Reinforcement Learning Configuration Interaction

Joshua J. Goings,[†] Hang Hu,[†] Chao Yang,[*,‡] and Xiaosong Li[*,†]

[†]*Department of Chemistry, University of Washington, Seattle, WA, 98195*

[‡]*Computational Research Division, Lawrence Berkeley National Laboratory, CA 94720*

E-mail: cyang@lbl.gov; xsli@uw.edu

Abstract

Selected configuration interaction (sCI) methods exploit the sparsity of the full configuration interaction (FCI) wave function, yielding significant computational savings and wave function compression without sacrificing the accuracy. Despite recent advances in sCI methods, the selection of important determinants remains an open problem. We explore the possibility of utilizing reinforcement learning approaches to solve the sCI problem. By mapping the configuration interaction problem onto a sequential decision-making process, the agent learns on-the-fly which determinants to include and which to ignore, yielding a compressed wave function at near-FCI accuracy. This method, which we call reinforcement learned configuration interaction (RLCI), adds another weapon to the sCI arsenal and highlights how reinforcement learning approaches can potentially help solve challenging problems in electronic structure theory.

1 Introduction

Methods that allow for the efficient simulation of strongly correlated molecules and materials are critical to energy, quantum information, and materials applications. Strong correlation arises when the system cannot be qualitatively described by a single Slater determinant,

such as in bond dissociation and in the description of many transition metal complexes. Because of the breakdown of the single determinant approximation, methods like Kohn-Sham density functional theory or perturbative techniques such as Møller-Plesset perturbation theory cannot be used. Instead, the reference wave function must be modified to include multiple determinants from the outset.

In principle, the full configuration interaction (FCI) method[1] solves the electronic problem exactly for a given basis, but it scales exponentially, rendering FCI impractical for all but the smallest of problems. However, it is well-known that the solution to the FCI wave function is generally sparse. That is, many determinants do not significantly contribute to the overall wave function; these determinants are sometimes referred to as computational "deadwood."[2] From a practical standpoint, this means that significant compression of the wave function is possible while retaining near-FCI accuracy. Approaches for wave function compression based on the singular value decomposition[3] and compressive sensing,[4] for example, have been explored. How to best exploit this sparsity is an open question,[5,6] but it is a question with important ramifications.

Exploring wave function compression is not just important for classical electronic structure calculations, where the sparsity may be exploited for significant computational savings. As pointed out by Stair, *et al.*,[5] highly compressed wave functions are desirable for chemical applications of quantum computing on near-term devices. The reason is that many quantum algorithms, such as variational quantum eigensolvers, rely on wave function parameterization. Because the goal of these quantum algorithms is to yield an advantage over classical simulation, one way to characterize the advantage of these methods is to compare against the classical efficiency, in order determine if there is indeed a quantum advantage over classical approaches.

In this light, selected configuration interaction (sCI) methods have come to the fore, driven by several recent important methodological advances. The goal of selected CI methods is to find sparse approximate solutions to the full configuration interaction (FCI) problem,

which is generally done in an iterative manner where the FCI parameter space is efficiently searched, ranked, and top contributors included in the process of obtaining a sparse solution to the CI problem. Most selected CI methods begin by selecting an initial variational space (such as a Hartree-Fock, HF, reference or complete active space self-consistent field, CASSCF, reference) and obtaining the variational ground state wave function within this space. After the initial wave function is obtained, search algorithms explore the space outside the current set of determinants to select important additional determinants to be included in the variational space. The estimated significance of a determinant may be based off perturbative or energetic heuristic considerations. Furthermore, the space may be optionally pruned to eliminate determinants deemed no longer significant. After the searching and pruning, the ground state of the Hamiltonian in this new variational space is obtained. The process is then repeated until some convergence criteria is reached.

Although all flavors of sCI follow this rough algorithm, they differ on the specifics of how to achieve the goal: for example, there are deterministic,[2,4,7–20] stochastic,[21–27] and semi-stochastic[28–31] variations of sCI methods. The crucial ingredient is the searching/ranking of determinants and the pruning of determinants no longer deemed significant. Although several heuristics have been developed to achieve these goals, one open question is whether or not these heuristics can be learned and improved upon via a machine learning (ML) approach.

Machine learning methods have had a significant impact across many domains, not least in chemistry.[32,33] New methods for the analysis and simulation of molecules and materials have emerged based on developments in machine learning. For example, machine learning has been used to design new materials,[34] to predict protein structure,[35] ground[36] and excited[37] state molecular properties, and even aid in the interpretation of complex molecular dynamics trajectories.[38,39] For the selected CI problem, machine learning has been used to predict important configurations using supervised learning on data generated on-the-fly.[40,41] This enables more accurate potential energy surfaces with fewer iterations as compared to a

stochastic sCI approach. Most applications of ML for quantum chemistry are in either the domain of supervised or unsupervised learning and require large amounts of data. The quality of the application depends on the amount and type of data, and even then the training may not generalize to previously unseen scenarios, making (un-)supervised learning difficult to apply to broad problem classes in general.

However, a related alternative—reinforcement learning—has been relatively unexplored for quantum chemical application. Whereas unsupervised learning seeks to find underlying structure in unlabeled data, and supervised learning uses labeled data to train a parameterized model, reinforcement learning is designed to learn from experiences or mistakes. It uses feedback observed from an appropriately defined environment to improve strategies to achieve some objective by taking an optimal sequence of actions. By exploring the environment that provides rewards (or penalties), the agent gradually learns which actions to take and which to avoid. In contrast to supervised learning, no answer or action is directly given to the agent, and there is no need to label data or worry about preparing training data, as all data is incorporated as it is generated in the form of a reward signal. Because data can be generated on-the-fly and is used as-is (*e.g.*, there is no need to label data or split data into training, test, and validation sets), reinforcement learning is an attractive approach for the selected CI problem.

Here we propose an alternative approach to the selected configuration interaction problem based on reinforcement learning, where the sCI problem is mapped to a sequential decision-making process that implicitly learns the optimal actions to take to return an accurate, compressed approximation to the FCI wave function. The optimal ranking of configurations can be learned on-the-fly due to observing the impact of including or removing (groups of) determinants on the ground state energy. After detailing the mathematical aspects of the reinforcement learning configuration interaction (RLCI) method, we explore the performance of RLCI methods against several prototypical cases, as well as some larger, strongly correlated hydrogen ring systems.

2 Methods

In configuration interaction methods, the goal is to obtain the ground state energy E of a system by solving the eigenvalue problem corresponding to the molecular electronic Hamiltonian $\hat{H}$, i.e.

$$\hat{H}|\Psi_{\mathrm{CI}}\rangle = E|\Psi_{\mathrm{CI}}\rangle \tag{1}$$

where the CI wave function $|\Psi_{\mathrm{CI}}\rangle$ is represented as a linear expansion of excited Slater determinants out of a reference state $|R\rangle$. That is to say,

$$|\Psi_{\mathrm{CI}}\rangle = \sum_{R}^{N_{\mathrm{ref}}} C_R|R\rangle + \sum_{S} C_S|S\rangle + \sum_{D} C_D|D\rangle + \cdots \tag{2}$$

The coefficients C indicate the contribution of the determinant to the final wave function expansion and S, D, ..., indicate the excitation level out of the reference state. The reference $|R\rangle$ may be a single Slater determinant, such as a Hartree-Fock state, or it may be a CASSCF reference. In contrast to a linear expansion out of the reference that is truncated at some excitation level (e.g. configuration interaction with singles and doubles, CISD), the goal of sCI methods is to minimize the expectation value of the molecular electronic Hamiltonian $\hat{H}$ spanned by a subset of all possible determinants, without regard to an *a priori* truncation. That is, we seek an optimal solution to

$$\min_{\|\Psi_{\mathrm{CI}}\|_0 \leq k} \frac{\langle\Psi_{\mathrm{CI}}|\hat{H}|\Psi_{\mathrm{CI}}\rangle}{\langle\Psi_{\mathrm{CI}}|\Psi_{\mathrm{CI}}\rangle} \tag{3}$$

where $\|\Psi_{\mathrm{CI}}\|_0$ is the cardinality (the number of non-zero elements) of the CI vector. The goal of selected CI methods is to find the solution to Eq. (3) where k is the maximum number of determinants to be considered in the sCI problem. The value of k is a user-defined parameter, and will depend on the particular system and the desired accuracy. If k is equal to the dimension of all possible determinants for a problem, the solution to Eq. (3) is equivalent to the FCI problem. Ultimately, this problem can be viewed as a special instance

of a combinatorial optimization problem, akin to the knapsack problem or the traveling salesman problem. Given a subset of possible determinants to include, what is the optimal combination of determinants that minimizes the expected value of the Hamiltonian?

In reinforcement learning, an agent is trained to take a series of actions in order to maximize a reward.[42] Herein we follow much of the notation and framework laid out for the k-sparse eigenproblem in Ref 43 and apply it to the sCI case. The reinforcement learning process proceeds in a series of episodes, where the agent explores the environment and obtains rewards or penalties for its actions. In order to map the sCI procedure onto a reinforcement learning framework, it is necessary to define the state, the environment, actions, and rewards. The state is the current set of selected determinants to include in the approximate calculation of the sCI problem. The environment is the space of all possible Slater determinants. During each step of an episode, the local reward r can be determined by the change in the obtained eigenvalue λ_{new} from the previous step λ_{old}, that is

$$r = \lambda_{\text{old}} - \lambda_{\text{new}}. \tag{4}$$

Herein we will utilize a Q-learning approach,[42,44] wherein the goal is to learn the optimal state-action value function $Q(s, a)$. Although several different RL approaches could be used, Q-learning is advantageous for its simplicity, robustness in planning, and ability to learn the optimal policy while following a different exploration policy.[42] In Q-learning, the learned function $Q(s, a)$ returns the value of taking a particular action a out of a state s. For a given set s of determinants that make up the current sCI space, possible actions a are removing the p-th determinant from the current set and replacing it with a SD q that is outside the current sCI space, *i.e.*,

$$a = (p, q), \text{ for } p \in s \text{ and } q \notin s \tag{5}$$

So when the optimal $Q(s, a)$ is obtained, from any state s it is possible to obtain the optimal

set of determinants by following the policy $\pi(s)$ for a given state, *i.e.*,

$$\pi(s) = \underset{a}{\text{argmax}}\, Q(s,a) \tag{6}$$

Although the ultimate goal is to learn the optimal policy in Eq. (6) *via* learning the optimal state-action value function $Q(s,a)$, the behavioral policy—that is to say the actions taken during training—need not follow the policy that is being learned in Eq. (6). Indeed, this is undesirable in that the agent will not explore actions which may appear sub-optimal yet may ultimately lead to the global optimal $Q(s,a)$: this is the exploration-exploitation trade off. Q-learning, because it is off-policy (meaning it need not follow Eq. (6) during training) has the flexibility to follow physics-inspired and potentially more efficient search policies, all the while ensuring it learns the optimal $Q(s,a)$ and $\pi(s)$.

The dimension of $Q(s,a)$ is $\dim(s) \times \dim(a)$. Because $\dim(s)$ scales as $\binom{N_{\text{det}}}{k}$, where N_{det} is the dimension of the FCI space, and for any state s, $\dim(a)$ scales as $k \times (N_{\text{det}} - k)$, the size of $Q(s,a)$ is clearly too large to store and utilize directly. To overcome this limitation, we use a linear approximation to $Q(s,a) \approx Q_{\mathbf{w}}(s,a)$, where

$$Q_{\mathbf{w}}(s,a) = \sum_{i \in N_{\text{det}}} w_i f_i(s,a) = \mathbf{w}^\top \mathbf{f}. \tag{7}$$

Here, $Q_{\mathbf{w}}(s,a)$ is the inner product between weights w_i and feature vectors $f_i(s,a)$, which have yet to be defined. One possible definition of the feature vectors $f_i(s,a)$, and the one used in this work, is

$$f_i(s,a) = \begin{cases} 1 & \text{if } i \in s \text{ and } i \neq p, \text{ or if } i \notin s \text{ and } i = q \\ -1 & \text{if } i \in s \text{ and } i = p \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

which then reduces Eq. (7) to a difference in weights w_i, namely

$$Q_{\mathbf{w}}(s,a) = \sum_{i \in s'} w_i - w_p \tag{9}$$

where s' is the active set of determinants subsequent to taking the action $a = (p,q)$, as defined in Eq. (5). One advantage of this choice is that the information in $Q_{\mathbf{w}}(s,a)$ is completely captured in the weights $\mathbf{w}$. Furthermore, the weights $\mathbf{w}$, which are independent of s and a, can be interpreted as maintaining a global ranking for the set of all possible determinants. The definition of the feature vectors in Eq. (8) can be interpreted as mapping the expected value of an action $a = (p,q)$ to the anticipated change in energy by removing determinant p and replacing it with determinant q. In other words, this choice for $f_i(s,a)$ enforces consistency between $\mathbf{w}$ and $Q_{\mathbf{w}}(s,a)$: for a given action $a = (p,q)$ it will always return a positive value if w_p is lower ranked than w_q, or a negative value if w_p is higher ranked than w_q.

In Q-learning, the Q-values (therefore, the weights $\mathbf{w}$) are updated every step according to the conventional update rule[42]

$$w_i \leftarrow w_i + \alpha \frac{dQ_{\mathbf{w}}(s,a)}{dw_i}\delta \tag{10}$$

where the Bellman error δ is

$$\delta = r + \gamma \max_{a'} Q_{\mathbf{w}}(s',a') - Q_{\mathbf{w}}(s,a) = r + \gamma \mathbf{w}^\top \mathbf{f}' - \mathbf{w}^\top \mathbf{f} \tag{11}$$

and $\mathbf{f}'$ denotes the feature vector after taking the action a'. α is the learning rate and γ is the discount factor, which are user-defined parameters and both should take values in $(0,1]$. Larger values of γ favor rewards in the future, rather than immediate rewards which are given by smaller values of γ. Although the update in Eq. (10) appears to be a gradient descent method, for approximate state-action value functions such as that used in Eq. (7), this is no

longer the case. This has been pointed out in Ref 45, Appendix I. To see this, assume that the update in Eq. (10) has been mapped onto the general gradient descent expression

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial \mathbf{J}(\mathbf{w})}{\partial w_i} \tag{12}$$

where $\mathbf{J}(\mathbf{w})$ is some loss function. In the case of Eq. (10), we have

$$\begin{aligned} \frac{\partial \mathbf{J}(\mathbf{w})}{\partial w_i} &= -\frac{\partial Q_{\mathbf{w}}(s,a)}{\partial w_i}\delta & (13)\\ &= \frac{\partial \mathbf{w}^\top \mathbf{f}}{\partial w_i}\left(\mathbf{w}^\top \mathbf{f} - (r + \gamma \mathbf{w}^\top \mathbf{f}')\right) & (14)\\ &= f_i\left(\mathbf{w}^\top \mathbf{f} - (r + \gamma \mathbf{w}^\top \mathbf{f}')\right) & (15) \end{aligned}$$

However, to be a suitable gradient descent method, the second derivatives must be symmetric. Yet for approximate $Q_{\mathbf{w}}(s,a)$ it can be shown that this is not the case:

$$\frac{\partial^2 \mathbf{J}(\mathbf{w})}{\partial w_j \partial w_i} = f_i(f_j - \gamma f_j'); \qquad \frac{\partial^2 \mathbf{J}(\mathbf{w})}{\partial w_i \partial w_j} = f_j(f_i - \gamma f_i'). \tag{16}$$

Therefore, the weights $\mathbf{w}$ are not guaranteed to converge in approximate Q-learning. However, it is possible to modify the weight update in Eq. (10) in order to yield a suitably convergent learning algorithm. To this end, we modify the Greedy-GQ method of Ref 46, which relies on minimizing not the Bellman error directly, but rather the Bellman error projected onto the basis of feature vectors. This is to compensate the gradient due to the lack of an incomplete basis. These terms can be understood in analogy to the Pulay forces[47] in conventional electronic structure nuclear gradients, where additional terms arise due to the fact that the electronic structure calculations are performed in an incomplete basis.

Therefore, the loss function $\mathbf{J}(\mathbf{w})$ is chosen to minimize the projected Bellman error,

rather than the Bellman error (Eq. (11)) itself[42,46]

$$\begin{aligned}\mathbf{J}(\mathbf{w}) &= \|\mathbf{\Pi}\cdot\delta\|^2 \\ &= \left(\delta\cdot\mathbf{f}^\top\right)\left(\mathbf{f}\cdot\mathbf{f}^\top\right)^{-1}(\delta\cdot\mathbf{f})\end{aligned} \tag{17}$$

which leads to the gradient expression

$$\begin{aligned}\frac{1}{2}\frac{\partial\mathbf{J}(\mathbf{w})}{\partial w_i} &= \delta\cdot(\gamma\mathbf{f}'-\mathbf{f})\cdot\mathbf{f}^\top\left(\mathbf{f}\cdot\mathbf{f}^\top\right)^{-1}\mathbf{f} \\ &= -\delta\cdot\mathbf{f}\cdot\mathbf{f}^\top\left(\mathbf{f}\cdot\mathbf{f}^\top\right)^{-1}\mathbf{f}+\gamma\mathbf{f}'\cdot\mathbf{f}^\top\left(\mathbf{f}\cdot\mathbf{f}^\top\right)^{-1}\mathbf{f} \\ &= -\delta\cdot\mathbf{f}+\gamma\mathbf{f}'\cdot\mathbf{f}^\top\cdot\mathbf{v}\end{aligned} \tag{18}$$

where we define a new vector of auxiliary weights $\mathbf{v}$ in order to avoid the computational overhead of inverting matrices of feature vectors, namely

$$\mathbf{v} = \left(\mathbf{f}\cdot\mathbf{f}^\top\right)^{-1}(\delta\cdot\mathbf{f})\,. \tag{19}$$

The vector $\mathbf{v}$ need not be explicitly formed, and will be learned on-the-fly during the training as will be shown. Thus, with the above results, the new update formula for $\mathbf{w}$ is

$$\mathbf{w}\leftarrow\mathbf{w}+\alpha\cdot\left(\delta\cdot\mathbf{f}-\gamma\cdot(\mathbf{f}^\top\mathbf{v})\cdot\mathbf{f}'\right) \tag{20}$$

and

$$\mathbf{v}\leftarrow\mathbf{v}+\beta\cdot(\delta-\mathbf{f}^\top\mathbf{v})\cdot\mathbf{f}. \tag{21}$$

$\mathbf{v}$ is of the same dimension as $\mathbf{w}$ and may be initialized to zero at the beginning of the RLCI algorithm. The update for $\mathbf{v}$ in Eq. (21) is derived from the Least Mean Square (LMS) rule that seeks $\mathbf{v}$ so as to minimize the squared error $\left(\mathbf{f}^\top\mathbf{v}-\delta\right)^2$. This modifies the update scheme to account for the fact that the linear approximation to $Q(s,a)$ is not complete. Note that upon setting the parameter β to zero the algorithm will recover the

classic approximate Q-learning algorithm. In this gradient corrected method, the update in Eq. (20) implicitly assumes that the update in Eq. (21) is at, or near, steady-state. In other words, the secondary learning rate β needs to be greater than the learning rate α.[46] Because α and β take values between 0 and 1, in our experience a reasonable value for β is $\sqrt{\alpha}$, where α is the learning rate. This eliminates the need to set an additional hyperparameter, while maintaining the condition that $\beta > \alpha$.

2.1 Algorithm overview

Here we summarize the algorithm used to generate the results discussed in subsequent sections.

1. From a given reference, initialize the set s through the greedy probing algorithm proposed in Ref 48. In this work, we use the HF wave function as the reference. Briefly, the greedy probing method builds up the set s incrementally through a perturbation-based approach until $\dim(s) = k$, at which the procedure terminates. The new determinants to be added to s are obtained from the first-order perturbative wave function estimate, as is commonly used in other sCI methods, such as CIPSI and ASCI

$$c_i^{(1)} = \frac{\sum_{j \neq i} H_{ij} c_j^{(0)}}{(E^{(0)} - H_{ii})} \tag{22}$$

   where H_{pq} are the Hamiltonian matrix elements between determinants p, q, and $E^{(0)}$ is the energy for the Hamiltonian in the current determinant basis indexed by j with eigenvector components $c_j^{(0)}$. At each iteration of the greedy probing algorithm, the determinant i corresponding to the largest value of $|c_i^{(1)}|$ is added to the set s at each iteration. Unless otherwise stated, we add a single determinant to the set s at each iteration, but we note that the selection procedure need not be limited to adding a single determinant at each iteration and determinants may be added in "batches", e.g. adding the five determinants with the largest magnitude of $|c_i^{(1)}|$ to the set s until

a dimension of k is reached.

2. Given the initial set s, initialize $\mathbf{w}$ using the magnitude of the eigenvector components for the $k \times k$ Hamiltonian submatrix spanned by the determinants in s. For values of $w_i \notin s$, the initial value is estimated using the perturbation theory expression in Eq. (22). In general, the values of $w_i \in s$ and $w_i \notin s$, which combine to form the initial vector $\mathbf{w}$, are not on the same scale as they are estimated from different procedures. Therefore, each component of w_i (i.e., the component $w_i \in s$ and the component $w_i \notin s$ are each normalized individually to the unit vector and subsequently scaled by its proportion of $\mathbf{w}$. This ensures that the initial values of w_i spanning the full determinant space are approximately on the same scale. The auxiliary vector $\mathbf{v}$ is initialized to zero, following Ref 46.

3. Once initialized, the behavioral search policy is as follows. In order to efficiently prune and expand the active determinant space s, we generate two ranked lists: the first list S_1 contains candidate determinants within the current space to be removed (given by low value in $\mathbf{w}$) and the second list S_2 contains candidate determinants outside the current space s to be added (given by high value in Eq. (22)). The dimension of S_1 is the same as the selected subspace k, and nominally the dimension of S_2 is $(N_{\text{det}} - k)$, but for efficiency considerations, S_2 may be limited to the most important external determinants. In this work we make the top 150 determinants available for consideration in the search policy. Note that if an action is selected, the loop over S_1 is immediately terminated. Action pairs $a = (p, q), p \in s, q \notin s$ from the two lists are iterated over, and an action a is taken if it satisfies a Metropolis-like criterion as detailed in Ref 43. If no action is selected, the learning procedure terminates.

4. Once the action a is selected, the search policy terminates and the local reward r is computed according to Eq. (4) and the weights $\mathbf{w}$ and $\mathbf{v}$ are updated according to Eq. (20) and Eq. (21), respectively. Upon taking an action $a = (p, q)$, S_1 can be

updated by removing p and adding q, and the next candidate external determinant in S_2 can be considered. This prevents the need to re-construct S_1 and S_2 at each step in an episode. The episode terminates when no further suitable candidate actions can be found, or once the space is exhausted.

5. These training steps are iterated through until the completion of an episode. To reinitialize the state for a new training episode, the largest k values of $\mathbf{w}$ may be used. We have also found it useful to occasionally and randomly initialize with the best s obtained during the training procedure.

Algorithm 1: Reinforcement Learning Configuration Interaction (RLCI)

Input: matrix A (efficiently represented), number of determinants k

Output: Approximate solution $(\lambda, |\psi_{\text{CI}}\rangle)$ with $\|\psi_{\text{CI}}\|_0 \leq k$

Initialize learning rate α, discount rate γ, exploration rate τ, weights $\mathbf{w}$, and $\mathbf{v}$;

for episode in $1, 2, ..., \text{max\_episode}$ **do**

  Select an initial state s;

  Construct S_1 from $i \in s$ with smallest w_i's;

  Construct S_2 from $j \notin s$ with largest $|c_j|$'s;

  for $j = 1, 2, ..., |S_2|$ **do**

    for $i = 1, 2, ..., |S_1|$ **do**

      $s' = ((s \setminus S_1[i]) \cup S_2[j])$;

      Compute smallest eigenvalue λ' of $A(i \in s', i \in s')$;

      Generate a random number $\epsilon \sim U(0, 1)$;

      if $\lambda' < \lambda \cdot (1 - \tau \cdot \epsilon)$ **then**

        Let $p = S_1[i], q = S_2[j]$, take action $a = (p, q)$ to get new state s';

        Evaluate local reward r;

        Update $\mathbf{w}$ and $\mathbf{v}$;

      end

    end

  end

  Output best approximate $(\lambda, |\psi_{\text{CI}}\rangle)$ during training;

end

Although the cost of the exploration policy can be greatly reduced by limiting the search space in S_2 to the top m external determinants, the RLCI algorithm may still require multiple matrix diagonalizations. However, the brunt of this cost can be greatly reduced by utilizing an iterative subspace diagonalizer, such as Davidson's algorithm,[49] which reduces the cost to $\mathcal{O}(k^2)$, where k is the subspace dimension. The cost is reduced further still by solving

for the lowest eigenpair once at the beginning of each episode, then caching the resulting eigenvector for re-use as an initial guess in subsequent search iterations. Since each action involves the replacement of a single pair of determinants, the wave function overlap between successive actions is very high, and Davidson's algorithm will converge in a few number of iterations. As each new eigenproblem is solved, the guess vector may be overwritten and re-used for the next matrix diagonalization, ensuring that the guess for the current Davidson diagonalization differs from the previous one by no more than one row.

Regarding memory utilization, the weights $\mathbf{w}$ and $\mathbf{v}$ are currently stored explicitly with dimension of the full Hilbert space and this proves to be the largest memory bottleneck. However, many of the (potential) weights are never accessed during the RLCI iterations, and may never need to be stored explicitly. To this end, sparse storage techniques for the learned weights may be utilized. For future work, it may be that moving beyond the linear parameterization of the state-action value function and utilizing deep neural networks can provide a more compact representation of the state and action space.

3 Results

3.1 Prototypical cases: dissociation curves

In order to test the performance of the proposed RLCI method, we have computed potential energy curves for the symmetric dissociation of N_2, CO, and an H_8 chain. These systems span a range of strong to weak correlation and allow us to evaluate the prototypical performance of RLCI methods for use in quantum chemistry. Note that we assume a point group of C_1 for all systems investigated here, and all data is obtained with an RHF reference wave function using canonical RHF orbitals, and the underlying integrals and Hamiltonians were obtained using an interface to the PySCF software package.[50] Although currently the code is far from optimized, we aim to show that RLCI is a promising route forward for generating highly compact wave functions at chemical accuracy. In this work, we choose $\alpha = 0.5$ and

$\gamma = 0.99$ for all cases. Learning rates are not damped to avoid early convergence to local minima, and the somewhat larger value of the learning rate α is suitable considering the deterministic CI environment.[42] The exploration rate decays as a function of episode, with $\tau = \exp(-0.5 \cdot \text{episode})$. Additionally, all RL runs are terminated within 30 episodes. Other choices of hyperparameters may be explored in future work.

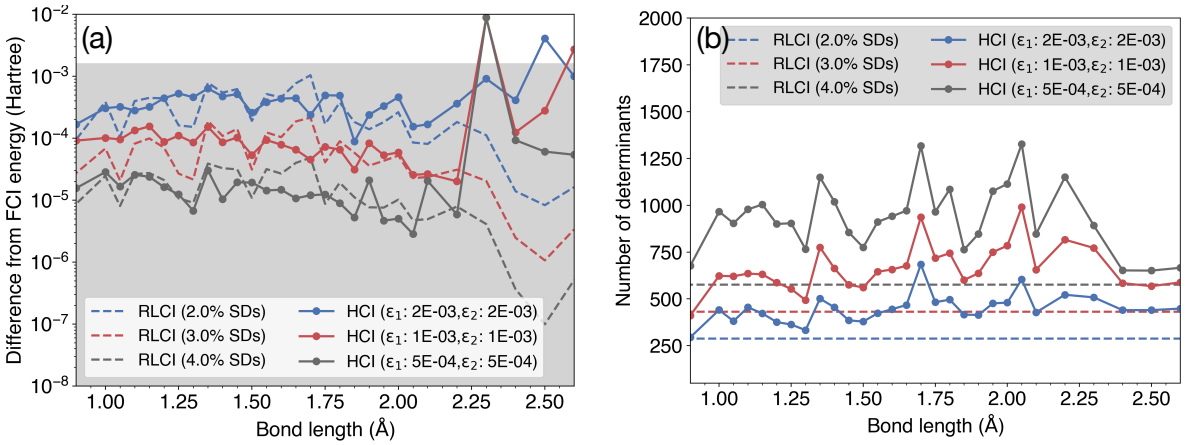

Figure 1. Comparison of RLCI and HCI methods for the dissociation of the N_2 molecule with the STO-6G basis with different levels of approximation. (a) Difference between FCI and RLCI or HCI potential energy curves, relative to the FCI minimum. The shaded gray area indicates region of chemical accuracy (1 kcal/mol). (b) Comparison of the number of determinants used to calculate each point along the potential energy surface. The size of the full space is 14,400 determinants.

An exploration of the dissociation behavior of N_2 using the RLCI method is given in Fig. 1. The errors with respect to FCI for N_2 with the STO-6G basis is given in Fig. 1a. N_2 dissociation is a challenging problem in quantum chemistry, and most single-reference methods will fail to describe this process, particularly at larger separations. This is due to the high amount of strong correlation required to dissociate the triple bond. To compare with existing sCI methods, we compared with the heat-bath selected configuration interaction (HCI) method[15] in PySCF with varying levels of approximation. HCI depends on two parameters, ϵ_1 and ϵ_2, and values for these parameters were chosen to reasonably mimic the level of accuracy obtained with the RLCI method at different levels of sparsity. This is to give a sense of how much each method can compress the wave function. Because

the parameters that govern the accuracy of each method differ (the RLCI parameter is the number of determinants k, whereas the HCI ϵ values govern determinant cutoff parameters) they cannot be compared one-to-one, however by observing the error with respect to FCI and the number of determinants used in Fig. 1b, general trends can be observed. The HCI and RLCI methods are competitive in the sense that both methods can be tuned to yield errors below chemical accuracy with reasonable wave function compression (worst case for all methods for N_2 here are still $< 10\%$ of the total FCI space). As can be seen, in order to get the accuracies to roughly agree in Fig. 1a, the number of parameters (determinants) for HCI is roughly double that of RLCI. In the more strongly correlated regime (> 2.0 Å), the agreement in accuracy between HCI and RLCI diverges, despite HCI utilizing a greater number of parameters. In fairness, for the current implementation HCI is a significantly faster method, but the savings in terms of wave function compression for RLCI suggest that pursuing further improvements to the method may be warranted.

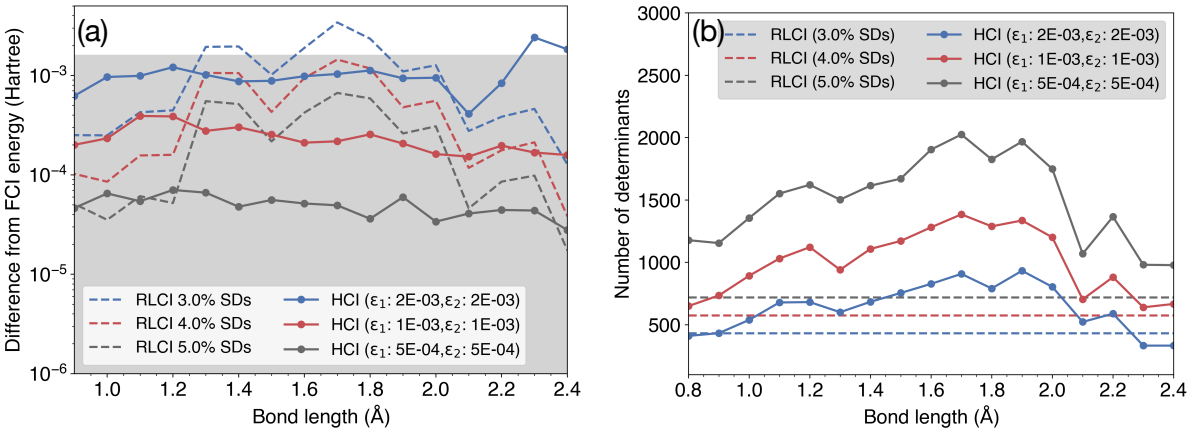

Figure 2. Comparison of RLCI and HCI methods for the dissociation of the CO molecule with the STO-6G basis with different levels of approximation. (a) Difference between FCI and RLCI or HCI potential energy curves, relative to the FCI minimum. The shaded gray area indicates region of chemical accuracy (1 kcal/mol). (b) Comparison of the number of determinants used to calculate each point along the potential energy surface. The size of the full space is 14,400 determinants.

Next, we explore the dissociation of the CO molecule in Fig. 2. Though HCI maintains a roughly constant accuracy throughout the dissociation, it requires roughly $2-3\times$ the number

of determinants as RLCI. In contrast, RLCI can maintain kcal/mol accuracy or better while utilizing 4.0% of the determinant space. That said, the errors for RLCI increase during dissociation up until around 1.7 Å, at which point they begin to decrease again, whereas HCI is relatively constant accuracy. This suggests that investigating ways to dynamically set the RLCI parameter k may be beneficial. For HCI, the free parameters govern the cutoffs used to determine the importance of a determinant, which in this particular case seems to yield more constant energy errors, though this requires the use of an increasing determinant space.

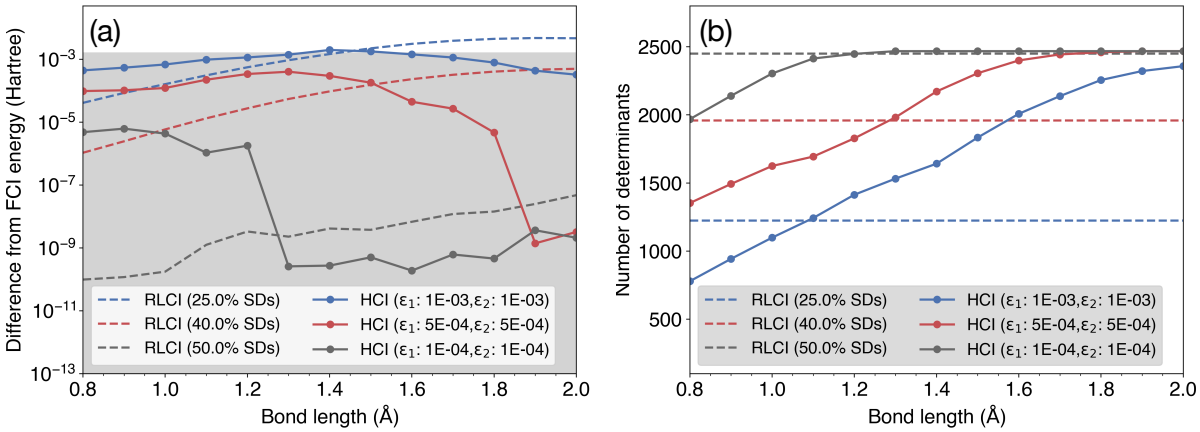

Figure 3. Comparison of RLCI and HCI methods for the symmetric dissociation of an H_8 chain with the STO-6G basis with different levels of approximation. (a) Difference between FCI and RLCI or HCI potential energy curves, relative to the FCI minimum. The shaded gray area indicates region of chemical accuracy (1 kcal/mol). (b) Comparison of the number of determinants used to calculate each point along the potential energy surface. The size of the full space is 4,900 determinants.

Finally, we explore the symmetric dissociation of the H_8 chain in Fig. 3. Despite its apparent simplicity, this is a challenging system with rapidly increasing amounts of strong correlation as the system dissociates. This is reflected in the linear increase in log error in Fig. 3a, and the linear increase in determinants required for HCI to maintain its sub-chemical accuracy observed in Fig. 3b. Despite these challenges, RLCI is able to obtain better accuracy for fewer determinants compared to HCI for most points along the dissociation curve, similar to what has been observed in the other prototype calculations for CO and N_2.

At high intra-atomic separations, the necessary parameters to achieve chemical accuracy in the (very) strongly correlated limit seem to limit toward a unified value of roughly 25% of the determinant space. We note that there is no orbital optimization in either the RLCI or HCI methods presented here, which may prove beneficial in cases such as these. Or, there may simply be limits to how well determinant-based methods can compress the wave function and other wave function ansätze, such as matrix[51] or tensor[18] product states may prove more efficient. Utilizing configuration state function (CSFs) may also additionally compress the wave function, though the RLCI will lose some of the computational advantages of a determinant-based approach. Thus, further improvements may be found in both extending RLCI to treat orbital optimization, as well as extending RL methods to optimize non-determinant-based wave function ansätze, like CSFs or tensor product states.

An example of the convergence behavior with respect to episode for the previous three test cases is given in Fig. 4, which depicts the difference between the best estimate of the energy at each episode and the lowest energy obtained overall. Each of the molecules (CO, H_8 chain, and N_2) was fixed with an interatomic distance of 1.5 Å and otherwise used the same parameters as those given previously. To explore the impact of the the dimension of the external determinant search space $|S_2|$ was allowed to vary, with $|S_2| = 50,\ 100,\ 200$. As can be seen, there is a rapid initial drop in energy in the first few episodes, which serve to quickly correct the initial guess. As later episodes explore the determinant space, the agent continues to decrease the energy and refine the set of determinants. Larger external determinant spaces may provide faster convergence with respect to episode, but the larger set of candidate actions results in longer episodes, which may explain this behavior. If the external determinant space is too small, the agent may end up in a higher energy local minima, as may be the case in Fig. 4b for the $|S_2| = 50$ case after episode 28. In most cases, minimal energetic refinement is obtained after 30 episodes, though for larger systems it may be necessary to consider additional episodes.

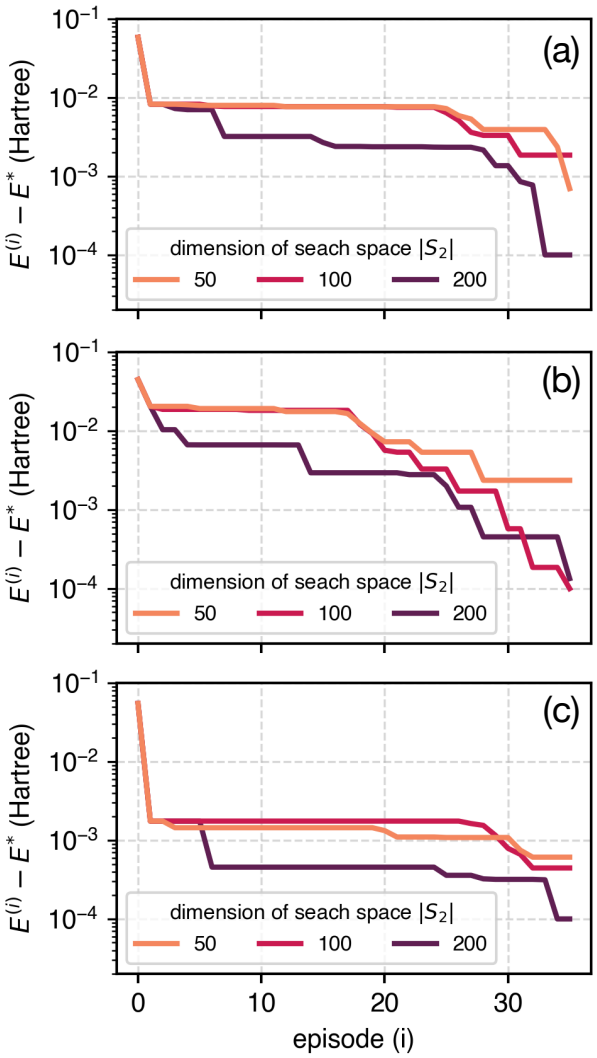

Figure 4. Difference between the best estimate of the energy at each episode i and the lowest energy obtained overall for (a) CO, (b) H_8 chain, and (c) N_2. Each molecule used a STO-6G basis and had an interatomic distance of 1.5 Å. The dimension of the subspace k was 200 determinants, and the dimension of the external determinant search space $|S_2|$ was varied to be 50, 100, or 200. All other parameters correspond to those used in the rest of the work. A small scalar constant of 1×10^{-4} was added to the lowest energy obtained overall in order to avoid plotting zero on a log scale.

3.2 Larger examples: hydrogen rings

In order to get a sense of the performance for larger systems, we have applied the RLCI algorithm to larger systems of hydrogen rings with a STO-6G basis. These hydrogen rings have been investigated recently in order to explore the ability of several methods to compress the wave function in strongly correlated materials.[5] In order to apply the RLCI method to these larger systems, we have implemented the RLCI algorithm in the Chronus Quantum electronic structure program,[52] with efficient determinant manipulation based on Ref. 53. At the moment, the algorithm uses a canonical generalized Hartree-Fock reference, which results in much larger and far more sparse determinant spaces. As an example, an H_{10}/STO-6G ring with canonical RHF orbitals has a full Hilbert space dimension of 63504, whereas with GHF the full Hilbert space is three times larger with dimension 184756. In any case, the RLCI algorithm is agnostic to the type of reference: if the method can handle a more challenging GHF reference, it can also handle a RHF reference.

Table 1. Comparison of the percent FCI correlation energy (% corr.) captured versus RLCI subspace dimension k as a function of the percentage of the full Hilbert space (% N_{det}) for hydrogen rings with n atoms, an interatomic separation of 1.5 Å, using a STO-6G basis.

n	k	N_{det}	% N_{det}	% corr.
10	2000	184756	1.083%	96.2%
10	4000	184756	2.165%	98.3%
10	6000	184756	3.248%	99.2%
12	4000	2704156	0.148%	74.3%
12	6000	2704156	0.222%	77.2%
12	10000	2704156	0.370%	95.6%
14	6000	40116600	0.015%	86.2%
14	10000	40116600	0.025%	88.6%
14	20000	40116600	0.050%	91.3%
16	10000	601080390	0.002%	47.4%
16	20000	601080390	0.003%	49.6%
16	50000	601080390	0.008%	52.4%

In Tab. 1, we compare the performance of the RLCI algorithm for a series of hydrogen rings with an interatomic separation of 1.5 Å and an STO-6G basis. This interatomic

distance was previously found to exhibit strong correlation in hydrogen rings.[5] Using the GHF reference, fully correlating all electrons, and utilizing rings ranging from 10 to 16 atoms, the full Hilbert space spans from 10^5 to nearly 10^9 determinants. All parameters for the calculations were the same as those used previously, with the exception that the greedy initialization for each trial was done in batches corresponding to 1/10 of the subspace k. As observed in the prototypical cases, only a small percent of the full Hilbert space is necessary to capture a significant portion of the FCI correlation energy. For the H_{10} ring, 2% of the Hilbert space is sufficient to capture 98% of the correlation energy. For the H_{14} ring, even 0.05% of the Hilbert space is sufficient to capture over 91% of the correlation energy. The results in Tab. 1 are consistent with the observations found for the smaller, prototypical systems.

4 Conclusion

Here we have explored the potential of using reinforcement learning techniques to solve the selected configuration interaction problem. In the prototypical cases explored, RLCI outperformed HCI in terms of generating more compact wave functions without neglecting chemical accuracy ($<$ 1 kcal/mol from FCI). Although we do not claim that the current implementation is necessarily *faster* than existing sCI methods, we have provided support that approaches based on reinforcement learning may yield more optimally compact wave functions, at least at the determinantal level. In these dissociation curves presented here, HCI appears to require on the order of $2-3\times$ the number of determinants for comparable accuracy to RLCI. As is seen in the strongly correlated case of the stretched H_8 chain, there may be an upper limit to how compressed a wave function can become. In these cases, other wave function ansätze may be required, or it may speak to the importance of orbital optimization in the strongly correlated cases. Both avenues are worth pursuing with a RL approach. Other improvements of the RLCI method include modifying the action space to

allow more than one determinant to be added or removed from the state, optimizing the learning rate and the discount factor, and gaining a better understanding of the trade-off between exploration and exploitation. Further work to explore the transfer of data between different Hamiltonians in order to improve efficiency should also be explored. Additional investigations of perturbative corrections on top of the RLCI-learned wave function may also yield robust convergence to the FCI limit with compact wave function references, as has been observed in other sCI methods[5,18,54–56]

Acknowledgement

The development of multi-reference method is supported by the U.S. Department of Energy in the Heavy-Element Chemistry program (Grant No. DE-SC0021100 to X.L.). Y.C. and X.L. acknowledge the support to develop reduced scaling electronic structure methods from the Computational Chemical Sciences (CCS) Program of the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division in the Center for Scalable and Predictive methods for Excitations and Correlated phenomena (SPEC) at the Pacific Northwest National Laboratory. The development of the open source software package, ChronusQ, is supported by the U.S. National Science Foundation (OAC-1663636).

References

(1) Knowles, P. J.; Handy, N. C. A Determinant Based Full Configuration Interaction Program. *Comp. Phys. Comm.* **1989**, *54*, 75–83.

(2) Bytautas, L.; Ruedenberg, K. A Priori Identification of Configurational Deadwood. *Chem. Phys.* **2009**, *356*, 64–75.

(3) Taylor, P. R. Lossless Compression of Wave Function Information using Matrix Factorization: A "gzip" for Quantum Chemistry. *J. Chem. Phys.* **2013**, *139*, 074113.

(4) Knowles, P. J. Compressive Sampling in Configuration Interaction Wavefunctions. *Mol. Phys.* **2015**, *113*, 1655–1660.

(5) Stair, N. H.; Evangelista, F. A. Exploring Hilbert Space on a Budget: Novel Benchmark Set and Performance Metric for Testing Electronic Structure Methods in the Regime of Strong Correlation. *J. Chem. Phys.* **2020**, *153*, 104108.

(6) Eriksen, J. J. The Shape of Full Configuration Interaction to Come. *J. Phys. Chem. Lett.* **2020**, 418–432.

(7) García, V. M.; Castell, O.; Caballol, R.; Malrieu, J. P. An Iterative Difference-dedicated Configuration Interaction. Proposal and Test Studies. *Chem. Phys. Lett.* **1995**, *238*, 222–229.

(8) Neese, F. A Spectroscopy Oriented Configuration Interaction Procedure. *J. Chem. Phys.* **2003**, *119*, 9428–9443.

(9) Nakatsuji, H.; Ehara, M. Iterative Cl General Singles and Doubles (ICIGSD) Method for Calculating the Exact Wave Functions of the Ground and Excited States of Molecules. *J. Chem. Phys.* **2005**, *122*, 194108.

(10) Abrams, M. L.; Sherrill, C. D. Important Configurations in Configuration Interaction and Coupled-cluster Wave Functions. *Chem. Phys. Lett.* **2005**, *412*, 121–124.

(11) Roth, R. Importance Truncation for Large-scale Configuration Interaction Approaches. *Phys. Rev. C* **2009**, *79*, 064324.

(12) Evangelista, F. A. Adaptive Multiconfigurational Wave Functions. *J. Chem. Phys.* **2014**, *140*, 124114.

(13) Liu, W.; Hoffmann, M. R. iCI: Iterative CI Toward Full CI. *J. Chem. Theory Comput.* **2016**, *12*, 1169–1178.

(14) 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.

(15) Holmes, A. A.; Tubman, N. M.; Umrigar, C. J. Heat-Bath Configuration Interaction: An Efficient Selected Configuration Interaction Algorithm Inspired by Heat-Bath Sampling. *J. Chem. Theory Comput.* **2016**, *12*, 3674–3680.

(16) Schriber, J. B.; Evangelista, F. A. Adaptive Configuration Interaction for Computing Challenging Electronic Excited States with Tunable Accuracy. *J. Chem. Theory Comput.* **2017**, *13*, 5354–5366.

(17) Tubman, N. M.; Levine, D. S.; Hait, D.; Head-Gordon, M.; Birgitta Whaley, K. An Efficient Deterministic Perturbation Theory for Selected Configuration Interaction Methods. *arXiv preprint.* **2018**, `https://arxiv.org/abs/1808.02049`.

(18) Abraham, V.; Mayhall, N. J. Selected Configuration Interaction in a Basis of Cluster State Tensor Products. *J. Chem. Theory Comput.* **2020**,

(19) Garniron, Y.; Scemama, A.; Giner, E.; Caffarel, M.; Loos, P.-F. Selected Configuration Interaction Dressed by Perturbation. *J. Chem. Phys.* **2018**, *149*, 064103.

(20) Loos, P.-F.; Damour, Y.; Scemama, A. The Performance of CIPSI on the Ground State Electronic Energy of Benzene. *J. Chem. Phys.* **2020**, *153*, 176101.

(21) Gyorffy, W.; Bartlett, R. J.; Greer, J. C. Monte Carlo Configuration Interaction Predictions for the Electronic Spectra of Ne, CH2, C2, N2, and H2O Compared to Full Configuration Interaction Calculations. *J. Chem. Phys.* **2008**, *129*, 064103.

(22) Coe, J. P.; Paterson, M. J. Development of Monte Carlo Configuration Interaction: Natural Orbitals and Second-order Perturbation Theory. *J. Chem. Phys.* **2012**, *137*, 204108.

(23) Coe, J. P.; Murphy, P.; Paterson, M. J. Applying Monte Carlo Configuration Interaction to Transition Metal Dimers: Exploring the Balance Between Static and Dynamic Correlation. *Chem. Phys. Lett.* **2014**, *604*, 46–52.

(24) Coe, J. P.; Paterson, M. J. State-averaged Monte Carlo Configuration Interaction Applied to Electronically Excited States. *J. Chem. Phys.* **2013**, *139*, 154103.

(25) Greer, J. C. Monte Carlo Configuration Interaction. *J. Comput. Phys.* **1998**, *146*, 181–202.

(26) Greer, J. C. Estimating Full Configuration Interaction Limits from a Monte Carlo Selection of the Expansion Space. *J. Chem. Phys.* **1995**, *103*, 1821–1828.

(27) Dash, M.; Moroni, S.; Scemama, A.; Filippi, C. Perturbatively Selected Configuration-Interaction Wave Functions for Efficient Geometry Optimization in Quantum Monte Carlo. *J. Chem. Theory Comput.* **2018**, *14*, 4176–4182.

(28) Li, J.; Otten, M.; Holmes, A. A.; Sharma, S.; Umrigar, C. J. Fast Semistochastic Heat-bath Configuration Interaction. *J. Chem. Phys.* **2018**, *149*, 214110.

(29) Chien, A. D.; Holmes, A. A.; Otten, M.; Umrigar, C. J.; Sharma, S.; Zimmerman, P. M. Excited States of Methylene, Polyenes, and Ozone from Heat-Bath Configuration Interaction. *J. Phys. Chem. A* **2018**, *122*, 2714–2722.

(30) Holmes, A. A.; Umrigar, C. J.; Sharma, S. Excited States using Semistochastic Heat-bath Configuration Interaction. *J. Chem. Phys.* **2017**, *147*, 164111.

(31) 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.

(32) Aspuru-Guzik, A.; Lindh, R.; Reiher, M. The Matter Simulation (R)evolution. *ACS Cent. Sci.* **2018**, *4*, 144–152.

(33) Dral, P. O. Quantum Chemistry in the Age of Machine Learning. *J. Phys. Chem. Lett.* **2020**, *11*, 2336–2347.

(34) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse Molecular Design Using Machine Learning: Generative Models for Matter Engineering. *Science* **2018**, *361*, 360–365.

(35) Senior, A. W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A. W. R.; Bridgland, A.; Penedones, H.; Petersen, S.; Simonyan, K.; Crossan, S.; Kohli, P.; Jones, D. T.; Silver, D.; Kavukcuoglu, K.; Hassabis, D. Improved Protein Structure Prediction Using Potentials from Deep Learning. *Nature* **2020**, *577*, 706–710.

(36) Schütt, K. T.; Gastegger, M.; Tkatchenko, A.; Müller, K.-R.; Maurer, R. J. Unifying Machine Learning and Quantum Chemistry with a Deep Neural Network for Molecular Wavefunctions. *Nat. Commun.* **2019**, *10*, 5024.

(37) Westermayr, J.; Marquetand, P. Machine Learning for Electronically Excited States of Molecules. *Chem. Rev.* **2020**,

(38) Häse, F.; Fdez Galván, I.; Aspuru-Guzik, A.; Lindh, R.; Vacher, M. How Machine Learning Can Assist the Interpretation of Ab Initio Molecular Dynamics Simulations and Conceptual Understanding of Chemistry. *Chem. Sci.* **2019**, *10*, 2298–2307.

(39) Goings, J. J.; Hammes-Schiffer, S. Nonequilibrium Dynamics of Proton-Coupled Electron Transfer in Proton Wires: Concerted but Asynchronous Mechanisms. *ACS Cent. Sci.* **2020**, *6*, 1594–1601.

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

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

(42) Sutton, R. S.; Barto, A. G. *Reinforcement Learning: An Introduction*; MIT press, 2018.

(43) Zhou, L.; Yan, L.; Caprio, M. A.; Gao, W.; Yang, C. Solving the k-sparse Eigenvalue Problem with Reinforcement Learning. *CSIAM Trans. Appl. Math.* **2020**, *submitted,* `arXiv:2009.04414v1`, 1–29.

(44) Watkins, C. J. C. H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292.

(45) Barnard, E. Temporal-difference Methods and Markov Models. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 357–365.

(46) Maei, H. R.; Szepesvári, C.; Bhatnagar, S.; Sutton, R. S. Toward Off-Policy Learning Control with Function Approximation. International Conference on Machine Learning. 2010.

(47) Pulay, P. Ab Initio Calculation of Force Constants and Equilibrium Geometries in Polyatomic Molecules. *Mol. Phys.* **1969**, *17*, 197–204.

(48) Hernandez, T. M.; Van Beeumen, R.; Caprio, M. A.; Yang, C. A Greedy Algorithm for Computing Eigenvalues of a Symmetric Matrix with Localized Eigenvectors. *Numer. Linear Algebra Appl.* **2020**, e2341.

(49) Davidson, E. The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices. *J. Comput. Phys.* **1975**, *17*, 87–94.

(50) Sun, Q.; Berkelbach, T. C.; Blunt, N. S.; Booth, G. H.; Guo, S.; Li, Z.; Liu, J.; McClain, J. D.; Sayfutyarova, E. R.; Sharma, S.; Wouters, S.; Chan, G. K. PySCF: the Python-based Simulations of Chemistry Framework. 2017.

(51) Olivares-Amaya, R.; Hu, W.; Nakatani, N.; Sharma, S.; Yang, J.; Chan, G. K.-L. The Ab-Initio Density Matrix Renormalization Group in Practice. *J. Chem. Phys.* **2015**, *142*, 034102.

(52) Williams-Young, D. B.; Petrone, A.; Sun, S.; Stetina, T. F.; Lestrange, P.; Hoyer, C. E.; Nascimento, D. R.; Koulias, L.; Wildman, A.; Kasper, J.; Goings, J. J.; Ding, F.; DePrince III, A. E.; Valeev, E. F.; Li, X. The Chronus Quantum (ChronusQ) Software Package. *WIREs Comput. Mol. Sci.* **2019**, e1436.

(53) Scemama, A.; Giner, E. An Efficient Implementation of Slater-Condon Rules. *arXiv preprint.* **2013**, `https://arxiv.org/abs/1311.6244`.

(54) Tubman, N. M.; Freeman, C. D.; Levine, D. S.; Hait, D.; Head-Gordon, M.; Whaley, K. B. Modern Approaches to Exact Diagonalization and Selected Configuration Interaction with the Adaptive Sampling CI Method. *J. Chem. Theory Comput.* **2020**, *16*, 2139–2159.

(55) Levine, D. S.; Hait, D.; Tubman, N. M.; Lehtola, S.; Whaley, K. B.; Head-Gordon, M. CASSCF with Extremely Large Active Spaces Using the Adaptive Sampling Configuration Interaction Method. *J. Chem. Theory Comput.* **2020**, *16*, 2340–2354.

(56) Zhang, N.; Liu, W.; Hoffmann, M. R. Iterative Configuration Interaction with Selection. *J. Chem. Theory Comput.* **2020**, *16*, 2296–2316.

Graphical TOC Entry