New Journal of Physics

The open access journal at the forefront of physics

PAPER

Quantum codes from neural networks

Johannes Bausch[1] and **Felix Leditzky**[2]

[1]   DAMTP, University of Cambridge, United Kingdom
[2]   JILA & CTQM, University of Colorado Boulder, United States of America

E-mail: jkrb2@cam.ac.uk and felix.leditzky@jila.colorado.edu

Abstract

We examine the usefulness of applying neural networks as a variational state ansatz for many-body quantum systems in the context of quantum information-processing tasks. In the neural network state ansatz, the complex amplitude function of a quantum state is computed by a neural network. The resulting multipartite entanglement structure captured by this ansatz has proven rich enough to describe the ground states and unitary dynamics of various physical systems of interest. In the present paper, we initiate the study of neural network states in quantum information-processing tasks. We demonstrate that neural network states are capable of efficiently representing quantum codes for quantum information transmission and quantum error correction, supplying further evidence for the usefulness of neural network states to describe multipartite entanglement. In particular, we show the following main results: (a) neural network states yield quantum codes with a high coherent information for two important quantum channels, the generalized amplitude damping channel and the dephrasure channel. These codes outperform all other known codes for these channels, and cannot be found using a direct parametrization of the quantum state. (b) For the depolarizing channel, the neural network state ansatz reliably finds the best known codes given by repetition codes. (c) Neural network states can be used to represent absolutely maximally entangled states, a special type of quantum error-correcting codes. In all three cases, the neural network state ansatz provides an efficient and versatile means as a variational parametrization of these highly entangled states.

1. Introduction

The exponential growth of the Hilbert space dimension in the number of particles is both a blessing and curse for quantum science: On the one hand, it is crucial to the widely-believed computational advantage of quantum computers over classical ones, but on the other hand it renders many questions about properties of many-body systems intractable. Yet we know that the 'physical' corner of this Hilbert space has to be small: local Hamiltonians with highly-entangled ground states only require a polynomial number of parameters to describe, as do quantum circuits of polynomial depth.

This fact motivates the use of variational representations of quantum states to solve a large class of problems. At the heart of any variational ansatz is the idea to preserve as much information about the quantum state as possible, while discarding irrelevant features. Quantum mechanical properties of a state are fundamentally dictated by its entanglement, which captures quantum correlations between its subsystems.

For instance, correlation length in many-body spin systems is tightly linked to the existence of a spectral gap [Has07, GH16]. For gapped one-dimensional systems (which follow an entanglement entropy area law), one can use matrix product states (MPS) with polynomial bond dimension to efficiently represent ground states [FNW92, LVV15, Ara+13]. The MPS ansatz has further proven useful e.g. in the study of critical systems [Pir+12] or in the continuum limit [Cue+17]. Other tensor network states include MERA and higher-dimensional variants such as PEPS—applied e.g. in the context of renormalization [Vid07, VC04], and proven similarly successful as part of numerical techniques [Orú14].

A relatively recent development is the use of neural network states as a variational ansatz, where the network is used as a function to calculate the state amplitudes [CT17]. There are many possible neural network architectures to choose from: one proposed model is to use restricted Boltzmann machines (RBMs) to represent e.g. the ground states and unitary dynamics of a transverse-field Ising model and the antiferromagnetic Heisenberg model [CT17], volume-law entanglement and the ground state of even long-range Hamiltonians [DLD17], as well as ground states of various stabilizer Hamiltonians, including the surface code [Jia+18]. While there exist local Hamiltonians that cannot be represented efficiently with shallow RBM architectures, it has been shown that deep RBM networks can in fact represent *most* physical states, which includes those that can be created by poly-depth quantum circuits, or ground states of local Hamiltonians with a $1/\mathrm{poly}$ spectral gap [GD17].

Apart from describing the physics of many-body systems, entanglement also plays a crucial role in information-processing tasks: teleportation [Ben+93], superdense coding [BW92], and entanglement-assisted classical [Ben+99] and quantum [DHW04] communication all build on *bipartite* entanglement as a resource. In contrast, for certain tasks such as quantum information transmission through many uses of a quantum channel, or the encoding of quantum information in quantum error correction codes, the crucial property is *multipartite entanglement*, which encapsulates correlations among all the constituents of the system simultaneously [Has07].

1.1. Main results

We demonstrate that neural network states with only polynomially many parameters in the system size (which, in this context, we call *efficient*) are capable of representing quantum codes for quantum information transmission and quantum error correction. In particular, we show the following:

- The neural network state ansatz finds new quantum codes with a high coherent information (CI) that outperform all previously known codes for two channel models, the generalized amplitude damping channel (GADC) and the dephrasure channel. For the GADC, the new codes also increase the threshold of the channel, i.e. the boundary of the interval in the parameter space with positive quantum capacity. For both channels, the new codes cannot be found with 'traditional' numerical methods, i.e. a direct parametrization of the complex amplitudes of the quantum state.

- For the depolarizing channel, neural network states can efficiently represent the best known codes. We carry out a detailed comparison of different network architectures, showing that FF networks converge faster than RBMs with comparable parameter counts in almost all tested cases. Furthermore, we constructively prove that the best known codes (repetition codes, and products thereof) can be obtained efficiently with both an RBM and a FF architecture.

- Neural network states can be used to parametrize so-called 'absolutely maximally entangled' (AME) states. These AME states, defined on n systems of local dimension d each, are examples of quantum error-correcting codes with the property that they are completely mixed after tracing out at least half of the systems. Besides their quantum error correction capabilities, AME states are useful in multi-user information-theoretic tasks such as open-destination teleportation, secret sharing or entanglement swapping that require maximal entanglement across different choices of bipartitions [HC13, Hel+12].

The properties of both quantum codes with high coherent information and AME states are the result of the non-trivial multipartite entanglement present in these states. The main finding of this paper is that for both high-CI states and AME states, a neural network state ansatz is able to faithfully represent this multipartite entanglement, which we demonstrate empirically for small problem instances. We furthermore provide numerical evidence that the variational ansatz vastly outperforms a full state parametrization for the respective learning tasks.

1.2. Structure of this paper

This paper is structured as follows. In section 2 we introduce the quantum capacity of a channel and state the corresponding coding theorem which expresses the quantum capacity as a regularized formula in terms of an entropic quantity called the coherent information. We then discuss how lower bounds on the quantum capacity can be obtained by solving an entropic optimization problem. In section 3 we review neural network states based on RBMs and feed-forward nets. We then present our main results about representing quantum codes with neural network states. In section 4 we discuss the GADC and the dephrasure channel. We show that the neural network state ansatz finds new quantum codes providing the strongest lower bounds to date on the quantum capacities of these channels. Moreover, we demonstrate that these new codes are not found using a 'direct' parametrization of quantum states. We then show in section 5 for the depolarizing channel how tensor products of repetition codes—i.e. the known optimal codes for $k \leqslant 9$ uses of this channel—can be efficiently represented

using FF and RBM networks, and comment on the trainability of our chosen network architectures. Finally, in section 6 we demonstrate how known examples of AME states can be efficiently represented using neural networks, and we comment on the trainability of the network architectures that we used. We conclude in section 7 with a discussion of our results and open problems.

In the appendices, we give more details about certain aspects of the paper. In appendices A and B we state explicit formulas for the coherent information of weighted repetition codes for the GADC and dephrasure channel, respectively, which serve as benchmarks for our quantum codes from neural networks. We also supply additional data obtained in our numerical investigations. In appendix C we give an overview of the best known codes for the depolarizing channel, and provide an analytical construction of these codes for neural networks with various architectures. In appendix D we provide some background information on AME states, and prove a useful bound on a trace distance parameter indicating how close a state is to being AME. In appendix E we discuss possible encodings of d-ary input strings to neural networks. In appendix F we comment on the role of activation functions for quantum codes; furthermore, we propose a novel NN Schmidt decomposition ansatz, which we benchmark against a full NN parametrization for the depolarizing channel. In appendix G we give a high-level explanation of the global derivative-free numerical optimization techniques used in our paper. Finally, we provide additional numerical data for some of our results in appendix H.

We encourage researchers to adopt our methods by providing full access to our code (in C++ and MATLAB) that was used to obtain the numerical results of this paper. These code files can be found in the 'Ancillary files' section of the arXiv post of this paper [Anc]. In MATLAB, we made use of the MATLAB Global Optimization Toolbox, as well as quantinf [Cub05] and QETLAB [Joh16]. In C++, we made use of NLopt [Joh] with the CCSA algorithm [Sva02] as well as PAGMO [BIM18].

2. The quantum capacity of a quantum channel

A point-to-point communication link between quantum systems can be modeled by a *quantum channel*. For quantum systems A and B with underlying (finite-dimensional) Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively, a quantum channel $\mathcal{N}: A \to B$ is a linear, completely positive, trace-preserving map between the algebras of linear operators $\mathcal{B}(\mathcal{H}_A)$ and $\mathcal{B}(\mathcal{H}_B)$. A quantum state ρ_A on A is a linear positive semidefinite operator with unit trace. A quantum state ψ_A with rank 1 is called pure, and can be identified with a normalized vector $|\psi\rangle_A \in \mathcal{H}_A$ such that $\psi_A = |\psi\rangle \langle\psi|_A$.

The communication capabilities of a quantum channel are characterized by various *capacities*, depending on what kind of information one attempts to transmit faithfully through the channel. The *quantum capacity* $Q(\mathcal{N})$ of a quantum channel $\mathcal{N}: A \to B$ characterizes the optimal rate of faithful quantum information transmission through the channel. $Q(\mathcal{N})$ can be defined in terms of the operational task of *entanglement generation* as follows.

Suppose Alice, the sender, prepares a pure state ψ_{RA^n} in her laboratory and sends the A^n-part to Bob through n independent uses of the quantum channel $\mathcal{N}$[3]. Upon receiving the quantum systems from Alice, Bob applies some decoding operation $\mathcal{D}_n: B^n \to R'$ to the output, yielding the final state $\sigma_{RR'} = (\mathrm{id}_R \otimes \mathcal{D}_n \circ \mathcal{N}^{\otimes n})(\psi_{RA^n})$. The goal for Alice and Bob is to obtain a final state $\sigma_{RR'}$ that is close (in a suitable distance measure) to a maximally entangled state $|\Phi^{M_n}\rangle_{RR'}$ of Schmidt rank M_n, i.e. $|\Phi^{M_n}\rangle_{RR'} = M_n^{-1/2} \sum_{i=1}^{M_n} |i\rangle_R \otimes |i\rangle_{R'}$ for some basis $\{|i\rangle\}_i$. If there is an entanglement generation protocol such that $\sigma_{RR'}$ converges to $\Phi_{RR'}^{M_n}$ with respect to the chosen distance as $n \to \infty$, then $\lim_{n\to\infty} \frac{1}{n} \log M_n$ is called an *achievable rate*. The quantum capacity $Q(\mathcal{N})$ is defined as the supremum over all achievable rates.

$Q(\mathcal{N})$ can be expressed in terms of a *regularized formula* as [Llo97, Sho02, Dev05]

$$Q(\mathcal{N}) = \lim_{n\to\infty} \frac{1}{n} Q^{(1)}(\mathcal{N}^{\otimes n}) = \sup_{n\in\mathbb{N}} \frac{1}{n} Q^{(1)}(\mathcal{N}^{\otimes n}), \tag{1}$$

where the *channel coherent information* $Q^{(1)}(\mathcal{N})$ is defined as

$$Q^{(1)}(\mathcal{N}) = \max_{|\psi\rangle_{RA}} Q^{(1)}(\psi_{RA}, \mathcal{N}) = \max_{\psi_{RA}} S(\mathcal{N}(\psi_A)) - S((\mathrm{id}_R \otimes \mathcal{N})(\psi_{RA})), \tag{2}$$

with the von Neumann entropy $S(\rho) := -\mathrm{tr}(\rho \log(\rho))$.

Formula 1 for the quantum capacity involves the evaluation of the channel coherent information $Q^{(1)}(\cdot)$ over an (in principle) unbounded number of channel copies. If the channel coherent information is *weakly additive*, $Q^{(1)}(\mathcal{N}^{\otimes n}) \leqslant n Q^{(1)}(\mathcal{N})$, then the regularization disappears and equation (1) becomes $Q(\mathcal{N}) = Q^{(1)}(\mathcal{N})$. Weak additivity of the channel coherent information is only known to hold for certain classes of channels such as degradable channels [DS05]. Moreover, there are examples of quantum channels for which the channel coherent information is strictly *superadditive*, $Q^{(1)}(\mathcal{N}^{\otimes n}) > n Q^{(1)}(\mathcal{N})$ for some n, rendering

---

[3] That is, Alice uses the A^n-part of ψ_{RA^n} as the input to the channel $\mathcal{N}^{\otimes n}$.

the regularization over n in the quantum capacity formula 1 necessary in general [DSS98]. However, for so-called *low-noise* channels that are close in diamond norm to a noiseless channel, the effect of superadditivity of coherent information cannot be too large, and the single-letter coherent information is essentially the right answer [LLS18b, Sut+17]. In this paper, we are interested in the *high-noise* regime where superadditivity of channel coherent information typically occurs.

An important part of the quantum capacity theorem in equation (1) is the fact that the channel coherent information is an *achievable rate* [Llo97, Sho02, Dev05]:

$$Q(\mathcal{N}) \geqslant Q^{(1)}(\mathcal{N}). \qquad (3)$$

Using block codes, this can be generalized to $Q(\mathcal{N}) \geqslant \frac{1}{n} Q^{(1)}(\mathcal{N}^{\otimes n})$ for all $n \in \mathbb{N}$. The rough proof idea of equation (3) is the following: assume that $|\psi\rangle_{RA}$ is a pure state with strictly positive coherent information, $Q^{(1)}(\psi, \mathcal{N}) > 0$. Once Alice and Bob share k copies of the state $\sigma_{RB} = (\mathrm{id}_R \otimes \mathcal{N})(\psi_{RA})$ (which they can achieve by Alice sending the A^k part of the state $\psi_{RA}^{\otimes k}$ to Bob through $\mathcal{N}^{\otimes k}$) for a sufficiently large k, there is a protocol defined in terms of the typical subspaces of $\sigma_{RB}^{\otimes k}$ that allows Alice and Bob to generate entanglement between them at a rate of $r - \delta$ for arbitrarily small $\delta \in (0, r)$, where r is equal to the coherent information of the state σ, that is, $r = I(R\rangle B)_\sigma = Q^{(1)}(\psi, \mathcal{N})$ [Dev05, Hay+08]. Here, $I(R\rangle B)_\sigma = S(\sigma_B) - S(\sigma_{RB})$ is the coherent information of the bipartite state σ_{RB}.

In this operational picture, we can think of ψ_{RA} as the *inner code*, whereas the (1-LOCC assisted) distillation protocol manipulating $\sigma_{RB}^{\otimes k}$ is the *outer code*. The rate at which the full protocol generates entanglement is solely determined by the (strictly positive) coherent information of the inner code ψ_{RA}. Hence, in this paper we refer to the inner code ψ_{RA} simply as a *quantum code*. The main objective of this paper is to find quantum codes $|\psi\rangle_{RA^n}$ that achieve high coherent information $\frac{1}{n} Q^{(1)}(\psi_{RA^n}, \mathcal{N}^{\otimes n}) > 0$. To find such quantum codes, we use the neural network state ansatz introduced in [CT17]. In the next section, we review different variants of this ansatz.

3. Neural network states

For simplicity we consider in the following a system consisting of n qubits, that is, a collection of n 2-dimensional quantum systems each described by a Hilbert space isomorphic to $\mathbb{C}^2$. The state space of the n qubits is described by the tensor space $(\mathbb{C}^2)^{\otimes n}$ with the 'computational basis' $\{|0\rangle, |1\rangle\}^{\otimes n}$, and a general pure normalized quantum state $|\psi\rangle \in \mathcal{H}^{\otimes n}$ can be written as

$$|\psi\rangle = \frac{1}{C} \sum_{i_1, \ldots, i_n = 0, 1} \psi(i_1, \ldots, i_n)|i_1\rangle \otimes \ldots \otimes |i_n\rangle = \frac{1}{C} \sum_{i^n \in \{0,1\}^n} \psi(i^n)|i^n\rangle. \qquad (4)$$

Here, C is a normalization constant ensuring $\langle\psi|\psi\rangle = 1$, the set of binary strings of length n is denoted by $\{0, 1\}^n$, and for a string $i^n = (i_1, \ldots, i_n) \in \{0, 1\}^n$ we define $|i^n\rangle := |i_1\rangle \otimes \ldots \otimes |i_n\rangle$. Evidently, a full description of the quantum state $|\psi\rangle$ consists of a list of the 2^n complex amplitudes $\psi(i^n)$, corresponding to $2 \cdot 2^n - 1$ real degrees of freedom.

For a neural network state ψ, the amplitude function $\psi(i^n)$ in equation (4) is computed from the input string i^n using a neural network. There are different network architectures that can be used, and we describe a few common choices in the following subsections.

3.1. Restricted Boltzmann states
The first architecture—and one of the most well-studied ones, see e.g. [Gla+18] for an excellent review—are RBM. They have proven particularly fruitful as a variational ansatz for representing various ground states of local Hamiltonians [CT17], notably surpassing fidelity as compared to other neural network architectures in some cases.

A Boltzmann machine has visible and hidden nodes (see figure 1). A set of complex variables is assigned to each node; we denote the visible units with $i_1, \ldots, i_n$, and the hidden units with $h_1, \ldots, h_m$. Each link between nodes corresponds to an Ising-type coupling, which defines an energy function (which one can think of as a Hamiltonian)

$$\mathbf{H}_{\mathrm{RBM}} = \sum_{l=1}^{n} a_l i_l + \sum_{l=1}^{m} b_l h_l + \sum_{k<l} W_{kl} i_k h_l. \qquad (5)$$

The two vectors $a \in \mathbb{C}^n$ and $b \in \mathbb{C}^m$ define a bias over the visible and hidden nodes, respectively, while the matrix $W \in \mathbb{C}^{m \times n}$ defines the coupling between the two layers. The energy of the system allows us to define a complex probability distribution over the vectors i and h via $\mathbb{P}(i, h) := \exp(-\mathbf{H}(i, h))/Z$ with partition function $Z = \sum_{i,k} \mathbb{P}(i, h)$.

To extract a weight $\psi(i^n)$ used to assemble a state via equation (4), we simply trace out the hidden nodes of the RBM, which yields a marginal probability distribution over the input nodes. We obtain
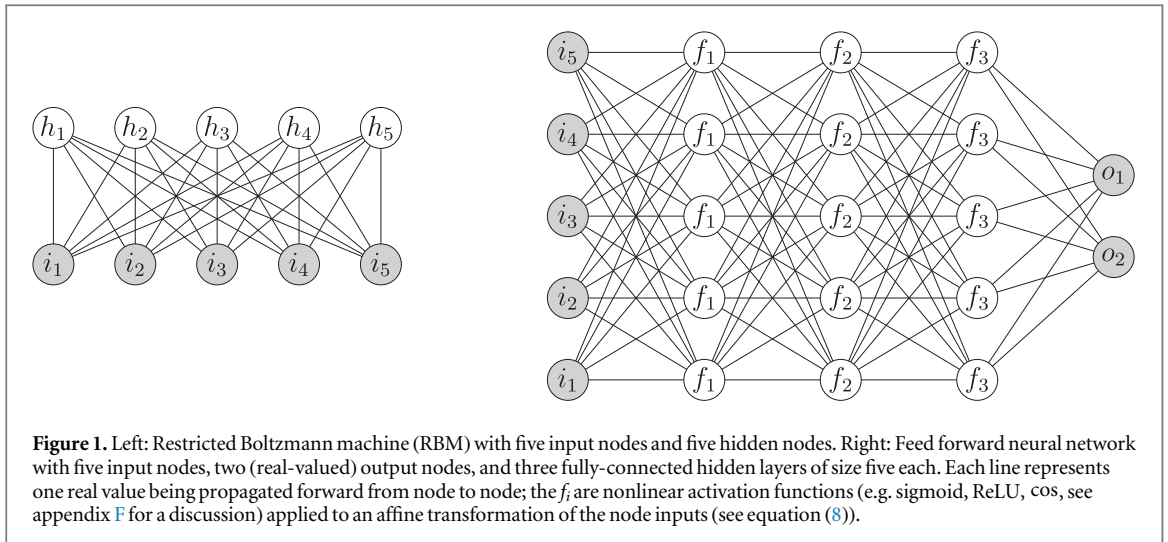
Figure 1. Left: Restricted Boltzmann machine (RBM) with five input nodes and five hidden nodes. Right: Feed forward neural network with five input nodes, two (real-valued) output nodes, and three fully-connected hidden layers of size five each. Each line represents one real value being propagated forward from node to node; the f_i are nonlinear activation functions (e.g. sigmoid, ReLU, $\cos$, see appendix F for a discussion) applied to an affine transformation of the node inputs (see equation (8)).
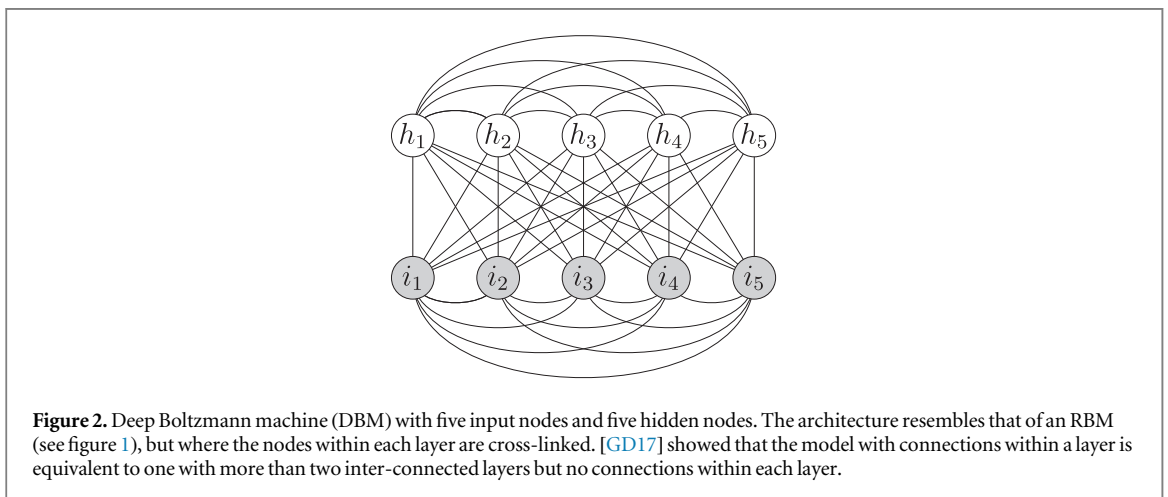

Figure 2. Deep Boltzmann machine (DBM) with five input nodes and five hidden nodes. The architecture resembles that of an RBM (see figure 1), but where the nodes within each layer are cross-linked. [GD17] showed that the model with connections within a layer is equivalent to one with more than two inter-connected layers but no connections within each layer.

$$|\psi_{\mathrm{RBM}}\rangle = \sum_{i^n \in \{0,1\}^n} \sum_{h^n \in \{0,1\}^n} \frac{\exp(-\mathbf{H}(i^n, h^n))}{Z} |i^n\rangle. \tag{6}$$

If we take all parameters a, b and W to be real-valued, the resulting state will only have real non-negative weights. In order to retain full generality in the RBM ansatz, the network weights are typically chosen to be complex [CT17].

3.2. Deep Boltzmann states
While RBM states struggle to represent e.g. ground states for local Hamiltonians with even mildly-decaying spectral gap, adding links between the nodes within each layer yields a model with vastly greater representative power [GD17, Gla+18]—deep Boltzmann machines (DBMs, see figure 2).

In analogy to equation (5), we can define an energy function for a DBM by introducing additional coupling matrices $D \in \mathbb{C}^{m \times m}$ and $C \in \mathbb{C}^{n \times n}$ for the hidden and visible nodes, respectively. This yields an overall Hamiltonian

$$\mathbf{H}_{\mathrm{DBM}} = \mathbf{H}_{\mathrm{RBM}} + \sum_{k<l} C_{kl} i_k i_l + \sum_{k<l} D_{kl} h_k h_l. \tag{7}$$

The way one obtains a state from a DBM follows the same method as for an RBM.

3.3. Feed-forward network states
The third architecture is obtained by using the most prominent neural network model to date, feed-forward nets, to represent quantum states. This has proven successful in a number of cases [CL18, Sai17].

A feed-forward network consists of a visible layer $v = i^n$ with input nodes i_1, ..., i_n, a fixed number H of hidden layers $h^{(j)}$ of width M_j, and an output layer o with two output nodes o_1 and o_2 (see figure 1). Each hidden neuron $h_k^{(j)}$ for $j \in [H]$ and $k \in [M_j]$ is assigned a *bias* $b_k^{(j)}$. Here, we use the notation $[n] := \{1, ..., n\}$ for

$n \in \mathbb{N}$. The interactions between two hidden layers $h^{(j-1)}$ and $h^{(j)}$ are mediated by *weight matrices* $(W_{kl}^{(j)})_{kl}$ where $k \in [M_j]$ and $l \in [M_{j-1}]$. The weight matrix $W^{(1)}$ mediates between the visible layer and the first hidden layer, and the weight matrix $W^{(H+1)}$ mediates between the last hidden layer $h^{(H)}$ and the output layer o with bias $b^{(H+1)}$. In each hidden layer $h^{(j)}$ the state of the neurons is processed with a nonlinear activation function f_j. In the following, we interpret the visible layer v, the hidden layers $h^{(j)}$, and the output layer o as column vectors, and functions are evaluated component-wise. Given the input $v = i^n$, the amplitude function $\psi(i^n)$ is computed as follows:

$$h^{(1)} = f_1(W^{(1)}v + b^{(1)})$$
$$h^{(j)} = f_j(W^{(j)}h^{(j-1)} + b^{(j)}) \quad \text{for } j = 2, \ldots, H$$
$$o = W^{(H+1)}h^{(H)} + b^{(H+1)}$$

$$\text{Cartesian:} \quad \psi(i^n) = o_1 + io_2$$
$$\text{Polar:} \quad \psi(i^n) = \exp(o_1 + io_2). \tag{8}$$

A network architecture is specified by the data $(H, \{M_j, f_j\}_{j \in [H]})$. Common choices for the activation functions are the sigmoid function $\sigma(x) := (1 + \exp(-x))^{-1}$, the hyperbolic tangent tanh, or the rectified linear unit $\mathrm{ReLU}(x) = \max\{0, x\}$, which are depicted in figure F1. From a theoretical point of view these choices are all equivalent, since feed-forward networks as described above are *universal*: With a single hidden layer, they can approximate any given function to arbitrary precision provided the activation function is non-constant and the number of hidden neurons is sufficiently large [Kol61, Hor91]. However, in practice the choice of activation functions has to be tailored to the problem at hand to achieve good numerical results. In appendix F, we elaborate on the heuristics of choosing activation functions for neural network states; of particular interest in this context is that periodic activation functions such as cosine seem to be able to capture more of the structure of various quantum states [CL18]. We prove analytically in appendix C.2 that periodic activation functions are also beneficial in representing good quantum codes.

4. New quantum codes using a neural network state ansatz

4.1. Generalized amplitude damping channel

The first quantum channel for which we investigate the neural network state ansatz is the GADC $\mathcal{A}_{\gamma,N}$. It is defined in terms of two parameters $\gamma, N \in [0, 1]$ and acts on a qubit state ρ as $\mathcal{A}_{\gamma,N} = \sum_{i=1}^{4} A_i \rho A_i^\dagger$, where
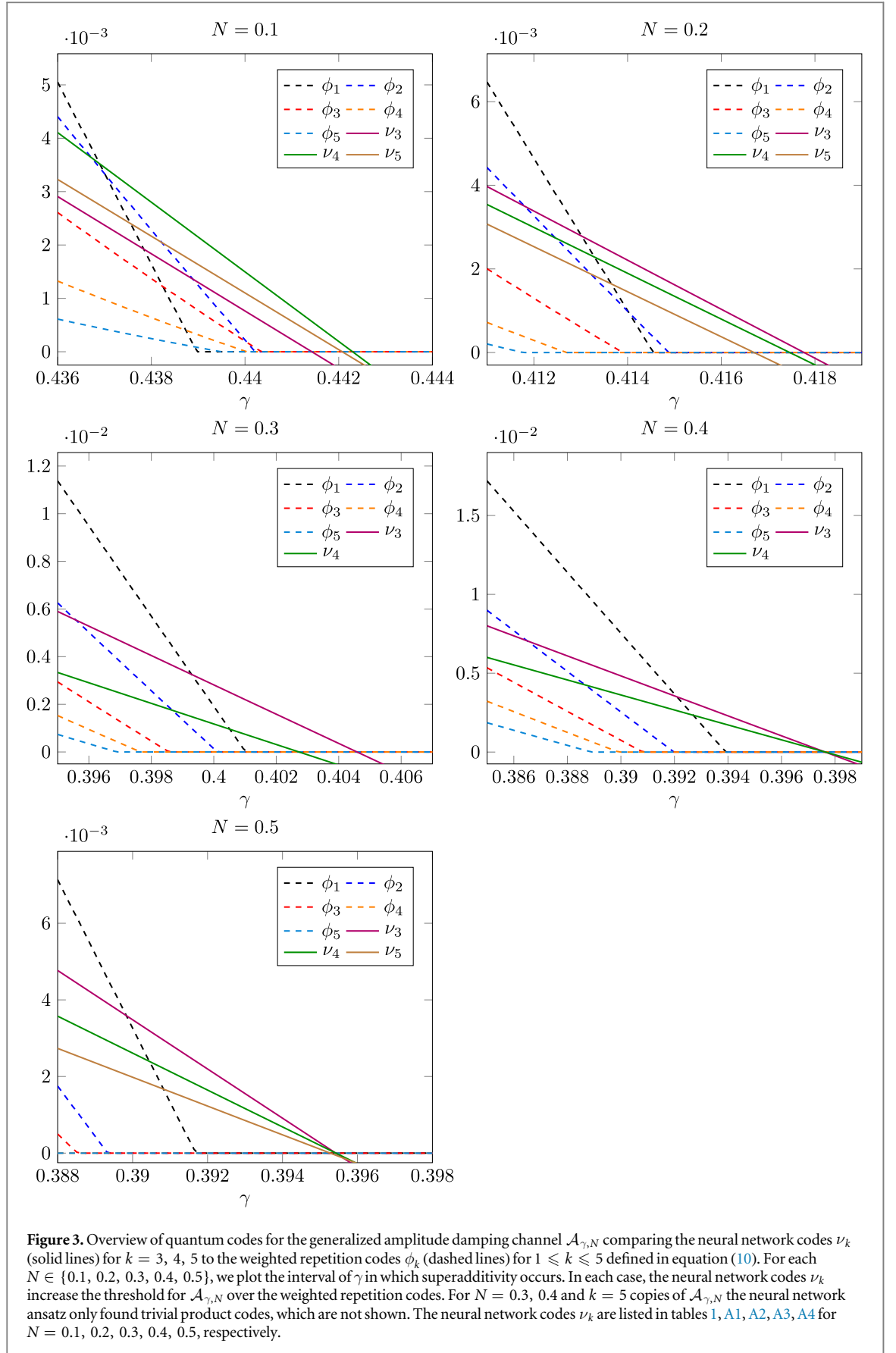
$$A_1 = \sqrt{1 - N}(|0\rangle\langle 0| + \sqrt{1 - \gamma}|1\rangle\langle 1|)$$
$$A_2 = \sqrt{\gamma(1 - N)}|0\rangle\langle 1|$$
$$A_3 = \sqrt{N}(\sqrt{1 - \gamma}|0\rangle\langle 0| + |1\rangle\langle 1|)$$
$$A_4 = \sqrt{\gamma N}|1\rangle\langle 0|. \tag{9}$$

The GADC models the dynamics of a qubit in contact with a thermal bath at temperature N and transition probability γ between the ground state $|0\rangle$ and the excited state $|1\rangle$. This quantum channel is a realistic noise model in various physical processes such as relaxation processes of spin systems, superconducting quantum computers, and loss processes in linear optical systems [Mya+00, Tur+00, CB08, Zou+17]. Furthermore, for $N = 0$ the GADC reduces to the well-known amplitude damping channel modeling energy dissipation of a qubit.

While the quantum capacity of the amplitude damping channel $\mathcal{A}_{\gamma,0}$ is equal to its (additive) single-letter coherent information for all $\gamma \in [0, 1]$ and can be computed efficiently [GF05], the quantum capacity of the more general noise model $\mathcal{A}_{\gamma,N}$ with $N \in (0, 1)$ is unknown. Various upper bounds on $Q(\mathcal{A}_{\gamma,N})$ have been computed in the recent work [KSW19], but so far achievable rates (i.e. lower bounds on $Q(\mathcal{A}_{\gamma,N})$) improving upon the single-letter coherent information $Q^{(1)}(\mathcal{A}_{\gamma,N})$ have not been studied extensively. We prove in this section that for $N \in (0, 1)$ and particular intervals of γ the channel coherent information $Q^{(1)}(\mathcal{A}_{\gamma,N})$ of the GADC is superadditive. As shown in the discussion below and in figure 3, superadditivity is achieved by, e.g. weighted repetition codes

$$|\phi_k^\lambda\rangle := \sqrt{\lambda}|0\rangle_R \otimes |0\rangle_A^{\otimes k} + \sqrt{1 - \lambda}|1\rangle_R \otimes |1\rangle_A^{\otimes k}. \tag{10}$$

A compact formula for the coherent information of this code in terms of an optimization over the weight $\lambda \in [0, 1]$ and arbitrary blocklength k can easily be derived (see appendix A). Note that the optimal single-letter coherent information $Q^{(1)}(\mathcal{A}_{\gamma,N})$ is achieved by equation (10) with $k = 1$ and optimized weight parameter $\lambda \in [0, 1]$ [GP+09]. We will show in this section that the neural network state ansatz finds superadditive codes for the GADC that substantially outperform weighted repetition codes.

Figure 3. Overview of quantum codes for the generalized amplitude damping channel $\mathcal{A}_{\gamma,N}$ comparing the neural network codes ν_k (solid lines) for $k = 3, 4, 5$ to the weighted repetition codes ϕ_k (dashed lines) for $1 \leqslant k \leqslant 5$ defined in equation (10). For each $N \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, we plot the interval of γ in which superadditivity occurs. In each case, the neural network codes ν_k increase the threshold for $\mathcal{A}_{\gamma,N}$ over the weighted repetition codes. For $N = 0.3, 0.4$ and $k = 5$ copies of $\mathcal{A}_{\gamma,N}$ the neural network ansatz only found trivial product codes, which are not shown. The neural network codes ν_k are listed in tables 1, A1, A2, A3, A4 for $N = 0.1, 0.2, 0.3, 0.4, 0.5$, respectively.

In the following, we restrict our attention to the interval $N \in [0, 1/2]$, as $\mathcal{A}_{\gamma,N}$ and $\mathcal{A}_{\gamma,1-N}$ are unitarily equivalent and hence their channel coherent informations (and quantum capacities) coincide [KSW19]. In the optimization procedure we consider the values $N \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and identify intervals of γ in which

Table 1. Table of the best neural network codes for the GADC $\mathcal{A}_{\gamma,N}$ with $(\gamma, N) = (0.440\,35, 0.1)$ and $k = 3, 4, 5$ channel copies. Only the non-zero amplitudes $\psi(s^n)$ indexed by the basis string s^n (with $n = 2k$) are shown (see equation (15)). The architecture used for the neural network codes is a feed-forward net with four hidden layers of width $2k$ each, activation functions cos and $3 \times \tanh$, and a Cartesian output layer (see section 4.1).

$|\nu_k\rangle$	$s^n \ (A^k|R)$	$\psi(s^n)$	$\frac{1}{k}Q^{(1)}(\nu_k, \mathcal{A}_{\gamma,N}^{\otimes k})$
$k = 3$	000|000	$-0.3934 + 0.2231i$	$5.7598 \cdot 10^{-4}$
	000|110	$-0.3136 + 0.2501i$	
	001|111	$-0.3956 + 0.2345i$	
	010|111	$-0.3956 + 0.2346i$	
	100|111	$-0.3955 + 0.2348i$	
$k = 4$	0101|1110	$+0.3482 - 0.2537i$	$1.2683 \cdot 10^{-3}$
	1010|1110	$+0.3354 - 0.2723i$	
	1111|0001	$+0.3986 - 0.3920i$	
	1111|1000	$+0.3980 - 0.3959i$	
$k = 5$	01010|00011	$+0.3010 + 0.1666i$	$9.1537 \cdot 10^{-4}$
	10101|00011	$+0.4389 + 0.1233i$	
	11111|10110	$+0.5660 + 0.0816i$	
	11111|11101	$+0.5844 + 0.0725i$	

weighted repetition codes are superadditive, that is, they yield a higher coherent information than the optimal single-letter coherent information. For $k = 3, 4, 5$ copies of $\mathcal{A}_{\gamma,N}$, we search for neural network codes using a feed-forward architecture as described in figure 1 with four hidden layers of width $2k$ each. We choose cos as the activation function in the first layer, the hyperbolic tangent function tanh as the activation function in the subsequent layers, and a Cartesian output layer (see equation (8)). In contrast to the more common gradient-based optimization techniques in machine learning, we choose to optimize the neural network parameters using stochastic *gradient-free* techniques. In particular, we use particle swarm optimization (PSO) algorithm followed by pattern search. We motivate our choice to use these algorithms in appendix G, which also contains high-level explanations of these techniques.

For all values $N \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ we find neural network codes outperforming the weighted repetition codes equation (10), as shown in figure 3. For each $N \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, the codes in figure 3 are obtained by first carrying out our optimization technique for a particular value of γ close to the threshold of the best weighted repetition code. We then plot the best neural network code found in this manner for the entire interval γ where superadditivity occurs. As a benchmark, we evaluate weighted repetition codes for up to $k = 16$ channel copies using the formula derived in appendix A; the codes ϕ_k for $1 \leqslant k \leqslant 5$ perform best and are shown in figure 3 for comparison.
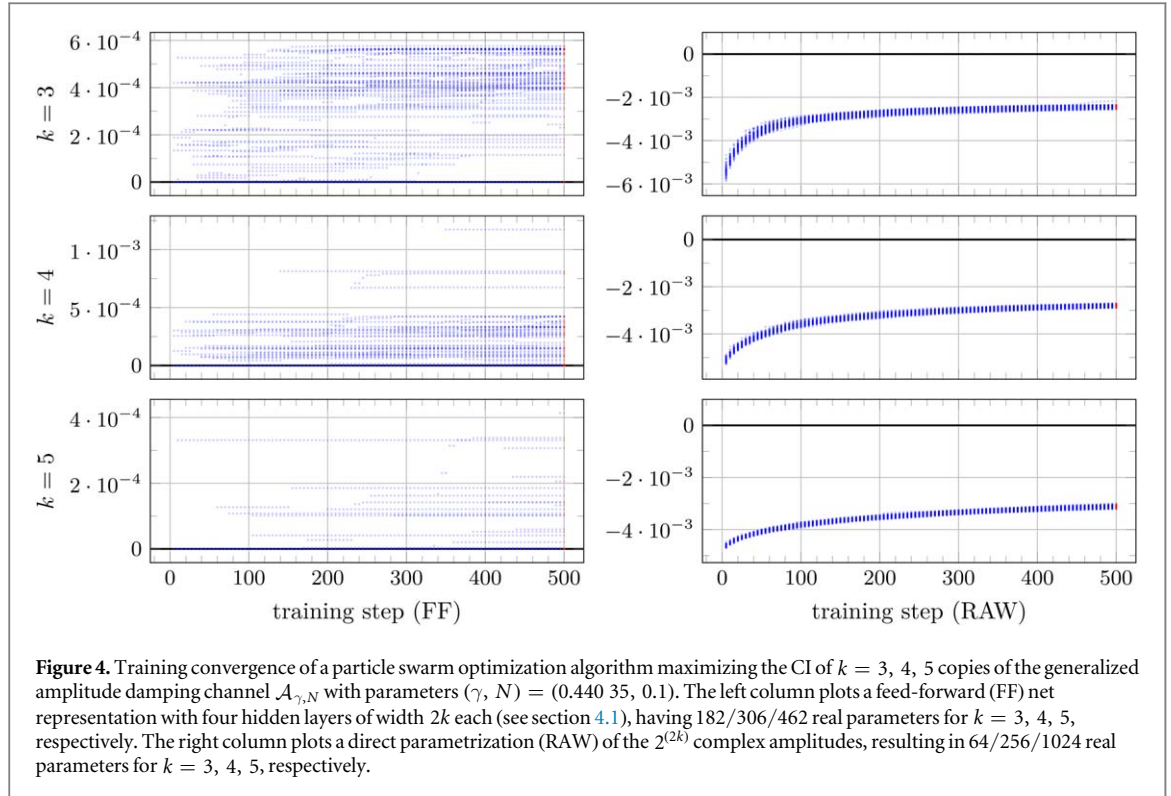
We focus here on the neural network codes found for the values $(\gamma, N) = (0.440\,35, 0.1)$ and $k = 3, 4, 5$ copies of $\mathcal{A}_{\gamma,N}$, and note that the neural network codes for the other values of (γ, N) are collected in appendix A. In table 1 we list the best codes (as plotted in figure 3) for each blocklength together with their coherent information. In figure 4 we plot the convergence of the PSO algorithm for $(\gamma, N) = (0.440\,35, 0.1)$ and $k = 3, 4, 5$ (FF), and compare its performance to a direct parametrization (RAW) of the 2^{2k} complex amplitudes in the quantum code $|\psi_n\rangle$, again optimized using PSO. Evidently, using comparable optimization parameters the raw ansatz is not able to find even trivial product codes with coherent information equal to zero. Note also that for $(\gamma, N) = (0.440\,35, 0.1)$ the weighted repetition codes in equation (10) do not yield positive coherent information up to at least $k = 16$. Hence, the neural network codes increase the *threshold* of the GADC substantially, as seen in figure 3. The threshold of a parametrized family of quantum channels is defined as the boundary of the region in which the channel has positive quantum capacity.

4.2. Dephrasure channel
The neural network ansatz is also able to find new quantum codes for the *dephrasure channel* that was introduced recently in [LLS18a]. It is defined in terms of probabilities $p, q \in [0, 1]$ as

$$\mathcal{N}_{p,q}(\rho) = (1 - q)((1 - p)\rho + pZ\rho Z) + q\mathrm{tr}(\rho)|e\rangle\langle e|, \tag{11}$$

where $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ is the Pauli Z-operator, and $|e\rangle$ is an erasure flag that is orthogonal to the input space. The name 'dephrasure' is derived from the fact that $\mathcal{N}_{p,q}$ first dephases an input state in the Z-basis with probability p, and then erases it with probability q. Despite the fact that both dephasing and erasure noise are well-understood in terms of quantum information transmission, the dephrasure channel—a concatenation of

Figure 4. Training convergence of a particle swarm optimization algorithm maximizing the CI of $k = 3, 4, 5$ copies of the generalized amplitude damping channel $\mathcal{A}_{\gamma, N}$ with parameters $(\gamma, N) = (0.440\,35, 0.1)$. The left column plots a feed-forward (FF) net representation with four hidden layers of width $2k$ each (see section 4.1), having 182/306/462 real parameters for $k = 3, 4, 5$, respectively. The right column plots a direct parametrization (RAW) of the $2^{(2k)}$ complex amplitudes, resulting in 64/256/1024 real parameters for $k = 3, 4, 5$, respectively.

the two—exhibits superadditivity of coherent information for as little as two uses of the channel [LLS18a]. As a result, the quantum capacity of the dephrasure channel is unknown for a large region in the parameter space.

As for the GADC in the previous section, superadditivity of coherent information for the dephrasure channel is again achieved by weighted repetition codes ϕ_k^λ as defined in equation (10). A compact formula for the coherent information $\frac{1}{k} Q^{(1)}(\phi_k^\lambda, \mathcal{N}_{p,q}^{\otimes k})$ of these codes was derived in [LLS18a], and we state it in appendix B. Similar to the GADC in section 4.1, we note that the optimal single-letter coherent information $Q^{(1)}(\mathcal{N}_{p,q})$ for the dephrasure channel is achieved by ϕ_1^λ for some $\lambda \in [0, 1]$ [LLS18a]. We show in this section that the neural network state ansatz finds new quantum codes demonstrating even larger superadditivity of coherent information for the dephrasure channel.

In the following, we focus our attention to the values $q \in \{0.1, 0.2, 0.3, 0.4\}$ of the erasure probability; for each q, we then investigate values of the dephasing probability p for which weighted repetition codes achieve superadditivity. Since the dephrasure channel maps a qubit to a qutrit, optimizing its coherent information is computationally more costly than for the GADC, which forces us to restrict our attention to $k = 2, 3, 4$ copies of $\mathcal{N}_{p,q}$ (we refer to section 7 for a discussion of these numerical limitations). We again use a feed-forward network as described in figure 1 with four hidden layers of width $2k$ each and cos as the activation function in the first layer. However, in contrast to section 4.1 we use ReLU as the activation function in the remaining layers, and an exponential output layer corresponding to a polar parametrization instead of a Cartesian one. We found these choices to perform significantly better for the dephrasure channel. As in section 4.1, the neural network parameters were optimized using the PSO algorithm followed by pattern search (see appendix G).

For all values $q \in \{0.1, 0.2, 0.3, 0.4\}$ we find neural network codes outperforming the weighted repetition codes equation (10), as shown in figure 5. For each $q \in \{0.1, 0.2, 0.3, 0.4\}$ and $k = 2, 3, 4$, the codes in figure 5 are obtained by first carrying out our optimization technique for a particular value of p close to the threshold of the best weighted repetition code. We then plot the best neural network code (labeled ν_k for $k = 2, 3, 4$) found in this manner for an interval of p where superadditivity occurs. We also individually optimized the coefficients of the basis strings s^n with non-zero weight across the shown interval of p, yielding even better codes ν_k^*. Curiously, such an additional optimization over coefficients gave no improvement for the neural network codes found for the GADC in section 4.1. In contrast, for fixed q there is an evident interplay between the dephasing probability p in the dephrasure channel $\mathcal{N}_{p,q}$ and the coefficients of the neural network codes ν_k, as evident from figure 5. As a benchmark, we evaluated weighted repetition codes ϕ_k for up to $k = 10$ channel copies using the formula in appendix B; the maximum $\max_k \phi_k$ over these codes for $1 \leqslant k \leqslant 10$ is shown in figure 5 for comparison, along with the optimal single-letter code ϕ_1.

We focus in the following on the neural network codes found for the values $(p, q) = (0.08, 0.4)$ and $k = 2, 3, 4$; the other neural network codes are listed in appendix B. In table 2 we list the best codes (as plotted in

Figure 5. Overview of quantum codes for the dephrasure channel $\mathcal{N}_{p,q}$ comparing the neural network codes ν_k (solid orange, magenta, and red lines) for $k = 2, 3, 4$ to the optimal single-letter code ϕ_1 (gray dashed line) and the maximum over all weighted repetition codes ϕ_k (black dashed line) for $2 \leqslant k \leqslant 10$ defined in equation (10). We also plot the neural network codes ν_k^* with optimized parameters over the shown interval (dashed–dotted lines), and the best code χ_3 on three channel qubits found with a direct parametrization of the quantum state amplitudes (blue line). For each $q \in \{0.1, 0.2, 0.3, 0.4\}$, we plot the interval of p in which superadditivity occurs. The neural network codes ν_k are listed in tables 2, B1, B2, B3 for $q = 0.4, 0.3, 0.2, 0.1$, respectively.
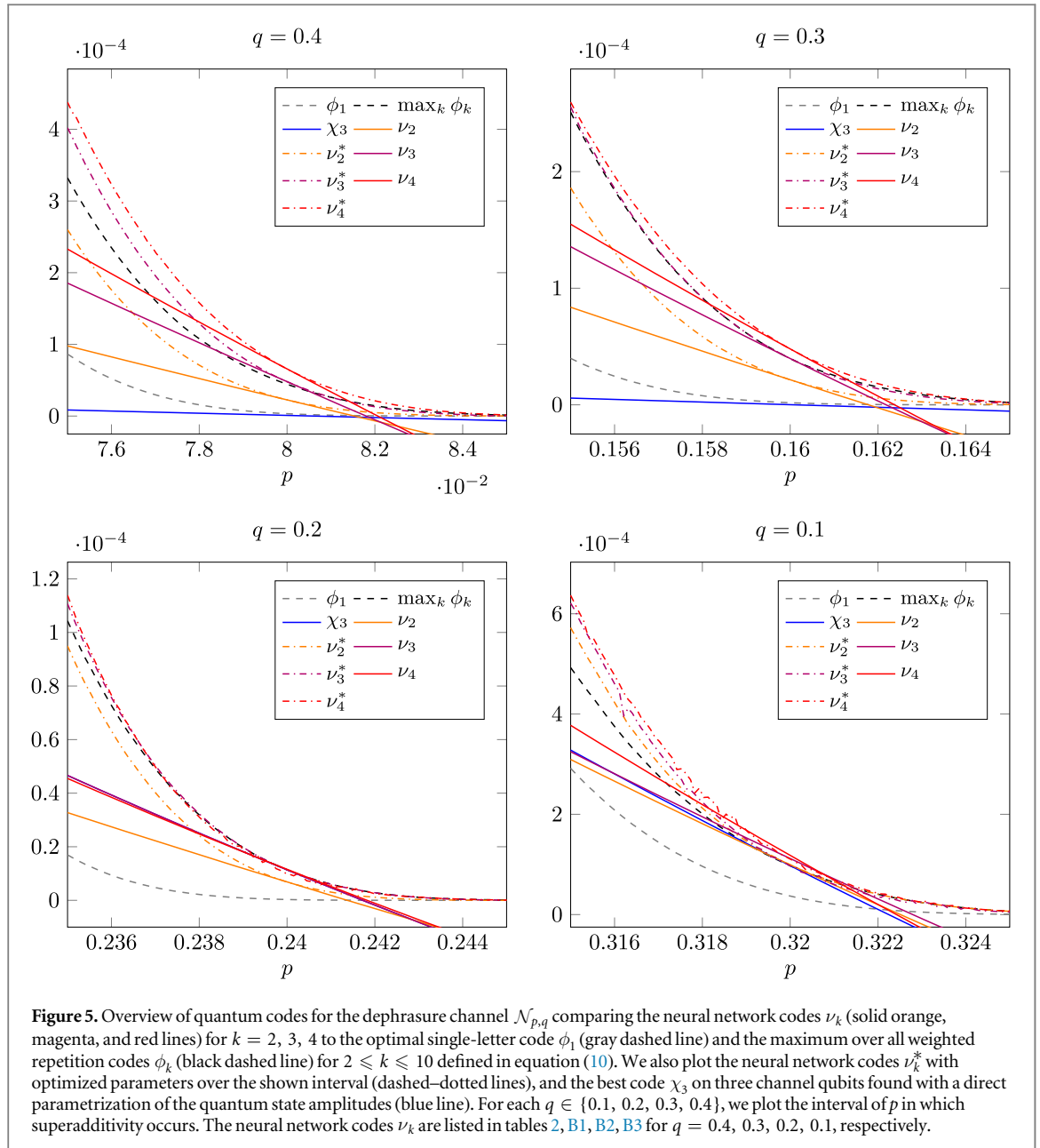
figure 5) for each blocklength together with their coherent information. In figure 6 we plot the convergence of the PSO algorithm for $(p, q) = (0.08, 0.4)$ and $k = 2, 3, 4$ (FF), and compare its performance to a direct parametrization (RAW) of the 2^{2k} complex amplitudes in the quantum code $|\psi_n\rangle$, again optimized using PSO. Similar to the GADC in section 4.1, the raw ansatz is not able to find codes with coherent information rates as high as the neural network codes. However, in contrast to the GADC the raw ansatz is indeed able to find superadditive quantum codes. For $k = 2$, these codes found using the raw ansatz are optimal (as already observed in [LLS18a]), while for $k = 3, 4$ they are clearly outperformed by our neural network codes. Another observation of [LLS18a] is that the dephasing part of $\mathcal{N}_{p,q}$ suggests a Schmidt ansatz for quantum codes, a neural network state version of which is discussed in equation (16) in section 5. However, in the high-noise regime investigated above, this Schmidt ansatz did not yield codes performing as well as the codes ν_k resp. ν_k^*.

5. Representing the best known codes for the depolarizing channel

The *depolarizing channel* is used as a model to describe qubit decoherence in a noisy environment. For a qubit in a state described by the density operator ρ, and for a real parameter $p \in [0, 4/3]$, the action of the channel is given by

Figure 6. Training convergence of a particle swarm optimization algorithm maximizing the CI of $k = 2$, 3, 4 copies of the dephrasure channel $\mathcal{N}_{p,q}$ with parameters $(p, q) = (0.08, 0.4)$. The left column plots a feed-forward (FF) net representation with four hidden layers of width $2k$ each (see section 4.2), having 90/182/306 real parameters for $k = 2$, 3, 4, respectively. The right column plots a direct parametrization (RAW) of the $2^{(2k)}$ complex amplitudes, resulting in 32/128/512 real parameters for $k = 2$, 3, 4, respectively. While the two parametrization find equivalent codes for $k = 2$, the feed-forward net representation finds strictly better codes for $k = 3$, 4 than the raw parametrization.
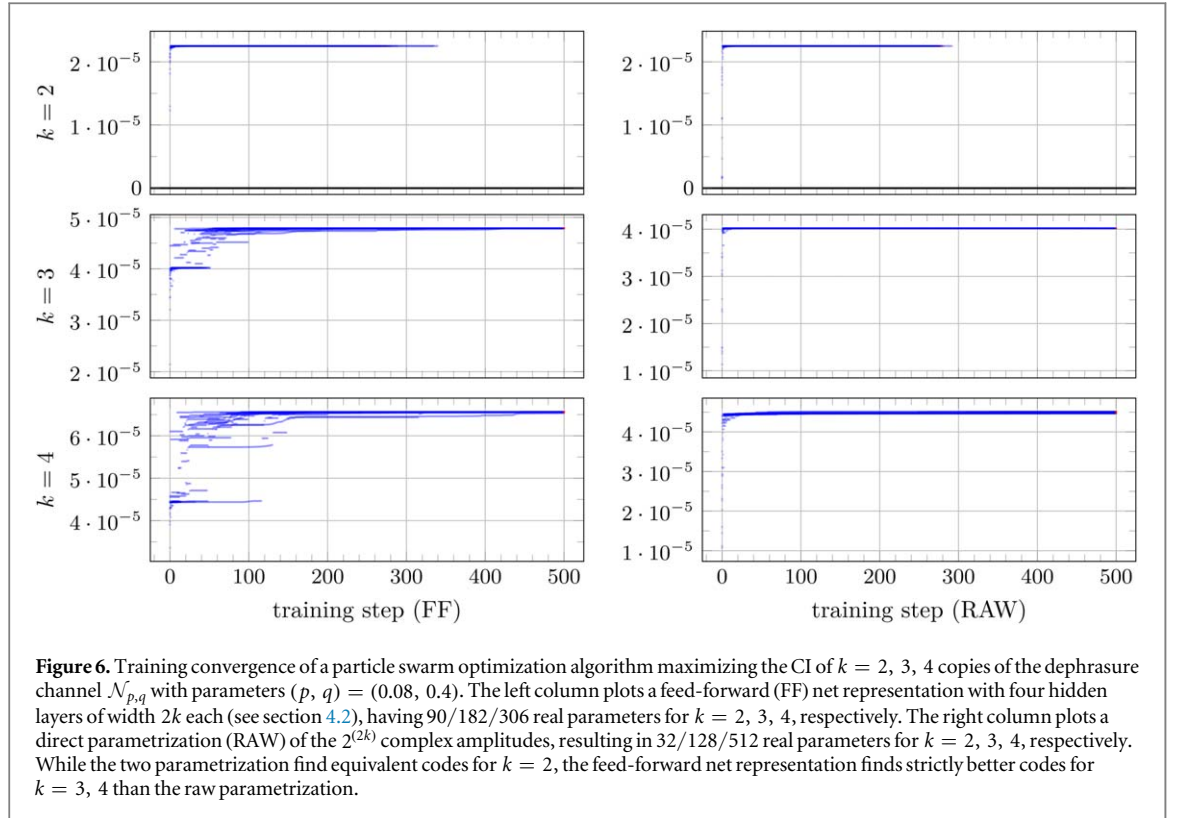
Table 2. Table of the best neural network codes for the dephrasure channel $\mathcal{N}_{p,q}$ with $(p, q) = (0.08, 0.4)$ and $k = 2$, 3, 4 channel copies. Only the non-zero amplitudes $\psi(s^n)$ indexed by the basis string s^n (with $n = 2k$) are shown (see equation (15)). The architecture used for the neural network codes is a feed-forward net with four hidden layers of width $2k$ each, activation functions $\cos$ and $3 \times \text{ReLU}$, and a Polar output layer (see section 4.2).

$|\nu_k\rangle$	$s^n \, (A^k|R)$	$\psi(s^n)$	$\frac{1}{k} Q^{(1)}(\nu_k, \mathcal{N}_{p,q}^{\otimes k})$
$k = 2$	00|00	$-0.250\,4 - 0.435\,2i$	$2.250\,2 \cdot 10^{-5}$
	00|01	$-0.694\,1 + 0.514\,2i$	
	11|01	$+0.037\,4 + 0.017\,1i$	
	11|11	$-0.000\,1 - 0.000\,1i$	
$k = 3$	000|011	$-0.030\,4 + 0.046\,5i$	$4.788\,1 \cdot 10^{-5}$
	001|011	$-0.046\,5 + 0.040\,8i$	
	111|000	$-0.695\,4 + 0.713\,8i$	
$k = 4$	0101|1000	$+0.002\,2 - 0.003\,1i$	$6.569\,9 \cdot 10^{-5}$
	0111|1000	$-0.916\,9 - 0.368\,6i$	
	0111|1110	$+0.005\,4 + 0.010\,2i$	
	1000|0111	$-0.093\,2 + 0.004\,9i$	
	1001|0111	$+0.000\,1 + 0.000\,0i$	
	1010|0111	$-0.034\,1 - 0.115\,6i$	

$$\mathcal{D}_p(\rho) := (1 - p)\rho + p\,\text{tr}(\rho)(|0\rangle\langle 0| + |1\rangle\langle 1|)/2, \tag{12}$$

i.e. the original state ρ is replaced by the maximally mixed state with 'probability' p (for $p \leqslant 1$); in other words, if on the Bloch sphere ρ has spin polarization vector $\vec{x}$, the channel $\mathcal{D}_p$ shrinks $\vec{x}$ by a factor $1 - p$.

For the depolarizing channel, the single-letter channel coherent information $Q^{(1)}(\mathcal{D}_p)$ is maximized by a Bell state $\frac{1}{\sqrt{2}}(|0\rangle_R|0\rangle_A + |1\rangle_R|1\rangle_A)$, and evaluates to [Wil16]

$$Q^{(1)}(\mathcal{D}_p) = 1 + \left(1 - \frac{3p}{4}\right)\log\left(1 - \frac{3p}{4}\right) + \frac{3p}{4}\log\frac{p}{4}. \tag{13}$$

$Q^{(1)}(\mathcal{D}_p)$ remains positive up to the threshold at $p = 0.252\,38$ (the *threshold* is defined as the highest p for which $Q^{(1)}(\mathcal{D}_p) > 0$). The next highest thresholds are achieved for $k = 3$ and 5 channel copies and a k-repetition code

$$|\phi_k\rangle = \frac{1}{\sqrt{2}}(|0\rangle_R |0\rangle_A^{\otimes k} + |1\rangle_R |1\rangle_A^{\otimes k}), \tag{14}$$

for which the channel coherent information $Q^{(1)}(\phi_k, \mathcal{D}_p^{\otimes k})$ reaches zero at $p = 0.253\,50$ and $p = 0.253\,80$, respectively. Both in terms of the rate and the threshold, these repetition codes are the best known codes up to 9 channel copies, which is discussed in more detail in appendix C.1.

We show in the following that a variational neural network ansatz achieves these codes for the depolarizing channel. We also contrast the various architectures (RBM, feed-forward, and their Schmidt variants) on an empirical level. To compute the amplitude function $\psi(i^n)$ in the tensor basis expansion

$$|\psi_n\rangle = \frac{1}{C} \sum_{i^n \in \{0,1\}^n} \psi(i^n)|i^n\rangle \in (\mathbb{C}^2)^{\otimes n}, \tag{15}$$

we use both an RBM architecture as well as an FF architecture with a cos activation function in the first hidden layer, and ReLU in two subsequent hidden layers. This setup, which has been shown to perform well in the context of representing quantum states of local Hamiltonians [CL18], clearly outperformed a ReLU-only architecture in our numerical investigations of the GADC and the dephrasure channel in section 4.

Furthermore, we propose a Schmidt-Ansatz similar to equation (15) given for $2l$ qubits by

$$|\psi_{2l}\rangle = \frac{1}{C} \sum_{i^l \in \{0,1\}^l} \psi(i^l)|i^l\rangle_R |i^l\rangle_A. \tag{16}$$

This approach greatly reduces the number of degrees of freedom required to parametrize $|\psi_{2l}\rangle$, but enforces the environment R to have the same dimension as the system A. Note that this may introduce redundancy, as e.g. a repetition code ordinarily only requires a single purifying qubit. The ansatz in equation (16) furthermore introduces a choice of basis for the channel input qubits, rendering it less general than the ansatz in equation (15).
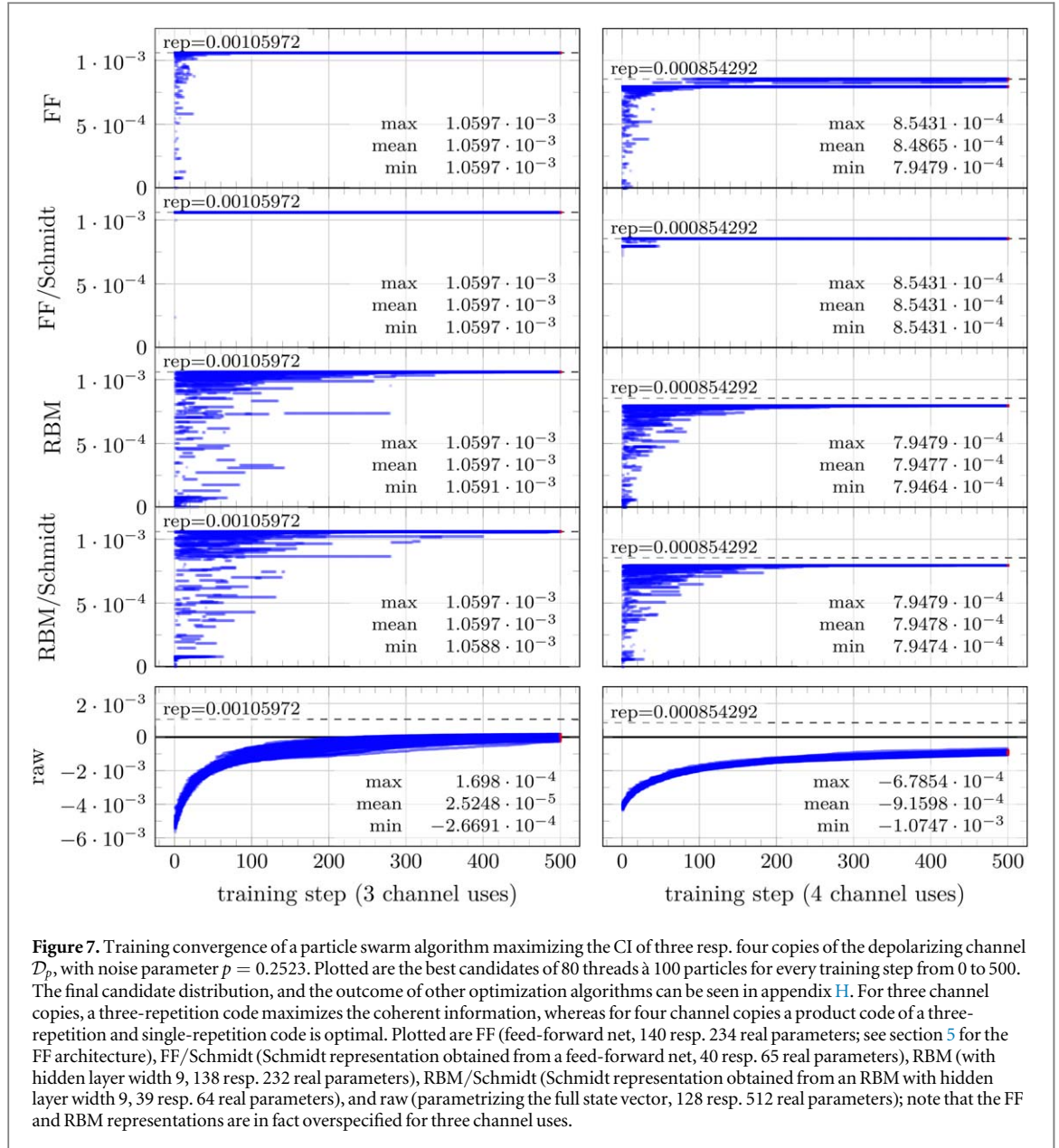
Using an explicit construction, we show that both FF and RBM architectures can efficiently represent products of repetition codes (which are discussed in appendix C.2): given k repetition codes on $n_1, \ldots, n_k$ qubits, respectively, an RBM with $\prod_i n_i$ visible units and k hidden nodes can represent the corresponding state amplitudes, and a FF net with first cos and second ReLU hidden layer width k, and a single final ReLU node suffices.

Empirically, we contrast FF, RBM and their corresponding Schmidt variants as a variational ansatz ψ_n (with $n = 2k$) to maximize $Q^{(1)}(\psi_n, \mathcal{D}_p^{\otimes k})$; the FF architecture consists of three hidden layers of width $n = 2k$ with cos-ReLU-ReLU for the activation functions and a Cartesian output layer. In comparison with a full state vector on n qubits with 2×2^n real parameters, we can see a significant improvement in convergence speed (see figure 7), both in the case that the best-known code is a single repetition code for three channel uses, or a three times one product repetition code (see appendix C.1 for an explanation of this terminology). For both FF and RBM architectures, the Schmidt ansatz equation (16) surpasses the standard parametrization equation (15), which is likely due to the significantly-reduced parameter count. FF networks further outperform RBM architectures with comparable parameter counts on three and four channel uses of a depolarizing channel, which we verified with various global derivative-free optimization techniques (see appendix G for an overview) to reduce the likelihood of a systematic bias in our numerical findings. The numerical data for these findings is collected in appendix H. We also note that a DBM ansatz as described in section 3.2 offered no advantage over an RBM ansatz, neither in terms of representability nor convergence speed.

6. Representing AME states

AME states are n-partite states having maximal correlation across any bipartition of the n parties into equal halves. These states are certain examples of quantum error-correcting codes, whose intricate multipartite entanglement structure mediates correlations between different subsets of the constituent systems.

AME states can be used as a resource for multi-user information-theoretic tasks such as open-destination teleporation, secret sharing or entanglement swapping that require maximal entanglement across different choices of bipartitions [HC13, Hel+12]. In a holographic context, where AME states are referred to as *perfect tensors*, they provide examples of holographic error-correcting codes [LS15, Pas+15, Li+17]. More generally, an arbitrary AME state on n qudits of local dimension d can be interpreted as a $\left(\left(n, 1, \left\lfloor \frac{n}{2} \right\rfloor + 1\right)\right)_d$ quantum error-correcting code, i.e. a code of distance $\left\lfloor \frac{n}{2} \right\rfloor + 1$ encoding a 1-dimensional subspace in n physical qudits [Sco04].

Figure 7. Training convergence of a particle swarm algorithm maximizing the CI of three resp. four copies of the depolarizing channel $\mathcal{D}_p$, with noise parameter $p = 0.2523$. Plotted are the best candidates of 80 threads à 100 particles for every training step from 0 to 500. The final candidate distribution, and the outcome of other optimization algorithms can be seen in appendix H. For three channel copies, a three-repetition code maximizes the coherent information, whereas for four channel copies a product code of a three-repetition and single-repetition code is optimal. Plotted are FF (feed-forward net, 140 resp. 234 real parameters; see section 5 for the FF architecture), FF/Schmidt (Schmidt representation obtained from a feed-forward net, 40 resp. 65 real parameters), RBM (with hidden layer width 9, 138 resp. 232 real parameters), RBM/Schmidt (Schmidt representation obtained from an RBM with hidden layer width 9, 39 resp. 64 real parameters), and raw (parametrizing the full state vector, 128 resp. 512 real parameters); note that the FF and RBM representations are in fact overspecified for three channel uses.

To define AME states in a precise way, we consider a pure state $|\psi_{n,d}\rangle \in (\mathbb{C}^d)^{\otimes n}$ on n qudits of local dimension d. For a subset $\mathcal{S} \subset [n] := \{1, \ldots, n\}$ of the n qudits we denote by $\rho_{\mathcal{S}} = \mathrm{tr}_{\mathcal{S}^c} \psi_{n,d}$ the marginal of $\psi_{n,d}$ on $\mathcal{S}$. Then $\psi_{n,d}$ is AME if $\rho_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} I_{\mathcal{S}}$ for every $\mathcal{S} \subset [n]$ with $|\mathcal{S}| = \lfloor \frac{n}{2} \rfloor$. We use the notation AME(n, d) for an AME state on n qudits of local dimension d.

Since an AME state is maximally entangled across all possible bipartitions into equal halves, monogamy of entanglement [CKW00] puts an obstruction on their existence. Furthermore, the fact that AME states are particular quantum error-correcting codes yields additional constraints via weight enumerator theory [SL97, Rai98]. Consequently, AME states do not exist for all (n, d) appendix D. For example, it is known that there is no AME$(4, 2)$ state [HS00]. On the other hand, an example of an AME$(4, 3)$ state is $|\Omega_{4,3}\rangle = \frac{1}{3}\sum_{i,j=0,1,2} |i\rangle |j\rangle |i + j(3)\rangle |i + 2j(3)\rangle$, where $k(d) \equiv k \bmod d$.

The property of $\psi_{n,d}$ being AME is related to the *linear entropy* $S_L(\rho_{\mathcal{S}}) = \frac{d^m}{d^m - 1}(1 - \mathrm{tr}(\rho_{\mathcal{S}}^2))$ of the marginals $\rho_{\mathcal{S}}$ for $\mathcal{S} \subset [n]$ with $|\mathcal{S}| = \lfloor \frac{n}{2} \rfloor$. Defining for $m = 1, \ldots, \lfloor \frac{n}{2} \rfloor$ the average linear entropy

$$Q_m(\psi_{n,d}) := \binom{n}{m}^{-1} \sum_{\mathcal{S} \subset [n]: |\mathcal{S}| = m} S_L(\rho_{\mathcal{S}}), \tag{17}$$

a pure state $\psi_{n,d}$ is AME if and only if $Q_{\lfloor \frac{n}{2} \rfloor}(\psi_{n,d}) = 1$ [Sco04]. Hence, to search for AME(n, d)-states $\psi_{n,d}$, we can use equation (17) with $m = \lfloor \frac{n}{2} \rfloor$ as the objective function and optimize the parameters in an ansatz for $\psi_{n,d}$ such that $Q_{\lfloor \frac{n}{2} \rfloor}(\psi_{n,d}) \approx 1$. As before, we use a neural network state ansatz for $\psi_{n,d}$ based on the following
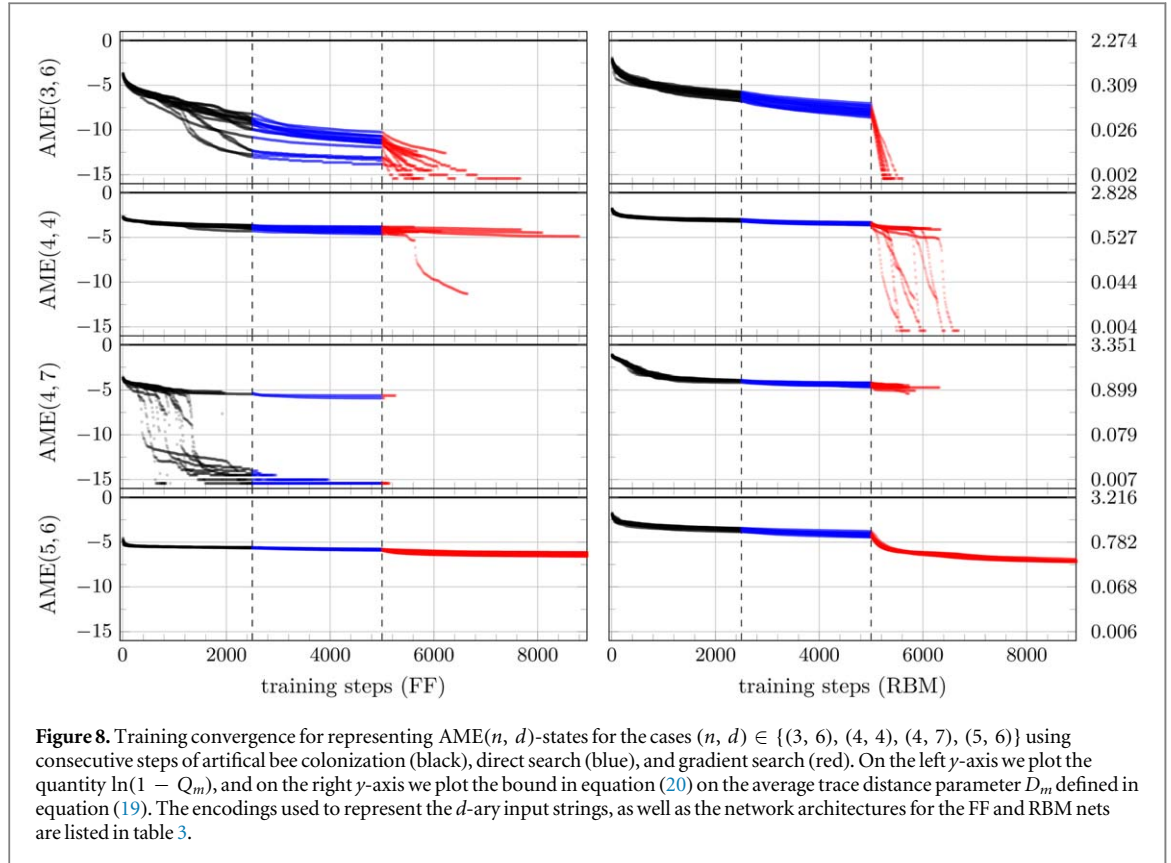
Figure 8. Training convergence for representing AME(n, d)-states for the cases (n, d) $\in$ {(3, 6), (4, 4), (4, 7), (5, 6)} using consecutive steps of artifical bee colonization (black), direct search (blue), and gradient search (red). On the left y-axis we plot the quantity $\ln(1 - Q_m)$, and on the right y-axis we plot the bound in equation (20) on the average trace distance parameter D_m defined in equation (19). The encodings used to represent the d-ary input strings, as well as the network architectures for the FF and RBM nets are listed in table 3.

decomposition with respect to a given basis $\{|i\rangle\}_{i=0}^{d-1}$:

$$|\psi_{n,d}\rangle = \frac{1}{C} \sum_{i^n \in [d]_0^n} \psi(i^n)|i^n\rangle, \tag{18}$$

where as before C is a normalization constant, and we use the notation $[d]_0 = \{0, \ldots, d - 1\}$. The amplitude function $\psi(i^n)$ is again computed by a neural network; since this is now a function from the set of all d-ary strings of length n into $\mathbb{C}$, there are multiple options how to encode i^n as the input to a neural network. We discuss these options in detail in appendix E.

We demonstrate in figure 8 that parametrizing $\psi_{n,d}$ with a neural network state ansatz yields AME(n, d)-states for the pairs (n, d) = (3, 6), (4, 4), (4, 7), and (5, 6). For the numerical optimization, we use the artificial bee colonization (ABC) algorithm, followed by pattern search and a final round of gradient search (see appendix G). These choices of parameters are only exemplary, and the neural network state ansatz is capable of representing AME(n, d)-states also for other pairs (n, d) such as (3, 3), (4, 3), and (4, 5). In the last three cases, the convergence is remarkably fast and only takes a few iterations in optimization algorithms such as ABC or PSO to reach a value of $Q_{\lfloor \frac{n}{2} \rfloor}$ sufficiently close to 1.

To assess our numerical results, we introduce an 'average trace distance' parameter

$$D_m(\psi_{n,d}) := \binom{n}{m}^{-1} \sum_{\mathcal{S} \subset [n]:|\mathcal{S}|=m} \|\rho_{\mathcal{S}} - \pi_{\mathcal{S}}\|_1, \tag{19}$$

where $\pi_{\mathcal{S}} := \frac{1}{|\mathcal{S}|} I_{\mathcal{S}}$ denotes the completely mixed state, and $\|X\|_1 = \operatorname{tr}\sqrt{X^\dagger X}$ is the trace norm of an operator X. The parameter $D_m(\psi_{n,d})$ measures the average trace distance of the marginals of a state $\psi_{n,d}$ on m subsystems to the completely mixed state. Clearly, $D_{\lfloor \frac{n}{2} \rfloor}(\psi_{n,d}) = 0$ if and only if $\psi_{n,d}$ is AME. We prove in appendix D that

$$D_m(\psi_{n,d}) \leqslant \sqrt{2 \log[d^m - (d^m - 1)Q_m(\psi_{n,d})]}. \tag{20}$$

This bound allows us to relate a value of Q_m to how close (on average) in trace distance a state is to being AME (see figure 8).

7. Discussion

In this work, we have shown that quantum codes for noisy quantum communication and certain quantum error-correcting codes can be modeled efficiently with various neural network representations. In particular, we investigated

Table 3. Encodings and network architectures used to represent AME(n, d)-states as plotted in figure 8. For the FF nets, the hidden layers are denoted by $(M_i)_i$, where M_i is the width of the ith hidden layer. We always use cos in the first hidden layer, ReLU in the following hidden layers, and a Cartesian output layer. For the RBM net, we list the width of the single hidden layer. The encodings are defined in table C2.

n	d	Architecture	Encoding	Hidden layer
3	6	FF	binary	(12, 12, 12)
		RBM	binary	12
4	4	FF	binary	(8, 8, 8)
		RBM	binary	8
4	7	FF	scaled	(4, 4, 4)
		RBM	scaled	4
5	6	FF	binary	(15, 15, 15, 15)
		RBM	binary	20

quantum codes that yield high coherent information for the GADC, the dephrasure channel, and the depolarizing channel. For the GADC and the dephrasure channel, the neural network ansatz finds codes that outperform the best known codes found with traditional numerical methods. For $k \leqslant 6$ of the depolarizing channel, we analyzed the representative power of neural network states with regards to the best known codes, repetition codes, and benchmarked how well they can be trained using a variety of global optimization algorithms. Finally, we demonstrated how neural network states can represent AME states on n qudits of local dimension d for an array of pairs (n, d).

An interesting question is, of course, whether a neural network state ansatz can be used to find better quantum codes for the depolarizing channel in the high noise regime: either in terms of a higher rate than, say, the 5-repetition code right below the noise threshold, or in terms of increasing the noise threshold itself. Our results indicate that in order to find such codes outperforming the repetition codes (or products thereof), one ought to increase the number of channel copies beyond 5, resulting in code states on 10 or more input qubits. While the (polynomial) scaling of the neural network ansatz in the number of input qubits is favorable, the calculation of the coherent information is the bottleneck here: The computation for a code on k qubits requires diagonalizing a dense $4^k \times 4^k$ matrix, which scales exponentially in runtime with the number of qubits. Due to these computational limitations, evaluating the coherent information for $k \gtrsim 7$ channel uses is thus an infeasible undertaking, and we would need to find an alternative approach—e.g. by exploiting symmetry considerations, or an approximate cost function that is faster to compute (see e.g. [WBS14], with the added difficulty that the coherent information is the difference between two entropies).

Furthermore, it could be possible that better quantum codes lie in maxima of measure almost zero, while the repetition code maxima dominate the potential landscape, making it difficult to find codes that surpass repetition codes. In fact, in all our simulations for $k \leqslant 6$ copies of the depolarizing channel, the variational NN ansatz converges to product repetition codes. Our results might be seen as indication that, among the states that can be represented using a neural network, repetition codes are in fact optimal for $k \leqslant 6$ copies of the depolarizing channel. We note that our techniques of finding quantum codes using neural network states can also be applied to other channels such as generalized Pauli channels, which includes the depolarizing channel. A thorough investigation of other channels in this class, such as the BB84 channel [BB84], is the subject of ongoing work.

We also applied our ansatz to search for AME(n, d)-states for values of (n, d) for which it is unknown yet whether these states exist. The smallest-dimensional instances of these cases are $(4, 6)$ and $(7, 4)$ (see appendix D). For $(n, d) = (4, 6)$ the best value we obtained was $Q_2(\psi_{4,6}) \approx 0.995\,6$, which translates via equation (20) to a bound on the average trace distance parameter of $D_2(\psi_{4,6}) \lesssim 0.642\,9$. The state $\psi_{n,d}$ achieving these values is an RBM state with binary encoding and a hidden layer width of $M = 12$. For $(n, d) = (7, 4)$, we obtained $Q_3(\psi_{7,4}) \approx 0.996\,2$, corresponding to $D_3(\psi_{7,4}) \lesssim 0.787\,0$, achieved by an FF state with binary encoding and hidden layers $(14, 14, 14)$ with activation functions cos-ReLU-ReLU. These results suggest that, assuming AME states do exist in these cases, one has to tweak the neural network ansatz or the numerical methods, or both, in order to obtain numerical instances of AME states.

Acknowledgments

Appendix A. Codes for the GADC

In this section we provide an overview of the quantum codes for the GADC defined in equation (9) found using the neural network state ansatz. To benchmark these neural network quantum codes we use weighted repetition codes

$$|\phi_k^\lambda\rangle := \sqrt{\lambda}|0\rangle_R \otimes |0\rangle_A^{\otimes k} + \sqrt{1-\lambda}|1\rangle_R \otimes |1\rangle_A^{\otimes k}, \tag{21}$$

whose simple structure allows for an efficient computation of the coherent information $Q^{(1)}(\phi_k^\lambda, \mathcal{A}_{\gamma,N}^{\otimes k})$. In the following, we first carry out this calculation, and then present the optimal neural network codes that we found for the GADC.

A.1. Formula for the coherent information of repetition codes

We first determine the action of the GADC $\mathcal{A}_{\gamma,N} = \sum_{i=1}^4 A_i \rho A_i^\dagger$ with A_i as defined in equation (9) on a single qubit:

$$\mathcal{A}_{\gamma,N}(|0\rangle\langle0|) = (1 - \gamma N)|0\rangle\langle0| + \gamma N|1\rangle\langle1|$$
$$\mathcal{A}_{\gamma,N}(|1\rangle\langle1|) = (\gamma - \gamma N)|0\rangle\langle0| + (1 - \gamma + \gamma N)|1\rangle\langle1|$$
$$\mathcal{A}_{\gamma,N}(|0\rangle\langle1|) = \sqrt{1-\gamma}|0\rangle\langle1|$$
$$\mathcal{A}_{\gamma,N}(|1\rangle\langle0|) = \sqrt{1-\gamma}|1\rangle\langle0|. \tag{22}$$

Setting $\sigma_{RB^k} = (\mathrm{id}_R \otimes \mathcal{A}_{\gamma,N}^{\otimes k})(\phi_k^\lambda)$, we have

$$\sigma_{B^k} = \lambda \mathcal{A}_{\gamma,N}(|0\rangle\langle0|)^{\otimes k} + (1-\lambda)\mathcal{A}_{\gamma,N}(|1\rangle\langle1|)^{\otimes k}, \tag{23}$$

which is a diagonal operator with eigenvalues

$$r_m := \lambda(1-\gamma N)^{k-m}(\gamma N)^m + (1-\lambda)(\gamma - \gamma N)^{k-m}(1 - \gamma + \gamma N)^m \tag{24}$$

with multiplicity $\begin{pmatrix} k \\ m \end{pmatrix}$ for $m = 0, \ldots, k$. Hence,

$$S(B^k)_\sigma = -\sum_{m=0}^k \begin{pmatrix} k \\ m \end{pmatrix} r_m \log r_m. \tag{25}$$

For the state on the joint system, we have

$$\sigma_{RB^k} = \lambda|0\rangle\langle0|_R \otimes \mathcal{A}_{\gamma,N}(|0\rangle\langle0|)^{\otimes k} + (1-\lambda)|1\rangle\langle1|_R \otimes \mathcal{A}_{\gamma,N}(|1\rangle\langle1|)^{\otimes k} \tag{26}$$

$$+ \sqrt{\lambda(1-\lambda)}\big[|0\rangle\langle1|_R \otimes \mathcal{A}_{\gamma,N}(|0\rangle\langle1|)^{\otimes k} + |1\rangle\langle0|_R \otimes \mathcal{A}_{\gamma,N}(|1\rangle\langle0|)^{\otimes k}\big] \tag{27}$$

$$= \lambda|0\rangle\langle0|_R \otimes [(1-\gamma N)|0\rangle\langle0| + \gamma N|1\rangle\langle1|]^{\otimes k} \tag{28}$$

$$+ (1-\lambda)|1\rangle\langle1|_R \otimes [(\gamma - \gamma N)|0\rangle\langle0| + (1 - \gamma + \gamma N)|1\rangle\langle1|]^{\otimes k} \tag{29}$$

$$+ \sqrt{\lambda(1-\lambda)}(1-\gamma)^{k/2}|0\rangle\langle1|_R \otimes |0\rangle\langle1|^{\otimes k} \tag{30}$$

$$+ \sqrt{\lambda(1-\lambda)}(1-\gamma)^{k/2}|1\rangle\langle0|_R \otimes |1\rangle\langle0|^{\otimes k}. \tag{31}$$

This operator can be written as

$$\sigma_{RB^k} = \mu_0 \sigma^{(0)} + \sum_{\substack{s^k \in \{0,1\}^k: \\ s^k \neq (0,\ldots,0)}} \mu_{s^k}^{(0)}|0s^k\rangle\langle0s^k| + \sum_{\substack{s^k \in \{0,1\}^k: \\ s^k \neq (1,\ldots,1)}} \mu_{s^k}^{(1)}|1s^k\rangle\langle1s^k|, \tag{32}$$

where $\mu_0 = \lambda(1-\gamma N)^k + (1-\lambda)(1 - \gamma + \gamma N)^k$,

$$\sigma^{(0)} = \frac{1}{\mu_0} \begin{pmatrix} \lambda(1-\gamma N)^k & \sqrt{\lambda(1-\lambda)}(1-\gamma)^{k/2} \\ \sqrt{\lambda(1-\lambda)}(1-\gamma)^{k/2} & (1-\lambda)(1 - \gamma + \gamma N)^k \end{pmatrix} \tag{33}$$

in the basis $\{|0\rangle_R|0\rangle_B^{\otimes k}, |1\rangle_R|1\rangle_B^{\otimes k}\}$, and

$$\mu_{s^k}^{(0)} = \lambda(1-\gamma N)^{k-|s^k|}(\gamma N)^{|s^k|} \tag{34}$$

$$\mu_{s^k}^{(1)} = (1-\lambda)(\gamma - \gamma N)^{k-|s^k|}(1 - \gamma + \gamma N)^{|s^k|}. \tag{35}$$

Table A1. Table of the best neural network codes for the GADC $\mathcal{A}_{\gamma,N}$ with $(\gamma, N) = (0.414\,88, 0.2)$ and $k = 3, 4, 5$ channel copies. Only the non-zero amplitudes $\psi(s^n)$ indexed by the basis string s^n (with $n = 2k$) are shown (see equation (15)). The architecture used for the neural network codes is a feed-forward net with four hidden layers of width $2k$ each, activation functions cos and $3 \times$ tanh, and a Cartesian output layer (see section 4.1).

		$(\gamma, N) = (0.414\,88, 0.2)$	
$\lvert\nu_k\rangle$	$s^n\ (A^k\lvert R)$	$\psi(s^n)$	$\frac{1}{k}Q^{(1)}(\nu_k, \mathcal{A}_{\gamma,N}^{\otimes k})$
$k = 3$	000|110	$-0.486\,1 - 0.399\,4i$	$1.692\,3 \cdot 10^{-3}$
	001|011	$-0.339\,4 - 0.293\,8i$	
	010|011	$-0.339\,2 - 0.293\,7i$	
	100|011	$-0.339\,3 - 0.293\,8i$	
$k = 4$	0110|1111	$+0.391\,8 + 0.006\,1i$	$1.413\,2 \cdot 10^{-3}$
	1011|1111	$+0.388\,1 + 0.124\,8i$	
	1101|1111	$+0.401\,4 + 0.119\,0i$	
	1111|1000	$+0.701\,8 - 0.109\,6i$	
$k = 5$	00110|01011	$+0.404\,3 + 0.201\,0i$	$9.802\,5 \cdot 10^{-4}$
	10001|01011	$+0.448\,5 + 0.029\,8i$	
	10111|11101	$+0.754\,2 + 0.159\,1i$	

Let r denote one of the eigenvalues of the state $\sigma^{(0)}$ in equation (33), let

$$s_m := \lambda(1 - \gamma N)^{k-m}(\gamma N)^m \tag{36}$$

$$t_m := (1 - \lambda)(\gamma - \gamma N)^{k-m}(1 - \gamma + \gamma N)^m \tag{37}$$

for $m = 1, \ldots, k - 1$, and

$$s_0 = 0 \quad t_0 = (1 - \lambda)(\gamma - \gamma N)^k \tag{38}$$

$$s_k = \lambda(\gamma N)^k \quad t_k = 0. \tag{39}$$

Then the entropy of σ_{RB^k} equals

$$S(RB^k)_\sigma = \mu_0 h(r) - \mu_0 \log \mu_0 - \sum_{m=0}^{k}\binom{k}{m}(s_m \log s_m + t_m \log t_m). \tag{40}$$

The coherent information $Q^{(1)}(\phi_k^\lambda, \mathcal{A}_{\gamma,N}^{\otimes k}) = S(B^k)_\sigma - S(RB^k)_\sigma$ can now be efficiently computed using equations (25) and (40) for blocklengths up to $k = 20$.

A.2. Neural network codes for the GADC
We list the best neural network codes found for the GADC $\mathcal{A}_{\gamma,N}$ in the following tables:

- Table 1: $(\gamma, N) = (0.440\,35, 0.1)$
- Table A1: $(\gamma, N) = (0.414\,88, 0.2)$
- Table A2: $(\gamma, N) = (0.401\,02, 0.3)$
- Table A3: $(\gamma, N) = (0.393\,92, 0.4)$
- Table A4: $(\gamma, N) = (0.391\,69, 0.5)$.

A comparison of these codes to weighted repetition codes is plotted in figure 3 in the main text.

Appendix B. Codes for the dephrasure channel

In the following, we give a summary of the results about the coherent information of the dephrasure channel $\mathcal{N}_{p,q}$ (defined in equation (11)) that were obtained in [LLS18a]. These results are concerned with the one-way quantum capacity, as defined in section 2; for a discussion of two-way capacities, see [PLB19].

Table A2. Table of the best neural network codes for the GADC $\mathcal{A}_{\gamma,N}$ with $(\gamma, N) = (0.401\,02, 0.3)$ and $k = 3, 4$ channel copies. For details, see table A1. For $k = 5$ copies of the GADC $\mathcal{A}_{\gamma,N}$ with $(\gamma, N) = (0.401\,02, 0.3)$ the neural network ansatz did not find any codes with positive coherent information.

	$(\gamma, N) = (0.401\,02, 0.3)$		
$|\nu_k\rangle$	$s^n\,(A^k|R)$	$\psi(s^n)$	$\frac{1}{k}Q^{(1)}(\nu_k, \mathcal{A}_{\gamma,N}^{\otimes k})$
$k = 3$	000|010	$+0.256\,6 - 0.360\,1i$	$2.188\,9 \cdot 10^{-3}$
	000|011	$+0.280\,2 - 0.370\,4i$	
	001|100	$+0.257\,2 - 0.360\,7i$	
	010|100	$+0.257\,2 - 0.360\,7i$	
	100|100	$+0.257\,3 - 0.360\,7i$	
$k = 4$	0101|1110	$+0.162\,4 - 0.362\,9i$	$7.363\,5 \cdot 10^{-4}$
	1010|1110	$+0.064\,1 - 0.574\,8i$	
	1111|0001	$+0.058\,2 - 0.581\,9i$	
	1111|1000	$+0.185\,8 - 0.361\,8i$	

Table A3. Table of the best neural network codes for the GADC $\mathcal{A}_{\gamma,N}$ with $(\gamma, N) = (0.393\,92, 0.4)$ and $k = 3, 4, 5$ channel copies. For details, see table A1.

	$(\gamma, N) = (0.393\,92, 0.4)$		
$|\nu_k\rangle$	$s^n\,(A^k|R)$	$\psi(s^n)$	$\frac{1}{k}Q^{(1)}(\nu_k, \mathcal{A}_{\gamma,N}^{\otimes k})$
$k = 3$	000|000	$+0.365\,3 - 0.332\,8i$	$2.345\,6 \cdot 10^{-3}$
	000|010	$+0.351\,7 - 0.235\,6i$	
	001|110	$+0.349\,8 - 0.264\,2i$	
	010|110	$+0.349\,9 - 0.266\,1i$	
	100|110	$+0.355\,8 - 0.254\,1i$	
$k = 4$	0100|1100	$-0.609\,1 + 0.227\,8i$	$1.759\,2 \cdot 10^{-3}$
	0101|0010	$-0.188\,1 + 0.396\,4i$	
	0110|0010	$-0.187\,7 + 0.396\,3i$	
	1100|0010	$-0.187\,7 + 0.396\,3i$	

Table A4. Table of the best neural network codes for the GADC $\mathcal{A}_{\gamma,N}$ with $(\gamma, N) = (0.391\,69, 0.5)$ and $k = 3, 4, 5$ channel copies. For details, see table A1.

	$(\gamma, N) = (0.391\,69, 0.5)$		
$|\nu_k\rangle$	$s^n\,(A^k|R)$	$\psi(s^n)$	$\frac{1}{k}Q^{(1)}(\nu_k, \mathcal{A}_{\gamma,N}^{\otimes k})$
$k = 3$	010|101	$+0.189\,4 + 0.390\,9i$	$2.394\,8 \cdot 10^{-3}$
	100|101	$+1.894\,3 + 0.390\,9i$	
	110|011	$+0.173\,5 + 0.432\,2i$	
	110|100	$+0.173\,5 + 0.432\,3i$	
	111|101	$+0.189\,5 + 0.390\,9i$	
$k = 4$	0100|0111	$-0.262\,9 - 0.412\,0i$	$1.791\,3 \cdot 10^{-3}$
	0100|1101	$-0.228\,7 - 0.377\,2i$	
	0101|0110	$-0.224\,8 - 0.372\,1i$	
	0110|0110	$-0.225\,8 - 0.373\,8i$	
	1100|0110	$-0.222\,5 - 0.370\,8i$	
$k = 5$	01100|01010	$-0.331\,1 + 0.269\,4i$	$1.339\,3 \cdot 10^{-3}$
	10100|01010	$-0.333\,7 + 0.320\,9i$	
	11000|01010	$-0.333\,6 + 0.320\,8i$	
	11100|10110	$-0.291\,1 + 0.300\,9i$	
	11100|11001	$-0.333\,5 + 0.320\,8i$	

Table B1. Table of the best neural network codes for the dephrasure channel $\mathcal{N}_{p,q}$ with $(p, q) = (0.16, 0.3)$ and $k = 2$, 3, 4 channel copies. Only the non-zero amplitudes $\psi(s^n)$ indexed by the basis string s^n (with $n = 2k$) are shown (see equation (15)). The architecture used for the neural network codes is a feed-forward net with four hidden layers of width $2k$ each, activation functions cos and $3 \times$ ReLU, and a Polar output layer (see section 4.2).

$|\nu_k\rangle$	$s^n\ (A^k|R)$	$\psi(s^n)$	$\frac{1}{k}Q^{(1)}(\nu_k, \mathcal{N}_{p,q}^{\otimes k})$
$k = 2$	00|11	$+0.129\,8 - 0.990\,5i$	$2.146\,5 \cdot 10^{-5}$
	11|00	$+0.025\,5 - 0.038\,6i$	
$k = 3$	000|011	$+0.000\,8 + 0.000\,0i$	$3.968\,6 \cdot 10^{-5}$
	001|100	$-0.692\,2 - 0.711\,4i$	
	011|010	$-0.000\,5 + 0.000\,4i$	
	101|011	$-0.000\,7 + 0.000\,3i$	
	110|001	$+0.075\,4 + 0.094\,8i$	
	110|101	$-0.000\,5 - 0.001\,4i$	
$k = 4$	0000|1000	$+0.094\,1 - 0.105\,9i$	$4.792\,2 \cdot 10^{-5}$
	0010|1000	$+0.049\,5 + 0.101\,7i$	
	1101|0000	$+0.936\,2 + 0.301\,2i$	

B.1. Formula for the coherent information of repetition codes

Superadditivity of the channel coherent information of the dephrasure channel can be achieved using a simple *weighted repetition code*

$$|\phi_k^\lambda\rangle := \sqrt{\lambda}\,|0\rangle_R \otimes |0^k\rangle_{A^k} + \sqrt{1 - \lambda}\,|1\rangle_R \otimes |1^k\rangle_{A^k}, \tag{41}$$

where $\lambda \in [0, 1]$. In [LLS18a], the following formula is derived for its channel coherent information:

$$Q^{(1)}(\phi_k^\lambda, \mathcal{N}_{p,q}^{\otimes k}) = ((1 - q)^k - q^k)h(\lambda) - (1 - q)^k\left(1 - u\,\text{artanh}\,u - \frac{1}{2}\log(1 - u^2)\right), \tag{42}$$

where $h(\lambda) = -\lambda\log\lambda - (1 - \lambda)\log(1 - \lambda)$ is the binary entropy (in terms of the binary logarithm), $\text{artanh}(x) := \frac{1}{2}\log\frac{1+x}{1-x}$, and

$$u = u(\lambda, p, k) = \sqrt{1 - 4\lambda(1 - \lambda)(1 - (1 - 2p)^{2k})}. \tag{43}$$

Moreover, it is shown in [LLS18a] that for $k = 1$ the formula in equation (42) maximized over $\lambda \in [0, 1]$ is in fact the optimal single-letter channel coherent information. That is, $Q^{(1)}(\mathcal{N}_{p,q})$ is optimized by states whose marginal on the system qubits is diagonal in the computational basis. Hence, the formula equation (42) can be used to find quantum codes that surpass the optimal code for a single copy of $\mathcal{N}_{p,q}$, demonstrating superadditivity of coherent information.

For $q \in \{0.1, 0.2, 0.3, 0.4\}$ and the relevant intervals of p, the rates of the weighted repetition code ϕ_k^λ (optimized over $\lambda \in [0, 1]$) for $1 \leqslant k \leqslant 5$ are plotted in figure 5. The lines corresponding to ϕ_1 represent the codes achieving the optimal single-letter coherent information $Q^{(1)}(\mathcal{N}_{p,q})$.

B.2. Neural network codes for the dephrasure channel

We list the best neural network codes found for the dephrasure channel $\mathcal{N}_{p,q}$ in the following tables:

- Table 2: $(p, q) = (0.08, 0.4)$

- Table B1: $(p, q) = (0.16, 0.3)$

- Table B2: $(p, q) = (0.24, 0.2)$

- Table B3: $(p, q) = (0.32, 0.1)$.

A comparison of these codes to weighted repetition codes is plotted in figure 5 in the main text.

Appendix C. Codes for the depolarizing channel

C.1. Product repetition codes for the depolarizing channel

In this appendix, we discuss the known optimal codes for the depolarizing channel, which are given by repetition codes

Table B2. Table of the best neural network codes for the dephrasure channel $\mathcal{N}_{p,q}$ with $(p, q) = (0.24, 0.2)$ and $k = 2, 3, 4$ channel copies. For details, see table B1.
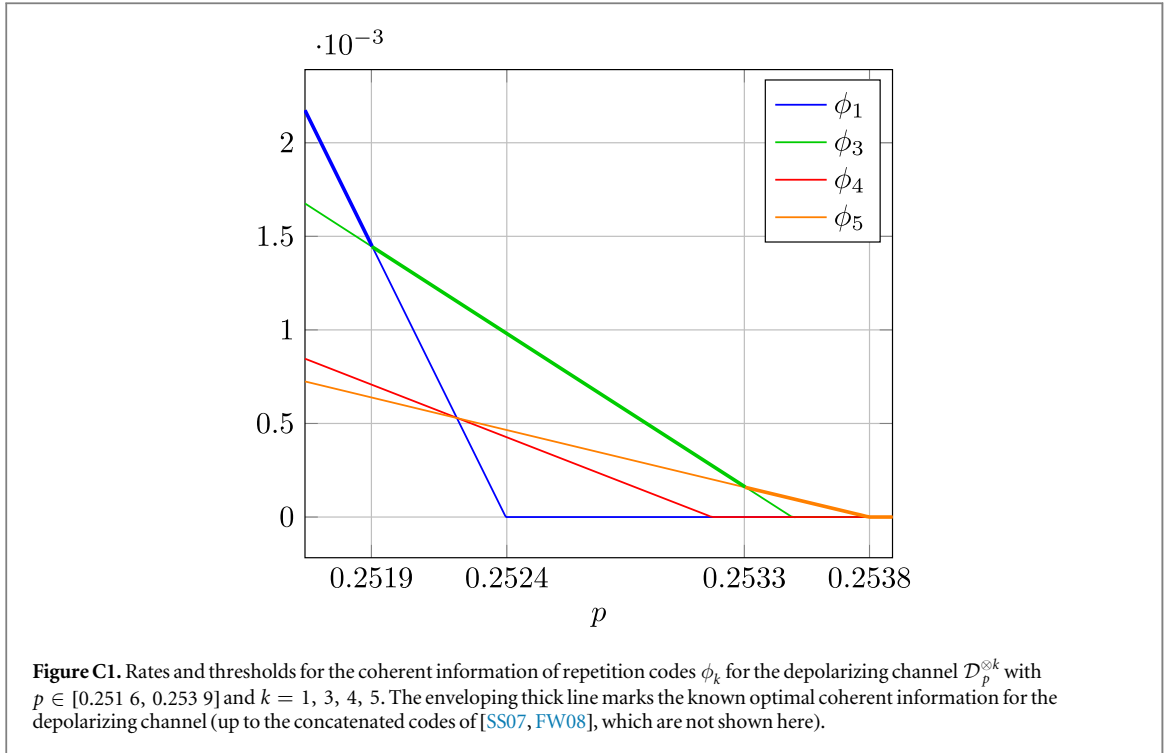
$\lvert\nu_k\rangle$	s^n $(A^k\lvert R)$	$\psi(s^n)$	$\frac{1}{k}Q^{(1)}(\nu_k, \mathcal{N}_{p,q}^{\otimes k})$
$k = 2$	01|00	$-0.007\,1 + 0.000\,5i$	$6.844\,7 \cdot 10^{-6}$
	01|01	$-0.082\,0 + 0.995\,9i$	
	10|10	$-0.032\,5 + 0.020\,2i$	
$k = 3$	000|100	$-0.000\,4 - 0.000\,9i$	$1.138\,2 \cdot 10^{-5}$
	001|000	$-0.002\,2 + 0.004\,8i$	
	001|001	$+0.023\,5 + 0.001\,1i$	
	001|010	$+0.025\,7 - 0.001\,9i$	
	001|101	$+0.160\,0 + 0.979\,5i$	
	001|110	$+0.010\,0 + 0.009\,4i$	
	011|110	$-0.000\,6 - 0.000\,9i$	
	101|110	$+0.001\,0 - 0.000\,5i$	
	110|011	$-0.054\,8 - 0.103\,1i$	
$k = 4$	0110|1110	$-0.104\,7 + 0.072\,7i$	$1.156\,1 \cdot 10^{-5}$
	0111|1110	$-0.053\,1 + 0.086\,3i$	
	1000|0010	$+0.140\,5 - 0.976\,6i$	
	1001|1100	$+0.000\,1 + 0.000\,0i$	
	1001|1101	$+0.000\,3 - 0.003\,1i$	

Table B3. Table of the best neural network codes for the dephrasure channel $\mathcal{N}_{p,q}$ with $(p, q) = (0.32, 0.1)$ and $k = 2, 3, 4$ channel copies. For details, see table B1.

$\lvert\nu_k\rangle$	s^n $(A^k\lvert R)$	$\psi(s^n)$	$\frac{1}{k}Q^{(1)}(\nu_k, \mathcal{N}_{p,q}^{\otimes k})$
$k = 2$	00|11	$+0.242\,5 - 0.031\,8i$	$9.920\,4 \cdot 10^{-5}$
	01|00	$-0.003\,8 - 0.007\,4i$	
	10|00	$-0.003\,8 - 0.007\,4i$	
	11|00	$+0.000\,0 - 0.000\,1i$	
	11|01	$-0.321\,5 - 0.914\,7i$	
$k = 3$	000|111	$-0.486\,3 - 0.820\,8i$	$1.117\,2 \cdot 10^{-4}$
	010|101	$+0.000\,0 - 0.000\,1i$	
	011|000	$+0.036\,7 - 0.162\,4i$	
	100|000	$-0.000\,8 + 0.000\,9i$	
	101|000	$+0.033\,2 - 0.162\,8i$	
	110|000	$+0.024\,6 - 0.164\,3i$	
	111|000	$-0.032\,8 - 0.076\,2i$	
$k = 4$	0000|1101	$-0.139\,7 - 0.056\,4i$	$1.180\,2 \cdot 10^{-4}$
	0100|0010	$-0.000\,1 - 0.000\,4i$	
	0101|1010	$-0.017\,1 - 0.087\,9i$	
	0110|0110	$-0.584\,3 + 0.357\,6i$	
	0111|1010	$+0.000\,2 + 0.000\,0i$	
	1001|0000	$+0.093\,3 - 0.678\,7i$	
	1010|0011	$-0.000\,1 + 0.000\,1i$	
	1010|0111	$-0.019\,6 - 0.087\,4i$	
	1011|0100	$+0.000\,4 - 0.000\,1i$	
	1111|0011	$-0.098\,5 - 0.114\,0i$	

$$\lvert\phi_k\rangle = \frac{1}{\sqrt{2}}(\lvert0\rangle_R\lvert0\rangle_A^{\otimes k} + \lvert1\rangle_R\lvert1\rangle_A^{\otimes k}). \tag{44}$$

For $p \lesssim 0.251\,9$, the single-letter coherent information equation (13) is optimal. For $0.251\,9 \lesssim p \lesssim 0.253\,3$, the 3-repetition code ϕ_3 (defined in equation (44)) is optimal, while for $0.253\,3 \lesssim p \lesssim 0.253\,8$ the 5-repetition code ϕ_5 is optimal. The point $p \simeq 0.253\,8$ marks the highest threshold for a single repetition code. This

Figure C1. Rates and thresholds for the coherent information of repetition codes ϕ_k for the depolarizing channel $\mathcal{D}_p^{\otimes k}$ with $p \in [0.251\ 6, 0.253\ 9]$ and $k = 1, 3, 4, 5$. The enveloping thick line marks the known optimal coherent information for the depolarizing channel (up to the concatenated codes of [SS07, FW08], which are not shown here).

threshold can be further extended using the concatenated codes of [SS07, FW08][4]. We summarize this in figure C1, where we compare the repetition codes and their rates and thresholds.

The above codes are the best known *information-theoretic* codes, yielding the best lower bounds on the quantum capacity of the depolarizing channel by equation (3). However, in numerical investigations we are facing a slightly different problem of maximizing the k-coherent information $\frac{1}{k}Q^{(1)}(\psi, \mathcal{D}_p^{\otimes k})$ over quantum codes ψ for *fixed* k, that is, solving

$$\underset{\psi}{\mathrm{argmax}} \frac{1}{k}Q^{(1)}(\psi, \mathcal{D}_p^{\otimes k}). \tag{45}$$
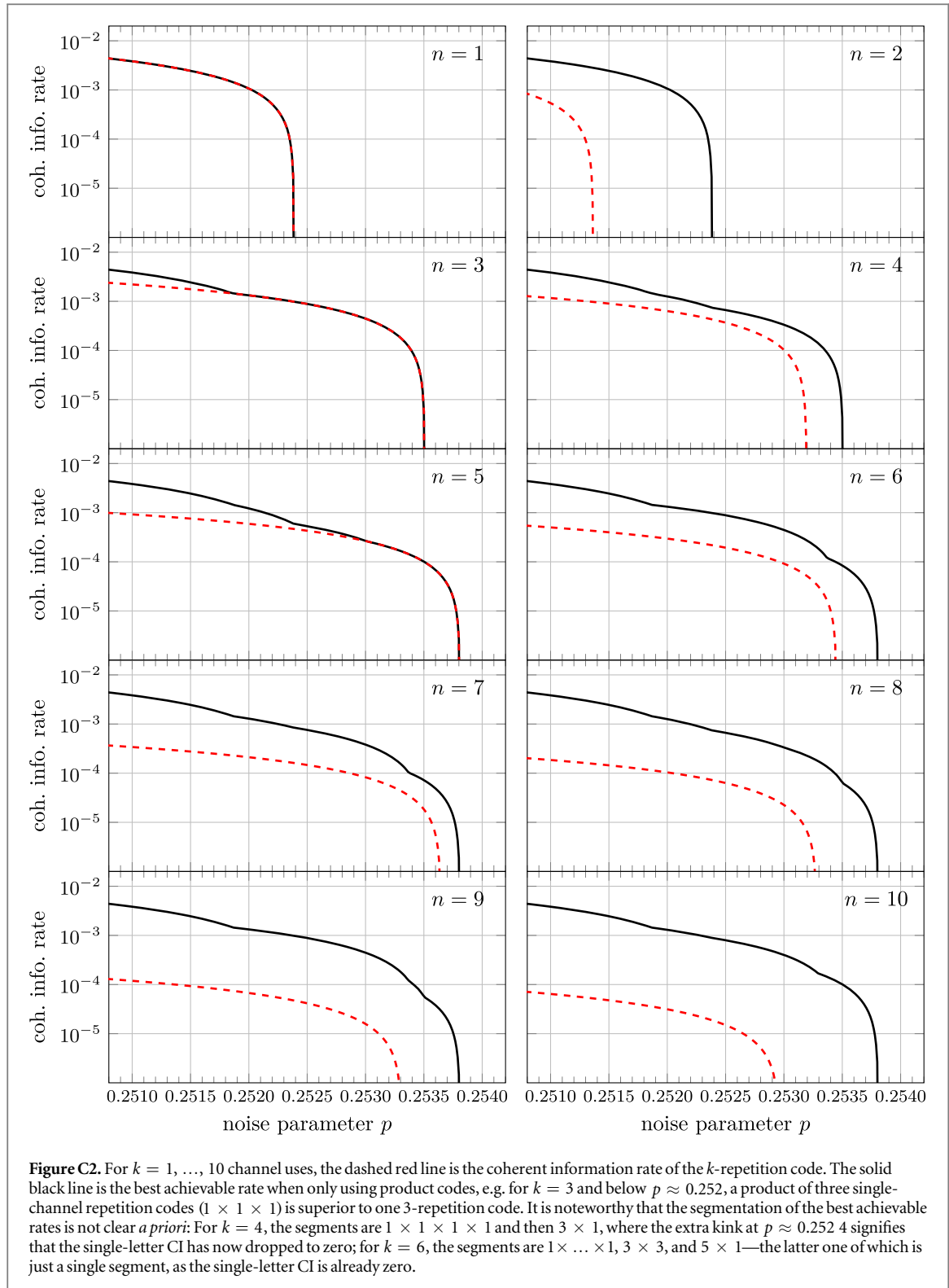
For $k \leqslant 9$ channel uses, the optimization problem equation (45) is solved by *products* of repetition codes,

$$|\Phi_{\mathbf{k}}\rangle = \bigotimes_{i=1}^{l} |\phi_{k_i}\rangle. \tag{46}$$

Here, $\mathbf{k} = (k_1, \ldots, k_l)$, and the resulting code $|\Phi_{\mathbf{k}}\rangle$ is a quantum code on $\sum_{i=1}^{l} k_i$ channel input qubits and l purifying qubits.

To illustrate this, consider 4 channel uses of the depolarizing channel, and recall that the single-letter coherent information equation (13) vanishes around $p \simeq 0.252\ 4$. The respective thresholds for the 4-repetition code ϕ_4 and the 3-repetition code ϕ_3 on *three* input qubits are $p \simeq 0.253\ 2$ and $p \simeq 0.253\ 5$, respectively (see figure C1 and the file `rep-codes-tabular.txt` in [Anc]). Hence, for $0.253\ 2 \leqslant p \leqslant 0.253\ 5$ it is clearly advantageous to 'freeze' one input qubit to some fixed pure state, and use a 3-repetition code on the remaining 3 input qubits. Since pure input states can never establish coherent information between Alice and Bob, the frozen input does not contribute to the overall coherent information, and the resulting code incurs a penalty in the rate. However, this code inherits the same threshold as the 3-repetition code on three input qubits, thus outperforming the plain 4-repetition code. Similarly, one finds that for $p \in [0.251\ 9, 0.252\ 4]$ the quantity $\frac{1}{4}Q^{(1)}(\mathcal{D}_p^{\otimes 4})$ is maximized by a 3-repetition code tensored with a 1-repetition code (i.e. using three of the four input qubits with one purifying qubit for a repetition code, and maximally entangling the remaining input qubit with another purifying qubit). In figure C2 and table C1, we provide an overview of the thresholds and rates of the optimal such combinations of repetition codes for $k \leqslant 10$ uses of the depolarizing channel. For $k \geqslant 10$ uses of the depolarizing channel, concatenated codes can surpass the best known repetition code thresholds [SS07, FW08].

---

[4] Note that these concatenated codes require at least 10 channel uses of the depolarizing channel, and thus their rate is far lower than the rates of the codes just described. Furthermore, investigating $n \geqslant 10$ channel uses of the depolarizing channel is at the moment out of reach for our numerical methods. For these reasons, we focus on the regime $p \leqslant 0.253\ 8$ within the threshold of ϕ_5.

Figure C2. For $k = 1, \ldots, 10$ channel uses, the dashed red line is the coherent information rate of the k-repetition code. The solid black line is the best achievable rate when only using product codes, e.g. for $k = 3$ and below $p \approx 0.252$, a product of three single-channel repetition codes ($1 \times 1 \times 1$) is superior to one 3-repetition code. It is noteworthy that the segmentation of the best achievable rates is not clear *a priori*: For $k = 4$, the segments are $1 \times 1 \times 1 \times 1$ and then 3×1, where the extra kink at $p \approx 0.252\,4$ signifies that the single-letter CI has now dropped to zero; for $k = 6$, the segments are $1 \times \ldots \times 1$, 3×3, and 5×1—the latter one of which is just a single segment, as the single-letter CI is already zero.

C.2. Products of repetition codes as benchmark for depolarizing noise

As a benchmark for finding quantum codes, we demand that the models we propose can at least achieve the product repetition codes described above; either because they can represent products of repetition codes directly, or because they achieve the target rates by some other means. In particular, this should serve as a sanity check for the models we propose, indicating whether we need to increase the width of a hidden layer, or the depth of the model. The relevant question for us is whether a state $|\Phi_{\mathbf{n}}\rangle$ as defined in equation (46) can always be represented accurately by the weights obtained from an RBM or an FF net.

Table C1. Intermediate product repetition code thresholds; before the first column at 0.251 86 the best code is given by the single-letter coherent information.

			p			
k	0.251 86	0.252 38	0.253 01	0.253 29	0.253 37	0.253 50
4	1×3					
5	$1 \times 1 \times 3$	2×3	5×0			
6	3×3				1×5	
7	$1 \times 3 \times 3$				2×5	
8	$1 \times 1 \times 3 \times 3$	$2 \times 3 \times 3$	3×5			
9	$3 \times 3 \times 3$				$1 \times 3 \times 5$	4×5
10	$1 \times 3 \times 3 \times 3$			5×5		

Table C2. Summary of the possible encodings of a symbol $i \in [d]_0$ in a d-ary string i^n of length n. We denote by $\mathrm{bin}_k(m)$ the binary representation of m of length k (with leading zeros if necessary), and by e_i a vector with a 1 in the ith component and 0s elsewhere.

	encoding ($i \in [d]_0$)	# input nodes
Scaled	$i \mapsto i/(d-1)$	n
Binary	$i \mapsto \mathrm{bin}_{\lceil \log d \rceil}(i)$	$\lceil \log d \rceil n$
One-hot	$i \mapsto e_i$	dn

C.2.1. RBM states.   First observe how the Hamiltonian $\mathbf{H}_{\mathrm{RBM}}$ describes a linear single-layer FF classifier (i.e. a linear function on the inputs i_k). Seen as a linear function on bit strings, the Hamiltonian can therefore represent a target state $|\psi\rangle$ as well as a linear model allows. For the simple case of products of repetition codes, where we subdivide the set of basis states into those of weight 0 and 1, respectively, this question is well-studied in the context of linear classifiers.

A single k-repetition code has the form $|0 \cdots 0\rangle + |1 \cdots 1\rangle =: |a\rangle + |b\rangle$. Since the RBM uses a scaled encoding (see table C2), the bit strings correspond to real entries in a k-dimensional vector, and thus $|b\rangle = 0|a\rangle$; a linear function L therefore necessarily satisfies $L(|b\rangle) = L(|a\rangle) = 0$. If we let $|b\rangle$ be a basis state (unnormalized) and complete the basis with $k - 1$ arbitrary orthogonal vectors, it therefore suffices to define L in such a way to have $\ker L = \mathrm{span}\{|b\rangle\}$.

Products of repetition codes always have the form $\bigotimes_{i=1}^{k}|\phi_{n_i}\rangle$; since basis states are bit strings for the RBM classifier, the corresponding code is a direct sum of the individual repetition codes. We can thus construct a classifier for the overall code by writing $L_1 \oplus \ldots \oplus L_k = L$, which is still linear.

Since $\mathbf{H}_{\mathrm{RBM}}$ appears in an exponential in equation (6), we can use the spectral gap of $\mathbf{H}_{\mathrm{RBM}}$ to obtain a lower bound on how close to zero an entry in the code can be set. For instance, if we were to represent a 3-repetition code $|000\rangle + |111\rangle$, we can require that $|111\rangle$ is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{H}_{\mathrm{RBM}}$; all other binary strings should have an energy that is as large as possible, such that the exponential function suppresses the corresponding weight. Consider the binary state $|011\rangle$, which has overlap $\sqrt{2/3}$ with $|111\rangle$ (assuming normalization). If Δ is the spectral gap of $\mathbf{H}_{\mathrm{RBM}}$—i.e. the difference between the ground state energy and the second lowest eigenvalue—then $\langle 001|\mathbf{H}_{\mathrm{RBM}}|001\rangle = 2\Delta/3$, yielding a lower bound between largest code weight and smallest code weight of $\exp(-\Delta)$. To get an empirical estimate, assume we flip a single bit—e.g. i_1—in equation (5). How large can the energy difference be? If all parameters are chosen (in magnitude) within a range $[-M, M]$, then a simple estimate would be $\Delta \leqslant M + M^2$; this is, of course, an upper bound to a lower bound. In practice we found that $M = 10$ is sufficient for our purposes.

C.2.2. DBM states.   Equation (7) introduces a quadratic term in the input. Since one can easily embed a 1-IN-3SAT instance into a quadratic polynomial (for three Boolean variables v_1, v_2, v_3 where TRUE $= 1$ and FALSE $= 0$ enforced by terms $v_i^2 - v_i = 0$, the equation $(v_1 + v_2 + v_3 - 1)^2 = 0$ if and only if exactly one of the v_i is TRUE; the existence or nonexistence of a root for the sum of all constraints thus answers the instance), it is clear that the discriminative power of DBM states should vastly outperform that of RBM states, albeit at a higher

computational cost. As discussed in the introduction, for various ground states of local Hamiltonians this intuition has empirically been shown to be correct.

C.2.3. Feed forward network states.   It is easy to explicitly construct weights for an FF net that can represent any product repetition code. As a first step, consider a single repetition code $|\phi_n\rangle$. We set up a three-layer neural network from n inputs, one hidden layer of width 1, and a single output node (for simplicity we disregard the imaginary part for the state output in equation (8)). The weights and activation functions to be chosen are

$$\{x_i\}_{i=1}^{n+1} \longmapsto y := \cos\left(\frac{2\pi}{n+1}\sum_{i=1}^{n+1}x_i\right) \longmapsto z := \text{ReLU}\left(\frac{y - \cos(2\pi/n+1)}{1 - \cos(2\pi/n+1)}\right), \tag{47}$$

and one can verify that the output is one on the all 1s and 0s input, and zero otherwise. We refer to appendix F for a more detailed discussion.

For a product code given by some $\mathbf{n} = (n_1, \ldots, n_k)$, we simply partition the input nodes into k subsets and dovetail those with a network given in equation (47); we obtain k outputs $z_1, \ldots, z_k$. Since we know that a logic AND gate corresponds to all the $z_i = 1$, we can use a final $\text{ReLU}(\sum_{i=1}^{k}z_i - k + 1)$ layer to enforce that the weights are 1 if all individual segments are valid repetition codes, and 0 otherwise—or merge all already existing ReLU nodes into one. Observe that we could always implement a single cos node in this function with two ReLU nodes, followed by another ReLU node to combine the outputs (as in appendix F). This would increase the hidden layer width by a factor of two; we can incorporate addition of the individual outputs into the last existing ReLU layer, so the depth should remain constant.

One immediate consequence is that any product code of k repetition codes can always be represented by a network architecture where the first hidden layer has width k; and we in fact empirically found that the trained weights of the first layer are similar to those in equation (47).

A final note on the parameter range necessary for the argument: the largest coefficients in absolute value in equation (47) and its final AND node are $\max_i(1 - \cos(2\pi/n(i))^{-1}$, or $k - 1$, whichever is larger. Restricting the network's parameter range artificially below this threshold could result in worse representability of product repetition codes.

C.2.4. Schmidt network states.   The argument is similar as for feed-forward network states. Note that, in general, Schmidt codes will be redundant, since for e.g. four channel uses we are forced to using more than just a single purifying qubit. The fact that the neural net calculates Schmidt coefficients means that the repetition codes always uses as many purifying dimensions as system dimensions.

Appendix D. AME states

An AME(n, d)-state is a pure state $|\psi_{n,d}\rangle \in (\mathbb{C}^d)^{\otimes n}$ on n qudits with local dimension $d \geqslant 2$ satisfying

$$\rho_\mathcal{S} = \text{tr}_{\mathcal{S}^c}\psi_{n,d} = \frac{1}{|\mathcal{S}|}I_\mathcal{S} \tag{48}$$

for every $\mathcal{S} \subset [n]$ with $|\mathcal{S}| = \lfloor\frac{n}{2}\rfloor$. As mentioned in the main text, whether or not an AME(n, d)-state exists depends on n and d. Using weight enumerator theory [SL97, Rai98], Scott proved that an AME(n, d)-state can only exist if $n \leqslant 2(d^2 - 1)$ for even n, and $n \leqslant 2d(d + 1) - 1$ for odd n [Sco04]. This technique was recently extended by Huber *et al* [Hub+18] to give further constraints on the existence of AME(n, d)-states. For fixed n an AME(n, d)-state always exists for sufficiently large local dimension d [HC13]. For example, AME(n, d)-states exist for d a prime power and $n \leqslant d$ [GBR04]. Recently, it was proved in [HGS17] that an AME state on seven qubits cannot exist. This result completely settled the case of qubit AME states: they exist for $n = 2, 3, 5, 6$, and only for these n. Furthermore, there are constructions for certain combinations of parameters (n, d) [GBR04, HC13, Hel13, Goy+15, Goy+17]. We refer to figure 2 in [Hub+18] as well as Problem 35 on the IQOQI Vienna Open Quantum Problems list [Opq] for a more complete overview of the known results about existence of AME(n, d)-states. Here, we merely mention that it is unknown whether AME(n, d)-states exist for $(n, d) = (4, 6)$ and $(n, d) = (7, 4), (7, 6)$.

Scott [Sco04] proved that a multipartite state $|\psi_{n,d}\rangle \in (\mathbb{C}^d)^{\otimes n}$ is AME if and only if the average linear entropy $Q_m(\psi_{n,d}) = 1$, where $Q_m(\cdot)$ is defined in equation (17). Since we are searching for AME states by maximizing $Q_m(\cdot)$, we need to make sure that a state $\psi_{n,d}$ with $Q_m(\psi_{n,d}) \approx 1$ is also approximately AME. We determine the latter by introducing the average trace distance parameter D_m defined in equation (19) that measures the average trace distance between the marginals of $\psi_{n,d}$ on m subsystems and the completely mixed state. The average trace distance parameter $D_m(\cdot)$ can be bounded from above in terms of $Q_m(\cdot)$, as stated in equation (20). We restate

this bound here for the reader's convenience:

$$D_m(\psi_{n,d}) \leqslant \sqrt{2\log[d^m - (d^m - 1)Q_m(\psi_{n,d})]}. \tag{49}$$

To prove equation (49), we use the quantum version of Pinsker's inequality, $D(\rho\|\sigma) \geqslant \frac{1}{2}\|\rho - \sigma\|_1^2$, where $D(\rho\|\sigma) = \mathrm{tr}(\rho\log\rho) - \mathrm{tr}(\rho\log\sigma)$ is the quantum relative entropy. We also use the 2-relative Rényi entropy $D_2(\rho\|\sigma) = \log\mathrm{tr}(\rho^2\sigma^{-1})$ [Pet86], and the well-known fact that $D(\rho\|\sigma) \leqslant D_2(\rho\|\sigma)$.

Observe first that, for $\pi = \frac{1}{d}I$, we have $D_2(\rho\|\pi) = \log\mathrm{tr}(\rho^2) + \log d$, and hence

$$\mathrm{tr}(\rho^2) = \frac{1}{d}\exp(D_2(\rho\|\pi)) \geqslant \frac{1}{d}\exp(D(\rho\|\pi)) \geqslant \frac{1}{d}\exp\left(\frac{1}{2}\|\rho - \pi\|_1^2\right). \tag{50}$$

Abbreviating $\pi_{\mathcal{S}} = \frac{1}{|\mathcal{S}|}I_{\mathcal{S}}$, we then bound

$$Q_m(\psi_{n,d}) = \binom{n}{m}^{-1}\frac{d^m}{d^m - 1}\sum_{\mathcal{S}\subset[n]:|\mathcal{S}|=m}(1 - \mathrm{tr}\rho_{\mathcal{S}}^2) \tag{51}$$

$$\leqslant \binom{n}{m}^{-1}\frac{d^m}{d^m - 1}\sum_{\mathcal{S}\subset[n]:|\mathcal{S}|=m}\left(1 - \frac{1}{d^m}\exp\left(\frac{1}{2}\|\rho_{\mathcal{S}} - \pi_{\mathcal{S}}\|_1^2\right)\right) \tag{52}$$

$$= \frac{1}{d^m - 1}\left(d^m - \binom{n}{m}^{-1}\sum_{\mathcal{S}\subset[n]:|\mathcal{S}|=m}\exp\left(\frac{1}{2}\|\rho_{\mathcal{S}} - \pi_{\mathcal{S}}\|_1^2\right)\right) \tag{53}$$

$$\leqslant \frac{1}{d^m - 1}\left(d^m - \exp\left(\frac{1}{2}D_m(\psi_{n,d})^2\right)\right), \tag{54}$$

where the last inequality follows from concavity of the function $x \mapsto -\exp\left(\frac{x^2}{2}\right)$. Rearranging equation (54) yields equation (49).

Since AME states are defined on tensor products of d-dimensional Hilbert spaces, the input string i^n to the neural network computing the amplitude $\psi(i^n)$ in the ansatz equation (15) is a d-ary string. Depending on the local dimension, we use different encodings of this d-ary input string, as explained in appendix E below.

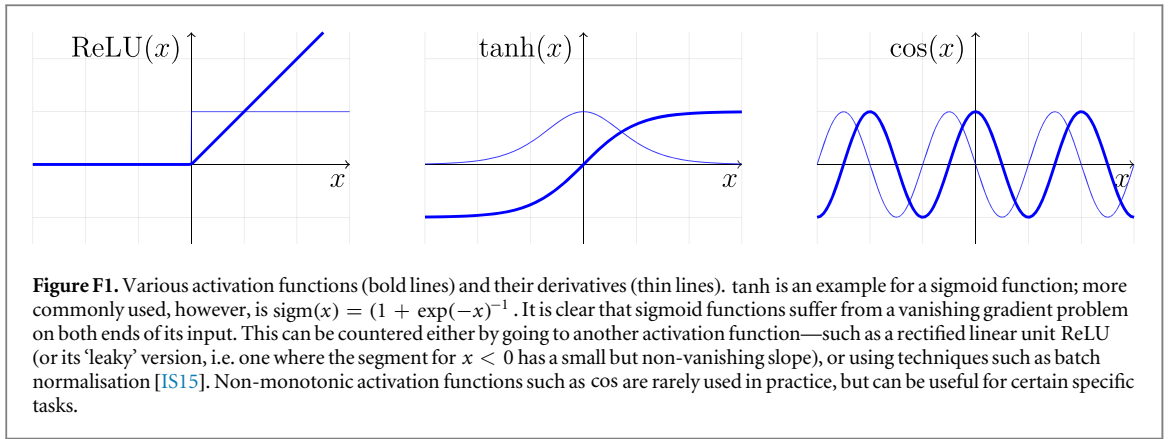## Appendix E. Input encoding of d-ary strings for neural networks

In order to parametrize quantum states on n qubits, it is rather straightforward to use the neural network ansatz described in section 3. In the case of AME(n, d)-states with local dimension $d > 2$, we slightly tweak the neural network ansatz. To this end, we fix a basis $\{|i\rangle\}_{i=0}^{d-1}$ for $\mathbb{C}^d$, and express a general quantum state $\psi_{n,d} \in (\mathbb{C}^d)^{\otimes n}$ as

$$|\psi_{n,d}\rangle = \frac{1}{C}\sum_{i^n \in [d]_0^n}\psi(i^n)|i^n\rangle, \tag{55}$$

where C is again a normalization constant ensuring $\langle\psi_{n,d}|\psi_{n,d}\rangle = 1$, and we use the notation $[d]_0 := \{0, \ldots, d-1\}$. We consider three different options of encoding the d-ary input string i^n in order to obtain the amplitudes $\psi(i^n)$ in equation (55) from a neural network:

1. (Scaled) direct encoding: Use the d-ary string i^n directly, with a possible scaling of the entries such that $i_k \in [0, 1]$ for $k \in [n]$.

2. Binary encoding: Convert each symbol $i_k \in [d]_0$ into a binary string, requiring $\lceil\log d\rceil$ 'physical' qubits per 'logical' qudit of $\psi_{n,d}$, and use the resulting binary string of length $\lceil\log d\rceil n$ as the input to the neural network.
   Example: For $d = 6$, the encoding is $0 \mapsto 000, 1 \mapsto 111, \cdots, 5 \mapsto 101$.

3. One-hot encoding: Encode each symbol in a 'one-hot' vector of length d and use the resulting binary string of length dn as the input to the neural network.
   Example: For $d = 6$, the encoding is $0 \mapsto 000001, 1 \mapsto 000010, \cdots, 5 \mapsto 100000$.

We have found that the performance of the specific encoding used in the neural network optimization depends on the local dimension d. For prime d, the neural network optimization using the scaled encoding converges quickly to known AME(n, d)-states such as AME(4, 7), as evident from figure 8 in the main text. On the other hand, for composite d the NN ansatz is more powerful using binary or one-hot encoding. Since binary encoding has a smaller overhead in terms of the 'physical' qubits used in the ansatz ($\lceil\log d\rceil n$ versus dn), we use binary encoding for composite local dimension d. We summarize the different encodings in table C2.

Figure F1. Various activation functions (bold lines) and their derivatives (thin lines). tanh is an example for a sigmoid function; more commonly used, however, is $\mathrm{sigm}(x) = (1 + \exp(-x))^{-1}$. It is clear that sigmoid functions suffer from a vanishing gradient problem on both ends of its input. This can be countered either by going to another activation function—such as a rectified linear unit ReLU (or its 'leaky' version, i.e. one where the segment for $x < 0$ has a small but non-vanishing slope), or using techniques such as batch normalisation [IS15]. Non-monotonic activation functions such as cos are rarely used in practice, but can be useful for certain specific tasks.

Appendix F. The role of activation functions for quantum codes

In machine learning, the use of nonlinear activation functions is crucial to a neural network's performance; otherwise, the network is just a single affine transformation and not useful beyond linear regression. The overall network can have varying activation functions per neuron (see figure 1). In essentially all cases, the activation functions are the same within a layer. The operation of such a layer is thus to perform an affine transformation on the input vector and then, element-wise, apply the nonlinearity *f*. For a single neuron *z* depending on $x = (x_1, \ldots, x_n)$, the mathematical operation can thus be visualized as
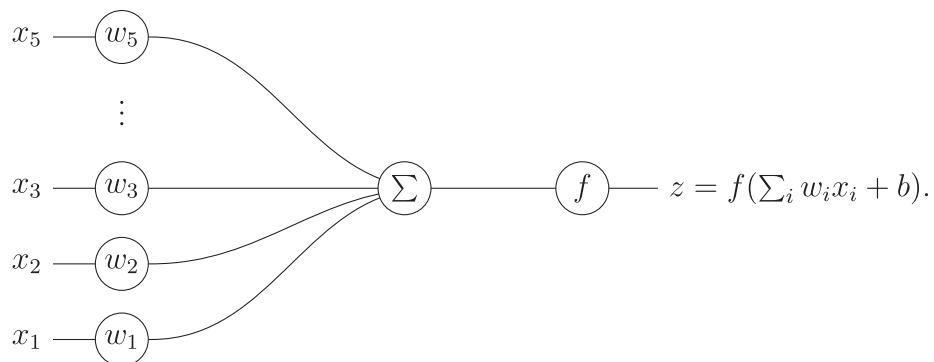
Commonly used activation functions are e.g. ReLU, sigm or tanh, which are plotted in figure F1; in addition to some thorough studies [IS15, He+15, KSH17], there seems to be a lot of empirical understanding which activation functions perform better in various scenarios [Phy]. One example is that e.g. sigm saturates (meaning the gradient vanishes for large or small values), whereas e.g. ReLU does not have the same problem. Furthermore, the general consensus seems to be that non-monotonic or periodic activation functions—such as e.g. sin—weaken the neural network's performance. We found conflicting evidence for this in the literature ([Sop99, GA16] and [GBC16], section 6.2.2), suggesting that such periodic functions can indeed be useful for specific tasks—especially in the context of representing ground states for local Hamiltonians [CL18].

In one example of such a task [CL18], use neural network states to approximate the ground states of certain Hamiltonians. They report good performance of feed-forward network architectures with a cosine activation function in the first layer for a 1D anti-ferromagnetic Heisenberg model, arguing that the cosine function is capable of handling the 'sign problem' typically found in the analysis of Hamiltonians. We found that using cosine in the first hidden layer also performs well in finding good quantum codes for quantum channels such as the depolarizing channel defined in equation (12), or the dephrasure channel defined in equation (11).

In the following, we want to give an intuition why a periodic activation function such as cos can be useful for learning quantum codes with a structure that can be easily derived from the binary signature of its state vector. To give an example, consider a repetition code on five qubits, given by $|00000\rangle + |11111\rangle$. A function $M \colon (\mathbb{C}^2)^{\otimes 5} \to \mathbb{C}$ with $M(|00000\rangle) = M(|11111\rangle) = 1$, and 0 elsewhere, is trivial to construct from elementary logic gates (i.e. either all bits are zero, or all bits are one).

For a feed-forward neural network, one could imagine adding up all bits within one neuron, and thresholding this value with a ReLU activator:

$$z_1 = \text{ReLU}\left(2\sum_{i=1}^{5} x_i - 9\right) = \begin{cases} 1 & \text{if } x_i = 1 \quad \forall\, i \\ 0 & \text{otherwise.} \end{cases}$$

A similar gate with flipped signs can activate only when all bits are zero; the two outputs can then be combined using a final ReLU node.

We can achieve the same activation using a single cos neuron, dovetailed by a ReLU in the next layer:

$$z_1 = \cos\left(\frac{2\pi}{5}\sum_{i=1}^{5} x_i\right) \quad \text{and} \quad z_2 = \text{ReLU}\left(\frac{z_1 - \cos(1/5)}{1 - \cos(1/5)}\right).$$

While this looks like a more complicated version of the same calculation, it quickly becomes obvious that one can easily perform modular arithmetic using this technique—what we have in fact calculated is whether $\sum_i x_i \equiv 0 \pmod{5}$.

Why is this an advantage? As a slightly more complicated example, let us consider an (unnormalized) tensor code built from a 3-repetition code $|\phi_3\rangle = |000\,000\rangle + |111\,111\rangle$ and a 1-repetition code (or simply maximally entangled state) $|\phi_1\rangle = |0\,0\rangle + |1\,1\rangle$. In both cases, the first block of qubits (3 resp. 1) is sent through the channel, and the second block form the purifying environment. On 4 qubits, the tensor code thus looks as follows (for visualization purposes we boldface the single channel repetition code):

$$|\phi_3\rangle \otimes |\phi_1\rangle = |\mathbf{000}0\,\mathbf{000}0\rangle + |\mathbf{000}1\,\mathbf{000}1\rangle + |\mathbf{111}0\,\mathbf{111}0\rangle + |\mathbf{111}1\,\mathbf{111}1\rangle. \tag{56}$$

Any tensor channel $\mathcal{N}^{\otimes n}$ is naturally covariant[5] with respect to permuting tensor factors, i.e. the unitary representation $\pi \mapsto U_\pi$ of the symmetric group S_n on $(\mathbb{C}^2)^{\otimes n}$ defined by $U_\pi |e_1\rangle \otimes \dots \otimes |e_n\rangle = |e_{\pi^{-1}(1)}\rangle \otimes \dots \otimes |e_{\pi^{-1}(n)}\rangle$. Since the coherent information $I(A\rangle B)$ is furthermore invariant under local unitaries of the form $U_A \otimes U_B$, codes that are permutations of each other yield the same value for the coherent information. For example, the code

$$(U_{(14)} \otimes U_{(24)})(|\phi_3\rangle \otimes |\phi_1\rangle) = |\mathbf{000}0\,\mathbf{000}0\rangle + |\mathbf{100}0\,\mathbf{010}0\rangle + |\mathbf{011}1\,\mathbf{101}1\rangle + |\mathbf{111}1\,\mathbf{111}1\rangle \tag{57}$$

is obtained from $|\phi_3\rangle \otimes |\phi_1\rangle$ by swapping channel qubits 1 and 4 and environment qubits 2 and 4,[6] and is thus equivalent for quantum information transmission[7]. Hence, within each block of four qubits (either channel or environment) the code is characterized by the Hamming weight of the code vectors (0, 1, 3 and 4 in the example above), and ideally this is identified by the neural network. With modular arithmetic, we can have a cos neuron identifying 0 and 4 (e.g. all Hamming weights $\equiv 0 \pmod{4}$), and another one identifying 1 and 3 (e.g. all odd Hamming weights).

While it is conceivable that for simple codes such as equation (56) one can write down relatively simple circuits with non-periodic activation functions, it should be clear that we do save space within the neural network representation if we can perform calculations such as the ones above within a single neuron.

Appendix G. Numerical optimization techniques

In most applications neural networks are trained using the backpropagation method, in which each network parameter is updated using the gradient of a loss or objective function with respect to that parameter. In our main application of neural networks, maximizing the coherent information of a quantum channel, the objective function is the coherent information itself. In the interesting case of a high-noise quantum channel (such as $\mathcal{D}_p$ for $p \gtrsim 0.252\,3$), a randomly selected quantum code (e.g. with respect to the Haar measure on pure states) has strictly negative coherent information with high probability, whereas a product state $|\psi_1\rangle_R \otimes |\psi_2\rangle_A$ always has vanishing coherent information, $I_c(\psi_1 \otimes \psi_2, \mathcal{N}) = S(\mathcal{N}(\psi_2)) - S(\psi_1 \otimes \mathcal{N}(\psi_2)) = 0$. Hence, the coherent information landscape is dominated by local maxima, and gradient-based optimization techniques are likely to get stuck in these local maxima.

---

[5] A quantum channel $\mathcal{M}: A \to B$ is *covariant* with respect to a group G if there are unitary representations $g \mapsto U_A(g)$ on $\mathcal{H}_A$ and $g \mapsto U_B(g)$ on $\mathcal{H}_B$ such that $\mathcal{M}(U_A(g) \cdot U_A(g)^\dagger) = U_B(g)\mathcal{M}(\cdot)U_B(g)^\dagger$ for all $g \in G$.

[6] Note that in equation (57) the two tensor products on the left-hand side are with respect to different tensor factors. For the first tensor product, the two factors correspond to channel input and purifying qubits, respectively.

[7] We do *not* claim that optimal codes are in any way symmetric due to this permutation invariance.

This intuition was confirmed in our numerical search for good quantum codes for the depolarizing channel and the dephrasure channel. In the search for $AME(n, d)$ states, the objective function is the function $Q_m(\psi)$ defined in equation (17). Here, numerical investigations also showed that gradient-based optimization was again likely to get stuck in local minima.

The failure of gradient-based optimization methods in both scenarios led us to consider gradient-free, stochastic global optimization techniques instead. In the following, we give high-level explanations of four popular such algorithms, PSO, ABC, pattern search (also known as direct search), and genetic evolution. A numerical comparison regarding their performance can be found in figures H1–H8.

G.1. Particle swarm optimization

PSO [KE95] is a meta-heuristic, derivative-free global optimization technique. The idea of PSO is to have multiple *particles* explore the landscape on the search for a global minimum, and communicate their individual best value to the swarm. At the same time, each particle records its own history and stores the personal best value. In each iteration, the update of a particle's velocity vector is determined by the current velocity, recurrence to the location of the personal best function value, and attraction towards the location of the global best value.

More precisely, fix model parameters α, β, $\gamma > 0$ and consider N particles with random initial position $\mathbf{x}_i^{(0)}$ and random initial velocity $\mathbf{v}_i^{(0)}$ for $i \in [N]$. For each particle i, the variable $\mathbf{p}_i$ stores the location of the personal best function value, while the variable $\mathbf{g}$ stores the location of the global best function value among the whole swarm. In the kth iteration, the velocity and position of a particle are updated according to

$$\mathbf{v}_i^{(k)} = \alpha \mathbf{v}_i^{(k-1)} + \beta r_\beta (\mathbf{p}_i - \mathbf{x}_i^{(k-1)}) + \gamma r_\gamma (\mathbf{g} - \mathbf{x}_i^{(k-1)}) \tag{58}$$

$$\mathbf{x}_i^{(k)} = \mathbf{x}_i^{(k-1)} + \mathbf{v}_i^{(k)}, \tag{59}$$

where r_β, $r_\gamma \in [0, 1]$ are drawn uniformly at random. The parameter α is called *inertia*, while β and γ are usually called *self-interaction* and *social interaction*, respectively. A common modification of the PSO is to limit the social interaction to neighborhoods of a certain size within the swarm, ensuring a more thorough exploration of the landscape by the swarm.

The MATLAB implementation of PSO, available in the Global Optimization Toolbox, uses the neighborhood modifications with variable neighborhood sizes and an adaptive adjustment of the inertia weight. We refer to the official documentation [Pso] for details of the algorithm, as well as the MATLAB files in [Anc] for the algorithm settings used in this paper. Furthermore, we used the 'inertia weight' variant of PSO in Pagmo [Abc], with parameter settings as found in the C++ source files [Anc].

G.2. Artificial bee colonization

ABC [Kar05] is another meta-heuristic, derivative-free global optimization technique based on the principle of swarm intelligence. The algorithm works as follows: The population consists of N *employer bees* and N *onlooker bees*. While the employer bees explore the neighborhood of randomly created 'food sources' (i.e. points in the landscape with a low objective function value for a minimization problem), the onlooker bees evaluate the food sources according to the promise given by the *fitness* of the food source, and join the employer bees in exploring the neighborhood of those food sources. If an employer bee cannot find any new food around its location for a certain number of iterations (i.e. it fails to find points in the neighborhood of the food source with a lower objective function value), it is converted into a *scout bee* and assigned to a new random food source.

In more detail, to minimize a function $f: \mathbb{R}^D \to \mathbb{R}$, an employer bee at site x_i randomly explores the neighborhood of x_i by probing the location x_i' which differs from x_i in exactly one randomly drawn component $j \in [D]$ according to

$$(x'_i)_j = (x_i)_j + r((x_i)_j - (x_k)_j), \tag{60}$$

where $x_k \neq x_i$ is another randomly drawn food source, and $r \in [-1, 1]$ is a uniform random number. If $f(x_i') < f(x_i)$, the employer bee switches to x'_i and continues exploring its neighborhood. The fitness of the food source x_i is defined as $\text{fit}_i := (1 + f(x_i)^{-1}$, and each onlooker bee reinforces the employer bee group by selecting a food source according to the probability distribution $\{\text{fit}_i / \sum_i \text{fit}_i\}_i$.

We use the standard implementation of ABC found in the C++ optimization library Pagmo [Abc], as well as our own implementation of the standard algorithm in MATLAB (see [Anc]).

G.3. Pattern search

The third derivative-free optimization technique we use in this paper is called *pattern search* or *direct search*. To minimize a function $f\colon \mathbb{R}^D \to \mathbb{R}$, the algorithm takes as input a starting point $x_0 \in \mathbb{R}^D$ together with the objective function value $f(x_0)$, and creates a mesh of probing points around the starting point. In each iteration or *poll*, the objective function is evaluated at each mesh point. If for one of the mesh points, say x_1, the objective function value is lower than the current one (at x_0), the algorithm centers at x_1 and creates a new mesh.

There are different ways in how the mesh at a new center point is created. In a popular variant called *generalized pattern search* (GPS), the new probing points y_i of the mesh are defined by a fixed set $\mathcal{S} \subset \mathbb{R}^D$ of vectors. Common choices are $\mathcal{S}_{2D} = \{\pm\mathbf{e}_i\}_{i=1}^D$, where $\mathbf{e}_i$ denotes the ith standard basis vector, or $\mathcal{S}_{D+1} = \{\mathbf{e}_i\}_{i=1}^D \cup \{-(\mathbf{e}_1 + \ldots + \mathbf{e}_D)\}$. In the kth round with center point x_{k-1}, the points of the mesh are defined as $y_i = x_{k-1} + \Delta v_i$, where $v_i \in \mathcal{S}$, and Δ is the *mesh constant*. In a successful poll (i.e. when a new point with a lower objective function value is found), the mesh constant for the new mesh is doubled. If the poll is unsuccessful, the center point remains the same and Δ is halved.

Another popular variant is called *mesh adaptive direct search*. Here, the set $\mathcal{R} \subset \mathbb{R}^D$ of vectors for the new mesh points is randomly created after each successful poll. In analogy to the GPS variant above, common choices are $\mathcal{R}_{2D} = \{\pm\mathbf{v}_i\}_{i=1}^D$ and $\mathcal{R}_{D+1} = \{\mathbf{v}_i\}_{i=1}^D \cup \{-(\mathbf{v}_1 + \ldots + \mathbf{v}_D)\}$, where in each case the $\mathbf{v}_i$ are random vectors.

The above variants of pattern search are available in the Global Optimization Toolbox of MATLAB [Psm]. We refer to the MATLAB files in [Anc] for the algorithm settings used in this paper.
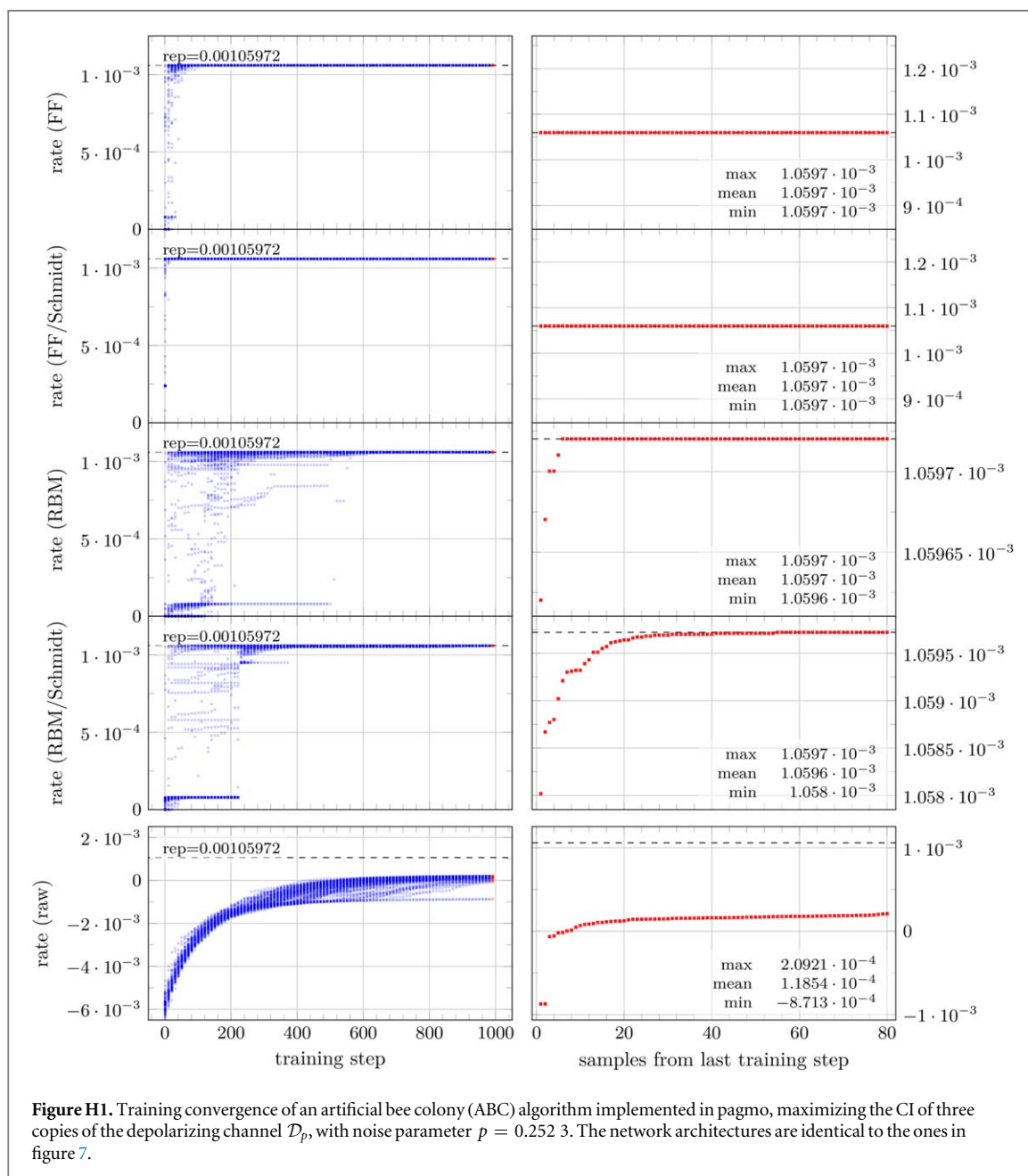
G.4. Simple genetic algorithm

The fourth derivative-free optimization algorithm is a genetic algorithm, which is related to evolutionary methods such as PSO and ABC, but motivated from the process of gene evolution.

Starting from a random selection of N so-called 'chromosomes' $\mathbf{x}_i^{(0)}$—where each vector component is called a 'gene'—a traditional implementation follows four steps.

- Pick random tuples of size s from the chromosome pool, and select the ones with the best function value within each tuple; this creates a selected chromosome pool of size less than N.

- Randomly select a parent tuple (can be more than two, and up to the entire selected pool). Merge the parents, e.g. by selecting a random chromosome, and replacing each gene (coordinate of $\mathbf{x}_i^{(0)}$) with some probability p by genes from other chromosomes. Continue creating child chromosomes until the new pool reaches size N.

- Randomize child genes within each chromosome according to some randomness distribution $\mathcal{D}$ and mutation probability m; a popular variant of which is called *polynomial mutation* where $\mathcal{D} \sim 1/\mathrm{poly}$, which introduces a stronger bias towards creating children close to their parents.

- Merge parent and child chromosome pool and select N of the fittest candidates.

We use Pagmo's standard implementation of a simple genetic algorithm with polynomial mutation (SGE, [Sge]), with parameters $s = 2$, $p = 0.9$ and $m = 0.02$.
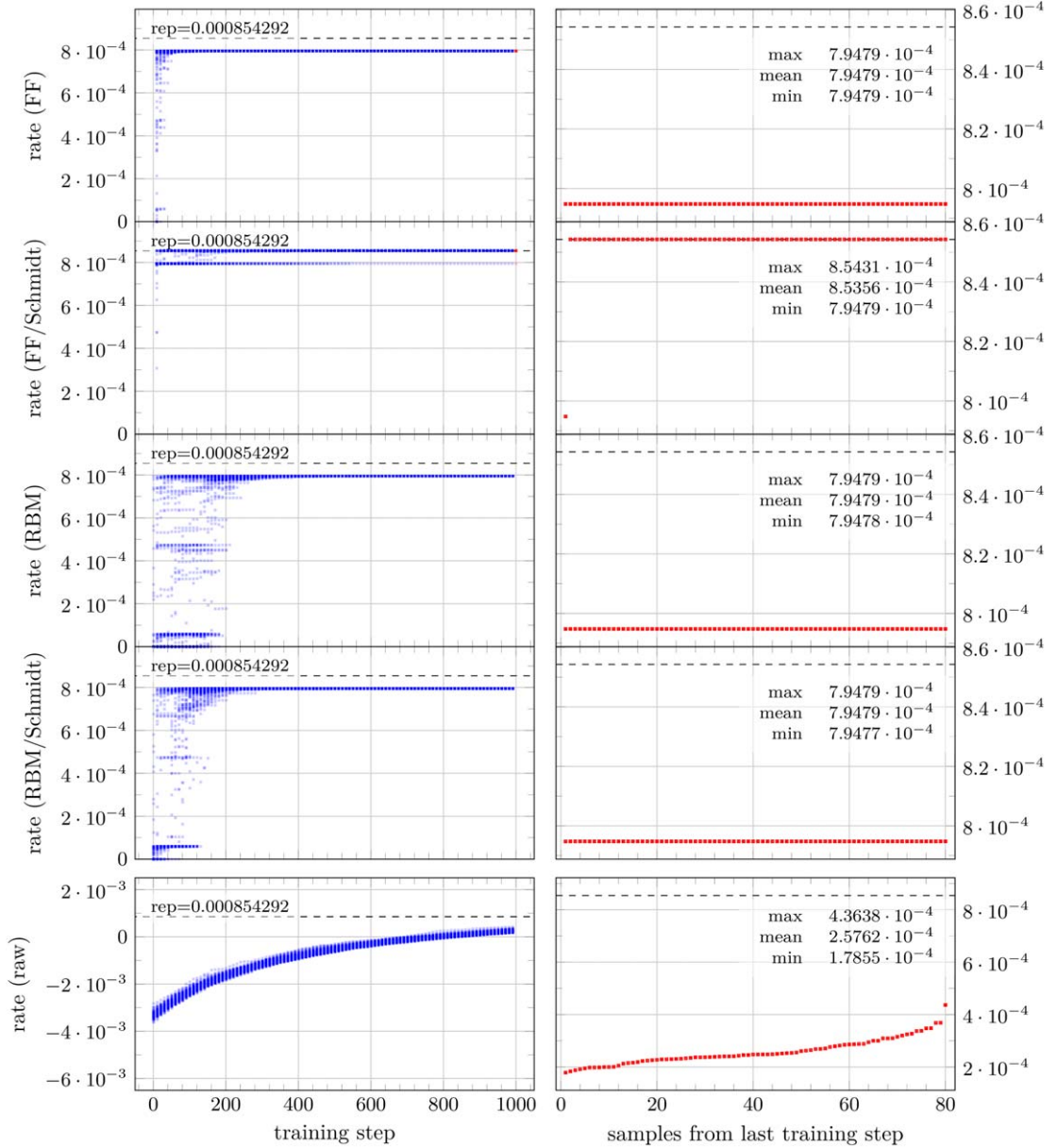
Appendix H. Additional numerical data

Figure H1. Training convergence of an artificial bee colony (ABC) algorithm implemented in pagmo, maximizing the CI of three copies of the depolarizing channel $\mathcal{D}_p$, with noise parameter $p = 0.252\,3$. The network architectures are identical to the ones in figure 7.

Figure H2. Training convergence of an artificial bee colony (ABC) algorithm implemented in pagmo, maximizing the CI of four copies of the depolarizing channel $\mathcal{D}_p$, with noise parameter $p = 0.252\,3$. The network architectures are identical to the ones in figure 7. We remark that ABC seems to have troubles moving beyond a local minimum around the three-repetition code with CI $= 0.000\,794\,8$ in all but the FF/Schmidt ansatz.
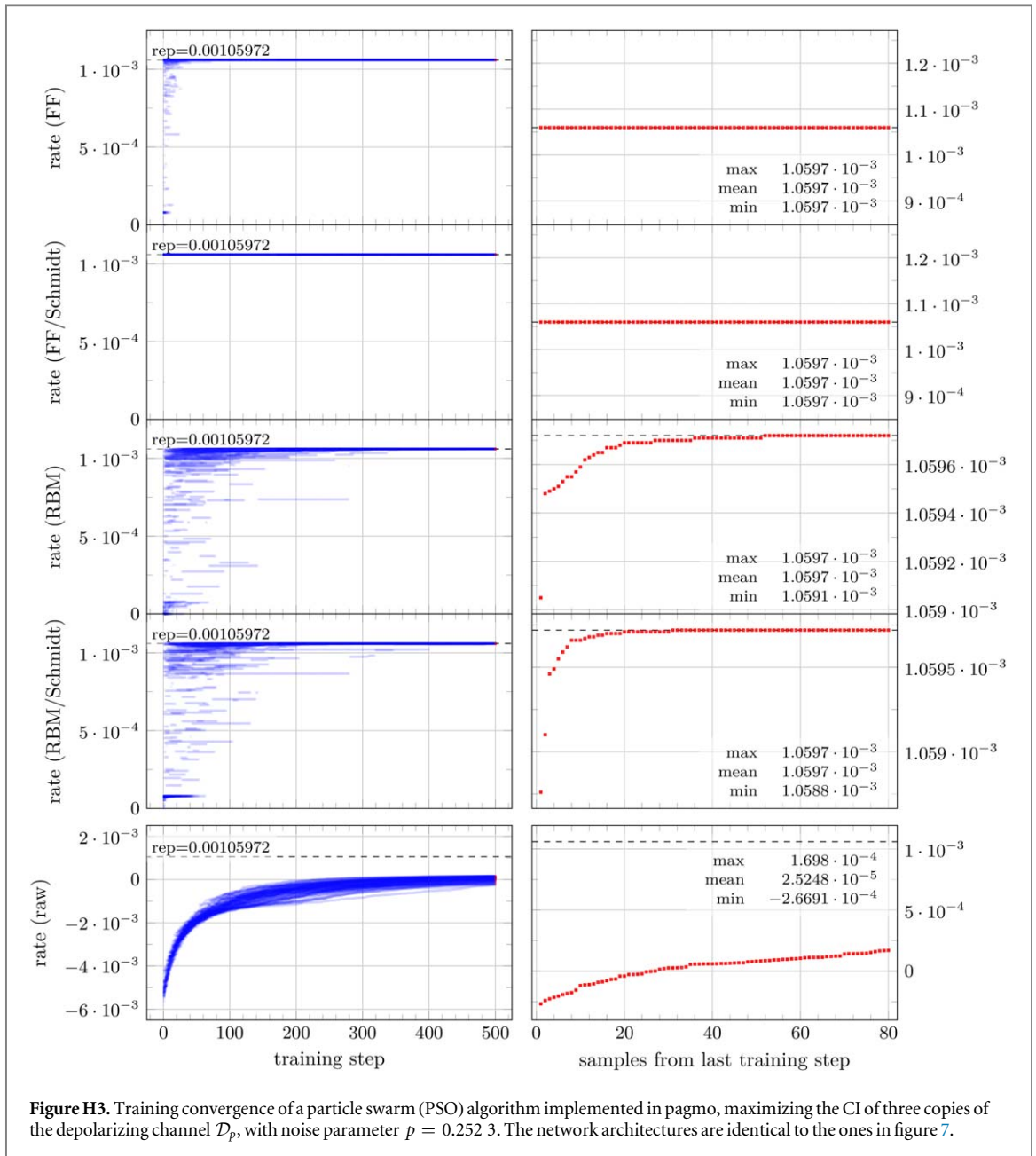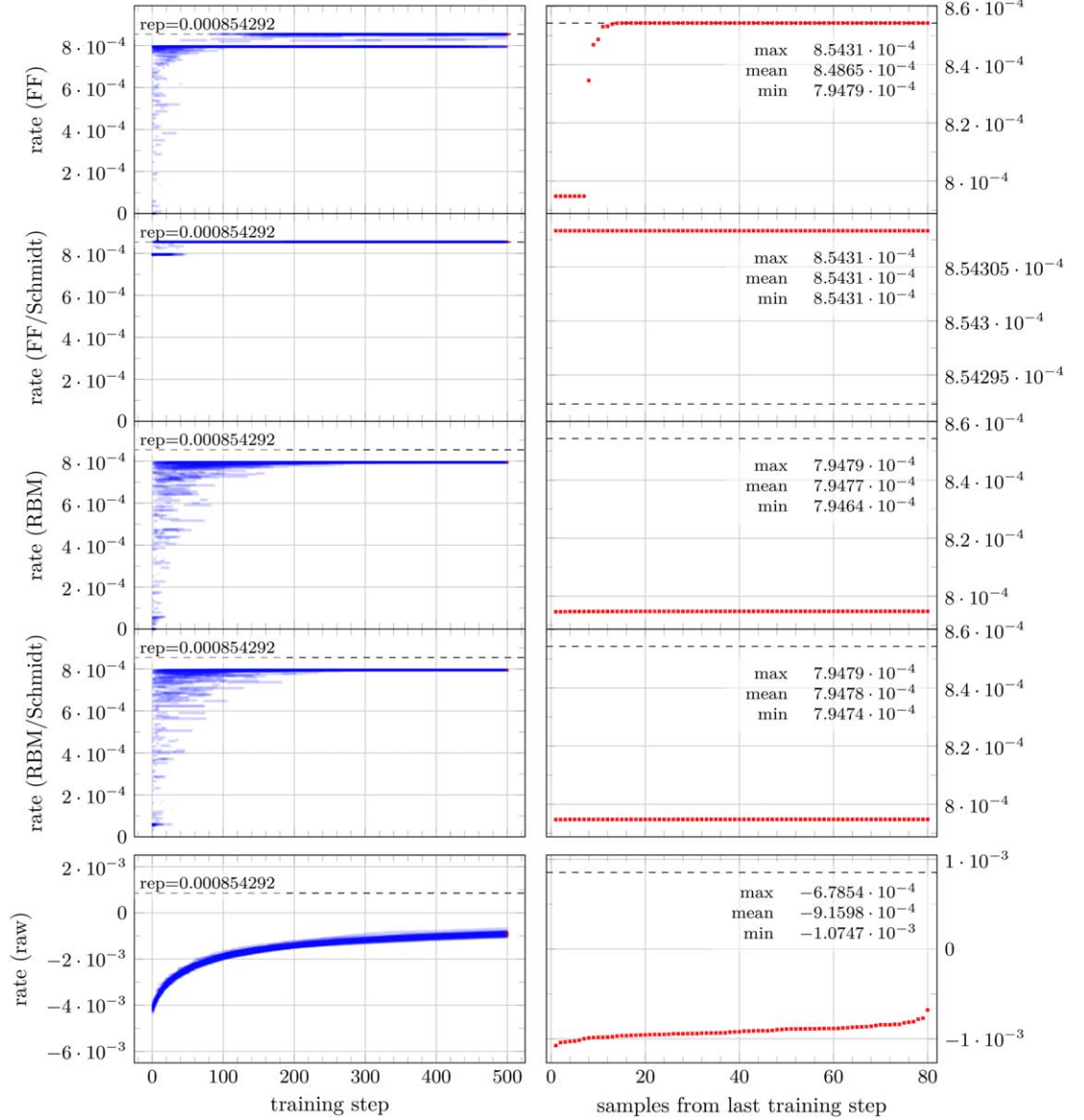
Figure H3. Training convergence of a particle swarm (PSO) algorithm implemented in pagmo, maximizing the CI of three copies of the depolarizing channel $\mathcal{D}_p$, with noise parameter $p = 0.252\ 3$. The network architectures are identical to the ones in figure 7.
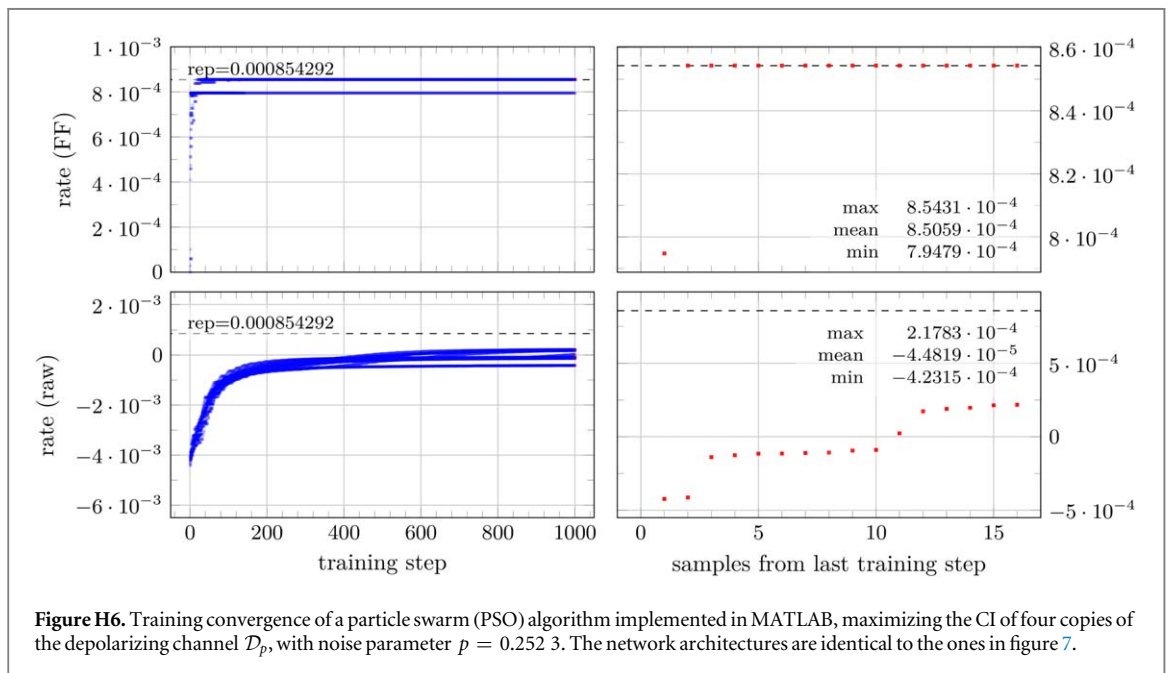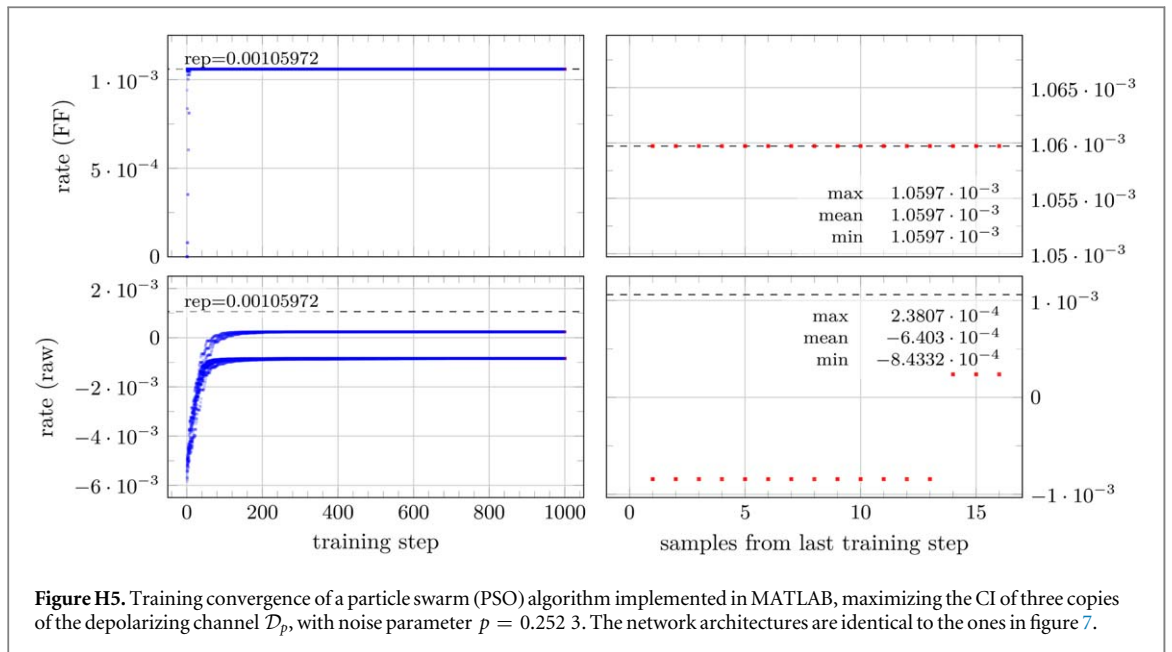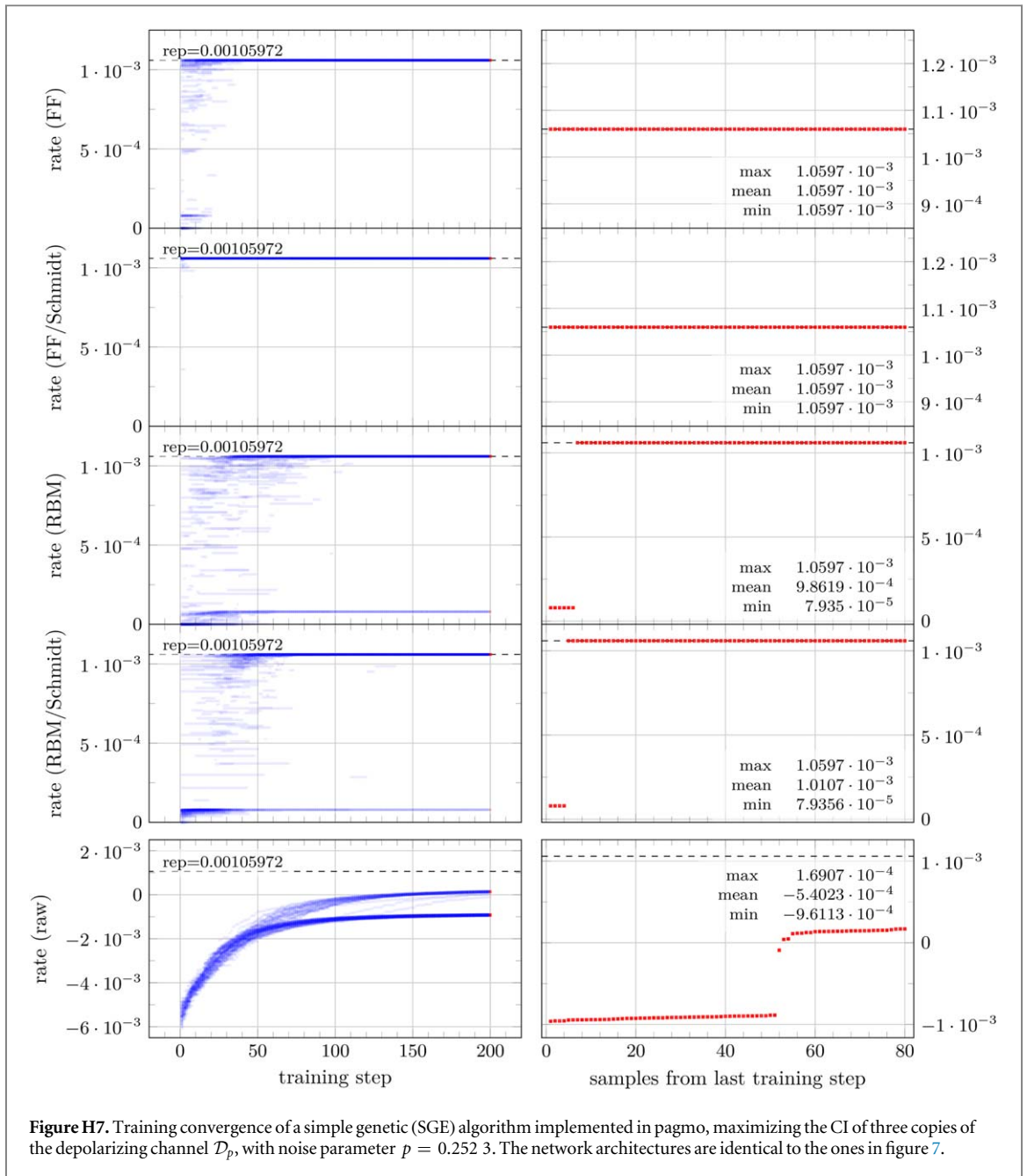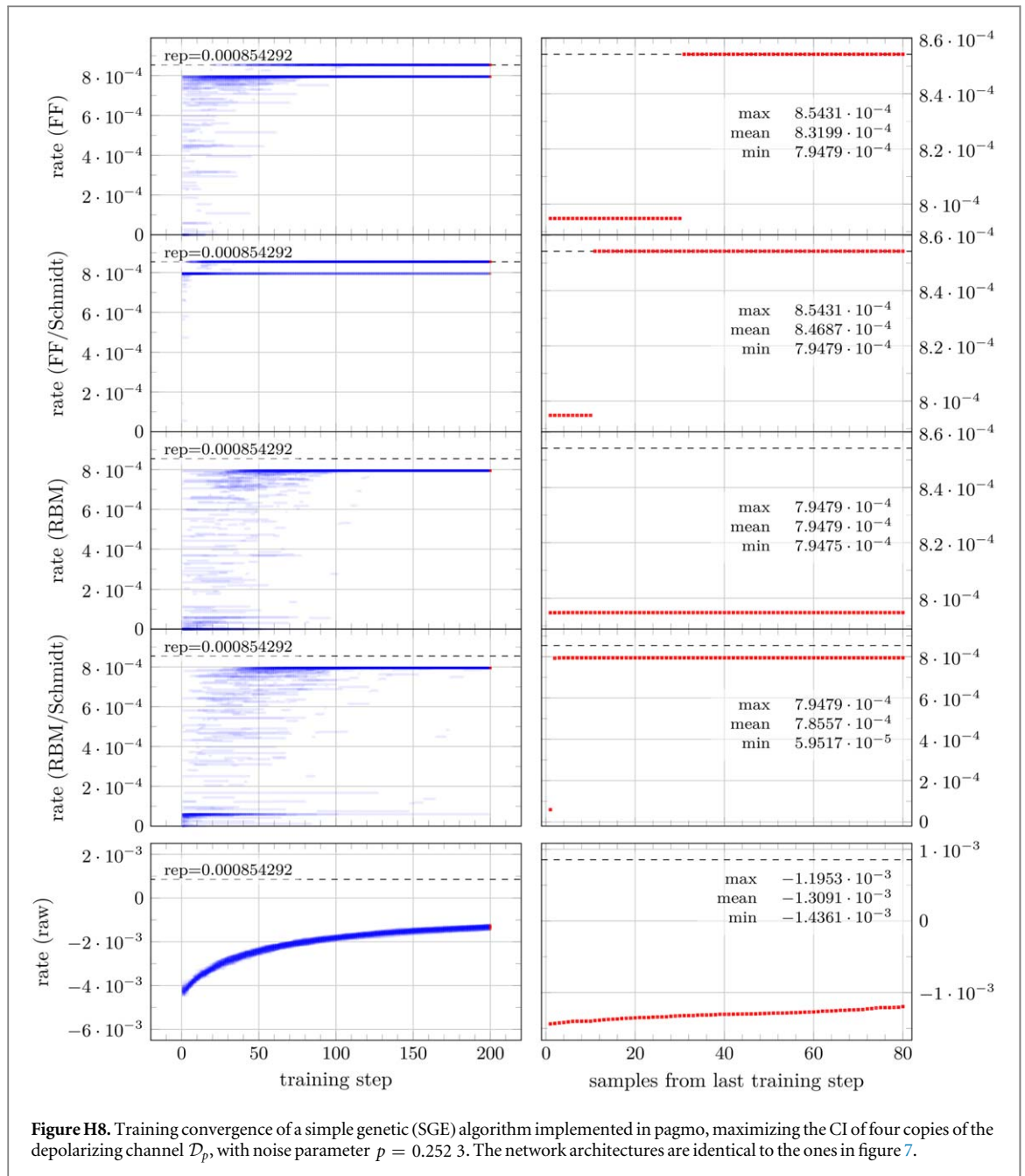
Figure H4. Training convergence of a particle swarm (PSO) algorithm implemented in pagmo, maximizing the CI of four copies of the depolarizing channel $\mathcal{D}_p$, with noise parameter $p = 0.252\,3$. The network architectures are identical to the ones in figure 7.

Figure H5. Training convergence of a particle swarm (PSO) algorithm implemented in MATLAB, maximizing the CI of three copies of the depolarizing channel $\mathcal{D}_p$, with noise parameter $p = 0.252\,3$. The network architectures are identical to the ones in figure 7.

Figure H6. Training convergence of a particle swarm (PSO) algorithm implemented in MATLAB, maximizing the CI of four copies of the depolarizing channel $\mathcal{D}_p$, with noise parameter $p = 0.252\,3$. The network architectures are identical to the ones in figure 7.

Figure H7. Training convergence of a simple genetic (SGE) algorithm implemented in pagmo, maximizing the CI of three copies of the depolarizing channel $\mathcal{D}_p$, with noise parameter $p = 0.252\,3$. The network architectures are identical to the ones in figure 7.

Figure H8. Training convergence of a simple genetic (SGE) algorithm implemented in pagmo, maximizing the CI of four copies of the depolarizing channel $\mathcal{D}_p$, with noise parameter $p = 0.252\,3$. The network architectures are identical to the ones in figure 7.

ORCID iDs

Johannes Bausch ⓘ https://orcid.org/0000-0003-3189-9162
Felix Leditzky ⓘ https://orcid.org/0000-0002-1073-9795

References

[Anc] 2018 Ancillary files: MATLAB and c++ code used in the paper arXiv:1806.08781v1
[Ara+13] Arad I, Kitaev A, Landau Z and Vazirani U 2013 An area law and sub-exponential algorithm for 1D systems arXiv:1301.1162
[Abc] Artificial bee colony algorithm (implementation in pagmo) https://esa.github.io/pagmo2/docs/cpp/algorithms/bee_colony.html (Accessed: 17 May 2018)
[BB84] Bennett C H and Brassard G 1984 Quantum cryptography: public key distribution and coin tossing *1984 IEEE Int. Conf. on Computers, Systems and Signal Processing (New York)* vol 175
[BW92] Bennett C H and Wiesner S J 1992 Communication via one- and two-particle operators on einstein-podolsky-rosen states *Phys. Rev. Lett.* **69** 2881–4
[Ben+99] Bennett C H, Shor P W, Smolin J A and Thapliyal A V 1999 Entanglement-assisted classical capacity of noisy quantum channels *Phys. Rev. Lett.* **83** 3081–4
[Ben+93] Bennett C H, Brassard G, Crépeau C, Jozsa R, Peres A and Wootters W K 1993 Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels *Phys. Rev. Lett.* **70** 1895–9

[BIM18] Biscani F, Izzo D and Märtens M 2018 esa/pagmo2: pagmo 2.7

[CL18] Cai Z and Liu J 2018 Approximating quantum many-body wave functions using artificial neural networks *Phys. Rev.* B **97** 035116

[CT17] Carleo G and Troyer M 2017 Solving the quantum many-body problem with artificial neural networks *Science* **355** 602–6

[CB08] Chirolli L and Burkard G 2008 Decoherence in solid-state qubits *Adv. Phys.* **57** 225–85

[CKW00] Coffman V, Kundu J and Wootters W K 2000 Distributed entanglement *Phys. Rev.* A **61** 052306

[Cub05] Cubitt T 2005–2017 quantinf. MATLAB package

[Cue+17] De las Cuevas G, Schuch N, Perez-Garcia D and Cirac J I 2017 Continuum limits of matrix product states *Phys. Rev.* B **98** 174303

[DLD17] Deng D-L, Li X and Sarma S D 2017 Quantum entanglement in neural network states *Phys. Rev.* X **7** 021021

[Dev05] Devetak I 2005 The private classical capacity and quantum capacity of a quantum channel *IEEE Trans. Inf. Theory* **51** 44–55

[DHW04] Devetak I, Harrow A W and Winter A 2004 A family of quantum protocols *Phys. Rev. Lett.* **93** 230504

[DS05] Devetak I and Shor P W 2005 The capacity of a quantum channel for simultaneous transmission of classical and quantum information *Commun. Math. Phys.* **256** 287–303

[DSS98] DiVincenzo D P, Shor P W and Smolin J A 1998 Quantum-channel capacity of very noisy channels *Phys. Rev.* A **57** 830

[FNW92] Fannes M, Nachtergaele B and Werner R F 1992 Finitely correlated states on quantum spin chains *Commun. Math. Phys.* **144** 443–90

[FW08] Fern J and Whaley K B 2008 Lower bounds on the nonzero capacity of pauli channels *Phys. Rev.* A **78** 062335

[GD17] Gao X and Duan L-M 2017 Efficient representation of quantum many-body states with deep neural networks *Nat. Commun.* **8** 662

[GP+09] García-Patrón R, Pirandola S, Lloyd S and Shapiro J H 2009 Reverse coherent information *Phys. Rev. Lett.* **102** 210501

[GA16] Gashler M S and Ashmore S C 2016 Modeling time series data with deep Fourier neural networks *Neurocomputing* **188** 3–11

[GF05] Giovannetti V and Fazio R 2005 Information-capacity description of spin-chain correlations *Phys. Rev.* A **71** 032314

[Gla+18] Glasser I, Pancotti N, August M, Rodriguez I D and Cirac J I 2018 Neural-network quantum states, string-bond states, and chiral topological states *Phys. Rev.* X **8** 011006

[GBC16] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)

[GH16] Gosset D and Huang Y 2016 Correlation length versus gap in frustration-free systems *Phys. Rev. Lett.* **116** 1–5

[Goy+15] Goyeneche D, Alsina D, Latorre J I, Riera A and Życzkowski K 2015 Absolutely maximally entangled states, combinatorial designs, and multiunitary matrices *Phys. Rev.* A **92** 032316

[Goy+17] Goyeneche D, Raissi Z, Di Martino S and Życzkowski K 2017 Entanglement and quantum combinatorial designs arXiv:1708.05946 [quant-ph]

[GBR04] Grassl M, Beth T and Rötteler M 2004 On optimal quantum codes *Int. J. Quantum Inf.* **02** 55–64

[Has07] Hastings M B 2007 Entropy and entanglement in quantum ground states *Phys. Rev.* B **76** 035114

[Hay+08] Hayden P, Horodecki M, Winter A and Yard J 2008 A decoupling approach to the quantum capacity *Open Syst. Inf. Dyn.* **15** 7–19

[He+15] He K, Zhang X, Ren S and Sun J 2015 Delving deep into rectifiers: surpassing human-level performance on ImageNet classification *2015 IEEE Int. Conf. on Computer Vision (ICCV)* (IEEE) pp 1026–34

[Hel13] Helwig W 2013 Absolutely maximally entangled qudit graph states arXiv:1306.2879 [quant-ph]

[HC13] Helwig W and Cui W 2013 Absolutely maximally entangled states: existence and applications arXiv:1306.2536 [quant-ph]

[Hel+12] Helwig W, Cui W, Latorre J I, Riera A and Lo H-K 2012 Absolute maximal entanglement and quantum secret sharing *Phys. Rev.* A **86** 052335

[HS00] Higuchi A and Sudbery A 2000 How entangled can two couples get? *Phys. Lett.* A **273** 213–7

[Hor91] Hornik K 1991 Approximation capabilities of multilayer feedforward networks *Neural Netw.* **4** 251–7

[HGS17] Huber F, Gühne O and Siewert J 2017 Absolutely maximally entangled states of seven qubits do not exist *Phys. Rev. Lett.* **118** 200502

[Hub+18] Huber F, Eltschka C, Siewert J and Gühne O 2018 Bounds on absolutely maximally entangled states from shadow inequalities, and the quantum macwilliams identity *J. Phys. A: Math. Theor.* **51** 175301

[IS15] Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *ICML'15: Proc. 32nd Int. Conf. on Int. Conf. on Machine Learning* Vol 37, pp 448–56

[Jia+18] Jia Z-A, Zhang Y-H, Wu Y-C, Kong L, Guo G-C and Guo G-P 2018 Efficient machine-learning representations of a surface code with boundaries, defects, domain walls, and twists *Phys. Rev.* A **99** 012307

[Joh] Johnson S G The nlopt nonlinear-optimization package http://ab-initio.mit.edu/nlopt

[Joh16] Johnston N 2016 QETLAB: A MATLAB toolbox for quantum entanglement version 0.9 http://qetlab.com

[Kar05] Karaboga D 2005 An idea based on honey bee swarm for numerical optimization *Technical Report* TR06 Erciyes University, Engineering Faculty, Computer Engineering Department

[KE95] Kennedy J and Eberhart R 1995 Particle swarm optimization *Proc. IEEE Int. Conf. on Neural Networks* vol 4 pp 1942–8

[KSW19] Khatri S, Sharma K and Wilde M M 2019 Information-theoretic aspects of the generalized amplitude damping channel arXiv:1903.07747 [quant-ph]

[Kol61] Kolmogorov A N 1961 On the representation of continuous functions of several variables by superpositions of contin- uous functions of a smaller number of variables *Dokl. Akad. Nauk SSSR* **108** 179–82

[KSH17] Krizhevsky A, Sutskever I and Hinton G E 2017 ImageNet classification with deep convolutional neural networks *Commun. ACM* **60** 84–90

[LVV15] Landau Z, Vazirani U and Vidick T 2015 A polynomial time algorithm for the ground state of one-dimensional gapped local Hamiltonians *Nat. Phys.* **11** 566–9

[LS15] Latorre J I and Sierra G 2015 Holographic codes arXiv:1502.06618 [quant-ph]

[LLS18a] Leditzky F, Leung D and Smith G 2018 Dephrasure channel and superadditivity of coherent information *Phys. Rev. Lett.* **121** 160501

[LLS18b] Leditzky F, Leung D and Smith G 2018 Quantum and private capacities of low-noise channels *Phys. Rev. Lett.* **120** 160503

[Li+17] Li Y, Han M, Grassl M and Zeng B 2017 Invariant perfect tensors *New J. Phys.* **19** 063029

[Llo97] Lloyd S 1997 Capacity of the noisy quantum channel *Phys. Rev.* A **55** 1613

[Mya+00] Myatt C J, King B E, Turchette Q A, Sackett C A, Kielpinski D, Itano W M, Monroe C W D J and Wineland D J 2000 Decoherence of quantum superpositions through coupling to engineered reservoirs *Nature* **403** 269

[Opq] Open quantum problems https://oqp.iqoqi.univie.ac.at/ Hosted by IQOQI Vienna (Accessed: 20 May 2018)

[Orú14] Orús R 2014 A practical introduction to tensor networks: matrix product states and projected entangled pair states *Ann. Phys.* **349** 117–58

[Pso] Particle swarm optimization algorithm (implementation in MATLAB) https://www.mathworks.com/help/gads/particle-swarm-optimization-algorithm.html (Accessed: 17 May 2018)

[Pas+15] Pastawski F, Yoshida B, Harlow D and Preskill J 2015 Holographic quantum error-correcting codes: toy models for the bulk/boundary correspondence *J. High Energy Phys.* JHEP06(2015)149

[Psm] Pattern search algorithm (implementation in MATLAB) https://www.mathworks.com/help/gads/how-pattern-search-polling-works.html (Accessed: 17 May 2018)

[Pet86] Petz D 1986 Quasi-entropies for finite quantum systems *Rep. Math. Phys.* **23** 57–65

[Phy] Phylliida. Comprehensive list of activation functions in neural networks with pros/cons. Cross Validated https://stats.stackexchange.com/q/154877 (version: 3 August 2017).

[PLB19] Pirandola S, Laurenza R and Banchi L 2019 Conditional channel simulation *Ann. Phys.* **400** 289–302

[Pir+12] Pirvu B, Vidal G, Verstraete F and Tagliacozzo L 2012 Matrix product states for critical spin chains: finite-size versus finite-entanglement scaling *Phys. Rev.* B **86** 075117

[Rai98] Rains E M 1998 Quantum weight enumerators *IEEE Trans. Inf. Theory* **44** 1388–94

[Sai17] Saito H 2017 Solving the Bose–Hubbard model with machine learning *J. Phys. Soc. Jpn.* **86** 093001

[Sco04] Scott A J 2004 Multipartite entanglement, quantum-error-correcting codes, and entangling power of quantum evolutions *Phys. Rev.* A **69** 052330

[Sho02] Shor P W 2002 The quantum channel capacity and coherent information *MSRI Workshop on Quantum Computation (CA, USA)*

[SL97] Shor P and Laflamme R 1997 Quantum analog of the macwilliams identities for classical coding theory *Phys. Rev. Lett.* **78** 1600–2

[Sge] Simple genetic algorithm (implementation in pagmo) https://esa.github.io/pagmo2/docs/cpp/algorithms/sga.html (Accessed: 22 June 2018)

[SS07] Smith G and Smolin J A 2007 Degenerate quantum codes for pauli channels *Phys. Rev. Lett.* **98** 030501

[Sop99] Sopena J M 1999 Neural networks with periodic and monotonic activation functions: a comparative study in classification problems *9th Int. Conf. on Artificial Neural Networks: ICANN '99 (IEE)* vol 1999 pp 323–8

[Sut+17] Sutter D, Scholz V B, Winter A and Renner R 2017 Approximate degradable quantum channels *IEEE Trans. Inf. Theory* **63** 7832–44

[Sva02] Svanberg K 2002 A class of globally convergent optimization methods based on conservative convex separable approximations *SIAM J. Optim.* **12** 555–73

[Tur+00] Turchette Q A, Myatt C J, King B E, Sackett C A, Kielpinski D, Itano W M, Monroe C and Wineland D J 2000 Decoherence and decay of motional quantum states of a trapped atom coupled to engineered reservoirs *Phys. Rev.* A **62** 053807

[VC04] Verstraete F and Cirac J I 2004 Renormalization algorithms for quantum-many body systems in two and higher dimensions arXiv:cond-mat/0407066

[Vid07] Vidal G 2007 Entanglement renormalization *Phys. Rev. Lett.* **99** 220405

[WBS14] Wihler T P, Bessire B and Stefanov A 2014 Computing the entropy of a large matrix *J. Phys. A: Math. Theor.* **47** 245201

[Wil16] Wilde M M 2016 *Quantum Information Theory* 2nd edn (Cambridge: Cambridge University Press)

[Zou+17] Zou W-J, Li Y-H, Wang S-C, Cao Y, Ren J-G, Yin J, Peng C-Z, Wang X-B and Pan J-W 2017 Protecting entanglement from finite-temperature thermal noise via weak measurement and quantum measurement reversal *Phys. Rev.* A **95** 042342