Toward Neural Network Simulation of Variational Quantum Algorithms

Oliver Knitter[1], **James Stokes**[1], **Shravan Veerapaneni**[1,2]
[1]Department of Mathematics, University of Michigan, Ann Arbor, MI 48109
[2]Flatiron Institute, Simons Foundation, New York, NY 10010
{knitter, stokesjd, shravan}@umich.edu

Abstract

Variational quantum algorithms (VQAs) utilize a hybrid quantum–classical architecture to recast problems of high-dimensional linear algebra as ones of stochastic optimization. Despite the promise of leveraging near- to intermediate-term quantum resources to accelerate this task, the computational advantage of VQAs over wholly classical algorithms has not been firmly established. For instance, while the variational quantum eigensolver (VQE) has been developed to approximate low-lying eigenmodes of high-dimensional sparse linear operators, analogous classical optimization algorithms exist in the variational Monte Carlo (VMC) literature, utilizing neural networks in place of quantum circuits to represent quantum states. In this paper we ask if classical stochastic optimization algorithms can be constructed paralleling other VQAs, focusing on the example of the variational quantum linear solver (VQLS). We find that such a construction can be applied to the VQLS, yielding a paradigm that could theoretically extend to other VQAs of similar form.

1 Introduction

Recent impressive progress in quantum computational technology has led to renewed interest in quantum algorithm research. Much of this excitement stems from the development of quantum algorithms exhibiting exponential speedups over their classical counterparts, including algorithms for large-scale linear algebra [7]. Furthermore, plausible complexity-theoretic assumptions strongly suggest [12] that quantum computers are capable of preparing quantum states whose output probability distributions are hard to sample classically. Despite this vast potential, quantum algorithm design suffers from a few caveats.

Firstly, exponential acceleration requires *fault-tolerant* quantum computation, which necessitates access to a prohibitively large number of physical qubits. Even under the assumption of polylogarithmic overhead in the asymptotic limit of problem size [10], the necessary resources for solving linear systems of practical utility vastly exceed the likely near-term capabilities of noisy intermediate-scale quantum (NISQ) devices. Secondly, the conjectured hardness results [12] for classical sampling pertain to probability distributions with no known practical utility, without any guidance toward which quantum states should be targeted for practical problems.

These two obstacles motivate a new research direction called *variational quantum algorithms* (VQAs), wherein one encodes a computational problem as a stochastic optimization problem whose solution is an unknown quantum state; in particular, the probabilistic nature of quantum states dictates that this solution be encoded in the output probabilities of the optimal state, which may be estimated through Monte Carlo sampling. Modeling quantum states classically requires overcoming a curse of dimensionality, since the number of dimensions of a multi-qubit state space grows exponentially with the number of qubits. The potential computational advantage of VQAs stems from their reliance on a Quantum Processing Unit (QPU) to perform state preparation within a hybrid quantum–

NeurIPS 2022 AI for Science Workshop.

classical framework, where the CPU performs gradient-based updates in order to direct the QPU toward the optimal quantum state. Variational algorithms have already been designed with the goal of solving hard combinatorial optimization [5] and for ground state preparation in quantum chemistry [9]; more recently, VQAs have emerged for solving fundamental linear algebra problems in high dimension, such as matrix–vector products [18], solutions of linear systems [1] and even singular value decomposition (SVD) [2, 17]. Nonetheless, the existence of a quantum computational advantage—as well as its underlying theoretical justification—has yet to be demonstrated for VQAs.

To better understand this potential for quantum advantage, we utilize the close relationship between VQAs and variational quantum Monte Carlo (VQMC), an area that has shown significant recent progress in expanding its capabilities to problems beyond its traditional purview [3]. Similarly to VQAs, the VQMC overcomes the curse of dimensionality by alternating between gradient-based optimization and Monte Carlo sampling from a parametrized quantum state. However, unlike VQAs—whose computational advantage hinges on the conjectured difficulty of sampling from a parametrized quantum circuit—the VQMC gains its power by modeling the quantum state using multi-layer parametrized neural networks [3]. The exploitation of flexible neural networks as many-body trial wavefunctions has made it possible to leverage the enormous success of machine learning (ML) in solving a variety of quantum many-body eigenvalue problems [3]. The close parallels between ideas developed in the very different fields of VQAs, VQMC, and ML has yielded many opportunities for technology transfer between the three, including the discovery of a VQMC-inspired natural gradient optimization algorithm for VQAs [14] and an expanding list of VQA-inspired algorithms for combinatorial optimization [6, 19, 8] and partial differential equations [22].

While VQMC has traditionally focused on ground state preparation and solving the initial value problem for the time-dependent Schrödinger equation, there is no fundamental obstruction to tackling other linear algebra problems of practical concern, and we strive to bridge this gap by pursuing VQMC approaches to linear algebra in exponentially high dimensions. Choosing to focus on VQMC isolates the question of quantum advantage from the inherent advantage of Monte Carlo for overcoming the curse of dimensionality. Using a recently discovered Rayleigh quotient reformulation, we introduce a VQMC-based solver for the linear system $Ax = b$, where A is a large row-sparse matrix, which we call the Variational Neural Linear Solver (VNLS). The VNLS provides a classical benchmark for assessing the quantum computational advantage of the Variational Quantum Linear Solver (VQLS) [1], which is a proposed VQA for solving sparse high-dimensional linear systems. The VNLS hopefully represents the first example of a new paradigm for efficiently solving a variety of large-scale linear algebra problems. On a longer timeframe, the insights gained by studying VQMC realizations of NISQ algorithms on classical hardware will provide the basis for further quantum algorithm technology transfer as quantum hardware matures.

The paper is organized as follows. We compare and contrast in section 2 the VQA-based and VQMC-based approaches for identifying the ground state of a quantum Hamiltonian. Section 3 describes the Variational Quantum Linear Solver (VQLS) and how it encodes the solution of a linear system into the ground state of some Hamiltonian. Section 4 introduces the Variational Neural Linear Solver (VNLS), a demonstrative example for producing fully-classical machine learning algorithms by applying VQMC techniques to VQA-inspired objectives. In section 5, we apply the VNLS to simple linear systems previously used in demonstration of the VQLS, before providing in section 6 some general suggestions for how this area may be further developed. Many of the finer details in this paper are reserved for the appendix.

2 VQAs and VQMC

We now provide an overview of variational quantum algorithms for Hamiltonian ground state identification, presented within the context of potential classical simulation of these algorithms. We directly contrast VQAs with variational Monte Carlo techniques for accomplishing the same task.

2.1 Variational Quantum Algorithms

A variational quantum algorithm (VQA) is a hybrid quantum-classical algorithm simultaneously executed on a classical CPU and a noisy intermediate-scale quantum processing unit (QPU). VQAs are iterative: at each step, the CPU proposes a set of parameters to the QPU, which uses these values to execute a fixed sequence of parameterized quantum gates operating on a set of n qubits. This

sequence of gates, called an *ansatz*, is kept shallow, meaning the number of gates in the sequence is polynomial with respect to n. The QPU passes information about the prepared state back to the CPU, which then performs some type of gradient-based update to the parameter set.

VQAs can solve a variety of tasks: the one most relevant to this paper is ground state identification for quantum Hamiltonians. Consider the set $\{|\psi_\theta\rangle \in \mathbb{C}^{2^n} : \theta \in \mathbb{R}^d\}$ of all quantum state vectors of an n-qubit system that may be represented by different parameterizations of some shallow ansatz. We immediately observe, since the state vector $|\psi_\theta\rangle$ of our ansatz has dimension 2^n, that the number of gates used to execute our ansatz is polylogarithmic with respect to the dimension of $|\psi_\theta\rangle$. Given a quantum Hamiltonian (Hermitian operator) H on this n-qubit state space, the VQA seeks to find the parameterization θ minimizing the expected value of H in state $|\psi_\theta\rangle$, which is derived from the Rayleigh-Ritz principle applied to a $2^n \times 2^n$ complex Hermitian matrix H,

$$L(\theta) = \frac{\langle\psi_\theta|H|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} \tag{1}$$

The minimal expected value equals the smallest eigenvalue of H, which is achieved if and only if $|\psi_\theta\rangle$ is an associated eigenvector. Even under this specific type of VQA paradigm, one can hypothetically solve a variety of problems; all one has to do to make a problem solvable by a ground state identification VQA is to identify a Hamiltonian H such that the solution of the desired problem is, in some way, encoded in its ground state. The Variational Quantum Linear Solver described in section 3 provides an example of how one may construct such a Hamiltonian, but for the remainder of section 2 we will only consider H as an abstract Hamiltonian.

2.2 Foundations of Classical Simulation of VQAs

The overarching goal of this paper is to explore the utility of "classicalizing" variational quantum algorithms by replacing the QPU with a classical neural network, simulating all the quantum state preparation and expected value estimation tasks via Monte Carlo methods. Van den Nest [16] introduces useful sufficient conditions under which quantum expectation values can be so estimated; in particular, the states and operators entering into quantum expectation values must satisfy conditions called classical tractability and efficiently computable sparsity, respectively. A *classically tractable* (CT) n-qubit state is one such that for any index $x \in [2^n]$ we may directly calculate the corresponding coordinate $\langle x|\psi_\theta\rangle$ with respect to the standard basis, and secondly, we may sample from the probability distribution on indices given by the Born rule,

$$\pi(x) = \frac{|\langle x|\psi_\theta\rangle|^2}{\langle\psi_\theta|\psi_\theta\rangle}. \tag{2}$$

The normalization constant introduces a slight ambiguity of terminology, as quantum states are generally considered normalized, and true classical simulation of a CT state should maintain this property. However, it is certainly possible to model a CT state using an unnormalized vector, in which case one may approximate sampling from the corresponding Born distribution using Monte Carlo techniques. In this case, the Born rule includes this normalization constant, and it is this formulation of the rule that we follow in this paper.

We also require that the Hamiltonian H, which has dimension $2^n \times 2^n$, is *efficiently computable*, meaning it obeys the following sparsity constraint: given any row $x \in [2^n]$ of H, we may calculate the column indices and values of all nonzero entries of row x of H in polynomial time with respect to n. Since H is Hermitian, it is true that this property holds for all rows of H if and only if it holds for all columns as well. It is clear that for H to be efficiently computable, the number of nonzero entries in each row of H must be polynomial in n; it is not, however, necessary that the number of *all* nonzero entries in H is polynomial in n. As we discuss in further detail in section 4.1, the tensor product of local Pauli operators is efficiently computable, and the sum of a small—polynomial with respect to n—number of efficiently computable operators is efficiently computable.

2.3 Variational Quantum Monte Carlo

We now give a description of the VQMC algorithm, which provides the foundation for converting VQAs into fully classical algorithms. A brief description of how to implement this algorithm may be found in A.3. Consider a (possibly unnormalized) parameterized family of vectors $|\psi_\theta\rangle$ residing in a 2^n-dimensional complex vector space. Let $\psi_\theta(x) := \langle x|\psi_\theta\rangle$ denote the components of ψ_θ relative

to the standard orthornormal basis $\{|x\rangle : x \in [2^n]\}$ of $\mathbb{C}^{2^n}$. Once normalized, $|\psi_\theta\rangle$ represents one potential quantum state in the state space of an n-qubit system. The starting point of the VQMC algorithm is the observation that the Rayleigh quotient for a Hermitian operator $H \in \mathbb{C}^{2^n \times 2^n}$ acting on this state space can be expressed in the following probabilistic form,

$$\frac{\langle\psi_\theta| H |\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} = \mathop{\mathbb{E}}_{x\sim\pi_\theta}\left[l_\theta(x)\right] \qquad l_\theta(x) := \frac{(H\psi_\theta)(x)}{\psi_\theta(x)} \ , \tag{3}$$

where the function $l_\theta(x)$ is referred to—for historical reasons—as the *local energy*, and where the distribution π_θ follows (2). Equation (3) follows from simple manipulations given in A.1. The variance of the local $l_\theta(x)$ obeys the identity

$$\mathop{\text{var}}_{x\sim\pi_\theta}\left(l_\theta(x)\right) := \mathop{\mathbb{E}}_{x\sim\pi_\theta}\left[\left|l_\theta(x) - L(\theta)\right|^2\right] = \frac{\langle\psi_\theta| H^2 |\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} - \left[\frac{\langle\psi_\theta| H |\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle}\right]^2 \ . \tag{4}$$

A remarkable consequence of (4) is that the variance of $l_\theta(x)$ approaches zero if ψ_θ approaches any eigenvector of H. This has the practical implication that convergence of the algorithm can be assessed by gathering statistics for the local energy.

The local energy $l_\theta(x)$ can be efficiently computed provided that the non-zero entries of the vector $H\psi_\theta$ can be extracted in a time which is polynomial in n, which is true if $|\psi_\theta\rangle$ is a CT state and H is an efficiently computable Hamiltonian, as defined in section 2.2.

The VQMC procedure relies on the approximation of the Rayleigh quotient in (3) by averaging batches sampled according to π_θ using Markov Chain Monte Carlo. More specifically, VQMC optimizes (10) by using a variant of stochastic gradient descent called stochastic reconfiguration [13], which is similar to stochastic natural gradient descent. In natural gradient descent, the parameter set θ is updated according to the rule $\theta \leftarrow \theta - \gamma I^+(\theta)\nabla L(\theta)$, where γ is the learning rate, $\nabla L(\theta)$ is the loss function gradient, and $I^+(\theta)$ represents the Moore–Penrose pseudoinverse of the Fisher information matrix at θ, encoding the geometry of the optimization landscape near θ.

In the case that ψ_θ is a real vector—a slight simplification of the complex case—the gradient of (3) and the Fisher information matrix at the parameterization θ of ψ_θ are given, respectively, by

$$\nabla L(\theta) = \mathop{\mathbb{E}}_{x\sim\pi_\theta}\left[\left(l_\theta(x) - L(\theta)\right)\nabla_\theta \log|\psi_\theta(x)|\right], \ I(\theta) = \mathop{\mathbb{E}}_{x\sim\pi_\theta}\left[\nabla_\theta \log\pi_\theta(x) \otimes \nabla_\theta \log\pi_\theta(x)\right]. \tag{5}$$

In practice, the gradient and Fisher information matrix are estimated stochastically—the gradient is estimated using batches of local energies sampled from π_θ, and the Fisher information is estimated by computing sample covariances between the logarithmic partial derivatives $\frac{\partial}{\partial\theta_i}\left(\log|\psi_\theta(x)|\right), \ x \sim \pi_\theta$ of ψ_θ with respect to π_θ.

3 VQLS

This section is dedicated to the Variational Quantum Linear Solver (VQLS) [1]. We give a high-level description of the VQLS in section 3.1, followed by some common methods and examples for assessing its performance.

3.1 The Variational Quantum Linear Solver

The VQLS is a VQA designed to target linear systems in exponentially high dimensions of the form

$$A|x\rangle \propto |b\rangle \tag{6}$$

where A denotes an invertible $2^n \times 2^n$ complex matrix and $|b\rangle \in \mathbb{C}^{2^n}$ is a nonzero vector. The output of the algorithm is an approximation of the quantum state equal to the normalization of the solution vector $A^{-1}|b\rangle$. For simplicity of exposition we assume that $A = A^\dagger := (\overline{A})^{\mathrm{t}}$ is a Hermitian matrix representing a quantum observable of the system. The original formulation of the VQLS merely presumes that A is invertible, for which a more general formulation of the solver is given in [1]. Furthermore it is demonstrated in [7] that one may always embed a non-Hermitian matrix on an n-qubit space within a Hermitian matrix using one additional ancilla qubit; this formulation is given in section A.2 of the appendix. Given a linear system of the form (6), consider the projection $P_b^\perp$ onto the subspace orthogonal to the vector $|b\rangle$,

$$P_b^\perp := \mathbb{1} - P_b \ , \qquad P_b := \frac{|b\rangle\langle b|}{\langle b|b\rangle}, \qquad H = A^\dagger P_b^\perp A, \tag{7}$$

with H being the resulting Hamiltonian for which the VQLS seeks to minimize (1). This minimization occurs if and only if $|\psi_\theta\rangle$ is an eigenvector associated with the smallest eigenvalue of H, which in this case is 0. In particular, as A is assumed to be invertible, the eigenspace associated with 0 is one-dimensional. Therefore for each θ, equation (1) gives a nonnegative value, equalling zero if and only if $|\psi_\theta\rangle \propto A^{-1}|b\rangle$; section 3.2 further shows exactly how the loss value bounds the accuracy of the VQLS.

3.2 Error Bounding for the VQLS

Modulo a few scaling factors, the square root of the VQLS loss $L(\theta)$ gives an upper bound on the trace distance between the current model state $|\psi_\theta\rangle$ and the true solution state $\left|A^{-1}b\right\rangle$; more specifically, we find (after correcting a missing factor of $\|A\|$ in [1]) that

$$\mathrm{dist}_{\mathrm{Tr}}(\psi_\theta, A^{-1}b) = \frac{1}{2}\,\mathrm{Tr}\left(\sqrt{\left(|\psi_\theta\rangle\langle\psi_\theta| - |A^{-1}b\rangle\langle A^{-1}b|\right)^2}\right) \leq \frac{\kappa\sqrt{L(\theta)}}{||A||}, \tag{8}$$

where κ is the condition number of A and $||A||$ is its operator norm. The trace distance formulation given here only holds for pure quantum states, which suffices for our purposes since our proposed VQLS-inspired solver (the VNLS) uses pure states. Another useful metric for comparing two quantum states is their *fidelity*, which for two pure states equals the square of the cosine of the angle between their corresponding state vectors; in particular, identical quantum states have a fidelity of 1, and orthogonal quantum states have fidelity of 0. We discuss the trace distance bound, alongside how it implies an equivalent bound on the fidelities between these two quantum states, in appendix A.4.

3.3 The Ising-Inspired VQLS Problem

As a benchmark for assessing the performance of the VQLS, the authors of [1] introduced an example problem which they term the Ising-inspired VQLS problem. In particular, given an n qubit system and a condition number κ they define

$$A = \frac{1}{\zeta}\left(\sum_{j=1}^{n} X_j + 0.1\sum_{j=1}^{n-1} Z_j Z_{j+1} + \eta I\right), \tag{9}$$

where X_j and Z_j denote the Pauli-X and Pauli-Z operators applied locally to qubit j, and ζ and η are scaling factors specifically chosen such that A has largest eigenvalue 1 and smallest eigenvalue $\frac{1}{\kappa}$. The vector $|b\rangle$ is taken to be the equal superposition state on n qubits given by $|b\rangle = |+\rangle^{\otimes n}$ where $|+\rangle = (1/\sqrt{2})(|0\rangle + |1\rangle)$. One noteworthy property of this problem is that the target $|b\rangle$ and solution $\left|A^{-1}b\right\rangle$ states become indistinguishable as the problem size increases. We give a further description and analysis of this phenomenon in appendix A.5 and A.6.

4 VNLS

In this section we describe an instantiation of VQMC that we term the variational neural linear solver (VNLS). We focus on a linear system solver for concreteness, but the general strategy theoretically generalizes to a wide range of linear algebra problem, such as the SVD. The VNLS targets linear systems $A|x\rangle \propto |b\rangle$ of the same form as in section 3.1. Analogous to the VQLS, we consider the space $\{|\psi_\theta\rangle \in \mathbb{C}^{2^n} : \theta \in \mathbb{R}^d\}$ of parameterized vectors representing some collection of n-qubit CT states. Given a $2^n \times 2^n$ Hermitian matrix A and nonzero vector $|b\rangle$, we consider the VQLS objective

$$L(\theta) := \frac{\left\langle\psi_\theta\left|AP_b^\perp A\right|\psi_\theta\right\rangle}{\langle\psi_\theta|\psi_\theta\rangle}\ . \tag{10}$$

Just as is the case for VQLS, we have that $L(\theta) \geq 0$, with minimum value equal to 0. As an alternative to preparing and storing $|\psi_\theta\rangle$ on an actual quantum device, we will take the approach of recasting the optimization problem into a form amenable to the variational quantum Monte Carlo, akin to the description given in section A.3.

4.1 Applying VQMC to the linear systems problem

Here we give a formulation of the VNLS, alongside a description of how it may be efficiently implemented. The key observation is that VNLS is an instantiation of the VQMC using (7) as the

input Hermitian Hamiltonian. As shown in A.7, we may apply the VQMC objective to this choice of H and obtain a stochastic objective analogous to (3), with a new local energy $l_\theta(x)$ defined by

$$l_\theta(x) := \frac{1}{\psi_\theta(x)} \left(\left(A^2\psi_\theta\right)(x) - \left(Ab\right)(x) \mathop{\mathbb{E}}_{x' \sim \beta} \left[\frac{\left(A\psi_\theta\right)(x')}{b(x')} \right] \right) . \tag{11}$$

The expectation within (11) is taken with respect to the distribution $\beta(x) = |b(x)|^2 / \langle b|b\rangle$. Thus we may approximate both the objective (10) and its gradient by estimating the expectation values over π_θ and β using batches of data obtained using MCMC or other sampling methods. Algorithm 2 in A.7 depicts this exact procedure and how it differs from A.3, traditional VQMC. For simplicity, we use the same batch size when sampling from π_θ and β, but doing so is not necessary in general.

The variational formulation given above places no additional theoretical restrictions on the structure of A and $|b\rangle$, but some practical considerations have to be made in accordance with the more general Hamiltonian restrictions required for VQMC. The computational complexities of calculating $\left(A^2\psi_\theta\right)(x)$, $\left(Ab\right)(x)$, and $\left(A\psi_\theta\right)(x)$ grow with the number of nonzero entries in row x of A, which can be as large as 2^n. Similarly, the computational complexity of storing and sampling from the distribution β grows with the number of nonzero entries of $|b\rangle$. For these reasons, we must require that A is efficiently computable and that $|b\rangle$ is a CT state.

We ensure A is efficiently computable in the following way. We observe that A, as a $2^n \times 2^n$ Hermitian matrix, may be expressed as a real linear combination of products of local Pauli operators X, Y, and Z. Our matrix A will be efficiently computable so long as the number of terms in this sum is kept sufficently small. A description of what this decomposition entails is given in A.9. There are many different approaches for ensuring that $|b\rangle$ is CT, but for the purposes of this paper we simply store the nonzero elements of $|b\rangle$ directly and require that $|b\rangle$ either be sufficiently sparse or restrict our testing to problems of sufficiently small dimension.

An important detail to keep in mind when training the VNLS using the natural gradient is that scaling either the matrix A or vector b by any nonzero constant does not alter the problem in any way that the VNLS can meaningfully recognize, and does not provide any meaningful advantage in performance or accuracy. Detailed explanations of these results are given in section A.8 of the appendix.

5 Experimental Results

As a benchmark test, we apply the VNLS to the Ising-inspired problem discussed in 3.3. We fix $\kappa = 10$ and we take $|b\rangle$ to have all its entries equal to 1; while VQLS uses the normalized form of $|b\rangle$, scaling $|b\rangle$ does not influence the performance of VNLS in any significant way, as discussed in 4.1.

5.1 Real RBMs

Initializing the weights of a real RBM by sampling from a zero-mean Gaussian distribution with sufficiently small variance will, with high probability, place the corresponding state $|\psi_\theta\rangle$ in a small neighborhood of $|b\rangle$ with respect to the trace distance from 3.2. As discussed in section 3.3, since $|b\rangle$ lies in a neighborhood of $A^{-1}|b\rangle$, we observe the Ising-inspired problem is trivially easy for a real RBM to learn, as shown in the left-most image in figure 1; this figure depicts, for qubit numbers 11 through 14, the fidelity between $|\psi_\theta\rangle$ and $A^{-1}|b\rangle$ over training. In all observed cases, the RBM initializes with very high fidelity, and only improves from there. These results are comparable with those obtained on larger problem sizes with analogous architecture.

These tests were done for $\kappa = 10$, where the RBM was trained with SR using a learning rate of 0.005, over 1000 epochs, with a batch size of 1024 VQMC samples. Local energy samples were drawn from 8 Monte Carlo chains. While directly calculating the fidelity between ψ_θ and $A^{-1}|b\rangle$ to assess performance is not viable at scale due to memory concerns, plotting this fidelity over the training period for small problems proves a better way for visually evaluating the model's performance than directly observing the stochastic loss plots. As the RBM initializes at a point with a small loss value, the noise in the stochastic loss makes it difficult to visually discern that the loss is decreasing over time from the loss graph alone. Though it does not harm the underlying point, we should note that the curve for the 14 qubit case is exhibiting roundoff errors, as it is impossible for the fidelity of two quantum states to be greater than 1.

Figure 1: *Fidelities over training for a few $\kappa = 10$ Ising-inspired VQLS problems. Left: fidelities for training with a real RBM, which initializes closely to the true solution in all tested cases. Center: fidelities for training with a complex RBM. Right: evaluation loss curves for the same complex RBM tests shown in the center plot. For nontrivial $(n \geq 10)$ problem sizes, the complex RBM loss plots inversely correlate with the corresponding increases in fidelity, as expected.*

5.2 Complex RBMs

In addition to the fact that a quirk of the Ising-inspired VQLS problem allows the real RBM to essentially initialize itself at the true solution state, real RBMs are likely not expressive enough to be of much use for the VNLS in practice, as they only learn quantum state vectors with entirely positive entries; there are certainly many linear systems of practical interest where the entries of the solution vector do not share the same sign. Furthermore, since quantum state vectors have complex entries in general, it is natural to explore the complex RBM (cRBM) for use in the VNLS.

Testing of the cRBM VNLS was performed using the $\kappa = 10$ Ising-inspired problem with the same hyperparemeter configuration used in 5.1, yielding interesting qualitative results that would likely be more indicative of the VNLS's general behavior, in comparison with the somewhat trivial example of applying the real RBM (rRBM) to the Ising-inspired problem.

Unlike for rRBMs, the cRBM generally does not initialize in a state close to the solution. However, we may observe qualitatively that the variance between sampled losses of nearby epochs decreases over the course of training, as expected. The second observation is that the cRBM exhibits more heterogeneous learning behavior at small problem sizes $(n \leq 10)$, occasionally struggling to learn the solution to the same standard of accuracy. We do not consider this a significant issue for two reasons: lowering the learning rate and training for more epochs appears to mitigate this issue somewhat, and moreover, just like the VQLS, the VNLS is not intended to be used for problem sizes suitable for standard numerical matrix inversion methods; we are primarily interested in its behavior at scale.

At larger problem sizes, the VNLS exhibits more uniform behavior, as demonstrated by the center and right-most images in figure 1. For the right-most image, depicting loss plots, we have smoothed the loss curves using a Savitzky–Golay filter, which better reveals the general trend. The translucent, noisy curves represent the true sampled loss plots; here, we can also observe the efficacy of the algorithm from the fact that the variances of nearby sample loss values decrease during training, which is a result of equation 4. Such behavior indicates that, relative to problem size, the optimization landscape of the Ising-inspired problem becomes easier for the VNLS to navigate. One likely explanation is that the deliberate fixing of the condition number across all problem sizes—which does not generally occur for matrices derived from physical applications—contributes to this phenomenon.

Figure 2 demonstrates the effect that altering the batch size or learning rate has on the performance of the VNLS. Batch size was varied between 512, 1024, and 2048 samples per epoch. In order to make the loss plots more comparable qualitatively, epoch numbers were scaled so that the total number of VQMC samples drawn remains fixed. We observe that while increasing the batch size does improve both accuracy and rate of convergence, this improvement does not scale linearly with the change in batch size. For instance, doubling the batch size from 512 to 1024 allow us to achieve comparable accuracy in half the time, though the VNLS requires a larger fraction of the training time to achieve it. in contrast, doubling the batch size from 1024 to 2048 does not halve the required training time in quite the same way.

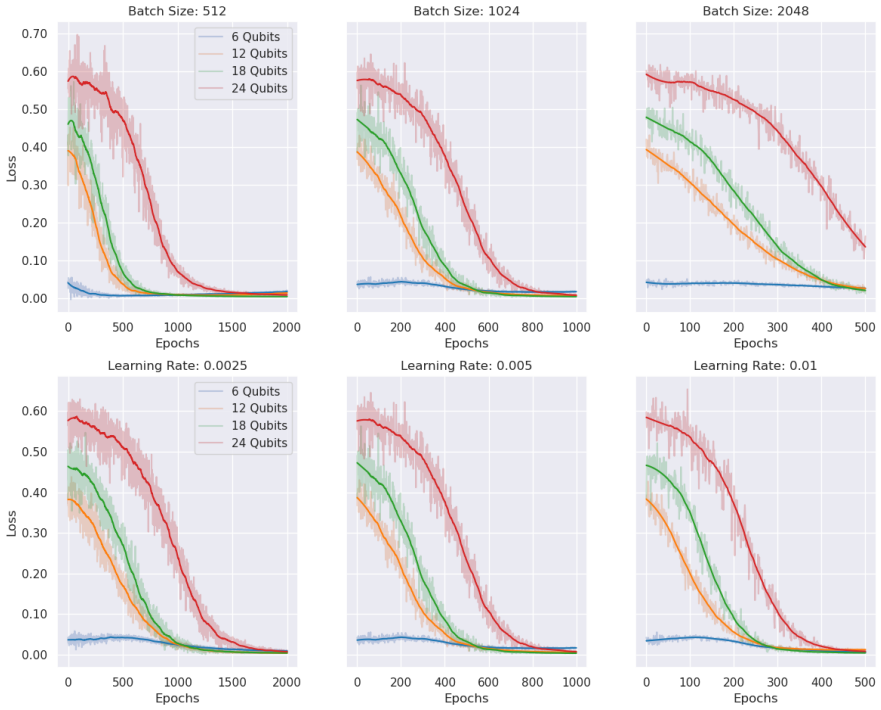

Figure 2: *Top row: effect of changing batch size on cRBM VNLS. Bottom row: effect of changing the learning rate. The epoch number is proportionately scaled so the graphs are qualitatively comparable.*

For larger problem sizes, solely altering the learning rate within reasonable tolerances does not appear to affect performance of the cRBM VNLS in any qualitatively significant way beyond the expected commensurate change in the rate of convergence, as demonstrated by figure 2. Here the number of training epochs was adjusted inversely in accordance with the change in learning rate. Across all larger ($n > 10$) problem sizes tested, these learning rates were able to capture the same accuracy and general learning trend. It should be noted that for the smallest problem sizes, decreasing the learning rate did help to both improve accuracy and smooth out both the loss curve. These observations corroborate the idea that larger Ising-inspired problems are easier for the VNLS to solve.

6 Future Work

Through the application of the VNLS to solving linear systems, we have demonstrated a new paradigm for adapting VQAs to a wholly classical architecture by simulating the quantum circuit component with a neural network. There are a few natural points of expansion for this general area. Firstly, the choice of using an RBM for the neural network component is not vital to the algorithm itself, and there are no theoretical or empirical guarantees that RBMs are best suited for this specific task: one could instead leverage the recent success of autoregressive neural network quantum states [11] or alternatively flow-based modeling for infinite-dimensional linear systems. Recent work in this direction indicates that it presents a scalable, practical alternative to RBMs [20]. Another possible point of expansion would be to develop VQMC-based methods for performing other linear-algebraic operations in high dimensions, such as SVD or QR decomposition, by constructing Monte Carlo estimators based on recent developments in the VQA literature [2]. In addition to providing an ongoing benchmark for assessing quantum computational advantage of VQAs, continued research on VQA-inspired classical algorithms might provide valuable insight into the structure of quantum states for which VQAs might yield quantum advantage.

Acknowledgments and Disclosure of Funding

Authors acknowledge support from NSF under grant DMS-2038030.

References

[1] Carlos Bravo-Prieto, Ryan LaRose, Marco Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J Coles. Variational quantum linear solver. *arXiv preprint arXiv:1909.05820*, 2019.

[2] Carlos Bravo-Prieto, Diego García-Martín, and José I Latorre. Quantum singular value decomposer. *Physical Review A*, 101(6):062310, 2020.

[3] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[4] Kenny Choo, Antonio Mezzacapo, and Giuseppe Carleo. Fermionic neural-network states for ab-initio electronic structure. *Nature Communications*, 11(1), May 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-15724-9. URL http://dx.doi.org/10.1038/s41467-020-15724-9.

[5] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[6] Joseph Gomes, Keri A McKiernan, Peter Eastman, and Vijay S Pande. Classical quantum optimization with neural network quantum states. *arXiv preprint arXiv:1910.10675*, 2019.

[7] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[8] Mohamed Hibat-Allah, Estelle M Inack, Roeland Wiersema, Roger G Melko, and Juan Carrasquilla. Variational neural annealing. *arXiv preprint arXiv:2101.10154*, 2021.

[9] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

[10] Robert Raussendorf and Jim Harrington. Fault-tolerant quantum computation with high threshold in two dimensions. *Physical review letters*, 98(19):190504, 2007.

[11] Or Sharir, Yoav Levine, Noam Wies, Giuseppe Carleo, and Amnon Shashua. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Physical review letters*, 124(2):020503, 2020.

[12] Dan Shepherd and Michael J Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2105): 1413–1439, 2009.

[13] Sandro Sorella. Green function Monte Carlo with stochastic reconfiguration. *Physical Review Letters*, 80(20), 1998.

[14] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020.

[15] Yi ğit Subaş ı, Rolando D. Somma, and Davide Orsucci. Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Phys. Rev. Lett.*, 122:060504, Feb 2019. doi: 10.1103/PhysRevLett.122.060504. URL https://link.aps.org/doi/10.1103/PhysRevLett.122.060504.

[16] Maarten Van den Nest. Simulating quantum computers with probabilistic methods. *arXiv preprint arXiv:0911.1624*, 2009.

[17] Xin Wang, Zhixin Song, and Youle Wang. Variational quantum singular value decomposition. *arXiv preprint arXiv:2006.02336*, 2020.

[18] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational algorithms for linear algebra. *arXiv preprint arXiv:1909.03898*, 2019.

[19] Tianchen Zhao, Giuseppe Carleo, James Stokes, and Shravan Veerapaneni. Natural evolution strategies and variational Monte Carlo. *Machine Learning: Science and Technology*, 2(2): 02LT01, 2020.

[20] Tianchen Zhao, Saibal De, Brian Chen, James Stokes, and Shravan Veerapaneni. Overcoming barriers to scalability in variational quantum monte carlo. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–13, 2021.

[21] Tianchen Zhao, James Stokes, and Shravan Veerapaneni. Scalable neural quantum states architecture for quantum chemistry, 2022. URL https://arxiv.org/abs/2208.05637.

[22] Tianchen Zhao, Chuhao Sun, Asaf Cohen, James Stokes, and Shravan Veerapaneni. Quantum-inspired variational algorithms for partial differential equations: Application to financial derivative pricing. *arXiv preprint arXiv:2207.10838*, 2022.

A Appendix

A.1 Derivation of the Local Energy

Expanding ψ_θ in the standard orthonormal basis, using anti-linearity in the first argument gives

$$\frac{\langle\psi_\theta|\, H\, |\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} = \sum_{x\in[2^n]} \frac{\overline{\psi_\theta(x)}}{\langle\psi_\theta|\psi_\theta\rangle}(H\psi_\theta)(x) = \sum_{x\in[2^n]}{}' \frac{\overline{\psi_\theta(x)}}{\langle\psi_\theta|\psi_\theta\rangle}(H\psi_\theta)(x) \tag{12}$$

where prime in the second summation indicates restriction to the nonzero entries of ψ_θ. Dividing and multiplying the summand by $\psi_\theta(x)$, one obtains the desired result:

$$\frac{\langle\psi_\theta|\, H\, |\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} = \sum_{x\in[2^n]}{}' \frac{|\psi_\theta(x)|^2}{\langle\psi_\theta|\psi_\theta\rangle}\frac{(H\psi_\theta)(x)}{\psi_\theta(x)} = \mathop{\mathbb{E}}_{x\sim\pi_\theta}\left[l_\theta(x)\right] \quad . \tag{13}$$

A.2 Embeddings of Non-Hermitian Matrices

As demonstrated in [7], when presented with a non-Hermitian $2^n \times 2^n$ matrix A, corresponding with an n-qubit state space, one can always embed A into a Hermitian matrix at the cost of one additional qubit, yielding the following modified linear system,

$$\begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix} |x\rangle = \begin{bmatrix} |b\rangle \\ 0 \end{bmatrix}, \qquad \text{whose solution is given by} \qquad \begin{bmatrix} 0 \\ A^{-1}\,|b\rangle \end{bmatrix}. \tag{14}$$

A.3 Implementing VQMC

Algorithm 1: Variational Quantum Monte Carlo using Neural Networks

Input: Hamiltonian H, quantum state neural network ψ_θ, batch size k, learning rate γ
Initialize θ
while not done **do**
  Obtain k samples $x_1, \ldots, x_k$ from π_θ
  for $i = 1, \ldots, k$ **do**
    Obtain $(H\psi_\theta)(x)$ by multiplying the nonzero entries of row x of H with the corresponding entries of ψ_θ, and summing the results
    Compute local energy $l_\theta(x_i)$
    Compute logarithmic gradients $\nabla_\theta \log|\psi_\theta(x)|$ using automatic differentiation
  end for
  Estimate objective using sample mean $\hat{L}(\theta) = \frac{1}{k}\sum_{i=1}^{k} l_\theta(x_i)$
  Estimate gradient and Fisher information using sample mean $\nabla\hat{L}(\theta)$ and covariance matrix $\hat{I}(\theta)$
  Perform parameter update $\theta \leftarrow$ OPTIMIZER $\left(\gamma, \theta, \nabla\hat{L}, \hat{I}\right)$
end while

Algorithm 1 gives a succinct description of how VQMC may be implemented in practice. One may make a direct comparison between the high-level structures of VQMC and VQAs for Hamiltonian

ground state identification. The essential difference between the two that in VQMC we assume $|\psi_\theta\rangle$ is a neural-network quantum state in the vein of what is described in section 2.2. For example, given an input $x \in [2^n]$ one can model $\psi_\theta(x)$ by the output of a restricted Boltzmann machine (RBM) with n visible nodes; inputting the binary encoding of x into the RBM returns component $\psi_\theta(x)$.

It should be noted here that VQMC—and, by extension, VNLS—is network-agnostic. Nothing in what follows will depend upon the use of RBMs, and there is good reason to consider more sophisticated architectures including autoregressive models, which are popular in natural language processing.

A.4 VQLS Error Bound Discussion

Here we describe how the exact value of the objective function (7) informs us of the accuracy of the VQLS, even when the loss value is nonzero. The key idea behind this correspondence is given in [1], though there is a slight modification that must be made to the result, which we note.

The primary method for assessing and bounding the error of the VQLS is by using the trace distance between two quantum states, which is given for two pure states $|\phi\rangle$ and $|\psi\rangle$ by

$$\mathrm{dist}_{\mathrm{Tr}}(\phi, \psi) = \frac{1}{2} \mathrm{Tr}\left(\sqrt{(|\phi\rangle\langle\phi| - |\psi\rangle\langle\psi|)^2}\right). \tag{15}$$

A more general definition of trace distance is unnecessary for our purposes since our own VQLS-inspired classical solver, the VNLS, does not model mixed quantum states, only pure ones. Under the assumption that $||A|| \leq 1$, it is stated in [1] that

$$\mathrm{dist}_{\mathrm{Tr}}(\psi_\theta, A^{-1}b) \leq \kappa\sqrt{L(\theta)}, \tag{16}$$

where κ is the condition number of A. As discussed in section 4.1, scaling A by a positive constant $c < 1$ can make the right-hand side of (16) arbitrary small without actually changing the trace distance between the learned and true states, meaning this bound does not hold in its current form.

The discrepancy in the proof of this bound comes from an intermediate result taken from [15], which states that the smallest eigenvalue of the matrix A^2, which equals $\frac{||A||^2}{\kappa^2}$, is bounded below by

$$\frac{||A||^2}{\kappa^2} \geq \frac{1}{\kappa^2} \tag{17}$$

when $||A|| \leq 1$. In this case, $\frac{1}{\kappa^2}$ is actually an upper bound, not a lower one. Adapting around this small error, we may still follow the rest of the argument given in [1] to obtain a new error bound,

$$\mathrm{dist}_{\mathrm{Tr}}(\psi_\theta, A^{-1}b) \leq \frac{\kappa\sqrt{L(\theta)}}{||A||}, \tag{18}$$

which agrees with the scaling argument given above. This new bound does not require any restriction on $||A||$, and is equivalent to saying that the trace distance between $|\psi_\theta\rangle$ and $A^{-1}|b\rangle$ is bounded above by the product of $\sqrt{L(\theta)}$ with the smallest singular value of A.

Alternately, since the VQLS ansatz $|\psi_\theta\rangle$ and the true solution $A^{-1}|b\rangle$ correspond with a pair of pure quantum states when normalized, an equivalent way of measuring the solver's error would be through using the fidelity between these two states, given here for two general states $|\psi\rangle$ and $|\phi\rangle$:

$$\mathrm{Fid}(\phi, \psi) = \frac{|\langle\phi|\psi\rangle|^2}{\langle\phi|\phi\rangle\,\langle\psi|\psi\rangle}. \tag{19}$$

The fidelity equals 1 if and only if one vector is a scalar multiple of the other, and 0 if the two vectors are orthogonal.

The two measures are essentially equivalent for pure states; more specifically, we have that

$$\mathrm{dist}_{\mathrm{Tr}}(\psi_\theta, A^{-1}b) = \sqrt{1 - \mathrm{Fid}(\psi_\theta, A^{-1}b)}, \tag{20}$$

from which it follows that any upper bound on the trace distance corresponds uniquely with a lower bound on the fidelity; thus the bound given in (18) could be used in either case.

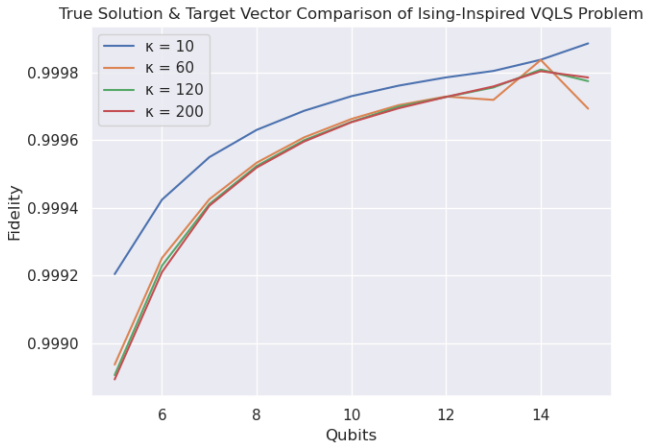

Figure 3: *Fidelity, with respect to problem size between target* $|b\rangle$ *and solution* $A^{-1}|b\rangle$ *vectors for the Ising-inspired VQLS problem.*

A.5 Ising-Inspired VQLS Problem Discussion

Here we give a more detailed description and analysis of the Ising-inspired VQLS problem from [1]. Given a qubit number n and a condition number κ, we consider the matrix

$$A = \frac{1}{\zeta}\left(\sum_{j=1}^{n} X_j + 0.1\sum_{j=1}^{n-1} Z_j Z_{j+1} + \eta I\right), \tag{21}$$

where X_j and Z_j refer to the Pauli-X and Pauli-Z operators applied locally to qubit j, and ζ and η are chosen so that A has largest eigenvalue 1 and smallest eigenvalue $\frac{1}{\kappa}$. The vector $|b\rangle$ is taken to be the equal superposition state on n qubits given by

$$|b\rangle = \left(\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\end{bmatrix}^T\right)^{\otimes n} \tag{22}$$

We observe that while A is surely not the identity matrix, it does behave similarly to the identity when applied to $|b\rangle$ for nontrivial problem sizes. We first note that for a given problem size n and condition number κ, we may take coefficients

$$\eta = n\left(\frac{\kappa+1}{\kappa-1}\right) \text{ and } \zeta = n + \eta = \frac{2n\kappa}{\kappa-1}. \tag{23}$$

This choice is based on a heuristic observation that the largest and smallest eigenvalues of

$$\sum_{j=1}^{n} X_j + 0.1\sum_{j=1}^{n-1} Z_j Z_{j+1} \tag{24}$$

are approximately n and $-n$, in cases for which the qubit number is small enough that these eigenvalues can be easily computed numerically.

The vector $\begin{bmatrix}1 & 1\end{bmatrix}^T$ is an eigenvector of the matrix X with eigenvalue 1, meaning that applying X locally to any one of the n qubits does not change the system state $|b\rangle$. Thus $|b\rangle$ is an eigenvector of $\sum_{j=1}^{n} X_j$ with eigenvalue n. It follows that the matrix A perturbs $|b\rangle$ in the following way:

$$A|b\rangle = |b\rangle + \frac{0.05(\kappa-1)}{n\kappa}\sum_{j=1}^{n-1} Z_j Z_{j+1}|b\rangle. \tag{25}$$

It follows that A, when applied to $|b\rangle$, will perturb each entry of $|b\rangle$ by at most $2^{-\frac{n}{2}}(0.05)$, an explanation for which is given in section A.6 of the appendix.

Just as A perturbs the vector $|b\rangle$, it appears that A^{-1} acts on $|b\rangle$ in a similar manner. Experimentally, we have found that for problem sizes of at least five qubits, the fidelity between $|b\rangle$ and $A^{-1}|b\rangle$ is significantly close to 1. Figure 3 demonstrates that this fidelity grows larger with problem size, and that this trend persists even as the condition number κ increases.

Seeking to explain this phenomenon, we examine the Euclidean distance between $|b\rangle$ and $A^{-1}|b\rangle$. We observe that

$$\left|\left||b\rangle - A^{-1}|b\rangle\right|\right|^2 \leq \frac{0.0025(\kappa-1)^2}{n}\left(\frac{n-1}{n}\right) \to 0 \text{ as } n \to \infty, \tag{26}$$

for which a proof is also given in A.6. While it is possible that this upper bound is not tight, it is nonetheless clear that for a fixed choice of κ, the vectors $|b\rangle$ and $A^{-1}|b\rangle$ become indistinguishable by the VQLS and the VNLS as the qubit number becomes arbitrarily large. Figure 3 demonstrates this phenomenon in practice.

A.6 Correspondence between Target and Solution States for Ising-Inspired VQLS Problem

We recall equation 25, which states for the Ising-inspired VQLS problem that

$$A|b\rangle = |b\rangle + \frac{0.05(\kappa-1)}{n\kappa}\sum_{j=1}^{n-1} Z_j Z_{j+1}|b\rangle\,. \tag{27}$$

We now show how exactly the local Z term in 25 perturbs the entries of $|b\rangle$. The Z-matrix maps $[1 \quad 1]^T$ to $[1 \quad -1]^T$, so for each $i = 0, \ldots, n-1$, applying Z locally to qubits i and $i+1$ converts the state $|b\rangle$ into

$$2^{-\frac{n}{2}}\begin{bmatrix}1\\1\end{bmatrix}_1 \otimes \cdots \otimes \begin{bmatrix}1\\-1\end{bmatrix}_i \otimes \begin{bmatrix}1\\-1\end{bmatrix}_{i+1} \otimes \cdots \otimes \begin{bmatrix}1\\1\end{bmatrix}_n \tag{28}$$

By exploiting the structure of the Kronecker product in exactly the same manner as is done in A.9, we may calculate specific coordinates of $Z_j Z_{j+1}|b\rangle$ as follows: consider entry $x \in \left\{0, 1, \ldots, 2^{n-1}\right\}$ of $Z_j Z_{j+1}|b\rangle$, represented in binary by $(x_1, \ldots, x_n)$. The value of entry x equals $2^{-\frac{n}{2}}$ multiplied by the product, taken over all $k = 1, \ldots, n$ qubits, of entry x_k of either $[1 \quad 1]^T$ or $[1 \quad -1]^T$, depending on whether or not $k \in \{j, j+1\}$.

Thus we may observe that entry x of $Z_j Z_{j+1}|b\rangle$ equals $2^{-\frac{n}{2}}$ if x_j and x_{j+1} agree, and $-2^{-\frac{n}{2}}$ if they do not. Since $\sum_{j=1}^{n-1} Z_j Z_{j+1}$ iterates over every pair of consecutive qubits, it follows that entry x of $\sum_{j=1}^{n-1} Z_j Z_{j+1}|b\rangle$ equals the difference, scaled by $2^{-\frac{n}{2}}$, between the number of consecutive binary digits in $(x_1, \ldots, x_n)$ with the same value, and the number of consecutive binary digits with opposing values. It follows from (25) that applying A to $|b\rangle$ perturbs each entry of $|b\rangle$ by at most $2^{-\frac{n}{2}}(0.05)$.

We now show the effect of A^{-1} on $|b\rangle$. We observe that the square of the Euclidean distance between $A^{-1}|b\rangle$ and $|b\rangle$ may be bounded above as follows:

$$
\begin{aligned}
\left|\left||b\rangle - A^{-1}|b\rangle\right|\right|^2 &= \left|\left|A^{-1}\left(A|b\rangle - |b\rangle\right)\right|\right|^2 \\
&\leq \left|\left|A^{-1}\right|\right|^2 \left|\left|\left(A|b\rangle - |b\rangle\right)\right|\right|^2 \\
&= \kappa^2 \left|\left| \frac{0.05(\kappa-1)}{n\kappa} \sum_{j=1}^{n-1} Z_j Z_{j+1} |b\rangle \right|\right|^2 \\
&= \frac{0.0025(\kappa-1)^2}{n^2} \left|\left| \sum_{j=1}^{n-1} Z_j Z_{j+1} |b\rangle \right|\right|^2 \\
&= \frac{0.0025(\kappa-1)^2}{n^2} \left\langle \sum_{j=1}^{n-1} Z_j Z_{j+1} b \middle| \sum_{k=1}^{n-1} Z_k Z_{k+1} b \right\rangle \\
&= \frac{0.0025(\kappa-1)^2}{n^2} \sum_{j=1}^{n-1}\sum_{k=1}^{n-1} \langle Z_j Z_{j+1} b | Z_k Z_{k+1} b\rangle \\
&= \frac{0.0025(\kappa-1)^2}{n^2} \sum_{j=1}^{n-1}\sum_{k=1}^{n-1} \langle b | Z_{j+1} Z_j Z_k Z_{k+1} | b\rangle
\end{aligned} \tag{29}
$$

Equation 29 follows from the line preceding it by the fact that the local Z-operator is Hermitian. To complete the bound, we fix any $j, k \in \{1, \ldots, n-1\}$ and consider the inner product $\langle b | Z_{j+1} Z_j Z_k Z_{k+1} b\rangle$. Since $|b\rangle$ is the tensor product of n copies of the vector $2^{-\frac{1}{2}} \begin{bmatrix} 1 & 1 \end{bmatrix}^T$, and the operator $Z_{j+1} Z_j Z_k Z_{k+1}$ acts locally on $|b\rangle$, it follows that we may decompose the inner product into

$$
\langle b | Z_{j+1} Z_j Z_k Z_{k+1} | b\rangle = 2^{-n} \prod_{\ell=1}^{n} \begin{bmatrix} 1 & 1 \end{bmatrix} O_\ell \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{30}
$$

where for each ℓ the local operator O_ℓ is given by

$$
O_\ell = \begin{cases} I & \text{if } \ell \text{ does not equal any of } j, j+1, k, k+1 \\ Z & \text{if } \ell \text{ equals exactly one of } j, j+1, k, k+1 \\ Z^2 = I & \text{if } \ell \text{ equals both one of } j, j+1 \text{ and one of } k, k+1. \end{cases} \tag{31}
$$

Since the products

$$
\begin{bmatrix} 1 & 1 \end{bmatrix} I \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1 + 1 = 2 \text{ and } \begin{bmatrix} 1 & 1 \end{bmatrix} Z \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1 - 1 = 0, \tag{32}
$$

it follows from (30) that $\langle b | Z_{j+1} Z_j Z_k Z_{k+1} b\rangle$ equals 1 if $j = k$ and 0 otherwise. Therefore we have from (29) that

$$
\begin{aligned}
\left|\left||b\rangle - A^{-1}|b\rangle\right|\right|^2 &\leq \frac{0.0025(\kappa-1)^2}{n^2} \sum_{j=1}^{n-1}\sum_{k=1}^{n-1} \langle b | Z_{j+1} Z_j Z_k Z_{k+1} | b\rangle \\
&= \frac{0.0025(\kappa-1)^2}{n^2} \sum_{j=1}^{n-1} 1 \\
&= \frac{0.0025(\kappa-1)^2}{n} \left(\frac{n-1}{n}\right) \to 0 \text{ as } n \to \infty
\end{aligned} \tag{33}
$$

A.7 VNLS Objective Derivation

Evaluating the VQMC objective for this specific choice of H one finds

$$L(\theta) = \frac{\left\langle \psi_\theta \left| A^2 - \frac{A|b\rangle\langle b|A}{\langle b|b\rangle} \right| \psi_\theta \right\rangle}{\langle \psi_\theta | \psi_\theta \rangle} \ , \tag{34}$$

$$= \frac{\langle \psi_\theta | \, A^2 \, | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} - \frac{\langle \psi_\theta | \, \langle b | \, A \, | \psi_\theta \rangle \, A \, | b \rangle}{\langle \psi_\theta | \psi_\theta \rangle \, \langle b | b \rangle} \ , \tag{35}$$

$$= \frac{\langle \psi_\theta | \, A^2 \, | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} - \left(\frac{\langle \psi_\theta | \, A \, | b \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \right) \left(\frac{\langle b | \, A \, | \psi_\theta \rangle}{\langle b | b \rangle} \right) \ . \tag{36}$$

By similar manipulations as in section 2.3, we find that the VNLS objective function takes the probabilistic form

$$L(\theta) = \mathop{\mathbb{E}}_{x \sim \pi_\theta} \left[l_\theta(x) \right] \ , \tag{37}$$

where the local energy is

$$l_\theta(x) := \frac{1}{\psi_\theta(x)} \left(\big(A^2 \psi_\theta\big)(x) - \big(Ab\big)(x) \mathop{\mathbb{E}}_{x' \sim \beta} \left[\frac{\big(A\psi_\theta\big)(x')}{b(x')} \right] \right) \ , \tag{38}$$

where the distribution β is defined by $\beta(x) = |b(x)|^2 \, / \, \langle b|b \rangle$. Algorithm 2 demonstrates the VNLS in full, highlighting its similarities and differences with traditional VQMC.

Algorithm 2: Variational Neural Linear Solver

Input: Matrix A, vector $|b\rangle$, quantum state neural network ψ_θ, batch size k, learning rate γ
Initialize θ
while not done **do**
  Obtain k samples $x_1, \ldots, x_k$ from π_θ and $x'_1, \ldots, x'_k$ from β
  for $i = 1, \ldots, k$ **do**
    Calculate $\frac{(A\psi_\theta)(x'_i)}{b(x'_i)}$
  end for
  Estimate expected value w.r.t. β using $\frac{1}{k} \sum_{i=1}^{k} \frac{(A\psi_\theta)(x'_i)}{b(x'_i)}$
  for $i = 1, \ldots, k$ **do**
    Obtain $(A^2\psi_\theta)(x)$ by multiplying nonzero entries of row x of A^2 with the corresponding entries of $|\psi_\theta\rangle$, summing the results
    Do the same for $(Ab)(x)$
    Construct local energy $\hat{l}_\theta(x_i)$ estimate
    Compute logarithmic gradients $\nabla_\theta \log |\psi_\theta(x)|$ using automatic differentiation
  end for
  Estimate objective using sample mean $\hat{L}(\theta) = \frac{1}{k} \sum_{i=1}^{k} \hat{l}_\theta(x_i)$
  Estimate gradient and Fisher information using sample mean $\nabla \hat{L}(\theta)$ and covariance $\hat{I}(\theta)$
  Perform parameter update $\theta \leftarrow$ OPTIMIZER $\left(\gamma, \theta, \nabla \hat{L}, \hat{I}\right)$
end while

A.8 Equivalency of VNLS Problems Under Scaling

We demonstrate here how the scaling of a VNLS problem, as described in section 4.1, has no meaningful effect on the ability of VNLS to solve it using SR.

Recalling that the goal of VNLS is to find, for a given Hermitian matrix A and vector $|b\rangle$, an unnormalized vector $|x\rangle$ such that $A\,|x\rangle$ is proportional to $|b\rangle$, we observe that if

$$A\,|x\rangle \propto |b\rangle \ , \text{ then } c_1 A\,|x\rangle \propto c_2\,|b\rangle \tag{39}$$

for any pair of nonzero scalars c_1 and c_2. For this reason, the solution set of any VNLS problem is invariant under scaling of both A and $|b\rangle$. In practice, one may consider that scaling either A or $|b\rangle$ by

some positive value might improve the learning process in some cases, but we find that careful scaling of A or $|b\rangle$ yields no unique advantage for VNLS performance under stochastic reconfiguration.

In particular, we find that keeping A, θ, and x fixed, scaling $|b\rangle$ by any positive constant c does not change the VNLS objective $L(\theta)$ or its gradient. Similarly, keeping $|b\rangle$, θ, and x fixed, scaling A by any positive constant c changes the objective $L(\theta)$ and its gradient $\nabla L(\theta)$ to $c^2 L(\theta)$ and $c^2 \nabla L(\theta)$, which we now explain.

For a given VNLS problem $\{A, |b\rangle\}$ and parameterized, unnormalized quantum state $|\psi_\theta\rangle$, we recall from (11) that the local energy of the VNLS Hamiltonian at basis state x is given by

$$l_\theta(x) = \frac{1}{\psi_\theta(x)} \left(\big(A^2\psi_\theta\big)(x) - \big(Ab\big)(x) \mathop{\mathbb{E}}_{x' \sim \beta} \left[\frac{\big(A\psi_\theta\big)(x')}{b(x')} \right] \right), \tag{40}$$

with the VNLS objective being the expected value of $l_\theta(x)$ with respect to the distribution π_θ corresponding with state $|\psi_\theta\rangle$. Keeping A, θ, and x fixed, we observe that replacing $|b\rangle$ with $c\,|b\rangle$ for any positive scalar c preserves the local energy:

$$\frac{1}{\psi_\theta(x)} \left(\big(A^2\psi_\theta\big)(x) - \big(A(cb)\big)(x) \mathop{\mathbb{E}}_{x' \sim \beta} \left[\frac{\big(A\psi_\theta\big)(x')}{cb(x')} \right] \right) \tag{41}$$

$$= \frac{1}{\psi_\theta(x)} \left(\big(A^2\psi_\theta\big)(x) - \frac{c}{c}\big(Ab\big)(x) \mathop{\mathbb{E}}_{x' \sim \beta} \left[\frac{\big(A\psi_\theta\big)(x')}{b(x')} \right] \right) = l_\theta(x). \tag{42}$$

This replacement also has no effect on the distribution β, and consequently scaling the vector b does not affect VNLS training in any way.

On the other hand, scaling A while holding $|b\rangle$, θ, and x fixed does alter the local energy:

$$\frac{1}{\psi_\theta(x)} \left(\big((cA)^2\psi_\theta\big)(x) - \big(cAb\big)(x) \mathop{\mathbb{E}}_{x' \sim \beta} \left[\frac{\big(cA\psi_\theta\big)(x')}{b(x')} \right] \right) \tag{43}$$

$$= \frac{c^2}{\psi_\theta(x)} \left(\big(A^2\psi_\theta\big)(x) - \big(Ab\big)(x) \mathop{\mathbb{E}}_{x' \sim \beta} \left[\frac{\big(A\psi_\theta\big)(x')}{b(x')} \right] \right) = c^2 l_\theta(x). \tag{44}$$

It follows that our objective $L(\theta) = \mathop{\mathbb{E}}_{x' \sim \pi_\theta} [l_\theta(x)]$ then becomes $c^2 L(\theta)$, hence the gradient of our objective is replaced with $c^2 \nabla L(\theta)$.

Under stochastic reconfiguration, solving the VNLS problem $\{cA, |b\rangle\}$ using learning rate γ is then tantamount to solving the original problem $\{A, |b\rangle\}$ using learning rate $c^2\gamma$, with the caveat that the objective function is scaled by c^2 as well. This latter change notwithstanding, scaling A does not affect SR in any way that could not similarly be achieved by changing the learning rate.

A.9 Pauli Basis Decomposition of Hermitian Matrices

From this point onward we will treat the rows and columns of every matrix discussed as being indexed starting from 0. As discussed in section 4.1, we may decompose any $2^n \times 2^n$ Hermitian matrix A into a linear combination

$$A = \sum_i c_i \left(A_{i_1} \otimes \cdots \otimes A_{i_n} \right), \tag{45}$$

where each A_{i_j} is either the identity operator or one of the Pauli operators

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \text{ and } Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{46}$$

Here the notation $\otimes$ refers to the Kronecker product

$$A \otimes B = \begin{bmatrix} A_{00}B & A_{01}B \\ A_{10}B & A_{11}B \end{bmatrix}, \tag{47}$$

which may be applied recursively to produce each summand in A.

This exact decomposition of A is made possible by the fact that the four matrices I, X, Y, and Z form a basis for the real vector space of 2×2 Hermitian matrices. The real vector space of $2^n \times 2^n$ Hermitian matrices comprises the tensor product of n instances of the 2×2 matrix space, for which the set of matrices of the form $A_1 \otimes \cdots \otimes A_n$—where $A_i \in \{I, X, Y, Z\}$ for every i—forms a basis.

In general, the number of terms comprising A is $O\left(4^n\right)$; we shall consider observables for which the number of terms is $O(n^k)$ for some (reasonably small) k. We can impose the same constraint on the number of nonzero entries in b. Any product of local Pauli operators

$$c\left(A_1 \otimes \cdots \otimes A_n\right) \tag{48}$$

is represented by a $2^n \times 2^n$ sparse matrix with exactly one nonzero entry in each row; this fact follows by induction on the number of factors n in $c\left(A_1 \otimes \cdots \otimes A_n\right)$, using the definition of the Kronecker product in (47) alongside the fact that each of the 2×2 Pauli matrices—and the identity matrix—has exactly one nonzero entry in each of its rows.

Given the index x of some row of $c\left(A_1 \otimes \cdots \otimes A_n\right)$, it is possible to find the column index and value of the corresponding nonzero entry in $O(n)$ time. The heart of this idea may be found in appendix B of [4], and in equation 16 of [21]. If A comprises $O(n^k)$ terms in this local Pauli product form, then for each row of A we may find the column indexes and values of all its nonzero entries in $O(n^{k+1})$ time by simply looking at each term individually. Since $(A\psi_\theta)(x)$ equals the inner product of row x of A with ψ_θ, placing this constraint on A ensures that we can always calculate the local energy of A at any row in $O(n^{k+1})$ time.