Artificial Neural Networks Applied as Molecular Wave Function Solvers

Yang Peng-Jian,[*,†] Mahito Sugiyama,[*,‡,⊥] Koji Tsuda,[*,¶,§,#] and Takeshi Yanai[*,‖,†,⊥]

†*Department of Chemistry, Nagoya University, Furocho, Chikusa Ward, Nagoya, Aichi 464-8601, Japan*

‡*National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan*

¶*Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5 Kashiwa-no-ha, Kashiwa, Chiba 277-8561, Japan*

§*RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan*

‖*Institute of Transformative Bio-Molecules (WPI-ITbM), Nagoya University, Furocho, Chikusa Ward, Nagoya, Aichi 464-8601, Japan*

⊥*JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan*

#*Research and Services Division of Materials Data and Integrated System, National Institute for Materials Science, Ibaraki 305-0047, Japan*

E-mail: pj.yang_lester@berkeley.edu; mahito@nii.ac.jp; tsuda@k.u-tokyo.ac.jp; yanait@chem.nagoya-u.ac.jp

Abstract

We use artificial neural networks (ANNs) based on the Boltzmann machine (BM) architectures as an encoder of *ab initio* molecular many-electron wave functions represented with the complete active space configuration interaction (CAS-CI) model. As first introduced by the work of Carleo and Troyer for physical systems, the coefficients of the electronic configurations in the CI expansion are parameterized with the BMs as a function of their occupancies that act as descriptors. This ANN based wave function ansatz is referred to as the neural-network quantum state (NQS). The machine learning is used for training the BMs in terms of finding a variationally optimal form of the ground-state wave function on basis of the energy minimization. It is relevant to reinforcement learning and does not use any reference data nor prior knowledge of the wave function, while the Hamiltonian is given based on a user-specified chemical structure in the first-principles manner. Carleo and Troyer used the restricted Boltzmann machine (RBM), which has hidden units, for the neural network architecture of NQS, while in this study, we further introduce its replacement with the BM that has only visible units but with different orders of connectivity. For this hidden-node free BM, the second- and third-order BMs based on quadratic and cubic energy functions, respectively, were implemented. We denote these second- and third-order BMs as BM2 and BM3, respectively. The pilot implementation of the NQS solver into an exact diagonalization module of the quantum chemistry program was made to assess the capability of variants of the BM based NQS. The test calculations were performed by determining the CAS-CI wave functions of illustrative molecular systems, indocyanine green and dinitrogen dissociation. The simulated energies have been shown to converge to CAS-CI energy in most cases by improving RBM with increasing number of hidden nodes. BM3 systematically yields lower energies than BM2, reproducing the CAS-CI energies of dinitrogen across potential energy curves within an error of 50 μE_h.

1 Introduction

Recent advances in machine learning (ML) and its versatility have led to a wide range of promising applications in chemical science from reaction prediction, drug discovery, and syntheses planning[1–3] to *ab initio* determination of force field parameters, and prediction of adaptive basis sets in large scale DFT calculations.[4,5] In the physics community, the potential applicability of the ML algorithms has been investigated as an alternative approach to tackle the compression problem related to quantum many-body systems.[6,7] Due to its strong ability to compress information and extract features from large data quantity, artificial neural networks (ANNs) have recently become the subject of active research in condensed matter physics and quantum information.[8–10]

In the seminal work by Carleo and Troyer, an ANN model based on the restricted Boltzmann machine (RBM) was introduced as a method of exact diagonalization, in which it serves as a representation of many-electron wave function,[11] hereafter referred to as the neural-network quantum state (NQS). The RBM is a type of generative models capable of reproducing probability distribution over some data of unknown probability distribution. Without prior knowledge of the exact particle distribution, Carleo and Troyer designed a reinforcement learning algorithm for learning optimal network parameters that give the best possible representations of ground states of many-body quantum systems or associated strong correlation in a variational manner given the Hamiltonian. This approach can achieve accurate results for physical systems of both one and two dimensions modeling networks of strongly-interacting spins or fermions. Further investigations have demonstrated an intimate relation between RBM and other 2-dimensional tensor-network or strongly-correlated states; hence, RBM's ability to account for higher dimensional systems, beyond an 1D capacity of the matrix product state (MPS) ansatz.[12–14] The incorpration of deep learning architecture into RBM based NQS was demonstrated to yield a promising improvement.[15,16] Naturally, it should thus be of great interest to investigate the applicability of this ANN ansatz to molecular systems in scope of quantum chemical (QC) research.

Motivated by the promise of Carleo's work, our interest runs into the adaptation of Carleo's ML inspired algorithm to the QC solver to describe static correlation or multireference (MR) electron correlation, analogy of strong correlation of physical systems. The efficient computation of the static correlation is considered to remain challenging; it needs to be calculated in QC applications when dealing with molecular electronic states relevant to multiplet structures, state degeneracies, open d-shells, bond-breaking, catalytic or reactive processes, and photochemical or radical activities of aromatic systems, etc. In general, cumbersome quantum complexity arises in static correlation because its underlying description is associated with a lengthy expansion into configuration interactions (CIs) of Hilbert space with basis of valance-bond structures or Slater determinants (SD). The *ab initio* building of chemically important part of the Hilbert space, results in a well-suited model space for static correlation, accommodating the framework of the complete active space (CAS) scheme.[17] CAS is a reduced space that is schematically derived from the use of user-selected orbitals for what is correlated in molecular electronic systems. The quantum states resulting from exact diagonalization of the given CAS Hamiltonian are referred to as CAS-CI wave functions, and corresponds to a description of the static correlation.

The algorithms to determine the MR wave functions have been extensively investigated and led to proposals of various capable methods that effectively overcome the steeply increasing complexity of the exact diagonalization. Among the proposed are density matrix renormalization group (DMRG),[18–20] its underlying MPS,[21,22] spin-projected matrix product states (SP-MPS),[23] spin tree tensor network states (TTNS),[24–26] complete-graph tensor network states (CGTNS),[27] Projected Entangled Pair States (PEPS),[28] and self-adaptive tensor network states (SATNS),[29] as well as (semi-)stochastic or quantum Monte-Calro (QMC) approaches, such as full configuration interaction quantum monte carlo (FCIQMC)[30] and related multi-state quantum monte carlo (MSQMC).[31] Along this line, a renewed interested has been received in the selected configuration interaction (CI) method.[32–35] We cannot cite all the notable studies on MR theory in QC developments due to limitations of space. In the earlier studies, one of the authors was involved in the research to combine DMRG with the aforementioned CAS framework along with various related QC extensions, including orbital optimization and a postprocess to account for weak or dynamical electron correlation as a critical correction to the static correlation.

The objective of this study is to develop the implementation of the NQS learning and training algorithm to determine the CAS-CI wave functions,[17] and demonstrate its basic capability to reproduce CAS-CI results for molecular QC systems. The value of this investigation lies in shedding light on the physical structure of static correlation from neural network perspectives, which somewhat differ from the MR models established by the aforementioned studies. Carleo et al. has used the RBM as a learning model of NQS. The RBM is a well-understood ANN model in ML; historically, it is known as an origin on which Hinton built the neural network of deep learning by introducing cascades of hidden layers. In addition, the authors just noticed a report of Choo et al. on the implemention of the NQS with RBM for quantum chemistry molecular Hamiltonian in a line similar to our study, while it uses a somewhat different wave function ansatz with the mapping between the fermionic quantum chemistry molecular Hamiltonian and corresponding spin Hamiltonians.[36] Along a similar line, there are several works recently published, which apply the ML technologies to learn the features of quantum chemical wave functions rather than energies.[37–41]

In addition to RBM for the NQS model, we attempted to use higher-order BM (HBM) as an extension of BM to a different dimension. Specifically, we employ the BM (i.e., the second-order BM) and its third-order variant that both have no hidden nodes nor restriction in connectivity. The BM formulated with quadratic or second-order global energy functions in a similar manner to RBM is hereafter referred to as BM2, whereas the third-order BM, designated BM3, is based on cubic energy functions where triples of visible nodes are allowed to interact. A central difference of BM2 and BM3 from RBM is thus that they have only visible nodes. They are based on concave log-likelihood functions $L(\theta)$, whereas RBM's $L(\theta)$ is non-concave due to the presence of its hidden nodes.[42] The use of BM2 and BM3 other than RBM allows us to possibly seek for better convergence to global minima. This comparison is merited as RBM and BM3 are both models capable of extracting higher-order features, and their total errors, measured as the Kullback-Leibler (KL) divergence between true (unknown) distribution and model distribution, are similar in magnitude for sufficiently large data sets.[43]

The rest of this article is structured as follows. In section 2, we focus on the underlying theory in order to discuss the specific merits and demerits of our ansatz. In 2.1, a brief introduction of generative artificial neural networks, and Boltzmann machines is given. Section 2.2 describes the Markov chain Monte Carlo (MCMC) sampling to evaluate NQS related objects. Section 2.3 explains the neural-network quantum state (NQS) ansatz. In section 3, we detail our formulation and implementation of the algorithm to QC solver for MR calculations. In section 4, benchmark results for molecules, indocyanine green (ICG) and dinitrogen, were analyzed and discussed. Finally, section 5 gives the conclusion.

2 Recapitulation

Since the NQS based on the Boltzmann machines (BM) was recently introduced by Carleo and Troyer[11] in the community of physics, we will give a brief review of its basics for novices, providing notations and mathematical definitions for throughout comprehensibility in the presentation of our quantum chemical variant. We will mainly recapitulate the Boltzmann machines in the context of the machine learning and their adaptation to quantum many-body states developed in the original work of Carleo and Troyer. Besides the restricted BM architecture used by Carleo and Troyer, an attention will be casted on the fully visiable BM with the bipartite and tripartite graphs as the higher-order BM based NQS introduced in this study.

2.1 Boltzmann Machines

Generative artificial neural networks are graph structures that can reproduce a probability distribution given some training data with the distribution $p_{\text{data}}(\mathbf{x})$, such that its ML model $p_{\text{model}}(\mathbf{x})$ is a good approximation to $p_{\text{data}}(\mathbf{x})$. In general, Boltzmann machines are a type of generative ANNs represented with undirected graphs of $G = (V, E)$ with vertex set of

$$V = \{v_1, v_2, ..., v_n, h_1, h_2, ..., h_m\} \tag{1}$$

and edge set of

$$E \subseteq \{(x_i, x_j) \mid x_i, x_j \in V\}. \tag{2}$$

The vertices of the model are composed of binary observed variables $\mathbf{v} \in \{0,1\}^n$ that represent the direct observations on the data set, and binary hidden variables $\mathbf{h} \in \{0,1\}^m$ that are latent feature detectors.[43] Since every vertex is connected to all other vertices in BM, let us generalize the vertex set V as $(x_1, x_2, ..., x_{n+m}) \in \{0,1\}^{(n+m)}$ and the energy function of the joint configuration $\mathbf{x} \equiv (\mathbf{v}, \mathbf{h})$ is given by:

$$E^{\mathrm{BM}}(\mathbf{x}) = -\sum_{i=1}^{n+m} x_i b_i - \sum_{i,j=1}^{n+m} x_i x_j w_{ij} \tag{3}$$

where b_i is the bias of the vertex that quantifies the importance of vertex x_i, and w_{ij} is the weight on the edge that quantifies the importance of connection between x_i and x_j. In the BM, the distribution of $\mathbf{x}$ assigned to every possible pair of visible and hidden vectors is defined using $E^{\mathrm{BM}}(\mathbf{x})$ as proportional to

$$f(\mathbf{x}) = e^{-E^{\mathrm{BM}}(\mathbf{x})}\,. \tag{4}$$

The probability is thus expressed as,

$$p(\mathbf{x}) = \frac{1}{Z}\, f(\mathbf{x}) \tag{5}$$

using the normalization constant Z, also known as the partition function given by:

$$Z = \sum_{\mathbf{x}} f(\mathbf{x})\,, \tag{6}$$

where $\sum_{\mathbf{x}}$ represents $\sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_{n+m}=0,1}$. In the RBM, the visible vertices and latent vertices are organized into two separate layers, forming a bipartite graph.[44] The "visible layer" and the "hidden layer" each do not include intralayer edges. The energy function equivalent to eqn. (3) can be specified for RBM as

$$E^{\mathrm{RBM}}(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{n} v_i b_i - \sum_{j=1}^{m} h_j c_j - \sum_{i=1}^{n} \sum_{j=1}^{m} v_i h_j w_{ij} \tag{7}$$

where c_j is the bias on the hidden unit h_j and a part of $\{b_i\}$ such that $c_j \equiv b_{j+n}$. The joint probability distribution formulated as eqns. (5) and (6) are specified in the similar manner. By marginalizing over all hidden vectors in eqn. (5), the probability assigned to a visible vector is obtained as,

$$p(\mathbf{v}) = \frac{1}{Z} f^{\mathrm{RBM}}(\mathbf{v}) \tag{8}$$

with

$$f^{\mathrm{RBM}}(\mathbf{v}) = \sum_{\mathbf{h}} e^{-E^{\mathrm{RBM}}(\mathbf{v},\mathbf{h})}. \tag{9}$$

The RBMs are fundamental building blocks of deep neural nets (DNNs) and can be expanded with additional hidden layers.[16,45]

From a training perspective, eqn. (8) is used for $p_{\mathrm{model}}(\mathbf{v})$, which is optimized with respect to the BM parameters $\mathbf{b}, \mathbf{w} \in \boldsymbol{\theta}$ so as to well reproduce the true distribution $p_{\mathrm{data}}(\mathbf{v})$. The optimization or learning algorithm is formulated based on the likelihood function. For a given set of training data or big data, $\left\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, ..., \mathbf{v}^{(N_{\mathrm{data}})}\right\}$ with the distribution $p_{\mathrm{data}}(\mathbf{v})$, the likelihood function $L(\boldsymbol{\theta})$ is given by

$$L(\boldsymbol{\theta}) = \prod_{s=1}^{N_{\mathrm{data}}} p(\mathbf{v}^{(s)}; \boldsymbol{\theta})\,, \tag{10}$$

and the log-likelihood $\log L(\boldsymbol{\theta})$ is written as

$$\log L(\boldsymbol{\theta}) = \sum_{s=1}^{N_{\mathrm{data}}} \log f(\mathbf{v}^{(s)}; \boldsymbol{\theta}) - N_{\mathrm{data}} \log Z(\boldsymbol{\theta})\,, \tag{11}$$

where $p(\mathbf{v}; \boldsymbol{\theta})$ represents $p(\mathbf{v})$ parameterized by $\boldsymbol{\theta}$. The goal of the learning is to get the model $p(\mathbf{v}; \boldsymbol{\theta})$ trained or optimized on the data set by finding suited $\boldsymbol{\theta}$. This can be done by maximizing $\log L(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ where the log-likelihood gradients, $\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta})$, satisfies the following condition:

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}) = 0\,. \tag{12}$$

This condition is equivalently minimizing the KL divergence (distance) between the trained model and data distribution. The training is carried out by iteratively updating $\boldsymbol{\theta}$ with

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \alpha_t \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}^{(t)})\,. \tag{13}$$

Considering that the normalization Z is solely responsible for the model, the log-likelihood gradients can be expanded as

$$\frac{1}{N_{\text{data}}} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}) = \underbrace{\frac{1}{N_{\text{data}}} \sum_{s=1}^{N_{\text{data}}} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log f(\mathbf{v}^{(s)})}_{\text{data}} - \underbrace{\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta})}_{\text{model}} \tag{14}$$

The partition function Z, as shown in eqn. (6), involves the exhaustive summation over all configurations. This means that its evaluation is computationally intractable; however, the following identity can be assumed for most machine learning systems:[46]

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log Z = \mathbb{E}_{\mathbf{v} \sim p(\mathbf{v})} \big[\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log f(\mathbf{v}) \big] \tag{15}$$

Taking advantage of this identity, Markov chain Monte Carlo (MCMC) sampling algorithms replace the intractable computation with random sampling that, after appropriate burn in, approximate the gradient of partition function with respect to network parameters,

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log Z \approx \frac{1}{N_{\text{samp}}} \sum_{s=1}^{N_{\text{samp}}} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log f(\mathbf{v}^{(s)}; \boldsymbol{\theta}) \tag{16}$$

In other words, a maximum likelihood estimation (MLE) estimates the unknown $p_{\text{data}}(\mathbf{v})$ by sampling a sufficiently large number of samples from the training set, and dividing the summation of their unnormalized probability by N_{samp}. More details on Markov chain Monte Carlo will be discussed later.

For the purpose of this investigation, it is critical that our proof-of-concept systems achieve a stable global minima, corresponding to the ground state energy of quantum states. As described in the introduction, the optimization landscape for the RBM is non-concave;

thus, local optimizers such as gradient descent may prevent us from reaching a globally optimal solution. Here we consider the viability of the BM compared to RBM. If BM does not contain any hidden vertices, the probability of a unit being on (a vertex with value 1) is given by a linear model of logistic regression because of the polynomial energy function.[46] One simplest BM is thus written as the following quadratic functions, and hereafter referred to as BM2:

$$E^{\mathrm{BM2}}(\mathbf{v}) = -\sum_{i=1}^{n} v_i b_i - \sum_{i,j=1}^{n} v_i v_j w_{ij} \,. \tag{17}$$

In such a case, the complexity of the encoded interaction in BM is limited in contrast to RBM (eqn. (7)), and it cannot model for higher order interactions because of the absence of the hidden layer. Even though the concave $L(\boldsymbol{\theta})$ is desirable, BM with only visible variables is at a disadvantage due to this inflexibility. Another way of encoding for higher-order interaction is by introducing higher-order terms into the energy function. The BM based on the third order energy function can be formulated by extending $E^{\mathrm{BM2}}(\mathbf{v})$ (eqn. (17)) to

$$E^{\mathrm{BM3}}(\mathbf{v}) = E^{\mathrm{BM2}}(\mathbf{v}) - \sum_{i,j,k=1}^{n} v_i v_j v_k p_{ijk} \,. \tag{18}$$

which we refer as BM3. This inclusion of higher order terms decrease the learning time, but comes with the tradeoff of increased number of connections, which induce higher computational cost.[47] However, as Luo and Sugiyama have shown extensively, by measuring $\mathbb{E}[D_{KL}(p_{\mathrm{data}}, p_{\mathrm{MLE}})]$, the difference in divergence for a third-order BM and a RBM is within an order of magnitude across a wide variety of sample sizes and number of model parameters.[43] Hence, investigation to its efficacy is still warranted. Note that BM2 and BM3 have a close relation with two-body and three-body denisty-based Jastrow wave functions.[48,49]

2.2 Stochastic Sampling of Configuration Space

Given the complete orthonormal configuration basis $\{\mathbf{v}\}$, the wave function $|\Psi\rangle$ can be expressed as the following CI expansion,

$$|\Psi\rangle = \sum_{\mathbf{v}} |\mathbf{v}\rangle \langle \mathbf{v}|\Psi\rangle = \sum_{\mathbf{v}} \Psi(\mathbf{v}) |\mathbf{v}\rangle \tag{19}$$

where we may use Slater determinants for $\mathbf{v}$ expressing configurations or n-site occupancies, and $\Psi(\mathbf{v})(= \langle \mathbf{v}|\Psi\rangle)$ serve as coefficients, which are expressed as a function of the n-bit descriptor $\mathbf{v}$. The ground state eigenvalue of Schrödinger equation $\mathcal{H} |\Phi_0\rangle = E_0 |\Phi_0\rangle$ corresponds to the expectation value of the Hamiltonian $\langle \mathcal{H} \rangle$ in the following form [eqn. (20)],

$$E_0 = \langle \mathcal{H} \rangle = \frac{\langle \Psi|\mathcal{H}|\Psi\rangle}{\langle \Psi|\Psi\rangle} = \frac{\sum_{\mathbf{v}} \langle \Psi|\mathbf{v}\rangle \ \langle \mathbf{v}|\mathcal{H}|\Psi\rangle}{\sum_{\mathbf{v}} \langle \Psi|\mathbf{v}\rangle \langle \mathbf{v}|\Psi\rangle} \tag{20}$$

with the correct $\Psi(\mathbf{v})$, actually given by $\langle \mathbf{v}|\Phi_0\rangle$ determined through the variational principle. Note that the number of configurations in sum grows at a combinatorial speed with increasing n, the number of single-particle sites. This can also be seen as analogy in the complexity of the evaluation of the partition function Z [eqn. (6)] in BM or, in a more general sense, probability theory.

This combinatorial complexity can be somewhat addressed by stochastic sampling approaches in ML using MCMC techniques, and in quantum many-body simulations using the variational Monte Carlo (VMC) method. Let us first form an ansatz, or a theoretically valid guess, to the function form of the wave function $|\Psi\rangle$, and parameterize this form with only polynomially many parameters $\boldsymbol{\theta}$. This specified function form should restrict the search space to a particular class, and different ansatz classes are usually suited for different systems due to their mathematical structures. Now if we reorganize eqn. (20) in a general way for evaluating the operator $\mathcal{O}$,

$$\langle \mathcal{O} \rangle = \frac{\sum_{\mathbf{v}} |\langle \Psi | \mathbf{v} \rangle|^2 \frac{\langle \mathbf{v} | \mathcal{O} | \Psi \rangle}{\langle \mathbf{v} | \Psi \rangle}}{\sum_{\mathbf{v}} |\langle \Psi | \mathbf{v} \rangle|^2} = \frac{\sum_{\mathbf{v}} |\langle \Psi | \mathbf{v} \rangle|^2 \mathcal{O}^{\mathrm{loc}}(\mathbf{v})}{\sum_{\mathbf{v}} |\langle \Psi | \mathbf{v} \rangle|^2} = \sum_{\mathbf{v}} \mathcal{P}(\mathbf{v}) \mathcal{O}^{\mathrm{loc}}(\mathbf{v}) \tag{21}$$

we can define the *local estimator* of the operator $\mathcal{O}$,

$$\mathcal{O}^{\mathrm{loc}}(\mathbf{v}) = \frac{\langle \mathbf{v} | \mathcal{O} | \Psi \rangle}{\langle \mathbf{v} | \Psi \rangle} \tag{22}$$

and more importantly, the classical probability distribution $\mathcal{P}(\mathbf{v})$,

$$\mathcal{P}(\mathbf{v}) = \frac{|\langle \Psi | \mathbf{v} \rangle|^2}{\sum_{\mathbf{v}} |\langle \Psi | \mathbf{v} \rangle|^2} \tag{23}$$

Therefore, computing the quantum average of operator $\mathcal{O}$ equates to taking the expectation of the local estimator with respect to the probability distribution $\mathcal{P}_{\mathrm{exact}}(\mathbf{v})$.[50] For the case of the Hamiltonian operator, the local estimator can be viewed as the *local energy* of $\mathcal{H}$,

$$\mathcal{E}^{\mathrm{loc}}(\mathbf{v}) = \frac{\langle \mathbf{v} | \mathcal{H} | \Psi \rangle}{\langle \mathbf{v} | \Psi \rangle} \tag{24}$$

The unknown probability distribution $\mathcal{P}(\mathbf{v})$ can be generated by a stochastic algorithm such as a MCMC in which a sequence of configurations are randomly sampled from $\{\mathbf{v}\}$.

$$\mathcal{P}_{\mathrm{prior}}(\mathbf{v}) \xrightarrow{\textsf{sufficient sampling}} \mathcal{P}_{\mathrm{eq}}(\mathbf{v}) \tag{25}$$

Eqn. (25) shows some initial prior distribution $\mathcal{P}_{\mathrm{prior}}(\mathbf{v})$ converging to the equilibrium distribution $\mathcal{P}_{\mathrm{eq}}(\mathbf{v})$ after a sequence of sampled distribution. After this equilibration, the quantum expectation value $\langle \mathcal{O} \rangle$ in eqn. (21) can be approximated as the mean of the local estimator over the N_{samp} visited configurations:

$$\begin{aligned}\langle \mathcal{O} \rangle = \mathbb{E}_{\mathbf{v}\sim\mathcal{P}_{\text{exact}}}\big[\mathcal{O}(\mathbf{v})\big] &= \sum_{\mathbf{v}} \mathcal{P}_{\text{exact}}(\mathbf{v})\mathcal{O}(\mathbf{v}) \\ &\approx \sum_{\mathbf{v}} \mathcal{P}_{\text{eq}}(\mathbf{v})\mathcal{O}^{\text{loc}}(\mathbf{v}) = \frac{1}{N_{\text{samp}}}\sum_{s=1}^{N_{\text{samp}}} \mathcal{O}^{\text{loc}}(\mathbf{v}^{(s)})\end{aligned} \tag{26}$$

The energy evaluation can thus be performed in this sampling form (eqn. (26)) using the local energy $\mathcal{E}^{\text{loc}}(\mathbf{v})$ ((24)) in place of $\mathcal{O}^{\text{loc}}(\mathbf{v})$. It is noted that $\mathcal{P}_{\text{eq}}(\mathbf{v})$ and $\mathcal{E}^{\text{loc}}(\mathbf{v})$ vary depending on $\boldsymbol{\theta}$, so that $\boldsymbol{\theta}$ serve as variational parameters of the energy. To indicate this $\boldsymbol{\theta}$ dependence, we denote the resultant energy as $E_{\boldsymbol{\theta}}$

To optimize the variational wave function in terms of minimizing the energy $E_{\boldsymbol{\theta}}$, we will also require the gradient of the energy with respect to the variational parameters: $\partial E_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}_I$, which can also be formulated as an expectation value with respect to $\mathcal{P}_{\text{eq}}(\mathbf{v})$ and evaluated along with eqn. (26) using the stochastic algorithm. The evaluated derivative permits us to update the variational parameters $\boldsymbol{\theta}$ with a multitude of gradient based optimization methods, such as gradient descent, Newton-Raphson, and stochastic reconfiguration, etc. In its simplest form, the gradient descent method updates the parameters in the negative derivative direction with

$$\boldsymbol{\theta}_I^{(t+1)} = \boldsymbol{\theta}_I^{(t)} - \alpha_t \, \frac{\partial E_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_I} \tag{27}$$

where α is a scaling factor or step size. This optimization function is iterated until the expectation value converges. In essence, VMC method allows us to map the eigenvalue problem as an optimization problem according to the variational principle. Unlike DMRG or the extended class of tensor-network methods, VMC provides wide applicability in the sense that the complexity of the ansatz can be arbitrarily independent of the range of interactions and the dimension of local Hilbert space, and we can thus treat a wide class of systems.[50]

2.3 Neural Network Quantum States

As described in the introduction, the discovery of neural networks for efficient data compression[45] has prompted the investigation into the potential of neural networks as an ansatz of wave functions, which is called NQS.[6,11] In particular, RBMs have shown promising results as an NQS ansatz.[11,13] If we reimagine the inputs to RBM at the binary observed variables $\mathbf{v}$ as physical configurations or n-site occupations in a chosen basis, and the binary hidden variables $\mathbf{h}$ as the correlation encoder, the probability distribution for calculating the expectation value of the operator in eqn. (23) can be reformulated as the joint probability distribution over the vertices of the RBM with $\mathcal{P}(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta})$ (eqn. (5)). This is a function of the visible units for the configuration $\mathbf{v}$, as well as arbitrary-m hidden vertices that determines the complexity the network can encode. Similarly, the marginalized probability $\mathcal{P}(\mathbf{v}; \boldsymbol{\theta})$ (eqn. (8)) carries the physical meaning of the likelihood of configuration v_i over the network, parameterized by $\boldsymbol{\theta}$, by summing over all hidden vertices.

Let eqn. (8) correspond to an ansatz of the variational expression of the quantum state, then the objects of quantum states (wave functions, density matrices or functions, etc) can be parameterized with the neural network $f^{\text{RBM}}(\mathbf{v}; \boldsymbol{\theta})$ (eqn. (9)). It acts as the computational black box that is a variational quantum object mapping the n-dimensional bit set $\mathbf{v}$ to the polynomially many variational parameters. Updating $\mathbf{b}, \mathbf{w} \in \boldsymbol{\theta}$ with eqn. (27) or other gradient based learning rules iteratively then leads to an accurate approximation. Notably, because of the auxiliary m size, the quality of the representation in principle increases proportionally with m. In practice, this feature is upper bounded by the computational power available. However, unlike the auxiliary indices in MPS, which only encodes 1D-like correlation, the correlation encoded by hidden vertices in RBM spans over the whole neural-network, suggesting that the NQS may be favored as a descriptor of systems of arbitrary dimension. Besides the RBM distribution $f^{\text{RBM}}(\mathbf{v}; \boldsymbol{\theta})$ (eqn. (9)), this study casts attention on the BM2 and BM3 architectures for the NQS model,

$$f^{\mathrm{BM2}}(\mathbf{v};\boldsymbol{\theta}) = e^{-E^{\mathrm{BM2}}(\mathbf{v};\boldsymbol{\theta})}\,, \tag{28}$$

$$f^{\mathrm{BM3}}(\mathbf{v};\boldsymbol{\theta}) = e^{-E^{\mathrm{BM3}}(\mathbf{v};\boldsymbol{\theta})}\,, \tag{29}$$

where the number of the parameters $\boldsymbol{\theta}$ for BM2 and BM3 is $(n + n^2)$ and $(n + n^2 + n^3)$, respectively.

It has been shown that deep Boltzmann machines (DBM), which has intralayer connections and capacity for more hidden layers, can efficiently represent (scaling with the number of particle at most polynomially) ground state of polynomially gapped Hamiltonians, and quantum states generated by any polynomial-size quantum circuits, some of which RBM fails to represent efficiently.[15] Nonetheless, the existence of an efficient representation does not equate to efficient evaluation. Since the problem of finding the ground state for a many-body system is intrinsically hard as previously described,[15] Ref. 15 indicated that without an efficient algorithm for training, obtaining convergence on this representation is thought to be a difficult task. The higher complexity in the DBM structure means the complicated contraction and required inference necessitates approximations. On the other hand, RBM can efficiently represent many highly entangled states with analytical contraction (as shown in eqn. (7)) and stochastic evaluation via VMC. Interestingly, neural networks have long been guaranteed by representability theorems to have network approximates of sufficiently smooth and regular high dimensional functions.[51,52] Furthermore, RBM in particular has proven ability to approximate any probability distribution with arbitrary accuracy if the efficiency is not capped.[15,53]For our purpose of solving chemical MR systems, the NQS with RBM and higher-order BM structures should in theory provide satisfactory results.

3 Quantum Chemical Formulation and Implementation

Our implementation of the Boltzmann machine based NQS builds over a general QC computer program, and is relatively straightforward. Given the molecular orbitals (MOs), which are obtained by a precursor Hartree-Fock or other-type MO calculation and expressed with a linear combination of user-input atomic orbital (AO) basis functions, the active-space Hamiltonian in the CAS approach is specified as,

$$\mathcal{H} = H_0 + \sum_{pq}^{K} (p|\hat{h}|q)\, \hat{E}_{pq} + \frac{1}{2} \sum_{pqrs}^{K} (pq|rs) \left(\hat{E}_{pq} \hat{E}_{rs} - \delta_{qr} \hat{E}_{ps} \right) \tag{30}$$

where p, q, ... refer to the orbital indices running over the known K MOs selected as active orbitals for the CAS treatment[17]. The second quantization is written using the spin-averaged substitution (or excitation) operator $\hat{E}_{pq}$, given by $\hat{E}_{pq} = \sum_{\sigma=\alpha,\beta} \hat{a}^{\dagger}_{p\sigma} \hat{a}_{q\sigma}$ with the annihilation (creation) operators $\hat{a}^{(\dagger)}_{p\sigma}$, where σ is a spin index taking either α (or ↑) or β (or ↓). These K MOs respectively serve as single-particle occupation sites of $\mathbf{v}$ for the CI expansion (eqn. (19)). Eqn. (30) is predetermined in the sense that the one- and two-electron integrals,[54] denoted $(p|\hat{h}|q)$ and $(pq|rs)$, respectively, along with the constant H_0 are generated prior to the CI or NQS calculations, through the AO to MO basis transformation computation.

3.1 Form of wave function

Our formulation is built upon the CI expansion of $|\Psi\rangle$ using the occupation representations of the configurations, formulated as follows:

$$|\Psi\rangle = \sum_{\mathbf{v}} C_{\mathbf{v}} |\mathbf{v}\rangle \tag{31}$$

$$|\mathbf{v}\rangle = \left| v_1^{\alpha}, v_2^{\alpha} \ldots v_K^{\alpha}, v_1^{\beta}, v_2^{\beta} \ldots v_K^{\beta} \right\rangle \tag{32}$$

$$v_j^{\sigma} = \{0, 1\} \tag{33}$$

In eqns. (32) and (33), we use $v_j^{\sigma} = 0$ for unoccupied spin orbitals, and $v_j^{\sigma} = 1$ for occupied spin orbitals. These states of the spin orbitals directly act as the states of the visible units of Boltzmann machines. This indicates that there are $2K$ visible units built into our neural network, i.e., setting $n = 2K$. The occupancies for $\mathbf{v}$ considered in eqn. (31) are subject to the fact that $\sum_{j=1}^{K} v_j^{\alpha}$ gives the total N_{α} electrons in the K MOs, similarly for β. As an example, eqn. (32) of indocyanine green in CAS(4e,4o), and the corresponding molecular orbitals are shown in Figure 1.

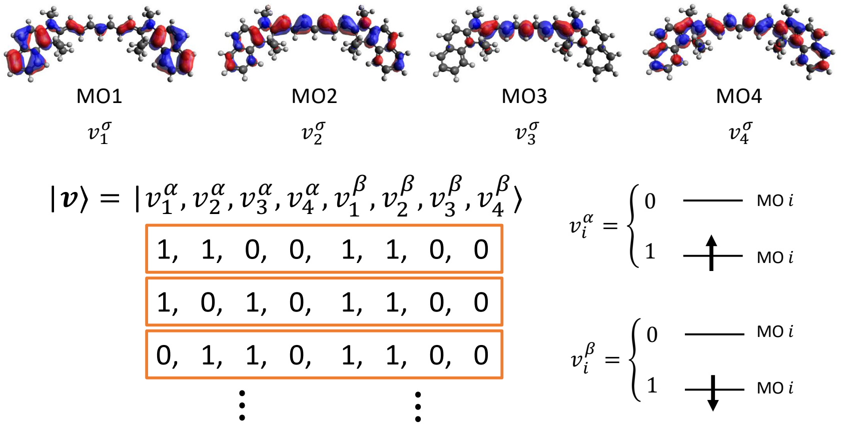


Figure 1: Configurations of indocyanine green in CAS(4e,4o) for the singlet state, and the corresponding molecular orbitals. This should give a total of $[\binom{4}{2}]^2 (= 36)$ configurations.

The expansion coefficients $C_{\mathbf{v}}$ are now a central machine-learned object. We proceed to the formulation of its neural network using the following ansatz form,

$$C_{\mathbf{v}} = \underbrace{e^{\frac{\imath}{2}\log f(\mathbf{v};\boldsymbol{\tau})}}_{\text{phase}} \underbrace{\sqrt{\frac{1}{Z(\boldsymbol{\theta})} f(\mathbf{v};\boldsymbol{\theta})}}_{\text{amplitude}} \ . \tag{34}$$

Two real-value-parameterized Boltzmann machines $f(\mathbf{v};\boldsymbol{\theta})$ and $f(\mathbf{v};\boldsymbol{\tau})$ are used to encode amplitude and phase of $C_{\mathbf{v}}$, respectively. Visually, this network distribution with a common input layer but two separate layers of hidden nodes gives it a *sandwich* structure, depicted in Figure 2. This, in contrast to eqn. (9), enables a visible layer with input $\mathbf{v}$ to be connected to two separate hidden layers through two different parameter sets $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$. We write the RBM distributions $f(\mathbf{v};\boldsymbol{\theta})$ and $f(\mathbf{v};\boldsymbol{\tau})$ based on eqns. (7) and (9) as:

$$\begin{aligned} f(\mathbf{v};\boldsymbol{\theta}) &= \sum_{\mathbf{h}} e^{-E^{\mathrm{RBM}}(\mathbf{v},\mathbf{h};\boldsymbol{\theta})} = \exp\left\{ \sum_i^{2K} v_i b_i + \sum_j^{m} \log\left(1 + c^{c_j + \sum_i^{2K} v_i w_{ij}}\right)\right\} \\ f(\mathbf{v};\boldsymbol{\tau}) &= \sum_{\mathbf{h}} e^{-E^{\mathrm{RBM}}(\mathbf{v},\mathbf{h};\boldsymbol{\tau})} = \exp\left\{ \sum_i^{2K} v_i d_i + \sum_j^{m} \log\left(1 + e^{e_j + \sum_i^{2K} v_i x_{ij}}\right)\right\} \end{aligned} \tag{35}$$

with $\boldsymbol{\theta} = \{\mathbf{b}, \mathbf{c}, \mathbf{w}\}$ and $\boldsymbol{\tau} = \{\mathbf{d}, \mathbf{e}, \mathbf{x}\}$. There is a single partition function in eqn. (34), which is a function of $\boldsymbol{\theta}$, evaluated to be $Z(\boldsymbol{\theta}) = \sum_{\mathbf{v}} f(\mathbf{v};\boldsymbol{\theta})$.

In addition to the real-valued phase-amplitude parameterization of NQS used in this study (Eqn. 34), we tested the complex-valued NQS parameterization, which is more commonly found in the literature and indeed employed in Ref. 36. The complex-valued NQS can avoid having a separated NQS for extra representing the phases of Fermionic wave functions and can thus be based on a single BM energy function. However, with our test implementation, we observed serious numerical difficulties in optimizing complex-valued NQS parameters.

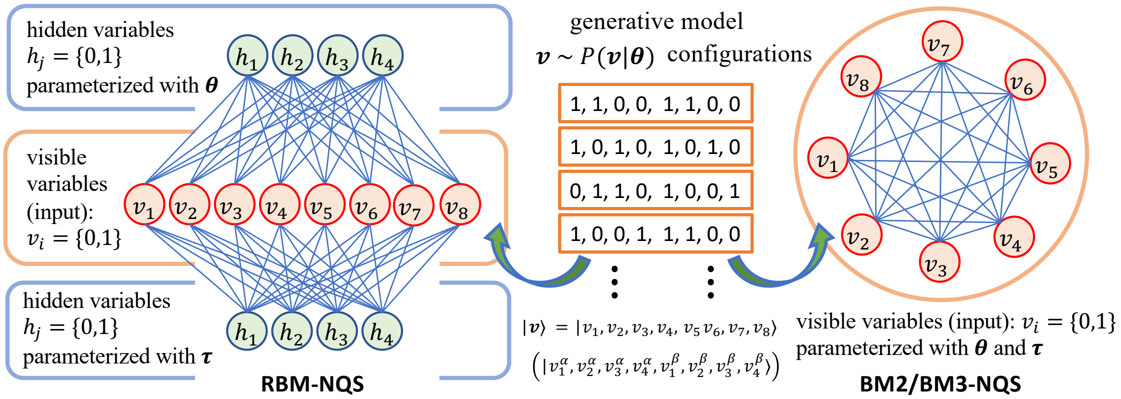

hidden variables
$h_j = \{0,1\}$
parameterized with $\boldsymbol{\theta}$
visible
variables
(input):
$v_i = \{0,1\}$
hidden variables
$h_j = \{0,1\}$
parameterized with $\boldsymbol{\tau}$
h_1 h_2 h_3 h_4
v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8
RBM-NQS
generative model
$\boldsymbol{v} \sim P(\boldsymbol{v}|\boldsymbol{\theta})$ configurations
1, 1, 0, 0, 1, 1, 0, 0
1, 0, 1, 0, 1, 0, 1, 0
0, 1, 1, 0, 1, 0, 0, 1
1, 0, 0, 1, 1, 1, 0, 0
$|\boldsymbol{v}\rangle = |v_1, v_2, v_3, v_4, v_5 v_6, v_7, v_8\rangle$
$(|v_1^\alpha, v_2^\alpha, v_3^\alpha, v_4^\alpha, v_1^\beta, v_2^\beta, v_3^\beta, v_4^\beta\rangle)$
visible variables (input): $v_i = \{0,1\}$
parameterized with $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$
BM2/BM3-NQS

Figure 2: An example of the implemented *sandwich* RBM and BM2/BM3 architectures. A CAS(4e,4o) calculation with 4 molecular orbitals will give 8 input nodes to contain both α and β spins, while arbitrary number of hidden nodes can be specified. In this case, 4 hidden nodes for both θ and τ. Configurations are fed into visible nodes as inputs.

The square of $C_{\mathbf{v}}$ (eqn. (34)) is interpreted in quantum mechanics as the probability of the occurrence of the associated configuration $|\mathbf{v}\rangle$. It is written as

$$\begin{aligned} |C_{\mathbf{v}}|^2 &= \left(e^{\frac{i}{2}\log f(\mathbf{v};\boldsymbol{\tau})}\sqrt{\frac{1}{Z(\boldsymbol{\theta})}f(\mathbf{v};\boldsymbol{\theta})}\right)^* \left(e^{\frac{i}{2}\log f(\mathbf{v};\boldsymbol{\tau})}\sqrt{\frac{1}{Z(\boldsymbol{\theta})}f(\mathbf{v};\boldsymbol{\theta})}\right) \\ &= \frac{1}{Z(\boldsymbol{\theta})}f(\mathbf{v};\boldsymbol{\theta}) \\ &= \mathcal{P}(\mathbf{v};\boldsymbol{\theta}) \end{aligned} \tag{36}$$

which is reduced to the $\boldsymbol{\tau}$ (phase) independent representation. This indicates that we use a Boltzman machine distribution $f(\mathbf{v};\boldsymbol{\theta})$ as the probability density of the quantum mechanical wave function, in exactly the same way as using it for ML. In the ML, this system is known as the autoencoder.

The extension of our NQS model to the BM2 and BM3 architectures can be readily formulated by replacing the energy function with the corresponding expressions $E^{\text{BM2}}(\mathbf{v})$ (eqn. (17)) and $E^{\text{BM3}}(\mathbf{v})$ (eqn. (18)). The amplitude and phase distributions of the BM3 based NQS are thus given by

$$
\begin{aligned}
f(\mathbf{v};\boldsymbol{\theta}) &= \sum_{\mathbf{h}} e^{-E^{\mathrm{BM3}}(\mathbf{v},\mathbf{h};\boldsymbol{\theta})} = \exp\left\{\sum_{i}^{2K} v_i b_i + \sum_{ij}^{2K} v_i v_j w_{ij} + \sum_{ijk}^{2K} v_i v_j v_k p_{ijk}\right\}, \\
f(\mathbf{v};\boldsymbol{\tau}) &= \sum_{\mathbf{h}} e^{-E^{\mathrm{BM3}}(\mathbf{v},\mathbf{h};\boldsymbol{\tau})} = \exp\left\{\sum_{i}^{2K} v_i d_i + \sum_{ij}^{2K} v_i v_j x_{ij} + \sum_{ijk}^{2K} v_i v_j v_k q_{ijk}\right\},
\end{aligned}
\tag{37}
$$

respectively, with $\boldsymbol{\theta} = \{\mathbf{b}, \mathbf{w}, \mathbf{p}\}$ and $\boldsymbol{\tau} = \{\mathbf{d}, \mathbf{x}, \mathbf{q}\}$. The BM2 variant is obtained by the truncation of the above BM3 expressions to the second order.

3.2 Energy expression

Now the expectation value of an operator $\mathcal{O}$, shown in eqn. (20) can be derived as,

$$
\langle \mathcal{O} \rangle = \frac{\langle \Psi|\mathcal{O}|\Psi\rangle}{\langle \Psi|\Psi\rangle} = \frac{\sum_{\mathbf{v}\mathbf{v}'} \langle \mathbf{v}|\, \mathcal{O}\, |\mathbf{v}'\rangle\, C^*_{\mathbf{v}} C_{\mathbf{v}'}}{\sum_{\mathbf{v}} C^*_{\mathbf{v}} C_{\mathbf{v}}} = \sum_{\mathbf{v}\mathbf{v}'} \langle \mathbf{v}|\, \mathcal{O}\, |\mathbf{v}'\rangle\, C^*_{\mathbf{v}} C_{\mathbf{v}'}
\tag{38}
$$

using the normalization of the probability $\mathcal{P}(\mathbf{v};\boldsymbol{\theta})$ (eqn. (36)) and wave function $|\Psi\rangle$. Inserting eqn. (34) to eqn. (38), we can further derive $\langle \mathcal{O} \rangle$,

$$
\langle \mathcal{O} \rangle = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{v}\mathbf{v}'} \langle \mathbf{v}|\, \mathcal{O}\, |\mathbf{v}'\rangle\, e^{\frac{\imath}{2} \log \frac{f(\mathbf{v}';\boldsymbol{\tau})}{f(\mathbf{v};\boldsymbol{\tau})}} \sqrt{f(\mathbf{v};\boldsymbol{\theta}) f(\mathbf{v}';\boldsymbol{\theta})}
\tag{39}
$$

At each iteration of the variational learning, the local energy $\mathcal{E}^{\mathrm{loc}}$ are evaluated, combining the above equations and eqn. (24) as,

$$
\mathcal{E}^{\mathrm{loc}}(\mathbf{v}) = \frac{\langle \mathbf{v}|\mathcal{H}|\Psi\rangle}{C_{\mathbf{v}}} = \sum_{\mathbf{v}'} \langle \mathbf{v}|\, \mathcal{H}\, |\mathbf{v}'\rangle \frac{C_{\mathbf{v}'}}{C_{\mathbf{v}}},
\tag{40}
$$

with

$$
\frac{C_{\mathbf{v}'}}{C_{\mathbf{v}}} = e^{\frac{\imath}{2} \log \frac{f(\mathbf{v}';\boldsymbol{\tau})}{f(\mathbf{v};\boldsymbol{\tau})}} \sqrt{\frac{f(\mathbf{v}';\boldsymbol{\theta})}{f(\mathbf{v};\boldsymbol{\theta})}},
\tag{41}
$$

which notably does not involve the partition function $Z(\boldsymbol{\theta})$. This local energy depends on the variational parameters $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$; thus it may be written as $\mathcal{E}^{\mathrm{loc}}(\mathbf{v};\boldsymbol{\theta},\boldsymbol{\tau})$. The energy

E, the expectation value of $\mathcal{H}$ (eqn. (20)), is rewritten as an expectation value of the local energy,

$$\begin{aligned} E(\boldsymbol{\theta}, \boldsymbol{\tau}) &= \left\langle \mathcal{E}^{\text{loc}}(\mathbf{v}; \boldsymbol{\theta}, \boldsymbol{\tau}) \right\rangle_{\mathbf{v} \sim \mathcal{P}(\mathbf{v}; \boldsymbol{\theta})} \\ &= \sum_{\mathbf{v}} \mathcal{P}(\mathbf{v}; \boldsymbol{\theta})\, \mathcal{E}^{\text{loc}}(\mathbf{v}; \boldsymbol{\theta}, \boldsymbol{\tau}) \end{aligned} \tag{42}$$

and is calculated to be an average of the simulated ensemble of the local energies,

$$E(\boldsymbol{\theta}, \boldsymbol{\tau}) \approx \frac{1}{N_{\text{samp}}} \sum_{s=1}^{N_{\text{samp}}} \mathcal{E}^{\text{loc}}(\mathbf{v}^{(s)}; \boldsymbol{\theta}, \boldsymbol{\tau})\,. \tag{43}$$

This can be evaluated with the random samples for $\mathbf{v}^{(s)}$ generated by the Metropolis-Hasting algorithm, a MCMC random sampling algorithm that we will discuss at the end of this section.

Additionally noted is that for given $\mathbf{v}$, the local energy $\mathcal{E}^{\text{loc}}(\mathbf{v})$ (eqn. (40)) is computed at a polynomial cost due to the sparsity of the matrix $\langle \mathbf{v} | \mathcal{H} | \mathbf{v}' \rangle$ associated to the fact that the Hamiltonian $\mathcal{H}$ (eqn. (30)) contains up to two-body operators.

3.3 Optimization of NQS

The goal of the training is to optimize the wave function by finding ANN's parameters $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ that satisfy the following variational conditions:

$$\begin{cases} F_\theta = \dfrac{\partial}{\partial \theta} E(\boldsymbol{\theta}, \boldsymbol{\tau}) = 0 \quad \forall\, \theta \in \boldsymbol{\theta} \\ F_\tau = \dfrac{\partial}{\partial \tau} E(\boldsymbol{\theta}, \boldsymbol{\tau}) = 0 \quad \forall\, \tau \in \boldsymbol{\tau} \end{cases} \tag{44}$$

As described earlier, the optimization is done using the iterative gradient based algorithms. With the identity $\frac{\partial}{\partial \alpha} C_{\mathbf{v}} = C_{\mathbf{v}} \frac{\partial}{\partial \alpha} \log C_{\mathbf{v}}$, the samplig form to evaluate eqn. (44) or *force* is derived in a similar style of the energy evaulation (eqn. (42)) as

$$\begin{aligned} F_\theta &= 2\,\mathrm{Re}\left[\sum_{\mathbf{v}} \mathcal{P}(\mathbf{v};\boldsymbol{\theta})\,\mathcal{E}^{\mathrm{loc}*}_{\mathbf{v}}(O^\theta_{\mathbf{v}} - \langle O^\theta\rangle)\right] = 2\,\mathrm{Re}\left[\left\langle \mathcal{E}^{\mathrm{loc}*}O^\theta\right\rangle - \left\langle \mathcal{E}^{\mathrm{loc}*}\right\rangle \left\langle O^\theta\right\rangle\right], \\ F_\tau &= 2\,\mathrm{Re}\left[\sum_{\mathbf{v}} \mathcal{P}(\mathbf{v};\boldsymbol{\theta})\,\mathcal{E}^{\mathrm{loc}*}_{\mathbf{v}}(O^\tau_{\mathbf{v}} - \langle O^\tau\rangle)\right] = 2\,\mathrm{Re}\left[\left\langle \mathcal{E}^{\mathrm{loc}*}O^\tau\right\rangle - \left\langle \mathcal{E}^{\mathrm{loc}*}\right\rangle \left\langle O^\tau\right\rangle\right], \end{aligned} \tag{45}$$

where $\mathcal{E}^{\mathrm{loc}}_{\mathbf{v}} \equiv \mathcal{E}^{\mathrm{loc}}(\mathbf{v};\boldsymbol{\theta},\boldsymbol{\tau})$, and we define the *local derivatives* of $\log C_{\mathbf{v}}$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ by

$$\begin{aligned} O^\theta_{\mathbf{v}} &= \frac{\partial}{\partial\theta}\,\log C_{\mathbf{v}}\,, \\ O^\tau_{\mathbf{v}} &= \frac{\partial}{\partial\tau}\,\log C_{\mathbf{v}}\,. \end{aligned} \tag{46}$$

The gradient $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\,\log C_{\mathbf{v}}$ can be written using the gradients of $f(\mathbf{v};\boldsymbol{\theta})$ and $Z(\boldsymbol{\theta})$ as:

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}}\,\log C_{\mathbf{v}} = \frac{1}{2}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\log f(\mathbf{v};\boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\log Z(\boldsymbol{\theta})\,. \tag{47}$$

This relation allows us to rewrite the gradients of the energy F_θ (eqn. (45)) as:

$$F_\theta = \underbrace{\frac{1}{N_{\mathrm{samp}}}\sum_{s=1}^{N_{\mathrm{samp}}} \mathrm{Re}\left[\mathcal{E}^{\mathrm{loc}*}_{\mathbf{v}^{(s)}}\ \boldsymbol{\nabla}_{\boldsymbol{\theta}}\log f(\mathbf{v}^{(s)})\right]}_{\mathrm{X}} - \underbrace{\left\langle \mathcal{E}^{\mathrm{loc}*}\right\rangle \boldsymbol{\nabla}_{\boldsymbol{\theta}}\log Z(\boldsymbol{\theta})}_{\mathrm{Y}}\ . \tag{48}$$

where we compute the X and Y terms using the sampled *data* generated from the generative BM-based model via the MCMC method. We note that there is a close formal relation between eqn. 48 and the KL divergence (distance) $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\log L(\boldsymbol{\theta})$ (eqn. (14)) where X and Y in eqn. (48) correspond to 'data' and 'model,' respectively, in eqn. (14). It should be stressed that the KL divergence, often discussed in the ML algorithm to train the BM model with given big data, itself is not computed in this study. Nonetheless, it is seemingly interesting to formally relate the KL divergence to the gradients of energy with respect to the BM parameters (eqn. (48)), which can be viewed as a *variant* of the KL divergence written with gradient components weighted with local energies.

The total derivatives $\langle O^\theta \rangle$ and $\langle O^\tau \rangle$ are obtained as:

$$\begin{aligned}
\langle O^\theta \rangle &= \langle O^\theta_{\mathbf{v}} \rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} = \sum_{\mathbf{v}} \mathcal{P}(\mathbf{v};\boldsymbol{\theta})\, O^\theta_{\mathbf{v}}\,, \\
\langle O^\tau \rangle &= \langle O^\tau_{\mathbf{v}} \rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} = \sum_{\mathbf{v}} \mathcal{P}(\mathbf{v};\boldsymbol{\theta})\, O^\tau_{\mathbf{v}}\,.
\end{aligned} \tag{49}$$

Due to the highly non-linear dependence on variational parameters, the simple steepest descent approach often gives unsatisfactory results.[50] To deal with this issue, a gradient based method called the stochastic reconfiguration (SR) is implemented. The SR defines the Hermitian covariance matrices that update the biases and weights with the rule:

$$\begin{aligned}
\theta^{(t+1)} &= \theta^{(t)} - \alpha_t \sum_{\theta'} S^{-1}_{\theta\theta'} F_{\theta'}\,, \\
\tau^{(t+1)} &= \tau^{(t)} - \alpha_t \sum_{\tau'} S^{-1}_{\tau\tau'} F_{\tau'}\,.
\end{aligned} \tag{50}$$

The Hermitian covariance matrices $S_{\theta\theta'}$ and $S_{\tau\tau'}$ are derived as metrics in the space of normalized wave functions, and relate to the distance between two wave functions. This distance metric improves the convergence to the minimal variational energy with respect to the steepest descent approach because the non-equivalent parameters can now be suitably updated at different scales.[50] Just as the forces F_θ and F_τ (eqn. (45)), the matrices $S_{\theta\theta'}$ and $S_{\tau\tau'}$ are given as,

$$\begin{aligned}
S_{\theta\theta'} &= \mathrm{Re}\left[\sum_{\mathbf{v}} \mathcal{P}(\mathbf{v};\boldsymbol{\theta})\left(O^\theta_{\mathbf{v}} - \langle O^\theta\rangle\right)^*\left(O^{\theta'}_{\mathbf{v}} - \langle O^{\theta'}\rangle\right)\right] \\
S_{\tau\tau'} &= \mathrm{Re}\left[\sum_{\mathbf{v}} \mathcal{P}(\mathbf{v};\boldsymbol{\theta})\left(O^\tau_{\mathbf{v}} - \langle O^\tau\rangle\right)^*\left(O^{\tau'}_{\mathbf{v}} - \langle O^{\tau'}\rangle\right)\right]
\end{aligned} \tag{51}$$

and can be evaluated within the Monte Carlo sampling.

We now turn to the formulations of the programmable forms of the local derivatives $O^\theta_{\mathbf{v}}$ and $O^\tau_{\mathbf{v}}$ (eqn. (47)) specific to the RBM, BM2, and BM3 architectures. For the RBM, they are obtained based on eqn. (35) for the differentiations with respect to the parameters $b_I, c_J, w_{IJ} \in \boldsymbol{\theta}$ and $d_I, e_J, x_{IJ} \in \boldsymbol{\tau}$ for the given $\mathbf{v}'$,

$$\begin{aligned}
O_{\mathbf{v}'}^{b_I} &= \frac{1}{2}\left[v_I' - \langle v_I \rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} \right] , \\
O_{\mathbf{v}'}^{c_J} &= \frac{1}{2}\left[\mathrm{Sig}\left(c_J + \sum_i v_i' w_{iJ} \right) - \left\langle \mathrm{Sig}\left(c_J + \sum_i v_i w_{iJ} \right) \right\rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} \right] , \\
O_{\mathbf{v}'}^{w_{IJ}} &= \frac{1}{2}\left[v_I' \; \mathrm{Sig}\left(c_J + \sum_i v_i' w_{iJ} \right) - \left\langle v_I \; \mathrm{Sig}\left(c_J + \sum_i v_i w_{iJ} \right) \right\rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} \right] , \\
O_{\mathbf{v}'}^{d_I} &= \frac{i}{2}\, v_I' \, , \\
O_{\mathbf{v}'}^{e_J} &= \frac{i}{2}\; \mathrm{Sig}\left(d_J + \sum_i v_i' x_{iJ} \right) , \\
O_{\mathbf{v}'}^{x_{IJ}} &= \frac{i}{2}\, v_I' \; \mathrm{Sig}\left(d_J + \sum_i v_i' x_{iJ} \right) ,
\end{aligned} \tag{52}$$

respectively, where $\mathrm{Sig}\,(x)$ is the sigmoid function defined by $\mathrm{Sig}\,(x) = \frac{e^x}{e^x+1}$. Given the configuration $\mathbf{v}'$, the local derivatives for BM3 formulated using eqn. (35) reach the following expressions:

$$\begin{aligned}
O_{\mathbf{v}'}^{b_I} &= \frac{1}{2}\left[v_I' - \langle v_I \rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} \right] \\
O_{\mathbf{v}'}^{w_{IJ}} &= \frac{1}{2}\left[v_I' v_J' - \langle v_I v_J \rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} \right] \\
O_{\mathbf{v}'}^{p_{IJK}} &= \frac{1}{2}\left[v_I' v_J' v_K' - \langle v_I v_J v_K \rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} \right] \\
O_{\mathbf{v}'}^{d_I} &= \frac{i}{2}\, v_I' \\
O_{\mathbf{v}'}^{x_{IJ}} &= \frac{i}{2}\, v_I' v_J' \\
O_{\mathbf{v}'}^{q_{IJK}} &= \frac{i}{2}\, v_I' v_J' v_K'
\end{aligned} \tag{53}$$

for the derivatives with respect to BM3's parameters $b_I, w_{IJ}, p_{IJK} \in \boldsymbol{\theta}$ and $d_I, x_{IJ}, q_{IJK} \in \boldsymbol{\tau}$, respectively. The BM2 local derivatives are the same as the first and second order terms of the above expressions.

3.4 Deterministic and Stochastic Implementations

The network can now update with eqn. (50) to variationally learn the ground state wave function that gives the minimal variational energy. The most computationally intensive step of the NQS training is to evaluate the expectation values of the $\mathbf{v}$ dependent objects, say, $f(\mathbf{v})$, over the configuration space with the probability distribution $\mathcal{P}(\mathbf{v};\boldsymbol{\theta})$, as denoted $\langle f(\mathbf{v})\rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})}$. We have implemented two approaches for this step from different perspectives:

(a) *Deterministic approach*, which performs the summation over the CI space by sweeping $\mathbf{v}$ across all the configurations, written as $\langle f(\mathbf{v})\rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} = \sum_{\mathbf{v}} \mathcal{P}(\mathbf{v};\boldsymbol{\theta})\, f(\mathbf{v})$;

(b) *Stochastic approach*, which takes a statistical average from the MCMC sampling of the configurations, $\langle f(\mathbf{v})\rangle_{\mathbf{v}\sim\mathcal{P}(\mathbf{v};\boldsymbol{\theta})} = \frac{1}{N_{\text{samp}}}\sum_{s=1}^{N_{\text{samp}}} f(\mathbf{v}^{(\mathbf{s})})$.

The deterministic approach [(a)] is useful for benchmarks for assessing the BM and RBM models, although it fundamentally embraces the combinatorial complexity in the summation and thus is not scalable. It can eliminate the random sampling introduced with MCMC for a more facile evaluation of network convergence. Furthermore, we stress this study as a proof-of-concept one, and even small system size should give us sufficient results to understand the performance of the NQS models. Note that the probability function $\mathcal{P}(\mathbf{v};\boldsymbol{\theta})$ is evaluated with the partition function $Z(\boldsymbol{\theta})$, which also requires a summation over all the configurations.

The real ML implementation is based on the stochastic approach [(b)] considered to be a way of circumventing the combinatorial complexity. The variant of Markov chain Monte Carlo sampling algorithm we have applied is the Metropolis-Hasting (MH) algorithm. In MH, a set of n_{w} walkers sample randomly from the set of configurations $\{\mathbf{v}\}$, then each configuration in the new set of n_{w} sampled configurations, labeled as x', are accepted with the conditional probability $A(x'|x_u)$, where x_u is the last configuration sampled by the same walker, and u is the MH iteration number. Metaphorically, the n_{w} walkers take random walks in the space, which for our case spans set $|\mathbf{v}\rangle$, forming n_{w} Markov chains. Mathematically, the probability distribution $\mathcal{P}$ that the algorithm is sampling from is guaranteed to converge as described in section 2.2 with equation (25) when the detailed balance condition is satisfied. For the Metropolis algorithm, this means the conditional probability A for accepting new samples is:

$$A(x'|x_u) = \mathrm{Min}\left\{1, \frac{\mathcal{P}_{\mathrm{eq}}(x')}{\mathcal{P}_{\mathrm{eq}}(x_u)}\right\} \tag{54}$$

Practically, with a given x_u, x' is generated randomly with uniform distribution from all possible configurations that preserve a given number of particles. In the implementation, a random probability (η) is generated from $[0, 1)$ uniformly, and x' is accepted if $A(x'|x_u) \geq \eta$. As written in Sec. 3.1, every x preserves the total numbers of the system's α and β electrons N_α and N_β.

On a further note, due to the nature of the sampling algorithm, our stochastic implementation is easily parallelized. The n_{w} walkers are evenly distributed among n_{proc} computation processes via the Message Passing Interface (MPI) programming model, and the walkers in the same processes are further distributed among n_{thread} threads via Open Multi-Processing (OpenMP). In this manner, computational power of each computing core is maximized such that each thread of a computational core handles a minimal proportion of n_{w}. MPI further allow us to distribute this task among several computing cores. To prompt more efficient training, we have opted for a dynamic burn-in for the MCMC sampling, during which the sampling iteration is low at the beginning of the training to promote better convergence, and avoid local minima. The sampling iteration is then dramatically increased after the energy stabilized to within 10^{-1} hartree over a 10 iteration period to provide accuracy. A simplified graphical representation of our implemented algorithm is shown in Figure 3. We used our in-house program suites ORZ[55] for performing quantum chemistry calculations ranging from Hartree-Fock to RBM calculations.

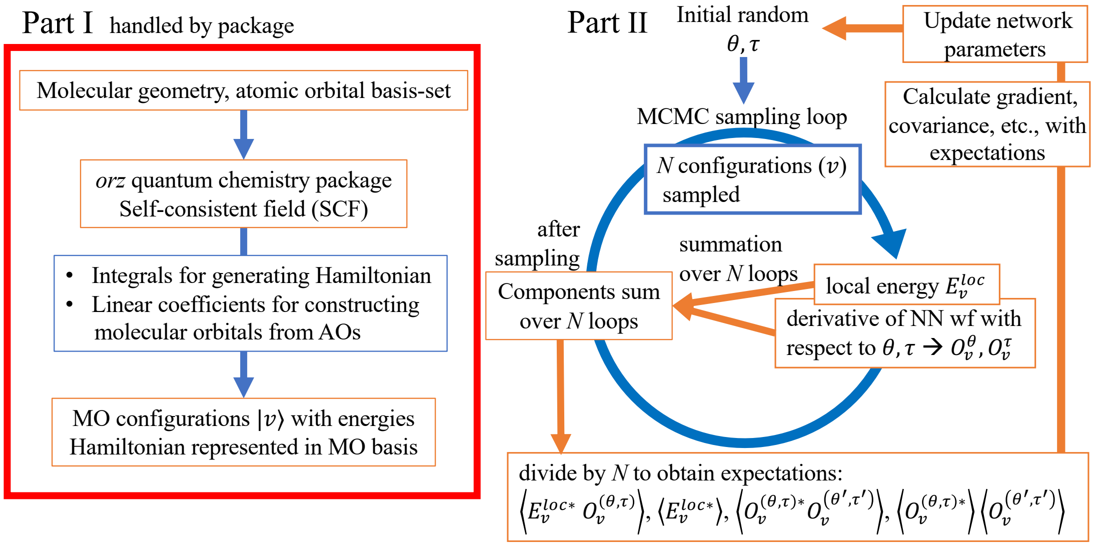

Part I handled by package
Molecular geometry, atomic orbital basis-set
orz quantum chemistry package
Self-consistent field (SCF)
• Integrals for generating Hamiltonian
• Linear coefficients for constructing molecular orbitals from AOs
MO configurations $|v\rangle$ with energies
Hamiltonian represented in MO basis
Part II
Initial random θ, τ
Update network parameters
Calculate gradient, covariance, etc., with expectations
MCMC sampling loop
N configurations (v) sampled
after sampling
summation over N loops
local energy E_v^{loc}
derivative of NN wf with respect to $\theta, \tau \rightarrow O_v^{\theta}, O_v^{\tau}$
Components sum over N loops
divide by N to obtain expectations:
$\left\langle E_v^{loc*} O_v^{(\theta,\tau)} \right\rangle, \langle E_v^{loc*} \rangle, \left\langle O_v^{(\theta,\tau)*} O_v^{(\theta',\tau')} \right\rangle, \left\langle O_v^{(\theta,\tau)*} \right\rangle \left\langle O_v^{(\theta',\tau')} \right\rangle$

Figure 3: A graphical depiction of our implemented variational Monte Carlo neural-network quantum state algorithm for quantum chemical calculations.

Let us briefly discuss the computational complexity of the stochastic approach. The most demanding portion of the algorithm is the processes executed within the MCMC sampling; thus, their computational complexities are expressed as proportional to the number of MCMC samplings, N_{samp}, which is a large factor. It is mainly decomposed into three important parts: the evaluations of the local energies $\mathcal{E}^{\text{loc}}$ (eqn. (40)), local gradients $O_{\mathbf{v}}^{\theta}$ or $O_{\mathbf{v}}^{\tau}$ (eqn. (47)), and covariance matrices $\mathbf{S}$ (eqn. (51)). Let these complexities per a given sampled configuration be denoted as K_{eloc}, K_{deriv}, and K_S, for local energies, local gradients, and covariance matrices, respectively. Then, the number of the operation counts of the MCMC step are in total estimated to be

$$O(N_{\text{samp}}(K_{\text{eloc}} + K_{\text{deriv}} + K_S)) \tag{55}$$

Assuming that the numbers of α and β spins are equal, i.e. $N_\alpha \simeq N_\beta$, the local energies can be computed at the cost formulated as $K_{\text{eloc}} = O(N_\alpha^2(K - N_\alpha)^2)$. As shown in eqns. (47) and (51), K_{deriv} and K_S basically depend on the number of the ANN parameters $\{\boldsymbol{\theta}, \boldsymbol{\tau}\}$, denoted N_{param}. They are approximately formulated as $K_{\text{deriv}} = O(N_{\text{param}})$ and $K_S = O(N_{\text{param}}^2)$

where N_{param} depends on the model used, given as

$$N_{\text{param}} = \begin{cases} 4K + 2m + 4Km & \text{(RBM)} \\ 4K + 8K^2 & \text{(BM2)} \\ 4K + 8K^2 + 16K^3 & \text{(BM3)} \end{cases} \tag{56}$$

As shown above, K_{eloc}, K_{deriv}, and K_S are all fitted to polynomial forms. The cost of the SR based update to the parameters of RBM and BM, however, grows steeply with N_{param}. This may be mitigated by the use of the subspace technique developed by Neuscamman et al.[56]

4 Numerical examples

To assess the implementation, we performed calculations of CAS-CI wave functions using the NQS models on indocyanine green (ICG) and ground state dinitrogen ($X^1 \sum_g^+$) as test cases. The training of the NQS was judged to be converged when the force of amplitude F_θ dropped below 1×10^{-5} E_h. Due to the non-concave log-likelihood function of RBM as well as the numerical sensitivity of NQS architectures, energy convergence falling into excited states as solution has been observed possible. However, only the ground state convergence are reported here. We used restricted Hartree-Fock orbitals for doubly-occupied and active MOs of CAS-CI wave functions. The MOs were represented using the STO-3G basis set, which is minimal atomic basis but adequate for the sake of solely describing valence electron correlation within the CAS framework. The invariance of CAS-CI wave functions with the unitary transformation of the active orbitals motivated us to use their canonical and localized forms, referred to as CMO and LMO, respectively, for the NQS calculations. The ground-state electronic state for ICG was modeled with CAS(4e,4o), CAS(6e,6o), and CAS(8e, 8o). The dimension of CI space for CAS(4e,4o), CAS(6e,6o), and CAS(8e, 8o) with (without) use of point group symmetry is 20 (36), 200 (400), and 2468 (4900), respectively. For CAS(4e,4o) and CAS(6e,6o), calculations were performed for RBM, BM2, and BM3 using LMO and CMO, while CAS(8e,8o) computations were only performed with RBM in the CMO basis due to computational cost. For all BM2 and BM3 calculations, only the deterministic approach was taken. Note that the implementation of BM2 and BM3 with the stochastic approach is straightforward. The effects of random sampling in MCMC on energy convergence were also investigated for CAS(4e,4o). After these preliminary investigations, application to reaction was investigated by calculating the potential energy curves of the ground state dinitrogen using CAS(6e,6o), which is modeled with a set of bonding and antibonding orbitals that play a major role in triple bond dissociation. Unless specified otherwise, ΔE denotes the difference in energy between the calculated NQS results and the CAS-CI energy of the same active space. A set of HOMO$-i$ and LUMO$+i$ with $i = 0, \ldots, n-1$ was used as the active orbirtals of the CAS($2n$e,$2n$o). All collected data can be referenced in the Supporting Information.

4.1 Energy Calculations of Indocyanine Green

The ICG (Figure 4) has a long π-conjugation as its main framework. For an initial test, we examined the ability of the RBM algorithm with CAS(4e,4o) to describe a part of ICG's π correlation. This test first focuses on the model's representability, forgoing check of statistical stability from the stochastic sampling. Figure 7 displays the weights of the configurations in the expansion of the CAS-CI wave functions determined using the CMO and LMO basis for the active orbitals. With the CMO basis, a single HF determinant accounts for 98% of the wave function in weight. This means the MR character is insignificant with this representation. This characteristic is largely changed with the use of LMO basis, turning into a representation of a state with a certain amount of MR character, in which five determinants are each weighted to the state with a ratio of over 10% and the rest of configurations also have some weight. The plots of the LMOs used for CAS(4e,4o) are provided in the Supporting Information.

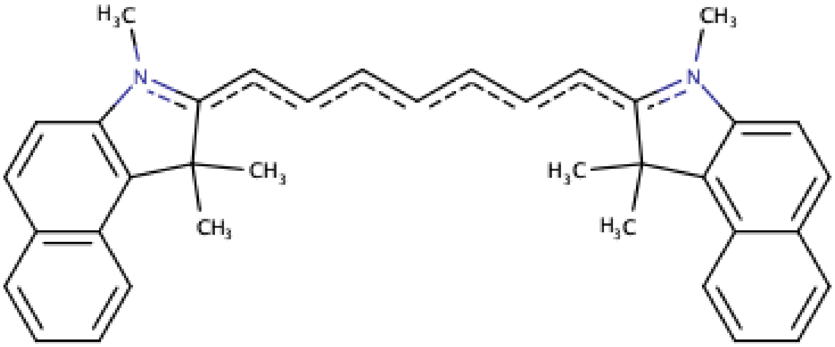


Figure 4: Modeled indocyanine green (ICG). The geometric information used in the calculations is given in the Supporting Infomration.

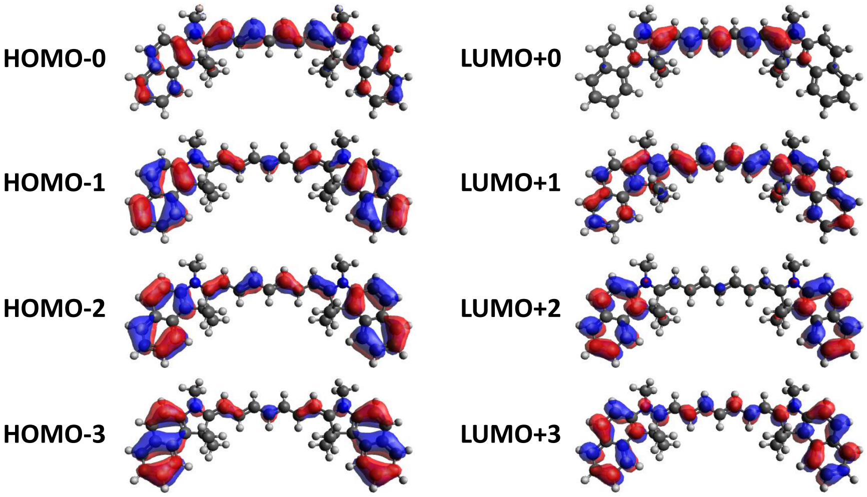


Figure 5: HOMO$-i$ and LUMO$+i$ of ICG for $i = 0, \ldots, 3$ used as active orbitals of CAS-CI calculations with CAS(4e,4o), CAS(6e,6o), and CAS(8e,8o).

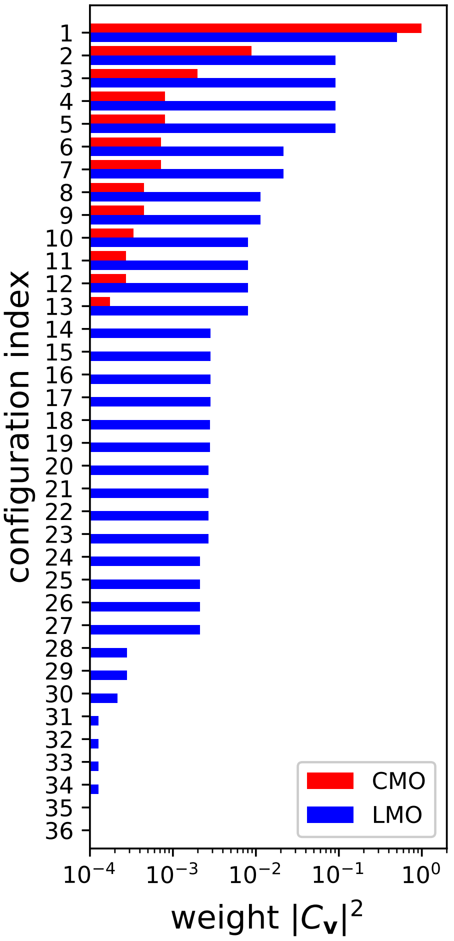


Figure 6: Weights of the configurations (i.e., determinants) $|C_{\mathbf{v}}|^2$ in the expansion of the CAS-CI(4e,4o) wave function for the ground state ICG represented using canonical molecular orbitals (CMO) and localized molecular orbitals (LMO) for the active space. Its energy is -1512.88042 E_{h} with either of CMO and LMO.

The NQS optimizations began with initializing their parameters with random values. We repeated the NQS training three times using different random seeds, set to 1000, 2000, and 3000, respectively. By training the RBM states with the deterministic approach, we observed the expected convergence at a large number of hidden nodes, reproducing the CAS-CI energy with accuracy of 10^{-5} E_{h}, as shown in Figure 7. The errors (ΔE) fell below 10^{-5} E_{h} for the RBM with the number of hidden nodes larger than 4 and 10 for CMO and LMO basis, respectively. The use of CMO is thus indicated to give a faster convergence, and this can be naturally understood because a highly concentrated structure of the CI distribution with

CMO, as shown in Figure 6, can be easily captured with a smaller number of hidden nodes. With a fewer hidden nodes, we observed that the training converged to different energies with more or less largely distributed errors depending on the random seeds chosen for the initialization of the RBM parameters. The forces of amplitude F_θ for these solutions were certainly lower than a convergence threshold; thus, they indicate the presence of local minima in the variational energy function. The errors associated with the seed dependence are not necessarily increasingly reduced with increasing number of hidden nodes; nonetheless, at a large number of hidden nodes, RBM is able to achieve highly accurate results with an inappreciable seed error. Table 1 compiles the lowest or best energies obtained with three training runs using different random seeds, along with ΔE and seed errors of the results. All converged energies are still lower bounded by the exact ground state energy as the algorithm is variational in nature. However, as mentioned above, there are some cases for the RBM model to fall into high-energy local minima and thus it may in a practical sense lose the satisfaction of variational characteristics with the number of hidden nodes. Figure 8 displays the values of the interaction matrices w_{ij} of $\boldsymbol{\theta}$, RBM's parameters of the amplitude $f(\mathbf{v};\boldsymbol{\theta})$ (eqn. (35)), determined using 2, 3, 4, 5, and 10 hidden nodes. The number of the parameters w_{ij}, $i = 1, \ldots, 2K$, and $j = 1, \ldots, m$, is $2Km$ where K and m refer to the number of active orbitals and hidden nodes, respectively. With increasing m, the numerical magnitude of elements generally decrease and the matrix appears to be increasingly more sparse.

Both BM2 and BM3 were able to generate energies much lower than RHF energy at convergence, as shown in Figure 7 and Table 1. In particular, BM3 was able to outperform BM2, giving a better energy by approximately one order magnitude, as expected due to its ability to learn higher-order correlation. In comparison to RBM, BM3 with CMO at convergence was able to achieve energies better than that of RBM with 3 hidden nodes. The error of LMO for BM3 (BM2) is reduced from 0.32 (1.05) mE_h to 0.01 (0.59) mE_h by using CMO, showing that CMO is an efficient basis for BM2 and BM3 similarly to RBM. It was demonstrated that the convergences with BM2 and BM3 to desirable solution were consistenly achieved regardless of the setting of the initial BM parameters. This is seemingly supported by a concave nature of the hidden node free energy functions of BM2 and BM3.

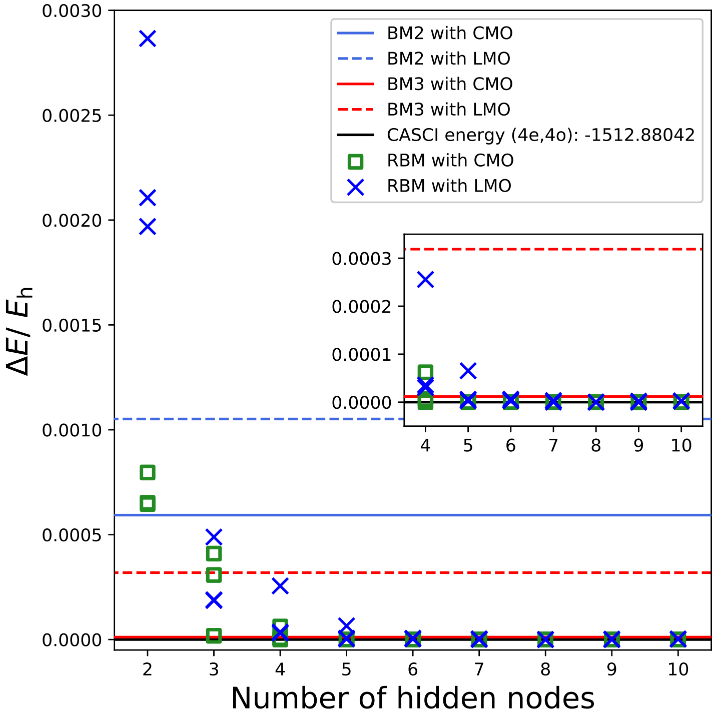


Figure 7: Errors of CAS-CI energies of ICG calculated using the RBM, BM2, BM3 models with CAS(4e,4o) in the CMO and LMO basis. The deterministic approach was used for training NQSs. ΔE is the difference in energy from the conventional CAS-CI result (-1512.88042 E_h). The energies are well below RHF energy (-1512.87513 E_h). The lowest energies are tabulated as a function of the models in Table 1.

Table 1: The CAS-CI energies of ICG determined using the deterministic approach using the RBM, BM2, BM3 models with CAS(4e,4o) in CMO and LMO basis, along with the conventional CAS-CI and HF results. The energies were determined using three different initial parameters that were randomly generated. The lowest energy from the training with the three initializations is shown as the best E in E_{h}, and the seed error is the largest difference of the three training runs from it. All numbers are shown in the unit of E_{h}.

	Hidden	CMO			LMO		
Model	nodes	Best E	ΔE	Seed error	Best E	ΔE	Seed error
RBM	1	-1512.87947	0.00095	0.00011	-1512.87518	0.00524	0.00167
RBM	2	-1512.87977	0.00065	0.00015	-1512.87845	0.00197	0.00090
RBM	3	-1512.88040	0.00002	0.00039	-1512.88023	0.00019	0.00030
RBM	4	-1512.88042	0.00000	0.00006	-1512.88039	0.00003	0.00023
RBM	5	-1512.88042	0.00000	0.00000	-1512.88041	0.00000	0.00006
RBM	10	-1512.88042	0.00000	0.00000	-1512.88042	0.00000	0.00000
RBM	20	-1512.88042	0.00000	0.00000	-1512.88042	0.00000	0.00000
RBM	40	-1512.88042	0.00000	0.00000	-1512.88042	0.00000	0.00000
BM2	0	-1512.87983	0.00059	0.00000	-1512.87937	0.00105	0.00000
BM3	0	-1512.88041	0.00001	0.00000	-1512.88010	0.00032	0.00000
CASCI		-1512.88042			-1512.88042		
HF		-1512.87513	0.00529				

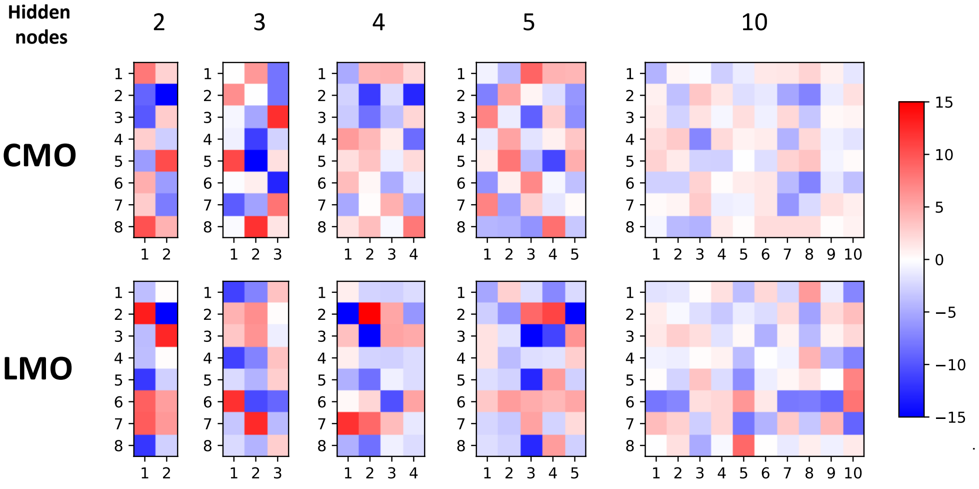


Figure 8: Interaction matrices w_{ij} of $\boldsymbol{\theta}$, the parameters of RBM's parameters of the amplitude $f(\mathbf{v};\boldsymbol{\theta})$ (eqn. (35)), determined using 2, 3, 4, 5, and 10 hidden nodes for IGC with CAS(4e,4o) in CMO and LMO basis.

Next, let us turn to the results obtained by the stochastic approach with RBM (Table 2). The MCMC integrations were carried out using 6×10^7 samples. The converged energies were obtained by taking the average of energy from the last 40 updates, which were also used for evaluating the standard deviations shown in the table. It was confirmed that our MCMC based implementation is overall capable of reproducing the corresponding results obtained with the deterministic approach to the accuracy close to 10^{-5} E_h. A marked seed-dependence error was observed to be 2.5 mE_h by RBM with 10 hidden nodes using CMO basis, while the corresponding deterministic approach did not suffer from this order of error. The sampling error in the MCMC integration seems to be properly managed because of use of a sufficient number of samples; however, its statistical noise may sensitively affect the optimization process and lead the solution to other local minima.

Table 2: The CAS-CI energies of ICG determined using the stochastic approach using the RBM model with CAS(4e,4o) in CMO and LMO basis. All numbers are shown in the unit of E_h.

Model	Hidden nodes	MO type	Best E	ΔE	Standard deviation	Seed error
RBM	5	CMO	-1512.88041	0.00001	5×10^{-7}	0.00016
RBM	5	LMO	-1512.88040	0.00002	5×10^{-7}	0.00036
RBM	10	CMO	-1512.88042	0.00000	4×10^{-8}	0.00248
RBM	10	LMO	-1512.88042	0.00000	1×10^{-7}	0.00000

To probe the reliability of the stochastic approach, converged parameters $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ were extracted from the deterministic calculations and fed into as initial parameters for a set of calculations performed with the stochastic approach of same number of hidden nodes. In all cases, the first iteration reproduced the deterministic energy with convergence, hence confirming the validity of also the stochastic approach. In Figure 9, the effect of stochastic sampling on energy is graphed. This is calculated with the same method described above for calculating stochastic approach in RBM. For five separate trials, decreasing ΔE calculated from the converged energy in deterministic approach demonstrates the ability for increasing number of sampling to accurately reproduce distribution evaluated with the deterministic approach, hence the small difference in energies.

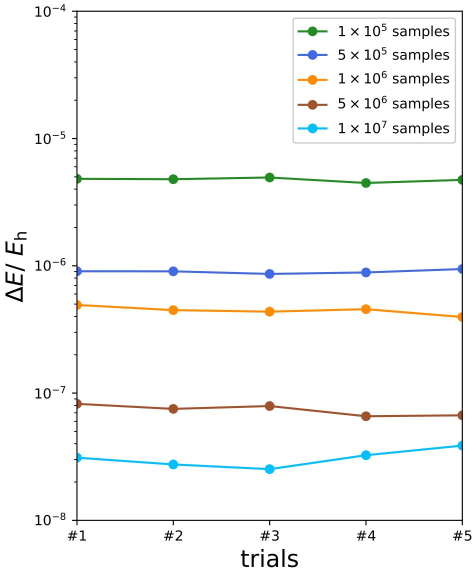


Figure 9: Dependence of configurational energy improvements on number of samples sampled per stochastic update.

After confirming viability in CAS(4e,4o), the active space was expanded to CAS(6e,6o) and CAS(8e,8o) for further investigation. The deterministic evaluation was only performed in order to mainly assess the representability of the models. With these sizes of CAS, deterministic evaluation is still possible and indeed able to perform training even faster than the stochastic approach, since the computational overhead of the sampling step is too large in the MCMC based optimization. Only when arriving at the exponential wall, as alluded to previously, is deterministic evaluation impossible, and sampling should then be preferred. Despite multi-node and multi-thread parallelization incorporated into our implementation, the task of massive data analysis is considered not to run the fastest on CPU architectures but to be most feasibly handled in the ML computation by GPGPU-based high throughput processing. We will return to this point in the conclusion. In CAS(6e,6o), the BM3 results with LMO were excluded due to extra complexity associated with lose of symmetry from the localization.

As shown in Figure 10, the lowest-energy RBM predictions in CAS(6e,6o) showed a desired converge to the CAS-CI energy with increasing number of hidden nodes, while several high-lying local minima were also obtained. With 40 hidden nodes, ΔE of the best energies with CMO and LMO basis are both less than 10^{-5} E_{h}. Just as for CAS(4e,4o), CMO basis better performed with a given number of hidden nodes than LMO basis. With CMO, RBM using 10 hidden nodes is still capable of delivering an accuracy of 10^{-4} E_{h}. In the LMO basis, the overestimated energies due to local minima with 20 hidden nodes lie even above those resulting from 10 hidden nodes. In the tests, the results with BM3 in CMO basis and BM2 in LMO basis were unaffected by the choice of the random seed for randomly initialized parameters, converging to consistent energies. This seems to effectively corroborate concave nature of these models free from hidden nodes. The BM2 results with CMO basis somewhat showed seed dependence with an error of 1.6 $\times 10^{-4}$ E_{h} in energy. With the CMO basis, the BM2 and BM3 were well trained, resulting in small ΔE, 11×10^{-4} E_{h} and 4×10^{-4} E_{h}, respectively, despite their linear model of logistic regression. This favored performance is related to the fact that the CI distribution with CMO is predominantly weighted to HF configuration as also seen in CAS(4e,4o). Also noted is the BM2 and BM3 energies shows variational convergence with the order of BM. Even though per iteration cost is much higher due to increased complexity for calculating the network energy function, at roughly 140 times the time cost of BM2 with the current implemenation, the update iteration required for BM3 to reach the reported energy is just a fraction of that of BM2, at 500 updates. Overall, RBM in CAS(6e,6o) gave promising results with convergent CASCI energy when using an adequate number of hidden nodes.

Finally, the results of RBM calculations performed in CAS(8e,8o) are graphed in Figure 11. With 50 or more hidden nodes, ΔE dropped below 2 $\times 10^{-4}$ E_{h} at convergence. However, the improvement gained by increasing hidden nodes over 50 was minor, and ΔE remains no better than 1.7 $\times 10^{-4}$ E_{h} with 100 hidden nodes, which is the largest RBM tested. It appears difficult to further reduce the error seemingly because of limitation of the model or some technical difficulties in training the model. A further increase in the number of hidden nodes should increase function flexibility to more accurately learn the CI distribution, but at the same time, increase difficulties in achieving sufficient optimization.

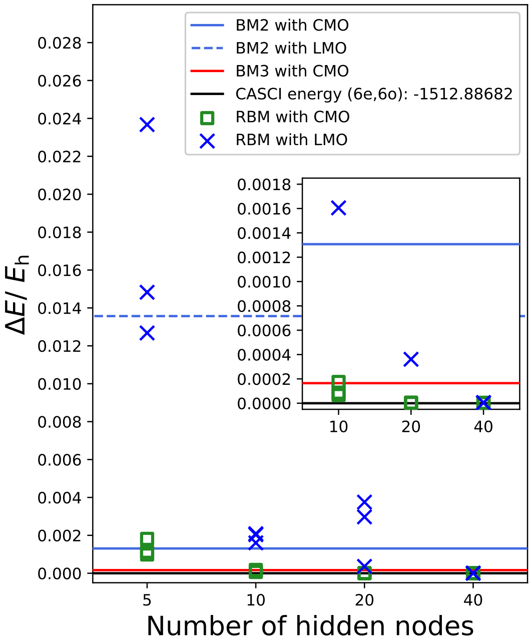


Figure 10: Calculation of ICG with RBM/BM2/BM3 algorithms in CAS(6,6). Errors of BM2 and BM3 in energy are 1.2×10^{-3} and 4.1×10^{-4} E_h, respectively, in CMO basis, while errors for RBM at 40 hidden nodes are belows 10^{-5} E_h.

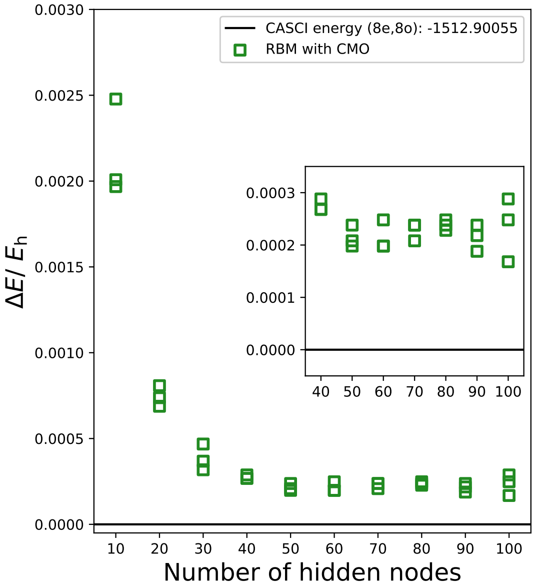


Figure 11: Calculation of ICG with RBM in CAS(8e,8o). ΔE for 100 hidden nodes is 1.7 $\times 10^{-4}$ E_h.

4.2 Potential Energy Curves of Dinitrogen

In this section the multireference characteristics of RBM, BM2, BM3 were studied and compared to energies of the conventional CAS-CI in CAS(6e,6o) by calculating the potential energy curve of ground state dinitrogen ($X^1 \sum_g^+$; $(1\sigma_g^2 1\sigma_u^2 2\sigma_g^2 2\sigma_u^2 1\pi_u^4 2\sigma_g^2)$). The canonical form of active orbitals were only tested; their plots are provided in the Supporting Information. Static correlation is known to play an important role in achieving balanced accuracy across the potential energy curve. A bond length of 1.09 Å is chosen as near equilibrium of N_2. Once again, the stochastic approach is only implemented in the RBM, but deterministic approach was carried out in RBM, BM2, and BM3. In this test, we used a single random seed for randomly initializing parameters of NQSs. The CI dimension in determinant basis is 56 under $D_{2\text{h}}$ point group symmetry. Five hidden nodes was used for training RBM using deterministic and stochastic approaches. In MCMC, we used 1.1 $\times 10^6$ samples for the stochastic evaluation of energies and gradients.

Calculation results tabulated in Table 3 demonstrated that all energies of the RBM model converge to CAS-CI energies with errors less than 0.4 mE_h across the potential energy curve, even at the stretched structure where the RHF prediction typically breaks down. BM2 and BM3 showed results similar to RBM, and their ΔE are on average similar in magnitudes. The dissociated nitrogen atoms can be characterized to a qualitatively correct manner by the CAS-CI wave function, which is a mixture of the ground state RHF configuration and its excited configurations associated with occupations in anti-bonding orbitals $1\pi_g$ and $3\sigma_u$. By confirming that calculations in RBM/BM2/BM3 algorithms contains small ΔE, the validity of these algorithms as alternative MR method is confirmed. In fact, the results in Table 3 from both the sampling and deterministic cases demonstrated their ability as consistent estimators of the CAS-CI results. The multireference characteristic is further substantiated by the configuration coefficients versus iterations graphed in Figure 13. It can be seen that for r = 1.09 Å, one single configuration dominated contribution in the left subfigure as the occupations and associated orbitals are optimized in the preceding HF calculations, but in the right subfigure for $r = 2.18\,\text{Å}$, significant number of different configurations all contributed to the description of the wave function.

The favorable convergence of RBM as well as BM2 and BM3 to CAS-CI energies is seemingly the result of the reduced molecule dimension and size. Meanwhile for the stochastic evaluation case in RBM, performance are expectantly worse in the shorter r ranges, most likely due to statistical errors arising from the MCMC integration. For larger r values, however, the contribution of statistical error to ΔE seems to become less significant, as observed in comparing the stochastic and deterministic approaches of RBM. Although the energy surface during the dissociation process suffers from the inherent instability where the single-determinant RHF theory fails to describe the MR character, it can be properly accounted for by the flexibility built into the BM architectures. Interestingly, compared to results obtained in ICG, BM3 was able to outperform RBM on larger r values. This may be related to the fact that the energy function of BM3 remained relatively concave, while the RBM has higher-energy local minima in the energy function. Further investigation with a more robust optimizer can contribute to a better result in the RBM. To further demonstrate the potential energy curve of nitrogen dissociation, calculations were taken in finer r steps to give calculation results in Figure 12, tabulated in the Supplementary Information. With finer r steps, the classical graph of potential energy during dinitrogen dissociation is obvious, and energies clearly converged to CAS-CI results. Crucially, reproduction of the potential energy curve reiterates the viability of NQS and our algorithms as a MR molecular wave function solver with a high potential. Additionally, configuration coefficients over iterations are given in Figures 13 and 14 for the deterministic and stochastic approaches, respectively. From this figure, the statistical instability resulting from the MCMC sampling necessary for larger system size calculations is observed.

Table 3: Calculation results for dinitrogen with the bond length r. CAS(6e,6o) with canonical Hartree-Fock orbitals was used for active space of CAS-CI, RBM, BM2, and BM3 calculations.

	Energy/ E_h			ΔE / E_h		
r / Å	CASCI	RHF	RBM (MCMC)	RBM (det)	BM2 (det)	BM3 (det)
0.545	-102.339743	0.031725	< 1e-6	< 1e-6	4e-6	< 1e-6
1.09	-107.617344	0.123813	1.12e-4	< 1e-6	1.25e-4	1e-6
1.64	-107.501623	0.347167	1.95e-4	< 1e-6	1.86e-4	< 1e-6
2.18	-107.432667	0.670026	1.92e-4	3.87e-4	2.748e-3	5e-6
2.73	-107.435787	0.891737	3.83e-4	1.88e-4	3.00e-4	4.56e-5
3.27	-107.437204	1.000766	8.8e-5	5.7e-5	8.8e-5	1.61e-5

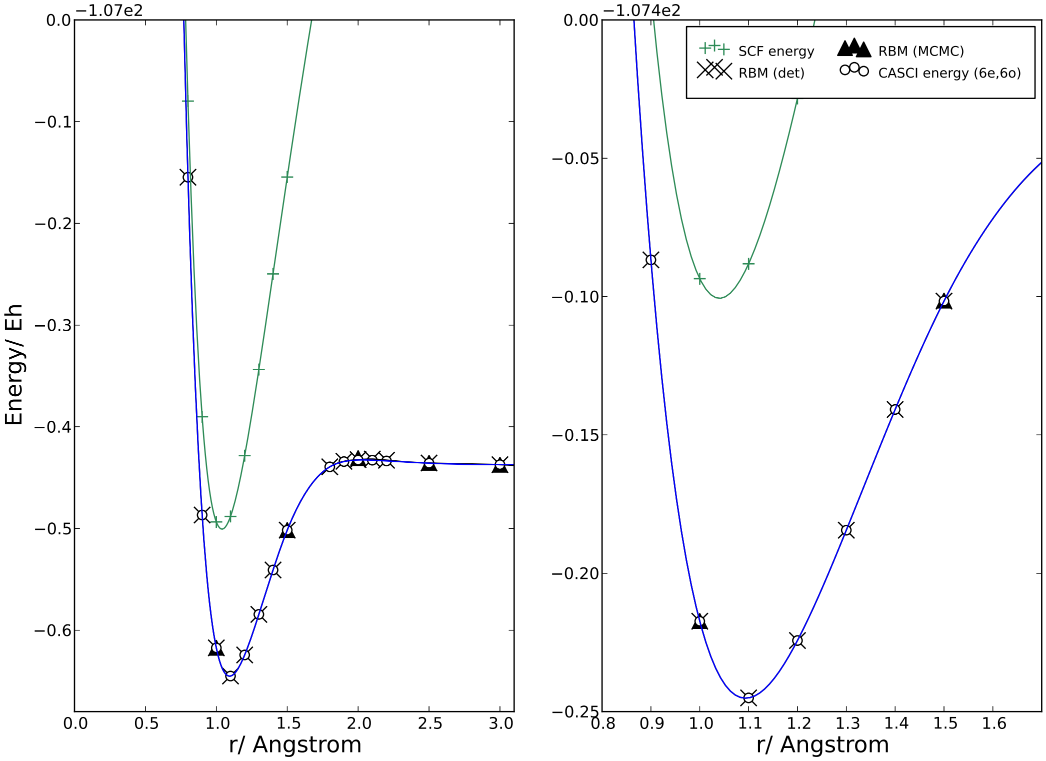


Figure 12: Dissociation of dinitrogen with RBM algorithm in CAS(6e,6o) for deterministic and MCMC approaches. Bond lengths are taken in finer steps for the deterministic case. Right subgraph is zoomed for a smaller energy range. Agreement between the stochastic and deterministic approaches illustrates viability of sampling as evaluation method for large active spaces, as described for Figure 9.

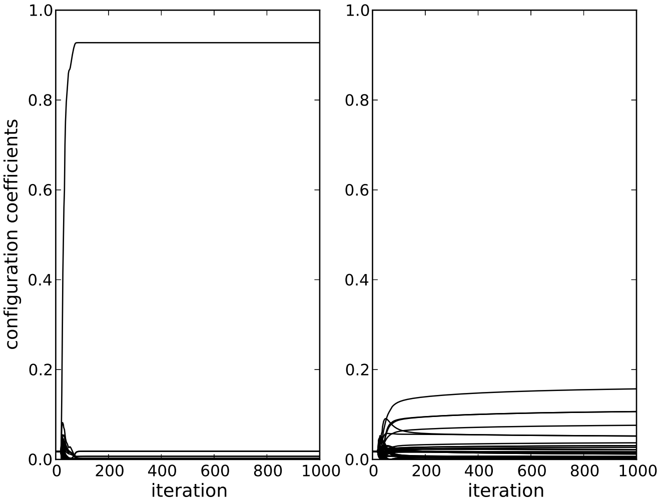


Figure 13: Configuration coefficients versus iterations for dinitrogen in the deterministic approach. Left: $r = 1.09\,\text{Å}$; right: $r = 2.18\,\text{Å}$. The dramatically increased importance of multiple configurations at larger internuclear distances is shown.

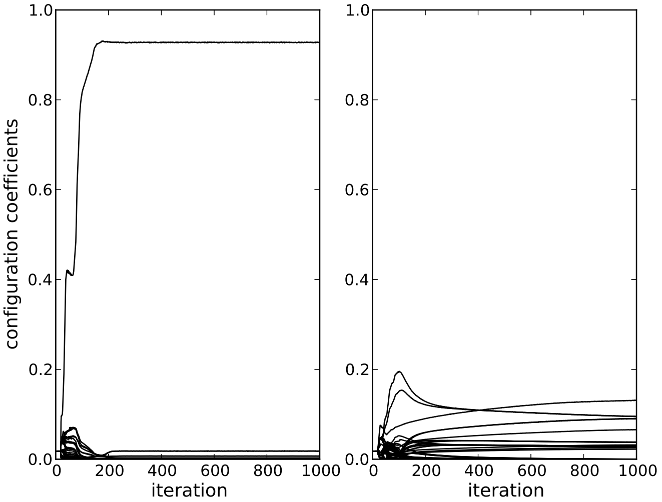


Figure 14: Configuration coefficients versus iterations for dinitrogen in the stochastic approach. Left: $r = 1.09\,\text{Å}$; right: $r = 2.18\,\text{Å}$. In comparison to Figure 13, statistical instability in the form of jitters is observed.

5 Conclusion

In conclusion, we have successfully presented a pilot implementation of an alternative CAS-CI method utilizing the neural-network quantum state ansatz. Using the ML models, the CI coefficients are parameterized as a function of their occupancies that act as descriptors. Specifically, restricted Boltzmann machines (RBM), second-order Boltzmann machines (BM2), and third-order Boltzmann machines (BM3) were used for the neural-network architectures. The BMs as a molecular wave function are trained or optimized with the energy minimization, in which neither reference data nor prior knowledge of the wave function are used in a manner relevant to reinforcement learning. In theory, the advantages of this approach over tensor network methods such as DMRG include extensibility in dimension as its parameterization does not assume any nonuniform connectivity of correlation network. However, this remains to be proven in practice with calculations in large active spaces. In this work, focusing on representability test of valence electron correlation, we performed calculations on multireference CAS-CI wave functions with CAS(4e,4o), CAS(6e,6o), and CAS(8e,8o) of indocyanine green (ICG) with localized orbitals as well as CAS(6e,6o) of dinitrogen dissociation. The energies have shown to converge to CAS-CI energy in most cases, with the RBM system reproducing the CAS-CI results in CAS(6e,6o) to ΔE of 10^{-5} at 40 (20) hidden nodes for ICG with LMO (CMO) basis. In reproducing the potential energy curve of dinitrogen, the representability of the MR feature for chemical processes such as dissociation was demonstrated. The ability for our method to account for MR character of wave functions is a consequence of its inherent foundation on the NQS model. We confirmed that the stochastic integration of energy and gradients can be incorporated into the implementation of the NQS models in a conceptually feasible manner.

The RBM was introduced as an underlying network architecture of NQS in the work of Carleo and Troyer, but in this study we have shed a new light on the comparable capabilities of the BM2 and BM3 architectures. Unlike RBM, they are hidden-node-free in our modeling, thus giving concave log-likelihood functions. The RBM results appeared to be susceptible to choice of starting parameters due to non-concave nature of its energy landscapes, while using the same optimizer, BM2 and BM3 convergences were shown to be relatively robust. Improvement on the accuracy of the RBM model is made by increasing the number of hidden nodes, and a recent attempt to extend it using the deep learning architecture should be promising.[15,16] Having more hidden nodes complements numerical flexibility of these models; however, this may generally cause a technical difficulty in finding a unique global minimum in a well-defined manner. In this study, we demonstrated that increase in the order of the BM model instead of the hidden nodes is an alternative approach to systematically improve the BM based model with a certain numerical stability. In our tests, the BM3 produced a satisfactory accuracy throughout the dinitrogen dissociation curve within an error of 5 $\times 10^{-5}$ E_{h}, surpassing the chemical accuracy. The electron correction effects are well formulated in QC with a hierarchy of many-body treatment based on the levels of the second quantized excitation operators, which is realized in a well-known series of the coupled-cluster theory, CCSD, CCSDT, CCSDTQ, and so forth. This systematic structure has an analogy with the fact that BM-k in principle converges to a full CI expansion with increasing k, as we observed that BM3 in general outperforms BM2. In the present form, the effects brought by the higher order connectivity in BM3 beyond the bipartite graph of BM2 cannot be directly interpreted using the conventional many-body formalism. Nonetheless, it may be interesting to illuminate any relation between high-order BMs and the conventional physical picture of many-body hierarchy of electron correlation. The denisty-based Jastrow formalism may have a ceratin relation along this line.[48,49]

As a proof-of-concept work, our implementation has demonstrated promising results, but still remains to be further improved for calculations in larger active space, and for network training in a more robust environment. To achieve ability for calculation in larger active space, transition to network training with general-purpose computing on graphical processing units (GPGPU) should be a promising direction of the future development. The advent of artificial neural-network application in nearly every possible way is primarily a result of its training with GPGPU, which is considerably more advantageous over training on traditional central processing units (CPUs). Due to the inherently graphical structure of artificial neural networks, GPUs, which are designed from hardware to perform graphical tasks, are incredibly suited for high throughput training of artificial neural networks. To implement our system for training in GPGPU would lead to a significant performance boost that would allow it to handle calculations for larger chemical systems, and active spaces. Also due to the inherent finesse associated with artificial neural networks, more fine tuning would be necessary to produce a robust environment that allows for optimal results. Once the program is able to calculate with larger active space, comparison with other state of the art methods will allow us to further improve and understand the extent to which the theory falls in line with reality.

Acknowledgement

T.Y. is supported by JST, PRESTO Grant Number 17937609. M.S. is supported by JSPS KAKENHI Grant No. JP16H02870 and JST, PRESTO Grant No. JPMJPR1855. K.T. is supported by NEDO P15009, SIP (Technologies for Smart Bio-industry and Agriculture), JST PRISM JPMJCR18Y3, JST CREST JPMJCR1502 and JST ERATO.

Supporting Information Available

The Supporting Information is available free of charge on the ACS Publications website.

- achemso-demo-sup.pdf:

References

(1) Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Cent. Sci.* **2017**, *3*, 434–443.

(2) Schneider, G. Automating drug discovery. *Nat. Rev. Drug Discov.* **2017**, *17*, 97–113.

(3) Segler, M. H. S.; Preuss, M.; Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **2018**, *555*, 604–610.

(4) Mackerell, A. Machine Learning Force Field Parameters from Ab Initio Data. *J. Chem. Theory Comput.* **2017**, *13*, 4492–4503.

(5) Schutt, O.; Vandevondele, J. Machine Learning Adaptive Basis Sets for Efficient Large Scale Density Functional Theory Simulation. *J. Chem. Theory Comput.* **2018**, *14*, 4168–4175.

(6) Cai, Z.; Liu, J. Approximating quantum many-body wave functions using artificial neural networks. *Phys. Rev. B* **2018**, *97*, 035116.

(7) Saito, H. Solving the Bose-Hubbard Model with Machine Learning. *J. Phys. Soc. Jpn.* **2017**, *86*, 093001.

(8) Torlai, G.; Mazzola, G.; Carrasquilla, J.; Troyer, M.; Melko, R.; Carleo, G. Many-body quantum state tomography with neural networks. *Nat. Phys.* **2018**, *14*, 447–450.

(9) Amin, M. H.; Andriyash, E.; Rolfe, J.; Kulchytskyy, B.; Melko, R. Quantum Boltzmann Machine. *Phys. Rev. X* **2018**, *8*, 021050.

(10) Lei, W. Discovering phase transitions with unsupervised learning. *Phys. Rev. B* **2016**, *94*, 195105.

(11) Carleo, G.; Troyer, M. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science* **2017**, *355*, 602–606.

(12) Nomura, Y.; Darmawan, A. S.; Yamaji, Y.; Imada, M. Restricted Boltzmann machine learning for solving strongly correlated quantum systems. *Phys. Rev. B* **2017**, *96*, 205152.

(13) Glasser, I.; Pancotti, N.; August, M.; Rodriguez, I.; Cirac, J. Neural-Network Quantum States, String-Bond States, and Chiral Topological States. *Phys. Rev. X* **2018**, *8*, 011006.

(14) Melko, R. G.; Carleo, G.; Carrasquilla, J.; Cirac, J. I. Restricted Boltzmann machines in quantum physics. *Nat. Phys.* **2019**, *15*, 887–892.

(15) Gao, X.; Duan, L.-M. Efficient representation of quantum many-body states with deep neural networks. *Nat. Commun.* **2017**, *8*, 662.

(16) Carleo, G.; Nomura, Y.; Imada, M. Constructing exact representations of quantum many-body systems with deep neural networks. *Nat. Commun.* **2018**, *9*, 5322.

(17) Roos, B. O.; Lindh, R.; Malmqvist, P. A.; Veryazov, V.; Widmark, P. *Multiconfigurational Quantum Chemistry*, 1st ed.; John Wiley and Sons, Inc.: New Jersey, 2016; pp 93–130.

(18) White, S. R. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.* **1992**, *69*, 2863–2866.

(19) White, S. R. Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B* **1993**, *48*, 10345–10356.

(20) Chan, G. K.-L.; Sharma, S. The density matrix renormalization group in quantum chemistry. *Annu. Rev. Phys. Chem.* **2011**, *62*, 465–481.

(21) Schollwöck, U. The density-matrix renormalization group in the age of matrix product states. *Ann. Phys.* **2011**, *326*, 96–192.

(22) Chan, G. K.-L.; Keselman, A.; Nakatani, N.; Li, Z.; White, S. R. Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms. *J. Chem. Phys.* **2016**, *145*, 014102.

(23) Li, Z.; Chan, G. K.-L. Spin-Projected Matrix Product States: Versatile Tool for Strongly Correlated Systems. *J. Chem. Theory Comput.* **2017**, *13*, 2681–2695.

(24) Gunst, K.; Verstraete, F.; Wouters, S.; Legeza, O.; Neck, D. V. T3NS: Three-Legged Tree Tensor Network States. *J. Chem. Theory Comput.* **2018**, *14*, 2026–2033.

(25) Nakatani, N.; Chan, G. K.-L. Efficient tree tensor network states (TTNS) for quantum chemistry: Generalizations of the density matrix renormalization group algorithm. *J. Chem. Phys.* **2013**, *138*, 134113.

(26) Murg, V.; Verstraete, F.; Legeza, Ö.; Noack, R. M. Simulating strongly correlated quantum systems with tree tensor networks. *Phys. Rev. B* **2010**, *82*, 205105.

(27) Marti, K. H.; Bauer, B.; Reiher, M.; Troyer, M.; Verstraete, F. Complete-graph tensor network states: a new fermionic wave function ansatz for molecules. *New J. Phys.* **2010**, *12*, 103008.

(28) Verstraete, F.; Murg, V.; Cirac, J. I. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Adv. Phys.* **2008**, *57*, 143–224.

(29) Kovyrshin, A.; Reiher, M. Self-adaptive tensor network states with multi-site correlators. *J. Chem. Phys.* **2017**, *147*, 214111.

(30) Booth, G. H.; Thom, A. J.; Alavi, A. Fermion Monte Carlo without fixed nodes: A game of life, death, and annihilation in Slater determinant space. *J. Chem. Phys.* **2009**, *131*, 054106.

(31) Ten-no, S. Stochastic determination of effective Hamiltonian for the full configuration interaction solution of quasi-degenerate electronic states. *J. Chem. Phys.* **2013**, *138*, 164126.

(32) Neese, F. A spectroscopy oriented configuration interaction procedure. *J. Chem. Phys.* **2003**, *119*, 9428–9443.

(33) Schriber, J. B.; Evangelista, F. A. Communication: An adaptive configuration interaction approach for strongly correlated electrons with tunable accuracy. *J. Chem. Phys.* **2016**, *144*, 161106.

(34) Tubman, N. M.; Lee, J.; Takeshita, T. Y.; Head-Gordon, M.; Whaley, K. B. A deterministic alternative to the full configuration interaction quantum Monte Carlo method. *J. Chem. Phys.* **2016**, *145*, 044112.

(35) Sharma, S.; Holmes, A. A.; Jeanmairet, G.; Alavi, A.; Umrigar, C. J. Semistochastic Heat-bath Configuration Interaction method: selected configuration interaction with semistochastic perturbation theory. *J. Chem. Theory Comput.* **2017**, *13*, 1595–1604.

(36) Choo, K.; Mezzacapo, A.; Carleo, G. Fermionic neural-network states for ab-initio electronic structure. 2019, arXiv:1909.12852. arXiv.org ePrint archive. https://arxiv.org/abs/1909.12852 (accessed Nov 13, 2019).

(37) Coe, J. P. Machine Learning Configuration Interaction. *J. Chem. Theory Comput.* **2018**, *14*, 5739–5749.

(38) Townsend, J.; Vogiatzis, K. D. Data-Driven Acceleration of the Coupled-Cluster Singles and Doubles Iterative Solver. *J. Phys. Chem. Lett.* **2019**, *10*, 4129–4135.

(39) Coe, J. P. Machine Learning Configuration Interaction for ab Initio Potential Energy Curves. *J. Chem. Theory Comput.* **2019**, *15*, 6179–6189.

(40) Margraf, J. T.; Reuter, K. Making the Coupled Cluster Correlation Energy Machine-Learnable. *J. Phys. Chem. A* **2018**, *122*, 6343–6348.

(41) Nudejima, T.; Ikabata, Y.; Seino, J.; Yoshikawa, T.; Nakai, H. Machine-learned electron

correlation model based on correlation energy density at complete basis set limit. *J. Chem. Phys.* **2019**, *151*, 024104.

(42) Montufar, G. Restricted Boltzmann Machines: Introduction and Review. *In: Ay N., Gibilisco P., Matus F. (eds) Information Geometry and Its Applications. IGAIA IV 2016. Springer Proceedings in Mathematics & Statistics* **2018**, *252*, 75–115.

(43) Luo, S.; Sugiyama, M. Bias-variance trade-off in hierarchical probabilistic models using higher-order feature interactions. Proceedings of the AAAI Conference on Artificial Intelligence. 2019; pp 4488–4495.

(44) Marsland, S. *Machine Learning: An Algorithmic Perspective*; CRC Press, 2015.

(45) Hinton, G. E.; Salakhutdinov, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507.

(46) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press, 2016.

(47) Sejnowski, T. J. Higher-order Boltzmann machines. *AIP Conf. Proc.* **1986**, *151*, 398–403.

(48) Thibaut, J.; Roscilde, T.; Mezzacapo, F. Long-range entangled-plaquette states for critical and frustrated quantum systems on a lattice. *Phys. Rev. B* **2019**, *100*, 155148.

(49) Sorella, S. Wave function optimization in the variational Monte Carlo method. *Phys. Rev. B* **2005**, *71*, 241103.

(50) Becca, F.; Sorella, S. *Quantum Monte Carlo Approaches For Correlated Systems*; Cambridge University Press, 2017.

(51) Kolmogorov, A. N. On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. *Dokl. Akad. Nauk SSSR* **1961**, *108*, 179–182.

(52) Le Roux, N.; Bengio, Y. Representational power of restricted Boltzmann machines and deep belief networks. *Neural Comput.* **2008**, *20*, 1631–1649.

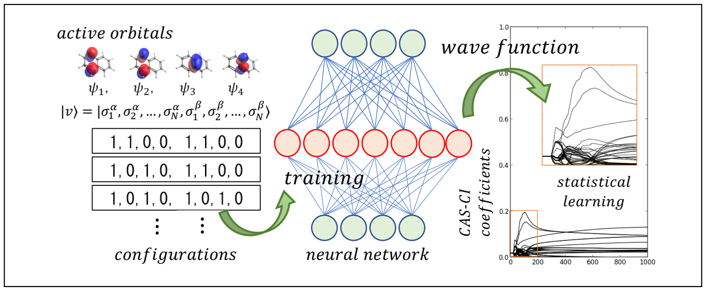
(53) Poulin, D.; Qarry, A.; Somma, R.; Verstraete, F. Quantum Simulation of Time-Dependent Hamiltonians and the Convenient Illusion of Hilbert Space. *Phys. Rev. Lett.* **2011**, *106*, 170501.

(54) Helgaker, T.; Jørgensen, P.; Olsen, J. *Molecular Electronic Structure Theory*; John Wiley & Sons, LTD: Chichester, 2000.

(55) Yanai, T.; Kurashige, Y.; Mizukami, W.; Chalupský, J.; Lan, T. N.; Saitow, M. Density matrix renormalization group for ab initio Calculations and associated dynamic correlation methods: A review of theory and applications. *Int. J. Quantum Chem.* **2015**, *115*, 283–299.

(56) Neuscamman, E.; Umrigar, C. J.; Chan, G. K.-L. Optimizing large parameter sets in variational quantum Monte Carlo. *Phys. Rev. B* **2012**, *85*, 045103.

Graphical TOC Entry



Artificial neural networks (ANNs) based on the Boltzmann machine (BM) architectures used as an encoder of *ab initio* molecular many-electron wave functions.