Electronic Supplementary Material (ESI) for Digital Discovery.

Atomic Fragment Approximation from Tensor Network

Haoxiang Lin and Xi Zhu*

School of Science and Engineering,

The Chinese University of Hong Kong, Shenzhen,

Shenzhen, Guangdong, China†

Email: zhuxi@cuhk.edu.cn

This file includes:

Supplementary Material S1: The error accumulation of DFA-NN

Supplementary Material S2: The Geometry-Enhanced Representation

Supplementary Material S3: Details on TN states representation

Supplementary Material S4: Pseudocodes

Supplementary Material S5: Mathematical interpretation for AFA

Supplementary Material S6: Physically interpretation for radical summation

Supplementary Material S7: Description of time complexity for contraction

Supplementary Material S8: Description of assisting chemical reaction prediction

Supplementary Material S9: Details on dataset Information

Supplementary Material S10: List of Selected drugs

Supplementary Material S11: Calculation details for QDF

Supplementary Material S12: More experiment results for AFA

Supplementary Material S13: The memory requirement of AFA

Supplementary Material S14: Extension of AFA in intermolecular interaction

Supplementary Material S1: The error accumulation of DFA-NN

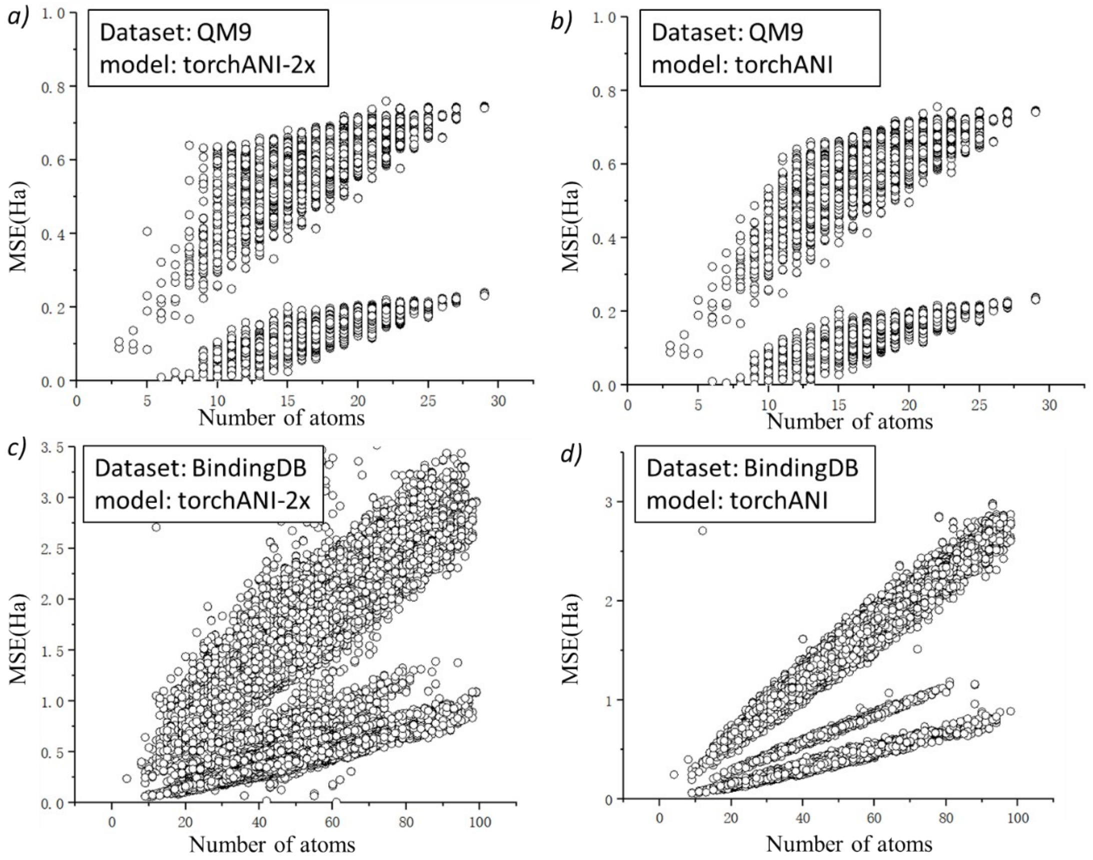

Figure S1: The performance of current state-of-the-art models, torchANI and ANI-2x, on datasets QM9 and Binding DB as labelled in the graph. The results reveal a significant tendency of error accumulation as the number of atoms in the molecule increases.

In this section, we assessed the performance of torchANI and ANI-2x. We use the sample from the dataset QM9 and BindingDB, the properties are calculated through Gaussian 09[1] under PBE level functional with basis 6-31G, including K space properties and R space properties. In the training of our model, we convert the unit of Hartree to the unit of eV and reduce the atomic energy of each atom. We can see a significant tendency of error accumulation, especially for model torchANI applied in the BindingDB dataset. This observation highlights the limitations of existing models in accurately predicting properties or reactions for larger molecules, emphasizing the need for more robust and scalable approaches in the field of molecular simulations and predictions.

Supplementary Material S2: The Geometry-Enhanced Representation

This section illustrated the detailed description of AFA, which includes the Geometry-Enhanced Representation[2], the construction of the atom's MPS, the construction of the bond's MPO, and the contraction of the entire TN structure.

In the first step, we map the molecules into the Geometry Enhanced Representation. This step is necessary since we expect the network to focus on each bond rather than their Cartesian coordinates. In the atom-bond graph G, the atom u from the entire atom set S is treated as the node of G, while the bond v between pairs of atoms from the entire bond set V serves as the edge of G. The bond-angle graph D shares a similar structure, but whose nodes are bond v from the entire bond set V, and edges are bond angles w from the entire bond angle set W. In traditional GNN structures, the only input is the atom-bond graph, while the lack of a bond-angle graph possibly raises confusion since two molecules may have the same topology but different geometries[2, 3]. A typical example comes from the trans-1,2-dichloroethene and cis-1,2-dichloroethene, whose two chlorine atoms lie on different sides. Notice that here the element of the graph matrix can be either a float representing the bond length or a simple bool value representing bond formation. The reason for this treatment is the total computation cost. Although the reliable identification of low-energy conformers for simple small molecules can be achieved through high-level quantum mechanical calculations, the computationally demanding optimized molecule structure may be even more expensive than that for property prediction based on this optimized molecular structure[4]. Meanwhile, the bond information is stored in the radical TN states. Thus, the bond information is conserved.

Take the methanol (CH_3OH) as an example, and the corresponding atom-graph matrix G can be:

$$G_{methanol} = \left[\begin{array}{ccccccc} & C & H1 & H2 & H3 & O & H4 \\ C & False & True & True & True & True & False \\ H1 & True & False & False & False & False & False \\ H2 & True & False & False & False & False & False \\ H3 & True & False & False & False & False & False \\ O & True & False & False & False & False & True \\ H4 & False & False & False & False & True & False \end{array}\right]$$

$\#(S2-1)$

In this case, we treat all bond lengths as the optimal lengths. This atom-graph matrix can also be:

$$G_{methanol} = \left[\begin{array}{ccccccc} & C & H1 & H2 & H3 & O & H4 \\ C & 1e6 & 1.096 & 1.096 & 1.096 & 1.427 & 1e6 \\ H1 & 1.096 & 1e6 & 1e6 & 1e6 & 1e6 & 1e6 \\ H2 & 1.096 & 1e6 & 1e6 & 1e6 & 1e6 & 1e6 \\ H3 & 1.096 & 1e6 & 1e6 & 1e6 & 1e6 & 1e6 \\ O & 1.427 & 1e6 & 1e6 & 1e6 & 1e6 & 0.956 \\ H4 & 1e6 & 1e6 & 1e6 & 1e6 & 0.956 & 1e6 \end{array}\right]$$

$\#(S2-2)$

Here $H1, H2, and\ H3$ are atoms that connect to the carbon atom, and $H4$ is the atom that binds to the oxygen atom. The element "1e6" will automatically be transformed into "False" during the calculation. We set the cut-off radius as "5", and any matrix element larger than five will automatically be transformed into false by default. The bond-angle graph matrix D can be:

$$D_{methanol} = \left[\begin{array}{cccccc} & C-O & C-H1 & C-H2 & C-H3 & O-H4 \\ C-O & False & True & True & True & True \\ C-H1 & True & False & True & True & False \\ C-H2 & True & True & False & True & False \\ C-H3 & True & True & True & False & False \\ O-H4 & True & False & False & False & False \end{array}\right]$$

$\#(S2-3)$

OR

$$D_{methanol} = \left[\begin{array}{cccccc} & C-O & C-H1 & C-H2 & C-H3 & O-H4 \\ C-O & 1e6 & 108.5 & 108.5 & 108.5 & 107 \\ C-H1 & 108.5 & 1e6 & 108.6 & 108.6 & 1e6 \\ C-H2 & 108.5 & 108.6 & 1e6 & 108.6 & 1e6 \\ C-H3 & 108.5 & 108.6 & 108.6 & 1e6 & 1e6 \\ O-H4 & 107 & 1e6 & 1e6 & 1e6 & 1e6 \end{array}\right]$$

$\#(S2-4)$

Any matrix element larger than 360, including the element "1e6", will automatically be transformed into "False" during the calculation. The codes for generating Geometry-Enhanced Representation can be found in ref[2],

Supplementary Material S3: Details on TN states representation

We first map each atom into its corresponding TN states in this work. This mapping contains two steps, the self-contribution, and the nearest-contribution. The first step gives initial TN states for each atom, while the nearest atoms further modify these TN states.

First, the initial TN states for the atom $A^{0}(a)$ is estimated using the feature vector of each atom V_{a}, here the shape of TN states is a hyper-parameter, but it is the same for all atoms.

$$A^{0}(a) = f^{encode}(V_{a})$$

$\#(S3-1)$

The encoding layer is an MLP, which takes the dimensionality reduction that reduces the computation complexity. The feature vector contains the orbital information, the atomic weight, and the atomic number:

$$V_{a} = \begin{bmatrix} Orbital\ informations \\ atomic\ informations \end{bmatrix}$$

$\#(S3-2)$

For the orbital information, we use the orbital points (OPs) to simulate the shape of orbitals. Equal-distant points are selected around the centers within a given radius. We calculate the wave function value for these points using the hydrogen atom wave functions. In the database, we do not have transition metals, so we only consider the s, p, and d orbital. The information of the f orbital can be added similarly. Notice that here only the outer-most orbitals are taken into consideration. Take the carbon atom as an example. The outer-most orbitals are 2s and 2p, whose wavefunction is approximated by the hydrogen atom orbitals:

$$|s(r)\rangle = \zeta\left(2 - \frac{r}{a_{0}}\right)\exp\left(-\frac{r}{2a_{0}}\right)$$

$$|p(r)\rangle = \zeta\frac{r}{a_{0}}exp\left(-\frac{r}{2a_{0}}\right)cos(\theta)$$

$\#(S3-3)$

where r is the relative distance from the point to the atom centre, θ is the angle between the $x/y/z$ axis and the $p_{x/y\,/\,z}$ orbital. The parameter ζ is used to ensure normalization. Here we set the value a_0 to be a tunable variable depending on the atom type. By default, this orbital information takes two s orbital, six p orbital, and ten d orbitals, but their values are zero if no electron occupies them.

Besides OPs that represent the orbital shape, the correlation energy is be concerned. In Nesbet's theorem[5], the correlation energy can be written exactly as a sum of contributions from occupied pairs of spin orbitals (I, J),

$$E_{corr} = \sum_{IJ} \epsilon_{IJ}$$

$\#(S3-4)$

For closed-shell systems, this energy can be specified as:

$$\epsilon_{ij} = \epsilon_{i\uparrow j\uparrow} + \epsilon_{i\uparrow j\downarrow} + \epsilon_{i\downarrow j\uparrow} + \epsilon_{i\downarrow j\downarrow}$$

$\#(S3-5)$

However, here we are seeking to construct the map through

$$\epsilon_{IJ} = \sum_{AB} \langle \Phi_0 | H | \Phi_{IJ}^{AB} \rangle \langle \Phi_{IJ}^{AB} | \psi_0 \rangle$$

$\#(S3-6)$

where $|\Phi_0\rangle$ is the HF ground state, $|\Phi_{IJ}^{AB}\rangle$ is the HF excited state by exciting orbital I, J to orbital A, B, and $|\psi_0\rangle$ is the true ground state. However, the computation cost to calculate all the pairs of energy scales O naively $(N_{orbital}^{5})$ so we adapted the correlation energy and its related values for each atom calculated from ref[6].

The atomic information includes the atomic number, the standard atomic weight, the reference energy, and the number of outmost electrons. (6, 12.0096, -37.846772,2 for carbon). Notice that this mapping step only depends on the atomic type.

The nearest atoms also modify the TN states here we calculate the terms of modification from the feature vectors:

$$C(a) = f^{contri}(A^0(a))$$

$\#(S3 - 7)$

Then we take the matrix product to obtain the final TN states:

$$A(a) = A^0(a) + C \cdot D$$

$\#(S3 - 8)$

Here only the nearest atoms contribute to the atom's MPS, but if we want to consider the second atoms, we only need to take the matrix product iteratively:

$$A^{2nd}(a) = A(a) + f^{contri}(A(a)) \cdot D$$

$\#(S3 - 9)$

Similar cases hold for the third-nearest case.

The construction of MPOs for bonds is similar to constructing MPSs for atoms, but here we are utilizing the bond-angle graph matrix.

Supplementary Material S4: Pseudocodes

Algorithm 1: Estimating properties through atoms

```
Input: the Geometry-Enhanced Representation of the molecule, including the bond-angle graph $D$ and the atom-bond graph $G$
Output: the desired property $P$, like the atomization energy
1. For each atom $a$, find the corresponding feature vector $V(a)$. This feature vector includes the orbital information and the atomic number.
2. $A^0(a) = f^{encode}(V_a)$
3. $C(a) = f^{contri}(A^0(a))$
4. Update the MPS of each atom, $A = A^0 + C \cdot D$. Here the TN state of each atom is modified using the nearest atoms.
5. Repeat steps 1-4 for bond-angle graph D, now we obtain multiple MPOs of bonds $O^0, O^0, ..., O^K$.
6. Contract the entire tensor network, i.e. $T = A \cdot (DO) \cdot A$
7. Calculate the final output $P = f^{decode}(T)$
```

The MPS of each atom only depends on itself and its nearest atoms, while this algorithm can be extended to second/third/… nearest atoms by extending the feature vectors.

Algorithm 2: The addition of properties

```
Input: The contracted result of two fragment $T_1$ and $T_2$ without the atom that forms the bond connecting these two fragments. The feature vector of two atoms that forms this bond $V(a_1)$ and $V(a_2)$. The feature vector of the nearest atoms for these two atoms.
Output: the desired property $P$, like the atomization energy
1. Calculate the MPS of these two atoms using algorithm 1.
2. Calculate the MPO of the bond that connects these two fragments.
3. Calculate the contracted result addition $T_{connect} = A(DO)A$
4. $T \ = T_1 \ + T_2 \ + \ T_{connect}$
5. Calculate the final output $P = f^{decode}(T)$
```

Supplementary Material S5: Mathematical interpretation for AFA

In this work, we mainly focus on the prediction of atomic property P with a given configuration f_i of N atoms. Here the atomic property is atomization energy, which is a scalar, and the features, atom-bond graph together with bond-angle graph has SO(3) invariant atomic representations. A natural starting point is to formulate a coarse-grained model given by the summation of all dependent variables, $P = p^0 + \sum_{f_i} p(f_i)$, where p^0 is a fixed constant and $p(f_i)$ calculates the property contribution from each feature. This model can be extended by adding the interaction terms between multiple dependent features, $P = p^0 + \sum_{f_i} p(f_i) + \sum_{f_i,f_j} p(f_i,f_j) + \sum_{f_i,f_j,f_k} p(f_i,f_j,f_k) + ...$ The interaction terms can be decomposed by projecting them into some particular basis sets

$\Phi_{s_1,s_2,\dots s_n}$, like the STO-3G, 6-31G, and cc-pVDZ basis sets, where

$$P(f_i,f_j,\dots) = \sum_{\{s_1,s_2,\dots s_n\}} C_{s_1,s_2,\dots s_n}\Phi_{s_1,s_2,\dots,s_n}(f_1,f_2,\dots f_n)$$

$\#(S5-1)$

with the general interaction coefficients $C_{s_1,s_2,\dots s_n}$.

Taking a chain molecule as an example, In this work's framework, each atom inside the molecule is transferred to a "node", a matrix containing all necessary information. The contraction of the entire tensor network is taking the matrix production. Let us assume that the total atomization energy is simply the summation of each atom's energy. The matrix of each atom is a 2-by-2 matrix. The identity matrix multiplies its energy. Now the network refers to the case: $P = p^0 + \sum_{f_i} p(f_i)$ If we further do a simplification, adding the matrix of the internal atom's tensor by a matrix M_2:

$$M_2 = \begin{bmatrix} I & 0 & 0 \\ E_{2term} & 0 & 0 \\ 0 & E_{2term} & I \end{bmatrix}$$

$\#(S5-2)$

Now the contraction of all these tensors gives a two-term summation, like the $P = P^0 + \sum_{f_i} p(f_i) + \sum_{f_i,f_j} p(f_i,f_j)$. Also, if the three-body correlation term is also required, like the contribution from bond angles, i.e. three-body correlation or the stretching of orbitals, adding the matrix of the internal atom's tensor by a matrix M_3:

$$M_3 = \begin{bmatrix} I & 0 & 0 & 0 \\ E_{3term} & 0 & 0 & 0 \\ 0 & E_{3term} & 0 & 0 \\ 0 & 0 & E_{3term} & I \end{bmatrix}$$

$\#(S5-3)$

Supplementary Material S6: Physically interpretation for radical summation

In this part, we talk about the way to estimate energy from each radical fragment. Here we focus on a special case, a molecule M with two radicals A and B, like the molecule CH3OH is made up of radical CH3- and radical -OH. The general assumption is that:

1. The total Hamiltonian H_{total} is the summation of two parts plus an extra potential term, i.e. $H_{total} = H_A + H_B + V$
2. The molecular orbital Φ is the linear combination of all the valence atomic orbitals χ_r from both radicals $\mathcal{X}$, i.e. $\psi = \sum_i c_i \chi_i$, while these valence atomic orbitals are always constant.

The term V in assumption 1 has multiple forms. In Semi-Empirical SCF Molecular Orbital Treatment, this term is estimated through a one-centre repulsion integral; this term can also be the radical stabilization energy in intermolecular orbital theory.

Therefore, the natural idea is to utilize the perturbation theory for estimation. The total Hamiltonian matrix $H_{total} = H_A + H_B + V = H_0 + V$, where H_0 simulates the system where radical A and radical B are entirely identical. Now the eigenvector of the matrix H_0 should be:

$$\left|\psi_{total}^{0}\right\rangle = \left|\psi_A^0\right\rangle \otimes \left|\psi_B^0\right\rangle$$

$\#(S6-1)$

Assuming the Hamiltonian H_{total} is a $N \times N$ matrix, where the first m terms come from radical A and the term $m+1$ to N corresponds to radical B. The elements H_{ij} of H_{total} is the same as that for H_0 for $(i,j) \leq m$ and $m+1 \leq (i,j)$. The element H_{ij} are 0 for H_0, but only several terms H_{ij} are nonzero terms for H_{total}, for $(i \leq m \& j > m)\ OR\ (i > m \& j \leq m)$. An assumption is that only close-enough terms are nonzero, as the carbon atom's orbital from radical CH3- and the oxygen atom's orbital from radical -OH.

Here we take the perturbation theory. The first-order derivative term shall be:

$$E^1 = \int \left|\psi_A^0\right\rangle V \left|\psi_B^0\right\rangle$$

#(S6 − 1)

Notice that this is a constant only depending on the type of radical A and radical B. The second-order derivative term shall be:

$$E^2 = \sum_{(m,n) \neq (i,j)} \frac{|V_{mn}|^2}{E_n^0 + E_m^0 - E_i^0 - E_j^0}$$

#(S6 − 2)

Where the subscript m,i (n,j) covers all available orbitals from radical A (B), notice now that the energy term also depends only on the type of radical A and B. Meanwhile, all of them are constant.

In the previous demonstration, we have shown that the TN framework can easily simulate linear regression, while in this case, the contraction of two connected radicals can be treated as the perturbation theory approach for estimating the molecular energy.

Supplementary Material S7: Time complexity for contraction

Here we give the proof of constant complexity for radical contraction. Recall that the MPS and MPO are obtained by the surrounding environment, as shown in equations (1) and (2). Since the number of atoms/bonds around each atom/radical/bond must have an upper limit, the complexity of obtaining MPS state A and the MPO O has an upper limit C_M. The revision of obtained MPS and MPO are written in the previous section. Equation (4) has two parts: the tensor network contraction process and the MLP forwarding process. Since the input of MLP is always a vector with a fixed dimension, the complexity of this MLP forwarding process is always a constant C_{NN}. Now the only problem is the tensor network contraction process. Here we refer to the graph matrix M_C of a molecule C formulated by the contraction of two radicals A and B:

$$M_C = \begin{bmatrix} M_A & M_{AB} \\ M_{AB} & M_B \end{bmatrix}$$

#(S7 − 1)

The contracting result (the element in equation 4) is actually from this graph matrix. Recall that in assumption, the graph matrix M_A and M_B are already known, and there only exists one bond that connects radical A with radical B. Therefore, the complexity of this bond must be a constant C_b. Aggregating all the above complexities, the total complexity of radical contraction must not exceed a constant:

$$C_{total} = n \times C_M + C_{NN} + C_b$$

$\#(S7-2)$

Here n is the atom that lies near the formulated bond, which is typically two.

In this work, we are dealing with the connecting graph for each molecule. Here we use the calculation of methanol CH_3OH as an example by adding two radicals, methyl $-CH_3$ and hydroxyl $-OH$. The connecting matrix $M_{methanol}$ for the entire methanol is:

$$M_{methanol} = \left[\begin{array}{cccccc} & C & H1 & H2 & H3 & O \\ C & 0 & TN(C,H1) & TN(C,H2) & TN(C,H3) & TN(C \\ H1 & TN(C,H1) & 0 & 0 & 0 & 0 \\ H2 & TN(C,H2) & 0 & 0 & 0 & 0 \\ H3 & TN(C,H3) & 0 & 0 & 0 & 0 \\ O & TN(C,O) & 0 & 0 & 0 & 0 \\ H4 & 0 & 0 & 0 & 0 & TN(O \end{array}\right.$$

$\#(S7-3)$

Where the function TN corresponds to the tensor network output, like $TN(CH1)$ corresponds to the TN output from the carbon *C* and hydrogen atom *H1*. This matrix is decomposed into $M_{methanol} = M_{CH3} + M_{OH} + M_{connect}$ with:

$$M_{methyl} = \left[\begin{array}{ccccccc} & C & H1 & H2 & H3 & O & H4 \\ C & 0 & TN(C,H1) & TN(C,H2) & TN(C,H3) & 0 & 0 \\ H1 & TN(C,H1) & 0 & 0 & 0 & 0 & 0 \\ H2 & TN(C,H2) & 0 & 0 & 0 & 0 & 0 \\ H3 & TN(C,H3) & 0 & 0 & 0 & 0 & 0 \\ O & 0 & 0 & 0 & 0 & 0 & 0 \\ H4 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$M_{OH} = \left[\begin{array}{ccccccc} & C & H1 & H2 & H3 & O & H4 \\ C & 0 & 0 & 0 & 0 & 0 & 0 \\ H1 & 0 & 0 & 0 & 0 & 0 & 0 \\ H2 & 0 & 0 & 0 & 0 & 0 & 0 \\ H3 & 0 & 0 & 0 & 0 & 0 & 0 \\ O & 0 & 0 & 0 & 0 & 0 & TN(O,H) \\ H4 & 0 & 0 & 0 & 0 & TN(O,H) & 0 \end{array}\right]$$

$$M_{connect}=\left[\begin{array}{ccccccc} & C & H1 & H2 & H3 & O & H4 \\ C & 0 & 0 & 0 & 0 & TN(C,O) & 0 \\ H1 & 0 & 0 & 0 & 0 & 0 & 0 \\ H2 & 0 & 0 & 0 & 0 & 0 & 0 \\ H3 & 0 & 0 & 0 & 0 & 0 & 0 \\ O & TN(C,O) & 0 & 0 & 0 & 0 & 0 \\ H4 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$\#(S7-4)$

It should be noted that M_{CH3} only depends on radical methyl and M_{OH} only depends on the radical hydroxyl. Notice that the connection graph has only one or few elements since they correspond to the bond formatted through the "adding" of these two radicals. Therefore, $M_{connect}$ has a fixed complexity O(1). Although we use the example of a molecule formed through two radicals, this fixed complexity consequence also holds for other similar cases, like the radical formed from two radicals.

Supplementary Material S8: Description of assisting chemical reaction prediction

Our method for assisting chemical reaction prediction using AFA consists of the following steps:

1. Calculate the energy of reactants and products using AFA, storing intermediate fragment results.
2. Identify all bonds among reactants and products and calculate bond-breaking energies, excluding those involved in ring breaking or formation. Store bond energies for those that can be broken but not for those changing types due to ring breaking or formation.
3. Find all possible combinations of bond breaking, assuming that intermediates are formed only through bond breaking without bond formation.
4. Calculate the bond-breaking energy for all possible combinations, using a summation of energies calculated in Step 2. Exclude alternatives with excessively high energy or reactions that involve ring breaking or formation; recalculate energy for these fragments as bond types change.
5. Determine the resulting molecules for both reactants and products by breaking chosen bonds and identifying the resulting molecular structures. Generate two

alternative choice lists of intermediates for reactants and products.

6. Compare the alternative choice lists of intermediates for reactants and products, selecting intermediates that appear in both sets.
7. Calculate the energy of all selected intermediate molecules using the summation of fragments stored in Step 1. Choose the intermediate with the lowest transition energy as the predicted intermediate.

This comprehensive method accounts for various bond-breaking scenarios, allowing for accurate predictions of intermediates in chemical reactions. Generally speaking, small effects may affect the possible reaction pathways, so further validation and refinement are required to clearly identify the reaction pathway.

Supplementary Material S9: Dataset Information

In this work, we create a molecule dataset that contains one million molecules. The topology of molecules is obtained from QM9[7], bindingDB[8], Chembl[9], and BDE[10]. The unstable structures that failed the consistency check are removed. Molecules with the number of atoms larger than 100 are removed. Properties are calculated through Gaussian 09[1] under PBE level functional with basis 6-31G, including K space properties and R space properties. These properties include:

1. SMILES
2. The atom positions (x, y, z)
3. The self-consistent field energy
4. The orbital energies, including HOMO, HOMO-1, HOMO-2, HOMO-3, HOMO-4, HOMO-5
5. The orbital energies, including LUMO, LUMO+1, LUMO+2, LUMO+3, LUMO+4, LUMO+5

We transformed the SCF energy into the atomization energy in units of eV, which can be calculated using “Atomref” of the QM9 dataset[7], that is, $E = U_0 - \sum_{i \in all\ atoms} E_i$, and 1 Hartree = 27.2114 eV. The “Atomref” of the QM9 dataset can be found from:

https://figshare.com/articles/dataset/Atomref_Reference_thermochemical_energies_of_H_C_N_O_F_atoms_/1057643?backTo=/collections/Quantum_chemistry_structures_and_properties_of_134_kilo_molecules/978904

Supplementary Material S10: List of Selected drugs

The SMILES and names of selected drugs are listed below, these data are achieved from BindingDB[8] and ChEMBL[9]:

	Name	SMILES
1	8-phenyl-octanecarboxamide peptidomimetic, 49	COCCCOc1cc(C[C@@H](C[C@H](N)[C@@H](O)C[C@@H](C)C(=O)NCCN2CCCCC2)C(C)C)ccc1OC
2	17-Hydroxywortmannin	COC[C@H]1OC(=O)C(=CN(CCCN(C)C)CCCN(C)C)C2C(=O)C(=O)C3=C([C@@H](C[C@]4(C)[C@@H](O)CCC34)OC(C)=O)[C@@]12C
3	Dipyridamole Analogue, 15	OCCN(CCO)c1nc(N2CCCCCCCC2)c2nc(nc(N3CCCCCCCC3)c2n1)N(CCO)CCO
4	(3S,6S,9S,12R)-3-[(2R)-butan-2-yl]-6-[(1-methoxyindol-3-yl)methyl]-9-(6-oxooctyl)-1,4,7,10-tetrazabicyclo[10.4.0]hexadecane-2,5,8,11-tetrone	CC[C@@H](C)[C@@H]1NC(=O)[C@H](Cc2cn(OC)c3ccccc23)NC(=O)[C@H](CCCCCC(=O)CC)NC(=O)[C@H]2CCCCN2C1=O
5	3-(2,4-Difluoro-phenyl)-1-heptyl-1-[5-(1,4,5-triphenyl-1H-imidazol-2-ylsulfanyl)-pentyl]-urea	CCCCCCCN(CCCCCSc1nc(c(-c2ccccc2)n1-c1ccccc1)-c1ccccc1)C(=O)Nc1ccc(F)cc1F
6	Aliskiren	COCCCOc1cc(CC(CC(N)C(O)CC(C(=O)NCC(C)(C)C(N)=O)C(C)C)C(C)C)ccc1OC
7	(S)-5-Guanidino-2-{(S)-2-[(S)-3-(3H-imidazol-4-yl)-2-(3,3,3-trifluoro-propionylamino)-propionylamino]-3-phenyl-propionylamino}-pentanoic acid [(S)-1-carbamoyl-2-(1H-indol-3-yl)-ethyl]-amide	NC(=N)NCCC[C@H](NC(=O)[C@H](Cc1ccccc1)NC(=O)[C@H](Cc1cnc[nH]1)NC(=O)CC(F)(F)F)C(=O)N[C@@H](Cc1c[nH]c2ccccc12)C(N)=O
8	Ac-YFR-AMOK 10b	CC(=O)NC(Cc1ccc(O)cc1)C(=O)NC(Cc1ccccc1

		)C(=O)NC(CCCN=C(N)N)C(=O)COC(=O)c1c(C)cccc1C
9	Gly-Ala-Val-Val-Asn-Asp-Leu	CC(C)C[C@H](NC(=O)[C@H](CC(O)=O)NC(=O)[C@H](CC(N)=O)NC(=O)[C@@H](NC(=O)[C@@H](NC(=O)[C@H](C)NC(=O)CN)C(C)C)C(C)C)C(O)=O
10	hentriacont-12-ynoic acid	CCCCCCCCCCCCCCCCCCC#CCCCCCCCCCCC(=O)O
11	N-[(1-Cyclohexylmethyl-2,3-dihydroxy-5-methyl-hexylcarbamoyl)-ethylsulfanyl-methyl]-2-(morpholine-4-sulfonylamino)-3-phenyl-propionamide	CCSC(NC(=O)[C@H](Cc1ccccc1)NS(=O)(=O)N1CCOCC1)C(=O)N[C@@H](CC1CCCCC1)[C@@H](O)[C@@H](O)CC(C)C
12	N-[(1-Cyclohexylmethyl-2,3-dihydroxy-5-methyl-hexylcarbamoyl)-ethylsulfanyl-methyl]-2-(morpholine-4-sulfonylamino)-3-phenyl-propionamide	CCSC(NC(=O)C(Cc1ccccc1)NS(=O)(=O)N1CCOCC1)C(=O)NC(CC1CCCCC1)C(O)C(O)CC(C)C
13	(2S)-2-[[(2S)-2-[[(2S)-2-[[(2S,3S)-3-amino-2-hydroxy-1-oxodecyl]-methylamino]-4-methylsulfinyl-1-oxobutyl]amino]-3-(4-hydroxyphenyl)-1-oxopropyl]amino]-3-(4-hydroxyphenyl)propanoic acid	CCCCCCC[C@H](N)[C@H](O)C(=O)N(C)[C@@H](CCS(C)=O)C(=O)N[C@@H](Cc1ccc(O)cc1)C(=O)N[C@@H](Cc1ccc(O)cc1)C(O)=O
14	Beta-carotene	C\C(\C=C\C=C(/C)\C=C\C1=C(C)CCCC1(C)C)=C/C=C/C=C(\C)/C=C/C=C(\C)/C=C/C1=C(C)CCCC1(C)C

15	N-[4-[(4-ethyl-1-piperazinyl)methyl]-3-(trifluoromethyl)phenyl]-4-methyl-3-[(E)-2-[5-[[2-methyl-6-(1-piperazinyl)-4-pyrimidinyl]amino]-2-pyrazinyl]ethenyl]benzamide	CCN1CCN(Cc2ccc(NC(=O)c3ccc(C)c(\C=C\c4cnc(Nc5cc(nc(C)n5)N5CCNCC5)cn4)c3)cc2C(F)(F)F)CC1
16	(2S)-2-[[(2S)-2-[[(2S)-2-[[(2S,3S)-3-amino-2-hydroxy-1-oxodecyl]-methylamino]-4-methylsulfinyl-1-oxobutyl]amino]-3-(4-hydroxyphenyl)-1-oxopropyl]amino]-3-(4-hydroxyphenyl)propanoic acid	CCCCCCC[C@H](N)[C@H](O)C(=O)N(C)[C@@H](CCS(C)=O)C(=O)N[C@@H](Cc1ccc(O)cc1)C(=O)N[C@@H](Cc1ccc(O)cc1)C(O)=O
17	(5R)-1-[[(2S)-1-[(2S)-2-[(4R)-2-azanylidene-3-[2-(3,4-dichlorophenyl)ethyl]-4-(phenylmethyl)imidazolidin-1-yl]pentyl]pyrrolidin-2-yl]methyl]-5-(2-methylpropyl)-4,5-dihydroimidazol-2-amine	CCC[C@@H](CN1CCC[C@H]1CN1[C@H](CC(C)C)CN=C1N)N1C[C@@H](Cc2ccccc2)N(CCc2ccc(Cl)c(Cl)c2)C1=N

Supplementary Material S11: Calculation details for QDF

In this work, we use QDF as an example for DFA-simulating NN. We use the model published online, one may access it through:

https://github.com/masashitsubaki/QuantumDeepField_molecule

The adapted parameter is the same as the published example. Specifically, the training dataset is the subset of QM9, whose number of atoms is smaller than or equal to 14. The test dataset is another subset of QM9, whose number of atoms is larger than 15. The grid radius is set as 0.75 Å, and the grid interval is set as 0.3 Å. The training basis is 6-31G. The decoding neural network has an intermediate dimension of 250, with a depth of 3 layers. The operation is set as ‘sum’ for the R space property, while it is set as ‘mean’ for HOMO and LUMO. The batch size for training is set as 4, the learning rate is set as 1e-4, and the learning rate decay is set as 0.4. The step size for learning rate decay is 200. The total number of iterations is set as 2000.

Even though the parameter might not be optimal, the tendency of error accumulation always holds for DFA-simulating NN.

Supplementary Material S12: More experiment results for AFA

We first talk about the experiment result for the QM9 dataset. In this experiment, we do not optimal the intermediate radical TN basis. The results are obtained directly from the contraction of atoms. The entire dataset is divided into two parts, the small molecule dataset for training and the large molecule dataset for testing. Here the training (testing) datasets only contain molecules whose total number of atoms is smaller (larger) than 16. This division is conducted since the DFT calculation can easily deal with small molecules with an acceptable time-costing and spatial costing, but such complexity quickly increases exponentially with molecule size. We set the separation to be 16 since we want to include the methylbenzene inside training sets. Therefore the geometry of aromatic molecules can be learned.

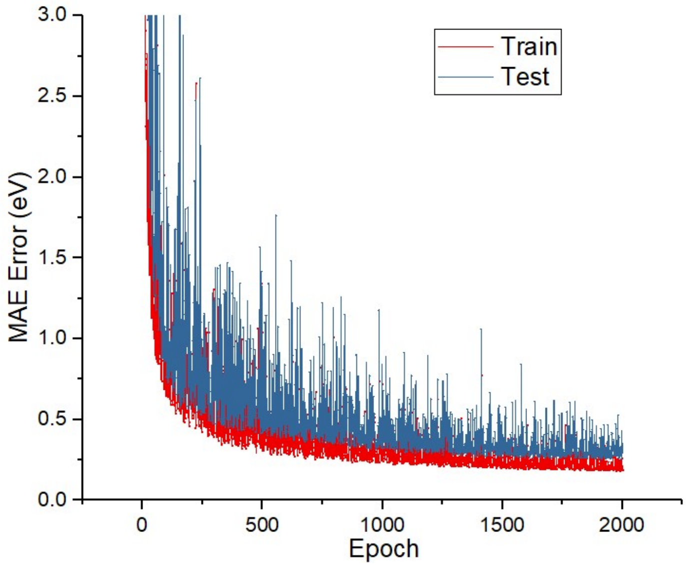

Figure S12-1: The convergence graph. This learning curve is performed on the QM9 dataset, whose training (testing) datasets only contain molecules whose total number of atoms is smaller (larger) than 16. This curve converges at approximately 500 epochs, while finally, the mean absolute error (MAE) is approximately 0.18 eV for the training dataset and 0.25eV for the testing dataset.

Figure S12-1 shows the convergence graph while the error is the mean absolute error for each molecule. For the first 100 epochs, the training error and the testing error drop down exponentially since the initial guess is approximately 0 eV for each molecule. During the first several epochs, the model learned some knowledge from it. After 500 epochs, the learning curve is still fluctuating significantly, but it shows a tendency to be flattened.

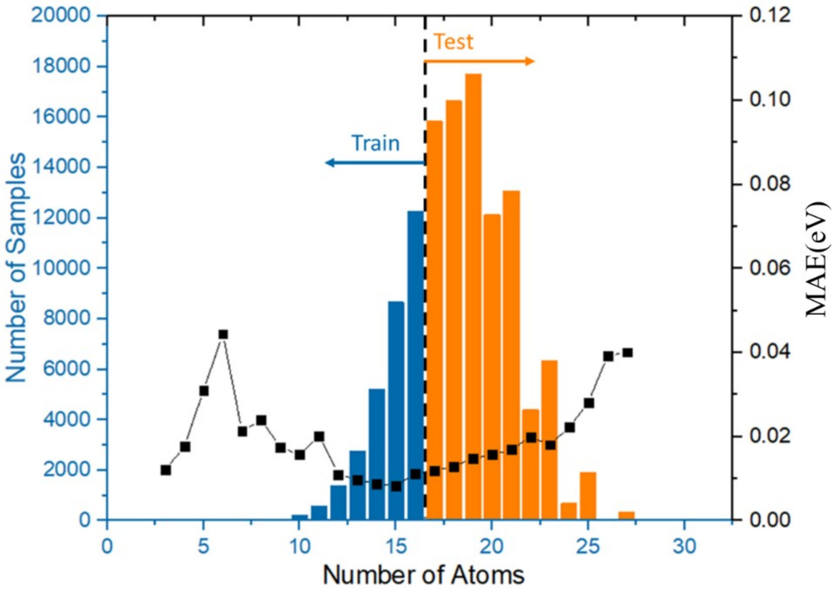

Figure S12-2: Error and number of samples, using only the QM9 database. The histogram represents the data distribution of training and testing samples. Here the training (testing) datasets only contain molecules whose total number of atoms is smaller (larger) than 16. The black curve is the average mean absolute error (MAE) for each atom. The error does not increase a lot despite different datasets.

The dataset information is shown in Figure S12-2, the histogram. We can see that the testing dataset is approximately ten times larger than the training set. The black curve in Figure S12-2 shows the mean absolute error for each atom with a different subset. The error of molecules whose number of atoms is smaller than 8 is large, a possible reason is the lack of input samples. The model may hardly obtain the features. However, a fortune is that the DFT algorithm can easily deal with these molecules without the insistence of deep learning models. For the subsets whose number of samples is larger than 1000, the average error quickly dropped down. A noticeable feature is that the error does not increase even for the testing sets. Now features of molecules in testing sets are distinct from those in the training set, in the aspect of the number of atoms. However, no significant error is observed.

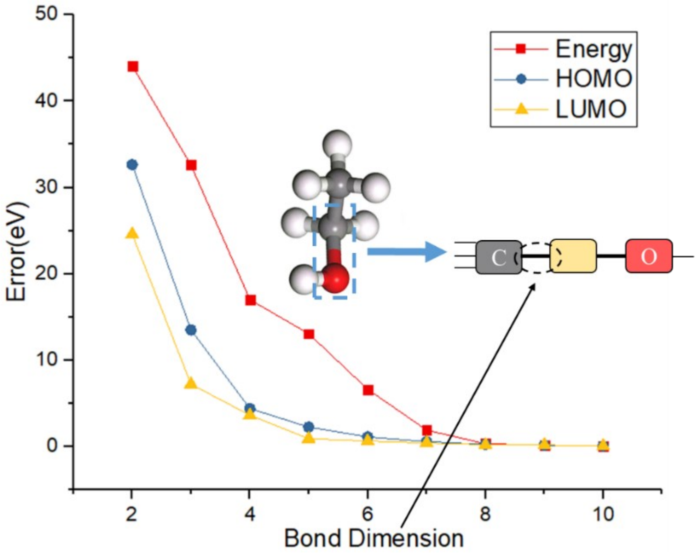

Figure S12-3: the error convergence with bond dimension. Here the bond dimension refers to the dimension of the TN tensor (equation 2 in the manuscript). We can see that with an increment of bond dimension, the error is quickly reduced. It is expected that with sufficient computation conditions, better accuracy can be obtained.

Figure S12-3 shows the MSE error with dependence on bond dimension. One of the hyper-parameters is the bond dimension of MPO and MPS. The bond dimension represents the number of parameters included in the calculation. A higher bond dimension means a higher number of parameters included during the calculation process. We can see that the error quickly drops with the increment of bond dimension. However, our device does not support a large bond dimension, but better performance is expected with the development of devices.

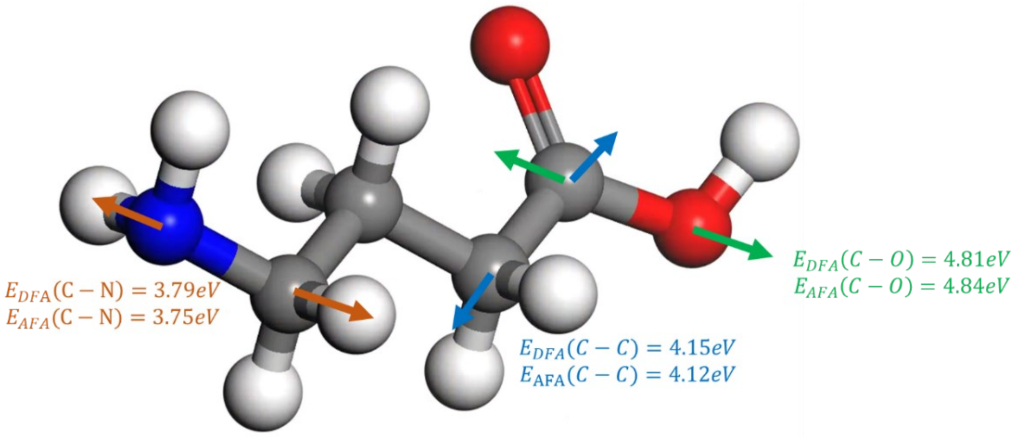
$E_{DFA}(C-O) = 4.81eV$
$E_{AFA}(C-O) = 4.84eV$
$E_{DFA}(\mathrm{C-N}) = 3.79eV$
$E_{AFA}(\mathrm{C-N}) = 3.75eV$
$E_{DFA}(C-C) = 4.15eV$
$E_{\mathrm{AFA}}(C-C) = 4.12eV$

Figure S12-4: The calculation of bond dissociation energy. Here the brown, blue, and green arrows refer to the bond dissociation energy for the target C-N, C-C, and C-O bonds. We notice that the predicted bond dissociation energy has a great agreement with DFA results.

Thanks to the ability to predict the energy of radicals, AFA can calculate the bond dissociation energy. The bond dissociation energy is calculated from the difference between two separated radicals and the entire molecules. Figure S12-4 shows an example of a molecule. We can see that the predicted dissociation energy is close to the DFA results. It is common that for a colossal structure, researchers simply focus on a few local bonds. DFA must calculate the whole electron density of each molecule, as well as all radicals separated from this molecule. This process leads to redundant calculation, which can be improved through AFA.

We train the model with the following hyper-parameters: The embedding layer maps each atom into an MPS with bond dimension 10. The contracted results of the TN structure go through a deep neural network with four hidden layers, while each hidden layer has 250 neurons, their activation functions are set as the ReLU function. We use the PyTorch framework to build the models and use a batch stochastic gradient descent optimizer using the Adam method with batch size 64 to optimize the model. The learning rate is set to 1e-4 initially, and the learning rate will be halved after every 200 iterations, the overall training procedure contains 2000 epochs. These parameters are the same for both real space property estimation and reciprocal space property estimation.

Supplementary Material S13: the memory requirement of AFA

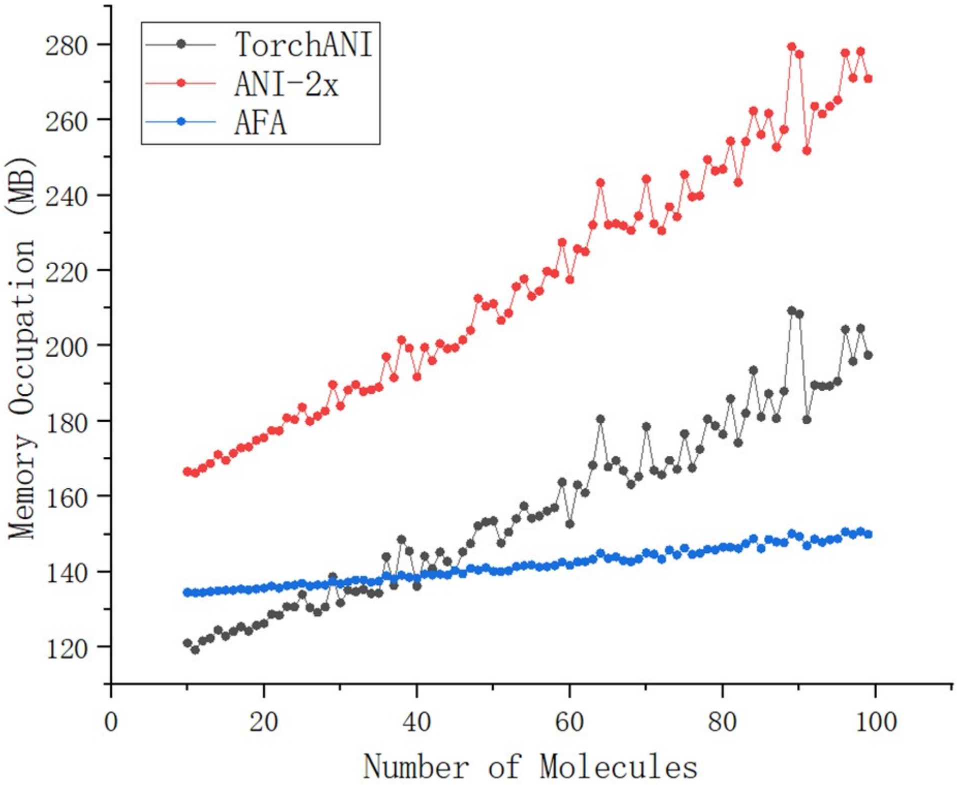

Figure S13-1: The memory usage comparison of the TorchANI, ANI-2x, and AFA. The red curve with dots represents the TorchANI model, the black curve is for ANI-2x, and the blue curve is for the AFA model. It can be observed that the memory requirement of AFA remains relatively flat in comparison to the increasing memory requirements of both the TorchANI and ANI-2x models.

Figure S13-1 presents a comparison of memory usage for TorchANI, ANI-2x, and AFA. We first divided the entire dataset into several sub-datasets, each containing samples with a fixed number of atoms. From each subset, we randomly selected 30 samples. We then employed TorchANI, ANI-2x, and AFA models to estimate the properties of these samples and recorded their respective memory usage. The memory requirement of AFA remains relatively flat compared to the increasing memory requirements of both TorchANI and ANI-2x models. This is because the memory requirement of AFA generally grows with respect to radicals, rather than atoms, while the memory requirements of TorchANI and ANI-2x models primarily increase with respect to the number of atoms.

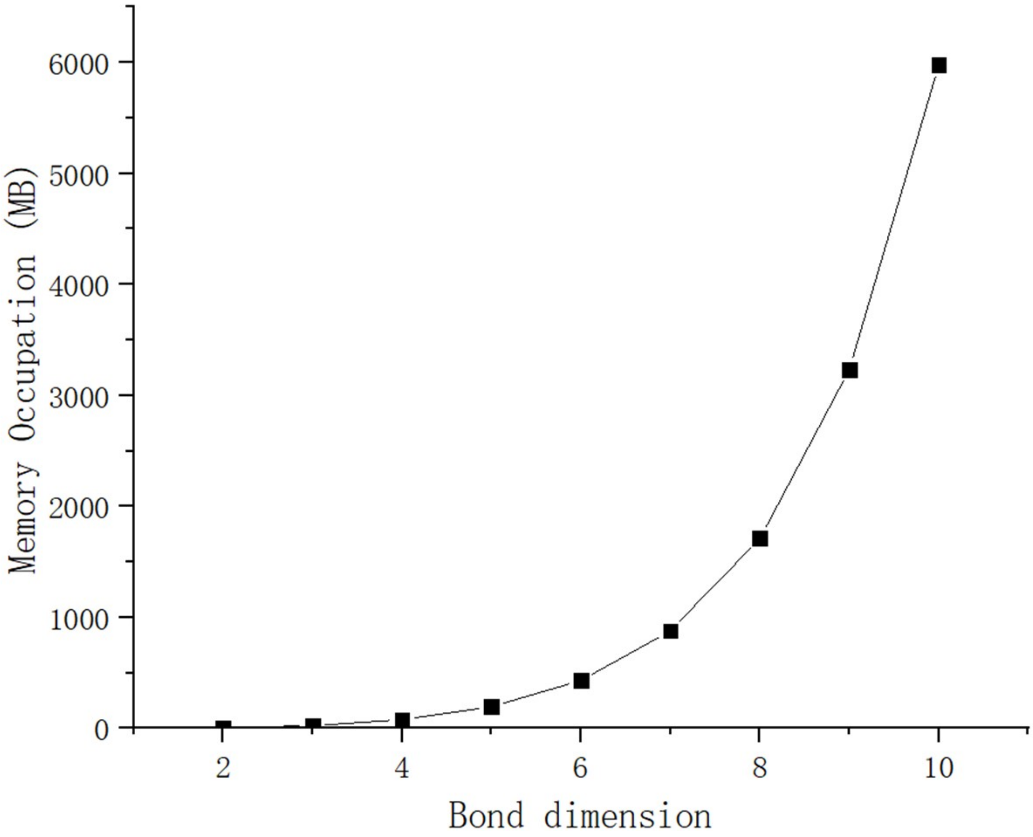

Figure S13-2: The memory requirement of AFA as a function of bond dimension. As depicted, the memory occupation exhibits exponential growth with respect to the bond dimension.

Figure S13-2 illustrates the memory requirement of AFA in relation to bond dimension, a parameter associated with the storage of correlations between radicals. Our current device has a maximum memory capacity of 10GB, which allows for a bond dimension of up to 10. We believe that AFA's performance could be further enhanced with a more advanced device offering greater memory capacity.

Supplementary Material S14: Extension of AFA in intermolecular interaction

While AFA was originally designed to predict the properties of individual molecules, we are able to modify it to predict intermolecular interaction energies by incorporating hydrogen-bond interactions and non-rigid bonding interactions. The calculation of these interactions is obtained from ref[11]. We set the intra-molecular energy calculator to be AFA and then conducted the experiment. Here we define the resulting model as

AFA-hydrogen. The method is similar to that in ref [12].

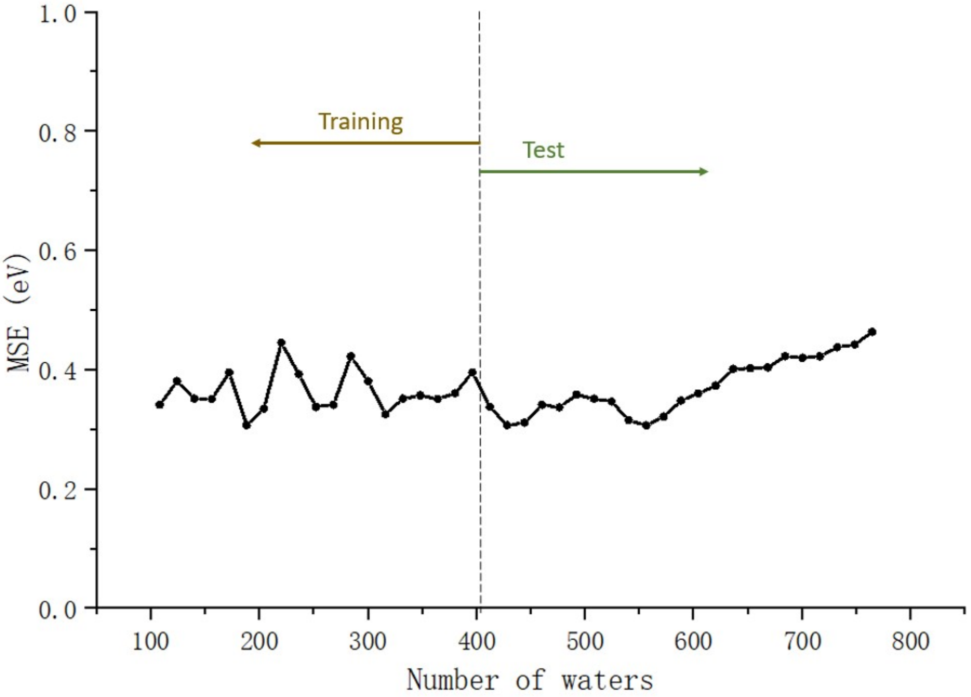

Figure S14-1: the performance of AFA-hydrogen on intermolecular interaction energies. We use the water systems with less than 400 atoms for training, while we test the performance of the water systems with more than 400 atoms. We can see that the error does not increase a lot, AFA-hydrogen still performs well.

We have tested the modified AFA model, referred to as AFA-hydrogen on the water system, which includes a typical intermolecular hydrogen bond interaction. We trained the model using water systems with fewer than 400 and tested it on water systems between 400 and 768. Our results, as shown in Figure S14-1, indicate that AFA still performs well in predicting the intermolecular interaction energies, as the error does not increase significantly as the size of the system increases.

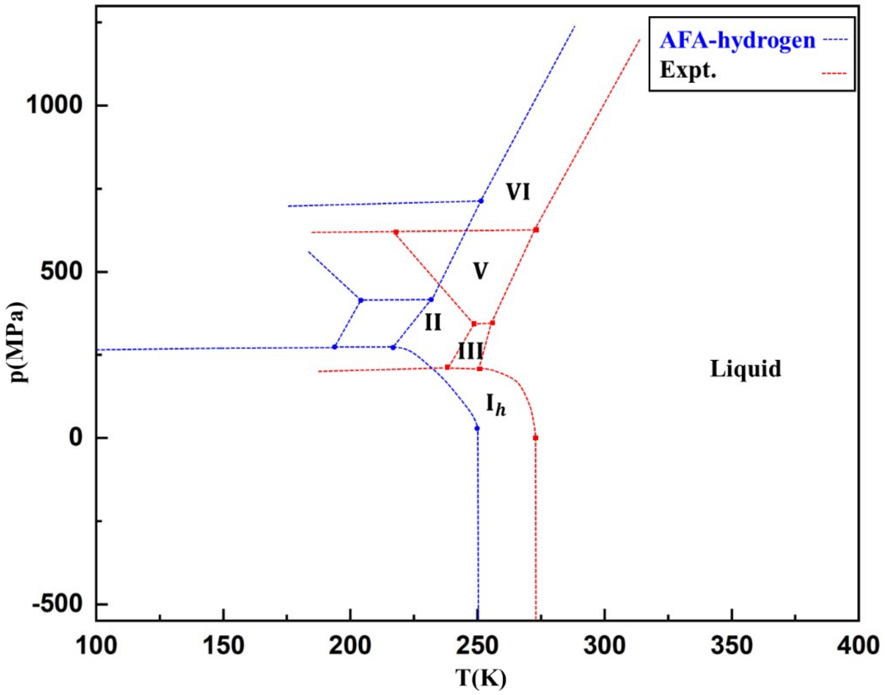

Figure S14-2: the phase diagram of AFA-meta compared to the experiment results.

Figure S14-2 shows the phase diagram comparison between AFA-hydrogen and the experiment results. Here the blue curve refers to the prediction result of AFA-hydrogen, and the red curve represents the experimental result. The Gibbs-Duhem integration method was used to compute the phase diagram of water using the AFA-hydrogen force field. This method requires the prediction of not only energy but also other properties like pressure, which can be estimated by AFA-hydrogen. Our model has these advantages compared with other models:

1. Our model can effectively capture a variety of water phases, such as Ice III, which some water models[13] are unable to capture.
2. Our model predicts ice Ih to be the most stable phase under ambient conditions, aligning well with experimental findings. However, the TIP3P model[14] does not support this result.
3. Our model accurately estimates the melting point at 1 bar to be around 250K, whereas the TIP4P model[15] predicts a value of 232K. The experimental result stands at 273K.

However, capturing long-range interactions presents a challenge for tensor network methods, and without incorporating hydrogen-bond and non-rigid bonding interactions, it would be very difficult to achieve.

References:

(1) Frisch, M.; Trucks, G.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. gaussian 09, Revision d. 01, Gaussian. *Inc., Wallingford CT* **2009**, *201*.

(2) Fang, X.; Liu, L.; Lei, J.; He, D.; Zhang, S.; Zhou, J.; Wang, F.; Wu, H.; Wang, H. Geometry-enhanced molecular representation learning for property prediction. *Nat. Mach. Intell* **2022**, 1-8.

(3) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided Mol. Des.* **2016**, *30* (8), 595-608.

(4) Brockherde, F.; Vogt, L.; Li, L.; Tuckerman, M. E.; Burke, K.; Müller, K.-R. Bypassing the Kohn-Sham equations with machine learning. *Nat. Comm.* **2017**, *8* (1), 1-10.

(5) Nesbet, R. Brueckner's Theory and the Method of Superposition of Configurations. *Phys. Rev.* **1958**, *109* (5), 1632.

(6) Yamagami, H.; Takada, Y.; Yasuhara, H.; Hasegawa, A. Improvement on the correlated-Hartree-Fock method and application to atoms. *Phys. Rev. A* **1994**, *49* (4), 2354.

(7) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1* (1), 1-7.

(8) Liu, T.; Lin, Y.; Wen, X.; Jorissen, R. N.; Gilson, M. K. BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Res.* **2007**, *35* (suppl_1), D198-D201.

(9) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40* (D1), D1100-D1107.

(10) St John, P. C.; Guan, Y.; Kim, Y.; Etz, B. D.; Kim, S.; Paton, R. S. Quantum chemical calculations for over 200,000 organic radical species and 40,000 associated closed-shell molecules. *Sci. Data* **2020**, *7* (1), 1-6.

(11) Mayo, S. L.; Olafson, B. D.; Goddard, W. A. DREIDING: a generic force field for molecular simulations. *J..Phys. Chem.* **1990**, *94* (26), 8897-8909.

(12) Gao, Y.; Lu, Y.; Zhu, X. Mateverse, the Future Materials Science Computation Platform Based on Metaverse. *J. Phys. Chem. Lett.* **2022**, *14*, 148-157.

(13) Zhang, L.; Wang, H.; Car, R.; Weinan, E. Phase diagram of a deep potential water model. *Phys. Rev. Lett.* **2021**, *126* (23), 236001.

(14) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* **1983**, *79* (2), 926-935.

(15) Jorgensen, W. L.; Madura, J. D. Temperature and size dependence for Monte Carlo simulations of TIP4P water. *Mol. Phys.* **1985**, *56* (6), 1381-1392.