

Quantum autoencoders for efficient compression of quantum data

Jonathan Romero,¹ Jonathan Olson,¹ and Alan Aspuru-Guzik^{1,*}

¹Department of Chemistry and Chemical Biology, Harvard University, Cambridge, Massachusetts 02138, United States

(Dated: December 9, 2016)

Classical autoencoders are neural networks that can learn efficient codings of large datasets. The task of an autoencoder is, given an input x , to simply reproduce x at the output with the smallest possible error. For one class of autoencoders, the structure of the underlying network forces the autoencoder to represent the data on a smaller number of bits than the input length, effectively compressing the input. Inspired by this idea, we introduce the model of a quantum autoencoder to perform similar tasks on quantum data. The quantum autoencoder is trained to compress a particular dataset, where an otherwise efficient compression algorithm cannot be employed. The training of the quantum autoencoder is done using classical optimization algorithms. We show that efficient quantum autoencoders can be obtained using simple circuit heuristics. We apply our model to compress molecular wavefunctions with applications in quantum simulation of chemistry.

I. INTRODUCTION

Quantum technologies, ranging from quantum computing to quantum cryptography, are being found to have increasingly powerful applications for a modern society. Quantum simulators for chemistry, for example, have been recently shown to be capable of efficiently calculating molecular energies for small systems [1]; the capability for larger scale simulations promises to have deep implications for materials design, pharmacological research, and an array of other potentially life-changing functions [2]. For nearly all of these applications, however, the limiting factor in scaling is the number of quantum resources that can be realized in an experiment. For experiments now and in the near future, then, any tool which can reduce the experimental overhead in terms of these resources is especially valuable.

For classical data processing, machine learning via an autoencoder is one such tool for dimensional reduction [3, 4], as well as having application in generative data models [5]. A classical autoencoder is a circuit that is optimized across a training set of data which, given an $(n+k)$ -bit input string x , attempts to reproduce x . Part of the specification of the circuit, however, is to erase some k bits during the process. If an autoencoder is successfully trained to reproduce x at the output (or something close to x), then the remaining n bits immediately after the erasure (referred to as the *latent space*) represent a compressed encoding of the string x . Thus, the circuit “learns” to encode information that is close to training set.

In this paper, we introduce the concept of a quantum autoencoder which is inspired by this design for an input of $n+k$ qubits. The motivation for a *quantum* autoencoder is simple; such a model allows us to perform analogous machine learning tasks for quantum systems without exponentially costly classical memory, for instance dimension reduction of quantum data. A related work proposing a quantum autoencoder model appeared on the arXiv e-print server very recently [6]. This work establishes a formal connection between classical and quantum feedforward neural networks. In this work, we

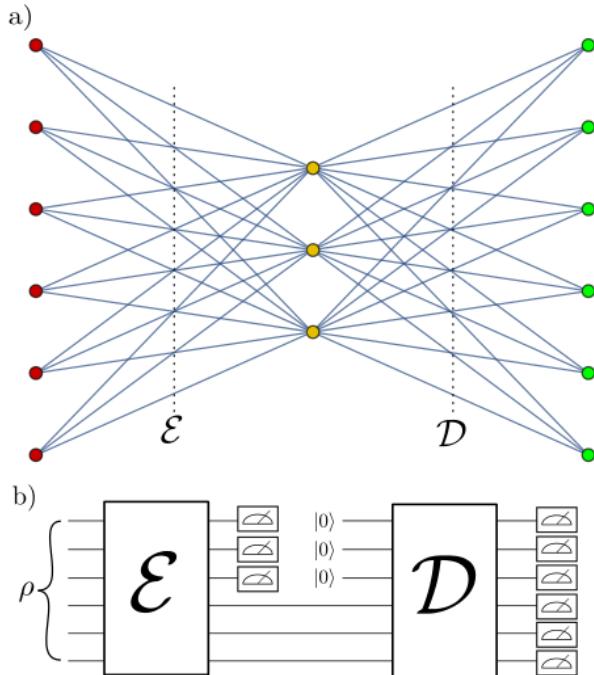


Figure 1. a) A graphical representation of a 6-bit autoencoder with a 3-bit latent space. The map \mathcal{E} encodes a 6-bit input (red dots) into a 3-bit intermediate state (yellow dots), after which the decoder \mathcal{D} attempts to reconstruct the input bits at the output (green dots). b) Circuit implementation of a 6-3-6 quantum autoencoder.

provide a straightforward model which we believe more easily captures the essence of an autoencoder, as well as suggesting a variety of different problems which quantum autoencoders may be employed to solve.

II. QUANTUM AUTOENCODER MODEL

In analogy with the model of classical autoencoders, the quantum network has a graphical representation consisting of an interconnected group of nodes. In the graph of the quantum network, each node represents a qubit, with the first layer of

* aspuru@chemistry.harvard.edu

the network representing the input register and the last layer representing the output register. In our representation, the edges connecting adjacent layers represent a unitary transformation from one layer to the next. Autoencoders, in particular, shrink the space between the first and second layer, as depicted in Figure 1a.

For a quantum circuit to embody an autoencoder network, the information contained in some of the input nodes must be discarded after the initial “encoding” \mathcal{E} . We imagine this takes place by tracing over the qubits representing these nodes (in Figure 1b, this is represented by a measurement on those qubits). Fresh qubits (initialized to some reference state) are then prepared and used to implement the final “decoding” evolution \mathcal{D} , which is then compared to the initial state.

The learning task for a quantum autoencoder is to find unitaries which preserve the quantum information of the input through the smaller intermediate latent space. To this end, it is important to quantify the deviation from the initial input state, $|\psi_i\rangle$, to the output, ρ_i^{out} . Here, we will use the expected fidelity [7] $F(|\psi_i\rangle, \rho_i^{out}) = \langle \psi_i | \rho_i^{out} | \psi_i \rangle$. We thus describe a successful autoencoding as one in which $F(|\psi_i\rangle, \rho_i^{out}) \approx 1$ for all the input states.

A more formal description of a quantum autoencoder follows: Let $\{p_i, |\psi_i\rangle_{AB}\}$ be an ensemble of pure states on $n+k$ qubits, where subsystems A and B are comprised of n and k qubits, respectively. Let $\{U^j\}$ be a family of unitary operators acting on $n+k$ qubits, and let $|a\rangle_{B'}$ be some fixed pure reference state of k qubits. Using classical learning methods, we wish to find the unitary U^j which maximizes the average fidelity, which we define to be the cost function,

$$C_1 = \sum_i p_i \cdot F(|\psi_i\rangle, \rho_{i,j}^{out}), \quad (1)$$

where,

$$\rho_{i,j}^{out} = (U^j)_{AB'}^\dagger \text{Tr}_B \left[U_{AB}^j [\psi_{iAB} \otimes a_{B'}] (U_{AB}^j)^\dagger \right] (U^j)_{AB'}, \quad (2)$$

and we have abbreviated $|\psi_i\rangle \langle \psi_i|_{AB} = \psi_{iAB}$ and $|a\rangle \langle a|_{B'} = a_{B'}$. Equivalently, the goal is to find the best unitary U^j which, on average, best preserves the input state of the circuit in Figure 2 where instead of tracing over the B system, we employ a swap gate and trace over the B' system.

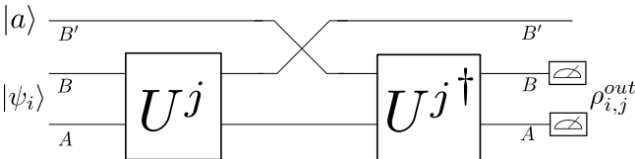


Figure 2. A quantum autoencoder circuit. The goal is to find j such that the averaged $F(|\psi\rangle_i, \rho_{i,j}^{out})$ is maximized.

To prove this, consider the fidelity of the input and output of the entire system of Figure 2 for some fixed j where we

have denoted the swap operation by the unitary V ,

$$\begin{aligned} & F(|\psi_i\rangle_{AB} \otimes |a\rangle_{B'}, U_{AB}^\dagger V_{BB'} U_{AB} |\psi_i\rangle_{AB} \otimes |a\rangle_{B'}) = \\ & F(U_{AB} |\psi_i\rangle_{AB} \otimes |a\rangle_{B'}, V_{BB'} U_{AB} |\psi_i\rangle_{AB} \otimes |a\rangle_{B'}) = \\ & F(|\psi_i'\rangle_{AB} \otimes |a\rangle_{B'}, V_{BB'} |\psi_i'\rangle_{AB} \otimes |a\rangle_{B'}) = \\ & F(|\psi_i'\rangle_{AB} \otimes |a\rangle_{B'}, |\psi_i'\rangle_{AB'} \otimes |a\rangle_B), \end{aligned} \quad (3)$$

where we have denoted $U|\psi_i\rangle = |\psi_i'\rangle$. The terms in the cost function are found by tracing out over the B' system,

$$\begin{aligned} & F(\text{Tr}_{B'} [\psi_{iAB}' \otimes a_{B'}], \text{Tr}_{B'} [\psi_{iAB'}' \otimes a_B]) = \\ & F(\psi_{iAB}'', \rho_A' \otimes a_B), \end{aligned} \quad (4)$$

where $\rho_A' = \text{Tr}_{B'} [|\psi_i'\rangle \langle \psi_i'|_{AB'}]$. However, consider instead tracing over the AB system and looking at the “trash system” of B' ,

$$\begin{aligned} & F(\text{Tr}_{AB} [\psi_{iAB}' \otimes a_{B'}], \text{Tr}_{AB} [\psi_{iAB'}' \otimes a_B]) = \\ & F(|a\rangle_{B'}, \rho_{B'}'), \end{aligned} \quad (5)$$

where $\rho_{B'}' = \text{Tr}_A [|\psi_i'\rangle \langle \psi_i'|_{AB'}]$. We henceforth refer to $\rho_{B'}'$ as the “trash state” of the circuit. It is straightforward to see in the above circuit that perfect fidelity (i.e. $C_1 = 1$) can be achieved by a unitary U if and only if, for all i :

$$U|\psi_i\rangle_{AB} = |\psi_i^c\rangle_A \otimes |a\rangle_B. \quad (6)$$

where $|\psi_i^c\rangle_A$ is some compressed version of $|\psi_i\rangle$. This follows because, if the B and B' systems are identical when the swap occurs, the entire circuit reduces to the identity map. However, this occurs precisely when the trash state is equal to the reference state, i.e., $F(|a\rangle_{B'}, \rho_{B'}') = 1$. This implies that it is possible to accomplish the learning task of finding the ideal U^j by training *only on the trash state*. Furthermore, because Eq. (6) is completely independent of U^\dagger , this suggests that the circuit of Figure 2 can be reduced further. We then consider an alternative definition of the cost function in terms of the trash state fidelity,

$$C_2 = \sum_i p_i \cdot F(\text{Tr}_A [U^j |\psi_i\rangle \langle \psi_i|_{AB} (U^j)^\dagger], |a\rangle_B), \quad (7)$$

Note, however, that the cost functions of Eq. (1) and Eq. (7) are not in general the same (in fact, $C_1 \leq C_2$).

It is interesting to note that, if we only care about circuits where $C_2 \approx 1$, we can re-imagine the problem to being one of finding a particular decoupling. Information-theoretic bounds have been explored in this context before, both in the context of one-shot compression and one-shot decoupling [8, 9]. However, because the heuristics involved in choosing efficient-to-implement families of unitaries are largely ad-hoc, it is difficult to say if these bounds are meaningful in the context of a quantum autoencoder.

III. IMPLEMENTATION OF THE QUANTUM AUTOENCODER MODEL

To implement the quantum autoencoder model on a quantum computer we must define the form of the unitary, U^j

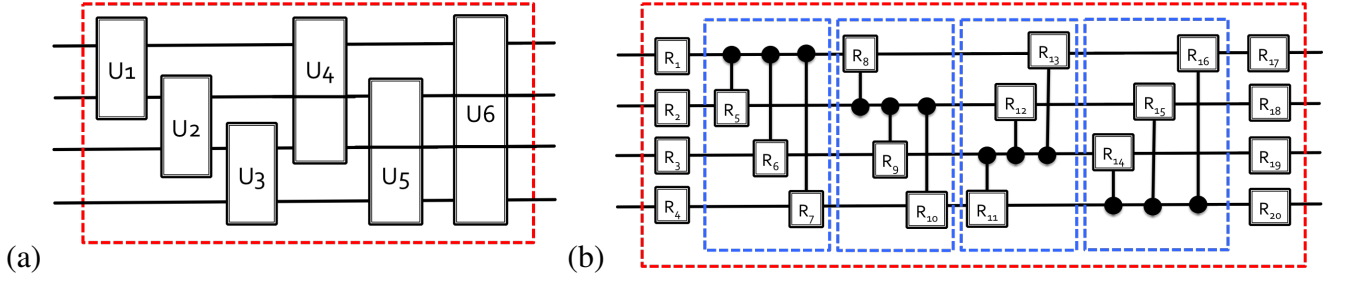


Figure 3. Two programmable circuits employed as autoencoder models: a) Circuit A: a network of all the possible two-qubit gates (denoted by U_i) among the qubits. b) Circuit B: a network comprising all the possible controlled general single-qubit rotations (denoted by R_i) in a qubit set, plus single qubit rotations at the beginning and at the end of the unit-cell. All the circuits are depicted in the case of a four-qubit input. The unit-cell is delimited by the red dotted line.

(Eq. (2)) and decompose it into a quantum circuit suitable for optimization. For the implementation to be efficient, the number of parameters and the number of gates in the circuit should scale polynomially with the number of input qubits. This requirement automatically discards the possibility of using a $(n+k)$ -qubit general unitary as U^j due to the exponential scaling in the number of parameters needed to generate them.

One alternative for the generation of U^j is to employ a programmable quantum circuit [10]. This type of circuit construction consists of a fixed networks of gates that can generate different unitaries by modifying a polynomial number of parameters, represented by the vector \vec{p} . The pattern defining the network of gates is regarded as a unit-cell. This unit-cell can ideally be repeated to increase the flexibility of the model. For the numerical assessment presented in this work, we employed two simple programmable circuits illustrated in Figure 3.

Circuit A has a unit-cell comprising a network of general two-qubit gates where we have considered all the possible pairings between qubits, as illustrated in Figure 3a for the four-qubit case. Accordingly, this model requires $15n(n-1)/2$ training parameters per unit-cell. A network of arbitrary two qubit gates can be easily implemented using state of the art superconducting qubit technologies [11] and the standard decomposition of a two-qubit gate into three CNOT gates and single-qubit rotations [12]. Arbitrary two qubit-gates has been also implemented using ion traps [13] and quantum dots [14].

Circuit B has a unit-cell comprising all the possible controlled one-qubit rotations among a set of qubits, complemented with a set of single qubit rotations at the beginning and at the end of the unit-cell, as shown in Figure 3b for the four-qubit case. We start considering the rotations controlled by the first qubit, followed by the rotations controlled by the second qubit and so on. Accordingly, our second model comprises $3n(n-1)+6n$ training parameters per unit-cell and can be implemented in state of the art quantum hardware using the standard decomposition of controlled unitaries into two CNOT gates and single-qubit rotations [15]. This model is also general and can be modified by adding constraints to the parameters. For instance, one could consider the initial and final layers of rotations to be all the same.

Once the circuit model has been chosen, we must train the

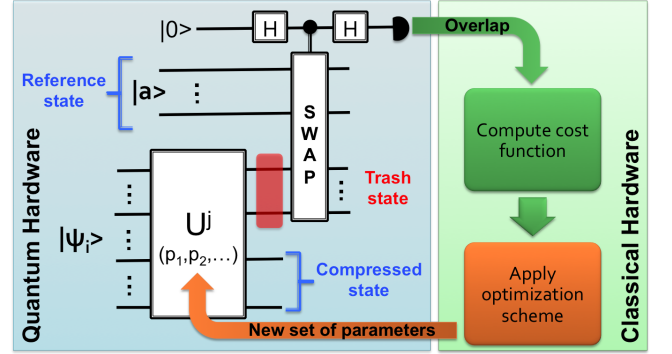


Figure 4. Schematic representation of the hybrid scheme for training a quantum autoencoder. After preparation of the input state, $|\psi_i\rangle$, the state is compressed by the application of the parameterized unitary, $U^j(\vec{p})$. The overlap between the reference state and the trash state produced by the compression is measured via a SWAP test. The results for all the states in the training set are collected to compute the cost function that is minimized using a classical optimization algorithm. The process is repeated until achieving convergence on the cost function and/or the values of the parameters, $\vec{p} = (p_1, p_2, \dots)$.

network by maximizing the autoencoder cost function Eq. (7), in close analogy to classical autoencoders. Our training procedure adopts a quantum-classical hybrid scheme in which the state preparation and measurement are performed on the quantum computer while the optimization is realized via an optimization algorithm running on a classical computer. Such hybrid schemes has been proposed in the context of quantum machine learning [16, 17] and variational algorithms for quantum simulation [18–20]. In the later case, several experimental demonstrations have been successfully carried out [18, 21, 22].

As described in Section II, the cost function of the quantum autoencoder is defined as the weighted average of fidelities between the trash state produced by the compression, and the reference state. These fidelities can be measured via a SWAP test [23] between the reference state and the trash state. Accordingly, our quantum register must comprise the input state, $|\psi_i\rangle$, and the reference state. In a single iteration of our training algorithm, we perform the following steps for each of the states in the training set:

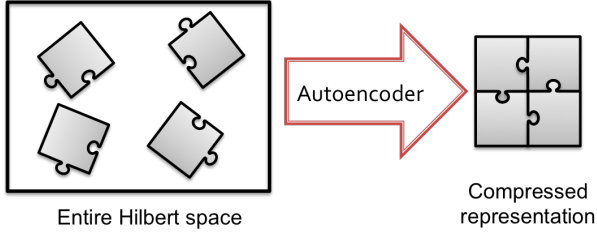


Figure 5. Graphical representation of Hilbert space compression. Given that the set of states of interest have support on only a subset \mathcal{S} of the Hilbert space (gray pieces), the quantum autoencoder finds an encoding that uses a space of size $|\mathcal{S}|$.

1. Prepare of the input state, $|\psi_i\rangle$ and the reference state. We assume these preparations to be efficient.
2. Evolve under the encoding unitary, $U^j(\vec{p})$, where \vec{p} is the set of parameters at a given optimization step.
3. Measure the fidelity between the trash state and the reference state via a SWAP test.

With the estimates of all the fidelities, the cost function (Eq. (7)) is computed and fed into a classical optimization routine that returns a new set of parameters for our compression circuit. These steps are repeated until the optimization algorithm converges. Given that value of the cost function ranges from 0 to 1, we can minimize the log of the cost function instead. This procedure is widely used in machine learning applications and helps prevent numerical instabilities [24]. A graphical summary of the hybrid scheme for training a quantum autoencoder is shown in Figure 4.

IV. APPLICATION TO QUANTUM SIMULATION

Consider a set of states, $\{|\psi_i\rangle\}$, with support on a subset of a Hilbert space $\mathcal{S} \subset \mathcal{H}$. Using a quantum autoencoder, we could find an encoding scheme that employs only $\log |\mathcal{S}|$ qubits to represent the states instead of $\log |\mathcal{H}|$, with a trash state of size $\log |\mathcal{H} - \mathcal{S}|$. This idea is graphically depicted in Figure 5. This situation is usually encountered for eigenstates of many-body systems due to special symmetries.

For instance, Fermionic wavefunctions are eigenfunctions of the particle number operator, same as the Fermionic state vectors. Consequently, an eigenstate of a system with η particles is spanned exclusively by the subspace of Fermionic state functions with the same number of particles [25], that has size $\binom{N}{\eta}$ with N the number of Fermionic modes. This result has direct implications for the design of quantum algorithms for simulation, suggesting that the number of qubits required to store Fermionic wavefunctions could be reduced up to $\log \binom{N}{\eta}$ if an appropriate mapping can be found. The same situation is encountered for the spin projection operator, thus reducing the size of the subspace spanning a specific Fermionic wavefunction even further.

Generally, the number of particles in the system is part of the input for the quantum simulation problem. In many ap-

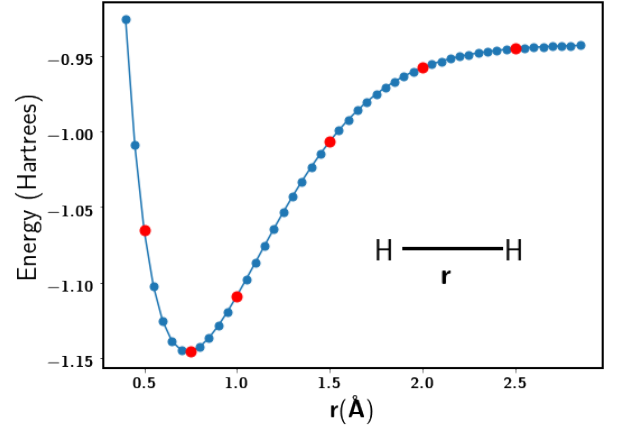


Figure 6. Potential energy surface for the hydrogen molecule using a STO-6G basis set. The ground states at the red dots were used as training set for the quantum autoencoder. The ground states at the blue dots were used for testing.

plications, such as quantum chemistry, the spin projection of the target state of the simulation might be also known. Many classical algorithms for simulating quantum systems take advantage of these constraints to reduce their computational cost [25]. However, the standard transformations employed to map Fermionic systems to qubits, namely the Bravyi-Kitaev and the Jordan-Wigner mappings [26, 27], do not exploit these symmetries and thus employ more qubits than formally needed.

In this scenario, a quantum autoencoder could be trained to compress the Fermionic wavefunctions obtained from a quantum simulation algorithm using the standard transformations. The compression schemes obtained through this procedure could be employed to reduce the use of quantum memory, if the wavefunction needs to be stored. It also could save quantum resources for the simulation of systems with similar symmetries. To illustrate this idea, we simulated a quantum autoencoder applied to molecular wavefunctions.

Within the Born-Oppenheimer approximation, the electronic Hamiltonian of a molecular system is represented by a sum of local terms, $H = \sum_i^M c_i H_i$, with M scaling as $O(N^4)$. H_i represent a set of local operators and c_i represent real coefficients. In a quantum computer, the operators H_i correspond to strings of Pauli matrices. The coefficient c_i are linear combinations of the one-electron and two-electron molecular integrals [25] and capture the electron-electron and the classical nuclear-electron interactions. For a fixed set of nuclei and a given number of electrons, the coefficients c_i are a function of the internal coordinates of the molecule, \vec{R} .

For instance, consider the Hamiltonian of molecular hydrogen in the STO-6G minimal basis set [25]. Using the Jordan-Wigner transformation, the corresponding Hamiltonian acting

on four qubits adopts the generic form [26]:

$$\begin{aligned}
 H = & c_0 I + c_1(Z_0 + Z_1) + c_2(Z_2 + Z_3) + c_3 Z_0 Z_1 + \\
 & c_4(Z_0 Z_2 + Z_1 Z_3) + c_5(Z_1 Z_2 + Z_0 Z_3) + c_6 Z_2 Z_3 \\
 & + c_7(Y_0 X_1 X_2 Y_3 - X_0 X_1 Y_2 Y_3 - Y_0 Y_1 X_2 X_3 + X_0 Y_1 Y_2 X_3)
 \end{aligned}
 \quad (8)$$

In this case, the coefficients c_i are a function of the internuclear distance, r . By solving the Schrödinger equation for the Hamiltonians at different values of r , we can obtain the ground state energy for molecular hydrogen and construct the potential energy surface (PES) shown in Figure 6. We expect that the ground state wavefunctions along the PES conserve the same number of particles and projection spin symmetries, turning this set of states into an excellent target for compression.

To illustrate the previous idea, we classically simulated a quantum autoencoder taking six ground states of the hydrogen molecule at different values of r , $\{|\Psi(r_i)\rangle\}_{i=1}^6$, as our training set. In this case, the weights of the states are chosen to be all equal. In real applications, we can imagine that the ground states are obtained using a quantum algorithm such as the quantum variational eigensolver [18]. We trained the circuit model described in Figure 3 to compress the training set of four-qubit states to two qubits and to one qubit, using $|0\rangle^{\otimes 2}$ and $|0\rangle^{\otimes 3}$ as reference states, respectively. Once the circuits were trained we tested them on 44 ground states corresponding to values of r different from those of the training set. This selection of the training and testing sets is shown in Figure 6.

The classical simulation was performed using a Python script supplemented with the QuTiP library [28, 29]. To simulate general two-qubit gates we employed the decomposition described in Ref.[13]. Arbitrary single-qubit rotations were implemented by decomposing them into Pauli-Z and Pauli-Y rotations, $R = R_z(\theta_1)R_y(\theta_2)R_z(\theta_3)$, ignoring global phases [15]. The optimization was performed using the SciPy implementation of the Basin-Hopping (BS) algorithm [30]. We also employed the L-BFGS-B method [31] with a numerical gradient (central difference formula with three points and spacing $h = 10^{-8}$). In the implementation of both circuit models, the parameters were constrained to the range $[0, 4\pi)$.

Table I shows the average error in the fidelities and the energies obtained after a cycle of compression and decompression through the optimal quantum autoencoder. We observe that both circuit models are able to achieve high fidelities for the encoding, producing decoded wavefunctions with energies that are close to the original values within chemical accuracy ($1\text{kcal/mol} \equiv 1.6 \times 10^{-3}$ Hartrees $\equiv 43.4$ meV). This accuracy requirement assures that quantum chemistry predictions have enough quality to be used for the estimation of thermochemical properties such as reaction rates [32].

Figure 7 illustrates the optimization process of a quantum autoencoder. We compared two different optimization algorithms, L-BFGS-B and Basin-Hopping. Both algorithms required a similar number of cost function evaluations to achieve similar precision and exhibit a monotonic reduction of the log of the cost function with the number of function evaluations. The implementation of the quantum autoencoder in state of the art architectures can benefit from the use of

Table I. Average fidelity error after one cycle of compression and decompression using the quantum autoencoder trained from ground states of the Hydrogen molecule. We also report the error in the energy of the decoded state. 6 states were used for training and 44 more were used for testing. Logarithms were taken to base 10.

Circuit	Final size # qubits	Set	log Fidelity MAE	log Energy MAE (Hartrees)
Model A	2	Training	-6.96 ± 0.12	-6.64 ± 0.78
	2	Testing	-6.99 ± 0.12	-6.76 ± 0.96
	1	Training	-6.92 ± 0.11	-6.60 ± 0.39
	1	Testing	-6.96 ± 0.11	-6.72 ± 0.52
Model B	2	Training	-6.11 ± 0.11	-6.00 ± 0.15
	2	Testing	-6.07 ± 0.12	-6.03 ± 0.15
	1	Training	-3.95 ± 1.08	-3.74 ± 0.50
	1	Testing	-3.81 ± 0.42	-3.62 ± 0.32

* MAE: Mean Absolute Error. Log chemical accuracy in Hartrees ≈ -2.80

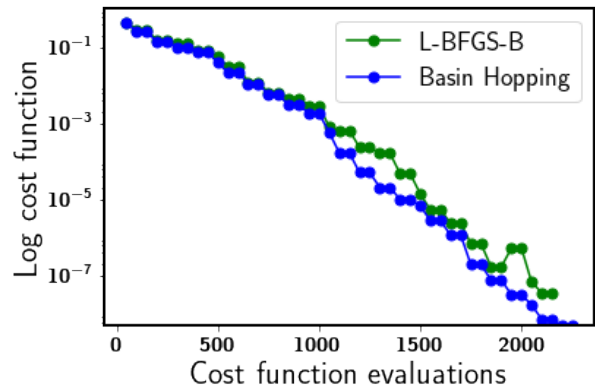


Figure 7. A plot of the cost function vs the number of cost function evaluations during the training process. This example corresponds to a quantum autoencoder to compress the wavefunction of molecular hydrogen from 4 to 2 qubits using circuit A. We compared the L-BFGS-B and the Basin-Hopping algorithms for optimization.

algorithms that do not require gradient evaluations and have a larger tolerance to the presence of noise in the hardware, such as Basin-Hopping. Numerical explorations, currently in progress, also indicate that some optimization techniques developed for training neural networks and based on Gaussian-processes [33] are promising for training quantum autoencoders.

To gain insights into the compression process, we plotted the density matrices of the compressed states and compared them with the density matrix of the original state in Figure 8, for three different values of r . The sparsity of the original input density matrix is due to the symmetry of the Hamiltonian for molecular hydrogen, whose eigenvectors have support on only 2 computational basis states, allowing for a compression up to a single qubit. Although the quantum autoencoder achieves high fidelity with both types of circuit, the structure of the density matrices indicates that the form of the compressed space and therefore the form of the encoding unitaries



Figure 8. Visualization of the input space and the latent (compressed) spaces for three different testing instances (different values of r), represented as the density matrices of the input and compressed states. Letters (A) and (B) denote the type of circuit employed to construct the encoding unitary. The size of the register (in qubits) appears within parenthesis. Integer labels starting at 1 denote the computational basis states from $|00 \dots 0\rangle$ to $|11 \dots 1\rangle$.

is different. As the values of r change, the relation between the features of the input space, here represented by the elements of the density matrix, and the features of the compressed space become apparent.

V. DISCUSSION

We have introduced a general model for a quantum autoencoder – a quantum circuit augmented with learning via classical optimization – and shown that it is capable of learning unitaries which can compress quantum data, particularly in the context of quantum simulations. We imagine that the model can be used to find other useful systems, such as compression protocols for quantum communication, error-correcting circuits, or perhaps to solve particular problems directly.

Are there any obvious limitations to this kind of quantum autoencoder? One consideration is that the von-Neumann entropy [7] of the density operator representing the ensemble $\{p_i, |\psi_i\rangle_{AB}\}$ limits the number of qubits to which it can be noiselessly compressed. However, finding the entropy of this density operator is not trivial – in fact, given a circuit that constructs a density operator ρ , it is known that, in general, even estimating the entropy of ρ is QSZK-complete [34]. This then opens the possibility for quantum autoencoders to efficiently give *some* estimate of the entropy of a density operators.

It is natural to consider whether the quantum autoencoder structure we have defined is actually a generalization of its classical counterpart, as in the construction of [6]. It may certainly be possible that some particular family of unitaries, to-

gether with certain choices for n and k , can be constructed so that a mapping exists. However, it is unclear that such a correspondence would even be desirable. Rather, we believe the value of autoencoders in general lies in the relatively simple structure of forcing a network to preserve information in a smaller subspace, as we have defined here.

Another topic of interest for any quantum computing model is the computational complexity exhibited by the device. For our construction, it is clear that any complexity result would be dependent upon the family of unitaries that is chosen for the learning to be optimized over. As the training itself is based on classical optimization algorithms (with no clear ‘optimal’ learning method), this further obfuscates general statements regarding the complexity of the model.

VI. ACKNOWLEDGEMENTS

The authors would like to thank Yudong Cao, Peter Johnson, Ian Kivlichan, Nicolas Sawaya, Libor Veis, and Dominic Berry for very insightful discussions and suggestions. JR and AAG acknowledge the Air Force Office of Scientific Research for support under Award: FA9550-12-1-0046. AA-G acknowledges the Army Research Office under Award: W911NF-15-1-0256. AA-G and JO acknowledge support from the Vannevar Bush Faculty Fellowship program sponsored by the Basic Research Office of the Assistant Secretary of Defense for Research and Engineering and funded by the Office of Naval Research through grant N00014-16-1-2008.

-
- [1] P. J. J. O’Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, et al., *Physical Review X* **6**, 031007 (2016).
 - [2] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, *Science* **309**, 1704 (2005).
 - [3] C.-Y. Liou, J.-C. Huang, and W.-C. Yang, *Neurocomputing* **71**, 3150 (2008).
 - [4] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, *Neurocomputing* **139**, 84 (2014).
 - [5] R. Gmez-Bombarelli, D. Duvenaud, J. M. Hernandez-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, *arXiv preprint cs.LG/1610.02415* (2016).
 - [6] K. H. Wan, O. Dahlsten, H. Kristjansson, R. Gardner, and M. Kim, *arXiv preprint arXiv:1612.01045* (2016).
 - [7] M. M. Wilde, *Quantum Information Theory* (Cambridge University Press, 2012).
 - [8] N. Datta, J. M. Renes, R. Renner, and M. M. Wilde, *IEEE Transactions on Information Theory* **59**, 8057 (2013).
 - [9] F. Dupuis, M. Berta, J. Wullschleger, and R. Renner, *Communications in Mathematical Physics* **328**, 251284 (2014).
 - [10] P. B. Sousa and R. V. Ramos, *arXiv preprint quant-ph/0602174* (2006).
 - [11] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, Y. Campbell Chen, et al., *Nature* **508**, 500 (2014).
 - [12] B. Kraus and J. Cirac, *Physical Review A* **63**, 062309 (2001).
 - [13] D. Hanneke, J. Home, J. Jost, J. Amini, D. Leibfried, and D. Wineland, *Nat. Phys.* **6**, 13 (2010).
 - [14] M. Veldhorst, C. Yang, J. Hwang, W. Huang, J. Dehollain, J. Muhonen, S. Simmons, A. Laucht, F. Hudson, K. M. Itoh, et al., *Nature* **526**, 410 (2015).
 - [15] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge university press, 2010).
 - [16] S. Gammelmark and K. Mølmer, *New. J. Phys* **11**, 033017 (2009).
 - [17] J. Bang, J. Ryu, S. Yoo, M. Pawłowski, and J. Lee, *New. J. Phys* **16**, 073017 (2014).
 - [18] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, *Nat. Commun.* **5**, 4213 (2014).
 - [19] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, *New. J. Phys* **18**, 023023 (2016).
 - [20] D. Wecker, M. B. Hastings, and M. Troyer, *Phys. Rev. A* **92**, 042303 (2015).
 - [21] P. J. J. O’Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, et al., *Phys. Rev. X* **6**, 031007 (2016).
 - [22] R. Santagati, J. Wang, A. Gentile, S. Paesani, N. Wiebe, J. McClean, S. Short, P. Shadbolt, D. Bonneau, J. Silverstone, et al., *arXiv preprint arXiv:1611.03511* (2016).
 - [23] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, *Physical Review Letters* **87**, 167902 (2001).
 - [24] C. M. Bishop, *Machine Learning* **128** (2006).

- [25] T. Helgaker, J. Olsen, and P. Jorgensen, *Molecular Electronic Structure Theory* (Wiley, 2013).
- [26] J. T. Seeley, M. J. Richard, and P. J. Love, *J. Chem. Phys.* **137**, 224109 (2012).
- [27] A. Tranter, S. Sofia, J. Seeley, M. Kaicher, J. McClean, R. Babush, P. V. Coveney, F. Mintert, F. Wilhelm, and P. J. Love, *International Journal of Quantum Chemistry* **115**, 1431 (2015).
- [28] J. Johansson, P. Nation, and F. Nori, *Comput. Phys. Commun* **183**, 1760 (2012).
- [29] J. Johansson, P. Nation, and F. Nori, *Comput. Phys. Commun* **184**, 1234 (2013).
- [30] D. J. Wales and J. P. Doye, *J. Phys. Chem. A* **101**, 5111 (1997).
- [31] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *SIAM. J. Sci. Comput* **16**, 1190 (1995).
- [32] K. A. Peterson, D. Feller, and D. A. Dixon, *Theor. Chem. Acc.* **131**, 1 (2012).
- [33] J. Snoek, H. Larochelle, and R. P. Adams, in *Advances in neural information processing systems* (2012), pp. 2951–2959.
- [34] A. Ben-Aroya and A. Ta-Shma, *arXiv preprint quant-ph/0702129* (2007).