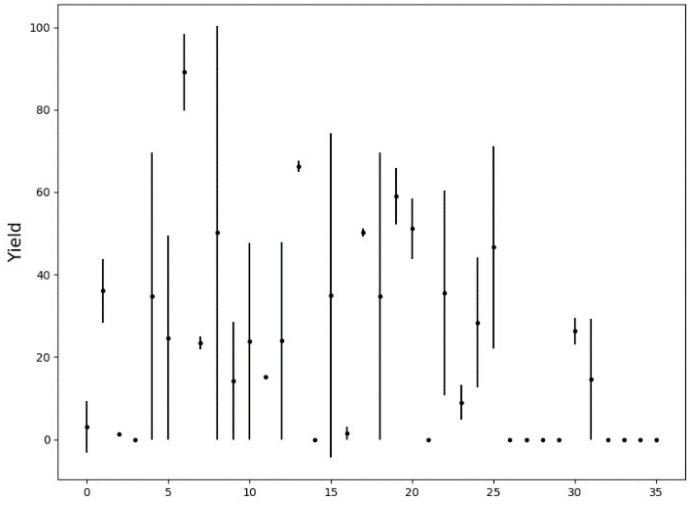
Supporting Information for:
On the Use of Real-World Data sets for Reaction Yield Prediction

Authors: Mandana Saebi,[‡a] Bozhao Nan,[‡b] John E. Herr,[b] Jessica Wahlers,[b] Zhichun Guo,[a] Andrzej M. Zurański,[c] Thierry Kogej,[d] Per-Ola Norrby,[e] Abigail G. Doyle,[c,f] Nitesh V. Chawla*[a] and Olaf Wiest*[b]
[a] Department of Computer Science and Engineering and Lucy Family Institute for Data and Society, University of Notre Dame, Notre Dame, IN 46556, USA.
[b] Department of Chemistry and Biochemistry, University of Notre Dame, Notre Dame, IN 46556, USA.
[c] Department of Chemistry, Princeton University, Princeton, NJ 08544, USA.
[d] Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Pepparedsleden 1, SE-431 83 Mölndal, Sweden
[e] Data Science and Modelling, Pharmaceutical Sciences, R&D, AstraZeneca, Gothenburg, Pepparedsleden 1, SE-431 83 Mölndal, Sweden
[f] Department of Chemistry and Biochemistry, University of California, Los Angeles, CA 90095, USA

Table of Contents:

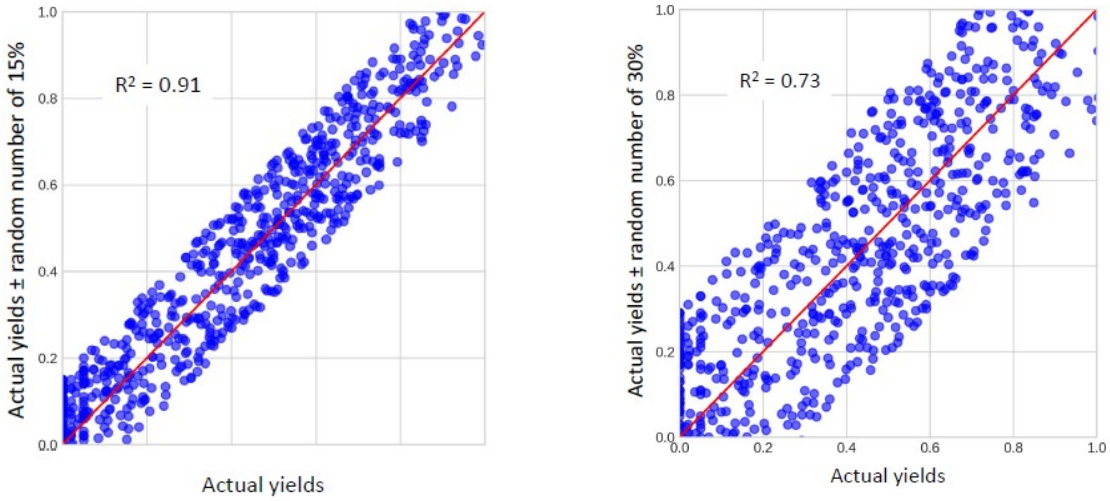
Generation and Curation of ELN dataset. The raw dataset was collected from the electronic laboratory notebooks at AstraZeneca using the NextMove software. To curate the raw data, a series of Jupyter notebooks were created, which can be found in the github repository. First, the original data format (.xml) was converted to the internally used library files. The scripts include several steps of data processing for automated curation of the dataset. Examples demonstrating the data format workflows for the generation of the features from the structures are described below. Next, the yield-related information was generated including reactants information, reaction variables (e.g. temperature, volume, and reaction scale). In many cases, the information was contained in the comment section of the .xml file rather than the appropriate data field. In these cases, the information was transferred (either through scripting or manual curation) to the correct field based on the preparation section which is shown as text form in the original dataset. Approximately half of the dataset consisted of reactions without yield information, denoted as 'None'. These non-valued reactions include incomplete reaction, reaction with no product, reaction with product but in very small quantity. These cases reported were classified to four types of non-yield reactions: (A) no reaction occurred (104 of 173), (B) trace amount of product (41 of 173) and (C) complex mixture of reaction products (28 of 173). This was done using a word filter and some manual filtering work of the reaction description in the comment field of the ELN entries. Firstly, incomplete reactions were filtered out which was the incomplete reaction abandoned for several reasons, such as being aimed at testing, optimization or having incorrect reactants. For complete reactions with missing yield values, zero was set for those with a description indicating that the desired product was not present, these reaction was classified as a 'no-yield' reaction. For reactions that reported the presence of a desired product but too little to produce a measurable result, the yield was also set to zero, as the lowest reaction yield reported in the dataset was 0.5%.
Finally, MDL molfiles were generated for each molecule from the compounds database included in the ELN, which were then used to generate SMILES strings. The SMILES files were converted into Cartesian coordinates for the Gaussian calculations using RDKit(1) and OpenBabel.(2)
Analysis of the ELN dataset identified 35 cases with the same reaction components and conditions. In some cases, the reported yields were significantly different as shown in Fig. 1. These cases were manually curated based on the comment section of the ELN dataset. For example, reaction 8 contained a comment “No sign of desired product. Now know that starting material needs to be columned. Reaction not progressed” and was therefore removed from the dataset.

Figure S1: Yield distribution of 35 reactions in ELN dataset with identical reaction components and conditions



The dependence of the fit from the experimental variability was tested by plotting the reported yield against the reported yield with random noise of ±15% and ±30% added, which leads to R^2 values of 0.91 and 0.73, respectively.

Figure S2: Correlation plots of actual yields vs. actual yields ±15% (left) and ±30% (right)



Feature generation and selection. For each molecule, two sets of chemical features were obtained. The first source is the full set of descriptors available in the RDKit library. The second source are the features from DFT calculations of each of the reactants using Gaussian16(3) with the B3LYP functional and 6-31G* basis set for geometry optimization and 6-311G* basis sets for single point calculations. The remaining features include the surface area generate from pymol, pKa of the base, solvent dielectric constant from the compound database. The following set shows the chemical features used for model training:
Molecular features: molecular volume, surface area, ovality, molecular weight, HOMO/LUMO Energy, electronegativity, hardness, and dipole moment.
Atomic features: Electrostatic charge and NMR shift
Reaction features: Temperature, Reaction scale and volume for some of reactions

To pre-select features, the features from above sources were combined, and a random forest model was trained on ten 70:30 random splits. Then, all features with feature importance of 10^{-4} or greater in any of the 30 random forest models were retained and included in the AGNN. Note that no feature engineering on the structural features is performed, the structural features are automatically generated by the GNN model. The random forest-based pre-selection helps reduce the number of the parameters used in the deep learning YieldGNN model.

Test of standard ML architectures We tested the several types of widely used ML models covering Linear, SVM, KNN and NN models, which were also evaluated in prediction of Buchwald-Hartwig HTE dataset by Ahneman et al. The models were performed on the processed dataset with all pre-selected features with or without RDKit features using 70% of the data as training set to predict 30% of the data as test set in 30 random splits. The Python code could be found in the scrips folder which can be used once the data file was created from the first step of data loading. The parameters of all the ML models were selected by using Grid Search on 5 cross-validations.

Table S1: Additional ML architectures tested

	Suzuki-Miyara [HTE]	Buchwald-Hartwig [HTE]	Buchwald-Hartwig [ELN]
Lasso (w/o rdkit feat.)	0.455±0.013 (0.177±0.002)	0.615±0.012 (0.135±0.002)	0.061±0.037 (0.244±0.010)
Lasso_CV (with rdkit feat.)	0.512±0.012 (0.164±0.001)	0.699±0.011 (0.120±0.0017)	--- ---
Lasso_CV (w/o rdkit feat.)	0.455±0.013 (0.177±0.002)	0.615±0.012 (0.135±0.002)	0.051±0.031 (0.246±0.009)
SVM (w/o rdkit feat.)	0.651±0.012 (0.129±0.002)	0.761±0.009 (0.102±0.002)	0.165±0.052 (0.217±0.009)
NN (with rdkit feat.)	0.735±0.026 (0.111±0.007)	0.840±0.018 (0.082±0.005)	--- ---
NN (w/o rdkit feat.)	0.631±0.0345 (0.135±0.007)	0.769±0.018 (0.096±0.004)	-1.575±2.866 (0.246±0.022)
KNN (w/o rdkit feat.)	0.776±0.011 (0.090±0.002)	0.5176±0.017 (0.1486±0.003)	0.041±0.038 (0.245±0.009)
Shuffle yield test w/o rdkit feat.	-0.127±0.018 (0.262±0.004)	-0.0656±0.017 (0.240±0.005)	-0.132±0.050 (0.247±0.011)

Figure S3: Correlation plots of predicted vs. actual yields for KNN, Lasso, NN and SVM models with RDKit features

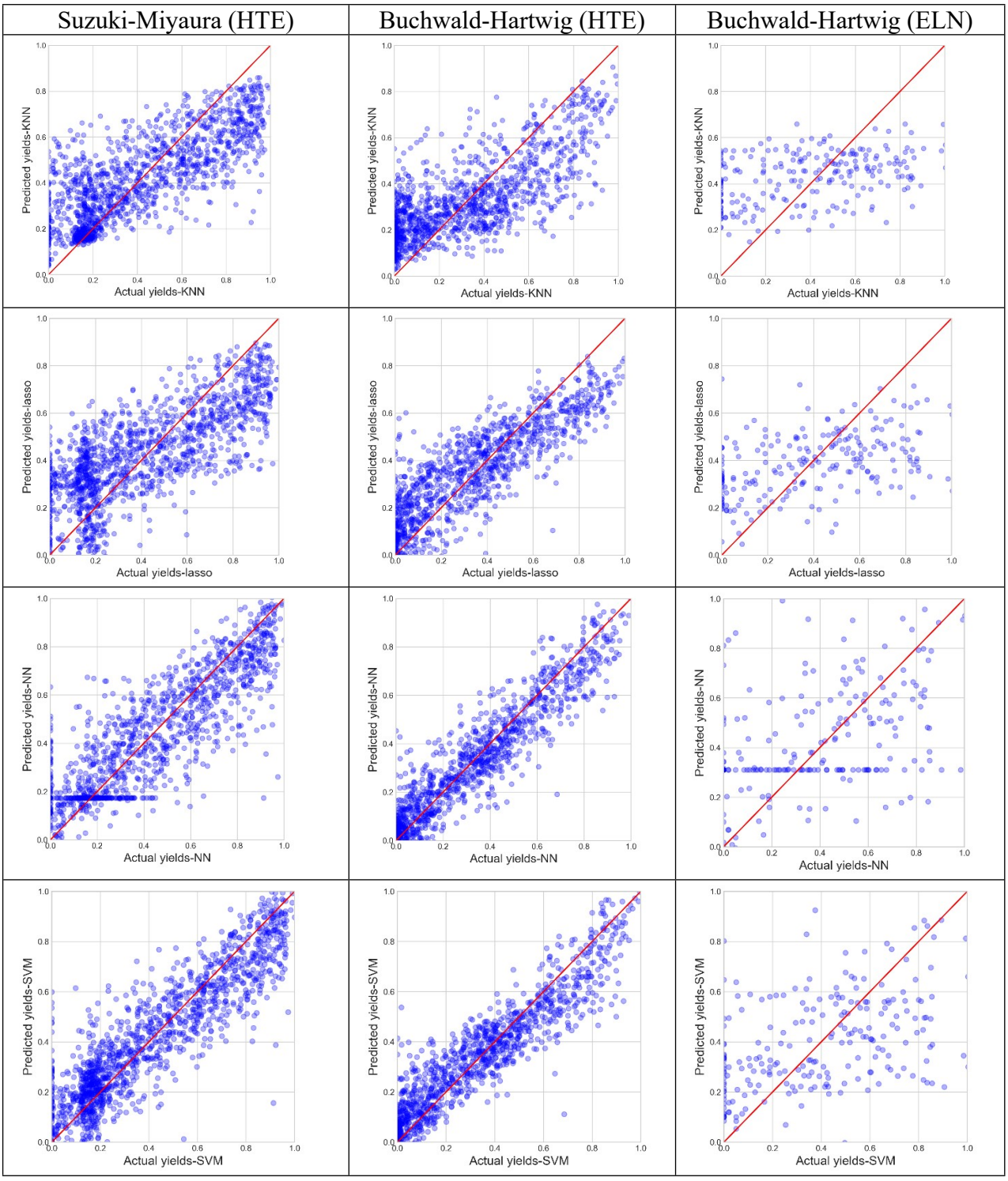
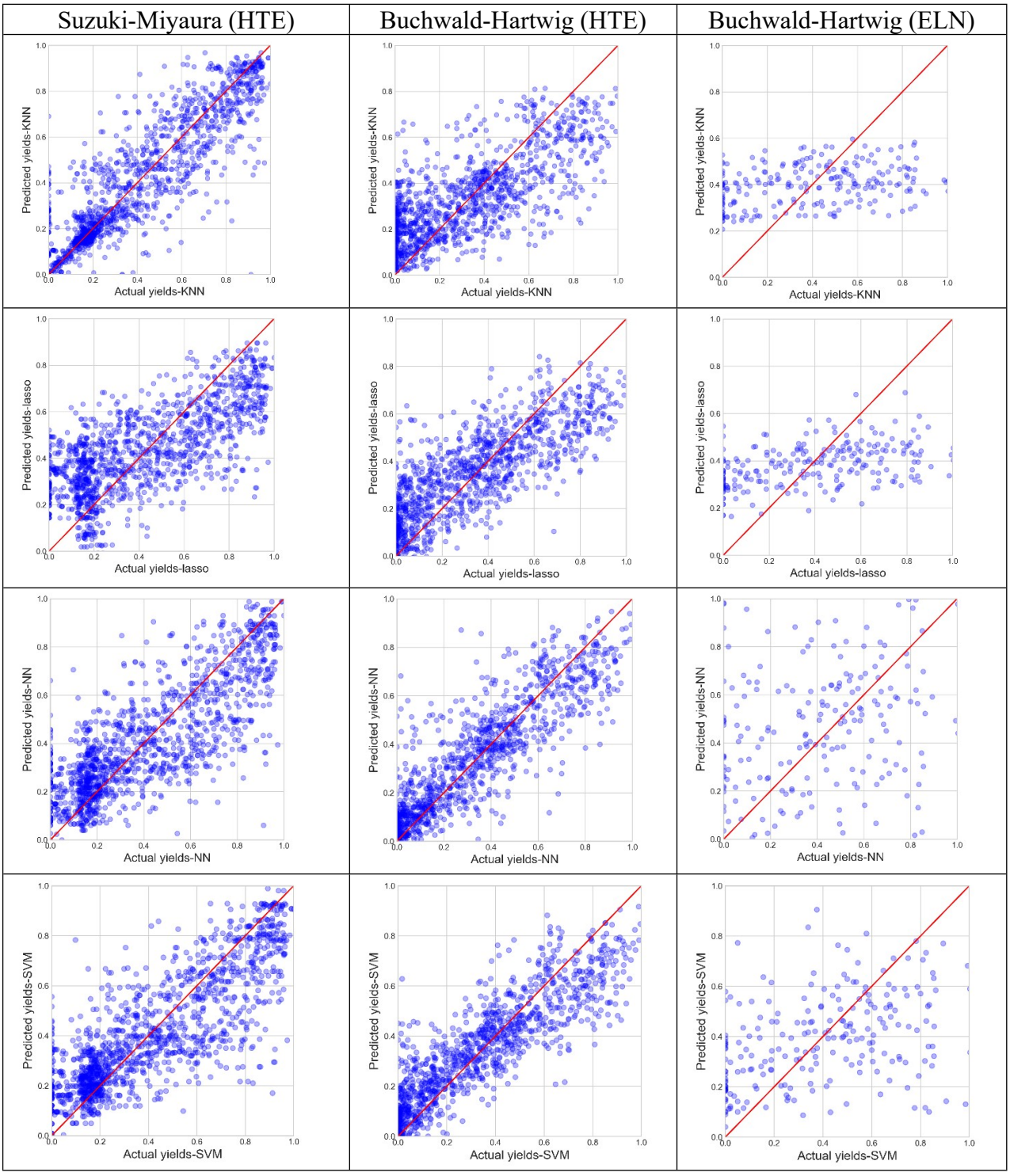


Figure S4: Correlation plots of predicted vs. actual yields for KNN, Lasso, NN and SVM models without RDKit features



YieldGNN Model architecture. The model integrates both the chemical features and the structural features for reaction molecules using two main components. An overview of the model is shown in Figure 3 of the main manuscript. The top component represents the AGNN which learns the structural features and the bottom module captures the chemical features. In this section the process of structural feature generation is detailed.

For each reaction, attributed graphs containing atom and bond features for each molecule are build first. Each atom feature contains atomic number, formal charge, degree of connectivity, explicit and implicit valence, and aromaticity. The bond features include the bond type, bond order and ring status. Atom features around the atom neighborhood are aggregated in an iterative manner using the Weisfeiler-Lehman Network (WLN)(4) to obtain the local atom and bond features. WLN is a graph kernel based on the Weisfeiler-Lehman test for graph isomorphism. Two graphs are isomorphic if they are topologically equivalent, and the WL test is a necessary condition for graph isomorphism. Thus, the WLN is one of the most expressive GNN methods and is used here. In each iteration, the atom feature representation is updated according to:

$$h_v^l = Relu\,(U_1 h_v^{l-1} + U_2 \sum_{u \in N(v)} Relu\,(V_1 h_v^{l-1} + V_2 h_{uv}))$$

where h_v^l is the atom feature representation at iteration l $(1 \le l \le L)$. U_1, U_2,V_1,V_2 are paramters to be learned, which are shared across L iterations. The final atom feature representation for atom v is obtained at the end of the final iteration using:

$$h_v = \sum_{u \in N(v)} (\theta_1 h_u^L \odot \theta_2 h_{uv}^L \odot \theta_1 h_v^L)$$

where $\odot$ is the convolution operation and θ are the model weights. For the HTE datasets, two iterations are used to capture the 2-hop neighborhoods. Therefore, for these datasets the above operation translates to two iterations to obtain the local representation of atoms.

Next, the local structural features are fed to an attention layer to capture the global structural features. The intuition behind including attention(5) is that different components of the reaction may influence the reaction yield differently. The attention layer is meant to capture the degree to which different atoms influence each other. The global representation of atom v is equivalent to the weighted sum of all other atoms in the reaction:

$$\hbar_v = \sum_z \alpha_{vz} h_v$$

The attention score for a given atom pair (v,z) is calculated using:

$$\alpha_{vz} = \sigma(u^T \times Relu\,(W_1 h_v + W_2 h_z + W_3 b_{vz}))$$

where $\sigma(.)$ is the sigmoid function, b_{vz} is the binary features for atom pair (v,z) and W is the attention weights to be learned by the model. Both global and local structural features are concatenated to generate the final structural features. The YieldGNN model provides two yield scores, one from the structural features (Yield(graph)) and the other from the chemical features (Yield(chem)). The two scores are fed to a linear layer to generate the final reaction yield predictions in analogy to earlier work by Coley et al.,(6) but for prediction of the reaction yield through combining both structural graph-based features as well as chemical properties.

Parameter Selection. A grid-search for each hyper parameter is performed and tuned for each dataset separately. For all datasets, batch size, dropout ratio, and initial learning rate are set to 40, 0.04, 0.01 and 0.005 respectively. A learning rate decay ratio of 0.5 is used on all datasets if the loss plateaus. A 2-hop neighborhood is used for the HTE datasets and a 3-hop neighborhood for the ELN data. The size of all hidden layers is set to 100 for the ELN data and 200 for the HTE data. The gradient is clipped with a 0.8 ratio on all datasets to avoid the exploding gradient problem. The model is trained for 200 epochs for HTE data and 100 epoch for the ELN data using Adam optimizers(7) with β_1=0.9 and β_2=0.99.

For pre-training, the models developed by Hu et al.(8) we use Graph Isomorphism Network (GIN), which based on the author's finding resulted in the best performance. Following the best parameters suggested by

the authors in their original work, we set the batch size, number of layers, and dropout ratio to 256, 5, and 0.15, respectively. We use an embedding dimension of 300 and a learning rate of 0.001, and pre-train the models for 100 epochs. For the fine-tuning module, we set batch size, number of layers, embedding dimensions, and dropout ratio to 32, 5, 0.5, and 300, respectively based on the author's recommended parameters. We set the learning rate to 0.001 and decay it with a 0.9 rate upon loss plateau. We use mean pooling for GNN during both pre-training and fine-tuning. We fine-tune the pre-trained model for 100 epochs as well.

Model Evaluation

Evaluation metrics. The model is compared to a series of standard ML architectures (Tables 1 and S1-S3). The performance of each model using coefficient of determination, denoted as R^2, and the mean absolute error, denoted as MAE. 30 models with different random splits of each dataset are run and the mean and standard deviation of the 30 experiments is reported. As an alternate metric, we created three pairs of data consisting of both training and test data, along with their corresponding predicted values, and merged training and test data with its predicted value. We then performed t-tests for each pair of samples using the 'ttest_ind()' function in scipy.stats library's, which calculate the t-statistic and p-value. Finally, we calculated the mean value for the 30 random splits. Note that this produces the p-value for the model hypothesis rather than the null hypothesis, which is more commonly reported.

Table S2: Results for three reaction datasets. For each dataset, the p-value for the model hypothesis for the test set and total dataset (in parenthesis) is shown

Method	Suzuki-Miyaura [HTE][18]	Buchwald-Hartwig [HTE][10]	Buchwald-Hartwig [ELN]
RF[a]	0.784 (0.875)	0.826 (0.891)	0.430 (0.634)
RF[b]	0.746 (0.837)	0.827 (0.891)	0.434 (0.624)
BERT[22]	0.729 (0.646)	0.656 (0.735)	0.307 (0.259)
Lasso[a]	0.631 (0.784)	0.724 (0.840)	0.408 (0.586)
SVM[a]	0.762 (0.759)	0.767 (0.813)	0.418 (0.525)
KNN[a]	0.415 (0.294)	0.685 (0.623)	0.202 (0.131)
One-hot Encoding	0.774 (0.863)	0.782 (0.849)	0.390 (0.393)
Shuffle[a]	0.480 (0.664)	0.480 (0.663)	0.376 (0.582)

a: with rdkit features; b: without rdkit features

Table S3: Results for three reaction datasets. Results for three reaction datasets. For each dataset, the p-value for the model hypothesis for the test set and total dataset (in parenthesis) is shown

Method	Suzuki-Miyaura [HTE][18]	Buchwald-Hartwig [HTE][10]	Buchwald-Hartwig [ELN]
YieldGNN[a]	0.880 (0.931)	0.802 (0.893)	0.473 (0.663)
YieldGNN[b]	0.778 (0.878)	0.867 (0.925)	0.217 (0.594)
YieldGNN[c]	0.820 (0.875)	0.766 (0.775)	0.335 (0.389)

a: with RDKit features; b: without RDKit features; c: without chemical features

Figure S5: Correlation plots of predicted vs. actual yields for training (A) and test (B) sets. (C): Learning plots of R^2 vs. epoch of YieldGNN. (D): Weights of the two components of the YieldGNN as function of epoch

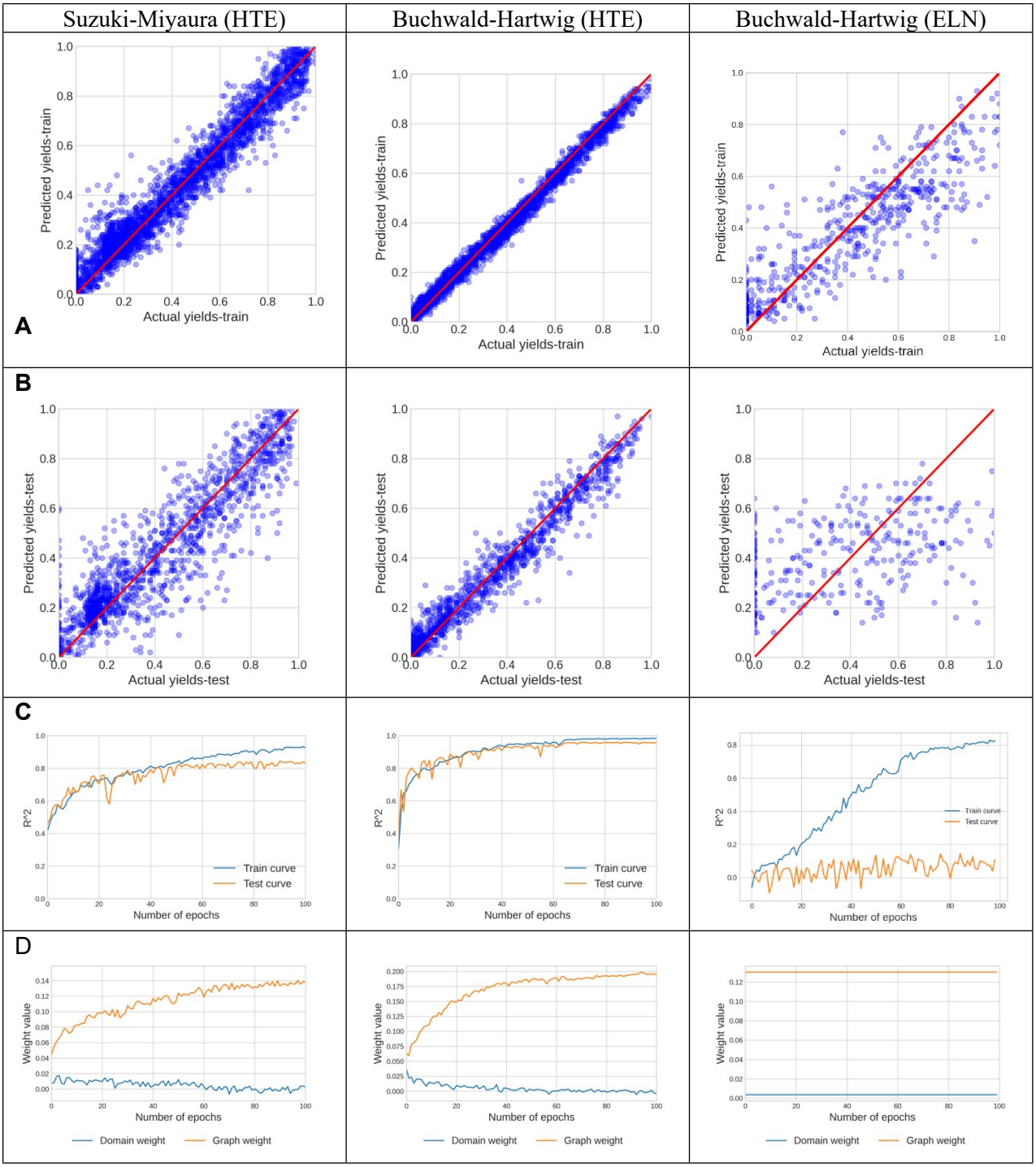


Table S4. Learned weights for the chemical properties in YieldGNN model of Suzuki-Miyaura HTE dataset with RDKit features

Rank	*Feature name*	*Weight*
1	ligand_SlogP_VSA4	-0.542
2	ligand_dipole_moment	0.507
3	halide_EState_VSA8	0.469
4	ligand_fr_NH0	0.404
5	halide_Kappa1	-0.374
6	solvent_Kappa1	-0.322
7	ligand_FpDensityMorgan3	-0.295
8	ligand_Kappa2	0.275
9	ligand_SlogP_VSA7	-0.271
10	halide_Chi1n	0.268
11	boronic Acid_PEOE_VSA4	-0.257
12	ligand_SlogP_VSA6	0.247
13	halide_PEOE_VSA6	0.246
14	halide_EState_VSA1	-0.246
15	ligand_BCUT2D_LOGPHI	0.241
16	base_VSA_EState4	-0.24
17	ligand_BCUT2D_CHGHI	0.24
18	ligand_PEOE_VSA14	-0.235
19	ligand_FpDensityMorgan2	-0.235
20	ligand_MolWt	0.235
21	halide_BalabanJ	-0.234
22	boronic Acid_BCUT2D_MWLOW	-0.234
23	halide_SlogP_VSA7	-0.234
24	base_BCUT2D_MWHI	-0.233
25	ligand_fr_para_hydroxylation	0.232
26	ligand_.P1_electrostatic_charge	-0.227
27	halide_Chi4v	0.226
28	halide_NumAliphaticRings	0.224
29	base_SlogP_VSA10	0.221
30	boronic Acid_NumSaturatedRings	-0.219
31	ligand_NumValenceElectrons	0.217
32	base_fr_COO2	-0.217
33	halide_BCUT2D_LOGPLOW	-0.214
34	boronic Acid_EState_VSA8	-0.213
35	ligand_fr_methoxy	-0.213

Table S5. Learned weights for the chemical properties in YieldGNN model of Suzuki-Miyaura HTE dataset without RDKit features

Rank	*Feature name*	*Weight*
1	halide_.C4_electrostatic_charge	0.605
2	halide_V2_frequency	-0.575
3	halide_V1_frequency	0.461
4	solvent_1	-0.428
5	halide_V2_intensity	-0.355
6	halide_.C9_NMR_shift	0.351
7	boronic Acid_dipole_moment	0.305
8	boronic Acid_E_LOMO	-0.304
9	halide_V1_intensity	0.302
10	halide_.C5_electrostatic_charge	-0.295
11	halide_.C8_NMR_shift	-0.288
12	halide_molecular_volume	-0.285
13	halide_.C3_NMR_shift	-0.284
14	ligand_molecular_weight	0.263
15	halide_.C1_NMR_shift	-0.253
16	solvent_3	0.243
17	halide_.C3_electrostatic_charge	-0.237
18	ligand_E_HOMO	-0.225
19	ligand_hardness	0.223
20	halide_.C9_electrostatic_charge	0.22
21	halide_hardness	0.199
22	boronic Acid_.C16_electrostatic_charge	0.197
23	halide_molecular_weight	0.193
24	halide_ovality	0.191
25	halide_.C15_electrostatic_charge	-0.182
26	halide_V0_intensity	0.181
27	halide_.N3_electrostatic_charge	0.176
28	halide_.C8_electrostatic_charge	0.174
29	halide_.C10_NMR_shift	-0.173
30	boronic Acid_.C12_NMR_shift	0.168
31	halide_.C13_electrostatic_charge	0.162
32	boronic Acid_.C8_electrostatic_charge	0.152
33	boronic Acid_.C3_electrostatic_charge	0.148
34	halide_.C2_NMR_shift	-0.145
35	boronic Acid_.C4_electrostatic_charge	0.143

Table S6. Learned weights for the chemical properties in YieldGNN model of Buchwald-Hartwig HTE dataset with RDKit features

Rank	*Feature name*	*Weight*
1	ligand_.C10_electrostatic_charge	-0.47
2	halide_PEOE_VSA9	0.452
3	ligand_MaxPartialCharge	-0.451
4	ligand_MinAbsEStateIndex	-0.431
5	ligand_.C11_NMR_shift	0.427
6	halide_PEOE_VSA8	-0.419
7	halide_MinAbsPartialCharge	-0.374
8	ligand_.C10_NMR_shift	0.363
9	ligand_VSA_EState4	0.361
10	ligand_.C4_electrostatic_charge	0.356
11	ligand_.C8_electrostatic_charge	-0.353
12	halide_.C1_NMR_shift	0.351
13	ligand_.C12_electrostatic_charge	-0.344
14	Additive_Chi1n	0.323
15	Additive_VSA_EState5	-0.319
16	ligand_.C5_electrostatic_charge	-0.311
17	ligand_electronegativity	0.309
18	ligand_NumHeteroatoms	-0.297
19	ligand_.C7_electrostatic_charge	0.297
20	halide_VSA_EState3	-0.282
21	ligand_Chi1n	-0.282
22	ligand_.H9_electrostatic_charge	0.281
23	ligand_molecular_weight	-0.275
24	ligand_.C15_NMR_shift	-0.27
25	ligand_.C4_NMR_shift	0.269
26	product_MinAbsPartialCharge	0.268
27	ligand_VSA_EState6	-0.267
28	base_BalabanJ	0.266
29	Additive_.C4_NMR_shift	0.266
30	halide_HallKierAlpha	-0.262
31	Additive_PEOE_VSA11	0.261
32	ligand_.C14_NMR_shift	-0.256
33	ligand_.H11_NMR_shift	-0.254
34	halide_Chi3v	0.248
35	ligand_.C6_NMR_shift	0.244

Table S7. Learned weights for the chemical properties in YieldGNN model of Buchwald-Hartwig HTE dataset without RDKit features

Rank	*Feature name*	*Weight*
1	halide_.H2_electrostatic_charge	1.009
2	Additive_.C4_electrostatic_charge	0.738
3	halide_.C4_electrostatic_charge	0.659
4	halide_.H3_electrostatic_charge	0.523
5	halide_.H3_NMR_shift	-0.482
6	amine_.C7_electrostatic_charge	0.452
7	Additive_.N1_electrostatic_charge	-0.442
8	amine_dipole_moment	0.396
9	Additive_.C5_electrostatic_charge	0.39
10	amine_.C2_electrostatic_charge	0.386
11	amine_.C2_NMR_shift	0.381
12	halide_dipole_moment	-0.363
13	halide_.C1_electrostatic_charge	0.362
14	base_pka	0.357
15	amine_E_HOMO	-0.336
16	halide_.C3_electrostatic_charge	-0.332
17	Additive_molecular_weight	0.309
18	amine_ovality	-0.295
19	Additive_.C3_NMR_shift	-0.27
20	amine_.C4_NMR_shift	0.267
21	halide_molecular_volume	0.267
22	ligand_.C17_NMR_shift	-0.266
23	halide_surface_area	-0.266
24	halide_.H2_NMR_shift	0.254
25	amine_hardness	-0.247
26	ligand_.C6_electrostatic_charge	0.226
27	amine_.C7_NMR_shift	0.226
28	amine_electronegativity	0.225
29	halide_.C4_NMR_shift	0.225
30	amine_.N1_NMR_shift	-0.224
31	halide_.C2_electrostatic_charge	-0.224
32	amine_.C1_NMR_shift	-0.221
33	amine_surface_area	-0.214
34	ligand_molecular_weight	0.21
35	Additive_.O1_electrostatic_charge	0.201

Table S8. Feature importance for the chemical properties in RF model of Buchwald-Hartwig ELN dataset with and without RDKit features.

	No RDKit Features		With RDKit Features		
Rank	***Feature name***	***Weight***	***Rank***	***Feature name***	***Weight***
1	ligand_molecular_weight	0.048171	1	temperature	0.021893
2	temperature	0.043841	2	product_SlogP_VSA2	0.01427
3	halide_ovality	0.042808	3	product_VSA_EState2	0.010126
4	amine_ovality	0.039912	4	halide_MolLogP	0.009896
5	amine_E_LOMO	0.035032	5	halide_VSA_EState10	0.009477
6	halide_molecular_weight	0.03335	6	product_MinAbsPartialCharge	0.00876
7	halide_.C1_electrostatic_charge	0.031769	7	product_MinAbsEStateIndex	0.008755
8	halide_V1_frequency	0.030433	8	product_MolLogP	0.008598
9	halide_molecular_volume	0.028384	9	product_Chi3n	0.008443
10	halide_.C2_electrostatic_charge	0.028137	10	product_Chi4n	0.008369
11	amine_dipole_moment	0.027836	11	ligand_qed	0.008162
12	halide_V0_frequency	0.027743	12	product_Chi2n	0.007621
13	halide_dipole_moment	0.026655	13	product_VSA_EState3	0.007247
14	halide_V2_intensity	0.026558	14	product_Chi3v	0.007051
15	solvent_1	0.026309	15	product_MaxPartialCharge	0.007023
16	halide_V2_frequency	0.026306	16	product_Chi1v	0.006745
17	amine_.N1_electrostatic_charge	0.02544	17	amine_electronegativity	0.006711
18	amine_electronegativity	0.024788	18	halide_BCUT2D_LOGPLOW	0.006647
19	amine_.N1_NMR_shift	0.024255	19	product_FpDensityMorgan3	0.006561
20	halide_V0_intensity	0.023632	20	halide_V0_frequency	0.006513

Table S9. Results of "Leave on

	test_mae	test_rmse	test_scores	train_mae	train_rmse	tra
5-phenylisoxazole	0.139777	0.174087	0.397206	0.035099	0.054472	
ethyl-3-methylisoxazole-5-carboxylate	0.066617	0.086625	0.917887	0.042550	0.063432	
ethyl-5-methylisoxazole-3-carboxylate	0.076732	0.102210	0.862042	0.019464	0.025639	
4-phenylisoxazole	0.068234	0.093198	0.868071	0.023371	0.031031	
3-phenylisoxazole	0.135447	0.214485	0.463342	0.018641	0.024367	
3-methylisoxazole	0.102956	0.140315	-0.344360	0.018945	0.024534	
5-methylisoxazole	0.121247	0.147447	0.239393	0.031357	0.041469	
benzo[c]isoxazole	0.179396	0.230424	0.085125	0.022057	0.029362	
3,5-dimethylisoxazole	0.169722	0.230911	0.093221	0.025011	0.033070	
methyl-isoxazole-5-carboxylate	0.142576	0.172699	0.617068	0.020486	0.026596	
ethyl-isoxazole-3-carboxylate	0.189750	0.251831	-1.036693	0.019648	0.025541	
ethyl-5-methylisoxazole-4-carboxylate	0.071385	0.103543	0.882123	0.022456	0.029244	
ethyl-isoxazole-4-carboxylate	0.123400	0.167765	0.718299	0.020760	0.027389	
benzo[d]isoxazole	0.152192	0.213862	0.271748	0.027755	0.037135	
ethyl-3-methoxyisoxazole-5-carboxylate	0.092118	0.118181	0.801740	0.024211	0.031335	
3-methyl-5-phenylisoxazole	0.047480	0.063187	0.938902	0.026862	0.035255	
N,N-dibenzylisoxazol-3-amine	0.048559	0.062316	0.946653	0.024088	0.031242	
methyl-5-furan-2-ylisoxazole-3-carboxylate	0.121119	0.163422	0.507613	0.019931	0.026227	
5-2,6-difluorophenylisoxazole	0.077902	0.092831	0.893097	0.023798	0.030676	
N,N-dibenzylisoxazol-5-amine	0.056776	0.073315	0.913446	0.022344	0.029281	
5-methyl-3-1H-pyrrol-1-ylisoxazole	0.092155	0.108650	0.721471	0.024368	0.032373	
methyl-5-thiophen-2-ylisoxazole-3-carboxylate	0.093664	0.123694	0.751298	0.022680	0.029766	
Average:	0.107483	0.140862	0.524045	0.024319	0.032321	

Figure S6: Shared atom features and atom numbering for each reaction component

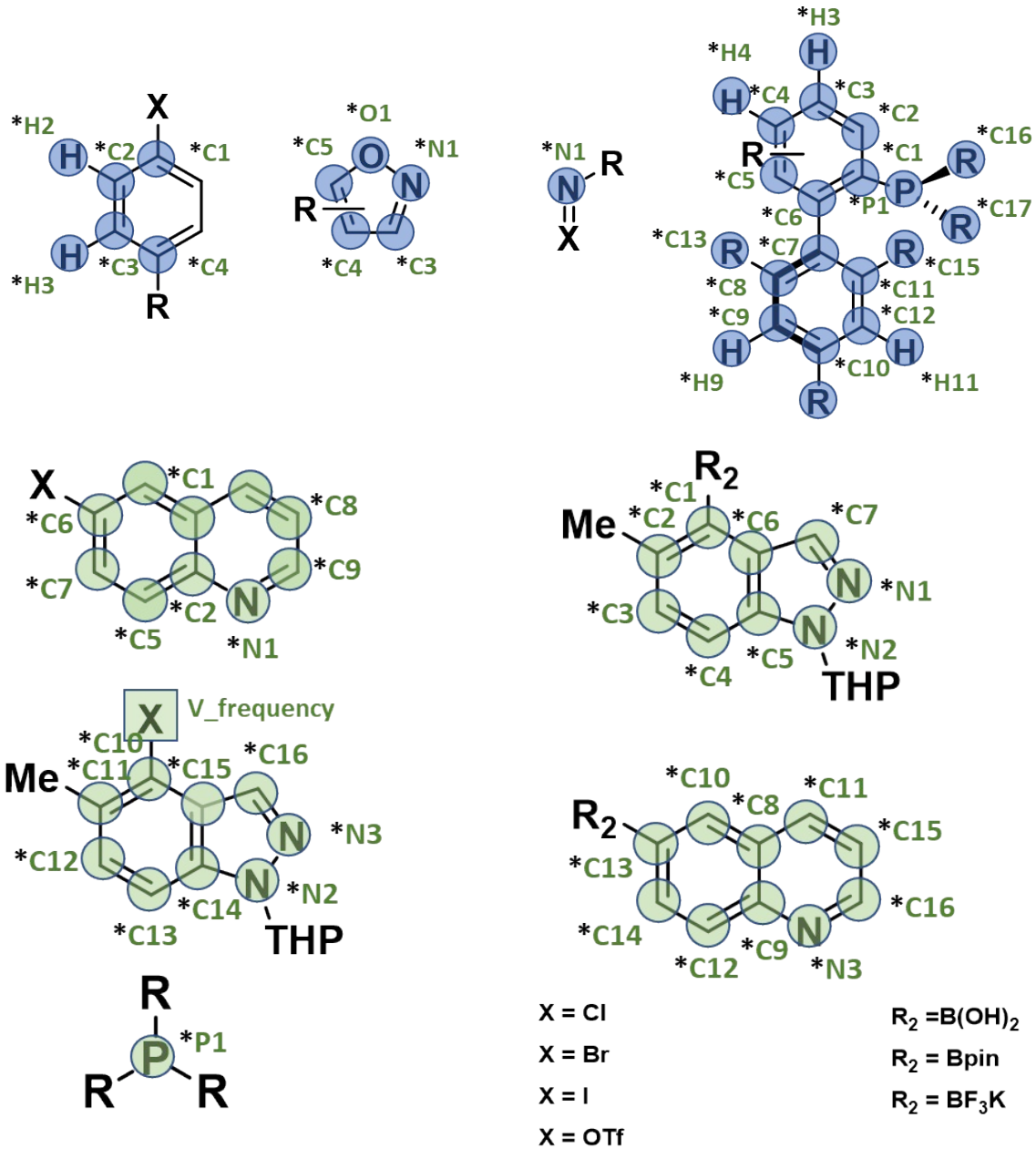


Table S10: Table of chemical properties in the prediction model

	Function	***Basis set***	***Method***	***Software***
charges	HF	6-31G*, 6-311G*	Pop = chelpg	G16(3)
NMR shift	B3LYP	6-31G*, 6-311G*	GIAO	G16
volume	B3LYP	6-31G*, 6-311G*	Opt	G16
E(HOMO)	B3LYP	6-31G*, 6-311G*	SCF	G16
E(LUMO)	B3LYP	6-31G*, 6-311G*	SCF	G16
vibration	B3LYP	6-31G*, 6-311G*	Freq	G16
dipole_moment	B3LYP	6-31G*, 6-311G*	Pop = full	G16
surface area	---	---	---	Pymol(9)
RDKit Features	https://www.rdkit.org/docs/cppapi/namespaceRDKit_1_1Descriptors.html			

The atom properties used as domain features in the model and the level of theory at which they were generated are listed in Table S8. A list of features generated in RDKit can be obtained from the website listed. These atom features were embedded following the atom numbering scheme shown in Fig S6. The features are used in the model to find the effect that may be attributed to certain types of atoms on the reaction yield. The feature weights of domain features for HTE datasets are shown in Tables S2-S5, which are generated from the trained linear feature selection model. A positive value means the positive influence on the reaction yield value, while the negative value means the negative influence on the reaction yield value. The details for the molecule properties are included in the feature tables as shown in Table S8.

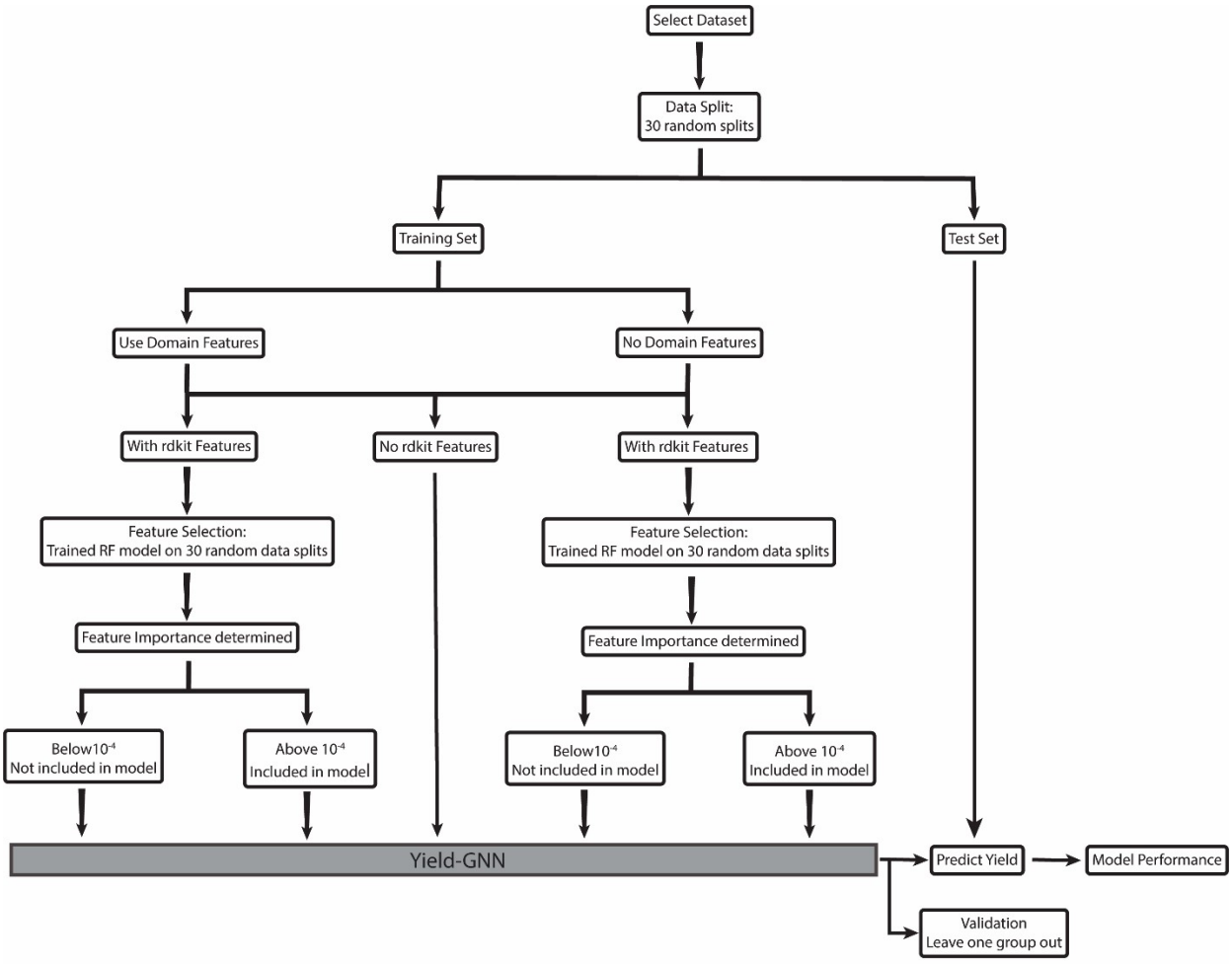
The feature importance for the ELN dataset in the Random Forest model is shown in Table S6. Information that was not found to have a significant feature weight in this analysis (e.g. reaction volume, base amount, reactant amount) was removed from the feature vector. It should be noted that the data variance on reaction scale, temperature, reaction volume, reactants amount can make the prediction challenging.

Figure S7: Highlighted atom weights in molecular graph neural network in two examples

correct prediction **incorrect prediction**

Molecular maps with highlighted atoms for two prediction examples that were generated from the Graph Neural Network (Figure S7). The interactions between the labeled parts were considered as important influences on reaction yield. In the model shown on the right, the prediction was able to correctly capture the interaction between the phosphorus atom and carbon atom in the ligand, while on the right the model did not identify the influence of the atom interactions in the ligand, leading to an incorrect prediction.

Figure S8: Yield-GNN flow chart showing model architecture



The Yield-GNN model was employed to predict the reaction performance for the three reaction datasets included in the data directory (./su, ./dy, ./az) of the model. The model includes an embedded molecular Graph Neural Network and a linear model dealing with chemical properties (Figure 3 in the main text). The predictions were performed using the Yield-GNN model with 30 randomized splits into test (30%) and training set (70%) with a learning rate of 0.005 and a dropout rate of 0.2. The model can generate RDKit features that are useful for reaction prediction through feature selection. In the feature selection process, all the features generated from the RDKit library will go through a random forest model to collect the names of the higher-ranked features into a file, and then use the file to generate the features in reaction prediction. Finally, the leave-one-group-out (LOGO) validation was performed on the types of additives used in Buchwald-Hartwig reaction dataset (see Table S7).

Table S11 Table of Contents for the GitHub repository.

Name	*Description*
README.md	General instruction for downloading the data, installing the necessary packages, and running the code.
01_prepare_data.py	Python script for converting the .json format data into clean .csv format, and splitting it into 10 sets of train/test data.
02_train_rf.py	Python script for training the random forest models and feature selection.
03_train_yield.py	Python script for training the YieldGNN model.
rxntorch	Python module including all necessary data containers and models for running 03_train_yield.py.
scripts	Contains a set of Python scripts for auxiliary functions that are used for plotting the results or loading the model for analysis.
jupyter_notebooks	Contains Jupyter notebooks for loading and analyzing the model.
RawDataPreparation	This directory is under jupyter_notebooks, and contains the notebooks used for processing the ELN data and converting it into the .json format.

Model operation guide

Setting up of the environment

The Yield-GNN model is based on chemical toolkit RDKit and machine learning toolkits PyTorch and Scikit-learn. These toolkits are required and can be downloaded using the Anaconda package manager. The version should be the same as the version shown in the READE.md file in the Github repository.

Preparation of dataset

The input of the three datasets (su, dy, az) here have been provided and can be downloaded from the Github repository. The prediction for other data sets needs to generate data files in JSON format, including reactant SMILES information and feature vectors containing the chemical information described in Table S8. The model will use the reaction SMILES information to generate the molecular graph. The properties in the feature vector will undergo feature selection and are forwarded to the linear part of the YieldGNN model.

Loading of dataset

The three datasets (.su,.dy,.az) are provided on Github and the JSON files are loaded using different arguments for the data loading script. Then, the JSON file is curated and processed and generates a .csv file that contains all possible features (depending on whether you use RDKit features or not). Examples of the formats are provided in the Github directory. It also generates 30 sets of train-test indexes in the .pickle file which are then used to split the data later for training and testing.

Feature selection

If RDKit features are considered in the Yield-GNN model, a feature selection step is required. The script will first read the .csv file and the .pickle file to load and split the data and read all possible features. A Random Forest model is trained on each data split. The script will then output the R2 value calculated for each model and the selected feature, and select features based on the feature importance. The selected features are written to the selected_feats.txt file.

Running the YieldGNN model

To run the YieldGNN model, the script will first load the .pickle file, the selected_feats.txt file (depending on whether or not using the RDKit features), and .csv file to split the dataset to train and test, read the names of feature selected and the values for the selected features. It provides an output for output R2, MAE, RMSE for the selected model for a given random split. The model parameter can be adjusted by using different arguments as provided in the code.

After the prediction is finished, the performance of the prediction model can be further analyzed by using the script in ./scripts and the ./jupyter_notebooks. These analysis tools include loading previous models, generating true and predicted values, model performance vs epochs, and feature weights for chemical properties.

Model evaluation metrics

All metrics used to calculate model performance are calculated using the sklearn python library. package. To evaluate the model performance, R^2, RMSE, and MAE values are calculated between predicted result and true result for each of epoch of the prediction model using:

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=0}^{n_{samples}-1} (yi - \hat{y}i)^2}{\sum_{i=0}^{n_{samples}-1} (yi - \bar{y}i)^2}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \left|y_i - \bar{y}\right|$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (yi - \hat{y})^2}$$

References

1. Landrum G. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. RDKit.org2013.
2. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. Open Babel: An open chemical toolbox. J Cheminf. 2011;3:1-14.
3. Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, Scalmani G, Barone V, Petersson GA, Nakatsuji H, Li X, Caricato M, Marenich AV, Bloino J, Janesko BG, Gomperts R, Mennucci B, Hratchian HP, Ortiz JV, Izmaylov AF, Sonnenberg JL, Williams, Ding F, Lipparini F, Egidi F, Goings J, Peng B, Petrone A, Henderson T, Ranasinghe D, Zakrzewski VG, Gao J, Rega N, Zheng G, Liang W, Hada M, Ehara M, Toyota K, Fukuda R, Hasegawa J, Ishida M, Nakajima T, Honda Y, Kitao O, Nakai H, Vreven T, Throssell K, Montgomery Jr. JA, Peralta JE, Ogliaro F, Bearpark MJ, Heyd JJ, Brothers EN, Kudin KN, Staroverov VN, Keith TA, Kobayashi R, Normand J, Raghavachari K, Rendell AP, Burant JC, Iyengar SS, Tomasi J, Cossi M, Millam JM, Klene M, Adamo C, Cammi R, Ochterski JW, Martin RL, Morokuma K, Farkas O, Foresman JB, Fox DJ. Gaussian 16 Rev. B.01. Wallingford, CT2016.
4. Lei T, Jin W, Barzilay R, Jaakkola T, editors. Deriving neural architectures from sequence and graph kernels. International Conference on Machine Learning; 2017: PMLR.
5. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473. 2014.
6. Coley CW, Jin W, Rogers L, Jamison TF, Jaakkola TS, Green WH, Barzilay R, Jensen KF. A graph-convolutional neural network model for the prediction of chemical reactivity. Chem Sci. 2019;10:370-7.
7. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014.
8. Hu W, Liu B, Gomes J, Zitnik M, Liang P, Pande V, Leskovec J. Strategies for pre-training graph neural networks. arXiv preprint arXiv:190512265. 2019.

9. DeLano WL. PyMOL, Schrödinger, 2020.