High-throughput Computational Workflow for Ligand Discovery in Catalysis with the CSD

Marc A. S. Short, Clare A. Tovee, Charlotte E. Willans, Bao N. Nguyen*

Electronic Supporting Information

Contents

1 Benchmarking of computational methods for copper(I) complexes 2

2 Method development with literature ligands for Ullmann-Goldberg reactions 9
2.1 DFT calculation of the Ullmann-Goldberg catalytic cycle . . . 9
2.2 Benchmarking $\Delta G^{\ddagger}$ from B97-3c//GFN2-xTB . . . 9
2.3 Accounting for the asymmetric geometry of **TSOA** . . . 10
2.4 Protonation and deprotonation rules for ligands coordinating to Cu(I) . . . 11
2.5 Literature Ligand Analysis . . . 12
2.6 Tuning the Value of S_0 . . . 13

3 Ligand searching with CSD-CrossMiner 14
3.1 Building CatSD, the structural database for catalysis . . . 14
3.2 CatSD Features . . . 14
3.3 Generation of Catalophores . . . 17
3.4 Search Settings . . . 19
3.4.1 Annotation Filters . . . 19
3.4.2 Structure Filters . . . 20

4 High-throughput B97-3c//GFN2-xTB calculations of $\Delta G^{\ddagger}$ with different datasets 22
4.1 Automated building of structures of Cu(I) complexes from ligands with molSimplify . . . 22
4.1.1 Generation of stable/intermediate structures . . . 22
4.1.2 Generation of transition state structures . . . 25
4.1.3 Correction of Coordinating Atom Indexes . . . 27
4.2 Automated generation of GFN2-xTB and B97-3c input files . . . 28
4.3 Automated extraction of computational results . . . 28
4.3.1 Automated screening of failed B97-3c//GFN2-xTB calculations . . . 28
4.3.2 Automated extraction of $\Delta G^{\ddagger}$ from computational output files . . . 28
4.3.3 Automated extraction of descriptors from computational output files . . . 29

5 Machine Learning 29
5.1 Descriptors and descriptors selection . . . 29
5.1.1 List of initial descriptors and their sources . . . 29
5.2 Scree and Q^2 plots . . . 32
5.3 Correlation between $\Delta G^{\ddagger}$ and descriptors . . . 33
5.4 Initial ML models building with transition state descriptors . . . 36
5.4.1 Initial model metrics . . . 37
5.4.2 Descriptors trimming based on permutation and feature importance . . . 43
5.4.3 Models with Trimmed Descriptors and Optimised Hyperparameters . . . 48
5.4.4 Permutation Importance of Trimmed Models . . . 53
5.5 ML models building without transition state descriptors . . . 58
5.5.1 Metrics of initial ML models without transition state descriptors . . . 58
5.5.2 Transition State Independent Descriptors with Trimmed Descriptors . . . 66
5.5.3 Improving models through improved calculations of electronic descriptors . . . 73

1 Benchmarking of computational methods for copper(I) complexes

Semi-empirical and DFT methods were performed using Gaussian09.[1] Composite methods were performed using ORCA 4.2.1.[2] Extended Tight Binding calculations were performed with xtb 6.3.3.[3] The University of Leeds supercomputers, ARC3 and ARC4, were used on standard nodes with 24-core Broadwell E5-2650v4 CPUs at 2.2GHz with turbo and 128GB of memory and 40-core Intel Xeon Gold 6138 CPUs at 2.0GHz with 196GB of memory respectively.[4] Complexes were optimised in the gas phase using the default convergence criteria unless stated otherwise. XTB calculations were optimised to the *vtight* criteria unless stated otherwise.

Structures for benchmarking were chosen based on three criteria which describe the active catalytic state in the Ullmann-Goldberg reaction:

1. a mononuclear three-coordinate copper(I) centre
2. a deprotonated nitrogen nucleophile
3. either one bidentate or two monodentate ligand(s)

Only 10 structures were available in the Cambridge Structural Database (CSD) at the time of searching (Figure S1).

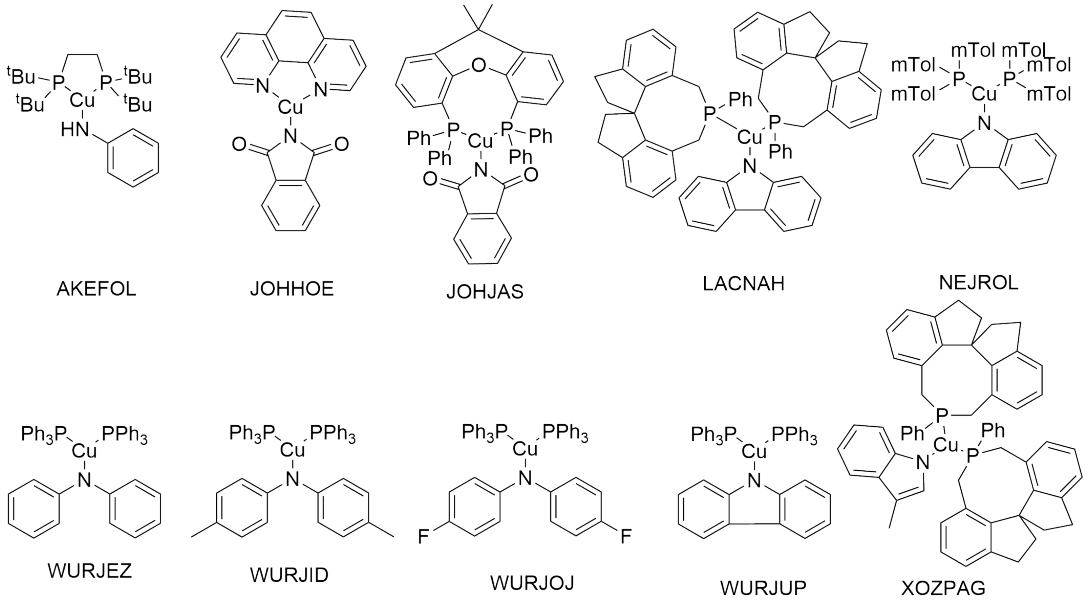


Fig. S1 Chemical structures and CSD refcodes of the benchmarking complexes.

Nine of the ten structures are phosphorus-based ligands, with three out of 10 being bidentate. Due to the lack of 3D crystal structures available for catalytic species, these were the closest to the active catalytic state available and therefore chosen for benchmarking.

Table S1 Structures used for benchmarking, their ligand types and X-ray structure accuracies.

CSD Ref	Ligand Type	Denticity	R factor (%)	Average σ(C-C) (Å)
AKEFOL	PP	Bidentate	6.70	0.011-0.030
JOHHOE	NN	Bidentate	5.09	0.006-0.010
JOHJAS	PP	Bidentate	4.24	0.001-0.005
LACNAH	P, P	Monodentate	6.59	0.006-0.010
NEJROL	P, P	Monodentate	4.78	0.001-0.005
WURJEZ	P, P	Monodentate	4.03	0.001-0.005
WURJID	P, P	Monodentate	3.11	0.001-0.005
WURJOJ	P, P	Monodentate	4.25	0.001-0.005
WURJUP	P, P	Monodentate	3.11	0.001-0.005
XOZPAG	P, P	Monodentate	4.61	0.001-0.005

Mean Absolute Error between optimized structures and x-ray structures from the CSD are calculated with the following formula:

$$MAE = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j>1}^{N} \left| R_{ij}(Calc) - R_{ij}(X\text{-}ray) \right| \tag{1}$$

Bond lengths and bond angles measured are between the copper atom and the coordinating atoms (L_1, L_2, N).

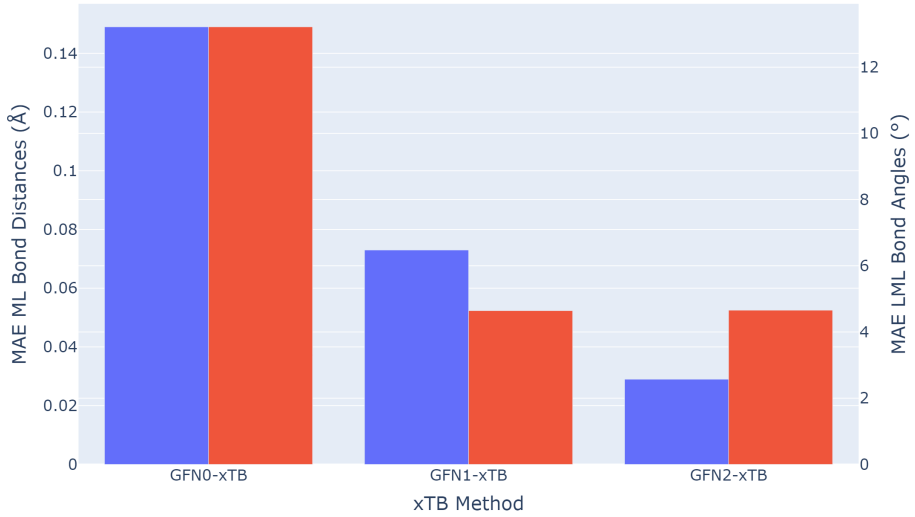


Fig. S2 Mean Average Error of bond lengths (blue) and bond angles (red) for xTB calculations on the benchmark structure set.

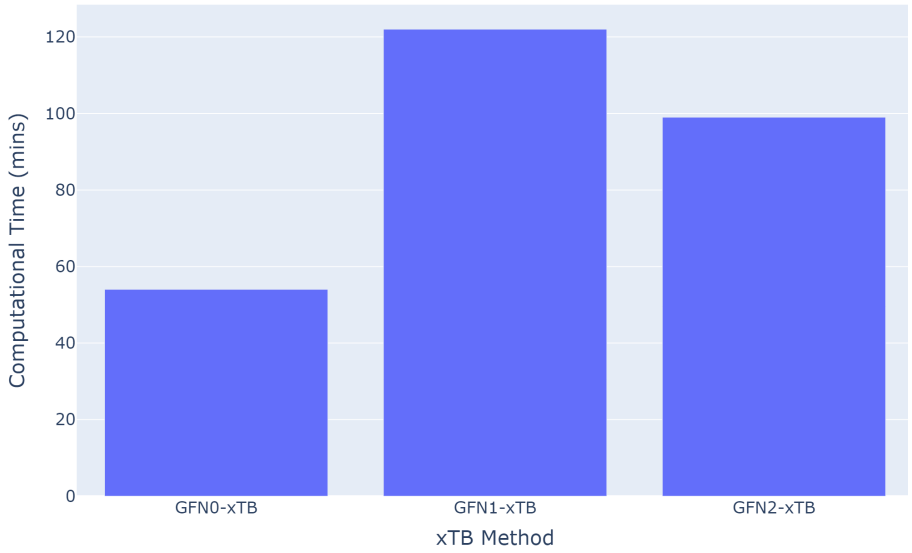


Fig. S3 Single core computational Time for xTB calculations on the benchmark structure set.

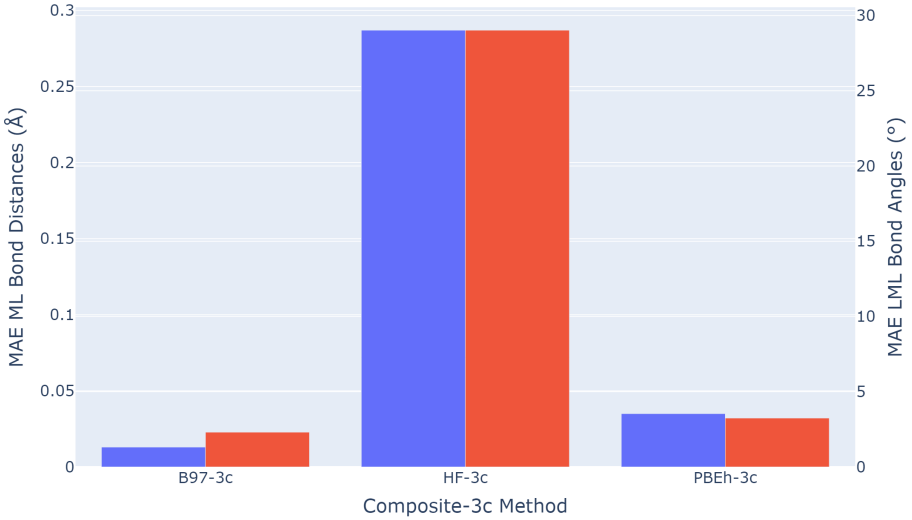


Fig. S4 Mean Average Error of bond lengths (blue) and bond angles (red) for 3c calculations on the benchmark structure set.

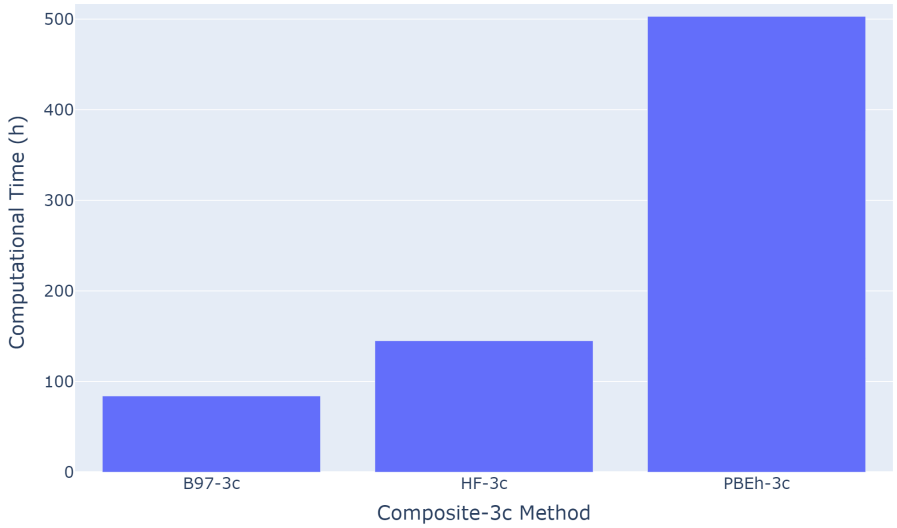


Fig. S5 Single core computational Time for 3c calculations on the benchmark structure set.

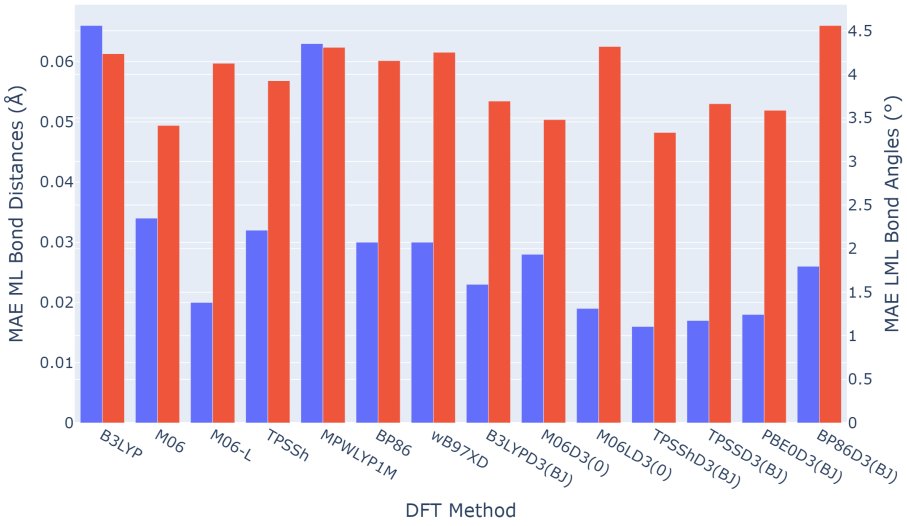


Fig. S6 Mean Average Error of bond lengths (blue) and bond angles (red) for DFT methods on the benchmark structure set.

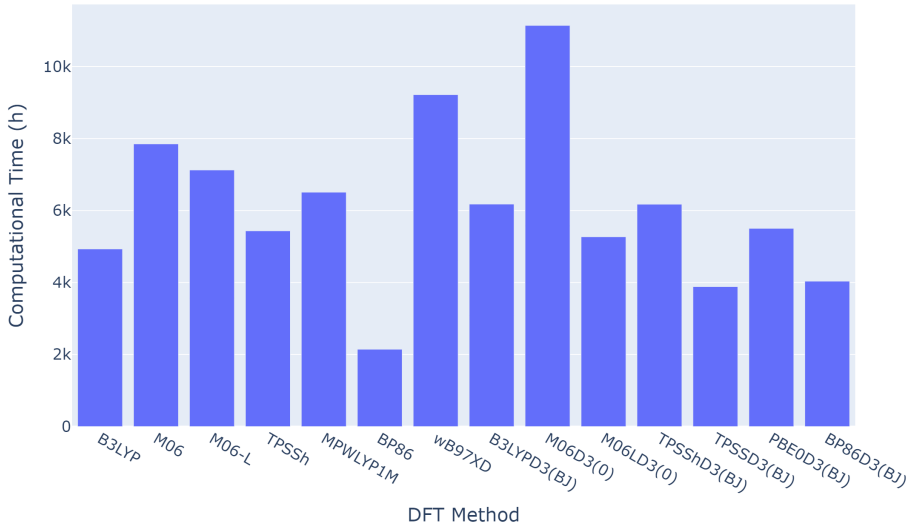


Fig. S7 Single core computational time for DFT methods on the benchmark structure set.

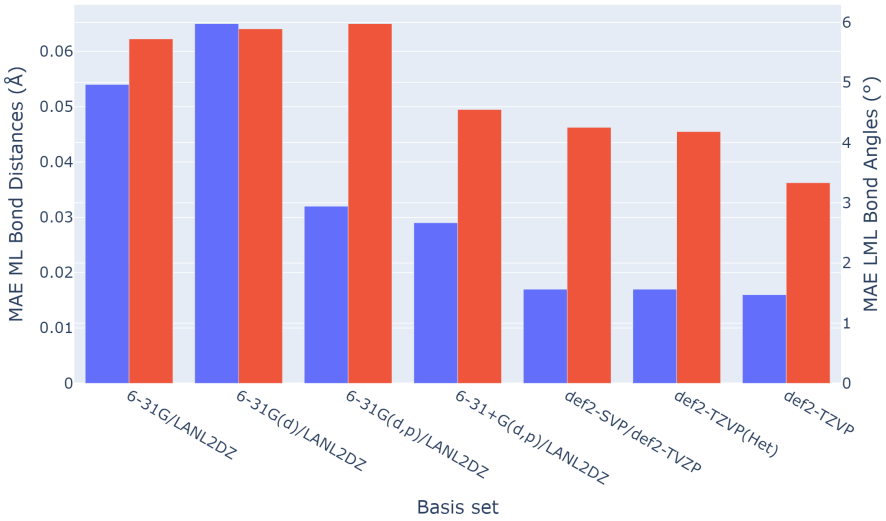


Fig. S8 Mean Average Errors of bond lengths (blue) and bond angles (red) for different basis sets using the TPSSh functional for the benchmark structure set.

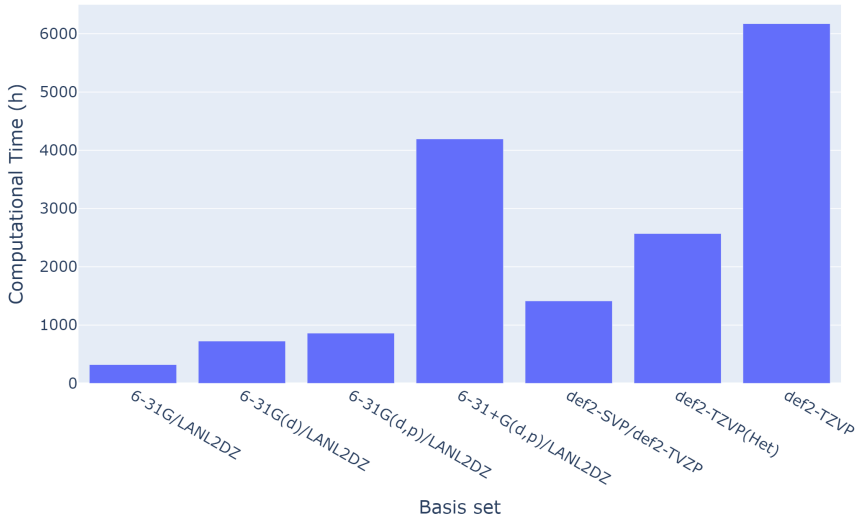


Fig. S9 Single core computational time for different basis sets using the TPSSh functional for the benchmark structure set.

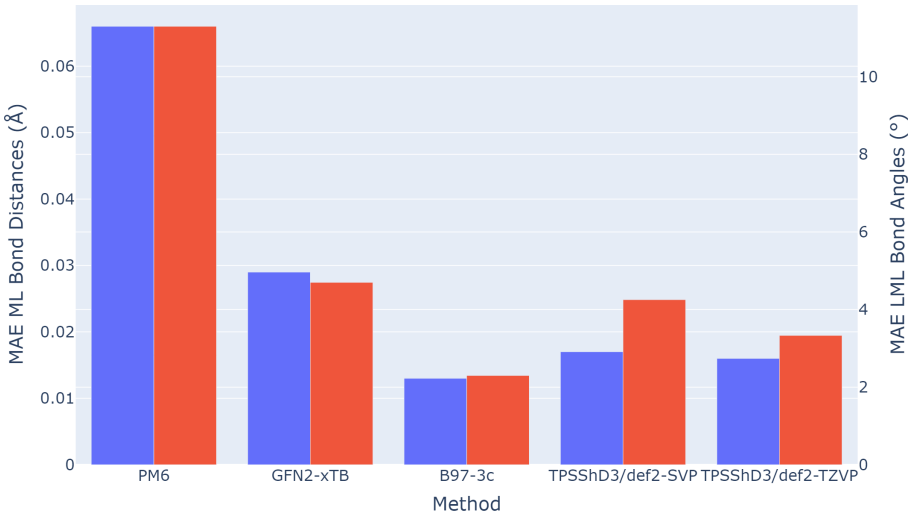


Fig. S10 Comparison of mean average errors of bond lengths (blue) and bond angles (red) between the best methods from each class on the benchmark structure set.

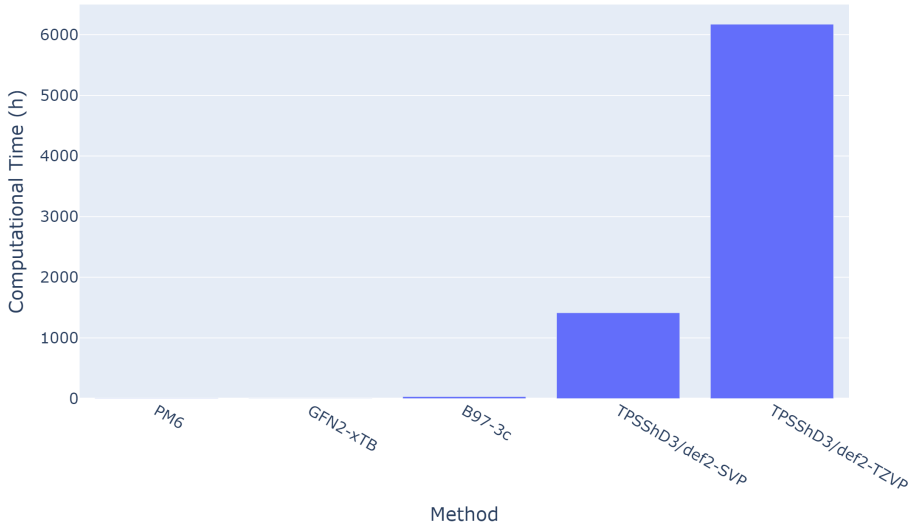


Fig. S11 Comparison of single core computational time for the best methods from each class on the benchmark structure set.

2 Method development with literature ligands for Ullmann-Goldberg reactions

B97-3c and DLPNO-CCSD(T)/def2-TZVPP calculations were performed using ORCA 4.2.1.[2] GFN2-xTB calculations were performed with standalone xtb 6.3.3 with the *vtight* optimisation criteria for non-TS optimisation steps and xtb 6.3.3 interfaced with ORCA 4.2.1 for TS optimisations and frequency calculations with default convergence criteria.[3] For TS optimisations the exact hessian was recalculated every 5 optimisation steps. DMF was used as the solvent and caesium carbonate as the base. Structures were generated with a custom version of molSimplify.[5] Ligands were supplied as SMILES strings.

2.1 DFT calculation of the Ullmann-Goldberg catalytic cycle

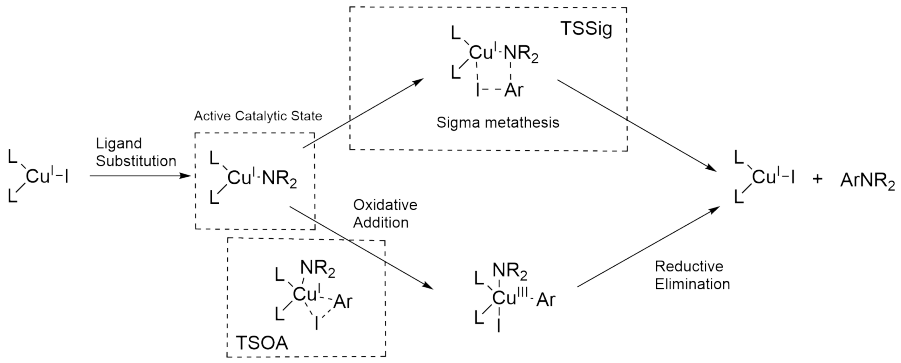
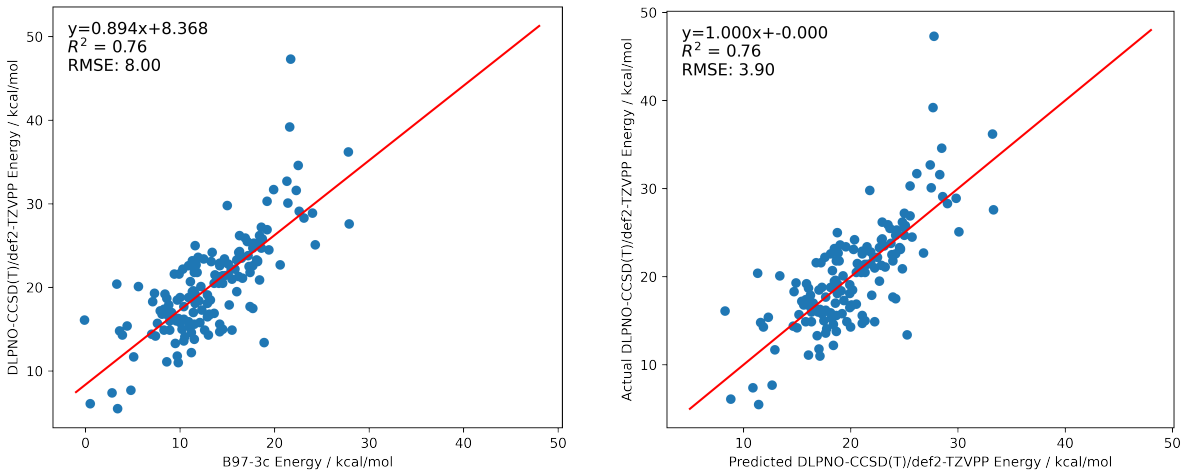


Fig. S12 Oxidative Addition and Sigma metathesis pathways for the Ullmann-Goldberg reaction.

Two closed shell mechanistic pathways were investigated, oxidative-addition/reductive-elimination (**TSOA**) and sigma metathesis (**TSSig**). This requires the generation of four structures, the two transition states (oxidative addition is the rate-determining step for OA/RE), the active catalytic state and the precatalyst after ligand exchange/product complex. Structures for these complexes were generated and the Gibbs free energies were calculated.

2.2 Benchmarking $\Delta G^{\ddagger}$ from B97-3c//GFN2-xTB



(a) Unscaled **(b)** Scaled

Fig. S13 Comparison of activation energies for **TSOA** and **TSSig** between B97-3c//GFN2-xTB and DLNPO-CCSD(T)/def2-TZVPP//GFN2-xTB calculated energies for all literature ligands.

To assess the error within the activation energy calculation B97-3c energies were benchmarked against DLPNO-CCSD(T)/def2-TZVPP. 100 randomly selected ligands were taken from the ***ligands_lit_set***, optimised at the GFN2-xTB level of theory and the activation energies were calculated at the B97-3c and DLPNO-CCSD(T)/def2-TZVPP levels of theory. The comparison of energies for each method are shown in Fig. S13. B97-3c energies were scaled to a DLPNO-CCSD(T)/def2-TZVPP energy using the equation of the line using raw B97-3c energy, resulting in an RMSE of 3.9 $\mathrm{kcal\,mol^{-1}}$.

2.3 Accounting for the asymmetric geometry of TSOA

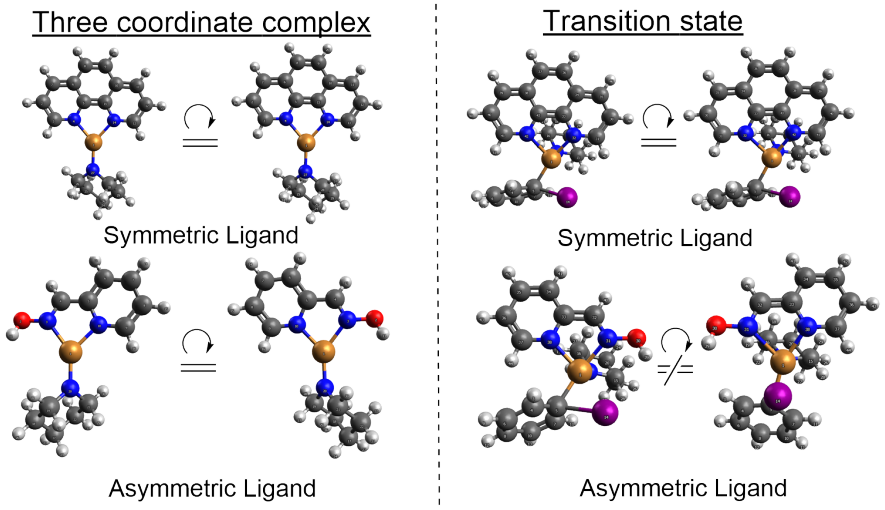


Fig. S14 Overview of asymmetric ligand flipping.

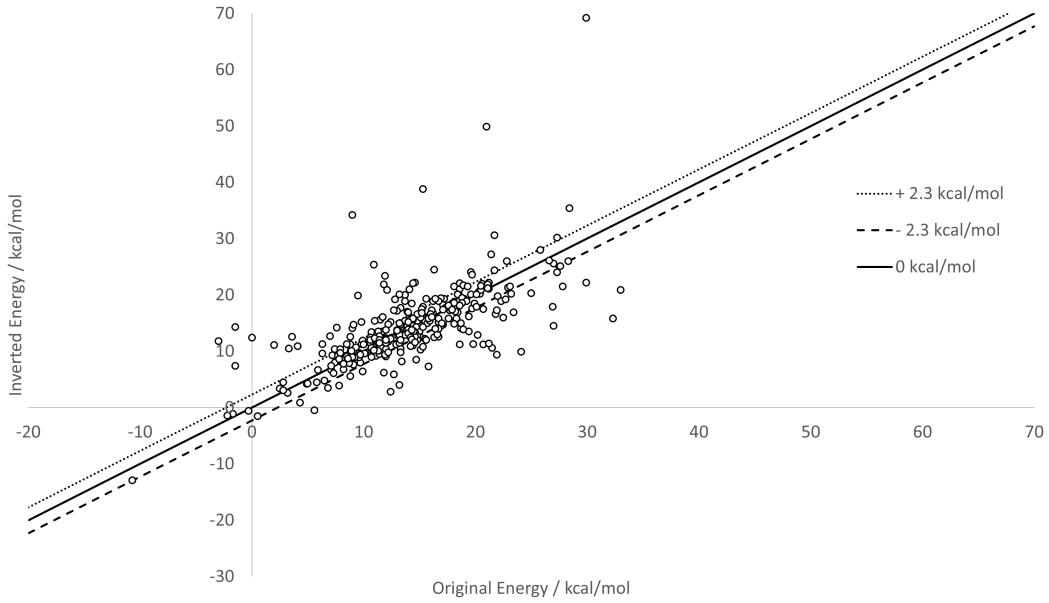


Fig. S15 Comparison of activation energies between original and flipped ligands for all asymmetric ligands in the literature dataset.

Transition state structures for asymmetric ligands, in the case of the Ullmann-Goldberg reaction, have two possible isomers which need to be compared to predict the total activity of the ligand (Fig. S14). In order to assess the need to generate both structures for each transition state in the computational workflow, all asymmetric ligands in the ***ligands_lit_set*** were used to compare the activation energies for both isomers. The structures were optimised to a transition state using GFN2-xTB and their activation

energies were calculated at the B97-3c level of theory. The comparison of activation energies of the original and inverted ligand (Fig. S15). The correlation of original activation energy and activation energy upon ligand inversion has an average difference of 2.3 $\mathrm{kcal\,mol^{-1}}$ (2.8 and 1.9 $\mathrm{kcal\,mol^{-1}}$ for **TSOA** and **TSSig** respectively) between isomers and is within the error of the calculation (3.9 $\mathrm{kcal\,mol^{-1}}$). As the Ullmann-Goldberg reaction is a symmetric reaction, only one isomer was deemed necessary. The need to generate only one isomer significantly reduces the computational time required. For asymmetric synthesis tasks, however, it is recommended to generate both isomers.

2.4 Protonation and deprotonation rules for ligands coordinating to Cu(I)

The protonation states of common ligand coordinating functional groups were analysed by comparing the number of protonated and deprotonated structures available for their copper complexes in the Cambridge Structural Database containing only one copper atom (Table S2). The SMARTS pattern for each functional group which deprotonated upon coordination to copper were generated and added to the self.remHsmarts list in ~/molSimplify/Classes/globalvars.py.

Table S2 Protonation states of common ligand functional group from the Cambridge Structural Database and their associated pKa range in DMSO.[6]

Functional Group	**Number of entries**	**Unchanged**	**Deprotonated**	**pKa**[6]
Amine	3793	3747	46	~ 40
Aniline	199	165	34	25-31
Hydrazine	102	101	1	25-29
Imine	218	215	3	~ 31
Amide	906	14	882	17-25
Carboxylic acid	4622	18	4604	9-13
Alcohol	1169	991	178	~ 30
Thiol	34	1	33	5-12
Phenol	2939	99	2840	10-19
Thiophenol	68	0	68	5-12
Phosphonic acid	209	4	205	~ 2

Table S3 SMILES strings and associated functional group for the protonation rules used for structure generation.

Functional Group	SMILES String
Amide	O=CN
Carboxylic Acid	O=CO
Aromatic nitrogen	n
Amidine	N=CN
N-amino	nN
Phenol	Oc
Thiophenol	Sc
Thiol	SC
N-oxide	ON
Aromatic N-oxide	On
Sulphonic acid	OS
N-Carbene	NCN
Phosphonic	OP

2.5 Literature Ligand Analysis

Table S4 Frequency of functional groups which coordinate to the copper centre in the ***ligands_lit_set*** for bidentate ligands only. 1 group represents only one of the two coordinating functional groups, 2 groups are both functional groups.

Functional Group	Frequency	1 Group	2 Groups
alcohol	22	6	8
aldehyde	1	1	0
amide (N)	51	33	9
amide (O)	38	38	0
amine	138	48	45
carbene	7	1	3
carboxylic acid	41	39	1
ester	4	4	0
ether	1	1	0
hydrazine	13	11	1
imidazole	5	1	2
imine	41	13	14
indole	4	2	1
ketone	24	8	8
nitrile	2	0	1
N-oxide	15	13	1
oxime	10	6	2
phenol	30	24	3
phosphate	2	2	0
phosphine	12	4	4
phosphine oxide	2	2	0
pyrazole	3	1	1
pyridine	56	30	13
pyrimidine	1	1	0
pyrrole	26	24	1
selenophene	1	1	0
tetrazole	1	1	0
thiol	2	2	0
thiophene	2	2	0
thiophenol	1	1	0
triazole	1	1	0

Table S5 Number of bridging atoms between the ligand coordinating atoms for the ligands in the ***ligands_lit_set***.

Number of Bridging Atoms	Frequency
1	2
2	187
3	73
4	9
5+	7

2.6 Tuning the Value of S_0

Table S6 Successful identification of a correct transition state with varying values of S_0 for 198 ligands in the PIP_set_TSOA dataset. The 'Not used' entry only uses the magnitude of the imaginary frequency to determine if the transition state is correct.

S_0	Correctly Identified Transition States	Incorrectly Identified Transition States	Success Rate (%)
Not used	175	23	88.4
0.20	197	1	99.4
0.25	180	18	90.9
0.30	142	56	71.7
0.33	114	84	57.6

3 Ligand searching with CSD-CrossMiner

3.1 Building CatSD, the structural database for catalysis

The structural database is built upon CSD_541 with the Mar20, May20, Aug20 and Feb21 updates. To enable easier searching of the database a set of annotations is used to define specific properties of each entry in the database. The annotations refine the database to create a subset of structures that (a) are not polymeric, (b) have no disorder, (c) for which 3D coordinates have been determined and (d) have a maximum *R*-factor of 10%. This resulted in a database containing approximately 658,000 structures.

Each entry has an annotation taken directly from the CSD, using the CSD Python API, containing the following values: (1) CSD identifier, (2) CSD Refcode, (3) database name, (4) chemical formula, (5) *R*-factor, (6) is_organic and (7) is_organometallic.

The following elements were excluded from the searches: Br, Cl, I, Li, Na, K, Ca, Mg, Be. The maximum molecular weight for structures was set to 500Da. The maximum rmsd between catalophore and hit was set to 1.0. A maximum of 1 hit per structure was allowed to prevent duplicates. Any subsequent duplicates were removed via. SMILES matching. Both three cubed packing and complete small molecules were enabled. Only organic structures were returned.

3.2 CatSD Features

A series of features were generated to describe common coordinating functional groups in organometallic ligands. All queries were combined into a single general feature catsd_coordinating_atom_general, the SMARTS queries are summarised in Table S7. Features were also separated into individual functional groups to enable more refined searching. The features were indexed into **CatSD** to enable searching the CSD with these features. These features can be added to a catalophore to define the location and properties of a coordinating atom within a ligand. Standard CSD-CrossMiner features can be used to define additional properties such as aromatic rings, heavy atoms and hydrogen bonding.

Table S7 Features used in the CatSD *coordinating_atom_general* feature.

Number	SMARTS Definition	Base Index	Geometry	Functional Group
1	[#6](-[#6])(-[#6]=[#6](-[#6])-[#8-])=[#8]	5 6	trigonal	2,5-diketone
2	[#8-]-[#6]	0	tetrahedral	alcohol
3	[#8H1]-[#6X4]	0	tetrahedral	alcohol
4	[#1][#8H1]-[#6X4]	0	linear_nb	alcohol
5	[#8X1]=[#6](-[#6;#7])(-[#7])	0	trigonal	amide
6	[#7X2]-[#6](=[#8])	0	trigonal	cyclic amide
7	[#1][#7X3H2]-[#6](=[#8])	0	tetrahedral	primary amide
8	[NX3H2](-[C]([!#8;!#7])([!#8;!#7]))	0	tetrahedral	primary amide
9	[#1][#7X3H1]-[#6](=[#8])	0	linear_nb	secondary amide
10	[NX3H2](-[C]([!#8;!#7])([!#8;!#7]))	0	tetrahedral	secondary amide
11	[NX3H0](-[C]([!#8;!#7])([!#8;!#7]))(-[C]([!#8;!#7])([!#8;!#7]))-[C]([!#8;!#7])([!#8;!#7])	0	tetrahedral	tertiary amide
12	[#1][NX3H2](-[c]([!#8;!#7])([!#8;!#7]))	0	linear_nb	analine
13	[#1][NX3H2](-[c]([!#8])([!#8]))	0	linear_nb	analine
14	n-[#1]	0	linear_nb	aromatic nitrogen
15	n	0	trigonal	aromatic nitrogen
16	[#7]=[#7][#6]	1	trigonal	azo
17	[#7]=[#7][#6]	0	tetrahedral	azo
18	[#8H1][#6]=[#8]	0	trigonal	carboxylic acid
19	[#1][#8H1][#6]=[#8]	0	linear_nb	carboxylic acid
20	[#1][#8H1][#6]=[#8]	3	trigonal	carboxylic acid
21	[#8X1][#6]=[#8]	2	trigonal	carboxylic acid
22	[#8X1][#6]=[#8]	0	tetrahedral	carboxylic acid
23	[#6](-[#6])(-[#8]-[#6])=[#8]	4	trigonal	ester
24	[#7]=[#6]-[#7][#1]	3	linear_nb	imidazole
25	[#6;#1;#8;#7;#16][#7X2]=[#6]	1	trigonal	imine
26	[#6]=[#7X3](-[#6;#1;#8;#7;#16])-[#1]	3	linear_nb	imine
27	[#6]=[#7]-[#6]	1	trigonal	imine

Continuation of Table S7

Number	SMARTS Definition	Base Index	Geometry	Functional Group
28	[#6]#[#7]-[#6]	0	linear	isocyano
29	[#8X1]=[#6]([#6])([#6])	0	trigonal	ketone
30	[#8-]-[#7]-[!#8]	0	trigonal	n-oxide
31	[#1]-[#8]-[#7]=[#6]	0	linear_nb	oxime
32	[#8-]-[#7]=[#6]	0	tetrahedral	oxime
33	[#1]-[#8]-[#7]=[#6]	1	tetrahedral	oxime
34	HOc	1	tetrahedral	phenol
35	HOc	0	linear_nb	phenol
36	[PX3](-[#8H0][#6])(-[#8H0][#6])-[#8H0][#6]	0	tetrahedral	phosphate ester
37	[PX3](-[#6;#7])(-[#6;#7])-[#6;#7]	0	tetrahedral	phosphine
38	[#7X2H1]=[#6]	0	tetrahedral	primary imine
39	[#7][#7][#1]	2	linear_nb	pyrazole
40	c-[#7](-[#1])-(c)	2	linear_nb	pyrrole
41	c[nX2]c	1	trigonal	aromatic nitrogen ring
42	[#16-]-[#6]	0	tetrahedral	thiol
43	[#16H1]-[#6X4]	0	tetrahedral	thiol
44	[#1][#16H1]-[#6X4]	0	linear_nb	thiol
45	HSc	1	tetrahedral	thiophenol
46	HSc	0	linear_nb	thiophenol
47	[NX3H2]([#6]([!#8;!#7])([!#8;!#7]))	0	tetrahedral	primary amine
48	[NX3H1]([#6]([!#8;!#7])([!#8;!#7]))([#6]([!#8;!#7])([!#8;!#7]))	0	tetrahedral	secondary amine
49	[NX3H0](-[#6]([!#8;!#7])([!#8;!#7]))(-[#6]([!#8;!#7])([!#8;!#7]))-[#6]([!#8;!#7])([!#8;!#7])	0	tetrahedral	tertiary amine

3.3 Generation of Catalophores

Catalophores were generated using the CSD-CrossMiner software through the GUI. Reference structures were generated for each transition state (**TSOA** and **TSSig**) at the GFN2-xTB level of theory using 3,4,7,8-tetramethyl-1,10-phenanthroline (TMPHEN) as the ligand, iodobenzene as the substrate and the nucleophile of interest. Transition states were confirmed by visualisation of the imaginary frequency. The **TSOA** reference structure was used to generate the catalophore as it is more sterically demanding around the copper centre. For each nucleophile, the **TSOA** reference structure was imported into CSD-CrossMiner and the **CatSD** feature database was loaded. catsd_coordinating_atom_general features were placed on each TMPHEN nitrogen atom with a tolerance of 0.75Å and the projected sphere was placed on top of the copper atom. heavy_atom features were placed on the two bridging carbon atoms with a tolerance of 0.75Å and the features were constrained to be intramolecular (Fig. S16).

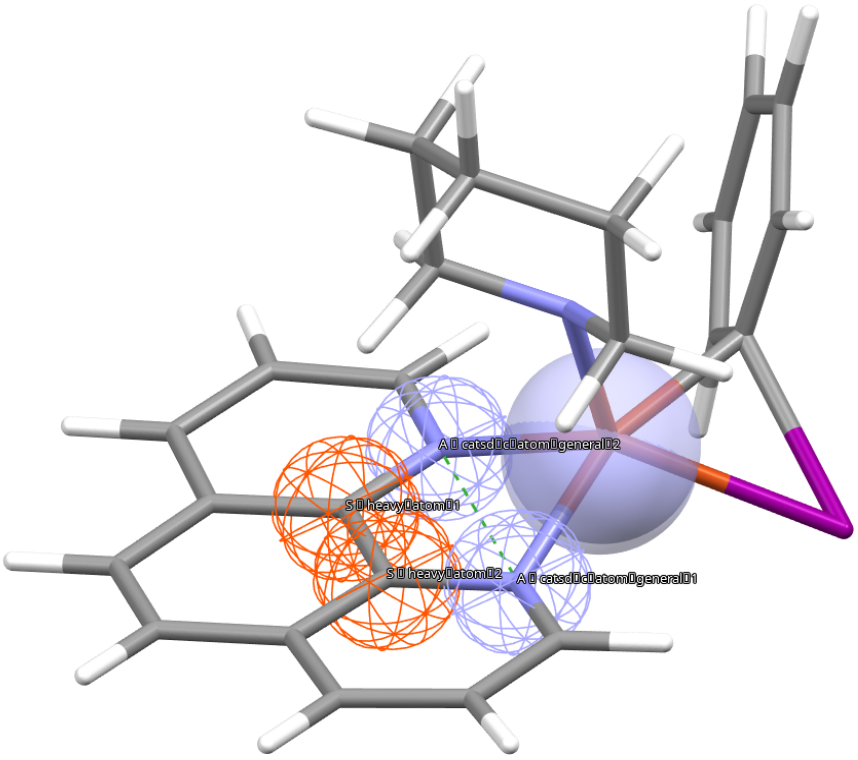


Fig. S16 Example definition of ligand coordination in CSD-CrossMiner.

To define the substrate sites excluded volumes were placed on every atom of the substrates and the copper atom. The tolerance of the excluded volumes were set to the van der Waals radii of the base atom. (Fig. S17, S18). A smaller tolerance of 1.5Å was used for copper to allow for coordinating atoms to occupy the space around the copper, while also preventing ligand atoms from occupying the space of the metal.

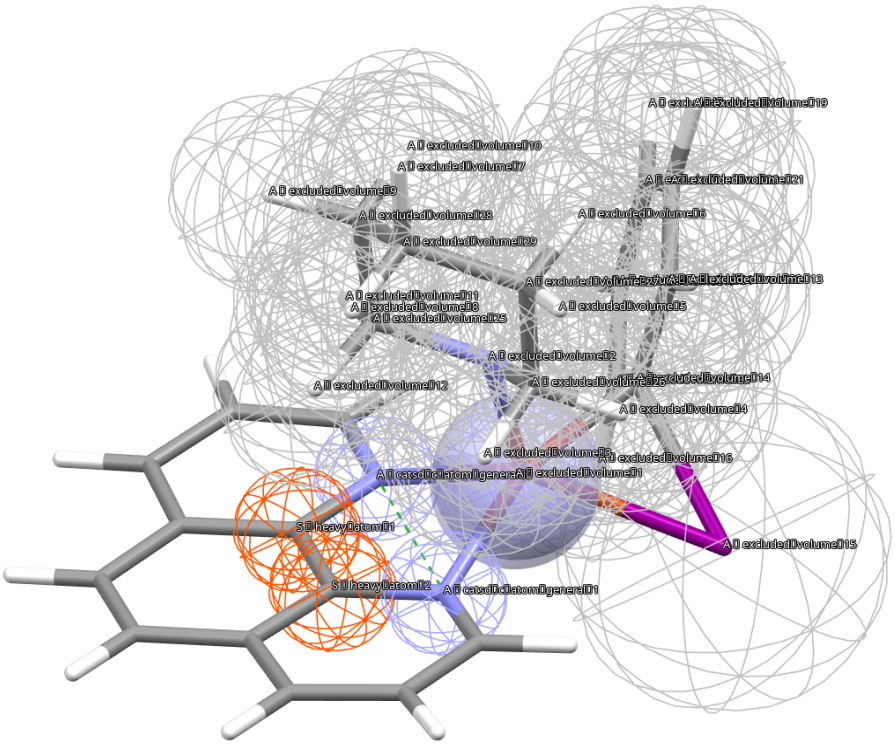


Fig. S17 Example definition of a full catalophore in CSD-CrossMiner including substrate definition.

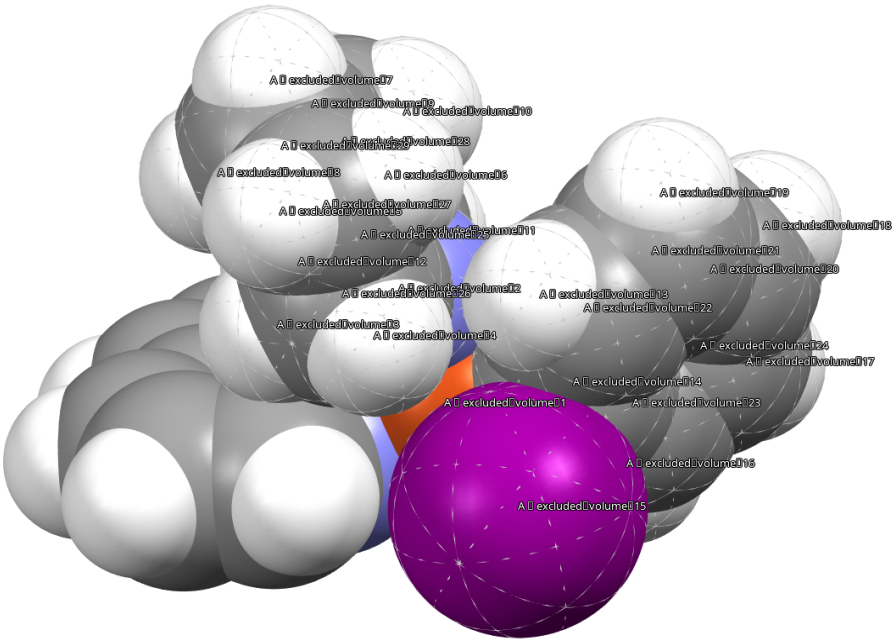


Fig. S18 Example definition of a full catalophore in CSD-CrossMiner including substrate definition with spacefill.

The catalophore was saved as a *.cm* file. Searching **CatSD** is only possible via the CSD-PythonAPI, the search script is provided in the provided Zenodo repository. Note: A CSD-Discovery license is required to use both CSD-CrossMiner and the search script through the CSD-PythonAPI.

3.4 Search Settings

The script supports the following arguments to adjust the search procedure:

`-n, --name`

The name of the search, which will be applied as a prefix to all output files. Value=str.

`-d, --database`

The feature database to search. The **CatSD** feature database should be used. The value should be a file path to the location of the feature database, e.g. './CatSD.feat' if the feature database is in the same folder as the search script. If no feature database is supplied the script will default to using the standard CSD-CrossMiner feature database.

`-c, --catalophore`

The catalophore file, e.g. 'example_catalophore.cm'. Values include the text string of the name of the catalophore file or the file path of the catalophore file.

`-m, --max-hit-structures`

The maximum number of results to return from a search. Default=50000. Value=int.

`-r, --rmsd`

The maximum value of the rmsd between the catalophore and the hit structures. Default=1. Value=float.

`-w, --max-molecular-weight`

The maximum molecular weight of the hit structures. Default=500. Value=int.

`-t, --threads`

The number of CPU threads to use for the search. Default=4. Value=int. An example input is shown below:

```
python cm_search.py -n "example_search" -c "example_catalophore.cm"
-d "./CatSD.feat" -t 8 -m 10000
```

This will search the **CatSD** feature database using the 'example_catalophore' catalophore with a maximum number of 10,000 hit structures using 8 threads and providing output files with the prefix 'example_search'. Further search refinement can be used to alter the search procedure by modification of the python script. The following settings are used by default:

```
searcher.settings.max_hits_per_structure = 1
searcher.settings.three_cubed_packing = True
searcher.settings.complete_small_molecules = True
```

By default, the script only returns a maximum of one hit per structure. For some use cases, such as comparing coordinating sites within the same ligand, this may not be desirable and should be either increased or removed. Three cubed packing (3x3x3 packing) is enabled, which restricts the search to 26 unit cells around the central unit cell. This allows for symmetry-related copies of the feature points to be considered for a small molecule crystal structure that matches. Complete small molecules is also enabled, this ensures that the entire molecule is returned and not just the section that is within the catalophore bounding sphere. Additional search settings can be found in the CSD-PythonAPI documentation.[7]

3.4.1 Annotation Filters

Hit structures can be filtered based on the annotations present in the feature database in two ways. First, the annotation can be defined in the catalophore file. Second, the annotation filter can be applied within the search script. Annotation filtering is a textual filtering rule that must be defined within the search query, using values that are present within the feature database. An annotation filter consists of a *'key'* and a *'value'*, where the *key* corresponds to the annotation name and the *value* corresponds to the value

for each structure for that annotation. Below is an example of how to apply an annotation filter within the script.

```
model.add_feature(Pharmacophore.AnnotationFilter('is_organic', 'True'))
```

This will return only organic structures from within the database. Custom annotations can be used with custom databases to enable personalised filtering of hits.

3.4.2 Structure Filters

Hits can also be filtered based on chemical structure. Structural filtering is useful in cases where the presence of a certain element in a ligand could potentially lead to prominent side reactions.

Elements which the user wishes to exclude from the search hits can be defined in the following line. Elements must be defined using their atomic symbol.

```
not_elements = ['Br', 'Cl', 'I', 'Li', 'Na', 'K', 'Ca', 'Mg', 'Be']
```

It is also useful to be able to set a limit for the molecular weight of returned hit structures. Therefore, a molecular weight limit can be applied/adjusted using the −w keyword when running the search.

The catalophore searches were conducted using the CSD-PythonAPI with a maximum molecular weight of 500 Da, a maximum root-mean-square-deviation (RMSD) in geometry between catalophore and the hit of 1,[8,9] with Br, Cl, I, Li, Na, K, Ca, Mg, Be and transition metals excluded.[7] Only organic structures were included in the search by setting *is_organic* to True. SMILES code matching was used to remove duplicate structures. 3D structures were cleaned by assigning all unknown bond types, adding all missing hydrogens and setting all formal charges.

For piperidine, the catalophore search resulted in 26022 total hits, 14483 of which were unique. For 2-pyrrolidinone, the catalophore search resulted in 33780 total hits, 18886 of which were unique. Indexes of the coordinating atoms were automatically identified from the hit structure by matching the catsd_coordinating_atom_general feature, used to define the coordinating atoms, to the base atom, with a tolerance of 0.1Å for the x, y and z coordinated, and exported for use in structure generation. For linear_nb features the matched atom is hydrogen. In this case, as the hydrogen atom is deprotonated upon coordination to the metal centre, the atom index of the atom it is bonded to is required instead. The bonds that the hydrogen forms are retrieved from the *.mol* CSD entry and the atom index of the bonded atom extracted instead.

The results of a search are saved in the *'name.csv'* file. This file contains all of the hit structures, as their CSD Identifiers and important information in the following format.

```
CSD_Identifier, Index, Chemical Name, Structure File, Coord Atoms,
Freq, rmsd
```

Index is a unique suffix for each CSD Identifier which is required when duplicates are not removed, to distinguish between different crystal structures or coordination modes within the same hit. Chemical Name is the chemical name of the hit structure. Structure File is the name of the *.mol* file that is saved from the search. Coord Atoms are the coordinating atom indexes identified from the search and is used to define the metal-ligand bonds in the molSimplify input files. Freq is the frequency for the ligand to be used in the molSimplify input files. rmsd is the root mean squared deviation between the hit structure and the catalophore.

A molSimplify *.dict* file is also generated containing all of the relevant data required to use the 3D structures to generate organometallic complexes using molSimplify. The file contains the following information in the format:

```
CSD_Identifier, 3D structure file name, CSD_Identifier_1,
coordinating atom indexes, 'build custom custom', 'BA', formal charge
```

Values in quotes are fixed string values. CSD_Identifier is the CSD Identifier. 3D structure file name is the name of the ligand *.mol* file saved from the search. CSD_Identifier_1 is a unique name for each ligand and cannot be the same as CSD_Identifier. The same suffix is used as in the *.csv* Index. By default _1 is used as a suffix if there are no duplicate structures. If duplicate removal is turned off the suffix _X, where X is an integer, is used for structures with the same CSD Identifier. coordinating atom indexes are the atom indexes of the coordinating atoms extracted from the CSD-CrossMiner search and are used to form the bonds between the ligand and the metal centre. 'build custom custom' tells molSimplify that the ligand is used to build a custom complex. 'BA' is the type of force field optimization to use by default when using the ligand to build a complex. formal charge is the charge of the ligand and is used to calculate the total charge of the output complex. All hit structures are saved locally in the 3D *.mol* format. The resulting ligand sets were named ***ligands_CSD_PIP_set*** and ***ligands_CSD_PYR_set*** respectively.

4 High-throughput B97-3c//GFN2-xTB calculations of $\Delta G^{\ddagger}$ with different datasets

All automation scripts and data analysis is performed using Python 3. Organometallic structure generation is performed with a customised version of the molSimplify python package (version 1.2.7-alpha).[5] This workflow is only available on Linux due to software restrictions.

B97-3c calculations were performed using ORCA 4.2.1 with the *TightSCF* and *SlowConv* convergence criteria.[2] For piperidine, GFN2-xTB calculations were performed with standalone xtb 6.3.3 with the *tight* optimisation criteria for non-TS optimisation steps and xtb 6.3.3 interfaced with ORCA 4.2.1 for TS optimisations and frequency calculations with default convergence criteria.[3] For 2-pyrrolidinone, GFN2-xTB calculations were performed with xtb 6.3.3 interfaced with ORCA 4.2.1 for all calculations with the *TightOpt* criteria for non-TS optimisations and the default convergence criteria for TS and frequency calculations.[3] All TS calculations recalculate the hessian every 5 optimisation steps. All calculations use 4 CPU cores and 4GB of RAM. DMF was used as the solvent and caesium carbonate as the base.

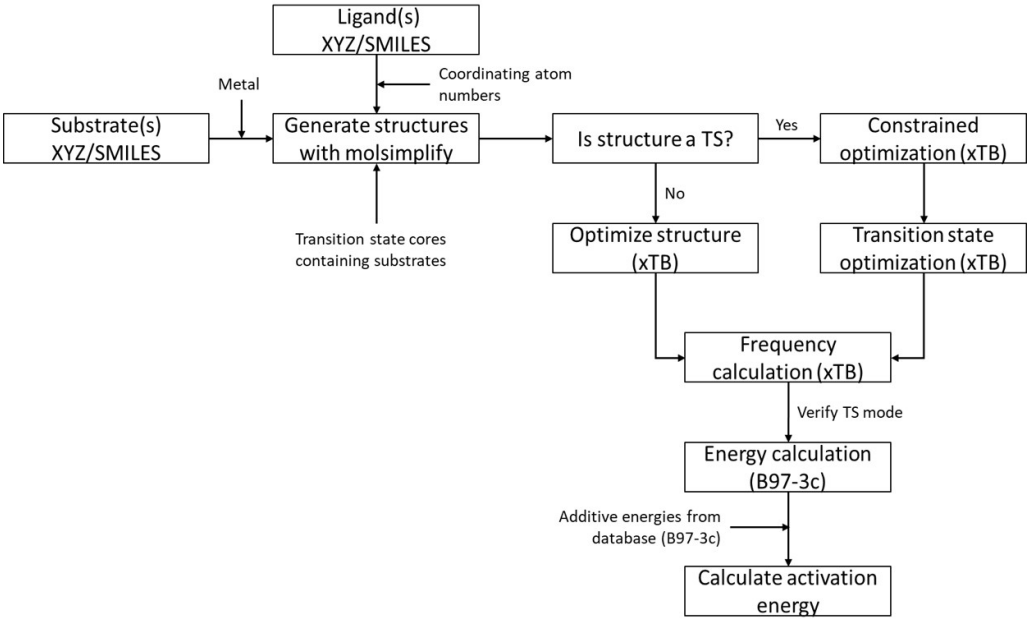


Fig. S19 Workflow for the prediction of activation energies

4.1 Automated building of structures of Cu(I) complexes from ligands with molSimplify

4.1.1 Generation of stable/intermediate structures

Organometallic complexes are generated in an automated manner using the molSimplify Python toolkit.[5] For each organometallic complex in a mechanistic pathway the metal centre including oxidation state and spin (in most cases), coordination number, and geometry is constant between complexes in the same step. This creates a template structure where only the ligand(s) needs to be adjusted for each structure. The ligand(s) and their frequency in the complex can then be defined for each unique ligand, the values of which depend on the ligand denticity or requirements for the final complex (additional fixed ligands). Any ligand that is required in every complex (fixed) can be included within the template. MolSimplify generates the charge of the complex automatically using openbabel.[10] In cases where a ligand must be deprotonated upon coordination to the metal centre, deprotonation is done automatically using a set of deprotonation rules which use SMILES matching to match functional groups. The deprotonation rules can be customised based on the user's needs. Structures can also be optimised with a force field to clean up the structures both before and after ligand addition. An example molSimplify input file for

the generation of the active catalytic species, which is a stable three-coordinate Cu(I) complex with one bidentate ligand and one deprotonated nucleophile is shown below.

```
-name AADMPY10_CuLpyr_1
-core copper
-oxstate I
-coord 3
-geometry tpl
-lig AADMPY10, pyrrolidinone
-ligocc 1, 1
-spin 1
-ff uff
-ffoption ba
-keepHs auto, False
-ligalign true
-skipANN true
```

The file contains all of the relevant information regarding the complex to be generated, including information about the metal centre, the structure of the ligands, how to deprotonate the ligands and whether to clean up the structure using a force field. Let's break down each line.

```
-name AADMPY10_CuLpyr_1
```

The −name line defines the name of the structure to be used in the output files. For example, the above example will output the structure as a *.xyz* file to the location *./AADMPY10_CuLpyr_1/AADMPY10_CuLpyr_1/AADMPY10_CuLpyr_1.xyz*. This value should contain all of the relevant information required to identify the complex. This provides a file structure where all of the complexes are separated allowing for easier handling of computations and analysis.

```
-core copper
-oxstate I
-spin 1
```

The metal centre is defined using the above terms. −core states the element to be used as the metal centre. The value can either be the name of the element or the atomic symbol (e.g. copper, cu, iron, fe). The oxidation state of the metal is defined with the −oxstate keyword. Values use roman numerals (e.g. I, IV, V) for positive oxidation states or negative numbers (-1, -2, ...) for negative oxidation states. Finally, the spin of the metal centre is defined with the −spin keyword and its value is the spin multiplicity of the metal centre (e.g. 1 - singlet, 2 - doublet, 3 - triplet).

```
-coord 3
-geometry tpl
```

To define the geometry of the complex the following keywords are used. −coord defines the coordination number of the complex (number of bonds between the metal centre and ligands). −geometry defines the geometry of the complex (e.g. tpl - trigonal planar). Custom geometries can be used if required, please see the official molSimplify documentation for instructions.[5]

```
-lig AADMPY10, pyrrolidinone
-ligocc 1, 1
```

Next, the ligand(s) to be added to the metal are defined. −lig contains the names of the ligands to be added. The names defined here are looked up from the ligands.dict file located in the default molSimplify folder which is by default located at ∼/molSimplify/Ligands/. The CSD-CrossMiner search script generates a **.dict* (where * is the name of the search) file containing all of the relevant values for each ligand from the CSD. The contents of which must be copied directly into the default

ligands.dict file. 3D structure files exported from the CSD-CrossMiner search must also be copied to the ~/molSimplify/Ligands/ folder.

The −ligocc keyword defines the frequency of each ligand in the same order they are stated in −lig. In this example, **AADMPY10** is a bidentate ligand and therefore only one is required for the trigonal planar complex along with one nucleophile. If **AADMPY10** was a monodentate ligand −ligocc 2, 1 would be the correct values to include two monodentate ligands and one nucleophile to fill all coordination sites.

```
-lig AADMPY10.mol, C1CC(=O)NC1
-ligocc 1, 1
-smicat [2, 19], [4]
```

Structures can also be generated from the 3D structure files located in the same folder as the input files by defining the file name of the ligand 3D structure (e.g. AADMPY10.mol) or by SMILES string and the indexes of the coordinating atoms (e.g. [2, 19]) in the SMILES string. Atom indexes must be enclosed in square brackets for each ligand and separated by a comma using the −smicat keyword. The indexes of the coordinating atoms must be defined for every explicitly defined file/SMILES string.

```
-ff uff
-ffoption ba
```

The complex can be cleaned up using a force field at several stages during structure generation. −ff defines the force field to be used. For organometallic structures, we recommend the Univeral Force Field (UFF). −ffoption defines when the structure is optimised. **b** optimises the ligands before they are added to the metal centre. This is only recommended when using SMILES strings. **a** optimises the structure after the ligands have been added. Both options can be used together (**ba**), which will optimise the ligands before addition and the complex after all ligands are added.

```
-keepHs auto, False
```

It is common that upon coordination to a metal, the ligand is deprotonated. In these cases, hydrogen atoms need to be removed from the starting 3D structure of the ligand. This can be done using the −keepHs keyword. In order to deprotonate the ligand, for example, the active catalytic state in the Ullmann-Goldberg reaction has a deprotonated nucleophile, the False value can be used. To keep all hydrogen atoms use the True value. Custom deprotonation rules can be used for deprotonation for a large range of ligands containing a variety of functional groups. This can be achieved using the auto value which uses SMARTS matching to deprotonate matching functional groups. Custom deprotonation rules can be defined by altering the self.remHsmarts line in the ~/molSimplify/Classes/globalvars.py file by replacing the list of SMARTS strings with a user-defined list.

```
-ligalign true
```

The −ligalign keyword is used to call the ligand alignment tool. This ensures that the ligands are added to the metal in order of steric bulk. This improves the structures generated as adding bulky ligands last can often lead to them not having enough space causing incorrect structures or failure of the program. Possible values are true and false.

```
-skipANN true
```

The −skipANN keyword is used to call the use of ANN-calculated bond lengths. This is only supported by molSimplify for specific elements and geometries. Ensure that the metal and geometry you are using are supported otherwise it can be turned off using the true value to save computational time.

Structures are generated automatically with the command:

```
molsimplify -i {input_file}
```

To generate all structures instead of just one, bash can be used to loop over all molSimplify input files:

```
for f in *.inp; do
    molsimplify -i {f};
done
```

The calculated charge and spin values for the complex are output to the terachem_input file and are important for the generation of computational input files.

4.1.2 Generation of transition state structures

Transition state structures are much harder to generate from scratch due to the non-standard bond lengths and bond angles present in the transition state. To generate a good starting structure for a transition state calculation the ligand replacement tool in molSimplify is used to replace the ligand in a transition state template with the ligands retrieved from CSD-CrossMiner, or from a SMILES string or other 3D structure file (e.g. *.xyz* or *.mol*).[5]

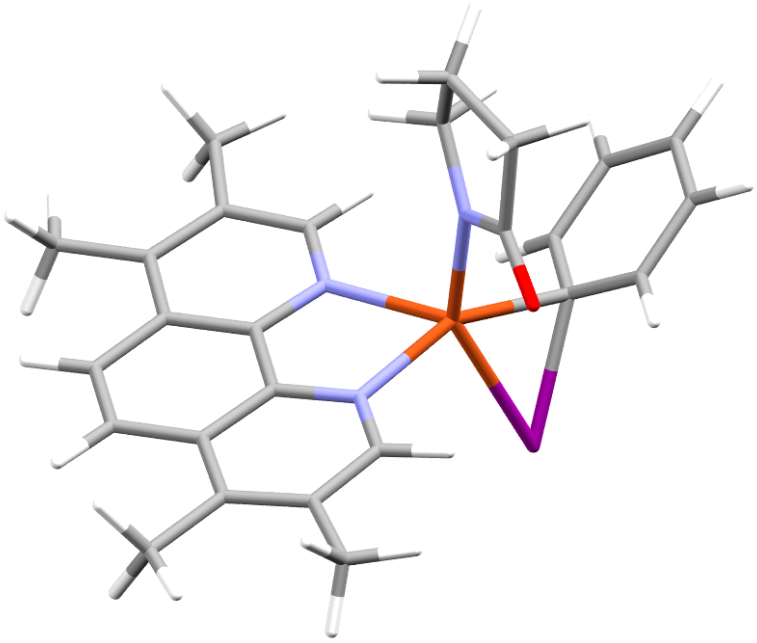


Fig. S20 Example transition state core.

Structures are built from a template structure that uses a 'simple' and 'common' ligand as a base. The template should be an optimised transition state of the transition state of interest. This structure should be optimised at the same level of theory as the computational calculations to be employed for the entire ligand dataset. An example 'core' is shown in Fig. S20 for the oxidative-addition transition state for 2-pyrrolidinone and iodobenzene using 3,4,7,8-tetramethyl-1,10-phenanthroline (TMPHEN) as the base ligand. An example input file for automated ligand replacement with molSimplify is shown below.

```
-name AADMPY10_TSOA_pyr_1
-core tsoa_pyr
-oxstate 0
-spin 1
-replig true
-lig AADMPY10
-ligocc 1
-ccatoms 6,15
-ligloc true
-ligalign true
-keepHs auto
-ffoption c
-skipANN true
```

The input file has several differences compared to the stable intermediates. Let's break down each line.

```
-name AADMPY10_TSOA_pyr_1
-ligalign true
-skipANN true
```

The above lines are the same as the stable/intermediate structure generation.

```
-core tsoa_pyr
-oxstate 0
-spin 1
```

Instead of a metal centre −core defines the name of the transition state template. This is the structure of the optimised transition state to be used as a template for ligand replacement. The structure file should be placed in the ∼/molSimplify/Cores/ folder and an entry added to the cores.dict file containing the following information in the following format: *'alias':'name of the XYZ file','indexes of the coordinating atoms', 'maximum denticity'*. For the oxidative-addition transition state for 2-pyrrolidinone, where the structures in contained in *TSOA_PYR.xyz*, the entry is *'tsoa_pyr:TSOA_PYR.xyz,6 15,6'*. Where atom indexes 6 and 15 are the indexes of the two TMPHEN nitrogen atoms and the core has a maximum denticity of 6. The −oxstate keyword is the oxidation state of the entire core. In this example the metal centre is copper(I) and is bound to a deprotonated nucleophile of charge, -1, and a neutral ligand so the core has an oxidation state of 0. −spin is the spin of the core.

```
-replig true
-lig AADMPY10
-ligocc 1
```

In order to enable ligand replacement the −replig true command is required. The ligand to be added is defined as −lig 'ligand'. As with the stable complexes the ligand can be defined as a 3D structure from the *ligands.dict* file or be defined as a 3D structure file in the current folder or a SMILES string. The ligand coordinating atoms are taken directly from the *ligands.dict* file. When using a structure not defined in ligands.dict, e.g. a SMILES string, the −smicat keyword should be used to define the indexes of the coordinating atoms as described for intermediate structures. The number of each ligand is defined using −ligocc 'frequency'.

```
-ccatoms 6,15
```

−ccatoms defines the atom indexes of the atoms in the ligand to be replaced in the core which coordinates to the metal centre. −ccatoms 6,15 refers to the two nitrogen atoms in the TMPHEN ligand in the core.

```
-ligloc true
```

The −ligloc keyword enforces ligand location. This ensures that the ligand is placed in the correct position around the metal centre.

```
-keepHs auto
```

As in the stable complexes −keepHs is used to deprotonate the ligand structures. As only the ligand is added to the structure only auto need to be used if using custom deprotonation rules. true and false can be used if no deprotonation or forced deprotonation is required respectively.

```
-ffoption c
```

In order to maintain the transition state mode in structure generation both force field options a new −ffoption **c** has been implemented. **c** stands for core-constrained and freezes all the atoms in the core, resulting in only the ligand being optimised. **c** can be used alongside **b** as **b** does not optimise the core, only the ligand before addition.

Table S8 Success rate for different force field optimization methods for transition state generation for the Ullmann-Goldberg reaction, (TSOA: oxidative-addition, TSSig: sigma-metathesis) for all ligands in ***ligands_lit_set***

Success Rates	Before (**b**)	Core (**c**)	Before + Core (**bc**)	No Force Field
TSOA	71%	87%	86%	68%
TSSig	82%	93%	93%	79%
Total	76%	90%	90%	74%

Table S8 shows the success rate for each force field option for the generation of two transition states (TSOA and TSSig). Ligands were added to the structure via SMILES string. Using core-constrained optimization improves the success rate of structure generation by ∼15% compared to using no force field and before optimization. Using both before and core-constrained optimization offers little advantage compared to just core-constrained force field optimization but comes at an additional computational cost from the additional force field optimisation step.

4.1.3 Correction of Coordinating Atom Indexes

Due to the deprotonation of some ligands during complex generation, the atom indexes of the coordinating atoms in the ligand may be different from those extracted from CSD-CrossMiner. This is due to the removal of the hydrogens in the atom lists. The location at which the hydrogens were present in the atom list determines whether correction of these indexes is required. Correction of the atom indexes is required for the correct analysis of these atoms when extracting specific properties from computational output files. To correct the indexes of these atoms the following method is employed:

1. The atom lists for the **CuLI** and ligand is read from their respective *.xyz*/*.mol* files.
2. The Cu and I atoms are removed from the **CuLI** complex atom list.
3. The lists of atoms from the CuLI complex and the free ligand are compared.

The complex atom list is iterated through and the atomic symbol is compared to the free ligand atom list. When a miss-match is found the index (n) is recorded and then compared against the next item (n + number of miss-matched items) in the list. If the index of the miss-matched item is greater than the index of the coordinating atoms no adjustments are needed. If the index of the miss-matched item is before the index of the coordinating atom a hydrogen has been removed. Therefore, the coordinating atom index is adjusted based on the number of miss-matched indexes lower than the coordinating atom index. This process fails if a hydrogen has been removed from a section of an atom list containing multiple hydrogen atoms. In this case, the following process is employed:

For each atom at location (n) in the list, the list is propagated forward to find the length of the section in the list with the same repeating atomic symbol. This process is done for both structures. The lengths

of the repeating sections are then compared, if they are not equal then the deprotonated hydrogen was present in that section of the atom list. The starting index of the section containing the removed hydrogen is then compared to the index of the coordinating atoms. The index is then adjusted using the same comparison method.

4.2 Automated generation of GFN2-xTB and B97-3c input files

Computational input files are generated automatically using Python. The script to generate these input files is provided and can be modified to use custom parameters or change the method used. The script must in run in the folder with the *.xyz* file generated by molSimplify. This can be done on scale using a bash script to loop through each folder. ORCA input files use the *.xyz* file to obtain the coordinates of the structures this allows for easy chaining between calculations so only one input file generation step is required. For example the name of the output *.xyz* file of the optimisation step is predictable and therefore can be placed in the frequency and energy input file before the calculation is run. Charge and spin data is obtained from the *terachem_input* file created by molSimplify and is automatically extracted and added to the ORCA input files. The folder can then be copied to a HPC to run the calculations or run locally.

4.3 Automated extraction of computational results

4.3.1 Automated screening of failed B97-3c//GFN2-xTB calculations

In order to validate the structure of the transition state a modified version of the TS vetting requirements presented by Jacobsen et al. is used.[11] Structures are automatically tested by running the *ts_check.py* script. This script requires the *.xyz* file for the optimisation step and the frequency output file. This script will iterate through each folder and check that intermediate structures are at a minimum and that transition state structures meet all of the following three criteria: i) exactly one imaginary frequency of the hessian. ii) the TS active bond (bond being broken or formed) must be of an intermediate length:

$$1.7 \geq \frac{r_{ij}}{(r_i^{cov} + r_j^{cov})} > 1.0 \tag{2}$$

where r_{ij} is the bond length between atoms i and j and r_i^{cov} and r_j^{cov} are the covalent radii of atoms i and j. iii) the eigenvector corresponding to the imaginary frequency should have motion along one of the TS active bond stretching modes:

$$\left| v_i^{stretch} \cdot v^{ts} \right| \geq S_0 \tag{3}$$

where $v_i^{stretch}$ is the eigenvector of the imaginary frequency, $v_i^{stretch}$ is the unit vector of the stretching mode of bond i and S_0 is the amount of overlap between the two vectors. S_0 is a constant of default value 0.33. The value of S_0 needs to be tuned depending on the type of transition state. Transition states which are not a simple bond stretch along the TS active bonds are not well described with an S_0 value of 0.33. For example, in transition states possessing a bend-like character, commonly observed in some oxidative additions, tuning the value of S_0 is required.

An output *.csv* file is produced containing a summary of each structure. The script can be adjusted to change the tolerances. While the script is written for this reaction only it can be adjusted for other reaction types if required.

4.3.2 Automated extraction of $\Delta G^{\ddagger}$ from computational output files

Activation energies are calculated using Python (*Energy_Analysis_CrossMiner.py*). The script iterates through each folder and calculates $\Delta G^{\ddagger}$ for each ligand and transition state. E_{el} is taken from the *_energy.out* file and the vibrational correction, $G_{correction}$ is taken from the frequency output. The Gibbs free energy is then calculated with the following formula:

$$G^0 = E_{el}(B97-3c) + G_{correction}(GFN2-xTB) \tag{4}$$

Additive (e.g. $CsCO_3$, HCO_3) energies are taken from a database. The activation energy of each pathway was calculated as the difference between the Gibbs free energy of the transition state and the lowest energy intermediate structure. Relative energies were calculated relative to the CuLI complex.

$$E_A = \sum_1^n G_{products} - \sum_1^n G_{reactants} \tag{5}$$

The energy is converted from Eh to $\mathrm{kcal\,mol^{-1}}$ and output to a *.csv* file.

4.3.3 Automated extraction of descriptors from computational output files

Steric and electronic descriptors are calculated from the computational output files. Extraction of descriptors is performed using Python 3 with the morfeus package[12] for steric descriptors and a customised version of the cclib package (version 1.7.1)[13] with expanded functionality for the extraction of additional electronic data from ORCA output files. The morfeus package calculates properties for monodentate ligands by default. To ensure Cone Angle and Sterimol parameters (B1, B5 and L) are calculated correctly for bidentate ligands a dummy hydrogen atom is placed at the midpoint of the two ligand coordinating atoms. The indexes of the two coordinating atoms, L1 and L2, define the *xy* and *z* planes. For Buried Volume, the substrates were removed from the complex to ensure that only the Buried Volume of the ligand is calculated. Buried Volumes were calculated at 3.5Å 5Å and 7Å with hydrogen atoms, excluding the dummy atom, included. For descriptors describing a change in length or angle, the difference is taken between the transition state and the **CuLI** intermediate. Descriptors are saved in a *.csv* file.

5 Machine Learning

5.1 Descriptors and descriptors selection

Machine learning models were created using the *scikit-learn* module with Python 3.[14] Hyperparameters were optimised using the *optuna* Python module.[15] DFT methods used in the machine learning electronic descriptor calculations were performed using ORCA 5.0.1, all methods use D4 dispersion correction and DMF as the solvent.[16] The *TightSCF* and *SlowConv* convergence criteria were used in all cases.

5.1.1 List of initial descriptors and their sources

All calculated descriptors for each of the four datasets are available as separate *.csv* files in the provided Zenodo repository.

5.1.1.1 Transition-state-dependent descriptors

Table S9 Full list of descriptors and their source. Entries highlighted in red and blue are for TSOA and TSSig only respectively. Descriptors 25-31 are extracted for the Cu, nucleophile N, ligand coordinating atom 1 (L1), ligand coordinating atom 2 (L2) and the iodobenzene C and I atoms. For 2-pyrrolidinone the amide C and O atoms are also included for the **TSSig** transition state. No d-orbitals were extracted for the nucleophile nitrogen.

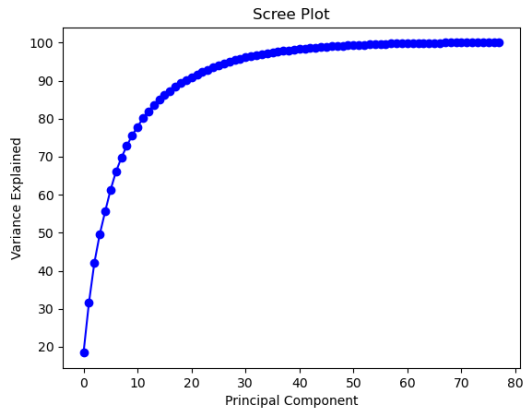
No.	Descriptor	Source	Description
1	Bite_Angle	Python	Ligand bite angle (°)
2	D_Bite_Angle	Python	Change in bite angle from the Cu-I intermediate to the transition state (°)
3	Cone_Angle	Python	Ligand Cone Angle (°)
4	Sterimol_B1	Python	Smallest distance perpendicular to the Cu-dummy vector to the edge of the ligand (Å)
5	Sterimol_B5	Python	Largest distance perpendicular to the Cu-dummy vector to the edge of the ligand (Å)
6	Sterimol_L	Python	Distance along the Cu-dummy vector to the edge of the ligand (Å)
7	PC_Buried_Volume_3-5Å	Python	Percentage buried volume at a 3.5Å radius (%)
8	PC_Buried_Volume_5Å	Python	Percentage buried volume at a 5Å radius (%)
9	PC_Buried_Volume_7Å	Python	Percentage buried volume at a 7Å radius (%)
10	SASA	Python	Solvent Accessible Surface Area (Å^2)
11	HOMO_Energy	ORCA	Energy of the HOMO (eV)
12	LUMO_Energy	ORCA	Energy of the LUMO (eV)
13	Cu-L$_x$	Python	Bond distance between Cu and ligand atom x (Å)
14	D_Cu-L$_x$	Python	Change in bond distance between Cu and ligand atom x between the CuLI intermediate and transition state (Å)
15	Cu-I	Python	Cu-I bond distance (Å)
16	Cu-C	Python	Cu-C bond distance (Å)
17	Cu-N	Python	Cu-N bond distance (Å)
18	C-I	Python	C-I bond distance (Å)
19	I-C-Cu	Python	I-C-Cu bond angle (°)
20	N-Cu-I	Python	N-Cu-I bond angle (°)
21	Cu-I-C	Python	Cu-I-C bond angle (°)
22	I-C-N	Python	I-C-N bond angle (°)
23	C-N-Cu	Python	C-N-Cu bond angle (°)
24	C-Cu-I	Python	C-Cu-I bond angle (°)
25	Lowdin_Charge	ORCA	Lowdin atomic charge of the atom
26	Bonded_Valence	ORCA	Number of bonds formed by the atom
27	Atomic_Population	ORCA	Number of electrons localised on the atom
28	Bond_Order	ORCA	Number of bonds between two atoms
29	Orbital_Charge_s	ORCA	Orbital charge of the s orbital
30	Orbital_Charge_p and subshells	ORCA	Orbital charge of the p orbital and its subshells
31	Orbital_Charge_d and subshells	ORCA	Orbital charge of the d orbital and its subshells
32	Img_Freq	ORCA	Magnitude of the imaginary frequency (cm^{-1})

5.1.1.2 Transition-state-independent descriptors

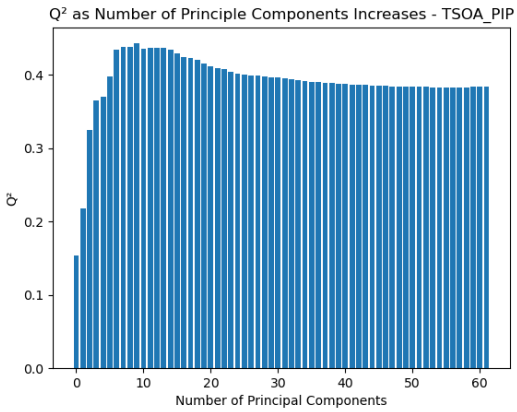
Table S10 Full list of descriptors for the TS independent descriptor sets and their source. Descriptors are calculated from the CuLNu active catalytic state. Descriptors 15-20 were extracted for the Cu, nucleophile N, ligand coordinating atom 1 (L1) and ligand coordinating atom 2 (L2) atoms.

No.	Descriptor	Source	Description
1	Bite_Angle	Python	Ligand bite angle (°)
2	Cone_Angle	Python	Ligand Cone Angle (°)
3	Sterimol_B1	Python	Smallest distance perpendicular to the Cu-dummy vector to the edge of the ligand (Å)
4	Sterimol_B5	Python	Largest distance perpendicular to the Cu-dummy vector to the edge of the ligand (Å)
5	Sterimol_L	Python	Distance along the Cu-dummy vector to the edge of the ligand (Å)
6	PC_Buried_Volume_3-5Å	Python	Percentage buried volume at a 3.5Å radius (%)
7	PC_Buried_Volume_5Å	Python	Percentage buried volume at a 5Å radius (%)
8	PC_Buried_Volume_7Å	Python	Percentage buried volume at a 7Å radius (%)
9	SASA	Python	Solvent Accessible Surface Area ($Å^2$)
10	HOMO_Energy	ORCA	Energy of the HOMO (eV)
11	LUMO_Energy	ORCA	Energy of the LUMO (eV)
12	Cu-L_x	Python	Bond distance between Cu and ligand atom x (Å)
13	Cu-N	Python	Cu-N bond distance (Å)
14	Bond_Order	ORCA	Number of bonds between two atoms
15	Lowdin_Charge	ORCA	Lowdin atomic charge of the atom
16	Bonded_Valence	ORCA	Number of bonds formed by the atom
17	Atomic_Population	ORCA	Number of electrons localised on the atom
18	Orbital_Charge_s	ORCA	Orbital charge of the s orbital
19	Orbital_Charge_p and sub-shells	ORCA	Orbital charge of the p orbital and its sub-shells
20	Orbital_Charge_d and sub-shells	ORCA	Orbital charge of the d orbital and its sub-shells

5.2 Scree and Q^2 plots

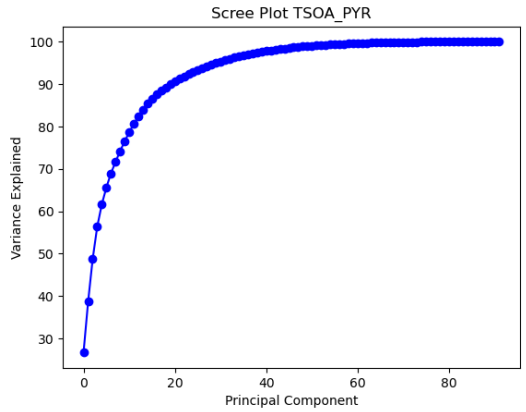


(a) Scree plot.

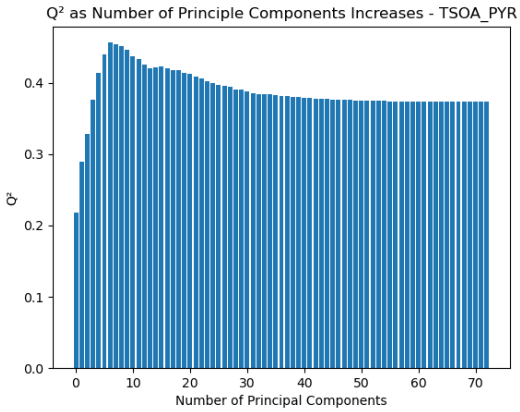


(b) Q^2 plot.

Fig. S21 Exploratory factor analysis for the PIP_set_TSOA dataset.

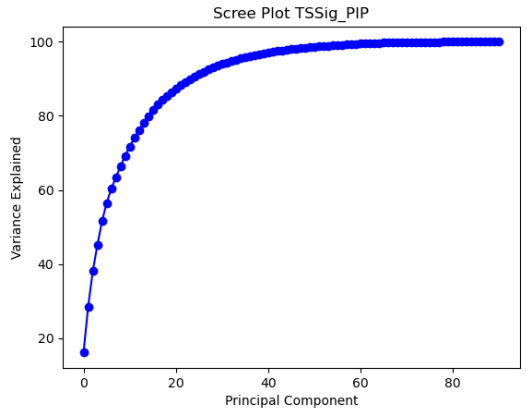


(a) Scree plot.

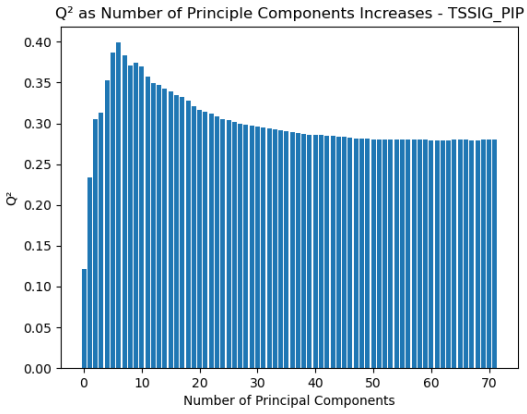


(b) Q^2 plot.

Fig. S22 Exploratory factor analysis for the PYR_set_TSOA dataset.

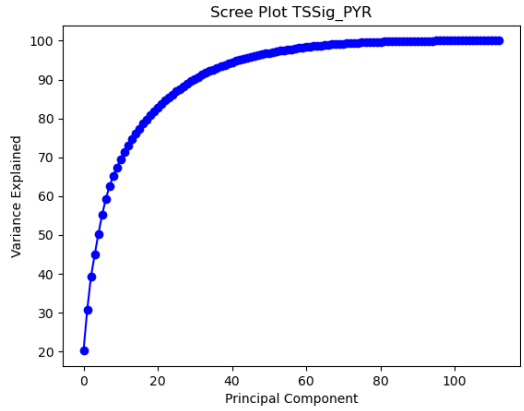


(a) Scree plot.

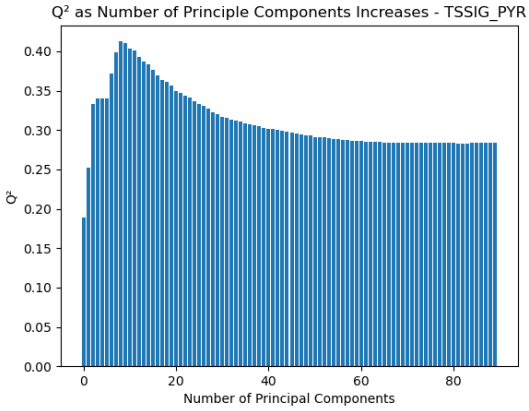


(b) Q^2 plot.

Fig. S23 Exploratory factor analysis for the PIP_set_TSSig dataset.



(a) Scree plot.



(b) Q^2 plot.

Fig. S24 Exploratory factor analysis for the PYR_set_TSSig dataset.

5.3 Correlation between $\Delta G^{\ddagger}$ and descriptors

Table S11 Pearson's R^2 between each descriptor and activation energy for every descriptor in the ***PIP_set_TSOA***, ***PYR_set_TSOA***, ***PIP_set_TSSig*** and ***PYR_set_TSSig*** datasets.

Descriptor	R^2			
	PIP_set_TSOA	PYR_set_TSOA	PIP_set_TSSig	PYR_set_TSSig
Bite Angle	−0.03	−0.02	−0.29	−0.10
Change in Bite Angle	−0.03	0.11	−0.06	0.06
Cone Angle	0.11	0.23	−0.22	−0.02
Sterimol B1	0.02	−0.01	0.00	−0.06
Sterimol B5	−0.05	0.03	−0.11	−0.01
Sterimol L	−0.09	−0.13	0.05	−0.11
PC_Buried_Volume_3-5A	0.16	0.26	−0.17	0.07
PC_Buried_Volume_5A	0.12	0.21	−0.18	0.03
PC_Buried_Volume_7A	0.04	0.09	−0.16	−0.03
SASA	−0.06	−0.05	−0.09	−0.09
HOMO Energy	0.29	0.29	0.05	0.47
LUMO Energy	0.17	0.24	0.10	0.46
Cu-L1	−0.05	0.05	0.11	0.03
Cu-L2	−0.03	0.07	0.07	0.04
D_Cu-L1	0.05	−0.08	0.08	−0.04
D_Cu-L2	0.00	−0.14	0.03	−0.02
Cu-I	0.13	0.25	−0.24	0.17
Cu-N	-	-	−0.21	0.19
Cu-C	0.32	0.16	−0.27	0.38
Cu-O	-	-	-	−0.04
C-I	−0.29	−0.18	0.12	−0.18
Amide C-O	-	-	-	0.27
Amide C-N	-	-	-	−0.32
N-Cu-I	-	-	0.27	−0.23

Continuation of Table S11

Descriptor	PIP_set_TSOA	PYR_set_TSOA	PIP_set_TSSig	PYR_set_TSSig
Cu-I-C	0.35	0.15	0.07	0.33
I-C-N	-	-	−0.23	0.04
C-N-Cu	-	-	−0.29	−0.05
I-C-Cu	0.14	0.20	-	-
C-Cu-I	−0.31	−0.21	-	-
Amide O-C-N	-	-	-	0.29
Lowdin Charge (Cu)	−0.14	−0.14	−0.07	−0.05
Lowdin Charge (N)	-	-	0.00	−0.28
Lowdin Charge (C)	−0.29	−0.20	0.05	−0.32
Lowdin Charge (I)	0.30	0.17	0.10	0.03
Lowdin Charge (L1)	0.07	0.03	−0.01	−0.14
Lowdin Charge (L2)	0.09	0.02	−0.04	−0.16
Lowdin Charge (Amide C)	-	-	-	−0.18
Lowdin Charge (Amide O)	-	-	-	−0.28
Bonded Valence (Cu)	−0.11	−0.03	0.12	0.00
Bonded Valence (N)	-	-	0.14	−0.29
Bonded Valence (C)	0.01	0.03	−0.06	0.23
Bonded Valence (I)	0.33	0.14	0.24	0.01
Bonded Valence (L1)	0.07	0.07	−0.04	−0.09
Bonded Valence (L2)	0.10	0.06	−0.06	−0.1
Bonded Valence (Amide C)	-	-	-	0.06
Bonded Valence (Amide O)	-	-	-	−0.14
Atomic Population (Cu)	−0.11	0.07	0.37	0.38
Atomic Population (N)	-	-	−0.10	−0.20
Atomic Population (C)	0.16	0.16	0.11	−0.12
Atomic Population (I)	−0.19	0.05	−0.04	0.10
Atomic Population (L1)	−0.05	−0.03	−0.01	0.04
Atomic Population (L2)	−0.07	−0.08	0.05	0.05
Atomic Population (Amide C)	-	-	-	−0.05
Atomic Population (Amide O)	-	-	-	0.13
Bond Order (Cu-I)	−0.28	−0.27	0.27	−0.03
Bond Order (Cu-C)	−0.34	−0.15	0.03	0.06
Bond Order (C-I)	0.25	0.21	−0.20	0.21
Bond Order (Cu-N)	-	-	0.05	−0.30
Bond Order (Cu-L1)	0.04	0.06	−0.07	0.02
Bond Order (Cu-L2)	0.05	0.00	−0.01	0.00
Bond Order (Amide C-O)	-	-	-	−0.16
Bond Order (Amide C-N)	-	-	-	0.23
Orbital Charge C(s)	0.31	0.10	0.13	0.11
Orbital Charge C(p)	0.26	0.20	−0.06	0.28
Orbital Charge C(pz)	0.04	0.17	−0.05	−0.02
Orbital Charge C(px)	−0.02	0.13	0.03	−0.11
Orbital Charge C(py)	0.12	−0.08	−0.03	0.17
Orbital Charge N(s)	-	-	−0.17	0.13
Orbital Charge N(p)	-	-	0.03	0.21

Continuation of Table S11

Descriptor	PIP_set_TSOA	PYR_set_TSOA	PIP_set_TSSig	PYR_set_TSSig
Orbital Charge N(pz)	-	-	0.06	0.16
Orbital Charge N(px)	-	-	−0.06	0.04
Orbital Charge N(py)	-	-	0.01	−0.17
Orbital Charge I(s)	−0.39	−0.25	−0.23	−0.26
Orbital Charge I(p)	−0.33	−0.20	−0.17	−0.06
Orbital Charge I(pz)	−0.14	0.11	0.02	0.11
Orbital Charge I(px)	−0.15	−0.11	−0.08	−0.13
Orbital Charge I(py)	0.03	−0.13	−0.03	−0.02
Orbital Charge I(d)	0.48	0.33	0.21	0.40
Orbital Charge I(dxz)	0.23	0.06	0.06	0.09
Orbital Charge I(dyz)	0.16	0.21	0.08	0.06
Orbital Charge I(dxy)	0.20	0.19	0.17	0.25
Orbital Charge I(dz2)	0.25	0.10	0.01	0.06
Orbital Charge I(dx2y2)	0.04	0.26	0.10	0.32
Orbital Charge Cu(s)	0.11	0.00	0.01	0.00
Orbital Charge Cu(p)	−0.08	−0.02	0.05	−0.11
Orbital Charge Cu(pz)	−0.01	−0.22	0.18	−0.20
Orbital Charge Cu(px)	−0.15	−0.07	−0.09	0.06
Orbital Charge Cu(py)	0.01	0.25	−0.02	−0.02
Orbital Charge Cu(d)	0.27	0.24	0.05	0.19
Orbital Charge Cu(dxz)	0.16	0.04	0.05	0.11
Orbital Charge Cu(dyz)	0.04	0.06	−0.08	0.31
Orbital Charge Cu(dxy)	−0.03	−0.11	−0.03	0.05
Orbital Charge Cu(dz2)	0.04	0.18	0.09	−0.22
Orbital Charge Cu(dx2y2)	0.05	0.06	0.10	−0.16
Orbital Charge L1(s)	−0.03	−0.02	0.00	0.04
Orbital Charge L1(p)	−0.03	0.00	−0.02	0.06
Orbital Charge L1(pz)	−0.01	0.05	0.00	0.13
Orbital Charge L1(px)	0.03	0.02	−0.07	0.01
Orbital Charge L1(py)	−0.09	−0.06	0.05	0.05
Orbital Charge L1(d)	−0.09	−0.08	−0.02	0.01
Orbital Charge L1(dxz)	−0.11	−0.06	0.05	0.11
Orbital Charge L1(dyz)	−0.05	0.00	−0.08	0.02
Orbital Charge L1(dxy)	−0.08	−0.12	−0.03	−0.04
Orbital Charge L1(dz2)	−0.10	−0.01	0.00	0.08
Orbital Charge L1(dx2y2)	−0.08	−0.11	−0.01	−0.05
Orbital Charge L2(s)	−0.05	−0.05	0.06	0.05
Orbital Charge L2(p)	−0.06	−0.04	0.04	0.08
Orbital Charge L2(pz)	−0.04	0.01	0.04	0.14
Orbital Charge L2(px)	0.01	−0.04	−0.02	0.03
Orbital Charge L2(py)	−0.10	−0.07	0.12	0.05
Orbital Charge L2(d)	−0.08	−0.09	0.00	−0.01
Orbital Charge L2(dxz)	−0.09	−0.08	0.08	0.07
Orbital Charge L2(dyz)	−0.06	−0.04	−0.07	−0.01
Orbital Charge L2(dxy)	−0.08	−0.11	−0.01	−0.06

Continuation of Table S11

Descriptor	PIP_set_TSOA	PYR_set_TSOA	PIP_set_TSSig	PYR_set_TSSig
Orbital Charge L2(dz2)	−0.08	−0.06	0.03	0.06
Orbital Charge L2(dx2y2)	−0.07	−0.10	0.00	−0.06
Orbital Charge Amide C(s)	-	-	-	−0.23
Orbital Charge Amide C(p)	-	-	-	0.34
Orbital Charge Amide C(pz)	-	-	-	−0.11
Orbital Charge Amide C(px)	-	-	-	−0.03
Orbital Charge Amide C(py)	-	-	-	0.16
Orbital Charge Amide O(s)	-	-	-	−0.10
Orbital Charge Amide O(p)	-	-	-	0.27
Orbital Charge Amide O(pz)	-	-	-	0.12
Orbital Charge Amide O(px)	-	-	-	−0.14
Orbital Charge Amide O(py)	-	-	-	0.19
Magnitude of the Imaginary Frequency	0.00	0.16	0.02	0.33

5.4 Initial ML models building with transition state descriptors

Eight machine learning models were employed; Multiple Linear Regression (MLR), Gaussian Process Regression (GP), Artificial Neural Networks (ANN), Support Vector Machine (SVM), Partial Least Squares (PLS), Random Forest (RF), ExtraTrees (ET) and Bagging (Bag). Default parameters were used with the following exceptions: for GP only the Matern. RBF and RationalQuadratic kernel were used; for ANN, n_nodes (number of nodes in the hidden layers) was optimised with the number of hidden layers varied; for SVM the radial basis function (RBF) kernel was used with C, epsilon and gamma being optimised; for PLS, n_components (number of components to retain after dimension reduction) was optimised; and for RF, ET and Bag, n_estimators (number of trees) and max_depth was optimised. Machine learning was performed in Python 3 with the scikit-learn module. Prior to machine learning, all descriptors were scaled using scikit-learn's StandardScaler() method. Where parameters were optimised the Optuna python package was used.[15] All parameters were optimised to maximise the Coefficient of Determination (R^2).

Datasets were split into training and test sets by binning the data in intervals of 1 kcalmol^{-1}. A proportional amount of data was taken from each bin to form a training set ($\sim$ 80% of the data) and a test set ($\sim$ 20% of the data). Each model was trained on the same training set and tested on the same unseen test set.

Performance metrics are obtained by splitting the data into k groups using the scikit-learn's KFolds method ensuring the dataset was shuffled before splitting. Each group is used as a test set and the remaining $k-1$ groups are used as the training set. After each group, the performance metrics are stored and the model discarded. All stated uses of K-fold cross-validation use 10 folds.

5.4.1 Initial model metrics

Table S12 Performance metrics for the machine learning models for the PIP_set_TSOA dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.39	8.64	58.8
GPR	0.25	9.70	72.9
ANN	0.45	8.55	61.4
SVM	0.33	9.14	74.4
PLS	0.44	8.31	60.1
RF	0.43	8.44	74.2
ExtraTrees	0.49	7.90	75.4
Bagging	0.41	8.56	73.1

Table S13 Performance metrics for the machine learning models for the PYR_set_TSOA dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.37	8.71	43.6
GPR	0.51	7.48	63.7
ANN	0.52	8.00	57.0
SVM	0.55	7.18	64.0
PLS	0.34	8.92	43.6
RF	0.60	6.76	64.6
ExtraTrees	0.65	6.32	66.1
Bagging	0.61	6.72	64.3

Table S14 Performance metrics for the machine learning models for the PIP_set_TSSig dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.28	6.39	67.7
GPR	0.21	6.89	77.2
ANN	0.36	6.23	73.2
SVM	0.31	6.28	80.4
PLS	0.26	6.47	70.1
RF	0.39	5.92	77.3
ExtraTrees	0.39	5.93	77.9
Bagging	0.39	5.92	77.3

Table S15 Performance metrics for the machine learning models for the PYR_set_TSSig dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.53	6.24	56.1
GPR	0.35	7.86	62.1
ANN	0.52	6.51	63.2
SVM	0.59	5.80	67.4
PLS	0.50	6.47	55.3
RF	0.63	5.57	67.3
ExtraTrees	0.63	5.52	68.5
Bagging	0.62	5.58	67.3

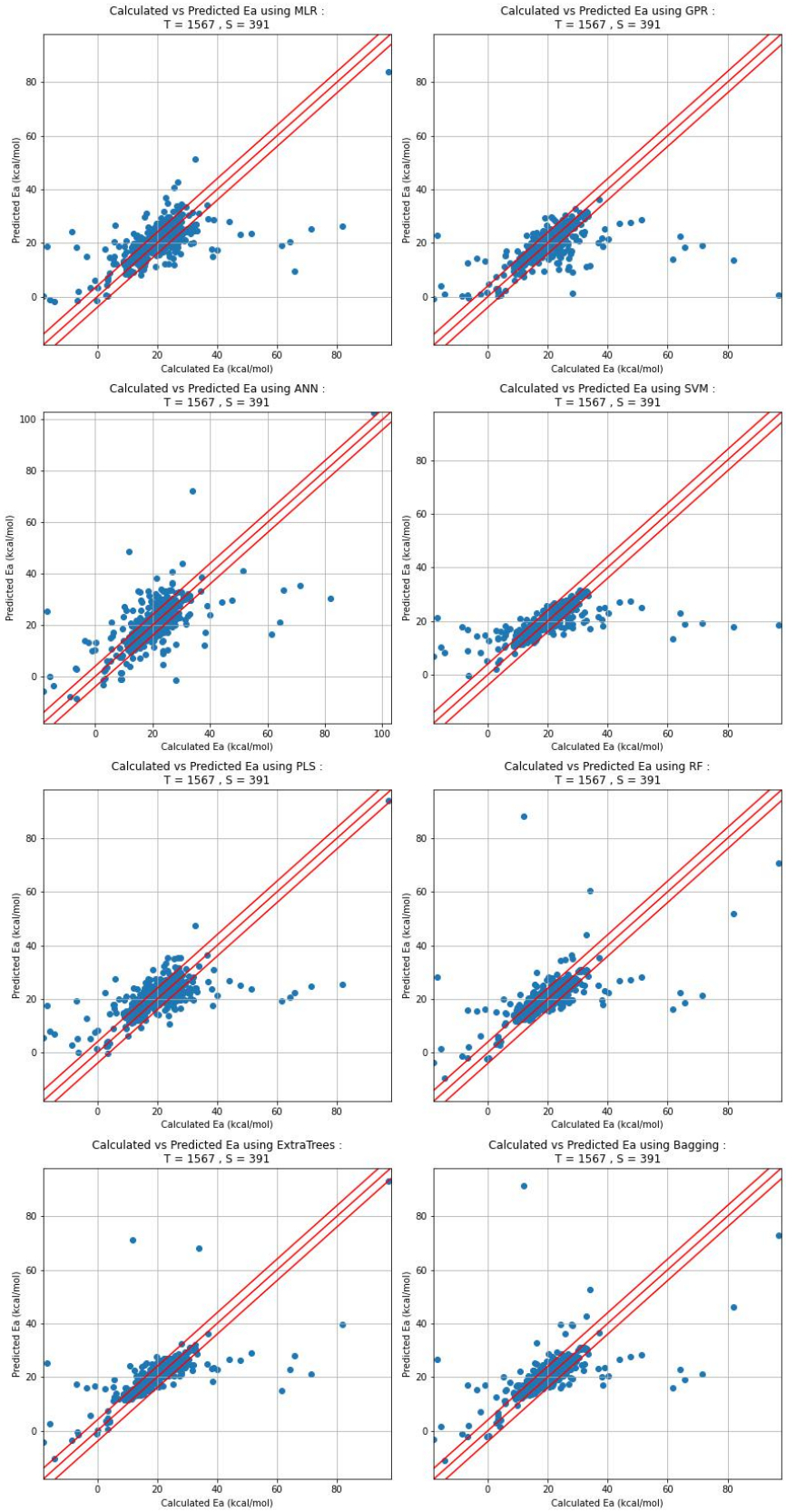


Fig. S25 Machine learning model predictions for the PIP_set_TSOA dataset.

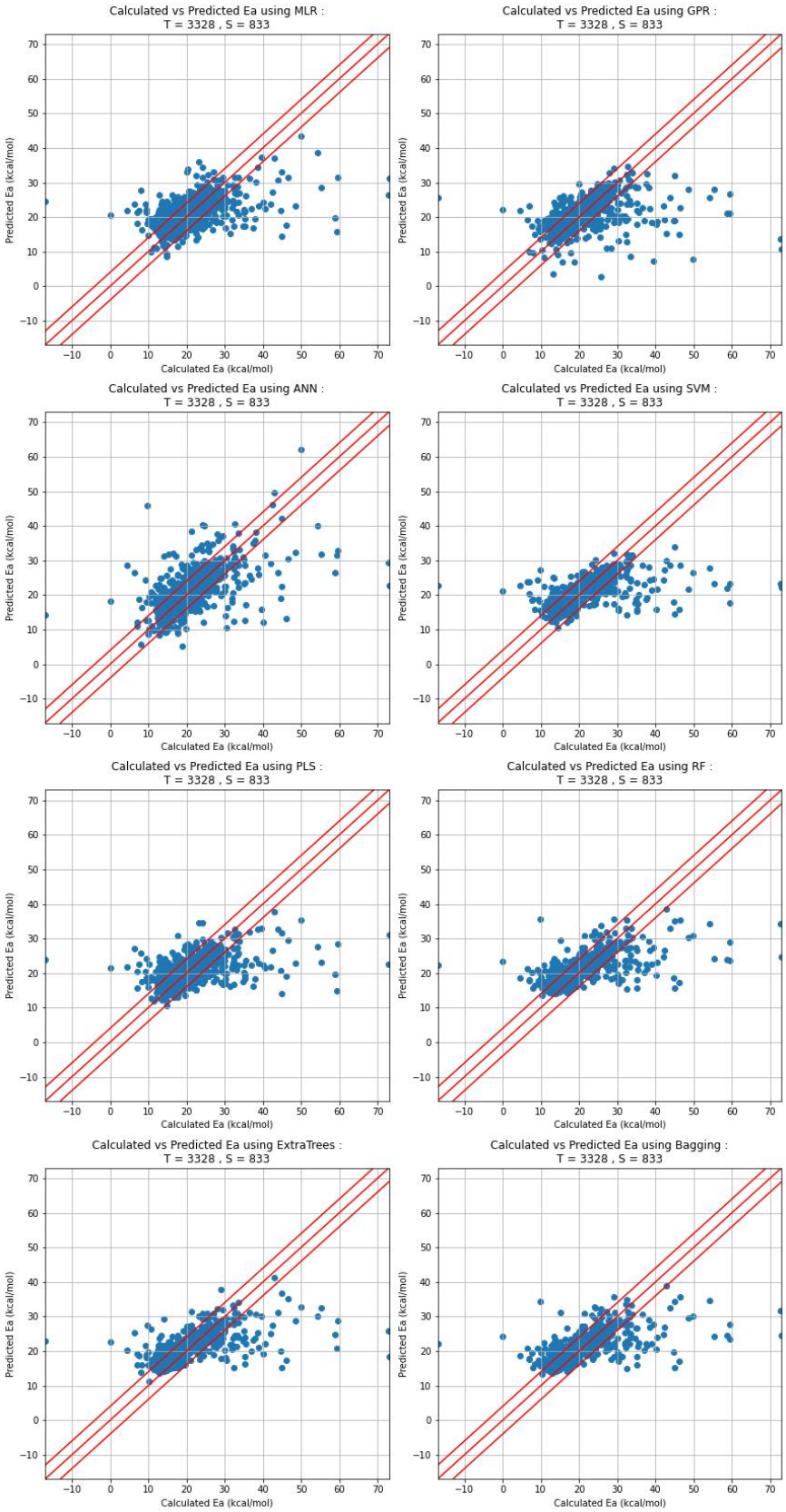


Fig. S26 Machine learning model predictions for the PIP_set_TSSig dataset.

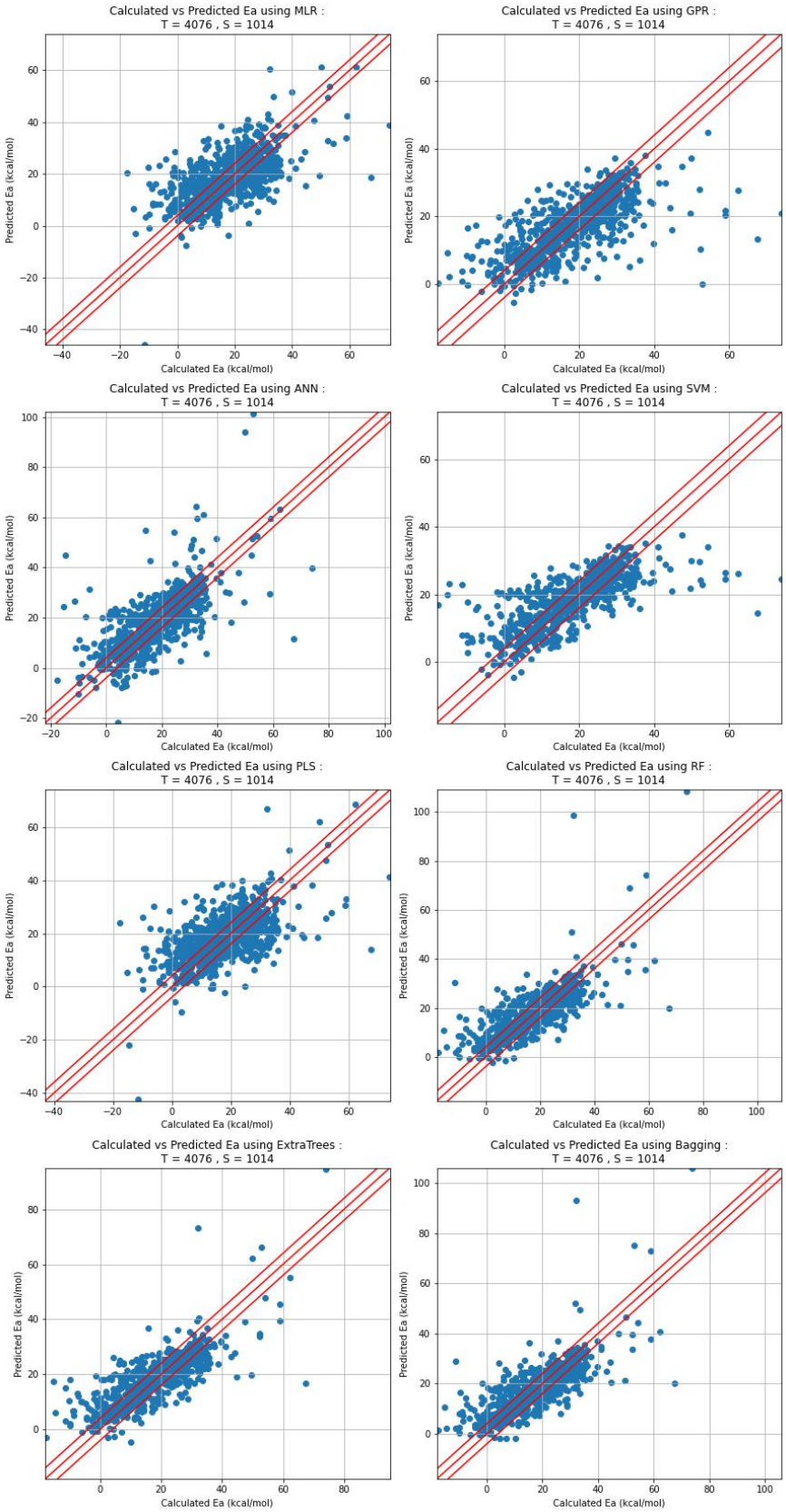


Fig. S27 Machine learning model predictions for the PYR_set_TSOA dataset.

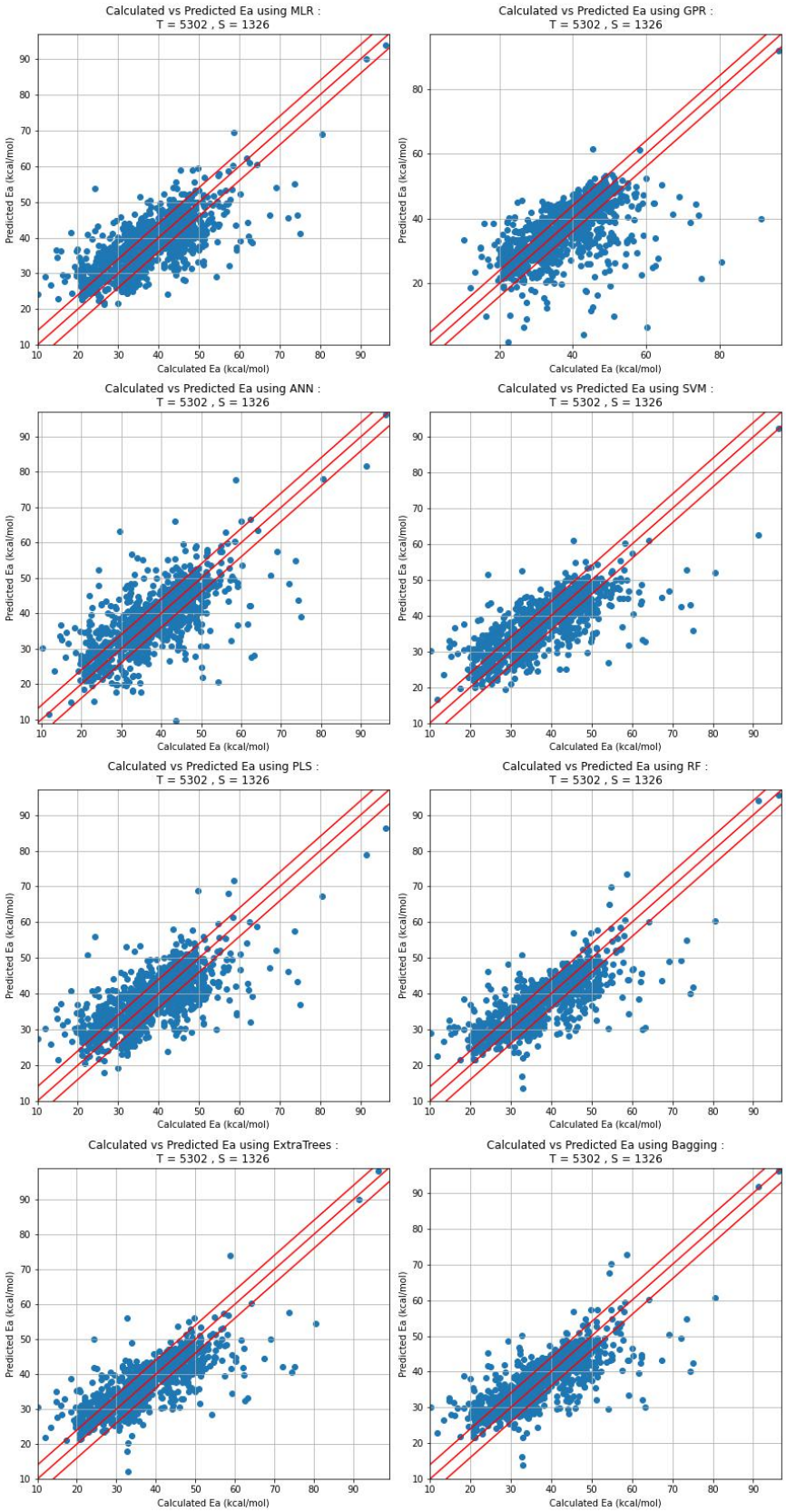


Fig. S28 Machine learning model predictions for the PYR_set_TSSig dataset.

5.4.2 Descriptors trimming based on permutation and feature importance

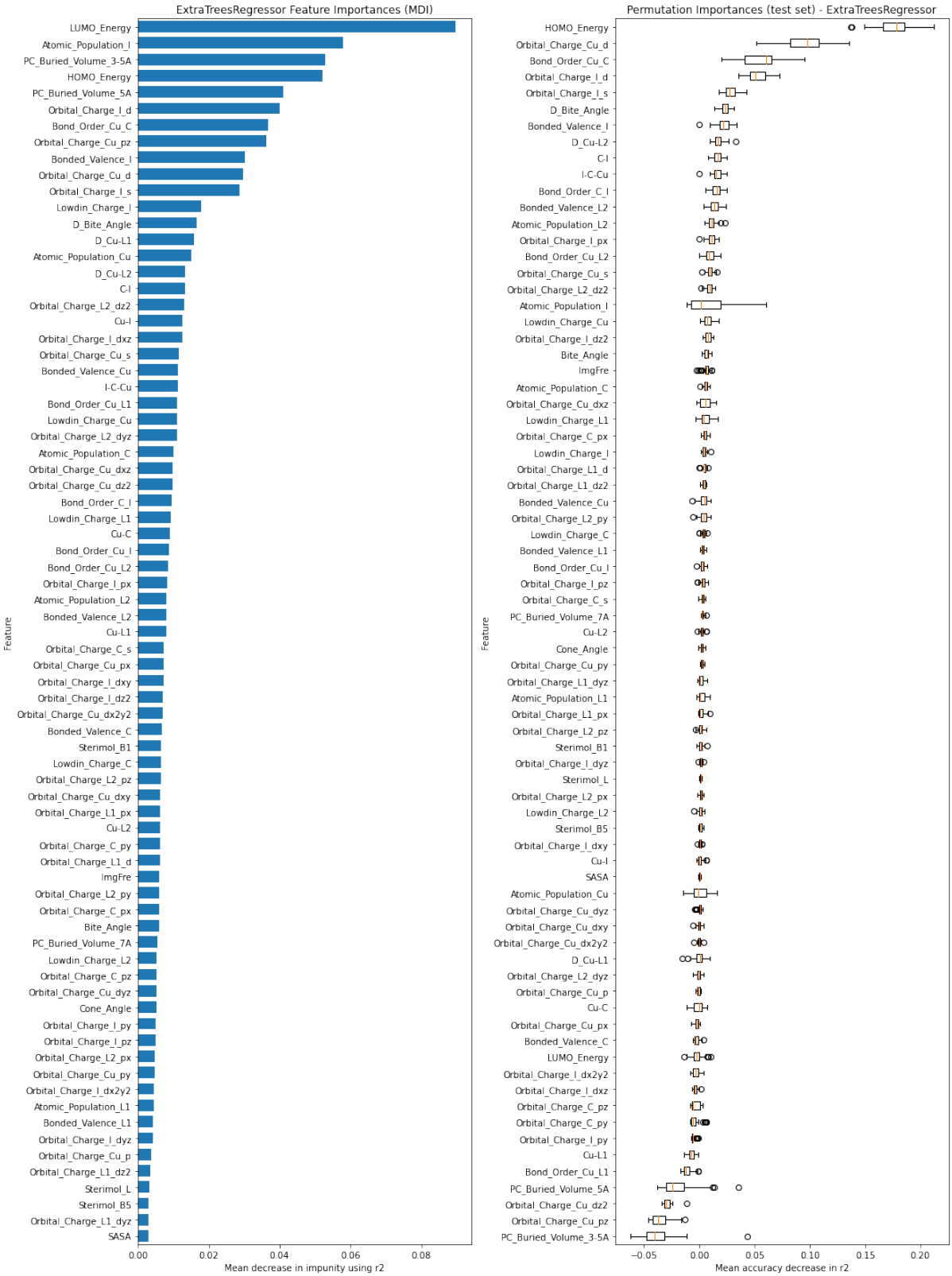


Fig. S29 Permutation importance of the initial ExtraTrees model of the ***PIP_set_TSOA*** dataset.

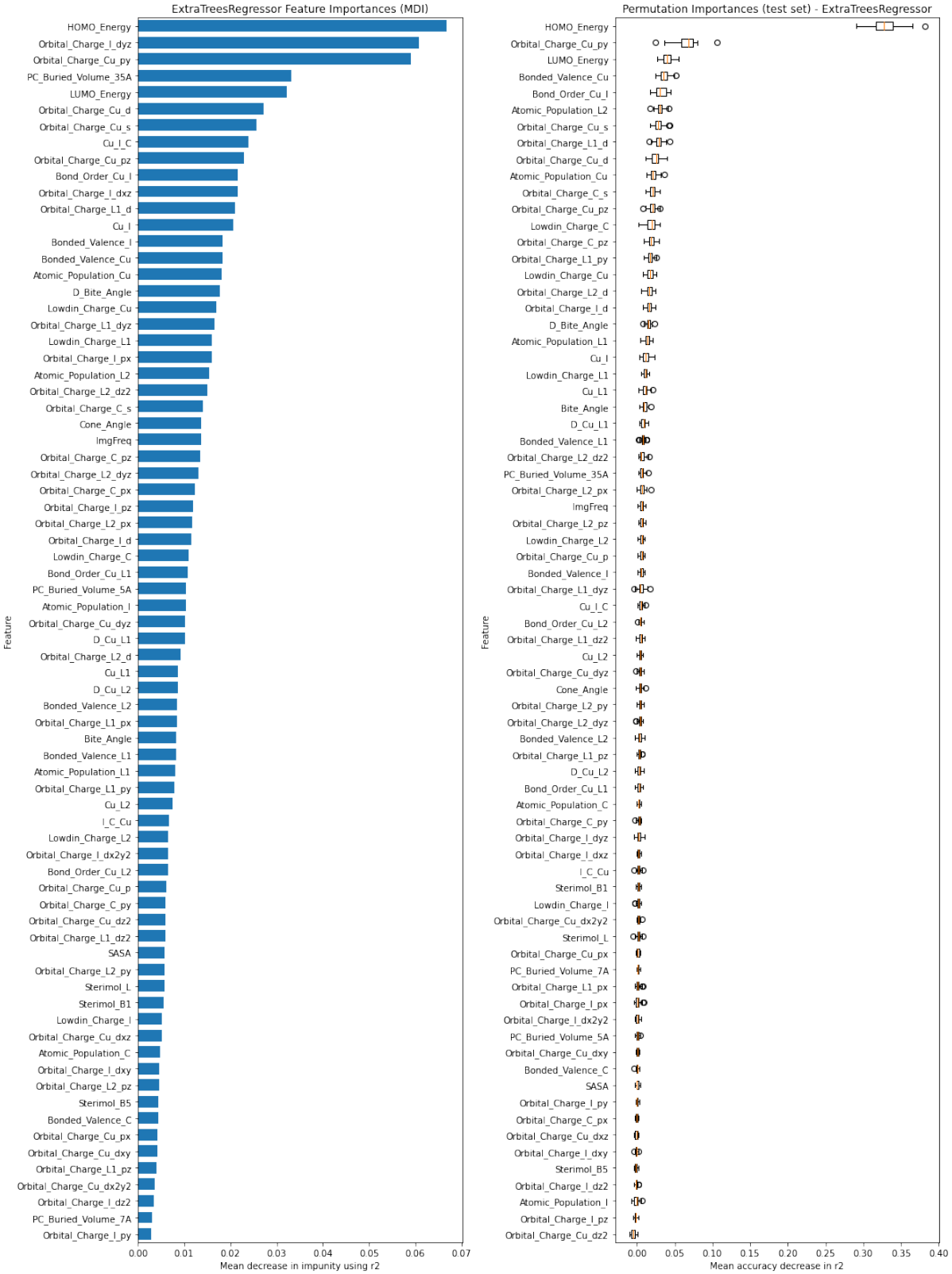


Fig. S30 Permutation importance of the initial ExtraTrees model of the ***PYR_set_TSOA*** dataset.

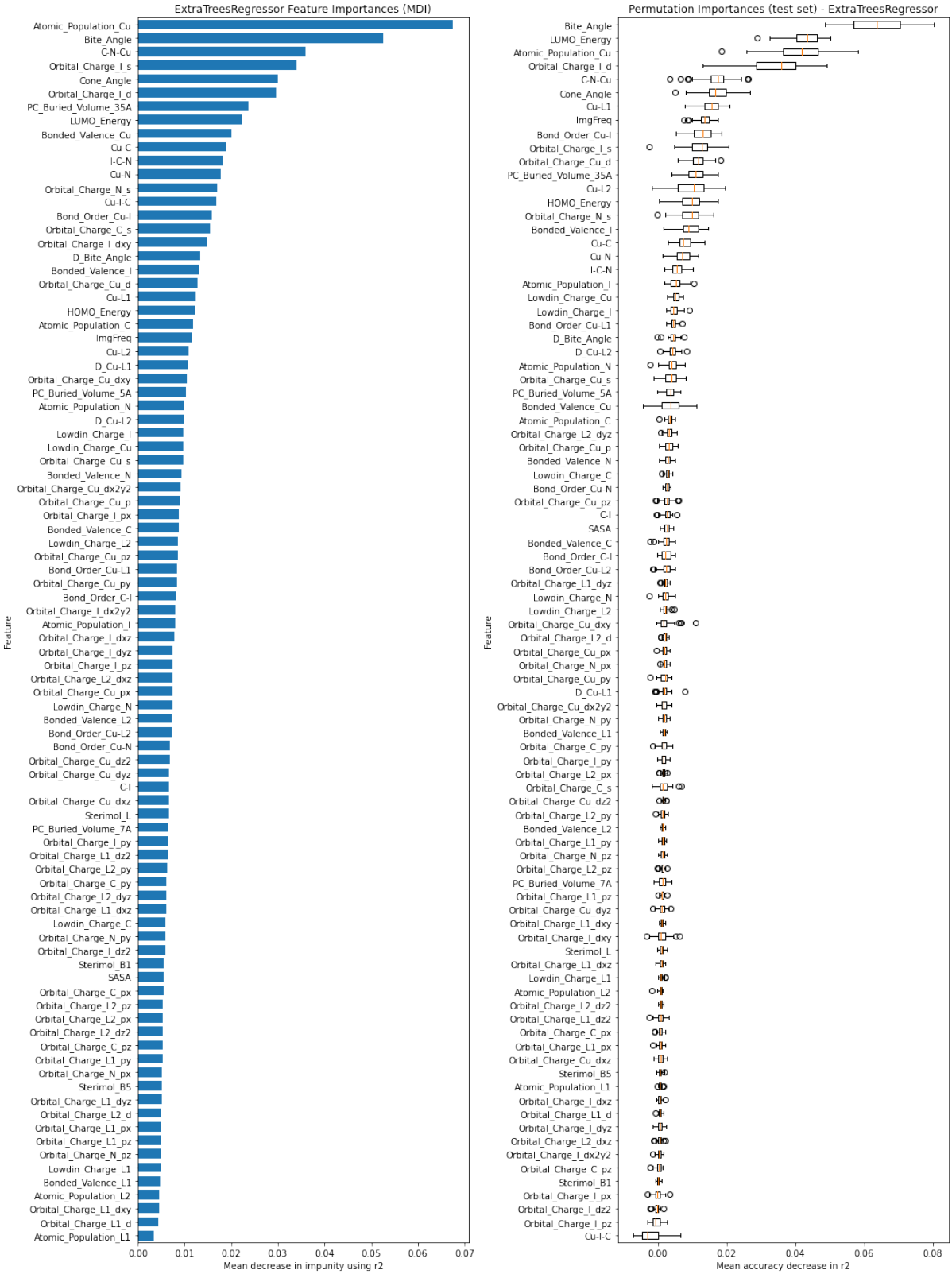


Fig. S31 Permutation importance of the initial ExtraTrees model of the ***PIP_set_TSSig*** dataset.

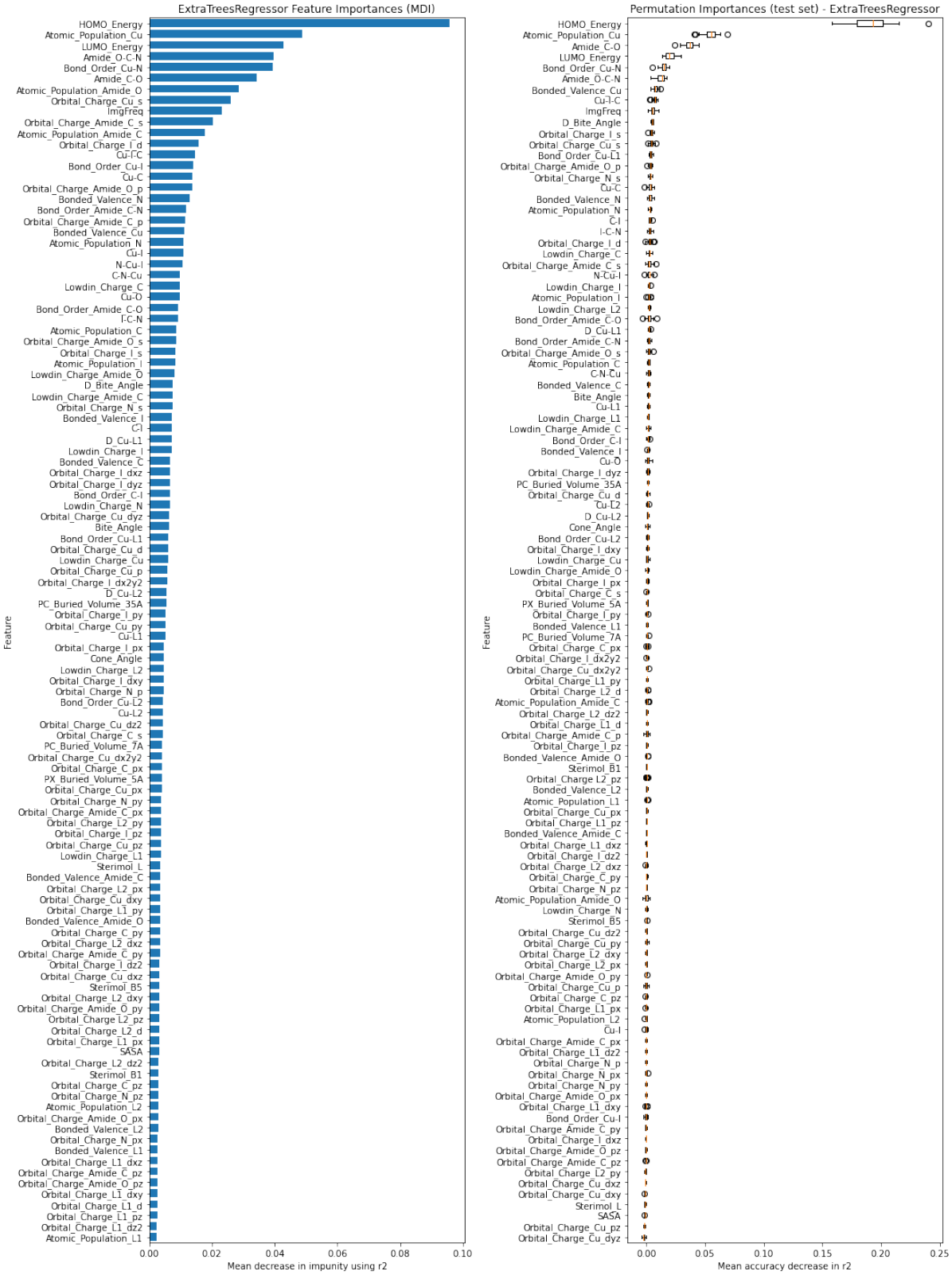


Fig. S32 Permutation importance of the initial ExtraTrees model of the ***PYR_set_TSSig*** dataset.

5.4.2.1 List of Trimmed Descriptors

PIP_set_TSOA: Bite_Angle, D_Bite_Angle, Cone_Angle, Sterimol_B1, HOMO_Energy, LUMO_Energy, Lowdin_Charge_Cu, Lowdin_Charge_C, Bonded_Valence_I, Bonded_Valence_L1, Atomic_Population_C, Atomic_Population_L2, Bond_Order_Cu_L2, Orbital_Charge_C_pz, Orbital_Charge_I_d, Orbital_Charge_Cu_s, Orbital_Charge_Cu_p, Orbital_Charge_Cu_d, Orbital_Charge_L2_px, Orbital_Charge_L2_dz2.

PYR_set_TSOA: Bite_Angle, D_Bite_Angle, Cone_Angle, Sterimol_B1, HOMO_Energy, LUMO_Energy, Lowdin_Charge_Cu, Lowdin_Charge_C, Bonded_Valence_I, Bonded_Valence_L1, Atomic_Population_C, Atomic_Population_L2, Bond_Order_Cu_L2, Orbital_Charge_C_pz, Orbital_Charge_I_d, Orbital_Charge_Cu_s, Orbital_Charge_Cu_p, Orbital_Charge_Cu_d, Orbital_Charge_L1_pz, Orbital_Charge_L1_px, Orbital_Charge_L1_py, Orbital_Charge_L1_d, Orbital_Charge_L2_px, Orbital_Charge_L2_dz2.

PIP_set_TSSig: Bite_Angle, D_Bite_Angle, Cone_Angle, PC_Buried_Volume_35A, PC_Buried_Volume_7A, HOMO_Energy, LUMO_Energy, Cu-L1, Cu-N, I-C-N, C-N-Cu, Lowdin_Charge_Cu, Lowdin_Charge_N, Lowdin_Charge_C, Lowdin_Charge_I, Bonded_Valence_I, Bonded_Valence_L1, Atomic_Population_Cu, Atomic_Population_I, Bond_Order_Cu-I, Bond_Order_C-I, Bond_Order_Cu-N, Orbital_Charge_N_s, Orbital_Charge_N_pz, Orbital_Charge_N_px, Orbital_Charge_N_py, Orbital_Charge_I_s, Orbital_Charge_I_px, Orbital_Charge_Cu_py, Orbital_Charge_Cu_d, Orbital_Charge_L1_pz, Orbital_Charge_L1_dyz, Orbital_Charge_L2_pz, Orbital_Charge_L2_px, Orbital_Charge_L2_d, ImgFreq.

PYR_set_TSSig: Bite_Angle, D_Bite_Angle, PC_Buried_Volume_7A, HOMO_Energy, LUMO_Energy, Cu-L1, Cu-L2, D_Cu-L1, D_Cu-L2, C-I, N-Cu-I, Cu-I-C, Lowdin_Charge_I, Lowdin_Charge_L1, Lowdin_Charge_L2, Bonded_Valence_Cu, Bonded_Valence_N, Bonded_Valence_I, Atomic_Population_Cu, Atomic_Population_N, Atomic_Population_I, Atomic_Population_Amide_O, Bond_Order_C-I, Bond_Order_Cu-N, Bond_Order_Cu-L1, Bond_Order_Amide_C-N, Orbital_Charge_N_s, Orbital_Charge_I_pz, Orbital_Charge_I_py, Orbital_Charge_I_dxy, Orbital_Charge_I_dx2y2, Orbital_Charge_Cu_s, Orbital_Charge_Cu_d, Orbital_Charge_L1_pz, Orbital_Charge_L2_d, Orbital_Charge_L2_dxz, Orbital_Charge_Amide_C_s, Orbital_Charge_Amide_O_s, ImgFreq, Amide_C-O, Amide_O-C-N.

5.4.3 Models with Trimmed Descriptors and Optimised Hyperparameters

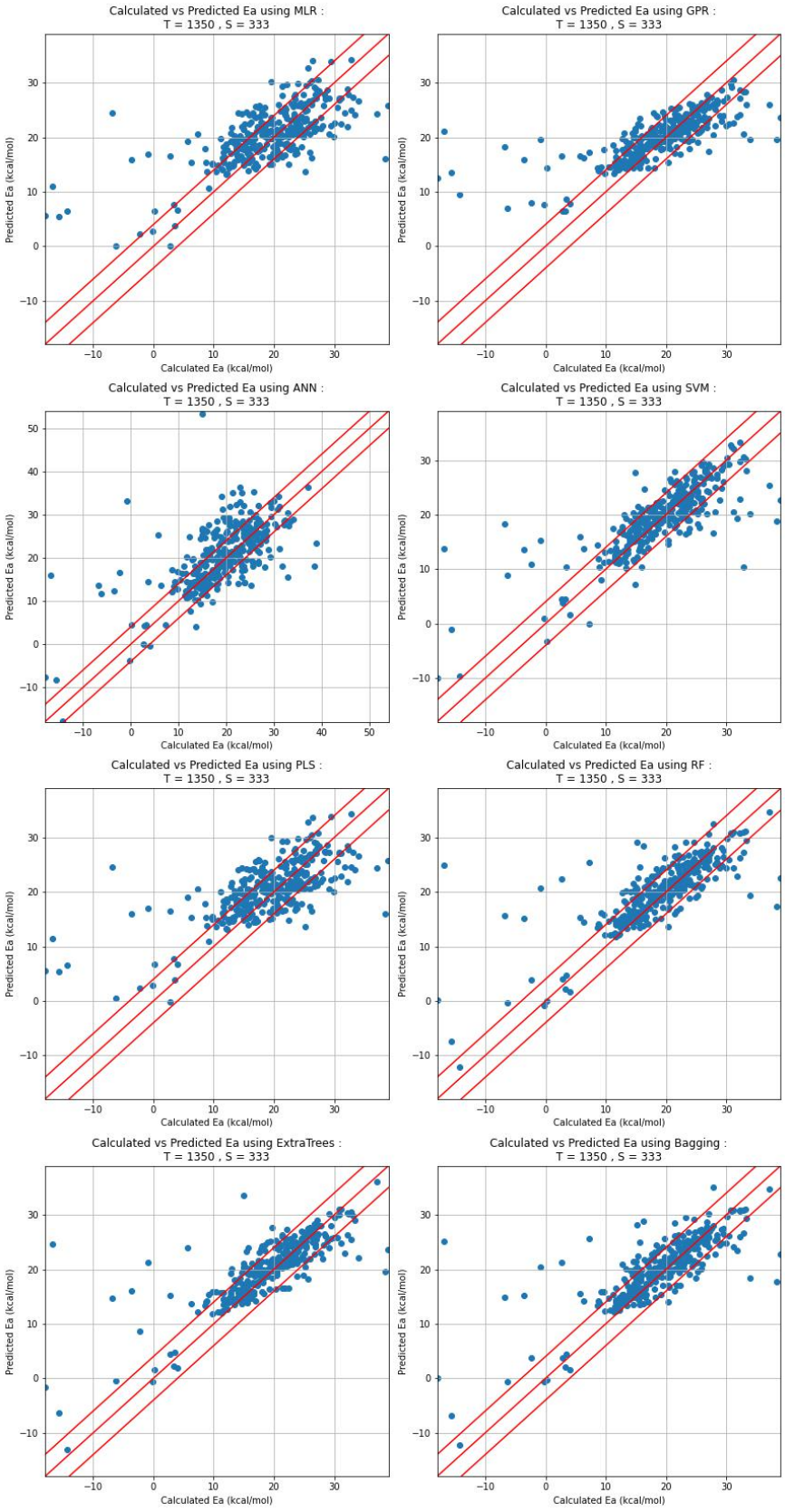


Fig. S33 Machine learning models with trimmed descriptors and optimised hyperparameters for the ***PIP_set_TSOA*** dataset.

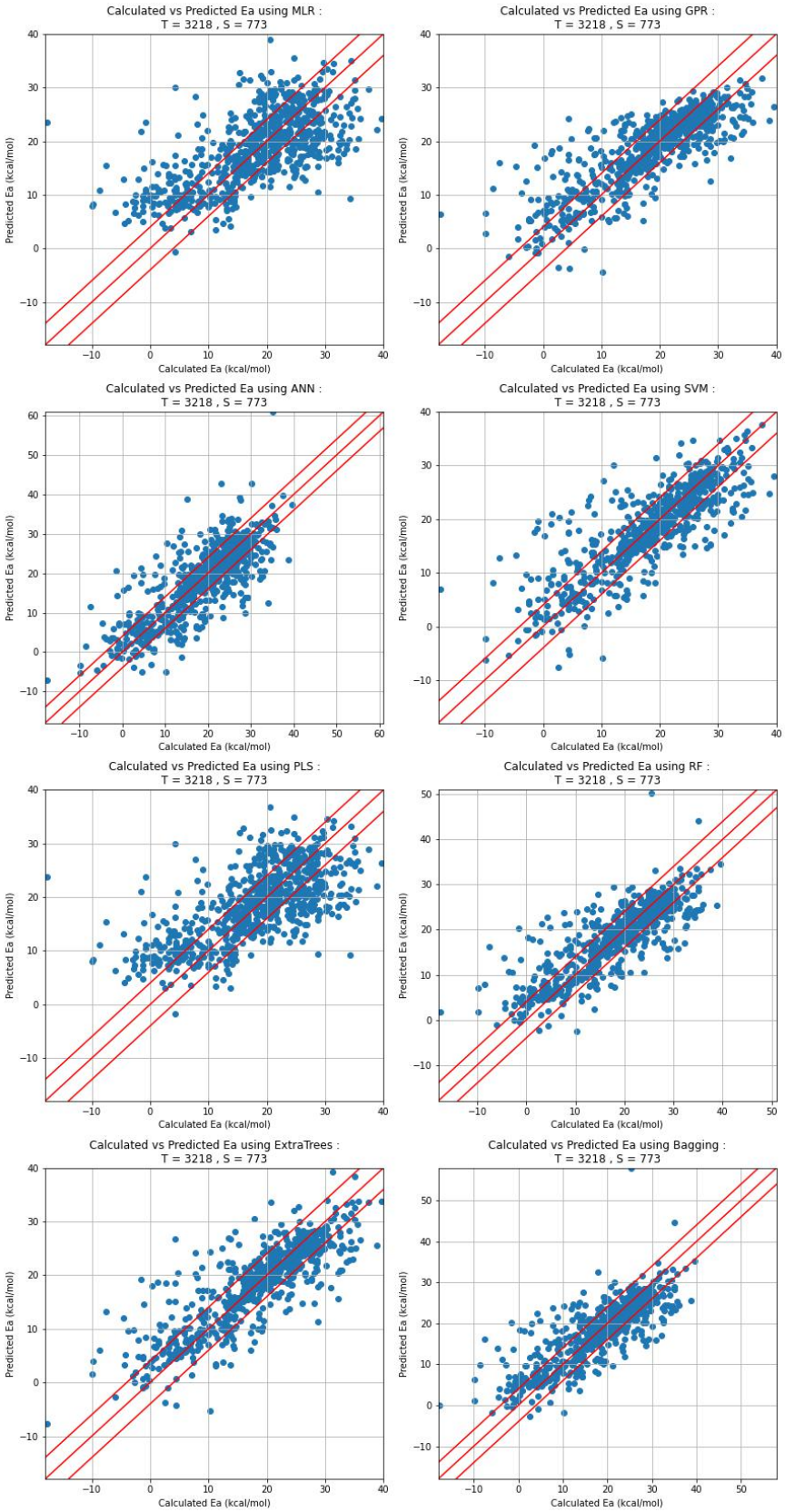


Fig. S34 Machine learning models with trimmed descriptors and optimised hyperparameters for the ***PYR_set_TSOA*** dataset.

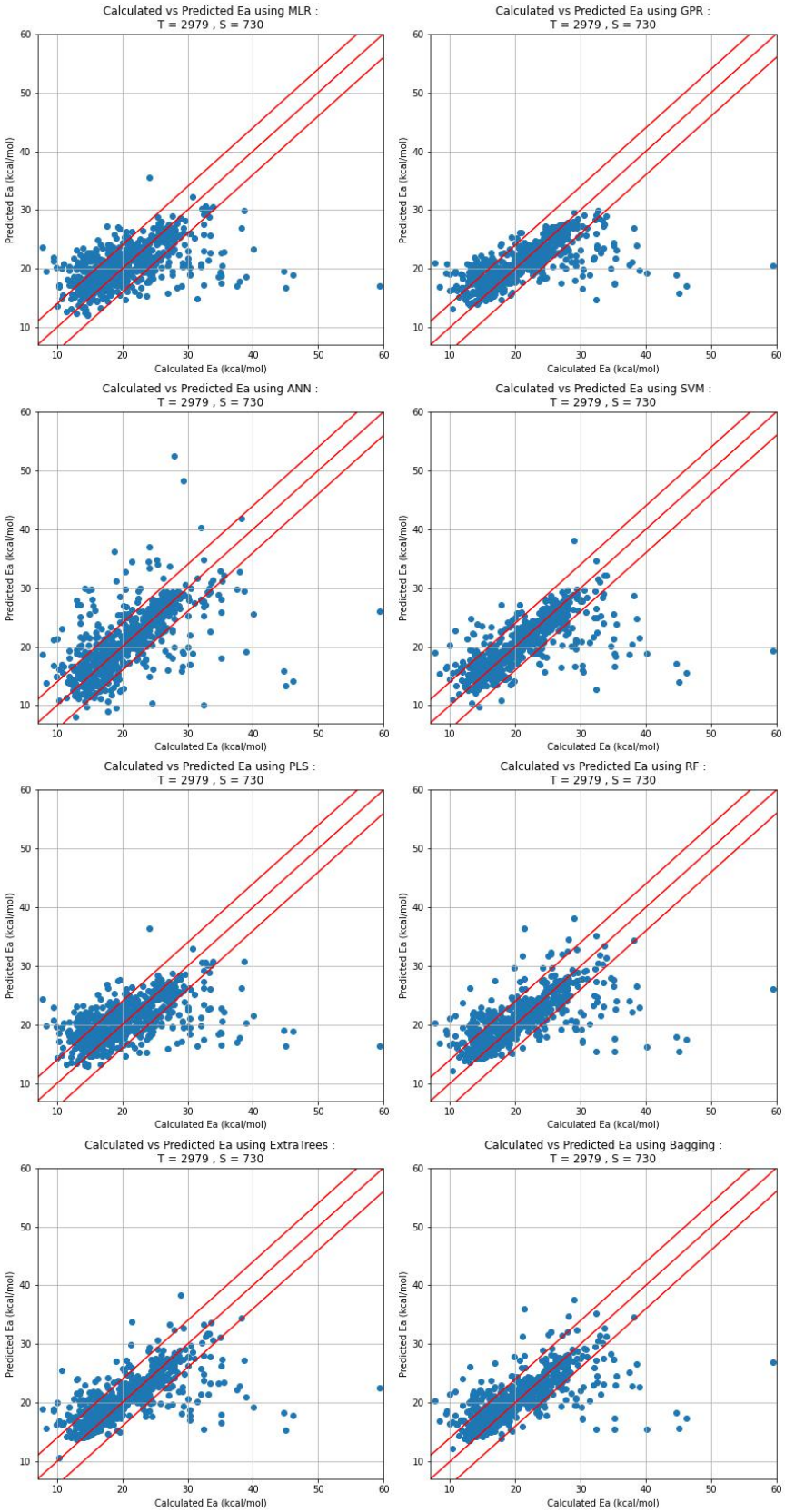


Fig. S35 Machine learning models with trimmed descriptors and optimised hyperparameters for the ***PIP_set_TSSig*** dataset.

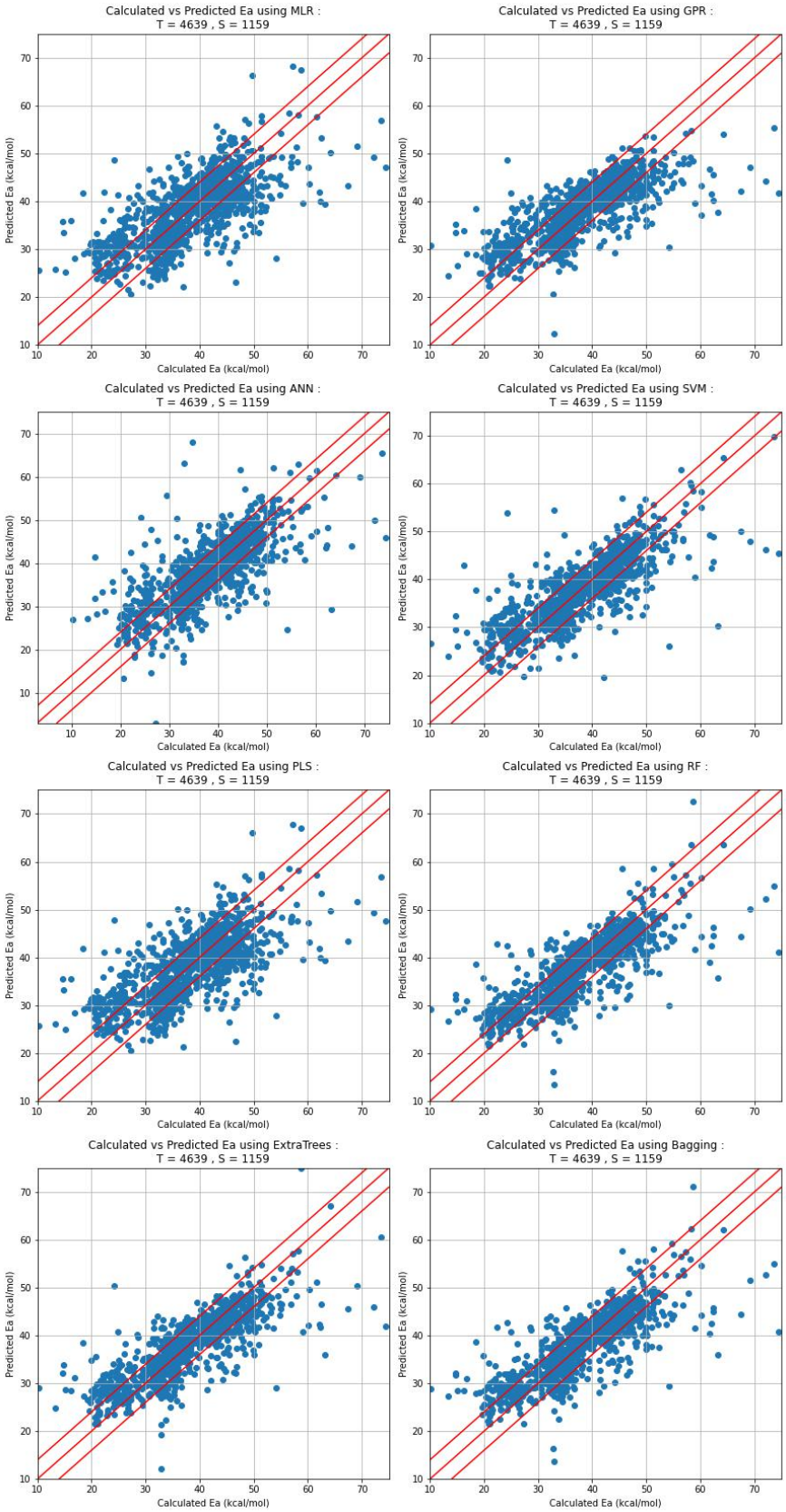


Fig. S36 Machine learning models with trimmed descriptors and optimised hyperparameters for the ***PYR_set_TSSig*** dataset.

Table S16 Machine learning metrics with trimmed descriptors and optimised hyperparameters for the ***PIP_set_TSOA*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.49	5.85	59.5
GPR	0.57	5.65	73.9
ANN	0.43	6.54	61.6
SVM	0.64	4.81	78.7
PLS	0.49	5.86	60.1
RF	0.62	5.01	75.7
ExtraTrees	0.66	4.81	79.6
Bagging	0.63	5.00	76.6

Table S17 Machine learning metrics with trimmed descriptors and optimised hyperparameters for the ***PYR_set_TSOA*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.44	6.81	50.6
GPR	0.68	5.17	68.0
ANN	0.64	5.63	66.1
SVM	0.68	5.09	71.7
PLS	0.44	6.77	51.2
RF	0.69	5.06	70.2
ExtraTrees	0.71	4.86	71.3
Bagging	0.68	5.10	70.2

Table S18 Machine learning metrics with trimmed descriptors and optimised hyperparameters for the ***PIP_set_TSSig*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.32	4.97	73.7
GPR	0.45	4.49	79.7
ANN	0.41	4.95	78.1
SVM	0.47	4.41	84.7
PLS	0.32	4.98	73.8
RF	0.46	4.44	80.7
ExtraTrees	0.48	4.33	81.5
Bagging	0.47	4.39	81.0

Table S19 Machine learning metrics with trimmed descriptors and optimised hyperparameters for the ***PYR_set_TSSig*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.48	6.16	55.9
GPR	0.60	5.39	66.3
ANN	0.55	5.89	65.2
SVM	0.64	5.10	71.0
PLS	0.48	6.17	55.8
RF	0.65	4.99	70.6
ExtraTrees	0.66	4.95	70.6
Bagging	0.66	4.97	70.8

5.4.4 Permutation Importance of Trimmed Models

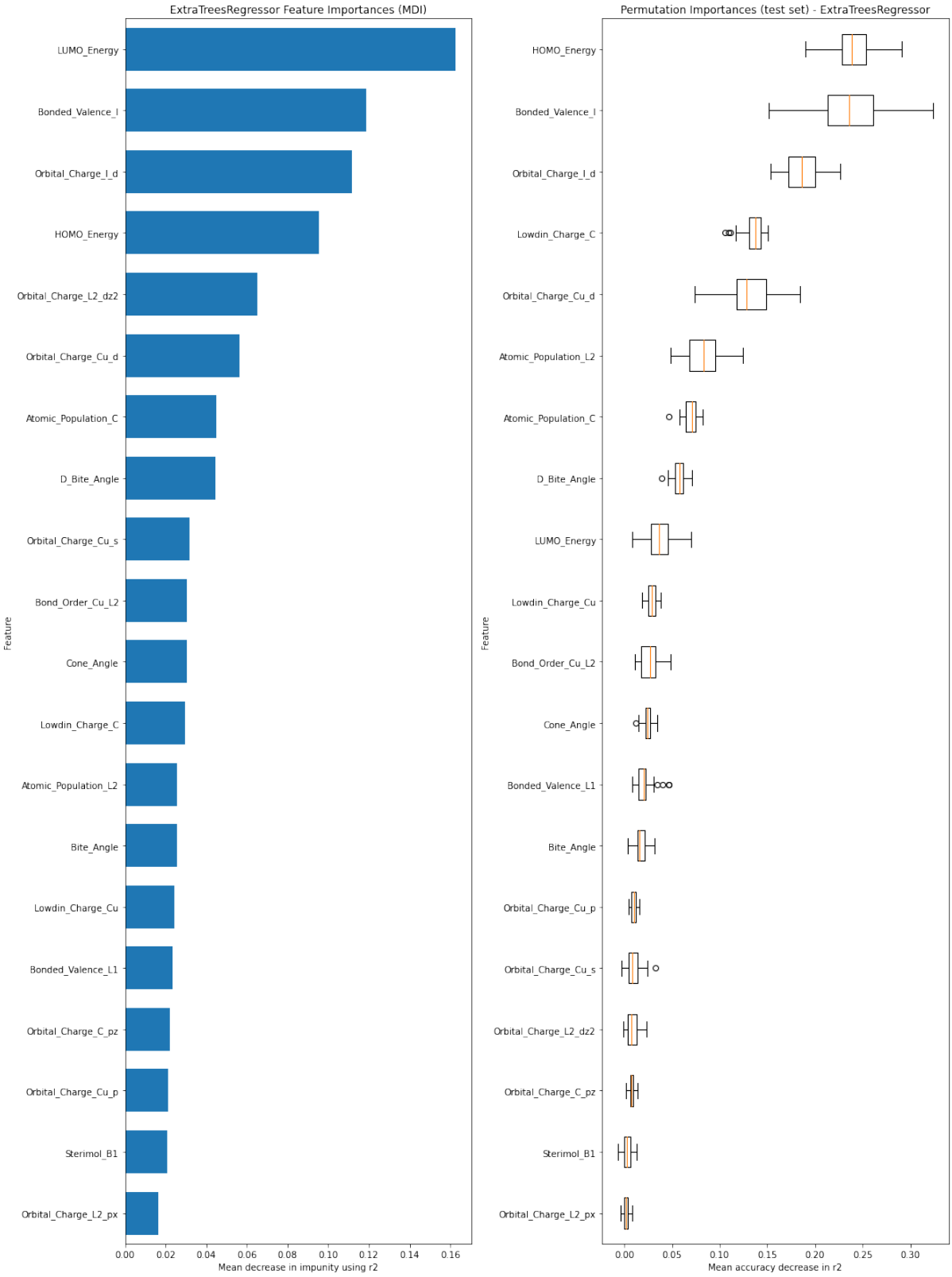


Fig. S37 Permutation importance of the initial trimmed model of the ***PIP_set_TSOA*** dataset.

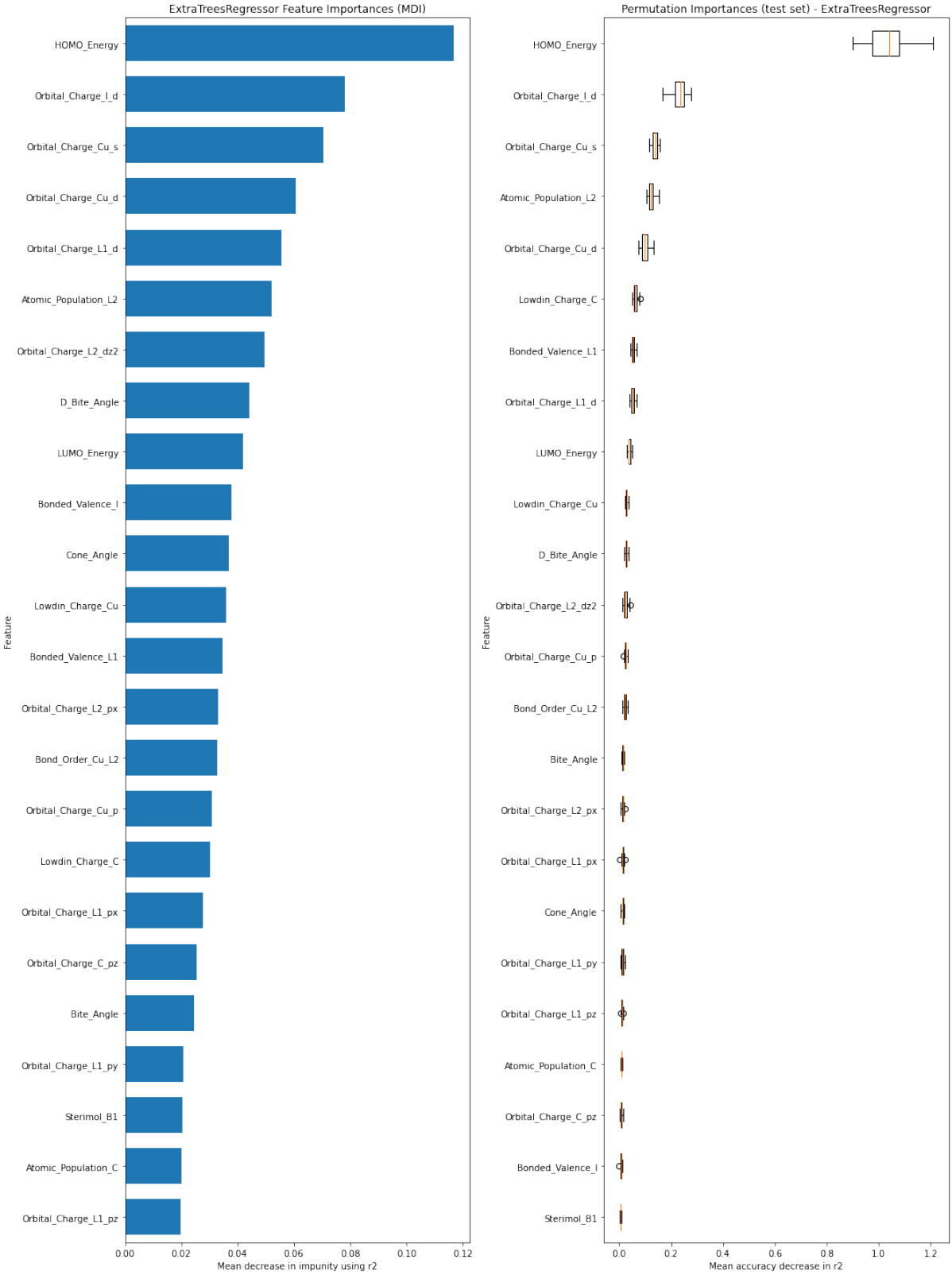


Fig. S38 Permutation importance of the trimmed ExtraTrees model of the ***PYR_set_TSOA*** dataset.

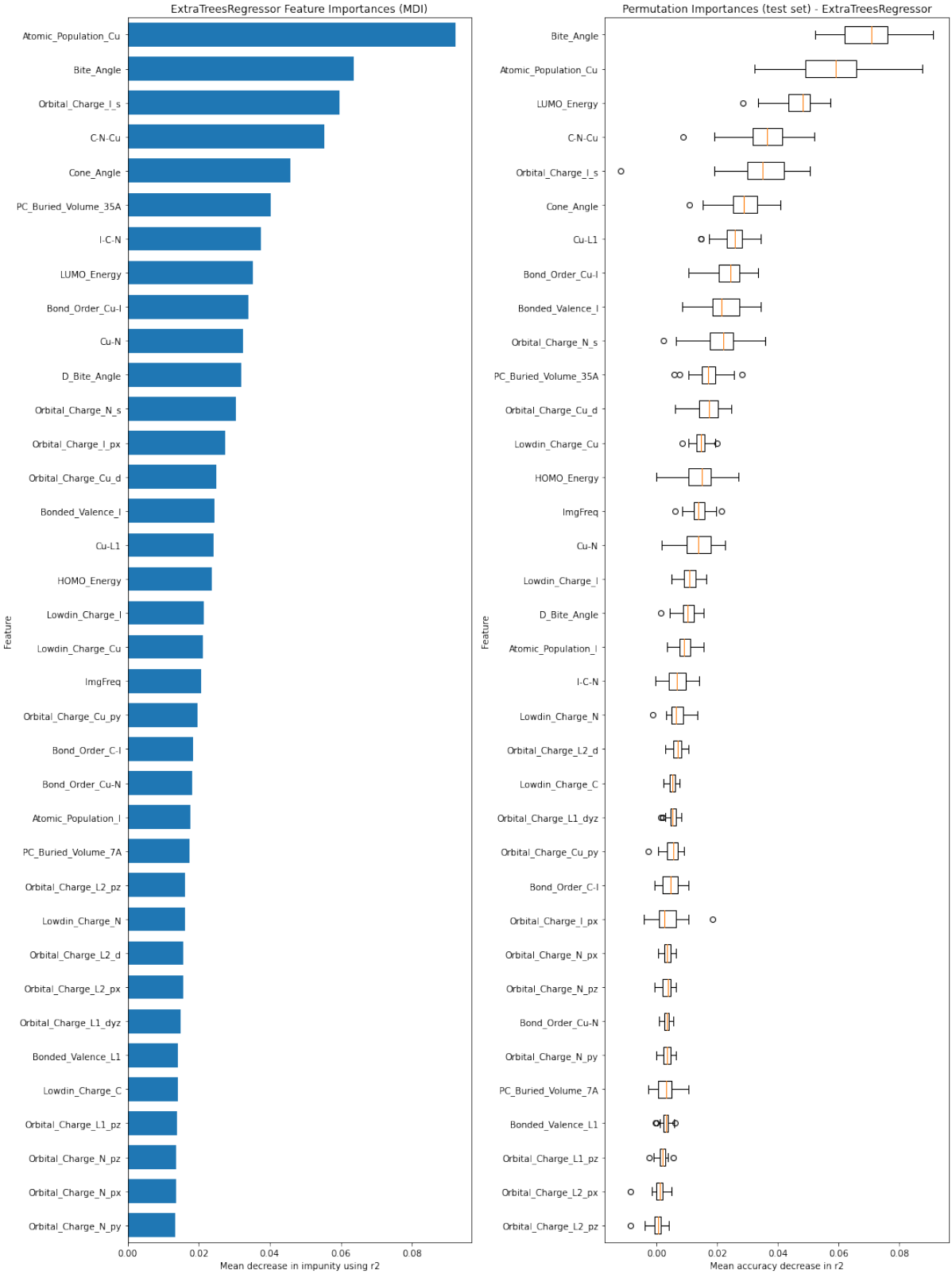


Fig. S39 Permutation importance of the trimmed ExtraTrees model of the ***PIP_set_TSSig*** dataset.

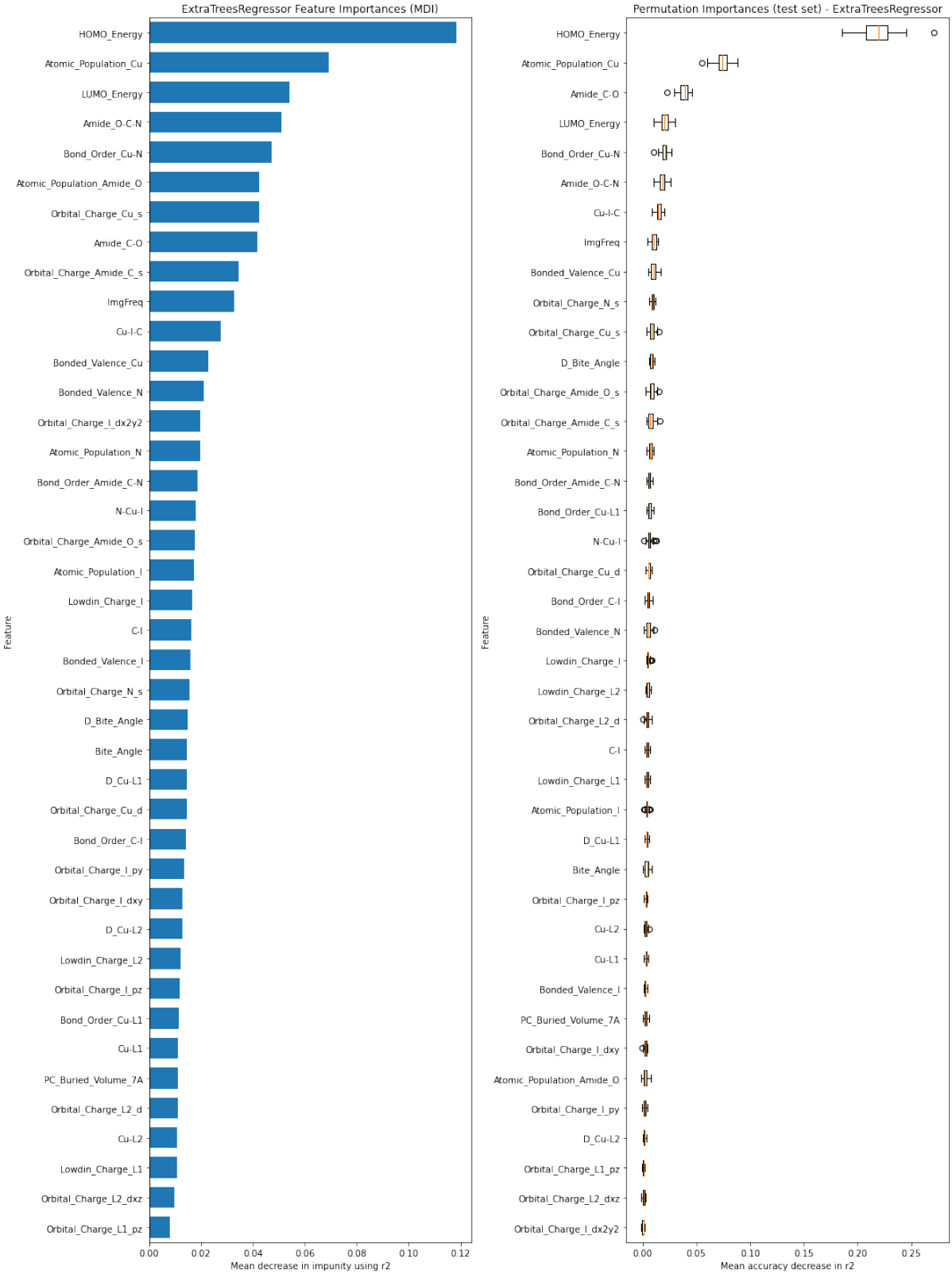


Fig. S40 Permutation importance of the trimmed ExtraTrees model of the ***PYR_set_TSSig*** dataset.

Table S20 Ten most important descriptors with each dataset based on permutation importance analysis of ET models

Pip_set_TSOA	***Pyr_set_TSOA***	***Pip_set_TSSig***	***Pyr_set_TSSig***
Orbital_charge_I_d	HOMO_energy	Atomic_population_Cu	HOMO_energy
HOMO_energy	Orbital_charge_Cu_d	Bite_angle	Atomic_population_Cu
Orbital_charge_Cu_d	Orbital_charge_Cu_s	Orbital_charge_N_s	N-Cu-I_angle
Lowdin_charge_C	Orbital_charge_L1_px	LUMO_energy	Atomic_population_-amide_O
D_bite_angle	LUMO_energy	Orbital_charge_I_s	Atomic_population_N
LUMO_energy	Orbital_charge_L2_px	C-N-Cu_angle	Orbital_charge_Cu_s
Bond_order_Cu_L2	Atomic_population_L2	Orbital_charge_Cu_d	Bond_order_Cu-N
Lowdin_charge_Cu	D_bite_angle	HOMO_energy	Cu-I-C_angle
Bonded_valence_I	Lowdin_charge_Cu	Cu-L1_distance	Orbital_charge_amide_-C_s
Orbital_charge_C_pz	Orbital_charge_Cu_p	Cone_angle	Bond_order_Cu-I

5.5 ML models building without transition state descriptors

5.5.1 Metrics of initial ML models without transition state descriptors

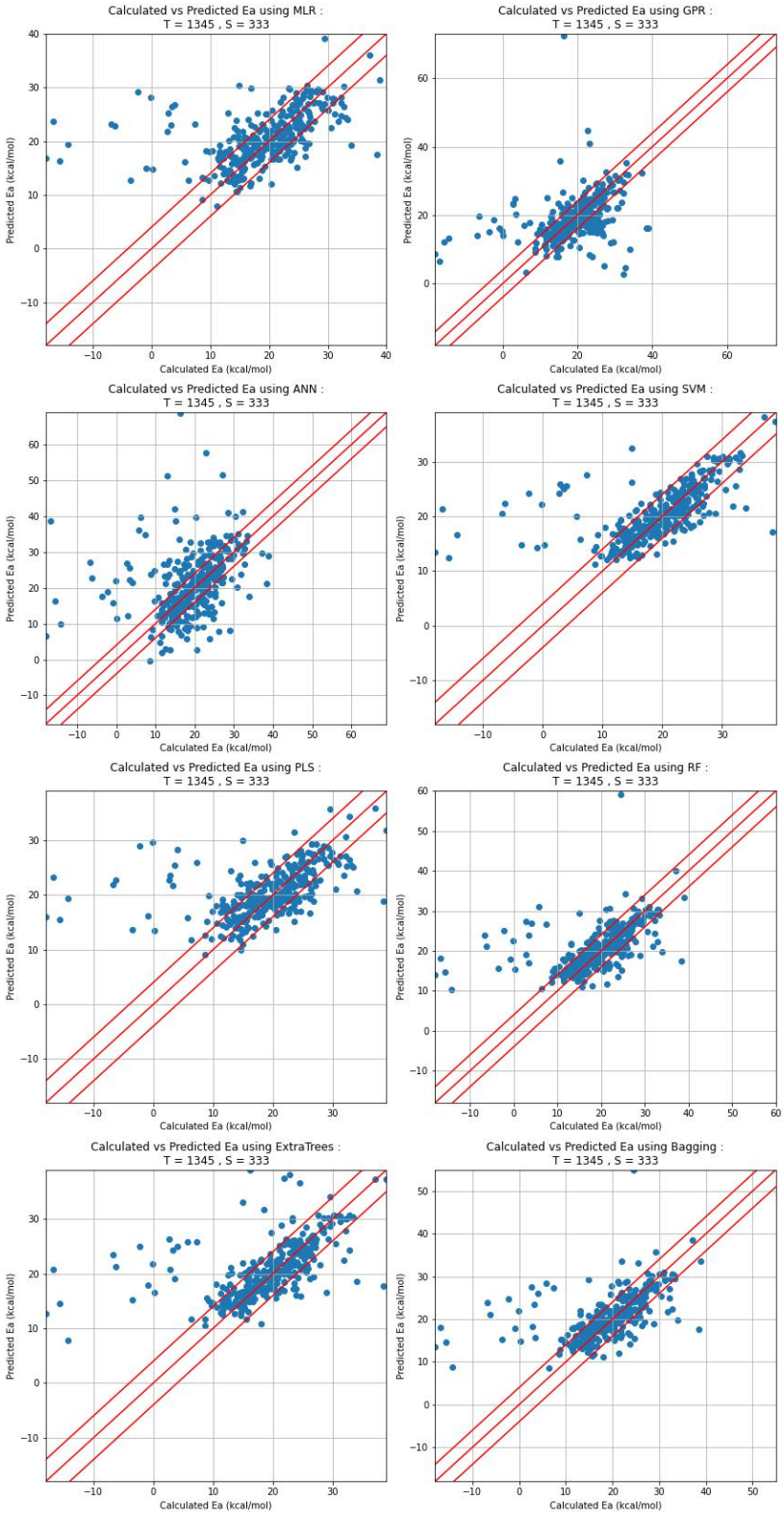


Fig. S41 Machine learning model predictions for the PIP_set_TSOA_NoTS dataset with optimized parameters.

Table S21 Performance metrics for the machine learning models for the PIP_set_TSOA_NoTS dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.22	7.24	66.7
GPR	0.15	8.07	64.3
ANN	0.08	10.59	46.6
SVM	0.35	6.56	79.6
PLS	0.23	7.13	65.2
RF	0.28	7.01	73.0
ExtraTrees	0.32	6.81	76.0
Bagging	0.28	6.96	73.6

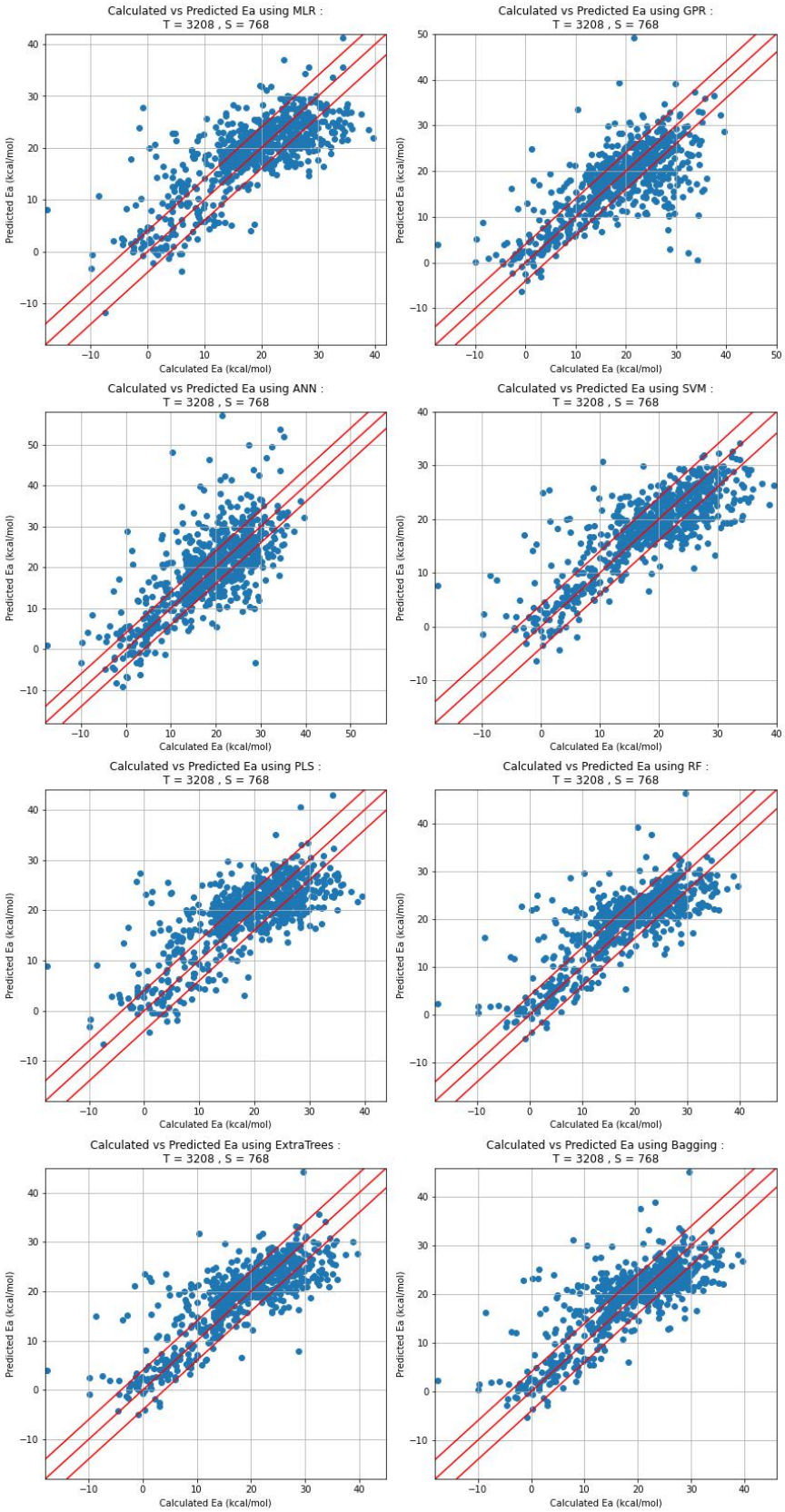


Fig. S42 Machine learning model predictions for the PYR_set_TSOA_NoTS dataset with optimized parameters.

Table S22 Performance metrics for the machine learning models for the PYR_set_TSOA_NoTS dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.59	5.86	54.8
GPR	0.55	6.17	62.6
ANN	0.53	6.98	55.1
SVM	0.67	5.16	68.1
PLS	0.56	6.03	54.2
RF	0.66	5.36	67.1
ExtraTrees	0.68	5.18	67.7
Bagging	0.65	5.40	66.8

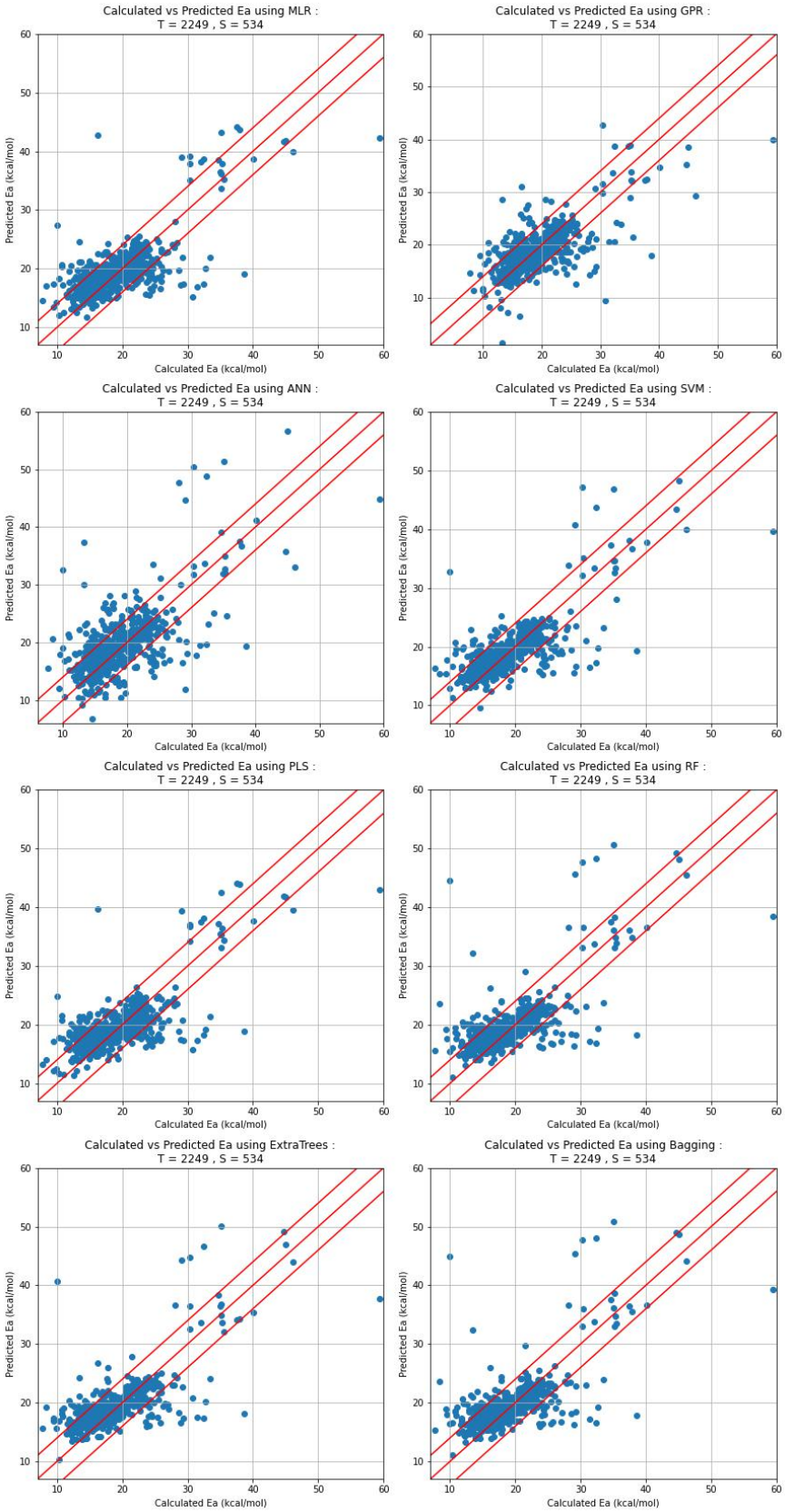


Fig. S43 Machine learning model predictions for the PIP_set_TSSig_NoTS dataset with optimized parameters.

Table S23 Performance metrics for the machine learning models for the PIP_set_TSSig_NoTS dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.53	3.97	79.8
GPR	0.48	4.14	78.8
ANN	0.43	4.74	74.5
SVM	0.57	3.74	83.7
PLS	0.54	3.87	79.0
RF	0.50	4.16	83.3
ExtraTrees	0.54	3.94	82.6
Bagging	0.50	4.16	83.2

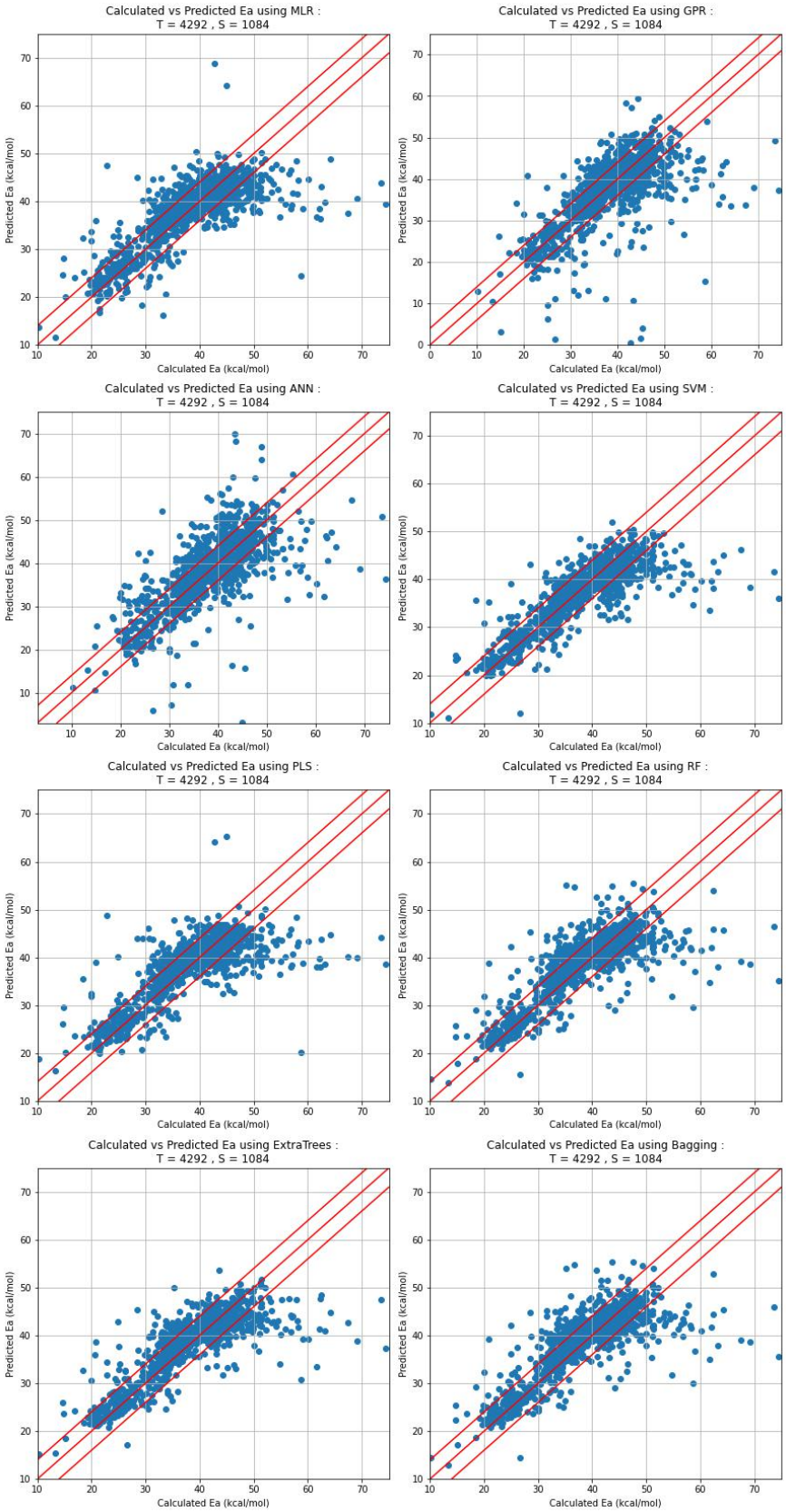


Fig. S44 Machine learning model predictions for the PYR_set_TSSig_NoTS dataset with optimized parameters.

Table S24 Performance metrics for the machine learning models for the PYR_set_TSSig_NoTS dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.57	5.45	66.5
GPR	0.47	6.65	64.0
ANN	0.51	6.31	61.3
SVM	0.66	4.94	75.6
PLS	0.56	5.54	68.1
RF	0.62	5.12	73.2
ExtraTrees	0.66	4.83	75.8
Bagging	0.62	5.14	73.4

5.5.2 Transition State Independent Descriptors with Trimmed Descriptors

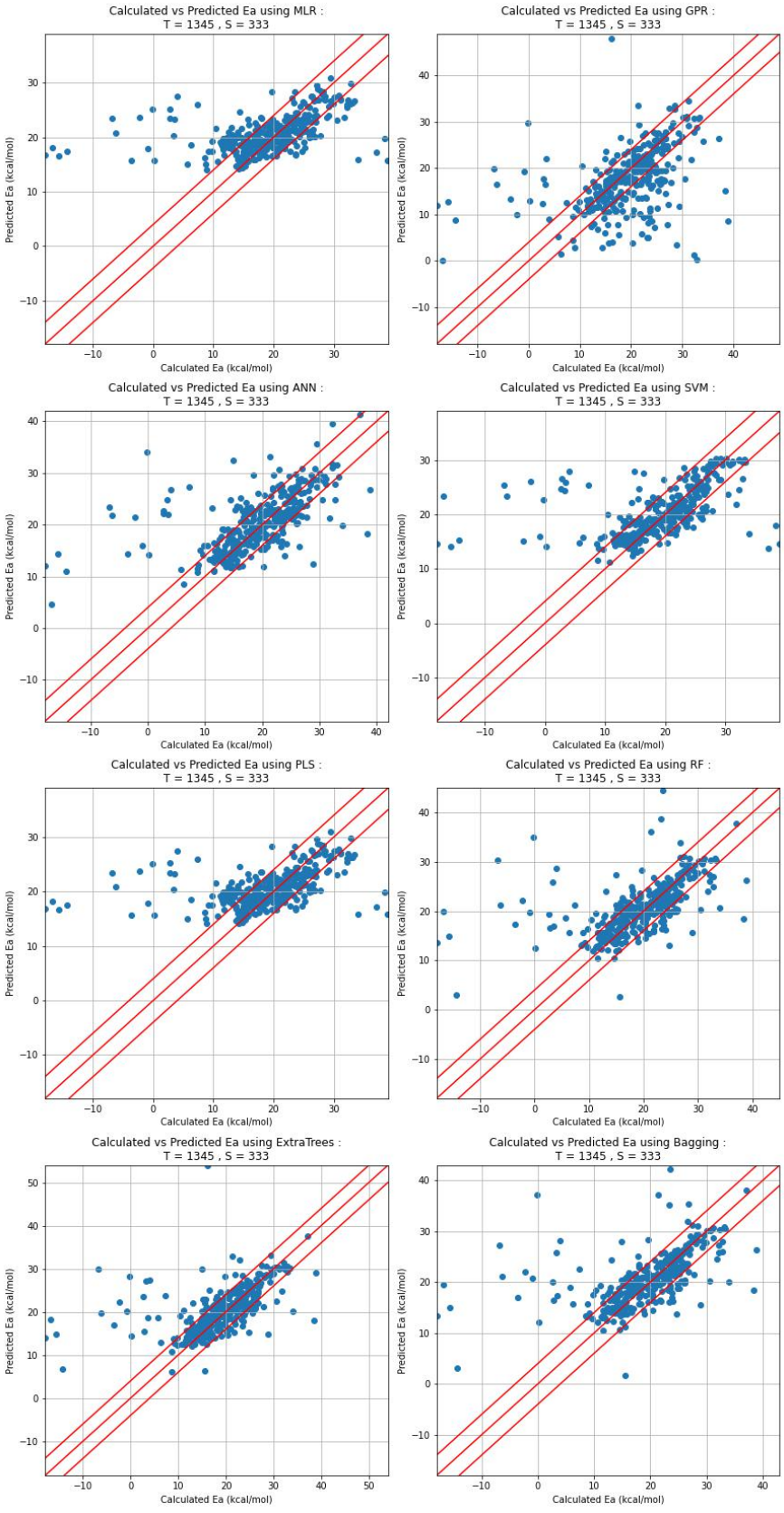


Fig. S45 Machine learning model predictions for the PIP_set_TSOA_NoTS dataset with optimized parameters and trimmed descriptors.

Table S25 Performance metrics for the machine learning models for the PIP_set_TSOA_NoTS dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.22	7.22	65.8
GPR	0.19	8.03	58.3
ANN	0.33	6.73	68.5
SVM	0.24	7.07	75.1
PLS	0.22	7.22	65.8
RF	0.28	6.99	71.2
ExtraTrees	0.29	6.95	76.3
Bagging	0.28	6.95	72.1

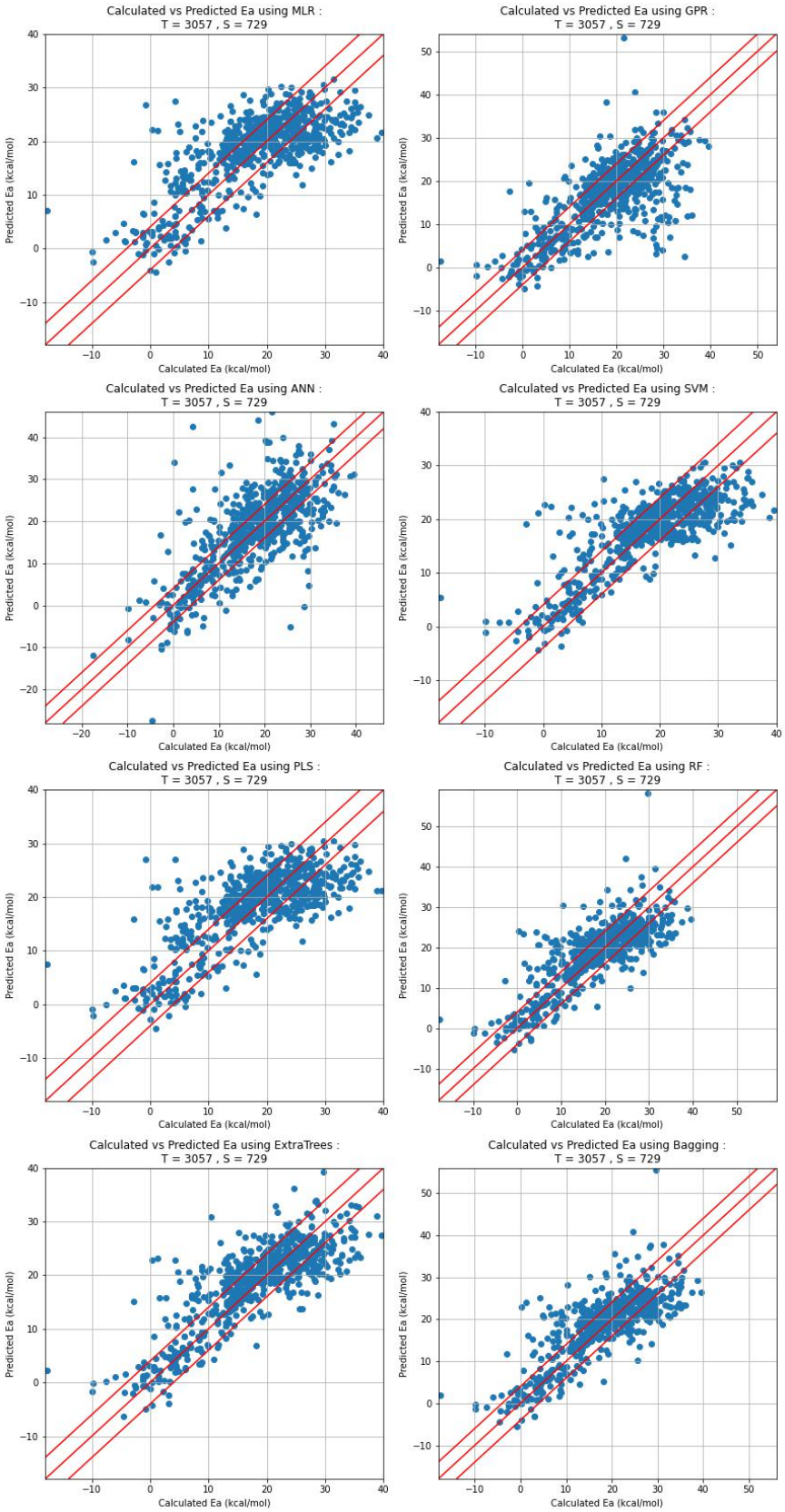


Fig. S46 Machine learning model predictions for the PYR_set_TSOA_NoTS dataset with optimized parameters and trimmed descriptors.

Table S26 Performance metrics for the machine learning models for the PYR_set_TSOA_NoTS dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.55	5.98	52.3
GPR	0.52	6.46	58.2
ANN	0.52	6.80	55.1
SVM	0.65	5.23	66.3
PLS	0.55	5.95	52.1
RF	0.66	5.24	66.0
ExtraTrees	0.69	4.97	66.3
Bagging	0.66	5.25	65.3

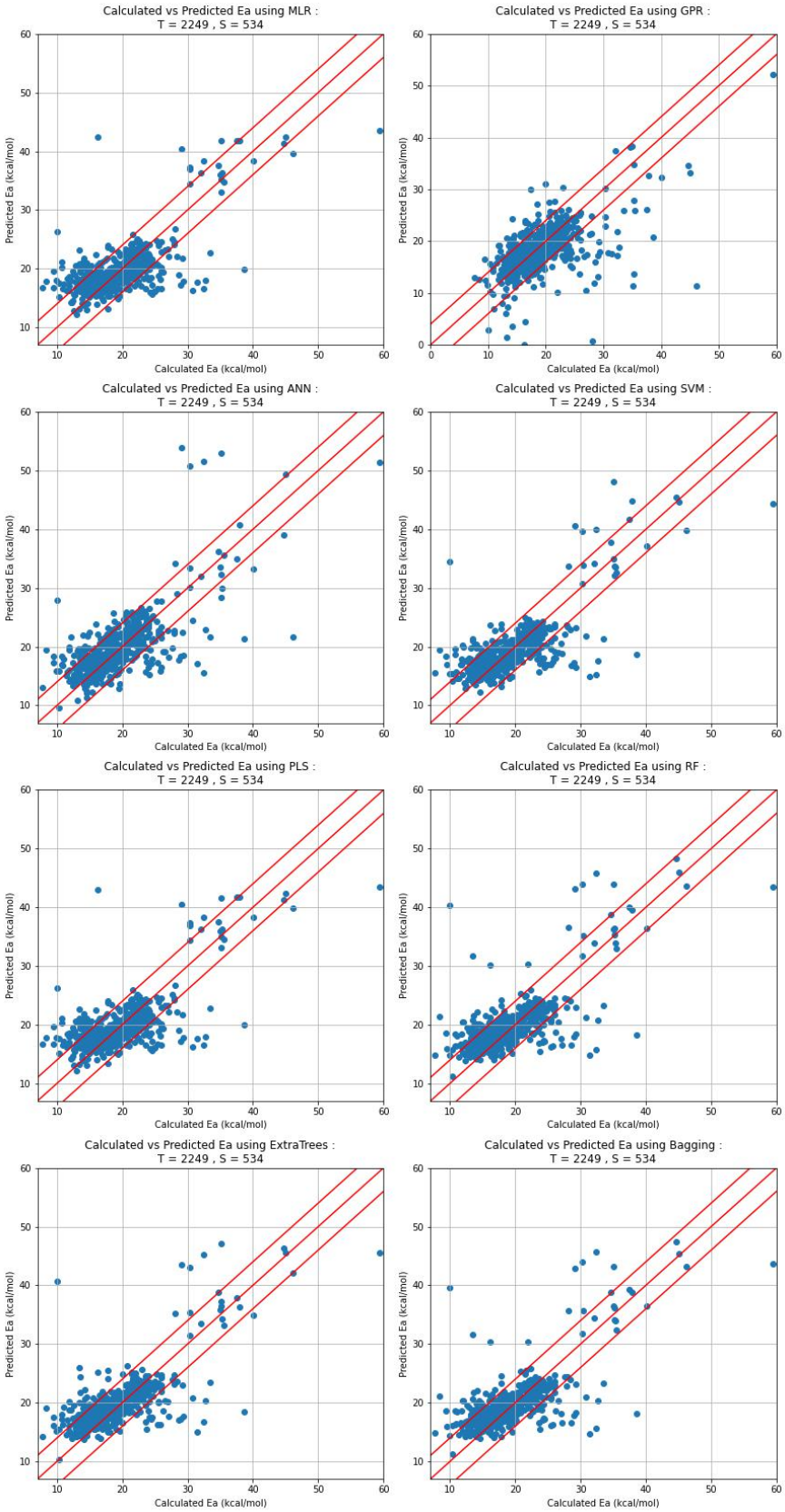


Fig. S47 Machine learning model predictions for the PIP_set_TSSig_NoTS dataset with optimized parameters and trimmed descriptors.

Table S27 Performance metrics for the machine learning models for the PIP_set_TSSig_NoTS dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.50	4.04	77.3
GPR	0.35	4.84	74.7
ANN	0.51	4.18	80.2
SVM	0.56	3.78	82.8
PLS	0.50	4.04	77.2
RF	0.53	4.00	82.0
ExtraTrees	0.56	3.87	83.0
Bagging	0.53	3.98	82.4

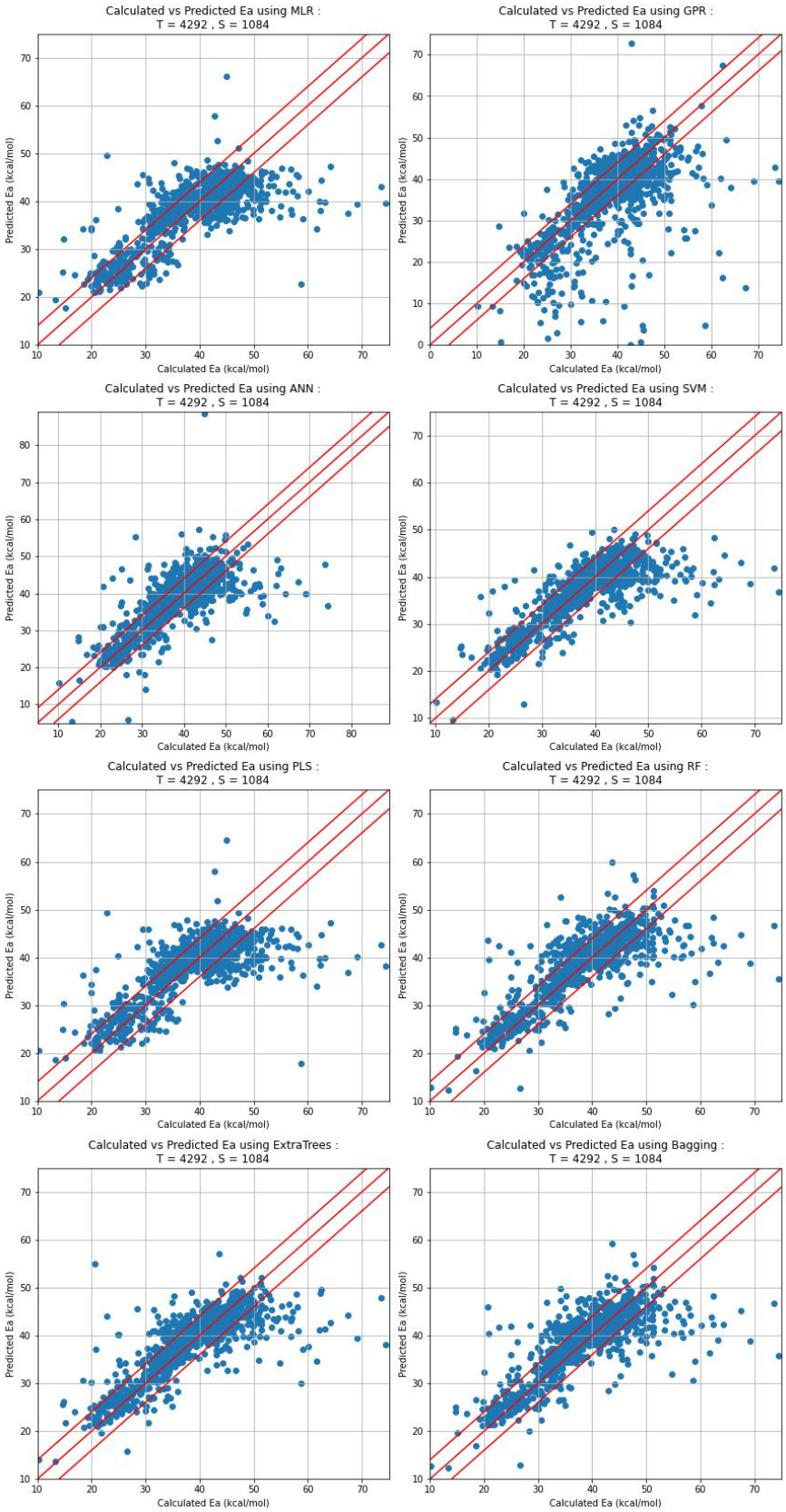


Fig. S48 Machine learning model predictions for the PYR_set_TSSig_NoTS dataset with optimized parameters and trimmed descriptors.

Table S28 Performance metrics for the machine learning models for the PYR_set_TSSig_NoTS dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.55	5.56	65.7
GPR	0.40	7.92	62.2
ANN	0.57	5.59	71.2
SVM	0.62	5.21	73.5
PLS	0.54	5.65	66.2
RF	0.62	5.16	74.4
ExtraTrees	0.64	4.98	75.8
Bagging	0.62	5.16	73.8

5.5.3 Improving models through improved calculations of electronic descriptors

5.5.3.1 Correlation of Electronic Descriptors Between Functional and DLPNO-CCSD(T)

Orbital charges, Lowdin charges, LUMO energies and bond orders on the ligating atoms from all methods correlated well with $\Delta G^{\ddagger}$ calculated with DLPNO-CCSD(T)/def2-TZVPP. A higher percentage of HF exchange has a better correlation with DLPNO-CCSD(T)/def2-TZVPP (TPSS < TPSSh (10%) < PBE0(20%)). This suggests that a large amount of HF exchange is required to correctly describe the bonding between the copper centre and the coupling nitrogen atom. All other descriptors can be sufficiently described without the inclusion of HF exchange. Descriptors calculated at the PBE0/def2-TZVP level of theory provide the most accurate electronic descriptors and activation energies for the same computational cost of the transition state calculations

Table S29 Correlation of electronic descriptors between B97-3c, PBE0, TPSS and TPSSh with DLPNO-CCSD(T)/def2-TZVPP. Red is a low correlation, green is a high correlation.

Method	HOMO Energy (eV)	LUMO Energy (eV)	Lowdin Charge (Cu)	Lowdin Charge (N)	Lowdin Charge (L1)	Lowdin Charge (L2)	Bonded Valence (Cu)	Bonded Valence (N)
B97-3c	0.48	0.85	0.01	0.71	0.78	0.67	0.5	0.21
PBE0	0.75	0.9	0.71	0.91	0.78	0.94	0.54	0.67
TPSS	0.52	0.85	0.5	0.87	0.71	0.91	0.38	0.6
TPSSh	0.63	0.87	0.58	0.89	0.75	0.92	0.46	0.64
Method	Bonded Valence (L1)	Bonded Valence (L2)	Atomic Population (Cu)	Atomic Population (N)	Atomic Population (L1)	Atomic Population (L2)	Bond Order (Cu-N)	Bond Order (Cu-L1)
B97-3c	0.98	0.99	0.2	0.43	1	0.99	0.84	0.9
PBE0	1	1	0.76	0.71	1	1	0.85	0.96
TPSS	0.99	0.99	0.76	0.51	1	1	0.83	0.95
TPSSh	1	1	0.77	0.6	1	1	0.84	0.96
Method	Orbital Charge N(s)	Orbital Charge N(p)	Orbital Charge Cu(s)	Orbital Charge Cu(p)	Orbital Charge Cu(d)	Orbital Charge Cu(dxz)	Orbital Charge Cu(dyz)	Orbital Charge Cu(dxy)
B97-3c	0.78	0.69	0.89	0.91	0.31	0.47	0.3	0.1
PBE0	0.94	0.89	0.97	0.97	0.82	0.78	0.65	0.71
TPSS	0.94	0.82	0.95	0.96	0.69	0.67	0.49	0.49
TPSSh	0.94	0.85	0.96	0.96	0.74	0.71	0.55	0.58
Method	Orbital Charge Cu(dz2)	Orbital Charge Cu(dx2y2)	Orbital Charge L1(s)	Orbital Charge L1(p)	Orbital Charge L1(d)	Orbital Charge L2(s)	Orbital Charge L2(p)	Orbital Charge L2(d)
B97-3c	0.71	0.04	0.99	1	−0.86	1	1	−0.89
PBE0	0.92	0.83	1	1	1	1	1	1
TPSS	0.82	0.48	1	1	1	1	1	1
TPSSh	0.87	0.64	1	1	1	1	1	1

5.5.3.2 Correlation of Activation Energies for each Functional and DLPNO-CCSD(T)

Table S30 Correlation of activation energies between B97-3c and 4 other DFT methods and DLPNO-CCSD(T) for 50 ligands. RMSE_Actual is the RMSE of the raw value of the activation energy compared to DLPNO-CCSD(T)/def2-TZVPP. RMSE_Scaled is the RMSE of the scaled activation energy using the equation of the line to convert to a DLPNO-CCSD(T)/def2-TZVPP energy.

	TSOA Activation Energy			TSSig Activation Energy		
	R^2	RMSE_Actual	RMSE_Scaled	R^2	RMSE_Actual	RMSE_Scaled
B97-3c	0.91	5.97	3.78	0.87	8.96	3.46
PBE0	0.96	4.67	2.54	0.97	4.34	1.65
TPSS	0.88	10.01	4.30	0.88	8.87	3.33
TPSSh	0.92	8.40	3.67	0.93	7.14	2.57

Table S31 Comparison of single core computational time for the DFT energy calculations compared to the total single core time to calculate each transition state for the machine learning datasets.

	Single Core CPU time (h)			
Nulceophile	Transition States (TSOA+TSSig)	B97-3c Energy	TPSS Energy	PBE0 Energy
IPip	44 139	1433	3264	8384
IPyr	48 319	2165	4606	11 510

5.5.3.3 PBE0 Models

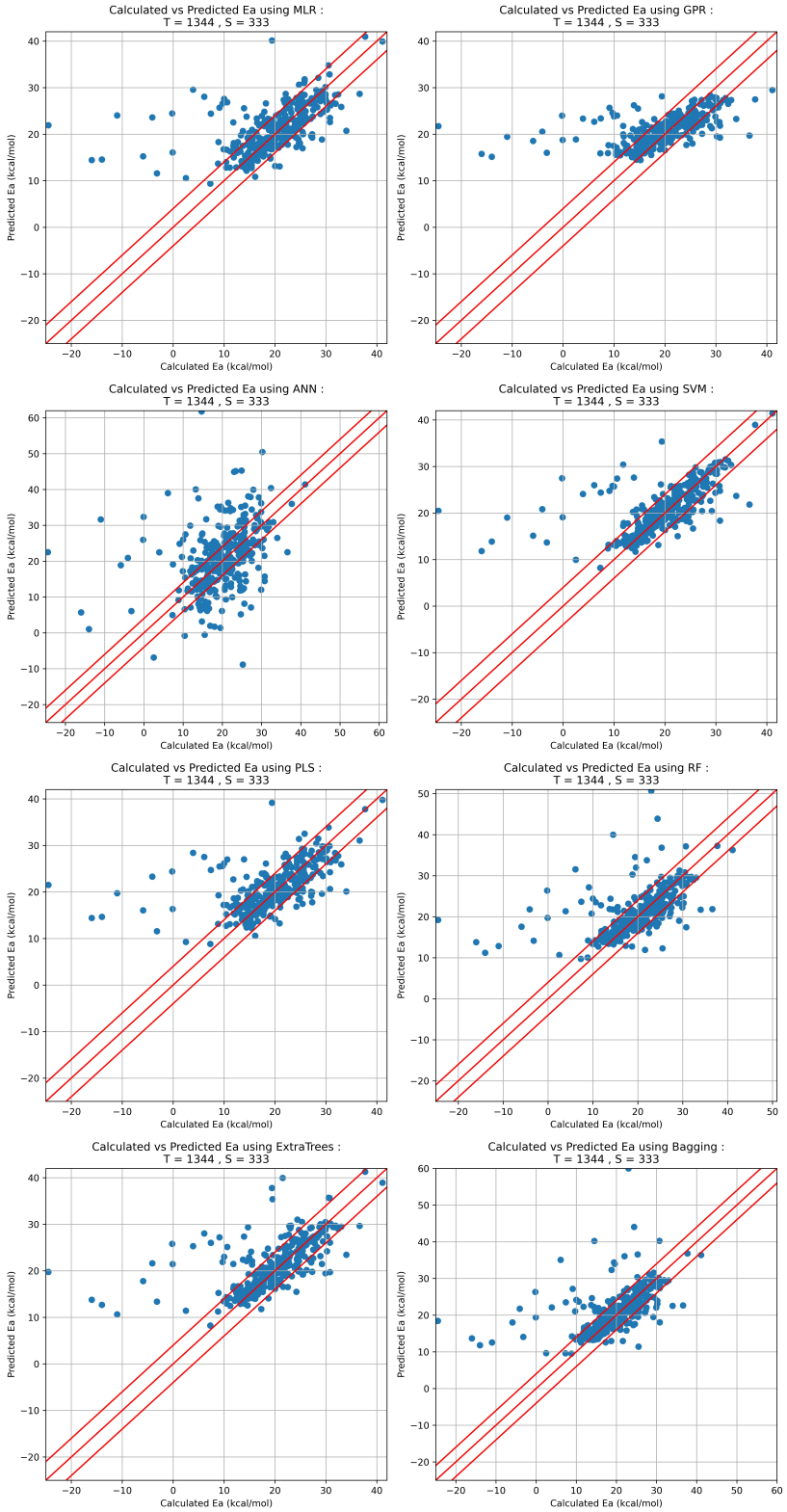


Fig. S49 Machine learning models using descriptors calculated using PBE0/def2-TZVP for the ***PIP_set_-TSOA_NoTS*** dataset.

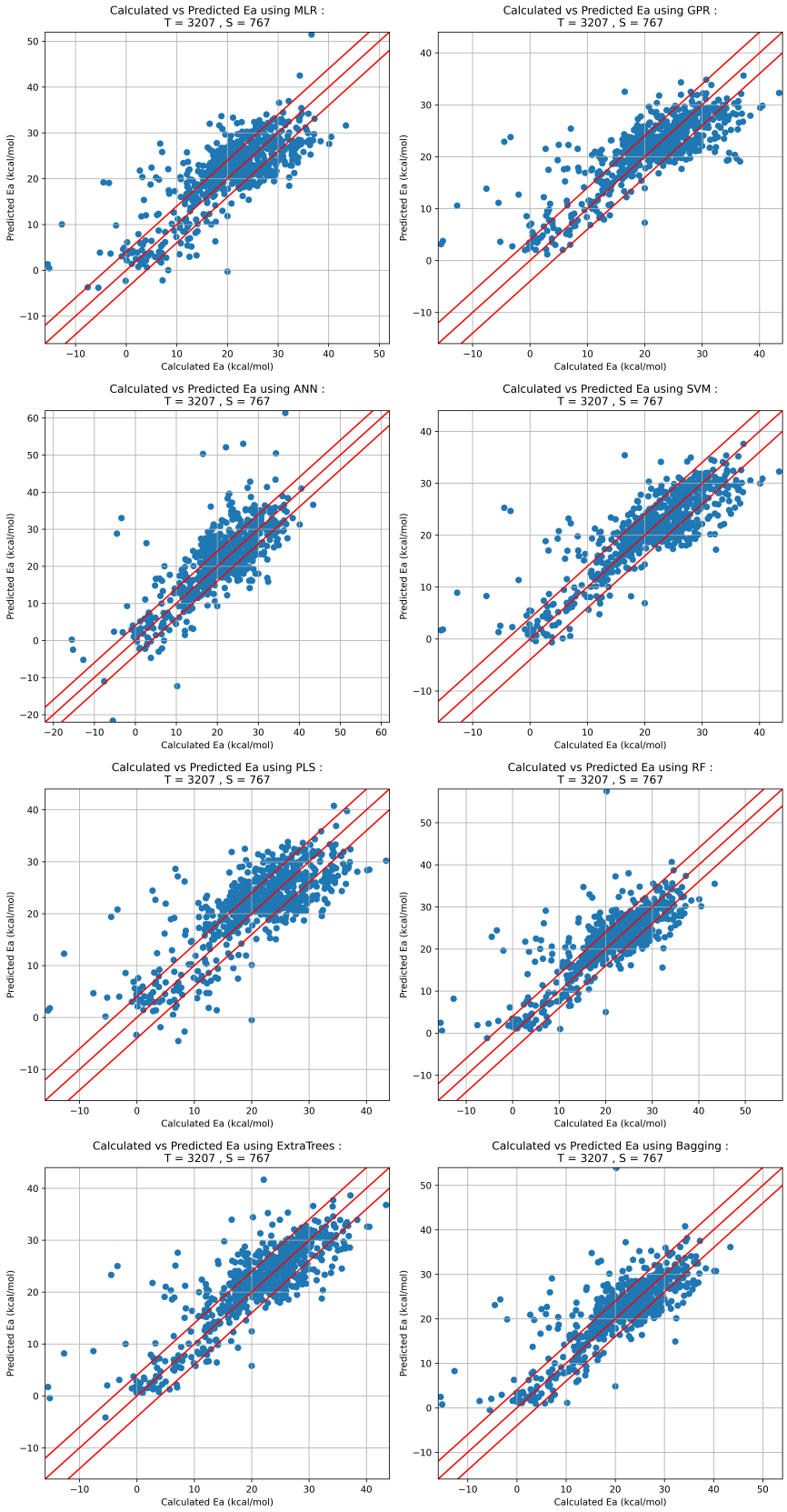


Fig. S50 Machine learning models using descriptors calculated using PBE0/def2-TZVP for the ***PYR_set_-TSOA_NoTS*** dataset.

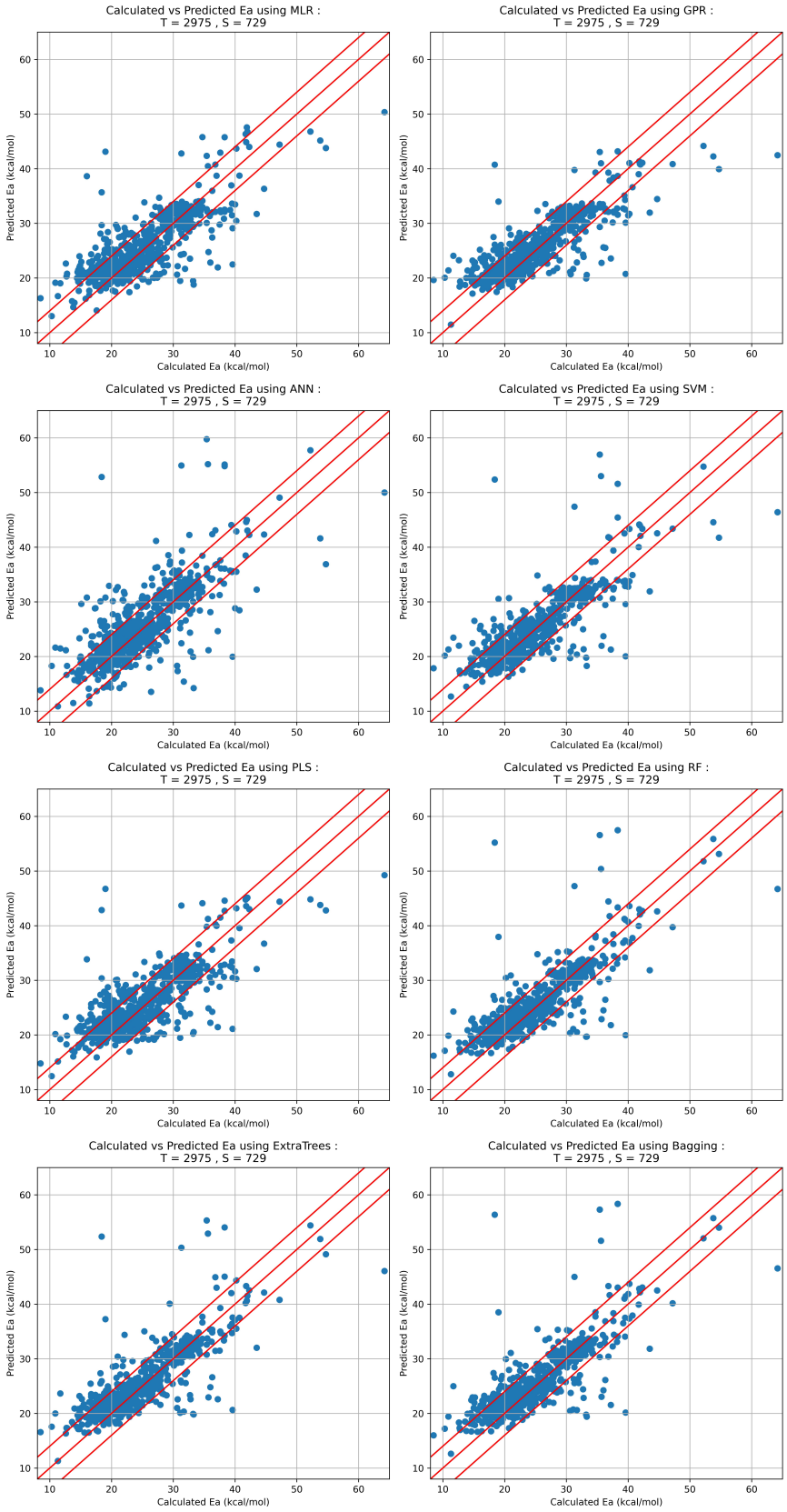


Fig. S51 Machine learning models using descriptors calculated using PBE0/def2-TZVP for the ***PIP_set_-TSSig_NoTS*** dataset.

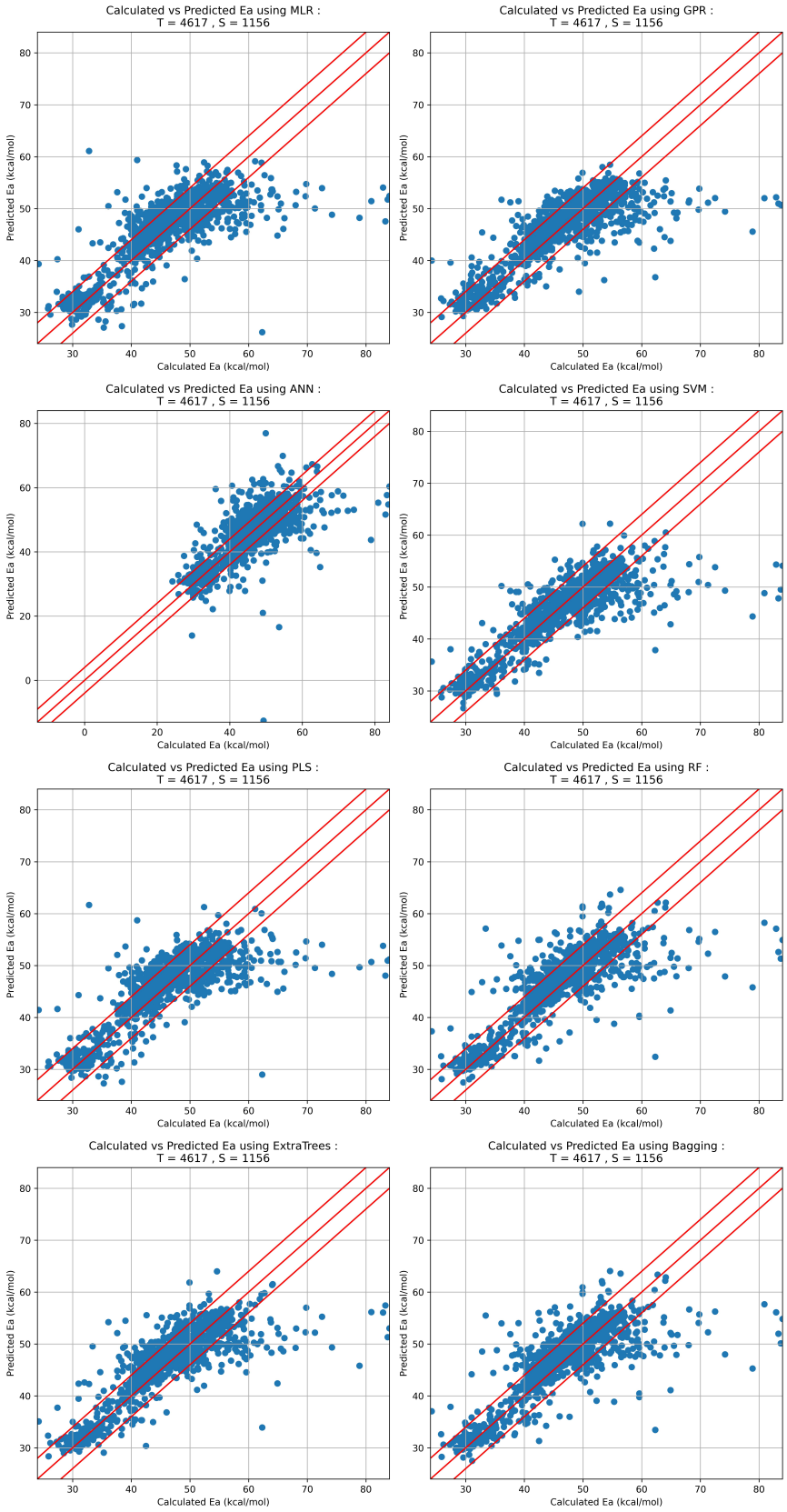


Fig. S52 Machine learning models using descriptors calculated using PBE0/def2-TZVP for the ***PYR_set_-TSSig_NoTS*** dataset.

Table S32 Machine learning metrics using descriptors calculated using PBE0/def2-TZVP for the ***PIP_set_TSOA_NoTS*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.30	6.63	71.8
GPR	0.34	6.48	71.8
ANN	0.13	9.62	47.4
SVM	0.40	6.11	82.3
PLS	0.32	6.51	73.0
RF	0.32	6.62	77.2
ExtraTrees	0.38	6.28	77.8
Bagging	0.30	6.84	77.5

Table S33 Machine learning metrics using descriptors calculated using PBE0/def2-TZVP for the ***PYR_set_TSOA_NoTS*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.62	5.35	59.6
GPR	0.66	5.03	67.3
ANN	0.62	5.90	62.5
SVM	0.71	4.59	72.6
PLS	0.61	5.43	57.2
RF	0.67	5.00	71.7
ExtraTrees	0.71	4.68	72.9
Bagging	0.68	4.97	71.7

Table S34 Machine learning metrics using descriptors calculated using PBE0/def2-TZVP for the ***PIP_set_TSSig_NoTS*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.65	3.99	76.1
GPR	0.71	3.66	81.8
ANN	0.60	4.58	76.0
SVM	0.69	3.78	84.6
PLS	0.63	4.12	77.2
RF	0.69	3.80	84.4
ExtraTrees	0.69	3.79	84.4
Bagging	0.69	3.83	84.4

Table S35 Machine learning metrics using descriptors calculated using PBE0/def2-TZVP for the ***PYR_set_TSSig_NoTS*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.60	5.16	72.5
GPR	0.62	5.03	74.7
ANN	0.51	6.11	66.2
SVM	0.68	4.72	79.3
PLS	0.61	5.12	73.8
RF	0.65	4.85	77.9
ExtraTrees	0.68	4.66	78.0
Bagging	0.65	4.88	77.3

5.5.3.4 TPSS Models

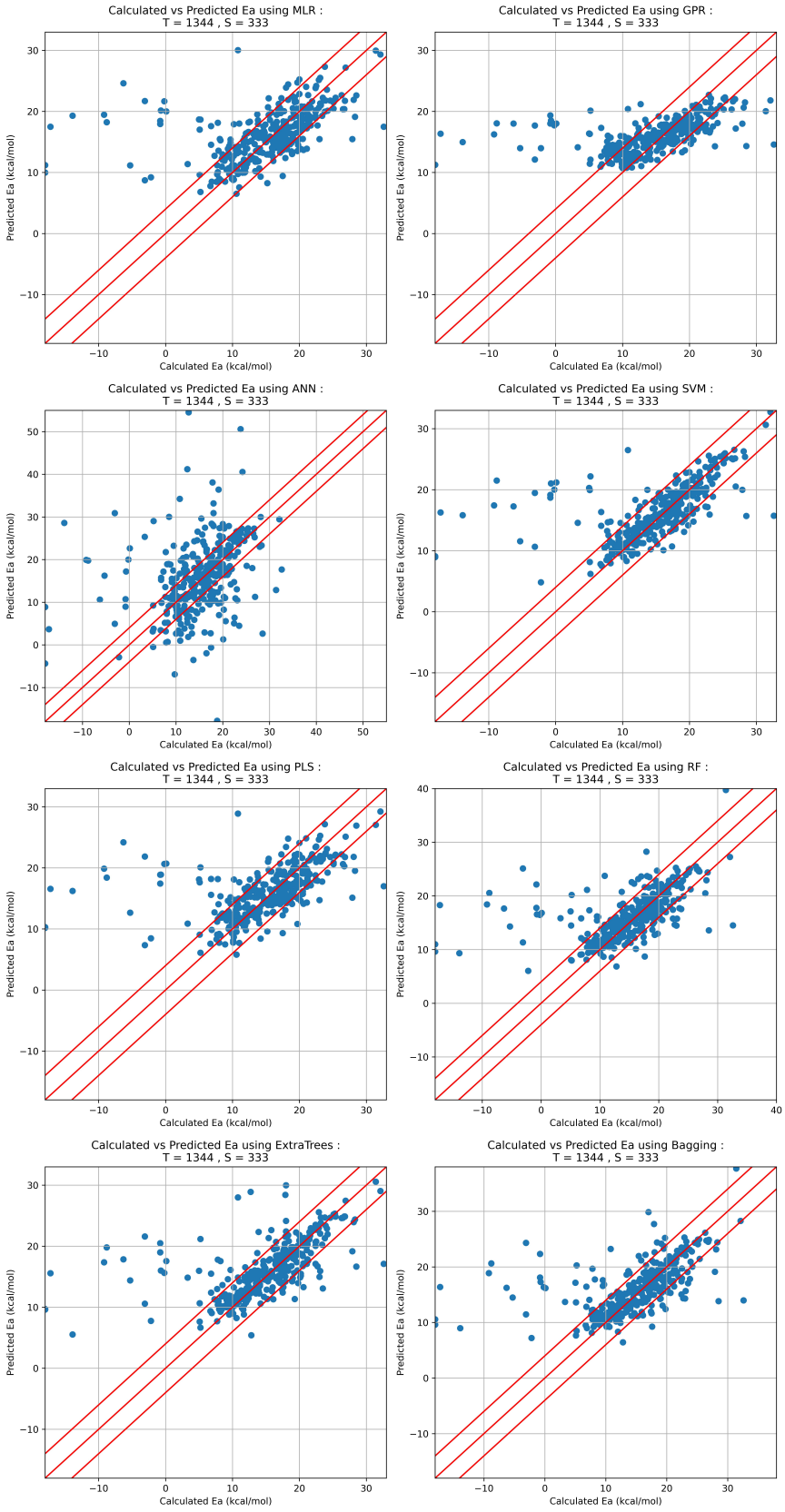


Fig. S53 Machine learning models using descriptors calculated using TPSS/def2-TZVP for the ***PIP_set_-TSOA_NoTS*** dataset.

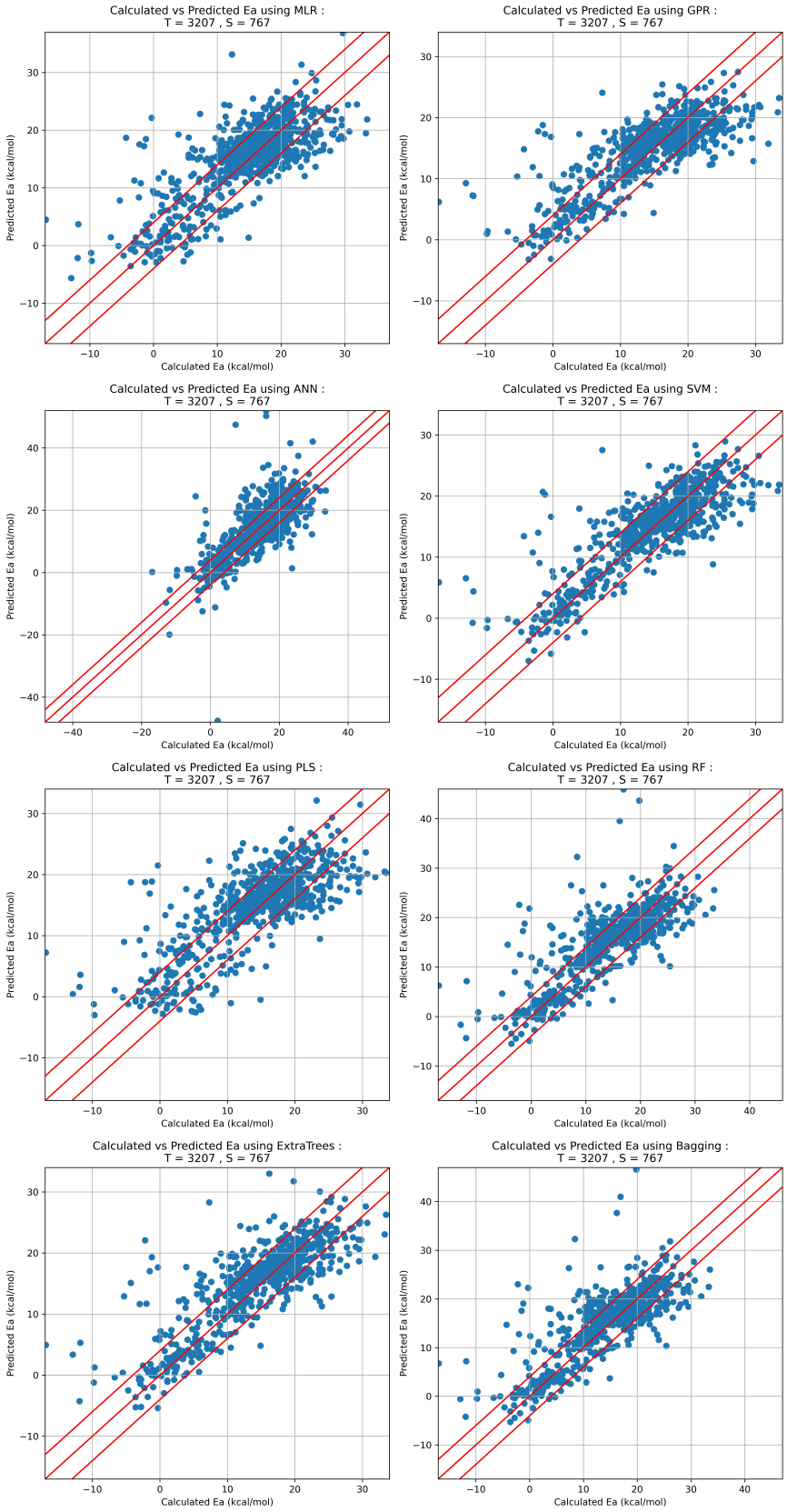


Fig. S54 Machine learning models using descriptors calculated using TPSS/def2-TZVP for the ***PYR_set_-TSOA_NoTS*** dataset.

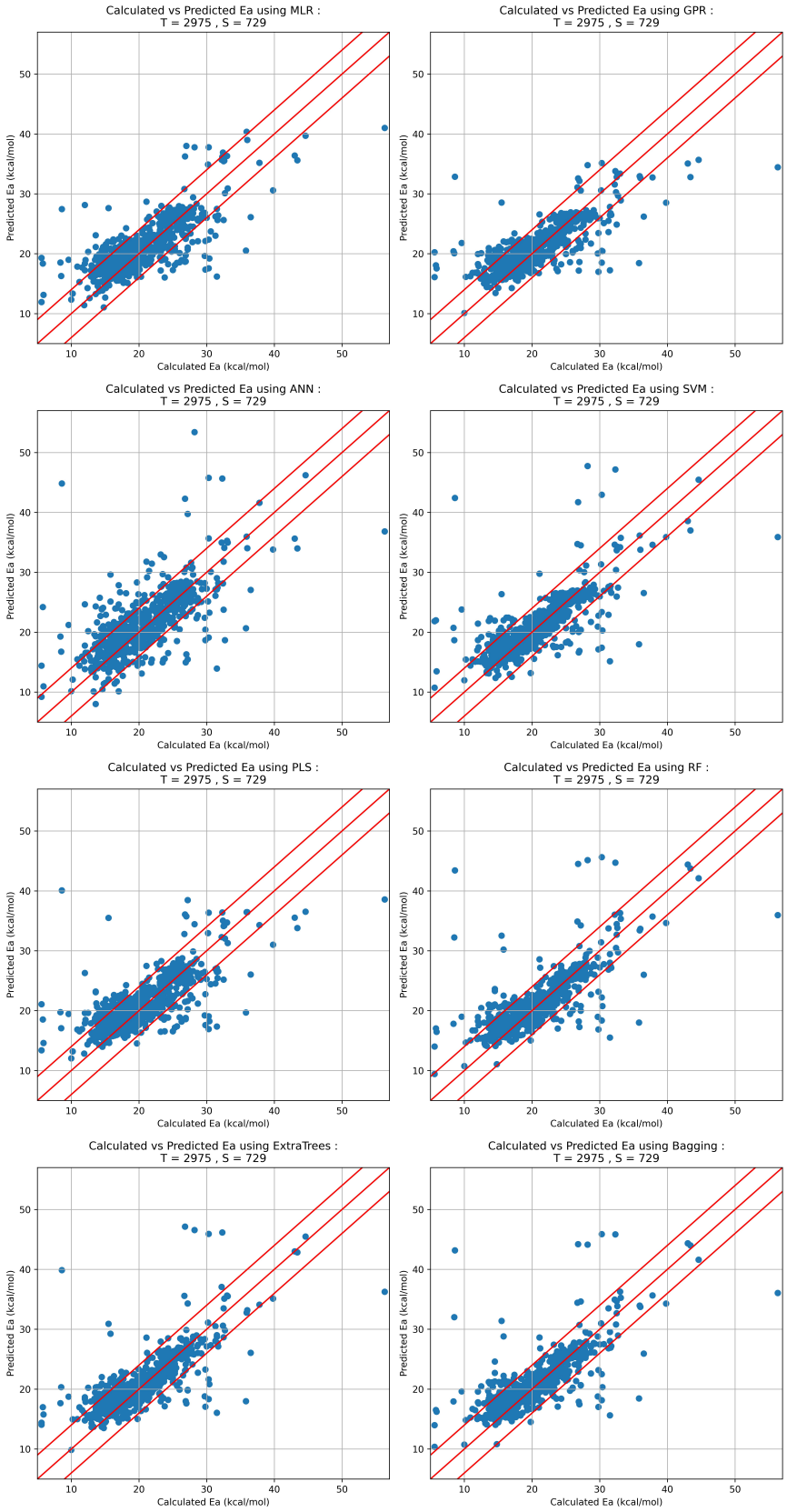


Fig. S55 Machine learning models using descriptors calculated using TPSS/def2-TZVP for the ***PIP_set_-TSSig_NoTS*** dataset.

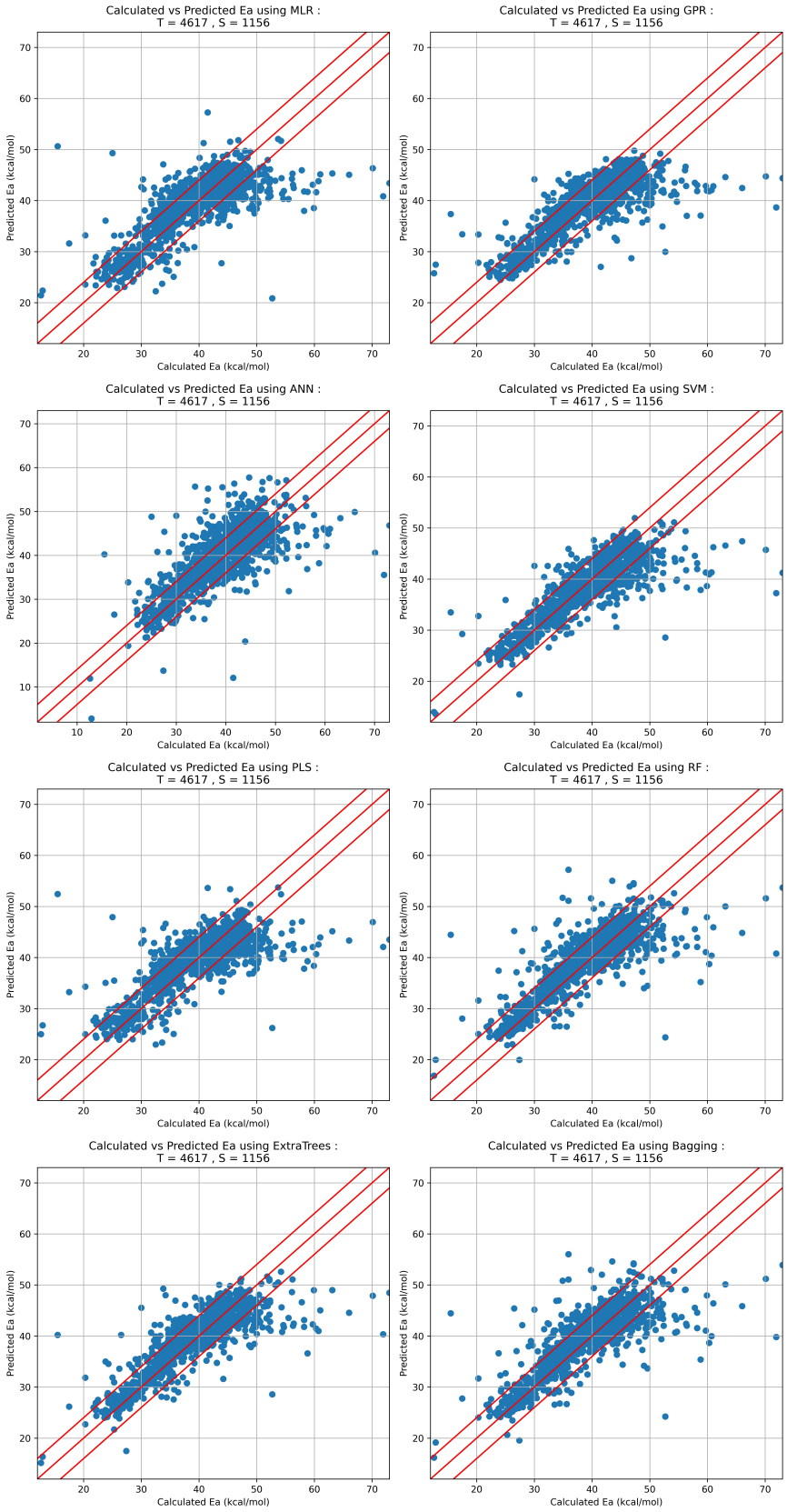


Fig. S56 Machine learning models using descriptors calculated using TPSS/def2-TZVP for the ***PYR_set_-TSSig_NoTS*** dataset.

Table S36 Machine learning metrics using descriptors calculated using TPSS/def2-TZVP for the ***PIP_set_TSOA_NoTS*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.22	6.54	74.5
GPR	0.27	6.36	70.9
ANN	0.10	9.31	51.7
SVM	0.33	6.07	82.3
PLS	0.24	6.44	73.9
RF	0.29	6.26	77.8
ExtraTrees	0.34	6.03	80.5
Bagging	0.29	6.23	77.5

Table S37 Machine learning metrics using descriptors calculated using TPSS/def2-TZVP for the ***PYR_set_TSOA_NoTS*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.59	5.01	64.1
GPR	0.64	4.69	69.6
ANN	0.53	6.05	63.0
SVM	0.69	4.27	75.4
PLS	0.58	5.09	62.3
RF	0.63	4.84	74.3
ExtraTrees	0.68	4.42	74.8
Bagging	0.63	4.81	74.2

Table S38 Machine learning metrics using descriptors calculated using TPSS/def2-TZVP for the ***PIP_set_TSSig_NoTS*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.61	3.45	83.1
GPR	0.63	3.38	84.9
ANN	0.51	4.15	79.4
SVM	0.60	3.53	87.1
PLS	0.56	3.68	83.4
RF	0.59	3.58	88.1
ExtraTrees	0.62	3.46	87.8
Bagging	0.60	3.57	88.1

Table S39 Machine learning metrics using descriptors calculated using TPSS/def2-TZVP for the ***PYR_set_TSSig_NoTS*** dataset.

Model	R^2	RMSE	% within 4.0
MLR	0.58	4.83	72.6
GPR	0.63	4.53	77.2
ANN	0.54	5.28	66.7
SVM	0.69	4.19	80.8
PLS	0.58	4.83	73.0
RF	0.65	4.39	78.7
ExtraTrees	0.70	4.09	80.4
Bagging	0.65	4.39	78.8

References

1 M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman and D. J. Fox, *Gaussian 09 Revision D.01*, 2009, Gaussian Inc. Wallingford CT.

2 F. Neese, *WIREs Computational Molecular Science*, 2018, **8**, e1327.

3 *Semiempirical Extended Tight-Binding Program Package*, `https://github.com/grimme-lab/xtb`, Accessed: March 2020.

4 *University of Leeds ARC Supercomputer*, `https://arc.leeds.ac.uk/`, Accessed: April 2020.

5 E. Ioannidis, T. Gani and H. Kulik, *Journal of computational chemistry*, 2016, **37**,.

6 H. J. Reich, *Bordwell pKa Table (in DMSO)*, 2020, `https://www.chem.wisc.edu/areas/reich/pkatable/`.

7 C. R. Groom, I. J. Bruno, M. P. Lightfoot and S. C. Ward, *Acta Crystallographica Section B*, 2016, **72**, 171–179.

8 W. Kabsch, *Acta Crystallographica Section A*, 1976, **32**, 922–923.

9 O. Korb, B. Kuhn, J. Hert, N. Taylor, J. Cole, C. Groom and M. Stahl, *Journal of Medicinal Chemistry*, 2016, **59**, 4257–4266.

10 N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch and G. R. Hutchison, *Journal of Cheminformatics*, 2011, **3**, 33.

11 L. D. Jacobson, A. D. Bochevarov, M. A. Watson, T. F. Hughes, D. Rinaldo, S. Ehrlich, T. B. Steinbrecher, S. Vaitheeswaran, D. M. Philipp, M. D. Halls and R. A. Friesner, *Journal of Chemical Theory and Computation*, 2017, **13**, 5780–5797.

12 *GitHub - kjelljorner/morfeus: A Python package for calculating molecular features — github.com*, `https://github.com/kjelljorner/morfeus#readme`, [Accessed Oct-2021].

13 E. Berquist, G. Hutchison, K. M. Langner, N. M. O'Boyle, A. L. Tenderholt and S. Upadhyay, *Release of cclib version 1.7*, 2021, `https://doi.org/10.5281/zenodo.4420433`.

14 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, *Journal of Machine Learning Research*, 2011, **12**, 2825–2830.

15 T. Akiba, S. Sano, T. Yanase, T. Ohta and M. Koyama, Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

16 F. Neese, *WIREs Computational Molecular Science*, 2022, **12**, e1606.