Accurate and Efficient Open-Source Implementation of Domain-Based Local Pair Natural Orbital (DLPNO) Coupled-Cluster Theory Using a t_1-Transformed Hamiltonian

Andy Jiang,[1] Zachary L. Glick,[2] David Poole,[2] Justin M. Turney,[1, a)] C. David Sherrill,[2, b)] and Henry F. Schaefer III[1]

[1)] *Center for Computational Quantum Chemistry, Department of Chemistry, University of Georgia, Athens, GA 30602, United States of America*

[2)] *Center for Computational Molecular Science and Technology, School of Chemistry and Biochemistry, School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0400, United States of America*

(Dated: 26 July 2024)

We present an efficient, open-source formulation for coupled-cluster theory through perturbative triples with domain-based local pair natural orbitals [DLPNO-CCSD(T)]. Similar to the implementation of the DLPNO-CCSD(T) method found in the ORCA package, the most expensive integral generation and contraction steps associated with the CCSD(T) method are linear-scaling. In this work, we show that the t_1-transformed Hamiltonian allows for a less complex algorithm when evaluating the local CCSD(T) energy without compromising efficiency or accuracy. Our algorithm yields sub-kJ mol^{-1} deviations for relative energies when compared with canonical CCSD(T), with typical errors being on the order of 0.1 kcal mol^{-1}, using our TightPNO parameters. We extensively tested and optimized our algorithm and parameters for non-covalent interactions, which have been the most difficult interaction to model for orbital (PNO)-based methods historically. To highlight the capabilities of our code, we tested it on large water clusters, as well as insulin (787 atoms).

[a)] Electronic mail: justin.turney@uga.edu

[b)] Electronic mail: sherrill@gatech.edu

I. INTRODUCTION

Coupled-cluster (CC) theory[1,2] is one of the greatest triumphs of modern quantum chemistry, allowing for the accurate evaluation of the electronic energy of a molecule in polynomial time, as an approximation to solving the time-independent Schrödinger equation. Full configuration interaction (FCI)[3,4] seeks to provide the exact energy and wave function, within a finite basis set. Unfortunately, FCI scales as $\mathcal{O}(N!)$ with respect to the size of the molecule, rendering it very challenging for molecules larger than a few atoms. CC theory allows for a systematic series of approximations to FCI, and its exponential ansatz allows for size-extensivity of electronic energies. The basic equations for CC theory are

$$|\Psi_{\mathrm{CC}}\rangle = e^{T}\,|\Psi_0\rangle \ , \tag{1}$$

$$E_{\mathrm{CC}} = \left\langle \Psi_0 \mid e^{-T} H e^{T} \mid \Psi_0 \right\rangle \ , \tag{2}$$

where Ψ_0 is the reference Hartree–Fock wave function given by a single Slater determinant, T is the electron excitation operator, and H is the molecular Hamiltonian, within the Born-Oppenheimer approximation. With the T operator, any number of electron excitations can be considered, up to the number of electrons in the system. More excitations considered means a larger runtime, in exchange for greater accuracy. Coupled-cluster methods are defined by the highest level of electronic excitations that are considered. For the CCSD method, $T = T_1 + T_2$, such that

$$|\Psi_{\mathrm{CCSD}}\rangle = e^{(T_1+T_2)}\,|\Psi_0\rangle \ , \tag{3}$$

$$E_{\mathrm{CCSD}} = \left\langle \Psi_0 \mid e^{-(T_1+T_2)} H e^{(T_1+T_2)} \mid \Psi_0 \right\rangle \ . \tag{4}$$

where T_1 represents the excitation operator where one electron is excited from the ground-state wave function, while T_2 is the two-electron excitation operator. The CCSD(T) method[5] considers triples electronic excitations T_3 in a perturbative manner using the CCSD wave function. CCSD(T) is known as the "gold standard" method in quantum chemistry, with errors in relative energy (versus FCI) often around 1 kcal mol^{-1} or less,[6–9] and often with excellent agreement with experimental data.[10]

Unfortunately, the cost of evaluating the CCSD wave function for a molecule scales $\mathcal{O}(N^6)$, and CCSD(T) adds a non-iterative $\mathcal{O}(N^7)$ step on top of the iterative CCSD method.

This means that CCSD and CCSD(T) methods are intractable for systems with more than around 30 atoms on a typical workstation. Therefore, it is useful to devise a series of approximations to CCSD and CCSD(T) that allow them to be useful for larger molecular complexes, such as pharmaceutical molecules, protein fragments, and smaller whole proteins (like crambin and insulin) to allow for increased applicability of high-accuracy quantum chemistry to fields like drug discovery and computational biology. Currently, cheaper methods like density functional theory (DFT)[11–13] or Møller–Plesset perturbation theory (MP2)[14,15] are applied to these problems, but they do not have the accuracy of coupled-cluster.

One such approach to increase the efficiency of coupled-cluster based methods is through rank reduction. Parrish et al. have used orthogonal projectors to transform CCSD amplitudes into smaller-ranked tensors.[16–18] Lesiuk has successfully applied such an approach to CCSD(T).[19] Rank reduction can also be used in conjunction with tensor hypercontraction (THC) methods.[20–23] Through the use of the CANDECOMP/PARAFAC (CP) decomposition[24] of the orthogonal projectors, Hohenstein et al. have applied THC to CCSD amplitudes,[18] while Jiang et al. recently applied this approach to the (T) correction.[25]

Another approach to this problem is to reformulate coupled-cluster theory through local correlation methods,[26–46] especially methods that use pair natural orbitals (PNOs)[29,30] (triples natural orbitals (TNOs)[35] are used for the triples terms). State-of-the-art PNO-based coupled-cluster methods include the in the domain-based local pair natural orbital [DLPNO-CCSD(T)] method[40,44] in ORCA;[47] the pair natural orbital local [PNO-LCCSD(T)] method[41,45] in Molpro;[48] and the local natural orbital [LNO-CCSD(T)] method[43,46] in MRCC.[49] ORCA's DLPNO-CCSD(T) algorithm has been executed on system sizes containing more than 1000 atoms,[40] far greater than the 30 atoms using canonical CCSD(T) methods. As used in practice, these methods are reasonably accurate approximations to canonical CCSD(T) at a greatly reduced computational cost. Approximations are often made to the (T) correction which render it not fully equivalent to its canonical variant, such as through the semi-canonical (T0) algorithm.[35] However, alternative formulations, such as the iterative (T) algorithm, also known as (T1) (not to be confused with the t_1 transformation),[44] do yield the canonical result in the limit of very tight cutoffs. In either case, errors relative to canonical CCSD(T) are typically proportional to system size (are "size extensive"), and good accuracy can typically be maintained by proper selection of parameters and/or (T) algorithm.[50]

To contribute to the development of these highly efficient and popular PNO-based methods, and to make these methods more accessible to the quantum chemistry community, we implemented our own version of the DLPNO-CCSD(T) method in the open-source Psi4 package.[51] While working on this project, it has been brought to our attention that the DLPNO-CCSD(T) method has previously been implemented in the open-source SERENITY program.[52,53] However, the implementation in SERENITY only treats triples through the semi-canonical (T0) treatment,[35] so this work, to the best of our knowledge, represents the first open-source implementation of DLPNO-CCSD(T) that includes an iterative (T) algorithm to account for the non-canonical local orbitals.[44] The difference between semi-canonical (T0) and iterative (T) will be further elaborated in later sections.

In this work, we make use of the t_1-transformed Hamiltonian to reduce the complexity of the CCSD equations,[54,55] and we present our own set of LCCSD working equations that minimize common sources of error in PNO-based methods, like PNO projection error. We have also developed a set of parameters that allow our code to yield relative energies with deviations on the order of 0.1 kcal mol^{-1} from canonical CCSD(T), called TightPNO, following the convention of Neese et al.[34,40] We test our code extensively on relative energies, including interaction energies and conformation energies. Weak, non-covalent interactions have historically been a challenge for local correlation methods.[36] We also present results for some of the largest systems on which a canonical CCSD(T) computation have been performed, the 16 and 17-molecule water cluster conformers with an aug-cc-pVTZ basis set.[56] We compare our results to ORCA's implementation of DLPNO-CCSD(T), as well as a canonical MP2/CCSD(T) many-body expansion method.[57] Finally, we benchmark our algorithm on a whole insulin chain (787 atoms),[58] a system significantly beyond the reach of conventional coupled-cluster theory.

II. THEORY

A. Notation

We use the following conventions to describe the indices of matrices and tensors appearing in this work:

- $\mu, \nu, \lambda, \sigma$: atomic orbitals; these range from 1 to n_{bf}, the number of basis functions

- i, j, k, l: canonical and local occupied molecular orbitals; these range from 1 to n_{occ}, the number of occupied orbitals
- a, b, c, d: canonical virtual molecular orbitals; these range from 1 to n_{virt}, the number of virtual orbitals
- p, q, r, s: general canonical molecular orbitals; these range from 1 to $n_{occ} + n_{virt}$
- $\widetilde{\mu}, \widetilde{\nu}, \widetilde{\lambda}, \widetilde{\sigma}$: projected atomic orbitals; these range from 1 to n_{bf}
- $\widetilde{\mu}_{ij}, \widetilde{\nu}_{ij}, \widetilde{\lambda}_{ij}, \widetilde{\sigma}_{ij}$: projected atomic orbitals localized to pair ij; these range from 1 to $n_{pao,ij}$, number of PAOs local to LMO pair ij
- $\widetilde{\mu}_{ijk}, \widetilde{\nu}_{ijk}, \widetilde{\lambda}_{ijk}, \widetilde{\sigma}_{ijk}$: projected atomic orbitals localized to triplet ijk; these range from 1 to $n_{pao,ijk}$, number of PAOs local to LMO triplet ijk
- $a_{ij}, b_{ij}, c_{ij}, d_{ij}$: Pair natural orbitals in each pair domain ij; these range from 1 to $n_{pno,ij}$, number of PNOs in the domain of LMO pair ij
- $a_{ijk}, b_{ijk}, c_{ijk}, d_{ijk}$: Triples natural orbitals in each triplet domain ijk; these range from 1 to $n_{tno,ijk}$, number of TNOs in the domain of LMO triplet ijk
- P, Q: auxiliary basis functions for density-fitted ERIs; these range from 1 to n_{aux}, number of auxiliary basis functions for density fitting
- P_{ij}, Q_{ij}: local auxiliary basis functions in each pair domain ij; these range from 1 to $n_{aux,ij}$, number of auxiliary basis functions local to LMO pair ij
- P_{ijk}, Q_{ijk}: local auxiliary basis functions in each triplet domain ijk; these range from 1 to $n_{aux,ijk}$, number of auxiliary basis functions local to LMO triplet ijk

The relative sizes of these indices are typically:

$$n_{pno,ij} < n_{tno,ijk} \ll n_{pao,ij} < n_{pao,ijk} < n_{aux,ij} < n_{aux,ijk} \sim \mathcal{O}(1) \ . \tag{5}$$

$$n_{occ} \ll n_{virt} < n_{bf} < n_{naux} \sim \mathcal{O}(N) \ . \tag{6}$$

where N is the system size represented by the number of atoms.

B. t_1-transformed Formulation of CCSD

In the CCSD method, the T cluster operator is truncated to only include single and double excitation contributions. An alternate way to formulate CCSD is to fold the effects of the single excitations back into the Hamiltonian operator.[54] In this alternate formulation

$$E_{\text{CCSD}} = \left\langle \Psi_0 \mid e^{-T_2} \widetilde{H} e^{T_2} \mid \Psi_0 \right\rangle \ , \tag{7}$$

where

$$\widetilde{H} = e^{-T_1} H e^{T_1} \ , \tag{8}$$

$$T_1 = t_i^a E_{ai} \ , \tag{9}$$

$$T_2 = t_{ij}^{ab} E_{ai} E_{bj} \ . \tag{10}$$

In singlet, closed-shell CCSD, E_{ai} can be formulated as

$$E_{ai} = a_a^\dagger a_i + \bar{a}_a^\dagger \bar{a}_i \ , \tag{11}$$

where the barred creation/annihilation operators refer to the beta spin orbitals and non-barred refer to the alpha spin orbitals. The quantity t_i^a is known as the singles amplitude, and t_{ij}^{ab} is known as the doubles amplitude. In the t_1-transformed formalism, the amplitudes are updated through iteratively solving the corresponding residual equations

$$R_i^a = \left\langle \Psi_i^a \mid e^{-T_2} \widetilde{H} e^{T_2} \mid \Psi_0 \right\rangle \ , \tag{12}$$

$$R_{ij}^{ab} = \left\langle \Psi_{ij}^{ab} \mid e^{-T_2} \widetilde{H} e^{T_2} \mid \Psi_0 \right\rangle \ , \tag{13}$$

with

$$t_i^a = t_i^a - \frac{R_i^a}{\epsilon_a - \epsilon_i} \ , \tag{14}$$

$$t_{ij}^{ab} = t_{ij}^{ab} - \frac{R_{ij}^{ab}}{\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j} \ , \tag{15}$$

where ϵ_i and ϵ_a represent orbital energies obtained from the diagonal elements of the Fock operator in the MO basis. The residual update equations are much more simplified compared

to the traditional formulation of CCSD[59] since all terms involving singles excitations no longer arise explicitly. The t_1-transformed formalism of CCSD initially did not see much use after its introduction due to the cost of transforming conventional, four-center ERIs every iteration.[54] However, DePrince et al.[55] showed that using this formalism for CCSD is much more advantageous in the context of using the density-fitting (DF)/resolution-of-the-identity (RI)[60–68] or Cholesky decomposition (CD)[69–71] approximations for the two-electron integrals. In this formalism, the two-electron integrals are approximated as:

$$(pq|rs) \approx (pq|P)(P|Q)^{-1}(Q|rs) \ , \tag{16}$$

where P and Q represent auxiliary basis functions. This can be rewritten as:

$$(pq|rs) \approx B^Q_{pq} B^Q_{rs} \ , \tag{17}$$

where

$$B^Q_{pq} = (Q|P)^{-\frac{1}{2}}(P|pq) \ . \tag{18}$$

We present our working equations based on the formalism of DePrince et al.,[55] with equation 31 in this work reflecting a corrected sign error from the original work. Terms with a single overhead tilde represent t_1-dressed quantities, and their explicit form is defined later.

$$R^{ab}_{ij} = \widetilde{K}^{ab}_{ij} + A^{ab}_{ij} + B^{ab}_{ij} + \hat{P}^{ab}_{ij}\left[\frac{1}{2}C^{ab}_{ij} + C^{ab}_{ji} + D^{ab}_{ij} + E^{ab}_{ij} + G^{ab}_{ij}\right] \ , \tag{19}$$

where

$$\widetilde{K}_{ij}^{ab} = \widetilde{B}_{ai}^{Q}\widetilde{B}_{bj}^{Q} \ , \tag{20}$$

$$A_{ij}^{ab} = t_{ij}^{cd}\widetilde{B}_{ac}^{Q}\widetilde{B}_{bd}^{Q} \ , \tag{21}$$

$$B_{ij}^{ab} = t_{kl}^{ab}\beta_{ij}^{kl} \ , \tag{22}$$

$$C_{ij}^{ab} = -t_{kj}^{bc}\gamma_{ki}^{ac} \ , \tag{23}$$

$$D_{ij}^{ab} = \frac{1}{2}u_{jk}^{bc}\delta_{ik}^{ac} \ , \tag{24}$$

$$E_{ij}^{ab} = t_{ij}^{ac}\widetilde{\widetilde{F}}_{bc} \ , \tag{25}$$

$$G_{ij}^{ab} = -t_{ik}^{ab}\widetilde{\widetilde{F}}_{kj} \ , \tag{26}$$

with

$$\beta_{ij}^{kl} = \widetilde{B}_{ki}^{Q}\widetilde{B}_{lj}^{Q} + t_{ij}^{cd}B_{kc}^{Q}B_{ld}^{Q} \ , \tag{27}$$

$$\gamma_{ki}^{ac} = \widetilde{B}_{ki}^{Q}\widetilde{B}_{ac}^{Q} - \frac{1}{2}t_{li}^{ad}B_{kd}^{Q}B_{lc}^{Q} \ , \tag{28}$$

$$\delta_{ik}^{ac} = (2\widetilde{B}_{ai}^{Q}B_{kc}^{Q} - \widetilde{B}_{ki}^{Q}\widetilde{B}_{ac}^{Q}) + \frac{1}{2}u_{il}^{ad}(2B_{ld}^{Q}B_{kc}^{Q} - B_{lc}^{Q}B_{kd}^{Q}) \ , \tag{29}$$

$$\widetilde{\widetilde{F}}_{bc} = \widetilde{F}_{bc} - u_{kl}^{bd}B_{ld}^{Q}B_{kc}^{Q} \ , \tag{30}$$

$$\widetilde{\widetilde{F}}_{kj} = \widetilde{F}_{kj} + u_{lj}^{cd}B_{kd}^{Q}B_{lc}^{Q} \ . \tag{31}$$

$\hat{P}_{ij}^{ab}$ is a permutation operator and is defined $\hat{P}_{ij}^{ab}(X_{ij}^{ab}) = X_{ij}^{ab} + X_{ji}^{ba}$. The quantity u_{ij}^{ab} is the antisymmetrized doubles amplitude and is defined as $u_{ij}^{ab} = 2t_{ij}^{ab} - t_{ij}^{ba}$. The singles residual takes the form

$$R_i^a = \widetilde{F}_{ai} + A_i^a + B_i^a + C_i^a \ , \tag{32}$$

where

$$A_i^a = u_{ki}^{cd}B_{kc}^{Q}\widetilde{B}_{ad}^{Q} \ , \tag{33}$$

$$B_i^a = -u_{kl}^{ac}\widetilde{B}_{ki}^{Q}B_{lc}^{Q} \ , \tag{34}$$

$$C_i^a = \widetilde{F}_{kc}u_{ik}^{ac} \ . \tag{35}$$

The DF/RI or CD integrals dressed with the singles amplitude take the form[55]

$$\widetilde{B}^Q_{ki} = B^Q_{ki} + B^Q_{ka} t^a_i \ , \tag{36}$$

$$\widetilde{B}^Q_{ia} = B^Q_{ia} \ , \tag{37}$$

$$\widetilde{B}^Q_{ai} = B^Q_{ai} - t^a_k B^Q_{ki} + B^Q_{ab} t^b_i - t^a_k B^Q_{kb} t^b_i \ , \tag{38}$$

$$\widetilde{B}^Q_{ab} = B^Q_{ab} - t^a_k B^Q_{kb} \ , \tag{39}$$

Since B^Q_{ia} does not transform under t_1-dressing (Equation 37), the terms in the singles and doubles amplitudes involving integrals of that type do not need to be dressed. The dressed Fock matrices are, analogously,

$$\widetilde{F}_{ki} = \bar{F}_{ki} + \bar{F}_{ka} t^a_i \ , \tag{40}$$

$$\widetilde{F}_{ia} = \bar{F}_{ia} \ , \tag{41}$$

$$\widetilde{F}_{ai} = \bar{F}_{ai} - t^a_k \bar{F}_{ki} + \bar{F}_{ab} t^b_i - t^a_k \bar{F}_{kb} t^b_i \ , \tag{42}$$

$$\widetilde{F}_{ab} = \bar{F}_{ab} - t^a_k \bar{F}_{kb} \ , \tag{43}$$

where

$$\bar{F}_{rs} = F_{rs} + [2(rs|kc) - (rc|ks)] t^c_k \ , \tag{44}$$

The energy expression, is

$$E_{\mathrm{CCSD}} = (t^{ab}_{ij} + t^a_i t^b_j)[2(ia|jb) - (ib|ja)] \ . \tag{45}$$

C. Perturbative Triples Correction in CCSD(T)

Though CCSD, with its size-extensive treatment of single and double excitation operators, provides a good description of dynamic electron correlation, it is often not sufficient for chemical accuracy.[72–78] Chemical accuracy, in this context, is defined to be a relative energy error of 1 kcal mol^{-1} or lower, compared to either the FCI energy or experimental results. A full treatment of triples (CCSDT) costs, iteratively, $\mathcal{O}(N^8)$. A cheaper way to consider the effect of triples is the perturbative (T) treatment as devised by Raghavachari et al.[5] In restricted, single-reference, closed-shell coupled cluster theory, $E_{(\mathrm{T})}$ can be expressed as[25]

$$E_{(\mathrm{T})} = \frac{1}{3}\frac{(4W_{ijk}^{abc} + W_{ijk}^{bca} + W_{ijk}^{cab})(V_{ijk}^{abc} - V_{ijk}^{cba})}{\epsilon_i + \epsilon_j + \epsilon_k - \epsilon_a - \epsilon_b - \epsilon_c} \ , \tag{46}$$

with

$$W_{ijk}^{abc} = P_L[(ia|bd)t_{kj}^{cd} - (ia|jl)t_{kl}^{cb}] \ , \tag{47}$$

$$V_{ijk}^{abc} = W_{ijk}^{abc} + P_S[t_i^a(jb|kc)] \ , \tag{48}$$

Following the formalism of Lesiuk,[79] we define P_L and P_S, or the "long" and "short" permutation operators as

$$P_L(A_{ijk}^{abc}) = A_{ijk}^{abc} + A_{ikj}^{acb} + A_{jik}^{bac} + A_{jki}^{bca} + A_{kij}^{cab} + A_{kji}^{cba} \ , \tag{49}$$

$$P_S(A_{ijk}^{abc}) = A_{ijk}^{abc} + A_{jik}^{bac} + A_{kij}^{cab} \ . \tag{50}$$

In the (T) formalism, the triples amplitude takes the form

$$t_{ijk}^{abc} = \frac{W_{ijk}^{abc}}{\epsilon_i + \epsilon_j + \epsilon_k - \epsilon_a - \epsilon_b - \epsilon_c} \ , \tag{51}$$

Using the triples amplitude, as well as the permutational symmetry of the energy denominator, one can rewrite the expression for the (T) energy as:

$$E_{(T)} = t_{ijk}^{abc} \cdot (\frac{4}{3}V_{ijk}^{abc} - 2V_{ijk}^{cba} + \frac{2}{3}V_{ijk}^{cab}) \ , \tag{52}$$

In order to reduce memory costs, in our implementation of the DLPNO-CCSD(T) algorithm, the indices are restricted such that $i \leq j \leq k$ (no restriction on the virtual indices). The energy expression can now be rewritten as,

$$\begin{aligned} E_{(T)} =& \frac{t_{ijk}^{abc}}{1 + (\delta_{ij} + \delta_{jk} + \delta_{ik}) + 2\delta_{ij}\delta_{jk}\delta_{ik}} \\ & \times (8V_{ijk}^{abc} - 4V_{ijk}^{bac} - 4V_{ijk}^{acb} - 4V_{ijk}^{cab} + 2V_{ijk}^{bca} + 2V_{ijk}^{cab}) \ . \end{aligned} \tag{53}$$

D. Overview of Domain-Based Pair Natural Orbital (DLPNO)

In this section, we provide a brief overview of all of the different localization techniques involved in the DLPNO approach as defined by Neese et al.[29,34,35,40] For a more comprehensive understanding, the reader is referred to the original papers.

1. Local Molecular Orbitals (LMOs)

To localize the occupied molecular orbitals, one applies a unitary transformation to the Hartree–Fock/SCF molecular orbitals, to limit their spatial extent[80]

$$C^L_{\mu i} = C_{\mu j} U_{ji} \ . \tag{54}$$

The Foster–Boys[81,82] or Pipek–Mezey localization[82,83] approaches can be used to effectively localize the MOs. For all computations presented here, we use the Foster–Boys approach, following the work of Riplinger et al.[40] Localizing molecular orbitals reduces the number of "strongly correlated pairs" of molecular orbitals ij from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. In this context, we define "significantly correlated pairs" to be pairs that need to be treated with MP2 or a higher level of correlation. Otherwise, a dipole estimate[39] is sufficient for a description of non-significantly correlated pairs. In our work, similar to the previous work by Valeev, Neese, and coworkers,[40] we divide our LMO pairs ij into four classes: dipole pairs, semi-canonical MP2 pairs, weak MP2 pairs, and strong pairs. Dipole pairs [which scale $\mathcal{O}(N^2)$] are treated using an inexpensive dipole estimate. Semi-canonical MP2 pairs, scaling $\mathcal{O}(N)$, are treated using semi-canonical MP2 in the projected atomic orbital (PAO) basis, while weak MP2 pairs, scaling $\mathcal{O}(N)$, are treated with full iterative LMP2. The surviving pairs, the strong pairs, scaling $\mathcal{O}(N)$, are treated at the CCSD level. For the (T) correction, triplets ijk are determined from strong pairs and weak MP2 pairs, so the number of relevant triplets is also linear scaling. For clarity, "semi-canonical MP2" is obtained by using the standard (canonical) MP2 energy expression, and the effect of off-diagonal LMO Fock matrix elements that would contribute in the case of non-canonical Hartree-Fock orbitals are neglected. In the case of canonical molecular orbitals, "semi-canonical MP2" is the exact MP2 energy. However, when localized molecular orbitals are used, the full MP2 energy requires an iterative solution.

Though the Foster–Boys or Pipek–Mezey localization procedure is $\mathcal{O}(N^3)$ and determining the dipole pair contribution is $\mathcal{O}(N^2)$, these steps have such a small prefactor that they do not significantly affect the computation time of systems studied in this work.

2. Projected Atomic Orbitals (PAOs)

Compared to localizing the occupied space, localizing the virtual space is challenging. One of the earliest attempts at virtual space localization was through projected atomic orbitals (PAOs).[84] Since the atomic orbital space spans the same subspace as the complete MO space, a complete, localized, and linearly-dependent description of the virtual space can be determined from the atomic orbitals and occupied MO coefficients. PAOs have a more local character compared to canonical virtual molecular orbitals. The following equations represent how PAOs are formed by projecting out the occupied MO space from the complete AO space:

$$C^{\mathrm{PAO}}_{\mu\widetilde{\nu}} = \delta_{\mu\nu} - C^{L}_{\mu i} C^{L}_{\sigma i} S^{\mathrm{AO}}_{\sigma\nu} \ , \tag{55}$$

$$S^{\mathrm{PAO}}_{\widetilde{\mu}\widetilde{\nu}} = C^{\mathrm{PAO}}_{\lambda\widetilde{\mu}} S^{\mathrm{AO}}_{\lambda\sigma} C^{\mathrm{PAO}}_{\sigma\widetilde{\nu}} \ . \tag{56}$$

The C^{PAO} coefficients give the contribution of atomic orbital μ to PAO $\widetilde{\nu}$, and S^{PAO} represents the overlap matrix between two PAOs. Next, the PAOs are normalized

$$C^{\mathrm{PAO}}_{\mu\widetilde{\nu}} = (S^{\mathrm{PAO}}_{\widetilde{\nu}\widetilde{\nu}})^{-\frac{1}{2}} C^{\mathrm{PAO}}_{\mu\widetilde{\nu}} \ . \tag{57}$$

and the PAO overlaps are non-iteratively recomputed using the new PAO coefficients.

One early attempt at creating local-correlation algorithms was by Schütz, Hetzer, and Werner,[85] who used LMOs and PAOs to implement a local version of MP2.[14] In their work, they gave each occupied molecular orbital pair its own set of PAOs, taking advantage of the limited spatial overlap between LMOs and PAOs. The concept of giving every MO pair its own virtual space is a precursor to PNO (pair natural orbital) based algorithms. Werner and coworkers later extended the same framework to CCSD[86] and CCSD(T)[87] methods. In these methods, a set of redundant, linearly-dependent PAOs is assigned to each LMO based on the spatial overlap of the PAO with the LMO. In our work, following Pinski, Riplinger, Valeev, and Neese,[39] the overlap is computed through a measure called the "differential overlap integral" (DOI).

$$\mathrm{DOI}_{i\widetilde{\mu}} = (i\widetilde{\mu}|i\widetilde{\mu})^{\frac{1}{2}} \ . \tag{58}$$

If the value of the integral is greater than a given tolerance $T_{\mathrm{CUT\_DO}}$, then the PAO $\widetilde{\mu}$ is included in the domain of LMO i. The PAOs included in the domain of a pair ij are the union of the PAOs included in the domain of LMO i combined with the PAOs in the domain of LMO j. After the PAOs in the pair domain of pair ij are determined, linear dependencies are removed through an algorithm like Partial Cholesky, and then the resulting space is transformed into a canonical basis (forming a diagonal Fock matrix, and thus orbital energies for these transformed versions of the virtual orbitals for LMO pair ij). The resulting PAOs will be called canonical PAOs.

$$F_{\widetilde{\mu}_{ij}\widetilde{\nu}_{ij}} = \epsilon_{\widetilde{\mu}_{ij}}\delta_{\widetilde{\mu}_{ij}\widetilde{\nu}_{ij}} = X_{\widetilde{\mu}\widetilde{\mu}_{ij}}F^{\mathrm{PAO}}_{\widetilde{\mu}\widetilde{\nu}}X_{\widetilde{\nu}\widetilde{\nu}_{ij}} \ . \tag{59}$$

3. Pair Natural Orbitals (PNOs)

To mitigate the high crossover points associated with using projected atomic orbitals (PAOs), Neese et al. introduced pair natural orbitals (PNOs) for correlated methods such as CEPA,[29] and CCSD.[30] PNOs are eigenvectors of the pair density of a molecular orbital pair ij.

$$D^{ab}_{ij} = \frac{1}{1+\delta_{ij}}[u^{ac}_{ij}t^{bc}_{ij} + u^{ca}_{ij}t^{cb}_{ij}] \ , \tag{60}$$

The pair density can computed through canonical virtual orbitals or PAOs. In their original work, Neese et al.[29] constructed PNOs from canonical virtual orbitals, using amplitudes from a preceding MP2 calculation. In a later work, Riplinger et al.[34] updated their methodology by computing PNOs using canonical PAOs, from *semicanonical* MP2 amplitudes. This is known as the domain-based local pair natural orbital (DLPNO) approach.

$$D^{\widetilde{\mu}_{ij}\widetilde{\nu}_{ij}}_{ij} = X^{\mathrm{PNO},ij}_{\widetilde{\mu}_{ij}a_{ij}}n^{occ,ij}_{a_{ij}}X^{\mathrm{PNO},ij}_{\widetilde{\nu}_{ij}a_{ij}} \ . \tag{61}$$

The eigenvectors, $X^{PNO,ij}_{\widetilde{\mu}_{ij}a_{ij}}$, represent the transformation from canonical PAOs to PNOs, and their eigenvalues $n^{occ,ij}_{a_{ij}}$ represent the occupation numbers corresponding to each pair natural orbital.

The PNOs are then truncated to form a more compact description of the virtual space spanned by each pair ij. In our method, there are three criteria we use for determining

significant PNOs. If any one of the following three criteria is met, then the PNO is considered significant.

- **Occupation Criterion:** All PNOs with an occupation number greater than $T_{\text{CUT\_PNO}}$ will be included. We will dub this the occupation cutoff.
- **Energy Criterion:** Every PNO is included, from highest to lowest occupation number, until the pair energy computed from only those PNOs, as a ratio of the total semicanonical MP2 energy for the pair ij, is greater than $T_{\text{CUT\_ENERGY}}$.
- **Trace Criterion:** Every PNO is included, from highest to lowest occupation number, until the sum of their occupation numbers, divided by the total virtual occupation number sum, is greater than $T_{\text{CUT\_TRACE}}$.

In the demonstrations of the algorithm presented here, we will use tighter cutoffs in order to best capture the effects of non-covalent interations. We present results using $T_{CUT\_PNO} = 10^{-7}$, $T_{\text{CUT\_ENERGY}} = 0.997$, and $T_{\text{CUT\_TRACE}} = 0.999$. The occupation criterion was the original method of truncating PNOs introduced by Riplinger et al. in the ORCA package[47] in the DLPNO-CCSD algorithm.[34] The energy criterion was first introduced by the work of Schwilk et al. in the Molpro package[48] in their PNO-LCCSD algorithm.[41] After the truncated PNO basis is constructed, the truncated PNOs are canonicalized to give orbital energies for the pair ij.

For diagonal pairs ii, a tighter occupation cutoff is used, with the occupation number criterion $T_{\text{CUT\_PNO}}$ scaled by $T_{\text{DIAG\_SCALE}} = 10^{-3}$ when determining significant PNOs. These PNOs are also assigned to the singles amplitudes of orbital i.

For the (T) algorithm, it is possible to build a compact virtual space for LMO triplets ijk, by forming a triplet density,[34] through the average of pair densities of pairs ij, jk, and ik.

$$D_{ijk} = \frac{1}{3}(D_{ij} + D_{jk} + D_{ik}) \ . \tag{62}$$

In our algorithm, a PAO space is first built for triplet ijk by merging the PAO spaces of LMOs i, j, and k, at a looser tolerance $T_{CUT\_DO\_TRIPLES}$ (default 10^{-2}), and then the combined PAO space is canonicalized. Next, the pair densities for ij, jk, and ik are computed using converged LCCSD amplitudes for strong pairs, and LMP2 amplitudes for weak pairs,

and then each projected into the canonical PAO space for triplet ijk. The projected densities are then averaged to form the triplet density, which is subsequently diagonalized to form the triples natural orbitals (TNOs). In this work, only the occupation criterion T_{CUT_TNO} (default 10^{-9}) is used for the selection of TNOs. The use of an analogous energy or trace criterion for triples is not considered but would be an interesting avenue for future research.

4. Local Density Fitting

To reduce the cost of integral computation, in this algorithm, similar to the previous work by Riplinger et al.,[34,40] only a subset of auxiliary basis functions is used, rather than the full set of auxiliary basis functions, in using the DF/RI approximation for two-electron integrals. The Mulliken population of electrons of LMO i for each center A is used to determine local auxiliary function domains.[88]

$$P^i_{\mu\nu} = C^L_{\mu i} S_{\mu\nu} C^L_{\nu i} \ , \tag{63}$$

$$q_{iA} = 2 \sum_{\mu \in A} \sum_{\nu} P^i_{\mu\nu} \cdot \frac{P^i_{\mu\mu}}{P^i_{\mu\mu} + P^i_{\nu\nu}} \ . \tag{64}$$

If q_{iA} for local molecular orbital i on atom A is greater than $T_{\mathrm{CUT\_MKN}}$, then all of the auxiliary basis functions centered on atom A are in the local auxiliary domain of LMO i. Thus, the subset of auxiliary basis functions (Q_{ij}) local to pair ij is the union of the local auxiliary domains on LMO i and LMO j.

III. WORKING EQUATIONS

In this section we present the working equations for our DLPNO-CCSD(T) implementation. We will use our sets of working equations for density-fitted, t_1-dressed CCSD and (T) as presented in the Theory section as a starting point, following the work of DePrince et al.[55] For a baseline derivation, we use these following heuristics:

- The virtual space of singles amplitudes for LMO i uses the diagonal PNOs of pair ii
- The virtual space of doubles amplitudes for LMO pair ij uses PNOs of pair ij

- PNO overlap matrices are used in the event of a mismatch in virtual spaces (defined below)

$$S^{a_{ij}}_{a_{kl}} = X^{\mathrm{PNO},ij}_{\widetilde{\mu}a_{ij}} S^{\mathrm{PAO}}_{\widetilde{\mu}\widetilde{\nu}} X^{\mathrm{PNO},kl}_{\widetilde{\nu}a_{kl}} \ , \tag{65}$$

For example, equation 22 becomes

$$B^{a_{ij}b_{ij}}_{ij} = (S^{a_{ij}}_{a_{kl}} t^{a_{kl}b_{kl}}_{kl} S^{b_{ij}}_{b_{kl}})\beta^{kl}_{ij} \ . \tag{66}$$

as the PNOs of pair kl from the doubles amplitudes need to be transformed into the PNOs of pair ij.

Converting integrals to the PNO basis is less straightforward, and the different ways to formulate integrals from a speed/accuracy trade-off perspective is presented in the next section.

A. Discussion of PNO Projection Error

Integrals can either be directly formed from the PAO basis, or approximated using PNO overlap matrices. For example, integrals of type $(ia_{kl}|jb_{kl})$ can be derived in two ways

$$(ia_{kl}|jb_{kl}) = X^{\mathrm{PNO},kl}_{\widetilde{\mu}a_{kl}} (i\widetilde{\mu}|j\widetilde{\nu}) X^{\mathrm{PNO},kl}_{\widetilde{\nu}b_{kl}} \ , \tag{67}$$

or

$$(ia_{kl}|jb_{kl}) \approx S^{a_{kl}}_{a_{ij}} (ia_{ij}|jb_{ij}) S^{b_{kl}}_{b_{ij}} \ . \tag{68}$$

Using the projection approximation is advantageous in that building and storing integrals of type $(ia_{kl}|jb_{kl})$ is significantly more expensive (requiring an index loop over ij and kl in storage) than projecting integrals of type $(ia_{ij}|jb_{ij})$ (only requiring an index loop over ij). Using the projection approximation, such as in equation 68, is akin to using the PNO basis of ij in a resolution of the identity (RI) operator

$$\sum_{a_{ij}} |a_{ij}\rangle \langle a_{ij}| \approx 1 \ . \tag{69}$$

This is not always a good approximation, as it assumes that the span of the PNOs of pairs ij is close enough to the span of the PNOs of pairs kl. The error resulting from building integrals from the projection approximation is defined as the "projection error."[34,41] The projection error decreases as the PNO cutoff is tightened. Interestingly enough, using the projection approximation does not induce large errors for most terms, even if it is sometimes even used repeatedly. However, for contributions to R_{ij}^{ab} that are linear in t_{ij}^{ab}, as well as certain terms involved in dressing the Fock matrix, the projection approximation cannot be applied without bringing large errors. After extensive experimentation, we have determined a set of working equations that best balance speed and accuracy, derived from the original set of equations presented in this work, transformed to the local basis. For terms that explicitly show four-center integrals, the integrals are first computed from the sparse three-center integrals through local density-fitting and LMO/PAO sparsity[39,40] and stored explicitly in sparse-format as four-index quantities, while for terms that involve three-center integrals, the four-index quantities are never explicitly formed. Select terms in some equations are bolded to ease reader comprehension and highlight design choices that balance accuracy and efficiency.

B. LCCSD Working Equations

First, let us define some integral and amplitude intermediates:

$$\widetilde{t}_i^{a_{kl}} = S_{a_{ii}}^{a_{kl}} t_i^{a_{ii}} \ , \tag{70}$$

$$J_{pq}^{rs} = (pq|rs) \ , \tag{71}$$

$$K_{pq}^{rs} = (pr|qs) \ , \tag{72}$$

$$L_{pq}^{rs} = 2K_{pq}^{rs} - K_{pq}^{sr} \ , \tag{73}$$

$$M_{pq}^{rs} = 2K_{pq}^{rs} - J_{pq}^{rs} \ . \tag{74}$$

For the contributions to the R_{ij}^{ab} residual,

$$\widetilde{K}_{ij}^{a_{ij}b_{ij}} = \widetilde{B}_{a_{ij}i}^{Q_{ij}} \widetilde{B}_{b_{ij}j}^{Q_{ij}} \ , \tag{75}$$

$$A_{ij}^{a_{ij}b_{ij}} = t_{ij}^{c_{ij}d_{ij}} \widetilde{B}_{a_{ij}c_{ij}}^{Q_{ij}} \widetilde{B}_{b_{ij}d_{ij}}^{Q_{ij}} \ , \tag{76}$$

$$B_{ij}^{a_{ij}b_{ij}} = (S_{a_{kl}}^{a_{ij}} t_{kl}^{a_{kl}b_{kl}} S_{b_{kl}}^{b_{ij}}) \beta_{ij}^{kl} \ , \tag{77}$$

$$C_{ij}^{a_{ij}b_{ij}} = -S_{a_{ki}}^{a_{ij}} \gamma_{ki}^{a_{ki}c_{ki}} S_{c_{kj}}^{c_{ki}} t_{kj}^{b_{kj}c_{kj}} S_{b_{kj}}^{b_{ij}} - \boldsymbol{J_{ik}^{a_{ij}c_{kj}} S_{b_{kj}}^{b_{ij}} t_{kj}^{b_{kj}c_{kj}}} \ , \tag{78}$$

$$D_{ij}^{a_{ij}b_{ij}} = \frac{1}{2} S_{a_{ik}}^{a_{ij}} \delta_{ik}^{a_{ik}c_{ik}} S_{c_{jk}}^{c_{ik}} u_{jk}^{b_{jk}c_{jk}} S_{b_{jk}}^{b_{ij}} + \boldsymbol{\frac{1}{2} M_{ik}^{a_{ij}c_{kj}} S_{b_{jk}}^{b_{ij}} u_{jk}^{b_{jk}c_{jk}}} \ , \tag{79}$$

$$E_{ij}^{a_{ij}b_{ij}} = t_{ij}^{a_{ij}c_{ij}} \widetilde{\widetilde{F}}_{b_{ij}c_{ij}} \ , \tag{80}$$

$$G_{ij}^{a_{ij}b_{ij}} = -(S_{a_{ik}}^{a_{ij}} t_{ik}^{a_{ik}b_{ik}} S_{b_{ik}}^{b_{ij}}) \widetilde{\widetilde{F}}_{kj} \ , \tag{81}$$

The intermediates are redefined as

$$\beta_{ij}^{kl} = \widetilde{B}_{ki}^{Q_{ij}} \widetilde{B}_{lj}^{Q_{ij}} + t_{ij}^{c_{ij}d_{ij}} B_{kc_{ij}}^{Q_{ij}} B_{ld_{ij}}^{Q_{ij}} \ , \tag{82}$$

$$\gamma_{ki}^{a_{ki}c_{ki}} = \boldsymbol{-\widetilde{t}_{l}^{a_{ki}} J_{ki}^{lc_{ki}} + \widetilde{t}_{i}^{b_{ki}} J_{kb_{ki}}^{a_{ki}c_{ki}} - \widetilde{t}_{i}^{b_{kl}} K_{kl}^{b_{kl}c_{kl}} S_{c_{kl}}^{c_{ki}} \widetilde{t}_{l}^{a_{ki}}} - \frac{1}{2} S_{a_{li}}^{a_{ki}} t_{li}^{a_{li}d_{li}} S_{d_{kl}}^{d_{li}} K_{kl}^{d_{kl}c_{kl}} S_{c_{kl}}^{c_{ki}} \ , \tag{83}$$

$$\delta_{ik}^{a_{ik}c_{ik}} = \boldsymbol{-\widetilde{t}_{l}^{a_{ik}} M_{ik}^{lc_{ik}} + \widetilde{t}_{i}^{b_{ik}} M_{kb_{ki}}^{c_{ki}a_{ki}} - \widetilde{t}_{i}^{b_{lk}} L_{lk}^{b_{lk}c_{lk}} S_{c_{lk}}^{c_{ki}} \widetilde{t}_{l}^{a_{ik}}} + \frac{1}{2} S_{a_{il}}^{a_{ik}} u_{il}^{a_{il}d_{il}} S_{d_{kl}}^{d_{il}} L_{kl}^{c_{kl}d_{kl}} S_{c_{kl}}^{c_{ik}} \ , \tag{84}$$

$$\widetilde{\widetilde{F}}_{b_{ij}c_{ij}} = \widetilde{F}_{b_{ij}c_{ij}} - S_{b_{kl}}^{b_{ij}} u_{kl}^{b_{kl}d_{kl}} K_{kl}^{c_{kl}d_{kl}} S_{c_{kl}}^{c_{ij}} \ , \tag{85}$$

$$\widetilde{\widetilde{F}}_{kj} = \widetilde{F}_{kj} + (S_{c_{lj}}^{c_{lk}} u_{lj}^{c_{lj}d_{lj}} S_{d_{lj}}^{d_{lk}}) K_{lk}^{c_{lk}d_{lk}} \ . \tag{86}$$

Similarly, the singles residuals in the diagonal PNO basis take the form

$$R_{i}^{a_{ii}} = \widetilde{F}_{a_{ii}i} + A_{i}^{a_{ii}} + B_{i}^{a_{ii}} + C_{i}^{a_{ii}} \ , \tag{87}$$

where

$$A_{i}^{a_{ii}} = u_{ki}^{c_{ki}d_{ki}} K_{ka_{ki}}^{c_{ki}d_{ki}} S_{a_{ki}}^{a_{ii}} - \widetilde{t}_{l}^{a_{ii}} u_{ki}^{c_{ki}d_{ki}} S_{c_{kl}}^{c_{ki}} K_{kl}^{c_{kl}d_{kl}} S_{d_{kl}}^{d_{ki}} \ , \tag{88}$$

$$B_{i}^{a_{ii}} = -S_{a_{kl}}^{a_{ii}} u_{kl}^{a_{kl}c_{kl}} [K_{kl}^{ic_{kl}} + \widetilde{t}_{i}^{b_{kl}} K_{kl}^{b_{kl}c_{kl}}] \ , \tag{89}$$

$$C_{i}^{a_{ii}} = S_{a_{ik}}^{a_{ii}} u_{ik}^{a_{ik}c_{ik}} \widetilde{F}_{kc_{ik}} \ . \tag{90}$$

Here are the relevant DF integrals dressed with the singles amplitudes

$$\widetilde{B}^{Q_{ij}}_{ki} = B^{Q_{ij}}_{ki} + B^{Q_{ij}}_{ka_{ij}} \widetilde{t}^{a_{ij}}_{i} \ , \tag{91}$$

$$\widetilde{B}^{Q_{ij}}_{a_{ij}i} = B^{Q_{ij}}_{a_{ij}i} - \widetilde{t}^{a_{ij}}_{k} B^{Q_{ij}}_{ki} + B^{Q_{ij}}_{a_{ij}b_{ij}} \widetilde{t}^{b_{ij}}_{i} - \widetilde{t}^{a_{ij}}_{k} B^{Q_{ij}}_{kb_{ij}} \widetilde{t}^{b_{ij}}_{i} \ , \tag{92}$$

$$\widetilde{B}^{Q_{ij}}_{a_{ij}b_{ij}} = B^{Q_{ij}}_{a_{ij}b_{ij}} - \widetilde{t}^{a_{ij}}_{k} B^{Q_{ij}}_{kb_{ij}} \ . \tag{93}$$

For the dressed Fock matrices, since we are not using canonical MOs, we derived these equations from the work of Werner and coworkers[41]

$$\widetilde{F}_{ij} = \overline{F}_{ij} + \overline{F}_{ic_{jj}} t^{c_{jj}}_{j} \ , \tag{94}$$

$$\widetilde{F}_{ia_{ij}} = S^{a_{ij}}_{a_{ik}} L^{a_{ik}c_{ik}}_{ik} \widetilde{t}^{c_{ik}}_{k} \ , \tag{95}$$

$$\widetilde{F}_{a_{ii}i} = \overline{F}_{a_{ii}i} - \widetilde{t}^{a_{ii}}_{k} \overline{F}_{ki} + \overline{F}_{a_{ii}b_{ii}} t^{b_{ii}}_{i} - \widetilde{t}^{a_{ii}}_{k} \overline{F}_{kb_{ii}} t^{b_{ii}}_{i} \ , \tag{96}$$

$$\widetilde{F}_{a_{ij}b_{ij}} = \overline{F}_{a_{ij}b_{ij}} - \widetilde{t}^{a_{ij}}_{k} \overline{F}_{kb_{ij}} \ , \tag{97}$$

where

$$\overline{F}_{ij} = F_{ij} + [2J^{kc_{ij}}_{ij} - K^{kc_{ij}}_{ji}] \widetilde{t}^{c_{ij}}_{k} \ , \tag{98}$$

$$\overline{F}_{ia_{kl}} = [2B^{Q_{kl}}_{ia_{kl}} B^{Q_{kl}}_{mc_{kl}} - B^{Q_{kl}}_{ic_{kl}} B^{Q_{kl}}_{ma_{kl}}] \widetilde{t}^{c_{kl}}_{m} \ , \tag{99}$$

$$\overline{F}_{a_{ii}i} = [2B^{Q_{ii}}_{a_{ii}i} B^{Q_{ii}}_{kc_{ii}} - B^{Q_{ii}}_{a_{ii}c_{ii}} B^{Q_{ii}}_{ki}] \widetilde{t}^{c_{ii}}_{k} \ , \tag{100}$$

$$\overline{F}_{a_{ij}b_{ij}} = \epsilon_{a_{ij}} \delta_{a_{ij}b_{ij}} + [\boldsymbol{2B^{Q_{ij}}_{a_{ij}b_{ij}} B^{Q_{ij}}_{kc_{ij}} - B^{Q_{ij}}_{a_{ij}c_{ij}} B^{Q_{ij}}_{kb_{ij}}}] \boldsymbol{\widetilde{t}^{c_{ij}}_{k}} \ . \tag{101}$$

In our formalism, we dress our Fock matrices directly in the PNO space, rather than the PAO space, as is done by Werner and coworkers in their PNO-LCCSD algorithm in Molpro[41]. In our working equations for DLPNO-CCSD, the terms have a slightly different structure than the original set of working equations from canonical t_1-transformed DF-CCSD. One notable modification is the expansion of the t_1-dressed integrals and the removal of the leading two-virtual integrals in equations 83 and 84 from their canonical counterparts and expressing their contributions explicitly in equations 78 and 79. This is done since PNO projection errors are greatest in terms containing linear doubles amplitude contributions to the doubles residual. In addition, even though the R^{ab}_{ij} and R^{a}_{i} residuals are only updated over strong pairs, weak MP2 pairs also contribute to the residual of strong pairs. Because of this, the DF integrals from equations 91–93 are only constructed over strong pairs to save

memory costs. Therefore, the t_1-dressed integrals are expanded explicitly in other parts of the working equations as well, in equations 78, 79, 88 and 89. Certain Fock matrix contributions also have a unique form. Equation 95 is not constructed from equation 99, since the former is looped over strong pairs and the latter is looped over both strong pairs and weak LMP2 pairs. Finally, equation 101 is built with the explicit DF integrals, and not through projecting integrals of type $(kc_{kk}|a_{kk}b_{kk})$ (errors too large) or building integrals of type $(kc_{kk}|a_{ij}b_{ij})$ (too expensive to store) as a trade off between speed and accuracy. For the sake of absolute clarity, and reproducibility, the energy expression we used was

$$E_{\mathrm{LCCSD}} = (t_{ij}^{a_{ij}b_{ij}} + \widetilde{t}_i^{a_{ij}}\widetilde{t}_j^{b_{ij}})L_{ij}^{a_{ij}b_{ij}} \ . \tag{102}$$

where the singles and doubles residual updates can be computed in three equivalent, equally valid formalisms

$$t_i^{a_{ii}} -= \frac{R_i^{a_{ii}}}{\epsilon_{a_{ij}} - F_{ii}} \ , \tag{103}$$

$$t_i^{a_{ii}} -= \frac{R_i^{a_{ii}}}{F_{a_{ij}a_{ij}} - F_{ii}} \ , \tag{104}$$

$$t_i^{a_{ii}} -= \frac{R_i^{a_{ii}}}{\widetilde{F}_{a_{ij}a_{ij}} - \widetilde{F}_{ii}} \ , \tag{105}$$

$$t_{ij}^{a_{ij}b_{ij}} -= \frac{R_{ij}^{a_{ij}b_{ij}}}{\epsilon_{a_{ij}} + \epsilon_{b_{ij}} - F_{ii} - F_{jj}} \ , \tag{106}$$

$$t_{ij}^{a_{ij}b_{ij}} -= \frac{R_{ij}^{a_{ij}b_{ij}}}{F_{a_{ij}a_{ij}} + F_{b_{ij}b_{ij}} - F_{ii} - F_{jj}} \ , \tag{107}$$

$$t_{ij}^{a_{ij}b_{ij}} -= \frac{R_{ij}^{a_{ij}b_{ij}}}{\widetilde{F}_{a_{ij}a_{ij}} + \widetilde{F}_{b_{ij}b_{ij}} - \widetilde{F}_{ii} - \widetilde{F}_{jj}} \ . \tag{108}$$

As shown above, both dressed and undressed Fock matrices can be used for the energy denominators for the residual updates. At convergence, the computed energies will be equivalent. This is presented for reader comprehension and to reduce the confusion between the equations as presented in other works.

C. (T) Working Equations

Most of the triples equations are trivially carried over from the canonical (T) equations. Still, we present the W and V intermediates as they are computed in the TNO basis in our code for the sake of completeness. The two-electron integrals are never stored in the (T) algorithm, but computed on-the-fly from sparse three-center integrals and contracted into the intermediates as needed:

$$W_{ijk}^{a_{ijk}b_{ijk}c_{ijk}} = P_L[(ia_{ijk}|b_{ijk}d_{ijk})[S_{c_{kj}}^{c_{ijk}} t_{kj}^{c_{kj}d_{kj}} S_{d_{kj}}^{d_{ijk}}] - (jl|kc_{ijk})S_{a_{il}}^{a_{ijk}} t_{il}^{a_{il}b_{il}} S_{b_{il}}^{b_{ijk}}] \ , \tag{109}$$

$$V_{ijk}^{a_{ijk}b_{ijk}c_{ijk}} = W_{ijk}^{a_{ijk}b_{ijk}c_{ijk}} + P_S[t_i^{a_{ii}} S_{a_{ii}}^{a_{ijk}} (jb_{ijk}|kc_{ijk})] \ , \tag{110}$$

Since we are in the LMO basis where the Fock matrix is not diagonal, we need to iteratively solve for the full (T) energy.[44,45] The approximation where the triples amplitudes are not corrected for off-diagonal LMO Fock matrix elements is called the semi-canonical (T0) approximation.[34,44] In this work, we will *not* use the semicanonical (T0) approximation in any of our test cases, as it is known to be problematic for certain systems.[44,45] The triples amplitudes are iteratively updated as

$$\begin{aligned} R_{ijk}^{a_{ijk}b_{ijk}c_{ijk}} &= W_{ijk}^{a_{ijk}b_{ijk}c_{ijk}} - t_{ijk}^{a_{ijk}b_{ijk}c_{ijk}}(\epsilon_{a_{ijk}} + \epsilon_{b_{ijk}} + \epsilon_{c_{ijk}} - f_{ii} - f_{jj} - f_{kk}) \\ &- \sum_{l\neq i} f_{il} t_{ljk}^{a_{ljk}b_{ljk}c_{ljk}} S_{a_{ljk}b_{ljk}c_{ljk}}^{a_{ijk}b_{ijk}c_{ijk}} - \sum_{l\neq j} f_{jl} t_{ilk}^{a_{ilk}b_{ilk}c_{ilk}} S_{a_{ilk}b_{ilk}c_{ilk}}^{a_{ijk}b_{ijk}c_{ijk}} - \sum_{l\neq k} f_{kl} t_{ijl}^{a_{ijl}b_{ijl}c_{ijl}} S_{a_{ijl}b_{ajl}c_{ajl}}^{a_{ijk}b_{ijk}c_{ijk}} \ , \end{aligned} \tag{111}$$

$$t_{ijk}^{a_{ijk}b_{ijk}c_{ijk}} - = \frac{R_{ijk}^{a_{ijk}b_{ijk}c_{ijk}}}{\epsilon_{a_{ijk}} + \epsilon_{b_{ijk}} + \epsilon_{c_{ijk}} - F_{ii} - F_{jj} - F_{kk}} \ . \tag{112}$$

In the iterative (T) algorithm, unlike in the semi-canonical (T0) algorithm, all three $t_{ijk}^{a_{ijk}b_{ijk}c_{ijk}}$, $W_{ijk}^{a_{ijk}b_{ijk}c_{ijk}}$, and $V_{ijk}^{a_{ijk}b_{ijk}c_{ijk}}$ intermediates need to be stored. The T3 amplitudes need to be stored and updated, with the W intermediate stored to compute the residual, and the V intermediate stored to compute the energy (Equation 53). Though the memory costs for storing these quantities may appear to be excessive due to its $\mathcal{O}(n_{ijk}n_{pno\_ijk}^3)$ scaling, the memory requirements are often similar or less than the memory requirements for the

preceding DLPNO-CCSD computation. This is because the TNO space used in the iterative (T) computation is smaller than the one used in the semi-canonical (T0) computation, and additionally, applying the index restriction $i \leq j \leq k$ reduces costs by a factor of 6. There is an option to perform disk I/O with these triples intermediates, through the keyword WRITE_TRIPLES, but we do not anticipate the average user needing to use this. For example, in a uracil dimer computation in the cc-pVDZ basis set (TightPNO convergence), the total storage requirement for the DLPNO-CCSD computation is around 8.40 GiB, while the requirements for the triples intermediates is around 2.36 GiB each (total of around 7.09 GiB).

IV. IMPLEMENTATION DETAILS

A. CCSD Algorithm Details

Much of our prescreening to classify pairs in the DLPNO-CCSD algorithm is derived from the original work of Valeev, Neese, and coworkers.[39,40] We first screen out the dipole pairs based on the $T_{\text{CUT\_DO\_ij}}$ and $T_{\text{CUT\_PRE}}$ cutoffs, the overlap and energy criteria used to ensure that LMOs i and j are non-overlapping. Next, we determine the semicanonical MP2 pairs as all non-dipole pairs with an energy contribution less than $T_{\text{CUT\_PAIRS\_MP2}}$. The latter is done using an initial prescreening procedure that us looser in cutoffs.[40] In the third step, we recompute the semicanonical LMP2 amplitudes for surviving pairs (non-dipole or semicanonical) through the refined prescreening procedure,[40] and compute PNOs for the LMP2 procedure using $T_{\text{CUT\_PNO\_MP2}}$, $T_{\text{CUT\_ENERGY\_MP2}}$, and $T_{\text{CUT\_TRACE\_MP2}}$. Pairs are then divided into weak LMP2 pairs or strong pairs based on their energy through $T_{\text{CUT\_PAIRS}}$. Next, LMP2 energies and amplitudes are computed for both weak and strong pairs using the tighter PNOs. Finally, the PNOs are recomputed at looser cutoffs from converged LMP2 amplitudes with $T_{\text{CUT\_PNO}}$, $T_{\text{CUT\_ENERGY}}$, and $T_{\text{CUT\_TRACE}}$, with only the strong pair amplitudes being updated in the LCCSD iterations; the weak MP2 pair amplitudes are saved for the (T) algorithm. The total DLPNO-CCSD energy thus contains contributions from all four pair classes, as well as a PNO truncation correction[39] from strong pairs and weak MP2 pairs. The PNO truncation is computed as the difference between the semicanonical LMP2 energy computed in the initial, tighter PNO basis ($T_{\text{CUT\_PNO\_MP2}}$) and the PAO basis,

summed with the difference between the LMP2 energy computed using the tighter PNOs used for LMP2 and the looser PNOs ($T_{\text{CUT\_PNO}}$) used for LCCSD.

$$E_{\text{DLPNO-CCSD}} = E_{\text{LCCSD[strong pairs]}} + \Delta E_{\text{LMP2[weak MP2 pairs]}} + \Delta E_{\text{SC-LMP2[semicanonical MP2 pairs]}} + \Delta E_{\text{dipole[dipole pairs]}} + \Delta E_{\text{PNO[strong pairs + weak MP2 pairs]}} \cdot \quad (113)$$

We present values for the most relevant parameters used in the DLPNO-CCSD algorithm, as reported in this section and throughout this work, in Table I. The presented values are those corresponding to the **TightPNO**/NormalPNO convergence settings for our code. We will present all our results with TightPNO.

TABLE I. Parameters of our DLPNO-CCSD algorithm for TightPNO and NormalPNO settings

Parameter	**Description**	**TightPNO Value**	**NormalPNO Value**
$T_{\text{CUT\_PNO}}$	LCCSD PNO occupation criterion	$\mathbf{10^{-7}}$	3.33×10^{-7}
$T_{\text{CUT\_ENERGY}}$	LCCSD PNO energy criterion	**0.997**	0.99
$T_{\text{CUT\_TRACE}}$	LCCSD PNO trace criterion	**0.999**	0.99
$T_{\text{CUT\_PNO\_MP2}}$	LMP2 PNO occupation criterion	$\mathbf{10^{-9}}$	3.33×10^{-9}
$T_{\text{CUT\_ENERGY\_MP2}}$	LMP2 PNO energy criterion	**0.999**	0.997
$T_{\text{CUT\_TRACE\_MP2}}$	LMP2 PNO trace criterion	**0.9999**	0.999
$T_{\text{DIAG\_SCALE}}$	Scale of $T_{\text{CUT\_PNO}}$ for diagonal pairs	**0.001**	0.001
$T_{\text{CUT\_DO}}$	LMO/PAO DOI criterion for pair domains	**0.005**	0.01
$T_{\text{CUT\_DO\_ij}}$	LMO/LMO DOI criterion for dipole pairs	$\mathbf{10^{-5}}$	10^{-5}
$T_{\text{CUT\_PRE}}$	Dipole energy cutoff for pair screening	$\mathbf{10^{-7}}$	10^{-6}
$T_{\text{CUT\_PAIRS}}$	Strong/weak pair cutoff	$\mathbf{10^{-5}}$	10^{-4}
$T_{\text{CUT\_PAIRS\_MP2}}$	Weak/semicanonical pair cutoff	$\mathbf{10^{-6}}$	10^{-6}
$T_{\text{CUT\_MKN}}$	Local density fitting Mullikan tolerance	$\mathbf{10^{-3}}$	10^{-3}

B. Triples Algorithm Procedures

We model our triples algorithm based on a combination of features from DLPNO-CCSD(T) in ORCA[44] and PNO-LCCSD(T) in Molpro,[45] in order to optimize speed and accuracy. First, we use the triples prescreening algorithm as presented by Ma and Werner.[45] We first compute the semicanonincal (T0) energy for each possible triplet ijk, derived from combinations of pairs ij, jk, and ik, at least one of which is a strong pair, at a weaker TNO tolerance ($T_{\text{CUT\_TNO\_PRE}}$). All triplets ijk for which the absolute value of the energy is lower than ($T_{\text{CUT\_TRIPLES\_PRE}}$) are screened out and not further considered, but the sum of their energy contributions is saved and accounted for. We will term these triplets that did not survive the prescreening as the "screened triplets." The rest of the algorithm is derived from the work of Neese and coworkers,[44] where the TNOs of the surviving triplets are then recomputed at a tighter tolerance (T_{CUT_TNO}), in order to obtain a more accurate semi-canonical (T0) energy. The TNOs are then recomputed at a looser tolerance for the iterative (T) step to reduce the cost of storing triples amplitudes and intermediates. To this end, the energies of the triplets are sorted and the approximately 20% of triplets that account for at least 90% of the semi-canonical (T0) energy are deemed "strong triplets,"[44] and the rest deemed "weak triplets," For the "strong triplets," the TNOs are recomputed at a looser tolerance $T_{\text{CUT\_TNO}} \cdot T_{\text{STRONG\_SCALE}}$ and the "weak triplets" at $T_{\text{CUT\_TNO}} \cdot T_{\text{WEAK\_SCALE}}$ for the full iterative (T) algorithm. The final (T) energy is as follows:

$$E_{\text{DLPNO-(T)}} = \sum_{i \le j \le k} E^{ijk}_{\text{(T0)}}[T_{\text{CUT\_TNO}}] + \sum_{ijk \in \text{strong triplets}} (E^{ijk}_{\text{(T)}} - E^{ijk}_{\text{(T0)}})[T_{\text{CUT\_TNO}} \times T_{\text{STRONG\_SCALE}}]$$
$$+ \sum_{ijk \in \text{weak triplets}} (E^{ijk}_{\text{(T)}} - E^{ijk}_{\text{(T0)}})[T_{\text{CUT\_TNO}} \times T_{\text{WEAK\_SCALE}}] + \sum_{ijk \in \text{screened triplets}} \Delta E^{ijk}_{\text{(T0)}}[T_{\text{CUT\_TNO\_PRE}}] \,. \tag{114}$$

The default values for the triples parameters are presented in Table II (values are the same across all PNO convergences)

TABLE II. Default values of our DLPNO-(T) parameters for all settings

Parameter	Description	Default Value
$T_{\text{CUT\_TNO}}$	TNO occupation criterion	$\mathbf{10^{-9}}$
$T_{\text{CUT\_TNO\_PRE}}$	TNO occupation criterion in "screened triplets"	$\mathbf{10^{-7}}$
$T_{\text{CUT\_TRIPLES\_PRE}}$	"Screened triplets" energy cutoff	$\mathbf{10^{-7}}$
$T_{\text{CUT\_DO\_TRIPLES}}$	LMO/PAO DOI criterion for triples domains	$\mathbf{10^{-2}}$
$T_{\text{CUT\_MKN\_TRIPLES}}$	Local density fitting tolerance for triples	$\mathbf{10^{-2}}$
$T_{\text{STRONG\_SCALE}}$	Iterative (T) strong triplet $T_{\text{CUT\_TNO}}$ scaling	**10**
$T_{\text{WEAK\_SCALE}}$	Iterative (T) weak triplet $T_{\text{CUT\_TNO}}$ scaling	**100**

C. Discussion on Memory Usage

For the integrals, the quantities are stored in RAM based on their form in the working equations. For example, $K_{ij}^{a_{ij}b_{ij}}$ and $J_{ij}^{a_{ij}b_{ij}}$ are stored as a list of matrices (of dimension $n_{pno_{ij}} \cdot n_{pno_{ij}}$) indexed by the index of pair ij, while the non-projected integrals $J_{ik}^{a_{ij}c_{kj}}$ and $K_{ik}^{a_{ij}c_{kj}}$ (used to form $M_{ik}^{a_{ij}c_{kj}}$) are stored as a nested list of matrices of dimension $n_{pno_{ij}} \cdot n_{pno_{kj}}$, with the first index being the LMO pair ij, and the second index being the LMO index k_{ij}, which represents all LMOs k such that ki and kj are both strong or weak pairs. The DF integrals $B_{k_{ij}a_{ij}}^{Q_{ij}}$ $B_{a_{ij}b_{ij}}^{Q_{ij}}$ are similarly stored as a nested list, with the first index being the LMO pair ij, and the second index being the auxiliary function index q_{ij} in the domain of ij for each set of matrices. The non-projected J and K integrals, as well as the DF integrals, need only to be stored for strong pairs. There is an option to store the expensive $B_{k_{ij}a_{ij}}^{Q_{ij}}$ $B_{a_{ij}b_{ij}}^{Q_{ij}}$ integrals to disk, through the keywords WRITE_QIA_PNO, and WRITE_QAB_PNO. Algorithms that explain how the non-projected integrals ($J_{ik}^{a_{ij}c_{kj}}$ and $K_{ik}^{a_{ij}c_{kj}}$) are formed are included in the appendix.

All the dressed Fock matrices are computed and stored, while only the dressed integrals $\widetilde{B}_{a_{ij}i}^{Q_{ij}}$ (list of matrices of dimension $n_{aux_{ij}} \cdot n_{pno_{ij}}$ indexed by ij) and $\widetilde{B}_{k_{ij}i}^{Q_{ij}}$ (list of matrices of dimension $n_{aux_{ij}} \cdot n_{lmo_{ij}}$ indexed by ij) are stored, due to the reduced memory requirements).

For most computations, the largest contributor to memory cost is the $S_{b_{kl}}^{a_{ij}}$ PNO overlap integrals, which are stored as a nested list indexed by LMO pair ij and then LMO pair kl. Since these quantities are only used for the $B_{ij}^{a_{ij}b_{ij}}$ and $E_{ij}^{a_{ij}b_{ij}}$ intermediates (Equations 77

and 80), to reduce the cost of the PNO overlap integrals, we have developed a semi-direct low memory overlap algorithm for PNO matrices, where PNO overlaps of the form $S_{b_{kk}}^{a_{ij}}$, as well as $S_{b_{kj}}^{a_{ij}}$ and $S_{b_{ik}}^{a_{ij}}$ are stored in RAM, and $S_{b_{kl}}^{a_{ij}}$ is computed through the semi-direct algorithm. This can be toggled by the setting LOW_MEMORY_OVERLAP. The algorithm for this is presented in the appendix.

D. Computational Details

For all correlated computations, the frozen-core approximation is used. Unless otherwise stated, TightPNO convergence is used for all computations. All timings are performed on 16 cores of an Intel Xeon 6136 CPU (3.0 GHz processing speed with 1 TB of RAM) unless otherwise stated. Typical quantum chemistry algorithms benefit from access to large amounts of RAM, especially for computations on the largest systems considered here. Yet, DLPNO-CCSD(T) computations can also be carried out with the resources available to the average user, who can find our code in a development branch of the freely available Psi4 program.[51]

V. RESULTS

A. Dimer Interaction Energies

We first present the results of our DLPNO-CCSD and (T) algorithms on the S22 data set,[89] consisting of 22 dimers of sizes ranging from water-water to adenine-thymine. The interactions of the S22 dimers can be primarily hydrogen-bonded (HB), dispersion-dominated (DD), or mixed influence (MX).[90] For a fair comparison of our algorithm, we compared the results to canonical DF-CCSD and DF-CCSD(T) as implemented in Psi4, using the same RI basis sets for the correlated computations.[55] For these tests, we use the cc-pVDZ, jun-cc-pVDZ, and cc-pVTZ orbital basis sets.[91,92] For the unfamiliar reader, the jun-cc-pVDZ basis set adds a set of diffuse functions for all heavy atoms up to shell $l-1$, where l represents the highest angular momentum shell, from the cc-pVDZ basis. All computations are performed with the counterpoise (CP) correction.[93] The results comparing DLPNO-CCSD to canonical DF-CCSD are presented in Table III, and comparisons between DLPNO-CCSD(T) to DF-CCSD(T) are shown in Table IV. We present the results of the interaction energy errors

in aggregate for each basis set, as well as for each interaction type for each basis set.

As shown in the tables, at the TightPNO convergence, the algorithm is accurate enough to yield interaction energy errors of MAE 0.06 kcal mol^{-1} or less in every basis set for CCSD, and 0.10 kcal mol^{-1} or less for CCSD(T). For the NormalPNO convergence, the MAE is typically on the order of 0.2 – 0.3 kcal mol^{-1} for both CCSD and CCSD(T), with the MAE for the dispersion-dominated complexes being larger, around 0.5 – 0.6 kcal mol^{-1} (detailed results are presented in the supporting information). Generally speaking, DLPNO-CCSD/DLPNO-CCSD(T) is the most accurate when modeling dimers bound by mixed influences, versus modeling hydrogen-bonded or dispersion-dominated dimers. It is also shown that adding diffuse functions (as in the case of the jun-cc-pVDZ basis set) helps reduce the errors associated with modeling dispersion interactions relative to DF-CCSD(T). For dispersion-dominated complexes, it is *not* recommended to use NormalPNO convergnce, which is consistent with the findings of Kallay and coworkers.[36]

TABLE III. TightPNO DLPNO-CCSD error statistics compared to canonical DF-CCSD reference (kcal mol^{-1}). jun-cc-pVDZ augments the cc-pVDZ basis by adding diffuse functions for all heavy atoms up to shell $l_{\max} - 1$.

Basis Set (Count)	**ME**	**MAE**	**RMSE**	**Std Dev**	**Min**	**Max**
cc-pVDZ (22)	0.017	**0.063**	0.088	0.086	**–0.168**	**0.175**
HB (7)	0.046	0.047	0.066	0.048	–0.006	0.132
DD (8)	0.033	0.105	0.128	0.124	–0.168	0.175
MX (7)	–0.029	0.031	0.037	0.024	–0.074	0.007
jun-cc-pVDZ (22)	0.005	**0.058**	0.083	0.083	**–0.243**	**0.122**
HB (7)	0.044	0.054	0.068	0.052	–0.033	0.115
DD (8)	–0.011	0.089	0.119	0.119	–0.243	0.122
MX (7)	–0.015	0.025	0.031	0.027	–0.054	0.020
cc-pVTZ (22)	0.021	**0.052**	0.078	0.075	**–0.200**	**0.141**
HB (7)	0.054	0.054	0.071	0.046	–0.001	0.128
DD (8)	0.013	0.086	0.110	0.109	–0.200	0.141
MX (7)	–0.003	0.011	0.015	0.014	–0.030	0.020

TABLE IV. TightPNO DLPNO-CCSD(T) error statistics compared to canonical DF-CCSD(T) reference (kcal mol^{-1}). jun-cc-pVDZ augments the cc-pVDZ basis by adding diffuse functions for all heavy atoms up to shell $l_{\max} - 1$.

Basis Set (Count)	**ME**	**MAE**	**RMSE**	**Std Dev**	**Min**	**Max**
cc-pVDZ (22)	0.079	**0.084**	0.132	0.105	**–0.025**	**0.312**
HB (7)	0.116	0.116	0.145	0.088	0.008	0.280
DD (8)	0.115	0.115	0.170	0.126	0.002	0.312
MX (7)	0.003	0.018	0.020	0.020	–0.025	0.032
jun-cc-pVDZ (22)	0.070	**0.079**	0.113	0.089	**–0.054**	**0.270**
HB (7)	0.117	0.117	0.139	0.076	0.028	0.270
DD (8)	0.072	0.089	0.127	0.105	–0.054	0.229
MX (7)	0.020	0.028	0.046	0.042	–0.027	0.116
cc-pVTZ (22)	0.100	**0.100**	0.141	0.099	**0.003**	**0.296**
HB (7)	0.140	0.140	0.171	0.099	0.017	0.296
DD (8)	0.116	0.116	0.162	0.113	0.003	0.282
MX (7)	0.042	0.042	0.054	0.034	0.005	0.099

B. Potential Energy Surfaces

Next, we examined the potential energy surface along the dissociation of a uracil dimer pair, from the S66x8 data set of Rezac et al.[94] The dimer is displaced along an axis parallel to the two hydrogen bonds, at 0.9, 0.95, 1.0, 1.05, 1.1, 1.25, 1.5, and 2.0 times the average distance of the two hydrogen bonds (R_{eq} = 1.805 Å). As shown in Figure 1, the DLPNO-CCSD/DLPNO-CCSD(T) dissociation curves match their respective canonical references, and DLPNO-CCSD(T) also effectively captures the (T) correlation effects. The computation is performed in the cc-pVTZ basis set, with counterpoise (CP) correction, with DF-CCSD/(T) used for the canonical reference. For canonical DF-CCSD(T), the computations take around 8 hours per dimer on 48 cores of the Intel Xeon 6136 specified in the Computational Details, while for DLPNO-CCSD(T), it takes around 35 minutes per dimer, with the computation involving more separated dimers being faster (39 minutes for $0.9R_{eq}$,

26 minutes for $2.0R_{eq}$).

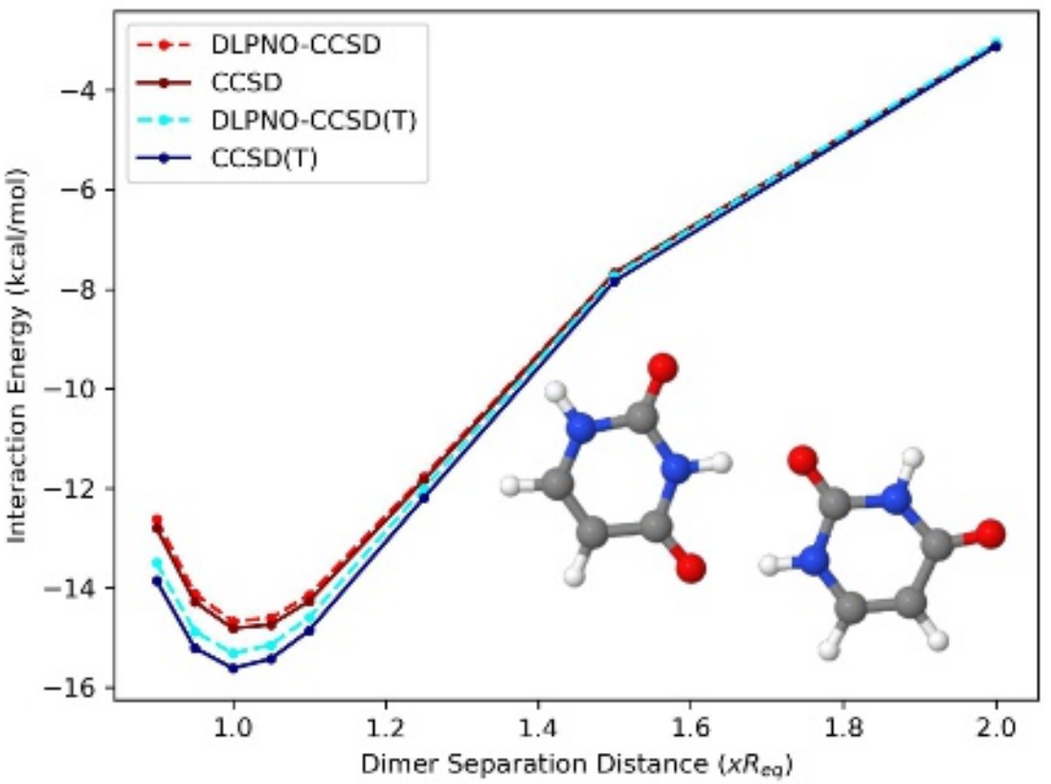

FIG. 1. Counterpoise (CP) corrected cc-pVTZ interaction energies for uracil dimer base pair along the (frozen monomer) dissociation curve

We also plot the errors of DLPNO-CCSD/(T) with respect to the canonical DF-CCSD/(T) results along the dissociation curve, as shown in Figure 2. As expected, the errors decrease as the system becomes more well separated, since the magnitude of the inter-molecular correlation decreases. The CCSD and CCSD(T) errors are 0.17 and 0.36 kcal mol^{-1}, respectively, at $0.9R_{eq}$ separation, and at $2.0R_{eq}$, the errors decrease to 0.01 and 0.08 kcal mol^{-1}. The relatively higher magnitude of errors around equilibrium and closer contacts is to be expected due to the local approximations, and are controllable through tightening the parameters.

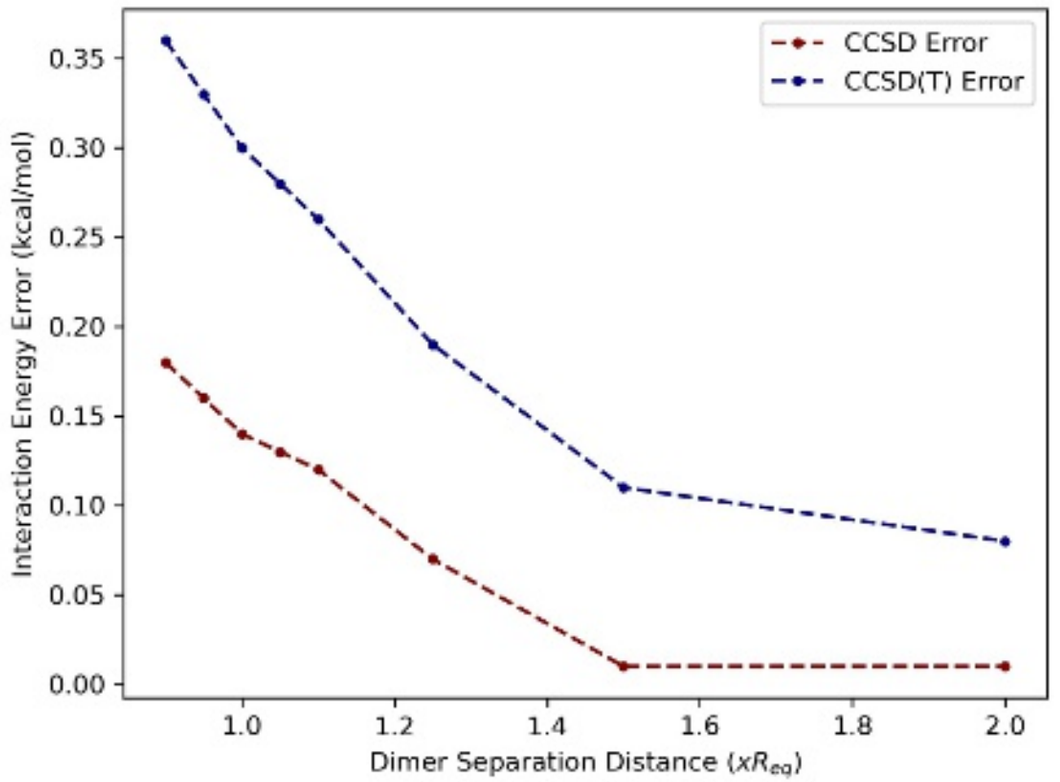

FIG. 2. Errors along the frozen monomer dissociation curve for uracil dimer base pair, defined as the canonical DF-CCSD/(T) interaction energy subtracted from the DLPNO-CCSD/(T) result

C. Large Water Cluster Conformation Energies

For a more rigorous test of the accuracy of our DLPNO-CCSD(T) implementation, we consider some of the larger systems for which canonical CCSD(T) results are available: conformers of $(H_2O)_{16}$ and $(H_2O)_{17}$, in the aug-cc-pVTZ basis, from the work of Xantheas and coworkers.[56] In their work, they computed canonical CCSD(T) reference values using supercomputing resources at the massive ORNL Leadership Computing Facility. They used the CRAY XT5 partition, containing a total of 18,684 compute nodes and 224,208 processing cores with more than 300 TB of memory.[56] In contrast, all of our computations were run on a single computing node with 32 CPU cores and 450 GB of RAM. Xantheas and coworkers presented results for five different conformers of $(H_2O)_{16}$ and two different conformers of $(H_2O)_{17}$. There are four possible conformers of $(H_2O)_{17}$, but not all their computations

finished since they ran out of computing time. Table V presents the relative conformational energies for $(H_2O)_{16}$ and $(H_2O)_{17}$ as computed by Xantheas and coworkers, by our DLPNO-CCSD(T) algorithm, and by the DLPNO-CCSD(T) algorithm in ORCA. For ORCA, we used their default TightPNO cutoff values (different than ours), for version 5.0.4. For completeness, we also include results from the work of Bates, Tschumper, and coworkers[57] who developed a highly accurate many-body expansion method to obtain absolute energies for the conformers close to the canonical CCSD(T) values, called the "CCSD(T):MP2 3-body:many-body" method, which means that all monomer, dimer, and trimer contributions are treated at the CCSD(T) level, and the other contributions are treated with MP2. We will refer to that as "3b:mb" as shorthand.

$$E_{\text{3b:mb/CCSD(T):MP2}} = E_{\text{MP2}}(\text{all}) + \sum_A E_{\text{CCSD(T)}}(A) - E_{\text{MP2}}(A) + \sum_{AB} \Delta E_{\text{CCSD(T)}}(AB) - \Delta E_{\text{MP2}}(AB) + \sum_{ABC} \Delta E_{\text{CCSD(T)}}(ABC) - \Delta E_{\text{MP2}}(ABC) \ . \quad (115)$$

The conformation energies are computed with respect to the lowest energy conformer for each series of conformers. All three approximate methods correctly identify the lowest energy conformer (albeit in ORCA, boat-a and 4444-a have nearly identical energies). Compared to the DLPNO-CCSD(T) algorithm in ORCA, the DLPNO-CCSD(T) algorithm we implemented gives closer answers to the canonical CCSD(T) value for the conformation energies in all cases. We believe this to be the case because of our use of additional criteria (energy and occupation) for PNO selection, as well as a tighter $T_{\text{DIAG\_SCALE}}$ compared to ORCA (0.001 compared to 0.03). This is evident in the boat-a conformation for $(H_2O)_{16}$, where the number of PNOs for our pairs range from 11-149 (average of 44), while ORCA ranges from 6-73 (average of 25). Additionally, our criterion for the initial consideration of triplets (2 weak pairs vs 1 in ORCA) allow our code to recover more triples energy compared to ORCA. For the boat-a configuration, our DLPNO-CCSD absolute energy is 0.3 kcal mol^{-1} lower, while the (T) contribution is 0.4 kcal mol^{-1} lower, leading to a net difference of around 0.7 kcal mol^{-1}.

Surprisingly, even though the 3b:mb method gives more accurate absolute energies than our DLPNO-CCSD(T) method, the two methods reproduce conformation energies of comparable accuracy. This shows that with a tight enough tolerance, local correlation methods can capture subtle higher-order, many-body electron correlation effects. The absolute energies

are included in the supplementary material.

TABLE V. Relative conformation energies (kcal mol^{-1}) using canonical CCSD(T) from the work of Yoo et al.,[56] compared with DLPNO-CCSD(T) as implemented in this paper and the implementation in ORCA,[44] as well as the 3b:mb MP2/CCSD(T) method of Bates et al.[57]

isomer	$\Delta\mathbf{E_{canon}}$	$\Delta\mathbf{E_{PSI4}}$	$\Delta\mathbf{E_{ORCA}}$	$\Delta\mathbf{E_{3b:mb}}$
$\mathbf{(H_2O)_{16}}$				
boat-a	**0.25**	0.35	0.00	0.36
boat-b	**0.42**	0.51	0.15	0.60
antiboat	**0.51**	0.63	0.27	0.67
4444-a (abab)	**0.00**	0.00	0.00	0.00
4444-b (aabb)	**0.54**	0.52	0.59	0.57
$\mathbf{(H_2O)_{17}}$				
sphere	**0.00**	0.00	0.00	0.00
522'5	**0.71**	0.77	0.39	0.77
441'44	X	0.79	0.78	1.10
L-Shape	X	1.49	1.34	1.55

D. Timings and Scaling

For timings, we tested our code on a growing series of three-dimensional water and benzene clusters (geometries available in the supporting information), as shown in Figures 3 and 4. We performed our tests in the cc-pVDZ, jun-cc-pVDZ, and cc-pVTZ basis sets. In these figures, we present log-log plots, using our timings to calculate the empirical scaling of our algorithm. We used a log-log regression of the walltime (in minutes) compared to the system size (by number of basis functions) to perform our analysis, in order to fit a function of the form $t = a \cdot n^b$, where the t is the run-time, a is the pre-factor, n the system size, and b the computational scaling. The results of our analysis are presented in Figures 3 and 4. For each system, across all basis sets, the empirical scalings are all below cubic scaling, with non-diffuse basis sets scaling quadratic or less. Even though the linear scaling regime has not been achieved yet with our system sizes, the observed scaling show a drastic improvement from the seventh power scaling of canonical CCSD(T). Due to the three-dimensional nature of these systems, we do not expect the onset of linear scaling to occur until later, as opposed to previous tests on linear alkanes in ORCA.[40] The steps that are formally higher scaling, such as molecular orbital localization, and dipole pair prescreening, have such a low prefactor that they do not affect timings at all, and none of the new features we incorporated in this algorithm, such as the t_1-transformation of integrals and Fock matrices, scale more than linearly. The results of the analysis performed for the DLPNO-CCSD as well as the DLPNO-(T) components of the computation are presented in the supplementary material.

Finally, to assess the limits of the capabilities of our algorithm, we tested our code on an insulin peptide hormone, as show in Figure 5. The geometry was obtained from the work of Bykov et al.,[58] and is also presented in the supplementary material. For this computation, we used the def2-SV(P) basis set of Weigend et al.,[95] with 6458 basis functions. In Table VI, we present the DLPNO-CCSD(T) correlation energy, as well as its various contributions, at both the NormalPNO and TightPNO convergences. In Table VII, we present timings for the most important portions of the computation (excluding SCF). For these sets of timings, we used 32 CPU cores on a single 2nd generation AMD EPYC Rome (2.9 GHz processing speed, 2000 GB RAM) as part of the Sapelo2 computing cluster at the University of Georgia. The computation is performed completely in-core (excluding SCF). Finally, in Table VIII, we present local domain information at both levels of convergence. As shown in Table VI,

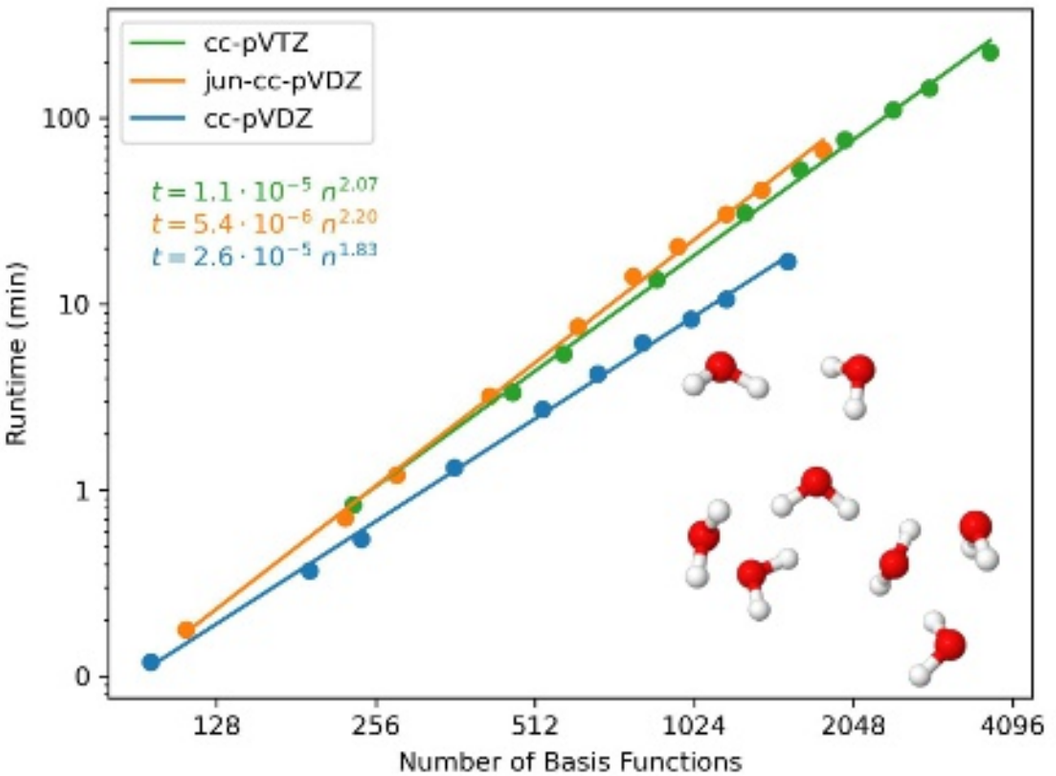

FIG. 3. Water cluster timings and empirical scaling with the DLPNO-CCSD(T) method. System sizes range from 4–64 water molecules, presented for each basis set.

NormalPNO recovers about 99.96% of the TightPNO DLPNO-CCSD correlation energy, and 99.94% of the overall DLPNO-CCSD(T) correlation energy.

Although the LCCSD strong pair correlation energy at the NormalPNO convergence is significantly less than the LCCSD strong pair energy at the TightPNO convergence, the LMP2 weak pair correction makes up the majority of that difference. This highlights the importance of the contribution of weak pairs in DLPNO-CCSD(T). In the timings on Table VII, the TightPNO computation takes about 3 times as long as the NormalPNO computation, with DLPNO-CCSD taking much longer due to significantly larger PNO domain sizes for the tighter criterion. For this computation, we used a preliminary version of the LOW_MEMORY_OVERLAP algorithm, where only the PNO overlap matrices of the form $S^{a_{ij}}_{b_{kj}}$ and $S^{a_{ij}}_{b_{ik}}$ are stored in RAM, and all other types are computed on the fly. This is signif-

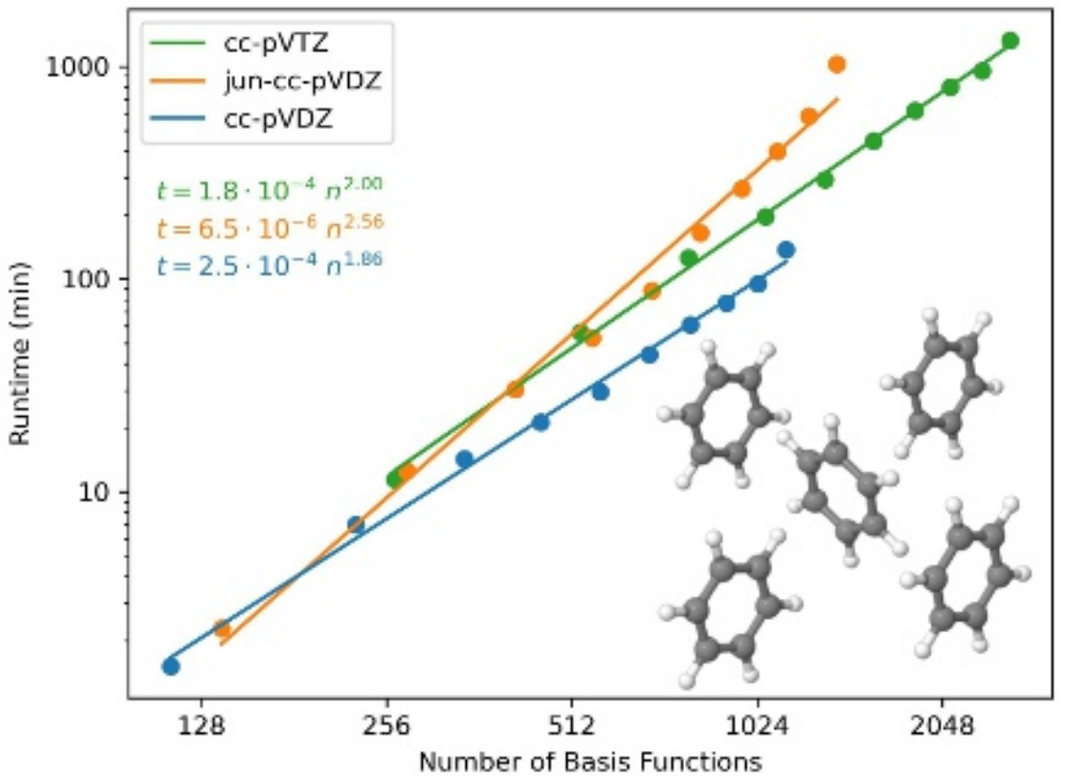

FIG. 4. Benzene cluster timings and empirical scaling with the DLPNO-CCSD(T) method. System sizes range from 1–10 benzene molecules, presented for each basis set.

icantly less efficient than the current implementation (as described earlier in Section IV), so this led to a the disproportionate amount of time spent in the LCCSD iterations. The time spent in the LCCSD iterations would be significantly less with the current code. One fact to highlight is that the dipole correction, though nominally scaling $\mathcal{O}(N^2)$, is far from being a bottleneck in either computation. The triples contribution becomes less of a bottleneck at tighter PNO convergences, due to the efficiency of our triples prescreening algorithm, as highlighted in Table VIII. Table VIII also highlights the locality of the pair and triplet domains in a large system like insulin, with both the virtual and auxiliary domains being significantly less than the span of the entire molecule.

FIG. 5. 3D Structure of insulin (787 atoms)

TABLE VI. Energy information for insulin/def2-SVP at NormalPNO and TightPNO convergence (mEh, unless otherwise stated)

	NormalPNO	TightPNO	Diff.	Diff. (kcal mol^{-1})
DLPNO-CCSD(T) Correlation Energy	–62393.798	–62433.591	–39.8	–25.0
DLPNO-CCSD Contribution	–60265.340	–60289.967	–24.6	–15.5
LCCSD Correlation Energy	–58868.909	–59952.074	–1083.2	–679.7
Weak Pair Contribution	–1248.806	–253.246	995.6	624.7
Semicanonical Contribution	–45.786	–54.920	–9.1	–5.7
Dipole Pair Correction	–23.966	–8.017	15.9	10.0
PNO Truncation Correction	–77.873	–21.711	56.2	35.2
DLPNO-(T) Contribution	–2128.458	–2143.624	–15.2	–9.5
DLPNO-(T0) Energy	–1999.977	–2011.485	–11.5	–7.2
Iterative (T) Contribution	–118.583	–119.840	–1.3	–0.8
Prescreened Triplets Correction	–9.897	–12.299	–2.4	–1.5

TABLE VII. Timings for insulin using DLPNO-CCSD(T) algorithm at NormalPNO and TightPNO

	NormalPNO	TightPNO
Total Wall Time (s)	44906 (0.52 days)	128090 (1.48 days)
DLPNO-CCSD	**66%**	**83%**
Orbital Localization/Sparsity Prep	5%	2%
Dipole Pair Correction	0%	0%
Semicanonical MP2 Pair Correction	0%	0%
LMO/PAO DF Ints	16%	7%
PNO Formation and PNO-LMP2	1%	1%
PNO Overlap Integrals	0%	0%
Integral PNO Transformation	6%	7%
Local CCSD Iterations	38%	66%
DLPNO-(T)	**34%**	**17%**
TNO Formation	6%	2%
Semicanonical (T0)	23%	12%
Iterative (T)	6%	3%

TABLE VIII. Domain information for insulin/def2-SVP at NormalPNO and TightPNO convergence

	NormalPNO	TightPNO
Orbital Information		
Atoms	787	787
Basis Functions	6458	6458
Frozen Core Orbitals	429	429
Active Core Orbitals	1117	1117
Virtual Orbitals	4912	4912
Auxiliary Basis Functions (RI)	24872	24872
Pair Information		
Total LMO pairs (non-unique)	1247689	1247689
Dipole pairs	996002 (79.8%)	857900 (68.8%)
Semicanonical LMP2 pairs	147500 (11.8%)	284386 (22.8%)
Weak Pairs	87590 (7.0%)	64954 (5.2%)
Strong Pairs	16597 (1.3%)	40449 (3.2%)
NAUX per pair	597	598
PAOs per pair	255	348
PNOs per pair (LMP2)	33	60
PNOs per pair (LCCSD)	16	30
Triplet Information		
Total LMO triplets (unique)	232901202	232901202
Initial Triplets	601572 (0.3%)	955435 (0.4%)
Final Triplets	292805 (0.1%)	296016 (0.1%)
TNOs per triplet (Prescreening)	26	22
TNOs per triplet (T0)	52	60
TNOs per triplet (T)	30	32

VI. CONCLUSION

In this work, we have presented an open-source DLPNO-CCSD(T) algorithm, and demonstrated that it can accurately model non-covalent interactions. The deviations for relative energies compared to canonical CCSD(T) are typically on the order of 0.1 kcal mol^{-1} or less with our given set of parameters (at the TightPNO convergence). Our emphasis on accuracy has resulted in some modifications to previously published local PNO based CCSD(T) algorithms, such that the resulting code appears to provide improved accuracy when using "TightPNO" cutoff parameters. We have also shown our code to be competitive in accuracy compared to many-body expansion methods based on canonical CCSD(T). Even when taking advantage of our code's high accuracy, we have been able to keep many of the useful properties associated with the original DLPNO-CCSD(T) algorithm,[34,35,40,44] such as its low scaling and increased efficacy with larger basis sets. This code is now publicly available to view and execute in a development branch of Psi4, and will be widely available in a future release of the Psi4 software. In the future, we plan to test and optimize our code to study even larger systems than those presented in this paper, as well as explore additional ways to reduce projection errors without compromising the efficiency of our algorithm. Combined with advances in computing technology such as with massively parallel computing[96–112] and GPUs,[113–120] an open-source version of DLPNO-CCSD(T) will also allow the development of local coupled-cluster codes which take advantage of these new hardware developments, enabling coupled-cluster calculations on much larger systems than previously imaginable.

VII. SUPPLEMENTARY MATERIAL

The geometries for insulin, as well as the water and benzene clusters used for the scaling tests are available in the supplementary material. We have also included NormalPNO error statistics for the S22 test set, absolute energies for the water clusters from Xantheas and coworkers, as well as scaling analyses for the DLPNO-CCSD and DLPNO-(T) portions of the computation for the water and benzene clusters. We have additionally included (DLPNO)-CCSD/(T) relative energies for each of the S22 dimers, the uracil dimer, and wall times for each of the scaling tests.

ACKNOWLEDGEMENTS

AJ and CDS would like to acknowledge Jose Madriaga for discussions on the working equations in the PNO basis and Jonathon Misiewicz for helpful discussions on non-orthogonal LMO Fock matrices. AJ also acknowledges Dr. Gregory Tschumper for useful discussions on the large water clusters. AJ, JMT, and CDS would like to thank Dr. Susi Lehtola for helpful comments during the revision process of the paper. AJ, JMT, and HFS gratefully acknowledge financial support from the U.S. Department of Energy, Basic Energy Sciences, Chemistry Division Computational and Theoretical Chemistry (CTC) Grant DE-SC0018164. ZLG and CDS also gratefully acknowledge financial support from the National Science Foundation Grant CHE-1955940. DP was supported by a fellowship from The Molecular Sciences Software Institute under NSF grant CHE-2136142.

DATA AVAILABILITY

The data that supports the findings of this study are available with the article and its supplementary material. The source code is available at `https://github.com/andyj10224/psi4/tree/dlpno_ccsd_t_jan_24`.

Appendix A: Computation and Storage of Non-Projected Integrals

In the following algorithms, both in this section and the next section, for user clarity, algorithms are presented with restricted index $i \leq j$ over all strong pairs to increase efficiency. lmopair_to_paos[ij] represents all PAOS $\widetilde{\mu}_{ij}$ that are in the domain of ij (determined by $T_{\mathrm{CUT\_DO}}$), lmopair_to_ribfs[ij] represents all auxiliary basis functions on atoms in the Mulliken fitting domain of pair ij (determined by $T_{\mathrm{CUT\_MKN}}$), while lmopair_to_lmos[ij] represents all LMOs k, such that ik and kj are both strong or weak pairs.

Algorithm 1 Linear-scaling Computation of $K_{ik}^{a_{ij}b_{kj}}$ - Used in Equation 79 (as part of $M_{ik}^{a_{ij}b_{kj}}$)

for ij `in` strong_pairs **do** ▷ Loop over all strong ij pairs, $i \leq j$

  $(P_{ij}|i\widetilde{\nu}_{ij}) \leftarrow (P|m\widetilde{\nu})[\text{lmopair\_to\_ribfs}[ij], i, \text{lmopair\_to\_paos}[ij]]$ ▷ Get slice from $(P|m\widetilde{\nu})$

  $A_{ia_{ij}}^{Q_{ij}} \leftarrow (Q_{ij}|P_{ij})^{-1}(P_{ij}|i\widetilde{\nu}_{ij})X_{\widetilde{\nu}_{ij}a_{ij}}^{PNO}$

  if $i \neq j$ **then** ▷

    $(P_{ij}|j\widetilde{\nu}_{ij}) \leftarrow (P|m\widetilde{\nu})[\text{lmopair\_to\_ribfs}[ij], j, \text{lmopair\_to\_paos}[ij]]$ ▷ Get slice from $(P|m\widetilde{\nu})$

    $A_{ja_{ij}}^{Q_{ij}} \leftarrow (Q_{ij}|P_{ij})^{-1}(P_{ij}|j\widetilde{\nu}_{ij})X_{\widetilde{\nu}_{ij}a_{ij}}^{PNO}$

  end if

  for Q_{ij} `in` lmopair_to_ribfs$[ij]$ **do**

    for k_{ij} `in` lmo_pairs_to_lmos$[ij]$ **do**

      $\overline{K}_{k_{ij}\widetilde{\nu}_{kj}} \leftarrow (Q|m\widetilde{\nu})[Q_{ij}, k_{ij}, \text{lmo\_pair\_to\_lmos}[kj]]$ ▷ Get slice from $(Q|m\widetilde{\nu})$

      $\overline{K}_{k_{ij}b_{kj}} \leftarrow \overline{K}_{k_{ij}\widetilde{\nu}_{kj}}X_{\widetilde{\nu}_{kj}b_{kj}}^{PNO}$

      $K_{ik}^{a_{ij}b_{kj}} \mathrel{+}= A_{ia_{ij}}^{Q_{ij}}\overline{K}_{k_{ij}b_{kj}}$

      if $i \neq j$ **then**

        $\overline{K}_{k_{ij}\widetilde{\nu}_{ij}} \leftarrow (Q|m\widetilde{\nu})[Q_{ij}, k_{ij}, \text{lmo\_pair\_to\_lmos}[ik]]$ ▷ Get slice from $(Q|m\widetilde{\nu})$

        $\overline{K}_{k_{ij}b_{ik}} \leftarrow \overline{K}_{k_{ij}\widetilde{\nu}_{ik}}X_{\widetilde{\nu}_{ik}b_{ik}}^{PNO}$

        $K_{jk}^{a_{ij}b_{ik}} \mathrel{+}= A_{ja_{ij}}^{Q_{ij}}\overline{K}_{k_{ij}b_{ik}}$ ▷ Form jk analog

      end if

    end for

  end for

end for

Algorithm 2 Linear-scaling Computation of $J_{ik}^{a_{ij}b_{kj}}$ - Used in Equations 78 and 79 (as part of $M_{ik}^{a_{ij}b_{kj}}$)

for ij `in` strong_pairs **do** ▷ Loop over all strong ij pairs, $i \leq j$

  pair_ext_domain ← List[int] ▷ Form extended domain for pairs

  for k_{ij} `in` lmo_pairs_to_lmos[ij] **do**

    $k \leftarrow$ lmo_pairs_to_lmos[ij][k_{ij}]

    pair_ext_domain = pair_ext_domain $\cup$ lmo_to_paos[k]

  end for

  $(P_{ij}|ik_{ij}) \leftarrow (P|mn)[\text{lmopair\_to\_ribfs}[ij], i, \text{lmopair\_to\_lmos}[ij]]$ ▷ Get slice from $(P|mn)$

  $A_{ik_{ij}}^{Q_{ij}} \leftarrow (Q_{ij}|P_{ij})^{-1}(P_{ij}|ik_{ij})$

  if $i \neq j$ **then**

    $(P_{ij}|jk_{ij}) \leftarrow (P|mn)[\text{lmopair\_to\_ribfs}[ij], j, \text{lmopair\_to\_lmos}[ij]]$ ▷ Get slice from $(P|mn)$

    $A_{jk_{ij}}^{Q_{ij}} \leftarrow (Q_{ij}|P_{ij})^{-1}(P_{ij}|jk_{ij})$

  end if

  for Q_{ij} `in` lmopair_to_ribfs[ij] **do**

    $\overline{J}_{\widetilde{\mu}_{ij}\widetilde{\nu}_{ij_{\text{ext}}}} \leftarrow (Q|\widetilde{\mu}\widetilde{\nu})[Q_{ij}, \text{lmopair\_to\_paos}[ij], \text{pair\_ext\_domain}]$ ▷ Get slice from $(Q|\widetilde{\mu}\widetilde{\nu})$

    $\overline{J}_{a_{ij}\widetilde{\nu}_{ij_{\text{ext}}}} \leftarrow X_{\widetilde{\mu}_{ij}a_{ij}}^{PNO}\overline{J}_{\widetilde{\mu}_{ij}\widetilde{\nu}_{ij_{\text{ext}}}}$

    for k_{ij} `in` lmo_pairs_to_lmos[ij] **do**

      $\overline{J}_{a_{ij}\widetilde{\mu}_{kj}} \leftarrow \overline{J}_{a_{ij}\widetilde{\nu}_{ij_{\text{ext}}}}[\text{All}, \text{pair\_ext\_domain} \cap \text{lmo\_pair\_to\_paos}[kj]]$

      $\overline{J}_{a_{ij}b_{kj}} \leftarrow \overline{J}_{a_{ij}\widetilde{\mu}_{kj}}X_{\widetilde{\mu}_{kj}b_{kj}}^{PNO}$

      $J_{ik}^{a_{ij}b_{kj}}$ += $A_{ik_{ij}}^{Q_{ij}}\overline{J}_{a_{ij}b_{kj}}$

      if $i \neq j$ **then**

        $\overline{J}_{a_{ij}\widetilde{\mu}_{ik}} \leftarrow \overline{J}_{a_{ij}\widetilde{\nu}_{ij_{\text{ext}}}}[\text{All}, \text{pair\_ext\_domain} \cap \text{lmo\_pair\_to\_paos}[ik]]$

        $\overline{J}_{a_{ij}b_{ik}} \leftarrow \overline{J}_{a_{ij}\widetilde{\mu}_{ik}}X_{\widetilde{\mu}_{ik}b_{ik}}^{PNO}$

        $J_{jk}^{a_{ij}b_{ik}}$ += $A_{jk_{ij}}^{Q_{ij}}\overline{J}_{a_{ij}b_{ik}}$ ▷ Form jk analog

      end if

    end for

  end for

end for

Appendix B: Low Memory PNO Overlap Algorithm

Algorithm 3 Semi-direct algorithm for forming $S_{b_{kl}}^{a_{ij}}$, and relavent contractions

for ij `in` strong_pairs **do** ▷ Loop over all strong ij pairs, $i \leq j$

  pair_ext_domain ← List[int] ▷ Form extended domain for pairs

  for k_{ij}, l_{ij} `in` lmo_pairs_to_lmos[ij] × lmo_pairs_to_lmos[ij] **do**

    $k, l \leftarrow$ lmo_pairs_to_lmos[ij][k_{ij}], lmo_pairs_to_lmos[ij][l_{ij}]

    $kl \leftarrow$ lmo_pair_index[k][l]

    if $kl \in$ strong_pairs ∪ weak_pairs **then**

      pair_ext_domain = pair_ext_domain ∪ lmo_pairs_to_paos[kl]

    end if

  end for

  $\mathbf{S_{ij}} \leftarrow$ submatrix_rows_and_columns($\mathbf{S^{PAO}}$, lmo_pair_to_paos[ij], pair_ext_domain)

  $\mathbf{S'_{ij}} \leftarrow \mathbf{X_{ij}^{PNO}} \mathbf{S_{ij}}$ ▷ Transform first index of overlap matrix to PNO space of ij

  $B_{ij}^{a_{ij}b_{ij}} \leftarrow 0$ ▷ Initialize $B_{ij}^{a_{ij}b_{ij}}$

  $\widetilde{\widetilde{F}}_{b_{ij}c_{ij}} \leftarrow \widetilde{F}_{b_{ij}c_{ij}}$ ▷ Equation 85a

  for k_{ij}, l_{ij} `in` lmo_pairs_to_lmos[ij] × lmo_pairs_to_lmos[ij] **do**

    $k, l \leftarrow$ lmo_pairs_to_lmos[ij][k_{ij}], lmo_pairs_to_lmos[ij][l_{ij}]

    $kl \leftarrow$ lmo_pair_index[k][l]

    if $kl \in$ strong_pairs ∪ weak_pairs **then**

      $\mathbf{S''_{ij}} \leftarrow$ submatrix_columns($\mathbf{S'_{ij}}$, pair_ext_domain ∩ lmo_pair_to_paos[kl])

      $S_{b_{kl}}^{a_{ij}} \leftarrow (S''_{ij})_{a_{ij}\widetilde{\nu}_{kl}} X^{PNO}_{\widetilde{\nu}_{kl}b_{kl}}$

      $B_{ij}^{a_{ij}b_{ij}} \mathrel{+}= (S_{a_{kl}}^{a_{ij}} T_{kl}^{a_{kl}b_{kl}} S_{b_{kl}}^{b_{ij}}) \beta_{ij}^{kl}$ ▷ Equation 77

      $\widetilde{\widetilde{F}}_{b_{ij}c_{ij}} \mathrel{-}= S_{b_{kl}}^{b_{ij}} u_{kl}^{b_{kl}d_{kl}} K_{kl}^{c_{kl}d_{kl}} S_{c_{kl}}^{c_{ij}}$ ▷ Equation 85b

    end if

  end for

  $E_{ij}^{a_{ij}b_{ij}} \leftarrow t_{ij}^{a_{ij}c_{ij}} \widetilde{\widetilde{F}}_{b_{ij}c_{ij}} + \widetilde{\widetilde{F}}_{a_{ij}c_{ij}} t_{ij}^{c_{ij}b_{ij}}$ ▷ Equation 80, second term added to account for P_{ij}^{ab}

  $R_{ij}^{a_{ij}b_{ij}} \mathrel{+}= B_{ij}^{a_{ij}b_{ij}} + E_{ij}^{a_{ij}b_{ij}}$ ▷ Add contributions to doubles residual

  if $i \neq j$ **then**

    $R_{ji}^{b_{ij}a_{ij}} \mathrel{+}= B_{ji}^{b_{ij}a_{ij}} + E_{ji}^{b_{ij}a_{ij}}$ ▷ Add relevant contributions for ji

  end if

end for

REFERENCES

[1]T. D. Crawford and H. F. Schaefer, Rev. Comp. Chem. **14**, 33 (2007).

[2]R. J. Bartlett and M. Musial, Rev. Mod. Phys. **79**, 291 (2007).

[3]C. D. Sherrill and H. F. Schaefer, Advances in Quantum Chemistry **34**, 143 (1999).

[4]C. J. Cramer, *Essentials of Computational Chemistry* (2002) pp. 191–232.

[5]K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, Chem. Phys. Lett. **157**, 479 (1989).

[6]J. F. Stanton, Chem. Phys. Lett. **281**, 130 (1997).

[7]R. J. Bartlett, J. Watts, S. Kucharski, and J. Noga, Chem. Phys. Lett. **165**, 513 (1990).

[8]J. D. Watts, I. Cernusak, J. Noga, R. J. Bartlett, J. Bauschlicher, Charles W., T. J. Lee, A. P. Rendell, and P. R. Taylor, J. Chem. Phys. **93**, 8875 (1990).

[9]G. E. Scuseria and T. J. Lee, J. Chem. Phys. **93**, 5851 (1990).

[10]L. Gyevi-Nagy, M. Kállay, and P. R. Nagy, J. Chem. Theory Comput. **16**, 366 (2020).

[11]P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).

[12]W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).

[13]W. Dawson, A. Degomme, M. Stella, T. Nakajima, L. E. Ratcliff, and L. Genovese, Wiley Interdiscip. Rev. Comput. Mol. Sci. **12**, e1574 (2022).

[14]C. Møller and M. S. Plesset, Phys. Rev. **46**, 618 (1934).

[15]D. Cremer, Wiley Interdiscip. Rev. Comput. Mol. Sci. **1**, 509 (2011).

[16]R. M. Parrish, Y. Zhao, E. G. Hohenstein, and T. J. Martínez, J. Chem. Phys. **150**, 164118 (2019).

[17]E. G. Hohenstein, Y. Zhao, R. M. Parrish, and T. J. Martínez, J. Chem. Phys. **151**, 164121 (2019).

[18]E. G. Hohenstein, B. S. Fales, R. M. Parrish, and T. J. Martínez, J. Chem. Phys. **156**, 054102 (2022).

[19]M. Lesiuk, J. Chem. Phys. **156**, 064103 (2022).

[20]R. M. Parrish, E. G. Hohenstein, T. J. Martínez, and C. D. Sherrill, J. Chem. Phys. **137**, 224106 (2012).

[21]E. G. Hohenstein, R. M. Parrish, and T. J. Martínez, J. Chem. Phys. **137**, 044103 (2012).

[22]E. G. Hohenstein, R. M. Parrish, C. D. Sherrill, and T. J. Martínez, J. Chem. Phys. **137**, 221101 (2012).

[23]R. M. Parrish, C. D. Sherrill, E. G. Hohenstein, S. I. L. Kokkila, and T. J. Martínez, J. Chem. Phys. **140**, 181102 (2014).

[24]T. G. Kolda and B. W. Bader, SIAM Review **51**, 455 (2009).

[25]A. Jiang, J. M. Turney, and H. F. Schaefer, III, J. Chem. Theory Comput. **19**, 1476 (2023).

[26]S. Li, J. Ma, and Y. Jiang, J. Comput. Chem. **23**, 237 (2002).

[27]S. Li, J. Shen, W. Li, and Y. Jiang, J. Chem. Phys. **125**, 074109 (2006).

[28]W. Li, P. Piecuch, J. R. Gour, and S. Li, J. Chem. Phys. **131**, 114109 (2009).

[29]F. Neese, F. Wennmohs, and A. Hansen, J. Chem. Phys. **130**, 114108 (2009).

[30]F. Neese, A. Hansen, and D. G. Liakos, J. Chem. Phys. **131**, 064103 (2009).

[31]W. Li and P. Piecuch, J. Phys. Chem. A **114**, 8644 (2010).

[32]Z. Rolik and M. Kállay, J. Chem. Phys. **135**, 104111 (2011).

[33]Z. Rolik, L. Szegedy, I. Ladjánszki, B. Ladóczki, and M. Kállay, J. Chem. Phys. **139**, 094105 (2013).

[34]C. Riplinger and F. Neese, J. Chem. Phys. **138**, 034106 (2013).

[35]C. Riplinger, B. Sandhoefer, A. Hansen, and F. Neese, J. Chem. Phys. **139**, 134101 (2013).

[36]D. G. Liakos, M. Sparta, M. K. Kesharwani, J. M. L. Martin, and F. Neese, J. Chem. Theory Comput. **11**, 1525 (2015).

[37]H.-J. Werner, G. Knizia, C. Krause, M. Schwilk, and M. Dornbach, J. Chem. Theory Comput. **11**, 484 (2015).

[38]Q. Ma and H.-J. Werner, J. Chem. Theory Comput. **11**, 5291 (2015).

[39]P. Pinski, C. Riplinger, E. F. Valeev, and F. Neese, J. Chem. Phys. **143**, 034108 (2015).

[40]C. Riplinger, P. Pinski, U. Becker, E. F. Valeev, and F. Neese, J. Chem. Phys. **144**, 024109 (2016).

[41]M. Schwilk, Q. Ma, C. Köppl, and H.-J. Werner, J. Chem. Theory Comput. **13**, 3650 (2017).

[42]Q. Ma, M. Schwilk, C. Köppl, and H.-J. Werner, J. Chem. Theory Comput. **13**, 4871 (2017).

[43]P. R. Nagy and M. Kállay, J. Chem. Phys. **146**, 214106 (2017).

[44]Y. Guo, C. Riplinger, U. Becker, D. G. Liakos, Y. Minenkov, L. Cavallo, and F. Neese, J. Chem. Phys. **148**, 011101 (2018).

[45]Q. Ma and H.-J. Werner, J. Chem. Theory Comput. **14**, 198 (2018).

[46]P. R. Nagy, G. Samu, and M. Kállay, J. Chem. Theory Comput. **14**, 4193 (2018).

[47]F. Neese, F. Wennmohs, U. Becker, and C. Riplinger, J. Chem. Phys. **152**, 224108 (2020).

[48]H.-J. Werner, P. J. Knowles, F. R. Manby, J. A. Black, K. Doll, A. Heßelmann, D. Kats, A. Köhn, T. Korona, D. A. Kreplin, Q. Ma, I. Miller, Thomas F., A. Mitrushchenkov, K. A. Peterson, I. Polyak, G. Rauhut, and M. Sibaev, J. Chem. Phys. **152**, 144107 (2020).

[49]M. Kállay, P. R. Nagy, D. Mester, Z. Rolik, G. Samu, J. Csontos, J. Csóka, P. B. Szabó, L. Gyevi-Nagy, B. Hégely, I. Ladjánszki, L. Szegedy, B. Ladóczki, K. Petrov, M. Farkas, P. D. Mezei, and Ganyecz, J. Chem. Phys. **152**, 074107 (2020).

[50]I. Sandler, J. Chen, M. Taylor, S. Sharma, and J. Ho, J. Phys. Chem. A **125**, 1553 (2021).

[51]D. G. A. Smith, L. A. Burns, A. C. Simmonett, R. M. Parrish, M. C. Schieber, R. Galvelis, P. Kraus, H. Kruse, R. D. Remigio, A. Alenaizan, A. M. James, S. Lehtola, J. P. Misiewicz, M. Scheurer, R. A. Shaw, J. B. Schriber, Y. Xie, Z. L. Glick, D. A. Sirianni, J. S. O'Brien, J. M. Waldrop, A. Kumar, E. G. Hohenstein, B. P. Pritchard, B. R. Brooks, H. F. Schaefer, A. Y. Sokolov, K. Patkowski, A. E. DePrince, U. Bozkaya, R. A. King, F. A. Evangelista, J. M. Turney, T. D. Crawford, and C. D. Sherrill, J. Chem. Phys. **152**, 184108 (2020).

[52]N. Niemeyer, P. Eschenbach, M. Bensberg, J. Tölle, L. Hellmann, L. Lampe, A. Massolle, A. Rikus, D. Schnieders, J. P. Unsleber, and J. Neugebauer, WIREs Computational Molecular Science **13**, e1647 (2023).

[53]M. Bensberg and J. Neugebauer, J. Chem. Theory Comput. **16**, 3607 (2020).

[54]H. Koch, O. Christiansen, R. Kobayashi, P. Jørgensen, and T. Helgaker, Chem. Phys. Lett. **228**, 233 (1994).

[55]A. E. DePrince and C. D. Sherrill, J. Chem. Theory Comput. **9**, 2687 (2013).

[56]S. Yoo, E. Aprà, X. C. Zeng, and S. S. Xantheas, J. Phys. Chem. Lett. **1**, 3122 (2010).

[57]D. M. Bates, J. R. Smith, T. Janowski, and G. S. Tschumper, J. Chem. Phys. **135**, 044123 (2011).

[58]D. Bykov, K. Kristensen, and T. Kjærgaard, J. Chem. Phys. **145**, 024106 (2016).

[59]J. F. Stanton, J. Gauss, J. D. Watts, and R. J. Bartlett, J. Chem. Phys. **94**, 4334 (1991).

[60]J. L. Whitten, J. Chem. Phys. **58**, 4496 (1973).

[61]M. Feyereisen, G. Fitzgerald, and A. Komornicki, Chem. Phys. Lett. **208**, 359 (1993).

[62]B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, J. Chem. Phys. **71**, 3396 (1979).

[63]O. Vahtras, J. Almlöf, and M. Feyereisen, Chem. Phys. Lett. **213**, 514 (1993).

[64]A. P. Rendell and T. J. Lee, J. Chem. Phys. **101**, 400 (1994).

[65]F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs, Chem. Phys. Lett. **294**, 143 (1998).

[66]F. Weigend, Phys. Chem. Chem. Phys. **4**, 4285 (2002).

[67]A. Sodt, J. E. Subotnik, and M. Head-Gordon, J. Chem. Phys. **125**, 194109 (2006).

[68]H.-J. Werner, F. R. Manby, and P. J. Knowles, J. Chem. Phys. **118**, 8149 (2003).

[69]N. H. F. Beebe and J. Linderberg, Int. J. Quantum Chem. **12**, 683 (1977).

[70]I. Røeggen and T. Johansen, J. Chem. Phys. **128**, 194107 (2008).

[71]T. B. Pedersen, S. Lehtola, I. Fdez. Galván, and R. Lindh, Wiley Interdiscip. Rev. Comput. Mol. Sci. **14**, e1692 (2024).

[72]K. E. Riley, M. Pitoňák, P. Jurečka, and P. Hobza, Chem. Rev. **110**, 5023 (2010).

[73]A. Karton, E. Rabinovich, J. M. L. Martin, and B. Ruscic, J. Chem. Phys. **125**, 144108 (2006).

[74]A. Tajti, P. G. Szalay, A. G. Császár, M. Kállay, J. Gauss, E. F. Valeev, B. A. Flowers, J. Vázquez, and J. F. Stanton, J. Chem. Phys. **121**, 11599 (2004).

[75]K. L. Bak, P. Jørgensen, J. Olsen, T. Helgaker, and W. Klopper, J. Chem. Phys. **112**, 9229 (2000).

[76]B. W. Hopkins and G. S. Tschumper, J. Phys. Chem. A **108**, 2941 (2004).

[77]R. J. Bartlett, J. Watts, S. Kucharski, and J. Noga, Chem. Phys. Lett. **165**, 513 (1990).

[78]A. Dutta and C. D. Sherrill, J. Chem. Phys. **118**, 1610 (2003).

[79]M. Lesiuk, J. Chem. Theory Comput. **16**, 453 (2020).

[80]F. Jensen, *Introduction to Computational Chemistry* (2007) pp. 204–208.

[81]J. M. Foster and S. F. Boys, Rev. Mod. Phys. **32**, 300 (1960).

[82]J. W. Boughton and P. Pulay, J. Comput. Chem. **14**, 736 (1993).

[83]J. Pipek and P. G. Mezey, J. Chem. Phys. **90**, 4916 (1989).

[84]P. Pulay, Chem. Phys. Lett. **100**, 151 (1983).

[85]M. Schütz, G. Hetzer, and H.-J. Werner, J. Chem. Phys. **111**, 5691 (1999).

[86]M. Schütz, J. Chem. Phys. **113**, 9986 (2000).

[87]M. Schutz and H.-J. Werner, J. Chem. Phys. **114**, 661 (2001).

[88]R. S. Mulliken, J. Chem. Phys. **23**, 1833 (1955).

[89]P. Jurečka, J. Šponer, J. Černý, and P. Hobza, Phys. Chem. Chem. Phys. **8**, 1985 (2006).

[90]M. S. Marshall, L. A. Burns, and C. D. Sherrill, J. Chem. Phys. **135**, 194102 (2011).

[91]T. H. Dunning, J. Chem. Phys. **90**, 1007 (1989).

[92]E. Papajak and D. G. Truhlar, J. Chem. Theory Comput. **7**, 10 (2011).

[93]S. Boys and F. Bernardi, Mol. Phys. **19**, 553 (1970).

[94]J. Řezáč, K. E. Riley, and P. Hobza, J. Chem. Theory Comput. **7**, 2427 (2011).

[95]F. Weigend and R. Ahlrichs, Phys. Chem. Chem. Phys. **7**, 3297 (2005).

[96]S. Hirata, J. Phys. Chem. A **107**, 9887 (2003).

[97]A. A. Auer, G. Baumgartner, D. E. Bernholdt, A. Bibireata, V. Choppella, D. Cociorva, X. Gao, R. Harrison, S. Krishnamoorthy, S. Krishnan, C.-C. Lam, Q. Lu, M. Nooijen, R. Pitzer, J. Ramanujam, P. Sadayappan, and A. Sibiryakov, Mol. Phys. **104**, 211 (2006).

[98]T. Janowski, A. R. Ford, and P. Pulay, J. Chem. Theory Comput. **3**, 1368 (2007).

[99]T. Janowski and P. Pulay, J. Chem. Theory Comput. **4**, 1585 (2008).

[100]H. van Dam, W. de Jong, E. Bylaska, N. Govind, K. Kowalski, T. Straatsma, and M. Valiev, Wiley Interdiscip. Rev. Comput. Mol. Sci. **1**, 888 (2011).

[101]E. Deumens, V. F. Lotrich, A. Perera, M. J. Ponton, B. A. Sanders, and R. J. Bartlett, Wiley Interdiscip. Rev. Comput. Mol. Sci. **1**, 895 (2011).

[102]R. Kobayashi and A. P. Rendell, Chem. Phys. Lett. **265**, 1 (1997).

[103]V. M. Anisimov, G. H. Bauer, K. Chadalavada, R. M. Olson, J. W. Glenski, W. T. C. Kramer, E. Aprà, and K. Kowalski, J. Chem. Theory Comput. **10**, 4307 (2014).

[104]E. Solomonik, D. Matthews, J. R. Hammond, J. F. Stanton, and J. Demmel, J. Parallel Distr. Com. **74**, 3176 (2014).

[105]C. Peng, J. A. Calvin, F. Pavošević, J. Zhang, and E. F. Valeev, J. Phys. Chem. A **120**, 10231 (2016).

[106]D. I. Lyakh, Int. J. Quantum Chem. **119**, e25926 (2019).

[107]L. Gyevi-Nagy, M. Kállay, and P. R. Nagy, J. Chem. Theory Comput. **16**, 366 (2020).

[108]C. Peng, C. A. Lewis, X. Wang, M. C. Clement, K. Pierce, V. Rishi, F. Pavošević, S. Slattery, J. Zhang, N. Teke, A. Kumar, C. Masteran, A. Asadchev, J. A. Calvin, and E. F. Valeev, J. Chem. Phys. **153**, 044120 (2020).

[109]D. Datta and M. S. Gordon, J. Chem. Theory Comput. **17**, 4799 (2021).

[110]L. Gyevi-Nagy, M. Kállay, and P. R. Nagy, J. Chem. Theory Comput. **17**, 860 (2021).

[111]K. Kowalski, R. Bair, N. P. Bauman, J. S. Boschen, E. J. Bylaska, J. Daily, W. A. de Jong, T. Dunning, N. Govind, R. J. Harrison, M. Keçeli, K. Keipert, S. Krishnamoorthy, S. Kumar, E. Mutlu, B. Palmer, A. Panyala, B. Peng, R. M. Richard, T. P. Straatsma, P. Sushko, E. F. Valeev, M. Valiev, H. J. J. van Dam, J. M. Waldrop, D. B. Williams-

Young, C. Yang, M. Zalewski, and T. L. Windus, Chem. Rev. **121**, 4962 (2021).

[112]J. A. Calvin, C. Peng, V. Rishi, A. Kumar, and E. F. Valeev, Chem. Rev. **121**, 1203 (2021).

[113]D. Datta and M. S. Gordon, J. Chem. Theory Comput. **19**, 7640 (2023).

[114]S. Seritan, C. Bannwarth, B. S. Fales, E. G. Hohenstein, S. I. L. Kokkila-Schumacher, N. Luehr, J. W. Snyder, C. Song, A. V. Titov, I. S. Ufimtsev, and T. J. Martínez, J. Chem. Phys. **152**, 224110 (2020).

[115]Z. Wang, M. Guo, and F. Wang, Phys. Chem. Chem. Phys. **22**, 25103 (2020).

[116]C. Peng, J. A. Calvin, and E. F. Valeev, Int. J. Quantum Chem. **119**, e25894 (2019).

[117]I. A. Kaliman and A. I. Krylov, J. Comput. Chem. **38**, 842 (2017).

[118]A. E. DePrince, M. R. Kennedy, B. G. Sumpter, and C. D. Sherrill, Mol. Phys. **112**, 844 (2014).

[119]W. Ma, S. Krishnamoorthy, O. Villa, and K. Kowalski, J. Chem. Theory Comput. **7**, 1316 (2011).

[120]A. E. DePrince and J. R. Hammond, J. Chem. Theory Comput. **7**, 1287 (2011).