

RESEARCH ARTICLE | AUGUST 22 2024

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 ; Zachary L. Glick ; David Poole ; Justin M. Turney ; C. David Sherrill ; Henry F. Schaefer, III 



J. Chem. Phys. 161, 082502 (2024)

<https://doi.org/10.1063/5.0219963>



Articles You May Be Interested In

Communication: An improved linear scaling perturbative triples correction for the domain based local pair-natural orbital based singles and doubles coupled cluster method [DLPNO-CCSD(T)]

J. Chem. Phys. (January 2018)

Orbital pair selection for relative energies in the domain-based local pair natural orbital coupled-cluster method

J. Chem. Phys. (August 2022)

Direct orbital selection within the domain-based local pair natural orbital coupled-cluster method

J. Chem. Phys. (December 2021)



The Journal of Chemical Physics

Special Topics Open for Submissions

[Learn More](#)

Accurate and efficient open-source implementation of domain-based local pair natural orbital (DLPNO) coupled-cluster theory using a t_1 -transformed Hamiltonian

Cite as: J. Chem. Phys. 161, 082502 (2024); doi: 10.1063/5.0219963

Submitted: 21 May 2024 • Accepted: 5 August 2024 •

Published Online: 22 August 2024



Andy Jiang,^{1,a)} Zachary L. Glick,² David Poole,² Justin M. Turney,^{1,b)} C. David Sherrill,^{2,c)} and Henry F. Schaefer III¹

AFFILIATIONS

¹ Center for Computational Quantum Chemistry, Department of Chemistry, University of Georgia, Athens, Georgia 30602, USA

² Center for Computational Molecular Science and Technology, School of Chemistry and Biochemistry, School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0400, USA

^{a)} Author to whom correspondence should be addressed: ajiang10224@gmail.com

^{b)} Electronic mail: justin.turney@uga.edu

^{c)} Electronic mail: sherrill@gatech.edu

ABSTRACT

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).

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0219963>

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

$$|\Psi_{CC}\rangle = e^T |\Psi_0\rangle, \quad (1)$$

$$E_{CC} = \langle \Psi_0 | e^{-T} H e^T | \Psi_0 \rangle, \quad (2)$$

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

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

$$E_{\text{CCSD}} = \langle \Psi_0 | e^{-(T_1+T_2)} H e^{(T_1+T_2)} | \Psi_0 \rangle, \quad (4)$$

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) method⁵ 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 as $\mathcal{O}(N^6)$, 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 (such as crambin and insulin) to allow for increased applicability of high-accuracy quantum chemistry to fields such as drug discovery and computational biology. Currently, cheaper methods, such as 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.* used orthogonal projectors to transform CCSD amplitudes into smaller-ranked tensors.^{16–18} Lesiuk 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) decomposition²⁴ of the orthogonal projectors, Hohenstein *et al.* applied THC to CCSD amplitudes,¹⁸ while Jiang *et al.* have 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 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, rendering 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 a 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 Sec. III C.

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, such as the 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 (MOs); 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 $n_{\text{occ}} + n_{\text{virt}}$.
- $\tilde{\mu}, \tilde{\nu}, \tilde{\lambda}, \tilde{\sigma}$: projected atomic orbitals (PAOs); these range from 1 to n_{bf} .
- $\tilde{\mu}_{ij}, \tilde{\nu}_{ij}, \tilde{\lambda}_{ij}, \tilde{\sigma}_{ij}$: projected atomic orbitals localized to pair ij ; these range from 1 to $n_{\text{pao}, ij}$, the number of PAOs local to local molecular orbital (LMO) pair ij .
- $\tilde{\mu}_{ijk}, \tilde{\nu}_{ijk}, \tilde{\lambda}_{ijk}, \tilde{\sigma}_{ijk}$: projected atomic orbitals localized to triplet ijk ; these range from 1 to $n_{\text{pao}, ijk}$, the 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 $n_{pno, ij}$, the 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 $n_{tno, ijk}$, the 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} , the 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 $n_{aux, ij}$, the 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 $n_{aux, ijk}$, the number of auxiliary basis functions local to LMO triplet ijk .

The relative sizes of these indices are typically

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

$$n_{occ} \ll n_{virt} < n_{bf} < n_{aux} \sim \mathcal{O}(N), \quad (6)$$

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,

$$E_{\text{CCSD}} = \langle \Psi_0 | e^{-T_2} \tilde{H} e^{T_2} | \Psi_0 \rangle, \quad (7)$$

where

$$\tilde{H} = e^{-T_1} H e^{T_1}, \quad (8)$$

$$T_1 = t_i^a E_{ai}, \quad (9)$$

$$T_2 = t_{ij}^{ab} E_{ai} E_{bj}. \quad (10)$$

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

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

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,

$$R_i^a = \langle \Psi_i^a | e^{-T_2} \tilde{H} e^{T_2} | \Psi_0 \rangle, \quad (12)$$

$$R_{ij}^{ab} = \langle \Psi_{ij}^{ab} | e^{-T_2} \tilde{H} e^{T_2} | \Psi_0 \rangle, \quad (13)$$

with

$$t_i^a = t_i^a - \frac{R_i^a}{\epsilon_a - \epsilon_i}, \quad (14)$$

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

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 CCSD⁵⁹ 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

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

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

$$(pq|rs) \approx B_{pq}^Q B_{rs}^Q, \quad (17)$$

where

$$B_{pq}^Q = (Q|P)^{-\frac{1}{2}} (P|pq). \quad (18)$$

We present our working equations based on the formalism of DePrince and Sherrill,⁵⁵ with Eq. (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,

$$R_{ij}^{ab} = \tilde{K}_{ij}^{ab} + A_{ij}^{ab} + B_{ij}^{ab} + \tilde{P}_{ij}^{ab} \left[\frac{1}{2} C_{ij}^{ab} + C_{ji}^{ab} + D_{ij}^{ab} + E_{ij}^{ab} + G_{ij}^{ab} \right], \quad (19)$$

where

$$\tilde{K}_{ij}^{ab} = \tilde{B}_{ai}^Q \tilde{B}_{bj}^Q, \quad (20)$$

$$A_{ij}^{ab} = t_{ij}^{cd} \tilde{B}_{ac}^Q \tilde{B}_{bd}^Q, \quad (21)$$

$$B_{ij}^{ab} = t_{kl}^{ab} \rho_{ij}^{kl}, \quad (22)$$

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

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

$$E_{ij}^{ab} = t_{ij}^{ac} \tilde{F}_{bc}, \quad (25)$$

$$G_{ij}^{ab} = -t_{ik}^{ab} \tilde{F}_{kj}, \quad (26)$$

with

$$\beta_{ij}^{kl} = \tilde{B}_{ki}^Q \tilde{B}_{lj}^Q + t_{ij}^{cd} B_{kc}^Q B_{ld}^Q, \quad (27)$$

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

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

$$\tilde{F}_{bc} = \tilde{F}_{bc} - u_{kl}^{bd} B_{ld}^Q B_{kc}^Q, \quad (30)$$

$$\tilde{F}_{kj} = \tilde{F}_{kj} + u_{lj}^{cd} B_{kd}^Q B_{lc}^Q. \quad (31)$$

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

$$R_i^a = \tilde{F}_{ai} + A_i^a + B_i^a + C_i^a, \quad (32)$$

where

$$A_i^a = u_{ki}^{cd} B_{kc}^Q \tilde{B}_{ad}^Q, \quad (33)$$

$$B_i^a = -u_{kl}^{ac} \tilde{B}_{ki}^Q B_{lc}^Q, \quad (34)$$

$$C_i^a = \tilde{F}_{kc} u_{ik}^{ac}. \quad (35)$$

The DF/RI or CD integrals dressed with the singles amplitude take the form⁵⁵

$$\tilde{B}_{ki}^Q = B_{ki}^Q + B_{ka}^Q t_i^a, \quad (36)$$

$$\tilde{B}_{ia}^Q = B_{ia}^Q, \quad (37)$$

$$\tilde{B}_{ai}^Q = B_{ai}^Q - t_k^a B_{ki}^Q + B_{ab}^Q t_i^b - t_k^a B_{kb}^Q t_i^b, \quad (38)$$

$$\tilde{B}_{ab}^Q = B_{ab}^Q - t_k^a B_{kb}^Q. \quad (39)$$

Since B_{ia}^Q does not transform under t_1 -dressing [Eq. (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,

$$\tilde{F}_{ki} = \tilde{F}_{ki} + \tilde{F}_{ka} t_i^a, \quad (40)$$

$$\tilde{F}_{ia} = \tilde{F}_{ia}, \quad (41)$$

$$\tilde{F}_{ai} = \tilde{F}_{ai} - t_k^a \tilde{F}_{ki} + \tilde{F}_{ab} t_i^b - t_k^a \tilde{F}_{kb} t_i^b, \quad (42)$$

$$\tilde{F}_{ab} = \tilde{F}_{ab} - t_k^a \tilde{F}_{kb}, \quad (43)$$

where

$$\tilde{F}_{rs} = F_{rs} + [2(rs|kc) - (rc|ks)] t_k^c. \quad (44)$$

The energy expression is

$$E_{\text{CCSD}} = (t_{ij}^{ab} + t_i^a t_j^b) [2(ia|jb) - (ib|ja)]. \quad (45)$$

C. Perturbative triples correction in CCSD(T)

Although 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, $E_{(T)}$ can be expressed as²⁵

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

with

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

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

Following the formalism of Lesiuk,⁷⁹ we define P_L and P_S , or the “long” and “short” permutation operators, as

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

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

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

$$t_{ijk}^{abc} = \frac{W_{ijk}^{abc}}{\epsilon_i + \epsilon_j + \epsilon_k - \epsilon_a - \epsilon_b - \epsilon_c}. \quad (51)$$

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

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

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

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

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

$$C_{\mu i}^L = C_{\mu j} U_{ji}. \quad (54)$$

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 estimate³⁹ is sufficient for a description of non-significantly correlated pairs. In our work, similar to the previous work by Riplinger *et al.*,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 as $\mathcal{O}(N^2)$] are treated using an inexpensive dipole estimate. Semi-canonical MP2 pairs, scaling as $\mathcal{O}(N)$, are treated using semi-canonical MP2 in the projected atomic orbital (PAO) basis, while weak MP2 pairs, scaling as $\mathcal{O}(N)$, are treated with full iterative LMP2. The surviving pairs, the strong pairs, scaling as $\mathcal{O}(N)$, 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 is 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.

Although 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 pre-factor 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:

$$C_{\mu\tilde{\nu}}^{\text{PAO}} = \delta_{\mu\nu} - C_{\mu i}^L C_{\sigma i}^L S_{\sigma\nu}^{\text{AO}}, \quad (55)$$

$$S_{\mu\tilde{\nu}}^{\text{PAO}} = C_{\lambda\mu}^{\text{PAO}} S_{\lambda\sigma}^{\text{AO}} C_{\sigma\nu}^{\text{PAO}}. \quad (56)$$

The $C_{\mu\tilde{\nu}}^{\text{PAO}}$ coefficients give the contribution of atomic orbital μ to PAO $\tilde{\nu}$, and $S_{\mu\tilde{\nu}}^{\text{PAO}}$ represents the overlap matrix between two PAOs. Next, the PAOs are normalized,

$$C_{\mu\tilde{\nu}}^{\text{PAO}} = (S_{\nu\tilde{\nu}}^{\text{PAO}})^{-\frac{1}{2}} C_{\mu\tilde{\nu}}^{\text{PAO}}, \quad (57)$$

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 co-workers later extended the same framework to CCSD⁸⁶ 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),

$$\text{DOI}_{i\tilde{\mu}} = (\tilde{i}\mu|\tilde{i}\mu)^{\frac{1}{2}}. \quad (58)$$

If the value of the integral is greater than a given tolerance $T_{\text{CUT_DO}}$, then the PAO $\tilde{\mu}$ 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, such as 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,

$$F_{\tilde{\mu}\tilde{\nu}}^{\text{PAO}} = \epsilon_{\tilde{\mu}\tilde{\nu}} \delta_{\tilde{\mu}\tilde{\nu}} = X_{\tilde{\mu}} F_{\mu\nu}^{\text{PAO}} X_{\tilde{\nu}}^{\text{PAO}}. \quad (59)$$

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 CEPA²⁹ and CCSD.³⁰ PNOs are eigenvectors of the pair density of a molecular orbital pair ij ,

$$D_{ij}^{ab} = \frac{1}{1 + \delta_{ij}} [u_{ij}^{ac} t_{ij}^{bc} + u_{ij}^{ca} t_{ij}^{cb}]. \quad (60)$$

The pair density can be 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 and Neese³⁴ updated their methodology by computing PNOs using canonical PAOs, from *semi-canonical* MP2 amplitudes. This is known as the domain-based local pair natural orbital (DLPNO) approach,

$$D_{ij}^{\mu\nu} = X_{\mu ij a_{ij}}^{\text{PNO},ij} n_{a_{ij}}^{\text{occ},ij} X_{\nu ij a_{ij}}^{\text{PNO},ij}. \quad (61)$$

The eigenvectors, $X_{\mu ij a_{ij}}^{\text{PNO},ij}$, represent the transformation from canonical PAOs to PNOs, and their eigenvalues $n_{a_{ij}}^{\text{occ},ij}$ 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, we use three criteria 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 semi-canonical 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 interactions. We present results using $T_{\text{CUT_PNO}} = 10^{-7}$, $T_{\text{CUT_ENERGY}} = 0.997$, 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 package⁴⁷ in the DLPNO-CCSD algorithm.³⁴ The energy criterion was first introduced by the work of Schwilk *et al.* in the Molpro package⁴⁸ 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 ,

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

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 $T_{\text{CUT_DO_TRIPLES}}$ (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 $T_{\text{CUT_TNO}}$ (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

$$P_{\mu\nu}^i = C_{\mu i}^L S_{\mu\nu} C_{\nu i}^L, \quad (63)$$

$$q_{iA} = 2 \sum_{\mu \in A} \sum_{\nu} P_{\mu\nu}^i \cdot \frac{P_{\mu\mu}^i}{P_{\mu\mu}^i + P_{\nu\nu}^i}. \quad (64)$$

If q_{iA} for local molecular orbital i on atom A is greater than $T_{\text{CUT_MKN}}$, 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 Sec. II as a starting point, following the work of DePrince and Sherrill.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),

$$S_{a_{kl}}^{a_{ij}} = X_{\mu a_{ij}}^{\text{PNO},ij} S_{\mu\nu}^{\text{PAO}} X_{\nu a_{kl}}^{\text{PNO},kl}. \quad (65)$$

For example, Eq. (22) becomes

$$B_{ij}^{a_{ij}b_{ij}} = (S_{a_{kl}}^{a_{ij}} t_{kl}^{a_{kl}b_{kl}} S_{b_{kl}}^{b_{ij}}) \beta_{ij}^{kl} \quad (66)$$

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 Sec. III A.

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,

$$(ia_{kl}|jb_{kl}) = X_{\mu a_{kl}}^{\text{PNO},kl} (\tilde{i}|\tilde{\mu})(\tilde{\mu}|\tilde{j}) X_{\nu b_{kl}}^{\text{PNO},kl} \quad (67)$$

or

$$(ia_{kl}|jb_{kl}) \approx S_{a_{ij}}^{a_{kl}} (ia_{ij}|jb_{ij}) S_{b_{ij}}^{b_{kl}}. \quad (68)$$

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 Eq. (68), is akin to using the PNO basis of ij in a resolution of the identity (RI) operator,

$$\sum_{a_{ij}} |a_{ij}\rangle \langle a_{ij}| \approx 1. \quad (69)$$

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 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 bold-faced 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,

$$\tilde{t}_i^{a_{kl}} = S_{a_{ii}}^{a_{kl}} t_i^{a_{kl}}, \quad (70)$$

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

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

$$L_{pq}^{rs} = 2K_{pq}^{rs} - K_{pq}^{sr}, \quad (73)$$

$$M_{pq}^{rs} = 2K_{pq}^{rs} - J_{pq}^{rs}. \quad (74)$$

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

$$\tilde{K}_{ij}^{a_{ij}b_{ij}} = \tilde{B}_{a_{ij}i}^{Q_{ij}} \tilde{B}_{b_{ij}j}^{Q_{ij}}, \quad (75)$$

$$A_{ij}^{a_{ij}b_{ij}} = t_{ij}^{c_{ij}d_{ij}} \tilde{B}_{a_{ij}c_{ij}}^{Q_{ij}} \tilde{B}_{b_{ij}d_{ij}}^{Q_{ij}}, \quad (76)$$

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

$$C_{ij}^{a_{ij}b_{ij}} = -S_{a_{ki}}^{a_{ij}} \gamma_{ki}^{a_{kl}c_{kl}} S_{c_{kj}}^{c_{ij}} t_{kj}^{b_{kl}c_{kl}} S_{b_{kj}}^{b_{ij}} - J_{ik}^{a_{ij}c_{ij}} S_{b_{kj}}^{b_{ij}} t_{kj}^{b_{kl}c_{kl}}, \quad (78)$$

$$D_{ij}^{a_{ij}b_{ij}} = \frac{1}{2} S_{a_{ik}}^{a_{ij}} \delta_{ik}^{a_{kl}c_{kl}} S_{c_{jk}}^{c_{ij}} u_{jk}^{b_{kl}c_{kl}} S_{b_{jk}}^{b_{ij}} + \frac{1}{2} M_{ik}^{a_{ij}c_{ij}} S_{b_{jk}}^{b_{ij}} u_{jk}^{b_{kl}c_{kl}}, \quad (79)$$

$$E_{ij}^{a_{ij}b_{ij}} = t_{ij}^{a_{ij}c_{ij}} \tilde{F}_{b_{ij}c_{ij}}, \quad (80)$$

$$G_{ij}^{a_{ij}b_{ij}} = -(S_{a_{ik}}^{a_{ij}} t_{ik}^{a_{kl}b_{kl}} S_{b_{ik}}^{b_{ij}}) \tilde{F}_{kj}. \quad (81)$$

The intermediates are redefined as

$$\beta_{ij}^{kl} = \tilde{B}_{ki}^{Q_{ij}} \tilde{B}_{lj}^{Q_{ij}} + t_{ij}^{c_{ij}d_{ij}} B_{kcij}^{Q_{ij}} B_{ldij}^{Q_{ij}}, \quad (82)$$

$$\gamma_{ki}^{a_{kl}c_{kl}} = -\tilde{t}_i^{a_{kl}} J_{ki}^{c_{kl}} + \tilde{t}_i^{b_{kl}} J_{kbki}^{a_{kl}c_{kl}} - \tilde{t}_i^{b_{kl}} K_{kl}^{b_{kl}c_{kl}} S_{c_{kl}}^{c_{kl}} \tilde{t}_i^{a_{kl}} - \frac{1}{2} S_{a_{ii}}^{a_{kl}} t_{li}^{a_{kl}d_{kl}} S_{d_{kl}}^{d_{ij}} K_{kl}^{d_{kl}c_{kl}} S_{c_{kl}}^{c_{kl}}, \quad (83)$$

$$\delta_{ik}^{a_{kl}c_{kl}} = -\tilde{t}_i^{a_{kl}} M_{ik}^{c_{kl}} + \tilde{t}_i^{b_{kl}} M_{kbki}^{a_{kl}c_{kl}} - \tilde{t}_i^{b_{kl}} L_{lk}^{b_{kl}c_{kl}} S_{c_{kl}}^{c_{kl}} \tilde{t}_i^{a_{kl}} + \frac{1}{2} S_{a_{ii}}^{a_{kl}} u_{il}^{a_{kl}d_{kl}} S_{d_{kl}}^{d_{ij}} L_{kl}^{d_{kl}c_{kl}} S_{c_{kl}}^{c_{kl}}, \quad (84)$$

$$\tilde{F}_{b_{ij}c_{ij}} = \tilde{F}_{b_{ij}c_{ij}} - S_{b_{kl}}^{b_{ij}} u_{kl}^{b_{kl}d_{kl}} K_{kl}^{c_{kl}d_{kl}} S_{c_{kl}}^{c_{ij}}, \quad (85)$$

$$\tilde{F}_{kj} = \tilde{F}_{kj} + (S_{c_{ij}}^{c_{kl}} u_{ij}^{c_{ij}d_{ij}} S_{d_{ij}}^{d_{kl}}) K_{lk}^{c_{kl}d_{kl}}. \quad (86)$$

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

$$R_i^{a_{ii}} = \tilde{F}_{a_{ii}i} + A_i^{a_{ii}} + B_i^{a_{ii}} + C_i^{a_{ii}}, \quad (87)$$

where

$$A_i^{a_{ii}} = u_{ki}^{c_{kl}d_{kl}} K_{ka_{ki}}^{c_{kl}d_{kl}} S_{a_{ki}}^{a_{ii}} - \tilde{t}_i^{a_{ii}} u_{ki}^{c_{kl}d_{kl}} S_{c_{kl}}^{c_{kl}} K_{kl}^{c_{kl}d_{kl}} S_{d_{kl}}^{d_{kl}}, \quad (88)$$

$$B_i^{a_{ii}} = -S_{a_{kl}}^{a_{ii}} u_{kl}^{a_{kl}c_{kl}} [K_{kl}^{ic_{kl}} + \tilde{t}_i^{b_{kl}} K_{kl}^{b_{kl}c_{kl}}], \quad (89)$$

$$C_i^{a_{ii}} = S_{a_{ik}}^{a_{ii}} u_{ik}^{a_{kl}c_{kl}} \tilde{F}_{kcik}. \quad (90)$$

Here are the relevant DF integrals dressed with the singles amplitudes:

$$\tilde{B}_{ki}^{Q_{ij}} = B_{ki}^{Q_{ij}} + B_{ka_{ij}}^{Q_{ij}} \tilde{t}_i^{a_{ij}}, \quad (91)$$

$$\tilde{B}_{a_{ij}i}^{Q_{ij}} = B_{a_{ij}i}^{Q_{ij}} - \tilde{t}_k^{a_{ij}} B_{ki}^{Q_{ij}} + B_{a_{ij}b_{ij}}^{Q_{ij}} \tilde{t}_i^{b_{ij}} - \tilde{t}_k^{a_{ij}} B_{kb_{ij}}^{Q_{ij}} \tilde{t}_i^{b_{ij}}, \quad (92)$$

$$\tilde{B}_{a_{ij}b_{ij}}^{Q_{ij}} = B_{a_{ij}b_{ij}}^{Q_{ij}} - \tilde{t}_k^{a_{ij}} B_{kb_{ij}}^{Q_{ij}}. \quad (93)$$

For the dressed Fock matrices, since we are not using canonical MOs, we derived these equations from the work of Schwilk *et al.*,⁴¹

$$\tilde{F}_{ij} = \bar{F}_{ij} + \bar{F}_{ic_{ij}} t_j^{c_{ij}}, \quad (94)$$

$$\tilde{F}_{ia_{ij}} = S_{a_{ik}}^{a_{ij}} L_{ik}^{a_{ik}c_{ik}} \tilde{t}_k^{c_{ik}}, \quad (95)$$

$$\tilde{F}_{a_{ii}} = \bar{F}_{a_{ii}} - \tilde{t}_k^{a_{ii}} \bar{F}_{ki} + \bar{F}_{a_{ii}b_{ii}} t_i^{b_{ii}} - \tilde{t}_k^{a_{ii}} \bar{F}_{kb_{ii}} t_i^{b_{ii}}, \quad (96)$$

$$\tilde{F}_{a_{ij}b_{ij}} = \bar{F}_{a_{ij}b_{ij}} - \tilde{t}_k^{a_{ij}} \bar{F}_{kb_{ij}}, \quad (97)$$

where

$$\bar{F}_{ij} = F_{ij} + [2J_{ij}^{kc_{ij}} - K_{ji}^{kc_{ij}}] \tilde{t}_k^{c_{ij}}, \quad (98)$$

$$\bar{F}_{ia_{kl}} = [2B_{ia_{kl}}^{Q_{kl}} B_{m_{cl}}^{Q_{kl}} - B_{ic_{kl}}^{Q_{kl}} B_{ma_{kl}}^{Q_{kl}}] \tilde{t}_m^{c_{kl}}, \quad (99)$$

$$\bar{F}_{a_{ii}} = [2B_{a_{ii}}^{Q_{ii}} B_{kc_{ii}}^{Q_{ii}} - B_{a_{ii}c_{ii}}^{Q_{ii}} B_{ki}^{Q_{ii}}] \tilde{t}_k^{c_{ii}}, \quad (100)$$

$$\bar{F}_{a_{ij}b_{ij}} = \epsilon_{a_{ij}} \delta_{a_{ij}b_{ij}} + [2B_{a_{ij}b_{ij}}^{Q_{ij}} B_{kc_{ij}}^{Q_{ij}} - B_{a_{ij}c_{ij}}^{Q_{ij}} B_{kb_{ij}}^{Q_{ij}}] \tilde{t}_k^{c_{ij}}. \quad (101)$$

In our formalism, we dress our Fock matrices directly in the PNO space, rather than the PAO space, as is done by Werner and co-workers 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 Eqs. (83) and (84) from their canonical counterparts and expressing their contributions explicitly in Eqs. (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_{ij}^{ab} and R_i^a 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 Eqs. (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 Eqs. (78), (79), (88), and (89). Certain Fock matrix contributions also have a unique form. Equation (95) is not constructed from Eq. (99), since the former is looped over strong pairs and the latter is looped over both strong pairs and weak LMP2 pairs. Finally, Eq. (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

$$E_{\text{LCCSD}} = (t_{ij}^{a_{ij}b_{ij}} + \tilde{t}_i^{a_{ii}} \tilde{t}_j^{b_{ij}}) L_{ij}^{a_{ij}b_{ij}}, \quad (102)$$

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

$$t_i^{a_{ii}} = \frac{R_i^{a_{ii}}}{\epsilon_{a_i} - F_{ii}}, \quad (103)$$

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

$$t_i^{a_{ii}} = \frac{R_i^{a_{ii}}}{\tilde{F}_{a_{ij}a_{ij}} - \tilde{F}_{ii}}, \quad (105)$$

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

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

$$t_{ij}^{a_{ij}b_{ij}} = \frac{R_{ij}^{a_{ij}b_{ij}}}{\tilde{F}_{a_{ij}a_{ij}} + \tilde{F}_{b_{ij}b_{ij}} - \tilde{F}_{ii} - \tilde{F}_{jj}}. \quad (108)$$

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,

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

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

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 semi-canonical (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

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

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

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 [Eq. (53)]. Although the memory costs for storing these quantities may appear to be excessive due to its $\mathcal{O}(n_{ijk}^3 n_{pno-ijk})$ scaling, the memory requirements are often similar to 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 in addition, 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 are 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 co-workers.^{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 semi-canonical 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 is looser

in cutoffs.⁴⁰ In the third step, we recompute the semi-canonical LMP2 amplitudes for surviving pairs (non-dipole or semi-canonical) through the refined prescreening procedure⁴⁰ 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 correction³⁹ from strong pairs and weak MP2 pairs. The PNO truncation is computed as the difference between the semi-canonical 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,

$$\begin{aligned} E_{\text{DLPNO-CCSD}} = & E_{\text{LCCSD}}[\text{strong pairs}] + \Delta E_{\text{LMP2}}[\text{weak MP2 pairs}] \\ & + \Delta E_{\text{SC-LMP2}}[\text{semi-canonical MP2 pairs}] \\ & + \Delta E_{\text{dipole}}[\text{dipole pairs}] \\ & + \Delta E_{\text{PNO}}[\text{strong pairs} + \text{weak MP2 pairs}]. \end{aligned} \quad (113)$$

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.

B. Triples algorithm procedures

We model our triples algorithm based on a combination of features from DLPNO-CCSD(T) in ORCA⁴⁴ and PNO-LCCSD(T) in Molpro,⁴⁵ in order to optimize speed and accuracy. First, we use the

TABLE I. Parameters of our DLPNO-CCSD algorithm for TightPNO and NormalPNO settings. Boldface denotes default values for the TightPNO setting.

Parameter	Description	TightPNO value	NormalPNO value
$T_{\text{CUT_PNO}}$	LCCSD PNO occupation criterion	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	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	10^{-5}	10^{-5}
$T_{\text{CUT_PRE}}$	Dipole energy cutoff for pair screening	10^{-7}	10^{-6}
$T_{\text{CUT_PAIRS}}$	Strong/weak pair cutoff	10^{-5}	10^{-4}
$T_{\text{CUT_PAIRS_MP2}}$	Weak/semi-canonical pair cutoff	10^{-6}	10^{-6}
$T_{\text{CUT_MKN}}$	Local density fitting Mullikan tolerance	10^{-3}	10^{-3}

triples prescreening algorithm as presented by Ma and Werner.⁴⁵ We first compute the semi-canonical (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 Guo *et al.*,⁴⁴ where the TNOs of the surviving triplets are then recomputed at a tighter tolerance ($T_{\text{CUT_TNO}}$), 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 ~20% of triplets that account for at least 90% of the semi-canonical (T0) energy are deemed “strong triplets,”⁴⁴ and the rest are 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:

$$E_{\text{DLPNO-(T)}} = \sum_{i \leq j \leq k} E_{(T0)}^{ijk} [T_{\text{CUT_TNO}}] + \sum_{ijk \in \text{strong triplets}} (E_{(T)}^{ijk} - E_{(T0)}^{ijk}) \times [T_{\text{CUT_TNO}} \times T_{\text{STRONG_SCALE}}] + \sum_{ijk \in \text{weak triplets}} (E_{(T)}^{ijk} - E_{(T0)}^{ijk}) \times [T_{\text{CUT_TNO}} \times T_{\text{WEAK_SCALE}}] + \sum_{ijk \in \text{screened triplets}} \Delta E_{(T0)}^{ijk} [T_{\text{CUT_TNO_PRE}}]. \quad (114)$$

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

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 $n_{\text{pno}_{ij}} \cdot n_{\text{pno}_{ij}}$) 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 $n_{\text{pno}_{ij}} \cdot n_{\text{pno}_{kj}}$, 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 Appendixes A and B.

All the dressed Fock matrices are computed and stored, while only the dressed integrals $\tilde{B}_{a_{ij}i}^{Q_{ij}}$ (list of matrices of dimension $n_{\text{aux}_{ij}} \cdot n_{\text{pno}_{ij}}$ indexed by ij) and $\tilde{B}_{k_{ij}i}^{Q_{ij}}$ (list of matrices of dimension $n_{\text{aux}_{ij}} \cdot n_{\text{lmo}_{ij}}$ 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 the LMO pair ij and then the 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 [Eqs. (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 $S_{b_{kl}}^{a_{ij}}$, 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 Appendixes A and B.

D. Computational details

For all correlated computations, the frozen-core approximation is used. Unless stated otherwise, TightPNO convergence is used for all computations. All timings are performed on 16 cores of an Intel Xeon 6136 central processing unit (CPU) (3.0 GHz processing speed with 1 TB of RAM) unless stated otherwise. 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

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

Parameter	Description	Default value
$T_{\text{CUT_TNO}}$	TNO occupation criterion	10⁻⁹
$T_{\text{CUT_TNO_PRE}}$	TNO occupation criterion in “screened triplets”	10⁻⁷
$T_{\text{CUT_TRIPLES_PRE}}$	“Screened triplets” energy cutoff	10⁻⁷
$T_{\text{CUT_DO_TRIPLES}}$	LMO/PAO DOI criterion for triples domains	10⁻²
$T_{\text{CUT_MKN_TRIPLES}}$	Local density fitting tolerance for triples	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

TABLE III. TightPNO DLPNO-CCSD error statistics compared to canonical DF-CCSD reference (kcal mol^{-1}). jun-cc-pVDZ augments the cc-pVDZ basis set by adding diffuse functions for all heavy atoms up to shell $l_{\text{max}} - 1$. Boldface denotes summary MAE, minimum, and maximum errors for each basis set.

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

V. RESULTS

A. Dimer interaction energies

We first present the results of our DLPNO-CCSD and (T) algorithms on the S22 dataset,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) and DF-CCSD(T) are shown in Table IV. We present the results of

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 set by adding diffuse functions for all heavy atoms up to shell $l_{\text{max}} - 1$. Boldface denotes summary MAE, minimum, and maximum errors for each basis set.

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

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 [supplementary material](#)). Generally speaking, DLPNO-CCSD/DLPNO-CCSD(T) is the most accurate when modeling dimers bound by mixed influences, vs 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 convergence, which is consistent with the findings of Liakos *et al.*³⁶

B. Potential energy surfaces

Next, we examined the potential energy surface along the dissociation of a uracil dimer pair, from the S66x8 dataset of Řezáč *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 [Fig. 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 min per dimer, with the computation involving more separated dimers being faster (39 min for $0.9R_{eq}$; 26 min for $2.0R_{eq}$).

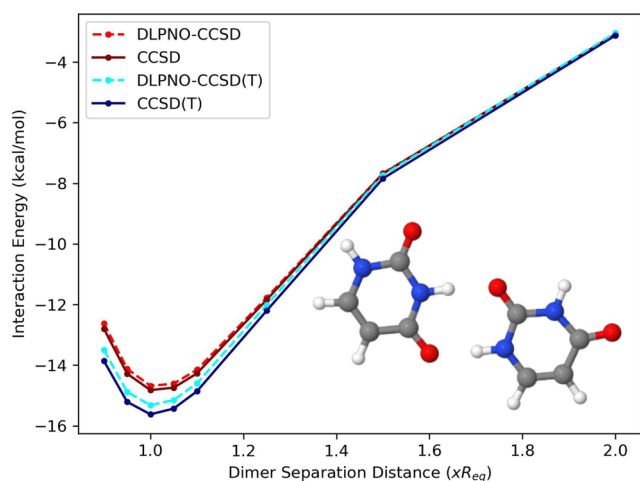


FIG. 1. Counterpoise (CP) corrected cc-pVTZ interaction energies for uracil dimer base pair along the (frozen monomer) dissociation curve.

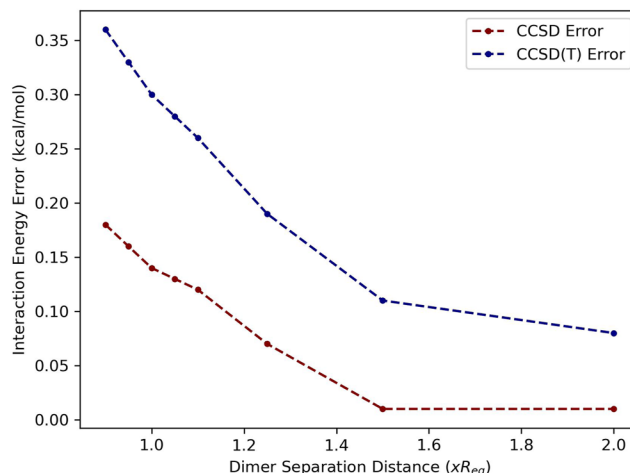


FIG. 2. Errors along the frozen monomer dissociation curve for uracil dimer base pair, defined as the canonical DF-CCSD(T) interaction energy subtracted from the DLPNO-CCSD(T) result.

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 [Fig. 2](#). As expected, the errors decrease as the system becomes more well separated, since the magnitude of the intermolecular 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. Larger errors relative to the canonical reference around equilibrium are to be expected due to the local approximations and are controllable through tightening the parameters.

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 (H₂O)₁₆ and (H₂O)₁₇, in the aug-cc-pVTZ basis set, from the work of Yoo *et al.*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 co-workers presented results for five different conformers of (H₂O)₁₆ and two different conformers of (H₂O)₁₇. There are four possible conformers of (H₂O)₁₇, but not all their computations finished since they ran out of computing time. [Table V](#) presents the relative conformational energies for (H₂O)₁₆ and (H₂O)₁₇ as computed by Xantheas and co-workers, 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 *et al.*,⁵⁷ who developed a highly accurate many-body expansion method to obtain absolute energies for the conformers close to the

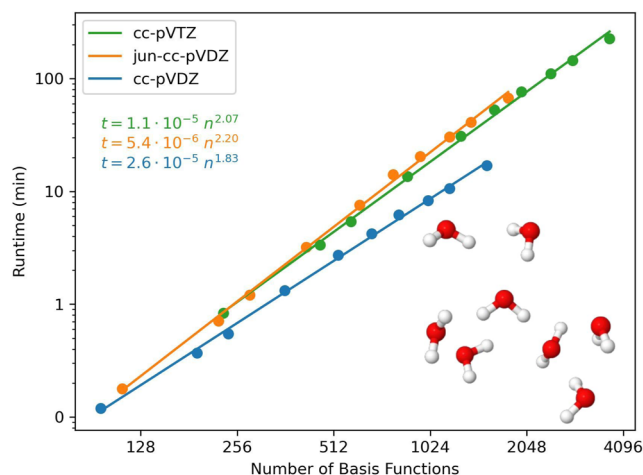
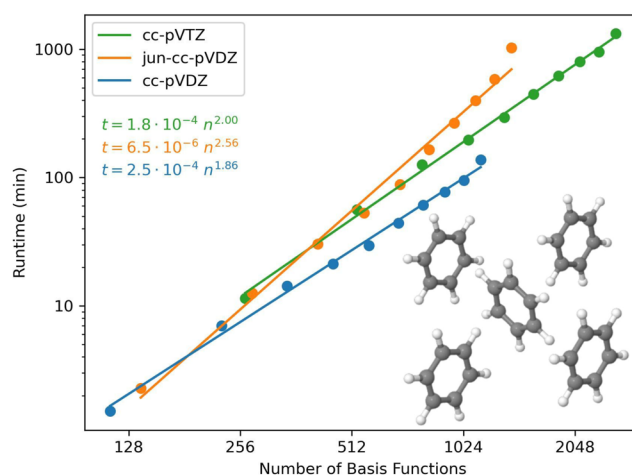
TABLE V. Relative conformation energies (kcal mol⁻¹) using canonical CCSD(T) from the work of Yoo *et al.*,⁵⁶ compared with DLPNO-CCSD(T) as implemented in this paper and the implementation in ORCA,⁴⁴ as well as the 3b:mb MP2/CCSD(T) method of Bates *et al.*⁵⁷ Boldface denotes the canonical values.

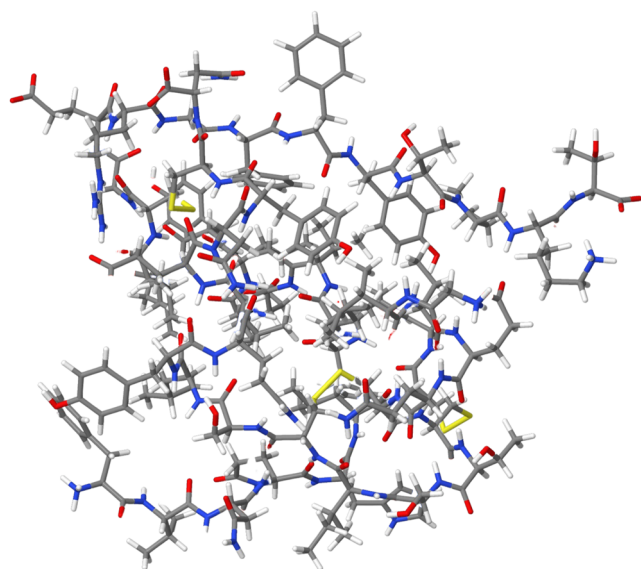
Isomer	ΔE_{canon}	ΔE_{PSI4}	ΔE_{ORCA}	$\Delta E_{\text{3b:mb}}$
$(\text{H}_2\text{O})_{16}$				
Boat-a	0.25	0.35	0.00	0.36
Boat-b	0.42	0.51	0.15	0.60
Antiboat	0.51	0.63	0.27	0.67
4444-a (abab)	0.00	0.00	0.00	0.00
4444-b (aabb)	0.54	0.52	0.59	0.57
$(\text{H}_2\text{O})_{17}$				
Sphere	0.00	0.00	0.00	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

canonical CCSD(T) values, called the “CCSD(T):MP2 3-body:manybody” 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,

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

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 $(\text{H}_2\text{O})_{16}$, where the number of PNOs for our pairs ranges from 11 to 149 (average of 44), while ORCA ranges from 6 to 73 (average of 25). In addition, our criterion for the initial consideration of triplets (2 weak pairs vs 1 in ORCA)

**FIG. 3.** Water cluster timings and empirical scaling with the DLPNO-CCSD(T) method. System sizes range from 4 to 64 water molecules, presented for each basis set.**FIG. 4.** Benzene cluster timings and empirical scaling with the DLPNO-CCSD(T) method. System sizes range from 1 to 10 benzene molecules, presented for each basis set.

**FIG. 5.** 3D structure of insulin (787 atoms).

allows our code to recover more triples energy compared to ORCA. For the boat-a configuration, our DLPNO-CCSD absolute energy is $0.3 \text{ kcal mol}^{-1}$ lower, while the (T) contribution is $0.4 \text{ kcal mol}^{-1}$ lower, leading to a net difference of around $0.7 \text{ kcal mol}^{-1}$.

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](#).

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 [supplementary material](#)), as shown in [Figs. 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 wall time (in minutes) compared to the system size (by the 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 runtime, a is a pre-factor, n is the system size, and b is the computational scaling. The results of our analysis are presented in [Figs. 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 shows 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 pre-factor 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 shown in [Fig. 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 and Ahlrichs,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.

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

	NormalPNO	TightPNO	Diff.	Diff. (kcal mol^{-1})
DLPNO-CCSD(T) correlation energy	-62 393.798	-62 433.591	-39.8	-25.0
DLPNO-CCSD contribution	-60 265.340	-60 289.967	-24.6	-15.5
LCCSD correlation energy	-58 868.909	-59 952.074	-1083.2	-679.7
Weak pair contribution	-1248.806	-253.246	995.6	624.7
Semi-canonical 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. Boldface denotes the proportion of time spent in DLPNO-CCSD compared to DLPNO-(T) for each setting.

	NormalPNO	TightPNO
Total wall time (s)	44 906 (0.52 days)	128 090 (1.48 days)
DLPNO-CCSD	66%	83%
Orbital localization/sparsity prep	5%	2%
Dipole pair correction	0%	0%
Semi-canonical 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%
Semi-canonical (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)	24 872	24 872
Pair information		
Total LMO pairs (non-unique)	1 247 689	1 247 689
Dipole pairs	996 002 (79.8%)	857 900 (68.8%)
Semi-canonical LMP2 pairs	147 500 (11.8%)	284 386 (22.8%)
Weak pairs	87 590 (7.0%)	64 954 (5.2%)
Strong pairs	16 597 (1.3%)	40 449 (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)	232 901 202	232 901 202
Initial triplets	601 572 (0.3%)	955 435 (0.4%)
Final triplets	292 805 (0.1%)	296 016 (0.1%)
TNOs per triplet (prescreening)	26	22
TNOs per triplet (T0)	52	60
TNOs per triplet (T)	30	32

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 second 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, 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 in Table VII, the TightPNO computation takes about three 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 $S_{b_{kj}}^{a_{ij}}$ and $S_{b_{jk}}^{a_{ij}}$ are stored in RAM, and all other types are computed on the fly. This is significantly less efficient than the current implementation (as described earlier in Sec. IV), so this led to 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 as $\mathcal{O}(N^2)$, 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, such as insulin, with both the virtual and auxiliary domains being significantly less than the span of the entire molecule.

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 devel-

ALGORITHM 1. Linear-scaling computation of $K_{ik}^{a_{ij}b_{kj}}$ —used in Eq. (79) (as part of $M_{ik}^{a_{ij}b_{kj}}$).

```

for  $ij$  in strong_pairs do                                ▷ Loop over all strong  $ij$  pairs,  $i \leq j$ 
   $(P_{ij}|i\tilde{v}_{ij}) \leftarrow (P|m\tilde{v})[\text{lmopair\_to\_ribfs}[ij], i, \text{lmopair\_to\_paos}[ij]]$ 
   $A_{ia_{ij}}^{Q_{ij}} \leftarrow (Q_{ij}|P_{ij})^{-1}(P_{ij}|i\tilde{v}_{ij})X_{\tilde{v}_{ij}a_{ij}}^{PNO}$ 
  if  $i \neq j$  then
     $(P_{ij}|j\tilde{v}_{ij}) \leftarrow (P|m\tilde{v})[\text{lmopair\_to\_ribfs}[ij], j, \text{lmopair\_to\_paos}[ij]]$ 
     $A_{ja_{ij}}^{Q_{ij}} \leftarrow (Q_{ij}|P_{ij})^{-1}(P_{ij}|j\tilde{v}_{ij})X_{\tilde{v}_{ij}a_{ij}}^{PNO}$ 
    ▷ Get slice from  $(P|m\tilde{v})$ 
  end if
  for  $Q_{ij}$  in  $\text{lmopair\_to\_ribfs}[ij]$  do
    for  $k_{ij}$  in  $\text{lmopair\_to\_lmos}[ij]$  do
       $\bar{K}_{k_{ij}\tilde{v}_{kj}} \leftarrow (Q|m\tilde{v})[Q_{ij}, k_{ij}, \text{lmopair\_to\_lmos}[kj]]$ 
       $\bar{K}_{k_{ij}b_{kj}} \leftarrow \bar{K}_{k_{ij}\tilde{v}_{kj}}X_{\tilde{v}_{kj}b_{kj}}^{PNO}$ 
       $K_{ik}^{a_{ij}b_{kj}} + = A_{ia_{ij}}^{Q_{ij}}\bar{K}_{k_{ij}b_{kj}}$ 
      ▷ Get slice from  $(Q|m\tilde{v})$ 
      if  $i \neq j$  then
         $\bar{K}_{k_{ij}\tilde{v}_{ij}} \leftarrow (Q|m\tilde{v})[Q_{ij}, k_{ij}, \text{lmopair\_to\_lmos}[ik]]$ 
         $\bar{K}_{k_{ij}b_{ik}} \leftarrow \bar{K}_{k_{ij}\tilde{v}_{ij}}X_{\tilde{v}_{ij}b_{ik}}^{PNO}$ 
         $K_{jk}^{a_{ij}b_{ik}} + = A_{ja_{ij}}^{Q_{ij}}\bar{K}_{k_{ij}b_{ik}}$ 
        ▷ 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 Eqs. (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  $\leftarrow$  List[int]                        ▷ Form extended domain for pairs
  for  $k_{ij}$  in lmpair_to_lmos[ $ij$ ] do
     $k \leftarrow$  lmpair_to_lmos[ $ij$ ][ $k_{ij}$ ]
    pair_ext_domain = pair_ext_domain  $\cup$  lmo_to_paos[ $k$ ]
  end for
  ( $P_{ij}|ik_{ij}$ )  $\leftarrow$  ( $P|mn$ )[lmpair_to_ribfs[ $ij$ ],  $i$ , lmpair_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
    ( $P_{ij}|jk_{ij}$ )  $\leftarrow$  ( $P|mn$ )[lmpair_to_ribfs[ $ij$ ],  $j$ , lmpair_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 lmpair_to_ribfs[ $ij$ ] do                                ▷ Get slice from ( $Q|\tilde{\mu}\tilde{\nu}$ )
     $\tilde{J}_{\tilde{\mu}_{ij}\tilde{\nu}_{j\text{ext}}} \leftarrow (Q|\tilde{\mu}\tilde{\nu})[Q_{ij}, \text{lmpair\_to\_paos}[ij], \text{pair\_ext\_domain}]$ 
     $\tilde{J}_{a_{ij}\tilde{\nu}_{j\text{ext}}} \leftarrow X_{\tilde{\mu}_{ij}a_{ij}}^{\text{PNO}} \tilde{J}_{\tilde{\mu}_{ij}\tilde{\nu}_{j\text{ext}}}$ 
    for  $k_{ij}$  in lmo_pairs_to_lmos[ $ij$ ] do
       $\tilde{J}_{a_{ij}\tilde{\mu}_k} \leftarrow \tilde{J}_{a_{ij}\tilde{\nu}_{j\text{ext}}}[\text{All}, \text{pair\_ext\_domain} \cap \text{lmpair\_to\_paos}[k]]$ 
       $\tilde{J}_{a_{ij}b_{kj}} \leftarrow \tilde{J}_{a_{ij}\tilde{\mu}_k} X_{\tilde{\mu}_k b_{kj}}^{\text{PNO}}$ 
       $J_{ik}^{a_{ij}b_{kj}} + = A_{ik_{ij}}^{Q_{ij}} \tilde{J}_{a_{ij}b_{kj}}$ 
      if  $i \neq j$  then
         $\tilde{J}_{a_{ij}\tilde{\mu}_{ik}} \leftarrow \tilde{J}_{a_{ij}\tilde{\nu}_{j\text{ext}}}[\text{All}, \text{pair\_ext\_domain} \cap \text{lmpair\_to\_paos}[ik]]$ 
         $\tilde{J}_{a_{ij}b_{ik}} \leftarrow \tilde{J}_{a_{ij}\tilde{\mu}_{ik}} X_{\tilde{\mu}_{ik} b_{ik}}^{\text{PNO}}$ 
         $J_{jk}^{a_{ij}b_{ik}} + = A_{jk_{ij}}^{Q_{ij}} \tilde{J}_{a_{ij}b_{ik}}$                                 ▷ Form  $jk$  analog
      end if
    end for
  end for
end for

```

opment of local coupled-cluster codes that take advantage of these new hardware developments, enabling coupled-cluster calculations on much larger systems than previously imaginable.

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 Xanthreas and co-workers, and 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.

ACKNOWLEDGMENTS

A.J. and C.D.S. thank Jose Madriaga for discussions on the working equations in the PNO basis set and Jonathon Misiewicz for helpful discussions on non-orthogonal LMO Fock matrices. A.J.

also thanks Dr. Gregory Tschumper for useful discussions on the large water clusters. A.J., J.M.T., and C.D.S. thank Dr. Susi Lehtola for helpful comments during the revision process of the paper. A.J., J.M.T., and HFS gratefully acknowledge the financial support from the U.S. Department of Energy, Basic Energy Sciences, Chemistry Division Computational and Theoretical Chemistry (CTC) under Grant No. DE-SC0018164. Z.L.G. and C.D.S. also gratefully acknowledge the financial support from the National Science Foundation under Grant No. CHE-1955940. D.P. was supported by a fellowship from The Molecular Sciences Software Institute under NSF Grant No. CHE-2136142.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Andy Jiang: Conceptualization (supporting); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead);

ALGORITHM 3. Semi-direct algorithm for forming $S_{b_{kl}}^{a_{ij}}$ and relevant contractions.

```

for  $ij$  in strong_pairs do                                ▷ Loop over all strong  $ij$  pairs,  $i \leq j$ 
  pair_ext_domain  $\leftarrow$  List[int]                        ▷ Form extended domain for pairs
  for  $k_{ij}, l_{ij}$  in lmopair_to_lmos[ $ij$ ]  $\times$  lmopair_to_lmos[ $ij$ ] do
     $k, l \leftarrow$  lmopair_to_lmos[ $ij$ ][ $k_{ij}$ ], lmopair_to_lmos[ $ij$ ][ $l_{ij}$ ]
     $kl \leftarrow$  lmo_pair_index[ $k$ ][ $l$ ]
    if  $kl \in$  strong_pairs  $\cup$  weak_pairs then
      pair_ext_domain = pair_ext_domain  $\cup$  lmopair_to_paos[ $kl$ ]
    end if
  end for
   $S_{ij} \leftarrow$  submatrix_rows_and_columns( $S^{\text{PAO}}$ , lmopair_to_paos[ $ij$ ], pair_ext_domain)
   $S'_{ij} \leftarrow X_{ij}^{\text{PNO}} S_{ij}$                                 ▷ Transform first index of overlap matrix to PNO space of  $ij$ 
   $B_{ij}^{a_{ij}b_{ij}} \leftarrow 0$                                 ▷ Initialize  $B_{ij}^{a_{ij}b_{ij}}$ 
   $\widetilde{F}_{b_{ij}c_{ij}} \leftarrow \widetilde{F}_{b_{ij}c_{ij}}$                                 ▷ Eq. (85a)
  for  $k_{ij}, l_{ij}$  in lmopair_to_lmos[ $ij$ ]  $\times$  lmopair_to_lmos[ $ij$ ] do
     $k, l \leftarrow$  lmopair_to_lmos[ $ij$ ][ $k_{ij}$ ], lmopair_to_lmos[ $ij$ ][ $l_{ij}$ ]
     $kl \leftarrow$  lmo_pair_index[ $k$ ][ $l$ ]
    if  $kl \in$  strong_pairs  $\cup$  weak_pairs then
       $S''_{ij} \leftarrow$  submatrix_columns( $S'_{ij}$ , pair_ext_domain  $\cap$  lmopair_to_paos[ $kl$ ])
       $S_{b_{kl}}^{a_{ij}} \leftarrow (S''_{ij})_{a_{ij}\widetilde{v}_{kl}} X_{\widetilde{v}_{kl}b_{kl}}^{\text{PNO}}$ 
       $B_{ij}^{a_{ij}b_{ij}} + = (S_{kl}^{a_{ij}} T_{kl}^{a_{kl}b_{kl}} S_{b_{kl}}^{b_{ij}}) \beta_{ij}^{kl}$                                 ▷ Eq. (77)
       $\widetilde{F}_{b_{ij}c_{ij}} - = S_{b_{kl}}^{b_{ij}} u_{kl}^{b_{kl}d_{kl}} K_{kl}^{c_{kl}d_{kl}} S_{c_{kl}}^{c_{ij}}$                                 ▷ Eq. (85b)
    end if
  end for
   $E_{ij}^{a_{ij}b_{ij}} \leftarrow t_{ij}^{a_{ij}c_{ij}} \widetilde{F}_{b_{ij}c_{ij}} + \widetilde{F}_{a_{ij}c_{ij}} t_{ij}^{c_{ij}b_{ij}}$                                 ▷ Eq. (80), second term added to account for  $P_{ij}^{ab}$ 
   $R_{ij}^{a_{ij}b_{ij}} + = B_{ij}^{a_{ij}b_{ij}} + E_{ij}^{a_{ij}b_{ij}}$                                 ▷ Add contributions to doubles residual
  if  $i \neq j$  then
     $R_{ji}^{b_{ij}a_{ij}} + = B_{ji}^{b_{ij}a_{ij}} + E_{ji}^{b_{ij}a_{ij}}$                                 ▷ Add relevant contributions for  $ji$ 
  end if
end for

```

Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (supporting). **Zachary L. Glick:** Software (supporting). **David Poole:** Data curation (supporting); Investigation (supporting); Methodology (supporting); Software (supporting); Writing – review & editing (supporting). **Justin M. Turney:** Supervision (equal); Writing – review & editing (equal). **C. David Sherrill:** Conceptualization (lead); Funding acquisition (equal); Project administration (equal); Supervision (equal); Validation (supporting); Writing – review & editing (equal). **Henry F. Schaefer:** Funding acquisition (equal); Project administration (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available within 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 Algorithms 1–3, both in [Appendixes A and B](#), 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 $T_{\text{CUT_DO}}$), `lmopair_to_ribfs[ $ij$ ]` represents all auxiliary basis functions on atoms in the Mulliken fitting domain of pair ij (determined by $T_{\text{CUT_MKN}}$), and `lmopair_to_lmos[ $ij$ ]` represents all LMOs k such that ik and kj are both strong or weak pairs.

APPENDIX B: LOW MEMORY PNO OVERLAP ALGORITHM

Here, we present our semi-direct algorithm for the formation of $S_{b_{kl}}^{a_{ij}}$ intermediates, as well as contractions into the $B_{ij}^{a_{ij}b_{ij}}$ [Eq. (77)] and $E_{ij}^{a_{ij}b_{ij}}$ [Eqs. (80) and (85)] intermediates.

REFERENCES

- ¹T. D. Crawford and H. F. Schaefer, *Rev. Comput. Chem.* **14**, 33 (2007).
- ²R. J. Bartlett and M. Musial, *Rev. Mod. Phys.* **79**, 291 (2007).
- ³C. D. Sherrill and H. F. Schaefer, *Adv. Quantum Chem.* **34**, 143 (1999).
- ⁴C. J. Cramer, *Essentials of Computational Chemistry* (Wiley, 2002), pp. 191–232.
- ⁵K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, *Chem. Phys. Lett.* **157**, 479 (1989).
- ⁶J. F. Stanton, *Chem. Phys. Lett.* **281**, 130 (1997).
- ⁷R. J. Bartlett, J. D. Watts, S. A. Kucharski, and J. Noga, *Chem. Phys. Lett.* **165**, 513 (1990).
- ⁸J. D. Watts, I. Cernusak, J. Noga, R. J. Bartlett, C. W. Bauschlicher, Jr., T. J. Lee, A. P. Rendell, and P. R. Taylor, *J. Chem. Phys.* **93**, 8875 (1990).
- ⁹G. E. Scuseria and T. J. Lee, *J. Chem. Phys.* **93**, 5851 (1990).
- ¹⁰L. Gyevi-Nagy, M. Kállay, and P. R. Nagy, *J. Chem. Theory Comput.* **16**, 366 (2020).
- ¹¹P. Hohenberg and W. Kohn, *Phys. Rev.* **136**, B864 (1964).
- ¹²W. Kohn and L. J. Sham, *Phys. Rev.* **140**, A1133 (1965).
- ¹³W. Dawson, A. Degomme, M. Stella, T. Nakajima, L. E. Ratcliff, and L. Genovese, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **12**, e1574 (2022).
- ¹⁴C. Møller and M. S. Plesset, *Phys. Rev.* **46**, 618 (1934).
- ¹⁵D. Cremer, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **1**, 509 (2011).
- ¹⁶R. M. Parrish, Y. Zhao, E. G. Hohenstein, and T. J. Martínez, *J. Chem. Phys.* **150**, 164118 (2019).
- ¹⁷E. G. Hohenstein, Y. Zhao, R. M. Parrish, and T. J. Martínez, *J. Chem. Phys.* **151**, 164121 (2019).
- ¹⁸E. G. Hohenstein, B. S. Fales, R. M. Parrish, and T. J. Martínez, *J. Chem. Phys.* **156**, 054102 (2022).
- ¹⁹M. Lesiuk, *J. Chem. Phys.* **156**, 064103 (2022).
- ²⁰R. M. Parrish, E. G. Hohenstein, T. J. Martínez, and C. D. Sherrill, *J. Chem. Phys.* **137**, 224106 (2012).
- ²¹E. G. Hohenstein, R. M. Parrish, and T. J. Martínez, *J. Chem. Phys.* **137**, 044103 (2012).
- ²²E. G. Hohenstein, R. M. Parrish, C. D. Sherrill, and T. J. Martínez, *J. Chem. Phys.* **137**, 221101 (2012).
- ²³R. M. Parrish, C. D. Sherrill, E. G. Hohenstein, S. I. L. Kokkila, and T. J. Martínez, *J. Chem. Phys.* **140**, 181102 (2014).
- ²⁴T. G. Kolda and B. W. Bader, *SIAM Rev.* **51**, 455 (2009).
- ²⁵A. Jiang, J. M. Turney, and H. F. Schaefer III, *J. Chem. Theory Comput.* **19**, 1476 (2023).
- ²⁶S. Li, J. Ma, and Y. Jiang, *J. Comput. Chem.* **23**, 237 (2002).
- ²⁷S. Li, J. Shen, W. Li, and Y. Jiang, *J. Chem. Phys.* **125**, 074109 (2006).
- ²⁸W. Li, P. Piecuch, J. R. Gour, and S. Li, *J. Chem. Phys.* **131**, 114109 (2009).
- ²⁹F. Neese, F. Wennmohs, and A. Hansen, *J. Chem. Phys.* **130**, 114108 (2009).
- ³⁰F. Neese, A. Hansen, and D. G. Liakos, *J. Chem. Phys.* **131**, 064103 (2009).
- ³¹W. Li and P. Piecuch, *J. Phys. Chem. A* **114**, 8644 (2010).
- ³²Z. Rolik and M. Kállay, *J. Chem. Phys.* **135**, 104111 (2011).
- ³³Z. Rolik, L. Szegedy, I. Ladjánszki, B. Ladóczki, and M. Kállay, *J. Chem. Phys.* **139**, 094105 (2013).
- ³⁴C. Riplinger and F. Neese, *J. Chem. Phys.* **138**, 034106 (2013).
- ³⁵C. Riplinger, B. Sandhoefer, A. Hansen, and F. Neese, *J. Chem. Phys.* **139**, 134101 (2013).
- ³⁶D. G. Liakos, M. Sparta, M. K. Kesharwani, J. M. L. Martin, and F. Neese, *J. Chem. Theory Comput.* **11**, 1525 (2015).
- ³⁷H.-J. Werner, G. Knizia, C. Krause, M. Schwilk, and M. Dornbach, *J. Chem. Theory Comput.* **11**, 484 (2015).
- ³⁸Q. Ma and H.-J. Werner, *J. Chem. Theory Comput.* **11**, 5291 (2015).
- ³⁹P. Pinski, C. Riplinger, E. F. Valeev, and F. Neese, *J. Chem. Phys.* **143**, 034108 (2015).
- ⁴⁰C. Riplinger, P. Pinski, U. Becker, E. F. Valeev, and F. Neese, *J. Chem. Phys.* **144**, 024109 (2016).
- ⁴¹M. Schwilk, Q. Ma, C. Köppl, and H.-J. Werner, *J. Chem. Theory Comput.* **13**, 3650 (2017).
- ⁴²Q. Ma, M. Schwilk, C. Köppl, and H.-J. Werner, *J. Chem. Theory Comput.* **13**, 4871 (2017).
- ⁴³P. R. Nagy and M. Kállay, *J. Chem. Phys.* **146**, 214106 (2017).
- ⁴⁴Y. Guo, C. Riplinger, U. Becker, D. G. Liakos, Y. Minenkov, L. Cavallo, and F. Neese, *J. Chem. Phys.* **148**, 011101 (2018).
- ⁴⁵Q. Ma and H.-J. Werner, *J. Chem. Theory Comput.* **14**, 198 (2018).
- ⁴⁶P. R. Nagy, G. Samu, and M. Kállay, *J. Chem. Theory Comput.* **14**, 4193 (2018).
- ⁴⁷F. Neese, F. Wennmohs, U. Becker, and C. Riplinger, *J. Chem. Phys.* **152**, 224108 (2020).
- ⁴⁸H.-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, T. F. Miller III, A. Mitrushchenkov, K. A. Peterson, I. Polyak, G. Rauhut, and M. Sibaev, *J. Chem. Phys.* **152**, 144107 (2020).
- ⁴⁹M. Kállay, P. R. Nagy, D. Mester, Z. Rolik, G. Samu, J. Csontos, J. Csóka, P. B. Szabó, L. Gyevi-Nagy, B. Hégyel, I. Ladjánszki, L. Szegedy, B. Ladóczki, K. Petrov, M. Farkas, P. D. Mezei, and Á. Ganyecz, *J. Chem. Phys.* **152**, 074107 (2020).
- ⁵⁰I. Sandler, J. Chen, M. Taylor, S. Sharma, and J. Ho, *J. Phys. Chem. A* **125**, 1553 (2021).
- ⁵¹D. G. A. Smith, L. A. Burns, A. C. Simmonett, R. M. Parrish, M. C. Schieber, R. Galvelis, P. Kraus, H. Kruse, R. Di 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).
- ⁵²N. Niemeyer, P. Eschenbach, M. Bensberg, J. Tölle, L. Hellmann, L. Lampe, A. Massolle, A. Rikus, D. Schnieder, J. P. Unsleber, and J. Neugebauer, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **13**, e1647 (2023).
- ⁵³M. Bensberg and J. Neugebauer, *J. Chem. Theory Comput.* **16**, 3607 (2020).
- ⁵⁴H. Koch, O. Christiansen, R. Kobayashi, P. Jørgensen, and T. Helgaker, *Chem. Phys. Lett.* **228**, 233 (1994).
- ⁵⁵A. E. DePrince and C. D. Sherrill, *J. Chem. Theory Comput.* **9**, 2687 (2013).
- ⁵⁶S. Yoo, E. Aprà, X. C. Zeng, and S. S. Xantheas, *J. Phys. Chem. Lett.* **1**, 3122 (2010).
- ⁵⁷D. M. Bates, J. R. Smith, T. Janowski, and G. S. Tschumper, *J. Chem. Phys.* **135**, 044123 (2011).
- ⁵⁸D. Bykov, K. Kristensen, and T. Kjærgaard, *J. Chem. Phys.* **145**, 024106 (2016).
- ⁵⁹J. F. Stanton, J. Gauss, J. D. Watts, and R. J. Bartlett, *J. Chem. Phys.* **94**, 4334 (1991).
- ⁶⁰J. L. Whitten, *J. Chem. Phys.* **58**, 4496 (1973).
- ⁶¹M. Feyereisen, G. Fitzgerald, and A. Komornicki, *Chem. Phys. Lett.* **208**, 359 (1993).
- ⁶²B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).
- ⁶³O. Vahtras, J. Almlöf, and M. W. Feyereisen, *Chem. Phys. Lett.* **213**, 514 (1993).
- ⁶⁴A. P. Rendell and T. J. Lee, *J. Chem. Phys.* **101**, 400 (1994).
- ⁶⁵F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs, *Chem. Phys. Lett.* **294**, 143 (1998).
- ⁶⁶F. Weigend, *Phys. Chem. Chem. Phys.* **4**, 4285 (2002).
- ⁶⁷A. Sodt, J. E. Subotnik, and M. Head-Gordon, *J. Chem. Phys.* **125**, 194109 (2006).
- ⁶⁸H.-J. Werner, F. R. Manby, and P. J. Knowles, *J. Chem. Phys.* **118**, 8149 (2003).
- ⁶⁹N. H. F. Beebe and J. Linderberg, *Int. J. Quantum Chem.* **12**, 683 (1977).
- ⁷⁰I. Roeggen and T. Johansen, *J. Chem. Phys.* **128**, 194107 (2008).
- ⁷¹T. B. Pedersen, S. Lehtola, I. Fdez Galván, and R. Lindh, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **14**, e1692 (2024).
- ⁷²K. E. Riley, M. Pitoňák, P. Jurečka, and P. Hobza, *Chem. Rev.* **110**, 5023 (2010).
- ⁷³A. Karton, E. Rabinovich, J. M. L. Martin, and B. Ruscic, *J. Chem. Phys.* **125**, 144108 (2006).
- ⁷⁴A. 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).
- ⁷⁵K. L. Bak, P. Jørgensen, J. Olsen, T. Helgaker, and W. Klopper, *J. Chem. Phys.* **112**, 9229 (2000).
- ⁷⁶B. W. Hopkins and G. S. Tschumper, *J. Phys. Chem. A* **108**, 2941 (2004).
- ⁷⁷J. Noga and R. J. Bartlett, *J. Chem. Phys.* **86**, 7041–7050 (1987).

- ⁷⁸A. Dutta and C. D. Sherrill, *J. Chem. Phys.* **118**, 1610 (2003).
- ⁷⁹M. Lesiuk, *J. Chem. Theory Comput.* **16**, 453 (2020).
- ⁸⁰F. Jensen, *Introduction to Computational Chemistry* (Wiley, 2007), pp. 204–208.
- ⁸¹J. M. Foster and S. F. Boys, *Rev. Mod. Phys.* **32**, 300 (1960).
- ⁸²J. W. Boughton and P. Pulay, *J. Comput. Chem.* **14**, 736 (1993).
- ⁸³J. Pipek and P. G. Mezey, *J. Chem. Phys.* **90**, 4916 (1989).
- ⁸⁴P. Pulay, *Chem. Phys. Lett.* **100**, 151 (1983).
- ⁸⁵M. Schütz, G. Hetzer, and H.-J. Werner, *J. Chem. Phys.* **111**, 5691 (1999).
- ⁸⁶M. Schütz, *J. Chem. Phys.* **113**, 9986 (2000).
- ⁸⁷M. Schütz and H.-J. Werner, *J. Chem. Phys.* **114**, 661 (2001).
- ⁸⁸R. S. Mulliken, *J. Chem. Phys.* **23**, 1833 (1955).
- ⁸⁹P. Jurečka, J. Šponer, J. Černý, and P. Hobza, *Phys. Chem. Chem. Phys.* **8**, 1985 (2006).
- ⁹⁰M. S. Marshall, L. A. Burns, and C. D. Sherrill, *J. Chem. Phys.* **135**, 194102 (2011).
- ⁹¹T. H. Dunning, *J. Chem. Phys.* **90**, 1007 (1989).
- ⁹²E. Papajak and D. G. Truhlar, *J. Chem. Theory Comput.* **7**, 10 (2011).
- ⁹³S. F. Boys and F. Bernardi, *Mol. Phys.* **19**, 553 (1970).
- ⁹⁴J. Řezáč, K. E. Riley, and P. Hobza, *J. Chem. Theory Comput.* **7**, 2427 (2011).
- ⁹⁵F. Weigend and R. Ahlrichs, *Phys. Chem. Chem. Phys.* **7**, 3297 (2005).
- ⁹⁶S. Hirata, *J. Phys. Chem. A* **107**, 9887 (2003).
- ⁹⁷A. 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).
- ⁹⁸T. Janowski, A. R. Ford, and P. Pulay, *J. Chem. Theory Comput.* **3**, 1368 (2007).
- ⁹⁹T. Janowski and P. Pulay, *J. Chem. Theory Comput.* **4**, 1585 (2008).
- ¹⁰⁰H. J. J. van Dam, W. A. de Jong, E. Bylaska, N. Govind, K. Kowalski, T. Straatsma, and M. Valiev, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **1**, 888 (2011).
- ¹⁰¹E. 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).
- ¹⁰²R. Kobayashi and A. P. Rendell, *Chem. Phys. Lett.* **265**, 1 (1997).
- ¹⁰³V. 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).
- ¹⁰⁴E. Solomonik, D. Matthews, J. R. Hammond, J. F. Stanton, and J. Demmel, *J. Parallel Distrib. Comput.* **74**, 3176 (2014).
- ¹⁰⁵C. Peng, J. A. Calvin, F. Pavošević, J. Zhang, and E. F. Valeev, *J. Phys. Chem. A* **120**, 10231 (2016).
- ¹⁰⁶D. I. Lyakh, *Int. J. Quantum Chem.* **119**, e25926 (2019).
- ¹⁰⁷P. R. Nagy, L. Gyevi-Nagy, B. D. Lorincz, and M. Kállay, *Mol. Phys.* **121**, e2109526 (2023).
- ¹⁰⁸C. 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).
- ¹⁰⁹D. Datta and M. S. Gordon, *J. Chem. Theory Comput.* **17**, 4799 (2021).
- ¹¹⁰L. Gyevi-Nagy, M. Kállay, and P. R. Nagy, *J. Chem. Theory Comput.* **17**, 860 (2021).
- ¹¹¹K. 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. Wal-drop, D. B. Williams-Young, C. Yang, M. Zalewski, and T. L. Windus, *Chem. Rev.* **121**, 4962 (2021).
- ¹¹²J. A. Calvin, C. Peng, V. Rishi, A. Kumar, and E. F. Valeev, *Chem. Rev.* **121**, 1203 (2021).
- ¹¹³D. Datta and M. S. Gordon, *J. Chem. Theory Comput.* **19**, 7640 (2023).
- ¹¹⁴S. 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).
- ¹¹⁵Z. Wang, M. Guo, and F. Wang, *Phys. Chem. Chem. Phys.* **22**, 25103 (2020).
- ¹¹⁶C. Peng, J. A. Calvin, and E. F. Valeev, *Int. J. Quantum Chem.* **119**, e25894 (2019).
- ¹¹⁷I. A. Kaliman and A. I. Krylov, *J. Comput. Chem.* **38**, 842 (2017).
- ¹¹⁸A. E. DePrince, M. R. Kennedy, B. G. Sumpter, and C. D. Sherrill, *Mol. Phys.* **112**, 844 (2014).
- ¹¹⁹W. Ma, S. Krishnamoorthy, O. Villa, and K. Kowalski, *J. Chem. Theory Comput.* **7**, 1316 (2011).
- ¹²⁰A. E. DePrince and J. R. Hammond, *J. Chem. Theory Comput.* **7**, 1287 (2011).