

Artificial intelligence and machine learning for quantum technologiesMario Krenn,^{1,*} Jonas Landgraf,^{1,2} Thomas Foesel,^{1,2} and Florian Marquardt^{1,2}¹Max Planck Institute for the Science of Light, Erlangen 91058, Germany²Department of Physics, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen 91054, Germany

(Received 7 August 2022; published 3 January 2023)

In recent years the dramatic progress in machine learning has begun to impact many areas of science and technology significantly. In the present perspective article, we explore how quantum technologies are benefiting from this revolution. We showcase in illustrative examples how scientists in the past few years have started to use machine learning and more broadly methods of artificial intelligence to analyze quantum measurements, estimate the parameters of quantum devices, discover new quantum experimental setups, protocols, and feedback strategies, and generally improve aspects of quantum computing, quantum communication, and quantum simulation. We highlight open challenges and future possibilities and conclude with some speculative visions for the next decade.

DOI: [10.1103/PhysRevA.107.010101](https://doi.org/10.1103/PhysRevA.107.010101)**I. INTRODUCTION**

The fields of machine learning [1–3] and quantum technologies [4–7] have much in common: Both started out with an amazing vision of applications (in the 1950s and 1980s, respectively), went through a series of challenges, and are currently extremely hot research topics. Of these two, machine learning has firmly taken hold beyond academia and beyond prototypes, triggering a revolution in technological applications during the past decade. This perspective article will be concerned with shining a spotlight on how techniques of classical machine learning (ML) and artificial intelligence (AI) hold great promise for improving quantum technologies in the future. A wide range of ideas have been developed at this interface between the two fields during the past five years; see Fig. 1. Whether one tries to understand a quantum state through measurements, discover optimal feedback strategies or quantum error correction protocols, or design new quantum experiments, machine learning can yield efficient solutions, optimized performance and, in the best cases, even new insights.

With the present review, we aim to take physicists with a background in quantum technologies on a tour of this rapidly growing area at the interface to classical machine learning. The readers are not expected to have a background in machine learning and will get a state-of-the-art view into how machine learning techniques are applied to quantum physics. We should state right away that in this perspective article we

tried to achieve our goal by focusing in each application domain on some selected illustrative examples. Our selection is necessarily subjective. We thus make no claim as to providing a comprehensive list of the literature and apologize to anyone who misses some favorite work.

We hope that after seeing the examples discussed in our review, the reader will appreciate how useful machine learning techniques could be for quantum technologies. At the same time, we also want to make the reader aware of how crucial it is to choose the right AI approach for a problem. It is by no means necessary that the most advanced AI method is the most suited tool for a given task. Often, modern deep learning methods (i.e., machine learning with deep, multilayered neural networks) can be significantly outperformed by rather simple methods if applied to the wrong task, as seen in other areas [8]. Here it is crucial to analyze the scope of the problem and decide on the best-suited algorithms.

Once the reader wants to understand how to apply these tools in practice, we recommend the very educational lecture notes introducing machine learning techniques with a view to their application to quantum devices, both brief [9] and very extended [10]. In addition, there are several reviews from recent years with a somewhat different focus than ours, e.g., about machine learning applied to physics in general [11,12] or machine learning for quantum physics [13], quantum many-body physics [14,15] and quantum photonics [16,17], and learning quantum machines [18].

We also remark that there is the whole field of quantum machine learning, which tries to discover potential quantum advantages when implementing new learning algorithms on quantum platforms. This also includes variational quantum circuits, which are very promising in the context of noisy intermediate-scale quantum (NISQ) computers. We do not cover these developments here, and we refer the interested reader to reviews for quantum machine learning [19–22] and variational quantum algorithms and NISQ devices [23,24]. While we focus on quantum technology here, we want to

*ML4qtech@mpl.mpg.de

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Open access publication funded by the Max Planck Society.

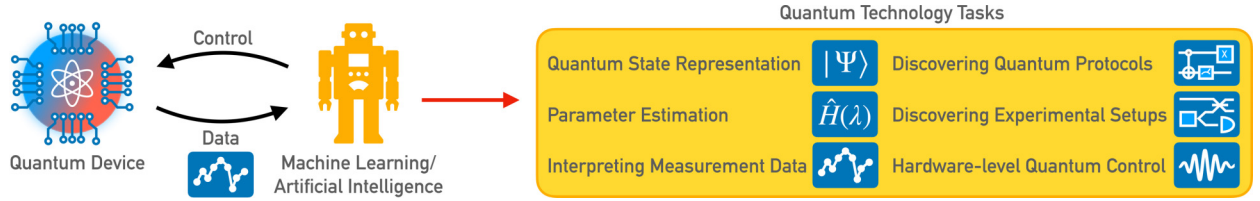


FIG. 1. Overview of tasks in the area of quantum technologies that machine learning and artificial intelligence can help solve better, as explained in this perspective article.

mention that there is also considerable work on ML for foundational quantum science [25].

In the following section, we will briefly introduce the basics of neural networks and other machine learning techniques, aiming to set the stage for subsequent discussions. The bulk of our review is contained in Sec. III, where we discuss the various applications of machine learning to quantum technologies. In each case we aim to remark on some of the challenges and potential future research directions. Finally, in the Outlook we speculate about how machine learning might have transformed quantum technologies in a dozen years from now.

II. BASIC TECHNIQUES OF MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

The purpose of this section is merely to provide a glimpse of the essential basics so that the subsequent discussion of applications becomes intelligible to the reader without prior exposure. Machine learning techniques have been around for several decades and involve many efficient approaches that predate the recent deep learning revolution, like “support vector machines” or “decision trees” [26]. However, the flexibility of deep neural networks (that is, neural networks with many layers) has made them a popular general-purpose choice, so we focus on those in our brief introduction.¹ In the next sections, we will give a short introduction to evolutionary algorithms (as they are commonly applied to quantum technologies, as we will see in Sec. III) and then focus on different aspects of neural networks in machine learning, including supervised, unsupervised, and reinforcement learning. Other techniques will be presented within Sec. III when they are necessary to understand the application.

A. Evolutionary algorithms

One set of algorithms that often been used in optimization tasks in the domain of artificial intelligence are evolutionary (genetic) algorithms [28,29]. There, the idea is to deal with a set of candidate solutions, each of them described via a

suitable vector. These solutions can be randomly changed (“mutated”), two solutions can be combined to form a new candidate (“crossover”), and finally only the best solutions can be kept for the next round of evolution (“selection”). Such methods can be surprisingly effective, and their applications to new problems is often straight forward. As we will describe in further sections, genetic algorithms have successfully been applied to quantum-technology-related tasks.

B. Neural networks: Structure

The structure of artificial neural networks, which can be trained to approximate arbitrary functions, is loosely motivated by neurons in the human brain. Each neuron receives multiple inputs and generates an output signal which serves as input for other neurons. During learning, the strength of connections between neurons changes.

More precisely, each artificial neuron receives N inputs x_j which are summed according to some weights w_j , representing connection strengths; see Fig. 2(a). After adding a bias (shift) b , a simple nonlinear “activation function” f is applied to yield the output y :

$$y = f(z), \quad (1a)$$

$$z = \sum_{j=1}^N w_j x_j + b. \quad (1b)$$

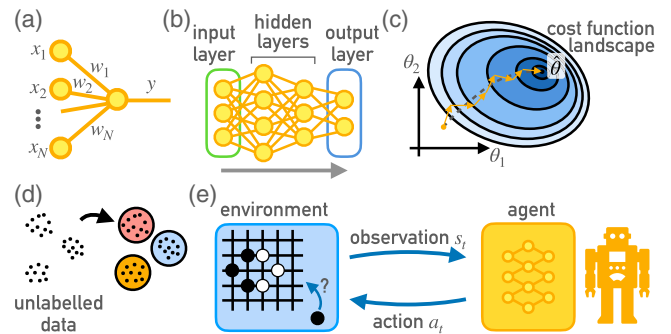


FIG. 2. Basics of neural networks and machine learning techniques. (a) Operation of a single artificial neuron. (b) Structure of a neural network with dense layers. (c) Evolution of the network’s parameters θ (which contains weights and biases of the neural network) in the cost function landscape using stochastic gradient descent. For every step (orange arrows) the gradients of the averaged cost function with respect to θ (gray dashed line) are approximated by the gradient averaged over a random batch. The parameters $\hat{\theta}$ minimize the averaged cost function. (d) Classification of unlabeled data. (e) Reinforcement learning problem modeled as Markov decision process.

¹Only a few years ago, implementing neural networks still required quite some effort. Nowadays, libraries like Tensorflow, PyTorch, JAX, and many others offer powerful tools to set up and train neural networks. We have created an online collection of resources to start with ML that contains information on different popular frameworks, as well as helpful tutorials, lecture notes, and reviews for machine learning in general. We plan to update this collection continuously and are happy to receive contributions from the community [27].

Without the nonlinear activation function between each layer, the whole neural network could be compressed into one single linear transformation, with very limited computational power. Popular activation functions are the “ReLU” ($f(z) = z$ for $z \geq 0$ and $f(z) = 0$ for $z \leq 0$) and the “sigmoid” function $f(z) = 1/(e^{-z} + 1)$, which represents a smoothed step function.

Multiple connected neurons form a neural network as shown in Fig. 2(b). These networks are typically structured into several layers, where the output from each layer serves as input for the next layer. Overall, there is one input layer, one or more intermediate “hidden” layers, and an output layer. The input being fed into such a network could be, e.g., pixel data from an image, and after successful training, the output neurons encode e.g., the category of the image (cat vs dog). A network with only one hidden layer and suitably many neurons is already able to approximate arbitrary functions with arbitrary precision [30]. However, networks with more hidden layers (called “deep” neural networks) may be able to fulfill the same task with fewer neurons.

In the network in Fig. 2(b), all outputs of one layer serve as input for all neurons of the next layer. Such a structure is called a *dense* layer, or “fully connected network.” However, this is often not the best choice, especially when the input has certain symmetries. For example, an image classification task can have a translational symmetry since the category of an object shown in a picture is independent of the object’s precise location. The typical choice for these kind of inputs are convolutional neural networks (CNNs). Here the output neuron is connected only to a set of neighboring input neurons and the applied weights depend on only the distance between the output and input neuron. Therefore, the number of parameters of the layer is drastically reduced and the multiplication between the weights and inputs corresponds to a convolution which ensures the translational symmetry.

Time series have a temporal structure that can be utilized using so-called “recurrent” neural networks (RNNs). The key idea is that the RNN receives the information for each subsequent time step sequentially as input but also keeps some memory about previous inputs. One of the most advanced and commonly used RNNs are so-called long short-term memory (LSTM) [31] networks.

C. Supervised and unsupervised learning of neural networks

A network can be made to approximate any function by suitably choosing its weights and biases, often summarily denoted as a parameter vector θ . To produce correct predictions with a neural network, its parameters have to be trained for the given problem. Supervised learning (SL) is a training method to learn a desired target function $f(x)$ from a data set containing pairs (“samples”) of inputs and associated outputs, $(x, f(x))$. A neural network represents an approximate input-output relation $f_\theta(x)$. The deviation between the network’s prediction and the correct output in the data set is quantified by a “cost” (or “loss”) function. For any given input x , this cost function can be calculated, $C(f_\theta(x), f(x))$, and eventually, it will be averaged over all samples in the data set. The trainable parameters θ of the NN $f_\theta(x)$ have to be chosen so as to minimize this (sample-averaged) cost function.

The choice of cost function depends on the problem. Typical fields of application for neural networks are regression and classification tasks. For regression the target function $f(x)$ is a continuous function of the input x . Then the so-called “mean-square error” is the canonical choice, which for n output neurons reads

$$C_{\text{MSE}}(x) = \sum_{i=1}^n [f_{\theta,i}(x) - f_i(x)]^2. \quad (2)$$

On the other hand, in classification tasks, the input should be assigned to certain predefined classes (e.g., categories of images). This is solved by having each output neuron correspond to one of the n classes. Each neuron value (or “activation”) can then be interpreted as the probability that the input is assigned to the corresponding class. In that situation, the typical cost function is the so-called “cross-entropy,” a means to compare probability distributions, which reads as

$$C_{\text{CE}} = -\frac{1}{n} \sum_{i=1}^n f_i(x) \ln f_{\theta,i}(x). \quad (3)$$

It is minimal if the probability distribution $f_\theta(x)$ matches the target distribution $f(x)$. $f_i(x)$ is 1 if the data set assigns the input x to class i , else 0.

In any case, neural network training relies on the minimization of the cost function using gradient descent. However, evaluating the cost function averaged over all samples of the data set is infeasible. Rather, in each update step, the cost function is averaged over a batch of randomly selected inputs. In the simplest version of the resulting “stochastic gradient descent” scheme [see Fig. 2(c)], the parameters θ are updated in each step according to $\theta \rightarrow \theta - \eta g$ with the gradient g :

$$g = \frac{\partial \langle C \rangle_{\text{batch}}}{\partial \theta}, \quad (4)$$

where $\langle \cdot \rangle_{\text{batch}}$ denotes the average over the random batch, and η is the “learning rate,” controlling the update’s size.

A neural network has many parameters (even small networks usually have thousands of parameters, the largest published networks have hundreds of billions of parameters). Therefore, it is crucial that there exists a highly efficient approach to calculating the gradient: the so-called “backpropagation” scheme [32,33]. As its name implies, this algorithm calculates the gradients layer by layer, starting from the output layer. Remarkably, it is computationally not more demanding than the original evaluation of the neural network (also called the forward propagation). A significant hardware advance for training neural networks are GPUs (graphical processor units) that are optimized to perform highly efficient manipulations of large matrices. This efficiency is essential for the success of ML in many applications.

For more details on neural network architectures, backpropagation, and gradient descent, the interested reader is referred to the many existing excellent introductions [3,9,10,34].

Unsupervised learning approaches can learn the structure of unlabeled data sets on their own, for example, as shown in Fig. 2(d). Typical fields of application are feature learning, where the machine is asked to find a compact representation of the data, and clustering, where the computer has to sort on its

own samples into classes with similar properties. Generative models are also often attributed to the field of unsupervised learning. Their purpose is to stochastically create new samples that follow the same distribution as the previously observed data set (e.g., images of the same type, though never seen before). The simplest approaches to feature learning still rely on the techniques of supervised learning (namely, “self-supervised” learning in so-called “autoencoders”). However, clustering and generative models are typically implemented using distinct techniques (see, e.g., [3]).

D. Reinforcement learning

Some important problems in artificial intelligence and machine learning can be seen as the attempt to find an optimal strategy to cope with a certain task, where the optimal strategy is unknown (thus, supervised learning cannot be applied). This is the domain of reinforcement learning (RL), which aims to discover the optimal action sequences in decision-making problems. Its power has been famously illustrated in board games like chess or Go [35] or video games [36], in all of which RL is able to reach superhuman performance. In many applications, RL can find the optimal strategy without prior knowledge about the actual dynamics of the system. When that is the case, we speak of “model-free” RL. The goal in any RL task is encoded by choosing a suitable “reward,” a quantity that measures how well the task has been solved.

The typical RL problem can be understood as a so-called “Markov decision process” (MDP); see Fig. 2(e). An MDP consists of an “agent” (the controller) and an “environment” (the world, or the system to be controlled), and both interact in multiple time steps. In each time step t , the environment’s state s_t is observed. Solely based on this observation, the agent decides on its next action a_t , which will change the environment’s state. The agent’s behavior is defined by the “policy” $\pi(a|s)$, which denotes the probability of choosing the action a given the observation s .

For each action a_t , the agent receives a reward r_t . For example, in a game, this reward could be $+1/-1$ at the last time step, when the agent has won or lost the game, and otherwise 0 in all previous steps. The sequence of observations, actions and rewards is called a trajectory. RL aims to maximize the return R (the cumulative reward of a trajectory)

$$R = \sum_{t=1}^T r_t, \quad (5)$$

where T is the total number of time steps.

Three major branches of RL algorithms exist: policy gradient, Q-learning, and actor-critic methods. For policy-gradient methods, the agent directly sets the policy $\pi_\theta(a|s)$. In deep RL, this agent is realized by a deep neural network with trainable parameters θ . To find the optimal strategy, a policy gradient estimates the gradient of the average return $\langle R \rangle$ with respect to θ . Here $\langle \cdot \rangle$ denotes the average over all trajectories for the current policy. At first sight, it is unclear how to take the gradients through the reward without knowing the model. However, one can compute the gradients of the frequency of a certain reward via the policy function π_θ . Thus, the gradient

turns out to be

$$g = \frac{\partial \langle R \rangle}{\partial \theta} = \left\langle R \sum_{t=1}^T \frac{\partial}{\partial \theta} \ln \pi_\theta(a_t | s_t) \right\rangle. \quad (6)$$

It is important to note that R depends on the full trajectory and the parameters θ (we suppressed these dependencies for brevity). Updates based on this equation will increase (“reinforce”) the probabilities of actions that occur predominantly in high-reward trajectories. Another approach to finding the optimal policy is so-called “Q-learning.” It employs the “Q function” $Q_\pi(s, a)$ that tries to estimate the quality of an action a : it is defined as the average expected future return starting from a state s and action a for a policy π . The optimal policy is then, by definition, to choose the action that maximizes Q in a given state s . In practice, the Q function is initially unknown. During training, an approximation to this function is learned, often using a deep neural network to represent Q .

Finally, the third group of RL algorithms are the so-called “actor-critic” methods. These try to combine the benefits of both policy-gradient and Q-learning approaches. The basic idea behind actor-critic methods is to estimate the expected reward given the current state, the so-called value function V . The success of any action is then measured by comparing the resulting reward against V . As might be expected, the value function is represented by a neural network (the “critic” network) which is trained by SL to approximate the true value function for the current policy.

For an introduction to the different RL algorithms, the interested reader is referred to [37].

E. Automatic differentiation and gradient-based optimization

Deep learning is efficient because of the backpropagation technique that can efficiently calculate gradients with respect to all the hundreds or millions of parameters in a neural network. This technique more generally leads to the concept of automatic differentiation. There the idea is to obtain the exact gradient with respect to any variable appearing in any kind of numerical calculation. As a numerical approach, this is distinct from symbolic differentiation applied in computer algebra programs. Modern frameworks used for neural networks offer various modes of automatic differentiation. This offers the chance to employ them for arbitrary gradient-based continuous optimization tasks, especially those involving many parameters, where efficiency is of concern. We will later show examples of how this can be used to discover new quantum experiments, quantum circuits, and in other contexts.

III. APPLICATIONS OF MACHINE LEARNING FOR QUANTUM TECHNOLOGIES

A. Measurement data analysis and quantum state representation

An important direct application of machine learning to quantum devices is the interpretation of measurement data. The application ranges from an improved understanding of the measurement apparatus itself to extracting some high-level

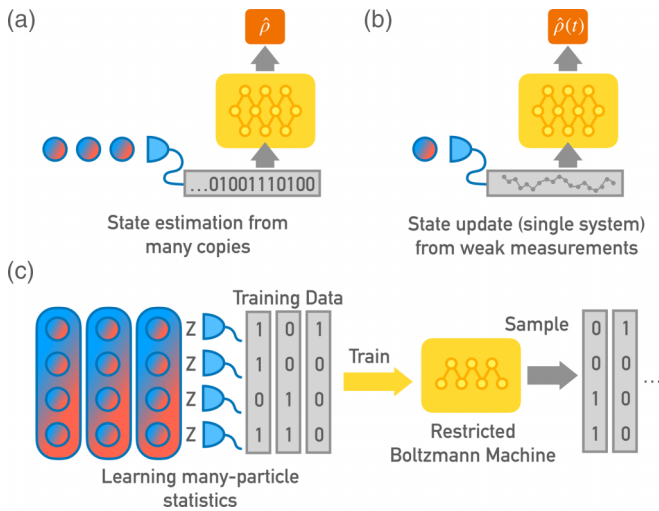


FIG. 3. State estimation via neural networks. (a) Measurements on many identical copies of a quantum state can be processed to produce an estimate of the quantum state. (b) A continuous weak measurement on a single quantum system can be used to update the estimated state. In both (a) and (b), a single network is trained to estimate arbitrary states correctly. (c) One can also train a network-based generative model to reproduce the statistics of a quantum state, i.e., to sample from the probability distribution. Training requires many identical copies that can be measured, so the statistics can be learned. Here one network represents only a single quantum state. It can be extended to handle measurements in arbitrary bases.

properties of the quantum system to the full reconstruction of the measured quantum state. In many cases, this can be phrased as a supervised learning task. One example might be to extract an approximate description of a quantum state from a sequence of measurement results on identically prepared copies; see Fig. 3(a). Provided the actual quantum state is known for each training example, the machine learning algorithm will learn to provide the best possible approximation to the quantum state. The same is true if the goal is to reconstruct certain properties of the state or the device instead of reconstructing the full quantum state. The choice of the cost function is essential, a simple example being the infidelity between the predicted and the actual state. Other choices will lead to slightly different optimal approximations predicted by the ML algorithm. Crucially, the algorithm can easily deal with distortions of the measurement data (such as extra technical noise), as it will learn the properties of these distortions and how to undo them.

1. Interpreting measurements

An interesting early example used machine learning to improve the readout fidelity of a qubit in a superconducting quantum device. There, the noisy measurement trace, obtained from a microwave signal passing through a readout resonator interacting with the qubit, can be used to deduce the qubit's logical state. However, classifying the qubit state is challenging. The authors of [38] use a machine learning technique called support-vector machine (SVM) to perform clustering of measurement traces in an unsupervised fashion, outperforming classical clustering algorithms. The idea of a

nonlinear SVM is to map data points to a higher-dimensional space and to find the best hyperplane for separating two classes of data points in that space. In this specific work, each measurement trajectory, which consists of hundreds of individual data points, is interpreted as a point in a high-dimensional space. The SVM's goal was to separate curves that originate from a zero state and one from a one state. The readout fidelity was improved compared to non-ML clustering techniques. Furthermore, this analysis has shown that the main noise contribution comes from physical bit flips (either heating or relaxation of the qubit). Without such events, the classification of the SVM becomes near-perfect. A similar approach has been demonstrated—using neural networks—to enhance the readout capability of trapped-ion qubits [39]. Here the authors show that the readout fidelity improves significantly compared to a non-ML clustering method, especially when the effective amount of data per measurement increases. Similar techniques have also been applied to NV center quantum devices [40].

In a pioneering experimental work, a neural network reconstructed the quantum dynamics of a quantum system directly from measurement data [41]. There, the authors considered again a superconducting qubit coupled to a readout resonator, whose noisy measurement trace is fed as input into the network (together with the initial preparation state and the final measurement basis). The network's task was to predict the statistics of arbitrary measurements at some given time during the evolution, i.e., effectively predict the evolution of the density matrix given the measurement record; see Fig. 3(b). The kind of network most suited to this task is a so-called recurrent network, i.e., a network able to process a time series (originally used for text or speech processing). The resulting fully trained network can map any measurement data to a quantum state.

Furthermore, machine learning techniques can be exploited to analyze the statistics of measurement outcomes in quantum experiments where the aim is to demonstrate the classical complexity of sampling from the quantum distribution. Boson sampling is the most well-known such scenario. In [42] unsupervised machine learning techniques (various clustering approaches) were employed to identify and rule out possible malfunction scenarios that would lead to a noticeably different distribution.

One exciting work from theoretical computer science connects the question of quantum state tomography with the theory of computational learning in the supervised setting [43]. Here a learner uses the training data set to produce a hypothesis about future measurements. Quantum state tomography, which requires a number of measurements that is exponential in the number of particles, can be seen as a learner that produces a hypothesis for every possible measurement on the quantum system. This might, however, not be necessary in most cases. The theory of *probably approximately correct* (PAC) models provides a hypothesis for every measurement close to the data set. Surprisingly, it was found that this question can be solved with a linear scaling of measurements. This computational learning strategy has been first demonstrated in quantum optics experiments with up to six photons. The experiment has confirmed the scaling behavior of PAC models, even in the presence of realistic experimental noise [44].

Another very interesting recent development shows that joint quantum measurements of several individual copies of a many-particle quantum state can lead to an exponential improvement over classical learning algorithms [45]. The authors show experimentally on a platform of 40 superconducting qubits that tasks such as predicting properties of the physical system can be significantly improved when the results of such joint measurements are fed as an input to a classical RNN. The result is particularly remarkable as it shows a clear advantage already for current, noisy, and not error-corrected quantum computers.

Many other interesting examples exist that use neural networks to analyze simulated or measured data of quantum systems. For example, it has been demonstrated that the Wigner negativity of a multimode quantum state can be approximated well even in the low-data regime [46], with important consequences for quantum technologies. Another vivid field is the neural-network-based detection of quantum phase transitions and classification of quantum phases in condensed matter physics. We will not go into detail here. Instead, we point to some exciting early works in this field [47–49], as well as some very modern applications that, for instance, use anomaly detection [50] or influence functions [51] to find phase transitions in an unsupervised way. Both tools have already been applied to detect phases directly from experimental measurements [52]. See a recent review on this topic in [14].

2. Approximation of quantum states

The direct application of numerical techniques to quantum devices requires, in many situations, the storage, and processing of the system's full quantum state. As the quantum state grows exponentially with the number of particles, the memory requirements quickly become enormous, even for moderately large quantum systems. For example, storing the full quantum state of a 42-qubit system requires 35 TB of memory. As demonstrated in the earliest quantum advantage experiments [53], this is directly related to the power of quantum computers and quantum simulators. However, it poses a significant problem for classical computational approaches that deal with large quantum systems and therefore for developing new large-scale quantum technologies.

To overcome this challenge, memory-efficient approximations of the quantum wave function are indispensable. Neural networks are one key candidate to approximate the quantum wave function. This approach has sometimes been called a neural quantum state (NQS).

A prominent approach tries to represent the quantum state in terms of a neural network [54]. This implies that for each new quantum state, another network will be trained, based on the associated measurement data for that state. In principle, that is considerably easier than asking a single network to be responsible for arbitrary states, i.e., the task considered above. As a consequence, much more complicated many-body states can be accessed. The whole approach can be seen as a neural-network-based version of quantum state tomography.

Several different ways exist to use a single neural network to represent a single quantum state. One straightforward approach, first introduced in [54] and then extended in subse-

quent works, employs a network that directly represents the wave function. Given a multiparticle configuration x as input, the network has to produce the wave function amplitude for that configuration as output: $\Psi_\theta(x)$.

Different structures can be used for the network, with a restricted Boltzmann machine (RBM) being a popular choice since it also allows direct sampling from the probability distribution of observations [55]. In a traditional RBM, the aim is to learn to sample from some observed probability distribution; see Fig. 3(c). It consists of binary visible units and hidden units connected to each other, and the statistics of these units are sampled from a Boltzmann distribution with an energy E that contains interaction terms bilinear in the hidden (h) and visible (v) unit values: $-E = \sum_j a_j v_j + \sum_k b_k h_k + \sum_{j,k} w_{jk} v_j h_k$. During training, the coupling constants are updated to obtain the desired probability distribution of v (observed in samples provided during training). A simple physics example would be a 1D spin chain, whose configurations are identified as sample vectors v . More generally, other so-called generative deep learning methods (such as normalizing flows, variational auto-encoders, and generative adversarial networks) can be used to learn probability distributions, including those representing the statistics of observables in quantum states in a given basis.

Quantum state tomography using an RBM-style ansatz for the wave functions was introduced in [56]. Since one wants to keep the wave function's phase φ as well as the probability p , the ansatz is now of the type $\Psi(x) = \sqrt{p(x)}e^{i\varphi(x)}$, where both p and φ are represented as networks and x corresponds to the visible units. A crucial idea in this approach is to match the probability distributions obtained from the experiment for observables in more than one basis (e.g., σ^z and σ^x for qubits). The evaluation of different bases can be carried out via unitary transformations acting on the wave function Ψ that is expressed in a single reference basis. Experimental implementations based on photonic quantum states have demonstrated that these techniques allow for full estimation of quantum states based on a smaller amount of experimental data compared to state-of-the-art methods [57,58].

In [59] this approach was applied to experimental data from snapshots of many-particle configurations taken in a Rydberg atom quantum simulator. The resulting network-based wave-function ansatz could then be used to reconstruct other expectation values and observables that were not directly accessed in the experiment.

The idea has been extended to mixed states [60–64]. In that way, NQS can be used to efficiently approximate open quantum systems which are notoriously difficult to capture.

3. Approximating quantum dynamics

Once a suitable quantum state representation is available, it can be exploited to evolve the state in time. This enables potentially efficient simulation of quantum many-body time evolution, which is important for predicting and benchmarking the dynamics of quantum simulators and quantum computing platforms. For the general case of dissipative quantum many-body dynamics, i.e., the time evolution of mixed states, this has been explored in [62].

Rather than explicitly storing the entire quantum state, another technique shows how one can directly compute a quantum state's complex properties just from the state's construction rules, i.e., the quantum experimental circuit. In [65] the authors show how a recurrent neural network [31] can approximate the properties that emerge from quantum experiments without ever storing the intermediate quantum state directly. These systems could then directly be applied for complex quantum design tasks, a topic we cover in Sec. III D. Another approach [66] also foregoes the representation of quantum states and instead trains a recurrent network to predict the evolution of observables under random external driving of a quantum many-body system (based on either simulated or possibly even experimental data). The trained network can then predict the evolution under arbitrary driving patterns (e.g., quenches). In a similar spirit, neural ordinary differential equations (ODEs) can be used to approximate the dynamics of quantum systems directly, again without storing the explicit information about the quantum wave function [67]. Interestingly, the approximation is of high enough quality that it is possible to rediscover some fundamental properties of quantum physics, such as the Heisenberg uncertainty relation.

4. Future challenges and opportunities

Improved data efficiency (both for the training but also when applying ML to interpret the data) will be an important challenge for the future, especially when the devices scale to more complex quantum systems.

It will be interesting to codiscover measurement strategy together with the data interpretation strategy. This might be particularly interesting if the AI algorithm is allowed to employ quantum measurements on numerous copies of the same state, as pioneered in [45].

When a neural network can find a suitable approximation for the computation of complex quantum systems, such as an NQS, it has learned a theoretical technique that might be useful for humans too. It will be interesting to learn how to extract the *per se* inaccessible knowledge from the weights and biases of the neural network. One method is so-called symbolic regression.

The extensions of NQS to complex quantum systems, such as higher dimensions and spins beyond qubits, will allow for more interesting applications.

B. Parameter estimation: Learning the properties of quantum systems

In this section, we will discuss machine-learning-based approaches for estimating experimental parameters. These could either be system parameters for the calibration of quantum devices or the estimation of external parameters in the form of quantum metrology.

1. Quantum metrology

Machine learning can be helpful for various challenges in quantum metrology [16]. Quantum metrology deals with the resource-efficient measurement of external parameters acting on a quantum system, like magnetic fields or optical properties

of a material sample. Broadly, the field can be separated into two different branches. First, nonadaptive approaches exploit how complex quantum entanglement can be exploited to reduce the required resources (for instance, the number of photons that interact with a sample) without changing the experimental setup or input states throughout the measurement sequence. The second class of approaches exploits feedback, meaning they employ adaptive strategies that change either the input or the measurement setting, depending on the previous measurement outcome. Naturally, adaptive strategies require more advanced experimental implementations, including fast switching or long-term stability of setups. For that reason, nonadaptive quantum metrology is so far much more explored in laboratories, while adaptive approaches are still at the stage of proof-of-principle experiments. AI has contributed to both approaches.

An example of the application of neural networks to the estimation of unknown parameters in a photonic experiment is provided in [69]. There the goal is to calibrate a device via neural network training that can later be used to estimate an unknown phase shift. The authors first accumulate a large set of calibration measurements using a controlled phase plate. In addition, the data are augmented to account for the statistical noise contribution. The neural network receives the measurement data (the number of detected photons) and has to estimate the corresponding phase. As soon as the neural network is able to model the connection between measurement data and calibration phases, it can be used to estimate unknown phases as well. This task shows that device calibration and estimation of external parameters for quantum sensors are closely related.

In addition, also still for nonadaptive approaches, various projects have explored the discovery of new experimental setups for the resource-efficient measurement of parameters of an external system [70,71]. We will talk about this approach in Sec. III D.

We now turn to focus on feedback-based quantum metrology schemes and how they could be improved with ML algorithms. A pioneering non-ML approach for such a task is called BWB (Berry-Wiseman-Breslin) strategy, after the authors of [72,73]. In the original setting of that strategy, the authors consider a Mach-Zehnder interferometer with two detectors at the two outputs; see Fig. 4(a). One of the arms contains an unknown phase shift, and the other arm contains a controllable phase that is modified depending on the previous measurement results. The goal of this approach is to send individual photons into the interferometer, and modify the controllable phase according to previous measurement results to decrease the phase uncertainty. This problem can therefore ideally be formulated as a decision tree [see Fig. 4(b)], where the actions must be chosen based on the measurement outcomes to maximize the information gain. In general, the expected information gain is defined as the reduction of entropy of the parameter distribution, averaged over possible measurement outcomes. The authors derived the feedback algorithm by applying Bayes theorem to the distribution of the unknown phase, which is then used to choose subsequent measurement settings with a large information gain. This strategy has been found not only to be highly resource-efficient, but also practically implementable in the laboratory [74].

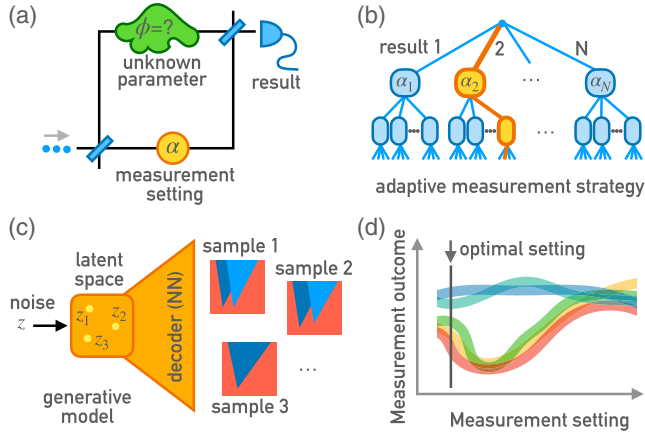


FIG. 4. Machine learning for parameter estimation in quantum devices. (a) A typical scenario, with the measurement result statistics depending on both some tuneable measurement setting and the unknown parameter(s), here represented as phase shifts in a Mach-Zehnder setup. (b) An adaptive measurement strategy can be illustrated as a tree, with branches on each level corresponding to different measurement outcomes. Depending on those outcomes, a certain next measurement setting (indicated as “ α_j ”) needs to be selected. Finding the best strategy is a challenging task, as it corresponds to searching the space of all such trees. (c) Neural generative models can be used to randomly sample possible future measurement outcomes (here 2D current-voltage maps as in [68]) that are compatible with previous measurement outcomes. This is helpful for selecting the optimal next measurement location. Different random locations in latent space result in different samples. (d) Measurement outcome vs measurement setting for five possible underlying parameter values (different curves; measurement uncertainty indicated via thickness). We aim to maximize the information gain, i.e., choose the setting which best pinpoints the parameter (which is not equivalent to maximizing the uncertainty of the outcome).

Interestingly, greedy adaptive strategies (i.e., to choose in every step the reference phase that yields the largest immediate information gain) do not necessarily lead to the largest information gain in the long run. To overcome this effect, one of the seminal early contributions to ML-based quantum metrology used particle swarm optimization of the feedback strategy [75]. In particle swarm optimization, a collection of different feedback strategies (each called a “particle”) iteratively moves in the space of all possible strategies. In each iteration, the particle moves towards a combination of the best local optimum and the currently best known global optimum known by the whole swarm. The experimental setting of [75,76] is the same as for the BWB strategy—a Mach Zehnder interferometer with an unknown phase in one arm and an adaptive phase in the other arm. Indeed, the swarm optimization algorithm finds (slightly) better strategies than the greedy Bayesian BWB approach. Interestingly, neither BWB nor swarm optimization can find the optimal strategy, which was identified for small photon numbers via an extensive computation of all possible strategies. Other early ML algorithms in this domain have applied evolutionary approaches to approximate the ideal feedback strategy [77,78].

Discovering a strategy is a problem that can directly be formulated as a reinforcement learning task. An early application

of RL to quantum parameter estimation was provided in [79], with frequency estimation of a qubit as a test case. In that work the idea was to optimize the quantum Fisher information for the parameter of interest. This can be done by finding a sequence of suitable control pulses applied during the noisy evolution of the quantum probe. No feedback is involved in this simple setting since the measurement itself is not part of the evolution controlled by RL.

However, the quantum Fisher information is useful only in cases where one is already fairly certain of the true parameter value. RL can be employed to study more complex situations, where updates are performed using the Bayes rule, starting from an arbitrary prior parameter distribution, and where the strategy is not greedy (i.e., more than a single step of the sequence is optimized). In [80] the authors provided information about the current Bayes distribution of the unknown parameter (as extracted from previous measurement results) and the previous measurement choices as input to an RL agent implemented by a neural network. It then has to suggest the next measurement. After the whole sequence of measurements, the agent is rewarded according to the total reduction in parameter variance. It was shown that this approach performs very competitively for an important test case, namely, parameter estimation for a qubit of unknown frequency in the presence of dephasing.

2. Device calibration

Future large-scale quantum devices will consist of a large number of components with adjustable parameters that need to be characterized and tuned automatically. A complete characterization of the device via quantum process tomography quickly becomes impractical. To find the actual parameters or the ideal operating point of the quantum device, it is therefore necessary to extract the relevant data with a very limited amount of information. This task can be formulated as a machine learning task, specifically applying “active learning” or “Bayesian optimal experimental design” [81] where the algorithm chooses the most informative measurement autonomously (see also [18] for a review of such methods in the context of quantum devices). Naturally, this is closely related to the estimation of external parameters in quantum metrology as discussed above.

We illustrate these techniques via a pioneering experimental application to quantum devices. This experiment [68] considered the calibration and measurement of a semiconductor quantum dot. Such a device can be tuned via applied gate voltages, and its resulting properties can be measured via a transport current. Here the goal was to explore the properties of the quantum dot, as defined by its current-voltage map $I(V_1, V_2)$, where the voltages include a bias voltage driving the current and a gate voltage deforming the dot’s potential. In a naive approach, even if only two voltages were scanned with 100 discretization steps each, one would need to perform 10 000 measurements to get a suitable resolved device characteristic.

To reduce the required number of current measurements, the authors tried to estimate which measurement (in the 2D voltage space) would yield the “maximum amount of additional information.” In practical terms, this would be the

measurement that is expected to place the tightest constraints on the current-voltage maps that are still compatible with all the observed values of the current (observed in this and prior measurements). It is obvious how this setting translates to other quantum platforms, e.g., measurements of microwave transmission through superconducting circuits controlled via gate voltages and magnetic fields or the optical response of tuneable atomic systems.

The first step towards this goal is to efficiently represent all current-voltage maps that might be observed, given the general physics of such a device, the assumed prior distribution of device parameters, and all previous measurement results. In general, this is the domain of “generative models,” which can sample from a probability distribution that is learned. In the case of [68], the authors used such a generative model, in their case a so-called “constrained variational autoencoder” (cVAE), to randomly create realistic current-voltage map that follow the probability distribution of the actual physical system; see Fig. 4(c). Additional input into the generative model provides a constraint, in the form of a few initially existing measurement results, and guides the reconstruction to sample only maps compatible with those constraints. In each step, 100 different voltage maps are sampled. Those maps are used to find the next measurement point in voltage space that would lead to the maximum information gain; see Fig. 4(d). With this technique, the total number of necessary measurements for the characterization of the device is reduced by a factor of 4. This clearly shows that the overhead of the deep learning algorithm is more than compensated by its efficiency improvement compared to the naive approach. A benefit of this technique is that generating new samples with the cVAE is very efficient. Thus it can be scaled to much larger devices, where even more significant efficiency gains are expected.

Another comparatively straightforward way to use machine learning in device characterization consists in training a network-based classifier to recognize “interesting” measurement results. This then allows one to tuning parameters until those results are obtained. Such an approach has been demonstrated in [82] for navigating charge-stability diagrams of multi-quantum-dot devices. In that setting, the algorithm’s goal was to automatically tune the charge occupation of the double quantum dot. The task is reformulated as a classification task, where the algorithm recognizes individual charge transitions when presented with a charge-stability diagram. Since such a diagram constitutes an image, CNNs are a suitable choice for the task.

3. Quantum Hamiltonian learning

Imagine the following parameter-estimation problem: One wants to estimate the parameters x_0 that affect the evolution of a quantum state under a *quantum many-body* Hamiltonian $H(x_0)$ [83]. Unfortunately, even the task of computing the dynamics scales exponentially with the system size (number of qubits) when tackled using a classical machine. The idea of quantum Hamiltonian learning (QHL) is to enlist the help of a quantum simulator to overcome this problem. The parameters x_0 can then be estimated with Bayesian methods. Thereby, the quantum simulator is used like a subroutine inside a classical ML approach. The first experimental implementation of this

idea was demonstrated in 2017 [84]. In that work, the authors wanted to estimate the parameters of an electron spin in a nitrogen-vacancy center, and they used a quantum simulator on an integrated photonics platform to perform the QHL. Interestingly, not only did the approach lead to a high-quality estimation of the dynamic system parameters, but it also indicated when the initial Hamiltonian model had deficits. In these cases the learning method informed the user that there are other dynamics in play that have not been considered, which inspires an improvement of the underlying Hamiltonian model.

While the QHL method indicates that when the model Hamiltonian is not ideal, it cannot adapt it. To overcome this hurdle and to learn the entire Hamiltonian structure (not only its parameters), the authors of [85] have introduced the idea of a quantum model learning agent (QMLA). This agent not only finds the parameters of a predefined Hamiltonian, but discovers the whole Hamiltonian that describes the dynamics of a system. The approach iteratively refines the initial Hamiltonian and uses QHL as a subroutine for finding suitable parameter settings. This approach has also been demonstrated in a hybrid quantum system involving a nitrogen-vacancy center. The underlying learning mechanism is very general, and it thus could become a powerful tool for learning the dynamics of unknown quantum systems.

4. Future challenges and opportunities

For adaptive approaches, one needs to consider the trade-off between speed and the sophistication of the approach. In these tasks, the time between measurement and feedback is often very short, so the decision must be taken quickly.

An interesting future approach for advanced quantum metrology approaches is to simultaneously codesign the experimental setup and the feedback strategy, rather than solving these tasks individually.

C. Discovering strategies for hardware-level quantum control

Challenges like quantum computing and quantum simulation are leading to rapidly increasing demands on the efficient and high-fidelity control of quantum systems. Tasks range from the preparation of complex quantum states and the synthesis of unitary gates via suitable control-field pulses all the way up to goals like feedback-based quantum state stabilization and continuously performed error correction. In trying to solve these tasks, the specific capabilities and restrictions of any hardware platform, from superconducting circuits to cavity quantum electrodynamics, need to be considered.

In this section we will highlight specifically how reinforcement learning has come to help with many of these challenges. In the form of model-free RL, it promises to discover optimal strategies directly on an experiment, which can be treated as a black box; see Fig. 5(a). All its unknowns and nonidealities will then be revealed only via its response to the externally imposed control drives. But even when used in a model-based way, using simulations, RL can be more flexible than simpler approaches. In particular, it offers ways to discover feedback strategies, i.e., strategies conditioned on measurement outcomes. These were not previously accessible to the usual numerical optimal control techniques.

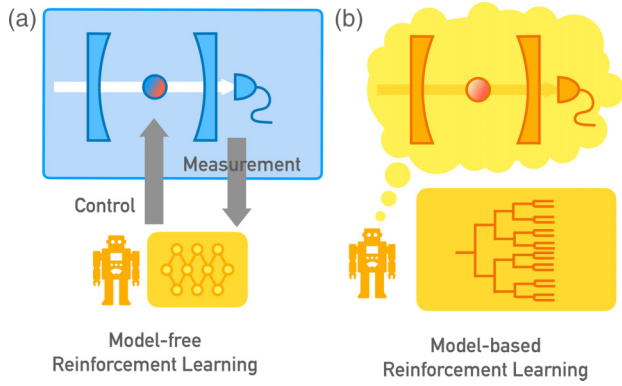


FIG. 5. (a) The eventual goal of model-free reinforcement learning is the direct application to experiments, which then can be treated as a black box. Many actual implementations, however, use model-free RL techniques applied to model-based simulations. (b) Model-based reinforcement learning directly exploits the availability of a model, e.g., taking gradients through differentiable dynamics.

The present section is firmly concerned with hardware-level control that is continuous in the time domain, discovering pulse shapes or feedback strategies based on time-continuous noisy measurement traces as they would emerge from weak measurements of quantum devices. There are some connections to the next section, but there we will be concerned with the discovery of protocols, control strategies, and whole experimental setups that are described on a higher level, composed of discrete building blocks like gates or experimental elements.

1. Quantum control tasks without feedback (open-loop control)

Prior to the application of machine learning techniques in this field, the focus was essentially on tasks without feedback, which was solved by direct optimization techniques, adapting the shape of control pulses applied to the quantum system to maximize some quantity (like the state fidelity). These direct optimization techniques include gradient-based approaches, with GRAPE [86] and the Krotov method [87,88] the most prominent examples, as well as approaches that do not rely on access to gradients, such as CRAB [89]. At the time of writing, these techniques still form the default toolbox for the case of open-loop control, even while the first applications of machine learning (described below) are taking hold. Evolutionary algorithms define another class of (stochastic) approaches that have been used successfully to find optimal control sequences [90].

State preparation is the most common quantum control problem, and yet it can already be challenging, especially for multiqubit settings. In probably the earliest application of RL to quantum physics, pure-state preparation with discrete control pulses was shown using a version of Q-learning for a spin-1/2 system and a three-level system [91]. A few years later, RL-based state preparation was demonstrated for a many-qubit system [92], also using Q-learning and discrete bang-bang type actions, with particular emphasis on analyzing the complexity of the control problem showing up in the form of a glassy control landscape. Both of these works used

some version of table-based Q-learning, which works well for a restricted number of states and actions. The first work to employ deep (i.e., neural-network-based) RL methods to open-loop control of quantum systems was [93], with both discrete and continuous controls and a recurrent network as an agent, as applied to dynamical decoupling and again state preparation, followed shortly afterwards by [94].

The RL approach can be used successfully to find suitable pulse sequences for unitary gates and optimize for the gate fidelity, as shown first in [95], and analyzed later also in [96]. Furthermore, RL was employed to maximize the average power of quantum thermal machines [97].

Recently deep RL has been applied for the first time to learn control strategies for a real quantum computing experiment [98]. The authors trained on a cloud-based quantum computing platform, collecting data for the current control policy, extracting rewards, and updating the policy. The goal was unitary gate synthesis, and the lack of real-time access to the device was not a concern since the task required only open-loop control. This first demonstration of RL-based quantum control on a real quantum experiment helped to illustrate the possibilities and challenges in this new approach.

Even though open-loop control pulse design means that the actual strategy in the experiment is not conditioned on any measurement outcomes, RL training for such tasks (when done on a computer simulation) may still benefit from the agent receiving input information like the current quantum state. Experience shows that this makes it easier to find a good strategy. Otherwise, only very sparse nominal information like the current time step and possibly the most recent selected action would be fed into the agent. In any case, however, once RL has found a control sequence, it could in principle be stored (e.g., as a waveform or pulse sequence) and sent to an experiment whenever needed. In other words, there is no need for the agent to be running during the actual experiment, which strongly relaxes requirements for the hardware: no real-time control is necessary.

2. Quantum feedback control (closed-loop control)

The successful control of quantum systems subject to noise, decay and decoherence requires either reservoir engineering (autonomous feedback) or active feedback control. The space of active feedback strategies is exponentially larger than that of open-loop control strategies (i.e., without feedback), owing to the number of potential measurement outcome sequences growing exponentially with time (each such sequence may require a different response). It is here that it is almost inevitable to use the power of RL, particularly deep RL, with its ability to process high-dimensional observations.

The first work to apply deep RL to *feedback-based* control of quantum systems was [99]. It employed discrete gates for quantum error correction, and we will discuss it in Secs. III D and III E. State preparation and stabilization in the presence of noise or an uncertain initial state are other natural candidates for RL feedback strategies. Examples include quantum state engineering via feedback [100], as well as control of a quantum particle in an unstable potential [101] and a double-well potential [102]. In some quantum systems, control may be very limited (e.g., only linear manipulations), but mea-

measurements can introduce nonlinearity and their exploitation through RL-based feedback strategies can enable powerful control, as shown in [103].

One challenge for model-free RL as applied to experiments is to make sure rewards can be extracted directly and reliably from experimental measurements, and to use a training procedure that really treats the quantum device as a black box (not relying, e.g., on simulations). These aspects were emphasized in [104], where state preparation in a cavity coupled to a qubit was analyzed.

An additional challenge for experimental deployment is that the agent (a neural network) must process the incoming measurement signals in real time and decide on the subsequent actions in timescales much shorter than the coherence time of the quantum system. In [105], the authors developed a low-latency neural network trained via model-free RL to control a superconducting qubit. The network is implemented on a field-programmable gate array (FPGA) to decide within a microsecond on the following action. This kind of challenge is specific to real-time feedback and does not exist for any of the other machine learning applications to quantum devices.

3. Model-free vs model-based RL

Applying model-free RL techniques, as described above, has a great advantage: the experimental quantum device can be treated as a black box, and its inner parameters and distortions of the control and measurement signals need not be known *a priori*. However, this also means that part of the training effort is spent on effectively learning an implicit model of the quantum device since that is the basis for a good control strategy.

In many situations relevant to modern quantum technologies, though, a good model is known since the Hamiltonian and Lindblad dissipation terms have been carefully calibrated. This allows one to consider model-based techniques explicitly; see Fig. 5(b). In principle, these can simply consist in applying model-free approaches to an RL-environment that is represented by a simulation of the model. However, this is useful only if running the experiment often would be expensive or time-consuming. A more direct approach takes gradients directly through the model dynamics. In the absence of feedback, this is what well-known approaches like GRAPE offer. In an interesting recent development, automatic differentiation (the cornerstone of deep learning frameworks) has been used to easily get access to the gradients needed for model-based control optimization. This was first presented in [106] and then applied to various quantum control tasks, especially for qubit systems [107,108], also employing neural networks to generate the control pulses [109,110].

Until recently, it was unclear, however, how to extend these ideas naturally to situations with feedback. The reason is that the stochastic choice of measurement outcomes is not directly compatible with taking gradients through smooth dynamics, unless special care is taken. A first example, applied to feedback based on weak linear measurements, was provided in [111], using automatic differentiation. The continuous measurement outcomes in such situations can be written as a simple function of a given Gaussian noise process. Very recently, a fully general approach, termed “feedback-GRAPE,”

was presented and analyzed in [112]. There it was pointed out how the effect of discrete stochastic measurements can be properly considered in such a gradient-based setting. This enables model-based optimization for feedback involving arbitrary strong discrete measurements, where the response to those outcomes can be represented via neural networks or trainable lookup tables.

4. Future challenges and opportunities

As of the time of writing, experimental applications of reinforcement-learning-based quantum control are still in their infancy. Even for the easier, open-loop control case, one has to set up a full pipeline where control sequences are delivered to the setup and a suitable reward is extracted from experimental measurement data, before being processed (e.g., externally, in a PC, implementing the RL algorithm). In a quantum system, where measurements collapse the state, this often means the reward can be obtained only at the end of an experimental run, making it harder to guide training. By contrast, having an immediate reward after each time step would improve training success because it assigns credit to the actions that immediately preceded a high reward.

The challenge becomes even larger when feedback control is called for. Then we require an agent (a neural network) that can process in real time the incoming measurement signals and decide on the subsequent actions. Depending on the hardware platform, this imposes severe constraints: for superconducting qubits, the time afforded to one such evaluation may be on the order of only a few hundred nanoseconds. Early proof-of-principle experiments have recently demonstrated the feasibility to control a single qubit [105]. How to control larger quantum systems is an important question for the future. This kind of challenge is specific to real-time feedback and does not exist for any of the other machine learning applications to quantum devices.

Almost all of the applications of model-free RL techniques mentioned above are numerical and rely on some simulation of the quantum device (if only because realizing these approaches in an experiment is still technically very demanding). In many publications, access to the simulation is furthermore utilized to make learning easier, e.g., by feeding the current quantum state obtained from the simulation as an “observation” into the agent. Since that would not be available in a real experiment, one possible solution is the use of a stochastic master equation to deduce the quantum evolution of the state based on noisy measurement traces during the experimental run in real time (see, e.g., [103] for comments on this). Another option is a “two-stage learning” procedure, where the successful RL agent, which still uses the state as input, is used for supervised training of another network that uses only measurement results as input. This new network can then be applied to an experiment [99]. However, both approaches require some calibration of experimental parameters since the original training still relies on simulations.

D. Discovering quantum experiments, protocols, and circuits

Inventing a blueprint for a quantum experiment, circuits or protocols requires ensuring that complex quantum phenomena play together to produce a quantum state or a quantum

transformation from a limited set of basic building blocks. In this section we will discuss design questions that involve discrete building, such as optical elements, superconducting circuit elements or quantum gates. While these systems also contain additional continuous parameters, the overarching discrete nature gives in some cases the possibility, as we will see, to *understand* and learn from the solutions.

1. Discovery of quantum experiments

An important and natural playground for discovery based on discrete building blocks is the invention of new experimental setups.

In the field of quantum optics, those building blocks contain lasers, nonlinear crystals, beam splitters, holograms, photon detectors, and the like. Put together in a specific way, those systems can lead to the generation of complex quantum entanglement or perform quantum teleportation—but exchange one single element, and the setup produces something entirely different and most likely not useful. Conventionally, experienced and creative scientists use their intuition and insights to design new quantum experiments. They translate an abstract task into a concrete layout that can be built in an experimental laboratory. However, human researchers are struggling to find suitable experimental setups for more complex quantum states and transformations.

This challenge was met recently with the introduction of automated discovery of quantum experiments from scratch [114,115]. The general idea is that an algorithm combines building blocks from a toolbox to produce a suitable experimental setup while optimizing for certain desired characteristics. In the setting of [114,115], the algorithm (here called *Melvin*), puts optical elements from a toolbox onto a virtual optical table. The toolbox consists of the experimental components available in the laboratory (such as lasers and crystals). The algorithm initially starts to put elements from the toolbox in random order on the table. If the candidate setup satisfies all sanity checks, a simulator computes the full experimental output. If the setup produces the desired quantum state, it is automatically simplified and reported to the user. In addition, the setup is then stored as a new part of the toolbox. In that way, over time, the algorithm learns useful macrocomponents which it can use in subsequent iterations. Thereby it can already access useful operations that significantly speed up the discovery process.

This algorithm has produced experimental blueprints that enabled the observation of numerous new quantum phenomena in laboratories [115]. Furthermore, new concepts and ideas have been discovered, understood, and generalized from some of the surprising solutions of the algorithm, such as an entirely new way to multiphoton interactions [116].

Sequentially building an experimental setup can be formulated as a reinforcement learning problem. This possibility has been explored in [117]. The approach led to the rediscovery of several experimental setups and to the automated simplification of experimental setups (which before was done using hand-crafted algorithms).

An alternative approach, called *Theseus*, which is orders of magnitudes faster than the previous techniques, is based on a new abstract representation of quantum exper-

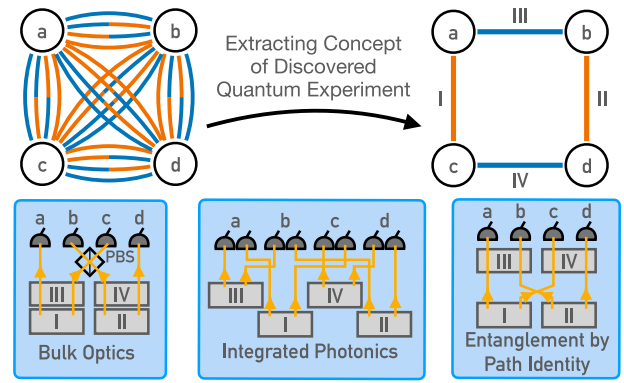


FIG. 6. Discovery of quantum experiments. Quantum optics experiments can be represented by colored graphs. Using the most general, complete graph as a starting representation, the AI's goal is to extract the conceptual core of the solution, which can then be understood by human scientists. The solution can then be translated to numerous different experimental configurations [113].

iments [113,118]. Here quantum experimental setups are translated into a graph-based representations; see Fig. 6. Any quantum optical experiment that can be built in the laboratory can be represented with a colored weighted graph, which translates an in principle infinite search space into a finite space with continuous (thus differentiable) parameters. In that way, the question of finding a certain quantum setup can be directly translated into discovering a graph with certain properties. As the parameters are continuous, highly efficient gradient-based optimization algorithms can be used to find the solutions. In addition, the graph's topology is eventually simplified, such that the human researcher is not only presented with a solution, but can immediately understand *why* and how the solution works, for example by discovering underlying resource structures [119]. The algorithm has been used to answer numerous open questions and has led to new concepts. It showcases that, when a simulator is available (i.e., a model) and no feedback needs to be taken into account, gradient-descent optimization at a large abstract representation often outperforms approaches such as genetic algorithms networks or reinforcement learning.

A different approach, denoted *Tachikoma*, aims at the discovery of new experimental setups for quantum metrology [70,71], using an evolutionary learning approach. The goal of *Tachikoma* is to find setups for quantum states that can measure phase shifts efficiently and with high precision. It uses a toolbox of optical elements from which it builds up a pool of candidate solutions. The next generation is produced from the best-performing parent setups. Those are merged and mutated to create the new pool of candidates. In that way, the population improves its performance over time and leads to numerous counterintuitive and exotic solutions. One of the computationally expensive operations and bottlenecks is the computation of the fidelity of the quantum state. For that, the authors have extended the approach by using a neural network that can classify the quantum states from the setup. This combination of an evolutionary algorithm with a neural network has discovered experimental blueprints with yet unachieved quantum metrology advantage. While much effort

has been put into constraining the system to realistic solutions, it remains to be seen whether these experiments can be built in the laboratory and achieve the expected quality.

An entirely different approach uses logical artificial intelligence for designing quantum optical experiments [120]. While the credo of the deep learning community is to build large neural networks that can solve arbitrary tasks given large enough training examples and compute power, this is not the only way towards “intelligent” algorithms. An alternative is logic AI [121]. Here the idea is to translate arbitrary problems to Boolean satisfiability expressions and solve them with powerful SAT solvers. In [120] the question of designing quantum experiments has been rephrased into logical expressions and solved with *MiniSAT*. It is shown that in some problems, a combination of Theseus and the logical approach is faster than the continuous optimization itself. The reason is that the unsatisfiability of candidate solutions is detected quickly with a logical approach, thus guiding the continuous optimizer towards more promising candidates. This approach is in its infancy. Given that the field of logic AI is growing fast due to computational and algorithmic advances, we expect a large increase in interest in this topic.

Deep generative models such as variational autoencoders became a standard tool in fields such as material design [122]. Here an encoder network transforms a (potentially discrete) representation into a continuous latent space. The decoder network is trained to take a point in the latent space and translate it back to the discrete structure. The encoder and decoder together are trained to perform an identity transformation, which by itself is not that interesting. However, as an exciting side effect, the system builds up an internal, continuous latent space that can be shaped during the training and used for gradient-based optimization. For the first time, such a system was demonstrated for quantum optics in [123]. The work focuses on understanding *what* the neural networks have learned and how they store the information in their internal latent space. The structure of the latent space shows surprising discrete structures that were then identified with concrete properties of the experimental setups. It will be interesting to see more advanced ways to investigate, navigate and understand the high-dimensional internal representations of neural networks that are built autonomously during training.

A conceptually related task is the design of superconducting circuits. The quantum behavior of superconducting circuits is defined by a network of inductances, capacitances, and Josephson junctions. As with quantum optical experiments, those systems are conventionally designed by experienced human researchers who aim to find suitable configurations for complex quantum transformations, such as coupling between two well-defined qubits in quantum computers. The search space of possible structures grows exponentially with the number of elements, and thus it quickly becomes infeasible for humans to find solutions for complex tasks. In [124] the authors addressed the question of designing superconducting circuits for the first time with a fully automated closed-loop optimization approach and designed a 4-local coupler by which four superconducting flux qubits interact. The algorithm *SCILLA* starts with a discrete circuit topology. The best candidates are further parametrically optimized, either with a direct gradient-based optimization or

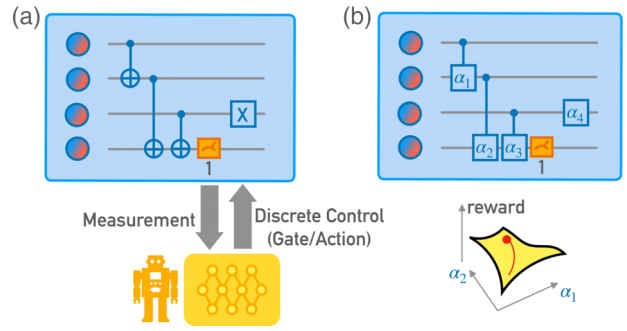


FIG. 7. Discovery of quantum circuits and feedback strategies with discrete gates. (a) A reinforcement-learning agent acts on a multiqubit system by selecting gates, potentially conditioned on measurement outcomes, finding an optimized quantum circuit or quantum feedback strategy. (b) A fixed layout quantum circuit with adjustable parameters that can be optimized via gradient ascent to achieve some goal like state preparation or variational ground state search (possibly including feedback).

with an evolutionary approach (to avoid local minima). The final design outperforms the only other (hand-crafted) 4-local coupler in terms of noise resilience and coupling strength.

2. Discovering quantum protocols and discrete feedback strategies

Discrete building blocks occur not only naturally in the construction of experiments, but also as part of discrete temporal sequences, specifically sequences of quantum gates and other operations. These sequences can represent quantum protocols or higher-level control strategies for quantum devices. The hardware-level control discussed in the previous Sec. III C could then be considered as a tool to implement the individual building blocks (e.g., an individual gate). In the following we will deal with protocols that also contain elements of feedback or other actions that are not merely unitary gates. The task of quantum circuit synthesis (building up unitaries out of elements) will be discussed further below.

Reinforcement learning is one suiting tool for the automated discovery of such sequences. This was first analyzed in [99], using deep RL, where the goal was to discover a strategy for quantum error correction in a quantum memory register made of a few qubits. This involves applying discrete unitary gates, which are conditioned on the outcomes of measurements, i.e., “real-time” feedback executed during the control sequence; see Fig. 7(a). Since the aim of [99] was primarily to find quantum error correction strategies, we will discuss some more aspects separately in the upcoming Sec. III E.

A reinforcement learning technique was subsequently also applied to the rediscovery of implementations for quantum communication protocols [125]. There the authors set up the task as an RL problem and explain the similarity of quantum communication and RL with the following intuition: A quantum communication protocol is a sequence of operations that leads to the desired outcome. Similarly, an RL agent learns a policy, that is, to perform sequences of operations that maximize a reward function. The authors task the RL agent to rediscover several important quantum communication schemes such as quantum teleportation, quantum state

purification, or entanglement swapping. Each of these tasks can be written as a simple network, where the nodes stand for the involved parties and edges indicate classical or quantum correlations between them. Let us take the quantum teleportation protocol as an example (the others follow similar ideas). The environment is a three-node network (the incoming unknown quantum state A, the sender B, and the receiver C). The environment starts with preshared entanglement between B and C. The agent now has to find a correct sequence of local measurements and classical communication steps that teleports the quantum state from A to C. After performing up to 50 operations, the transformations are evaluated, and the agent gets a (binary) reward for whether it succeeded or not. Over 100 000 trials, the agent finds with high probability an action-efficient strategy to perform the task. As there is no feedback from the environment and a model of the system exists, RL agents are not the only option to solve these questions, and direct gradient-based methods can be used for discovering quantum protocols.

3. Quantum circuits

The design of quantum circuits has some relation to the design of quantum experiments. In both situations, a discrete set of parametrized elements are carefully connected to form the topology of the circuit or experiment, while the continuous parameters of the elements (such as phases) are optimized. In contrast to quantum circuits that can use a universal gate set, for experimental design or the quantum protocols considered above it is not clear in the beginning whether certain targets can be reached with the available resources.

Quantum circuit design problems broadly fall into two branches: First, quantum circuit synthesis (QCS) (sometimes called quantum circuit compilation) addresses the problem of how to build from scratch a circuit that performs a specific task. Second, quantum circuit optimization (QCO) aims at turning a given circuit into a simplified, logically equivalent circuit.

The problem of quantum circuit synthesis is translating an algorithm into elementary gates from a finite universal set. There are two situations: the fully discrete case, where gates are fixed, and the case where gates can be tuned via continuous parameters (e.g., rotation angles). The first application of machine-learning techniques for the de-novo generation of quantum circuits used genetic algorithms [126]. The algorithm had access to a set of single- and two-qubit gates and was tasked to rediscover the quantum circuit for quantum teleportation. It indeed found the correct circuit with significantly less evaluations than an exhaustive search and could also present different solutions not discussed in the literature before. Genetic algorithms are a powerful tool for discrete discovery. Thus, related approaches are still being used two decades later [127,128].

The problem of designing a quantum circuit from an unparameterized set of elementary gates can be formulated as a reinforcement learning problem. In [129] the authors introduce a deep reinforcement learning algorithm that translates an arbitrary single-qubit gate into a sequence of elementary gates from a finite universal set for a topological quantum computer. The authors apply their algorithms to Fibonacci

anyons and discover high-quality braiding sequences. Similar machine learning techniques have been applied to the quantum circuits for gate-based quantum computers [130]. The design of unparameterized quantum circuits has an important application for fault-tolerant quantum computation, where only a finite list of unparameterized gates can be applied to the logical qubits.

In general, quantum circuits can have tunable parameters, for instance, parameterized X-gates. In that case, the problem has both a discrete and continuous element. An important application for this task is the hardware-aware design of circuits [131,132]. Here the algorithms consider the circuit's connectivity on the one hand and noise contributions on the other. The main goal is to find shallow circuits that are more noise-resistant than other (textbook) implementations of circuits. The authors of [131,132] present an algorithm that can significantly outperform textbook solutions for various state generation tasks in terms of fidelity under realistic noise conditions. The design of discrete circuits with continuous parameters has also been approached with deep reinforcement learning [133].

A crucial and heavily investigated topic in the area of quantum circuit design is the parametric optimization of a constant circuit topology. This task is essential for hybrid quantum-classical variational quantum algorithms (VQAs) that can run on near-term quantum computing hardware [23,24], as well as for quantum machine learning, where such parameterized circuits are used as quantum neural networks [21]. We will not go into the details of these topics, but refer the reader to the excellent reviews on this topic.

We will mention only a few selected but important results and ideas for completeness. First, it has been discovered that the gradients in a randomly initialized parametrized quantum circuit vanish exponentially with the number of qubits [134], which has become one of the main challenges in the field of quantum machine learning and VQAs; see Fig. 7(b). An important question then is how an expressive initial state of the circuit (Ansatz) allows for the efficient machine-learning-based optimization of the circuits [135]. Some exciting approaches involve reinforcement learning that explores economic and expressive initial Ansätze [136] or ideas that are inspired from neural network architecture search [137]. Besides the direct gradient-based optimization of parametrized quantum circuits, different approaches try to avoid the problem of vanishing gradients, by employing reinforcement learning [138], using ML-based prediction of suitable initial parameters (rather than optimizing the parameters directly) [139] or advanced gradient-free approaches that are naturally not susceptible to the barren plateau problem [140].

An interesting recent application of VQA-based systems is the *quantum-computer-aided design* of quantum hardware [141–143]. As described at the beginning of this chapter, the AI-based design of new quantum hardware on a classical computer has the problem of memory requirements increasing exponentially with the system size. One way to overcome this problem is to outsource the computation of the expensive quantum system to a quantum computer. Here the problem of designing new multiqubit couplers for superconducting quantum computers or the design of new quantum optics hardware

can be rephrased as in a VQA-style problem. A classical AI algorithm changes the parameter of a parameterized quantum circuit to minimize a fidelity function computed from the outcome of the quantum computation. After convergence, a mapping translates the final parametrized quantum circuit into the specific quantum hardware. This approach has been experimentally demonstrated in a proof-of-principle three-qubit superconducting circuit [143].

In general, it is not guaranteed that a direct compilation of an algorithm already yields the most efficient implementation of a quantum circuit. A powerful classical method to simplify (compile) quantum circuits is the ZX formalism [144], which reformulates the circuit into a graph, where predefined rules identify simplifications. However, this and similar approaches have been formulated in a hardware-independent way, operating on a global level. Alternatively, this problem can be approached by RL algorithms [145] that can autonomously simplify circuits, for example, in terms of circuit depth or gate counts, and this enables easily taking into account concrete hardware constraints. In [145] this approach was developed and found superior to simulated annealing (tested for circuits of up to 50 qubits). It has the potential to become an important tool for simplifying quantum circuits in the future.

4. Future challenges and opportunities

The simulation of quantum experiments becomes expensive as soon as the system grows in size. Neural networks could autonomously find approximate predictions for the dynamics of the quantum system. Such supervised systems need a lot of training examples. Thus the trade-off between the creation of training examples and the computational benefit of an approximation needs to be investigated.

The design of new experiments or hardware can be seen not only as *optimization* (in the sense of making an existing structure better) but as *discovery* in which we create new ideas that did not exist before, as shown in [113,118]. This point of view shows how machines can creatively contribute to science and act as an inspiration for human scientists. There will be a great potential for expanding these ideas. Automatic extraction of understandable building blocks (“subroutines”) can help with this challenge.

E. Quantum error correction

The ability to correct errors in a quantum computing device will be indispensable to realizing beneficial applications of quantum computation since real-world devices are not coherent enough to run an error-free calculation. The basic conceptual ideas in this domain are known since the pioneering work of Shor [146] and subsequent developments, most notably the surface code. In any case, the idea is to encode logical qubit information in many physical qubits robustly and redundantly. The presence of errors (like qubit dephasing and decay) must be detected via measurement of so-called syndromes, i.e., suitably chosen observables (often multiqubit operators). Finally, a good way to interpret the observed syndromes and apply some error correction procedure must be found. Despite the knowledge of good encodings and suitable syndromes, it remains a challenging problem how to best implement those in practice, for a given quantum device, with

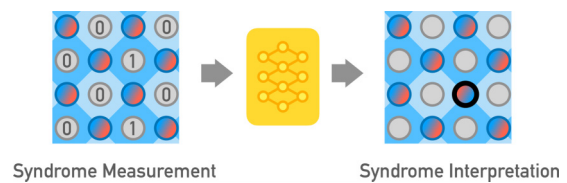


FIG. 8. Quantum error correction. Syndrome interpretation in a surface code as a task that a neural network can be trained to perform.

its available gate set and topology of connections between the qubits, and how to optimize them for a given noise model.

1. Syndrome interpretation

On the simplest level, we already assume an existing encoding and a fixed set of syndromes. The task then is to find the optimal way to interpret the observed syndrome, e.g., deciding which qubits are likely erroneous and must be corrected; see Fig. 8. This can be phrased as a supervised learning problem, where some errors are simulated, the syndrome is fed into a network, and the network must announce the location of the errors. In practice, the surface code is the most promising QEC architecture, and deducing the error from the syndrome is not trivial, though non-ML algorithms exist. Multiple works therefore trained neural networks to yield “neural decoders” [147–153]. In one early example [147], a modified, restricted Boltzmann machine was used, with two types of visible units, corresponding to syndrome and underlying error configuration. This was then trained on a data set of such pairs. Afterward, the machine could be used to sample the errors compatible with an observed syndrome. It is also possible to use reinforcement learning to discover better strategies in more complicated situations. In [152] this was applied to the surface code, exemplified in a situation with faulty syndrome measurements. In practical settings, the time required to decode the error syndrome of a surface codes scales badly with the size of the code. To overcome this challenge, the authors of [153] have developed a hierarchical decoder that combines a convolutional neural network to identify local errors with a conventional algorithm to detect long-range errors. The combined approach can scale up to large number of qubits while still offering a runtime only in the orders of milliseconds.

2. Code search

Going one step further, the question arises whether a machine can also find better codes. It is helpful to take an existing code and modify it. The surface code is usually formulated on a simple square lattice, but it can be implemented on more complicated geometries, both periodic and even more generally aperiodic. In [154] a reinforcement learning agent was asked to optimize the connectivity of a surface code given a number of data qubits. It was able to find the best performing code in scenarios important for real experiments. These included biased noise (not all error channels equally strong) or spatially localized noise (higher error rates in the vicinity of some qubits). The agent found interesting nontrivial connectivities as optimal solutions. Due to the availability of highly efficient simulation tools for establishing the performance of

surface codes, the authors of [154] were able to go up to 70 data qubits.

Autonomous quantum error correction consists of an experimental configuration that can intrinsically correct certain types of errors without active feedback. This idea can be implemented by introducing carefully additional drives and dissipation. The discovery of such mechanisms in real physical systems, under strict experimental constraints, is highly nontrivial. The authors of [155] show the automated discovery of an autonomous QEC that could be applied to Bosonic systems. The goal is to find an encoding of a logical qubit that is robust under the dynamics of the system. The algorithm denoted *AutoQEC* can then discover such an encoding by maximizing the average fidelity of the logical qubit. *AutoQEC* is further constrained to consider only systems within experimental capabilities. Indeed, the authors discover a new quantum code, denoted $\sqrt{3}$ -code, that has a longer lifetime than previously studied systems with the same concrete experimental constraints. The authors go on and, inspired by their numerical discovery, derive the analytical, logical state and analyze the new autonomous QEC system further.

3. Learning QEC circuit parameters

High-quality QEC codes are known for many setups, like the Gottesmann-Kitaev-Preskill (GKP) code for microwave cavities. However, due to imperfections in the calibration of the experiment, the codes cannot be realized in their most efficient way. In a pioneering experiment, [156], the authors implement the GKP code in a microwave cavity setup, where continuous parameters parametrize the required quantum gates. The parameters of the QEC circuit are optimized *in situ* via model-free RL to overcome the limitations given by the calibration. The resulting QEC scheme surpasses the lifetime of the uncorrected system by more than a factor of two.

4. Full QEC protocol discovery

Finally, one can adopt the attitude that neither the code itself, the code family, nor any other ingredients are assumed. In that case, one starts from scratch, and the goal of the machine is to discover ways to preserve the quantum information with high fidelity for as long as possible. In other words, it (re-)discovers all aspects of QEC and error mitigation, adapted to the given platform and noise model. Such an *ab initio* approach was demonstrated in [99], which we already mentioned above in the context of RL for feedback. Given a few qubits with arbitrary connectivity and gate set, as well as an arbitrary noise model, the agent is asked to preserve the quantum information as long as possible. To solve this challenging task, additional generally applicable insights were required, e.g., introducing a reward that can measure the amount of surviving quantum information without having discovered a proper decoding sequence. Beyond approaches that fall in the family of stabilizer codes, the same agent also discovered noise mitigation techniques based on adaptive measurements. The advantage of such an *ab initio* approach is its flexibility, but the price to pay is that so far, it works for only a handful of qubits due to the effort required in simulations. A future

challenge would be finding ways to make the RL work directly on the experiment.

5. Future challenges and opportunities

Regarding syndrome measurements, an important challenge for the future is ensuring that the neural networks interpreting those measurement results can be deployed in an actual device at sufficient speed: even for the classical algorithms, this is nontrivial. Another challenge is that *ab initio* discovery of quantum error correction strategies still relies on simulations, whose numerical effort scales exponentially in the number of qubits (for general quantum dynamics). An interesting direction could also be the codiscovery of autonomous QEC experiments together with QEC feedback strategies in these systems.

IV. OUTLOOK

With all these promising ideas in mind, let us look forward to the year 2035: how do we imagine machine learning to contribute to quantum technologies by that time?

1. Fully controlled and error-corrected quantum systems

As quantum platforms scale towards ever greater numbers of components and connections, machine learning will provide a way to harness this complexity—by automatically calibrating and fine-tuning the resulting huge number of parameters adaptively, by discovering optimized experimental setups in the first place, by extracting the maximum amount of information from rather indirect measurements of the underlying phenomena via complex observables, and by finding smart control strategies. Errors in such systems could be corrected by fully automated quantum error correction schemes discovered by an AI. When we think of quantum computers with thousands of qubits or quantum simulators with even more degrees of freedom, all of this will be a crucial part of the community's toolbox.

2. Specifying goals, not algorithms

One important aspect here will be that instead of defining an algorithm that tells the computer *how* to achieve some goal, we will typically *define the goal itself*. Such goals could be to retain a large fidelity during quantum operations, produce highly entangled states, or have a strong sensitivity to some external signal. The details of how to reach this goal will be left for the computer to discover. This change of perspective will enable a much higher level of description, which is one way to keep ahead of the growing complexity. Ultimately one might expect that the machine has access to the scientific literature and suggests goals and new experiments autonomously, as demonstrated in material science [157].

3. Discovering new algorithms

Rather than discovering experiments or feedback strategies, it will be very interesting to see whether ML agents can autonomously discover other higher-logic quantum programs such as quantum algorithms. This task is recently been tackled by large language models for classical algorithms [158,159],

and we expect that similarly quantum algorithms can be discovered with classical machine learning models.

4. How can the human learn?

Suppose that computers will be able to help us find solutions for many of the lower-level and even some higher-level, more conceptual tasks in the domain of quantum technologies. That raises the following notorious question, pervasive throughout machine learning and artificial intelligence: How can we human scientists understand what the machine has learned? Do we need to open the black box of neural networks, or can we use the algorithms as a source of inspiration in a different way [160]? We argue that while improved performance in the task at hand is great, being able to understand the essence of what the machine has discovered is crucial for the result to become of much wider applicability. In general, gaining *understanding* has been called the *essential aim of science* [161]. Here approaches where the solution involves discrete steps (e.g., discrete actions of an agent) or logic-based AI seem to be easier to interpret, explain and understand than results from deep learning-based methods. The field of symbolic regression (which extracts discrete explanations of neural network predictions) might be very fruitful in this approach.

5. What needs to be done?

To attain the visions described above, our community may adopt some proven methodologies from other areas. The idea of fair benchmarks and competitions is one of the powerful driving forces in the development of ML algorithms. One of the most famous examples being the ImageNet data set which provided the basis for a revolution of ML-based computer vision systems [162]. This idea was adapted in other fields

of science that apply AI methodologies, such as material discovery [163,164]. In contrast, the field of AI in quantum technology, at the moment, appears more like the wild west. There are no clear ways how to compare approaches from different papers, because most works apply their approaches to slightly different tasks, making them incomparable. We believe that fair and suitably curated benchmark data sets will steer the development of powerful and ever more generally applicable AI algorithms in quantum technology. The data sets could consist of simulated or (in the best case) experimental data for data interpretation tasks. Likewise, to facilitate the discovery of experimental setups and protocols, the community can develop a selection of well-curated objective functions and a set of simulated environments describing important prototypical quantum devices (see SciGym for a first attempt at this [165]). In a similar direction, we expect that cloud access to real quantum experiments will become available for significantly more systems. AI algorithms can then be trained on the data from these real machines with specific experimental constraints (such as connectivity or noise). This will boost the capabilities of algorithms that deal with important, real-world systems.

Finally, what Alan Turing remarked in his visionary article on intelligence and learning machines [166] is also valid here, in the field of machine learning applied to quantum technologies: “We can only see a short distance ahead, but we can see plenty there that needs to be done.”

ACKNOWLEDGMENTS

The authors thank S. Arlt and X. Gu for helpful comments on the manuscript. F.M. acknowledges funding by the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus.

-
- [1] M. I. Jordan and T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, *Science* **349**, 255 (2015).
 - [2] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature (London)* **521**, 436 (2015).
 - [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
 - [4] J. P. Dowling and G. J. Milburn, Quantum technology: The second quantum revolution, *Philos. Trans. R. Soc. London A* **361**, 1655 (2003).
 - [5] F. Flamini, N. Spagnolo, and F. Sciarrino, Photonic quantum information processing: A review, *Rep. Prog. Phys.* **82**, 016001 (2019).
 - [6] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, *Rev. Mod. Phys.* **86**, 153 (2014).
 - [7] J. Preskill, Quantum Computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
 - [8] M. C. Angelini and F. Ricci-Tersenghi, Cracking nuts with a sledgehammer: When modern graph neural networks do worse than classical greedy algorithms, [arXiv:2206.13211](https://arxiv.org/abs/2206.13211).
 - [9] F. Marquardt, Machine learning and quantum devices, *SciPost Phys. Lect. Notes* **29** (2021).
 - [10] A. Dawid, J. Arnold, B. Requena, A. Gresch, M. Płodzień, K. Donatella, K. A. Nicoli, P. Stornati, R. Koch, M. Büttner *et al.*, Modern applications of machine learning in quantum sciences, [arXiv:2204.04198](https://arxiv.org/abs/2204.04198).
 - [11] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, *Rev. Mod. Phys.* **91**, 045002 (2019).
 - [12] P. Mehta, M. Bukov, C.-H. Wang, A. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, *Phys. Rep.* **810**, 1 (2019).
 - [13] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: A review of recent progress, *Rep. Prog. Phys.* **81**, 074001 (2018).
 - [14] J. Carrasquilla, Machine learning for quantum matter, *Adv. Phys. X* **5**, 1797528 (2020).
 - [15] G. Torlai and R. G. Melko, Machine-learning quantum states in the NISQ era, *Annu. Rev. Condens. Matter Phys.* **11**, 325 (2020).
 - [16] E. Polino, M. Valeri, N. Spagnolo, and F. Sciarrino, Photonic quantum metrology, *AVS Quantum Sci.* **2**, 024703 (2020).
 - [17] F. Vernuccio, A. Bresci, V. Cimini, A. Giuseppi, G. Cerullo, D. Polli, and C. M. Valensise, Artificial intelligence in classical and quantum photonics, *Laser Photonics Rev.* **16**, 2100399 (2022).

- [18] V. Gebhart, R. Santagati, A. A. Gentile, E. Gauger, D. Craig, N. Ares, L. Bianchi, F. Marquardt, L. Pezze, and C. Bonato, Learning quantum systems, [arXiv:2207.00298](https://arxiv.org/abs/2207.00298).
- [19] M. Schuld, I. Sinayskiy, and F. Petruccione, An introduction to quantum machine learning, *Contemp. Phys.* **56**, 172 (2015).
- [20] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, *Nature (London)* **549**, 195 (2017).
- [21] V. Dunjko and P. Wittek, A non-review of quantum machine learning: Trends and explorations, *Quantum Views* **4**, 32 (2020).
- [22] L. Lamata, Quantum machine learning and quantum biomimetics: A perspective, *Mach. Learn.: Sci. Tech.* **1**, 033002 (2020).
- [23] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
- [24] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, Noisy intermediate-scale quantum algorithms, *Rev. Mod. Phys.* **94**, 015004 (2022).
- [25] K. Bharti, T. Haug, V. Vedral, and L.-C. Kwek, Machine learning meets quantum foundations: A brief survey, *AVS Quantum Sci.* **2**, 034101 (2020).
- [26] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. (Pearson, 2021).
- [27] <https://github.com/ML4QTech/Collection>.
- [28] D. Whitley, A genetic algorithm tutorial, *Stat. Comput.* **4**, 65 (1994).
- [29] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms* (Springer, Berlin, 2008).
- [30] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control Signal Syst.* **2**, 303 (1989).
- [31] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.* **9**, 1735 (1997).
- [32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning representations by back-propagating errors, *Nature (London)* **323**, 533 (1986).
- [33] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Netw.* **61**, 85 (2015).
- [34] M. A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, 2015).
- [35] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, Mastering the game of Go with deep neural networks and tree search, *Nature (London)* **529**, 484 (2016).
- [36] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, Grandmaster level in StarCraft II using multi-agent reinforcement learning, *Nature (London)* **575**, 350 (2019).
- [37] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA, 2018).
- [38] E. Magesan, J. M. Gambetta, A. D. Córcoles, and J. M. Chow, Machine Learning for Discriminating Quantum Measurement Trajectories and Improving Readout, *Phys. Rev. Lett.* **114**, 200501 (2015).
- [39] A. Seif, K. A. Landsman, N. M. Linke, C. Figgatt, C. Monroe, and M. Hafezi, Machine learning assisted readout of trapped-ion qubits, *J. Phys. B: At. Mol. Opt. Phys.* **51**, 174006 (2018).
- [40] G. Liu, M. Chen, Y.-X. Liu, D. Layden, and P. Cappellaro, Repetitive readout enhanced by machine learning, *Mach. Learn. Sci. Tech.* **1**, 015003 (2020).
- [41] E. Flurin, L. S. Martin, S. Hacothen-Gourgy, and I. Siddiqi, Using a Recurrent Neural Network to Reconstruct Quantum Dynamics of a Superconducting Qubit from Physical Observations, *Phys. Rev. X* **10**, 011006 (2020).
- [42] I. Agresti, N. Viggianiello, F. Flamini, N. Spagnolo, A. Crespi, R. Osellame, N. Wiebe, and F. Sciarrino, Pattern Recognition Techniques for Boson Sampling Validation, *Phys. Rev. X* **9**, 011013 (2019).
- [43] S. Aaronson, The learnability of quantum states, *Proc. R. Soc. A* **463**, 3089 (2007).
- [44] A. Rocchetto, S. Aaronson, S. Severini, G. Carvacho, D. Poderini, I. Agresti, M. Bentivegna, and F. Sciarrino, Experimental learning of quantum states, *Sci. Adv.* **5**, eaau1946 (2019).
- [45] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill *et al.*, Quantum advantage in learning from experiments, *Science* **376**, 1182 (2022).
- [46] V. Cimini, M. Barbieri, N. Treps, M. Walschaers, and V. Parigi, Neural Networks for Detecting Multimode Wigner Negativity, *Phys. Rev. Lett.* **125**, 160504 (2020).
- [47] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, *Nat. Phys.* **13**, 431 (2017).
- [48] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, *Nat. Phys.* **13**, 435 (2017).
- [49] S. J. Wetzel, Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders, *Phys. Rev. E* **96**, 022140 (2017).
- [50] K. Kottmann, P. Huembeli, M. Lewenstein, and A. Acín, Unsupervised Phase Discovery with Deep Anomaly Detection, *Phys. Rev. Lett.* **125**, 170603 (2020).
- [51] A. Dawid, P. Huembeli, M. Tomza, M. Lewenstein, and A. Dauphin, Phase detection with neural networks: Interpreting the black box, *New J. Phys.* **22**, 115001 (2020).
- [52] N. Kåmning, A. Dawid, K. Kottmann, M. Lewenstein, K. Sengstock, A. Dauphin, and C. Weitenberg, Unsupervised machine learning of topological phase transitions from experimental data, *Mach. Learn. Sci. Tech.* **2**, 035037 (2021).
- [53] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature (London)* **574**, 505 (2019).
- [54] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602 (2017).
- [55] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, Restricted Boltzmann machines in quantum physics, *Nat. Phys.* **15**, 887 (2019).
- [56] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nat. Phys.* **14**, 447 (2018).
- [57] E. S. Tiunov, V. V. Tiunova, A. E. Ulanov, A. Lvovsky, and A. K. Fedorov, Experimental quantum homodyne tomography via machine learning, *Optica* **7**, 448 (2020).

- [58] M. Neugebauer, L. Fischer, A. Jäger, S. Czischek, S. Jochim, M. Weidemüller, and M. Gärtner, Neural-network quantum state tomography in a two-qubit experiment, *Phys. Rev. A* **102**, 042604 (2020).
- [59] G. Torlai, B. Timar, E. P. L. van Nieuwenburg, H. Levine, A. Omran, A. Keesling, H. Bernien, M. Greiner, V. Vuletić, M. D. Lukin *et al.*, Integrating Neural Networks with a Quantum Simulator for State Reconstruction, *Phys. Rev. Lett.* **123**, 230504 (2019).
- [60] M. Schuld, I. Sinayskiy, and F. Petruccione, Neural networks take on open quantum systems, *Physics* **12**, 74 (2019).
- [61] A. Nagy and V. Savona, Variational Quantum Monte Carlo Method with a Neural-Network Ansatz for Open Quantum Systems, *Phys. Rev. Lett.* **122**, 250501 (2019).
- [62] M. J. Hartmann and G. Carleo, Neural-Network Approach to Dissipative Quantum Many-Body Dynamics, *Phys. Rev. Lett.* **122**, 250502 (2019).
- [63] F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, Variational Neural-Network Ansatz for Steady States in Open Quantum Systems, *Phys. Rev. Lett.* **122**, 250503 (2019).
- [64] N. Yoshioka and R. Hamazaki, Constructing neural stationary states for open quantum many-body systems, *Phys. Rev. B* **99**, 214306 (2019).
- [65] T. Adler, M. Erhard, M. Krenn, J. Brandstetter, J. Kofler, and S. Hochreiter, Quantum optical experiments modeled by long short-term memory, *Photonics* **8**, 535 (2021).
- [66] N. Mohseni, T. Fösel, L. Guo, C. Navarrete-Benlloch, and F. Marquardt, Deep learning of quantum many-body dynamics via random driving, *Quantum* **6**, 714 (2022).
- [67] M. Choi, D. Flam-Shepherd, T. H. Kyaw, and A. Aspuru-Guzik, Learning quantum dynamics with latent neural ordinary differential equations, *Phys. Rev. A* **105**, 042403 (2022).
- [68] D. T. Lennon, H. Moon, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, E. A. Laird, and N. Ares, Efficiently measuring a quantum device using machine learning, *npj Quantum Inf.* **5**, 79 (2019).
- [69] V. Cimini, I. Gianani, N. Spagnolo, F. Leccese, F. Sciarrino, and M. Barbieri, Calibration of Quantum Sensors by Neural Networks, *Phys. Rev. Lett.* **123**, 230502 (2019).
- [70] P. A. Knott, A search algorithm for quantum state engineering and metrology, *New J. Phys.* **18**, 073033 (2016).
- [71] R. Nichols, L. Mineh, J. Rubio, J. C. F. Matthews, and P. A. Knott, Designing quantum experiments with a genetic algorithm, *Quantum Sci. Tech.* **4**, 045012 (2019).
- [72] D. W. Berry and H. M. Wiseman, Optimal States and Almost Optimal Adaptive Measurements for Quantum Interferometry, *Phys. Rev. Lett.* **85**, 5098 (2000).
- [73] D. W. Berry, H. M. Wiseman, and J. K. Breslin, Optimal input states and feedback for interferometric phase estimation, *Phys. Rev. A* **63**, 053804 (2001).
- [74] B. L. Higgins, D. W. Berry, S. D. Bartlett, H. M. Wiseman, and G. J. Pryde, Entanglement-free Heisenberg-limited phase estimation, *Nature (London)* **450**, 393 (2007).
- [75] A. Hentschel and B. C. Sanders, Machine Learning for Precise Quantum Measurement, *Phys. Rev. Lett.* **104**, 063603 (2010).
- [76] A. Hentschel and B. C. Sanders, Efficient Algorithm for Optimizing Adaptive Quantum Metrology Processes, *Phys. Rev. Lett.* **107**, 233601 (2011).
- [77] N. B. Lovett, C. Crosnier, M. Perarnau-Llobet, and B. C. Sanders, Differential Evolution for Many-Particle Adaptive Quantum Metrology, *Phys. Rev. Lett.* **110**, 220501 (2013).
- [78] P. Palittapongarnpim, P. Wittek, E. Zahedinejad, S. Vedaie, and B. C. Sanders, Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics, *Neurocomputing* **268**, 116 (2017).
- [79] H. Xu, J. Li, L. Liu, Y. Wang, H. Yuan, and X. Wang, Generalizable control for quantum parameter estimation through reinforcement learning, *npj Quantum Inf.* **5**, 82 (2019).
- [80] J. Schuff, L. J. Fiderer, and D. Braun, Improving the dynamics of quantum sensors with reinforcement learning, *New J. Phys.* **22**, 035001 (2020).
- [81] E. G. Ryan, C. C. Drovandi, J. M. McGree, and A. N. Pettitt, A review of modern computational algorithms for Bayesian optimal design, *Int. Stat. Rev.* **84**, 128 (2016).
- [82] R. Durrer, B. Kratochwil, J. V. Koski, A. J. Landig, C. Reichl, W. Wegscheider, T. Ihn, and E. Greplová, Automated Tuning of Double Quantum Dots into Specific Charge States Using Neural Networks, *Phys. Rev. Appl.* **13**, 054019 (2020).
- [83] N. Wiebe, C. Granade, C. Ferrie, and D. G. Cory, Hamiltonian Learning and Certification Using Quantum Resources, *Phys. Rev. Lett.* **112**, 190501 (2014).
- [84] J. Wang, S. Paesani, R. Santagati, S. Knauer, A. A. Gentile, N. Wiebe, M. Petruzzella, J. L. O'Brien, J. G. Rarity, A. Laing *et al.*, Experimental quantum Hamiltonian learning, *Nat. Phys.* **13**, 551 (2017).
- [85] A. A. Gentile, B. Flynn, S. Knauer, N. Wiebe, S. Paesani, C. E. Granade, J. G. Rarity, R. Santagati, and A. Laing, Learning models of quantum systems from experiments, *Nat. Phys.* **17**, 837 (2021).
- [86] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms, *J. Magn. Reson.* **172**, 296 (2005).
- [87] V. F. Krotov, *Global Methods in Optimal Control Theory*, Monographs and Textbooks in Pure and Applied Mathematics, Vol. 195 (Marcel Dekker, New York, 1996).
- [88] J. Somló, V. A. Kazakov, and D. J. Tannor, Controlled dissociation of I_2 via optical transitions between the X and B electronic states, *Chem. Phys.* **172**, 85 (1993).
- [89] P. Doria, T. Calarco, and S. Montangero, Optimal Control Technique for Many-Body Quantum Dynamics, *Phys. Rev. Lett.* **106**, 190501 (2011).
- [90] R. S. Judson and H. Rabitz, Teaching Lasers to Control Molecules, *Phys. Rev. Lett.* **68**, 1500 (1992).
- [91] C. Chen, D. Dong, H.-X. Li, J. Chu, and T.-J. Tarn, Fidelity-based probabilistic Q-learning for control of quantum systems, *IEEE Trans. Neural Netw. Learn. Syst.* **25**, 920 (2013).
- [92] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement Learning in Different Phases of Quantum Control, *Phys. Rev. X* **8**, 031086 (2018).
- [93] M. August and J. M. Hernández-Lobato, Taking gradients through experiments: LSTMs and memory proximal policy optimization for black-box quantum control, in *High Performance Computing*, edited by R. Yokota, M. Weiland, J. Shalf, and S. Alam (Springer International Publishing, Cham, 2018), pp. 591–613.

- [94] X.-M. Zhang, Z.-W. Cui, X. Wang, and M.-H. Yung, Automatic spin-chain learning to explore the quantum speed limit, *Phys. Rev. A* **97**, 052333 (2018).
- [95] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, Universal quantum control through deep reinforcement learning, *npj Quantum Inf.* **5**, 33 (2019).
- [96] Z. An and D. L. Zhou, Deep reinforcement learning for quantum gate control, *Europhys. Lett.* **126**, 60002 (2019).
- [97] P. A. Erdman and F. Noé, Identifying optimal cycles in quantum thermal machines with reinforcement-learning, *npj Quantum Inf.* **8**, 1 (2022).
- [98] Y. Baum, M. Amico, S. Howell, M. Hush, M. Liuzzi, P. Mundada, T. Merkh, A. R. R. Carvalho, and M. J. Biercuk, Experimental deep reinforcement learning for error-robust gate-set design on a superconducting quantum computer, *PRX Quantum* **2**, 040324 (2021).
- [99] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Reinforcement Learning with Neural Networks for Quantum Feedback, *Phys. Rev. X* **8**, 031084 (2018).
- [100] J. Mackeprang, D. B. R. Dasari, and J. Wrachtrup, A reinforcement learning approach for quantum state engineering, *Quantum Mach. Intel.* **2**, 5 (2020).
- [101] Z. T. Wang, Y. Ashida, and M. Ueda, Deep Reinforcement Learning Control of Quantum Cartpoles, *Phys. Rev. Lett.* **125**, 100401 (2020).
- [102] S. Borah, B. Sarma, M. Kewming, G. J. Milburn, and J. Twamley, Measurement-Based Feedback Quantum Control with Deep Reinforcement Learning for a Double-Well Nonlinear Potential, *Phys. Rev. Lett.* **127**, 190403 (2021).
- [103] R. Porotti, A. Essig, B. Huard, and F. Marquardt, Deep reinforcement learning for quantum state preparation with weak nonlinear measurements, *Quantum* **6**, 747 (2022).
- [104] V. V. Sivak, A. Eickbusch, H. Liu, B. Royer, I. Tsioutsios, and M. H. Devoret, Model-Free Quantum Control with Reinforcement Learning, *Phys. Rev. X* **12**, 011059 (2022).
- [105] K. Reuer, J. Landgraf, T. Fösel, J. O'Sullivan, L. Beltrán, A. Akin, G. J. Norris, A. Remm, M. Kerschbaum, J.-C. Besse *et al.*, Realizing a deep reinforcement learning agent discovering real-time feedback control strategies for a quantum system, *arXiv:2210.16715*.
- [106] N. Leung, M. Abdelhafez, J. Koch, and D. Schuster, Speedup for quantum optimal control from automatic differentiation based on graphics processing units, *Phys. Rev. A* **95**, 042318 (2017).
- [107] M. Abdelhafez, D. I. Schuster, and J. Koch, Gradient-based optimal control of open quantum systems using quantum trajectories and automatic differentiation, *Phys. Rev. A* **99**, 052327 (2019).
- [108] M. Abdelhafez, B. Baker, A. Gyenis, P. Mundada, A. A. Houck, D. Schuster, and J. Koch, Universal gates for protected superconducting qubits using optimal control, *Phys. Rev. A* **101**, 022321 (2020).
- [109] F. Schäfer, M. Kloc, C. Bruder, and N. Lörch, A differentiable programming method for quantum control, *Mach. Learn. Sci. Tech.* **1**, 035009 (2020).
- [110] L. Coopmans, D. Luo, G. Kells, B. K. Clark, and J. Carrasquilla, Protocol discovery for the quantum control of majoranas by differentiable programming and natural evolution strategies, *PRX Quantum* **2**, 020332 (2021).
- [111] F. Schäfer, P. Sekatski, M. Koppenhöfer, C. Bruder, and M. Kloc, Control of stochastic quantum dynamics by differentiable programming, *Mach. Learn. Sci. Tech.* **2**, 035004 (2021).
- [112] R. Porotti, V. Peano, and F. Marquardt, Gradient ascent pulse engineering with feedback, *arXiv:2203.04271*.
- [113] M. Krenn, J. S. Kottmann, N. Tischler, and A. Aspuru-Guzik, Conceptual Understanding through Efficient Automated Design of Quantum Optical Experiments, *Phys. Rev. X* **11**, 031044 (2021).
- [114] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger, Automated Search for new Quantum Experiments, *Phys. Rev. Lett.* **116**, 090405 (2016).
- [115] M. Krenn, M. Erhard, and A. Zeilinger, Computer-inspired quantum experiments, *Nat. Rev. Phys.* **2**, 649 (2020).
- [116] X. Gao, M. Erhard, A. Zeilinger, and M. Krenn, Computer-Inspired Concept for High-Dimensional Multipartite Quantum Gates, *Phys. Rev. Lett.* **125**, 050501 (2020).
- [117] A. A. Melnikov, H. P. Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, Active learning machine learns to create new quantum experiments, *Proc. Natl. Acad. Sci. USA* **115**, 1221 (2018).
- [118] C. Ruiz-Gonzalez, S. Arlt, J. Petermann, S. Sayyad, T. Jaouni, E. Karimi, N. Tischler, X. Gu, and M. Krenn, Digital discovery of 100 diverse quantum experiments with Pytheus, *arXiv:2210.09980*.
- [119] S. Arlt, C. Ruiz-Gonzalez, and M. Krenn, Digital discovery of a scientific concept at the core of experimental quantum optics, *arXiv:2210.09981*.
- [120] A. Cervera-Lierta, M. Krenn, and A. Aspuru-Guzik, Design of quantum optical experiments with logic artificial intelligence, *Quantum* **6**, 836 (2022).
- [121] M. J. H. Heule and O. Kullmann, The science of brute force, *Comm. ACM* **60**, 70 (2017).
- [122] B. Sanchez-Lengeling and A. Aspuru-Guzik, Inverse molecular design using machine learning: Generative models for matter engineering, *Science* **361**, 360 (2018).
- [123] D. Flam-Shepherd, T. C. Wu, X. Gu, A. Cervera-Lierta, M. Krenn, and A. Aspuru-Guzik, Learning interpretable representations of entanglement in quantum optics experiments using deep generative models, *Nat. Mach. Intel.* **4**, 544 (2022).
- [124] T. Menke, F. Häse, S. Gustavsson, A. J. Kerman, W. D. Oliver, and A. Aspuru-Guzik, Automated design of superconducting circuits and its application to 4-local couplers, *npj Quantum Inf.* **7**, 49 (2021).
- [125] J. Wallnöfer, A. A. Melnikov, W. Dür, and H. J. Briegel, Machine learning for long-distance quantum communication, *PRX Quantum* **1**, 010301 (2020).
- [126] C. P. Williams and A. G. Gray, Automated design of quantum circuits, in *Quantum Computing and Quantum Communications*, edited by C. P. Williams (Springer, Berlin, 1999), pp. 113–125.
- [127] R. B. McDonald and H. G. Katzgraber, Genetic braid optimization: A heuristic approach to compute quasiparticle braids, *Phys. Rev. B* **87**, 054414 (2013).
- [128] U. Las Heras, U. Alvarez-Rodriguez, E. Solano, and M. Sanz, Genetic Algorithms for Digital Quantum Simulations, *Phys. Rev. Lett.* **116**, 230504 (2016).

- [129] Y.-H. Zhang, P.-L. Zheng, Y. Zhang, and D.-L. Deng, Topological Quantum Compiling with Reinforcement Learning, *Phys. Rev. Lett.* **125**, 170501 (2020).
- [130] L. Moro, M. G. A. Paris, M. Restelli, and E. Prati, Quantum compiling by deep reinforcement learning, *Commun. Phys.* **4**, 178 (2021).
- [131] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, Learning the quantum algorithm for state overlap, *New J. Phys.* **20**, 113022 (2018).
- [132] L. Cincio, K. Rudinger, M. Sarovar, and P. J. Coles, Machine learning of noise-resilient quantum circuits, *PRX Quantum* **2**, 010324 (2021).
- [133] J. Yao, P. Kotterring, H. Gundlach, L. Lin, and M. Bukov, Noise-robust end-to-end quantum control using deep autoregressive policy networks, in *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, Proceedings of Machine Learning Research, Vol. 145, edited by J. Bruna, J. Hesthaven, and L. Zdeborova (PMLR, 2022), pp. 1044–1081.
- [134] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [135] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, *Adv. Quantum Tech.* **2**, 1900070 (2019).
- [136] M. Ostaszewski, L. M. Trenkwalder, W. Masarczyk, E. Scerri, and V. Dunjko, Reinforcement learning for optimization of variational quantum circuit architectures, *Advances in Neural Information Processing Systems*, Vol. 34 (Curran Associates, Inc., 2021), p. 18182.
- [137] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Neural predictor based quantum architecture search, *Mach. Learn. Sci. Tech.* **2**, 045027 (2021).
- [138] J. Yao, L. Lin, and M. Bukov, Reinforcement Learning for Many-Body Ground-State Preparation Inspired by Counterdiabatic Driving, *Phys. Rev. X* **11**, 031070 (2021).
- [139] G. Verdon, M. Broughton, J. R. McClean, K. J. Sung, R. Babbush, Z. Jiang, H. Neven, and M. Mohseni, Learning to learn with quantum neural networks via classical neural networks, [arXiv:1907.05415](https://arxiv.org/abs/1907.05415).
- [140] A. Anand, M. Degroote, and A. Aspuru-Guzik, Natural evolutionary strategies for variational quantum computation, *Mach. Learn. Sci. Tech.* **2**, 045012 (2021).
- [141] T. H. Kyaw, T. Menke, S. Sim, A. Anand, N. P. D. Sawaya, W. D. Oliver, G. G. Guerreschi, and A. Aspuru-Guzik, Quantum Computer-Aided Design: Digital Quantum Simulation of Quantum Processors, *Phys. Rev. Appl.* **16**, 044042 (2021).
- [142] J. S. Kottmann, M. Krenn, T. H. Kyaw, S. Alperin-Lea, and A. Aspuru-Guzik, Quantum computer-aided design of quantum optics hardware, *Quantum Sci. Tech.* **6**, 035010 (2021).
- [143] F.-M. Liu, M.-C. Chen, C. Wang, S.-W. Li, Z.-X. Shang, C. Ying, J.-W. Wang, C.-Z. Peng, X. Zhu, C.-Y. Lu *et al.*, Quantum design for advanced qubits, [arXiv:2109.00994](https://arxiv.org/abs/2109.00994).
- [144] R. Duncan, A. Kissinger, S. Perdrix, and J. van de Wetering, Graph-theoretic simplification of quantum circuits with the ZX-calculus, *Quantum* **4**, 279 (2020).
- [145] T. Fösel, M. Y. Niu, F. Marquardt, and L. Li, Quantum circuit optimization with deep reinforcement learning, [arXiv:2103.07585](https://arxiv.org/abs/2103.07585).
- [146] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, *Phys. Rev. A* **52**, R2493(R) (1995).
- [147] G. Torlai and R. G. Melko, Neural decoder for topological codes, *Phys. Rev. Lett.* **119**, 030501 (2017).
- [148] S. Krastanov and L. Jiang, Deep neural network probabilistic decoder for stabilizer codes, *Sci. Rep.* **7**, 11003 (2017).
- [149] S. Varsamopoulos, B. Criger, and K. Bertels, Decoding small surface codes with feedforward neural networks, *Quantum Sci. Tech.* **3**, 015004 (2018).
- [150] P. Baireuther, T. E. O'Brien, B. Tarasinski, and C. W. J. Beenakker, Machine-learning-assisted correction of correlated qubit errors in a topological code, *Quantum* **2**, 48 (2018).
- [151] Y.-H. Liu and D. Poulin, Neural Belief-Propagation Decoders for Quantum Error-Correcting Codes, *Phys. Rev. Lett.* **122**, 200501 (2019).
- [152] R. Sweke, M. S. Kesselring, E. P. L. van Nieuwenburg, and J. Eisert, Reinforcement learning decoders for fault-tolerant quantum computation, *Mach. Learn. Sci. Tech.* **2**, 025005 (2021).
- [153] K. Meinerz, C.-Y. Park, and S. Trebst, Scalable Neural Decoder for Topological Surface Codes, *Phys. Rev. Lett.* **128**, 080505 (2022).
- [154] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, Optimizing quantum error correction codes with reinforcement learning, *Quantum* **3**, 215 (2019).
- [155] Z. Wang, T. Rajabzadeh, N. Lee, and A. H. Safavi-Naeini, Automated discovery of autonomous quantum error correction schemes, *PRX Quantum* **3**, 020302 (2022).
- [156] V. Sivak, A. Eickbusch, B. Royer, S. Singh, I. Tsioutsios, S. Ganjam, A. Miano, B. L. Brock, A. Z. Ding, L. Frunzio *et al.*, Real-time quantum error correction beyond break-even, [arXiv:2211.09116](https://arxiv.org/abs/2211.09116).
- [157] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain, Unsupervised word embeddings capture latent knowledge from materials science literature, *Nature (London)* **571**, 95 (2019).
- [158] S. d'Ascoli, P.-A. Kamienny, G. Lample, and F. Charton, Deep symbolic regression for recurrent sequences, [arXiv:2201.04600](https://arxiv.org/abs/2201.04600).
- [159] P. Veličković, A. P. Badia, D. Budden, R. Pascanu, A. Banino, M. Dashevskiy, R. Hadsell, and C. Blundell, The CLRS algorithmic reasoning benchmark, [arXiv:2205.15659](https://arxiv.org/abs/2205.15659).
- [160] M. Krenn, R. Pollice, S. Y. Guo, M. Aldeghi, A. Cervera-Lierta, P. Friederich, G. d. P. Gomes, F. Häse, A. Jinich, A. Nigam *et al.*, On scientific understanding with artificial intelligence, *Nat. Rev. Phys.* **4**, 761 (2022).
- [161] H. W. de Regt, *Understanding Scientific Understanding* (Oxford University Press, 2017).
- [162] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in 2009

- IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2009), pp. 248–255.
- [163] N. Brown, M. Fiscato, M. H. S. Segler, and A. C. Vaucher, GuacaMol: Benchmarking models for de novo molecular design, *J. Chem. Info. Model.* **59**, 1096 (2019).
- [164] D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov *et al.*, Molecular sets (MOSES): A benchmarking platform for molecular generation models, *Front. Pharmacol.* **11**, 565644 (2020).
- [165] Reinforcement learning for science, <https://www.scigym.net/>.
- [166] A. Turing, Computing machinery and intelligence, *Mind* **59**, 433 (1950).