Quantum error reduction with deep neural network applied at the post-processing stage

Andrey Zhukov · Walter Pogosov

Received: 30 September 2021 / Accepted: 18 January 2022 / Published online: 18 February 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Deep neural networks (DNN) can be applied at the post-processing stage for the improvement of the results of quantum computations on noisy intermediate-scale quantum (NISQ) processors. Here, we propose a method based on this idea, which is most suitable for digital quantum simulation characterized by the periodic structure of quantum circuits consisting of Trotter steps. A key ingredient of our approach is that it does not require any data from a classical simulator at the training stage. The network is trained to transform data obtained from quantum hardware with artificially increased Trotter steps number (noise level) toward the data obtained without such an increase. The additional Trotter steps are fictitious, i.e., they contain negligibly small rotations and, in the absence of hardware imperfections, reduce essentially to the identity gates. This preserves, at the training stage, information about relevant quantum circuit features. Two particular examples are considered that are the dynamics of the transverse-field Ising chain and XY spin chain, which were implemented on two real five-qubit IBM Q processors. A significant error reduction is demonstrated as a result of the DNN application that allows us to effectively increase quantum circuit depth in terms of Trotter steps.

Keywords Quantum algorithms · Machine learning · Neural networks · Error mitigation · Ising chain · DNN · NISQ processors · Trotter evolution · XY spin chain · IBMQ
1 Introduction

Quantum information is a fast developing field that aims to utilize quantum properties, such as quantum interference and entanglement [1], to perform functions related to computing [2, 3], communication [4, 5], and simulation [6]. State-of-the-art quantum computers are already capable of solving many problems [7–11], which, however, are not of practical importance yet, because of relatively high quantum hardware error rates. Particularly, such processors can be useful for solving evolutionary problems, such as investigation of the quantum dynamics of the Fermi-Hubbard model, see, e.g., Refs. [12, 13]. However, the simulation of the dynamics of such systems at long times requires a large number of Trotter decomposition steps of evolution operator. This leads to the fact that a large number of quantum gates are required for simulation, which means that the outcomes from the quantum computer become too noisy [14, 15].

An attempt to enhance capabilities of quantum devices, from one hand, and machine learning methods, from another hand, has led to the merger of these areas [16–20], which gave rise to a new discipline known as quantum machine learning (QML). For example, hybrid quantum-classical systems were proposed [21], where QML algorithms work with both classical data from quantum devices and quantum algorithms on a quantum processor. Thus using machine learning [22, 23] to reduce the burden of classical information processing for quantum computation results has recently become an area of intense interest. The development of classical machine learning technologies can be useful for such tasks as the verification of quantum devices [24], quantum error correction [25–27], quantum control [28–31], quantum state classification [32, 33], and quantum state tomography [34–38]. For example, using deep machine learning in quantum tomography, it became possible to reduce errors in the preparation and measurement of quantum states [39] or predict if the set of measurements is informationally complete to uniquely reconstruct any given quantum state [40].

The main difficulty associated with neural networks in the context of noisy quantum machines is that it is necessary to have an access to the “ideal” data for their training. In principle, such data can be obtained from the classical simulation, but in the case of a large number of qubits, this becomes impossible. Methods that can be utilized to overcome this problem [41–43] include the usage of quantum circuits largely composed of Clifford gates that can be simulated classically. In our work, we address the problem of digital simulation of quantum dynamics and propose another method for obtaining quasi-ideal data for training using only a quantum computer, without a classical simulation. Our method also respects a periodic quantum circuit structure. As a source of quasi-ideal data, we consider outcomes from real quantum computer, which correspond to a limited number of Trotter steps. At the training stage, circuit depth is artificially increased by incorporation of certain number of “empty” Trotter steps, which contain zero or negligible rotation angles and thus reduce to identity gates for the ideal execution of quantum gates. The neural network is trained to transform such data with increased noise level toward their quasi-ideal counterpart. After being trained, the network is applied to improve new noisy data, which are obtained by increasing Trotter number, while none of Trotter blocks is fictitious. We demonstrate this rather generic method by studying the time evolution of the magnetization of the
transverse-field Ising spin chain and XY spin chain using IBM Q Athens and Bogota quantum processors, respectively. Both processors consist of five qubits and have a linear connectivity topology that matches spin chain topology.

We believe that our approach can be of particular interest in the context of digital simulation of different quantum systems. Indeed, such a simulation generally suffers from error accumulation problems, so that a very limited number of systems have been simulated so far. For example, Ref. [12] reports on simulation of Fermi–Hubbard model in a one-dimensional geometry using state-of-the-art superconducting quantum processor, while much more interesting simulation of the same model in two dimensions remains challenging. In order to reduce errors, several techniques have been used [12], which allowed to increase Trotter step number and thus to trace the system’s dynamics at a longer time. The application of neural networks at the post-processing stage is a prospective tool that can be used in combination with other techniques.

The article is organized as follows. Section 2 deals with the formulation of the problem of the dynamics of the Ising spin chain in a transverse field. Section 3 details our method for reducing hardware errors using deep machine learning. Section 4 describes the practical implementation of the method on a real processor for the transverse-field Ising model. In Sect. 5, we apply our method to another system, which is XY spin chain. Section 6 presents Conclusions.

2 Trotterization of quantum dynamics

Our approach can be applied for the simulation of various quantum many-body systems, but in the beginning we are going to illustrate it in a detail with a paradigmatic example of a system that is widely used for benchmarking state-of-the-art quantum machines and their ability to work as digital, analog, or digital-analog [15] quantum simulators. This is the transverse-field Ising model, for which we choose a one-dimensional geometry because it perfectly matches the topology of a five-qubit IBM Q Athens processor, which has one of the lowest gate error rates among state-of-the-art quantum processors, as well as one of the largest quantum volumes, and therefore it is suitable for the demonstration of our concept, which requires quantum circuits of a rather large depth. In Sect. 5, we consider another example, which is XY spin chain. The transverse-field Ising Hamiltonian reads as

$$H = - \sum_j h_j \sigma_j^x - \sum_{<ij>} J_{ij} \sigma_i^z \sigma_j^z,$$

where $\sigma_j$ are Pauli operators acting in the space of $j$’s spin, represented directly by $j$’s qubit of the chain, $h_j$ are local fields, and $J_{ij}$ are coupling constants which are supposed to be nonzero only for nearest neighbors.

One of the standard approaches to simulate quantum systems on digital quantum processor is based on Trotter decomposition of the evolution operator. We are going to concentrate on the dynamics of magnetic characteristics. In the case of the Hamiltonian $H$ consisting of two non-commuting contributions, $H = H_A + H_B$, Trotter decomposition transforms the evolution operator for $H$ into the sequence of evolution
operators for $H_A$ and $H_B$

$$e^{-i(tH_A + H_B)} \approx \left(e^{-i H_A \frac{t}{N}} e^{-i H_B \frac{t}{N}}\right)^N,$$ \(2\)

which is asymptotically exact in the limit of large Trotter step number $N \rightarrow \infty$; otherwise, Trotterization produces digitization (discretization) error. For the transverse-field Ising Hamiltonian, the appropriate splitting of the full Hamiltonian is

$$H_A = -\sum_j h_j \sigma_j^x,$$ \(3\)

$$H_B = -\sum_{<ij>} J_{ij} \sigma_i^z \sigma_j^z,$$ \(4\)

so that the evolution operators for both $H_A$ and $H_B$ can be further split exactly into evolution operators for each $-h_j \sigma_j^x$ and $-J_{ij} \sigma_i^z \sigma_j^z$ terms, since these contributions do commute to each other. All these operators can be implemented in real quantum computers in a standard way using quantum circuits language.

Let us concentrate on the quantum circuit for $N_q = 5$ transverse-field Ising chain. We will present our results for the case, when all coupling constants and free fields for spins of the chain are equal to each other, $J_{ij} = J = 2$, $h_j = h = 1$. Our quantum circuit corresponding to the single Trotter step for the time interval $\delta t = t/N$ is shown in Fig. 1, where single-qubit rotations are defined as $R_x(\varphi_1) = e^{-i \frac{\varphi_1}{2} X}$ and $R_z(\varphi_2) = e^{-i \frac{\varphi_2}{2} Z}$, where $\varphi_1 = 2h \delta t$ and $\varphi_2 = 2J \delta t$. Hereafter, $X$, $Y$, and $Z$ refer to Pauli gates.

### 3 Error reduction with DNN

First of all, neural network needs to be trained. Imagine that we have reliable, i.e., noise-free data corresponding to the outcomes from an ideal quantum computer. They can be represented in a classical form by some characteristics of a quantum state, i.e., quantum mean values or correlators, that does not require a full quantum tomography. Similar data but affected by hardware imperfections can be obtained as outcomes from a noisy quantum computer. Next, the neural network can be trained to transform noisy data toward ideal results. After being trained, the network can be applied to other noisy data, which were not included in the initial training data set, but were affected...
Quantum error reduction with deep neural network applied…

Fig. 2  Schematic view of our approach. It consists of three steps: generation of quasi-ideal data with relatively shallow circuit (a); training the DNN—the data with artificially increased Trotter steps number are transformed toward quasi-ideal data (b); the trained DNN is applied to raw experimental data with the same Trotter step number as at the second stage (c).

by nearly the same amount of noise. The most attractive potential application of this rather general idea is an effective increase in an accessible quantum circuit length (for example, increase of Trotter step number) due to the improvement of data quality at the post-processing stage.

This general approach requires an “ideal” dataset to train the neural network. In principle, such a dataset can be obtained using a classical simulator. However, in the case of large quantum processors with many qubits, classical simulation will be possible only in some special cases. Thus, it might seem that an absence of an ideal training dataset makes this idea useless for real applications using quantum machines with many qubits.

Let us, however, assume that we have a reliable enough low-noise quantum computer. Let our task be to analyze the digitized evolution of the quantum system for some time interval \((0, T)\) taking into account as large number of Trotter steps as possible. To this end, we will proceed as follows. Let us execute the algorithm for \(N_1\) Trotter steps for the total evolution time \(T\), as shown schematically in Fig. 2a for \(N_1 = 2\), where \(\text{Tr}(\delta t)\) stands for the evolution operator consisting of a single Trotter step of length \(\delta t\). Each Trotter step has a length of \(T / N_1\) at this stage. The Trotter steps number \(N_1\) must be chosen in such a way as to keep the total gates error small. Such data generated by the quantum computer can be considered as quasi-ideal in the sense that errors due to hardware imperfections are small. Note that, in practical situations, optimal \(N_1\) can be estimated from the balance between total gate error, which increases as circuit depth grows, and Trotterization error, which decreases as circuit depth grows. Optimal \(N_1\) thus can be found from the condition of minimum of the total error.

Next, let us consider the method of training of the neural network using quasi-ideal data. Imagine that our goal is to increase the total Trotter step number from \(N_1\) to \(N_2 = cN_1\) for the same time interval \((0, T)\), where \(c\) is a natural number. Apparently,
the level of noise will be increased in this case and this must be taken into account at the training stage. The training can be performed as follows. We extend the quantum circuit for \( N_1 \) Trotter steps and the total evolution time \( T \) by adding \( N_2 - N_1 \) “empty” Trotter steps, as illustrated in Fig. 2b. These fictitious Trotter steps must include rotations on zero or negligibly small angles making each ideal “empty” Trotter block essentially equivalent to the identity gate. For the perfect execution of the quantum algorithm, the new data must coincide with the quasi-ideal data, but due to the increased number of noisy quantum gates, they do differ. The neural network is to be trained to transform these noisy data toward quasi-ideal data. After the training, we can run the quantum computer for \( N_2 \) Trotter steps and total evolution time \( T \), such that each Trotter step has a length \( T / N_2 \), see Fig. 2c. The quantum gate number as well as the structure of the quantum circuit and its composition in terms of quantum gates are similar to the case with \( N_1 \) “real” and \( N_2 - N_1 \) “empty” Trotter steps for which the neural network has been trained (\( N_2 \) Trotter blocks in total in both cases). Let us stress that it is known that circuit structure is of importance since it directly influences quantum computer performance [44]. Finally, at the post-processing stage, we apply the trained network for these last data, as illustrated in Fig. 2c, that allows to partially get rid of noise effects.

There are certain analogies between our approach and zero-noise extrapolation technique [45]. The latter is based on the artificial addition of different numbers of identity gates to the actual quantum circuit. These identity gates are composed of standard quantum gates, for instance from CNOTs, which have the largest error rates in state-of-the-art IBM quantum computers, see, e.g., Ref. [46]. In the absence of gate errors, the results of quantum computation corresponding to the considered set of quantum circuits must be the same; however, gate errors make them distinct. Addition of identity gates makes the results more and more noisy. Using the corresponding set of results, it is possible to extrapolate the result to the zero-noise limit utilizing a power series expansion of the target quantity in a noise parameter. In contrast, in our method, we do not add identity gates to the actual circuit. Instead, we consider shorter circuits to which we add noisy identity gates in order to artificially increase circuit depth making it similar to the depth of the actual circuit. This implies that we can consider circuits of larger depths since the addition of (noisy) identity gates would make the results even more noisy and the corresponding dependencies on time quite flat, i.e., difficult to analyze and to recover any valuable information. In addition, the final transformation is made automatically by the trained neural network rather than by addressing a series expansion in a noise parameter.

Note that there can be different permutations between positions of “real” and “empty” Trotter blocks in the quantum circuit at the training stage. This leads to the variety of different schemes, the efficiency of which is worth to be studied. Also note that our approach does not use ideal data at the training stage, and therefore, it is hard to expect that the post-processing based on the DNN application would demonstrate higher accuracy compared to quasi-ideal data with \( N_1 \) Trotter steps, which are influenced by gate errors.
Fig. 3 The feed-forward DNN architecture used in our problems for improving simulation results. Input layer consists of $K_{in}$ neurons. Output layer contains $K_{out}$ neurons. Two hidden layers consist of 200 neurons both. We use sigmoid activation function after both hidden and output layers. During training, the neural network changes its internal parameters in order to achieve a better mapping of the input noisy data to the quasi-ideal counterpart.

4 Magnetic properties of transverse-field Ising chain

Now we apply our idea for the digital quantum simulation of 5-spin transverse-field Ising chain using IBM Q Athens quantum processor. We provide an illustrative example of a neural network, which is trained to improve noisy data for the dynamics of magnetization of each spin in $z$ direction, which we define as $\langle M_j^z \rangle = (1 - \langle Z_j \rangle)/2$, where $j = 1 \ldots 5$. We utilize a simple feed-forward deep neural network whose architecture is shown in Fig. 3. The neural network performs the transformation of input data $x^{in}$ into output data $x^{out}$ defined as

$$\begin{align*}
x^1 &= \sigma \left( W^{01} x^{in} + b^1 \right), \\
x^2 &= \sigma \left( W^{12} x^1 + b^2 \right), \\
x^{out} &= \sigma \left( W^{23} x^2 + b^{out} \right),
\end{align*}$$

(5)

where $\sigma(z) = (1 + \exp(-z))^{-1}$ is a sigmoid activation function, $W^{ij}$ is weight matrix between layers $i$ and $j$, and $b^i$ is bias on layer $i$. The number of input neurons is chosen to be $K_{in} = 15$, which corresponds to the magnetization of each spin in three directions $x$, $y$, and $z$. DNN has two hidden layers consisting of 200 neurons both. The output is a 5-dimensional layer representing magnetization of each spin in the $z$ direction only since we have chosen magnetization in this direction as a target quantity. Note that, of course, the number of input and output neurons can be different from the case considered here since different characteristics of the spin system can be used in datasets, so the neural network shown in Fig. 3 is just a specific example. In Sect. 5, a similar network will be used for XY spin chain, but with $K_{in} = K_{out} = 5$ that corresponds to magnetizations only in $z$ direction, since our method will be supplemented by the post-selection of outcomes according to the excitation-number conservation condition.

For a given pair $(N_1, c)$, a set of curves for $\langle X_j \rangle$, $\langle Y_j \rangle$, $\langle Z_j \rangle$ as functions of time is obtained from real quantum computer for all possible initial conditions, which
correspond to the computational basis (such as \( |01101\rangle, |10110\rangle \), etc.). The same DNN is thus trained for all these initial conditions, and it is able to improve raw data for all of them. The number of such initial states is \( 2^{N_q} \), where \( N_q \) is the number of the qubits \((N_q = 5 \text{ in our example})\). As a whole, for a given Trotter step number, the quantum computer generates a quasi-ideal dataset of the size \( D = 2^{N_q} \times N_q \times K \times B \), where \( K \) is the number of points along a given time interval \((0, T)\) and \( B = 3 \) is the number of characteristics for each spin we are interested in (magnetization in \( x \), \( y \), and \( z \) directions in our example). As usual in the IBM Q system, 8192 measurements for each point are performed. For our DNNs, we tested different activation functions such as the sigmoid or ReLu and found that the sigmoid function yielded the best results.

The quality of data improvement is estimated by the comparison between quasi-ideal data and data improved by DNN for the whole time interval \( T \). The quality criterion for DNN is the mean square error (MSE), defined as

\[
E_{A,B} = \frac{1}{2D} \sum_{i,j,k,l} \left( \langle M_{k,l}^{j,l}(t_i) \rangle_A - \langle M_{k,l}^{j,l} \rangle_B \right)^2,
\]

which depends on difference between training dataset \((A)\) and quasi-ideal data \((B)\). Here \( M_{k,l}^{j,l}(t_i) \) is the value of magnetization of spin \( j \) along axes \( k = x, y, z \) for time \( t_i \) and initial state \( l \). Training is performed by minimization the MSE. For this, we applied the Adam algorithm [47]. The network was trained during 50,000 epochs. Network parameters were saved for every 100 epochs in order to find optimal training time. The networks are trained separately for each couple of parameters \( N_1 \) and \( c \). In our examples, we focus on \( N_1 = 2 \) and \( c = 2, 3, 4 \) (three DNNs in total). All data for training and testing the neural network were obtained using the real quantum processor. In our illustrative examples, the value of \( T = 2/J \) is taken to be several times larger than the typical maximum time for which the approximation with \( N_1 = 2 \) has a negligibly small digitization error. Of course, such a small value of \( N_1 \) as \( N_1 = 2 \) is meaningful only for demonstration purposes. An increase in \( N_1 \) leads to the enhancement of hardware error rates which become too high for our demonstration. Three DNNs are trained in total, each corresponding to a given \( c \).

Our results for errors \( E_{A,B} \) at \( k = z \) are presented in Fig. 4. This figure shows the MSE (6) between both raw data and data improved by DNN (A) and exact data for the same Trotter step number (B). Averaging over all initial conditions is performed. It
Fig. 5  The mean spin chain magnetization along the $Z$ axis as a function of time for the initial state $|00000\rangle$. Solid curves are the results of ideal simulations for given Trotter step number $N_1 = 2$. Raw experimental data for magnetization (x-shape symbols) were obtained on the real 5-qubit quantum processor IBM Athens. The results of applying the neural network are shown with o-shape symbols. The neural network was trained using ideal data from the simulator (see the text). The dotted line shows the reference ideal evolution of magnetization free of Trotterization errors. For all curves $J = 2, h = 1$

is seen from this figure that the improvement due to the neural network is significant generally decreasing MSE in several times. In order to demonstrate the ability of the DNN to improve the dynamics, in Fig. 5 we show the mean magnetization of 5 spins $\langle M_j^z \rangle$ averaged over spins $j$ as a function of time for the initial state $|00000\rangle$. It presents the results of improvement for quasi-ideal data corresponding to $N_1 = 2$. For illustrative purposes, in this particular case, DNN was trained using ideal data from the simulator with the same Trotter step number. This demonstration provides an important test—Fig. 5 shows that the DNN is able to transform raw data essentially to the ideal counterpart. The figure shows four curves: the result of the ideal simulation with two Trotter steps (solid curve), the exact solution corresponding to the Schrodinger equation in the absence of Trotterization errors (dotted curve), raw data (x-shaped symbols), and data improved by DNN (circle-shaped symbols). In other words, we used the algorithm illustrated in Fig. 2a to obtain both ideal (from the simulator) and training data (from the quantum processor) for training the neural network. Steps illustrated by Fig. 2b, c are not used in this demonstration. Figure 5 reveals a nearly perfect matching between ideal simulation and data improved by DNN. A similar level of improvement has been achieved for all other initial conditions considered.

Next, the improved results for $N_1 = 2$ and $c = 2, 3, 4$ are compared to the ideal results free of hardware errors, which include the results of the full solution of the time-dependent Schrodinger equation corresponding to the infinite Trotter steps numbers. In Fig. 6, we plot such time dependencies of $\langle M_j^z \rangle$ averaged over spins $j$, which are referred to as “exact results’, for the initial state $|00000\rangle$ (dotted lines). Also shown are the ideal results for a given Trotter step number (solid line), referred to as “ideal simulation”, raw results (x-shaped symbols) as well as the improved results (o-shaped symbols), which are plotted at three values of $c$: 2 (a), 3 (b), 4 (c). Figure 7 shows deviations between the results improved $\langle M_j^z \rangle_{DNN}$ and ideal results with the same Trotter step number $\langle M_j^z \rangle_{Trott}$, $\Delta M_{Trott} = \langle M_j^z \rangle_{DNN} - \langle M_j^z \rangle_{Trott}$ (a), as well as
Fig. 6 The mean spin chain magnetization along the Z axis as a function of time for the initial state $|00000\rangle$. Solid curves are the results of ideal simulations for given Trotter steps number $N_2$: 4 (a), 6 (b), 8 (c). Experimental data for magnetization (x-shape symbols) were obtained on the real 5-qubit quantum processor IBM Athens. The results of applying the neural network are shown with o-shape symbols. For each case, the neural network was trained using experimental data for $N_1 = 2$ and $c = 2, 3, 4$, respectively (see the text). The dotted line shows the exact reference evolution of magnetization free of Trotterization errors. In all cases $J = 2$, $h = 1$.

Fig. 7 Errors after DNN application as a function of time for the initial state $|00000\rangle$—a difference between improved results and ideal simulation for the same Trotter step number; b difference between improved results and exact results free of Trotterization error.

between the results of the full solution of Schrodinger equation $\langle M_z^j \rangle_{\text{exact}}$, $\Delta M_{\text{exact}} = \langle M_z^j \rangle_{\text{DNN}} - \langle M_z^j \rangle_{\text{exact}}$ (b).

In general, it is seen from Figs. 6c and 7b that DNN brings the raw results closer to the ideal curves for all values of c we considered. For example, raw data for $N_1 = N_2 = 2$, see Fig. 5, were reasonably close to the exact results free from both digitization and hardware errors up to the maximum time $t \lesssim 0.5$. The DNN extended this range to $t \lesssim 1.0$ for $N_2 = 4$ and $t \lesssim 1.5$ for $N_2 = 6$. The extension of the time interval, when there exists a good agreement between the results of the full solution of the time-dependent Schrodinger equation and the outcomes from the real quantum computer enhanced by DNN, is one of the key results of our proof-of-principle demonstration.

Let us also note that our approach typically leads to the less efficient improvement of the data quality at short times compared to the improvement averaged over the whole interval of time. We believe that this artifact is not crucial since quasi-ideal data turn out to be accurate in this range of parameters so that they can be treated as reference results generated by the quantum computer.
More advanced DNN architectures together with different types of input characteristics of the quantum state as well as the incorporation of other error mitigation techniques can further enhance the abilities of DNN at the post-processing stage. This requires an interplay between the physics of the underlying model we simulate and DNN architecture. Another improvement can be due to the more advanced Trotterization that accounts for higher orders of Trotter step length.

Note that we expect that, in general, for the successful training of a neural network, there is no need for data corresponding to the full set of initial states of the computational basis. All that is needed is the information about the dynamics for various initial states that correspond to different patterns of the system’s dynamics. Therefore, as it is expected, there will be no need to include all $2^{N_q}$ possible initial states to the training set that is crucial for large $N_q$. It is possible to apply, for example, Monte Carlo methods when choosing an initial state and stop training the neural network until the level of errors becomes acceptable, or the addition of new data ceases to improve results.

5 Magnetic properties of XY spin chain

In this section, we consider another quantum system, which is XY quantum chain, and apply our method for this particular case. The system considered is described by the Hamiltonian

$$H = -\hbar \sum_j \sigma_j^z - J \sum_{<ij>} \left( \sigma_i^x \sigma_j^x + \sigma_i^y \sigma_j^y \right),$$  \hspace{1cm} (7)

where the interaction acts between nearest neighbors in the chain. We concentrate on the chain consisting on five spins, which is simulated using five-qubit quantum IBM Q processor Bogota with linear connectivity. Quantum circuit for this chain corresponding to the structure of the single Trotter step is shown in Fig. 8, where $\varphi_1 = 2\hbar \delta t$ and $\varphi_2 = 2J \delta t$. We apply the same method by improving the results of the quantum simulation using the neural network at the post-processing stage, as illustrated in Figs. 2 and 3.

As a dataset for the neural network, we use magnetizations in $z$ direction only, $\langle M_j^z \rangle$. Thus, the parameters of the neural network, shown in Fig. 3, are $K_{in} = K_{out} = 5$. The Hamiltonian conserves the excitation number, and this allows us to use a post-selection as an additional tool to mitigate errors: at the training stage, as a quasi-ideal data we use the experimental data for $N_1 = 2$, for which we discard the data with
wrong excitation numbers. This allows us to noticeably improve the quality of the quasi-ideal dataset.

Figure 9 shows MSE between ideal simulation data for a given Trotter step number at \( k = z \) in Eq. (6) and experimental data improved by the network with post-selection (o-shape symbols) and without post-selection (triangle-shape symbols) as well as raw data (x-shape symbols and dashed line). It is seen from this figure that neural network is able to significantly improve the quality of the data by decreasing MSE in several times for \( N_2 = 4 \) and 6. Post-selection leads to a further improvement. In the case for \( N_2 = 8 \), there is almost no improvement in the results due to the fact that the IBM Bogota quantum processor used has too high error rate (CNOT errors \( \approx 2\% \)) to enable quantum simulation with so long quantum circuit depth. As a result, the training data for \( N_2 = 8 \) is very noisy so that the magnetization as a function of time is flat, since the probability of each qubit to be in the state 1 is close to 0.5 for any time. Such a data can not be used effectively to train a neural network.

In order to illustrate the dynamics, we focus on the initial conditions, which are characterized by the domain wall between polarized and unpolarized spins, such as \( |11100\rangle \), and consider a quantity known as half-difference of spin magnetization defined as

\[
d = \sum_{j \in S_{\text{up}}^{\text{init}}} \frac{1}{N_{\text{up}}^{\text{init}}} m_j - \sum_{j \in S_{\text{down}}^{\text{init}}} \frac{1}{N_{\text{down}}^{\text{init}}} m_j,
\]

where \( m_j = 2n_j - 1 \) is \( j \)-th spin magnetization, \( S_{\text{up}}^{\text{init}} \) and \( S_{\text{down}}^{\text{init}} \) are sets of spins with initial state \( |1\rangle \) and \( |0\rangle \), while \( N_{\text{up}}^{\text{init}} \) and \( N_{\text{down}}^{\text{init}} \) are their numbers, correspondingly. The half-difference of spin magnetization characterizes the robustness of the initial spin pattern (the domain wall in our case) to the interaction between spins during the free evolution. The dynamics of this quantity is shown in Fig. 10 for the initial state \( |11100\rangle \) and different \( N_2 \). In this particular case, e.g., \( S_{\text{up}}^{\text{init}} = \{1, 2, 3\} \), \( S_{\text{down}}^{\text{init}} = \{4, 5\} \), and \( N_{\text{up}}^{\text{init}} = 3 \), \( N_{\text{down}}^{\text{init}} = 2 \). The post-selection technique was used at the training stage. From this figure, we see an improvement of the data quality due to the neural network.
Fig. 10. The half-difference of spin magnetization along the $z$ axis as a function of time for the initial state $|11100⟩$. Solid curves are the results of ideal simulations for given Trotter steps number $N_2$: 4 (a), 6 (b), 8 (c). Experimental data for half-difference of spin magnetization (x-shape symbols) were obtained on the real 5-qubit quantum processor IBM Bogota. The results of applying the neural network are shown with o-shape symbols. For each case, the neural network was trained using experimental data for $N_1 = 2$ and $c = 2, 3, 4$, respectively (see the text). The dotted line shows the exact reference evolution of half-difference of spin magnetization free of Trotterization errors. In all cases $J = 2$, $h = 1$.

Fig. 11. Errors after DNN application as a function of time for the initial state $|11100⟩$—a difference between improved results and ideal simulation for the same Trotter step number; b difference between improved results and exact results free of Trotterization error.

Figure 11 shows deviations of the improved results $⟨d⟩_{DNN}$ from the ideal results with the same Trotter step number $⟨d⟩_{Trot}$, $Δd_{Trot} = ⟨d⟩_{DNN} − ⟨d⟩_{Trot}$ (a) as well as of the results of the full solution of Schrodinger equation $⟨d⟩_{exact}$ $Δd_{exact} = ⟨d⟩_{DNN} − ⟨d⟩_{exact}$ (b) as a function of time for the initial state $|11100⟩$. There is an improvement of the dynamics for all values of $N_2$ considered. The comparison with the exact result free of trotterization errors presented in Fig. 11b shows that the deviation from this result generally grows with time for all values of $N_2$ considered that is qualitatively similar to the behavior revealed for the transverse-field Ising chain depicted in Fig. 7b.

6 Conclusions

In this article, we have proposed a method for the application of classical neural networks for the improvement of the outcomes of noisy quantum computers at the post-processing stage. In contrast to other suggestions, using our approach, it is possible to get data for training a neural network without relying on a classical simulator or...
any other source of ideal data. The approach is most suitable to the very important problem of digital simulation of quantum dynamics, for which a limitation in accessible Trotter steps numbers is currently crucial. The limitation is associated with the error accumulation due to the hardware imperfections.

Our method is based on artificial increase of the quantum circuits depth on the training stage that can be done by incorporation of fictitious Trotter blocks formally equivalent to identity gates into the circuit. Their role is to increase noise level due to the hardware imperfections while preserving the circuit’s general structure and its relevant features. The network is trained to transform data obtained with such fictitious steps toward data obtained without them, that is, for rather shallow circuits, for which hardware errors are not critical. This idea seems to be more prospective for near-term generations of quantum computers with reduced gate errors, for which circuits at the training stage can already support large entanglement.

After being trained, the network can be applied to new data with the same Trotter step number, i.e., increased in the same way as at the training stage, but without fictitious Trotter steps. The amount of noise in this case is similar to that at the training stage. This trick allows for the effective increase in the Trotter number due to the post-processing, in the sense that errors become suppressed and results of simulations, which must have error rates below a given level, start to include data with larger Trotter step number.

Our method does not require a complete tomography of quantum states, which allows it to be scaled. The reason is that the network is trained to improve the data for a restricted number of quantum mean values such as spins magnetizations along different axes, order parameters, or characteristic correlators.

We have demonstrated the basic ingredients of our approach using two examples: digital quantum simulations of the dynamics of the transverse-field Ising chain and XY chain. Deep neural network with simple architectures was used at the post-processing stage. For XY chain, an additional post-selection of the results at the training stage was applied by discarding a part of the data, which does not conserve the excitation number, as required by the Hamiltonian. The proof-of-principle results obtained on a real 5-qubit IBM Athens and Bogota quantum processors show that our method allows us to increase the number of Trotter steps while maintaining the same level of errors. The significant error reduction is the main result of our demonstration. A single neural network is able to improve the data for different initial conditions.

We believe that the proposed approach can be useful in the context of error mitigation in noisy quantum devices (especially of next generations with hardware errors decreased and qubit number increased). Particularly, it can be used in the case of periodic quantum circuits and in combination with other error reduction tools, such as post-selection or partial error correction.

Acknowledgements  We acknowledge use of the IBM Quantum Experience for this work. The viewpoints expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum Experience team. A. A. Zh. acknowledges a support from RFBR (project no. 20-37-70028). W. V. P. acknowledges a support from RFBR (project no. 19-02-00421).
Data availability Raw data for this study were generated by running quantum circuits via QISKIT on real quantum processors IBM Athens and Bogota or simulator. The datasets generated during the current study are available from the corresponding author on request.

References

1. Horodecki, R., Horodecki, P., Horodecki, M., Horodecki, K.: Quantum entanglement. Rev. Mod. Phys. 81(2), 865 (2009)
2. Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J.M., Gambetta, J.M.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. Nature 549(7671), 242–246 (2017)
3. Barends, R., Lamata, L., Kelly, J., García-Álvarez, L., Fowler, A.G., Megrant, A., Jeffrey, E., White, T.C., Sank, D., Mutus, J.Y.: Digital quantum simulation of fermionic models with a superconducting circuit. Nat. Commun. 6(1), 1–7 (2015)
4. Mattle, K., Weinfurter, H., Kvit, P.G., Zeilinger, A.: Dense coding in experimental quantum communication. Phys. Rev. Lett. 76(25), 4656 (1996)
5. Zhukov, A.A., Kiktenko, E.O., Elistratov, A.A., Pogosov, W.V., Lozovik, Y.E.: Quantum communication protocols as a benchmark for programmable quantum computers. Quantum Inf. Proc. 18(1), 1–23 (2019)
6. Georgescu, I.M., Ashhab, S., Nori, F.: Quantum simulation. Rev. Mod. Phys. 86(1), 153 (2014)
7. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., O’Brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nat. Commun. 5(1), 1–7 (2014)
8. O’Malley, P.J., Babbush, R., Kivlichan, I.D., Romero, J., McClean, J.R., Barends, R., Kelly, J., Roushan, P., Tranter, A., Ding, N.: Scalable quantum simulation of molecular energies. Phys. Rev. X 6(3), 031007 (2016)
9. Mohseni, M., Read, P., Neven, H., Boixo, S., Denchev, V., Babbush, R., Fowler, A., Smelyanskiy, V., Martinis, J.: Commercialize quantum technologies in five years. Nat. News 543(7644), 171 (2017)
10. Li, Y., Benjamin, S.C.: Efficient variational quantum simulator incorporating active error minimization. Phys. Rev. X 7(2), 021050 (2017)
11. Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum 2, 79 (2018)
12. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Bengtsson, A., Boixo, S., Broughton, M., Buckley, B.B., Buell, D.A., Burkett, B., Bushnell, N., Chen, Y., Chen, Z., Chen, Y.-A., Chiaro, B., Collins, R., Cotton, S.J., Courtney, W., Demura, S., Derk, A., Dunningworth, A., Eppens, D., Eckl, T., Erickson, C., Farhi, E., Fowler, A., Foxen, B., Gidney, C., Giustina, M., Graff, R., Gross, J.A., Habetger, S., Harrigan, M.P., Ho, A., Hong, S., Huang, T., Huggins, W., Ioffe, L.B., Isakov, S.V., Jeffrey, E., Jiang, Z., Jones, C., Kafri, D., Kechedzhi, K., Kelly, J., Kim, S., Klimov, P.V., Korotkova, A.N., Kostritsa, F., Landhuis, D., Laptev, P., Lindmark, M., Lucero, E., Marthaler, M., Martin, O., Martinis, J.M., Marusczyk, A., McArdle, S., McClean, J.R., McCourt, T., McEwen, M., Megrant, A., Mejuto-Zaera, C., Mi, X., Mohseni, M., Mruczkiewicz, W., Mutus, J., Naaman, O., Neeley, M., Neill, C., Neven, H., Newman, M., Niu, M.Y., O’Brien, T.E., Ostby, E., Pató, B., Petukhov, A., Puttnaman, H., Quintana, C., Reiner, J.M., Roushan, P., Rubin, N.C., Sank, D., Satzinger, K.J., Smelyanskiy, V., Strain, D., Sung, K.J., Schmitteckert, P., Szalay, M., Tschumper, N., Vainsencher, A., White, T., Vogt, N., Yao, Z.J., Yeh, P., Zalcman, A., Zanker, S.: Observation of separated dynamics of charge and spin in the Fermi-Hubbard model (2020)
13. Cappellaro, P., Viola, L., Ramanathan, C.: Coherent-state transfer via highly mixed quantum spin chains. Phys. Rev. A 83, 032304 (2011)
14. Zhukov, A.A., Remizov, S.V., Pogosov, W.V., Lozovik, Y.E.: Algorithmic simulation of far-from-equilibrium dynamics using quantum computer. Quantum Inf. Proc. 17(9), 1–26 (2018)
15. Babukhin, D.V., Zhukov, A.A., Pogosov, W.V.: Hybrid digital-analog simulation of many-body dynamics with superconducting qubits. Phys. Rev. A 101, 052337 (2020)
16. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature 549(7671), 195–202 (2017)
17. Dunjko, V., Briegel, H.J.: Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Rep. Prog. Phys. 81(7), 074001 (2018)
18. Perdomo-Ortiz, A., Benedetti, M., Realpe-Gómez, J., Biswas, R.: Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. Quantum Sci. Technol. 3(3), 030502 (2018)

19. Ciliberto, C., Herbster, M., Ialongo, A.D., Pontil, M., Rocchetto, A., Severini, S., Wossnig, L.: Quantum machine learning: a classical perspective. Proc. Math. Phys. Eng. Sci. 474(2209), 20170551 (2018)

20. Schuld, M., Sinayskiy, I., Petruccione, F.: An introduction to quantum machine learning. Contemp. Phys. 56(2), 172–185 (2015)

21. Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M.: Parameterized quantum circuits as machine learning models. Quantum Sci. Technol. 4(4), 043001 (2019)

22. Nielsen, M.A.: Neural Networks and Deep Learning, vol. 25 (2015)

23. Carleo, G., Cirac, I., Cramer, N., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., Zdeborová, L.: Machine learning and the physical sciences. Rev. Mod. Phys. 91(4), 045002 (2019)

24. Lennon, D., Moon, H., Camenzind, L., Yu, L., Zumbühl, D., Briggs, G., Osborne, M., Laird, E., Ares, N.: Efficiently measuring a quantum device using machine learning. Npj Quantum Inf. 5(1), 1–8 (2019)

25. Nautrup, H.P., Delfosse, N., Dunjko, V., Briegel, H.J., Friis, N.: Optimizing quantum error correction codes with reinforcement learning. Quantum 3, 215 (2019)

26. Baireuther, P., O’Brien, T.E., Tarasinski, B., Beenakker, C.W.J.: Machine-learning-assisted correction of correlated qubit errors in a topological code. Quantum 2, 48 (2018)

27. Andreasson, P., Johansson, J., Liljestrand, S., Granath, M.: Quantum error correction for the toric code using deep reinforcement learning. Quantum 3, 183 (2019)

28. Kalantre, S.S., Zwolak, J.P., Ragole, S., Wu, X., Zimmerman, N.M., Stewart, M.D., Taylor, J.M.: Machine learning techniques for state recognition and auto-tuning in quantum dots. Npj Quantum Inf. 5(1), 1–10 (2019)

29. Baireuther, P., O’Brien, T.E., Tarasinski, B., Beenakker, C.W.J.: Machine-learning-assisted correction of correlated qubit errors in a topological code. Quantum 2, 48 (2018)

30. Neugebauer, M., Fischer, L., Jäger, A., Czischek, S., Jochim, S., Weidemüller, M., Gärttner, M.: Neural-network quantum state tomography in a two-qubit experiment. Phys. Rev. A 102(4), 042604 (2020)

31. Czarnik, P., Arrasmith, A., Coles, P.J., Cincio, L.: Error mitigation with clifford quantum-circuit data. arXiv:2005.10189 (2020)

32. Strikis, A., Qin, D., Chen, Y., Benjamin, S.C., Li, Y.: Learning-based quantum error mitigation. arXiv:2005.07601 (2020)

33. Kim, C., Park, K.D., Rhee, J.-K.: Quantum error mitigation with artificial neural network. IEEE Access 8, 188853–188860 (2020)

34. Proctor, T., Rudinger, K., Young, K., Nielsen, E., Blume-Kohout, R.: Measuring the Capabilities of Quantum Computers (2020)
45. Temme, K., Bravyi, S., Gambetta, J.M.: Error mitigation for short-depth quantum circuits. Phys. Rev. Lett. 119, 180509 (2017)
46. Fauseweh, B., Zhu, J.-X.: Digital quantum simulation of non-equilibrium quantum many-body systems. Quantum Inf. Proc. 20, 138 (2021)
47. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization (2017)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.