

PAPER • OPEN ACCESS

Quantum neural networks force fields generation

To cite this article: Oriel Kiss *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 035004

View the [article online](#) for updates and enhancements.



PAPER

Quantum neural networks force fields generation

OPEN ACCESS

RECEIVED
13 April 2022REVISED
13 June 2022ACCEPTED FOR PUBLICATION
29 June 2022PUBLISHED
15 July 2022

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.

Oriël Kiss^{1,2,3} , Francesco Tacchino^{1,*} , Sofia Vallecorsa³  and Ivano Tavernelli^{1,*} ¹ IBM Quantum, IBM Research—Zürich, 8803 Rüschlikon, Switzerland² Institute for Theoretical Physics, ETH Zürich, 8093 Zürich, Switzerland³ European Organization for Nuclear Research (CERN), 1211 Geneva, Switzerland

* Authors to whom any correspondence should be addressed.

E-mail: fta@zurich.ibm.com and ita@zurich.ibm.com**Keywords:** quantum neural networks, molecular dynamics, force fields**Abstract**

Accurate molecular force fields are of paramount importance for the efficient implementation of molecular dynamics techniques at large scales. In the last decade, machine learning (ML) methods have demonstrated impressive performances in predicting accurate values for energy and forces when trained on finite size ensembles generated with *ab initio* techniques. At the same time, quantum computers have recently started to offer new viable computational paradigms to tackle such problems. On the one hand, quantum algorithms may notably be used to extend the reach of electronic structure calculations. On the other hand, quantum ML is also emerging as an alternative and promising path to quantum advantage. Here we follow this second route and establish a direct connection between classical and quantum solutions for learning neural network (NN) potentials. To this end, we design a quantum NN architecture and apply it successfully to different molecules of growing complexity. The quantum models exhibit larger effective dimension with respect to classical counterparts and can reach competitive performances, thus pointing towards potential quantum advantages in natural science applications via quantum ML.

1. Introduction

Since more than half a century, atomistic simulations represent one of the sharpest tools available for scientific investigation in a wide range of research fields, such as chemistry, materials science and biology [1, 2]. To perform a molecular dynamics (MD) calculation, in which the classical equations of motion are numerically integrated for each atom in the system under study, an accurate knowledge of potential energy surfaces (PES) and local forces is required. This information, originating from the quantum mechanical behaviour of electrons and nuclei, could in principle be deduced from the exact solution of the Schrödinger equation. However, the complexity of such task makes it impractical beyond a few small-scale paradigmatic examples. In a delicate balance between performance and accuracy, approximate solution methods such as density functional theory (DFT) have therefore been proposed, leading to the family of so called *ab initio* MD techniques [3, 4]. These precise yet computationally demanding strategies can be applied up to medium-sized systems, while larger problems may only be tackled, typically at a lower accuracy, with empirical force fields (FFs) [5]. In fact, MD runs require on-the-fly computations of energy and forces at each time step and for each configuration visited by the system during its evolution: therefore, the use of simple parametrized functional potentials (e.g. the FFs) that can be evaluated in a fraction of the time required by actual quantum mechanical calculations is the only viable strategy when thousands of atoms are involved.

Recently, machine learning (ML) has emerged as a new technological paradigm offering promising and effective solutions for physics and chemistry [6, 7]. In the context of MD simulations, a pioneering approach to ML-powered FFs was proposed by Behler and Parrinello using neural networks (NN) [8]. The original idea has later been refined and extended [9–16], also promoting the development of specific software libraries [17, 18]. The fundamental insight behind the so called neural network potentials (NNPs) is two-fold: first, they incorporate the idea that large performance gains can be achieved by directly modelling some form of functional relationship between structure (i.e. atomic positions) and properties of interest (e.g. energies),

essentially bypassing the explicit solution of the underlying quantum mechanical problem. Second, NNPs typically enjoy the generalization capabilities of ML models, maintaining extremely good accuracy even on previously unseen configurations when trained on data sets constructed with *ab initio* methods.

Despite the success of classical ML techniques in the realm of atomic and molecular dynamical processes, the quantum mechanical character of the fundamental laws governing such phenomena immediately leads to the question whether quantum machine learning (QML) methods could provide further significant advantages [6, 19, 20]. The interested reader can refer to [21, 22] for a thorough introduction on quantum computation and QML, respectively.

Thanks to the high practical relevance of the problem, the learning and generation of molecular FFs may actually constitute a very natural and appealing playground in which QML could be tested and compared with state-of-the-art classical counterparts. An important aspect of such comparison lies in the fact that the properties to be learned, namely the relationship between configurations, energies and forces, are generally hard to be derived directly from first principles due to their quantum mechanical origin. At the same time, it has recently been suggested that information theoretical complexity considerations are strongly affected by the availability of training data even when quantum systems are involved [23], with classical ML methods showing competitive performances, e.g. in predicting non-trivial many-body properties [24]. Several important questions may therefore be addressed, from an overall assessment of the capabilities of QML protocols for PES and FFs reconstruction to systematic tests of classical versus quantum techniques for learning specific quantum mechanical properties through data.

In this work, we take a first step in such direction by establishing a direct connection between QML and NNPs. In particular, we demonstrate how quantum neural networks (QNNs) can be employed, in combination with classical data sets, as trainable models for the prediction of energies and forces in molecular systems. Although many different realizations of QNNs and quantum perceptrons are known in the literature [25–33], here we will focus on variational parametrized quantum circuits (PQC) [34–38], which offer the greater flexibility for near term applications. It is worth mentioning that some instances of QNNs, such as the ones that we will implement in the following, are known to exhibit greater power, as measured by the effective dimension in model space, compared to their classical equivalents [39]—a fact that places them among the most promising candidates in the quest for quantum advantage in ML. We also notice that, while variational quantum algorithms are often employed for the direct *ab initio* computation of Hamiltonian spectra [40, 41] and corresponding forces [42], here we take a rather different approach [43], using quantum resources to learn an implicit mapping between atomic coordinates, energy and forces, without any explicit solution of the quantum mechanical problem itself.

2. Model and methods

2.1. Quantum neural networks

We adopt a supervised learning approach with training sets of the form $\mathcal{A} = \{(\vec{C}_\alpha, E_\alpha, \vec{F}_\alpha)\}$, namely collections of n -atom molecular configurations $\vec{C}_\alpha = (x_{a_1}^\alpha, y_{a_1}^\alpha, z_{a_1}^\alpha, \dots, x_{a_n}^\alpha, y_{a_n}^\alpha, z_{a_n}^\alpha)$ with associated total energies E_α and forces $\vec{F}_\alpha = (F_{a_1,x}^\alpha, F_{a_1,y}^\alpha, F_{a_1,z}^\alpha, \dots, F_{a_n,x}^\alpha, F_{a_n,y}^\alpha, F_{a_n,z}^\alpha)$. Here, the index α runs over the different elements of the training set, c_i^α for $c = x, y, z$ is the Cartesian coordinate of the i th atom and:

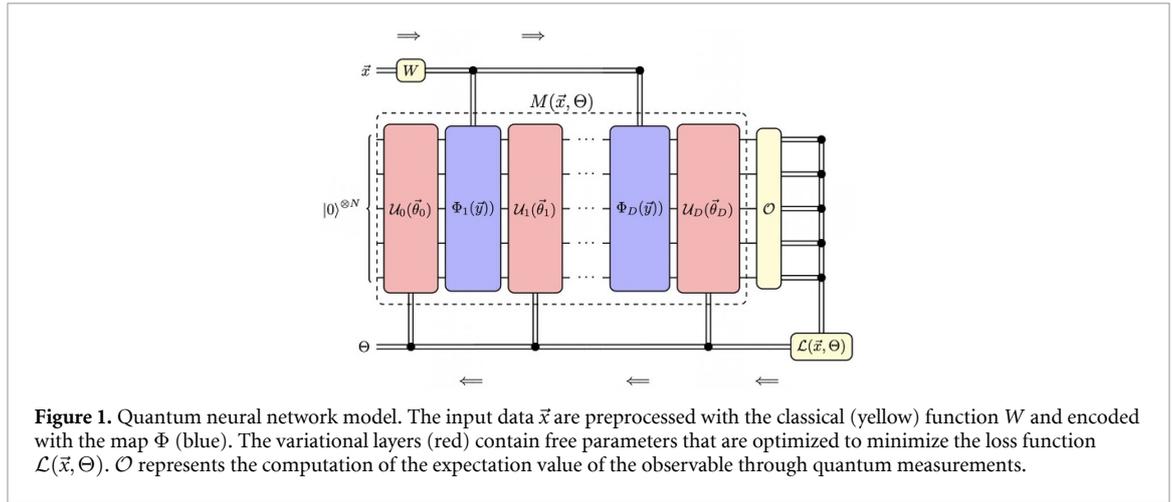
$$F_{a_i,c}^\alpha = -\frac{\partial E_\alpha}{\partial c_{a_i}^\alpha}, \quad (1)$$

is the force acting on atom a_i along the direction c . We will also denote with $|\mathcal{A}|$ the number of samples contained in \mathcal{A} .

The training data, derived from classical *ab initio* methods that will be specified in the following, are used to optimize the predictions made by quantum models, specifically QNNs. These are based on the general notion of PQCs [33, 35, 37], consisting of a reference initial state $|0\rangle$, an output observable \mathcal{O} and a model unitary $M(\vec{x}, \Theta)$, depending on both some input data \vec{x} and a set of trainable parameters $\Theta = \{\theta_0, \dots, \theta_D\}$. The function expressed by a QNN takes the general form:

$$f_\Theta(\vec{x}) = \langle 0|M(\vec{x}, \Theta)^\dagger \mathcal{O}M(\vec{x}, \Theta)|0\rangle, \quad (2)$$

and can be represented schematically as shown in figure 1. For our application of QNNs to PES learning, the Cartesian coordinates \vec{C}_α will be playing the role of the input \vec{x} . In practical realizations, we allow for an additional classical preprocessing map $\vec{y} = W(\vec{x})$, which in our specific context can serve the purpose of converting between Cartesian coordinates and more suitable molecular descriptors, as well as enhancing the effective nonlinearity of the model [34].



It is worth noticing that while the term QNNs is used to emphasize some significant similarities between parametrized quantum models and classical neural networks (NN)—most prominently the layered structure and the use of trainable parameters—some important differences also exist. Indeed, contrary to the usual feed-forward scheme of classical NN, QNNs are quite easily designed and interpreted in the form of re-uploading circuits [44, 45], where trainable layers $\mathcal{U}_\ell(\vec{\theta}_\ell)$, red in figure 1, are alternated with data encoding ones $\Phi_\ell(\vec{y})$, blue in figure 1, with the same classical input values appearing multiple times. The choice of this re-uploading architecture is actually related to a more fundamental difference between classical and quantum NNs, namely the fact that quantum unitary layers act linearly on quantum states, hence lacking an explicit implementation of the usual non-linear activation function as a separate step. Instead, the non-linear manipulation of the classical input data is essentially referred to the choice of the encoding map, the trainable quantum gates and the final measurements.

It has actually been shown that the re-uploading mechanism makes QNNs universal functional approximators [44–46]: more specifically, any QNN output function (equation (2)) can be recast into a truncated Fourier series with a set of available independent frequencies determined by the eigenvalues of the encoding map and growing with the number of re-uploading steps. While recent results suggest that, by expanding the size of the quantum registers, these models can actually be mapped back on simpler sequential ones in which all input operations appear at the beginning [47], one can still profit from the insights offered by the re-uploading picture as a guide for intuition in the actual design of application-specific QNNs. As an example, theory suggests that, by adjusting the number of input layers, the richness of the Fourier spectrum can be systematically increased.

The structure of QNNs is completed by choosing a physical observable \mathcal{O} and a suitable loss function \mathcal{L} , whose minimization drives the update of the trainable parameters via a classical optimization routine. In the following, we will make the simple choice:

$$\mathcal{O} = \sigma_z^1, \quad (3)$$

namely we will take the expectation value of the Pauli-z operator on the first qubit to construct the network output. Moreover, we will use a quadratic mean square error (MSE) loss function:

$$\mathcal{L}_\chi(\mathcal{A}, \Theta) = \text{MSE}(\text{Energy}, \Theta) + \chi \cdot \text{MSE}(\text{Forces}), \quad (4)$$

with the hyperparameter χ weighting the contribution of energy and forces [48]. In practice, we directly associate the output of the QNN with the energy potential surface by defining:

$$\text{MSE}(\text{Energy}) = \frac{1}{|\mathcal{A}|} \sum_{\alpha \in \mathcal{A}} \left(f_\Theta(\vec{C}_\alpha) - E_\alpha \right)^2. \quad (5)$$

In parallel, consistent predictions for the forces are obtained by taking the derivative of the quantum circuit output with respect to the molecular coordinates:

$$\text{MSE}(\text{Forces}) = \frac{1}{3n \cdot |\mathcal{A}|} \sum_{\alpha \in \mathcal{A}} \left\| \nabla_c f_\Theta(\vec{C}_\alpha) + \vec{F}_\alpha \right\|^2. \quad (6)$$

We leave for future works the investigation of alternative strategies, such as the use of two independent quantum circuits for the separate learning of energies and forces.

2.2. Encoding layers

As mentioned in the previous section, the encoding operations $\Phi_\ell(\vec{y})$ are used to input the molecular configurations in the QNN model. Here, we assume for simplicity that these encoding unitaries are identical across different layers or re-uploading stages, i.e. $\Phi_\ell(\vec{y}) \equiv \Phi(\vec{y})$. Moreover, let us denote by N the linear dimension of the feature vector \vec{y} , where in general N can differ from $3n$ due to the preprocessing stage. Following standard practices [49], we will use N qubits to encode and manipulate N -dimensional features $\vec{y} = (y_1, \dots, y_j, \dots, y_N)$.

The quantum feature map $\Phi(\vec{y})$ is then constructed according to the expression:

$$\Phi(\vec{y}) = \mathcal{E}(\vec{y})\mathcal{S}(\vec{y}), \quad (7)$$

where $\mathcal{S}(\vec{y})$ is a collection of single-qubit Pauli- y rotations:

$$\mathcal{S}(\vec{y}) = \prod_{j=1}^N \exp\left(-i\frac{\sigma_y^{(j)}}{2}y_j\right), \quad (8)$$

and $\mathcal{E}(\vec{y})$ is an entangling operation of the form:

$$\mathcal{E}(\vec{y}) = \prod_{(j,k) \in \mathcal{P}} \exp\left(-i(1/2)\sigma_z^{(j)}\sigma_z^{(k)}y_jy_k\right). \quad (9)$$

The latter directly resembles the so called ZZ feature map originally introduced in [49]. The set of qubit pairs \mathcal{P} can be chosen in different ways, balancing ease of implementation on physical architectures with functional expressivity. Standard examples include *linear* entanglement:

$$\mathcal{P}_{\text{linear}} = \{(j, j+1) | j = 1, \dots, N-1\}, \quad (10)$$

circular entanglement:

$$\mathcal{P}_{\text{circ}} = \{(j, j+1 \bmod N) | j = 1, \dots, N\}, \quad (11)$$

and *full* entanglement:

$$\mathcal{P}_{\text{full}} = \{(j, k) | j = 1, \dots, N-1, k > j\}. \quad (12)$$

For further generality, we also define a natural extension of the ZZ feature map to degree l interactions, adding factors of the form:

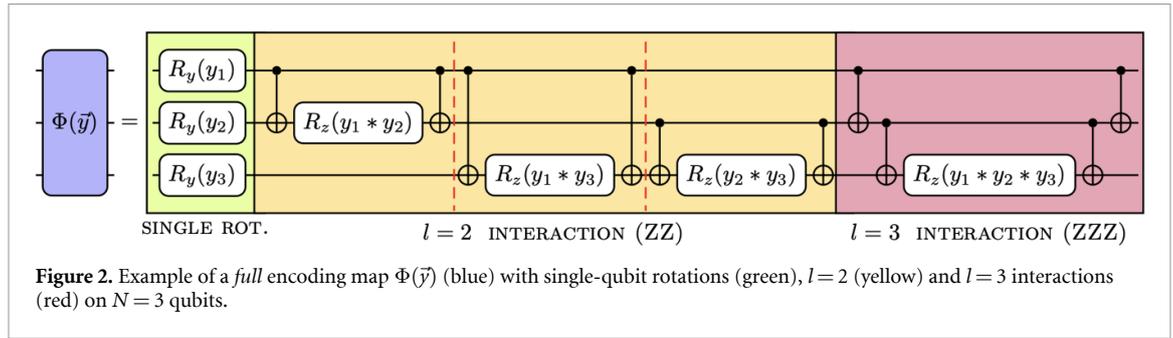
$$\exp\left(-i(1/2)\prod_{k=1}^l y_{j_k}\sigma_z^{(j_k)}\right), \quad (13)$$

where, in principle, the l qubits (j_1, \dots, j_l) can be arbitrarily chosen among the N available.

In figure 2 we explicitly show a 3-qubit example with a *full* 2-qubit entangling map and an additional $l=3$ operation. All multiple qubit operations are already decomposed into a standard universal set made of single-qubit rotations and CNOTs [50], which is typical of superconducting quantum computing architectures.

Before moving to the description of the trainable part of the QNN models, let us also remark a few points about the classical preprocessing step W (see figure 1). As suggested above, this classical manipulation is generally used to enhance the nonlinear behaviour of the network, for example by taking inverse trigonometric functions of the original inputs [34]. At the same time, we can use this initial step to embed the relevant set of physical symmetries into the abstract representation of the target molecular systems seen by the QNN. This is known to be crucial already for classical ML methods, where the role of symmetry preserving features is played, e.g. by the so called symmetry functions [8, 10].

To limit the complexity of the quantum models, in the following we will use a set of internal coordinates, namely bond distances and angles, which by design respect translational and rotational symmetries. The integration of more advanced techniques, including fragmentation of large systems into local atomic environments, the use of other classes of molecular fingerprints and, possibly, the realization of symmetry adapted quantum circuits all represent natural future extensions of the present work.



We explicitly notice that the conversion between Cartesian and internal coordinates ($\vec{I}_\alpha = W(\vec{C}_\alpha)$) must be taken into account when computing the quantum forces predictions, as these are defined with respect to the former class (equation (1)). As a result, equations (5) and (6) are more properly rewritten as:

$$\text{MSE(energy)} = \frac{1}{|\mathcal{A}|} \sum_{\alpha \in \mathcal{A}} \left(f_\Theta \left(W(\vec{C}_\alpha) \right) - E_\alpha \right)^2, \tag{14}$$

and

$$\text{MSE(Forces)} = \frac{1}{3n \cdot |\mathcal{A}|} \sum_{\alpha \in \mathcal{A}} \sum_{c, a_i} \left(\nabla_I f_\Theta(\vec{I}_\alpha) \cdot \frac{\partial W}{\partial c_{a_i}^\alpha} + F_{a_i, c}^\alpha \right)^2. \tag{15}$$

2.3. Trainable layers

The parametrized operations $\mathcal{U}_\ell(\vec{\theta}_\ell)$ represent the adjustable components of the QNN model. If suitably optimized during the learning phase, these select the most appropriate mapping from inputs (molecular coordinates) to outputs (energy and forces).

The precise structure of the $\mathcal{U}_\ell(\vec{\theta}_\ell)$ can vary significantly across different models and applications, with several popular choices known in the literature. For our specific functional regression problem, we follow the steps of Mitarai *et al* [34] and make use of a physics-inspired ansatz which is known to generate highly entangled states. In particular, we consider the following fully connected transverse field Ising Hamiltonian:

$$H = \sum_{j=1}^N a_j \sigma_y^{(j)} + \sum_{j=1}^N \sum_{k=j}^N J_{jk} \sigma_z^{(j)} \sigma_z^{(k)}, \tag{16}$$

and we use the induced time evolution operator $U(t) = e^{-iHt}$ as a template for the trainable layers. By using the approximate product formula:

$$e^{-iHt} \approx \prod_{j=1}^N e^{-ita_j \sigma_y^{(j)}} \cdot \prod_{j=1}^N \prod_{k=j}^N e^{-it J_{jk} \sigma_z^{(j)} \sigma_z^{(k)}}, \tag{17}$$

and by replacing $a_j t$ and $J_{jk} t$ with free parameters, we obtain a unitary operation of the form:

$$\mathcal{U}_\ell(\vec{\theta}_\ell) = \prod_{j=1}^N e^{-i\frac{\theta_\ell^j}{2} \sigma_y^{(j)}} \cdot \prod_{j=1}^N \prod_{k=j}^N e^{-i\frac{\theta_\ell^{jk}}{2} \sigma_z^{(j)} \sigma_z^{(k)}}, \tag{18}$$

where now $\vec{\theta}_\ell$ is the collection of all θ_ℓ^j and θ_ℓ^{jk} . It is easy to see that $\mathcal{U}_\ell(\vec{\theta}_\ell)$ can be implemented with single-qubit Pauli- y rotations and two-qubit ZZ operations, similarly to what happens for the encoding map Φ defined in the previous section. We stress however that, while the encoding layers are identical across different reuploading stages, as they repeatedly input the same classical data \vec{y} , the trainable parameters $\vec{\theta}_\ell$ are allowed to vary across different layers $\ell = 0, \dots, D$.

For $\ell = 0$, we make the special choice $\theta_\ell^{jk} = 0, \forall j, k$, namely we only use single-qubit rotations at the beginning of the computation (see figure 1). Moreover, we empirically find that including generalized l -qubit interactions up to $l = 3$, i.e. terms of the form $e^{-i(1/2)\theta_{jkm} \sigma_z^{(j)} \sigma_z^{(k)} \sigma_z^{(m)}}$ improves the overall performances of the

model at the cost of only a modest increase of circuit complexity. The resulting $M(\vec{x}, \Theta)$, as seen in figure 1, can thus be written as:

$$M(\vec{x}, \Theta) = \left[\prod_{\ell=D}^1 \mathcal{U}_\ell(\vec{\theta}_\ell) \cdot \Phi(W(\vec{x})) \right] \cdot \mathcal{U}_0(\vec{\theta}_0). \quad (19)$$

Notice that the product runs from $\ell = D$ to $\ell = 1$ from left to right.

2.4. Model training

We train QNN models by minimizing an average quadratic error that in principle contains both energy and forces labels, see equation (4). In most instances, we make use of an update rule which follows the negative gradient direction of the loss function:

$$\Theta^{t+1} = \Theta^t - \eta \nabla \mathcal{L}_\chi(\mathcal{A}, \Theta^t), \quad (20)$$

where η is the learning rate and Θ^t the set of trainable parameters at the optimization step t . For energy predictions derived from a quantum circuit, derivatives with respect to any given trainable parameter μ can be easily computed with the parameter shift rule [51]:

$$\frac{\partial E_\alpha^{\text{QNN}}}{\partial \mu} = \frac{1}{2} \left[f_{\mu+\frac{\pi}{2}; \Theta_\mu}(W(\vec{C}_\alpha)) - f_{\mu-\frac{\pi}{2}; \Theta_\mu}(W(\vec{C}_\alpha)) \right], \quad (21)$$

where $E_\alpha^{\text{QNN}} = f_\Theta(W(\vec{C}_\alpha))$ and Θ_μ is the set of all trainable parameters without μ . The corresponding result for the forces predictions is obtained with the iterative parameter shift rule [52], and schematically reads:

$$\begin{aligned} \frac{\partial F_{a_i, c}^{\alpha, \text{QNN}}}{\partial \mu} &= -\frac{\partial^2 E_\alpha^{\text{QNN}}}{\partial c_{a_i} \partial \mu} = -\frac{1}{4} \sum_j \left[f_{\mu+\frac{\pi}{2}; \Theta_\mu} \left(\vec{I}_\alpha + \frac{\pi}{2} \vec{e}_j \right) \right. \\ &\quad \left. - f_{\mu-\frac{\pi}{2}; \Theta_\mu} \left(\vec{I}_\alpha + \frac{\pi}{2} \vec{e}_j \right) - f_{\mu+\frac{\pi}{2}; \Theta_\mu} \left(\vec{I}_\alpha - \frac{\pi}{2} \vec{e}_j \right) \right. \\ &\quad \left. + f_{\mu-\frac{\pi}{2}; \Theta_\mu} \left(\vec{I}_\alpha - \frac{\pi}{2} \vec{e}_j \right) \right] \cdot \frac{\partial I_j^\alpha}{\partial c_{a_i}}, \end{aligned} \quad (22)$$

with $(\vec{e}_j)_k = \delta_{jk}$ the standard basis vectors.

Notice that the parameter shift rule of equation (21) can also be used to retrieve the actual QNN forces predictions, namely:

$$F_{a_i, c}^{\alpha, \text{QNN}} = -\nabla_I f_\Theta(\vec{I}_\alpha) \cdot \frac{\partial W}{\partial c_{a_i}} = -\sum_j \frac{\partial f_\Theta(\vec{I}_\alpha)}{\partial I_j^\alpha} \frac{\partial I_j^\alpha}{\partial c_{a_i}}. \quad (23)$$

In this case, factors of the form $\partial f / \partial I_j$ must be computed from the quantum circuit, while the chain rule factors taking into account the classical preprocessing are known analytically and are determined solely by the mapping function W .

At the beginning of the training, all free parameters are initialized at zero, so that each hidden layer acts as the identity operator. This essentially follows the recommendations given in [53] to promote effective optimization steps in the early training phase.

2.5. Effective dimension

In the remaining part of this work, we will provide a series of examples to demonstrate how QNN models are able to learn and give consistent predictions of PES and FFs for individual molecules. The overall performances will be assessed primarily through the evaluation of the average root MSE on suitable test sets.

In addition to that, we will also make use of the concept of effective model dimension—originally introduced in [39]—to inform the comparison with classical counterparts and to investigate potential advantages. As a brief summary, the effective dimension quantifies the capacity of both classical and quantum parametrized ML models [54] by measuring the portion of model space that they occupy. In other words, it estimates the capability of a model in covering the functional space defined by a particular model class by making a productive use of all its parameters, going beyond naive parameter-counting arguments. A high effective dimension is therefore related to a richer set of expressible functions and better trainability, as it can lead to a more favourable landscape for gradient based methods [39]. Moreover, the effective dimension can sensibly bound the model generalization error [39, 54].

For a given statistical model $y = f_{\Theta}(x)$, the effective dimension is defined as:

$$d_n(f_{\Theta}) = \frac{2 \log \left(\frac{1}{V_{\Omega}} \int_{\Omega} \sqrt{\det \left(\mathbb{1} + \frac{n}{2\pi \log n} F(\Theta) \right)} d\Theta \right)}{\log \left(\frac{n}{2\pi \log n} \right)}, \quad (24)$$

where $\Omega \subset \mathbb{R}^d$ is a d -dimensional parameter space of volume V_{Ω} , $n \in \mathbb{N}$ is the number of data samples. Here, we have also introduced the Fisher information matrix:

$$F = \mathbb{E} \left[\nabla_{\Theta} \log(p(x, y; \Theta)) \nabla_{\Theta} \log(p(x, y; \Theta))^T \right] \quad (25)$$

$$\approx \frac{1}{K} \sum_{k=1}^K \nabla_{\Theta} f_{\Theta}(x_k) \nabla_{\Theta} f_{\Theta}(x_k)^T, \quad (26)$$

with $p(x, y; \Theta) = p(y|x; \Theta)p(x)$ and $p(y|x; \Theta) = \exp(-\|y - f_{\Theta}(x)\|^2/2)/\sqrt{2\pi}$ being the probability distribution mass of the model. The effective dimension is bounded by the rank of the Fisher information matrix and is usually normalized with the total number of parameters d .

3. Results

3.1. Diatomic molecules

3.1.1. Lithium hydride

As a first proof-of-concept, we consider a single LiH molecule and we design a QNN model that learns its dissociation curve, and the corresponding FE, as a function of the bond length r .

The presence of a single internal coordinate, which is obtained from the Cartesian positions of the Li and H atoms with a mapping $W(\vec{C}) = |\vec{C}_{\text{Li}} - \vec{C}_{\text{H}}|$, makes the problem effectively one dimensional. We construct a classical dataset by numerically diagonalizing the second quantized Hamiltonian expressed in the STO3G basis set [41] for different bond lengths r in the range $[0.9, 4.5]$ Å. Forces are computed via finite differences over the exact PES.

To make better use of the Fourier series structure of the QNN, we make the exact PES symmetric by mirroring it around the $r = 4.5$ Å and selecting the data set over the extended range, see appendix A. Furthermore, we use an additional preprocessing step—on top of the Cartesian to internal coordinate transformation—to construct 3-dimensional features from the original 1D problem. First, we scale all inputs in $[-1, 1]$ with the `scikit-learn` `MinMaxScaler`, then we apply the map:

$$W: [-1, 1] \rightarrow [-\pi, \pi]^3$$

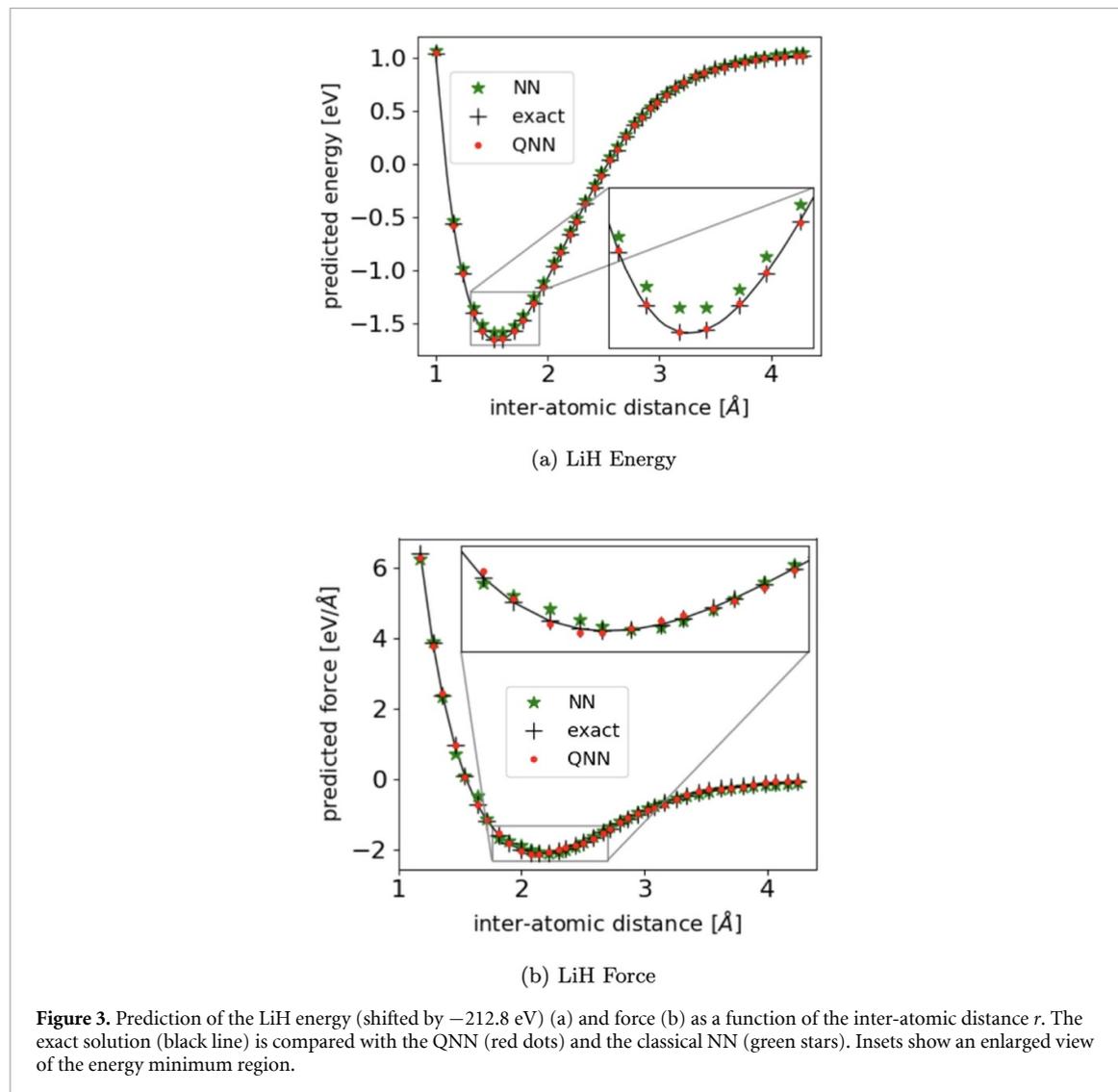
$$r \mapsto \begin{pmatrix} \pi r \\ \arcsin(r) \\ \arccos(r) \end{pmatrix}. \quad (27)$$

We employ a 3-qubit QNN model with $D = 10$ trainable layers, alternated with the same amount of input stages. Notice that, even if a single qubit could in principle be used to encode the 1-dimensional problem under study, we instead choose to engineer 3-dimensional feature vectors from the single independent internal coordinate available (interatomic distance) to enrich by design the set of expressible functions and to enable the use of genuine multi-qubit quantum resources such as entanglement. More in general, a very common choice when using a parametric type of encoding is to use one qubit per feature. However, similarly to the depth of the model and the overall number of trainable parameters, this is by no means a strict requirement and could be freely modified, subject to appropriate convergence tests, to improve performances or in response to additional constraints (e.g. limited number of qubits or finite quantum coherence). We use a *full* feature map, as introduced in section 2.2, with $l = 3$ (see also figure 2), and similar trainable layers with full entanglement and degree $l = 3$ interactions. The input of the feature map is $\vec{y} = W(r)$, with W as in equation (27) and r the Li-H inter-atomic distance.

We benchmark our model against a fully connected classical NN, which is given access to the same set of seven engineered inputs and products thereof shown in figure 2. The classical NN is composed of five layers with $[7, 4, 5, 2, 1]$ units using the hyperbolic tangent activation function. Such network topology is chosen through a random search over all possible configurations with the same number of parameters (identical to the QNN), selecting the one that achieves the best validation loss.

Table 1. Validation RMSE for LiH energy (eV) and forces (eV Å⁻¹), together with the normalized effective dimension (d_{50}/d), the total number of parameters (d) and the number of training epochs for both the QNN and the classical NN models.

LiH	RMSE (E)	RMSE (F)	d_{50}/d	d	Epochs
QNN	0.003	0.044	0.9	73	230
NN	0.034	0.21	0.38	73	1600



Both the classical and the QNN models are trained with 4000 steps of the ADAM [55] algorithm on 50 data points using the \mathcal{L}_0 ($\chi = 0$) loss. We remark that the small size of the classical NN makes it quite sensitive to parameters initialization, even when using the Glorot and Bengio [56] scheme. Hence, the model underfits in 65% of the cases and needs 1600 epochs to reach convergence. On the other hand, the QNN appears to be robust against weights initialization and converges in 230 epochs.

The validation root mean square errors (RMSE) for both models, computed on a test set with 120 data points, are given in table 1 and the respective predictions are shown in figure 3. We notice that, in this setting, the QNN outperforms the best classical counterpart with the same number of parameters in terms of stability and quality of the predictions, and also exhibits a larger effective dimension. Although the classical model can match the QNN results in absolute terms when its size grows or if the forces are explicitly included in the loss function ($\chi \neq 0$), the present comparison, supported by the effective dimension analysis, certifies the competitiveness of the proposed quantum models. In appendix B we also report an example of MD trajectories driven by the exact and learned FFs.

Table 2. Validation RMSE for Ar₂ energy (eV) and forces (eV Å⁻¹), together with the normalized effective dimension (d_{165}/d), the total number of parameters (d) for both the QNN and the classical NN models.

Ar ₂	RMSE (E)	RMSE (F)	d_{165}/d	d
QNN	2.6×10^{-5}	0.058	0.77	66
NN	1×10^{-4}	0.6	0.11	66

3.1.2. Argon dimer

As a second experiment, we consider two Argon atoms separated by a bond length r . This example covers a richer set of intermolecular interactions, as Ar₂ is loosely bounded via Van der Waals forces. The dataset is constructed using density functional theory (DFT/B3LYP with Grimme D3 corrections as implemented in Gaussian [57]), and smoothened by fitting to a $C_1/r^{12} - C_2/r^6$ function.

The training strategy is almost identical to the LiH case, including the QNN architecture (with $D = 9$) and the dataset mirroring. One notable difference is the use of the COBYLA optimizer, which empirically showed better performance than the ADAM one. The benchmark is performed against a fully connected classical NN, taking as input the bond length, composed of ten layers with [1, 7, 4, 3, 2, 5, 3, 4, 2, 1] hidden units. The architecture is chosen with a random search through the models with the same number of parameters as the QNN and variable number of inputs, selecting the one that achieves the best validation loss.

The validation RMSE for both models, computed on a test set with 120 data points, are given in table 2 and the respective predictions are shown in figure 4. As for the LiH case, we observe that the QNN outperforms the best classical counterpart with the same number of parameters in terms of stability and quality of the predictions, and also exhibits a larger effective dimension. Notably, the region around the PES minimum is better reconstructed with the QNN than the classical NN.

3.2. A single H₂O molecule

We move towards the multi-dimensional case by considering an individual H₂O molecule. For this more challenging test, we choose 3 internal coordinates, namely the two O–H bond lengths and the H–H planar angle, which we scale again in $[-1, 1]$ and preprocess to create the following 3-dimensional feature vector:

$$\vec{y} = \begin{pmatrix} \arcsin(r_{\text{OH},1}) \\ \arcsin(r_{\text{OH},2}) \\ \arcsin(\phi_{\text{HH}}) \end{pmatrix}, \quad (28)$$

which is then inputted in the model via the feature map. The classical configurations and the corresponding energy and forces labels, computed with DFT, are retrieved from [58]. To simplify the problem, we concentrate on configurations around the equilibrium position, discarding those with bond length outside $[1.6, 2.1]$ Å.

Our QNN model is constructed similarly to the LiH case, with three qubits and a depth of $D = 12$ encoding and trainable layers. We compare again its results with those of a classical NN model with the same number of parameters and designed with the same conditions applied in the LiH case presented above. Furthermore, we also use as a reference the state-of-the-art specialized n2p2 [59] classical package, which makes full use of symmetry functions and of the sub-network fragmentation idea [8], and from which we expect peak performance.

More specifically, the simplified classical NN model is a fully-connected 6-layer network with [7, 4, 6, 2, 2, 1] units and hyperbolic tangent activation function. Instead, the n2p2 network uses three sub-networks with two hidden layers, each with 15 units, and hyperbolic tangent activation. This model takes as input a set of 15 symmetry functions for the Oxygen atom and 20 for the two Hydrogen ones. The models are trained by minimizing a \mathcal{L}_1 loss function on 300 data points (1000 for n2p2). Notice that here we set $\chi = 1$: indeed, as pointed out in [60], it is important to incorporate the forces in the training if we wish to predict them accurately. However, this makes a full gradient descent computationally very intensive for the simulation of the quantum model, as it requires the calculation of the circuit Hessian matrix with recursive parameter shifts (see equation (22)). For this reason, we choose the gradient free optimizer COBYLA for the present demonstration, while the classical models are trained with the ADAM method.

The validation loss results (computed on a test set with 650 data points) are presented in table 3, while figure 5 shows the predicted energy and forces against the reference data points. The QNN is once again competitive with respect to classical counterparts, outperforming the non-specialized classical NN and reporting the largest normalized effective dimension.

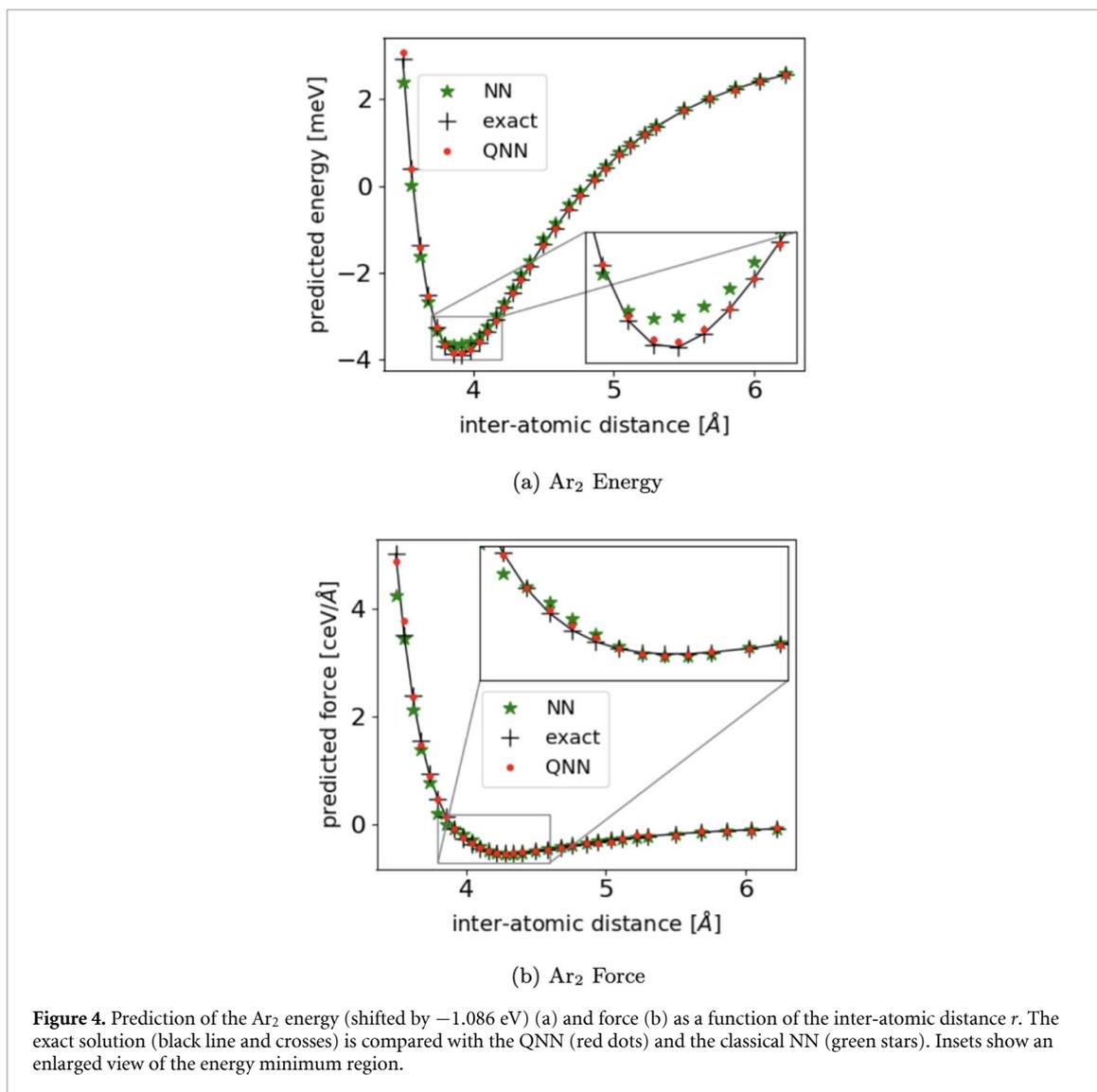


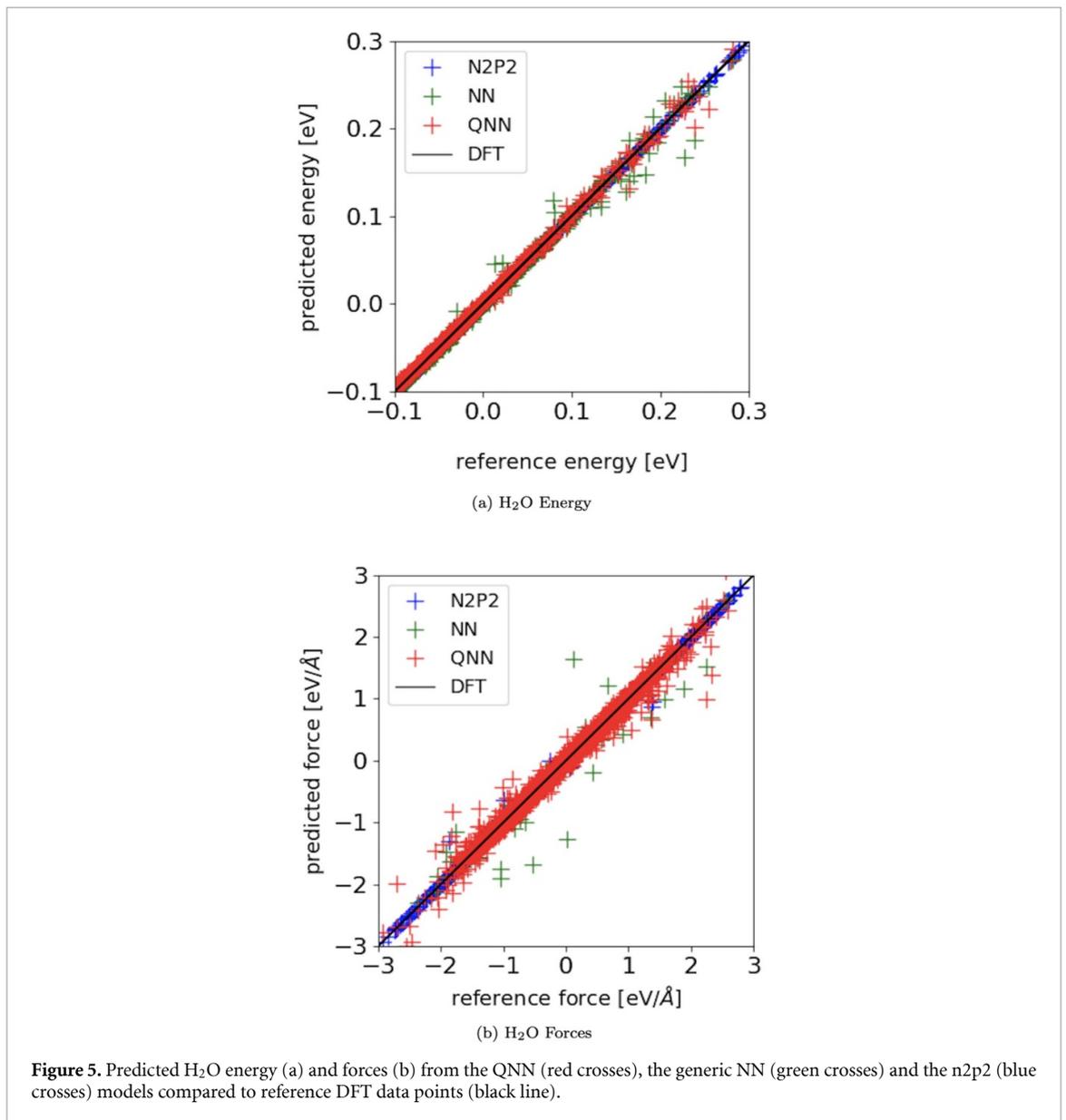
Table 3. Validation RMSE for H₂O energy (eV) and forces (eV \AA^{-1}), together with the normalized effective dimension (d_{300}/d) and the total number of parameters (d) for the QNN and the classical NN and n2p2 models.

H ₂ O	RMSE (E)	RMSE (F)	d_{300}/d	d
QNN	0.005	0.06	0.72	87
NN	0.006	0.1	0.25	87
n2p2	7×10^{-4}	0.01	0.04	1642

3.3. Umbrella motion of hydronium

In our last test, we consider a single hydronium (H_3O^+) molecule. Following [61], we prepare a training set by sampling configurations traversed along the inversion pathway shown in figure 6 using DFT-based MD simulations at 400 K and collecting the corresponding energies and forces. In order to sample the full profile along the dihedral angle ‘HHHO’ from -0.78 to 0.78 rad, we also applied a dynamical constraint with increments of 0.005 rad at each MD time step. All calculations were performed with the plane-wave (PW) code CPMD [62] using unrestricted Kohn–Sham DFT with the PBE functional [63], a PW cutoff of 70 Ry, a cubic simulation box of edge 14 \AA , and Trouiller–Martins pseudo-potentials [64]. For the MD, a time step 10 a.u. (0.242 fs) was used.

We describe each configuration with 6 internal degrees of freedom, namely three O–H bond lengths, two O–H angles and the dihedral angle formed by the four atoms (see appendix C for the formal definitions). The internal coordinates are preprocessed with the function $W(\cdot) = \arcsin(\cdot)$ and loaded in the model via



the feature map, see appendix D. We also make use of the explicit formulas for derivatives with respect to Cartesian coordinates provided in [65].

A 6-qubit QNN model is constructed with $D = 10$ repetitions of the encoding map and the trainable layers, both with *linear* entanglement and order $l = 3$ interactions. The latter are also placed with linear couplings across all neighbouring 3-qubit groups. Due to the complexity of the system and the high computational cost, we only include energy labels in the training set, leaving a more refined study of the forces for future investigations. Hence, we train the model by minimizing a \mathcal{L}_0 loss on 500 data points (9000 for n2p2) with 5000 steps of the ADAM algorithm.

As in the previous examples, we compare the QNN model with a classical NN, whose topology is optimized under the constraint that the number of trainable parameters be the same of the QNN, and with a n2p2-build model. In this case, the classical NN has [6, 14, 2, 1] units, while the n2p2 NN is composed of four sub-networks, each one with a structure similar to the case if the H₂O molecule in section 3.2.

The RMSE results for a test set with 500 data points are reported in table 4, and energy predictions are also depicted in figure 6. Here, the QNN model is still competitive with respect to the generic classical NN, although both of them are clearly outperformed by the specialized and much larger n2p2 model. It is also worth noticing that forces predictions are much worse than energy ones, which confirms the necessity of including them in the loss function to achieve good results in non-trivial systems.

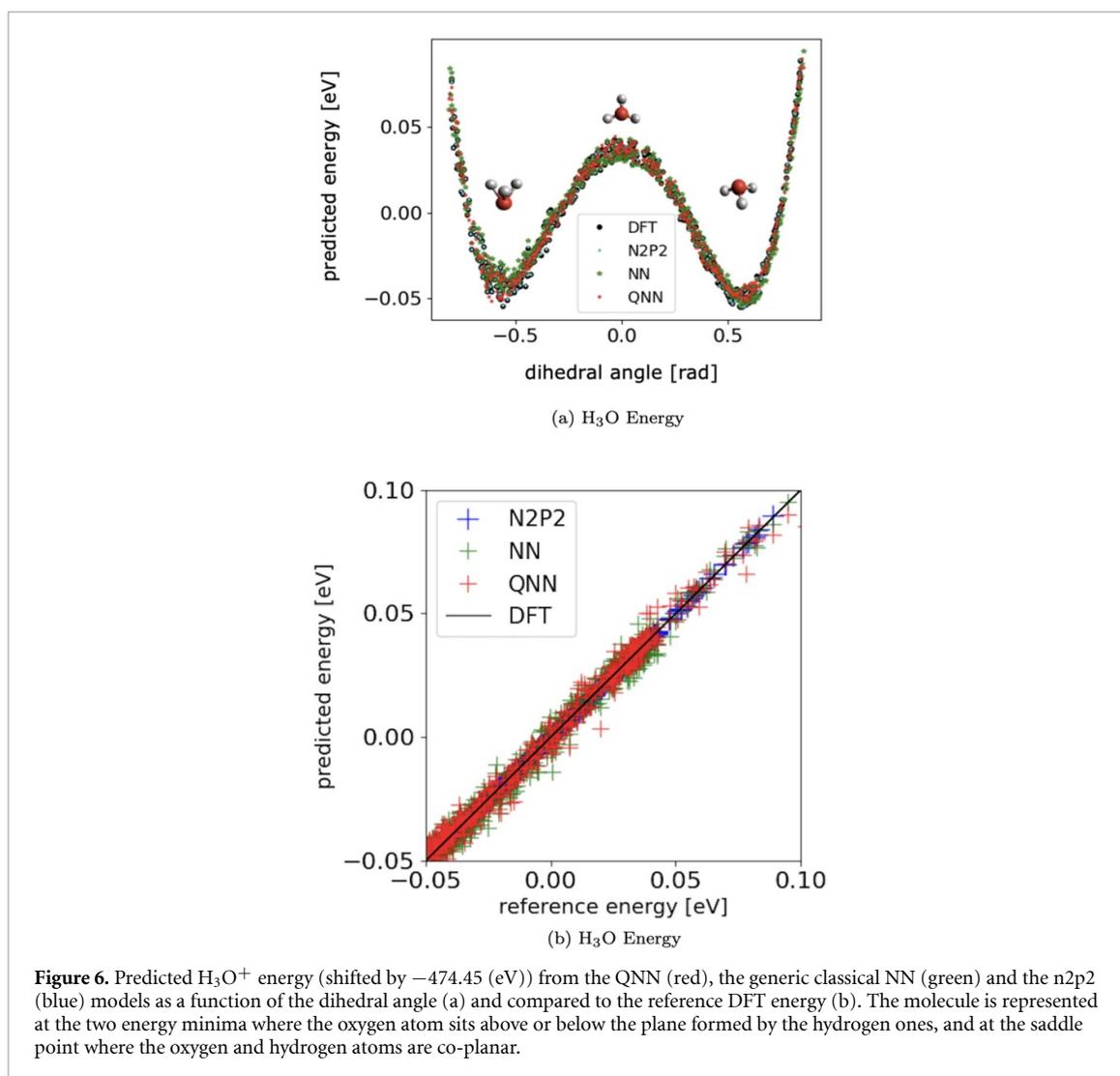


Figure 6. Predicted H₃O⁺ energy (shifted by -474.45 (eV)) from the QNN (red), the generic classical NN (green) and the n2p2 (blue) models as a function of the dihedral angle (a) and compared to the reference DFT energy (b). The molecule is represented at the two energy minima where the oxygen atom sits above or below the plane formed by the hydrogen ones, and at the saddle point where the oxygen and hydrogen atoms are co-planar.

Table 4. Validation RMSE for H₃O⁺ energy (eV) and forces (eV \AA^{-1}), together with the normalized effective dimension (d_{500}/d) and the total number of parameters (d) for the QNN and the classical NN and n2p2 models.

H ₃ O ⁺	RMSE (E)	RMSE (F)	d_{500}/d	d
QNN	3.7×10^{-3}	0.26	0.67	135
NN	4.2×10^{-3}	0.207	0.19	135
n2p2	5×10^{-4}	0.19	0.03	2214

4. Conclusions

In this work we have successfully demonstrated the systematic application of QML techniques, and specifically parametrized QNN, to the problem of learning molecular PES from classical *ab initio* data sets and for the generation of molecular FFs. In all our numerical simulations, the proposed QNN models already achieved competitive performances with respect to comparable classical ones, reporting good prediction accuracy for several paradigmatic single-molecule examples.

The present assessment naturally opens several questions and future research directions. On the one hand, the design of more specialized QNN architectures and molecular descriptors would allow a more refined and effective treatment of the problem, including the possibility of tackling bulk materials. For example, classical state-of-the-art implementations [59] crucially benefit from the fragmentation of large systems in local environments and corresponding sub-networks [8]. Similar techniques could possibly be engineered for QNNs, e.g. by resorting to more general perceptron-based models [30, 31]. These could explicitly be partitioned into local components and entangling connections between different sub-networks could eventually be realized. In addition to that, the development of efficient quantum gradient estimation

protocols, which currently represents one of the main algorithmic bottlenecks, will be crucial for larger scale implementations of the proposed methods and for an adequate treatment of forces. It is also worth mentioning that another viable alternative may be represented by quantum kernel methods [49], whose classical counterparts are already extensively used in the context of ML potentials [9], for example in combination with Coulomb matrix descriptors [11].

On the other hand, the most interesting open questions concern the potential quantum advantages brought by QML approaches. In this work, we focused on the problem of learning from classical data and we observed, following [39], that suitably designed QNN models exhibit a larger effective dimension than their classical equivalents. This in turn relates to stable and fast training capabilities, and points toward a more effective handling of larger systems. Indeed, large normalized effective dimensions signal, in the spirit of capacity measures, an effective use of the available model parameters, thus suggesting that quantum models could represent economic and manageable tools to tackle large molecular simulations.

At the same time, this analysis does not yet take into account the role of overparametrization, which is known to contribute in a crucial way to the performances of classical NN. In fact, our numerical experiments confirm that large classical models, such as the ones employed in sections 3.2 and 3.3, still achieve the best prediction and generalization accuracy. Interestingly, while the normalized effective dimension of those classical models is quite small, the absolute effective dimension is actually comparable to the one of their direct quantum counterpart, thus hinting at some form of ‘computational phase transition’ in their behaviour [66–68]. A few similar observations have already been made for QML models [68, 69], and a thorough exploration of QNNs capabilities in such highly overparametrized regime—including, among others, the question of whether this could be realized in a more effective way or with different qualitative behaviours compared to classical models—represents an interesting open research direction in general. We leave a complete analysis of its application in the context of NNPs for future investigations.

On a larger perspective, one may also envisage the use of QML methods on quantum data [20, 29, 70] retrieved from experiments or quantum chemistry simulations, e.g. in the form of quantum wavefunctions generated through variational [41, 71] or dynamical methods. The extraction of physical/chemical properties and the classification of materials directly at the quantum level of description could then likely represent one of the most advanced and exploratory efforts in the quest for quantum advantage with QML.

Data availability statement

The data and code that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

We thank Amira Abbas, Julian Schuhmacher and Stefan Wörner for insightful discussions. We acknowledge financial support from the Swiss National Science Foundation (SNF) through the Grant No. 200021-179312. O K acknowledges financial support through a scholarship from the Dominik and Patrick Gemperlé Foundation.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. The current list of IBM trademarks is available at www.ibm.com/legal/copytrade.

Appendix A. Mirroring

As we recalled in the main text, QNN models introduced in equation (2) can be expressed as partial Fourier Series [44]:

$$f_{\Theta}(x) = \sum_{n \in \Omega} c_n e^{inx}, \quad (\text{A1})$$

where Ω is the set of available frequencies and depends exclusively on the encoding map, while the coefficients c_n are determined by the trainable unitary gates and the observable. QNNs of this form become universal functional approximators in the limit $|\Omega| \rightarrow \infty$. In practice, we make use of the Fourier series interpretation of QNNs and of Fourier analysis to guide model design, for example by changing model complexity (in our case essentially determined by the depth D for fixed qubit number) to control the number of available frequencies.

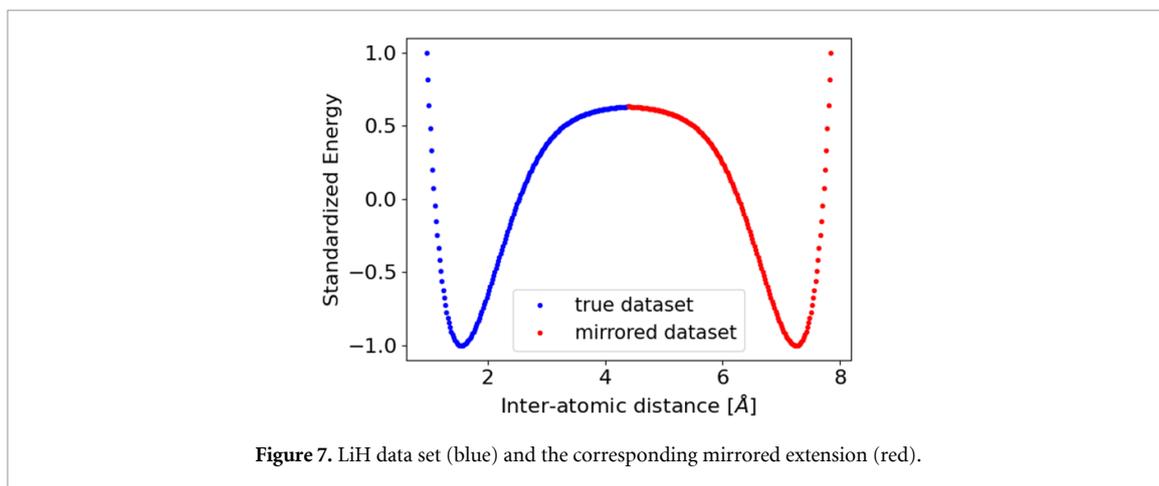


Figure 7. LiH data set (blue) and the corresponding mirrored extension (red).

In simple proof-of-principle experiments, such as LiH, we can also compare the model spectrum with the one of the data, although this may not be possible in general. We also empirically find that better performances are obtained for periodic data sets, which intuitively correspond to finite frequencies spectra and hence reduce spurious oscillations in the final solution. We apply this intuition to the 1-dimensional LiH and Ar₂ case, whose PES can be artificially made periodic by mirroring around $r = 4.5$ Å, (for LiH) and $r = 6.3$ Å (for Ar₂) as shown in figure 7.

Appendix B. LiH—molecular dynamics

The motivation behind both quantum and classical NNPs is ultimately to provide computationally effective access to highly accurate FFs to drive MD. In this section, we provide a first demonstration of a quantum NNP used to simulate the oscillations of a LiH molecule around its equilibrium position. We use the Velocity Verlet algorithm [72] to compute the time evolution of inter-atomic distance and velocity, starting from some given initial conditions. At each point, the forces are predicted with the trained QNN presented in section 3.1. For comparison, we also present results obtained with the classical NNP introduced in section 3.1 of the main text.

We start with the atoms at rest and close to each other, with an initial inter-atomic distance of 1.05 Å, and let the system evolve according to Newton's equations. The time evolution of the inter-atomic distance is shown in figure 8(a), where we compare the results obtained with classical and quantum NNPs to the exact solution obtained numerically. To assess the overall quality of the trajectories, we compute the frequency spectrum of the oscillations with a Fourier transform, which we apply to an evolution of 0.6 fs, repeated 1000 times and passed through an Hamming window. As reported in figure 8(b), the quantum and classical models can accurately reproduce the dominant frequencies.

Appendix C. H₃O⁺—internal coordinates

A hydronium molecule has $6 = 4 \times 3 - 6$ degrees of freedom. To describe its configurations, we used three bond lengths ($r_{\text{OH},1}$, $r_{\text{OH},2}$ and $r_{\text{OH},3}$), two angles ($\phi_{\text{H},1\text{OH},2}$ and $\phi_{\text{H},1\text{OH},3}$) and one dihedral angle ($d_{\text{OH},3\text{H},2\text{H},1}$). We used the following definition for the dihedral angle with atoms ($ijkl$):

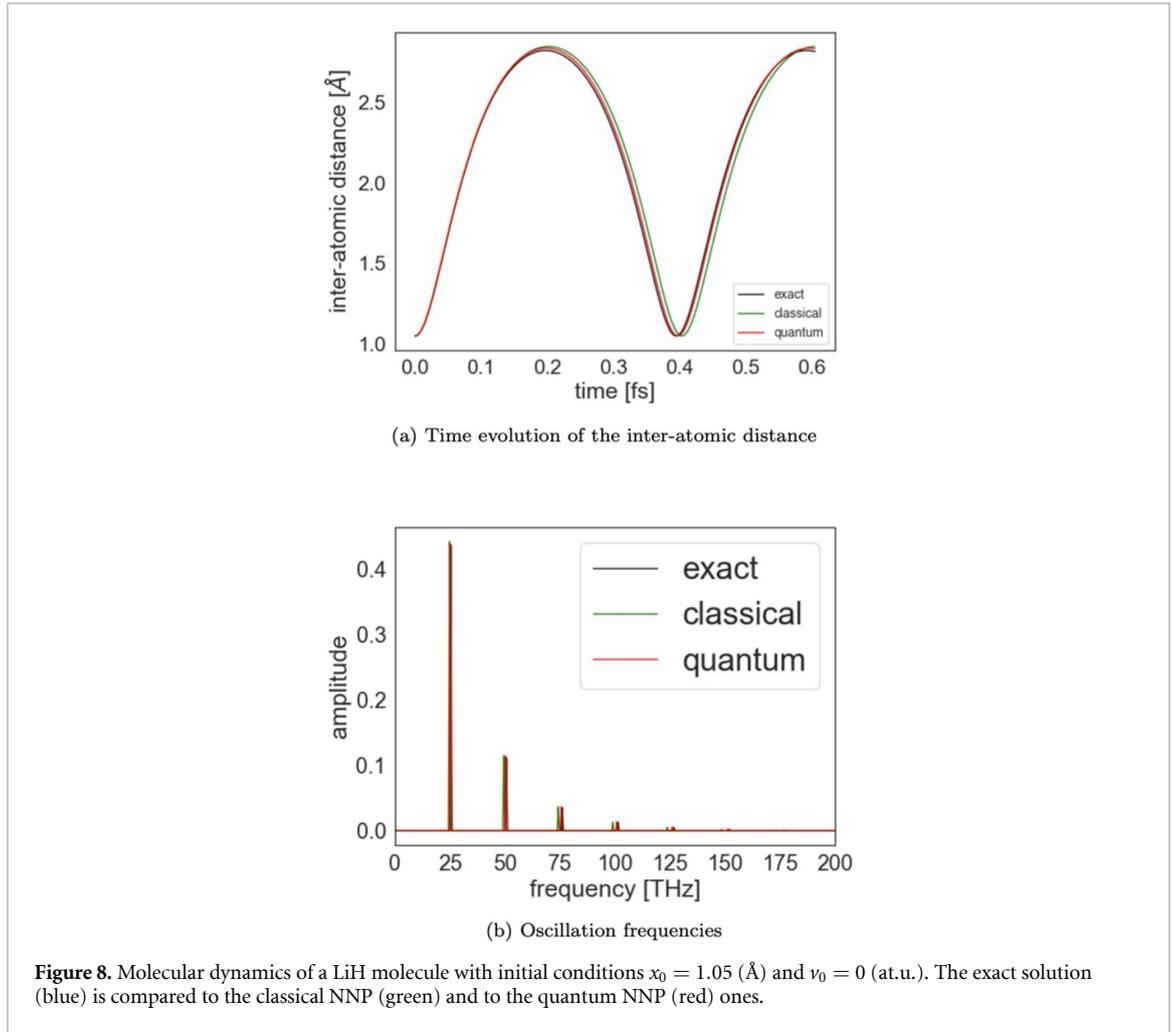
$$d_{ijkl} = \text{sign}(\chi) \arccos \frac{(\vec{r}_{ij} \times \vec{r}_{kj}) \cdot (\vec{r}_{kj} \times \vec{r}_{kl})}{|\vec{r}_{ij} \times \vec{r}_{kj}| \cdot |\vec{r}_{kj} \times \vec{r}_{kl}|} \quad (\text{C1})$$

$$\chi = \vec{r}_{kj} \cdot (\vec{r}_{ij} \times \vec{r}_{kl}) \times (\vec{r}_{kj} \times \vec{r}_{kl}), \quad (\text{C2})$$

where \vec{r}_{ij} is the distance vector between atom i and j . The above definitions and the formulas for their gradients can be found in [65].

Appendix D. Explicit QNN construction

In this section we show the explicit construction of the QNN in the case of a H₂O molecule. The data \vec{x} consist of three Euclidean coordinates for each of the three atoms, leading to nine features per individual configuration. However, by taking into account the planar structure of the molecule, as well as rotation and



translation symmetries, we can define a set of three independent internal coordinates. We choose them as the two O–H bond lengths and the H–H planar angle,

$$\vec{x} = \begin{pmatrix} r_{\text{OH},1} \\ r_{\text{OH},2} \\ \phi_{\text{HH}} \end{pmatrix}, \quad (\text{D1})$$

which we scale in $[-1, 1]$ and preprocess to create the following 3-dimensional feature vector:

$$\vec{y} = W(\vec{x}) = \begin{pmatrix} \arcsin(r_{\text{OH},1}) \\ \arcsin(r_{\text{OH},2}) \\ \arcsin(\phi_{\text{HH}}) \end{pmatrix} \equiv \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}. \quad (\text{D2})$$

For each data point in the training set and for any fixed value of the trainable parameters Θ during the model learning process, we compute the QNN output, defined as (see equation (2)):

$$f_{\Theta}(\vec{y}) = \langle 0 | M(\vec{y}, \Theta)^\dagger \mathcal{O} M(\vec{y}, \Theta) | 0 \rangle, \quad (\text{D3})$$

by executing the corresponding quantum circuit and measuring the expectation value of \mathcal{O} . As discussed in the main text, see section 2.1, the QNN model unitary $M(\vec{y}, \Theta)$ is constructed as a repetition of trainable and data encoding layers. The feature map is chosen as:

$$\Phi(\vec{y}) = \mathcal{E}(\vec{y}) \mathcal{S}(\vec{y}), \quad (\text{D4})$$

where $\mathcal{S}(\vec{y})$ is a collection of single-qubit Pauli- y rotations:

$$\mathcal{S}(\vec{y}) = \prod_{j=1}^3 \exp\left(-i \frac{\sigma_y^{(j)}}{2} y_j\right), \quad (\text{D5})$$

and $\mathcal{E}(\vec{y})$ is an entangling operation. Here, $\sigma_k^{(i)}$ ($k = x, y, z$) correspond to Pauli matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (\text{D6})$$

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad (\text{D7})$$

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (\text{D8})$$

acting on the i th qubit, i.e.

$$\sigma_y^{(i)} = \underbrace{\mathbb{1} \otimes \dots \otimes \mathbb{1}}_{i-1 \text{ times}} \otimes \sigma_y \otimes \underbrace{\mathbb{1} \otimes \dots \otimes \mathbb{1}}_{N-i \text{ times}}. \quad (\text{D9})$$

Notice that equation (D5) is written explicitly for the 3-qubit model that was used for H₂O, and it corresponds to the first set of operations highlighted in green in figure 2. As explained in section 3.2, we then used a *full* feature map with an order $l = 3$ interaction, i.e.

$$\begin{aligned} \mathcal{E}(\vec{y}) = & \underbrace{\prod_{j=1}^{N-1} \prod_{k=j+1}^N \exp\left(-i\sigma_z^{(j)}\sigma_z^{(k)}\frac{y_j y_k}{2}\right)}_{\text{full } l=2 \text{ interaction}} \\ & \cdot \underbrace{\prod_{j=1}^{N-2} \prod_{k=j+1}^{N-1} \prod_{m=k+1}^N \exp\left(-i\sigma_z^{(j)}\sigma_z^{(k)}\sigma_z^{(m)}\frac{y_j y_k y_m}{2}\right)}_{\text{full } l=3 \text{ interaction}}, \end{aligned} \quad (\text{D10})$$

as explicitly depicted in the quantum circuit of figure 2 for $N = 3$. In particular, the order $l = 2$ operations (yellow in figure 2) are decomposed as:

$$\exp\left(-i\sigma_z^{(j)}\sigma_z^{(k)}\frac{y_j y_k}{2}\right) = \text{CX}_{j,k} \text{R}_z^{(k)}(y_j y_k) \text{CX}_{j,k}, \quad (\text{D11})$$

while $l = 3$ ones (red in figure 2) read:

$$\exp\left(-i\sigma_z^{(j)}\sigma_z^{(k)}\sigma_z^{(m)}\frac{y_j y_k y_m}{2}\right) = \text{CX}_{j,k} \text{CX}_{k,m} \text{R}_z^{(m)}(y_j y_k y_m) \text{CX}_{k,m} \text{CX}_{j,k}, \quad (\text{D12})$$

here, $\text{R}_z^{(m)}(\cdot) = e^{-i(\cdot)\sigma_z^{(m)}/2}$ is a z -rotation on qubit m , and $\text{CX}_{j,k}$ denotes a CNOT [21] operation between qubits j and k .

The trainable layers follow the same structure, except the data entries of the form y_j , $y_j y_k$ and $y_j y_k y_m$ are replaced with distinct parameters θ_m , where m is an index assigned to each trainable gate.

ORCID iDs

Oriel Kiss  <https://orcid.org/0000-0001-7461-3342>

Francesco Tacchino  <https://orcid.org/0000-0003-2008-5956>

Sofia Vallecorsa  <https://orcid.org/0000-0002-7003-5765>

Ivano Tavernelli  <https://orcid.org/0000-0001-5690-1981>

References

- [1] Alder B J and Wainwright T E 1959 *J. Chem. Phys.* **31** 459–66
- [2] McCammon J A, Gelin B R and Karplus M 1977 *Nature* **267** 585–90
- [3] Car R and Parrinello M 1985 *Phys. Rev. Lett.* **55** 2471–4
- [4] Marx D and Hutter J 2009 *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods* (Cambridge: Cambridge University Press)
- [5] Unke O T, Koner D, Patra S, Käser S and Meuwly M 2020 *Mach. Learn.: Sci. Technol.* **1** 013001
- [6] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 *Rev. Mod. Phys.* **91** 045002
- [7] Pfau D, Spencer J S, Matthews A G D G and Foulkes W M C 2020 *Phys. Rev. Res.* **2** 033429

- [8] Behler J and Parrinello M 2007 *Phys. Rev. Lett.* **98** 146401
- [9] Behler J 2016 *J. Chem. Phys.* **145** 170901
- [10] Behler J 2011 *J. Chem. Phys.* **134** 074106
- [11] Rupp M, Tkatchenko A, Müller K-R and von Lilienfeld O A 2012 *Phys. Rev. Lett.* **108** 058301
- [12] Morawietz T, Singraber A, Dellago C and Behler J 2016 *Proc. Natl Acad. Sci.* **113** 8368–73
- [13] Gastegger M, Schwiedrzik L, Bittermann M, Berzsényi F and Marquetand P 2018 *J. Chem. Phys.* **148** 241709
- [14] Imbalzano G, Anelli A, Giofré D, Klees S, Behler J and Ceriotti M 2018 *J. Chem. Phys.* **148** 241730
- [15] Unke O T, Chmiela S, Sauceda H E, Gastegger M, Poltavsky I, Schütt K T, Tkatchenko A and Müller K-R 2021 *Chem. Rev.* **121** 10142–86
- [16] Cheng B, Mazzola G, Pickard C J and Ceriotti M 2020 *Nature* **585** 217–20
- [17] Singraber A, Behler J and Dellago C 2019 *J. Chem. Theory Comput.* **15** 1827–40
- [18] Singraber A, Morawietz T, Behler J and Dellago C 2019 *J. Chem. Theory Comput.* **15** 3075–92
- [19] Biamonte J, Witte P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195–202
- [20] Huang H Y et al 2021 Quantum advantage in learning from experiments (arXiv:2112.00778)
- [21] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge: Cambridge University Press)
- [22] Schuld M and Petruccione F 2021 *Machine Learning with Quantum Computers* (Cham: Springer)
- [23] Huang H-Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H and McClean J R 2021 *Nat. Commun.* **12** 2631
- [24] Huang H Y, Kueng R, Torlai G, Albert V V and Preskill J 2021 Provably efficient machine learning for quantum many-body problems (arXiv:2106.12627)
- [25] Schuld M, Sinayskiy I and Petruccione F 2015 *Phys. Lett. A* **379** 660–3
- [26] Cao Y, Guerreschi G G and Aspuru-Guzik A 2017 Quantum neuron: an elementary building block for machine learning on quantum computers (arXiv:1711.11240)
- [27] Torrontegui E and Garcia-Ripoll J J 2019 *Europhys. Lett.* **125** 030004
- [28] Tacchino F, Macchiavello C, Gerace D and Bajoni D 2019 *npj Quantum Inf.* **5** 26
- [29] Cong I, Choi S and Lukin M D 2019 *Nat. Phys.* **15** 1273–8
- [30] Tacchino F, Barkoutsos P, Macchiavello C, Tavernelli I, Gerace D and Bajoni D 2020 *Quantum Sci. Technol.* **5** 044010
- [31] Beer K, Bondarenko D, Farrelly T, Osborne T J, Salzmann R, Scheiermann D and Wolf R 2020 *Nat. Commun.* **11** 808
- [32] Mangini S, Tacchino F, Gerace D, Macchiavello C and Bajoni D 2020 *Mach. Learn.: Sci. Technol.* **1** 045008
- [33] Mangini S, Tacchino F, Gerace D, Bajoni D and Macchiavello C 2021 *Europhys. Lett.* **134** 10002
- [34] Mitarai K, Negoro M, Kitagawa M and Fujii K 2018 *Phys. Rev. A* **98** 032309
- [35] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 *Quantum Sci. Technol.* **4** 043001
- [36] Schuld M, Bocharov A, Svore K M and Wiebe N 2020 *Phys. Rev. A* **101** 032308
- [37] Cerezo M et al 2021 *Nat. Rev. Phys.* **3** 625–44
- [38] Tacchino F, Mangini S, Barkoutsos P K, Macchiavello C, Gerace D, Tavernelli I and Bajoni D 2021 *IEEE Trans. Quantum Eng.* **2** 3101110
- [39] Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A and Woerner S 2021 *Nat. Comput. Sci.* **1** 403–9
- [40] Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love P J, Aspuru-Guzik A and O'Brien J L 2014 *Nat. Commun.* **5** 4213
- [41] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 *Nature* **549** 242–6
- [42] Sokolov I O, Barkoutsos P K, Moeller L, Suchsland P, Mazzola G and Tavernelli I 2021 *Phys. Rev. Res.* **3** 013125
- [43] Xia R and Kais S 2020 *Entropy* **22** 828
- [44] Schuld M, Sweke R and Meyer J J 2021 *Phys. Rev. A* **103** 032430
- [45] Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E and Latorre J I 2020 *Quantum* **4** 226
- [46] Gil Vidal F J and Theis D O 2020 *Front. Phys.* **8** 297
- [47] Jerbi S, Fiderer L J, Nautrup H P, Kübler J M, Briegel H J and Dunjko V 2021 Quantum machine learning beyond kernel methods (arXiv:2110.13162)
- [48] Gastegger M and Marquetand P 2020 *Molecular Dynamics with Neural Network Potentials* (Cham: Springer International Publishing) pp 233–52
- [49] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 *Nature* **567** 209–12
- [50] Tacchino F, Chiesa A, Carretta S and Gerace D 2020 *Adv. Quantum Technol.* **3** 1900052
- [51] Schuld M, Bergholm V, Gogolin C, Izaac J and Killoran N 2019 *Phys. Rev. A* **99** 032331
- [52] Mitarai K and Fujii K 2019 *Phys. Rev. Res.* **1** 013006
- [53] Grant E, Wossnig L, Ostaszewski M and Benedetti M 2019 *Quantum* **3** 214
- [54] Abbas A, Sutter D, Figalli A and Woerner S 2021 Effective dimension of machine learning models (arXiv:2112.04807)
- [55] Kingma D and Ba J 2015 Adam: a method for stochastic optimization *Proc. 3rd Int. Conf. on Learning Representations (ICLR)*
- [56] Glorot X and Bengio Y 2010 Understanding the difficulty of training deep feedforward neural networks *Proc. 13th Int. Conf. on Artificial Intelligence and Statistics* vol 9 (PMLR) pp 249–56
- [57] Frisch M J et al 2016 *Gaussian 16 Revision C.01* (Wallingford, CT: Gaussian Inc.)
- [58] Morawietz T and Behler J 2019 HDNNP training data set for H₂O (<https://doi.org/10.5281/zenodo.2634098>)
- [59] Singraber A, Behler J and Dellago C 2019 *J. Chem. Theory Comput.* **15** 1827–40
- [60] Christensen A S and von Lilienfeld O A 2020 *Mach. Learn.: Sci. Technol.* **1** 045018
- [61] Schran C, Behler J and Marx D 2020 *J. Chem. Theory Comput.* **16** 88–99
- [62] CPMD Copyright IBM Corp 1990–2015, Copyright MPI für Festkörperforschung Stuttgart 1997–2001 (available at: www.cpmid.org/)
- [63] Perdew J P, Burke K and Ernzerhof M 1996 *Phys. Rev. Lett.* **77** 3865
- [64] Troullier N and Martins J L 1991 *Phys. Rev. B* **43** 8861–9
- [65] van Schaik R C, Berendsen H J C, Torda A E and van Gunsteren W F 1993 *J. Mol. Biol.* **234** 751–62
- [66] Belkin M, Hsu D, Ma S and Mandal S 2019 *Proc. Natl Acad. Sci.* **116** 15849–54
- [67] Nakkiran P, Kaplan G, Bansal Y, Yang T, Barak B and Sutskever I 2019 Deep double descent: where bigger models and more data hurt (arXiv:1912.02292)

- [68] Larocca M, Ju N, García-Martín D, Coles P J and Cerezo M 2021 Theory of overparametrization in quantum neural networks (arXiv:[2109.11676](https://arxiv.org/abs/2109.11676))
- [69] Liu J, Najafi K, Sharma K, Tacchino F, Jiang L and Mezzacapo A 2022 An analytic theory for the dynamics of wide quantum neural networks (arXiv:[2203.16711](https://arxiv.org/abs/2203.16711))
- [70] Kawai H and Nakagawa Y O 2020 *Mach. Learn.: Sci. Technol.* **1** 045027
- [71] McArdle S, Endo S, Aspuru-Guzik A, Benjamin S C and Yuan X 2020 *Rev. Mod. Phys.* **92** 015003
- [72] Verlet L 1967 *Phys. Rev.* **159** 98–103