Electronic Supplementary Information

for

Machine Learning Models for Catalytic Asymmetric Reactions of Simple Alkenes: From Enantioselectivity Predictions to Chemical Insights

Ajnabiul Hoque,[a] Nupur Jain,[a] Divya Chenna,[a] and Raghavan B. Sunoj[a,b,*]

[a]Department of Chemistry, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

[b]Centre for Machine Intelligence and Data Science, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

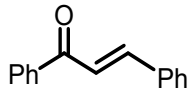
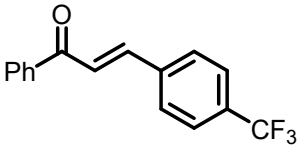
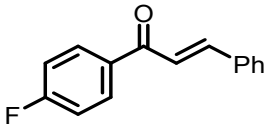
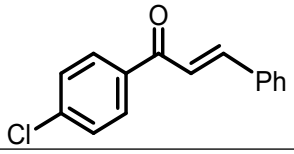
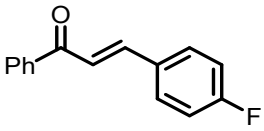
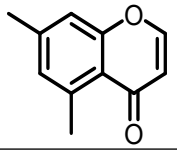
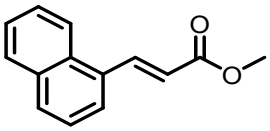
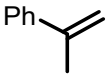
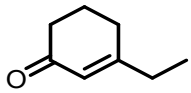
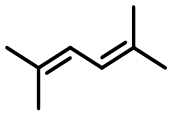
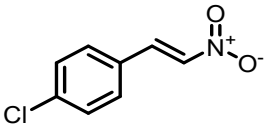
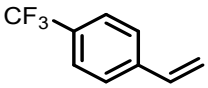
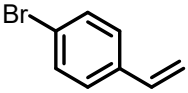
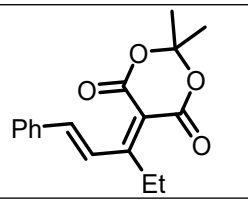
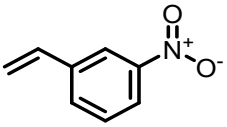
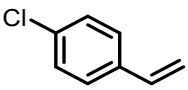
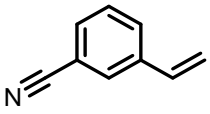
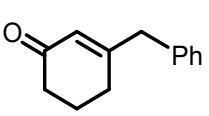
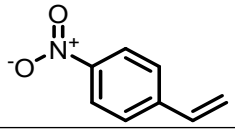
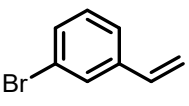
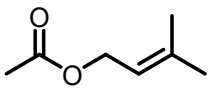
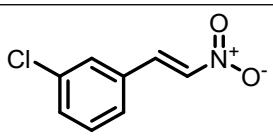
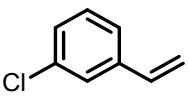
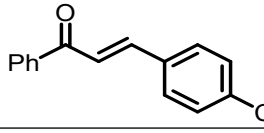
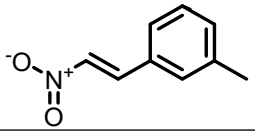
E-mail: sunoj@chem.iitb.ac.in

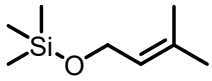
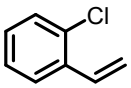
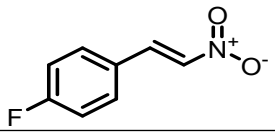
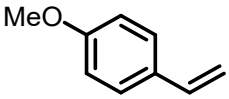
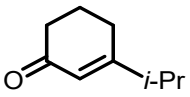
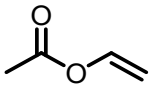
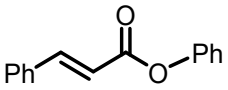
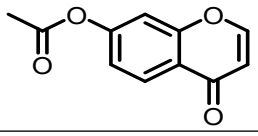
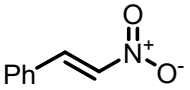
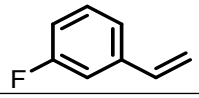
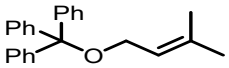
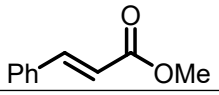
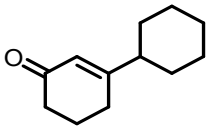
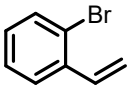
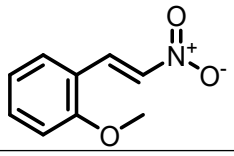
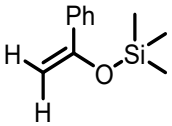
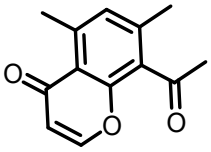
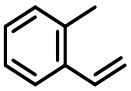
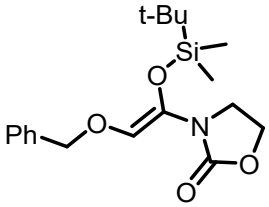
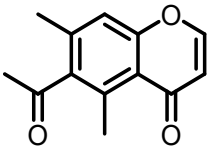
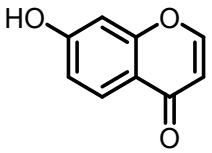
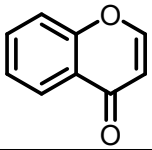
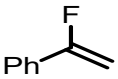
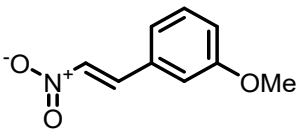
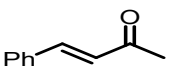
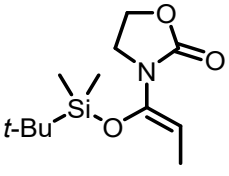
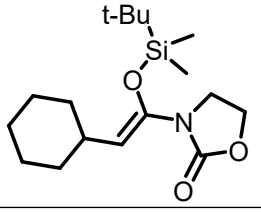
Contents

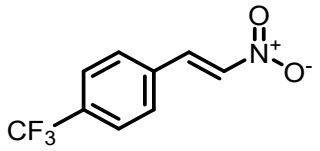
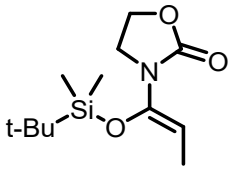
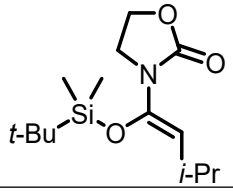
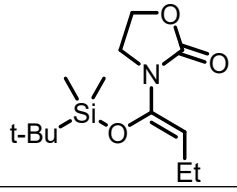
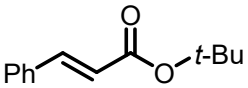
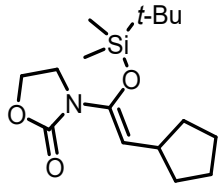
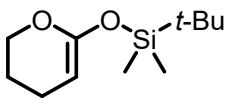
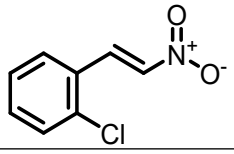
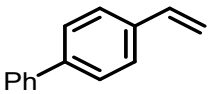
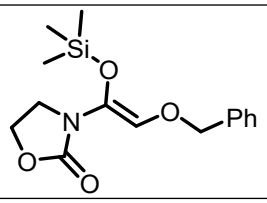
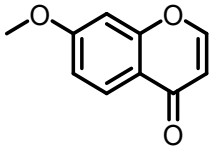
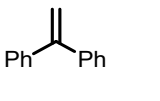
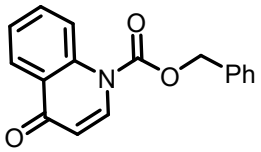
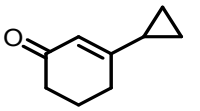
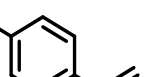
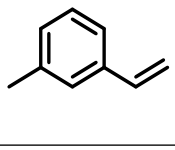
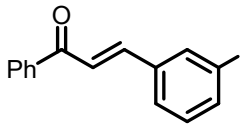
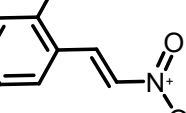
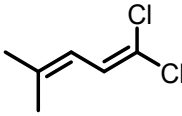
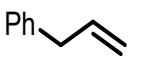
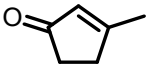
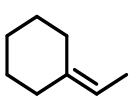
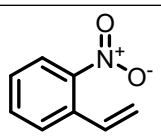
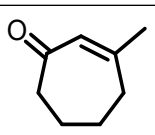
1. Details of all reacting components

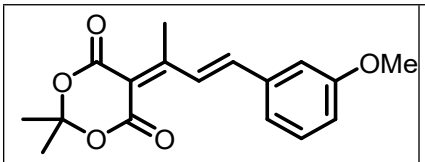
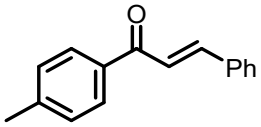
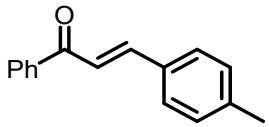
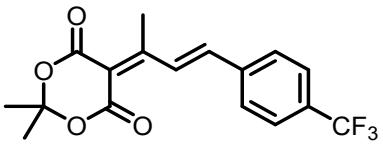
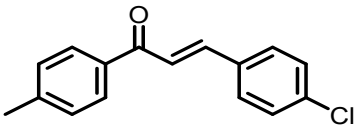
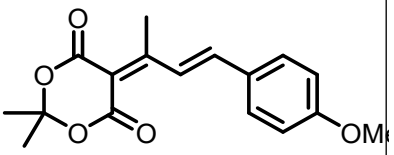
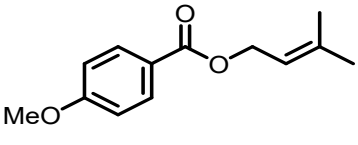
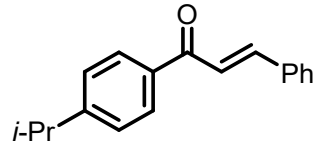
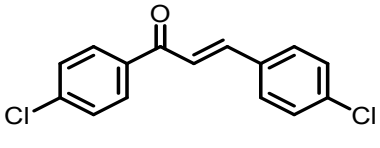
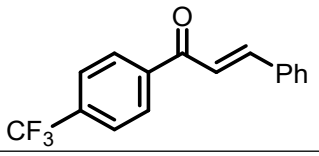
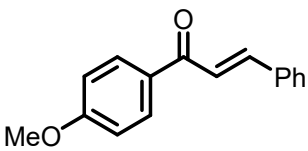
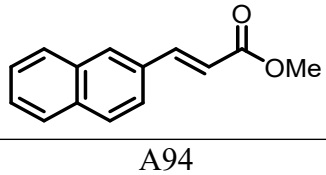
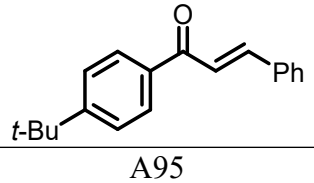
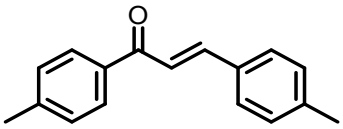
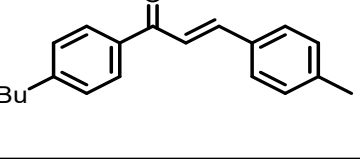
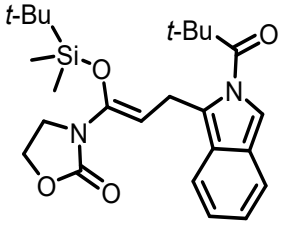
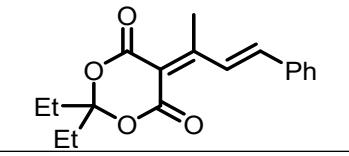
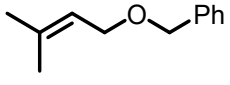
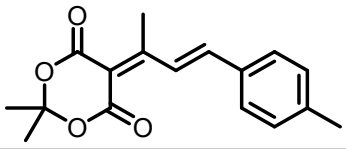
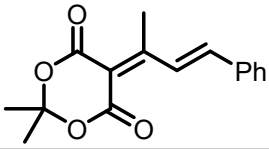
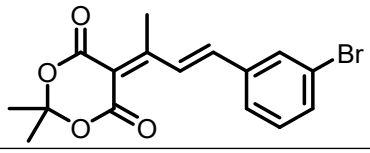
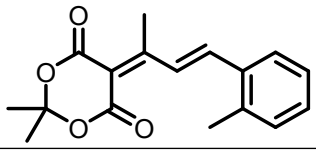
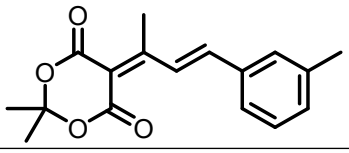
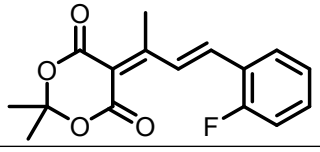
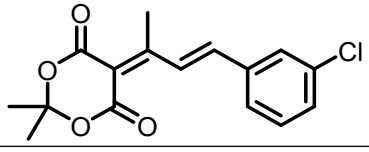
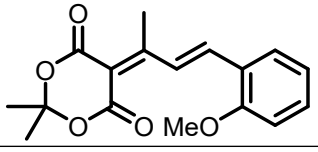
(1.1) Details of substrate library

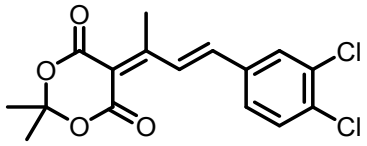
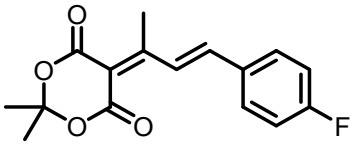
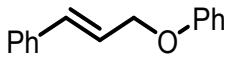
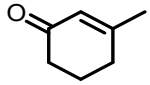
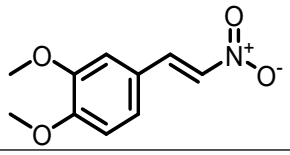
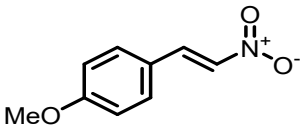
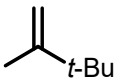
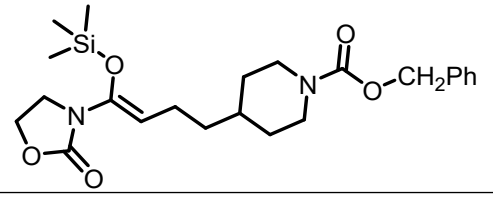
Table S1. Details of Substrates (alkenes)

		
A1	A2	A3
		
A4	A5	A6
		
A7	A8	A9
		
A10	A11	A12
		
A13	A14	A15
		
A16	A17	A18
		
A19	A20	A21
		
A22	A23	A24
		

A25	A26	A27
		
A28	A29	A30
		
A31	A32	A33
		
A34	A35	A36
		
A37	A38	A39
		
A40	A41	A42
		
A43	A44	A45
		
A46	A47	A48
		
A49	A50	A51
		
A52	A53	A54

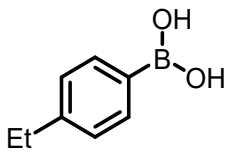
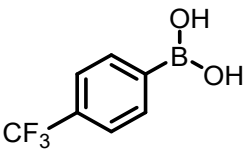
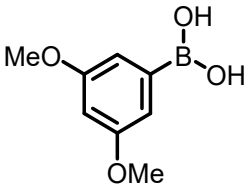
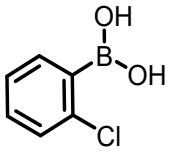
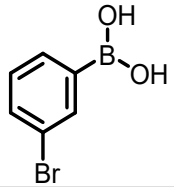
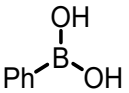
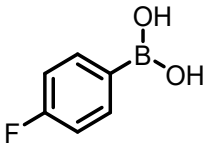
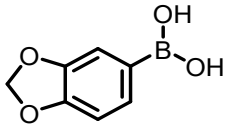
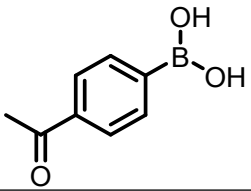
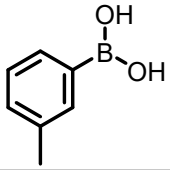
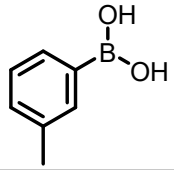
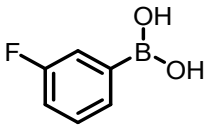
		
A55	A56	A57
		
A58	A59	A60
		
A61	A62	A63
		
A64	A65	A66
		
A67	A68	A69
		
A70	A71	A72
		
A73	A74	A75
		
A76	A77	A78
		
A79	A80	A81

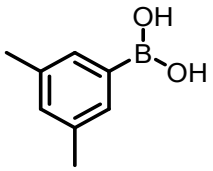
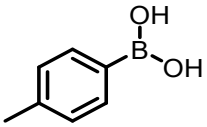
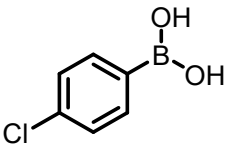
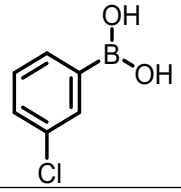
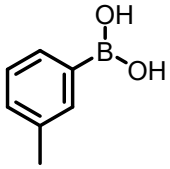
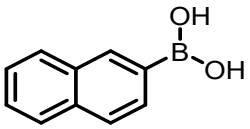
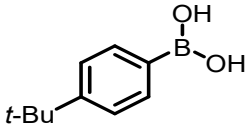
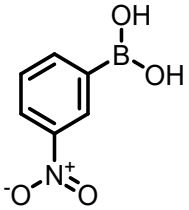
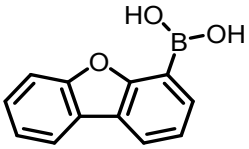
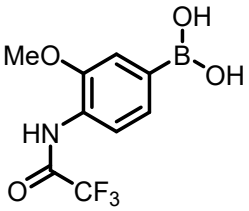
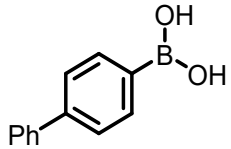
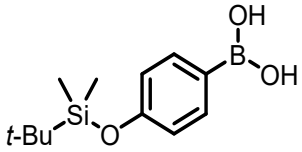
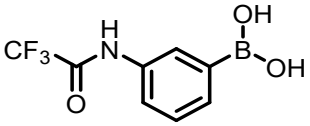
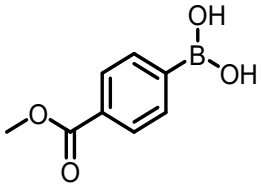
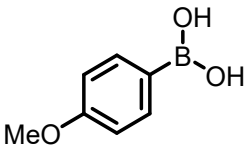
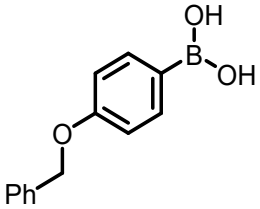
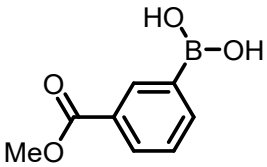
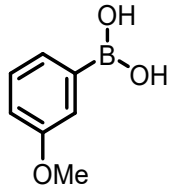
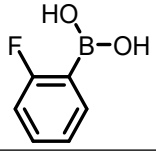
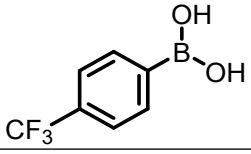
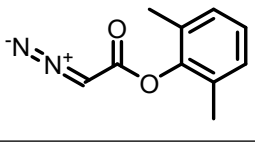
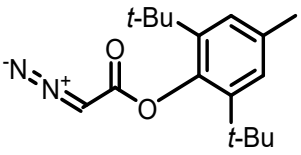
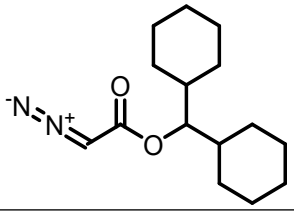
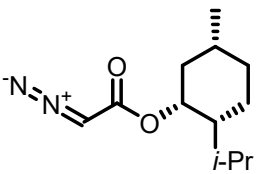
		
A82	A83	A84
		
A85	A86	A87
		
A88	A89	A90
		
A91	A92	A93
		
A94	A95	A96
		
A97	A98	A99
		
A100	A101	A102
		
A103	A104	A105
		

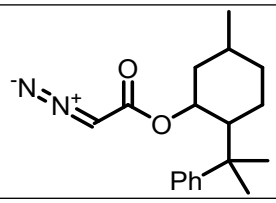
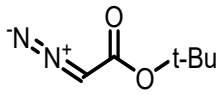
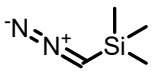
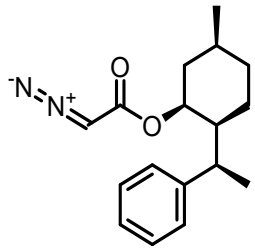
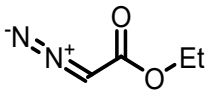
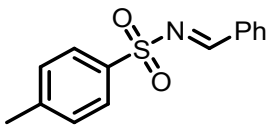
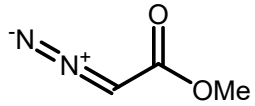
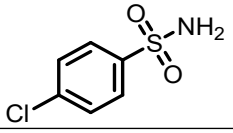
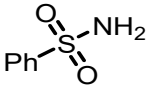
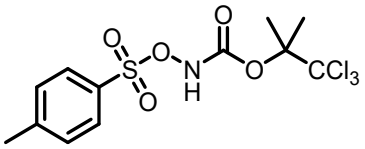
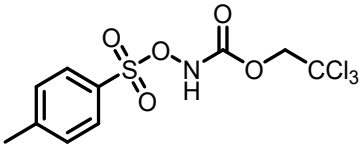
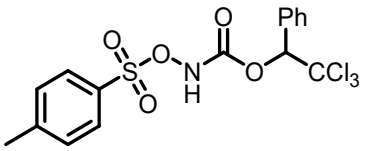
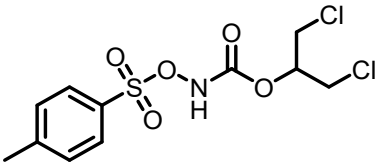
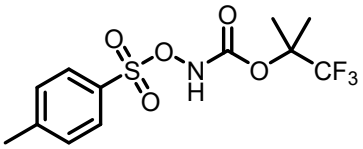
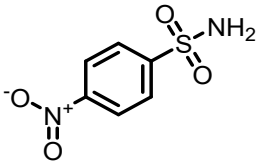
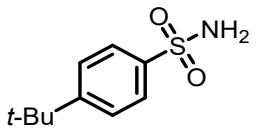
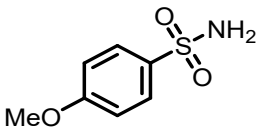
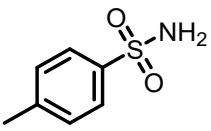
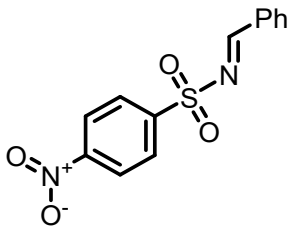
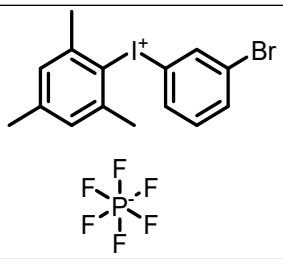
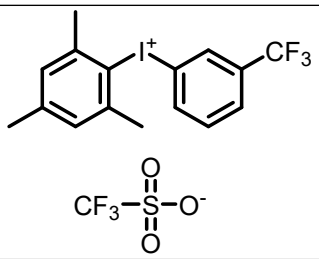
A106	A107	A108
		
A109	A110	A111
		
A112	A113	A114
		
A115	A116	

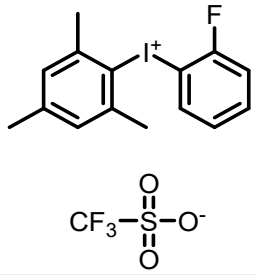
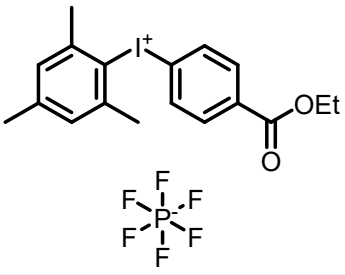
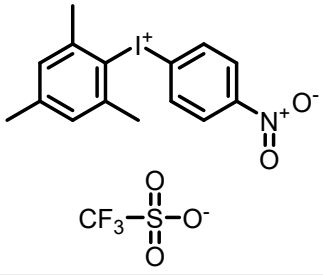
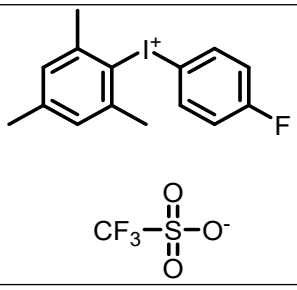
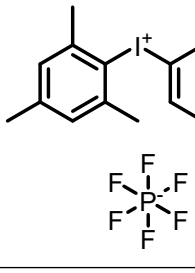
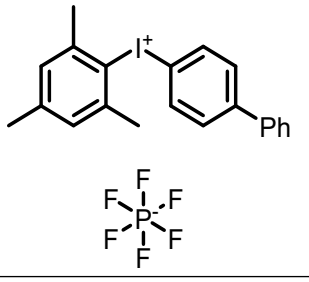
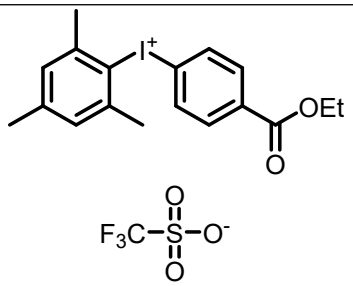
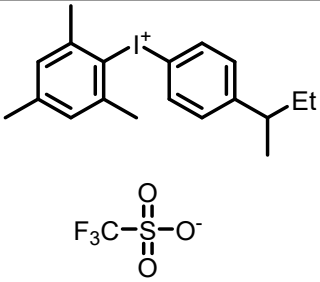
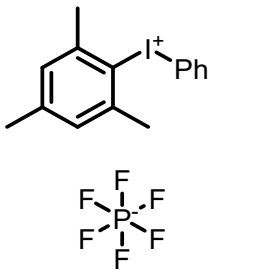
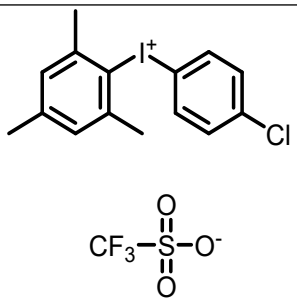
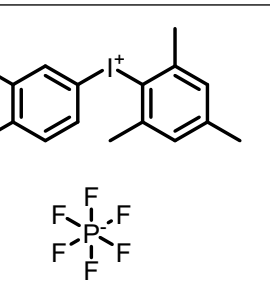
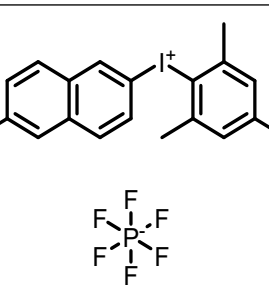
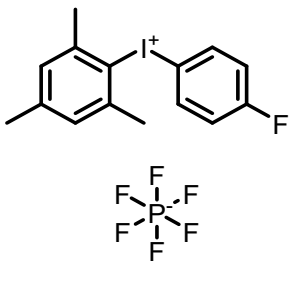
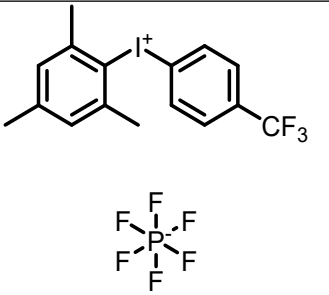
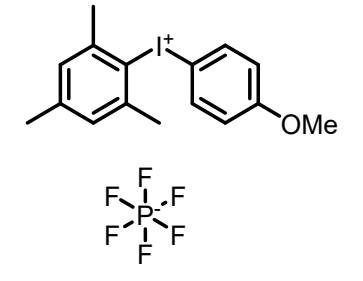
(1.2) Coupling partner library

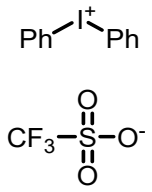
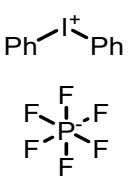
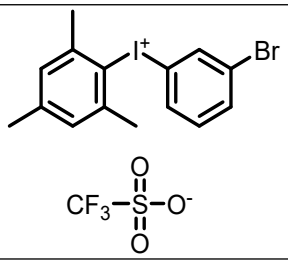
Table S2. The Coupling Partners

		
B1	B2	B3
		
B4	B4	B6
		
B7	B8	B9
		
B10	B11	B12

		
B13	B14	B15
		
B16	B17	B18
		
B19	B20	B21
		
B22	B23	B24
		
B23	B24	B25
		
B26	B27	B28
		
B29	B30	B31
		

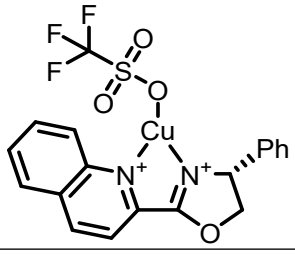
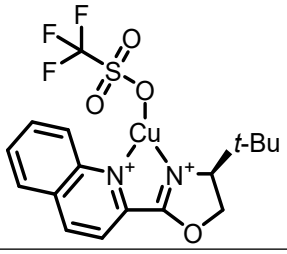
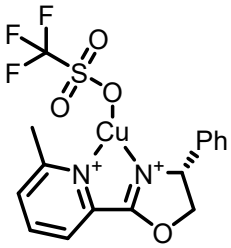
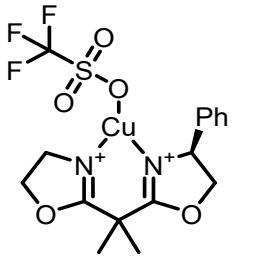
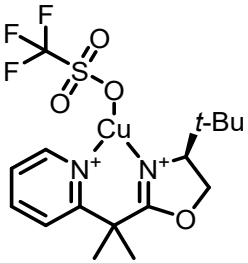
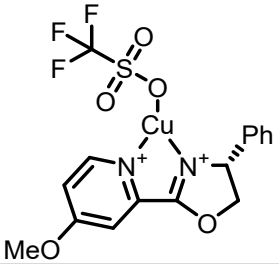
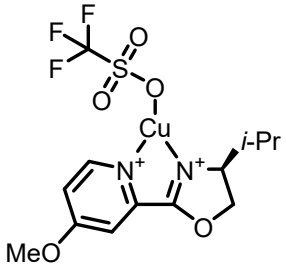
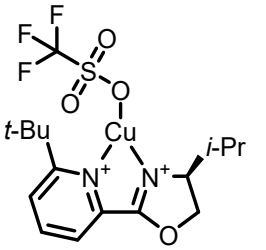
B32	B33	B34
		
B35	B36	B37
		
B38	B39	B40
		
B41	B42	B43
		
B44	B45	B46
		
B47	B48	B49
		
B50	B51	B52
		
B53	B54	B55

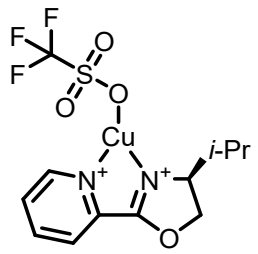
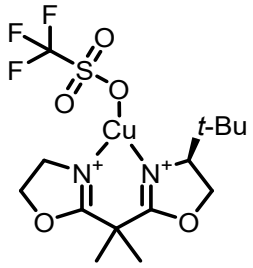
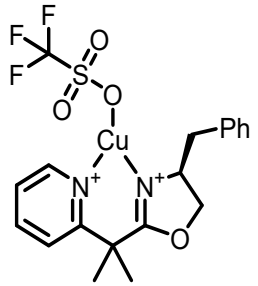
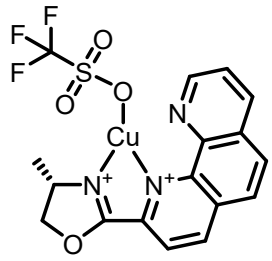
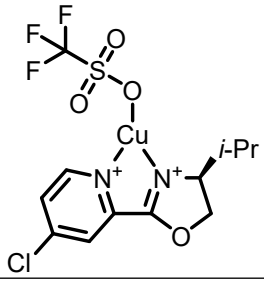
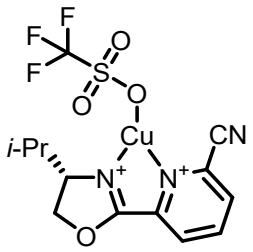
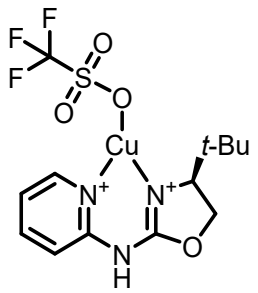
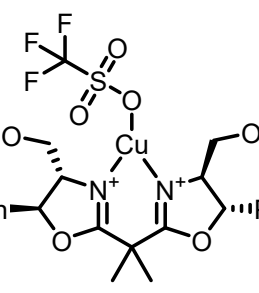
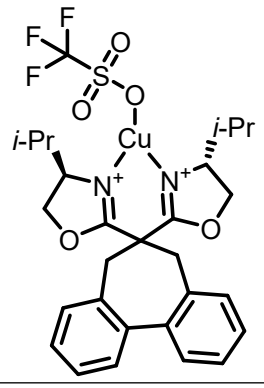
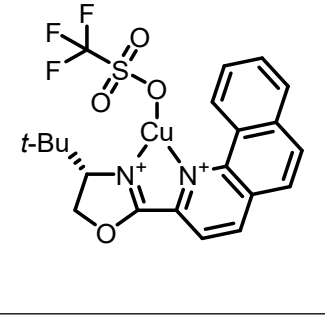
		
B56	B57	B58
		
B59	B60	B61
		
B62	B63	B64
		
B65	B66	B67
		
B68	B69	B70

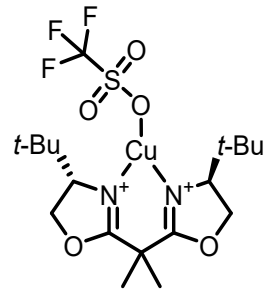
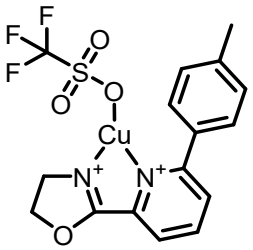
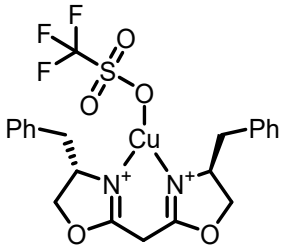
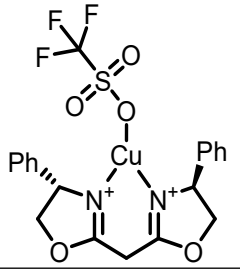
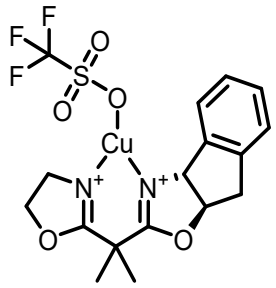
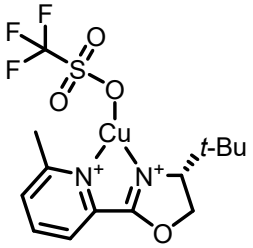
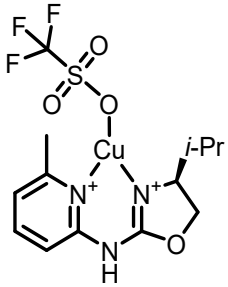
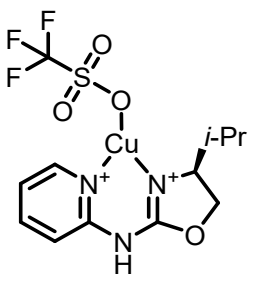
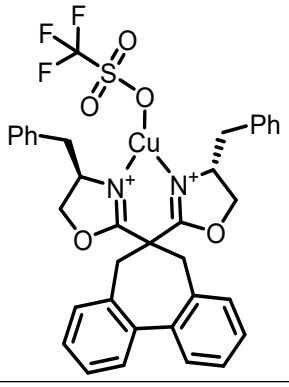
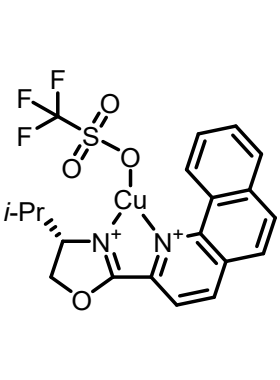
		
B71	B72	B73

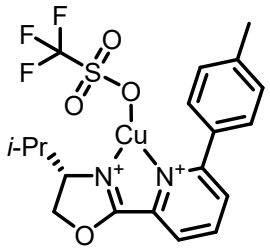
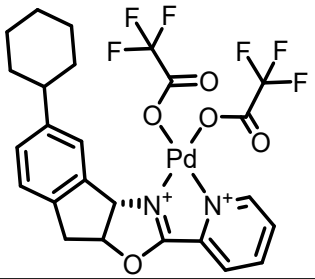
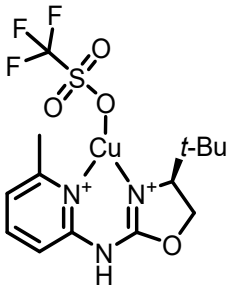
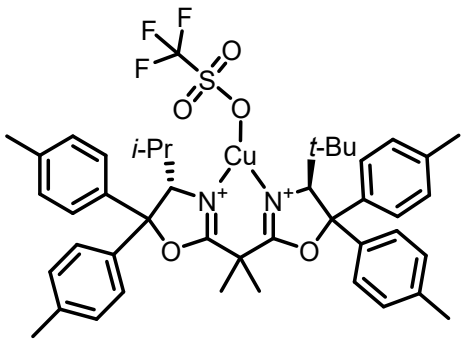
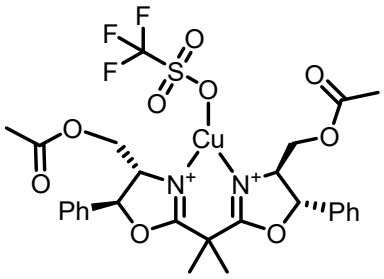
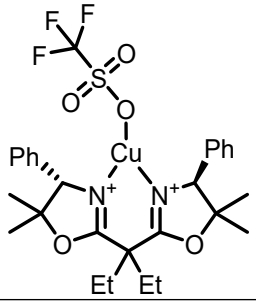
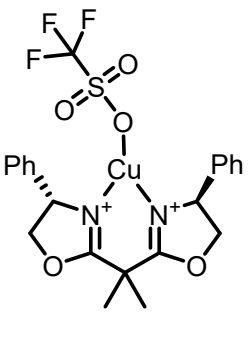
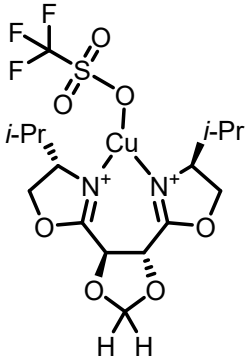
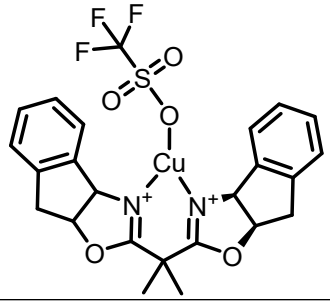
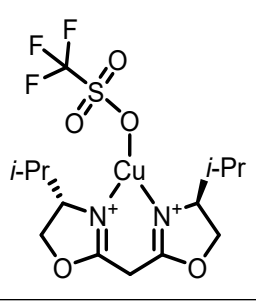
(1.3) Catalyst library

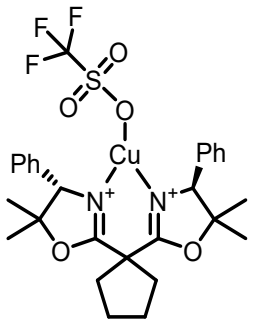
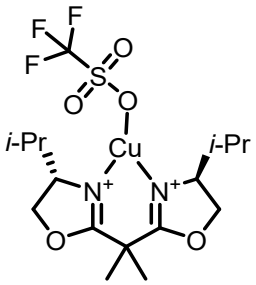
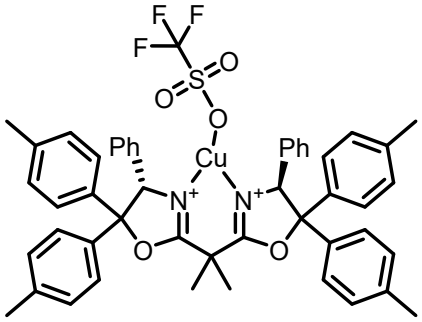
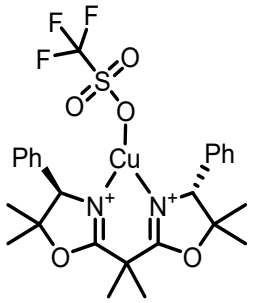
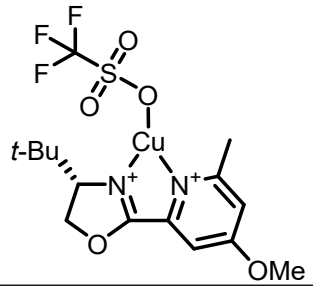
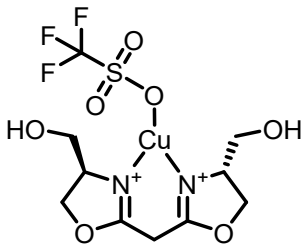
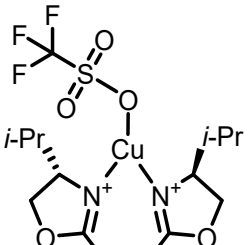
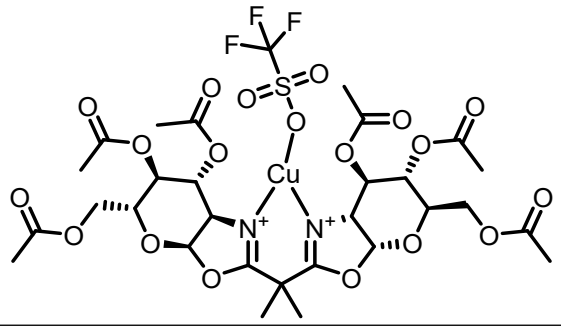
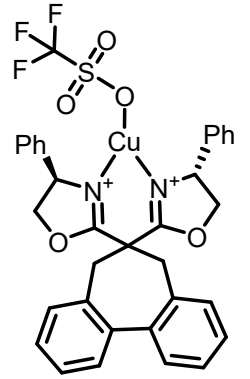
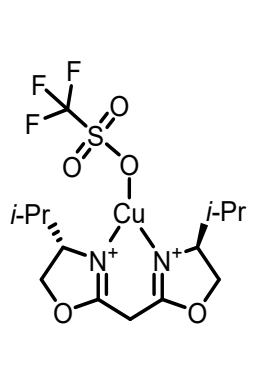
Table S3. Transition Metal Catalysts

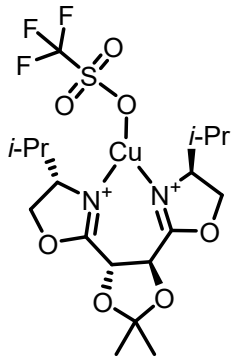
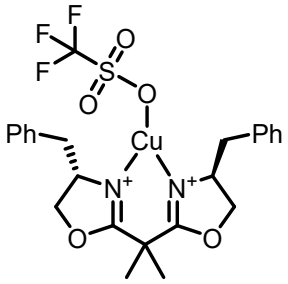
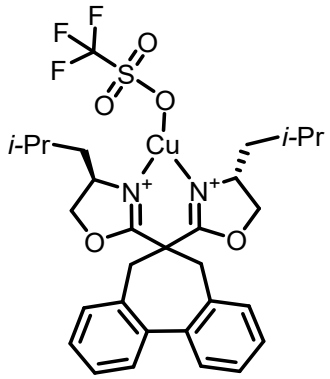
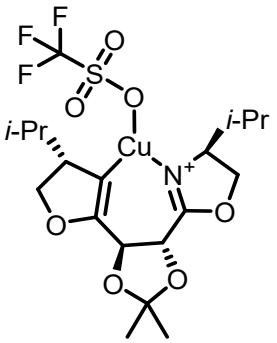
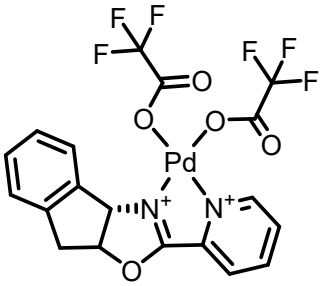
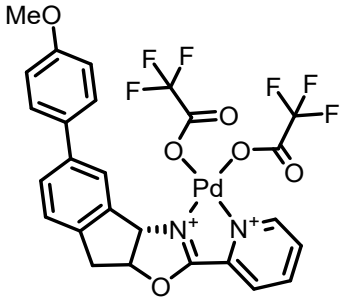
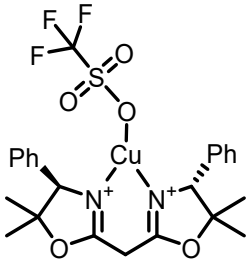
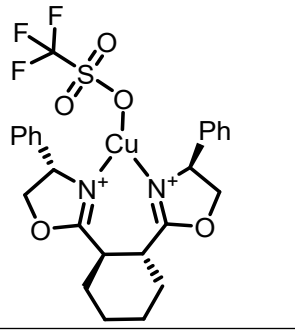
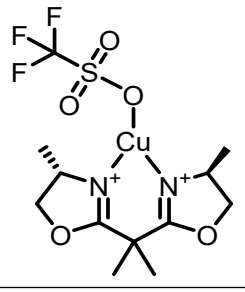
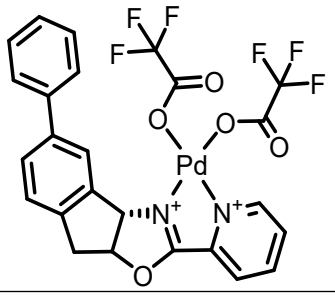
	
C1	C2
	
C3	C4
	
C5	C6
	
C7	C8

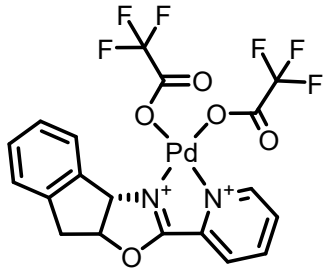
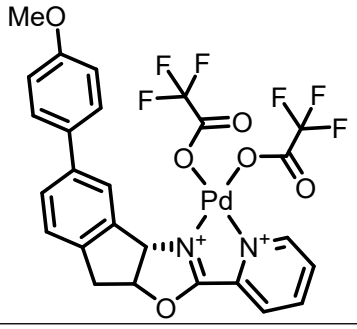
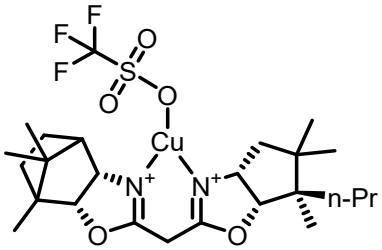
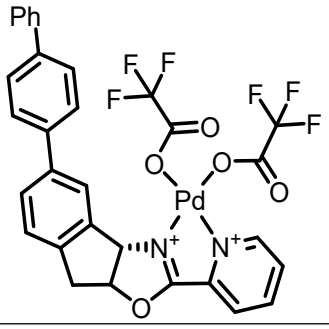
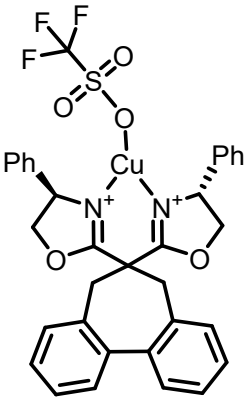
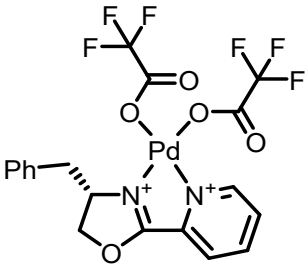
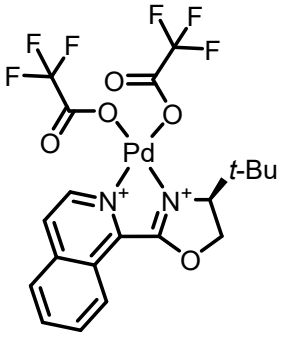
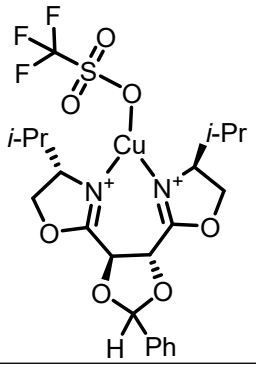
	
C9	C10
	
C11	C12
	
C13	C14
	
C15	C16
	
C17	C18

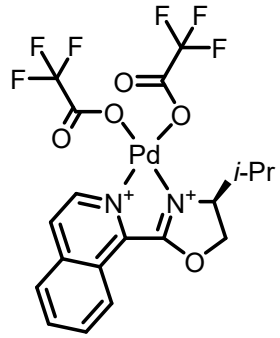
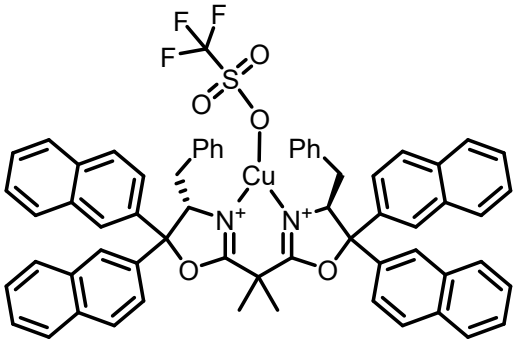
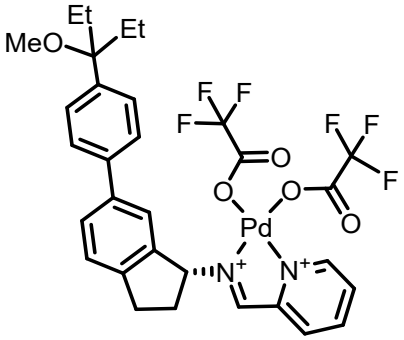
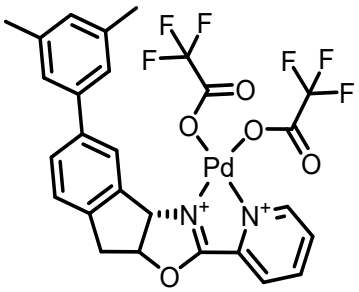
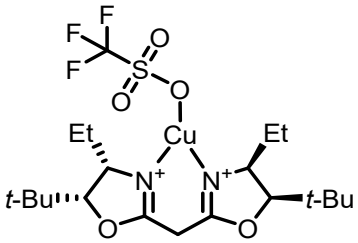
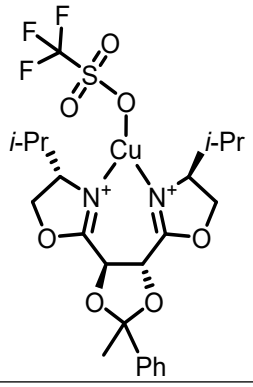
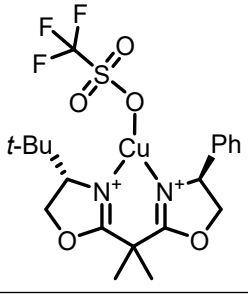
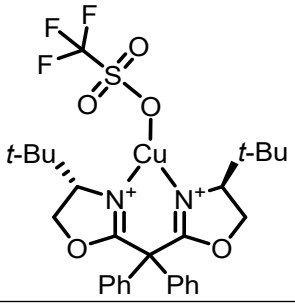
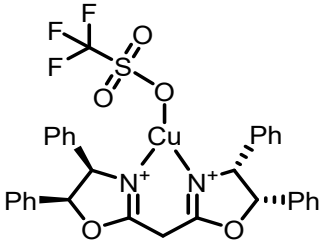
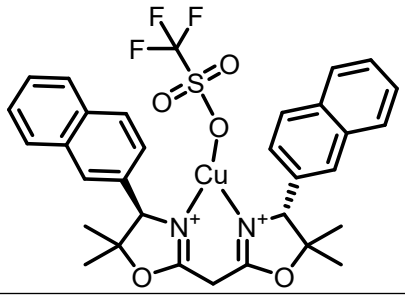
	
C19	C20
	
C21	C22
	
C23	C24
	
C25	C26
	
C27	C28

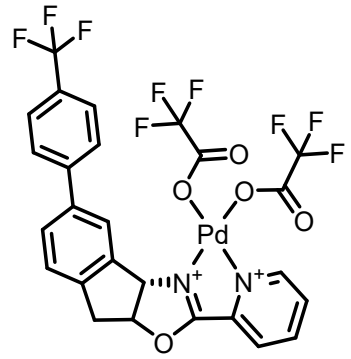
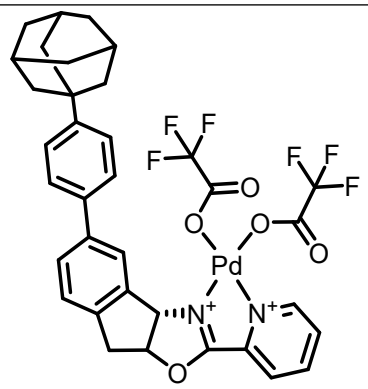
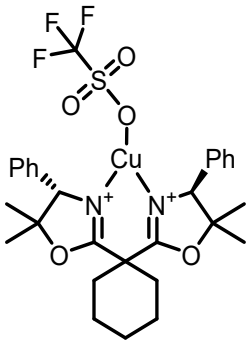
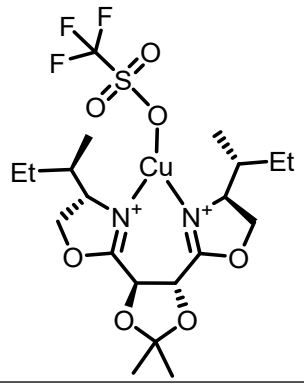
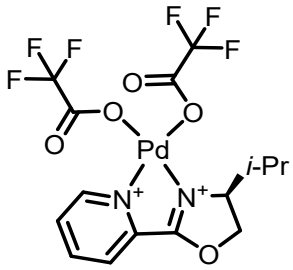
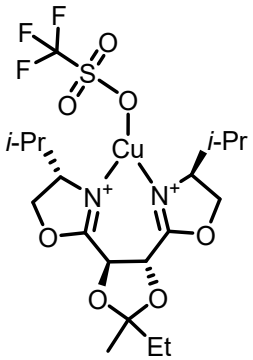
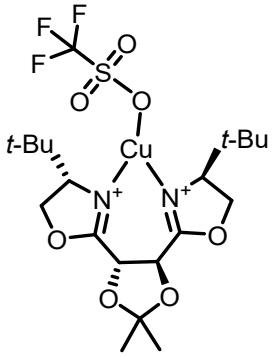
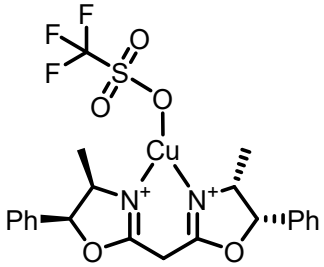
	
C29	C30
	
C31	C32
	
C33	C34
	
C35	C36
	
C37	C38

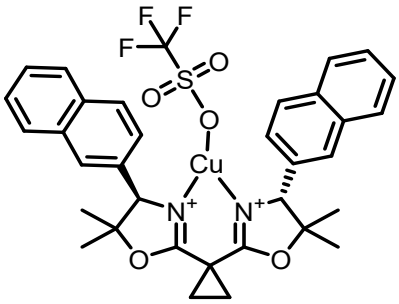
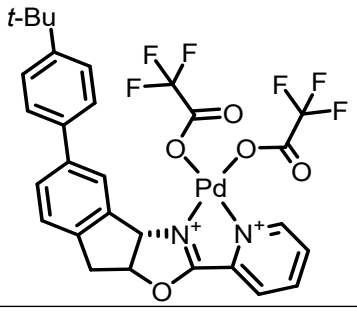
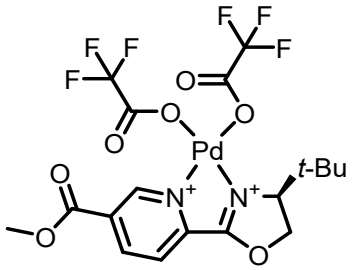
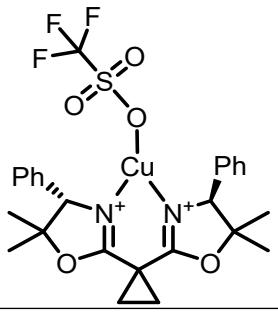
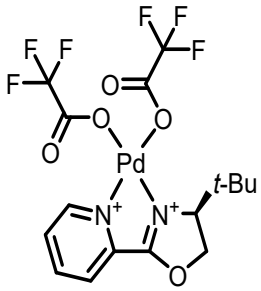
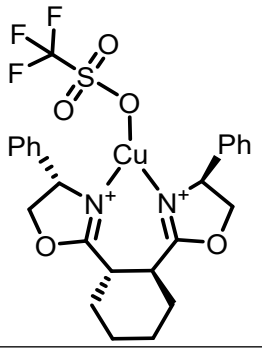
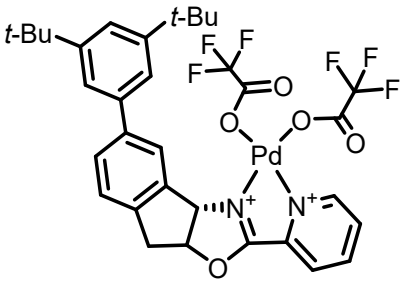
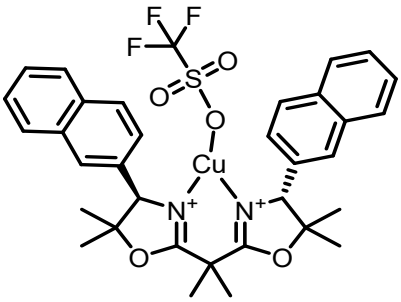
	
C39	C40
	
C41	C42
	
C43	C44
	
C45	C46
	
C47	C48

	
C49	C50
	
C51	C52
	
C53	C54
	
C55	C56
	

C57	C58
	
C59	C60
	
C61	C62
	
C63	C64
	
C65	C66

	
C67	C68
	
C69	C70
	
C71	C72
	
C73	C75
	

C76	C77
	
C78	C79
	
C80	C81
	
C82	C83
	
C84	C85

	
C86	C87
	
C88	C89
	
C90	C91
	
C92	C93

(1.4) Details of the other reaction datasets

In addition to the ART dataset, the performance of our ML models on four benchmark datasets is evaluated in this work. These are Buchwald-Hartwig (BHA) high throughput experimentation (HTE),[1] asymmetric hydrogenation (AH),[2] N,S-acetylation,[3] and USPTO (United States Patent and Trademark Office) dataset.[4] Each dataset is randomly partitioned

into training, validation, and test sets using a 70:10:20 split. Hyperparameters are optimized on the validation set and the best-performing configuration is subsequently evaluated on the test set. More details of each of these datasets are given below.

i) **BHA-HTE:** The BHA-HTE dataset comprises 3955 reactions obtained through HTE setup with near-identical reaction conditions. This Pd-catalyzed C–N cross-coupling space is composed of 1 aliphatic amine nucleophile, 15 alkyl halide electrophiles, a single catalyst precursor, four ligands, three bases, 22 additives, and one solvent.

ii) **AH:** This dataset contains 368 literature reported asymmetric hydrogenation reactions catalyzed by five distinct axially chiral binaphthyl-derived phosphoric acids. It encompasses 190 unique substrates (alkenes and imines) and 58 unique catalysts, representing structurally diverse BINOL- and BINAP-based systems widely used in enantioselective hydrogenation.

iii) **N,S-acetylation:** This dataset comprises enantioselective reactions between imines and thiols catalyzed by axially chiral phosphoric acids (CPAs). The reaction space comprises 5 imines, 5 thiols, and 43 CPA catalysts, resulting in 1075 reaction combinations. In the original report, enantioselectivity is quantified using the corresponding $\Delta\Delta G^{\ddagger}$ values, with a range spanning from -0.4 to +3.2 kcal/mol.

In the present study, we reformulated the prediction task as a direct regression problem on the magnitude of enantiomeric excess (%*ee*) on a 0-100% scale, rather than predicting $\Delta\Delta G^{\ddagger}$ values. To ensure consistency under this representation, reactions with negative reported *ee* values, which denote inversion of the major enantiomer, are removed during data curation. This refinement eliminated 48 reactions, resulting in a final dataset of 1027 entries. The resulting dataset is in line with previously works and hence serves as the benchmark for the N,S-acetylation case study for this work.[5]

iv) **USPTO:** The United States Patent and Trademark Office (USPTO) dataset consists of reactions automatically extracted from patent records and partitioned into gram-scale and subgram-scale subsets by Schwaller *et al.*[6] based on product mass. In this study, we use the gram-scale subset comprising approximately 1.9×10^5 reactions annotated with experimentally reported yields.

(1.5) Different types of featurizations

i) **One-hot encoding**: One-hot encoding (OHE) converts categorical variables into binary vectors that can be processed by machine learning algorithms.[7] In this work, OHE is used to represent reaction components by assigning each reagent a unique position in a binary vector. In the ART dataset, the reagent library includes 93 catalysts, 117 substrates, and 73 alkenes, with each entry mapped to a distinct bit position. Thus, a reaction employing "Catalyst #5" is encoded as [0, 0, 0, 0, 1, 0, …], where a single "1" identifies the selected reagent. This formulation captures only the presence or absence of a component and conveys no additional chemical structure.

ii) **Molecular fingerprints:** In this work, molecular fingerprints are used as numerical representations to encode molecular structures into fixed-length bit vectors for deep neural network (DNN) based prediction of enantiomeric excess (%*ee*) for the ART dataset.[8] Three RDKit-based fingerprinting schemes, such as Morgan (circular) fingerprints with a radius of 2 and 1024 bits, hashed atom-pair fingerprints (1024 bits), and layered fingerprints (1024 bits). Morgan fingerprints capture local atom-centered chemical environments,[9] whereas atom-pair and layered fingerprints encode longer-range topological and connectivity-based structural information.[10] For each molecule, the three fingerprint vectors are concatenated to form a single 3072-bit composite representation. The concatenated fingerprint vector is used as input to the machine learning models for *ee* prediction on the ART dataset.

iii) **SMILES:** SMILES (Simplified Molecular Input Line Entry Specification) is used to represent the molecular structures of all reaction components in the ART dataset, with individual strings assigned to substrates, coupling partners, and catalysts.[11] For each reaction, the SMILES of the participating species are contatenated using a dot (.) notation to create a single reaction-level input. These SMILES representations are used as inputs to sequence-based models (T5Chem, ULMFiT, and a transformer encoder) to predict the reaction enantioselectivity.

iv) **Molecule graph:** Molecular graphs represent molecules as undirected graphs, where atoms are nodes and bonds are edges, optionally enriched with node and edge features.[12] In this work, molecular graphs are used to represent reaction components for graph-based learning, with each molecule converted into a graph using the RDKit–DGL interface.[13] Atom-level and bond-level features are generated using the AttentiveFP atom and bond featurizers.[14] These molecular graphs serve as inputs to the AttentiveFP model for predicting the reaction outcome.

2. Details of various ML models

In this study, several ML methods are employed, spanning from tree-based approaches to deep neural networks (DNN). Thirty distinct splits are taken into consideration, maintaining a train:validation:test ratio of 70:10:20 for each split. Each of these splits are then used for training, with hyperparameter tuning conducted on the validation set. The reported root mean squared error (RMSE) is the average across the 30 random splits, ensuring a comprehensive evaluation of model performance.

(2.1) Tree-based models

A range of machine learning models, including tree-based algorithms such as Random Forest (RF), Decision Tree (DT), and Gradient Boosting (GB) as well as other models like Support

Vector Machines (SVM), were employed as baseline regressor. Binary molecular fingerprints were used as the input representation. We generated three sets of fingerprints for each instance of a reaction, employing diverse methods: Morgan fingerprints with radius 2, hashed atom-pair fingerprints, and layered fingerprints, each comprising 1024 bits. To optimize the hyperparameters, a Python framework called Optuna,[15] was employed to search for the best hyperparameters to improve model performance efficiently.

(i) Decision Tree (DT)

The first tree-based model is DT, a fundamental approach for regression and classification tasks. The default parameters and hyperparameters used in the DT algorithm are outlined in Table S4.

Table S4. List of Parameters used for DT

method	default parameters	hyperparameters
DT	criterion='rmse', splitter='best', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=42, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, presort=False	max_depth: {1 to 5}

Table S5. Performance of DT Algorithm in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameter {max_depth}
1	5.78	9.71	12.32	5
2	5.72	11.86	9.54	5
3	7.87	11.26	10.36	4
4	6.33	11.16	11.64	5
5	7.74	10.39	11.50	4
6	8.23	11.45	11.08	4
7	7.68	9.75	10.18	4
8	7.49	10.30	11.08	5
9	6.49	7.17	10.01	5
10	6.28	6.58	12.12	5
11	10.38	9.35	12.31	3
12	10.41	12.90	12.46	3
13	9.15	10.79	11.93	3
14	10.29	10.39	11.96	3

15	12.64	10.05	13.53	2
16	5.73	9.89	10.60	5
17	6.25	10.30	9.62	5
18	7.15	10.82	7.44	5
19	6.33	11.59	9.64	5
20	9.27	12.61	13.29	3
21	6.91	7.54	10.23	5
22	7.62	18.95	9.52	4
23	6.99	8.12	9.01	5
24	7.71	12.66	12.17	4
25	6.50	11.14	13.17	5
26	6.88	11.39	9.84	5
27	6.62	9.89	10.97	5
28	6.52	10.55	10.31	5
29	9.61	12.12	9.98	3
30	5.63	8.18	14.25	5
avg.±s.d.	7.61±1.71	10.63±2.23	11.07±1.53	

(ii) Random Forest (RF)

RF, an ensemble learning technique proposed by Breiman,[16] aggregates multiple decision trees to enhance predictive performance. The default parameters and hyperparameters used in the RF algorithm are specified in Table S6.

Table S6. List of Parameters used for the RF

method	default parameters	hyperparameters
RF	bootstrap=True, criterion='rmse', max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_jobs=1, oob_score=False, random_state=42, verbose=0, warm_start=False	n_estimators: {1 to 100}, max_depth: {1 to 100}

Table S7. Performance of RF Algorithm in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {n_estimators, max_depth}
1	3.694	6.823	10.513	38,30
2	4.068	12.709	9.254	8,17
3	3.641	7.9	7.998	99,15
4	3.513	10.039	8.178	95,18
5	4.054	8.113	7.886	34,28
6	3.459	8.836	8.591	10,51
7	3.319	7.106	8.253	84,25

8	3.369	7.413	8.202	77,71
9	3.357	7.093	9.21	73,45
10	3.61	6.227	8.663	66,71
11	3.706	10.057	7.537	10,97
12	3.617	7.252	8.48	22,51
13	3.441	9.601	9.793	25,62
14	4.561	7.988	10.37	5,19
15	3.861	8.606	8.608	21,70
16	3.64	9.12	7.982	26,23
17	4.593	8.973	9.339	5,67
18	4.718	10.132	6.613	47,7
19	3.855	9.424	7.691	10,68
20	3.425	7.639	11.229	22,99
21	3.459	6.014	8.043	18,57
22	7.543	15.049	10.533	1,60
23	3.582	7.603	7.839	69,22
24	3.305	7.964	9.437	40,96
25	4.972	10.54	10.423	4,20
26	5.136	9.5	10.645	3,100
27	4.186	8.225	8.286	93,9
28	3.331	8.879	8.979	100,62
29	3.599	9.389	6.885	41,82
30	3.23	7.494	9.671	100,26
avg.±s.d.	3.93±0.86	8.72±1.86	8.84±1.17	

(iii) Gradient Boosting (GB)

Gradient Boosting (GB), an ensemble learning method combining the principles of bagging and boosting, was employed in this study. While bagging trains multiple models in parallel and aggregates their predictions to reduce variance and improve stability, boosting builds models sequentially, where each subsequent tree attempts to correct the errors of its predecessors, thereby iteratively enhancing overall predictive performance. The default parameters and hyperparameters used in the GB algorithm are detailed in Table S8.

Table S8. List of Parameters used for the GB

method	default parameters	hyperparameters
GB	loss='huber', learning_rate=0.1, criterion='mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, min_impurity_decrease=0.0, min_impurity_split=None, init=None, random_state=42, max_features=None, verbose=0, max_leaf_nodes=None, warm_start=False, presort='auto'	n_estimators: {1 to 100} max_depth: {1 to 5}

Table S9. Performance of GB Algorithm in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {n_estimators, max_depth}
1	1.52	6.25	10.93	91,5
2	5.15	11.66	8.73	94,2
3	3.33	7.28	7.24	50,4
4	4.75	9.03	7.69	54,3
5	2.43	7.68	7.80	100,4
6	6.36	7.74	9.52	62,2
7	3.62	7.54	7.54	100,3
8	5.26	7.94	9.04	100,2
9	2.75	6.39	8.46	53,5
10	1.99	5.59	9.33	61,5
11	5.69	8.36	8.60	75,2
12	2.72	6.70	7.79	97,4
13	4.96	9.13	9.61	24,4
14	3.85	7.18	8.47	80,3
15	6.69	7.75	8.70	55,2
16	3.79	8.06	7.53	85,3
17	2.35	6.75	8.64	100,4
18	3.70	8.58	6.35	99,3
19	1.78	9.80	9.84	98,5
20	3.48	7.14	11.44	99,3
21	2.31	5.82	8.59	58,5
22	2.79	8.30	7.40	99,4
23	1.75	8.15	7.72	98,5
24	6.01	6.54	8.75	64,2
25	3.54	7.48	9.03	97,3
26	5.97	8.26	9.12	69,2
27	3.57	8.74	8.51	86,3
28	3.70	8.59	8.65	100,3
29	3.78	8.21	7.75	92,3
30	3.78	7.39	9.41	90,3
avg.±s.d.	3.78±1.45	7.80±1.23	8.61±1.07	

(iv) Support Vector Machine (SVM)

SVMs are powerful for classification and regression tasks, leveraging kernel functions to map data into high-dimensional spaces for effective modeling. The default parameters and hyperparameters applied in the SVM algorithm are listed in Table S10.

Table S10. List of Parameters used for SVM

method	default parameters	hyperparameters
SVM	degree=3, gamma='scale', coef0=0.0, tol=0.001,	epsilon: {e^{-2} to e^{1}},

	shrinking=True, cache_size=200, verbose=False, max_iter=-1	kernel:{linear, rbf, ploy}, C:{1 to 30}

Table S11. Performance of SVM Algorithm in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {C, epsilon, kernel}
1	4.05	7.29	10.79	1,5.1,linear
2	2.55	9.55	7.38	2,2.9,linear
3	4.16	8.45	7.75	15,4.9,linear
4	2.55	7.28	8.87	5,2.6,linear
5	4.74	10.03	8.49	1,6.0,linear
6	4.57	8.39	7.74	5,5.6,linear
7	3.10	8.16	8.15	3,3.5,linear
8	3.38	8.59	8.93	23,4.0,linear
9	3.66	7.90	8.11	2,4.3,linear
10	3.36	7.43	8.76	6,4.1,linear
11	3.86	9.66	8.63	30,0.0,rbf
12	4.95	8.45	12.37	30,0.0,poly
13	5.17	9.99	9.41	11,6.7,linear
14	4.54	7.12	9.29	3,5.7,linear
15	4.89	8.35	8.94	1,6.3,linear
16	5.58	8.72	8.64	2,7.1,linear
17	4.45	7.01	8.44	3,5.4,linear
18	3.02	8.20	9.09	20,3.4,linear
19	3.81	8.95	10.39	8,4.5,linear
20	1.70	7.31	10.99	5,1.4,linear
21	2.84	5.42	8.32	30,3.2,linear
22	3.29	6.97	7.27	23,3.7,linear
23	5.22	8.34	7.89	19,6.6,linear
24	5.84	7.28	9.86	26,7.4,linear
25	5.66	8.03	9.20	16,7.6,linear
26	5.95	9.01	8.95	8,8.0,linear
27	4.51	8.71	8.03	3,5.8,linear
28	3.20	8.27	9.38	19,3.6,linear
29	2.67	8.02	8.41	3,2.8,linear
30	3.80	7.84	9.89	18,4.6,linear
avg.±s.d.	4.04±1.10	8.16±0.99	8.95±1.12	

(2.2) Deep neural networks (DNN)

A DNN (deep neural network) comprises of multiple fully connected input layers, one or more hidden layers, and a single output layer. Usually, a DNN is trained using a supervised learning backpropagation algorithm, with the Adam optimization technique to update and

adjust the weights and biases of the neurons. Various hyperparameters, including the number of hidden layers, number of neurons, learning rate, and dropout rate, influence the performance of a DNN. It is crucial to determine the best combination of these hyperparameters in order to achieve good accuracy. We employed Optuna for hyperparameter tuning, and the optimal hyperparameters are selected based on the lowest validation RMSE. The default parameters and hyperparameters used in the DNN algorithm are listed in Table S12.

Table S12. List of parameters used for DNN

method	default parameters	hyperparameters
DNN	num_epochs = 300	n_layers: {1 to 5}, n_neurons: {1 to 100}, dropout_rate: {0.0 to 0.9}, learning_rate: { e^{-5} to e^{-1}}

Table S13. Performance of DNN Algorithm in terms of RMSE across Different Splits using Fingerprint as a Featurization Technique

split	train	validation	test	best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
1	4.35	8.16	12.51	1,66,0.10,0.09841
2	5.16	10.08	8.96	1,47,0.23,0.02825
3	4.97	7.86	7.69	1,87,0.34,0.00780
4	4.07	9.53	8.28	3,70,0.08,0.00953
5	3.89	8.18	9.55	1,86,0.08,0.02530
6	6.11	7.52	9.46	1,89,0.59,0.00522
7	3.64	8.60	8.29	1,74,0.05,0.09685
8	5.89	6.70	10.24	2,50,0.45,0.04353
9	5.91	6.62	9.77	3,47,0.09,0.01052
10	7.06	9.23	9.62	3,27,0.09,0.03964
11	7.21	7.66	8.63	1,100,0.68,0.00351
12	6.21	5.98	9.93	1,59,0.37,0.01642
13	5.27	9.82	10.35	1,88,0.36,0.00639
14	4.65	8.88	10.28	3,94,0.17,0.02302
15	6.17	9.36	8.58	3,68,0.32,0.00796
16	6.73	9.96	9.54	3,95,0.31,0.02816
17	2.62	6.57	10.51	2,66,0.01,0.04427
18	10.59	13.22	10.78	4,92,0.34,0.01210
19	6.16	8.22	9.13	2,46,0.35,0.01344

20	6.30	9.42	14.14	1,29,0.06,0.00918
21	6.04	5.27	8.74	1,44,0.09,0.01206
22	4.51	6.84	7.97	2,45,0.06,0.01395
23	5.11	8.39	8.66	1,88,0.22,0.02318
24	6.94	6.15	10.28	1,49,0.64,0.01869
25	5.81	8.65	8.36	1,54,0.48,0.09312
26	6.63	9.10	9.77	1,51,0.28,0.00563
27	4.45	8.57	8.32	1,76,0.13,0.02709
28	6.19	7.87	9.68	1,72,0.43,0.00774
29	3.54	9.59	8.64	1,93,0.04,0.05998
30	3.92	7.13	9.87	1,51,0.08,0.03893
avg.±s.d.	5.54±1.50	8.30±1.56	9.55±1.31	

Table S14. Performance of DNN Algorithm in terms of RMSE across Different Splits using OHE as a Featurization Technique

split	train	validation	test	best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
1	2.80	9.81	19.99	5,72,0.30,0.01766
2	9.49	16.33	13.84	2,68,0.89,0.08072
3	2.76	15.69	16.04	1,68,0.41,0.00420
4	3.85	14.19	11.35	4,77,0.56,0.00725
5	3.56	13.66	12.92	1,70,0.48,0.00369
6	1.45	15.84	9.43	2,98,0.09,0.00408
7	4.87	12.59	12.17	3,76,0.13,0.00238
8	1.87	10.74	15.19	1,83,0.37,0.05603
9	3.45	12.64	15.38	1,27,0.17,0.00527
10	4.37	15.13	13.86	4,65,0.58,0.03504
11	4.58	9.50	12.00	1,58,0.18,0.00281
12	2.83	12.35	15.78	1,52,0.25,0.00567
13	4.61	12.25	13.59	1,74,0.57,0.00286
14	2.98	10.70	13.90	3,9,0.03,0.09172
15	3.00	10.70	14.68	1,85,0.52,0.00469
16	4.67	15.38	11.33	1,72,0.45,0.00269
17	4.61	15.65	15.21	3,59,0.68,0.01477
18	1.49	11.74	11.46	1,72,0.07,0.00418
19	1.98	13.32	21.11	4,86,0.36,0.01402
20	1.77	11.41	18.89	1,45,0.18,0.00728
21	5.07	8.18	14.13	3,34,0.55,0.01222
22	3.87	12.99	11.70	1,70,0.63,0.05117
23	3.52	10.71	11.60	2,59,0.67,0.02403
24	2.78	11.22	13.87	3,57,0.46,0.01306
25	9.36	18.85	20.48	5,49,0.61,0.01331
26	5.72	14.25	11.90	1,32,0.60,0.00920
27	3.31	10.98	14.75	1,80,0.62,0.00677
28	2.10	12.14	15.73	1,92,0.50,0.06206

29	2.56	14.96	10.48	3,95,0.49,0.00539
30	0.93	7.27	20.15	3,65,0.02,0.01035
avg.±s.d.	3.67±1.94	12.7±2.55	14.43±3.05	

DNN-CI

To address the issue of class imbalance, we have built several DNN-CI model with majority-minority class-imbalance (CI) loss function (see Fig. 3). When training the models using the Class-Imbalanced (CI) loss function, various class boundaries are defined to split the dataset into a majority and a minority groups. These boundaries include fixed integer values (30, 40, 50, 60, 70) as well as statistically derived thresholds from the %*ee* distribution found in the ART dataset, specifically the mean μ (76) and μ – σ (54). Different models are built based on different class boundaries, and their performances are summarised in the following tables.

Table S15. Performance of DNN-CI Algorithm with a Class Boundary of 40 in terms of RMSE across Different Splits using Fingerprint as a Featurization Technique

split	train	validation	test	best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
1	3.70	8.34	12.31	1,100,0.12,0.04458
2	6.91	9.58	8.98	3,24,0.37,0.01774
3	4.66	7.44	7.60	1,92,0.26,0.00406
4	3.98	9.70	8.08	2,86,0.15,0.03008
5	6.56	8.50	9.62	1,21,0.28,0.01285
6	10.85	11.76	14.53	3,75,0.01,0.09898
7	5.49	8.97	7.50	2,91,0.44,0.00598
8	5.99	6.22	9.26	2,93,0.43,0.04962
9	5.16	6.64	9.73	3,35,0.08,0.04526
10	4.13	7.58	9.26	2,85,0.19,0.00963
11	6.25	8.44	8.07	1,83,0.72,0.01586
12	5.83	6.39	9.96	1,60,0.50,0.01475
13	5.88	9.36	10.12	1,90,0.73,0.01509
14	3.31	6.90	10.08	2,58,0.00,0.05323
15	6.29	9.21	9.69	2,41,0.07,0.08005
16	7.41	8.27	8.08	5,63,0.06,0.00493
17	4.02	7.31	10.72	2,100,0.09,0.01536
18	2.70	8.88	8.09	2,64,0.00,0.04338
19	6.34	7.40	8.50	2,46,0.31,0.01085
20	4.15	9.87	11.36	2,23,0.02,0.05704
21	5.50	5.50	8.38	3,65,0.28,0.01335
22	5.68	8.33	8.23	3,41,0.20,0.03618

23	6.32	8.25	9.52	1,10,0.12,0.07858
24	4.80	6.64	9.37	1,29,0.17,0.02577
25	4.38	8.99	8.09	1,35,0.11,0.01721
26	6.15	9.12	8.47	4,90,0.16,0.01365
27	6.20	8.71	9.21	1,84,0.59,0.02333
28	5.81	7.27	8.91	1,74,0.40,0.00532
29	4.26	9.92	8.36	1,96,0.00,0.01180
30	3.17	7.21	9.67	1,82,0.01,0.04462
avg.±s.d.	5.40±1.56	8.22±1.33	9.33±1.45	

Table S16. Performance of DNN-CI Algorithm with a Class Boundary of 50 in terms of RMSE across Different Splits using Fingerprint as a Featurization Technique

split	train	validation	test	best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
1	6.12	8.72	15.75	5,35,0.07,0.01827
2	3.93	10.05	8.61	1,68,0.08,0.02101
3	3.38	7.61	8.15	1,77,0.09,0.07824
4	3.72	9.75	7.89	1,57,0.15,0.09709
5	3.91	7.89	9.38	1,95,0.10,0.01401
6	8.64	9.13	11.56	1,10,0.20,0.02884
7	3.26	7.98	8.13	1,93,0.09,0.04263
8	3.19	7.21	9.82	2,80,0.13,0.01815
9	5.82	7.38	9.90	2,88,0.48,0.02097
10	5.08	7.74	9.83	2,25,0.07,0.02237
11	5.56	7.67	7.62	1,73,0.46,0.02222
12	6.00	6.14	10.13	1,47,0.40,0.02210
13	4.45	10.20	10.23	1,44,0.23,0.09898
14	5.13	8.36	10.59	3,52,0.07,0.09997
15	5.17	9.31	8.54	1,89,0.58,0.00760
16	6.92	8.26	8.68	4,85,0.47,0.01574
17	4.52	7.26	10.08	2,38,0.02,0.00696
18	5.57	9.61	6.83	2,57,0.11,0.03150
19	5.65	8.43	9.69	2,69,0.21,0.08802
20	3.80	8.78	12.50	2,75,0.13,0.00905
21	5.32	4.99	8.32	1,65,0.29,0.01031
22	3.69	7.17	8.13	2,92,0.08,0.01479
23	4.58	7.87	8.50	1,95,0.42,0.04917
24	5.41	7.05	9.83	1,81,0.40,0.00685
25	5.84	10.73	9.60	4,37,0.08,0.01155
26	3.90	8.90	8.43	1,87,0.05,0.00665
27	4.03	8.86	8.61	1,67,0.18,0.03961
28	5.82	6.80	9.61	1,21,0.11,0.01180
29	5.50	10.18	9.23	1,51,0.38,0.06982
30	5.03	8.08	10.30	3,79,0.05,0.00445
avg.±s.d.	4.96±1.18	8.27±1.27	9.48±1.65	

Table S17. Performance of DNN-CI Algorithm with a Class Boundary of 60 in terms of RMSE across Different Splits using Fingerprint as a Featurization Technique

split	train	validation	test	best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
1	7.75	8.37	16.22	4,81,0.41,0.00674
2	6.10	9.93	8.94	2,46,0.55,0.01402
3	5.00	7.76	7.64	2,44,0.27,0.00790
4	4.53	8.66	8.20	2,61,0.04,0.02711
5	4.74	8.59	9.53	1,47,0.36,0.05544
6	5.22	7.24	9.94	1,48,0.10,0.01461
7	6.31	9.14	8.16	1,51,0.61,0.05792
8	4.95	7.31	9.09	2,77,0.40,0.04329
9	6.69	6.80	10.73	4,94,0.40,0.01134
10	2.65	6.45	9.01	2,70,0.01,0.02177
11	6.80	7.47	8.20	1,77,0.70,0.00838
12	5.25	5.63	9.22	2,74,0.27,0.01839
13	6.41	9.63	11.34	1,47,0.54,0.01594
14	3.71	8.55	9.77	3,51,0.01,0.00810
15	7.00	9.92	8.40	2,89,0.66,0.02466
16	6.89	7.99	8.55	5,93,0.16,0.00619
17	5.30	7.77	10.15	2,72,0.32,0.03167
18	3.60	8.81	7.76	2,92,0.08,0.01480
19	6.63	8.02	8.73	2,87,0.60,0.02370
20	3.78	8.37	12.40	1,29,0.01,0.01182
21	5.64	4.84	8.80	1,34,0.07,0.01285
22	3.77	7.08	8.02	2,99,0.23,0.02447
23	6.35	7.62	9.73	1,78,0.57,0.09979
24	6.37	7.04	9.89	1,77,0.61,0.01587
25	5.22	8.97	9.02	4,51,0.01,0.00512
26	4.84	8.59	8.80	1,73,0.15,0.00921
27	4.95	9.10	9.34	1,37,0.17,0.01936
28	5.91	7.49	9.11	1,79,0.36,0.00399
29	5.22	9.72	8.35	1,38,0.28,0.05824
30	3.19	6.81	10.17	1,42,0.04,0.09396
avg.±s.d.	5.36±1.24	7.99±1.20	9.44±1.63	

Table S18. Performance of DNN-CI Algorithm with a Class Boundary of 70 in terms of RMSE across Different Splits using Fingerprint as a Featurization Technique

split	train	validation	test	best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
1	4.13	10.20	13.42	2,51,0.03,0.00788
2	3.00	10.20	8.82	1,55,0.00,0.09836
3	6.63	8.13	7.80	3,95,0.43,0.00537

4	2.90	8.94	7.73	2,57,0.01,0.02568
5	5.31	8.27	9.06	1,73,0.45,0.01939
6	7.35	7.30	11.25	4,95,0.36,0.01705
7	5.01	8.23	7.79	2,44,0.35,0.01430
8	4.57	5.90	10.50	2,68,0.28,0.00910
9	5.31	6.71	10.30	2,91,0.38,0.02661
10	5.21	7.39	9.42	2,99,0.10,0.02488
11	7.55	7.55	8.76	1,80,0.78,0.00704
12	6.53	6.37	10.27	1,36,0.54,0.02419
13	7.58	10.21	11.54	1,44,0.61,0.03641
14	4.22	7.68	11.31	2,14,0.00,0.09080
15	8.90	8.80	9.87	2,95,0.83,0.00538
16	3.69	8.73	8.78	3,62,0.06,0.02851
17	3.11	7.52	9.71	1,94,0.01,0.05577
18	5.78	9.82	6.90	1,89,0.22,0.00342
19	4.22	7.91	8.39	2,91,0.33,0.03047
20	5.78	10.37	14.21	3,23,0.11,0.01928
21	4.76	4.61	8.21	1,87,0.17,0.00731
22	3.48	7.27	8.67	2,37,0.01,0.01706
23	9.51	9.90	11.22	3,32,0.08,0.00568
24	6.53	6.46	9.82	1,80,0.61,0.00419
25	4.67	8.29	7.76	1,77,0.21,0.01683
26	5.90	8.97	8.67	4,17,0.02,0.01828
27	5.73	9.01	9.61	1,100,0.40,0.04114
28	5.30	8.09	9.24	1,9,0.15,0.04465
29	6.44	11.18	8.57	4,78,0.29,0.01210
30	9.26	11.16	12.02	1,17,0.17,0.07186
avg.±s.d.	5.61±1.75	8.37±1.55	9.65±1.68	

Table S19. Performance of DNN-CI Algorithm with a Class Boundary of 54 as Defined by (μ – σ) of the Distribution of %*ee* in terms of RMSE across Different Splits using Fingerprint as a Featurization Technique

split	train	validation	test	best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
1	6.80	8.45	14.68	1,73,0.33,0.00285
2	6.39	10.27	8.44	2,99,0.69,0.02680
3	2.75	7.75	8.15	1,100,0.01,0.07740
4	3.40	9.74	7.59	3,69,0.07,0.01717
5	6.44	8.33	10.29	2,15,0.33,0.06087
6	10.90	9.32	13.94	5,36,0.31,0.02237
7	5.53	8.43	7.32	2,35,0.16,0.00843
8	5.19	6.56	9.25	2,48,0.39,0.03477
9	5.60	7.08	9.73	2,60,0.37,0.04780
10	5.04	7.29	8.88	2,45,0.25,0.01099
11	8.86	8.24	9.91	1,70,0.71,0.00322

12	6.26	6.09	9.75	2,21,0.10,0.01829
13	6.92	10.07	11.63	1,21,0.30,0.01718
14	4.63	7.31	10.01	2,27,0.04,0.08632
15	5.75	9.23	9.46	2,56,0.38,0.03154
16	5.66	9.62	8.78	2,53,0.47,0.04501
17	3.86	7.13	10.25	2,54,0.08,0.00663
18	2.92	8.54	9.31	1,55,0.02,0.09465
19	6.13	7.70	7.98	2,22,0.25,0.02064
20	5.84	9.39	11.92	3,79,0.43,0.01401
21	5.99	4.98	8.70	1,49,0.09,0.01041
22	5.78	8.02	8.35	2,88,0.18,0.09179
23	5.21	7.96	8.67	1,65,0.28,0.05738
24	6.51	6.73	9.95	1,84,0.67,0.01125
25	6.29	9.10	8.10	1,92,0.53,0.03741
26	7.14	9.03	9.60	5,90,0.14,0.01194
27	7.05	9.19	10.34	1,36,0.49,0.08295
28	6.58	8.33	9.60	1,83,0.62,0.09075
29	4.63	10.07	8.72	1,70,0.28,0.01890
30	3.64	7.24	9.99	1,100,0.07,0.01158
avg.±s.d.	5.61±1.75	8.37±1.55	9.65±1.68	

Table S20. Performance of DNN-CI Algorithm with a Class Boundary of 76 as Defined by μ of the Distribution of %*ee* in terms of RMSE across Different Splits using Fingerprint as a Featurization Technique

split	train	validation	test	best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
1	5.22	11.76	13.92	2,7,0.04,0.05787
2	5.20	10.06	9.15	1,88,0.37,0.02494
3	3.50	7.35	7.90	1,88,0.07,0.04514
4	3.95	9.63	7.79	3,91,0.03,0.00544
5	4.58	8.04	9.38	1,53,0.07,0.01834
6	6.92	7.45	9.47	4,70,0.40,0.01352
7	6.67	8.63	8.37	1,93,0.71,0.03088
8	5.90	6.88	10.90	2,99,0.41,0.04034
9	5.63	6.99	9.71	4,61,0.03,0.00607
10	5.35	7.47	9.55	3,100,0.07,0.07144
11	6.41	7.37	8.04	1,97,0.70,0.01723
12	4.84	5.37	9.21	1,46,0.11,0.01355
13	7.41	10.19	11.60	1,74,0.69,0.02420
14	5.04	7.68	9.49	3,56,0.01,0.01901
15	6.50	9.39	8.25	4,36,0.18,0.01312
16	6.43	11.99	8.84	2,89,0.37,0.04087
17	4.01	7.23	9.70	1,98,0.09,0.00860
18	5.49	9.77	6.81	1,88,0.20,0.00346
19	5.45	8.48	8.40	4,86,0.09,0.00546

20	3.69	8.46	12.74	1,75,0.08,0.02047
21	5.15	4.58	8.46	1,85,0.38,0.00781
22	5.19	7.14	8.18	2,67,0.15,0.01654
23	5.59	7.45	9.11	1,85,0.39,0.03075
24	9.39	7.66	11.05	2,51,0.69,0.00472
25	4.04	8.40	8.10	1,84,0.17,0.06428
26	3.92	10.41	8.48	2,68,0.19,0.01781
27	6.32	8.89	9.81	1,81,0.58,0.05923
28	6.45	7.20	9.34	1,44,0.38,0.00961
29	3.86	9.10	8.66	1,56,0.10,0.03766
30	4.26	7.67	9.62	2,88,0.26,0.00640
avg.±s.d.	5.61±1.75	8.37±1.55	9.65±1.68	

(2.3) Transformer-based NLP models

(i) Yield BERT

Yield-BERT[6] leverages transformer-based NLP architectures, which have revolutionized natural language processing. In this case, the model processes tokenized sequences from the SMILES string representions of chemical reactions. By learning significant representations of molecules, it fine-tunes labelled data to predict the %*ee* of the reaction. The default hyperparameters (as listed in Table S21) are employed for training.

Table S21. Hyperparameters of Yield-BERT

hyperparameters	values
num_train_epochs	15
learning rate	0.00009659
gradient_accumulation_steps	1
evaluate_during_training	True
manual_seed	42
max_seq_length	300
train_batch_size	16
warmup_ratio	0.0
hidden_dropout_prob	0.7987

Table S22. Performance of Yield-BERT Algorithm in terms of RMSE across Different Splits

split	train	validation	test
1	4.15	10.78	10.99
2	5.00	11.29	11.47
3	2.81	12.03	8.40
4	8.01	10.91	12.67

5	8.46	8.40	12.52
6	3.94	12.06	9.67
7	2.95	9.85	8.94
8	7.12	16.89	10.14
9	8.34	9.14	10.61
10	2.60	7.93	10.31
11	2.52	8.76	8.00
12	6.30	7.35	11.61
13	7.60	9.81	9.97
14	5.19	10.21	11.39
15	5.45	5.92	10.78
16	1.10	10.41	10.29
17	4.50	10.16	9.51
18	3.17	12.12	10.15
19	5.49	10.74	10.11
20	6.93	8.61	10.31
21	3.27	5.49	7.97
22	4.23	9.10	9.99
23	3.75	10.55	7.55
24	5.10	9.37	11.23
25	7.50	11.45	9.81
26	5.79	10.20	11.66
27	5.82	8.87	11.15
28	5.64	10.64	10.83
29	2.38	10.51	8.85
30	4.27	8.47	11.74
avg.±s.d.	4.84±1.96	9.78±2.09	10.10±1.30

(ii) T5Chem

T5Chem[17] is a unified deep learning model built upon the Text-to-Text Transfer Transformer (T5) architecture, which has found successf in various NLP tasks. First, T5Chem is pre-trained using ~1.2B molecules from the PubChem databse. Then, it is fine-tuned for various tasks, including forward reaction prediction, retrosynthesis, reagents prediction, reaction type classification, reaction yield prediction, etc. Herein, we have fine-tuned the T5Chem for enantioselectivity predictions for the target dataset of interest (i.e., the ART dataset). The default hyperparameters (as listed in Table S23) are used for training.

Table S23. Hyperparameters of T5Chem

hyperparameters	values

tokenizer	simple
random_seed	8570
num_epoch	100
log_step	100
batch_size	64
init_lr	$5e^{-4}$

Table S24. Performance of the T5Chem Algorithm in terms of RMSE across Different Splits

split	train	validation	test
1	6.38	8.73	15.58
2	6.88	10.23	8.96
3	6.69	9.53	8.50
4	6.37	9.77	11.01
5	7.48	11.10	12.84
6	6.52	10.34	10.84
7	7.23	11.76	9.34
8	7.12	10.98	10.42
9	6.00	7.48	10.52
10	6.47	7.36	10.13
11	7.24	11.09	9.44
12	7.18	7.74	10.55
13	6.33	9.44	10.63
14	6.98	9.30	11.99
15	7.32	8.89	9.04
16	6.15	10.32	9.42
17	6.45	8.30	11.16
18	7.10	9.95	7.65
19	7.19	8.55	14.43
20	6.72	9.05	15.06
21	6.28	9.25	9.66
22	6.75	10.62	11.91
23	6.41	9.37	10.83
24	6.96	12.80	11.00
25	6.95	9.80	10.73
26	6.34	11.13	10.57
27	6.30	9.94	11.68
28	7.03	10.33	10.42
29	6.34	11.54	9.61
30	7.07	10.05	10.94
avg.±s.d.	6.74±0.39	9.82±1.23	10.83±1.73

T5Chem-CI

T5Chem-CI is a novel method employing a majority-minority class-aware loss (see Fig. 3) to address the issue of class imbalance. Models with different class boundaries are developed, and their performances are listed in the subsequent tables.

Table S25. Performance of T5Chem-CI Algorithm with a Class Boundary of 40 in terms of RMSE across Different Splits

split	train	validation	test
1	5.71	9.21	16.48
2	4.70	11.25	9.30
3	5.28	8.72	9.60
4	5.66	11.41	10.82
5	4.74	9.48	8.89
6	5.43	9.83	9.37
7	5.45	11.10	11.10
8	4.97	17.23	9.63
9	5.16	7.44	11.22
10	4.74	9.95	10.49
11	5.51	11.73	9.54
12	5.07	7.93	9.41
13	5.13	10.36	10.16
14	5.43	8.91	12.40
15	5.39	9.11	8.90
16	5.00	9.43	9.11
17	4.68	9.68	9.98
18	5.18	8.95	7.08
19	5.67	10.60	15.55
20	5.11	8.86	15.82
21	5.82	10.09	10.12
22	5.49	13.93	11.69
23	5.29	10.73	8.81
24	4.80	12.87	10.71
25	5.47	11.17	12.20
26	4.76	12.24	10.50
27	5.57	9.77	11.80
28	5.24	11.39	11.64
29	5.42	13.05	8.78
30	5.22	8.92	11.34
avg.±s.d.	5.24±0.32	10.51±1.95	10.75±2.10

Table S26. Performance of T5Chem-CI Algorithm with a Class Boundary of 50 in terms of RMSE across Different Splits

split	train	validation	test
1	4.46	7.84	15.30
2	5.34	11.64	10.20
3	4.70	11.91	9.61
4	4.69	12.76	9.91
5	5.35	10.13	10.04
6	5.71	10.12	10.13

7	4.86	11.29	9.58
8	5.69	15.78	10.06
9	5.09	7.53	12.03
10	5.26	10.31	11.55
11	5.20	10.77	9.28
12	4.72	7.76	9.76
13	5.54	10.57	9.99
14	4.78	12.33	11.45
15	5.10	10.12	7.63
16	5.00	9.15	9.61
17	4.63	8.44	9.51
18	5.73	10.41	7.79
19	5.45	8.82	13.57
20	4.96	8.97	15.74
21	5.52	9.14	9.18
22	5.52	13.15	11.53
23	5.11	10.11	8.34
24	5.35	12.76	10.88
25	5.23	9.96	9.65
26	4.72	10.89	11.59
27	5.13	9.79	11.68
28	5.07	10.29	10.28
29	5.38	13.72	8.42
30	4.87	9.50	11.80
avg.±s.d.	5.14±0.34	10.53±1.84	10.54±1.86

Table S27. Performance of T5Chem-CI Algorithm with a Class Boundary of 60 in terms of RMSE across Different Splits

split	train	validation	test
1	5.59	7.96	16.36
2	5.66	11.57	11.65
3	5.78	8.54	8.30
4	5.66	12.21	9.68
5	5.41	11.32	10.68
6	5.46	8.84	10.07
7	4.72	9.94	9.44
8	5.91	17.75	10.91
9	5.91	17.75	10.91
10	5.40	10.08	12.01
11	5.07	11.91	9.33
12	5.70	7.47	10.01
13	5.64	9.57	10.50
14	5.56	9.22	11.94
15	5.44	10.68	9.27
16	4.99	9.91	8.98
17	5.67	10.41	10.17
18	5.53	9.43	8.78

19	5.92	8.38	13.05
20	5.21	8.07	15.94
21	6.08	8.80	10.67
22	4.93	8.97	10.87
23	5.34	10.14	10.04
24	5.54	12.85	11.74
25	5.36	10.96	10.36
26	4.83	10.53	10.22
27	5.85	9.88	12.37
28	5.51	9.86	11.51
29	5.62	10.17	9.30
30	5.22	9.43	11.25
avg.±s.d.	5.48±0.33	10.42±2.32	10.88±1.79

Table S28. Performance of T5Chem-CI Algorithm with a Class Boundary of 70 in terms of RMSE across Different Splits

split	train	validation	test
1	5.51	8.73	15.58
2	6.03	10.23	8.96
3	5.92	9.53	8.50
4	5.71	9.77	11.01
5	6.63	11.10	12.84
6	5.81	10.34	10.84
7	6.56	11.76	9.34
8	6.29	10.98	10.42
9	5.21	7.48	10.52
10	5.74	7.36	10.13
11	6.42	11.09	9.44
12	6.29	7.74	10.55
13	5.60	9.44	10.63
14	6.22	9.30	11.99
15	6.59	8.89	9.04
16	5.50	10.32	9.42
17	5.78	8.30	11.16
18	6.34	9.95	7.65
19	6.50	8.55	14.43
20	5.96	9.05	15.06
21	5.54	9.25	9.66
22	5.92	10.62	11.91
23	5.67	9.37	10.83
24	6.11	12.80	11.00
25	6.22	9.80	10.73
26	5.54	11.13	10.57
27	5.58	9.94	11.68
28	6.31	10.33	10.42
29	5.65	11.54	9.61
30	6.21	10.05	10.94

avg.±s.d.	5.98±0.38	9.82±1.25	10.83±1.76

Table S29. Performance of T5Chem-CI Algorithm with a Class Boundary of 54 as Defined by μ–σ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test
1	4.93	8.73	15.58
2	5.28	10.58	10.69
3	5.02	11.60	8.74
4	4.98	12.44	9.68
5	4.83	11.32	9.86
6	5.29	8.85	9.62
7	5.20	10.74	10.16
8	4.88	17.25	9.62
9	5.23	7.28	10.41
10	4.83	12.22	11.33
11	5.42	11.51	10.51
12	5.40	7.84	11.11
13	5.20	9.69	9.99
14	5.20	9.39	11.64
15	5.59	8.32	8.73
16	5.30	10.08	10.14
17	5.17	10.33	10.83
18	5.87	10.14	7.43
19	5.65	7.58	14.75
20	4.92	7.55	16.21
21	5.02	8.44	9.27
22	5.41	8.43	11.29
23	4.89	9.71	9.25
24	5.16	14.83	10.95
25	4.51	9.35	9.92
26	4.99	11.99	11.30
27	5.37	9.73	11.56
28	5.29	9.02	11.01
29	5.17	13.06	8.80
30	5.47	10.98	12.16
avg.±s.d.	5.18±0.28	10.30±2.18	10.75±1.90

Table S30. Performance of T5Chem-CI Algorithm with a Class Boundary of 76 as Defined by μ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test
1	6.37	9.03	16.21
2	6.60	11.19	10.23
3	6.40	11.17	8.63
4	6.20	12.33	10.44
5	6.77	9.78	10.78

6	6.21	8.91	9.20
7	6.65	11.35	9.95
8	6.40	13.87	10.06
9	6.04	7.68	10.81
10	6.01	10.90	10.81
11	6.78	12.45	10.11
12	6.35	7.90	9.68
13	5.42	10.49	10.87
14	6.46	9.95	12.56
15	6.74	8.78	8.62
16	6.05	9.04	9.73
17	6.01	8.64	9.82
18	7.60	9.18	8.00
19	7.31	9.37	13.76
20	6.61	9.56	16.16
21	6.91	8.95	9.49
22	7.19	10.54	11.28
23	6.26	9.54	9.30
24	6.12	13.05	10.67
25	6.39	9.96	12.21
26	6.01	11.71	10.16
27	6.50	9.39	12.10
28	6.47	9.54	10.74
29	6.53	11.49	9.06
30	6.39	10.05	12.22
avg.±s.d.	6.46±0.43	10.19±1.47	10.79±1.90

(iii) Transformer encoder

The transformer encoder is used in predicting the %*ee* for the reactions of our interest.[18] Each transformer encoder layer consists of a multi-head self-attention layer followed by a position-wise feedforward layer. Reaction SMILES strings are used as molecular representations and processed through token embedding and positional encoding to extract meaningful features. The model is trained using the Adam optimizer, with Mean Squared Error (MSE) as the loss function. Various hyperparameters, such as the number of layers, number of attention heads, learning rate, and dropout rates, are tuned using Optuna. The default parameters and hyperparameters used in the transformer encoder algorithm are listed in Table S31.

Table S31. Hyperparameters of Transformer encoder

hyperparameters	values

hidden dimension	256
output dimension	128
num_features	200
epochs	50
batch_size	64

Table S32. Performance of the Transformer encoder Algorithm in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters { num_layer, num_heads, dropout_rate, learning_rate}
1	5.00	11.93	10.71	4,4,0.02,0.00054
2	3.47	11.73	9.35	2,4,0.00,0.00022
3	4.17	12.27	14.46	4,4,0.00,0.00010
4	6.98	13.65	16.92	4,4,0.09,0.00087
5	5.44	9.58	11.16	4,4,0.09,0.00075
6	5.69	10.61	11.04	4,2,0.02,0.00028
7	3.61	10.25	12.26	2,4,0.01,0.00069
8	4.66	12.14	12.75	2,4,0.08,0.00058
9	5.42	11.25	11.27	4,4,0.05,0.00083
10	5.27	10.04	11.03	4,4,0.00,0.00094
11	4.28	8.72	11.79	2,4,0.01,0.00032
12	5.59	14.20	12.73	2,4,0.18,0.00076
13	5.84	14.30	12.44	2,4,0.18,0.00099
14	5.94	11.09	12.21	4,4,0.04,0.00068
15	5.77	11.26	12.13	4,4,0.02,0.00037
16	4.53	10.13	12.97	4,4,0.00,0.00055
17	5.31	12.83	11.62	4,4,0.00,0.00018
18	8.19	13.52	9.96	4,4,0.20,0.00097
19	4.10	10.10	11.10	2,4,0.00,0.00037
20	4.72	15.45	18.84	2,4,0.10,0.00090
21	4.42	10.84	11.04	2,4,0.01,0.00046
22	8.36	14.00	14.36	4,2,0.07,0.00035
23	5.13	11.77	11.37	4,2,0.01,0.00040
24	6.95	11.25	11.51	4,2,0.08,0.00087
25	5.94	13.09	10.52	4,4,0.05,0.00028
26	5.66	11.49	14.61	4,4,0.07,0.00090
27	3.90	11.92	9.51	2,2,0.01,0.00069
28	5.26	13.07	13.45	2,4,0.12,0.00060
29	4.10	11.55	12.37	2,4,0.01,0.00018
30	4.82	11.03	12.23	2,2,0.05,0.00048
avg.±s.d.	5.28 ± 1.18	11.84 ± 1.56	12.26 ± 2.02	

Transformer encoder-CI

Table S33. Performance of Transformer encoder-CI Algorithm with a Class Boundary of 40 in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters { num_layer, num_heads, dropout_rate, learning_rate}
1	6.20	13.07	10.83	4,4,0.09,0.00039
2	13.06	10.98	13.27	2,2,0.00,0.00006
3	4.48	14.88	14.89	4,4,0.01,0.00032
4	5.02	11.50	13.89	2,2,0.02,0.00031
5	7.18	9.55	12.24	4,4,0.16,0.00041
6	5.52	11.32	13.28	2,2,0.03,0.00025
7	4.39	8.43	12.79	4,2,0.00,0.00029
8	4.90	11.57	10.37	4,2,0.00,0.00027
9	4.75	12.15	12.09	4,4,0.01,0.00016
10	4.67	11.26	13.32	2,4,0.03,0.00035
11	8.76	10.97	12.11	4,4,0.11,0.00061
12	5.93	13.49	12.15	4,2,0.04,0.00049
13	7.42	13.30	11.78	4,2,0.06,0.00052
14	3.85	9.20	9.95	2,4,0.00,0.00015
15	6.64	10.09	13.06	2,4,0.03,0.00016
16	5.64	13.57	13.70	4,4,0.12,0.00100
17	3.50	13.93	9.96	2,4,0.00,0.00038
18	4.93	13.79	9.55	4,4,0.00,0.00082
19	4.12	10.73	10.68	2,2,0.00,0.00030
20	6.81	14.88	17.77	4,4,0.14,0.00042
21	4.71	11.80	9.29	4,4,0.00,0.00035
22	4.71	9.90	10.38	4,4,0.01,0.00093
23	4.91	10.34	11.29	4,4,0.00,0.00056
24	6.13	11.75	11.86	4,4,0.00,0.00010
25	6.52	14.78	11.80	4,2,0.02,0.00037
26	4.28	10.17	12.14	2,2,0.01,0.00044
27	4.26	12.93	10.12	2,2,0.01,0.00023
28	3.75	11.76	12.14	2,4,0.01,0.00020
29	3.55	11.38	13.66	2,2,0.00,0.00097
30	3.77	10.80	10.15	4,4,0.00,0.00029
avg.±s.d.	5.48±1.92	11.81±1.73	12.02±1.81	

Table S34. Performance of Transformer encoder-CI Algorithm with a Class Boundary of 50 in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters { num_layer, num_heads, dropout_rate, learning_rate}
1	3.99	12.23	9.58	4,4,0.00,0.00020
2	9.33	10.91	13.36	2,2,0.00,0.00008
3	3.79	14.89	14.78	2,4,0.00,0.00026

4	7.49	12.02	15.61	4,4,0.15,0.00099
5	5.14	11.73	13.39	2,4,0.10,0.00057
6	6.80	17.57	17.45	4,4,0.19,0.00044
7	4.19	11.79	11.35	2,4,0.01,0.00038
8	6.40	12.04	12.61	4,2,0.12,0.00053
9	7.28	12.45	10.82	4,2,0.08,0.00065
10	6.67	12.66	12.56	4,2,0.05,0.00035
11	6.90	10.79	13.51	4,4,0.03,0.00068
12	4.92	12.14	8.17	2,2,0.02,0.00064
13	4.32	14.90	11.30	2,4,0.09,0.00098
14	9.40	11.29	13.76	4,4,0.03,0.00005
15	5.58	12.47	12.68	4,2,0.00,0.00016
16	5.33	10.79	14.47	2,2,0.08,0.00050
17	8.97	10.92	10.19	4,2,0.07,0.00052
18	4.89	12.37	10.79	4,2,0.00,0.00048
19	4.11	9.36	10.90	4,4,0.00,0.00057
20	5.16	15.61	17.53	2,2,0.09,0.00075
21	4.59	11.78	10.92	2,4,0.03,0.00053
22	5.07	13.62	10.29	4,2,0.01,0.00023
23	7.17	12.20	12.18	4,4,0.20,0.00088
24	6.35	11.72	12.49	4,4,0.03,0.00077
25	5.69	11.61	10.48	4,2,0.01,0.00019
26	5.76	11.51	13.80	2,4,0.01,0.00006
27	5.48	12.34	10.19	4,4,0.01,0.00059
28	5.05	11.95	13.10	2,2,0.02,0.00097
29	4.19	10.29	11.51	2,4,0.02,0.00074
30	6.40	11.89	11.47	4,2,0.04,0.00060
avg.±s.d.	5.88±1.55	12.26±1.65	12.37±2.17	

Table S35. Performance of Transformer encoder-CI Algorithm with a Class Boundary of 60 in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters { num_layer, num_heads, dropout_rate, learning_rate}
1	5.74	11.69	10.83	2,2,0.04,0.00037
2	5.06	13.43	13.76	4,4,0.02,0.00015
3	6.74	12.77	15.58	2,2,0.00,0.00006
4	5.04	10.51	14.93	4,4,0.05,0.00041
5	7.67	10.62	14.67	4,4,0.11,0.00043
6	7.60	16.39	16.92	4,4,0.19,0.00044
7	5.79	8.56	10.75	4,4,0.00,0.00014
8	5.34	11.31	10.62	4,4,0.02,0.00037
9	5.12	12.47	11.91	4,4,0.05,0.00039
10	3.85	9.53	10.71	4,4,0.00,0.00019
11	4.19	13.35	13.84	4,4,0.03,0.00020
12	7.02	13.19	9.45	4,4,0.00,0.00005
13	4.48	11.98	9.06	4,4,0.06,0.00036

14	4.58	11.03	12.49	2,2,0.00,0.00054
15	5.48	11.16	12.10	2,2,0.04,0.00029
16	5.41	11.09	14.97	4,4,0.00,0.00084
17	4.80	11.56	10.60	4,4,0.00,0.00083
18	4.62	11.36	10.58	2,2,0.03,0.00043
19	4.64	9.79	10.58	2,2,0.01,0.00086
20	3.81	10.92	15.17	4,4,0.00,0.00091
21	4.83	9.66	10.26	4,4,0.06,0.00058
22	5.80	14.36	11.99	2,2,0.00,0.00056
23	5.68	10.71	10.91	2,2,0.00,0.00010
24	4.69	11.92	11.14	4,4,0.02,0.00058
25	4.50	10.50	9.80	4,4,0.00,0.00023
26	4.49	10.10	15.06	4,4,0.02,0.00051
27	8.86	11.79	10.38	2,2,0.00,0.00005
28	5.24	12.51	13.67	4,4,0.04,0.00052
29	5.13	10.32	12.09	4,4,0.01,0.00016
30	8.12	11.46	11.74	2,0,0.04,0.00040
avg.±s.d.	5.48±1.27	11.54±1.58	12.22±2.11	

Table S36. Performance of Transformer encoder-CI Algorithm with a Class Boundary of 70 in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters { num_layer, num_heads, dropout_rate, learning_rate}
1	5.85	10.70	10.90	4,4,0.01,0.00096
2	3.89	10.86	11.30	2,4,0.00,0.00024
3	12.60	16.91	16.84	4,4,0.01,0.00006
4	7.98	11.30	14.50	2,2,0.01,0.00008
5	5.92	8.43	10.21	4,4,0.04,0.00047
6	5.04	9.85	12.75	4,4,0.03,0.00023
7	4.30	8.54	12.91	2,4,0.01,0.00054
8	4.07	10.98	11.40	4,4,0.00,0.00064
9	4.34	10.74	11.40	2,4,0.01,0.00057
10	5.83	9.83	12.16	4,2,0.00,0.00012
11	4.55	11.52	10.77	2,2,0.00,0.00045
12	5.58	12.79	9.23	4,4,0.03,0.00046
13	5.30	13.01	10.74	4,2,0.01,0.00054
14	5.77	9.72	13.16	4,4,0.01,0.00100
15	3.39	9.01	11.16	2,4,0.00,0.00028
16	4.61	12.75	13.82	4,4,0.03,0.00047
17	4.81	15.43	11.30	2,4,0.00,0.00014
18	4.41	12.75	11.49	2,4,0.05,0.00059
19	4.85	12.23	11.49	2,2,0.02,0.00051
20	5.35	11.99	15.77	4,4,0.05,0.00094
21	3.98	11.34	9.63	2,4,0.00,0.00075
22	8.06	13.61	11.81	4,2,0.17,0.00043
23	4.32	11.46	12.32	2,4,0.00,0.00089

24	4.81	12.68	12.86	2,4,0.02,0.00021
25	7.29	15.27	11.62	4,2,0.03,0.00062
26	4.88	11.02	12.57	4,4,0.07,0.00045
27	7.96	14.07	10.63	4,2,0.14,0.00098
28	4.41	13.70	12.47	2,4,0.01,0.00042
29	4.95	12.42	15.06	2,2,0.02,0.00019
30	5.55	13.97	12.88	2,2,0.08,0.00046
avg.±s.d.	5.49±1.81	11.96±2.03	12.17±1.73	

Table S37. Performance of Transformer encoder-CI Algorithm with a Class Boundary of 54 as Defined by μ–σ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters { num_layer, num_heads, dropout_rate, learning_rate}
1	6.80	10.83	11.71	2,2,0.18,0.00080
2	8.81	12.38	10.57	4,2,0.13,0.00097
3	3.92	12.04	15.25	2,4,0.01,0.00028
4	6.82	14.83	18.03	2,4,0.10,0.00032
5	8.28	11.66	15.00	4,4,0.15,0.00029
6	5.62	11.04	9.93	4,2,0.01,0.00041
7	9.26	11.28	11.46	4,4,0.05,0.00085
8	6.91	12.39	12.65	4,2,0.17,0.00099
9	3.77	12.14	9.94	2,4,0.05,0.00068
10	4.58	10.39	12.67	4,4,0.02,0.00021
11	4.29	9.32	10.34	4,4,0.01,0.00027
12	5.95	11.71	11.04	4,4,0.08,0.00027
13	5.89	13.36	12.36	4,2,0.09,0.00064
14	5.24	9.29	12.20	2,4,0.01,0.00061
15	3.55	10.91	12.36	2,4,0.00,0.00031
16	6.12	13.20	15.00	4,2,0.04,0.00028
17	4.27	14.77	10.17	2,2,0.03,0.00058
18	5.74	11.78	10.55	4,2,0.01,0.00015
19	5.60	10.56	10.18	4,4,0.00,0.00012
20	6.45	12.82	14.40	4,4,0.02,0.00091
21	8.56	13.75	13.44	4,2,0.03,0.00070
22	4.49	11.11	12.77	2,2,0.04,0.00040
23	4.08	11.55	12.10	2,2,0.01,0.00062
24	4.52	13.32	11.84	4,4,0.01,0.00032
25	5.69	15.06	13.61	4,2,0.08,0.00021
26	4.10	10.15	10.78	4,4,0.01,0.00033
27	3.98	11.13	9.94	2,4,0.00,0.00013
28	7.59	12.50	11.57	4,2,0.01,0.00048
29	3.66	9.88	11.88	2,4,0.01,0.00077
30	4.56	12.17	12.87	2,2,0.00,0.00012
avg.±s.d.	5.64±1.65	11.91±1.52	12.22±1.91	

Table S38. Performance of Transformer encoder-CI Algorithm with a Class Boundary of 76 as μ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters { num_layer, num_heads, dropout_rate, learning_rate}
1	5.77	11.56	10.13	4,4,0.04,0.00033
2	4.35	11.68	9.55	2,4,0.00,0.00009
3	5.35	11.39	15.67	4,4,0.13,0.00040
4	6.70	11.01	14.68	4,4,0.04,0.00092
5	7.21	12.26	14.95	4,4,0.04,0.00004
6	6.66	11.33	10.53	4,2,0.15,0.00083
7	7.49	10.80	11.86	2,4,0.04,0.00062
8	5.79	14.84	14.34	4,4,0.00,0.00046
9	5.94	12.11	12.34	4,2,0.01,0.00042
10	5.41	9.78	10.41	4,2,0.00,0.00010
11	4.13	10.06	11.31	2,4,0.00,0.00032
12	8.58	14.03	11.38	4,2,0.00,0.00092
13	3.87	11.93	9.12	2,2,0.05,0.00095
14	6.76	10.84	13.34	2,2,0.00,0.00038
15	6.53	10.93	10.64	4,2,0.01,0.00062
16	5.80	11.84	14.32	2,2,0.01,0.00025
17	8.40	12.43	12.63	4,4,0.09,0.00067
18	6.67	14.63	10.95	4,2,0.00,0.00011
19	6.36	10.90	11.06	4,2,0.01,0.00040
20	3.42	11.51	15.59	4,2,0.14,0.00023
21	4.62	11.96	12.36	4,4,0.13,0.00094
22	7.95	17.67	16.49	2,4,0.10,0.00050
23	5.67	10.52	11.64	2,4,0.00,0.00013
24	6.83	13.21	11.95	2,2,0.05,0.00092
25	9.33	9.98	13.03	4,2,0.00,0.00013
26	5.83	10.47	12.05	4,4,0.00,0.00051
27	9.69	10.74	10.48	4,4,0.00,0.00007
28	4.60	13.64	14.32	2,4,0.01,0.00017
29	3.75	9.57	11.15	2,4,0.02,0.00059
30	4.85	11.74	11.36	2,4,0.07,0.00075
avg.±s.d.	6.14±1.61	11.84±1.73	12.32±1.94	

(2.4) ULMFiT

This method, adapted from the ULMFiT (Universal Language Model Fine-Tuning), utilizes a transfer learning approach originally designed for NLP classification task.[5] It involves pre-training on a ChEMBL dataset containing 1M molecules, followed by task-specific fine-tuning for regression. The model architecture employed is AWD-LSTM (ASGD Weight-

Dropped LSTM). The model uses an embedding vector length of 400, with an encoder consisting of three LSTM layers. The first layer having an input size of 400, 1152 hidden units, and an output size of 400 in the last layer. The regressor uses the encoder output to make predictions. The regressor model consists of two feed-forward neural network layers. The first layer processes the concatenated output vectors from the last LSTM layer of the encoder (a combination of max pooling, mean pooling, and the last time step, resulting in a 1200-dimensional vector), followed by a ReLU activation. The model final output layer is single node for the regression task. The details of hyperparameter tuning are given in Table S39 to Table S45. For the test set, both canonical as well as augment SMILES (denoted as TTA – test time agumentation) are considered.

Table S39. Train, Validation, and Test RMSEs obtained by Varying Numbers of Augmented smiles.[i] Shown in Bold Font is the Optimal Model

No. of augmented smiles	train	validation	test(canonical)	test(TTA)
25	19.16	13.62	15.40	16.54
50	11.23	9.45	13.54	11.64
75	13.87	10.83	12.67	12.27
100	10.90	10.49	12.69	12.12
125	**11.12**	**11.70**	**13.71**	**12.67**
150	17.01	8.75	12.85	12.29

[i] σ_g = 0.0, batch size = 64, dropout ratio = 0.0, epoch = 20, learning rate = 0.001, weight decay= 0.0

Table S40. Train, Validation, and Test RMSEs obtained by Varying σ_g noise.[i] Shown in bold font is the Optimal Model

noise	train	validation	test(canonical)	test(TTA)
0.1	11.14	12.28	14.00	12.98
0.2	11.13	12.04	13.35	12.72
0.3	11.12	11.95	13.64	12.79
0.4	**11.13**	**11.94**	**13.35**	**12.61**
0.5	11.16	11.08	13.43	12.71

[i] augmented smiles = 100, batch size = 64, dropout ratio = 0.0, epoch = 20, learning rate = 0.001, weight decay= 0.0

Table S41. Train, Validation, and Test RMSEs obtained by Varying Dropout Rate.[i] Shown in bold font is the Optimal Model

dropout rate	train	validation	test(canonical)	test(TTA)
0.0	**11.13**	**11.94**	**13.35**	**12.61**
0.1	11.21	9.96	13.34	12.76
0.2	11.25	10.23	13.94	13.04
0.3	11.44	8.33	14.78	13.11
0.4	11.89	7.76	15.50	13.21
0.5	12.23	8.09	16.68	14.06

[i] σ_g = 0.5, augmented smiles = 100, batch size = 64, epoch = 20, learning rate = 0.001, weight decay= 0.0

Table S42. Train, Validation, and Test RMSEs obtained by Varying Batch Size.[i] Shown in bold font is the Optimal Model

batch size	train	validation	test(canonical)	test(TTA)
32	13.31	6.57	12.62	11.44
64	**11.13**	**11.94**	**13.35**	**12.61**
128	14.03	10.14	13.50	12.41

[i] σ_g = 0.5, augmented smiles = 100, epoch = 20, learning rate = 0.001, weight decay= 0.0, dropout ratio = 0.0

Table S43. Train, Validation, and Test RMSEs obtained by Varying Number of Epochs.[i] Shown in bold font is the Optimal Model

epoch	train	validation	test(canonical)	test(TTA)
20	**11.13**	**11.94**	**13.35**	**12.61**
30	12.01	9.91	12.71	11.84
40	14.48	9.44	12.95	11.37
50	12.67	9.11	11.64	11.02

[i] σ_g = 0.5, augmented smiles = 100, batch size = 64, learning rate = 0.001, weight decay= 0.0, dropout ratio = 0.0

Table S44. Train, Validation, and Test RMSEs obtained by Varying Learning Rate.[i] Shown in bold font is the Optimal Model

learning rate	train	validation	test(canonical)	test(TTA)
0.1	20.00	18.43	25.98	25.83
0.01	14.30	13.76	21.90	20.46
0.001	**11.13**	**11.94**	**13.35**	**12.61**
0.0001	63.33	71.89	69.10	70.19

[i] σ_g = 0.5, augmented smiles = 100, batch size = 64, epoch = 20, weight decay= 0.0, dropout ratio = 0.0

Table S45. Train, Validation, and Test RMSEs obtained by Varying Weight Decay.[i] Shown in bold font is the Optimal Model

weight decay	train	validation	test(canonical)	test(TTA)
0.01	11.12	10.25	12.55	12.69
0.02	11.15	11.20	12.94	12.32
0.03	11.17	12.57	13.87	13.06
0.0	**11.13**	**11.94**	**13.35**	**12.61**
0.1	11.40	12.46	14.03	13.40
0.2	11.97	11.75	14.49	13.68

[i] σ_g = 0.5, augmented smiles = 100, batch size = 64, epoch = 20, learning rate = 0.001, dropout ratio = 0.0

Table S46. Performance of the ULMFIT Model in terms of RMSE across Different Splits

split	train	validation	test(canonical)	test(TTA)
1	11.13	11.94	13.35	12.61
2	10.92	10.19	9.45	9.41
3	9.99	13.72	12.34	11.94
4	11.23	18.04	11.36	9.82
5	10.67	12.03	9.82	8.85
6	11.2	14.85	13.61	12.34
7	10.42	11.26	10.02	8.96
8	10.2	8.22	10.44	9.88
9	10.51	12.82	12.29	11.39
10	10.61	10.52	10.16	10.37
11	11	8.49	11.38	9.01
12	10.39	10.36	12.29	9.59
13	10.38	10.11	12.87	12.86
14	11.81	12.35	10.5	9.84
15	13.25	8.91	9.22	9.54
16	10.87	12.7	10.8	10.65
17	10.59	10.44	10.35	9.73
18	9.71	12.3	10.94	9.53
19	12.83	6.89	11.83	10.01
20	10.31	11.71	14.23	12.59
21	13.75	7.74	9.67	9.01
22	10.81	9.64	11.29	10.09
23	10.49	7.54	11.36	9.93
24	10.87	10.38	12.54	11.56
25	9.97	13.32	11.42	10.04
26	10.55	11.26	10.8	9.82
27	10.99	10.95	11.03	10.39
28	10.79	9.51	10.01	8.46
29	11.25	14.4	11.26	10.35
30	10.66	9.97	12.51	11.14
avg.±s.d.	10.94±0.90	11.09±2.39	11.30±1.27	10.32±1.20

ULMFiT-CI

Table S47. Performance of ULMFiT-CI Algorithm with a Class Boundary of 40 in terms of RMSE across Different Splits

split	train	validation	test(canonical)	test(TTA)
1	8.87	13.20	14.49	13.89
2	8.53	11.68	11.10	11.37
3	7.39	11.60	11.55	11.47
4	8.76	16.43	10.43	9.98
5	8.06	12.93	10.20	8.98
6	8.97	14.39	13.69	13.20
7	8.26	11.25	10.38	10.16
8	7.98	9.32	10.62	10.81
9	7.65	11.50	11.41	11.37
10	8.02	11.66	10.14	10.75
11	8.23	10.01	11.16	9.48
12	8.11	10.12	12.49	10.59
13	7.98	10.05	12.00	12.17
14	8.88	12.57	10.76	10.26
15	8.09	10.33	9.16	8.66
16	8.30	13.12	10.67	10.78
17	8.10	12.93	11.71	10.92
18	8.13	12.08	11.04	9.96
19	6.95	9.12	11.93	10.44
20	7.94	11.37	13.76	12.79
21	8.04	12.52	13.97	12.03
22	8.29	11.16	12.64	11.69
23	8.96	7.64	11.14	9.97
24	8.79	13.06	14.96	13.81
25	7.49	13.57	9.89	9.08
26	8.14	11.92	11.19	10.80
27	8.88	10.96	11.76	11.39
28	8.37	15.31	11.98	11.44
29	8.93	15.69	11.82	11.54
30	8.07	10.47	11.65	11.40
avg.±s.d.	8.24±0.50	11.93±1.97	11.66±1.39	11.04±1.30

Table S48. Performance of ULMFiT-CI Algorithm with a Class Boundary of 50 in terms of RMSE across Different Splits

split	train	validation	test(canonical)	test(TTA)
1	9.00	12.66	14.29	13.31
2	8.69	12.02	12.28	11.94
3	7.59	12.24	11.31	11.66
4	9.02	17.23	11.59	10.92

5	8.38	13.11	10.83	9.53
6	9.12	13.78	13.86	13.36
7	8.59	11.38	11.20	10.04
8	8.23	10.78	10.75	11.13
9	7.85	11.49	11.36	11.28
10	8.26	12.63	10.68	11.48
11	8.50	9.68	12.47	10.43
12	8.28	9.61	12.74	10.88
13	8.26	11.53	14.86	14.37
14	9.05	12.65	10.90	10.53
15	8.32	10.37	9.10	8.60
16	8.56	12.87	9.64	10.13
17	8.28	10.91	11.19	10.55
18	8.33	12.53	11.75	10.66
19	7.14	9.14	12.11	10.34
20	8.20	11.74	13.89	12.85
21	8.40	12.57	14.52	12.35
22	8.35	11.84	13.76	12.18
23	9.25	7.15	9.84	9.99
24	8.97	11.56	14.50	13.28
25	7.66	13.48	10.14	9.47
26	8.43	11.53	11.44	10.37
27	9.11	11.19	12.52	11.84
28	8.48	14.64	11.88	10.78
29	9.25	14.88	11.60	11.65
30	8.35	10.84	12.08	11.47
avg.±s.d.	8.46±0.50	11.93±1.89	11.97±1.53	11.25±1.31

Table S49. Performance of ULMFiT-CI Algorithm with a Class Boundary of 54 as Defined by μ–σ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test(canonical)	test(TTA)
1	9.03	12.68	14.38	13.37
2	8.73	12.30	12.30	12.18
3	7.62	13.22	12.11	12.27
4	9.03	19.33	11.99	10.93
5	8.37	12.32	10.04	9.25
6	9.18	15.31	14.24	13.47
7	8.63	11.39	11.00	9.85
8	8.29	11.73	11.81	11.39
9	7.86	11.35	11.84	11.36
10	8.31	12.64	11.02	11.48
11	8.54	10.61	11.09	9.82
12	8.31	11.48	16.18	12.24
13	8.31	9.92	13.17	13.23
14	9.10	13.21	10.70	10.10

15	8.35	11.01	9.42	8.82
16	8.61	11.84	9.27	10.14
17	8.34	13.15	11.27	10.71
18	8.36	12.16	10.25	9.96
19	7.20	8.59	12.11	10.43
20	8.24	12.44	13.78	13.06
21	8.44	11.59	13.00	11.38
22	8.36	12.57	13.18	12.43
23	9.33	8.09	10.48	10.32
24	9.05	11.93	13.99	13.12
25	7.68	14.91	10.54	9.63
26	8.49	11.52	11.56	10.99
27	9.19	11.74	12.92	12.23
28	8.51	12.56	10.96	10.47
29	9.27	14.60	11.93	11.76
30	8.36	10.43	11.44	11.32
avg.±s.d.	8.50±0.51	12.22±2.06	11.93±1.58	11.26±1.30

Table S50. Performance of ULMFiT-CI Algorithm with a Class Boundary of 60 in terms of RMSE across Different Splits

split	train	validation	test(canonical)	test(TTA)
1	9.14	12.75	14.86	14.18
2	8.84	12.17	11.49	11.99
3	7.74	10.63	10.67	11.35
4	9.17	19.07	12.22	11.46
5	8.51	14.03	10.23	9.62
6	9.27	9.74	12.01	12.10
7	8.72	11.59	11.36	10.29
8	8.40	12.56	12.34	11.49
9	7.98	11.85	11.69	11.84
10	8.34	12.60	11.18	11.31
11	8.66	9.56	10.69	9.51
12	8.42	10.11	11.32	10.38
13	8.36	10.42	12.75	12.81
14	9.23	13.41	11.12	10.37
15	8.48	10.98	9.01	8.90
16	8.67	12.84	10.44	10.44
17	8.35	13.02	11.45	10.88
18	8.47	12.53	9.47	9.75
19	7.31	7.99	11.82	10.40
20	8.33	11.72	13.53	12.61
21	8.58	11.28	12.02	11.41
22	8.48	12.10	13.23	12.27
23	9.36	8.07	9.71	9.74
24	9.17	12.89	14.55	13.57

25	7.80	12.13	10.13	9.64
26	8.59	12.26	12.22	11.37
27	9.23	11.05	11.17	11.15
28	8.63	12.85	10.89	10.71
29	9.41	15.42	12.44	11.99
30	8.52	10.61	11.86	11.52
avg.±s.d.	8.61±0.50	11.94±2.10	11.60±1.35	11.17±1.23

Table S51. Performance of ULMFiT-CI Algorithm with a Class Boundary of 70 in terms of RMSE across Different Splits

split	train	validation	test(canonical)	test(TTA)
1	9.28	12.58	14.31	13.13
2	9.11	11.93	11.62	11.37
3	8.06	14.03	13.67	12.76
4	9.48	18.74	11.99	10.99
5	8.57	11.96	9.19	9.00
6	9.44	10.82	12.61	12.20
7	9.02	11.17	10.99	10.52
8	8.63	11.29	12.06	11.28
9	8.32	13.80	12.31	11.96
10	8.63	12.12	10.78	11.10
11	9.44	13.75	8.73	8.82
12	8.72	9.94	13.96	12.08
13	8.96	13.35	13.53	13.12
14	8.60	13.88	11.43	10.92
15	8.80	12.98	9.64	8.98
16	7.58	10.21	10.52	11.58
17	8.63	12.30	12.25	11.27
18	8.80	11.72	10.87	10.23
19	8.72	12.82	13.06	11.22
20	9.64	8.00	13.76	13.01
21	9.29	11.43	12.12	11.24
22	8.03	10.93	12.48	11.78
23	8.85	11.90	11.02	10.23
24	9.36	11.79	14.98	13.65
25	8.96	11.30	10.21	9.65
26	9.57	13.83	12.74	12.14
27	8.72	11.37	11.97	12.00
28	9.44	13.75	11.10	10.34
29	8.72	9.94	12.85	12.23
30	8.96	13.35	12.19	12.02
avg.±s.d.	8.88±0.49	12.23±1.88	11.96±1.50	11.36±1.25

Table S52. Performance of ULMFiT-CI Algorithm with a Class Boundary of 76 as Defined by μ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test(canonical)	test(TTA)
1	9.36	13.04	14.90	13.59
2	9.21	12.51	11.82	11.69
3	8.12	10.61	10.69	10.78
4	9.52	17.59	11.37	10.90
5	8.70	12.15	9.73	9.01
6	9.56	12.47	13.31	13.01
7	9.11	10.81	10.58	10.36
8	8.78	11.86	11.97	11.32
9	8.39	11.73	12.07	11.68
10	8.71	12.44	10.92	11.12
11	9.01	10.17	10.82	9.64
12	8.77	10.61	14.11	11.82
13	8.58	10.23	12.50	13.05
14	9.54	13.16	11.43	10.80
15	8.81	10.46	9.66	9.32
16	9.06	12.68	10.62	11.09
17	8.69	12.01	11.42	10.51
18	8.83	12.99	9.89	10.12
19	7.66	8.95	12.21	10.60
20	8.74	11.26	14.22	13.10
21	8.88	11.68	13.16	11.86
22	8.83	12.04	13.02	12.12
23	9.77	8.35	11.17	10.33
24	9.35	11.97	15.71	14.31
25	8.11	11.31	10.21	9.68
26	8.92	12.46	13.63	12.32
27	9.49	11.47	11.65	11.41
28	9.03	14.78	12.24	11.07
29	9.64	14.11	11.91	11.85
30	8.79	9.96	12.20	11.59
avg.±s.d.	8.93±0.48	11.86±1.76	11.97±1.52	11.34±1.26

(2.5) Graph-based models

(i) MPNN

MPNN operates on molecular graphs, representing molecules as undirected graphs with vertices (atoms) and edges (bonds).[19] Unlike traditional methods that rely on pre-computed molecular features, MPNNs process raw features directly from the graph structure. They iteratively pass messages between neighbouring nodes, aggregating local information to

predict properties/reactivity of interests. For training, the default hyperparameters (as listed in Table S53) are employed in our work.

Table S53. Hyperparameters of MPNN

hyperparameters	values
num_step_message_passing	3
num_step_set2set	3
num_layer_set2set	1
readout_feats	1024
learning rate	10^{-4}
λ	0.1
epochs	100

Table S54. Performance of MPNN Algorithm in terms of RMSE across Different Splits

split	train	validation	test
1	7.91	8.63	19.17
2	8.66	9.84	9.03
3	8.52	10.40	8.24
4	8.15	9.85	13.43
5	6.88	11.71	10.42
6	7.48	8.60	9.86
7	9.94	9.21	9.06
8	7.33	14.40	9.56
9	7.42	6.56	11.57
10	6.98	9.24	10.53
11	6.21	9.96	9.60
12	7.34	8.40	10.10
13	7.20	11.64	10.64
14	6.95	8.33	13.14
15	11.04	10.36	11.58
16	8.50	9.12	9.74
17	9.50	11.30	10.26
18	8.98	10.70	9.58
19	8.76	12.32	14.89
20	6.37	11.14	14.75
21	6.91	7.78	9.54
22	8.05	9.35	10.62
23	9.16	9.77	9.39
24	9.13	7.21	10.84
25	7.04	9.60	9.89
26	9.04	9.51	10.11
27	7.04	9.63	10.25
28	6.62	9.11	12.41
29	12..97	12.54	11.22
30	9.21	8.88	10.48

avg.±s.d.	8.01±1.18	9.84±1.65	11.00±2.22

MPNN-CI

Table S55. Performance of MPNN-CI Algorithm with a Class Boundary of 40 in terms of RMSE across Different Splits

split	train	validation	test
1	5.62	8.53	8.92
2	5.95	9.39	8.90
3	6.88	9.92	8.59
4	6.80	11.45	14.13
5	5.51	10.47	11.20
6	6.98	7.84	11.61
7	5.19	8.10	8.75
8	6.04	15.79	10.33
9	6.13	7.05	11.52
10	6.18	8.99	11.27
11	4.87	11.26	9.20
12	6.97	7.85	11.54
13	5.15	11.57	10.28
14	5.35	8.62	12.04
15	6.12	8.61	9.46
16	5.65	9.38	8.92
17	7.37	11.59	11.55
18	5.83	10.18	9.37
19	6.79	12.84	16.23
20	5.06	10.69	15.22
21	10.88	9.12	13.40
22	6.45	9.08	10.42
23	5.71	8.94	9.51
24	5.78	7.56	10.59
25	6.26	9.95	10.93
26	6.90	9.90	10.24
27	5.37	9.61	9.46
28	5.32	8.87	11.38
29	6.89	12.01	9.29
30	7.18	8.76	10.78
avg.±s.d.	6.24±1.13	9.80±1.81	10.83±1.90

Table S56. Performance of MPNN-CI Algorithm with a Class Boundary of 50 in terms of RMSE across Different Splits

split	train	validation	test
1	6.51	8.57	9.12
2	6.88	9.26	8.83
3	7.44	9.52	8.50

4	7.56	11.24	14.58
5	5.39	9.71	11.04
6	5.59	7.67	12.21
7	5.42	8.35	8.70
8	6.17	16.01	9.92
9	4.48	7.47	10.42
10	5.64	9.09	10.40
11	4.89	11.21	9.07
12	6.22	8.33	11.96
13	5.26	11.63	10.48
14	7.83	9.57	12.21
15	6.51	8.67	9.44
16	6.63	9.06	9.06
17	7.30	12.06	11.67
18	6.71	10.28	9.58
19	6.67	12.81	15.89
20	5.42	10.90	15.45
21	10.83	8.86	13.71
22	6.14	8.61	10.30
23	6.26	8.96	9.60
24	6.70	7.47	10.85
25	6.95	10.20	10.06
26	5.75	9.30	9.98
27	5.34	9.57	9.51
28	5.43	8.65	11.42
29	6.61	12.75	10.19
30	6.78	8.84	11.15
avg.±s.d.	6.38±1.17	9.82±1.86	10.84±1.94

Table S57. Performance of MPNN-CI Algorithm with a Class Boundary of 54 as Defined by μ–σ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test
1	6.35	9.24	8.57
2	6.96	9.49	8.76
3	7.01	9.71	8.66
4	7.24	12.30	13.77
5	5.91	10.54	11.90
6	5.76	7.54	12.15
7	6.28	8.14	9.48
8	6.67	15.79	10.27
9	4.51	7.22	10.45
10	6.18	8.87	10.69
11	4.77	11.37	8.90
12	7.25	7.68	11.33
13	5.00	11.53	10.43
14	5.81	8.43	12.42

15	6.62	8.56	9.56
16	6.09	9.76	8.60
17	7.56	11.35	11.63
18	6.90	10.27	9.39
19	7.24	12.79	16.18
20	5.32	10.73	15.07
21	6.89	8.21	11.33
22	5.94	8.28	10.32
23	7.02	9.16	9.56
24	11.84	6.98	11.60
25	6.41	10.43	10.45
26	6.48	10.20	11.10
27	5.50	9.42	9.64
28	7.80	8.52	11.52
29	8.59	12.11	9.46
30	7.79	10.30	11.73
avg.±s.d.	6.66±1.35	9.83±1.91	10.83±1.84

Table S58. Performance of MPNN-CI Algorithm with a Class Boundary of 60 in terms of RMSE across Different Splits

split	train	validation	test
1	6.12	9.78	10.12
2	6.90	9.67	8.81
3	7.53	9.62	8.62
4	7.71	11.37	13.90
5	6.22	10.23	11.20
6	6.23	8.19	12.44
7	5.31	7.84	8.75
8	6.78	16.07	10.27
9	5.80	6.28	11.59
10	6.22	9.02	10.60
11	6.89	11.34	9.87
12	6.38	8.36	11.82
13	5.88	11.51	10.79
14	5.99	8.07	12.92
15	7.11	8.47	9.47
16	6.25	9.21	9.85
17	7.96	11.41	11.54
18	10.32	10.53	9.19
19	7.04	12.79	16.22
20	5.29	10.86	15.12
21	8.03	8.46	11.26
22	6.98	9.04	10.73
23	6.32	9.22	9.61
24	11.37	7.46	11.58
25	7.28	9.88	10.57

26	5.73	9.77	10.44
27	14.07	9.38	9.55
28	5.69	8.76	11.28
29	5.45	13.09	9.91
30	6.72	9.09	11.18
avg.±s.d.	7.05±1.88	9.83±1.93	10.97±1.77

Table S59. Performance of MPNN-CI Algorithm with a Class Boundary of 70 in terms of RMSE across Different Splits

split	train	validation	test
1	5.48	8.52	9.21
2	7.26	9.20	8.83
3	8.05	9.61	8.31
4	7.72	11.84	13.64
5	6.81	10.36	11.76
6	10.16	8.58	11.68
7	6.38	8.22	8.72
8	6.47	16.43	10.43
9	6.57	6.56	11.77
10	5.95	8.63	10.74
11	5.86	11.06	8.88
12	6.49	8.04	11.95
13	5.60	11.54	10.42
14	7.58	8.26	12.86
15	7.60	8.66	9.31
16	6.45	9.13	8.81
17	8.89	12.05	11.35
18	7.00	10.30	9.32
19	7.29	12.91	16.36
20	5.43	10.85	14.96
21	11.29	8.95	13.77
22	7.70	8.70	10.64
23	6.57	9.09	9.66
24	8.29	7.39	11.77
25	6.83	10.18	11.47
26	10.27	9.58	11.69
27	5.89	9.45	9.51
28	5.63	8.81	11.24
29	7.22	12.42	9.99
30	8.69	8.77	10.92
avg.±s.d.	7.25±1.46	9.80±1.96	11.00±1.93

Table S60. Performance of MPNN-CI Algorithm with a Class Boundary of 76 as Defined by μ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test

1	6.35	9.08	10.72
2	7.22	9.55	8.52
3	7.93	9.63	8.60
4	8.17	11.48	13.51
5	6.39	10.30	10.62
6	10.05	8.32	11.79
7	5.84	8.08	8.71
8	6.39	15.88	10.48
9	6.78	7.03	11.53
10	6.41	8.76	10.68
11	5.32	11.10	9.21
12	6.38	7.82	12.39
13	5.63	11.33	10.40
14	6.38	8.37	12.42
15	7.79	8.67	9.66
16	8.38	9.20	9.69
17	8.61	11.64	11.55
18	7.24	10.30	9.52
19	6.70	13.29	16.33
20	5.94	10.89	15.26
21	11.27	9.00	13.32
22	7.95	8.70	10.49
23	6.76	9.08	9.66
24	9.73	7.42	11.87
25	6.46	9.90	11.40
26	10.04	9.79	11.77
27	6.33	9.34	9.52
28	6.08	9.04	11.18
29	7.67	12.18	9.55
30	7.29	9.01	11.01
avg.±s.d.	7.35±1.47	9.80±1.84	11.05±1.83

(ii) AttentiveFP

In this study, we employed the AttentiveFP architecture[20] as our primary model to predict the %*ee* values. Given the computational challenges associated with neural networks, we adopted an automated hyperparameter optimization framework, Optuna. This involved optimizing common hyperparameters (such as learning rate) as well as model-specific hyperparameters (such as the number of layers, graph feature size, number of timesteps, and dropout rate). The optimal hyperparameters are then selected based on validation set performance. The default parameters and hyperparameters used in the AttentiveFP algorithm are listed in Table S61.

Table S61. List of Parameters used in AttentiveFP

method	default parameters	hyperparameters
AttentiveFP	node_feat_size = 39, edge_feat_size = 10	num_layers: {1 to 5}, graph_feat_size: {100 to 500}, dropout_rate: {1 to 5}, learning_rate: {e^{-5} to e^{-1}}, num_timesteps: {1 to 3}

Table S62. List of Atom and Bond Features

feature	size	description [i]
atom Symbol	16	atom symbols [B, C, N, O, F, Si, P, S, Cl, Br, I, metal]
degree	6	the number of covalent bonds [0, 1, 2, 3, 4, 5]
formal charge	1	electronic charge
radical electrons	1	the number of radical(unpaired) electrons
hybridization	6	hybridization types [sp, sp^2, sp^3, sp^3d, sp^3d^2, other]
aromaticity	1	indicating whether the atom is part of an aromatic system
hydrogens	5	the number of connected hydrogens [0, 1, 2, 3, 4]
chirality	1	indicating whether the atom is a chiral center
chirality type	2	[R, S]
bond type	4	[single, double, triple, aromatic]
conjugation	1	whether the bond is conjugated
ring	1	whether the bond is in a ring
stereo	4	one-hot encoding of stereochemical configuration [StereoNone, StereoAny, StereoZ, StereoE]

i all these features are represented as one-hot encoded vectors, except for formal charge and radical electrons, which are used as integer values

Table S63. Performance of AttentiveFP Algorithm in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps}
1	7.87	9.01	13.37	2,395,0.44,0.00374,2
2	5.44	8.98	8.69	1,271,0.26,0.00434,1
3	6.11	9.31	7.94	4,231,0.07,0.00058,1
4	5.68	11.52	8.39	5,418,0.07,0.00037,2
5	6.50	8.82	9.16	2,189,0.00,0.00126,2
6	7.73	8.75	9.19	4,439,0.50,0.00069,2
7	7.37	8.53	9.26	4,428,0.32,0.00101,2
8	8.97	8.34	10.58	5,456,0.41,0.00029,1
9	6.46	8.16	10.53	1,264,0.18,0.00016,2
10	7.59	7.41	10.24	4,254,0.22,0.00028,1
11	7.82	11.21	8.79	4,337,0.12,0.00174,2
12	9.50	9.08	10.87	4,314,0.40,0.00046,3
13	7.32	10.35	10.92	4,103,0.04,0.00079,2

14	5.56	7.68	10.73	5,336,0.32,0.00122,2
15	6.94	10.55	10.55	1,392,0.43,0.00487,2
16	7.14	13.50	11.55	2,361,0.39,0.00016,2
17	7.81	7.56	10.01	5,276,0.37,0.00034,2
18	5.13	8.87	9.85	4,334,0.19,0.00178,2
19	6.63	9.11	14.23	3,161,0.33,0.00146,1
20	6.79	8.88	12.55	1,330,0.17,0.00631,2
21	6.16	6.44	11.54	2,445,0.25,0.00021,1
22	7.96	9.88	12.44	3,181,0.11,0.00152,3
23	7.18	8.85	9.81	4,415,0.37,0.00068,2
24	6.50	9.00	10.82	5,411,0.37,0.00268,1
25	11.45	10.01	11.63	3,418,0.27,0.02194,2
26	8.55	8.39	10.62	3,461,0.07,0.00027,2
27	7.16	9.02	11.67	3,436,0.44,0.00034,1
28	7.29	8.62	9.39	4,496,0.21,0.00022,2
29	9.45	10.50	11.64	1,323,0.05,0.00003,3
30	6.96	9.24	10.14	3,498,0.04,0.00028,3
avg.±s.d.	7.30±1.32	9.58±1.59	10.56±1.43	

(2.6) AttentiveFP-CI

AttentiveFP-CI is a method incorporating a class-aware loss (see Fig. 3) which is considered in order to address class imbalance. Various models are created based on different class boundaries, and their performances are provided in the following tables.

Table S64. Performance of AttentiveFP-CI Algorithm with a Class Boundary of 30 in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps}
1	5.70	6.90	15.00	1,500,0.36,0.00144,3
2	6.10	10.00	9.30	1,429,0.22,0.00269,1
3	6.00	9.60	9.10	3,317,0.22,0.00057,1
4	5.20	9.50	8.90	3,447,0.10,0.00049,3
5	7.50	8.90	8.70	1,183,0.09,0.00057,1
6	8.30	8.40	8.60	1,223,0.50,0.00492,1
7	5.70	9.30	9.50	3,132,0.17,0.00136,2
8	6.20	7.60	9.30	4,440,0.31,0.00059,2
9	6.20	6.90	11.00	2,191,0.14,0.00083,3
10	6.20	6.90	9.60	5,297,0.16,0.00045,2
11	6.90	9.20	7.50	5,150,0.21,0.00061,3
12	7.90	8.20	9.90	4,242,0.41,0.00181,2

13	7.40	9.20	10.00	1,490,0.50,0.00038,2
14	4.40	8.60	11.00	5,482,0.12,0.00090,2
15	9.20	9.00	9.40	5,258,0.36,0.00016,3
16	7.00	8.00	9.70	3,440,0.43,0.00038,2
17	6.40	8.30	10.00	4,315,0.49,0.00139,3
18	6.20	9.90	8.30	4,285,0.11,0.00050,2
19	5.80	9.20	8.30	5,418,0.05,0.00057,3
20	6.40	10.00	12.00	5,272,0.18,0.00092,1
21	5.50	7.30	9.50	4,303,0.17,0.00141,2
22	7.50	9.90	10.00	3,211,0.10,0.00026,2
23	6.30	11.00	9.40	2,479,0.43,0.00084,2
24	4.50	9.40	9.20	4,437,0.06,0.00048,3
25	6.30	9.50	11.00	4,500,0.07,0.00026,3
26	7.00	9.50	9.90	1,103,0.03,0.00205,1
27	5.60	8.40	9.30	3,122,0.03,0.00072,3
28	8.70	8.90	11.00	1,273,0.22,0.00296,1
29	7.90	9.00	9.50	5,491,0.11,0.00156,3
30	6.10	7.70	11.00	4,318,0.29,0.00031,1
avg.±s.d.	6.50±1.10	8.80±1.00	9.80±1.40	

Table S65. Performance of AttentiveFP-CI Algorithm with a Class Boundary of 40 in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps}
1	9.06	10.28	16.47	2,395,0.44,0.00374,2
2	7.82	11.52	10.85	1,271,0.26,0.00434,1
3	6.01	9.56	9.03	4,231,0.07,0.00058,1
4	5.89	11.92	9.44	5,418,0.07,0.00037,2
5	5.74	9.63	10.48	2,189,0.00,0.00126,2
6	6.19	8.08	9.10	4,439,0.50,0.00069,2
7	6.44	9.10	8.46	4,428,0.32,0.00101,2
8	7.09	7.35	10.27	5,456,0.41,0.00029,1
9	8.88	9.73	12.10	1,264,0.18,0.00016,2
10	7.84	7.52	10.15	4,254,0.22,0.00028,1
11	5.65	11.01	8.61	4,337,0.12,0.00174,2
12	7.44	8.01	9.58	4,314,0.40,0.00046,3
13	5.45	10.72	14.51	4,103,0.04,0.00079,2
14	7.45	9.81	10.85	5,336,0.32,0.00122,2
15	9.22	10.02	11.64	1,392,0.43,0.00487,2
16	9.17	10.23	9.84	2,361,0.39,0.00016,2
17	7.31	8.42	10.31	5,276,0.37,0.00034,2
18	6.66	10.27	9.19	4,334,0.19,0.00178,2
19	6.65	8.87	8.92	3,161,0.33,0.00146,1
20	7.57	9.32	12.96	1,330,0.17,0.00631,2

21	7.45	6.98	10.36	2,445,0.25,0.00021,1
22	5.93	9.65	8.70	3,181,0.11,0.00152,3
23	7.15	10.87	8.99	4,415,0.37,0.00068,2
24	9.32	10.07	10.79	5,411,0.37,0.00268,1
25	13.31	10.11	12.77	3,418,0.27,0.02194,2
26	5.86	10.65	10.70	3,461,0.07,0.00027,2
27	7.05	9.05	8.08	3,436,0.44,0.00034,1
28	7.04	8.40	10.75	4,496,0.21,0.00022,2
29	10.89	11.57	12.18	1,323,0.05,0.00003,3
30	4.92	7.52	10.76	3,498,0.04,0.00028,3
avg.±s.d.	7.25±1.36	9.07±1.25	10.34±1.63	

Table S66. Performance of AttentiveFP-CI Algorithm with a Class Boundary of 50 in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps}
1	5.46	8.68	11.99	4,106,0.03,0.00271,2
2	5.65	11.55	10.42	4,327,0.07,0.00032,2
3	5.57	8.79	9.18	4,459,0.18,0.00045,1
4	6.30	9.94	9.44	2,428,0.34,0.00093,1
5	5.95	9.41	9.14	2,432,0.20,0.00066,2
6	9.59	9.71	10.74	2,243,0.23,0.00014,2
7	5.55	8.26	9.35	2,300,0.07,0.00118,3
8	5.81	8.44	10.87	5,173,0.24,0.00130,2
9	7.26	7.73	10.31	3,340,0.40,0.00085,3
10	5.52	7.58	9.80	4,367,0.13,0.00076,3
11	5.89	9.47	9.66	5,260,0.05,0.00052,3
12	6.80	8.99	8.64	4,441,0.34,0.00040,2
13	5.90	11.35	10.78	4,403,0.38,0.00056,2
14	6.79	9.09	10.28	3,139,0.04,0.00036,1
15	8.14	8.48	10.59	1,385,0.15,0.00083,3
16	6.00	11.14	10.09	4,274,0.34,0.00206,1
17	8.29	9.28	10.71	5,272,0.36,0.00023,1
18	6.84	8.22	7.64	3,464,0.13,0.00025,2
19	6.02	8.59	9.07	5,398,0.16,0.00026,2
20	5.76	8.85	13.02	2,234,0.14,0.00081,3
21	7.02	6.75	8.66	2,394,0.34,0.00031,1
22	5.40	8.62	9.89	2,399,0.04,0.00026,2
23	5.27	10.55	9.45	3,440,0.01,0.00022,2
24	7.06	10.90	11.71	1,392,0.11,0.00021,3
25	10.53	11.44	13.34	4,209,0.28,0.00375,2
26	5.77	10.63	9.91	3,341,0.26,0.00071,1
27	6.28	9.04	8.46	3,333,0.20,0.00036,3
28	7.45	9.10	10.58	1,354,0.44,0.00067,2

29	8.71	10.17	11.77	1,451,0.47,0.00010,2
30	6.41	7.43	11.07	4,306,0.39,0.00050,2
avg.±s.d.	6.63±1.30	9.27±1.25	10.22±1.29	

Table S67. Performance of AttentiveFP-CI Algorithm with a Class Boundary of 60 in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps}
1	13.23	12.19	19.34	2,181,0.42,0.01017,2
2	6.72	11.01	9.05	3,248,0.31,0.00043,1
3	6.74	9.28	9.32	2,374,0.22,0.00032,2
4	6.28	9.97	10.30	2,239,0.16,0.00190,2
5	6.89	10.89	10.09	2,389,0.18,0.00218,3
6	7.37	8.45	10.03	1,197,0.14,0.00183,3
7	5.65	8.96	8.86	4,147,0.09,0.00144,2
8	5.43	10.11	10.30	3,354,0.06,0.00075,2
9	5.77	8.35	10.28	4,415,0.15,0.00066,3
10	5.42	6.29	9.99	3,338,0.17,0.00066,2
11	7.41	9.12	9.35	1,244,0.48,0.00125,1
12	6.64	7.22	9.24	3,247,0.32,0.00245,2
13	6.89	8.85	10.24	1,281,0.17,0.00026,1
14	5.68	8.91	9.81	5,319,0.24,0.00156,2
15	7.85	8.21	10.36	1,193,0.41,0.00366,2
16	8.08	9.13	9.91	1,195,0.28,0.00042,1
17	7.19	8.89	12.90	2,474,0.20,0.00017,3
18	6.69	8.00	8.06	2,447,0.27,0.00037,2
19	6.14	8.23	8.81	5,422,0.11,0.00026,3
20	6.27	8.39	13.57	2,257,0.17,0.00047,3
21	6.54	7.14	9.01	2,371,0.24,0.00077,1
22	6.68	9.61	8.23	2,365,0.21,0.00186,2
23	5.78	9.26	9.60	5,456,0.05,0.00026,1
24	8.10	8.52	10.14	2,106,0.39,0.00257,2
25	5.49	9.06	10.29	5,469,0.15,0.00027,3
26	6.52	9.95	9.71	4,443,0.17,0.00039,3
27	8.80	10.38	11.67	5,240,0.17,0.00011,3
28	5.11	8.55	9.55	4,204,0.13,0.00132,1
29	5.25	8.94	9.82	1,447,0.00,0.00019,2
30	6.19	8.45	12.06	5,163,0.00,0.00032,3
avg.±s.d.	6.76±1.53	9.01±1.20	10.33±2.09	

Table S68. Performance of AttentiveFP-CI Algorithm with a Class Boundary of 76 as Defined by μ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters

				{num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps
1	4.92	9.25	11.79	3,351,0.02,0.00202,2
2	7.56	11.65	10.58	1,323,0.07,0.00261,2
3	5.50	9.73	8.86	3,266,0.14,0.00069,3
4	7.01	10.50	8.24	2,496,0.42,0.00047,1
5	7.45	9.04	8.42	1,232,0.17,0.00041,3
6	6.62	8.80	10.16	2,455,0.49,0.00044,1
7	5.96	8.24	8.69	4,249,0.14,0.00174,2
8	5.71	6.15	11.26	3,413,0.32,0.00086,1
9	7.69	8.22	11.29	1,365,0.26,0.00079,1
10	6.97	8.52	10.53	2,496,0.40,0.00094,3
11	6.81	9.91	9.74	2,474,0.24,0.00313,1
12	14.25	11.41	15.25	2,344,0.13,0.00709,3
13	6.60	12.35	10.88	5,296,0.21,0.00026,1
14	4.64	7.94	10.32	5,464,0.35,0.00092,1
15	7.98	9.72	10.69	1,358,0.40,0.00092,1
16	6.91	9.47	9.97	1,372,0.49,0.00138,3
17	7.59	8.66	10.86	1,142,0.31,0.00300,1
18	7.49	11.39	7.74	5,393,0.36,0.00021,3
19	7.39	7.50	11.31	4,124,0.06,0.00037,2
20	5.41	8.67	11.84	4,350,0.01,0.00173,2
21	6.47	6.36	9.71	2,187,0.17,0.00096,2
22	5.65	9.33	8.01	2,312,0.10,0.00051,2
23	5.19	8.91	9.55	5,499,0.35,0.00081,2
24	5.39	9.12	9.75	1,393,0.03,0.00038,3
25	7.24	10.46	9.71	1,395,0.18,0.00068,3
26	7.27	10.76	9.76	4,300,0.14,0.00217,3
27	8.53	9.88	9.60	1,410,0.47,0.00039,1
28	6.42	7.46	10.43	4,133,0.11,0.00125,1
29	5.29	8.93	11.32	3,197,0.06,0.00191,2
30	6.32	8.02	11.27	3,420,0.36,0.00025,3
avg.±s.d.	6.81±1.72	9.21±1.47	10.25±1.46	

Table S69. Performance of AttentiveFP-CI Algorithm with a Class Boundary of 54 as Defined by μ–σ of the Distribution of %*ee* in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps}
1	7.12	8.77	14.28	3,101,0.33,0.00087,1
2	6.59	10.28	9.35	2,375,0.05,0.00350,2
3	6.54	8.77	9.65	2,280,0.22,0.00045,1
4	6.47	8.57	8.72	4,232,0.25,0.00048,1
5	7.53	10.49	8.34	4,175,0.46,0.00123,2

6	8.98	8.95	10.50	3,209,0.44,0.00027,3
7	6.32	9.20	9.59	3,225,0.17,0.00207,3
8	5.49	9.45	10.01	5,282,0.15,0.00101,2
9	6.38	7.98	10.37	2,238,0.04,0.00027,2
10	6.57	10.04	10.25	3,498,0.50,0.00068,1
11	4.84	8.20	8.53	4,457,0.04,0.00028,2
12	6.08	8.46	10.69	5,296,0.13,0.00055,2
13	7.36	10.13	11.17	5,266,0.32,0.00088,1
14	5.83	8.38	10.31	3,299,0.13,0.00184,2
15	7.96	8.99	9.09	5,318,0.22,0.00148,2
16	7.00	8.48	10.59	3,358,0.39,0.00027,3
17	7.61	9.02	10.88	2,196,0.31,0.00046,2
18	6.33	9.35	8.27	3,109,0.05,0.00102,1
19	7.23	9.14	15.31	1,316,0.03,0.00012,2
20	6.01	9.32	13.98	2,267,0.30,0.00116,2
21	7.57	6.87	10.08	5,430,0.18,0.00094,2
22	6.01	9.79	8.74	4,418,0.07,0.00021,3
23	7.25	8.80	9.60	1,146,0.08,0.00095,3
24	9.71	12.96	11.92	5,272,0.08,0.00278,3
25	5.62	9.63	9.05	3,427,0.40,0.00136,2
26	6.70	10.64	9.19	1,386,0.18,0.00038,3
27	7.64	8.49	9.14	4,499,0.30,0.00019,3
28	8.46	10.19	10.64	1,410,0.31,0.00515,2
29	7.84	11.19	10.40	1,466,0.10,0.00010,3
30	5.00	7.29	11.42	2,234,0.01,0.00096,2
avg.±s.d.	6.87±1.11	9.26±1.19	10.33±1.70	

Table S70. Performance of AttentiveFP-CI Algorithm on the ART dataset under Scaffold-Based Data Splitting Employing Different Class Boundaries (indicated in column-1 B)

B	RMSE			R^2		
model	train	validation	test	train	validation	test
AttentiveFP	13.19±6.01	21.24±2.82	13.94±3.61	0.59±0.33	0.04±0.27	0.47±0.27
AttentiveFP-CI						
30 (μ)	13.09±5.05	20.18±2.5	13.85±3.41	0.62±0.28	0.14±0.21	0.48±0.25
76	14.47±5.04	18.43±2.02	14.09±3.65	0.54±0.30	0.28±0.15	0.46±0.28
54	17.02±4.60	19.29±2.48	15.67±3.73	0.39±0.28	0.21±0.18	0.34±0.28

Table S71. Performance of the AttentiveFP-CI Algorithm on the Test Set across Different Class Boundaries (indicated in column-1 B) on the ART Dataset, as the Averaged of 30 Independent Train-Validation-Test Splits

B	MAE	RMSE	R^2	RSE
30	7.10±0.77	9.80±1.40	0.84±0.04	0.16±0.04
40	6.80±0.72	10.34±1.63	0.84±0.03	0.16±0.04
50	7.10±0.77	10.22±1.29	0.83±0.04	0.17±0.04
54	7.10±0.67	10.33±1.70	0.83±0.03	0.17±0.03
76	6.20±1.30	10.25±1.46	0.84±0.07	0.16±0.07

(2.7) Comparison of AttentiveFP and DNN+Fingerprint model

We provide below a comparison between AttentiveFP and the DNN+fingerprint model, highlighting key strengths that make AttentiveFP a preferable model for the ART dataset over an alternative fingerprint based model(s).

a. Better generalization

The DNN+fingerprint model displays signs of overfitting, with a large train-test RMSE gap (5.54±1.50 vs 9.55±1.31). AttentiveFP exhibits a smaller gap (7.41±1.77 vs 10.56±1.86), indicating better generalization to unseen molecules. It is important to equip a model with improved generalizability to handle inference samples from outside the training samples. The train-test performance gap is considered as an indicator of how the model is likely to do with unseen samples.

b. Superior chemically meaningful interpretability

AttentiveFP provides atom-level attention weights that directly highlight the substructures and functional groups driving predictions (Fig. 5). This form of interpretability can be considered an actionable aspect toward planning synthensis, such as for designing new

catalysts/ligands. In contrast, fingerprint-based attribution, although informative, is less detailed and does not correspond as intuitively to the structural features chemists consider while evaluating or redesigning molecules.

c. Reduced information loss in diverse chemical space

Graph representations explicitly encode topology and connectivity, preserving structural distinctions across the highly diverse ART dataset. In contrast, fingerprint-based representations compress molecular structures into fixed-length vectors, which can cause distinct substructures to be mapped to the same features potentially leading to collision or hashing issues. This in turn could result in information loss and reduced discriminative power in chemically diverse datasets.

d. Task-adaptive end-to-end representation learning

AttentiveFP learns representations directly from the molecular graph in a task-specific manner, automatically capturing geometric and electronic features relevant to tasks such as reaction outcome prediction. On the other hand, fingerprint-based models rely on pre-defined, static encodings that are not necessarily optimal for every dataset or task and often require manual selection/tuning of fingerprint type and parameters.

While we appreciate that FPs are valuable for many applications due to their efficiency, for tasks that demand generalization, interpretability, and discrimination across structurally diverse molecules, contemporary graph-based models such as AttentiveFP can offer certain advantages. This aligns with prior demonstrations that graph neural networks can outperform or complement fingerprint-based approaches by learning richer, adaptive representations.[21]

3. Performance AttentiveFP model on the BHA dataset

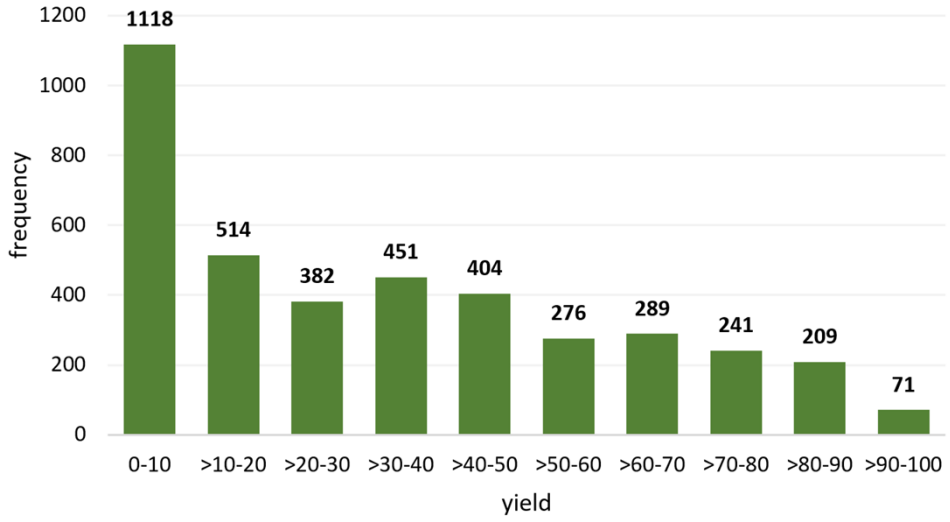
To investigate whether the performance of the AttentiveFP model on the ART dataset, with an RMSE of 10.20, originates from data sparsity and imbalance issues, we have chosen one

of the commonly used chemical reaction datasets for yield prediction models, i.e., BHA-HTE. BHA dataset carries a more balanced and densely distributed data. We trained the AttentiveFP model on this HTE dataset using five different splits, each with a train:validation:test ratio of 70:10:20. Again, we employed Optuna for efficient hyperparameter tuning. The resulting performances are summarized in Table S72.

Table S72. Performance of AttentiveFP Algorithm in terms of RMSE across Different Splits for BHA-HTE Dataset

split	train	validation	test	best hyperparameters {num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps}
1	3.17	6.81	6.25	3,302,0.40,0.00053,2
2	2.53	5.50	6.22	3,494,0.31,0.00033,2
3	2.70	5.58	6.49	4,281,0.31,0.00058,3
4	5.30	7.90	7.12	4,140,0.21,0.00101,3
5	3.25	6.03	6.37	3,456,0.35,0.00023,2
avg.±s.d.	3.39±0.99	6.37±0.90	6.49±0.33	

(a)



(b)

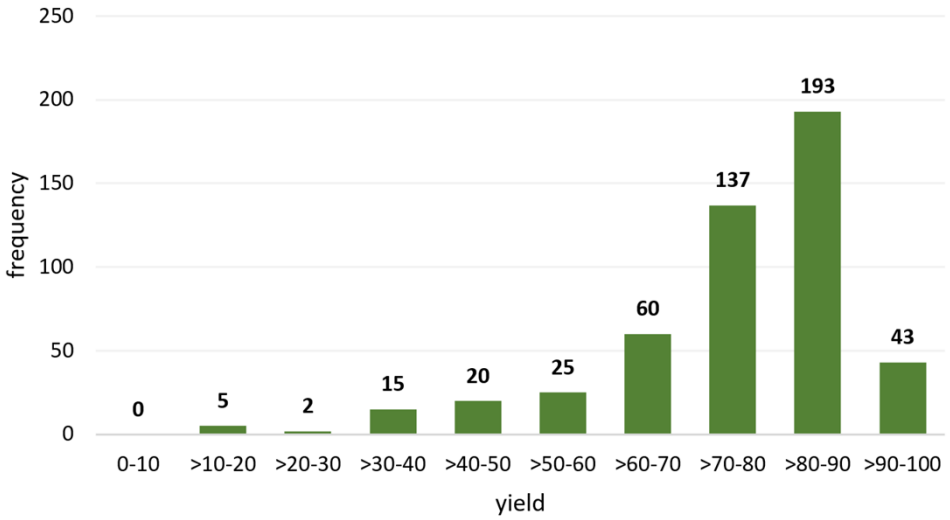


Figure S1. Yield distribution for (a) BHA-HTE, and (b) BHA-LTE dataset.

Furthermore, we have sampled several sparse and class-imbalanced subsets, referred to as the low-throughput dataset (LTE), from the full dataset, with a mean value of 75. A comparison of the yield distribution between the HTE dataset and the LTE subset is provided in Figure S1, highlighting the skewed distribution towards high-yield values. Following the same training protocol, we evaluated the performance of the AttentiveFP-CI algorithm using four such BHA-LTE datasets (Set-1 to Set-4). These datasets have similar yield distributions with an average value of around 75 containing different randomised samples in them. However, each subset slightly in their statistical parameters. Specifically, Set-1 has a mean (μ) of 75.11 and a standard deviation (σ) of 14.95, Set-2 has μ = 74.66 and σ = 15.06, Set-3 has μ = 75.33 and σ = 14.15, and Set-4 has μ = 75.25 and σ = 15.00. The performance of AttentiveFP algorithm using different class boundaries across these four sparse subsets is summarized in Table S73.

Table S73. Performance of the AttentiveFP Algorithm in terms of RMSE across the four BHA-LTE Dataset (Set-1 to Set-4)

	model (class boundary)	train	validation	test
	AttentiveFP	7.04±0.55	9.06±0.79	9.30±1.27
	AttentiveFP-CI			
	30	7.07±0.45	8.47±1.55	9.62±1.13

	40	8.26±1.40	9.43±1.16	10.00±0.94
	50	7.82±0.68	8.51±0.66	9.87±1.24
	60	7.60±1.22	8.69±0.63	9.79±1.11
	70	8.29±1.36	9.65±2.31	9.89±0.78
	75 (μ)	7.79±0.84	8.48±1.00	9.61±0.60
	54	7.09±1.30	8.97±1.42	9.54±0.66
Set-2	AttentiveFP	6.59±0.72	9.00±1.49	9.77±0.68
	AttentiveFP-CI			
	30	7.56±1.27	9.43±1.57	9.13±0.60
	40	7.85±1.26	9.15±1.16	9.42±1.56
	50	6.82±0.82	8.92±1.16	8.71±0.80
	60	6.95±0.42	8.94±1.19	8.98±1.12
	70	6.52±0.78	9.51±1.24	8.85±0.68
	75 (μ)	7.66±0.95	9.21±0.99	9.11±0.97
	54	7.67±1.74	9.99±1.59	9.42±1.5
Set-3	AttentiveFP	6.23±0.88	8.35±2.57	9.14±0.80
	AttentiveFP-CI			
	30	6.52±1.05	8.31±2.01	8.82±0.85
	40	7.13±2.08	8.88±2.04	8.99±1.09
	50	**7.04±0.79**	**8.56±2.05**	**8.70±0.52**
	60	7.07±0.92	8.59±1.94	8.72±0.55
	70	6.86±1.28	9.14±2.58	8.92±0.47
	75 (μ)	7.66±1.97	8.66±2.32	9.28±1.60
	54	7.81±0.87	9.01±2.36	9.01±0.68
Set-4	AttentiveFP	6.72±0.62	9.77±2.39	10.86±1.32
	AttentiveFP-CI			
	30	7.02±0.56	10.43±1.85	11.18±1.14
	40	7.77±2.02	10.46±2.68	11.79±0.65
	50	7.12±1.03	9.57±3.07	10.65±1.43
	60	7.79±2.09	10.00±3.17	11.19±1.37
	70	6.43±0.80	9.60±2.58	10.28±1.62
	75 (μ)	6.95±0.92	9.75±2.84	10.95±1.58

4. Performance of classification followed by regressor (CFR) approach

We have employed CFR approach as a baseline to mitigate the effect of CI/data sparsity.[22] For the classification task, the class boundaries in our CFR model are determined based on the natural distribution of %*ee* observed in the ART dataset. From a statistical perspective, the class boundaries for a binary classification could be placed at μ, (μ+σ), or (μ-σ), where μ and σ are the mean and standard deviation, respectively. The distribution of %*ee* across different boundaries is illustrated in Figure S2. In our ART dataset, μ of %*ee* is 76.16 and σ is

22.49. The Bayes error estimator (BER) analysis is employed to find out the optimal class boundary for the classifier.[23] It reveals that the maximum achievable classification accuracy is 0.985±0.003 with (μ–σ) as the class boundary. Based on this threshold, reactions with experimentally reported %*ee* values between 0 and 54 are classified as the CFR-minor class (65 reactions), while those exceeding this threshold belong to the CFR-major class (311 reactions).

Table S74. Bayes Error Estimator Analysis Across Various Class Boundaries Over 30 Independent Splits

	μ(76.16)	μ+σ(98.65)	μ-σ(53.67)
number of samples (CFR-minor, CFR-major)	130, 246	370, 6	65, 311
BER: accuracy	0.927±0.005	0.981±0.002	0.985±0.003

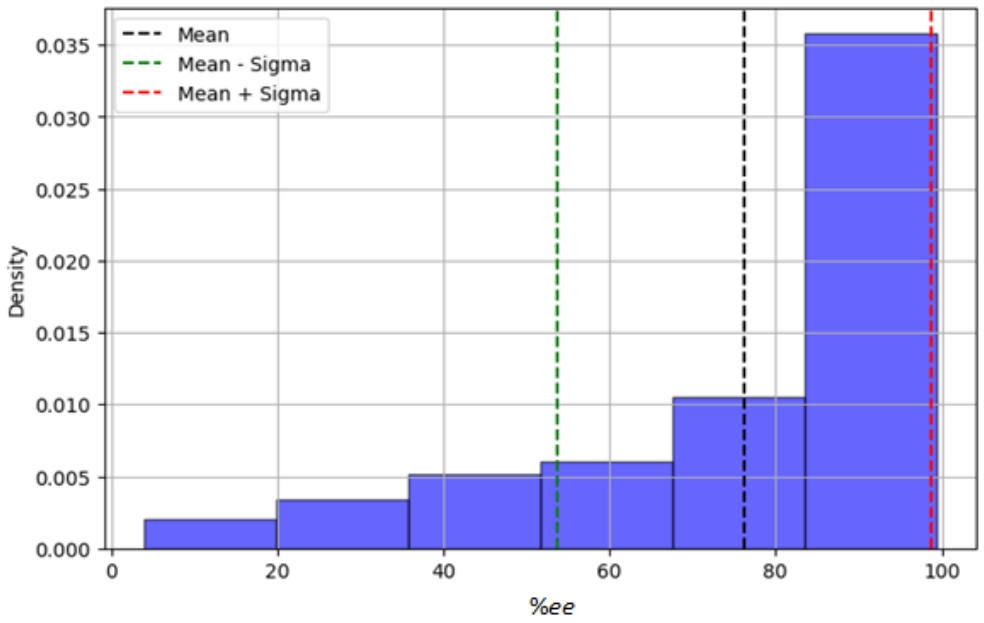


Figure S2. Distribution of %*ee* in our ART dataset.

(4.1) Classification performances

For classification tasks, two featurization techniques are taken into consideration. The first includes fingerprint-based molecular representations using the Morgan fingerprints of radius 2, hashed atom pair fingerprints, layered fingerprints, etc. The second approach uses a 400-

dimensional feature vector from the ULMFiT encoder applied to the concatenated reaction SMILES. These features are then used to train a deep neural network (DNN) classifier to distinguish between majority/minority class reactions. A randomised 70:10:20 train-validation-test split is used for model training with Deep Neural Network (DNN) classifier. Hyperparameter optimization is performed using the Optuna framework, which systematically evaluates combinations of hyperparameter values to identify the best-performing configuration (Tables S75 and S76). The DNN comprised of multiple fully connected layers, including an input layer, one or more hidden layers, and a single output layer, and is trained in a supervised manner using backpropagation with the Adam optimizer. Key hyperparameters influencing DNN performance include the number of hidden layers, number of neurons, learning rate, and dropout rate. To ensure robustness, the model is trained on 30 independent random splits, with the number of epochs set to 300 as the default parameter of the DNN classifier to distinguish between majority/minority class reactions. We evaluated the classification models based on accuracy and F1-score, with the results summarized in the following tables.

Table S75. Classification Performance of the DNN Algorithm Using the ULMFiT Encoding Vector on the ART Dataset

Split	accuracy			F1			best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
	train	validation	test	train	validation	test	
1	0.99	0.92	0.93	0.99	0.95	0.96	5,209,0.18,0.00002
2	1.00	0.95	0.91	1.00	0.97	0.94	5,672,0.30,0.00002
3	1.00	0.97	0.93	1.00	0.98	0.96	2,868,0.04,0.00001
4	1.00	1.00	0.93	1.00	1.00	0.96	3,730,0.43,0.03550
5	0.98	1.00	0.95	0.99	1.00	0.97	2,853,0.43,0.07712
6	1.00	0.92	0.93	1.00	0.96	0.96	2,532,0.06,0.00002
7	0.98	0.95	0.92	0.99	0.97	0.95	3,387,0.46,0.00002
8	1.00	0.89	0.95	1.00	0.94	0.97	1,657,0.36,0.00003
9	1.00	0.95	0.95	1.00	0.97	0.97	5,657,0.33,0.00021
10	1.00	0.95	0.92	1.00	0.97	0.95	4,733,0.11,0.00003

11	1.00	1.00	0.89	1.00	1.00	0.94	3,699,0.48,0.00063
12	0.99	0.87	0.91	0.99	0.92	0.95	3,439,0.43,0.00002
13	1.00	0.97	0.96	1.00	0.98	0.98	1,695,0.29,0.00021
14	1.00	1.00	0.97	1.00	1.00	0.98	2,104,0.29,0.02022
15	0.99	0.89	0.92	1.00	0.94	0.95	4,423,0.24,0.00001
16	0.98	0.97	0.97	0.99	0.99	0.98	2,532,0.31,0.09207
17	1.00	1.00	0.93	1.00	1.00	0.96	4,852,0.41,0.00032
18	1.00	0.92	0.93	1.00	0.95	0.96	2,831,0.15,0.00001
19	1.00	0.95	0.96	1.00	0.97	0.98	1,281,0.06,0.00009
20	1.00	0.95	0.99	1.00	0.97	0.99	1,945,0.25,0.00005
21	0.97	0.95	0.91	0.98	0.97	0.95	3,516,0.46,0.00001
22	0.99	0.97	0.95	0.99	0.98	0.97	5,395,0.35,0.00002
23	0.96	0.97	0.83	0.98	0.99	0.89	5,491,0.18,0.09918
24	1.00	0.87	0.97	1.00	0.92	0.98	2,572,0.44,0.00002
25	1.00	0.95	0.91	1.00	0.97	0.95	3,291,0.27,0.00003
26	1.00	0.95	0.91	1.00	0.96	0.94	1,256,0.04,0.00016
27	1.00	1.00	0.93	1.00	1.00	0.96	5,888,0.46,0.00037
28	1.00	0.97	0.95	1.00	0.98	0.97	2,217,0.39,0.00008
29	1.00	0.95	0.97	1.00	0.97	0.98	5,531,0.06,0.00095
30	1.00	1.00	0.92	1.00	1.00	0.95	4,894,0.46,0.00048
avg. ±s.d.	0.99 ±0.01	0.95 ±0.04	0.93 ±0.03	1.00 ±0.00	0.97 ±0.02	0.96 ±0.02	

Table S76. Classification Performance of the DNN Algorithm Using Fingerprint Vector on the ART Dataset

Split	accuracy			F1			best hyperparameters {n_layers, n_neurons, dropout_rate, learning_rate}
	train	validation	test	train	validation	test	
1	1.00	0.95	0.95	1.00	0.97	0.97	5,101,0.16,0.00002
2	1.00	1.00	0.95	1.00	1.00	0.96	5,539,0.27,0.02347
3	1.00	0.92	0.97	1.00	0.95	0.98	5,365,0.02,0.00002
4	0.98	0.97	0.92	0.99	0.98	0.95	1,985,0.13,0.06181
5	1.00	1.00	0.93	1.00	1.00	0.96	1,212,0.27,0.00016
6	1.00	0.95	0.95	1.00	0.97	0.97	4,394,0.10,0.00001
7	1.00	0.95	0.96	1.00	0.97	0.97	3,100,0.41,0.00001
8	1.00	0.89	0.93	1.00	0.93	0.96	3,103,0.37,0.00002
9	1.00	0.97	0.96	1.00	0.98	0.98	4,369,0.13,0.00002
10	1.00	0.97	0.95	1.00	0.98	0.97	1,982,0.47,0.03659
11	1.00	1.00	0.91	1.00	1.00	0.94	4,248,0.01,0.03383
12	1.00	0.97	0.95	1.00	0.98	0.97	1,228,0.28,0.00008
13	1.00	0.92	0.99	1.00	0.95	0.99	1,523,0.39,0.00002
14	1.00	1.00	0.93	1.00	1.00	0.96	3,942,0.47,0.00039
15	1.00	0.97	0.96	1.00	0.98	0.97	5,409,0.24,0.0018
16	1.00	0.97	0.97	1.00	0.98	0.98	4,168,0.23,0.00001

17	1.00	1.00	0.93	1.00	1.00	0.95	4,980,0.47,0.00010
18	1.00	0.92	0.99	1.00	0.95	0.99	4,509,0.03,0.00001
19	1.00	0.97	0.97	1.00	0.98	0.98	3,875,0.00,0.00001
20	1.00	0.92	1.00	1.00	0.96	1.00	3,109,0.06,0.00002
21	1.00	0.95	0.95	1.00	0.97	0.97	1,156,0.31,0.00009
22	1.00	0.97	0.96	1.00	0.98	0.97	5,164,0.08,0.00003
23	1.00	1.00	0.84	1.00	1.00	0.89	4,973,0.50,0.00050
24	1.00	0.95	0.97	1.00	0.97	0.98	1,509,0.00,0.00001
25	1.00	0.95	0.92	1.00	0.96	0.95	2,310,0.01,0.00001
26	1.00	0.97	0.96	1.00	0.98	0.98	1,822,0.11,0.00001
27	1.00	1.00	0.96	1.00	1.00	0.98	5,601,0.40,0.00031
28	1.00	0.95	0.95	1.00	0.97	0.97	2,305,0.17,0.00001
29	1.00	0.95	1.00	1.00	0.97	1.00	5,218,0.00,0.00005
30	1.00	1.00	0.95	1.00	1.00	0.97	3,417,0.19,0.01615
avg. ±s.d.	1.00 ±0.00	0.96 ±0.03	0.95 ±0.03	1.00 ±0.00	0.98 ±0.02	0.97 ±0.02	

(4.2) Regression performances for CFR-major and CFR-minor

(4.2.1) ULMFiT

Table S77. Hyperparameter Tuning for the ULMFiT Regressor for CFR-major Class. Performances are Reported in RMSE Metrics[i]

no. of augmented smiles	batch size	noise (σ_g)	drop out	epochs	learning rate	train	validation	test (canonical)	test (TTA)
varying the no. of augmented smiles									
50	64	0.0	0.0	20	0.001	6.04	6.24	7.47	7.54
75	64	0.0	0.0	20	0.001	5.84	5.60	8.50	8.10
100	64	0.0	0.0	20	0.001	6.00	5.68	8.35	8.03
110	64	0.0	0.0	20	0.001	5.80	5.65	8.77	8.31
120	64	0.0	0.0	20	0.001	5.50	6.28	8.80	8.47
130	64	0.0	0.0	20	0.001	5.68	5.19	8.32	8.06
140	64	0.0	0.0	20	0.001	5.34	6.36	8.77	8.60
150	64	0.0	0.0	20	0.001	5.22	6.03	8.79	8.61
varying the noise (σ_g)									
100	64	0.1	0.0	20	0.001	6.00	6.23	8.82	8.27
100	64	0.2	0.0	20	0.001	5.96	5.85	8.95	8.33
100	64	0.3	0.0	20	0.001	5.99	5.50	8.64	8.17
100	64	0.4	0.0	20	0.001	6.02	5.93	9.03	8.43
100	64	0.5	0.0	20	0.001	5.99	5.41	8.65	8.20
100	64	0.6	0.0	20	0.001	6.04	5.93	8.81	8.29
100	64	0.7	0.0	20	0.001	5.99	6.02	8.84	8.47
100	64	0.8	0.0	20	0.001	5.95	5.71	8.89	8.50
varying the dropout rate									
100	64	0.3	0.1	20	0.001	6.08	5.75	8.24	7.96
100	64	0.3	0.2	20	0.001	6.21	5.90	8.78	7.84

100	64	0.3	0.3	20	0.001	6.30	5.63	8.47	7.58
100	64	0.3	0.4	20	0.001	6.36	5.95	8.61	7.79
100	64	0.3	0.5	20	0.001	6.48	5.93	8.67	7.50
varying the learning rate									
100	64	0.3	0.1	20	0.01	8.63	7.52	12.83	9.38
100	64	0.3	0.1	20	0.001	6.08	5.75	8.24	7.96
100	64	0.3	0.1	20	0.0001	78.46	77.70	77.90	78.13
100	64	0.3	0.1	20	0.02	10.19	10.06	10.98	10.11
varying the no. of epochs									
100	64	0.3	0.1	20	0.001	6.08	5.75	8.24	7.96
100	64	0.3	0.1	30	0.001	5.78	5.96	8.85	8.38
100	64	0.3	0.1	40	0.001	5.73	6.47	9.09	8.43
100	64	0.3	0.1	50	0.001	5.49	7.03	9.18	8.54

[i]The values shown in red color represent the best hyperparameter for the regressor

Table S78. Hyperparameter Tuning for the ULMFiT Regressor for CFR-minor Class. Performances are Reported in RMSE Metrics[i]

no. of augmented smiles	batch size	noise (σ_g)	drop out	epochs	learning rate	train	validation	test (canonical)	test (TTA)
varying the no. of augmented smiles									
50	32	0.0	0.0	20	0.001	4.91	13.63	9.07	9.94
75	32	0.0	0.0	20	0.001	6.87	10.77	7.73	9.10
100	32	0.0	0.0	20	0.001	7.59	14.30	8.77	9.58
110	32	0.0	0.0	20	0.001	7.35	13.50	8.95	9.43
120	32	0.0	0.0	20	0.001	7.34	15.24	9.36	9.74
130	32	0.0	0.0	20	0.001	7.53	13.55	8.73	9.41
140	32	0.0	0.0	20	0.001	6.80	14.58	8.21	9.10
150	32	0.0	0.0	20	0.001	6.53	13.09	8.63	9.48
varying the noise (σ_g)									
75	32	0.1	0.0	20	0.001	6.89	14.70	7.85	9.44
75	32	0.2	0.0	20	0.001	6.86	14.05	9.14	9.55
75	32	0.3	0.0	20	0.001	6.86	12.84	8.57	9.17
75	32	0.4	0.0	20	0.001	6.89	15.55	9.24	9.49
75	32	0.5	0.0	20	0.001	6.85	14.46	8.80	9.55
75	32	0.6	0.0	20	0.001	6.87	13.48	8.00	9.10
75	32	0.7	0.0	20	0.001	6.87	13.44	8.18	9.28
75	32	0.8	0.0	20	0.001	6.87	12.83	8.74	9.56
varying the dropout rate									
75	32	0.5	0.1	20	0.001	6.88	13.30	8.41	9.89
75	32	0.5	0.2	20	0.001	6.97	14.39	8.73	10.07
75	32	0.5	0.3	20	0.001	7.05	17.57	9.45	10.11
75	32	0.5	0.4	20	0.001	7.11	13.45	7.98	9.92
75	32	0.5	0.5	20	0.001	7.26	15.93	8.14	9.95
varying the learning rate									

75	32	0.5	0.1	20	0.1	13.13	16.40	14.56	14.57
75	32	0.5	0.1	20	0.01	8.41	13.88	12.94	10.97
75	32	0.5	0.1	20	0.001	6.88	13.30	8.41	9.89
75	32	0.5	0.1	20	0.0001	34.75	37.50	35.50	35.46
varying the no. of epochs									
75	32	0.5	0.1	20	0.001	6.88	13.30	8.41	9.89
75	32	0.5	0.1	30	0.001	8.00	13.76	8.81	9.51
75	32	0.5	0.1	40	0.001	7.47	13.48	7.24	8.95

[r]The values shown in red color represent the best hyperparameter for the regressor

Table S79. Performance of ULMFiT Algorithm for CFR-major Class in terms of RMSE across Different Splits

split	train	validation	test(canonical)	test(TTA)
1	6.09	5.75	8.24	7.96
2	6.33	9.70	9.36	9.04
3	6.29	6.57	7.37	6.98
4	6.37	6.90	8.39	8.49
5	6.40	7.40	10.03	9.86
6	8.29	7.97	9.06	8.92
7	6.38	7.38	8.27	7.54
8	6.20	7.53	7.58	7.27
9	6.21	10.12	7.75	6.59
10	6.02	7.34	8.62	8.10
11	6.04	7.67	9.37	9.37
12	6.14	9.51	9.80	9.40
13	6.64	7.72	8.14	7.69
14	6.68	10.38	8.45	7.69
15	6.06	10.12	6.81	6.28
16	5.93	10.68	9.02	8.74
17	6.02	8.13	8.02	7.57
18	5.84	9.61	8.20	7.98
19	6.31	8.39	6.31	5.88
20	6.22	8.74	9.22	8.22
21	6.42	8.09	8.60	7.80
22	6.76	9.27	7.66	7.17
23	6.27	8.06	8.97	8.58
24	6.44	9.49	9.31	7.53
25	6.06	7.12	8.35	8.34
26	6.51	10.62	7.62	7.89
27	5.96	8.26	10.20	8.42
28	5.78	8.21	9.37	8.01
29	6.20	8.50	8.73	8.05
30	6.32	8.99	7.64	7.45
avg.±s.d.	6.31±0.44	8.47±1.26	8.48±0.91	7.96±0.90

Table S80. Performance of ULMFiT Algorithm for CFR-minor Class in terms of RMSE across Different Splits

split	train	validation	test(canonical)	test(TTA)
1	6.88	13.30	8.41	9.89
2	8.57	10.50	10.13	9.00
3	8.08	6.58	13.36	12.50
4	7.68	5.64	14.40	12.95
5	8.96	12.43	9.38	9.61
6	8.13	13.88	6.15	7.86
7	8.19	12.68	7.44	8.16
8	8.71	6.79	9.58	10.03
9	6.79	9.45	13.27	11.75
10	8.76	12.31	8.39	9.88
11	8.76	4.25	4.99	6.27
12	7.71	5.39	13.98	14.05
13	7.97	4.23	10.16	11.32
14	7.85	11.17	10.58	10.50
15	9.37	7.89	10.64	11.39
16	8.68	16.95	10.07	11.97
17	9.26	6.04	10.73	12.38
18	7.37	12.54	9.51	9.83
19	7.91	14.72	10.14	11.38
20	8.61	19.16	9.97	9.57
21	6.62	14.87	7.15	7.33
22	8.47	4.55	8.40	8.68
23	8.43	12.83	9.20	9.27
24	8.40	7.08	10.11	11.17
25	8.06	10.36	11.06	11.41
26	8.98	12.59	13.67	11.96
27	8.06	5.33	12.75	13.31
28	7.35	9.70	11.45	12.38
29	7.88	11.21	10.75	12.12
30	8.56	9.83	8.67	7.82
avg.±s.d.	8.17±0.69	10.14±3.96	10.15±2.26	10.52±1.92

(4.2.2) AttentiveFP

Table S81. Performance of AttentiveFP Algorithm for CFR-major Class in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps}
1	6.64	6.92	9.13	3,242,0.05,0.00023,2
2	8.10	8.25	11.18	2,451,0.10,0.00003,1

3	6.95	5.65	8.53	2,457,0.49,0.00057,2
4	5.75	8.69	9.77	1,168,0.05,0.00317,2
5	5.41	6.20	9.53	3,465,0.46,0.00165,1
6	5.86	5.79	10.79	2,264,0.20,0.00207,3
7	6.55	6.46	8.67	3,150,0.19,0.00098,1
8	4.95	6.37	8.36	3,327,0.01,0.00034,1
9	5.47	9.06	8.18	2,144,0.03,0.00207,2
10	5.49	6.87	9.21	5,380,0.16,0.00097,2
11	5.05	9.32	8.85	2,333,0.04,0.00043,2
12	7.06	9.78	8.58	1,215,0.12,0.00021,2
13	6.94	8.26	8.33	4,185,0.39,0.00093,2
14	5.64	10.64	7.94	5,342,0.21,0.00030,1
15	7.33	8.39	7.84	1,422,0.34,0.00378,1
16	9.20	11.11	8.80	1,104,0.13,0.07427,1
17	8.25	6.54	8.51	2,287,0.14,0.00006,2
18	6.24	9.93	8.50	2,273,0.18,0.00034,3
19	6.12	8.46	7.17	4,373,0.30,0.00249,2
20	3.77	5.60	8.84	4,330,0.01,0.00042,3
21	6.36	8.97	9.57	4,159,0.21,0.00100,2
22	6.76	8.80	9.95	3,347,0.00,0.00078,3
23	4.65	7.86	8.83	2,331,0.00,0.00168,2
24	7.24	7.57	8.04	2,480,0.00,0.00005,3
25	7.21	7.60	8.95	1,366,0.02,0.00006,2
26	6.15	10.62	7.55	2,174,0.33,0.00170,1
27	6.87	7.52	8.01	4,151,0.36,0.00072,1
28	7.31	7.96	7.70	1,320,0.29,0.00318,3
29	5.71	8.34	8.90	4,427,0.30,0.00201,1
30	4.62	9.63	7.85	3,462,0.06,0.00060,2
avg.±s.d.	6.32±1.17	8.11±1.53	8.74±0.90	

Table S82. Performance of AttentiveFP Algorithm for CFR-minor Class in terms of RMSE across Different Splits

split	train	validation	test	best hyperparameters {num_layers, graph_feat_size, dropout_rate, learning_rate, num_timesteps}
1	11.13	10.01	11.73	4,285,0.40,0.00009,2
2	3.43	11.20	10.44	3,149,0.14,0.00500,1
3	7.62	11.36	12.69	5,439,0.01,0.00200,2
4	5.52	7.30	11.77	2,166,0.50,0.00400,1
5	11.12	8.53	12.73	3,263,0.32,0.00002,1
6	6.88	11.07	7.63	5,286,0.09,0.00020,3
7	11.40	8.80	10.75	5,198,0.49,0.00004,2
8	7.43	7.71	8.29	5,326,0.19,0.00400,3
9	4.73	9.31	9.31	4,259,0.34,0.00060,2
10	6.95	14.64	10.08	1,318,0.49,0.00090,3
11	10.25	11.30	10.98	3,325,0.17,0.05000,2

12	12.83	6.54	10.58	4,261,0.32,0.00001,1
13	10.91	8.11	11.16	3,369,0.28,0.00800,2
14	8.52	9.43	12.90	5,384,0.34,0.00800,1
15	10.73	8.38	11.92	5,128,0.48,0.00007,3
16	11.03	9.32	10.39	4,348,0.41,0.00002,1
17	2.35	4.58	11.56	5,431,0.00,0.00050,2
18	11.54	11.09	8.69	3,352,0.50,0.00003,2
19	3.90	9.86	12.79	1,329,0.30,0.00100,3
20	10.39	15.07	11.16	5,254,0.13,0.00001,3
21	6.72	13.60	10.36	2,338,0.27,0.00040,1
22	3.05	4.55	10.91	3,251,0.38,0.00200,1
23	5.13	10.04	5.31	4,443,0.40,0.00060,2
24	7.34	8.00	12.96	4,257,0.30,0.00200,2
25	8.60	11.14	8.79	3,138,0.23,0.01000,3
26	5.87	15.07	11.70	3,129,0.14,0.01000,2
27	12.20	5.94	15.56	4,446,0.40,0.02000,3
28	11.42	9.95	9.60	5,313,0.41,0.00002,2
29	7.02	7.89	11.21	4,201,0.46,0.00030,3
30	6.38	11.34	8.29	1,410,0.49,0.00100,1
avg.±s.d.	8.08±3.02	9.70±2.71	10.74±1.98	

4.3. Regression performances for CFR-major and CFR-minor classes obtained using the Attentive-FP algorithm

Table S83. Performance of AttentiveFP Algorithm for CFR major and minor Classes in terms of RMSE across Different Class Boundaries (indicated in column-1 **B**)

B	samples	train	validation	test	train R^2	validation R^2	test R^2
30	27(minor)	4.97±2.78	6.82±2.99	9.13±1.79	0.56±0.37	-0.31±0.53	-0.54±0.95
30	349(major)	6.68±1.44	8.91±1.48	9.18±1.16	0.83±0.08	0.66±0.16	0.67±0.10
40	37(minor)	5.75±2.68	6.98±3.20	11.49±2.50	0.62±0.31	-0.25±0.91	-0.56±0.89
40	339(major)	6.65±1.78	8.43±1.57	9.07±1.15	0.78±0.13	0.64±0.14	0.60±0.10
50	60 (minor)	7.36±2.89	8.10±2.99	10.69±2.13	0.66±0.23	0.31±0.60	0.34±0.23
50	316(major)	6.07±1.10	7.74±1.43	8.13±1.07	0.71±0.11	0.46±0.27	0.47±0.16
54	65(minor)	7.58±2.74	8.89±2.39	11.24±2.39	0.66±0.20	0.30±0.56	0.29±0.33
54	311(major)	6.58±1.22	8.05±1.61	8.42±0.88	0.62±0.14	0.37±0.24	0.35±0.18
76	130(minor)	6.65±2.07	10.98±2.61	12.42±1.85	0.87±0.09	0.55±0.37	0.59±0.14
76	246(major)	4.00±0.56	4.89±0.71	4.74±0.45	0.37±0.17	-0.01±0.35	0.06±0.15

4.4. Weighted CFR performance

To ensure a fair evaluation of the CFR approach, we have also evaluated a weighted RMSE/R^2 in addition to a direct RMSE/R^2. Since the dataset is imbalanced, with significantly different sample sizes in the CFR-major and CFR-minor classes, a direct RMSE would be biased toward the majority class, masking the true predictive performance on the minority class. The weighted RMSE accounts for both classes by incorporating their respective sample sizes and error contributions, thereby providing a more representative and balanced assessment of the overall model performance.

The weighted RMSE and R^2 for the CFR approach is calculated as,

$$RMSE_{wa} = \sqrt{\frac{N_{major}\,(RMSE_{major})^2 + N_{minor}\,(RMSE_{minor})^2}{N_{major} + N_{minor}}}$$

$$R^2{}_{wa} = \frac{N_{major}R^2{}_{major} + N_{minor}R^2{}_{minor}}{N_{major} + N_{minor}}$$

where, N_{major} and N_{minor} are respectively the number of samples in the CFR-major and CFR-minor classes, while $\text{RMSE}_{\text{major}}$ and $\text{RMSE}_{\text{minor}}$ are the corresponding test RMSE values. $R^2{}_{major}$ and $R^2{}_{minor}$ are the coefficient of determination for the CFR major and minor classes respectively.

Table S84. Performance of AttentiveFP Algorithm in terms of $RMSE_{wa}$ and $R^2{}_{wa}$ across Different Class Boundaries (indicated in column-1 **B**)

B	Train $RMSE_{wa}$	Valid $RMSE_{wa}$	Test $RMSE_{wa}$	Train $R^2{}_{wa}$	Valid $R^2{}_{wa}$	Test $R^2{}_{wa}$
30	6.57±1.57	8.78±1.64	9.18±1.22	0.81±0.14	0.59±0.33	0.58±0.41
40	6.57±1.89	8.30±1.80	9.34±1.34	0.76±0.16	0.55±0.41	0.49±0.45
50	6.29±1.53	7.80±1.77	8.59±1.30	0.70±0.14	0.44±0.35	0.45±0.18
54	6.76±1.59	8.20±1.77	8.97±1.28	0.63±0.15	0.36±0.32	0.34±0.21
76	5.08±1.30	7.57±1.64	8.25±1.15	0.54±0.28	0.18±0.45	0.24±0.29

5. Choice of %*ee* vs. ΔΔG‡ in modeling enantioselectivity

In addition to using enantiomeric excess (%*ee*) as the target variable, the corresponding energy difference between the enantiocontrolling transition states (ΔΔG‡) is also considered for model training and evaluation.[24] The distribution of the ΔΔG‡ value for our ART dataset can be visualized in Figure S3. The results shown in Table S85 reveal that our AttentiveFP model achieves better performance with %*ee* as the training data, validating our choice as chemically more intuitive parameter despite the significance of ΔΔG‡.

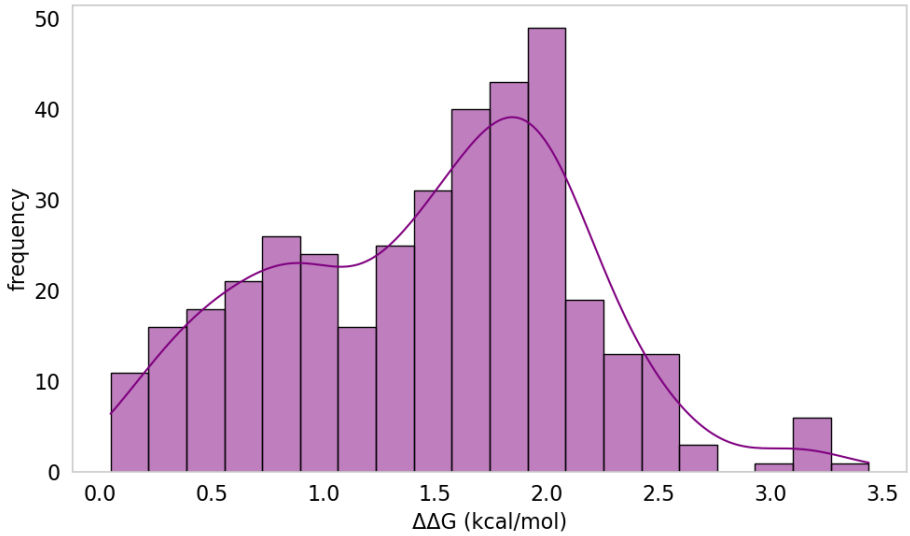


Figure S3. Distribution of ΔΔG‡ in our ART dataset.

Table S85. Performance of AttentiveFP Regressor in terms of RMSE and R^2 across Different Splits

output	train RMSE	validation RMSE	test RMSE	train R^2	validation R^2	test R^2
%*ee*	7.30±1.32	9.58±1.59	10.56±1.43	0.88±0.06	0.79±0.09	0.77±0.08
ΔΔG‡	0.19±0.08	0.40±0.09	0.43±0.05	0.90±0.08	0.64±0.14	0.58±0.10

We performed a similar analysis for the N,S-acetylation reaction dataset. To evaluate model performance, we first trained the AttentiveFP-CI model with varying class boundaries, chosen on the basis of the statistical distribution of the ΔΔG‡ values in the N,S-acetylation reaction dataset. The dataset comprises 1,075 reactions, with ΔΔG‡ as the target variable that ranges from -0.4 to 3.2 kcal/mol.

We trained the model under two data-splitting ratios, one 70:10:20 and the other 80:10:10 (train:validation:test), and then compared the results with previously reported baseline model. As shown in Tables S86 and S87, the AttentiveFP-CI model achieved a test R^2 of 0.90±0.02 and RMSE of 0.21±0.02 using the 80:10:10 split with a μ-based class boundary (0.98), comparable to the best-performing ChemAH model (R^2 = 0.915; RMSE = 0.197).[25]

Table S86. Performance of AttentiveFP-CI Algorithms For ΔΔG‡ Prediction in N,S-Acetylation Reaction Dataset (1075 Reactions) across Different Class Boundaries (Indicated in Column-1 B). Results are Averaged Over 10 Independent Random Splits Using a 70:10:20 Train-Validation-Test Scheme

B	RMSE			R^2		
model	train	validation	test	train	validation	test
AttentiveFP	0.17±0.06	0.2±0.02	0.23±0.02	0.94±0.03	0.92±0.02	0.89±0.02
AttentiveFP-CI						
0.98 (μ)	0.17±0.04	0.20±0.03	0.23±0.02	0.94±0.03	0.91±0.03	0.89±0.02
1.68 (μ+σ)	0.19±0.02	0.20±0.02	0.22±0.02	0.93±0.01	0.91±0.02	0.90±0.02
0.28 (μ-σ)	0.15±0.06	0.20±0.02	0.23±0.02	0.95±0.03	0.91±0.02	0.89±0.02

Table S87. Performance of AttentiveFP and AttentiveFP-CI Algorithms For ΔΔG‡ Prediction in N,S-Acetylation Reaction Dataset (1075 Reactions) across Different Class Boundaries (Indicated in Column-1 B). Results are Averaged Over 10 Independent Random Splits Using an 80:10:10 Train-Validation-Test Scheme

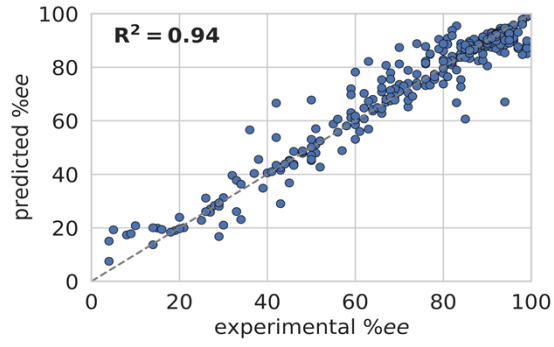
B	RMSE			R^2		
model	train	validation	test	train	validation	test
AttentiveFP	0.18±0.05	0.22±0.02	0.22±0.02	0.93±0.02	0.89±0.02	0.90±0.02

AttentiveFP-CI						
0.98(μ)	0.19±0.01	0.21±0.02	0.21±0.02	0.93±0.01	0.90±0.02	0.90±0.02
1.68 (μ+σ)	0.18±0.03	0.22±0.02	0.22±0.02	0.93±0.02	0.90±0.02	0.90±0.02
0.28 (μ-σ)	0.19±0.02	0.22±0.02	0.22±0.02	0.93±0.01	0.90±0.02	0.90±0.02

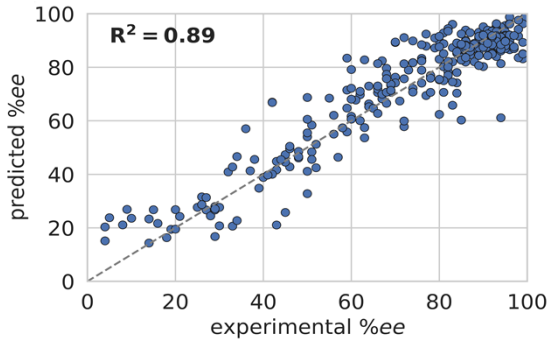
6. Experimental versus AttentiveFP-CI predicted %*ee* parity plot

We evaluated the performance of the AttentiveFP-CI, with a class boundary of 50, by computing R^2 values as obtained from the parity plots of experimental and predicted %*ee* values. The test set predictions is gathered from 30 independently trained models. Our dataset comprises 376 distinct reactions; however, due to the random partitioning into training and test sets across the models, each reaction may appear in the test set multiple times or just once. To ensure a fair comparison, we selected a single representative prediction per reaction, thus forming a unique reaction set. This representative was identified using ranking criteria e.g., the prediction with the smallest absolute deviation from the experimental value ('top-1'), the second smallest ('top-2'), and so forth.[26] This method resulted in a top-1 R^2 of 0.94, indicating a strong correlation between predicted and actual %*ee* values. Additional metrics (top-2 to top-5 R^2 = 0.89-0.72) and mean predictions (averaging all predictions per reaction) R^2 (0.84) further confirmed good confidence of the AttentiveFP-CI in enantioselectivity (%*ee*) predictions (Figure S4).

(a)



(b)



(c)

(d)

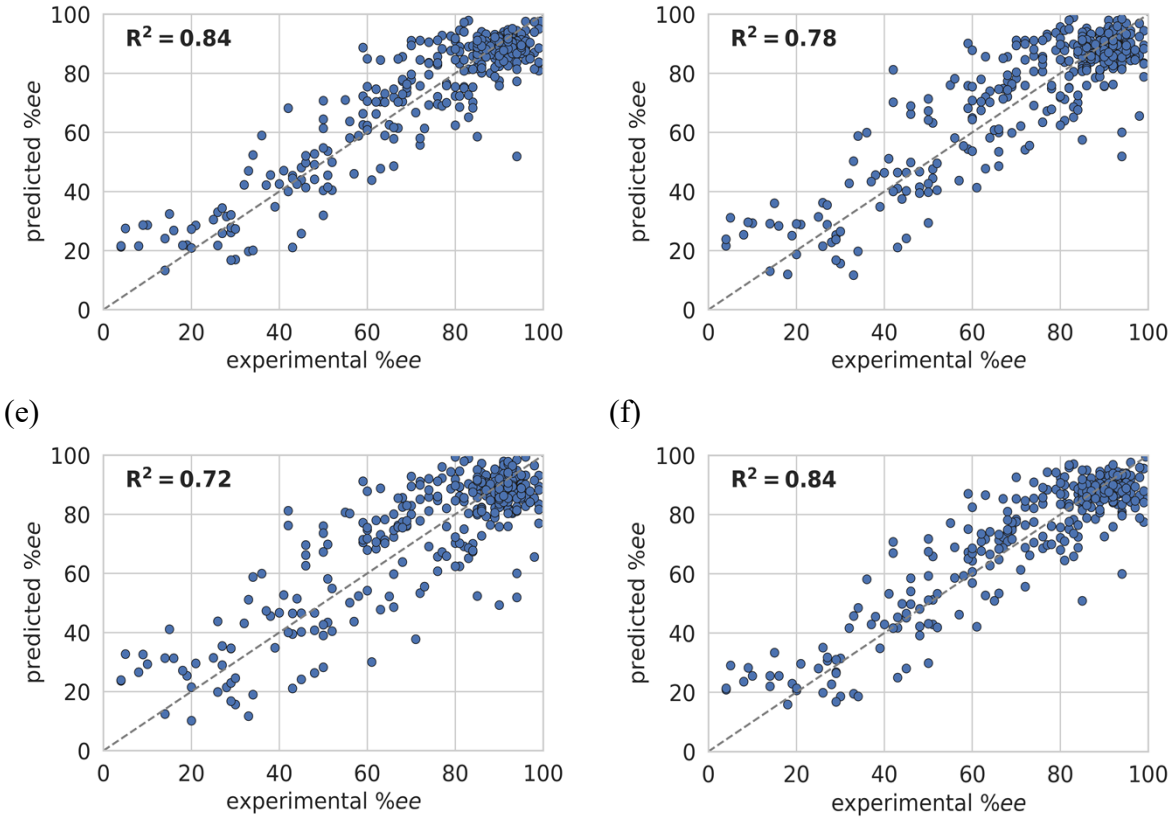


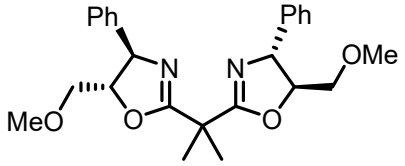
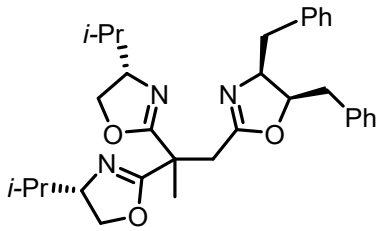
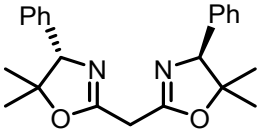
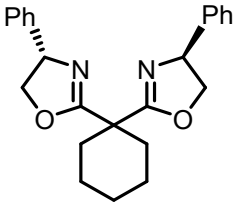
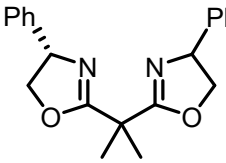
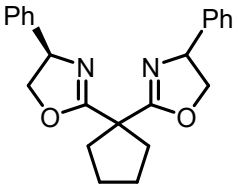
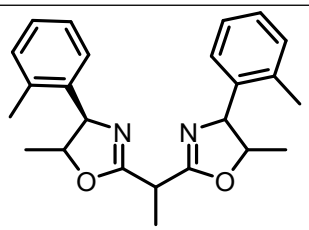
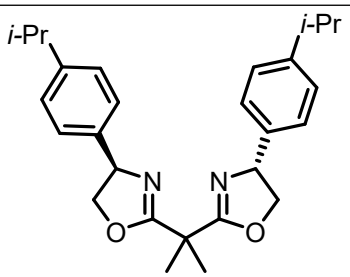
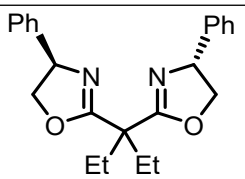
Figure S4. Parity plot with R^2 illustrating correlation between predicted and experimental %*ee* obtained using AttentiveFP-CI algorithm with the class boundary of 50, across (a) top-1, (b) top-2, (c) top-3, (d) top-4, (e) top-5 ranked, as well as (f) mean predictions.

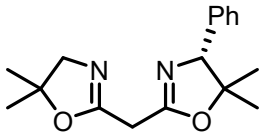
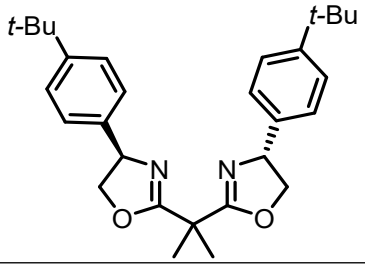
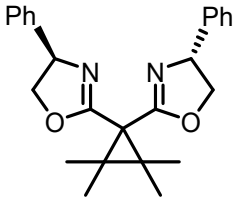
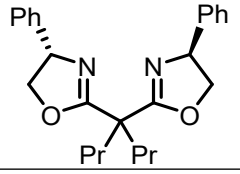
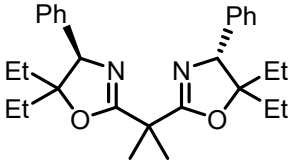
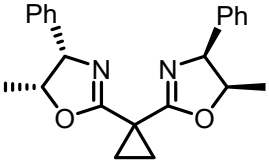
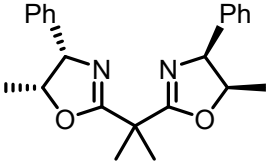
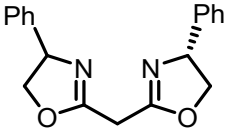
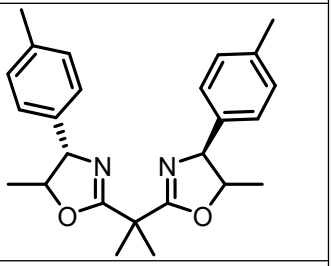
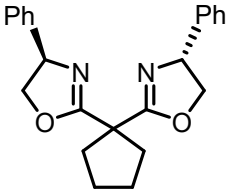
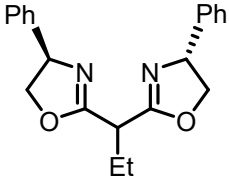
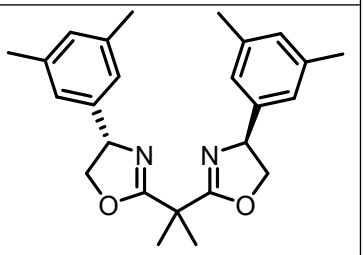
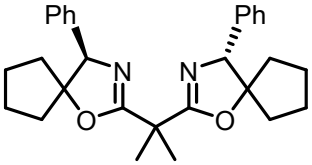
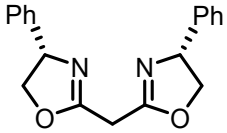
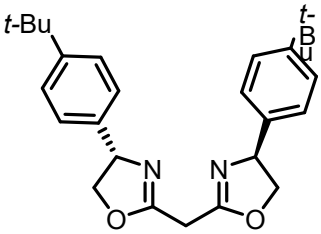
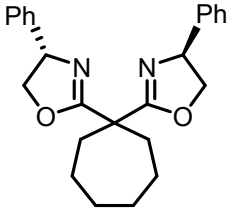
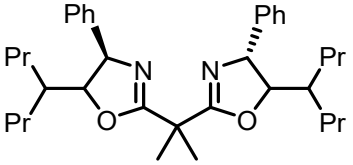
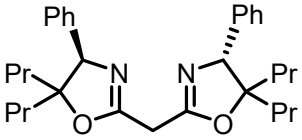
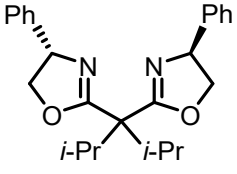
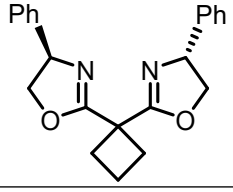
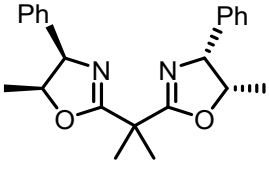
7. Predicted %*ee* for chiral ligands obtained from PubChem database

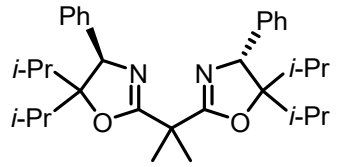
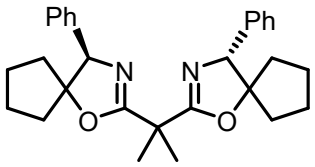
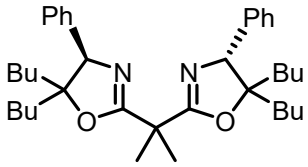
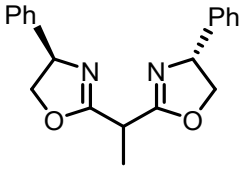
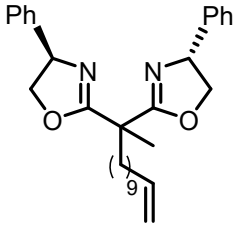
To expand the accessible ligand space beyond experimentally tested systems, we conducted a virtual screening campaign to identify and evaluate new chiral ligands from the PubChem database. A total of 164 oxazoline-based ligands are initially retrieved based on their structural similarity to the (*S,S*)-Ph-Box scaffold. These candidates are subjected to a rigorous cheminformatics filtration protocol implemented using RDKit, in which raw SMILES strings are sanitized to remove salts and solvents, retaining only the largest organic fragment. Subsequent screening filters these fragments to ensure they contain only permitted elements (C, H, O, N, halogens, S, P), possess a neutral formal charge, and exhibit defined stereochemistry with at least one confirmed chiral center. Qualifying ligands are then

complexed with a Cu(I) triflate precursor *in silico*; this is achieved by identifying two neutral nitrogen donor atoms, adjusting formal charges for coordinate bonding, and establishing bonds to the central Cu atom. The resulting metal complexes are canonicalized and cross-referenced against the existing training dataset to remove duplicates. This process yields a final set of 35 unique, stereochemically defined transition metal-ligand complexes. These complexes are subsequently combined with the alkene and coupling partner as the substrates of the reaction (detailed in Fig. 6 in the manuscript) to generate full reaction inputs. Finally, these inputs are evaluated using our main AttentiveFP-CI model to identify the potential candidates. The predicted %*ee* values with their corresponding ligand are summarized in the Table S88 below.

Table S88. List of Chiral Ligand Structure with their Predicted %*ee* using AttentiveFP-CI

		
94	100	98
		
94	90	89
		
86	92	90

		
58	101	87
		
93	98	89
		
91	85	87
		
90	87	95
		
97	98	100
		
95	98	96
		

96	89	91
		
102	97	102
		
86	95	

8. Performance of AttentiveFP-CI for N,S-acetylation and asymmetric hydrogenation

Table S89. Performance of AttentiveFP and AttentiveFP-CI Algorithms on N,S-Acetylation Reaction in Predicting %*ee*

model (class boundary)	train RMSE	validation RMSE	test RMSE	train R^2	valid R^2	test R^2
AttentiveFP	6.74±0.88	7.81±1.13	8.57±0.82	0.94±0.02	0.91±0.02	0.90±0.02
AttentiveFP-CI						
89 (μ+σ)	6.49±0.90	7.61±1.00	8.06±0.69	0.94±0.02	0.92±0.02	0.91±0.02
40(μ-σ)	6.51±0.85	7.53±0.90	8.18±0.69	0.94±0.02	0.92±0.02	0.91±0.02
61(μ)	6.42±1.05	7.58±0.83	8.19±0.77	0.94±0.02	0.92±0.02	0.91±0.02
50	6.43±0.69	7.57±0.99	8.10±0.71	0.94±0.01	0.92±0.02	0.91±0.02
70	6.24±0.81	7.57±0.83	8.19±0.72	0.95±0.01	0.92±0.02	0.91±0.02

Table S90. Performance of AttentiveFP and AttentiveFP-CI Algorithms on Asymmetric Hydrogenation Reaction in Predicting %*ee*

model (class boundary)	train RMSE	validation RMSE	test RMSE	train R^2	valid R^2	test R^2
AttentiveFP	7.50±2.00	10.0±2.70	11.0±2.50	0.79±0.12	0.50±0.24	0.52±0.30
AttentiveFP-CI						
30	8.40±2.60	10.0±2.80	11.0±2.40	0.73±0.19	0.51±0.27	0.56±0.21
40	7.80±2.40	11.0±3.40	11.0±3.20	0.77±0.16	0.39±0.56	0.49±0.33
50	8.10±2.80	11.0±3.70	12.0±3.50	0.75±0.18	0.43±0.47	0.48±0.35
60	7.90±1.90	11.0±3.30	11.0±3.00	0.77±0.11	0.45±0.37	0.53±0.24
70	7.80±2.40	9.60±2.80	11.0±1.80	0.77±0.15	0.57±0.26	0.53±0.30
80	7.50±2.00	11.0±3.40	11.0±2.20	0.79±0.12	0.36±0.48	0.53±0.29
90	7.20±1.10	10.0±1.90	10.0±1.60	0.81±0.06	0.50±0.29	0.58±0.23
69(μ-σ)	7.30±1.10	10.0±2.70	11.0±1.70	0.80±0.07	0.54±0.28	0.57±0.23
95	7.64±1.04	10.3±2.56	10.5±2.14	0.79±0.06	0.47±0.36	0.60±0.17

98	8.17±2.44	10.3±3.16	10.8±2.23	0.75±0.15	0.50±0.26	0.53±0.30
88(μ)	7.73±2.06	9.84±2.28	10.8±1.84	0.78±0.13	0.53±0.27	0.551±0.24

9. References

1. D. T. Ahneman, J. G. Estrada, S. Lin, S. D. Dreher and A. G. Doyle, *Science*, 2018, **360**, 186–190

2. S. Singh, M. Pareek, A. Changotra, S. Banerjee, B. Bhaskararao, P. Balamurugan and R. B. Sunoj, *Proc. Natl. Acad. Sci. U. S. A.*, 2020, **117**, 1339–1345.

3. A. F. Zahrt, J. J. Henle, B. T. Rose, Y. Wang, W. T. Darrow and S. E. Denmark, *Science*, 2019, **363**, eaau5631.

4. D. M. Lowe, Extraction of Chemical Structures and Reactions from the Literature. Ph.D. thesis, University of Cambridge, 2012.

5. S. Singh and R. B. Sunoj, *Digit. Discov.*, 2022, **1**, 303–312.

6. P. Schwaller, A. C. Vaucher, T. Laino and J.-L. Reymond, *Mach. Learn. Sci. Technol.*, 2021, **2**, 015016.

7. C. K. I. Williams, *arXiv*, 2024, DOI:10.48550/arxiv.2404.18190

8. F. Sandfort, F. Strieth-Kalthoff, M. Kühnemund, C. Beecks and F. Glorius, *Chem*, 2020, **6**, 1379-1390.

9. D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.

10. S. Shilpa, G. Kashyap and R. B. Sunoj, *J. Phys. Chem. A*, 2023, **127**, 8253–827.

11. D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.

12. D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik and R. P. Adams, *arXiv*, 2015, DOI:10.48550/arXiv.1509.09292

13. M. Li, J. Zhou, J. Hu, W. Fan, Y. Zhang, Y. Gu and G. Karypis, *ACS Omega*, 2021, **6**, 27233–27238.

14. Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang and M. Zheng, *J. Med. Chem.*, 2019, **63**, 8749–8760.

15. T. Akiba, S. Sano, T. Yanase, T. Ohta and M. Koyama, In *KDD*, 2019.

16. L. Breiman, *Mach. Learn.*, 2001, **45**, 5-32.

17. J. Lu and Y. Zhang, *J. Chem. Inf. Model.*, 2022, **62**, 1376–1387.

18. X. Lu, L. Xie, L. Xu, R. Mao, X. Xu and S. Chang, *Comput. Struct. Biotechnol. J*, 2024, **23**, 1666–1679.

19. Y. Kwon, D. Lee, Y.-S. Choi and S. Kang, *J. Cheminform.*, 2022, **14**, 2.

20. Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang and M. Zheng, *J. Med. Chem.*, 2019, **63**, 8749–8760.

21 (a) D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik and R. P. Adams, *arXiv*, 2015. DOI: https://arxiv.org/abs/1509.09292. (b) S. Kearnes, K. McCloskey, M. Berndl, V. Pande and P. Riley, *J. Comput. Aided Mol. Des.*, 2016, **30**, 595-608.

22. S. Ghosh, N. Jain and R. B. Sunoj, *arXiv*, 2025. DOI: https://arxiv.org/abs/2502.19976.

23. M. Noshad, L. Xu and A. Hero, *arXiv*, 2019. DOI: https://arxiv.org/abs/1909.07192.

24. M. Ruth, T. Gensch, and P. R. Schreiner, Contrasting Historical and Physical Perspectives in Asymmetric Catalysis: ΔΔG≠ versus Enantiomeric Excess. *Angew. Chem. Int. Ed.* 2024, **63**, e202410308.

25. L. Cheng, P.-L. Shao, J. Lv, H. Xiao, Y. Sun, J. Yang, Z. Xu, M. Lv, G. Wang, S. Zhao, J. Li, Z. Jin, X. Tan, G. Xing and B. Zhang, *Nat. Comput. Sci.*, DOI:10.1038/s43588-025-00920-8.

26. J. Schleinitz, A. Carretero-Cerdán, A. Gurajapu, Y. Harnik, G. Lee, A. Pandey, A. Milo and S. E. Reisman, *J. Am. Chem. Soc.*, 2025, **147**, 7476–7484.

---End of ESI File---