

## A VARIATIONAL BAYES

In this section we review variational Bayes.

Given  $M$  input/output pairs  $\mathbf{X} = (\mathbf{x}^1, \dots, \mathbf{x}^M)$ ,  $\mathbf{Y} = (\mathbf{y}^1, \dots, \mathbf{y}^M)$ , and a prior over weights  $p(\mathbf{W})$ , in a Bayesian approach we are interested in computing the posterior  $p(\mathbf{W}|\mathbf{X}, \mathbf{Y}) = p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W})/p(\mathbf{Y}|\mathbf{X})$ , which is used to make a prediction on a test point  $\mathbf{x}_*$ :  $p(\mathbf{y}|\mathbf{x}_*) = \mathbb{E}_{p(\mathbf{W}|\mathbf{X}, \mathbf{Y})}[p(\mathbf{y}|\mathbf{x}_*, \mathbf{W})]$ . However the posterior computation involves an intractable denominator. We proceed using variational Bayes and we introduce an approximate posterior  $q_\theta(\mathbf{W})$  which depends on hyperparameters  $\theta$ , chosen to maximize the evidence  $p(\mathbf{Y}|\mathbf{X})$  lower bound (ELBO) objective:

$$\text{ELBO} = \mathbb{E}_{q_\theta(\mathbf{W})}[\log p(\mathbf{Y}|\mathbf{X}, \mathbf{W})] - \text{KL}(q_\theta(\mathbf{W})||p(\mathbf{W})) \leq p(\mathbf{Y}|\mathbf{X}). \quad (27)$$

The resulting approximate posterior is then used for a prediction:  $q(\mathbf{y}|\mathbf{x}_*) := \mathbb{E}_{q_\theta(\mathbf{W})}[p(\mathbf{y}|\mathbf{x}_*, \mathbf{W})]$ . Maximizing the gap in variational Bayes is equivalent to minimizing the KL between the approximate and the true posterior Kingma et al. (2015):

$$\text{ELBO} - \log p(\mathbf{Y}|\mathbf{X}) = -\text{KL}(q_\theta(\mathbf{W})||p(\mathbf{W}|\mathbf{X}, \mathbf{Y})). \quad (28)$$

## B REVIEW OF QUANTUM MECHANICS

States in quantum mechanics (QM) are represented as abstract vectors in a Hilbert space  $\mathcal{H}$ , denoted as  $|\psi\rangle$ . Here, we will only be concerned with qubits which can have two possible states  $|0\rangle$  and  $|1\rangle$ . These states are defined relative to a particular frame of reference, e.g. spin values measured along the z-axis. More generally, a qubit is described as a complex linear combination (or superposition) of up and down z-spins:  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ .

$N$  disentangled qubits are described by a product state  $|\psi\rangle = \prod_{i=1}^N |\psi_i\rangle$ . Under interactions qubits entangle with each other. This means that the wave function is now a complex linear combination of an exponential number of  $2^N$  terms. We can write this as  $|\psi'\rangle = U|\psi\rangle$  where  $U$  is a unitary matrix in a  $2^N \times 2^N$  dimensional space. This entangled state is still 'pure' in the sense that there is nothing more to learn about it, i.e. its quantum entropy is zero and it represents maximal information about the system. However, in QM that does not mean our knowledge of the system is complete.

Time evolution in QM is described by a unitary transformation,  $|\psi(t)\rangle = U(t, 0)|\psi(0)\rangle$  with  $U(t, 0) = e^{iHt}$  where  $H$  is the Hamiltonian, a Hermitian operator. Note that time evolution entangles qubits. We will use time evolution to map an input state to an output state as a layer in a NN, not unlike a neural ODE Chen et al. (2018).

Measurements in QM are nothing else than projecting the state onto the eigenbasis of a symmetric positive definite operator  $\mathbf{A}$ . The quantum system collapses into a particular state with a probability given by Born's rule:  $p_i = |\langle\phi_i|\psi\rangle|^2$  where  $\{|\phi_i\rangle\}$  are the orthonormal eigenvectors of  $\mathbf{A}$ .

We will also need to describe "mixed states". A mixed state is a classical mixture of a number of pure quantum states. Probabilities in this mixture encode our uncertainty about what quantum state the system is in. This type of uncertainty is the one we are used to in AI, it results from a lack of knowledge about the system.

Mixed states are not naturally described by wave vectors. For that we need a tool called the density matrix  $\rho$ . For a pure state we use  $\rho = |\psi\rangle\langle\psi|$ , a rank-1 matrix (or outer product), where  $\langle\psi|$  is the complex transpose of the vector  $|\psi\rangle$ . But for a mixed state the rank will be higher and  $\rho$  can be decomposed as  $\rho = \sum_k p_k |\psi_k\rangle\langle\psi_k|$  with  $\{p_k\}$  (positive) probabilities that sum to 1. Note that a unitary transformation will change the basis but not the rank (and hence will keep pure states pure):  $\text{Rank}(\rho') = \text{Rank}(U\rho U^\dagger)$ . In particular, time evolution will preserve rank and keep pure states pure.

The probability of a measurement is given by the trace of the density matrix over the projector  $\mathbf{A}_i = |\phi_i\rangle\langle\phi_i|$ , namely  $p_i = \text{Tr}(\mathbf{A}_i\rho) = \text{Tr}(|\phi_i\rangle\langle\phi_i|\psi\rangle\langle\psi|) = |\langle\phi_i|\psi\rangle|^2$ , i.e. Born's rule. Similar to marginalisation in classical probability theory, we can trace over degrees of freedom we are not interested in, i.e.  $\rho_a = \text{Tr}_b(\rho_{ab})$ .

Further, if two operators  $A, A'$  commute, they have a common eigenbasis and we can simultaneously measure them leading to a joint probability distribution  $p(\lambda_\alpha, \lambda'_\beta) = \langle \psi | \Pi_{\lambda_\alpha} \Pi_{\lambda'_\beta} | \psi \rangle$ , where  $\lambda'_\beta$  are the eigenvalues of  $A'$  and  $\Pi_{\lambda'_\beta}$  ( $\Pi_{\lambda'_\alpha}$ ) projects onto the eigenspace of  $A$  ( $A'$ ).

### B.1 REVIEW OF QUANTUM PHASE ESTIMATION

We review here the quantum phase estimation, a quantum algorithm to estimate the eigenphases of a unitary  $U$ . Suppose first that we know an eigenvector  $|v\rangle$  of  $U$  with eigenvalue  $\exp(\frac{2\pi i}{2^t} \varphi)$ , and that  $\varphi$  can be represented with  $t$  bits:  $\varphi = 2^{t-1} \varphi^1 + \dots + 2^0 \varphi^t$ . Then introduce  $t$  ancilla qubits in equal weight superposition of all the  $2^t$  states:  $H^{\otimes t} |0\rangle^{\otimes t}$ ,

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H^{\otimes t} |0\rangle^{\otimes t} = \left( \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right)^{\otimes t} = \frac{1}{2^{t/2}} \sum_{\ell=0}^{2^t-1} |\ell\rangle. \quad (29)$$

where we used the identification:  $|0\rangle \equiv (1, 0)^T$  and introduced the basis  $\{|\ell\rangle\}_{\ell=0}^{2^t-1}$  for the ancilla qubits, i.e.  $\ell = 2^{t-1} \varphi^1 + \dots + 2^0 \varphi^t$ . We use the ancilla's as control qubits for applying powers of  $U$  on the input state  $|v\rangle$ , implementing the following unitary map:

$$\frac{1}{2^{t/2}} \sum_{\ell=0}^{2^t-1} |\ell\rangle \otimes |v\rangle \mapsto \frac{1}{2^{t/2}} \sum_{\ell=0}^{2^t-1} |\ell\rangle \otimes U^\ell |v\rangle = \frac{1}{2^{t/2}} \sum_{\ell=0}^{2^t-1} e^{\frac{2\pi i}{2^t} \ell \varphi} |\ell\rangle \otimes |v\rangle. \quad (30)$$

Next, we apply the inverse Fourier transform on the ancilla register to get:

$$\frac{1}{2^t} \sum_{\ell, k=0}^{2^t-1} e^{\frac{2\pi i}{2^t} (\varphi - k) \ell} |k\rangle \otimes |v\rangle = \sum_{k=0}^{2^t-1} \delta_{\varphi, k} |k\rangle \otimes |v\rangle = |\varphi\rangle \otimes |v\rangle. \quad (31)$$

The quantum complexity of this operation is linear in  $t$ . By linearity if the input is a generic state,  $|\psi\rangle = \sum_\alpha |v_\alpha\rangle \langle v_\alpha| \psi\rangle$ ,  $|v_\alpha\rangle$  an eigenstate of  $U$  with eigenvalue  $\exp(\frac{2\pi i}{2^t} \varphi_\alpha)$ , quantum phase estimation will act as:

$$|0\rangle^{\otimes t} \otimes |\psi\rangle \mapsto \sum_\alpha \langle v_\alpha | \psi \rangle |\varphi_\alpha\rangle \otimes |v_\alpha\rangle. \quad (32)$$

Here we also assumed that we can represent all the  $\varphi_\alpha$ 's with  $t$  bits, in which case the reduced density matrix of the ancilla state is diagonal

$$\rho_{\text{anc}} = \sum_{\alpha\beta} |\varphi_\alpha\rangle \langle \varphi_\beta| \langle v_\alpha | \psi \rangle \langle v_\beta | \psi \rangle \text{Tr}\{|v_\alpha\rangle \langle v_\beta|\} = \sum_\alpha |\varphi_\alpha\rangle \langle \varphi_\alpha| |\langle v_\alpha | \psi \rangle|^2, \quad (33)$$

so that the probability of measuring  $\varphi_\alpha$  is governed by  $|\langle v_\alpha | \psi \rangle|^2$ . In particular, the probability that the first ancilla bit is  $b$  is given by:

$$p(b) = \sum_\alpha |\langle v_\alpha | \psi \rangle|^2 \delta(\sigma(2^{-t} \varphi_\alpha) - b), \quad (34)$$

where  $\sigma$  is the threshold non-linearity introduced in equation 1 since  $2^{-t} \varphi = 2^{-1} \varphi^1 + \dots + 2^{-t} \varphi^t$  and if the first bit  $\varphi^1 = 0$  then  $2^{-t} \varphi < \frac{1}{2}$  and  $\sigma(2^{-t} \varphi) = 0$ , while if  $\varphi^1 = 1$ , then  $2^{-t} \varphi \geq \frac{1}{2}$  and  $\sigma(2^{-t} \varphi) = 1$ .

Note that computing the reduced density matrix can be done by simply discarding qubits on a quantum computer, while it can be exponentially hard classically. In practice,  $t$  can be chosen to be lower than the precision of  $\varphi$  and in that case one can estimate the accuracy of the measurement, see Nielsen & Chuang (2000) for details. We depict the quantum phase estimation with measurement of the first ancilla in figure 3.

## C DETAILED IMPLEMENTATION OF QUANTUM DEFORMED NEURAL NETWORKS

We here give a detailed derivation of the formulas related to implementing quantum circuit of a layer in a quantum deformed neural network depicted in figure 1 (a). The input state to that circuit

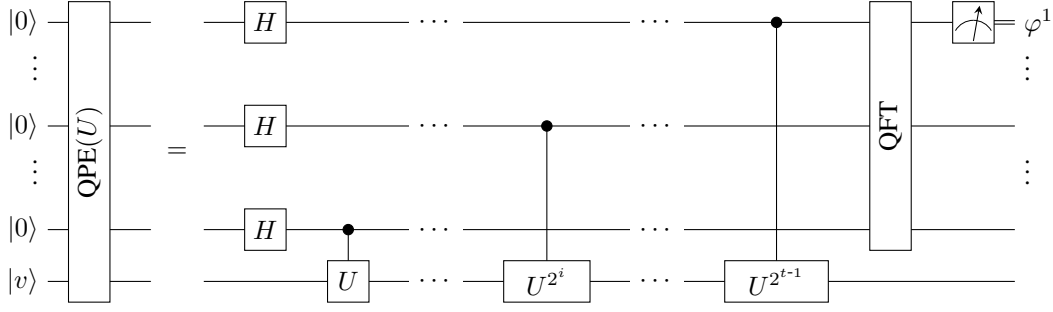


Figure 3: Quantum phase estimation circuit with measurement of the first ancilla qubit.

is  $|0\rangle^{\otimes tM} \otimes |\psi\rangle$ , where we recall that

$$|\psi\rangle = |\psi\rangle_h \otimes |\psi\rangle_W, \quad |\psi\rangle_W = \bigotimes_{i=1}^N \bigotimes_{j=1}^M \left[ \sqrt{q_{ji}(W_{ji}=0)} |0\rangle + \sqrt{q_{ji}(W_{ji}=1)} |1\rangle \right]. \quad (35)$$

We start by applying the first quantum phase estimation of  $U_1$  which involves only the first  $t$  ancillas. W.r.t. equation 8 we identify  $|v_\alpha\rangle \equiv |\mathbf{h}, \mathbf{W}\rangle_D$ ,  $|\varphi_\alpha\rangle \equiv |\varphi(\mathbf{h}, \mathbf{W}_{1,:})\rangle$ , to get:

$$|0\rangle^{\otimes t} \otimes |\psi\rangle \mapsto \sum_{\mathbf{h} \in \mathbb{B}^N} \sum_{\mathbf{W} \in \mathbb{B}^{NM}} D \langle \mathbf{h}, \mathbf{W} | \psi \rangle |\varphi(\mathbf{h}, \mathbf{W}_{1,:})\rangle \otimes |\mathbf{h}, \mathbf{W}\rangle_D. \quad (36)$$

Then we proceed to applying to the result the second quantum phase estimation involving  $U_2$  and the second batch of  $t$  ancilla qubits:

$$\sum_{\mathbf{h} \in \mathbb{B}^N} \sum_{\mathbf{W} \in \mathbb{B}^{NM}} D \langle \mathbf{h}, \mathbf{W} | \psi \rangle |0\rangle^{\otimes t} \otimes |\varphi(\mathbf{h}, \mathbf{W}_{1,:})\rangle \otimes |\mathbf{h}, \mathbf{W}\rangle_D \mapsto \quad (37)$$

$$\sum_{\mathbf{h} \in \mathbb{B}^N} \sum_{\mathbf{W} \in \mathbb{B}^{NM}} D \langle \mathbf{h}, \mathbf{W} | \psi \rangle |\varphi(\mathbf{h}, \mathbf{W}_{2,:})\rangle \otimes |\varphi(\mathbf{h}, \mathbf{W}_{1,:})\rangle \otimes |\mathbf{h}, \mathbf{W}\rangle_D. \quad (38)$$

We repeat the procedure  $M$  times to get to the final state:

$$\sum_{\mathbf{h} \in \mathbb{B}^N} \sum_{\mathbf{W} \in \mathbb{B}^{NM}} D \langle \mathbf{h}, \mathbf{W} | \psi \rangle |\varphi(\mathbf{h}, \mathbf{W})\rangle \otimes |\mathbf{h}, \mathbf{W}\rangle_D, \quad |\varphi(\mathbf{h}, \mathbf{W})\rangle \equiv \bigotimes_{j=1}^M |\varphi(\mathbf{h}, \mathbf{W}_{j,:})\rangle, \quad (39)$$

and compute the reduced density matrix of the ancilla qubits  $\rho_{\text{anc}}$  which is diagonal as in equation 33:

$$\rho_{\text{anc}} = \sum_{\mathbf{h} \in \mathbb{B}^N} \sum_{\mathbf{W} \in \mathbb{B}^{NM}} |D \langle \mathbf{h}, \mathbf{W} | \psi \rangle|^2 |\varphi(\mathbf{h}, \mathbf{W})\rangle \langle \varphi(\mathbf{h}, \mathbf{W})|. \quad (40)$$

Now we compute the outcome probability of a measurement of the first qubit in each of the  $M$  registers of ancilla qubits. Recalling equation 9 and the fact that the first bit of an integer is the most significant bit, determining whether  $2^{-t}\varphi(\mathbf{h}, \mathbf{W}_{j,:}) = (N+1)^{-1}\varphi(\mathbf{h}, \mathbf{W}_{j,:})$  is greater or smaller than  $1/2$ , the probability of outcome  $\mathbf{h}' = (h'_1, \dots, h'_M)$  is

$$p(\mathbf{h}') = \sum_{\mathbf{h} \in \mathbb{B}^N} \sum_{\mathbf{W} \in \mathbb{B}^{NM}} \delta(\mathbf{h}' - f(\mathbf{W}, \mathbf{h})) |\langle \psi | \mathbf{h}, \mathbf{W} \rangle_D|^2, \quad (41)$$

where  $f$  is the layer function introduced in equation 1.

## D THE CASE OF CONVOLUTIONAL LAYERS

We extend here the model of 3.3 to the convolution case, where the eigenphases we want to estimate using the quantum phase estimation are:

$$\varphi_{i,j,d} = \sum_{k=1}^{K_1} \sum_{\ell=1}^{K_2} \sum_{c=1}^C W_{k,\ell,c,d} h_{i+k,\ell+j,c}. \quad (42)$$

Here the kernel  $\mathbf{W}$  has size  $(K_1, K_2, C, C')$ , where  $K_1$  ( $K_2$ ) is the kernel along the height (width) direction, while  $C$  ( $C'$ ) is the number of input (output) channels and  $d = 1, \dots, C'$ . Classically, we can implement the convolution by extracting patches of size  $K_1 \times K_2 \times C$  from the image and perform the dot product of the patches with a flattened kernel for each output channel. Moving on to the quantum implementation discussed in section 3.3, we recall that the input activation distribution is factorized. Therefore it is encoded in a product state, and we define patches analogously to the classical case since there is no entanglement coupling the different patches. The quantum convolutional layer can then be implemented as outlined above in the classical case, replacing the classical dot product with the quantum circuit implementing the fully connected layer of section 3.3. The resulting quantum layer is a translation equivariant for any choice of  $\mathbf{D}_j$ .

## E DETAILS OF CLASSICAL SIMULATIONS

We compute here the mean and variance of equation 25 with the choice of equation 26. Denoting  $\mathbf{B}_{2i} = \mathbf{B}_i^H$ ,  $\mathbf{B}_{2i+1} = \mathbf{B}_i^W$ , we have

$$\mathbf{K}_i = \mathbf{D} \frac{1}{N} \mathbf{B}_{2i} \mathbf{B}_{2i+1} \mathbf{D}^{-1} = \mathbf{M}_i^{-1} \frac{1}{N} \mathbf{B}_{2i} \mathbf{B}_{2i+1} \mathbf{M}_i, \quad \mathbf{M}_i = \mathbf{Q}_{2i, 2i+1} \mathbf{P}_{2i-1, 2i} \mathbf{P}_{2i+1, 2i+2} \quad (43)$$

so the random variable associated to  $\mathbf{K}_i$  will have support only on the four qubits  $\{2i-1, 2i, 2i+1, 2i+2\}$ . Thus  $\langle \psi | \mathbf{K}_i \mathbf{K}_{i'} | \psi \rangle = \langle \psi | \mathbf{K}_i | \psi \rangle \langle \psi | \mathbf{K}_{i'} | \psi \rangle$  for  $|i - i'| > 1$  and the CLT can be applied. If we write  $|\psi\rangle = \otimes_{i=0}^{2N-1} |\psi_i\rangle$ , and denote:

$$\langle \mathbf{X} \rangle_{i:i'} \equiv \langle \psi_i | \dots \langle \psi_{i'} | \mathbf{X} | \psi_i \rangle \dots | \psi_{i'} \rangle, \quad (44)$$

the mean and variances are:

$$\mu = \sum_{i=0}^{N-1} \mu_i, \quad \mu_i = \begin{cases} \langle \mathbf{K}_0 \rangle_{0:2} & i = 0 \\ \langle \mathbf{K}_i \rangle_{2i-1:2i+2} & 0 < i < N-1 \\ \langle \mathbf{K}_{N-1} \rangle_{2N-3:2N-1} & i = N-1 \end{cases} \quad (45)$$

$$\sigma^2 = \sum_{ij} (\langle \psi | \mathbf{K}_i \mathbf{K}_j | \psi \rangle - \langle \psi | \mathbf{K}_i | \psi \rangle \langle \psi | \mathbf{K}_j | \psi \rangle) \quad (46)$$

$$= 2 \sum_{i < j} (\langle \psi | \mathbf{K}_i \mathbf{K}_j | \psi \rangle - \langle \psi | \mathbf{K}_i | \psi \rangle \langle \psi | \mathbf{K}_j | \psi \rangle) + \sum_{i=0}^{N-1} (\langle \psi | \mathbf{K}_i^2 | \psi \rangle - \langle \psi | \mathbf{K}_i | \psi \rangle^2) \quad (47)$$

$$= 2 \sum_{i=0}^{N-2} (\gamma_{i,i+1} - \mu_i \mu_{i+1}) + \sum_{i=0}^{N-1} (\mu_i - \mu_i^2) \quad (48)$$

$$\gamma_{i,i+1} = \langle \psi | \mathbf{K}_i \mathbf{K}_{i+1} | \psi \rangle = \begin{cases} \langle \mathbf{K}_0 \mathbf{K}_1 \rangle_{0:4} & i = 0 \\ \langle \mathbf{K}_i \mathbf{K}_{i+1} \rangle_{2i-1:2i+4} & 0 < i < N-2 \\ \langle \mathbf{K}_{N-2} \mathbf{K}_{N-1} \rangle_{2N-3:2N-1} & i = N-2 \end{cases} \quad (49)$$

In computing  $\sigma^2$  we used  $\mathbf{K}_i^2 = \mathbf{K}_i$ . Note that both  $\mu$  and  $\sigma^2$  can be computed in  $O(N)$  and parallelized. Naively, computing  $\langle \mathbf{K}_i \rangle_{2i-1:2i+2}$  and  $\langle \mathbf{K}_i \mathbf{K}_{i+1} \rangle_{2i-1:2i+4}$  takes  $(d^2)^3 + d$ , (the first term comes from the three layers of matrix vector products shown in figure 2 (b,c) and the second term from the final dot product) where  $d$  is the dimensionality of the input state, i.e.  $d = 2^4$  in the mean case and  $d = 2^6$  in the variance case. These constants can be further reduced by choosing an order of contractions of the tensors exploiting the local connectivities. These times can be further greatly reduced if we assume that  $\mathbf{P}_{2i-1, 2i} = 1$ , in which case  $\mathbf{K}_i$  is supported only on two sites and  $\mathbf{K}_i, \mathbf{K}_j$  are uncorrelated for all  $i, j$ .

## F DETAILS OF THE EXPERIMENTS AND ADDITIONAL NUMERICAL RESULTS

We summarize here details of the experiments discussed in section 4.2. We parametrize the  $4 \times 4$  unitaries  $\mathbf{P}_{i,i+1}^j, \mathbf{Q}_{i,i+1}^j$  as follows. For each  $4 \times 4$  unitary  $\mathbf{U}$  we introduce  $4 \times 4$  real matrices  $\mathbf{A}, \mathbf{B}$ , which are learnable. Then we compute  $\mathbf{U}$  as in Procedure 3. Note that the bandPart routine

effectively discards half of the parameters, halving the actual number of trainable variables, and we use this routine only for implementation convenience. We will keep that into account when counting parameters below.

---

**Procedure 3** Parametrization of a unitary matrix

---

**Input:**  $A, B$   $\{N \times N$  real matrices $\}$

**Output:**  $U$

$C \leftarrow A + iB$

$C \leftarrow \text{bandPart}(C, 0, -1)$  {Set lower triangular part, diagonal excluded, to zero}

$U \leftarrow \exp(C - C^\dagger)$   $\{N \times N$  unitary matrix $\}$

---

At the last layer of the neural network we produce a number of output probabilities equal to the number of classes, each of which is a number in  $[0, 1]$ . We then normalize these to interpret the normalized result as probability of the input to be in a given class, fed into a cross entropy loss.

For MNIST we preprocess the data by binarizing the images and using the binary value as input bit to the quantum neural network. For Fashion MNIST such procedure would result in a great loss of texture and therefore we simply normalize the pixels to  $[0, 1]$  and use that value as input probability. We train the models with the Adam optimizer and  $\beta = 10^{-6}$  coefficient of the variance regularization term as in Peters & Welling (2018). We train for a fixed budget of 50 epochs for MNIST Arch. A,B,C and FashionMNIST Arch. C, and 100 epochs for Fashion-MNIST Arch. A,B. We sweep over the coefficient of the  $L_2$  regularization for the deformation parameters between values  $\{0, 10^{-4}\}$  and also sweep the learning rate schedule, between constant and equal to 0.01 and piecewise decay from 0.01 to 0.001, choosing the best model among all these. On top of the learnable parameters discussed so far, we also add bias terms to the layers in all runs.

Table 2 extends the results of table 1. On top of the models discussed in the main text, we also present results for models with quantum circuits constrained to be translation invariant, which have a reduced parameter count. In general, those perform less well than the unconstrained models. In a few cases, the deformed models performed slightly worse than the undeformed ones because of the more complex optimization procedure.

To understand the practical benefit of the deformations studied, we also compare the results against classical baselines with increased number of parameters to match or slightly exceed the parameter count of the deformed layers. We start by noting that for a dense layer with  $n$  inputs and  $m$  outputs a classical probabilistic binary neural network baseline has  $mn + m$  parameters (weights plus bias). Recalling that a  $4 \times 4$  unitary can be parametrized in terms of its logarithm, an anti-Hermitian matrix with 16 real free parameters, for the quantum deformations considered we have the following ratio of deformed over classical parameters per dense layer:

$$[\text{PQ}] : \frac{mn(1 + 16 \times 2) + m}{mn + m} \sim 1 + 16 \times 2 = 33 \quad (50)$$

$$[\text{Q}] : \frac{mn(1 + 16) + m}{mn + m} \sim 1 + 16 = 17 \quad (51)$$

$$[\text{PQ T-inv}] : \frac{mn + m16 \times 2 + m}{mn + m} = 1 + \frac{32}{n + 1} \quad (52)$$

$$[\text{Q T-inv}] : \frac{mn + m16 + m}{mn + m} = 1 + \frac{16}{n + 1} \quad (53)$$

The  $\times 2$  accounts for  $P, Q$  and the deformation notation is explained in table 2. When considering the number of parameters, we note that in principle the parameterization of deformed layers is redundant. Indeed the weight parameters  $q_{ij}$  that make up the amplitudes of weights states  $|\psi_{2i-1}\rangle$ , and which correspond to the weights parameters of the classical baseline, could be incorporated into the definition of  $P, Q$  since those are generic unitaries. We however count them separately to reflect the parameterization used in our experiments.

Table 2: Test accuracies for MNIST and Fashion MNIST as function of parameters for various architectures. The notation  $cKsS - C$  indicates a conv2d layer with  $C$  filters of size  $[K, K]$  and stride  $S$ , and  $dN$  a dense layer with  $N$  output neurons. The deformations are:  $[/]$ :  $P_{i,i+1}^j = Q_{i,i+1}^j = \mathbf{1}$  (baseline Peters & Welling (2018)); [PQ]:  $P_{i,i+1}^j, Q_{i,i+1}^j$  generic; [PQ T-inv]:  $P_{i,i+1}^j = P^j, Q_{i,i+1}^j = Q^j$ ; [Q]:  $P_{i,i+1}^j = \mathbf{1}, Q_{i,i+1}^j$  generic; [Q] T-inv:  $P_{i,i+1}^j = \mathbf{1}, Q_{i,i+1}^j = Q^j$ .

Architecture	Deformation	# params	MNIST	Fashion MNIST
d10	$[/]$	7850	91.1	84.2
d10	[PQ]	258730	<b>94.3</b>	<b>86.8</b>
d10	[Q]	133290	91.6	85.1
d10	[PQ T-inv]	8170	91.9	84.3
d10	[Q T-inv]	8010	89.3	84.4
c3s2-8, c3s2-16, d10	$[/, /, /]$	7018	96.6	87.5
c3s2-8, c3s2-22, d10	$[/, /, /]$	9616	96.7	86.8
c3s2-9, c3s2-21, d10	$[/, /, /]$	9382	96.7	86.7
c3s2-10, c3s2-21, d10	$[/, /, /]$	9581	97.1	87.0
c3s2-12, c3s2-20, d10	$[/, /, /]$	9510	97.1	86.8
c3s2-8, c3s2-16, d10	[PQ, /, /]	9322	<b>97.6</b>	<b>88.1</b>
c3s2-8, c3s2-16, d10	[Q, /, /]	8170	96.8	87.8
c3s2-8, c3s2-16, d10	[PQ T-inv, /, /]	7274	97.4	87.8
c3s2-8, c3s2-16, d10	[Q T-inv, /, /]	7146	96.6	87.6
c3s2-32, c3s2-64, d10	$[/, /, /]$	41866	98.1	89.3
c3s2-43, c3s2-68, d10	$[/, /, /]$	51304	98.2	89.4
c3s2-44, c3s2-67, d10	$[/, /, /]$	51169	<b>98.4</b>	89.4
c3s2-48, c3s2-64, d10	$[/, /, /]$	51242	98.3	89.5
c3s2-32, c3s2-64, d10	[PQ, /, /]	51082	98.3	<b>89.6</b>
c3s2-32, c3s2-64, d10	[Q, /, /]	46474	98.3	89.5
c3s2-32, c3s2-64, d10	[PQ T-inv, /, /]	42890	98.2	89.13
c3s2-32, c3s2-64, d10	[Q T-inv, /, /]	42378	98.3	89.0