Machine Learning: Science and Technology

Deep quantum graph dreaming: deciphering neural network insights into quantum experiments

IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 015029 https://doi.org/10.1088/2632-2153/ad2628

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
4 October 2023

REVISED
20 December 2023

ACCEPTED FOR PUBLICATION
31 January 2024

PUBLISHED
15 February 2024

PAPER

Deep quantum graph dreaming: deciphering neural network insights into quantum experiments

Tareq Jaouni[1,2,*], Sören Arlt[1], Carlos Ruiz-Gonzalez[1], Ebrahim Karimi[1,2], Xuemei Gu[1] and Mario Krenn[1,*]

[1] Max Planck Institute for the Science of Light, Erlangen, Germany
[2] Nexus for Quantum Technologies, University of Ottawa, K1N 6N5, ON, Ottawa, Canada
[*] Authors to whom any correspondence should be addressed.

E-mail: tjaou104@uottawa.ca and mario.krenn@mpl.mpg.de

Keywords: neural network interpretability, deep dreaming, quantum physics

Abstract

Despite their promise to facilitate new scientific discoveries, the opaqueness of neural networks presents a challenge in interpreting the logic behind their findings. Here, we use a eXplainable-AI technique called *inception* or *deep dreaming*, which has been invented in machine learning for computer vision. We use this technique to explore what neural networks learn about quantum optics experiments. Our story begins by training deep neural networks on the properties of quantum systems. Once trained, we 'invert' the neural network—effectively asking how it imagines a quantum system with a specific property, and how it would continuously modify the quantum system to change a property. We find that the network can shift the initial distribution of properties of the quantum system, and we can conceptualize the learned strategies of the neural network. Interestingly, we find that, in the first layers, the neural network identifies simple properties, while in the deeper ones, it can identify complex quantum structures and even quantum entanglement. This is in reminiscence of long-understood properties known in computer vision, which we now identify in a complex natural science task. Our approach could be useful in a more interpretable way to develop new advanced AI-based scientific discovery techniques in quantum physics.

1. Introduction

Neural networks have been demonstrably promising towards solving various tasks in quantum science [1–3]. One notorious frustration concerning neural networks, however, lays in their inscrutability: modern architectures often contain millions of trainable parameters, and it is not readily apparent what role that they each play in the network's prediction. We may, therefore, inquire about what learned concepts from the data that the network utilizes to formulate its prediction, an important prerequisite in achieving scientific understanding [4]. This has since motivated the development of eXplainable-AI (XAI), which interprets how the network comes up with its solutions [5–8]. These developments have spurred physicists to address the problem of interpretability, resulting in the rediscovery of long-standing physics concepts [9, 10], the identification of phase transitions in quantum many-body physics [11–14], the compression of many-body quantum systems [15], and the study on the relationship between quantum systems and their entanglement properties [16, 17].

Here, we apply neural networks in the design of quantum optical experiments. The growing complexity of quantum information tasks has since motivated the design of computational methods capable of navigating the vast combintorical space of possible experimental designs that involve unintuitive phenomena [18]. To this end, scientists have developed automated design and machine learning routines [19], including some that leverage genetic algorithms [20, 21], active learning approaches [22] and the optimization of parameterized quantum circuits [23–25]. One may inquire if we may be able to learn new physics from the discoveries made by such algorithms. For instance, the computer algorithm Melvin [19], which topologically searches for arrangements of optical elements, has led to the discovery of new concepts

such as the generation of entanglement by path identity [26] and the creation of multipartite quantum gates [27]. However, the interpretability of these solutions is obfuscated by the stochasticity of the processes that create them as well as the unintuitiveness of their representations. The recent invention of THESEUS [24], and its successor PYTHEUS [25] addresses this through the topological optimization of highly interpretable, graph-based representation of quantum optical experiments. This has already enabled new scientific discoveries, such as a new form of multi-photon interference [28], and novel experimental schemes for high-dimensional quantum measurement [29].

To this point, the extraction and generalization of new concepts has largely been confined to analyzing the optimal solutions discovered by these algorithms. However, we may inquire if we can learn more physics by probing the rationale behind the computer's discoveries. Little attention has hitherto been given towards the application of XAI techniques on neural networks trained on quantum experiments, which may allow us to conceptualize what our algorithm has learned. In so doing, we may guide the creation of AI-based design techniques for quantum experiments that are more reliable and interpretable.

In this work, we present an interpretability tool based on the inceptionism technique in computer vision, better known as deep dreaming [30]. This technique has been applied to iteratively guide the automated design of quantum circuits [31] and molecules [32] towards optimizing a target property; it has also been applied in [33] to verify the reliability of a network trained to classify the entanglement spectra of many-body quantum systems. More importantly, it also lets us visualize what physical insights has the neural network gained from the training data. This lets us better discern the strategies applied throughout automated design processes, as well as to verify physical concepts rediscovered by the network, such as the thermodynamic arrow of time [34].

Here, we adapt this approach to quantum graphs. We train a deep neural network to predict properties of quantum systems, then inverse the training to optimize for a target property. We observe that the inverse training dramatically shifts the initial distribution of properties. We also show that, by visualizing the evolution of quantum graphs during inverse training, we are able to conceptualize the learned strategies applied by the neural network. We probe the network's rationale further by inverse training on the intermediate layers of the network. We find that the network learns to recognize simple features in the first layers and then builds up more complicated structures in later layers. Altogether, we synthesize a complete picture of what the trained neural network sees. We, therefore, posit that our tool may aid the design of more interpretable and reliable computer-inspired schemes to design quantum optics experiments.

2. Methodology

2.1. Graphs and quantum experiments

As developed in [24, 25, 35–37], we may represent quantum optical experiments in terms of colored, weighted, undirected multigraphs. This representation can be extended to integrated photonics [38–41] and entanglement by path identity [26, 42, 43]. The vertices of the graph represent photon paths to detectors, whereas edges between any two vertices, a and b, indicate correlation between two photon paths. We may assign an amplitude to them by introducing edge weights $\omega_{a,b}$, and we may assign the photons' internal mode number through different edge colorings. Each vertex inherits a color from the colored edge, defining the state of each photon.

Here, we consider graph representations of four-qubit, two-dimensional experiments dealing with state creation. Specifically, we consider graphs with vertices $V = \{0, 1, 2, 3\}$ and mode numbers 0 and 1 which we represent by coloring the edges blue and red respectively. Each graph, therefore, consists of 24 possible edges with real-valued edge weights between 1 and -1. We may determine the particular quantum state $|\Phi(\omega)\rangle$, where $\Phi(\omega)$ is the graph's weight function defined according to equation (2) in [25]. Specifically, we write

$$\Phi(\omega) = \sum_m \frac{1}{m!} \left(\sum_{e \in E(G)} \omega(e)\, x^\dagger(e)\, y^\dagger(e) + \text{h.c.} \right)^m , \tag{1}$$

where $E(G)$ is the set of edges of the graph G, $x^\dagger(e)$ and $y^\dagger(e)$ are creation operators of photons, which are represented as vertices x and y of edge e, and $h.c.$ is the hermitian conjugate, which includes annihilation terms. The quantum state can then be realized physically by applying the weight function to the vacuum state as $|\Phi(\omega)\rangle = \Phi(\omega)|\text{vacuum}\rangle$. On the whole, the neural network finds a way to decompose the quantum state into PMs of a graph. This is useful because we can experimentally implement arbitrary graphs in the laboratory, and the quantum state emerges as the coherent superposition of PMs.

We condition the creation of each term in the state on subsets of edges which contains every vertex in the graph exactly once, otherwise known as the PMs of the graph. This appears in $\Phi(\omega)$ as $m = 2$ order terms

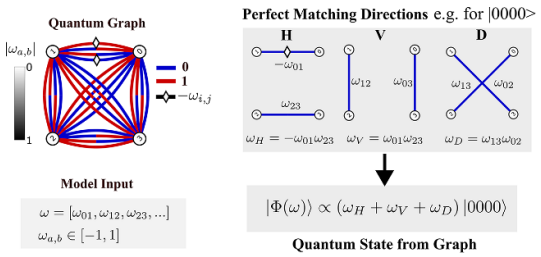

Figure 1. Brief overview of quantum graphs. In this work, we consider *complete* graph representations of two-dimensional, quadripartite quantum graphs. We let $\omega_{a,b}$ denote the weight of the edge connecting vertex a to vertex b. The weight's magnitude is indicated by the transparency of the edge and the presence of a diamond signifies a negative edge weight. The creation of every possible ket in the joint Hilbert space $\mathcal{H}$ is conditioned on the coherent superposition of the three possible types of perfect matchings (PMs), which are distinguished in terms of their direction. By considering the superposition of every PM, we obtain the quantum graph's state $|\Phi(\omega)\rangle$.

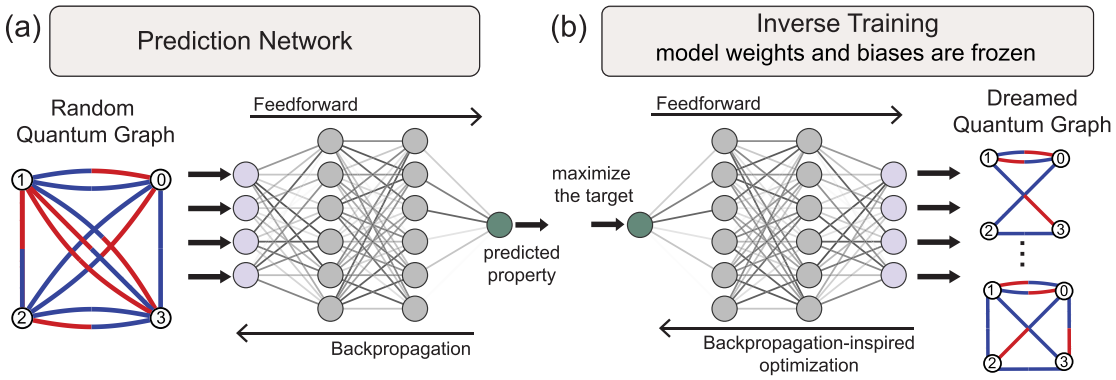

Figure 2. Quantum graph deep dreaming. (a) The weights and biases of a feed-forward neural network are continually updated during training to predict a property such as fidelity of a given input random quantum experiment represented by a graph. (b) In the deep dreaming process, the weights and biases of the network are frozen, and an initial input graph is fed onto the input layer of the neural network. The weights of an initial input graph are updated iteratively to maximize the activation of the output neuron of the feed-forward network, which gives the network's prediction on the aforementioned property. We expect that, when we feed the dreamed graph at the end of the process onto the neural network, the output activation is maximized.

and, physically, corresponds to conditioning one photon on each detector, a common technique in quantum optics. For each term, we can define three possible PMs, each distinguished by their 'directionality', which we show in figure 1. We obtain the amplitude of the term through the sum of weights of the three PMs, which are themselves determined by the product of edge weights. We permit multiple edges between the vertices to allow for the the superposition of different states. Applying this procedure for every possible ket in the joint Hilbert space $\mathcal{H} = \mathbb{H}_2 \otimes \mathbb{H}_2 \otimes \mathbb{H}_2 \otimes \mathbb{H}_2$, we may obtain the state $|\Phi(\omega)\rangle$. Larger, D-dimensional quantum systems consisting of n photons can be represented as a weighted graph with up to $D^2 \times \frac{n(n-1)}{2}$ edges.

2.2. Training

Figure 2 illustrates the basic workflow behind the dreaming process. A feed-forward neural network is first trained on the edge weights ω of a complete, quadripartite, two-dimensional quantum graph in order to make predictions on certain properties of the corresponding quantum state $|\Phi(\omega)\rangle$. We randomly initialize ω over a uniform distribution $[-1, 1]$. The neural network's own weights and biases are optimized for this task via mini-batch gradient descent and the mean squared error (MSE) loss function.

We consider the state fidelity $|\langle\Phi(\omega)||\psi\rangle|^2$ with respect to two well-known classes of multipartite entangled states within the joint Hilbert space $\mathcal{H}$. First, the Greenberger–Horne–Zeillinger (GHZ) State [44], $|\psi\rangle = |\text{GHZ}\rangle$, where

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}\left(|0000\rangle + |1111\rangle\right), \tag{2}$$

and, second, the W-state [45], $|\psi\rangle = |W\rangle$, where

$$|W\rangle = \frac{1}{\sqrt{2}}\left(|1000\rangle + |0100\rangle + |0010\rangle + |0001\rangle\right). \tag{3}$$

In addition, we also consider a measure of quantum state entanglement resulting from a graph—the concurrence [46]. Let A_1, A_2, A_3, A_4 each denote the subsystems of the joint quadripartite Hilbert space to which $|\Phi(\omega)\rangle$ is defined. Then assuming the pure state $\rho = |\Phi(\omega)\rangle\langle\Phi(\omega)|$, we may write

$$C(\rho) = \sum_{\mathcal{M}} C_{\mathcal{M}}(\rho) = \sum_{\mathcal{M}} \sqrt{2\left(1 - \mathrm{tr}\left(\rho_{\mathcal{M}}^2\right)\right)} \tag{4}$$

where $C(\rho)$ is the concurrence, $\mathcal{M}$ refers to a bipartitioning or 'split' of the subsystems into two disjoint subsystems (for example, $|0000\rangle = |0\rangle \otimes |000\rangle$ refers to a bipartitioning of subsystems into sets $\{A_1\}$ and $\{A_2, A_3, A_4\}$) and $\mathrm{tr}(\rho_{\mathcal{M}}^2)$ is the reduced density matrix obtained by tracing out $\mathcal{M}$. Each term in the sum of equation (4), then, corresponds to a different bipartitioning of subsystems. In this work, we train our networks to make predictions on the mean of $\mathrm{tr}(\rho_{\mathcal{M}}^2)$ across all biparitions $\mathcal{M}$, $\overline{\mathrm{tr}(\rho_{\mathcal{M}}^2)}$. Furthermore, for all cases considered, the network is trained on examples with a property value below a threshold of 0.5 to ensure that the network is not memorizing the best solutions in each case. This threshold remains fixed for cases involving the GHZ- and W- state fidelities and the value of $\overline{\mathrm{tr}(\rho_{\mathcal{M}}^2)}$. During this generation of the training data set, if examples are beyond the threshold, they are rejected.

Once convergence in the training has been achieved, we then execute the deep dreaming protocol to extract insights on what the neural network has learned. Given an arbitrary input graph, we select a neuron in the trained neural network. Then, we maximize the neuron's activation by updating the input graph via gradient ascent. In this stage, the weights and biases of the neural network are frozen, and we instead optimize for the edge weights of the input graph. During each iteration of the process, we calculate the loss—here, the negative of the network's activation—by evaluating the network's prediction with the intermediate, input graph. At the end of the process, the graph mutates into a configureation which most excites the neuron. However, this may not entirely represent all that the neuron over-interprets from the input graph, as it has been shown in [47] that individual neurons can be trained to recognize various possible features of the input. Therefore, to uncover all that the neuron sees, we repeat this procedure multiple times with different input quantum graphs.

3. Results

3.1. Dreaming on the output layer

Towards attaining a general idea of what the neural network has learned about select properties for the quantum state $|\Phi(\omega)\rangle$, we first apply the deep dreaming approach on the output layer. Figure 3(a) illustrates the mutation of an input graph by applying the deep dreaming approach on a $[400^3,10]$ (three hidden layers of 400 neurons, one hidden layer of 10 neurons) neural network, which has been trained to predict either the GHZ-state or the W-state fidelity. We also apply this approach on a $[800^7]$ neural network architecture, which has been trained to predict the mean value of $\mathrm{Tr}(\rho_{\mathcal{M}}^2)$. While dreaming, we task our network to find configureations which maximizes the property value. It should be stressed that, in particular, the optimal configureation that maximizes $\overline{\mathrm{tr}(\rho_{\mathcal{M}}^2)}$ minimizes the concurrence; we, therefore, anticipate the dreamed graph to correspond to a maximally separable state. Conversely, tasking the network to minimize $\overline{\mathrm{tr}(\rho_{\mathcal{M}}^2)}$ will influence it to dream graphs that realize maximally entangled states.

We obtain $|\Phi(\omega)\rangle$ from the reconstructed, mutated graph and recompute its true property value in each step. In all cases, we find that the graph evolves steadily towards the maximum property value. We repeat this procedure for 1000 different quantum graphs and plot the distribution of each graphs' initial versus dreamed fidelities in figure 3(b). In all three cases, we observe that the network consistently finds distinct examples with a property value outside the initial distribution's upper bounds. This demonstrates our approach's potential to discover novel quantum graphs which optimizes a specific quantum state property. The distribution of dreamed values of $\overline{\mathrm{tr}(\rho_{\mathcal{M}}^2)}$ is much more narrow than the dream distribution of either the GHZ- or W- state fidelities. This is due to the fact that a wide variety of states demonstrates separability across various $\mathcal{M}$ and, therefore, possess a high value of $\overline{\mathrm{tr}(\rho_{\mathcal{M}}^2)}$, whereas the only states which maximizes the GHZ/W state fidelities are the GHZ/W states themselves. Furthermore, the activation vastly exceeds the usual upper bounds of the property being predicted. This is due to the fact that, as the input graph mutates, the edge weights increase without bound and become unnormalized. Thus, in the end, we compute the normalization of all quantum states.

The intermediate steps of the dreaming process allow us to discern what strategies the neural networks are applying to a given optimization task. In figure 4, we summarize the evolution of different initial graphs during inverse training for different targets. In figure 4(a), we observe that the neural network tries to activate the $|0000\rangle$ and $|1111\rangle$ states either by creating PM of these terms in unused directions—the input graph had no PM in that direction previously—or by completing them with the assistance of an existing PM

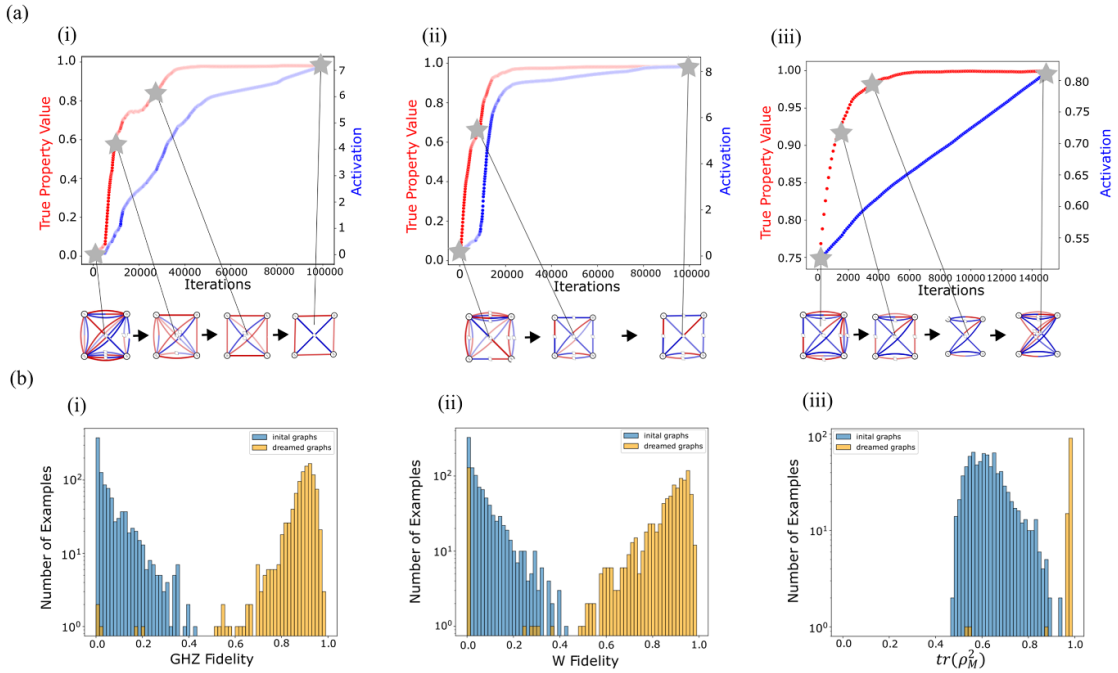

Figure 3. Dreaming results for on the output layer of different neural network architectures. (a) Plot of the evolution of an arbitrary input graph's activation, shown in blue, and true property value, shown in red, when we task the network to discover quantum graphs satisfying different optimization tasks. Specifically, we task the $[400^3, 10]$ network to find quantum graphs maximizing (i) the GHZ- state and (ii) W-state fidelities, and task the $[800^7]$ network to find graphs maximizing (iii) $\mathrm{tr}(\overline{\rho^2_{\mathcal{M}}})$. For each case, we show the intermediate steps of the input graphs' evolution to its dreamed counterpart and only show edges whose weights are above a threshold of 0.4. We, therefore, illustrate steps of the quantum graphs' evolution throughout each dreaming process. These intermediate steps reveal that, in inverse-training, edges of PMs which do not positively contribute to the target property are mitigated. (b) Distribution of initial vs. dreamed fidelities with respect to (i) the GHZ state and (ii) the W-state, as well as (iii) the mean value of $\mathrm{tr}(\rho^2_{\mathcal{M}})$. We observe that most dreamed examples exceed the upper bound of the original dataset, attesting to our tool's ability to find quantum graphs that are novel to the original dataset.

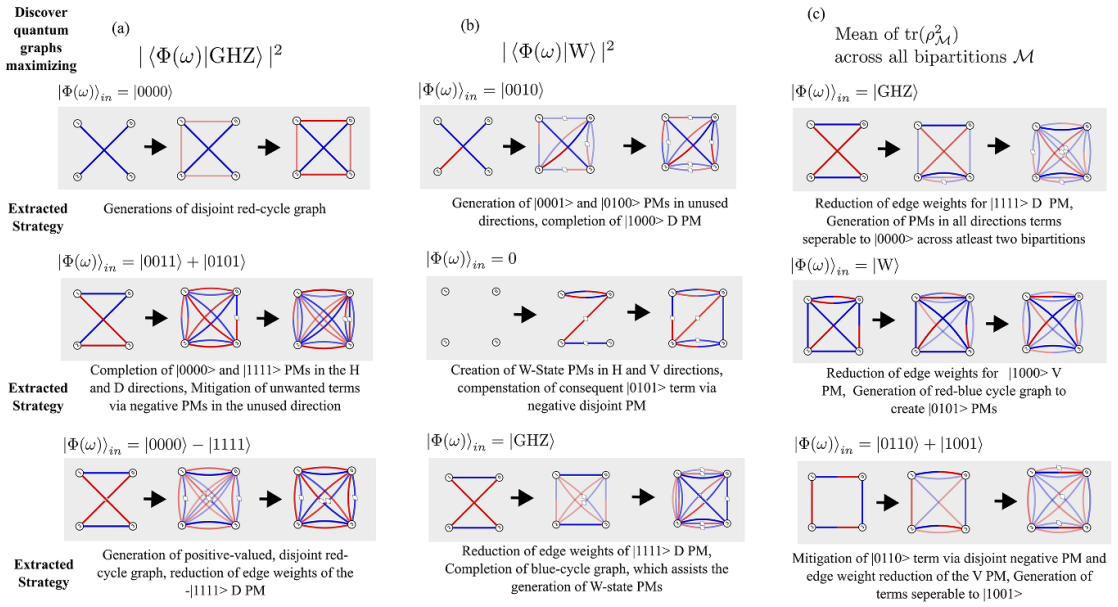

Figure 4. Extracted strategies from the evolution of certain states when dreaming on the output layer of the neural network. We discern the strategies employed by the inverse training routine when applied to a network tasked to optimize (a) the GHZ-State Fidelity, (b) the W-State Fidelity, and (c) the mean value of $\mathrm{tr}(\rho^2_M)$ by considering several initialisations for each case. For each graph, we only show edges with weights greater than 0.3. We find that the network attempts to construct perfect matchings (PMs) of terms which positively contribute to the property value and whose weights add up to 1. Conversely, we find that the network eliminates unwanted terms by either directly reducing the edge weights of the PM corresponding to that term, or by introducing negative, disjoint perfect matchings of that term. For (c), we observe that the network 'selects' a term in the initial state to be minimized, then creates terms that are separable across two or more bipartitions with respect to the remaining states.

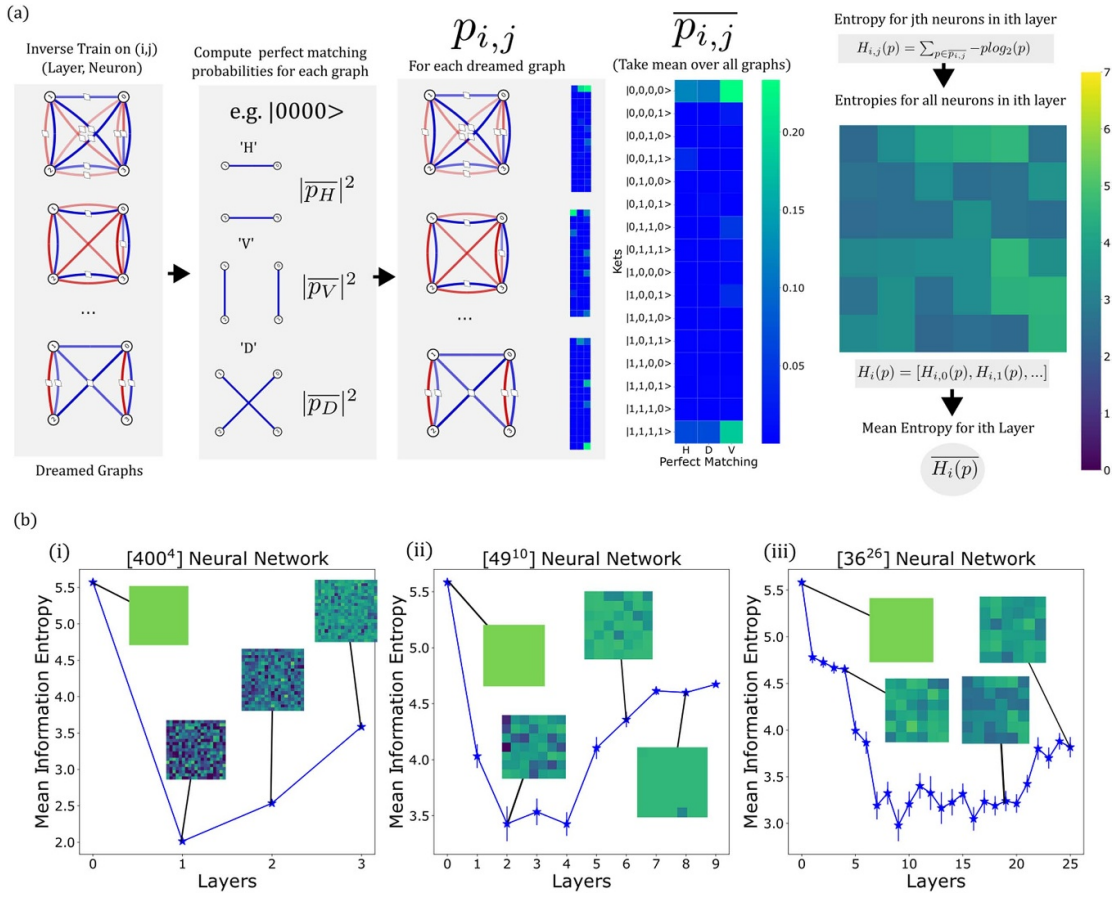

Figure 5. Information entropy throughout each different neural network architecture. (a) Workflow behind computing the mean information entropy for each layer of the trained neural network. We dream with multiple input graphs on each neuron in the neural network. To account for the diversity of structures that a neuron is interested in seeing, we compute the mean probability amplitudes for every possible perfect matching corresponding to each ket. We thereby observe the overall graph, which the neuron sees best. We may then compute the information entropy of each neuron, $H_{i,j}(p)$, and the mean information entropy of the layer, $\overline{H_{i,j}(p)}$. This gives us a measure of the complexity of structures seen by the neural network. As conveyed in the different $p_{i,j}$ for each dreamed graph, we note the variety of structures which the network over-interprets; this illustrates the multifactedness of the neurons. (b) Mean information entropy plots for the (i) $[400^4]$ (ii) $[49^{10}]$ and (iii) $[36^{26}]$ neural network architectures. A general trend that we may discern in all three cases is that the mean information entropy converges to a minimum in the lower layers and then gradually increases as we go deeper. We may attribute this to the intuition that the network initially learns to recognize simpler structures, then learns increasingly complicated ones as we go deeper within the network.

in some direction, as is seen in particular with the $|\Phi(\omega)\rangle = |0011\rangle + |0101\rangle$ initialization. The dreaming process creates these PMs such that their weights add up to 1. In circumstances where the initial graph starts with unwanted terms, or when the network unavoidably creates these terms while dreaming, the network attempts to eliminate them either by directly lowering the edge weights' magnitudes or by introducing negative weight PMs in different directions. We see this trend continue when the network is tasked with maximizing the W-state fidelity, as shown in figure 4(b), albeit instead in favouring the activation of the $|1000\rangle, |0100\rangle, |0010\rangle$, and $|0001\rangle$ states. In figure 4(c), the network attempts to maximize the separability of the initial, maximally entangled state by first eliminating one term from the initial state via edge-weight minimization or through negative PMs, then creating PMs of additional terms which are separable with respect to the intermediate graph state across two or more bipartitions. Through our deep dreaming approach, we have shown that the network learns about creating states through the graph representation in order to consistently achieve optimal values for select properties of the quantum state. We remark that, for each state property, the network was able to ascertain the configureations which maximizes them while only seeing configureations having property values below 0.50. This strongly suggests that the network is achieving its tasks from physical insights, rather than by memorizing the best examples.

3.2. Interpretability of neural network structure

We apply the deep dreaming approach on the neurons of its hidden layers to gain insight into the neural network's internal model, which generalizes well beyond the training data. We summarize the insights that we extract through our routine in figure 5. To showcase the universality of our approach, we consider several

different neural network architectures—the $[400^4]$, $[49^{10}]$ and $[36^{26}]$ networks—that have each been trained to predict the GHZ-state fidelity. For each network, we dream on the ith neuron in the jth hidden layer with 20 input graphs to best capture all of the possible structures exciting the neuron.

We take particular interest in how the complexity of the dreamed graphs evolves with the network depth. We obtain the greatest amount of information about our quantum graphs by considering all of the different ways, as seen through the graphs' PMs, that a ket is realized. We, therefore, attribute to each dreamed graph a 3×16 array, $p_{i,j}$, consisting of the probabilities of all possible PMs; through this, we gain insight into the state created by the graph, as well as all PM directions being used to that purpose. As we go deeper into the neural network, we observe that the dreamed graphs activate a greater number of PM directions and kets, which reflects the increasing complexity of structures the neural network has learned to recognize. We also verify the multifaceted nature of the neurons: different input graphs are observed to result in dreamed graphs that recreate different input states. As we see in the third inset of figure 5(a), the neuron may over-interpret parts of the graph that best creates the $|0000\rangle$ term, or it may either over-interpret different possible PM directions for $|0000\rangle$, or parts of the graph which instead realize the $|1111\rangle$ term.

We may quantify the complexity of structures recognized throughout the network with the information entropy $H_{i,j}$. We take the mean value of $p_{i,j}$ across all of the dreamed graphs, then use it to compute $H_{i,j}$ through the procedure outlined in appendix C. Repeating this procedure across all hidden layer neurons, we may then determine the average entropy observed across the jth layer, which gives us a general metric of the complexity of structures being recognized. We plot the trend of $\overline{H_{i,j}}$ observed across all three neural network architectures in figure 5(b). Intuitively, we expect that a deep neural network first learns to recognize simple structures, then more abstract features with network depth. Indeed, we observe consistently that, from an initial peak, the information entropy drops to its lowest values at the earlier layers, before gradually increasing near the end of the neural network. This certifies the universal assertion that the network identifies simple features of the input graph, such as edges that form one or two PMs to states, before forming more complicated graphical structures in the deeper layers that features a greater set of PMs.

4. Outlook

In this article, we showcase preliminary results for adapting the deep dreaming approach to quantum optical graphs for deep neural networks on different target quantities. We apply our routine to ascertain the strategies employed by the neural network on its predictive task by dreaming on the output layer and throughout the network. Crucially, we demonstrate that the trained neural network builds a non-trivial model of the quantum state properties produced by a quantum experiment, and we find that the deep dreaming approach does remarkably well in finding novel examples outside of the initial dataset. Lastly, in applying our approach to the hidden layers of the neural network, we find that the network gradually learns to recognize increasingly complicated structures, and that the individual neurons are multifaceted in the possible structures that excites them. In future work, further transparency of the learned representations can be possibly attained by applying regularization techniques such as α-norm [48], jitter [30], or by dreaming on the mean of a set of input graphs [47] to converge towards more interpretable solutions. It will also be interesting to see if we can obtain further insights by directly modifying the weights and biases of the neural network. Above all, we may also apply these tools to larger graphs with more dimensions and explore different applications beyond state creation, such as Quantum Measurements and Quantum Communication.

Thanks to their relative simplicity, the quadripartite graphs have been a good testing case for our inception approach, and the knowledge we extract from them can be used in other systems. Larger graphs and new targets will provide a novel and deeper understanding of quantum optics experiments as well as inspire new research. We foresee that our approach can be used to extend frameworks for automated setup design [4, 19, 25] as well as in generative molecular algorithms [32, 49] which adapt a surrogate neural network model. Through our approach, we can better decipher what these frameworks have learned about the underlying science, and understand the intermediate strategies taken towards a target configureation.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/artificial-scientist-lab/deepGraphDreaming.

Acknowledgments

T J and E K acknowledge the support of the Canada Research Chairs (CRC) and Max Planck-University of Ottawa Centre for Extreme and Quantum Photonics.

Appendix A

A.1. Training details

We generate 20 million input-output pairs using the digital discovery framework PyTheus [25]. Each input is a one-dimensional array consisting of the 24 real-valued edge weights which corresponds to the quantum graph, and the output is the property value of the graph's corresponding state, $|\Phi(\omega)\rangle$. The network is then forward-trained on these examples with a mini-batch size of 5000 and a 95:5 train-test split. We utilize a learning rate scheduler that is initially set 1×10^{-3} (1×10^{-5} for the $[36^{26}]$ training architecture) and is gradually decreased in factors of 0.95 if, after every 25 training epochs, the test MSE does not change significantly. The network is run until convergence in the test MSE is maintained for over 400 training epochs. Table A1 lists the network architectures considered and the corresponding training results for each network. Between the networks' hidden layers, We employ the ReLU nonlinear activation function for all of our architectures except for the $[36^{26}]$ network, in which the ELU activation function with $\alpha = 0.1$ was used. The networks were encoded using PyTorch [50], and we employ the Adam optimizer [51] for both the forward and inverse training steps. A hyperparameter search was carried out on the number of neurons, N, in the generic neural network architecture $[N^4]$ towards predicting the GHZ-State fidelity. We considered architectures with $N = 10, 25, 40, 100, 200$, and 400. The hyperparameter search stopped once satisfactory improvements in the test MSE with respect to the simplest model considered were attained, which was achieved with $N = 400$ neurons. We found this architecture to be satisfactory for the purposes of predicting the W-State Fidelity, but not the value of $\mathrm{tr}(\overline{\rho^2_{\mathcal{M}}})$, which prompted us to consider instead the $[800^7]$ neural network. Table A2 showcases the results of our hyperparameter search.

Table A1. Training Details of all Neural Network Architectures featured in this work. Each architecture is listed in the format $[N_1, N_2, \ldots N_i]$, where N_i refers to the number of neurons of the ith hidden layer.

NN Architecture	Test MSE	Training Epochs
$[400^4]$	2.48×10^{-6}	2550
$[800^7]$	2.95×10^{-6}	2300
$[49^{10}]$	3.60×10^{-4}	2300
$[36^{26}]$	6.24×10^{-4}	2700
	4.12×10^{-6} (GHZ)	
$[400^3, 10]$	6.89×10^{-6} (W)	4200 (Both)

Table A2. Hyperparameter search on the number of neurons, N, in the neural network architecture $[N^4]$ towards predicting GHZ-state fidelity. We show the minimal test MSE achieved after convergence.

NN Architecture	Test MSE
$[10^4]$	2.63×10^{-4}
$[25^4]$	8.33×10^{-4}
$[40^4]$	3.97×10^{-4}
$[100^4]$	7.02×10^{-5}
$[200^4]$	5.24×10^{-6}
$[300^4]$	2.47×10^{-6}
	1.53×10^{-6} (GHZ)
$[400^4]$	5.98×10^{-4} (Concurrence)

We perform inverse training on the jth neuron in layer i each of our trained networks' architecture by employing gradient ascent on random input graphs towards maximizing the neuron's activations. This is done by transferring all of the trained networks' parameters up to and including the $(i-1)$th layer. The ith layer, which now forms the output layer of this intermediate network, consists solely of neuron j. We then sample randomly from a dataset consisting of 1 million input graphs and perform the dreaming routine for

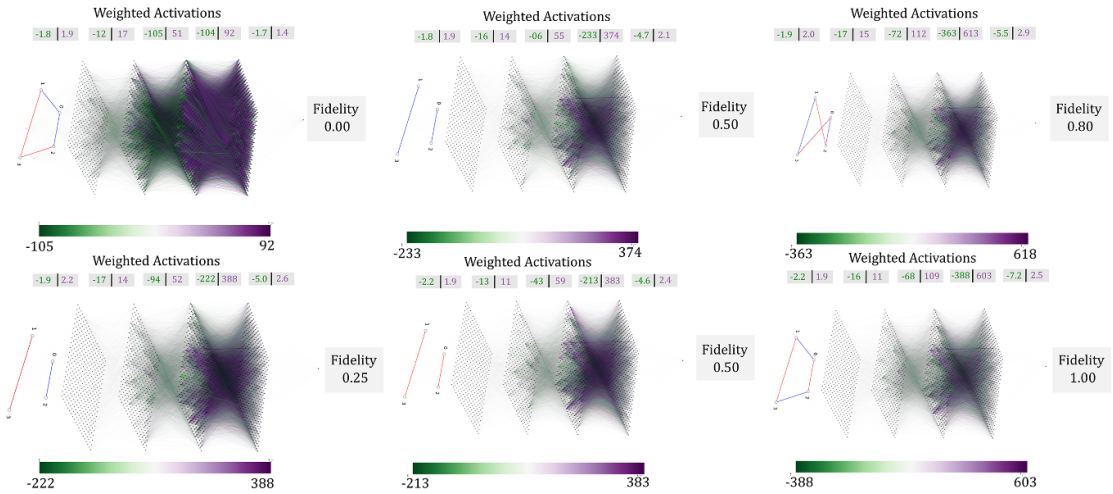

Figure A1. Plots of the *globally normalized* weighted activation for the $[400^4]$ neural network architectures when trained to predict the fidelity of the GHZ state. The plots are computed by taking the product of the trained neural network weights with the activations of the network when asked to predict $|\langle\Phi(\omega)\rangle \mathrm{GHZ}|^2$ for various states $|\Phi(\omega)\rangle$. Towards best visualizing each network, we filter out edges whose weighted activations are below a threshold of 0.05. We find that different PM directions and colourings change which neurons are activated in the intermediate layers, whereas the corresponding state's fidelity changes the magnitude of activations observed throughout the network. The different kinds of neuron activation patterns give us insight into how the neural network conceives its prediction.

100 000 iterations and a fixed learning rate of 1×10^{-4} when the network is tasked to optimize the GHZ- and W- state fidelities. For a single input graph, we observe inverse training times of approximately 5 min (0.003 seconds per iteration) for fidelity optimization tasks and 4 min (0.016 seconds per iteration) for concurrence minimization. We therefore dream with 15 000 iterations to compensate for the increased complexity of the architecture. Inverse training was done on an AMD Ryzen 5 4500U @ 2.38 GHZ CPU. It is possible to shorten the inverse training time by increasing the learning rate and implementing an early stopping criterion conditioned on the property value. For example, by increasing the learning rate to 1×10^{-3} and by stopping the dreaming process if, after 1000 iterations, the test MSE does not experience a change greater than 1×10^{-7}, the task of dreaming towards maximal GHZ- state fidelities finished after roughly 5000 iterations.

Appendix B. Neural network activation plots

Figure A1 displays the weighted activation patterns for the trained $[400^4]$ neural network architecture when tasked to make predictions on quantum graphs over a range of fidelity values. We observe that the activation patterns change depending on the kinds of PMs featured by the input graph. In tandem with our deep dreaming approach, we envision that we can reverse engineer what the neural network is observing in its computation of the GHZ-state fidelity by examining the activation patterns and through knowledge of what each individual neuron is precisely seeing.

Appendix C. Quantifying the quantum graph complexity

The complete Hilbert space $\mathcal{H} = \mathbb{H}_2 \otimes \mathbb{H}_2 \otimes \mathbb{H}_2 \otimes \mathbb{H}_2$ on which $|\Phi(\omega)\rangle$ is defined consists of 16 possible states. In the formulation of quantum graphs estabilshed in section 2.1, the probability amplitude of any quantum state $|\psi\rangle \in \mathcal{H}$ can be obtained via the weights of the three possible PMs which realize the state. The weights for each PM are obtained as follows,

$$p_{|\psi\rangle} = p_H^{|\psi\rangle} + p_V^{|\psi\rangle} + p_D^{|\psi\rangle} \tag{C1}$$

where

$$p_H^{|\psi\rangle} = |\omega_{01}\omega_{23}|^2 \tag{C2}$$

$$p_V^{|\psi\rangle} = |\omega_{03}\omega_{12}|^2 \tag{C3}$$

$$p_D^{|\psi\rangle} = |\omega_{02}\omega_{13}|^2 \tag{C4}$$

and $0 \leqslant |\omega_{a,b}|^2 \leqslant 1$, $a, b \in \{0, 1, 2, 3\}$ denote the weight of an edge with vertices a,b for the endpoint.

Repeating this procedure for every ket in $\mathcal{H}$, we obtain a 3×16 array of probabilities $p_{i,j}$ for the ith neuron of the jth hidden layer. We may think of the graphs' complexity in terms of the sum of probabilities that different possible events—here, the PMs corresponding to each quantum state—may occur. The overall complexity of structures observed by our neuron can, therefore, be quantified by calculating $\overline{p_{i,j}}$ over all of our representations, and computing the information entropy.

$$H_{i,j} = \sum_{p \in \overline{p_{i,j}}} -p \log_2 (p) \tag{C5}$$

We may iterate with this procedure across all of the neurons in the jth layer, yielding an array of information entropies out of which the mean information entropy of the layer, $\overline{H_j}$, may be obtained. This gives us an overall measure of the complexity of structures being observed at every point in the neural network.

ORCID iDs

Tareq Jaouni https://orcid.org/0009-0006-5661-2403
Carlos Ruiz-Gonzalez https://orcid.org/0000-0003-0545-360X
Ebrahim Karimi https://orcid.org/0000-0002-8168-7304

References

[1] Dawid A, Arnold J, Requena B, Gresch A, Płodzień M, Donatella K, Nicoli K A, Stornati P, Koch R and Büttner M 2022 Modern applications of machine learning in quantum sciences (arXiv:2204.04198)
[2] Krenn M, Landgraf J, Foesel T and Marquardt F 2023 Artificial intelligence and machine learning for quantum technologies *Phys. Rev.* A **107** 010101
[3] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 Machine learning and the physical sciences *Rev. Mod. Phys.* **91** 045002
[4] Krenn M *et al* 2022 On scientific understanding with artificial intelligence *Nat. Rev. Phys.* **4** 761
[5] Doran D, Schulz S and Besold T R 2017 *What Does Explainable ai Really Mean? A New Conceptualization of Perspectives* (arXiv:1710.00794)
[6] Tjoa E and Guan C 2021 A survey on explainable artificial intelligence (XAI):towards medical XAI *IEEE Trans. Neural Netw. Learn. Syst.* **32** 4793
[7] Burkart N and Huber M F 2021 A survey on the explainability of supervised machine learning *J. Artif. Intell. Res.* **70** 245
[8] Samek W, Wiegand T and Müller K-R 2017 Explainable artificial intelligence: understanding, visualizing and interpreting deep learning models (arXiv:2204.04198)
[9] Iten R, Metger T, Wilming H, del Rio L and Renner R 2020 Discovering physical concepts with neural networks *Phys. Rev. Lett.* **124** 010508
[10] Wetzel S J, Melko R G, Scott J, Panju M and Ganesh V 2020 Discovering symmetry invariants and conserved quantities by interpreting siamese neural networks *Phys. Rev. Res.* **2** 033499
[11] Dawid A, Huembeli P, Tomza M, Lewenstein M and Dauphin A 2021 Hessian-based toolbox for reliable and interpretable machine learning in physics *Mach. Learn. Sci. Technol.* **3** 015002
[12] Dawid A, Huembeli P, Tomza M, Lewenstein M and Dauphin A 2020 Phase detection with neural networks: interpreting the black box *New J. Phys.* **22** 115001
[13] Käming N, Dawid A, Kottmann K, Lewenstein M, Sengstock K, Dauphin A and Weitenberg C 2021 Unsupervised machine learning of topological phase transitions from experimental data *Mach. Learn.: Sci. Technol.* **2** 035037
[14] Wetzel S J and Scherzer M 2017 Machine learning of explicit order parameters: from the ising model to su(2) lattice gauge theory *Phys. Rev.* B **96** 184410
[15] Rocchetto A, Grant E, Strelchuk S, Carleo G and Severini S 2018 Learning hard quantum distributions with variational autoencoders *npj Quantum Inf.* **4** 28
[16] Flam-Shepherd D, Wu T C, Gu X, Cervera-Lierta A, Krenn M and Aspuru-Guzik A 2022 Learning interpretable representations of entanglement in quantum optics experiments using deep generative models *Nat. Mach. Intell.* **4** 544
[17] Frohnert F and van Nieuwenburg E 2024 Explainable representation learning of small quantum states *Mach. Learn.: Sci. Technol.* **5** 015001
[18] Krenn M, Erhard M and Zeilinger A 2020 Computer-inspired quantum experiments *Nat. Rev. Phys.* **2** 649
[19] Krenn M, Malik M, Fickler R, Lapkiewicz R and Zeilinger A 2016 Automated search for new quantum experiments *Phys. Rev. Lett.* **116** 090405
[20] Nichols R, Mineh L, Rubio J, Matthews J C F and Knott P A 2019 Designing quantum experiments with a genetic algorithm *Quantum Sci. Technol.* **4** 045012
[21] O'Driscoll L, Nichols R and Knott P A 2019 A hybrid machine learning algorithm for designing quantum experiments *Quantum Mach. Intell.* **1** 5
[22] Valcarce X, Sekatski P, Gouzien E, Melnikov A and Sangouard N 2023 Automated design of quantum-optical experiments for device-independent quantum key distribution *Phys. Rev.* A **107** 062607
[23] Arrazola J M, Bromley T R, Izaac J, Myers C R, Brádler K and Killoran N 2019 Machine learning method for state preparation and gate synthesis on photonic quantum computers *Quantum Sci. Technol.* **4** 024004
[24] Krenn M, Kottmann J S, Tischler N and Aspuru-Guzik A 2021 Conceptual understanding through efficient automated design of quantum optical experiments *Phys. Rev.* X **11** 031044
[25] Ruiz-Gonzalez C, Arlt S, Petermann J, Sayyad S, Jaouni T, Karimi E, Tischler N, Gu X and Krenn M 2023 Digital discovery of 100 diverse quantum experiments with PyTheus *Quantum* **7** 1204
[26] Krenn M, Hochrainer A, Lahiri M and Zeilinger A 2017 Entanglement by path identity *Phys. Rev. Lett.* **118** 080401

[27] Gao X, Erhard M, Zeilinger A and Krenn M 2020 Computer-inspired concept for high-dimensional multipartite quantum gates *Phys. Rev. Lett.* **125** 050501
[28] Arlt S, Ruiz-Gonzalez C and Krenn M 2022 Digital discovery of a scientific concept at the core of experimental quantum optics (arXiv:2210.09981)
[29] Jaouni T, Gao X, Arlt S, Krenn M and Karimi E 2023 Experimental solutions to the high-dimensional mean king's problem *Opt. Quantum* **1** 49–54
[30] Mordvintsev A, Olah C and Tyka M 2015 Inceptionism: going deeper into neural networks (available at: https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html)
[31] Lifshitz R 2022 Quantum deep dreaming: a novel approach for quantum circuit design (arXiv:2211.04343)
[32] Shen C, Krenn M, Eppel S and Aspuru-Guzik A 2021 Deep molecular dreaming: inverse machine learning for de-novo molecular design and interpretability with surjective representations *Mach. Learn.: Sci. Technol.* **2** 03LT02
[33] Schindler F, Regnault N and Neupert T 2017 Probing many-body localization with neural networks *Phys. Rev.* B **95** 245134
[34] Seif A, Hafezi M and Jarzynski C 2021 Machine learning the thermodynamic arrow of time *Nat. Phys.* **17** 105
[35] Krenn M, Gu X and Zeilinger A 2017b Quantum experiments and graphs: multiparty states as coherent superpositions of perfect matchings *Phys. Rev. Lett.* **119** 240403
[36] Gu X, Erhard M, Zeilinger A and Krenn M 2019a Quantum experiments and graphs II: quantum interference, computation and state generation *Proc. Natl Acad. Sci.* **116** 4147
[37] Gu X, Chen L, Zeilinger A and Krenn M 2019b Quantum experiments and graphs. III. High-dimensional and multiparticle entanglement *Phys. Rev.* A **99** 032338
[38] Ding Y, Llewellyn D, Faruque I, Paesani S, Bacco D, Rottwitt K, Laing A, Thompson M, Wang J and Oxenløwe L K 2020 Integrated quantum photonics on silicon platform *Optical Fiber Communication Conf. (OFC) 2020* (Optica Publishing Group) p W4C.6
[39] Feng L-T, Guo G-C and Ren X-F 2020 Progress on integrated quantum photonic sources with silicon *Adv. Quantum Technol.* **3** 1900058
[40] Pelucchi E *et al* 2022 The potential and global outlook of integrated photonics for quantum technologies *Nat. Rev. Phys.* **4** 194
[41] Bao J, Fu Z, Pramanik T, Mao J, Chi Y, Cao Y, Zhai C, Mao Y, Dai T and Chen X 2023 Very-large-scale integrated quantum graph photonics *Nat. Photon.* **1** 1204
[42] Qian K, Wang K, Chen L, Hou Z, Krenn M, Zhu S and Ma X-S 2023 Multiphoton non-local quantum interference controlled by an undetected photon *Nat. Commun.* **14** 1480
[43] Feng L-T, Zhang M, Liu D, Cheng Y-J, Guo G-P, Dai D-X, Guo G-C, Krenn M and Ren X-F 2023 On-chip quantum interference between the origins of a multi-photon state *Optica* **10** 105
[44] Greenberger D M, Horne M A and Zeilinger A 1989 Going beyond bell's theorem *Bell's Theorem, Quantum Theory and Conceptions of the Universe* (Springer) pp 69–72
[45] Cabello A 2002 Bell's theorem with and without inequalities for the three-qubit Greenberger-Horne-Zeilinger and W states *Phys. Rev.* A **65** 032108
[46] Wootters W K 2001 Entanglement of formation and concurrence *Quantum Inf. Comput.* **1** 27
[47] Nguyen A, Yosinski J and Clune J 2016 Multifaceted feature visualization: uncovering the different types of features learned by each neuron in deep neural networks (arXiv:1602.03616)
[48] Simonyan K, Vedaldi A, and Zisserman A 2013 Deep inside convolutional networks: visualising image classification models and saliency maps (arXiv:1312.6034)
[49] Hoogeboom E, Satorras V G, Vignac C and Welling M 2022 Equivariant diffusion for molecule generation in 3d *Int. Conf. on Machine Learning* (PMLR) pp 8867–87
[50] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N and Antiga L *et al* 2019 Pytorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* vol 32
[51] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)