Using deep learning to understand and mitigate the qubit noise environment

David F. Wise,1,2,* John J. L. Morton,1,2,3 and Siddharth Dhomkar1,†

1London Centre for Nanotechnology, University College London, London, WC1H 0AH, UK
2Quantum Motion Technologies, Nexus, Discovery Way, Leeds, West Yorkshire, LS2 3AA, UK
3Department of Electronic & Electrical Engineering, University College London, London, WC1E 7JE, UK

(Dated: May 5, 2020)

Understanding the spectrum of noise acting on a qubit can yield valuable information about its environment, and crucially underpins the optimization of dynamical decoupling protocols that can mitigate such noise. However, extracting accurate noise spectra from typical time-dynamics measurements on qubits is intractable using standard methods. Here, we propose to address this challenge using deep learning algorithms, leveraging the remarkable progress made in the field of image recognition, natural language processing, and more recently, structured data. We demonstrate a neural network based methodology that allows for extraction of the noise spectrum associated with any qubit surrounded by an arbitrary bath, with significantly greater accuracy than the current state-of-the-art. The technique requires only a two-pulse echo decay curve as input data and can further be extended either for constructing customized optimal dynamical decoupling protocols or for obtaining critical qubit attributes such as its proximity to the sample surface. Our results can be applied to a wide range of qubit platforms, and provide a framework for improving qubit performance with applications not only in quantum computing and nanoscale sensing but also in material characterization techniques such as magnetic resonance.

Robust isolation of a qubit from unwanted noise in its environment is a key factor in technologies such as quantum computers and sensors. There is a long history in magnetic resonance of developing so-called dynamical decoupling protocols, including Carr-Purcell-Meiboom-Gill (CPMG) [1–3], periodic dynamical decoupling (PDD) [4], concatenated dynamical decoupling (CDD) [5], and Uhrig dynamical decoupling (UDD) [6], as general methods to preserve spin coherence in the presence of noise. In practice, the experimental noise spectrum varies significantly between different qubits in ways which are non-trivial to either predict, or indeed to accurately extract from most common measurements [7–9]. As a result, it is difficult to predict a priori which of the several possible dynamical decoupling protocols would provide optimal suppression of decoherence. Indeed, one could envisage constructing a decoupling protocol customized for a particular qubit, but this is impossible without knowing the actual qubit noise spectrum with sufficient accuracy.

Significant advances have been made in machine learning and specifically deep learning techniques, for example in the fields of computer vision and natural language processing, and more recently they have been applied to problems in physics and quantum engineering [10]. Deep feed forward neural networks have been used to enhance extraction of material parameters in scanning probe microscopy [11] and for processing of magnetic resonance spectroscopy data, in NMR, EPR and DEER experiments [12–14]. In addition, deep reinforcement learning techniques have been applied to quantum metrology both as an efficient experiment design heuristic and to increase sensor sensitivity by more than order of magnitude over comparable approaches [15–17].

We propose that the challenges of accurately obtaining qubit noise spectra can be efficiently handled by employing deep learning algorithms. We show how a deep neural network can be trained to extract the noise spectrum from simple and widely-used time-dynamics measurements on qubits, such as the two-pulse ‘Hahn’ echo curves, and compare the accuracy of a deep learning approach with that of standard approximation techniques. Finally, we examine a neural network based technique of processing noisy experimental data and discuss potential uses of an accurate noise spectrum for quantum control. In addition to the possibility of optimising dynamical decoupling to extend qubit coherence, we explore how useful information about the qubit environment such as its proximity to particular noise sources can be deduced from its noise spectrum [18].

NOISE SPECTROSCOPY USING DYNAMICAL DECOUPLING

The noise spectrum of a qubit is an effective proxy to probe its surroundings and provides valuable information for qubit characterization. Furthermore, once the environmental noise spectrum is known, it is possible in principle to run an optimization protocol that minimizes ‘decoherence’ (which can be thought of as proportional to the overlap of the filter function $F(\omega t)$ associated with a given dynamical decoupling pulse sequence and the actual noise spectrum $S(\omega)$) by varying parameters in the sequence such as separation between pulses [7], or their amplitude, phase, and duration. Alternatively, dynamical decoupling protocols can be tailored under real-time experimental feedback [19], however, such methods can be experimentally expensive to perform, limiting their applicability. Therefore, we focus here on using the min-
imum experimental data from a given qubit, such as a single coherence decay curve under a particular pulse sequence, and using purely computational methods to identify the noise spectrum and optimized decoupling sequence.

The time dependence of qubit coherence under applied pulse sequences can be used for extracting useful information about the noise sources in the environment [9, 20–22]. Such sequences can typically be described as a set of \( n \) \( \pi \) pulses, each with duration \( \tau \pi \) applied at time \( t_k \), and possessing a characteristic filter function [23]:

\[
F(\omega t) = \left[1 + (-1)^{n+1}e^{i\omega n\tau} + 2\sum_{k=0}^{n}(-1)^{k}e^{i\omega t_k}\cos\left(\frac{\omega \tau \pi}{2}\right)\right]^2
\]

In principle, the noise spectrum, \( S(\omega) \), can then be extracted from the measured time-dependent coherence decay curve, \( C(t) \), by solving the following integral equation:

\[
\chi(t) = -\ln C(t) = \int_{0}^{\infty} \frac{d\omega}{2\pi} S(\omega) \frac{F(\omega t)}{\omega^2}
\]

where, \( \chi(t) \) is known as the decoherence functional [24]. However, in practice accurately solving such an integral equation is non-trivial, and without a sufficiently faithful noise spectrum, the technique cannot yield an optimized protocol to significantly suppress decoherence.

It is possible to simplify Eq. (2) by assuming the filter function at a given delay time to be a Dirac \( \delta \)-function localized at a desired frequency \( \omega_0 \) [21]. This assumption permits a simple mapping of the coherence decay curve on to a corresponding noise spectrum:

\[
S(\omega_0) = -\pi \frac{\ln C(t)}{t}
\]

Figure 1 shows an application of this simplified approach to obtain noise spectra corresponding to decoherence curves that are simulated from some underlying noise spectrum, using the integral equation in Eq. 2. However, as demonstrated in Fig. 1(c) the delta-function approximation is valid only where the spacing between pulses is much longer than the pulse duration. Increasing the number of \( \pi \) pulses produces a narrower peak in the filter function but with the expense of increased harmonics. As a result, the noise spectra inferred from the decoherence curves using Eq. 3 are a poor fit to the actual spectrum used to generate them.

A second challenge in obtaining accurate noise spectra from coherence decay curves is handling experimental noise in the measurement itself in an unprejudiced way. Depending on the type of qubit, the dominant noise sources can vary significantly, e.g. in the case of flux qubits and Si quantum dots the primary source of noise behaves as \( 1/f^\alpha \) where \( f \) is the frequency and \( \alpha \sim 1 \) [25, 26], telegraphic noise [27] dominates for GaAs quantum dots, bulk nitrogen vacancy (NV) centers in diamond are affected by Lorentzian-type noise [21, 28], whereas near-surface NV centers are prone to double-Lorentzian-type noise [18]. Despite this rich variety of noise sources, decoherence curves are often fitted with a stretched exponential function with essentially two parameters coherence lifetime, \( T_2 \), and power of the stretched exponential, \( \rho \):

\[
C(t) = e^{-t/T_2^\rho}
\]

In practice, qubits may be surrounded by multiple noise sources with varied functional forms, which can only be approximately represented by Eq.(4), and fitting experimental decay curves can remove or obscure valuable information regarding the true noise spectrum.
PROPOSED METHODOLOGY

As illustrated in the discussion above, solving Eq. (2) to extract the noise spectrum specific to the qubit under investigation is non-trivial, however, the reverse operation (evaluating the coherence decay for a given noise spectrum and decoupling sequence) is relatively simple. This asymmetry lends itself to a neural networks-based learning approach to achieve a significant improvement in accuracy, owing to the capacity of neural networks to act as universal function approximators [29, 30]. To successfully implement the deep learning technique we first developed an efficient method for generating a sufficiently large and diverse set of training data. We split this training data into train/validation/test tranches and explored a selection of possible network architectures before selecting the most effective option and training it for optimal performance on the validation set [31]. Finally, we verified the chosen network’s performance on the test set. A concise flow chart of the methodology is shown in Fig. 2(a) with detailed description provided below.

Generation of training data

We simulated a variety of noise spectra assuming commonly applicable models described above and computed the corresponding coherence decay curves using Eq. (2). In the first instance, we generated three different forms of noise spectra for training purposes:

1. A noise spectrum generated from a stretched exponential coherence decay. The delta function approximation shown in Eq. (4) is used to first produce an approximate noise spectrum for a given coherence decay.

2. 1/f noise, with functional form $A/f^\alpha$, generated using simulated parameters chosen to be in line with expected physical constants.

3. A noise spectrum with a Lorentzian form:

   $$\frac{\Delta^2 \tau_c}{\pi} \frac{1}{1 + (\omega \tau_c)^2}$$

   (5)

   where $\Delta$ and $\tau_c$ are coupling strength and correlation time, respectively, again generated using appropriate simulated parameters.

In theory, these data can then be used to train a neural network to output the noise spectrum corresponding to a given coherence decay measured using a Hahn echo sequence. Specifically, network training can be performed using coherence decays as inputs to the network, with noise spectra as the target outputs. Before training, the generated data must be split into training, validation, and test data sets. Training data is actively used to update the network parameters, whilst the network performance on validation data is monitored during training to avoid overfitting (see Fig. 2(d)) [31] and used for tuning the network hyperparameters (its overall structure, loss function etc.). The test data is held out for final evaluation of performance of the network on hitherto unseen data; error and loss rates quoted throughout this paper are always on the test set unless otherwise specified.

Choice of Network Architecture

Many different variants of neural network exist, the simplest being a deep feed forward neural network [31], consisting of multiple layers each with a number of units. The first layer takes as input the training data $X$, with the outputs of each layer fed as input to the next. The final layer output is compared with the training data $y$ to calculate the network loss. Each layer in a feed forward network can be represented by multiplication by a matrix of weights, $W$ followed by addition of a bias vector, $b$ and finally an activation function, $g$. The action of the layer, $h$, on its inputs $x$ can be written:

$$h(x) = g(x^\top W + b)$$

(6)

For each training step, the parameters of the neural network, $\phi$ are updated so that the overall function of the network $f$ is brought closer to the desired function $f^*$. This is done by taking the gradient of the chosen loss function with respect to the network parameters and updating those to reduce the loss, through a process known as back-propagation [31].

One problem with a feed forward network is that all inputs are connected to all units in each layer, which has the effect of destroying local correlations. For our purposes, a network architecture that preserves these local correlations is desirable. There are two obvious choices for working with time ordered data such as a coherence decay curve: convolutional and recurrent (RNN) networks [32, 33]. After exploration the architecture found to be most effective for the task at hand was a Long Short-Term Memory network (LSTM) [34], a special case of the RNN. LSTMs introduce the capacity for long range correlations in input data to be preserved and several recent studies have found the LSTM paradigm to be effective in working with time series data [35–37].

A simple RNN is a sequence of discrete units, each of which processes a single time step of the input data, $x_t$ and produces an output, $h_t$. This output is concatenated with the input of the next layer ($x_{t+1}$) to allow information earlier in the sequence to influence the output of layers later in the sequence. Although useful for maintaining short term correlations, one subtlety of the back
propagation method of training neural networks is that early layers have relatively little impact on the gradient of the loss function of the network as a whole, meaning that early information is effectively ‘forgotten’ by the network. To combat this, LSTMs introduce the concept of a ‘cell state’, $c_t$, to which information can be added (remembered, $i_t$) or subtracted (forgotten, $f_t$), allowing the maintenance of important long term information. A schematic of a single LSTM cell can be seen in Fig. 2(b) and its output is given by the following equations:

\[
\begin{align*}
    f_t &= \sigma_g(W_f(x_t \oplus h_{t-1}) + b_f) \\
    i_t &= \sigma_g(W_i(x_t \oplus h_{t-1}) + b_i) \\
    o_t &= \sigma_g(W_o(x_t \oplus h_{t-1}) + b_o) \\
    c_t &= \tanh(W_c(x_t \oplus h_{t-1}) + b_c) \\
    c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \\
    h_t &= o_t \cdot \tanh(c_t)
\end{align*}
\]

Many descriptions of the mathematical details and intuitions behind RNNs and LSTMs can be found [31, 38, 39]. To produce an output of the correct size, the output from the LSTM is connected to a dense layer matching the size of the $y$ training data (see Fig. 2(c)). Although the activation function for the output layer of a neural network used for regression is typically a linear function, in this case we select an exponential function. This aids the training of the neural network as weights and biases are typically initialized for relatively small output values ($< 5$) whilst the noise spectra used for regression in this case tend to have values $> 10^4$.

**Network Training**

The neural networks used for this study were defined using the Keras module of the Tensorflow 2.0 deep learning library distributed and maintained by Google [40, 41]. We carried out training using the previously discussed train/validation/test split. We use one cycle learning, as proposed by L. Smith [42], to control learning rate during training. Hyperparameter tuning is undertaken to determine optimum network features using Bayesian search and the Weights and Biases library [43].

**RESULTS AND DISCUSSION**

Figure 3 compares the performance of the traditional $\delta$-function approximation for calculating the noise spectrum against our neural network based approach. We show results for synthetic coherence decay curves generated from three different sources of noise: Lorentzian, $1/f$ and stretched exponential. The recorded performance of the neural network is for a single network trained on the three noise sources simultaneously, applied to the three different test data sets. Detailed statistics on the performance of the network are shown in Table I.

Our results show that the neural network approach significantly outperforms the methodology based on a $\delta$-function approximation to the sequence filter function for deducing noise spectra from qubit coherence decay curves. Of particular interest is that a single network is able to successfully reproduce noise spectra of different functional forms. A current limitation of this approach is that the network has only been trained on a set of coherence decays with a decoherence time constant $T_2$s in
FIG. 3. Comparison of two approaches for producing a noise spectrum from a measured coherence decay curve. Stacked histograms of errors in the predicted noise spectra from the holdout test data set are shown using (a) a δ function approximation, and (b) our neural network. Three different forms of noise spectrum are studied, each with ~ 8,500 individual noise spectra and associated decay curves. In (b), the tail of the histogram is clipped for clarity and the maximum error was 2.3%. Also shown are the true (solid curves) and predicted (dashed curves) noise spectra obtained under (c) the δ function approximation and (d) the neural network approach. The noise spectra used are at the 50th percentile error for each approach.

| Noise model      | NN approach Mean | NN approach σ | δ-function approx. Mean | δ-function approx. σ |
|------------------|------------------|---------------|------------------------|----------------------|
| Stretched Exp.   | 0.098            | 0.1           | 17.6                   | 12.2                 |
| f                | 0.10             | 0.1           | 10.2                   | 5.0                  |
| Lorentzian       | 0.39             | 0.21          | 28.4                   | 13.7                 |

TABLE I. Comparison of error statistics in % in noise spectrum extraction using the neural network approach introduced here versus the δ-function approximation for the pulse sequence filter function.

a relatively small range between approximately 120 μs to 600 μs. In principle, this can be circumvented by training a network on the same principles but for different length coherence decays. For example, in Fig. 4(a) and (b) a network is trained on coherence decays with $T_2$ s between 10–140 μs, generated using the phenomenological stretched exponential method discussed above. This neural network approach can therefore be applied generally, given additional training and data generation to cover the parameter range of interest. A more generalised approach would use the time vector data for training, in addition to the signal data, and represents a promising direction for further development.

Next, in Fig. 4(c) and (d) we show the performance of a network trained to reproduce noise spectra from coherence decays generated using the phenomenological method described above but in this case using a filter function associated with a 32 π pulse CPMG sequence. The similarity between the δ-function and the actual filter function of a CPMG sequence with sufficiently large number of π pulses is generally used to justify the approach of employing Eq. (3) to extract noise spectra. Surprisingly, we find the δ-function approach to be less accurate in this case than when a Hahn echo filter function is used as shown in Fig. 4. The neural network appears unaffected by the change in pulse sequence and remains able to successfully reproduce the noise spectra with a low error rate. We speculate that the reason for the poor performance of the δ-function approach in this case is that, whilst the central peak of the filter function becomes narrower at higher pulse numbers, the con-
FIG. 4. Comparison between estimated error in noise spectra predictions for three different neural networks and those obtained using a δ-function approximation. Histograms of error rates for predicting noise spectra generated using Eq. (3) from coherence decays with a coherence decay time constant $T_2$ between 10–140 μs obtained by means of (a) the δ-function approach and (b) a trained neural network. Histograms of error rates for predicting noise spectra generated from coherence decays constructed by implementing dynamical decoupling protocol comprising 32 π-pulses extracted by means of (c) the δ-function approach and (d) a second trained neural network; inset in (c) shows mean percent errors along with the maximum deviation as a function of number of π-pulses for conventional technique. Stacked histograms of error rates for predicting noise spectra having three different functional forms: $1/f$, Lorentzian, and double Lorentzian obtained by means of (e) the δ-function approach and (f) a third trained neural network, the tail of the neural network distribution is clipped for clarity with maximum error in this case 200.1% with high errors caused by noise spectra with large portions very close to 0 where small deviations in predicted noise spectrum magnify mean percentage error.

Having shown the robustness of the neural network approach to a range of coherence decay times and the number of π-pulses in the applied sequence, we now investigate the performance for quantum systems experiencing multiple simultaneous noise sources. Figs. 4(e) and 4(f) show the results of a network that is trained to predict noise spectra arising from three different models: $1/f$, Lorentzian and double Lorentzian (sum of two Lorentzians with different $\omega$, $\Delta$, and $\tau_c$ following Eq. (5)). Double Lorentzian noise spectra are characteristic of quantum systems experiencing noise from two uncorrelated sources, as observed in near-surface spins such as NV centers in diamond [18]. Our results show how a single trained network can successfully predict cases where a coherence decay is a result of a system experiencing multiple noise sources, whilst also differentiating multiple single noise sources. This proof-of-principle demonstration suggests that expanding the range of noise spectra that the network is able to identify, particularly where a noise spectrum is made up of multiple different noise sources, is a fruitful direction for future research.

Handling Experimental Data

The utility of our proposed approach requires a robust method for ingesting experimental coherence curves, including unavoidable experimental measurement noise. A traditional approach would be to apply a least squares fit of a stretched exponential function (given in Eq. 4) to the noisy data to gain a best approximation to the underly-
FIG. 5. Denoising the experimental data using curve fitting or neural networks. Stacked histograms of errors in smoothing synthetic experimental decoherence curves from four classes of noise model are shown using two denoising approaches: a) curve fitting to Eq. (4), or b) the neural network approach. In both cases, insets show mean errors and the maximum deviation. The percent errors are calculated using logarithmic values of decoherence curve. c) A synthetic noisy decoherence curve (from a $1/f$ noise model) along with a denoised curve obtained from the neural network.

| Noise Model | NN approach | $\delta$-function approx. |
|-------------|-------------|---------------------------|
| $1/f$       | Mean        | $\sigma$                  |
| Lorentzian  | 0.41        | 0.46                      |
| Double Lorentzian | 1.72    | 3.63                      |

TABLE II. Error statistics in % for neural network and $\delta$-function noise spectrum approach, for a network trained on $1/f$, Lorentzian and double Lorentzian noise spectra.

We propose instead a function-agnostic approach to removing experimental noise. Once again, we look to LSTM based neural networks to perform this task — an approach that has already been successfully used for denoising of electrocardiogram (ECG) data [37]. We trained the network by applying random noise of $\pm 5\%$ to the coherence curves generated from a) the three different noise models described above, or b) the phenomenological stretched exponential curves. This was used as the input $X$ data, whilst the original noiseless curves were used as the target $y$. To evaluate the success of this neural network technique we compare its ability to reconstruct a given coherence decay against the approach of fitting a stretched exponential to the same noisy data.

To infer the impact of the errors on reconstruction of the noise spectrum, we compare the mean absolute percentage errors in $\log C(t)$ of ground truth versus the noisy data, as the noise spectrum depends explicitly on this as seen in 2 — the results are shown in Fig. 5, with detailed statistics in Table III. Performance for the neural network is comparable to that of fitting for stretched exponential and $1/f$ coherence decays, although it does show some improvement. However, the advantage of the network becomes clear when we examine errors for coherence decays derived from Lorentzian and double Lorentzian noise spectra: whilst the neural network performs similarly on all decay types, the error in curve-fitting increases drastically for these cases. The results demonstrate that a functional-form agnostic approach to denoising — as permitted by the neural network — is essential for an accurate and general extraction of the noise spectrum from experimental data.

**OUTLOOK**

Accurately deducing the noise environment of a quantum system has many potential applications: for qubits, knowledge of the noise spectrum can be used to extend coherence times through bespoke dynamical decoupling sequences [24, 44], while for spin defects in solid state systems, the noise spectrum has been shown to provide information on parameters such as defect depth [18]. Figure 6 illustrates both such applications.

Using an approach based on the $\delta$-function approximation, Romach et al. find evidence for a double Lorentzian form for $S(\omega)$ for shallow NV centers in diamond, at-
tributed to the presence of two distinct noise sources with differing correlation times. The faster correlation time was attributed to surface-modified phonons and the slower to spin-spin coupling between a bath of surface spins. The coupling strength, $\Delta$, to each of these noise sources decreases as defect depth increases. Our technique provides a potential method for accurately estimating defect depth through simple coherence decay measurements, at a much lower experimental cost than previous approaches.

Figures 6(c)-(e) show the application of dynamical decoupling optimization to three different classes of noise spectra. Using the sequential least squares programming (SLSQP) minimization technique [45], provided by the SciPy library [46], we significantly enhance residual coherence over what can be achieved using a 32-pulse CPMG or UDD sequence. Underlying noise spectra are used to synthesise coherence decay curves under 32-pulse CPMG, and to generate bespoke pulse sequences (two examples of which are illustrated) that enhance the coherence at specific points in time. While UDD offers some limited enhancement in coherence over CPMG, substantial increases are seen using the optimized sequences, up to values of $4-8 \times$.

Whilst the results presented here are promising, the potential avenues for improvement are equally clear. The technique currently requires a specific input dimension and time scale, though an encoder-decoder architecture, as seen in sequence-to-sequence models and transformers, provides a possible solution [47, 48]. In addition, whilst the optimization techniques described above appear successful, the application of deep learning techniques to optimising qubit coherence times is a promising route for exploration. Furthermore, recent results have shown the successful application of deep reinforcement learning to quantum sensing problems, suggesting they may well have application in this field [16, 17].

In summary, we have demonstrated a multifaceted
toolbox that uses neural networks to accurately deduce the environmental noise spectrum of a qubit. We show that an LSTM network, properly trained with diverse training data, is capable of predicting the precise noise spectrum from a coherence decay curve recorded using dynamical decoupling protocols comprising variable numbers of $\pi$ pulses. To treat experimental data with measurement noise, we again employ an LSTM network to perform effective denoising that preserves the original functional form of the decoherence curve. This unprejudiced way of smoothing noisy experimental data avoids the use of a predetermined fitting function, enabling more accurate reconstruction of the qubit environment. Using these deep learning techniques one can even extract information about multiple noise sources, which can then be used to infer salient properties of the qubit under investigation. Finally, we show how the extracted noise spectrum can be used to generate a customized dynamical decoupling sequence that enhances coherence time significantly beyond what can be achieved with standard protocols. We have surveyed a variety of possible avenues related to the application of deep learning for problems related to qubit decoherence and have established that approach has several potential applications in different quantum technologies. Despite this promise, these concepts remain in their infancy and there is much scope for further improvement and development.

ACKNOWLEDGEMENTS

We thank Prasanna Vaidya, Gyde, India, for the fruitful discussions on neural networks. This research has received funding from the Engineering and Physical Sciences Research Council (EPSRC) via the Centre for Doctoral Training in Delivering Quantum Technologies (EP/L015242/1), as well as from the European Research Council (ERC) via the LOQO-MOTIONS grant (H2020-EU.1.1.), Grant No. 771493.

* d.wise.15@ucl.ac.uk
† s.dhomkar@ucl.ac.uk

[1] E. Hahn, Phys. Rev. 80, 580 (1950).
[2] H. Y. Carr and E. M. Purcell, Phys. Rev. 94, 630 (1954).
[3] S. Meiboom and D. Gill, Rev. Sci. 29, 688 (1958).
[4] L. Viola and S. Lloyd, Phys. Rev. A 58, 2733 (1998).
[5] K. Khodjasteh and D. A. Lidar, Phys. Rev. Lett. 95, 180501 (2005).
[6] G. S. Uhrig, Phys. Rev. Lett. 98, 100504 (2006), arXiv:0609203 [quant-ph].
[7] F. Poggiali, P. Cappellaro, and N. Fabbri, Phys. Rev. X 8, 21059 (2018), arXiv:1712.08256.
[8] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, Nature 458, 996 (2009), arXiv:0812.5095.
[9] G. A. Álvarez and D. Suter, Phys. Rev. Lett. 107, 230501 (2011).
[10] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schulz, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Rev. Mod. Phys. 91, 45002 (2019), arXiv:1903.10563.
[11] N. Borodinov, S. Neumayer, S. V. Kalinin, O. S. Ovchinnikova, R. K. Vasudevan, and S. Jesse, npj Comput. Mater. 5, 1 (2019).
[12] X. Qu, Y. Huang, H. Lu, T. Qiu, D. Guo, T. Agback, V. Orekhov, and Z. Chen, Angew. Chem. Int. Ed. 59, 1 (2020), arXiv:1904.05168.
[13] A. T. Taguchi, E. D. Evans, S. A. Dikanov, and R. G. Griffith, J. Phys. Chem. L 10, 1115 (2019).
[14] S. G. Worswick, J. A. Spencer, G. Jeschke, and I. Kuprov, Sci. Adv. 4, 1 (2018).
[15] X. M. Zhang, Z. Wei, R. Asad, X. C. Yang, and X. Wang, npj Quantum Inf. 5 (2019), arXiv:1902.02157.
[16] L. J. Fiderer, J. Schuff, and D. Braun, arXiv:2003.02183.
[17] J. Schuff, L. J. Fiderer, and D. Braun, New J. Phys. 22, 035001 (2020), arXiv:1908.08416.
[18] Y. Romach, C. Müller, T. Unden, L. J. Rogers, T. Isola, K. M. Itoh, M. Markham, A. Stacey, J. Meijer, S. Pezzagna, B. Naydenov, L. P. McGuinness, N. Bar-Gill, and F. Jelezko, Phys. Rev. Lett. 114, 1 (2015), arXiv:1404.3879.
[19] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, Nature 458, 996 (2009), arXiv:0812.5095.
[20] T. Yuge, S. Sasaki, and Y. Hirayama, Phys. Rev. Lett. 107, 170504 (2011), arXiv:1105.1594.
[21] N. Bar-Gill, L. M. Pham, C. Belthangady, D. Le Sage, P. Cappellaro, J. R. Maze, M. D. Lukin, A. Yacoby, and R. Walsworth, Nat. Commun. 3, 858 (2012), arXiv:1112.0667v2.
[22] S. Hernández-Gómez, F. Poggiali, P. Cappellaro, and N. Fabbri, Phys. Rev. B 98, 214307 (2018).
[23] H. Uys, M. J. Biercuk, and J. J. Bollinger, Phys. Rev. Lett. 103, 2 (2009), arXiv:0904.0036.
[24] L. Cywiński, R. M. Lutchyn, C. P. Nave, and S. Das Sarma, Phys. Rev. B. 77, 1 (2008), arXiv:0712.2225.
[25] J. Bylander, S. Gustavsson, F. Yan, F. Yoshihara, K. Harrabi, G. Fitch, D. G. Cory, Y. Nakamura, J. S. Tsai, and W. D. Oliver, Nat. Phys. 7, 565 (2011).
[26] J. Yoneda, K. Takeda, T. Otsuka, T. Nakajima, M. R. Delbecq, G. Allison, T. Honda, T. Kodera, S. Oda, Y. Hoshi, N. Usami, K. M. Itoh, and S. Tarucha, Nat. Nanotechnol. 13, 102 (2018).
[27] J. Medford, L. Cywiński, C. Barthel, C. M. Marcus, M. P. Hanson, and A. C. Gossard, Phys. Rev. Lett. 108, 086802 (2012).
[28] J. R. Klauder and P. W. Anderson, Phys. Rev. 125, 912 (1962).
[29] K. Hornik, Neural Networks 6, 1069 (1993).
[30] G. B. Huang, L. Chen, and C.-k. Siew, IEEE Trans. Neural. Netw. Learn. Syst. 17, 879 (2006).
[31] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning (MIT Press, 2016) http://www.deeplearningbook.org.
[32] K. Fukushima, Biological Cybernetics 36, 193 (1980).
[33] B. A. Pearlmutter, IJCNN Int Jt Conf Neural Network, 365 (1989).
[34] S. Hochreiter and J. Schmidhuber, Neural Comput. 9, 1735 (1997).
[35] C. T. C. Arsene, R. Hankins, and H. Yin, in 2019 27th European Signal Processing Conference (EUSIPCO) (2019) pp. 1–5.
[36] J. Chen, G. Q. Zeng, W. Zhou, W. Du, and K. D. Lu, Energy Convers. Manag. 165, 681 (2018).
[37] K. Antczak, arXiv:1807.11551.
[38] C. Olah, “Understanding LSTM Networks,” (2015).
[39] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Muller, Lecture Notes in Computer Science 11700, 435 (2019), arXiv:1606.06565.
[40] F. Chollet and Others, “Keras,” (2015).
[41] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” (2016), arXiv:1603.04467.
[42] L. N. Smith, arXiv:1803.09820.
[43] L. Biewald, “Experiment Tracking with Weights and Biases,” (2020).
[44] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, Nature 458, 996 (2009), arXiv:0812.5095.
[45] D. Kraft, ACM Trans. Math. Softw. 20, 262 (1994).
[46] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, b. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pe-dregosa, P. van Mulbregt, and S. . . Contributors, Nat. Methods 17, 261 (2020).
[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, in Advances in Neural Information Processing Systems 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017) pp. 5998–6008.
[48] N. Wu, B. Green, X. Ben, and S. O’Banion, arXiv:2001.08317.