode two atoms $\vec{x}_1, \vec{x}_2$ in four qubits:

$$\Phi(\vec{x}_1, \vec{x}_2) = \Phi^{(1)}(\vec{x}_1)\Phi^{(2)}(\vec{x}_2)\Phi^{(3)}(\vec{x}_1)\Phi^{(4)}(\vec{x}_2)$$



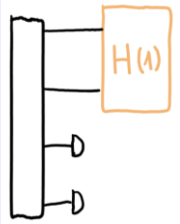
Push the symmetry action from data level to qubit level

Rotationally invariant operator (Heisenberg interaction)

$$H^{(i,j)}(J) = -J\left(X^{(i)}X^{(j)} + Y^{(i)}Y^{(j)} + Z^{(i)}Z^{(j)}\right), \qquad J \in \mathbb{R}$$

Equivariant parametrized operator

$$RH^{(i,j)}(J) = \exp\left(-iH^{(i,j)}(J)\right), \qquad J \in \mathbb{R}$$



Invariant observable

$$\mathcal{O} = X^{(0)}X^{(1)} + Y^{(0)}Y^{(1)} + Z^{(0)}Z^{(1)}$$

Diatomic case (LiH): rotational invariance

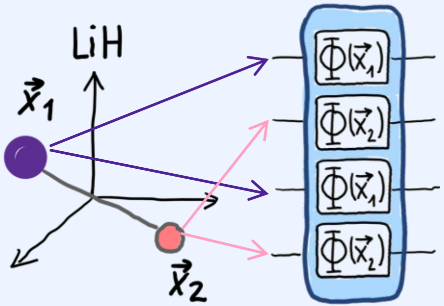
Rotationally equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode two atoms $\vec{x}_1, \vec{x}_2$ in four qubits:

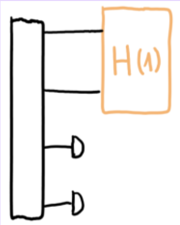
$$\Phi(\vec{x}_1, \vec{x}_2) = \Phi^{(1)}(\vec{x}_1)\Phi^{(2)}(\vec{x}_2)\Phi^{(3)}(\vec{x}_1)\Phi^{(4)}(\vec{x}_2)$$



Push the symmetry action from data level to qubit level

Rotationally invariant operator (Heisenberg interaction)

$$H^{(i,j)}(J) = -J\left(X^{(i)}X^{(j)} + Y^{(i)}Y^{(j)} + Z^{(i)}Z^{(j)}\right), \qquad J \in \mathbb{R}$$



Invariant observable

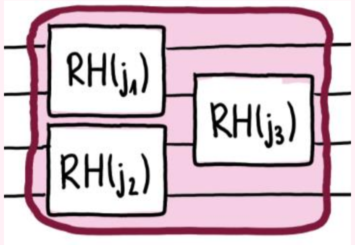
$$\mathcal{O} = X^{(0)}X^{(1)} + Y^{(0)}Y^{(1)} + Z^{(0)}Z^{(1)}$$

Equivariant trainable layer

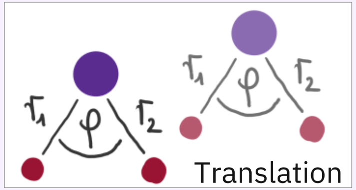
$$\mathcal{U}(\vec{j}) = RH^{(0,1)}(j_1)RH^{(2,3)}(j_2)RH^{(1,2)}(j_3)$$

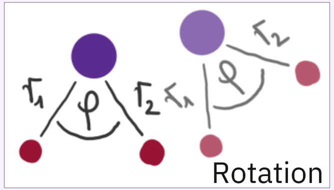
Equivariant parametrized operator

$$RH^{(i,j)}(J) = \exp\left(-iH^{(i,j)}(J)\right), \qquad J \in \mathbb{R}$$

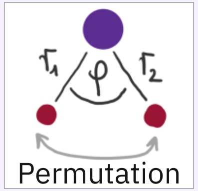


Triatomic case with two atom types: H2O

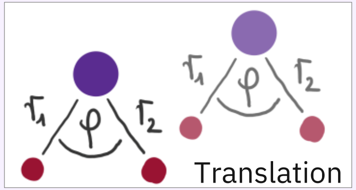


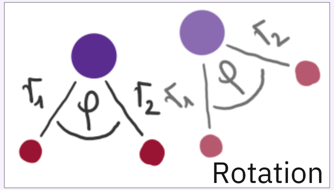


Relevant symmetries

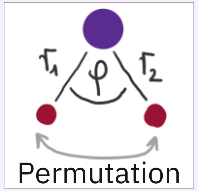


Triatomic case with two atom types: H2O





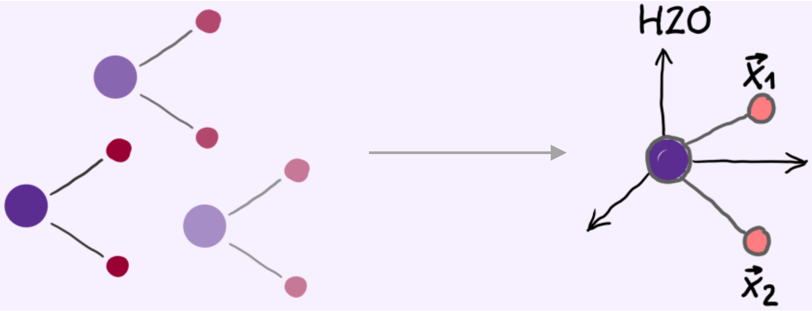
Relevant symmetries



Translational symmetry

Set O-atom into the origin

Input data: $\vec{x}_1, \vec{x}_2 \in \mathbb{R}^3$
but translational invariant



Triatomic case (H2O): permutational invariance

*Permutational representation on **data** level*

$$\sigma(\vec{x}_i, \vec{x}_j) = (\vec{x}_j, \vec{x}_i)$$

*Permutational representation on **qubit** level*

$$U(i,j) = SWAP(i,j)$$

Triatomic case (H2O): permutational invariance

*Permutational representation on **data** level*

$$\sigma(\vec{x}_i, \vec{x}_j) = (\vec{x}_j, \vec{x}_i)$$

*Permutational representation on **qubit** level*

$$U(i,j) = SWAP(i,j)$$

Permutationally invariant quantum state

Invariant two-qubit state: $|S\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$

Invariant four-qubit state: $|\psi_0\rangle = |S\rangle|S\rangle$

Let the initial state absorb the symmetry action

Permutationally equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode two atoms $\vec{x}_1, \vec{x}_2$:

$$\Phi(\vec{x}_1, \vec{x}_2) = \Phi^{(1)}(\vec{x}_1)\Phi^{(2)}(\vec{x}_2)$$

Equivariance:

$$\Phi\left(\sigma(\vec{x}_i, \vec{x}_j)\right) = SWAP(i,j)\Phi(\vec{x}_i, \vec{x}_j)SWAP(i,j)$$

Push the symmetry action from data level to qubit level

Triatomic case (H2O): permutational invariance

IBM **Quantum**

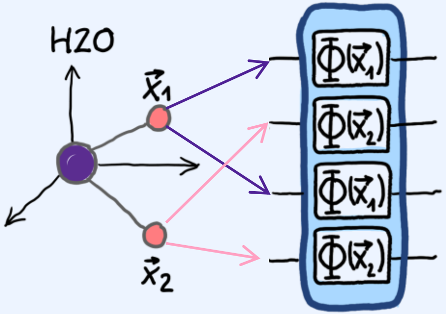
Permutationally equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode two atoms $\vec{x}_1, \vec{x}_2$ in four qubits:

$$\Phi(\vec{x}_1, \vec{x}_2) = \Phi^{(1)}(\vec{x}_1)\Phi^{(2)}(\vec{x}_2)\Phi^{(3)}(\vec{x}_1)\Phi^{(4)}(\vec{x}_2)$$



Push the symmetry action from data level to qubit level

Triatomic case (H2O): permutational invariance

IBM **Quantum**

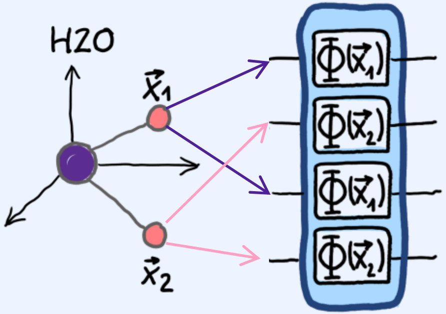
Permutationally equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ]), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode two atoms $\vec{x}_1, \vec{x}_2$ in four qubits:

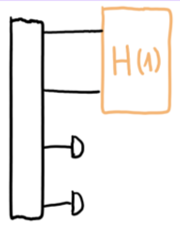
$$\Phi(\vec{x}_1, \vec{x}_2) = \Phi^{(1)}(\vec{x}_1)\Phi^{(2)}(\vec{x}_2)\Phi^{(3)}(\vec{x}_1)\Phi^{(4)}(\vec{x}_2)$$



Push the symmetry action from data level to qubit level

Permutation invariant operator (Heisenberg interaction)

$$H^{(i,j)}(J) = -J\left(X^{(i)}X^{(j)} + Y^{(i)}Y^{(j)} + Z^{(i)}Z^{(j)}\right), \qquad J \in \mathbb{R}$$



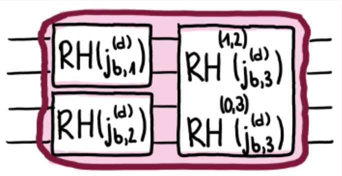
Invariant observable

$$\mathcal{O} = X^{(0)}X^{(1)} + Y^{(0)}Y^{(1)} + Z^{(0)}Z^{(1)}$$

Equivariant parametrized operator

$$RH^{(i,j)}(J) = \exp\left(-iH^{(i,j)}(J)\right), (i,j) \in \{(0,1),(2,3)\}$$

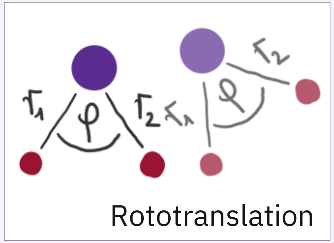
$$RH^{(1,2)}(J) \cdot RH^{(0,3)}(J), \qquad J \in \mathbb{R}$$

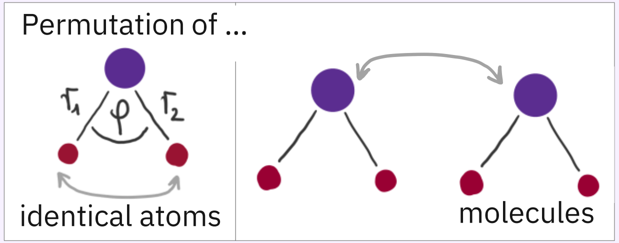


Equivariant trainable layer

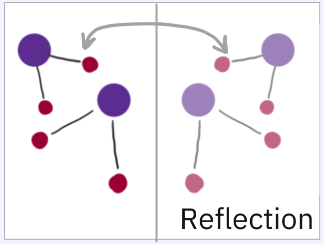
$$\mathcal{U}(\vec{J}) = RH^{(0,1)}(j_1)RH^{(2,3)}(j_2)RH^{(1,2)}(j_3)RH^{(0,3)}(j_3)$$

Triatomic dimer: H2O

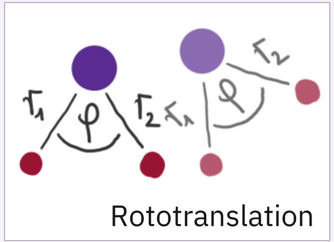


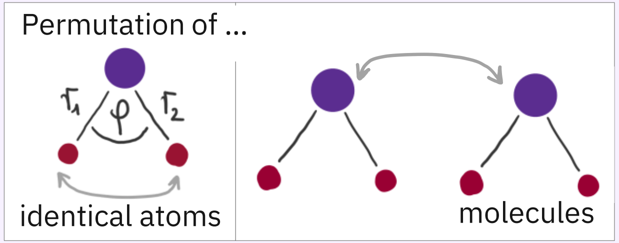


Relevant symmetries

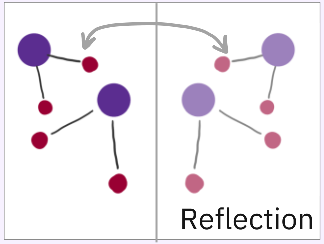


Triatomic dimer: H2O



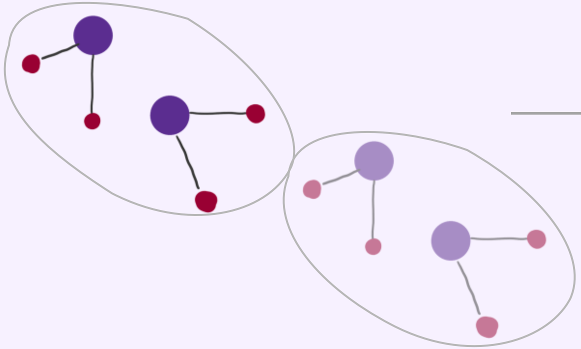


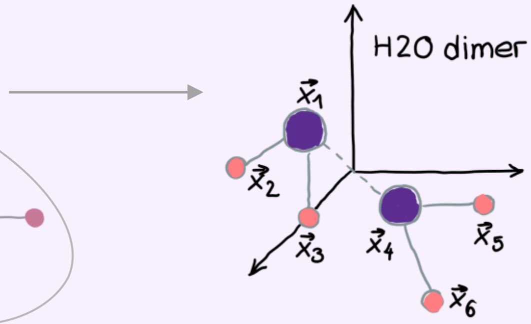
Relevant symmetries



Translational symmetry

Input data: $\vec{x}_i \in \mathbb{R}^3, i = 1 \dots 6$
but translational invariant





Triatomic dimer (H2O): reflection invariance

Reflection representation on ***data*** *level*

$$V(\vec{x}_i) = -\vec{x}_i$$

Reflection representation on ***qubit*** *level*

$$U = Z$$

Triatomic dimer (H2O): reflection invariance

*Reflection representation on **data** level*

$V(\vec{x}_i) = -\vec{x}_i$

*Reflection representation on **qubit** level*

$U = Z$

Reflection-equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ] \otimes X)$$

How to encode several data points $\vec{x}_i$:

$$\Phi(\mathcal{X}) = \prod_{i=1}^{N-1} \exp\left(-i\alpha_{\text{enc},a}\vec{x}_i\vec{\sigma}^{(i)}X^{(N)}\right)$$

Equivariance:

$$\Phi(-\mathcal{X}) = Z^{(N)}\Phi(\mathcal{X})Z^{(N)}$$

Push the symmetry action from data level to qubit level

Triatomic dimer (H2O): reflection invariance

IBM **Quantum**

*Reflection representation on **data** level*

$$V(\vec{x}_i) = -\vec{x}_i$$

*Reflection representation on **qubit** level*

$$U = Z$$

Reflection-invariant quantum state

Denote: $|S_{ij}\rangle = \frac{1}{\sqrt{2}}(|0_i 1_j\rangle - |0_i 1_j\rangle)$

Invariant 7-qubit state:

$$|\psi_0\rangle = |S_{12}\rangle|S_{45}\rangle|S_{30}\rangle|0\rangle$$

Let the initial state absorb the symmetry action

Reflection-equivariant data embedding

How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ] \otimes X)$$

How to encode several data points $\vec{x}_i$:

$$\Phi(\mathcal{X}) = \prod_{i=1}^{N-1} \exp\left(-i\alpha_{\mathrm{enc},a}\vec{x}_i\vec{\sigma}^{(i)}X^{(N)}\right)$$

Equivariance:

$$\Phi(-\mathcal{X}) = Z^{(N)}\Phi(\mathcal{X})Z^{(N)}$$

Push the symmetry action from data level to qubit level

Triatomic dimer (H2O): reflection invariance

IBM **Quantum**

Relfection equivariant data embedding

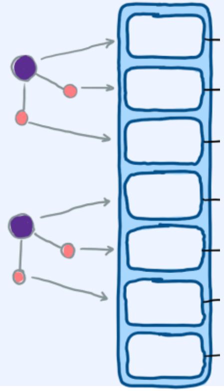
How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ] \otimes Z), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode 6 atoms $\vec{x}_i$ in $N = 7$ qubits:

$$\Phi(\mathcal{X}) = \prod_{i=1}^{N-1} \exp\left(-i\alpha_{\text{enc},a}\vec{x}_i\vec{\sigma}^{(i)}X^{(N)}\right)$$

Push the symmetry action from data level to qubit level



Triatomic dimer (H2O): reflection invariance

IBM **Quantum**

Relfection equivariant data embedding

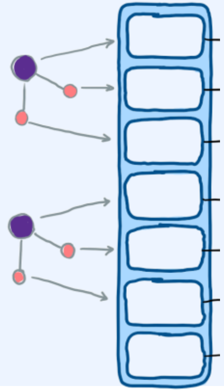
How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ] \otimes Z), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode 6 atoms $\vec{x}_i$ in $N = 7$ qubits:

$$\Phi(\mathcal{X}) = \prod_{i=1}^{N-1} \exp\left(-i\alpha_{\text{enc},a}\vec{x}_i\vec{\sigma}^{(i)}X^{(N)}\right)$$

Push the symmetry action from data level to qubit level



Reflection invariant operator (Heisenberg interaction extended)

$$H^{(i,j)}(J) = -J\left(X^{(i)}X^{(j)} + Y^{(i)}Y^{(j)} + Z^{(i)}Z^{(j)}\right) \cdot Z^{(k)}, \qquad J \in \mathbb{R}$$

Invariant observable

$$\mathcal{O} = X^{(0)}X^{(3)} + Y^{(0)}Y^{(3)} + Z^{(0)}Z^{(3)} \otimes Z^6$$

Triatomic dimer (H2O): reflection invariance

Relfection equivariant data embedding

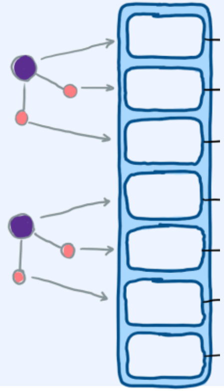
How to encode one data point $\vec{x} = (x, y, z)^T$:

$$\Phi(\vec{x}) = \exp(-i\alpha_{enc}[xX + yY + zZ] \otimes Z), \quad \alpha_{enc} \in \mathbb{R}$$

How to encode 6 atoms $\vec{x}_i$ in $N = 7$ qubits:

$$\Phi(\mathcal{X}) = \prod_{i=1}^{N-1} \exp\left(-i\alpha_{\mathrm{enc},a}\vec{x}_i\vec{\sigma}^{(i)}X^{(N)}\right)$$

Push the symmetry action from data level to qubit level



Reflection invariant operator (Heisenberg interaction extended)

$$H^{(i,j)}(J) = -J\left(X^{(i)}X^{(j)} + Y^{(i)}Y^{(j)} + Z^{(i)}Z^{(j)}\right) \cdot Z^{(k)}, \qquad J \in \mathbb{R}$$

Invariant observable

$$\mathcal{O} = X^{(0)}X^{(3)} + Y^{(0)}Y^{(3)} + Z^{(0)}Z^{(3)} \otimes Z^6$$

Equivariant parametrized operators

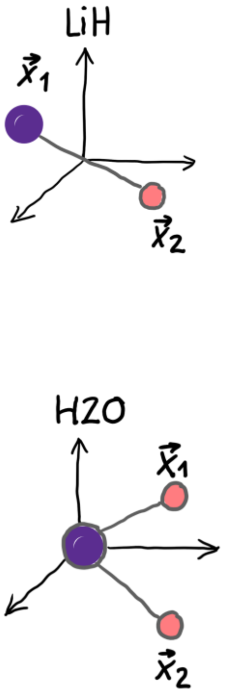
$$RH^{(i,j)}(J) = \exp\left(-iH^{(i,j)}(J)\right), (i,j) \in \{(1,2), (4,5), (0,4)\}$$

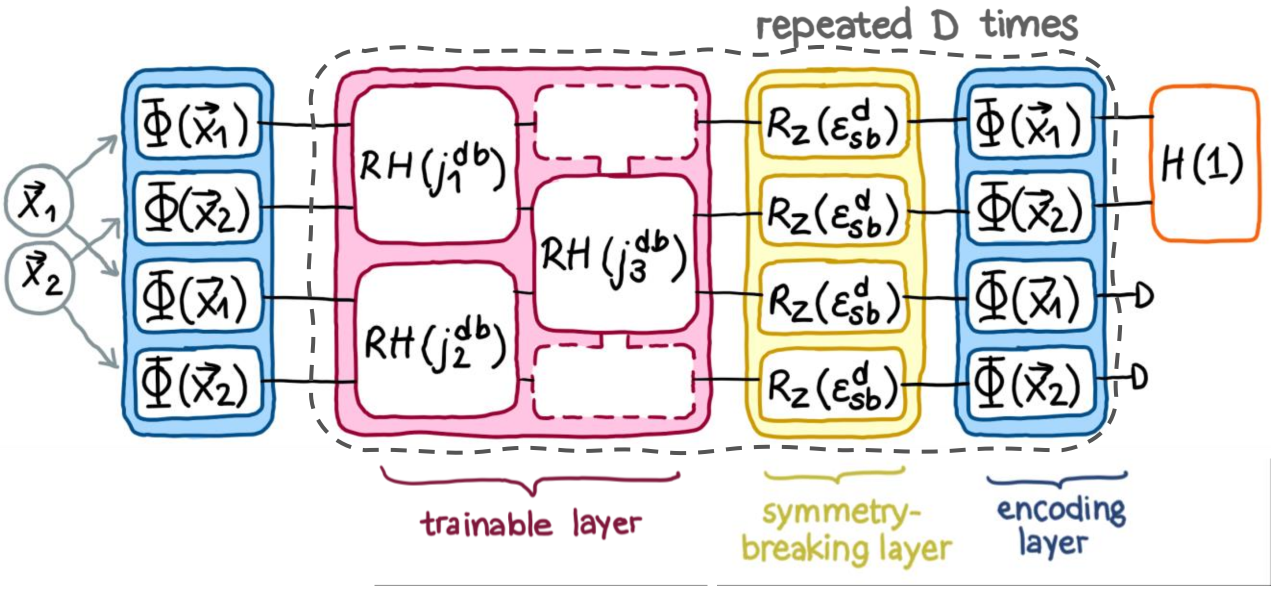
$$R_Z^{(6)}(J), \qquad J \in \mathbb{R}$$

Equivariant trainable layer

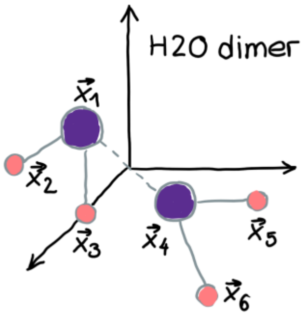
$$\mathcal{U}(\vec{J}) = RH^{(1,2)}(j_1)RH^{(4,5)}(j_2)RH^{(0,4)}(j_3)R_Z^{(6)}(j_4)$$

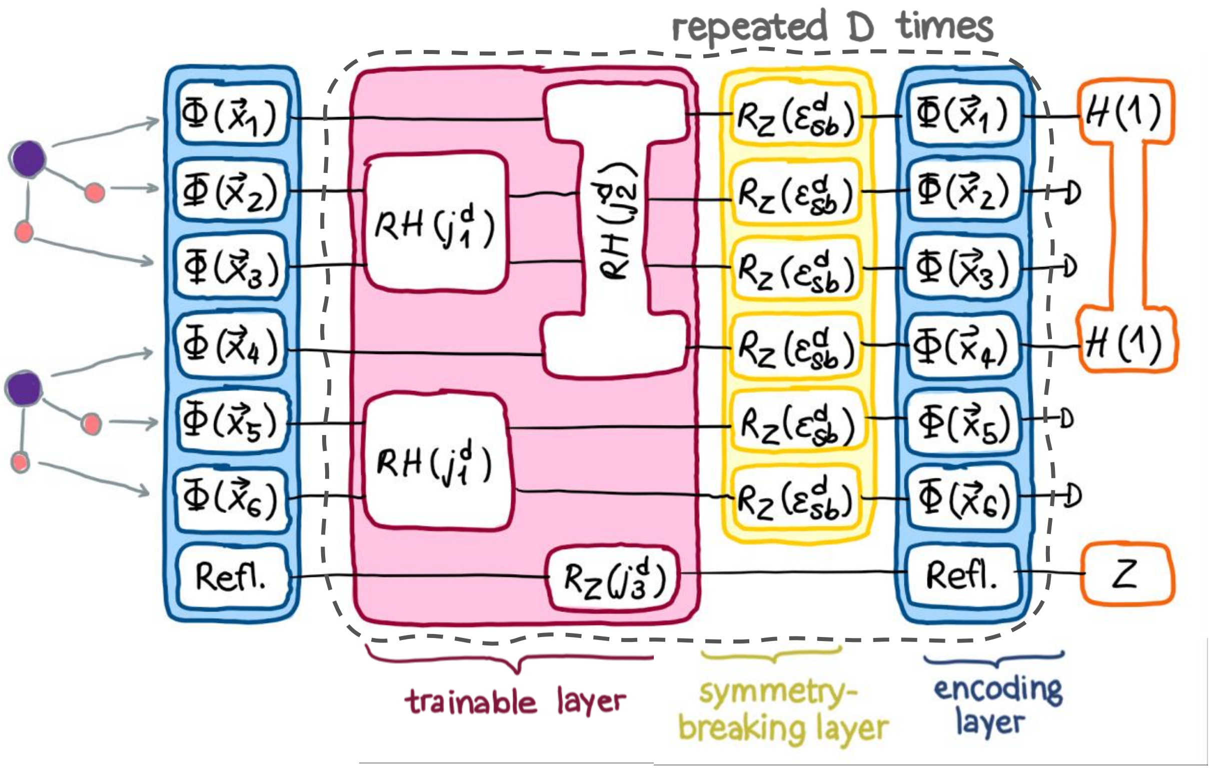
Single molecule (LiH+H2O): the overall model



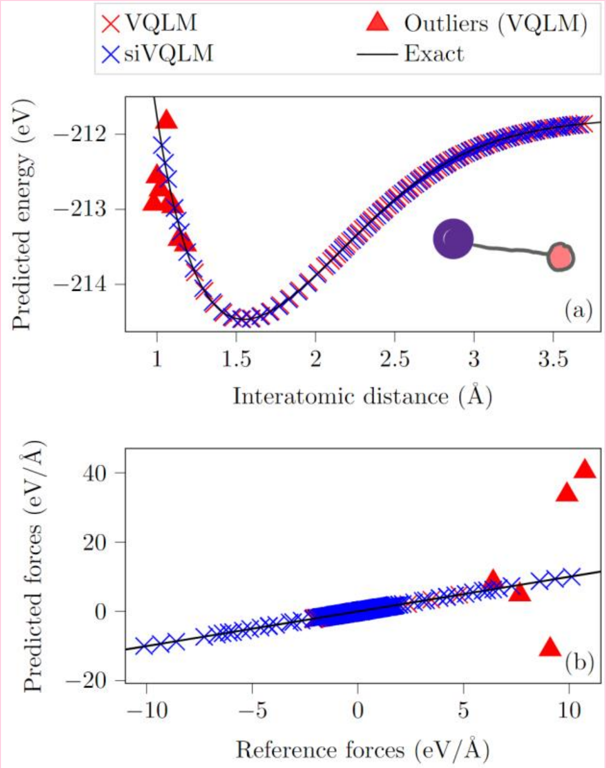


Dimer case (H2O): the overall model





Results



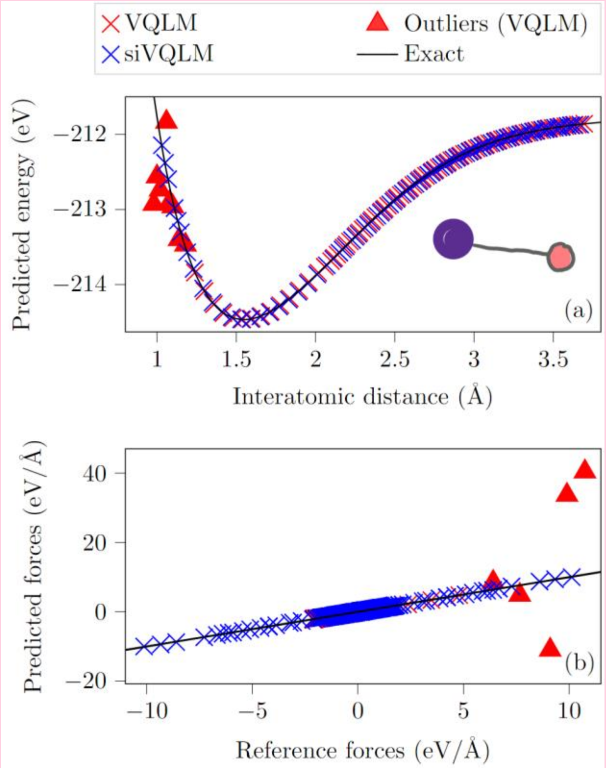
Prediction for LiH trained on exact data

No outliers in higher energy regime

Comparable energy prediction

Improvement in force prediction by one order of magnitude

Results

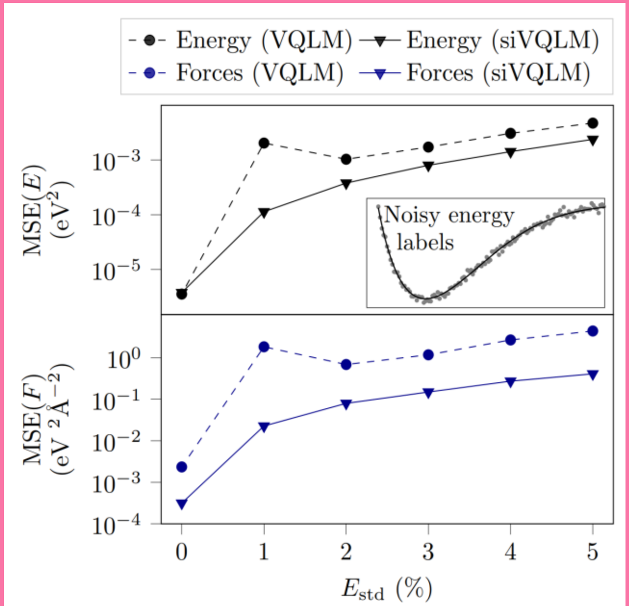


Prediction for LiH trained on exact data

No outliers in higher energy regime

Comparable energy prediction

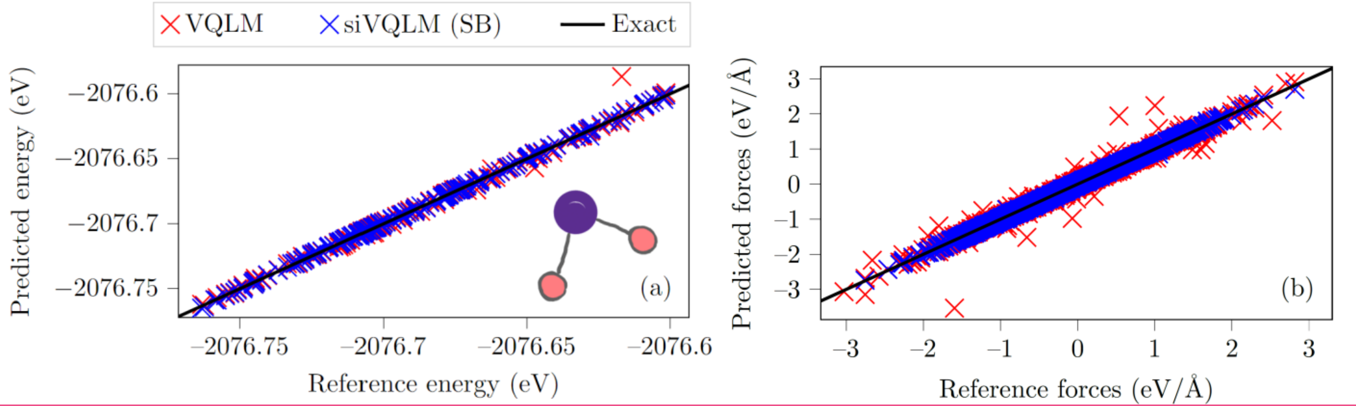
Improvement in force prediction by one order of magnitude



Prediction for LiH trained on noisy data

Less sensitive towards noisy labels

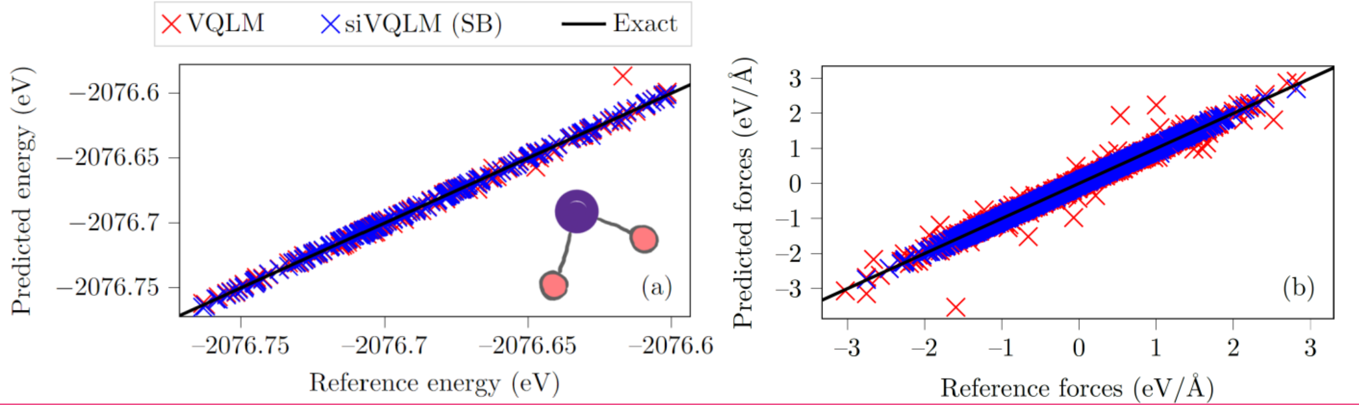
Results



Prediction for H2O

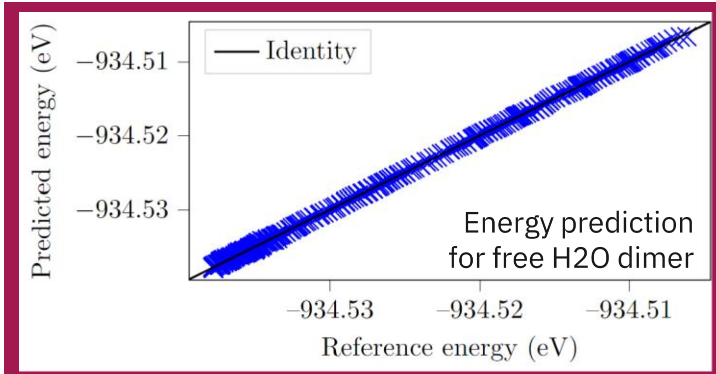
Symmetry-breaking can lead to prediction improvement of one order of magnitude

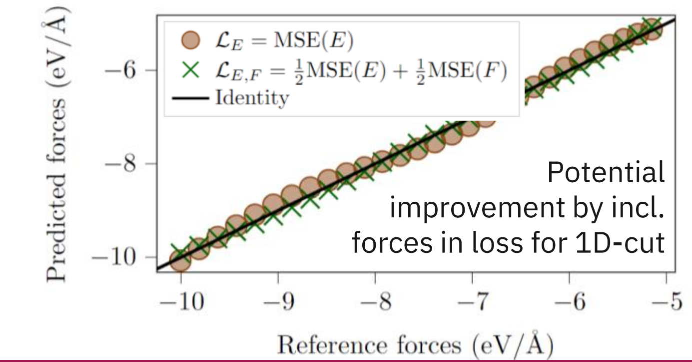
Results



Prediction for H2O

Symmetry-breaking can lead to prediction improvement of one order of magnitude



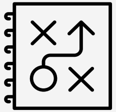


Prediction for H2O dimer

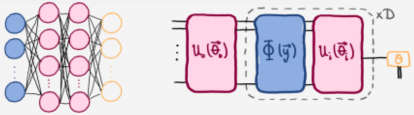
Energy can be learnt, forces need improvement, e.g., by including forces in loss

Summary

Challenge and existing tools

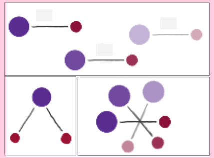


Molecular dynamics



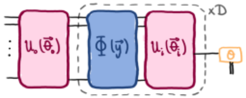
(Q)ML Force Fields

Symmetries and QML



"Raw data" ~~Invariance~~ in

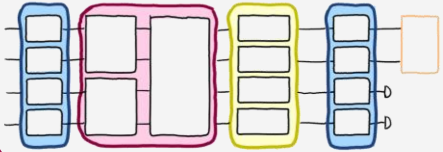
invariance out



Symmetry-invariant QML

Symmetry-invariant Quantum Machine Learning Force Fields

Results



- Examples: LiH, H2O, and H2O dimer
- Better force prediction
- Less noise sensitive
- Symmetry-breaking can improve training

Appendix

Numerical results: LiH

Table I: Numerical results for LiH. For the original VQLM a distinction is made between the full test set (including outliers) and the relevant test set (excluding outliers), while for the siVQLM only the full test set is taken into account.

Lithium hydride	VQLM (full)	VQLM (relevant)	siVQLM
$\mathrm{MSE}(E)_{\mathrm{test}}$ (eV^2)	$2.48 \cdot 10^{-2}$	$1.20 \cdot 10^{-6}$	$1.89 \cdot 10^{-6}$
$\mathrm{MSE}(F)_{\mathrm{test}}$ ($\mathrm{eV}^2\text{Å}^{-2}$)	24.91	$1.23 \cdot 10^{-3}$	$3.14 \cdot 10^{-4}$

Numerical results: H2O

Table II: Numerical results for H_2O of the original VQLM, a non-symmetry-breaking siVQLM (no SB), and a symmetry-breaking siVQLM (SB).

Water	VQLM	siVQLM (no SB)	siVQLM (SB)
$\mathrm{MSE}(E)_{\mathrm{train}}$ (eV^2)	$1.69 \cdot 10^{-6}$	$2.19 \cdot 10^{-6}$	$4.59 \cdot 10^{-7}$
$\mathrm{MSE}(F)_{\mathrm{train}}$ ($\mathrm{eV}^2\mathrm{Å}^{-2}$)	$3.29 \cdot 10^{-3}$	$1.36 \cdot 10^{-3}$	$2.10 \cdot 10^{-4}$
$\mathrm{MSE}(E)_{\mathrm{test}}$ (eV^2)	$3.84 \cdot 10^{-6}$	$3.40 \cdot 10^{-6}$	$4.54 \cdot 10^{-7}$
$\mathrm{MSE}(F)_{\mathrm{test}}$ ($\mathrm{eV}^2\mathrm{Å}^{-2}$)	$5.55 \cdot 10^{-3}$	$1.53 \cdot 10^{-3}$	$2.39 \cdot 10^{-4}$

Numerical results: H2O dimer

Table III: Numerical results for H_2O dimer obtained by the siVQLM on both the training set $\mathcal{A}_{\text{train}}$ and the test set $\mathcal{A}_{\text{test}}$.

Water dimer	$\mathcal{A}_{\text{train}}$	$\mathcal{A}_{\text{test}}$
MSE(E) (eV^2)	$7.26 \cdot 10^{-9}$	$7.45 \cdot 10^{-9}$
MSE(F) (eV^{2}Å^{-2})	$3.28 \cdot 10^{-3}$	$3.28 \cdot 10^{-3}$

Table IV: Numerical prediction results on the test set for the H_2O dimer 1D-cut for excluding ($\mathcal{L}_E$) and including ($\mathcal{L}_{E,F}$) forces in the loss function.

Water dimer 1D-cut	$\mathcal{L}_E$	$\mathcal{L}_{E,F}$
MSE$(E)_{\text{test}}$ (eV^2)	$1.04 \cdot 10^{-6}$	$3.19 \cdot 10^{-7}$
MSE$(F)_{\text{test}}$ (eV^{2}Å^{-2})	$3.79 \cdot 10^{-3}$	$4.58 \cdot 10^{-4}$