Quantum Error Correction with Quantum Autoencoders

David F. Locher[1,2], Lorenzo Cardarelli[1,2], and Markus Müller[1,2]

[1]Institute for Quantum Information, RWTH Aachen University, D-52056 Aachen, Germany

[2]Peter Grünberg Institute, Theoretical Nanoelectronics, Forschungszentrum Jülich, D-52425 Jülich, Germany

March 6, 2023

Active quantum error correction is a central ingredient to achieve robust quantum processors. In this paper we investigate the potential of quantum machine learning for quantum error correction in a quantum memory. Specifically, we demonstrate how quantum neural networks, in the form of quantum autoencoders, can be trained to learn optimal strategies for active detection and correction of errors, including spatially correlated computational errors as well as qubit losses. We highlight that the denoising capabilities of quantum autoencoders are not limited to the protection of specific states but extend to the entire logical codespace. We also show that quantum neural networks can be used to discover new logical encodings that are optimally adapted to the underlying noise. Moreover, we find that, even in the presence of moderate noise in the quantum autoencoders themselves, they may still be successfully used to perform beneficial quantum error correction and thereby extend the lifetime of a logical qubit.

1 Introduction

Experimental platforms for quantum information processing are unavoidably subject to noise, which can cause failures during quantum computations. The operation of reliable large-scale quantum computers will require active quantum error correction (QEC) procedures in order to cope with errors that dynamically occur during storage and processing of quantum information [1–3]. Quantum error correction relies on redundant encoding of logical quantum information, e.g. into specific multi-qubit states or bosonic modes [3, 4]. Standard qubit-based QEC protocols require measurements of qubits that are coupled to the encoded data, followed by real-time feedback operations. Experimental realizations of quantum error correction have seen great progress and range from repetition [5–7] and error detection codes [8, 9] to recent fault-tolerant implementations [10–13]. While performing in-sequence measurements and real-time feedback is experimentally challenging, it has been achieved in various hardware platforms and application contexts [14–17]. Repeated cycles of quantum error detection and correction are currently studied extensively with superconducting qubits [9, 18–21]. Experimental demonstrations of QEC that include in-sequence measurements and real-time feedback have been achieved in nitrogen-vacancy centers [13, 22], superconducting qubits [23, 24], trapped-ion platforms [12, 25] and bosonic qubits [26, 27]. The correction of errors on the encoded data requires suitable feedback operations based on the obtained measurement results. This task is known as decoding and is a subject of ongoing research [3] that includes also the application of classical neural networks [28–31]. To avoid the challenges posed by in-sequence measurements and feedback, self-correcting quantum memories are investigated [32] and protocols for autonomous corrections have been considered. Previous works in the latter direction include measurement-free QEC [33–35] or engineered dissipation [36–39], e.g. in bosonic codes [40–43]. Moreover, quantum machine learning represents a promising approach towards realizations of autonomous QEC that we want to follow in this work. The field of quantum machine learning is rapidly developing in several directions [44–46] ranging from variational quantum algorithms such as feedforward quantum neural networks (QNNs) [47–51] to quantum associative memories [52–54]. In this work, we focus on a type of multi-layered feedforward QNNs, called quantum autoencoders (QAEs), which have been investigated theoretically for the compression of quantum data [55–59]. Furthermore, QAEs have been proposed to denoise specific quantum states such as GHZ- or W-states [60–62]. Compression of quantum data using QAEs has already been achieved in experiments using single photons [63, 64] or superconducting qubits [65]. In other works, certain types of quantum neural networks were proposed to find suitable encodings of quantum information into logical states that allow for hardware-specific noise to be corrected [66–68].

In this paper we employ quantum autoencoders to perform quantum error correction and explore their utility with a focus on a quantum memory setting. In contrast to most previous works, we envisage QAEs as a flexible and powerful tool to denoise generic states

David F. Locher: d.locher@fz-juelich.de
Lorenzo Cardarelli: l.cardarelli@fz-juelich.de
Markus Müller: markus.mueller@fz-juelich.de

from a logical codespace instead of stabilizing specific quantum states, thereby extending the lifetime of encoded information. Differently from conventional quantum error correction protocols, QAEs are intended to perform the error correction autonomously, requiring neither in-sequence measurements nor classical processing for decoding and feedback. We show how QAEs can be used to correct computational errors on given logical states and also qubit erasures, which can be induced by the loss of qubits or leakage processes. Additionally, we show that QAEs are able to adopt correction strategies that are suited optimally for the noise, which the QNNs are trained for. We furthermore set up and analyze QNNs that can be used to unveil novel logical encodings in an unsupervised manner and without a-priori knowledge about the noise structure. The discovered encodings are optimally suited to protect quantum information against that specific noise. We propose and show that these QNNs can be directly transformed into QAEs ready to perform QEC on the newly discovered states without the need to conduct further training. Lastly, we probe the robustness of the networks when these are constituted by noisy gates. Our results show that even in the presence of moderate levels of intrinsic noise, QAEs can be used for beneficial quantum error correction, to extend the lifetime of a logical qubit.

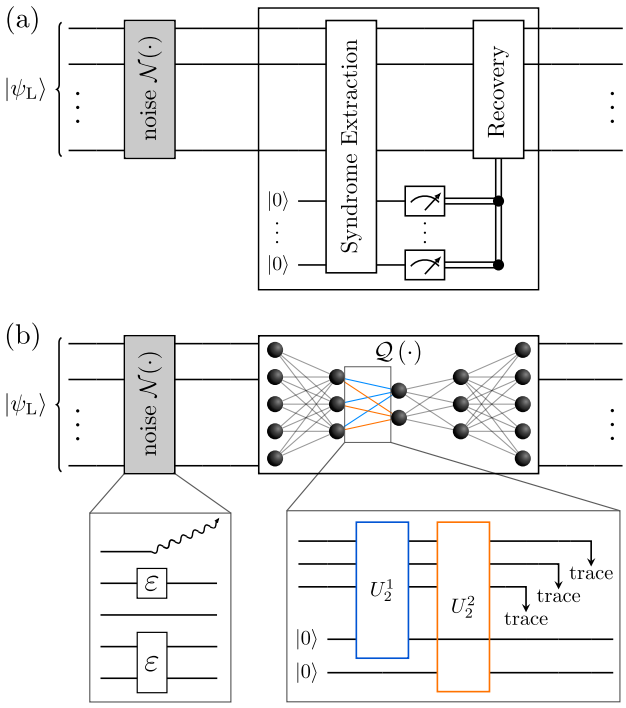


Figure 1: (a) Standard scheme of quantum error correction. Information is encoded in logical states $|\psi_L\rangle$ that undergo a noise process $\mathcal{N}$. Potential errors can possibly be detected by coupling ancilla qubits to the data and measuring the ancillas, yielding the error syndrome. Based on the syndrome an appropriate recovery operation is applied to the data qubits. (b) A quantum autoencoder being used for QEC. Instead of performing syndrome measurements and manually applying recovery operations we employ a QAE to perform the error correction autonomously. The network realizes a quantum channel $\mathcal{Q}$. We use the model of dissipative quantum neural networks to implement the QAE. Nodes in the graph represent individual qubits while edges represent unitary operations. The lower right box illustrates how layer-to-layer transitions are realized in a DQNN. Training the QAE amounts to learning the parameters of the unitary matrices. We find that QAEs can be successfully applied to correct computational errors as well as qubit erasures on logical states.

2 Background on QEC and QNNs

In this section we briefly summarize some basic quantum error correction concepts, which will be useful for the later benchmark of our QAEs against standard QEC codes. Moreover, we review a model of multi-layered feedforward quantum neural networks, known in the literature under the name dissipative quantum neural networks [49, 69, 70]. We then set up quantum autoencoders using this model and discuss how they can be used for QEC purposes.

2.1 Quantum Error Correction

To protect quantum information from errors it is necessary to encode the information redundantly using for instance particular entangled multi-qubit states [2, 71]. For a single encoded qubit, logical states $|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$ belong to the codespace $\mathcal{H}_L$ spanned by two basis states $|0_L\rangle$ and $|1_L\rangle$. Many quantum error-correcting codes are conveniently described in the stabilizer formalism, which uses operators instead of state vector amplitudes to efficiently describe quantum states [72]. An n-qubit stabilizer state is defined as the common $+1$-eigenstate of an Abelian group containing 2^n elements. This stabilizer group is a subgroup of the Pauli group. Without loss of generality, we focus on n physical qubits encoding a single logical qubit. The 2-dimensional logical codespace is defined by a stabilizer group that can be generated from $n-1$ group elements.

Errors occurring on the individual physical qubits can be detected if they map the state out of the logical codespace. The correction of errors is conducted in two steps. Measuring all stabilizer generators first determines for a possibly erroneous state the 2-dimensional subspace, which is orthogonal to the original code space. These measurements are performed by coupling ancilla qubits to the data qubits and measuring the ancillas, as depicted in Fig. 1(a) [2]. The measurements yield a set of ± 1 outcomes that form the error syndrome. A non-trivial syndrome indicates the occurrence of errors on the underlying logical state. In a second step, potential errors must be removed, mapping the state back to the logical codespace $\mathcal{H}_L$. This is achieved by applying a suit-

able recovery operation to the data qubits being determined from the error syndrome in the process of decoding. Different types of noise occurring on logical states may lead to different decoding strategies in order to achieve optimal error correction results [3].

2.2 Dissipative Quantum Neural Networks

A class of quantum feedforward neural networks having attracted attention in the last years are dissipative quantum neural networks (DQNNs) [49, 60, 69, 70]. A DQNN can be represented as a graph consisting of neurons arranged in subsequent coupled layers, as depicted in Fig. 1(b). A layer k of the network consists of n_k neurons that represent individual qubits. The network as a whole realizes a quantum channel $\mathcal{Q}$ that maps an input state ρ_{in} defined on the qubits of the first layer to an output state $\rho_{\text{out}} = \mathcal{Q}(\rho_{\text{in}})$ on the last layer. Each transition from a layer $k-1$ to a layer k realizes an individual map $\mathcal{E}_k$. The full network channel is the concatenation of all layer-to-layer maps:

$$\mathcal{Q}(\rho_{\text{in}}) = \mathcal{E}_{\text{out}}\left(\dots \mathcal{E}_3\left(\mathcal{E}_2\left(\rho_{\text{in}}\right)\right)\dots\right). \tag{1}$$

We regard the input layer as the first network layer, thus the layer-to-layer maps start at $\mathcal{E}_2$. All constituent maps are adjustable via a finite set of parameters which can be chosen such that the network implements a map $\mathcal{Q}^*$ which achieves a desired task. Training a network refers to the process of gradually adjusting the network parameters to eventually attain the target map. Thus, DQNNs are set up similarly to classical feedforward neural networks [73], however, they implement quantum channels instead of maps on classical data. The training of a DQNN is realized in a quantum-classical hybrid procedure: the network is implemented on actual quantum hardware while the optimization of the network parameters is performed on classical hardware. For supervised learning, the training of a feedforward neural network requires training pairs in the form $\{(\rho^i_{\text{in}}, \rho^i_{\text{targ}})\}$, where states ρ^i_{in} serve as input states for the network that one wants to be mapped to corresponding target states ρ^i_{targ}. To quantify the success of the neural network in achieving this task, a cost function is defined which assumes its minimal value if the output states $\rho^i_{\text{out}} = \mathcal{Q}(\rho^i_{\text{in}})$ equal the corresponding target states. A natural choice for a cost function is the averaged infidelity between training input and target states,

$$C = 1 - \frac{1}{N}\sum_{i=1}^{N} \mathcal{F}(\rho^i_{\text{out}}, \rho^i_{\text{targ}}), \tag{2}$$

where the fidelity between two quantum states ρ_1 and ρ_2 is defined as

$$\mathcal{F}(\rho_1, \rho_2) = \left(\text{Tr}\sqrt{\sqrt{\rho_2}\rho_1\sqrt{\rho_2}}\right)^2. \tag{3}$$

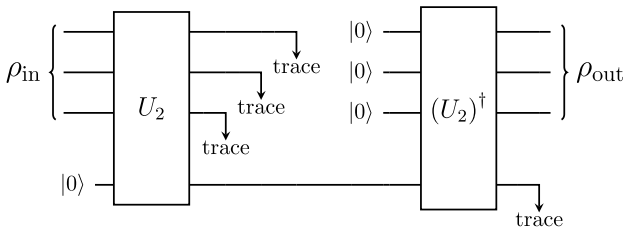


Figure 2: Quantum circuit that realizes a 3-1-3 QAE utilizing an architecture that we call *self-inverse* architecture. Using this ansatz the decoding channel implementing the transition from the single-qubit hidden layer to the 3-qubit output layer is set up from the inverse of the unitary matrix implementing the encoding channel. Compared to independently trained channels this ansatz leads to a reduction of training parameters.

The quantum hardware is thus used to map training input states to output states, which are then measured to evaluate the cost function. Classical optimization routines, such as widely used gradient descent algorithms, can then be used to find an updated set of network parameters that reduces the cost. Updating the network parameters and repeating this cycle eventually leads to a convergence of the cost. In this work we simulate the DQNNs on a classical computer. This allows us to apply an efficient training algorithm similar to a backpropagation algorithm known from classical machine learning. We sketch it briefly in Appendix B and refer to Ref. [49] for a detailed description.

We now describe how the layer-to-layer maps are realized in DQNNs. The graph representation of a DQNN can straightforwardly be translated into a quantum circuit, as indicated in Fig. 1(b). A layer-to-layer map $\mathcal{E}_k$ is implemented as follows. Layer $k-1$ of the network, consisting of n_{k-1} neurons, represents a quantum state ρ_{k-1}. This state is supplemented with new qubits in the state $|0\rangle^{\otimes n_k}$. A unitary matrix U_k is then applied to the qubits of both adjacent layers. Afterwards, qubits belonging to layer $k-1$ are discarded, resulting in a quantum state ρ_k on the k-th layer of the network [49]:

$$\rho_k = \mathcal{E}_k\left(\rho_{k-1}\right) = \underset{k-1}{\text{Tr}}\left[U_k\left(\rho_{k-1}\otimes|0\rangle\langle 0|^{\otimes n_k}\right)U_k^\dagger\right]. \tag{4}$$

The trace operation conducted in the maps gives rise to the term *dissipative* QNNs. The unitary operators U_k mediating the layer-to-layer transitions are the trainable quantities in this model. To reduce the number of training parameters or allow for an easy execution of the network map on actual hardware one may choose to set up the unitary matrices in various ways. Beer *et al.* [49] suggested to build an operator U_k from n_k individual unitary matrices U^j_k, each acting on all qubits in layer $k-1$ and a single qubit j in layer k: $U_k = U^{n_k}_k \dots U^1_k$. This explicit realization is shown in Fig. 1(b) and we adopt this approach in

our work. In an experiment it is often more practical to specify a parameterized ansatz for the unitaries, consisting of natively executable gates [70], instead of training completely generic unitary matrices.

2.3 Quantum Autoencoders

Autoencoders (AEs) are technically defined as feed-forward neural networks that are trained to reproduce their inputs at the output layer [73]. Typically, they comprise a hidden layer of width smaller than the input and output layers, meaning that some information must be discarded as the input states are processed. Such networks are called undercomplete autoencoders. Undercomplete AEs consist of two parts: a so-called encoder $\mathcal{E}$ maps the input to a latent state of smaller dimension. A decoder[1] $\mathcal{D}$ then tries to reconstruct the input from the latent state. The full map $\mathcal{Q}$ describing the action of the autoencoder is thus a concatenation of the encoder and the decoder map: $\mathcal{Q}(\cdot) = \mathcal{D}(\mathcal{E}(\cdot))$. Autoencoders can e.g. be used for data compression [74]. An undercomplete AE that succeeds to reproduce certain input data at the output layer is able to perform a lossless compression and reconstruction of the input. The latent states can thus be considered as compressed data which are found in an unsupervised manner since no compressed reference states have to be provided for training. Furthermore, AEs can be employed for denoising of data [75]. When an AE is trained to map noisy samples of the training data to noise-free instances, the network might learn to remove the noise and keep the relevant information while compressing the data. Noise-free samples can then be reconstructed at the output layer.

Quantum autoencoders are defined equivalently to their classical counterparts: In the quantum case, the input and output states are quantum states and the network realizes a quantum map. Just as classical AEs they can be split up into an encoding channel and a decoding channel, applied one after another.

In this work we employ denoising QAEs in the setting depicted in Fig. 1(b). We consider arbitrary states $|\psi_{\mathrm{L}}\rangle$ from a predefined codespace $\mathcal{H}_{\mathrm{L}}$ being affected by noise to serve as input states for a QAE. We first assume that the network dynamics is noise-free, while in Sec. 5 we discuss the generalization to noise occurring during the application of the network. The ultimate goal is that the QAE discards eventual errors while keeping the encoded quantum information as the noisy input states are processed. To achieve this, we apply a supervised learning scheme using a small number of logical states $|\psi_{\mathrm{L}}^i\rangle$ from the code space. We take noisy states $\mathcal{N}(|\psi_{\mathrm{L}}^i\rangle\langle\psi_{\mathrm{L}}^i|)$ as training inputs and the corresponding noise-free logical states as target states for the training. The cost function therefore reads

$$C = 1 - \frac{1}{N}\sum_{i=1}^{N} \langle\psi_{\mathrm{L}}^i|\, \rho_{\mathrm{out}}^i\, |\psi_{\mathrm{L}}^i\rangle\,, \tag{5}$$

with output states $\rho_{\mathrm{out}}^i = \mathcal{Q}(\mathcal{N}(|\psi_{\mathrm{L}}^i\rangle\langle\psi_{\mathrm{L}}^i|))$.

In general, every layer-to-layer map of a DQNN is realized by independent unitary matrices that are adjusted during the training process. However, the special form of a QAE consisting of an encoding and a decoding channel allows for a simpler ansatz. Inspired by Ref. [55] we may choose to not train the encoding and decoding channels independently but set up the decoder using the inverse matrices from the encoder, as depicted in Fig. 2. In the remaining part of the paper we will refer to this ansatz as *self-inverse architecture*. The self-inverse architecture certainly leads to a reduction of training parameters and comes with additional advantages that will be highlighted in Sec. 3.

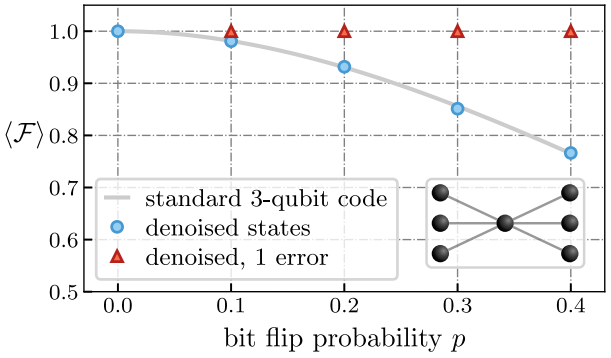


Figure 3: Validation of several 3-1-3 QAEs trained on logical states of the 3-qubit repetition code subjected to bit flip noise. To test the performance of a QAE, 10^4 randomly drawn logical states are subjected to Pauli X-errors occurring independently on any qubit with probability p. The corresponding QAE having been trained on noise strength p is then used to correct the errors on those states. The plot shows the averaged fidelity of denoised states w.r.t. the corresponding noise-free logical states. We find that the QAEs (blue circles) perform exactly as well as the standard 3-qubit repetition code (grey line), $\langle\mathcal{F}\rangle = 1 - \frac{2}{3}p_{\mathrm{L}}$ (see Appendix A). A closer analysis reveals that the QAEs learn to perfectly correct single X-errors (red triangles). Here and in the remainder of the paper we omit error bars because the statistical errors are smaller than the symbol sizes. The inset shows a sketch of the 3-1-3 QAE geometry used in this example.

3 Quantum Error Correction Results

In this section we present numerical results demonstrating that QAEs can be successfully used to perform quantum error correction. In particular, we show that QAEs can correct both computational errors and qubit loss (quantum erasures) occurring on logical states of quantum error-correcting codes.

[1] The term decoding in the context of autoencoders is not to be confused with the terminology of decoding as it is used in QEC.

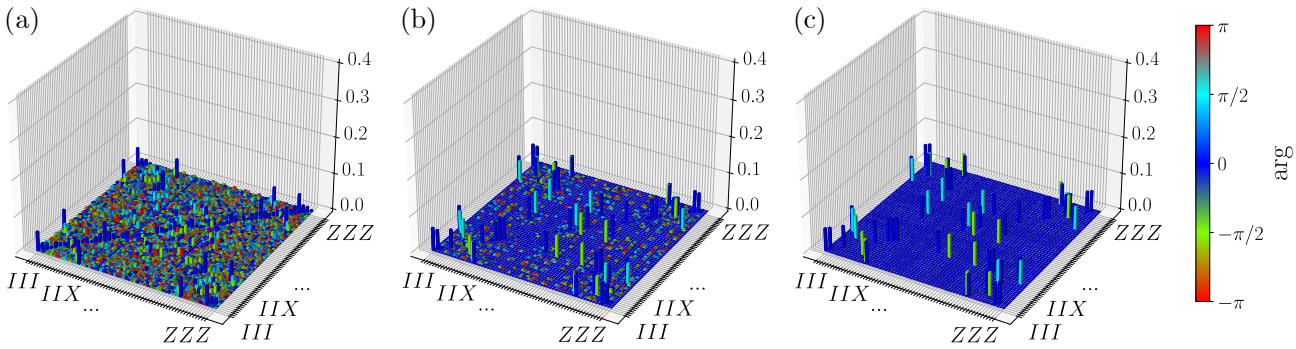


Figure 4: Quantum process tomographies of the maps which a 3-1-3 QAE, trained on bit flip noise of strength $p = 0.1$, implements after 20 (a), 50 (b) and 100 (c) training epochs. As the training progresses, the quantum channel realized by the quantum neural network converges towards a final map which corresponds to the quantum process that realizes the correction map of the standard 3-qubit code. Details are given in Appendix A.

3.1 Correction of Computational Errors

3.1.1 Correction of bit flips on 3 qubits

The 3-qubit repetition code (3QC) is a quantum error-correcting code able to correct single bit flip errors [2]. Here, we take the 3QC as an illustrative example to demonstrate that QAEs can be successfully trained and applied for QEC. One may choose the logical codespace to be spanned by the states $|0_\mathrm{L}\rangle = |000\rangle$ and $|1_\mathrm{L}\rangle = |111\rangle$. This space is stabilized by a group generated e.g. from the operators Z_1Z_2 and Z_2Z_3. The assignment of logical basis states above fixes the logical generators of the code to be $X_\mathrm{L} = X_1X_2X_3$ and $Z_\mathrm{L} = Z_1Z_2Z_3$, up to multiplication by elements of the stabilizer group [2, 72]. A bit flip error happening to one of the physical qubits can be detected by measuring the two stabilizer generators. Two bits of information allow for four different syndromes to be distinguished, corresponding to the noise-free case and the three different single-qubit X-errors. Removing the respective bit flip corresponds to an appropriate recovery operation. Bit flips occurring on two qubits simultaneously, e.g. X_1X_2, are misinterpreted as single bit flip errors on the complementary qubit, X_3 for this example. Therefore, a correction attempt causes a third bit flip, inducing a logical error $X_\mathrm{L} = X_1X_2X_3$ on the state. The presence of bit flip noise can be modelled via the bit flip channel which for a single qubit reads

$$\mathcal{N}_p^{\mathrm{bit}}(\rho) = (1-p)\rho + pX\rho X. \tag{6}$$

A 3QC state $|\psi_\mathrm{L}\rangle$ that is subjected to independent bit flip noise suffers no error with probability $(1-p)^3$, a single bit flip with probability $3p(1-p)^2$ and two or three flips with probabilities $3p^2(1-p)$ and p^3, respectively [2]. Since single bit flip errors are correctable on logical states of the 3QC, active error correction on a noisy state induces a logical bit flip with probability $p_\mathrm{L} = 3p^2(1-p)+p^3$ and recovers the noise-free state with prob. $1-p_\mathrm{L}$.

We start our study by training 3-1-3 QAEs, i.e. quantum neural networks with a 3-qubit input layer, a single-qubit hidden layer and a 3-qubit output layer, on logical states of the 3QC. A sketch of such a network is shown in Fig. 3. For the training process we consider the three states $|0_\mathrm{L}\rangle$, $|1_\mathrm{L}\rangle$ and $|+_\mathrm{L}\rangle$ which turn out to be enough training states for the network to learn to successfully generalize to arbitrary code states. These states are subjected to bit flip noise occurring independently on the three physical qubits with probability p. Concretely, we apply the corresponding noise channel to the states, meaning that mixed states $\mathcal{N}(|\psi_\mathrm{L}\rangle\langle\psi_\mathrm{L}|)$ are taken as inputs for the QAEs. As described in Sec. 2.3, the corresponding noise-free logical states are considered as target states. As a benchmark, Fig. 3 compares the trained QAEs and the standard 3QC in terms of the correction performance. Every value of p corresponds to a different QAE that has been trained on logical states subjected to noise of this strength. A standard procedure in machine learning tasks is to test a trained neural network on data which has not been used for the training process, called validation. Thus, for every trained QAE we randomly draw 10^4 pure logical states that uniformly cover the logical Bloch sphere. These states are subjected to Pauli X-errors occurring independently on every qubit with probability p to then serve as validation input states for a QAE. In an actual experiment, a network would be trained and run on the same platform, likely exposed to the same error conditions. Hence, at both the training and testing stage we apply noise of equal type and strength. At the validation stage we want to investigate how the QAE handles states that suffered no error, a single bit flip etc. Therefore, we use states $E_i|\psi_\mathrm{L}\rangle$ as validation input states, where the set $\{E_i\}$ consists of all combinations of bit flip errors from which we draw errors with appropriate probabilities. The averaged fidelity of denoised test states w.r.t. the corresponding noise-free test states serves as a measure for the performance of the trained QAE. For various noise strengths we find that the trained quantum networks perform just as well as the standard 3QC which corrects any single-qubit X-errors. A closer analysis reveals that

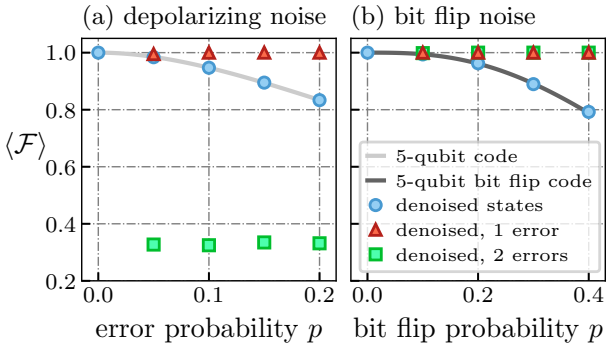


Figure 5: (a) Validation of several 5-1-5 QAEs trained on depolarizing noise. To test the performance of a network, 5×10^4 randomly drawn logical states of the 5-qubit code are subjected to Pauli errors occurring independently on every qubit with probability p. The QAE having been trained on the corresponding noise strength is then used to correct the errors on those states. The plot shows the averaged fidelity of denoised states w.r.t. the noise-free validation states. We find that the QAEs, indicated by the blue circles, perform exactly as well as the standard 5-qubit error correction code represented by the grey line. A closer analysis reveals that the QAEs learn to perfectly correct single-qubit Pauli errors, as can be seen from the red triangles. (b) Validation of 5-1-5 QAEs trained on bit flip noise. Each network is tested on 10^4 randomly drawn logical states subjected to independent X-errors. We find that the networks perform as well as a 5-qubit bit flip code, correcting all single- and two-qubit X-errors. Thus, we see that QAEs adopt different error correction strategies depending on the noise that is present during the training process.

the networks for any $p < 0.5$ but $p \neq 0$ indeed learn to implement a channel that perfectly corrects any single bit flip error on arbitrary logical states, indicated by the red triangles in Fig. 3. Hence, it is sufficient to train a QAE for one non-zero value of p to learn a correction strategy that successfully generalizes to other noise strengths $p < 0.5$. By performing quantum process tomographies of the maps $\mathcal{Q}$ realized by the fully trained QAEs, we are able to show that the learned channels are equivalent to the correction map of the standard 3-qubit code. Figure 4 shows quantum process tomographies of the maps that a 3-1-3 QAE implements at different stages of the training. After 100 training epochs there are no deviations of the quantum process from the map realizing the standard 3-qubit repetition code. Details on the quantum process tomographies can be found in Appendix A.

We sometimes observe a failure of the training process, manifesting itself in a channel $\mathcal{Q}$ that maps arbitrary input states to a fixed state such as $|000\rangle$. The training parameters realizing this map seem to correspond to a saddle point of the cost function, which is hard to escape using standard gradient descent methods. However, employing the self-inverse architecture, shown in Fig. 2, which reuses the unitary matrices from the encoding channel for the decoding channel we do not observe these failures when training 3-1-3 QAEs.

Considering QAEs whose latent space consists of a single qubit, it is clear that the first part of the network must conduct a combined correction and compression of erroneous logical states. The network decoder then performs the trivial task of re-encoding a logical state. Any errors left on the single-qubit intermediate state necessarily lead to logical errors on the final logical state. However, there exists a gauge freedom in the sense that the computational basis of the single-qubit state on the intermediate network layer can be arbitrarily rotated. The network encoder can thus map an input state $|\psi_\mathrm{L}\rangle$ to a state $R|\psi\rangle$ with R being an arbitrary single-qubit rotation as long as the network decoder reconstructs the desired logical state $|\psi_\mathrm{L}\rangle$ from $R|\psi\rangle$. In Appendix A we visualize this feature by means of quantum process tomographies.

3.1.2 Correction of arbitrary computational errors

To be able to correct Pauli X-, Y- and Z-errors occurring on physical qubits in a quantum memory one has to employ an encoding that uses at least five qubits [2]. We consider logical states of the 5-qubit error-correcting code [76] being generated by the stabilizer elements

$$\begin{aligned} g_1 &= XZZXI \\ g_2 &= IXZZX \\ g_3 &= XIXZZ \\ g_4 &= ZXIXZ. \end{aligned} \tag{7}$$

Logical states are $+1$-eigenstates of the operators g_1 to g_4 and the logical generators of the single encoded qubit may be chosen as follows:

$$X_\mathrm{L} = XXXXX, \quad Z_\mathrm{L} = ZZZZZ. \tag{8}$$

The 5-qubit code is the smallest distance-3 code which means that it can correct an arbitrary Pauli error happening to one of the physical qubits. It satisfies the quantum Hamming bound: four stabilizer generators allow for $2^4 = 16$ different error syndromes to be distinguished, corresponding to the 15 different single-qubit Pauli errors and the error-free case [2].

To further investigate the capabilities of quantum autoencoders to perform QEC we train QAEs with a 5-1-5 geometry and utilize them to correct errors on logical states of the 5-qubit code. We set up the QAEs employing the self-inverse architecture introduced in Sec. 2.3. Thus, the encoding channel is mediated via a single trainable 6-qubit unitary matrix while the decoding is realized using the inverse of that matrix. As training input states we employ the six logical X-, Y- and Z-eigenstates of the 5-qubit code subjected to depolarizing noise. The depolarizing channel for a single qubit reads

$$\mathcal{N}_p^\mathrm{depol}(\rho) = (1-p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z) \tag{9}$$

which we apply independently to the five physical qubits. Fig. 5(a) shows a validation of the trained QAEs. Every value of p corresponds to a different QAE whose performance is tested by exposing it to randomly drawn logical states subjected to random Pauli errors according to independent depolarizing noise of strength p. For various values of p we find that the QAEs perform just as well as the standard 5-qubit error correction code that corrects arbitrary single-qubit Pauli errors. Analyzing the action of those networks on different classes of errors exhibits that QAEs trained on $p \neq 0$ in fact implement a channel that perfectly corrects any single-qubit Pauli error. Pauli errors of weight two are, however, not correctable.

3.1.3 Adaptability to different types of noise

To investigate whether QAEs can learn different error correction strategies in the presence of different types of noise, we consider again QAEs with a 5-1-5 geometry. However, opposed to the previous example, we now train them on logical states that are subjected to solely bit flip noise. As can be seen from a validation of these networks in Fig. 5(b), the QAEs perform just as well as a five-qubit bit flip code. We find that the QAEs learn to perfectly correct up to two X-errors on arbitrary logical states. This illustrates that QAEs can learn various error correction strategies depending on the noise suffered by logical states during the training process.

In experimental quantum information processing devices, noise can be correlated in space and time [77, 78]. We want to study whether, in the presence of spatially correlated bit flip noise, QAEs can adopt correction strategies that perform better than the standard approach. To do so, we go back to logical states of the 3-qubit repetition code, subjected to correlated bit flip noise. We describe the noise by two parameters: an overall bit flip probability p and a correlation parameter η [78]. Choosing two qubits A and B, the correlation parameter is defined as

$$\eta = \frac{\Pr(\text{flip on A}|\text{B flipped})}{\Pr(\text{flip on A}|\text{B not flipped})}. \tag{10}$$

For simplicity, for any three qubits A, B and C we consider

$$\begin{aligned} &\Pr(\text{flip on A}|\text{B flipped and C flipped}) \\ &= \Pr(\text{flip on A}|\text{B flipped, C not flipped}), \end{aligned} \tag{11}$$

i.e. the probability for one of them to be flipped is the same, irrespective of whether one or both of the other qubits have suffered an error. The case $\eta = 1$ corresponds to uncorrelated bit flip noise, whereas $\eta > 1$ describes bunching of errors, meaning that bit flips tend to occur in pairs. Antibunching is characterized by $\eta < 1$. We train 3-1-3 QAEs on logical states of the 3-qubit repetition code suffering bit flips with fixed probability $p = 0.2$ but with different correlation strengths. We find that networks trained in the presence of correlations $\eta > 4$ implement a different correction map than networks that were trained on small correlations. This becomes clear from Fig. 6, where the performance of several trained QAEs is compared to the performance of the standard 3-qubit code and an alternative 3-qubit error correction strategy. The alternative correction strategy considers flips of two qubits as most likely error events, therefore correcting those while inducing a logical error for single bit flips. Indeed, if the correlation strength is increased beyond a certain threshold, it is advantageous to correct any two errors instead of single ones. Abbreviating the conditional probability Pr(flip on A|B flipped) as p_c, the turnover point resides at $p_c = 0.5$. From Bayes' theorem follows that Pr(flip on A|B not flipped) $= \frac{p(1-p_c)}{1-p}$, so we find the critical correlation strength to be $\eta_c = \frac{1-p}{p}$. During the training process, a QAE correctly determines and implements the correction strategy which yields the best error correction results.

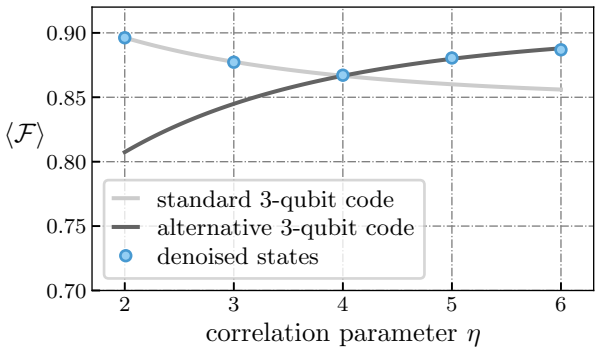


Figure 6: Validation of several 3-1-3 QAEs trained on logical states of the 3-qubit repetition code subjected to correlated bit flip noise with fixed overall bit flip probability $p = 0.2$ and varying correlation strength η. For $\eta < \frac{1-p}{p}$ the probability for a single bit flip error to occur is larger than the probability of two bit flips happening on a state and vice versa for $\eta > \frac{1-p}{p}$. Thus, for the overall error rate $p = 0.2$ the optimal error correction strategy is different for $\eta < 4$ and $\eta > 4$, denoted as standard / alternative 3-qubit code. QAEs automatically adopt the best possible denoising strategy during the training process. Each QAE is tested on 10^4 randomly drawn validation states.

These findings demonstrate that QAEs can adopt error correction strategies that are optimally suited for the type of noise present during training. We saw this at the example of QAEs adapting to depolarizing and bit flip noise or QAEs adjusting their correction strategy in the presence of correlations. In an experiment, a QAE would be trained on the device on which it is supposed to perform the error correction later on, thus implementing an optimal denoising strategy for the specific device. In the canonical scheme of syndrome-based quantum error correction the different strategies which a QAE can embrace correspond

to different decodings of syndromes into recovery operations.

3.2 Correction of Erasures and Computational Errors

Besides computational errors, losses and leakage errors pose a threat for successful quantum computations [79]. Ions or neutral atoms for example might escape the trapping potential in atomic systems [80, 81] or photons can get lost from a photonic quantum processor [82]. Moreover, leakage into states which are not part of the two-dimensional qubit subspace represents a risk, e.g. in superconducting [83–85] or atomic [86] quantum processors. The quantum erasure channel, which for a single qubit reads

$$\mathcal{N}_p^{\text{erasure}}(\rho) = (1-p)\rho + p\,|2\rangle\langle 2|\,, \tag{12}$$

is used frequently to model losses or incoherent leakage errors. The positions of possible erasures can be detected in experiments by performing quantum nondemolition measurements. These signal the occurrence of potential erasures while leaving the quantum state invariant if no erasures have happened. Such detection protocols have been proposed or even implemented for various architectures [80, 84, 87]. To be able to protect a logical qubit from single erasures, a code consisting of at least four physical qubits is necessary [79]. Moreover, any complete code of distance d can be used to correct $d-1$ erasures or located errors [79].

Here, we want to use QAEs to correct possible losses of physical qubits from logical quantum states. We consider the quantum erasure channel, thus, the positions of losses are known. Note that we restrict the investigation to erasures occurring on states before they enter a DQNN. Since erasures of different qubits are classically distinguishable, it is possible to apply a different recovery map for the correction of every possible erasure event. In this work we thus use a separate QAE for every possible loss. Together, these QAEs form a collection of networks, as depicted in Fig. 7. The individual QAEs from the collection have

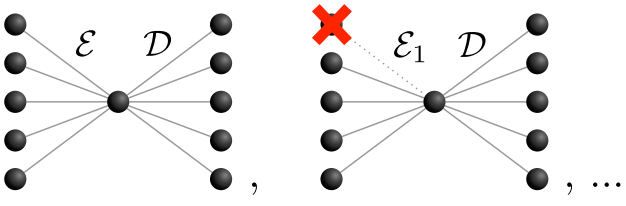


Figure 7: Collection of QAEs used to correct erasures of qubits. Since the positions of erasures are known, every possible combination of erasures is corrected by a separate QAE. States $\mathrm{Tr}_1\left(|\psi_L\rangle\langle\psi_L|\right)$ for example are processed by a QAE consisting of the channels $\mathcal{E}_1$ and $\mathcal{D}$. All the networks from the collection are trained separately on the respective erroneous states.

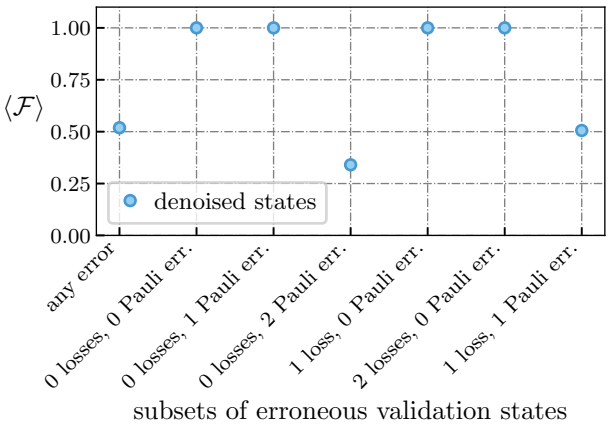


Figure 8: Validation of an x-1-5 collection of QAEs trained to denoise logical states of the 5-qubit code undergoing erasures with probability $p_{\text{loss}} = 0.4$ followed by depolarizing noise occurring independently on every qubit with probability $p_{\text{comp}} = 0.1$. The QAEs learn to correct arbitrary single-qubit Pauli errors and up to two arbitrary erasures of qubits. The collection of networks is tested on 5×10^4 randomly drawn validation states subjected to errors as outlined above.

to be trained separately on the respective erroneous states. We model erasures by tracing over the corresponding qubits of a logical state, leaving behind a state that is generally mixed. Networks that are used to correct single erasures on logical states of a code consisting of n physical qubits thus implement channels mapping $(n-1)$-qubit states to n-qubit states. In the following we will employ the self-inverse architecture to train the network that processes states which did not suffer any erasures. The trained decoding channel $\mathcal{D}$ will then be reused for all other networks of the collection such that they differ only with regard to the encoding channels $\mathcal{E}_i$, as indicated in Fig. 7.

As a minimal example we train a collection of x-1-4 QAEs on logical states of the 4-qubit erasure code being subjected to losses. We find that after sufficient training the networks learn to perfectly correct any single erasure while failing to correct two or more losses. In experimental devices for quantum information processing, erasures and computational errors occur side by side. Thus, as a further example we train a collection of x-1-5 QAEs to correct errors on logical states of the 5-qubit error-correcting code. The states suffer independent losses of qubits with probability $p_{\text{loss}} = 0.4$ followed by depolarizing noise of strength $p_{\text{comp}} = 0.1$. Fig. 8 shows the performance of the x-1-5 QAEs to correct errors on randomly drawn logical states subjected to losses and Pauli errors, with the same probabilities as for the training process. In particular, the plot shows denoising results for different subsets of errors. We see that the QAEs learn to correct single Pauli errors as well as any single or double erasure. This confirms our expectation regarding what is possible to achieve on logical states of the 5-qubit error-correcting code.

4 Encoding Discovery

In the previous section, we showed that quantum autoencoders are able to discover optimal strategies to correct computational errors and erasures on states from a predefined logical codespace. However, the denoising capabilities of QAEs are fundamentally limited by the logical encoding defined in advance. To allow for more flexible denoising strategies it would be desirable to search for entirely new logical encodings that are optimally suited to protect quantum information from unknown types of noise. Some schemes have already been proposed on how quantum neural networks can be used to achieve this [66–68]. Here, we choose quantum neural networks in the spirit of overcomplete autoencoders as shown in Fig. 9(a) to solve this task. The networks are trained to reconstruct the single-qubit input states at the output layer while noise is present in the interior of the network. In a first step, a trainable channel $\mathcal{D}$ maps the single-qubit input states to unspecified logical states: $\rho_{\mathrm{L}} = \mathcal{D}(|\psi\rangle\langle\psi|)$. The logical states are subjected to noise $\mathcal{N}$, corresponding to decoherence while the states are stored in memory. The subsequent channel $\mathcal{E}$ is trained to map these states back to the original single-qubit inputs. Thus, the network finds a suitable logical encoding that allows for errors induced by the intermediate noise channel to be corrected. Such an optimal encoding is found in an unsupervised manner. We propose then to rearrange the channels $\mathcal{D}$ and $\mathcal{E}$ of a trained network to form an undercomplete QAE with a single neuron on the central layer, as sketched in Fig. 9(b). This QAE is ready to perform QEC on logical states defined by the newly discovered encoding rule given by $\mathcal{D}$, without the need to perform further training. Exposing states ρ_{L} to noise and feeding those to the new QAE results in corrected logical states at the output layer.

To investigate whether the proposed QNNs can actually find suitable logical encodings and correction strategies we consider the following error model. The qubits of a state are subjected to spatially correlated dephasing noise followed by possible erasures. The collective dephasing arises from coherent Z-rotations of all qubits in the register, occurring probabilistically:

$$\mathcal{N}^{\mathrm{coll.deph.}}(\rho) = \int p(\alpha) U(\alpha) \rho U(\alpha)^{\dagger} \mathrm{d}\alpha, \tag{13a}$$

where

$$U(\alpha) = e^{-i\frac{\alpha}{2}\sum_n Z_n} \tag{13b}$$

and the quantity $p(\alpha)$ describes a probability distribution. Collective dephasing of idling qubits is a relevant type of noise in experimental setups of quantum processors. For instance, it occurs in ion-trap devices because of global fluctuations of the magnetic field strengths [88]. Collective dephasing noise can have detrimental effects on stored quantum information such as enhanced decoherence of entangled multiqubit states [88–90]. However, quantum information can be perfectly protected from collective dephasing noise by encoding logical states in a decoherence-free subspace (DFS) [91–93].

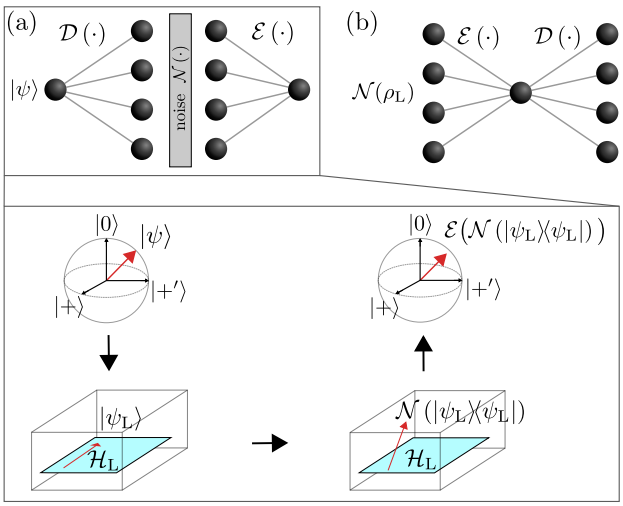


Figure 9: (a) Sketch of a quantum neural network that is used to unveil novel logical encodings protecting quantum information from noise $\mathcal{N}$. The network is trained to reproduce input states $|\psi\rangle$ at the output layer. The intermediate states $\mathcal{D}(|\psi\rangle\langle\psi|) = \rho_{\mathrm{L}}$ correspond to a logical encoding of the input information that is learned in an unsupervised manner. Those states are subjected to noise $\mathcal{N}$ whereupon the channel $\mathcal{E}$ conducts a combined error correction and compression of the possibly faulty logical states. A good retrieval of the original single-qubit states will be possible if the network finds a logical encoding $\rho_{\mathrm{L}} = |\psi_{\mathrm{L}}\rangle\langle\psi_{\mathrm{L}}|$ that is well suited to deal with the present noise. (b) Interchanging the encoder and the decoder of the network in (a) gives rise to an undercomplete QAE that can be used to perform quantum error correction on erroneous logical states $\mathcal{N}(\rho_{\mathrm{L}})$.

Here, we train a collection of 1-4-1 QNNs, based on the sketch in Fig. 9(a), and expose the 4-qubit states in the center of the quantum networks to collective dephasing noise according to Eq. (13), where $p(\alpha)$ is chosen to be a centered Gaussian with unit variance. Moreover, erasures of single qubits may occur on the intermediate states, triggering one of the channels $\mathcal{E}_i$, $i = 0, \ldots, 4$ from the collection to perform the combined compression and correction of faulty states. We employ the self-inverse architecture, meaning that the dissipative channel $\mathcal{E}_0$ embeds the inverse of the 5-qubit unitary matrix realizing the map $\mathcal{D}$. We find that the collection of quantum networks learns to encode logical states in a DFS. Moreover, the encoding allows for single erasures of qubits to be corrected. Logical states are thus perfectly protected from collective dephasing and partially protected from losses of qubits. In Appendix D we show the precise form of the discovered logical states and discuss the numerical results in more detail.

5 Robustness Against Internal Noise

So far, we have assumed that noise only acts on incoming qubits while the DQNNs themselves operate perfectly. This is, however, an idealization. In general, the application of gates as well as idling of qubits during a computation will introduce errors on the quantum states operated. In this work we focus on a quantum memory setting, i.e. stored quantum information to be protected from noise. One goal is to extend the lifetime of a logical qubit beyond the lifetime of a bare physical qubit. Reaching this "break-even" point is a present-day challenge. Using QEC to extend the lifetime of an encoded qubit has been demonstrated experimentally with bosonic codes [26, 27]. In other works, active quantum error correction was shown to be advantageous in some specific noise parameter regimes [7].

To assess whether we can use an intrinsically noisy QAE to extend the lifetime of an encoded qubit, we employ a measure proposed and discussed in Ref. [94], as sketched in Fig. 10. The goal is to protect quantum information from environmental decoherence for a time interval τ. To do so, one can either use a quantum state $|\psi\rangle$ on a bare physical qubit or decide to encode the information into a logical multi-qubit state $|\psi_{\mathrm{L}}\rangle$. In any case, all physical qubits are subjected to noise while being stored in memory. Now, the question arises whether it is beneficial to apply a round of imperfect quantum error correction to the encoded state after, say, half of the memory time, to correct errors that have accumulated thus far. It is not surprising that a very noisy QEC device rather deteriorates the encoded state than improving it. A "good" QEC device, however, can actually yield an advantage, as compared to doing nothing. To assess the usefulness of a noisy QEC device we therefore compare three scenarios. One starts either with a single-qubit quantum state $|\psi\rangle$ or an encoded logical state $|\psi_{\mathrm{L}}\rangle$. All qubits are subjected to noise $\mathcal{N}_{p_{\mathrm{i}}}$ caused by environmental decoherence while being stored in memory for a time $\tau/2$. Now, one can apply a round of imperfect QEC to the encoded state, assuming, for simplicity, that the application of the QEC cycle happens on a much shorter timescale than the idling time τ. Afterwards, the states are left in memory for a further time $\tau/2$, again inducing noise $\mathcal{N}_{p_{\mathrm{i}}}$. Finally, we project encoded logical states back to the codespace by performing a round of perfect QEC. We note that this last step is not part of an actual experimental protocol but rather serves as a tool for the quantitative assessment of the state's quality. The probability $\mathcal{P}$ of successful state discrimination is then given by the fidelity of a final state w.r.t. the corresponding noise-free initial state. We take this quantity as a measure for the quality of the quantum memory. Comparing the values of $\mathcal{P}$ for the different scenarios in Fig. 10 informs us whether the application of the intrinsically noisy QEC device can be beneficial.

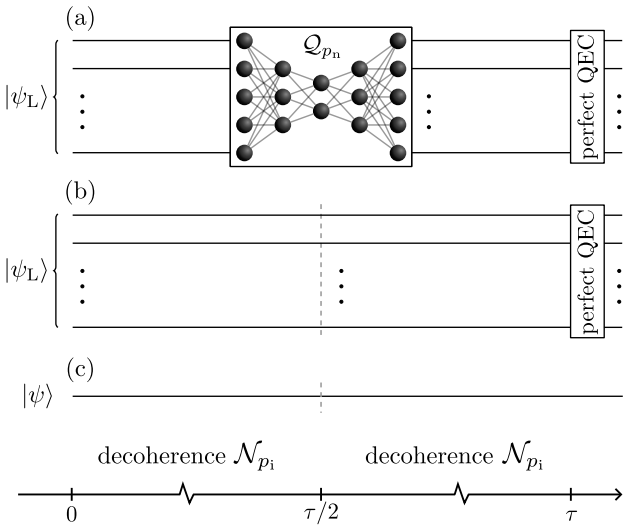


Figure 10: Scheme to evaluate the efficacy of QEC with a QAE in a quantum memory. A quantum state, either an encoded logical state $|\psi_{\mathrm{L}}\rangle$ (a,b) or a single-qubit state $|\psi\rangle$ stored in a bare physical qubit (c), is stored in memory for a time $\tau/2$ such that noise $\mathcal{N}_{p_{\mathrm{i}}}$ acts on all physical qubits. We may now choose to perform a round of imperfect QEC (a), or to not apply it (b). Here, we assume, for simplicity, that the QEC round happens on a much shorter timescale than τ. The qubits then idle for another time $\tau/2$ which introduces noise $\mathcal{N}_{p_{\mathrm{i}}}$. Finally, logical states are projected back to the codespace by a round of perfect QEC, which provides a tool to assess the state quality. The fidelity of the output w.r.t. the initial state determines the probability $\mathcal{P}$ of successful state discrimination and serves as a measure for the quality of the quantum memory.

To investigate whether noisy QAEs introduced in this paper can prove beneficial for quantum error correction, we analyze a minimal example, which is an intrinsically noisy 3-1-3 QAE to correct bit flip errors. We note, however, that the analysis we perform for this example could also be applied to larger codes and therefore other types of network structures and noise. In this work we do not focus on a specific physical platform and therefore consider a platform-agnostic noise model. In particular, we apply a multi-qubit depolarizing channel after every application of a unitary matrix. In this channel, any-weight Pauli errors occur with equal probabilities $p_{\mathrm{n}}/(4^m-1)$, where m is the number of qubits that the noise channel acts upon. Here we consider a 3-1-3 DQNN with standard architecture consisting of four independent unitary matrices, i.e. we do not use the self-inverse ansatz. A corresponding quantum circuit representation can be found in Appendix E. The internal noise thus acts at four positions in the circuit. We model the environmental decoherence in the quantum memory as bit flip errors occurring independently on every qubit with probability p_{i}. For various pairs of noise strengths $(p_{\mathrm{i}}, p_{\mathrm{n}})$ we perform numerical simulations according to Fig. 10 to analyze how well the intrinsically noisy QAE corrects

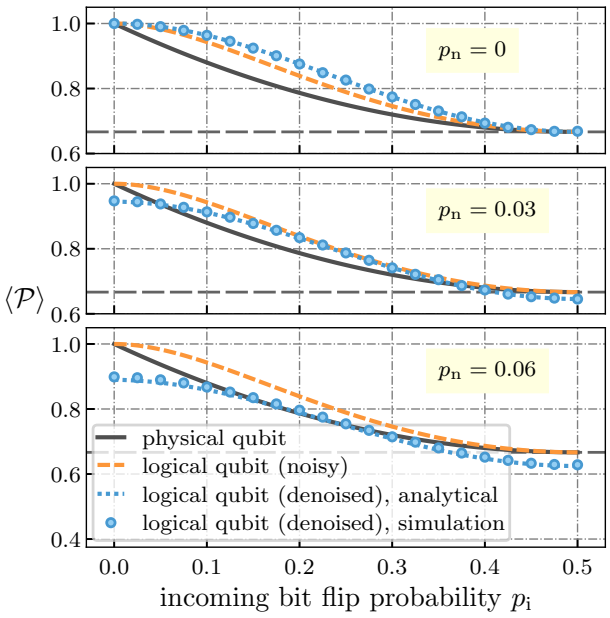


Figure 11: Comparison of the three different quantum memory settings shown in Fig. 10. A bare physical qubit (solid black line), an encoded but uncorrected logical qubit (dashed orange line) and a corrected logical qubit (dotted blue line and data points) are compared. As incoming noise we consider independent bit flips occurring with probability p_i on every qubit. We use logical states of the three-qubit repetition code and apply a 3-1-3 QAE for QEC. The QAE is intrinsically noisy with the strength of the network noise quantified by p_n. Encoding quantum information in logical states and performing active QEC is advantageous if the probability of successful state discrimination $\mathcal{P}$ of denoised states is larger than for single-qubit and uncorrected logical states. A feature of the three-qubit code is that the quality of the latter two saturates at $\langle \mathcal{P} \rangle = 2/3$ for $p_i = 0.5$, indicated by the horizontal dashed line. For a perfect QAE, applying active QEC is beneficial for $p_i < 0.5$. For an increasingly noisy QAE, the range of incoming noise strengths for which it is beneficial to apply QEC is reduced. The numerical data is obtained by averaging over 10^4 randomly drawn logical states for each data point.

bit flip errors on the logical states. Fig. 11 compares the quality of the three quantum memory settings introduced in Fig. 10 for various different network noise strengths p_n and incoming noise strengths p_i. We average the probability of successful state discrimination over a large number of different input states. First, we note that for bit flip probabilities $p_i < 0.5$ the uncorrected encoded qubit performs better than the bare physical qubit. This is a known property of the repetition code, resulting from the correctability of all single-qubit errors in the final perfect round of QEC. For a noise-free QAE, $p_n = 0$, the error correction is advantageous for incoming noise strengths in exactly that range. Small values of p_n reduce the interval of incoming noise strengths for which performing QEC is still beneficial. Above a certain threshold of p_n, the quality of an actively corrected logical qubit drops below the quality of an uncorrected encoded qubit. If we increase p_n even further, the scheme involving the noisy QAE is eventually outperformed by a bare physical qubit. In Appendix E we derive an approximate expression for the quality of denoised logical states as a function of p_i and p_n. We find that it scales linearly with p_n, highlighting that the design of the QAE is not fault-tolerant. This is, however, not surprising, since errors on the single bottleneck qubit inevitably result in logical errors on the final state. Performing an extensive analysis for various pairs of noise strengths (p_i, p_n) we obtain a phase diagram, depicted in Fig. 12, that indicates three regimes: one where the application of a noisy QAE for quantum error correction is advantageous compared to both a bare physical qubit and an encoded but uncorrected logical qubit; a regime where the quality of the actively corrected qubit lies between the latter two; and a third regime where the actively corrected qubit is outperformed by the other two approaches. We see that noisy QAEs can be successfully used for QEC as long as the internal noise of the quantum network stays below a certain p_i-dependent threshold. To obtain an approximate form of the phase boundaries we expand the output states of the noisy 3-1-3 QAE up to linear order in p_n. The calculation is sketched in Appendix E. For small values of p_n we observe excellent agreement of the analytical phase boundaries with the numerical results. For encoded logical states in the quantum memory, the final round of perfect QEC removes any single bit flip errors, such that the logical error rate and thus also the probability of successful state discrimination are quadratic in p_i. Therefore, also the boundary separating the blue-colored and orange-colored regions in Fig. 12 shows a quadratic behavior for small values of p_i. In contrast to this finding, the boundary which separates the orange-colored and the grey-colored regions is linear for small p_i. This results from the fact that the probability of successful state discrimination for a single-qubit state is linear in p_i. Typically, we are interested in the regime of small, though not too small incoming noise and small network noise strengths. In that case the phase boundary separating the blue and the orange phase can be very well approximated as $p_n^{\text{crit., logical}} = \frac{765}{351} p_i^2$, as derived in Appendix E. For the investigated QAE consisting of four unitary matrices the probability for a single error to occur during the application of the network is in leading order $p_{1\,\text{err.}} \approx 4 p_n$. This means that for $p_{1\,\text{err.}} \lesssim 4 \cdot \frac{765}{351} p_i^2 = \frac{3060}{351} p_i^2$ it is for this network structure advantageous to apply a QAE for error correction in the quantum memory.

6 Discussion and Outlook

In this paper we showed that quantum neural networks in the form of quantum autoencoders can be used to perform quantum error correction. QAEs are

able to correct errors on arbitrary states from a predefined logical codespace such that the lifetime of encoded logical qubits can be enhanced. In particular, if a logical encoding allows for various error correction strategies to be applied, a QAE can learn the strategy yielding the best possible denoising results. As an example we demonstrated that QAEs are able to adapt to spatially correlated bit flip noise. Moreover, we showed for the first time that the error correction abilities of QAEs are not limited to computational errors but extend also to the correction of qubit erasures. Other types of QNNs designed in the spirit of overcomplete quantum autoencoders can be used to find novel logical encodings being optimally suited to correct hardware-specific noise. We proposed and showed that these networks can be directly transformed into undercomplete QAEs ready to perform QEC on the discovered logical states without the need to perform further training. Lastly, even QAEs that are intrinsically noisy can be used successfully for QEC in a quantum memory if the internal noise of the quantum neural network is sufficiently low.

However, we note that we encountered difficulties in the training of DQNNs for QEC. We observe that especially the training of encoding finder networks, as discussed in Sec. 4, frequently converges towards solutions yielding non-optimal error correction strategies. These observations indicate the existence of saddle points or local minima in the cost function landscape. At the beginning of the training process, the unitary matrices composing a DQNN are initialized randomly, so convergence towards those non-optimal points is hard to avoid using standard gradient descent methods. To obtain a quantum network that is able to optimally correct errors, we thus have to perform a repeated random initialization of the training parameters followed by a gradient descent algorithm. These findings are, however, not unexpected since the occurrence of barren plateaus, where cost function gradients vanish exponentially in the number of qubits [95], is frequently observed for variational quantum algorithms. These training issues therefore impede a straightforward scaling to substantially larger quantum neural networks, where we expect the training difficulties to become more prominent for increasing depth and width of the networks. In fact, it has recently been shown explicitly that also DQNNs are affected by barren plateaus [69]. Ongoing research on the trainability of QNNs involves e.g. the investigation of different cost functions such that the occurrence of barren plateaus can be avoided [96]. Moreover, it is a challenge to find suitable variational ansätze that are sufficiently expressive while avoiding the occurrence of barren plateaus [97]. Eventually, one aims at finding hardware-specific ansätze with a reduced number of parameters that allow for an easy application of a QNN on available quantum hardware and do not suffer from serious trainability problems.

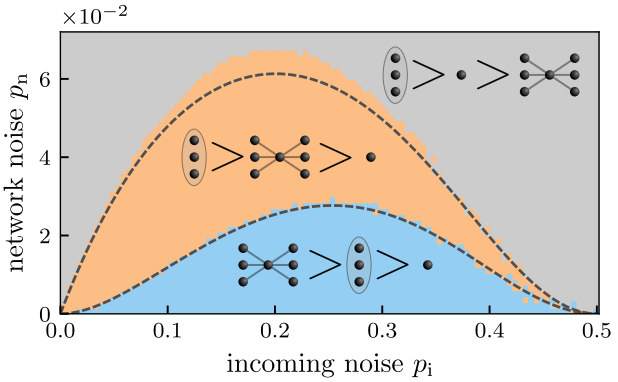


Figure 12: Phase diagram indicating the benefit of a QAE for QEC. It shows for which values of incoming bit flip noise strengths p_i and network noise strengths p_n a 3-1-3 QAE is beneficial for active QEC in a quantum memory. The range of noise strengths in which the QAE-corrected memory is superior to both a bare physical qubit and an encoded but uncorrected qubit is shown as the lower blue region. The orange-colored intermediate region corresponds to the actively corrected qubit performing better than a bare qubit but worse than the uncorrected logical qubit. In the grey-colored upper region, applying the QAE for QEC is inferior to both other cases. Each data point is obtained by randomly drawing 10^4 logical states, exposing them to a bit flip channel and processing these states $\mathcal{N}_{p_i}(|\psi_L\rangle\langle\psi_L|)$ with the noisy QAE. Then, another round of noise $\mathcal{N}_{p_i}$ and a perfect round of QEC are applied. We compare the probability of successful state discrimination to that of a physical qubit and an uncorrected encoded qubit. The phase boundaries are in some parts pixelated due to finite sampling statistics. The lines correspond to phase boundaries predicted from an analytical expansion to first order in p_n (see Appendix E). For small values of p_n the analytical boundaries match the actual boundaries very well, whereas small deviations for larger values of p_n are expected.

In summary, our work shows that QAEs could serve as a versatile tool for autonomous quantum error correction of a wide variety of error sources and characteristics in a quantum memory. The QAE framework is especially attractive for experimental setups where in-sequence measurements with real-time feedback as required for the conventional QEC approach are not readily available or inefficient. Thus, a comparison of quantum autoencoders with other autonomous QEC proposals would be interesting. Furthermore, possible future work could include the investigation of fault-tolerant designs of QAEs for quantum error correction to extend their applicability beyond the quantum memory setting. In this context, one could also imagine logical qubits corrected by QAEs being used as low-level autonomously running units that are trained to deal with the dominant error sources of a given architecture. Those could then form building blocks of more complex established QEC codes, in analogy to concatenating basic few-qubit codes with more advanced codes [98] or using bosonic codes as building blocks for scalable QEC schemes [4].

Acknowledgments

We thank D. Bondarenko, P. Feldmann and D. DiVincenzo for stimulating discussions and M. Rispler for feedbacks on the manuscript. We acknowledge support by the ERC Starting Grant QNets Grant Number 804247, the EU H2020-FETFLAG-2018-03 under Grant Agreement number 820495, by the German ministry of science and education (BMBF) via the VDI within the project IQuAn, by the Deutsche Forschungsgemeinschaft through Grant No. 449905436, and by US A.R.O. through Grant No. W911NF-21-1-0007, and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via US ARO Grant number W911NF-16-1-0070. All statements of fact, opinions or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of ODNI, the IARPA, or the US Government. The network coding and training was done in the Matlab programming language, based on code available on Github at [49, 60].

References

[1] Simon J Devitt, William J Munro, and Kae Nemoto. "Quantum error correction for beginners". Reports on Progress in Physics **76**, 076001 (2013).

[2] Michael A. Nielsen and Isaac L. Chuang. "Quantum computation and quantum information: 10th anniversary edition". Cambridge University Press. (2010).

[3] Barbara M. Terhal. "Quantum error correction for quantum memories". Rev. Mod. Phys. **87**, 307–346 (2015).

[4] B M Terhal, J Conrad, and C Vuillot. "Towards scalable bosonic quantum error correction". Quantum Science and Technology **5**, 043001 (2020).

[5] D. G. Cory, M. D. Price, W. Maas, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel, and S. S. Somaroo. "Experimental quantum error correction". Phys. Rev. Lett. **81**, 2152–2155 (1998).

[6] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. B. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer, R. Ozeri, and D. J. Wineland. "Realization of quantum error correction". Nature **432**, 602–605 (2004).

[7] Philipp Schindler, Julio T. Barreiro, Thomas Monz, Volckmar Nebendahl, Daniel Nigg, Michael Chwalla, Markus Hennrich, and Rainer Blatt. "Experimental repetitive quantum error correction". Science **332**, 1059–1061 (2011).

[8] Norbert M. Linke, Mauricio Gutierrez, Kevin A. Landsman, Caroline Figgatt, Shantanu Debnath, Kenneth R. Brown, and Christopher Monroe. "Fault-tolerant quantum error detection". Science Advances **3**, e1701074 (2017).

[9] Christian Kraglund Andersen, Ants Remm, Stefania Lazar, Sebastian Krinner, Nathan Lacroix, Graham J. Norris, Mihai Gabureac, Christopher Eichler, and Andreas Wallraff. "Repeated quantum error detection in a surface code". Nature Physics **16**, 875–880 (2020).

[10] J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, and U. G. Poschinger. "Fault-tolerant parity readout on a shuttling-based trapped-ion quantum computer". Phys. Rev. X **12**, 011032 (2022).

[11] Laird Egan, Dripto M. Debroy, Crystal Noel, Andrew Risinger, Daiwei Zhu, Debopriyo Biswas, Michael Newman, Muyuan Li, Kenneth R. Brown, Marko Cetina, and Christopher Monroe. "Fault-tolerant control of an error-corrected qubit". Nature **598**, 281–286 (2021).

[12] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. A. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz. "Realization of real-time fault-tolerant quantum error correction". Phys. Rev. X **11**, 041058 (2021).

[13] M. H. Abobeih, Y. Wang, J. Randall, S. J. H. Loenen, C. E. Bradley, M. Markham, D. J. Twitchen, B. M. Terhal, and T. H. Taminiau. "Fault-tolerant operation of a logical qubit in a diamond quantum processor". Nature **606**, 884–889 (2022).

[14] M. Riebe, H. Häffner, C. F. Roos, W. Hänsel, J. Benhelm, G. P. T. Lancaster, T. W. Körber, C. Becher, F. Schmidt-Kaler, D. F. V. James, and R. Blatt. "Deterministic quantum teleportation with atoms". Nature **429**, 734–737 (2004).

[15] M. D. Barrett, J. Chiaverini, T. Schaetz, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer, D. Leibfried, R. Ozeri, and D. J. Wineland. "Deterministic quantum teleportation of atomic qubits". Nature **429**, 737–739 (2004).

[16] Clément Sayrin, Igor Dotsenko, Xingxing Zhou, Bruno Peaudecerf, Théo Rybarczyk, Sébastien Gleyzes, Pierre Rouchon, Mazyar Mirrahimi, Hadis Amini, Michel Brune, Jean-Michel Raimond, and Serge Haroche. "Real-time quantum feedback prepares and stabilizes photon number states". Nature **477**, 73–77 (2011).

[17] D. Ristè, M. Dukalski, C. A. Watson, G. de Lange, M. J. Tiggelman, Ya. M. Blanter, K. W. Lehnert, R. N. Schouten, and L. DiCarlo. "Deterministic entanglement of superconducting qubits by parity measurement and feedback". Nature **502**, 350–354 (2013).

[18] Sebastian Krinner, Nathan Lacroix, Ants Remm,

Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes Herrmann, Graham J. Norris, Christian Kraglund Andersen, Markus Müller, Alexandre Blais, Christopher Eichler, and Andreas Wallraff. "Realizing repeated quantum error correction in a distance-three surface code". Nature **605**, 669–674 (2022).

[19] J. F. Marques, B. M. Varbanov, M. S. Moreira, H. Ali, N. Muthusubramanian, C. Zachariadis, F. Battistel, M. Beekman, N. Haider, W. Vlothuizen, A. Bruno, B. M. Terhal, and L. DiCarlo. "Logical-qubit operations in an error-detecting surface code". Nature Physics **18**, 80–86 (2022).

[20] Google Quantum AI. "Exponential suppression of bit or phase errors with cyclic error correction". Nature **595**, 383–387 (2021).

[21] Youwei Zhao, Yangsen Ye, He-Liang Huang, Yiming Zhang, Dachao Wu, Huijie Guan, Qingling Zhu, Zuolin Wei, Tan He, Sirui Cao, Fusheng Chen, Tung-Hsun Chung, Hui Deng, Daojin Fan, Ming Gong, Cheng Guo, Shaojun Guo, Lianchen Han, Na Li, Shaowei Li, Yuan Li, Futian Liang, Jin Lin, Haoran Qian, Hao Rong, Hong Su, Lihua Sun, Shiyu Wang, Yulin Wu, Yu Xu, Chong Ying, Jiale Yu, Chen Zha, Kaili Zhang, Yong-Heng Huo, Chao-Yang Lu, Cheng-Zhi Peng, Xiaobo Zhu, and Jian-Wei Pan. "Realization of an error-correcting surface code with superconducting qubits". Phys. Rev. Lett. **129**, 030501 (2022).

[22] J. Cramer, N. Kalb, M. A. Rol, B. Hensen, M. S. Blok, M. Markham, D. J. Twitchen, R. Hanson, and T. H. Taminiau. "Repeated quantum error correction on a continuously encoded qubit by real-time feedback". Nature Communications **7**, 11526 (2016).

[23] Christian Kraglund Andersen, Ants Remm, Stefania Lazar, Sebastian Krinner, Johannes Heinsoo, Jean-Claude Besse, Mihai Gabureac, Andreas Wallraff, and Christopher Eichler. "Entanglement stabilization using ancilla-based parity detection and real-time feedback in superconducting circuits". npj Quantum Information **5**, 69 (2019).

[24] Diego Ristè, Luke C. G. Govia, Brian Donovan, Spencer D. Fallek, William D. Kalfus, Markus Brink, Nicholas T. Bronn, and Thomas A. Ohki. "Real-time processing of stabilizer measurements in a bit-flip code". npj Quantum Information **6**, 71 (2020).

[25] V. Negnevitsky, M. Marinelli, K. K. Mehta, H.-Y. Lo, C. Flühmann, and J. P. Home. "Repeated multi-qubit readout and feedback with a mixed-species trapped-ion register". Nature **563**, 527–531 (2018).

[26] Nissim Ofek, Andrei Petrenko, Reinier Heeres, Philip Reinhold, Zaki Leghtas, Brian Vlastakis, Yehan Liu, Luigi Frunzio, S. M. Girvin, L. Jiang, Mazyar Mirrahimi, M. H. Devoret, and R. J. Schoelkopf. "Extending the lifetime of a quantum bit with error correction in superconducting circuits". Nature **536**, 441–445 (2016).

[27] L. Hu, Y. Ma, W. Cai, X. Mu, Y. Xu, W. Wang, Y. Wu, H. Wang, Y. P. Song, C.-L. Zou, S. M. Girvin, L.-M. Duan, and L. Sun. "Quantum error correction and universal gate set operation on a binomial bosonic logical qubit". Nature Physics **15**, 503–508 (2019).

[28] Giacomo Torlai and Roger G. Melko. "Neural decoder for topological codes". Phys. Rev. Lett. **119**, 030501 (2017).

[29] Ye-Hua Liu and David Poulin. "Neural belief-propagation decoders for quantum error-correcting codes". Phys. Rev. Lett. **122**, 200501 (2019).

[30] Nishad Maskara, Aleksander Kubica, and Tomas Jochym-O'Connor. "Advantages of versatile neural-network decoding for topological codes". Phys. Rev. A **99**, 052351 (2019).

[31] Ryan Sweke, Markus S Kesselring, Evert P L van Nieuwenburg, and Jens Eisert. "Reinforcement learning decoders for fault-tolerant quantum computation". Machine Learning: Science and Technology **2**, 025005 (2021).

[32] Benjamin J. Brown, Daniel Loss, Jiannis K. Pachos, Chris N. Self, and James R. Wootton. "Quantum memories at finite temperature". Rev. Mod. Phys. **88**, 045005 (2016).

[33] Gerardo A. Paz-Silva, Gavin K. Brennen, and Jason Twamley. "Fault tolerance with noisy and slow measurements and preparation". Phys. Rev. Lett. **105**, 100501 (2010).

[34] Daniel Crow, Robert Joynt, and M. Saffman. "Improved error thresholds for measurement-free error correction". Phys. Rev. Lett. **117**, 130503 (2016).

[35] Vickram N. Premakumar, M. Saffman, and Robert Joynt. "Measurement-free error correction with coherent ancillas" (2020). arXiv:2007.09804.

[36] Joseph Kerckhoff, Hendra I. Nurdin, Dmitri S. Pavlichin, and Hideo Mabuchi. "Designing quantum memories with embedded control: Photonic circuits for autonomous quantum error correction". Phys. Rev. Lett. **105**, 040502 (2010).

[37] Fernando Pastawski, Lucas Clemente, and Juan Ignacio Cirac. "Quantum memories based on engineered dissipation". Phys. Rev. A **83**, 012304 (2011).

[38] Eliot Kapit. "Hardware-efficient and fully autonomous quantum error correction in superconducting circuits". Phys. Rev. Lett. **116**, 150501 (2016).

[39] F. Reiter, A. S. Sørensen, P. Zoller, and C. A.

Muschik. "Dissipative quantum error correction and application to quantum sensing with trapped ions". Nature Communications **8**, 1822 (2017).

[40] Zaki Leghtas, Gerhard Kirchmair, Brian Vlastakis, Robert J. Schoelkopf, Michel H. Devoret, and Mazyar Mirrahimi. "Hardware-efficient autonomous quantum memory protection". Phys. Rev. Lett. **111**, 120501 (2013).

[41] Z. Leghtas, S. Touzard, I. M. Pop, A. Kou, B. Vlastakis, A. Petrenko, K. M. Sliwa, A. Narla, S. Shankar, M. J. Hatridge, M. Reagor, L. Frunzio, R. J. Schoelkopf, M. Mirrahimi, and M. H. Devoret. "Confining the state of light to a quantum manifold by engineered two-photon loss". Science **347**, 853–857 (2015).

[42] Jae-Mo Lihm, Kyungjoo Noh, and Uwe R. Fischer. "Implementation-independent sufficient condition of the knill-laflamme type for the autonomous protection of logical qudits by strong engineered dissipation". Phys. Rev. A **98**, 012317 (2018).

[43] Jeffrey M. Gertler, Brian Baker, Juliang Li, Shruti Shirol, Jens Koch, and Chen Wang. "Protecting a bosonic qubit with autonomous quantum error correction". Nature **590**, 243–248 (2021).

[44] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. "The quest for a quantum neural network". Quantum Information Processing **13**, 2567–2586 (2014).

[45] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. "Quantum machine learning". Nature **549**, 195–202 (2017).

[46] Vedran Dunjko and Hans J Briegel. "Machine learning & artificial intelligence in the quantum domain: a review of recent progress". Reports on Progress in Physics **81**, 074001 (2018).

[47] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. "Variational quantum algorithms". Nature Reviews Physics **3**, 625–644 (2021).

[48] S. Mangini, F. Tacchino, D. Gerace, D. Bajoni, and C. Macchiavello. "Quantum computing models for artificial neural networks". EPL (Europhysics Letters) **134**, 10002 (2021).

[49] Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J. Osborne, Robert Salzmann, Daniel Scheiermann, and Ramona Wolf. "Training deep quantum neural networks". Nature Communications **11**, 808 (2020).

[50] Lasse Bjørn Kristensen, Matthias Degroote, Peter Wittek, Alán Aspuru-Guzik, and Nikolaj T. Zinner. "An artificial spiking quantum neuron". npj Quantum Information **7**, 59 (2021).

[51] E. Torrontegui and J. J. García-Ripoll. "Unitary quantum perceptron as efficient universal approximator". EPL (Europhysics Letters) **125**, 30004 (2019).

[52] Dan Ventura and Tony Martinez. "Quantum associative memory". Information Sciences **124**, 273–296 (2000).

[53] Patrick Rebentrost, Thomas R. Bromley, Christian Weedbrook, and Seth Lloyd. "Quantum hopfield neural network". Phys. Rev. A **98**, 042308 (2018).

[54] Eliana Fiorelli, Igor Lesanovsky, and Markus Müller. "Phase diagram of quantum generalized potts-hopfield neural networks". New Journal of Physics **24**, 033012 (2022).

[55] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. "Quantum autoencoders for efficient compression of quantum data". Quantum Science and Technology **2**, 045001 (2017).

[56] L Lamata, U Alvarez-Rodriguez, J D Martín-Guerrero, M Sanz, and E Solano. "Quantum autoencoders via quantum adders with genetic algorithms". Quantum Science and Technology **4**, 014007 (2018).

[57] Hailan Ma, Chang-Jiang Huang, Chunlin Chen, Daoyi Dong, Yuanlong Wang, Re-Bing Wu, and Guo-Yong Xiang. "On compression rate of quantum autoencoders: Control design, numerical and experimental realization". Automatica **147**, 110659 (2023).

[58] Carlos Bravo-Prieto. "Quantum autoencoders with enhanced data encoding". Machine Learning: Science and Technology **2**, 035028 (2021).

[59] Chenfeng Cao and Xin Wang. "Noise-assisted quantum autoencoder". Phys. Rev. Applied **15**, 054012 (2021).

[60] Dmytro Bondarenko and Polina Feldmann. "Quantum autoencoders to denoise quantum data". Phys. Rev. Lett. **124**, 130502 (2020).

[61] Tom Achache, Lior Horesh, and John Smolin. "Denoising quantum states with quantum autoencoders – theory and applications" (2020). arXiv:2012.14714.

[62] Xiao-Ming Zhang, Weicheng Kong, Muhammad Usman Farooq, Man-Hong Yung, Guoping Guo, and Xin Wang. "Generic detection-based error mitigation using quantum autoencoders". Phys. Rev. A **103**, L040403 (2021).

[63] Alex Pepper, Nora Tischler, and Geoff J. Pryde. "Experimental realization of a quantum autoencoder: The compression of qutrits via machine learning". Phys. Rev. Lett. **122**, 060501 (2019).

[64] Chang-Jiang Huang, Hailan Ma, Qi Yin, Jun-Feng Tang, Daoyi Dong, Chunlin Chen, Guo-Yong Xiang, Chuan-Feng Li, and Guang-Can Guo. "Realization of a quantum autoencoder for lossless compression of quantum data". Phys. Rev. A **102**, 032412 (2020).

[65] Yongcheng Ding, Lucas Lamata, Mikel Sanz, Xi Chen, and Enrique Solano. "Experimental implementation of a quantum autoencoder via quantum adders". Advanced Quantum Technologies **2**, 1800065 (2019).

[66] Peter D. Johnson, Jonathan Romero, Jonathan Olson, Yudong Cao, and Alán Aspuru-Guzik. "Qvector: an algorithm for device-tailored quantum error correction" (2017). arXiv:1711.02249.

[67] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. "Quantum convolutional neural networks". Nature Physics **15**, 1273–1278 (2019).

[68] Chenfeng Cao, Chao Zhang, Zipeng Wu, Markus Grassl, and Bei Zeng. "Quantum variational learning for quantum error-correcting codes". Quantum **6**, 828 (2022).

[69] Kunal Sharma, M. Cerezo, Lukasz Cincio, and Patrick J. Coles. "Trainability of dissipative perceptron-based quantum neural networks". Phys. Rev. Lett. **128**, 180505 (2022).

[70] Kerstin Beer, Daniel List, Gabriel Müller, Tobias J. Osborne, and Christian Struckmann. "Training quantum neural networks on nisq devices" (2021). arXiv:2104.06081.

[71] Daniel A Lidar and Todd A Brun. "Quantum error correction". Cambridge University Press. (2013).

[72] Daniel Eric Gottesman. "Stabilizer codes and quantum error correction". PhD thesis. California Institute of Technology. (1997).

[73] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. "Deep learning". MIT Press. (2016). url: `http://www.deeplearningbook.org`.

[74] Michael Tschannen, Olivier Bachem, and Mario Lucic. "Recent advances in autoencoder-based representation learning" (2018). arXiv:1812.05069.

[75] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. "Extracting and composing robust features with denoising autoencoders". In Proceedings of the 25th International Conference on Machine Learning. Page 1096–1103. ICML '08New York, NY, USA (2008). Association for Computing Machinery.

[76] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. "Perfect quantum error correcting code". Phys. Rev. Lett. **77**, 198–201 (1996).

[77] Rochus Klesse and Sandra Frank. "Quantum error correction in spatially correlated quantum noise". Phys. Rev. Lett. **95**, 230503 (2005).

[78] Christopher T. Chubb and Steven T. Flammia. "Statistical mechanical models for quantum codes with correlated noise". Annales de l'Institut Henri Poincaré D **8**, 269–321 (2021).

[79] M. Grassl, Th. Beth, and T. Pellizzari. "Codes for the quantum erasure channel". Physical Review A **56**, 33–38 (1997).

[80] Roman Stricker, Davide Vodola, Alexander Erhard, Lukas Postler, Michael Meth, Martin Ringbauer, Philipp Schindler, Thomas Monz, Markus Müller, and Rainer Blatt. "Experimental deterministic correction of qubit loss". Nature **585**, 207–210 (2020).

[81] Jonathan M. Baker, Andrew Litteken, Casey Duckering, Henry Hoffmann, Hannes Bernien, and Frederic T. Chong. "Exploiting long-distance interactions and tolerating atom loss in neutral atom quantum architectures". In 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). Pages 818–831. (2021).

[82] Chao-Yang Lu, Wei-Bo Gao, Jin Zhang, Xiao-Qi Zhou, Tao Yang, and Jian-Wei Pan. "Experimental quantum coding against qubit loss error". Proceedings of the National Academy of Sciences **105**, 11050–11054 (2008).

[83] Austin G. Fowler. "Coping with qubit leakage in topological codes". Phys. Rev. A **88**, 042308 (2013).

[84] Boris Mihailov Varbanov, Francesco Battistel, Brian Michael Tarasinski, Viacheslav Petrovych Ostroukh, Thomas Eugene O'Brien, Leonardo DiCarlo, and Barbara Maria Terhal. "Leakage detection for a transmon-based surface code". npj Quantum Information **6**, 102 (2020).

[85] F. Battistel, B.M. Varbanov, and B.M. Terhal. "Hardware-efficient leakage-reduction scheme for quantum error correction with superconducting transmon qubits". PRX Quantum **2**, 030314 (2021).

[86] Natalie C. Brown and Kenneth R. Brown. "Comparing zeeman qubits to hyperfine qubits in the context of the surface code: $^{174}Yb^{+}$ and $^{171}Yb^{+}$". Phys. Rev. A **97**, 052301 (2018).

[87] Yue Wu, Shimon Kolkowitz, Shruti Puri, and Jeff D. Thompson. "Erasure conversion for fault-tolerant quantum computing in alkaline earth rydberg atom arrays". Nature Communications **13**, 4657 (2022).

[88] Thomas Monz, Philipp Schindler, Julio T. Barreiro, Michael Chwalla, Daniel Nigg, William A. Coish, Maximilian Harlander, Wolfgang Hänsel, Markus Hennrich, and Rainer Blatt. "14-qubit entanglement: Creation and coherence". Phys. Rev. Lett. **106**, 130506 (2011).

[89] T. Yu and J. H. Eberly. "Qubit disentanglement and decoherence via dephasing". Phys. Rev. B **68**, 165322 (2003).

[90] A. Bermudez, X. Xu, M. Gutiérrez, S. C. Benjamin, and M. Müller. "Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources". Phys. Rev. A **100**, 062307 (2019).

[91] D. A. Lidar, I. L. Chuang, and K. B. Whaley. "Decoherence-free subspaces for quantum computation". Phys. Rev. Lett. **81**, 2594–2597 (1998).

[92] Paul G. Kwiat, Andrew J. Berglund, Joseph B. Altepeter, and Andrew G. White. "Experimental verification of decoherence-free subspaces". Science **290**, 498–501 (2000).

[93] Edoardo G. Carnio, Andreas Buchleitner, and Manuel Gessner. "Robust asymptotic entanglement under multipartite collective dephasing". Phys. Rev. Lett. **115**, 010404 (2015).

[94] A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, F. Schmidt-Kaler, M. Biercuk, R. Blatt, S. Benjamin, and M. Müller. "Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation". Phys. Rev. X **7**, 041061 (2017).

[95] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. "Barren plateaus in quantum neural network training landscapes". Nature Communications **9**, 4812 (2018).

[96] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. "Cost function dependent barren plateaus in shallow parametrized quantum circuits". Nature Communications **12**, 1791 (2021).

[97] Zoë Holmes, Kunal Sharma, M. Cerezo, and Patrick J. Coles. "Connecting ansatz expressibility to gradient magnitudes and barren plateaus". PRX Quantum **3**, 010313 (2022).

[98] Andrew W. Cross, David P. DiVincenzo, and Barbara M. Terhal. "A comparative code study for quantum fault tolerance". Quantum Inf. Comput. **9**, 541–572 (2009).

[99] Sebastian Ruder. "An overview of gradient descent optimization algorithms" (2017). arXiv:1609.04747.

A Quantum Process Tomography

In this appendix we study some of the quantum autoencoders discussed in the main text in more detail by analyzing the maps which they implement using quantum process tomographies. The error correction map of the standard 3-qubit bit flip code is given by

$$\mathcal{R}(\cdot) = \sum_b M_b \cdot M_b^\dagger \,, \tag{14a}$$

with Kraus operators

$$\begin{aligned} M_{00} &= |000\rangle\langle 000| + |111\rangle\langle 111| \\ M_{01} &= |000\rangle\langle 001| + |111\rangle\langle 110| \\ M_{10} &= |000\rangle\langle 100| + |111\rangle\langle 011| \\ M_{11} &= |000\rangle\langle 010| + |111\rangle\langle 101| \,. \end{aligned} \tag{14b}$$

In Sec. 3 of the main text we demonstrate that our fully trained 3-1-3 QAEs correct bit flip errors on 3QC logical states as well as the standard 3QC does. To prove the assumption that the QAEs implement the map stated in Eq. (14), we perform quantum process tomographies of the channels realized by the networks. Given an operator basis $\{E_i\}$, a quantum channel $\mathcal{Q}$ can be written as

$$\mathcal{Q}(\cdot) = \sum_{i,j} \chi_{ij} E_i \cdot E_j^\dagger, \tag{15}$$

where the complex Hermitian matrix χ uniquely characterizes the quantum channel $\mathcal{Q}$ w.r.t. the basis $\{E_i\}$ [2]. Figure 4(c) in the main text shows the process matrix χ of a fully trained 3-1-3 QAE which is equal to the process matrix of the map in Eq. (14). Analyzing the trained QAEs shown in Fig. 3 in the main text, we find that all process matrices apart from the one corresponding to the network trained on $p = 0$ equal the matrix in Fig. 4(c). This demonstrates that QAEs which are trained on noisy states learn to implement the correction channel of the 3-qubit bit flip code, even though the number of different training states is very limited.

The analytical curve describing the performance of the standard 3-qubit code in Fig. 3 can be obtained as follows. As stated in the main text, a logical 3QC state $|\psi_\mathrm{L}\rangle$ undergoing independent bit flip noise of strength p and being actively corrected afterwards suffers a logical error with probability $p_\mathrm{L} = 3p^2(1-p) + p^3$ and no error with probability $1 - p_\mathrm{L}$, thus resulting in a state

$$\rho_\mathrm{denoised} = (1 - p_\mathrm{L}) \, |\psi_\mathrm{L}\rangle\langle\psi_\mathrm{L}| + p_\mathrm{L} X_\mathrm{L} \, |\psi_\mathrm{L}\rangle\langle\psi_\mathrm{L}| \, X_\mathrm{L}. \tag{16}$$

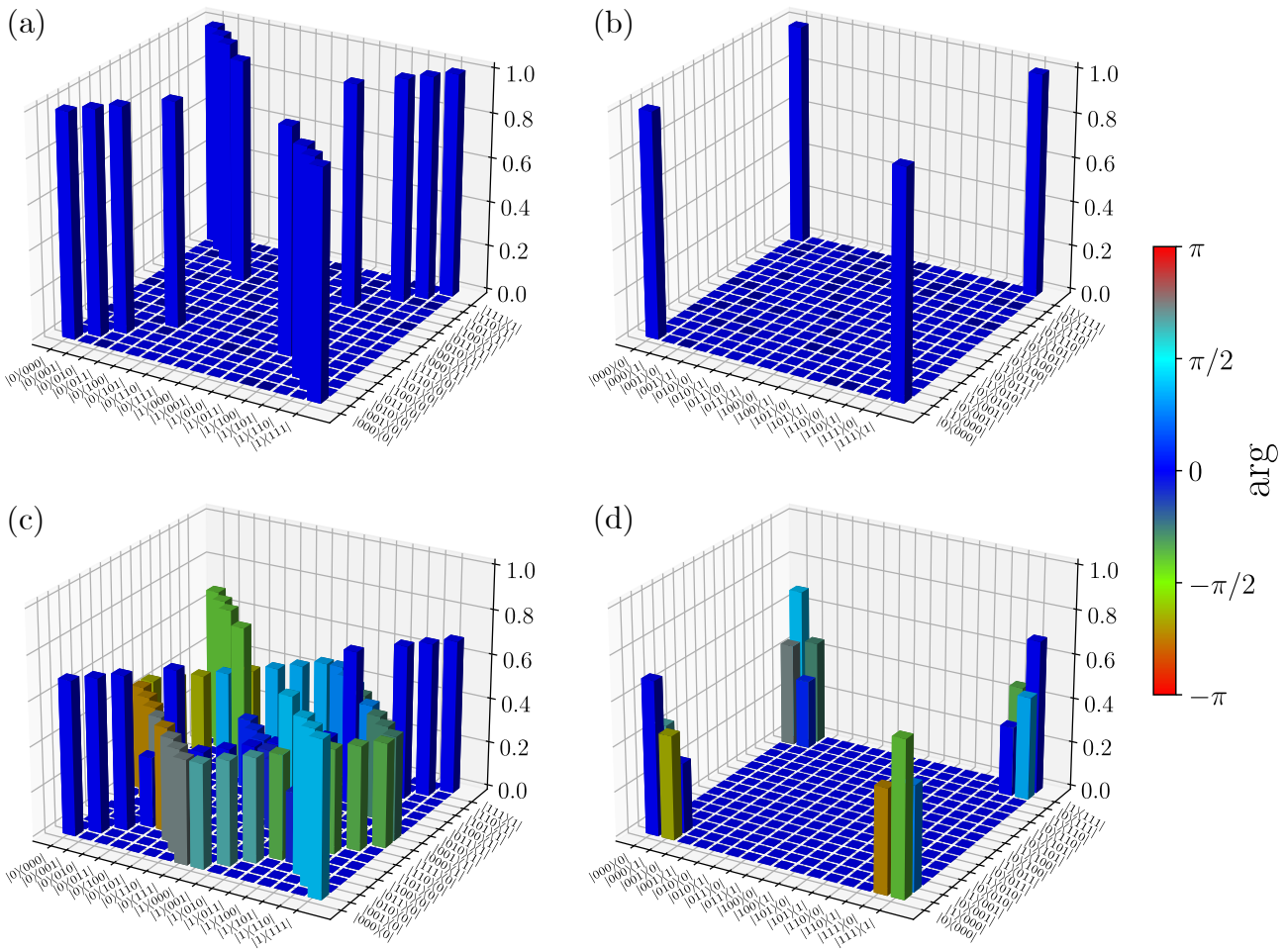


Figure 13: (a) Quantum process tomography of the quantum channel described by the Kraus operators in Eq. (18) which simultaneously corrects single bit flips on logical states of the 3QC and compresses them to single-qubit states. (b) Process matrix of the channel with Kraus operator in Eq. (19) which reconstructs logical states of the 3QC. (c) Process matrix of the 3-1 encoding channel of a trained QAE. (d) Quantum process tomography of the 1-3 decoding channel of a trained QAE. Performing the maps in (c) and (d) successively results in the same map that is obtained when the channels in (a) and (b) are applied one after another.

For pure states $|\psi_{\mathrm{L}}\rangle = \cos(\theta/2)\,|0_{\mathrm{L}}\rangle + e^{i\phi}\sin(\theta/2)\,|1_{\mathrm{L}}\rangle$, the averaged fidelity of denoised states w.r.t. the noise-free target states therefore reads

$$\langle \mathcal{F} \rangle = \iint \frac{\mathrm{d}\theta\,\mathrm{d}\phi}{4\pi} \sin(\theta)\,\langle \psi_{\mathrm{L}}|\,\rho_{\mathrm{denoised}}\,|\psi_{\mathrm{L}}\rangle = 1 - \frac{2}{3}p_{\mathrm{L}}. \tag{17}$$

Another point which we discuss in the main text is that the encoding channel of a 3-1-3 QAE conducts the combined compression and correction of erroneous input states while the decoding channel performs a trivial reconstruction of logical states. The single-qubit intermediate state can, however, be arbitrarily rotated against the logical input/output state. This can be seen from quantum process tomographies of the individual encoding and decoding channels for various trained 3-1-3 QAEs. Fig. 13(a) shows the process matrix of a handmade encoding channel described by the Kraus operators

$$\begin{aligned} M_{00} &= |0\rangle\langle 000| + |1\rangle\langle 111| \\ M_{01} &= |0\rangle\langle 001| + |1\rangle\langle 110| \\ M_{10} &= |0\rangle\langle 100| + |1\rangle\langle 011| \\ M_{11} &= |0\rangle\langle 010| + |1\rangle\langle 101|\,. \end{aligned} \tag{18}$$

The quantum process tomography of the corresponding decoding channel, characterized by a single Kraus operator

$$M_0 = |000\rangle\langle 0| + |111\rangle\langle 1|\,, \tag{19}$$

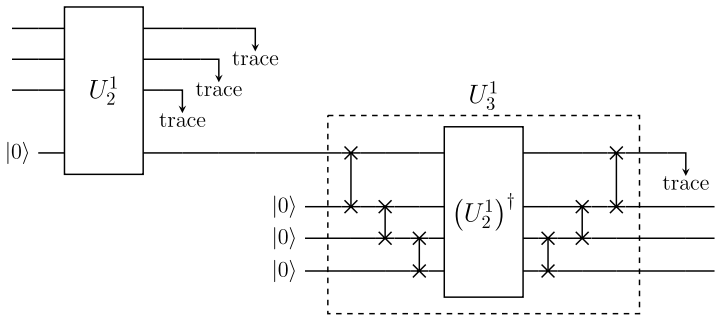


Figure 14: Quantum circuit realizing a 3-1-3 QAE with self-inverse architecture. The inverse of the matrix U_2^1 which occurs in the encoding channel is employed in the decoding channel. Since the three qubits belonging to the output layer are added after the intermediate qubit, a set of swap gates is required to permute the qubit indices.

is shown in Fig. 13(b). For an exemplary 3-1-3 QAE trained to implement the 3QC error correction map, Figs. 13(c) and (d) show process matrices of the respective encoding and decoding channels. The concatenation of the two maps yields the channel given by Eq. (14). Compared to the handmade channels we see, however, that the single-qubit intermediate state is rotated against the logical input and output state.

B Training QAEs with Self-inverse Architecture

Beer *et al.* [49] describe how DQNNs can be trained efficiently in numerical simulations. At time step s the gradient descent step is performed by updating all unitary matrices in the network according to the rule

$$U_k^{j_k}(s+\epsilon) = e^{i\epsilon K_k^{j_k}(s)} U_k^{j_k}(s), \tag{20}$$

where the update matrices $K_k^{j_k}$ are chosen such that the cost function

$$C(s) = 1 - \frac{1}{N}\sum_{i=1}^{N} \left\langle \phi_{\text{targ}}^i \right| \rho_{\text{out}}^i(s) \left| \phi_{\text{targ}}^i \right\rangle \tag{21}$$

is reduced as fast as possible. A careful analysis yields a simple rule to calculate the update matrices $K_k^{j_k}$, resembling a backpropagation algorithm. The training input states ρ_{in}^i are propagated forward through the network while the corresponding target states $\left|\phi_{\text{targ}}^i\right\rangle$ are propagated backwards. Commutators between the density matrices of the forward- and backpropagted states yield the quantities

$$\begin{aligned} M_k^{j_k}(s,i) = \Big[& U_k^{j_k}(s) \dots U_2^1(s) \Big(\rho_{\text{in}}^i \otimes |0\dots0\rangle\langle 0\dots 0|_{\text{in,hidden}} \Big) \big(U_2^1(s)\big)^\dagger \dots \big(U_k^{j_k}(s)\big)^\dagger \,, \\ & \big(U_k^{j_k+1}(s)\big)^\dagger \dots \big(U_{\text{out}}^{j_{\text{out}}^{\max}}(s)\big)^\dagger \Big(\mathbb{1}_{\text{in,hidden}} \otimes \left|\phi_{\text{targ}}^i\right\rangle\!\left\langle\phi_{\text{targ}}^i\right| \Big) U_{\text{out}}^{j_{\text{out}}^{\max}}(s) \dots U_k^{j_k+1}(s) \Big], \end{aligned} \tag{22}$$

which are involved in the calculation of the update matrices:

$$K_k^{j_k}(s) = i\frac{\dim(U_k^{j_k})}{2N} \sum_{i=1}^{N} \operatorname*{Tr}_{\substack{\text{qubits} \\ \text{not in } U_k^{j_k}}} \Big[M_k^{j_k}(s,i) \Big]. \tag{23}$$

For a detailed description the reader is referred to Ref. [49].

However, for QAEs with self-inverse architecture used in this paper the training algorithm requires some modifications which we describe in the following. These adjustments arise from the fact that the same unitary matrices occur at several positions in the network. For a numerical implementation it is convenient to always introduce new qubits of succeeding layers at the tail end of the latest state. Therefore, the unitary matrices used for the decoding channel are not just the inverses of the previously used matrices but they come with a set of swap gates, as depicted in Fig. 14. However, these gates only perform a permutation of qubit indices, so they do not have to be applied physically in the network. In a QAE with self-inverse architecture, a unitary matrix $U_k^{j_k}$ appearing in the encoding channel occurs in the decoder as the $\bar{j}_k$-th matrix realizing the transition to layer $\bar{k}$. Abbreviating the necessary swap gates as $S_k^{j_k}$ yields

$$U_{\bar{k}}^{\bar{j}_k} = S_k^{j_k} \big(U_k^{j_k}\big)^\dagger \big(S_k^{j_k}\big)^\dagger. \tag{24}$$

We consider QAEs where the unitaries of the encoder act on all qubits of the preceding layer and a single qubit in the succeeding layer. For a QAE consisting of L layers (including the input layer), one finds $\bar{k} = L + 2 - k$ and $\bar{j}_k = n_k + 1 - j_k$, where n_k is the width of the k-th layer. Identical matrices occurring at several positions within the DQNN require the training update rules to be modified. The goal is to find update matrices $K_k^{j_k}$ such that the unitary matrices from the encoder are again updated according to the rule

$$U_k^{j_k}(s+\epsilon) = e^{i\epsilon K_k^{j_k}(s)}\, U_k^{j_k}(s). \tag{25}$$

The updates of the unitary matrices assembling the decoder, however, follow from the updated encoding matrices:

$$\begin{aligned} U_{\bar{k}}^{\bar{j}_k}(s+\epsilon) &= S_k^{j_k}\left(U_k^{j_k}(s+\epsilon)\right)^\dagger \left(S_k^{j_k}\right)^\dagger \\ &= S_k^{j_k}\left(\left(U_k^{j_k}(s)\right)^\dagger e^{-i\epsilon K_k^{j_k}(s)}\right)\left(S_k^{j_k}\right)^\dagger \\ &= \left(S_k^{j_k}\left(U_k^{j_k}(s)\right)^\dagger e^{-i\epsilon K_k^{j_k}(s)}\, U_k^{j_k}(s)\left(S_k^{j_k}\right)^\dagger\right) U_{\bar{k}}^{\bar{j}_k}(s). \end{aligned} \tag{26}$$

We now derive how the update matrices $K_k^{j_k}$ are calculated such that the cost function C is reduced as quickly as possible. An expression for $\mathrm{d}C/\mathrm{d}s$ can be obtained:

$$\begin{aligned} \frac{\mathrm{d}C(s)}{\mathrm{d}s} = \frac{-i}{N}\sum_{i=1}^{N} \mathrm{Tr}\Big[& M_2^1(s,i)K_2^1(s) + \ldots + M_{(L+1)/2}^{j^{\max}}(s,i)K_{(L+1)/2}^{j^{\max}}(s) \\ & + M_{(L+3)/2}^1(s,i)J_{(L+3)/2}^1(s) + \ldots + M_{\mathrm{out}}^{j^{\max}}(s,i)J_{\mathrm{out}}^{j^{\max}}(s)\Big]. \end{aligned} \tag{27}$$

Here, $M_k^{j_k}$ is defined in Eq. (22) and the quantities $J_{\bar{k}}^{\bar{j}_k}$ are given by

$$J_{\bar{k}}^{\bar{j}_k} = -S_k^{j_k}\left(U_k^{j_k}\right)^\dagger K_k^{j_k}\, U_k^{j_k}\left(S_k^{j_k}\right)^\dagger. \tag{28}$$

A matrix $K_k^{j_k}$ thus occurs twice in the expression for $\mathrm{d}C/\mathrm{d}s$. Minimizing it therefore yields additional terms in the expressions for $K_k^{j_k}$:

$$K_k^{j_k}(s) = i\frac{\dim(U_k^{j_k})}{2N}\sum_{i=1}^{N} \underset{\substack{\text{qubits}\\ \text{not in } U_k^{j_k}}}{\mathrm{Tr}}\left[M_k^{j_k}(s,i)\right] - \underset{\substack{\text{qubits}\\ \text{not in } U_{\bar{k}}^{\bar{j}_k}}}{\mathrm{Tr}}\left[U_k^{j_k}(s)\left(S_k^{j_k}\right)^\dagger M_{\bar{k}}^{\bar{j}_k}(s,i)\, S_k^{j_k}\left(U_k^{j_k}(s)\right)^\dagger\right]. \tag{29}$$

An update matrix $K_k^{j_k}$ can therefore still be obtained by forward- and backpropagation of the training states. However, since the corresponding unitary matrix $U_k^{j_k}$ occurs at two positions in the network, it is necessary to propagate the training states to both places, for calculating the respective commutators between forward- and backpropagated states and constructing the update matrix for that specific unitary.

C Training Specifications

Here we give details on the training of the DQNNs discussed in the main text. All networks were trained numerically using the training algorithm described in Appendix B. A good overview of gradient descent variants for the training of classical neural networks can be found in a review by Ruder [99]. The set of states being used for the training of a DQNN is called training batch. We divide the batch into minibatches of size $S_{\mathrm{minibatch}}$ and perform a gradient descent step after states from one minibatch were exposed to the network. Presenting all minibatches to the network is called a training epoch. After a training epoch we shuffle all states in the batch, create new minibatches and continue the training for a total number of N_{epochs} training epochs. The gradient descent algorithm uses a learning rate ϵ. Moreover, we use the Nadam gradient descent optimizer with memory coefficients $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to achieve better training convergence [60, 99]. Table 1 summarizes the hyperparameters that were used for the training of the QAEs discussed in the paper.

	ϵ	N_{epochs}	$S_{\mathrm{minibatch}}$	training batch
Fig. 3	0.1	200	3	$\lvert 0_{\mathrm{L}}\rangle$, $\lvert 1_{\mathrm{L}}\rangle$, $\lvert +_{\mathrm{L}}\rangle$
Fig. 5	0.2	200	2	$\lvert 0_{\mathrm{L}}\rangle$, $\lvert 1_{\mathrm{L}}\rangle$, $\lvert +_{\mathrm{L}}\rangle$, $\lvert -_{\mathrm{L}}\rangle$, $\lvert +'_{\mathrm{L}}\rangle$, $\lvert -'_{\mathrm{L}}\rangle$
Fig. 6	0.1	200	3	$\lvert 0_{\mathrm{L}}\rangle$, $\lvert 1_{\mathrm{L}}\rangle$, $\lvert +_{\mathrm{L}}\rangle$, $\lvert -_{\mathrm{L}}\rangle$, $\lvert +'_{\mathrm{L}}\rangle$, $\lvert -'_{\mathrm{L}}\rangle$
Fig. 8	0.1	200	3	$\lvert 0_{\mathrm{L}}\rangle$, $\lvert 1_{\mathrm{L}}\rangle$, $\lvert +_{\mathrm{L}}\rangle$, $\lvert -_{\mathrm{L}}\rangle$, $\lvert +'_{\mathrm{L}}\rangle$, $\lvert -'_{\mathrm{L}}\rangle$ (50 each)

Table 1: Summary of training hyperparameters.

Note that the training batch lists the noise-free training states. These states undergo noise as described in the main text before serving as inputs for the QAEs. Computational errors are applied by subjecting the training states to the corresponding noise channel, thus only one copy of each training state is contained in the batch. Erasures, however, are applied probabilistically. To achieve good statistics we therefore include several copies of each state in the batch when training a collection of QAEs to correct losses of qubits.

D Discovered Encodings

As described in Sec. 4 of the main text, certain types of DQNNs can be used to discover logical encodings that optimally protect quantum information from specific kinds of noise. As an example we trained a collection of 1-4-1 DQNNs to find a logical encoding that perfectly protects states from collective desphasing and furthermore allows for single erasures of qubits to be corrected. For the training we use a learning rate $\epsilon = 0.1$ and a batch containing 50 of each of the states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |+'\rangle, |-'\rangle\}$. Minibatches contain 100 states and we train for a total number of 150 epochs. The logical codespace discovered by the network is spanned by the states

$$\begin{aligned} |0_{\mathrm{L}}\rangle &= (0.50+0.00i)\,|0011\rangle + (0.28+0.25i)\,|0101\rangle - (0.29+0.12i)\,|0110\rangle \\ &\quad + (0.33+0.15i)\,|1001\rangle - (0.30+0.25i)\,|1010\rangle - (0.31-0.36i)\,|1100\rangle\,, \\ |1_{\mathrm{L}}\rangle &= (0.15+0.46i)\,|0011\rangle + (0.07-0.45i)\,|0101\rangle - (0.14-0.23i)\,|0110\rangle \\ &\quad + (0.10-0.23i)\,|1001\rangle - (0.05-0.41i)\,|1010\rangle - (0.48+0.15i)\,|1100\rangle\,. \end{aligned} \tag{30}$$

Note that the vector amplitudes shown here are rounded and components whose squared modulus is smaller than 10^{-3} are omitted. One can clearly see that quantum information is encoded in a DFS. Logical states $|\psi_{\mathrm{L}}\rangle = \alpha\,|0_{\mathrm{L}}\rangle + \beta\,|1_{\mathrm{L}}\rangle$ are not altered by the application of a unitary $e^{-i\frac{\alpha}{2}(Z_1+Z_2+Z_3+Z_4)}$ since all computational basis states involved in the logical basis are eigenstates of the "total magnetization" $Z_1+Z_2+Z_3+Z_4$ with eigenvalue zero. Testing the collection of QAEs that results from the trained networks to correct erasures on randomly drawn logical states $|\psi_{\mathrm{L}}\rangle$ we find averaged fidelities between denoised states and corresponding target states that are shown in the first line of Table 2. The data indicates that erasures of qubits can be corrected very well. Given a state $|\psi_{\mathrm{L}}\rangle$, one therefore expects that the marginal state on any single qubit, $\rho_i = \mathrm{Tr}_{\{\text{code qubits}\}\setminus i}(|\psi_{\mathrm{L}}\rangle\langle\psi_{\mathrm{L}}|)$, is maximally mixed and thus the loss of a single qubit does not erase the encoded quantum information. In the second line of Table 2 we show averaged fidelities of those marginal states w.r.t. the maximally mixed state. The fact that they are not exactly equal to one indicates a slight dependence of a marginal state on the coefficients α and β of the original logical state.

	no losses	qubit 1 lost	qubit 2 lost	qubit 3 lost	qubit 4 lost
$\langle\mathcal{F}\rangle$	1.0000	0.9997	0.9990	0.9997	0.9995
$\langle\mathcal{F}(\rho_i, \mathbb{1}/2)\rangle$	–	0.9999	0.9995	0.9998	0.9997

Table 2: The first line shows how well a collection of x-1-4 QAEs resulting from trained encoding finder networks can denoise random logical states that suffered collective dephasing and qubit erasures. Each network from the collection is tested on 2000 validation states subjected to collective dephasing of strength $\sigma = 1$ and a corresponding loss. The second line shows the potential of the logical encoding to correct single erasure events. This is determined from the averaged fidelity of marginal states on a single qubit, $\rho_i = \mathrm{Tr}_{\{\text{code qubits}\}\setminus i}(|\psi_{\mathrm{L}}\rangle\langle\psi_{\mathrm{L}}|)$, w.r.t. the maximally mixed state $\mathbb{1}/2$.

E Analytical Approach for Noisy QAEs

In Sec. 5 of the main text we investigate the quality of a quantum memory that uses an intrinsically noisy QAE for active quantum error correction. We compare it to a bare physical qubit and to an encoded but uncorrected logical qubit. Here, we derive analytical expressions for the averaged probability of successful state discrimination that serves as a measure for the quality of such a memory.

A single physical qubit that is subjected to two rounds of bit flip noise of strength p_{i}, corresponding to two rounds of idling for a time $\tau/2$, suffers a bit flip with probability

$$p_{\mathrm{single}} = 2p_{\mathrm{i}}(1-p_{\mathrm{i}}). \tag{31}$$

The probability of successful state discrimination is given by the fidelity of the final state w.r.t. the noise-free

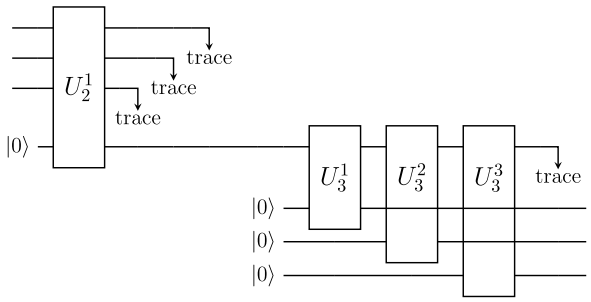


Figure 15: Quantum circuit implementing a 3-1-3 QAE.

initial state $|\psi\rangle = \cos(\theta/2)\,|0\rangle + e^{i\phi}\sin(\theta/2)\,|1\rangle$. Averaging uniformly over all states on the Bloch sphere yields

$$\begin{aligned}\langle \mathcal{P}_{\text{single}} \rangle =& \iint \frac{d\theta\, d\phi}{4\pi} \sin(\theta) \Big[(1 - p_{\text{single}})\, \langle\psi|\psi\rangle\langle\psi|\psi\rangle + p_{\text{single}}\ \langle\psi|X|\psi\rangle\langle\psi|X|\psi\rangle \Big] \\ =& (1 - p_{\text{single}}) + \frac{1}{3} p_{\text{single}} \\ =& 1 - \frac{4}{3} p_{\text{i}} \left(1 - p_{\text{i}}\right).\end{aligned} \tag{32}$$

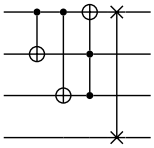
For an encoded logical qubit that is exposed to two rounds of bit flip noise and then subjected to a perfect round of QEC, the logical error rate is

$$\begin{aligned} p_{\text{uncorr.}} &= p_{\text{single}}^3 + 3 p_{\text{single}}^2 \left(1 - p_{\text{single}}\right) \\ &= 12 p_{\text{i}}^2 (1 - p_{\text{i}})^4 + 8 p_{\text{i}}^3 (1 - p_{\text{i}})^3 + 12 p_{\text{i}}^4 (1 - p_{\text{i}})^2, \end{aligned} \tag{33}$$

where p_{single} is given by Eq. (31). Thus, the averaged probability of successful state discrimination reads

$$\begin{aligned}\langle \mathcal{P}_{\text{uncorr.}} \rangle =& \iint \frac{d\theta\, d\phi}{4\pi} \sin(\theta) \Big[(1 - p_{\text{uncorr.}})\, \langle\psi_{\text{L}}|\psi_{\text{L}}\rangle\langle\psi_{\text{L}}|\psi_{\text{L}}\rangle + p_{\text{uncorr.}}\ \langle\psi_{\text{L}}|X_{\text{L}}|\psi_{\text{L}}\rangle\langle\psi_{\text{L}}|X_{\text{L}}|\psi_{\text{L}}\rangle \Big] \\ =& (1 - p_{\text{uncorr.}}) + \frac{1}{3} p_{\text{uncorr.}} \\ =& 1 - 8 p_{\text{i}}^2 + \frac{80}{3} p_{\text{i}}^3 - 40 p_{\text{i}}^4 + 32 p_{\text{i}}^5 - \frac{32}{3} p_{\text{i}}^6.\end{aligned} \tag{34}$$

Finally, we consider an intrinsically noisy 3-1-3 QAE that is used for QEC after the first round of bit flip noise. For weak internal noise, $p_{\text{n}} \ll 1$, output states of the QAE can be well approximated by expanding to linear order in p_{n}. We consider a hand-constructed QAE that implements the 3-qubit error correction channel stated in Eq. (14). The QAE consists of four unitary matrices, as shown in Fig. 15, where

$U_2^1 =$ 

and $U_3^1 = \text{CNOT}$, $U_3^2 = \text{CNOT}$, $U_3^3 = \text{SWAP}$. Each unitary matrix is followed by a multi-qubit depolarizing channel. In the absence of internal noise, a logical state corrupted by bit flip noise is mapped to a logical state ρ_{L} by the QAE. At the four positions in the circuit where noise is acting, the respective intermediate quantum state remains unaffected with probability $(1-p_{\text{n}})$ and suffers an error with probability p_{n}. Expanding an output state of the network to linear order in p_{n} therefore involves errors at one position in the circuit at most. For the analysis it turns out to be convenient to write the depolarizing channel acting on m qubits in the form

$$\mathcal{N}_{p_{\text{n}}}(\rho) = \left(1 - \frac{4^m\, p_{\text{n}}}{4^m - 1}\right) \rho + \left(\frac{4^m\, p_{\text{n}}}{4^m - 1}\right) \frac{I^{\otimes m}}{2^m}. \tag{35}$$

Denoting the output state of the noise-free QAE as ρ_{L}, one can write the output states of the circuit with only one of the four noise channels being present as follows:

$$\rho_{\text{out}}^{\text{noise 1}} = \left(1 - \frac{256}{255} p_{\text{n}}\right) \rho_{\text{L}} + \frac{256}{255} p_{\text{n}}\, \frac{|000\rangle\langle 000| + |111\rangle\langle 111|}{2}, \tag{36}$$

$$\rho_{\text{out}}^{\text{noise 2}} = \left(1 - \frac{16}{15}p_{\text{n}}\right)\rho_{\text{L}} + \frac{16}{15}p_{\text{n}}\,\frac{|000\rangle\langle 000| + |011\rangle\langle 011| + |100\rangle\langle 100| + |111\rangle\langle 111|}{4}, \tag{37}$$

$$\rho_{\text{out}}^{\text{noise 3}} = \left(1 - \frac{16}{15}p_{\text{n}}\right)\rho_{\text{L}} + \frac{16}{15}p_{\text{n}}\left(\langle 00_{23}|\,\rho_{\text{L}}\,|00_{23}\rangle + \langle 11_{23}|\,\rho_{\text{L}}\,|11_{23}\rangle\right) \otimes \frac{I^{\otimes 2}}{4}, \tag{38}$$

$$\rho_{\text{out}}^{\text{noise 4}} = \left(1 - \frac{16}{15}p_{\text{n}}\right)\rho_{\text{L}} + \frac{16}{15}p_{\text{n}}\left(\langle 0_3|\,\rho_{\text{L}}\,|0_3\rangle + \langle 1_3|\,\rho_{\text{L}}\,|1_3\rangle\right) \otimes \frac{I}{2}. \tag{39}$$

Collecting terms yields an approximate expression for a noisy output state of the quantum autoencoder that is linear in p_{n}, resulting from the circuit with all four noise channels present:

$$\rho_{\text{out}}^{\text{noisy}} \approx \rho_{\text{out}}^{\text{noise 1}} + \rho_{\text{out}}^{\text{noise 2}} + \rho_{\text{out}}^{\text{noise 3}} + \rho_{\text{out}}^{\text{noise 4}} - 3\rho_{\text{L}}. \tag{40}$$

To account for the round of bit flip noise before the application of the noisy QAE, we substitute $\rho_{\text{L}} = (1 - p_{\text{L}})\,|\psi_{\text{L}}\rangle\langle\psi_{\text{L}}| + p_{\text{L}} X_{\text{L}}\,|\psi_{\text{L}}\rangle\langle\psi_{\text{L}}|\,X_{\text{L}}$ into Eq. (40), where $p_{\text{L}} = p_{\text{i}}^3 + 3p_{\text{i}}^2(1-p_{\text{i}})$. After applying a second round of bit flip noise and performing a perfect round of QEC, we take the fidelity of the resulting state w.r.t. the original noise-free state $|\psi_{\text{L}}\rangle$ to obtain the probability of successful state discrimination. Averaging uniformly over all states on the Bloch sphere yields

$$\begin{aligned}\langle \mathcal{P}_{\text{corr.}} \rangle = &\left(1 - 4p_{\text{i}}^2 + \frac{8}{3}p_{\text{i}}^3 + 12p_{\text{i}}^4 - 16p_{\text{i}}^5 + \frac{16}{3}p_{\text{i}}^6\right) \\ &- p_{\text{n}}\left(\frac{156}{85} + \frac{8}{15}p_{\text{i}} - \frac{776}{51}p_{\text{i}}^2 + \frac{5312}{765}p_{\text{i}}^3 + \frac{13408}{255}p_{\text{i}}^4 - \frac{17152}{255}p_{\text{i}}^5 + \frac{17152}{765}p_{\text{i}}^6\right).\end{aligned} \tag{41}$$

Equating Eq. (41) with Eq. (32) or Eq. (34) and solving for p_{n} gives rise to analytical expressions for the phase boundaries as shown in Fig. 12 in the main text:

$$p_{\text{n}}^{\text{crit., single}} = \frac{255\left(p_{\text{i}} - 4p_{\text{i}}^2 + 2p_{\text{i}}^3 + 9p_{\text{i}}^4 - 12p_{\text{i}}^5 + 4p_{\text{i}}^6\right)}{351 + 102p_{\text{i}} - 2910p_{\text{i}}^2 + 1328p_{\text{i}}^3 + 10056p_{\text{i}}^4 - 12864p_{\text{i}}^5 + 4288p_{\text{i}}^6} = \frac{255}{351}p_{\text{i}} + \mathcal{O}(p_{\text{i}}^2) \tag{42}$$

and

$$p_{\text{n}}^{\text{crit., logical}} = \frac{765\left(p_{\text{i}}^2 - 6p_{\text{i}}^3 + 13p_{\text{i}}^4 - 12p_{\text{i}}^5 + 4p_{\text{i}}^6\right)}{351 + 102p_{\text{i}} - 2910p_{\text{i}}^2 + 1328p_{\text{i}}^3 + 10056p_{\text{i}}^4 - 12864p_{\text{i}}^5 + 4288p_{\text{i}}^6} = \frac{765}{351}p_{\text{i}}^2 + \mathcal{O}(p_{\text{i}}^3). \tag{43}$$