Electronic Supplementary Information

A Transfer Learning Protocol for Chemical Catalysis using Recurrent Neural Network Adapted from Natural Language Processing

Sukriti Singh[a,*] and Raghavan B. Sunoj[a,b,*]

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

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

1. Overview of ULMFiT

ULMFiT consists of key three steps as shown in Fig. S1. Additional details of each of these three steps with the respective model architecture are provided in the following sections.

Fig. S1. The steps involved in ULMFiT.

1.1 Training the language model (LM)

1.1.1 Model architecture

A brief overview of the model architecture used in the training of the LM is given in Fig. S2. For training, a regular LSTM with inbuilt optimization and regularization capabilities such as in the AWD-LSTM model architecture, is implemented.[1] This architecture involves an embedding layer, an encoder with three LSTM layers within, and a decoder layer. The embedding layer transforms the numericalized tokens (as shown in Fig. 2 in the main text) into real-valued vectors of fixed length. With an embedding size of 400 (a standard size, used in ULMFiT), each character in the SMILES string is represented by a 400-dimensional vector space. These vectors are initialized in the embedding layer and get updated during the training of the network. The embedding vectors contain the semantic relationship among the characters of the SMILES string. The output from the embedding layer, with a size of 400, is received as the input in the first of the 3 LSTM layers of the encoder consisting of 1152 hidden activations in each layer. The second LSTM layer then takes the hidden state of the previous layer, with a size of 1152, as its input. The output of the last layer of the encoder provides a hidden state of size 400, same as the embedded input. The output hidden state of the final LSTM layer is then decoded by a fully connected linear layer. Finally, a softmax function is applied which assigns the probability for every token in the vocabulary to be the next token.

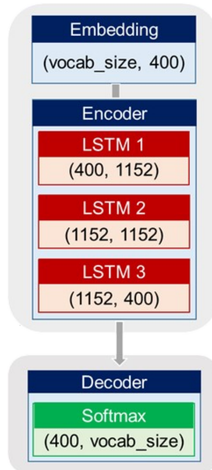


Fig. S2. The network architecture used for the training the language model using a 400-dimensional vector space generated from the SMILES strings of molecules.

1.1.2 General-domain LM pre-training: In the first step, the LM is pre-trained on a large dataset through which the model acquires the ability to predict the next character in a SMILES string. The pre-training step assists the model in understanding the inherent connectivity in molecules including grammar and semantics present in SMILES, which is beneficial for the downstream tasks.[2] Although this step is expensive, it needs to be done only once as it can be reused for other tasks. In the present study, the general-domain LM is trained on one million molecules as collected from the ChEMBL database, thus utilizing the vast amount of unlabeled chemical data for pre-training.[3] SMILES augmentation is also used while training wherein each molecule is augmented with four additional SMILES strings (Fig. S8).

1.1.3 Target-task LM fine-tuning: While the general-domain dataset used in the pre-training can span a large and diverse regions of the chemical space of interest, it is quite likely that the target task belongs to a different distribution. In keeping with the spirit of transfer learning, the knowledge gained from the pre-training step should be utilized for the target task. Thus, the target-task LM is fine-tuned using the pre-trained weights from the previous step (Fig. S3). Akin to that in the pre-training step, the model learns to predict the next character in a SMILES string, but at this stage, the model would have learned the task-specific features as well. With the pre-trained model, the fine-tuning step converges faster as it only has to adapt to the characteristics of the target task data. At this stage, the LM has learned the task-specific features and is ready for the regression task with some adjustments in the architecture (Section 1.2.1).

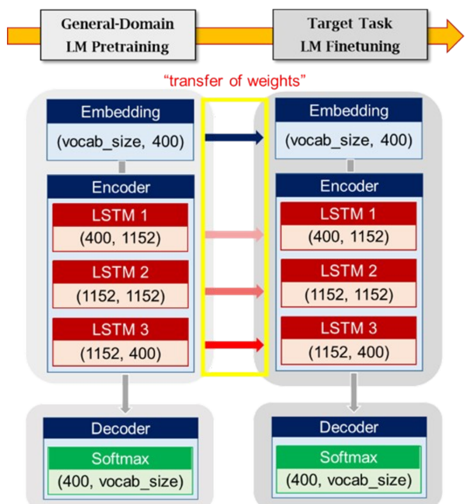


Fig. S3. Target-task LM fine-tuning that uses the pre-trained weights from the general-domain LM pre-training

1.2 Training the target-task regressor

1.2.1 Model architecture

For training a regressor, the LM architecture is slightly adjusted in the downstream (i.e., after the final LSTM layer) by adding two linear blocks with the ReLU activation function for the first linear layer, as shown in Fig. S4. Since the input sentences can contain several characters, the chance of losing some of the relevant information might be high if only the last hidden state is considered. To address this, we used the concat pooling technique by concatenating the last hidden state with both max-pooled and mean-pooled representations of all the hidden states (each of size 400) in the third LSTM layer. Such concatenation yields a feature vector of size 1200 for each character, which is then passed to a feed forward neural network serving as the linear decoder. Here, the first linear layer in the decoder has 50 activations, thereby reducing the size of the longer concatenated input feature vector to 50. It then serves as the input to the final linear layer where the dimension is further reduced to 1 for the regression task (as shown in Fig. 2 in the main text).

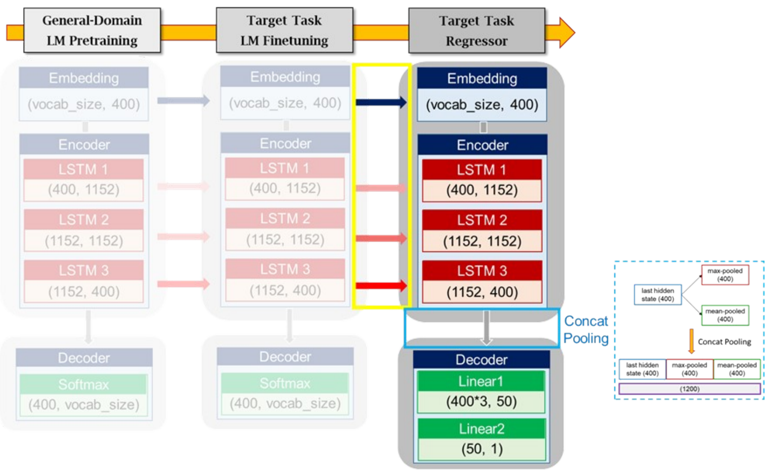


Fig. S4. Target-task regressor fine-tuning with concat pooling to make it conducive for the desired regression.

1.2.2 Target-task regressor fine-tuning

In the final stage, the actual target task activity, i.e., the prediction of % of *ee* or yield, is carried out. To utilize the knowledge gained through the pre-trained as well as the fine-tuned LMs, the embedding layer and the three LSTM layers are adopted as is, while the decoder and softmax layers are cut-off (depending on whether the regression task is desired to use the pre-trained or a fine-tuned weights). The two linear layers of the regressors in the decoder are then initialized by using randomly distributed weights and are trained from scratch. Fine-tuning the target regressor is crucial to transfer learning as an aggressive fine-tuning might even nullify the benefits of a trained LM. In addition to discriminative fine-tuning and fit-one-cycle methods, we have also used gradual unfreezing protocol for fine-tuning the regressor.

The fine-tuning of the target-task regressor is a crucial step in transfer learning. The first approach that we used in this study for fine-tuning involves the model initialization with the pre-trained (or fine-tuned) weights and training the full model at once. In other words, the method employing a fixed learning rate and without frozen weights constitute the first protocol. Other techniques like gradual unfreezing, discriminative learning rates etc., are the NLP-specific fine-tuning methods introduced with the ULMFiT (Supplementary section 2). In gradual unfreezing, we start with frozen weights first and the layers are unfrozen step-by-step during training and this process is repeated until the entire model is unfrozen and fine-tuned. The results presented in the manuscript are obtained by using the first protocol of fine-tuning. However, the performance comparison using both of these fine-tuning methods is also done (Supplementary sections 2 and 11).

A rigorous hyperparameter optimization is performed for fine-tuning the target-task regressor. The number of epochs and the learning rate are the hyperparameters, which are tuned on the validation set, in addition to the dropout rate. Also, the effect of number of augmented SMILES (termed as SMILES augmentation) and the gaussian noise added to the regression output is also considered for optimization (Supplementary sections 5.3, 5.8, 6.3, 6.6, 7.3, and 7.6).

2. Various techniques used for fine-tuning

2.1 Discriminative learning rates for pre-trained models

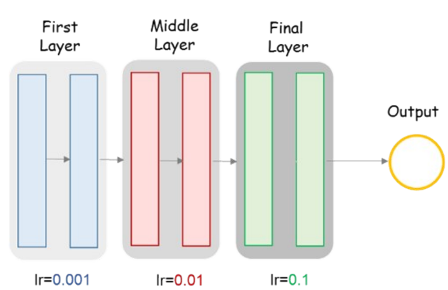


Fig. S5. Pictorial representation of a general case scenario using discriminative learning rates.

It has been known that different type of information is captured by different layers of the model, thus necessitating the use of the discriminative learning rate (lr). In a LM, the initial layers contain the general information of the language and would require minimum fine-tuning. The amount of fine-tuning required increases as one moves towards the final layer. Therefore, different learning rates can be used for each layer. The initial layers are trained with lower learning rate while a higher learning rate is used for the later layers (Fig. S5). In this way, the pre-trained weights do not get drastically altered and the layers near the output are trained relatively more aggressively.

2.2 Fit-one-cycle

In the fit-one-cycle method,[4] the learning rate (lr) is cycled between the minimum and maximum learning rates. For the duration of a training run, the lr goes from its minimum to maximum value and back again (one cycle). The higher lr during the middle of training acts as a regularization to prevent the model from over-fitting. A higher learning rate helps the network to get out of the saddle points. The lr and momentum goes in opposite directions, i.e., for a small lr, the momentum will be high and vice versa. This can help in accelerating the training.

2.3 Gradual unfreezing

Fine-tuning the target regressor is crucial to transfer learning, where gradual unfreezing is a useful technique (Fig. S6). The new linear layers that contain the least information compared to the other layers, are fine-tuned first while keeping the other layers frozen (i.e., weights are not updated). The layers are unfrozen step-by-step and this process is repeated until the entire model is unfrozen and fine-tuned.

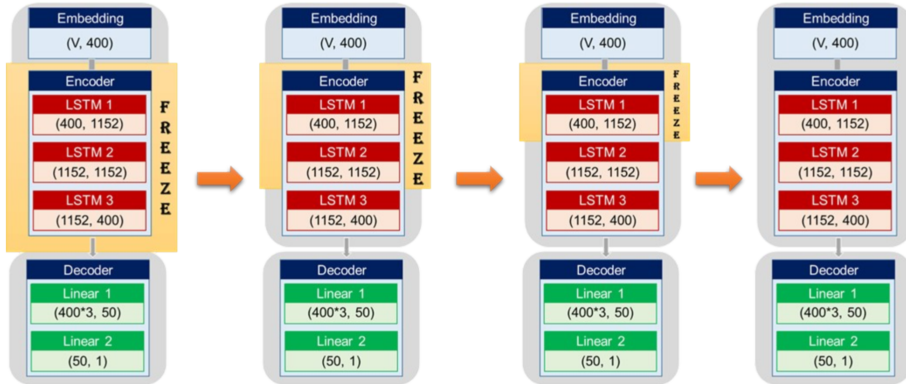


Fig. S6. Illustration of the gradual unfreezing approach used in the fine-tuning of the target-task regressor.

3. Dataset preparation

It should be noted that the reactions considered in this study, consist of multiple chemical entities such as a catalyst, substrates, additives/solvent etc., and that the reaction outcome depends on the nature of these participating species. The SMILES strings of the individual reaction partners are therefore merged together as shown in Fig. S7(a) for a representative reaction. The concatenated SMILES thus generated provides a composite representation for the desired reaction. To make these strings machine readable, the individual characters are generated through tokenization, as described in Fig. S7(b), wherein individual strings are split into tokens (e.g., 'C', 'o', '(', '=', 'p' *etc.*) separated by a dot (.). The list of unique possible tokens is called vocabulary, which is 13 for the example shown here. The total vocabulary size of 32 for reactions 1-2 and 40 for reaction-3 were required to represent all the samples in respective reaction class. These tokens are then numericalized to integers. Based on the location of a token, a unique id is assigned to each token. The encoded token is then matched to the embedding vector via one-hot encoding (Fig. S7(b)). The mapping of each of the tokens to their respective ids serves as an input for the deep learning model.

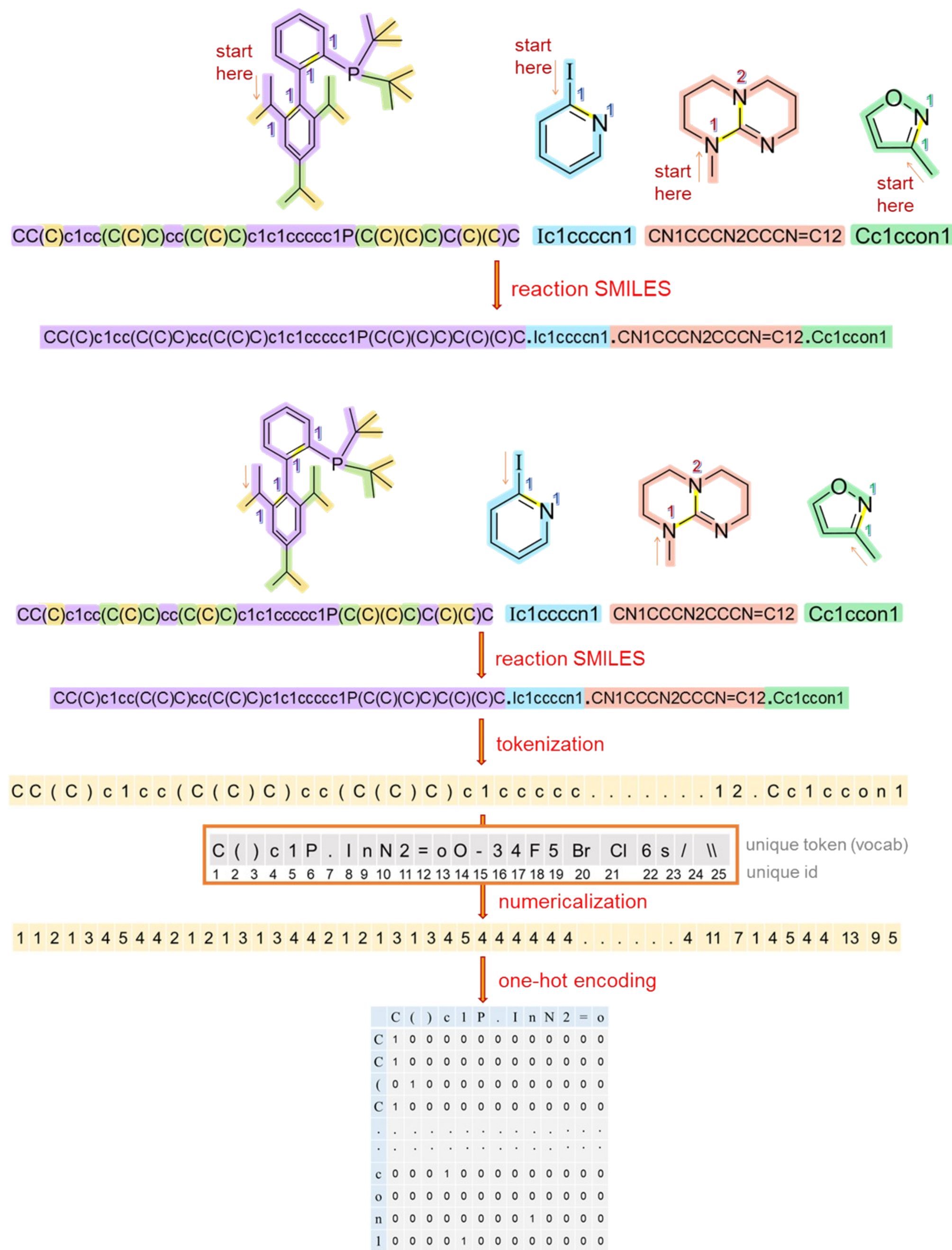


Fig. S7. (a) SMILES representation for a representative reaction obtained through concatenation of individual SMILES of the ligand, reactants, additive, and base. The arrow shown on each molecule indicates the starting atom considered in the generation of the SMILES representation, (b) various downstream conversions to machine-readable format to be used as input for the model.

3.1 SMILES augmentation

Since multiple unique SMILES can represent a given molecule (Fig. S8(a)), it also allows for desirable data augmentation, particularly for problems with relatively lower data size.[5] The one unique SMILES representation for a molecule, that satisfies certain set of rules[6], among all valid possibilities is known as the canonical SMILES. We have explored SMILES augmentation in this study. The data augmentation provides all valid SMILES with the key difference that the starting atom and direction of traversing the graph are chosen randomly. As shown in Fig. S8(a), one molecule is represented by five different SMILES representations. The process of generating these SMILES first involves the selection of the starting atom, as shown using the arrow on each molecule. Once the starting atom is selected, the direction of traversal of the 2D graph can be chosen. All these are chosen randomly and the augmented SMILES are therefore also known as randomized SMILES. For the regression task, a gaussian noise (with mean zero and standard deviation σ_{g_noise}) is added to the labels of the augmented SMILES during the training (Fig. S8(b)). The number of augmented SMILES and σ_{g_noise} is tuned on the validation set.

(a)

(b)

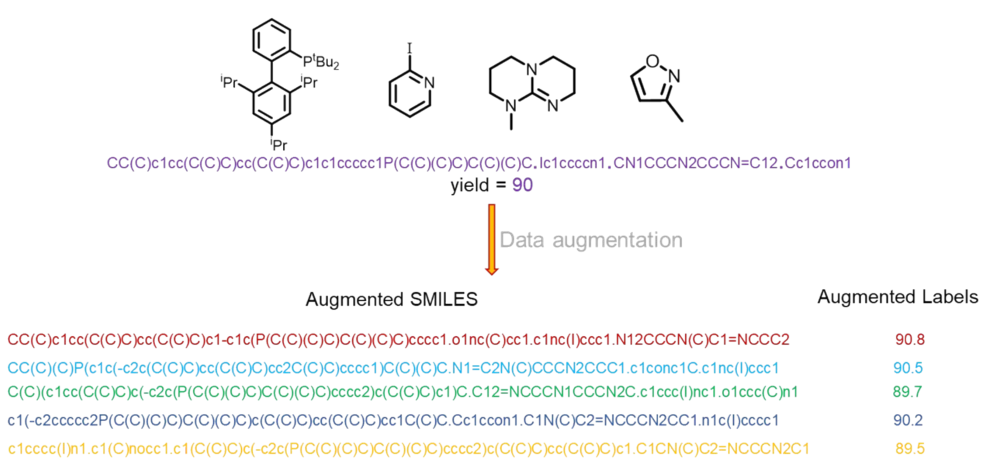


Fig. S8. (a) SMILES augmentation for a representative molecule. The arrow shown on each molecule indicates the starting atom considered in the generation of the SMILES representation, and (b) the data augmentation used for a reaction in the training.

3.2 Test time augmentation (TTA)

The test set performance is evaluated using the predictions based on the canonical SMILES as well as that employing test-time augmentation (TTA). In the former, each sample is represented by a unique SMILES whereas several augmented SMILES per sample is used in TTA and the average of the predicted values obtained from augmented SMILES is taken as the final prediction (Fig. S9).

Fig. S9. Illustration of the test time augmenation (TTA) procedure.

4. Programming details

The model is implemented using PyTorch[7] deep learning framework and fast.ai library[8]. All the calculations are run using the Google Colab Pro. It provides access to T4 and P100 GPUs with memory up to 25 GB. Code, data, and instructions will be made available at https://github.com/Sunojlab.

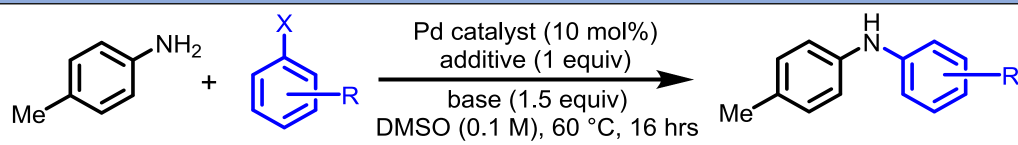
Reaction-1

5. Pd-catalyzed Buchwald-Hartwig reaction

5.1 Summary of reactions

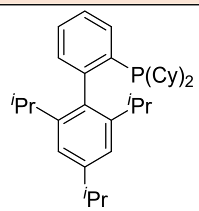
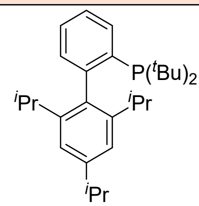
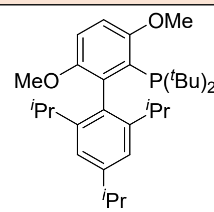
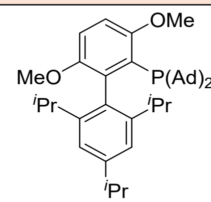
Table S1. Details of Reaction Components

General Reaction Conditions

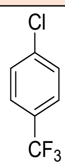
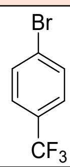
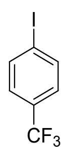
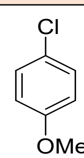
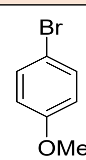
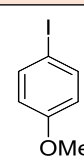
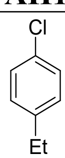
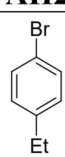
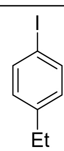
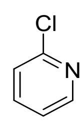
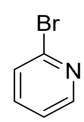
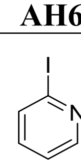


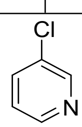
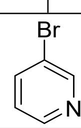
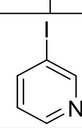
Reaction Components

Ligands

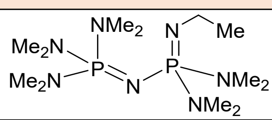
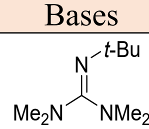
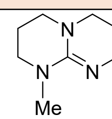
			 (Ad = adamantyl)
L1	**L2**	**L3**	**L4**

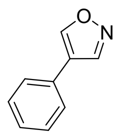
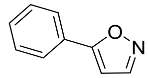
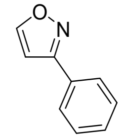
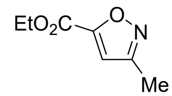
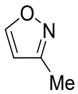
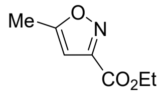
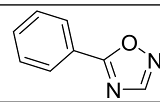
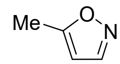
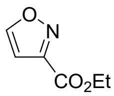
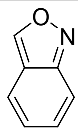
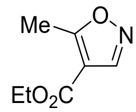
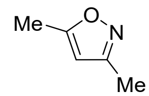
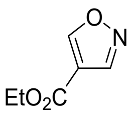
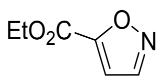
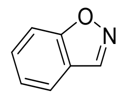
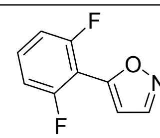
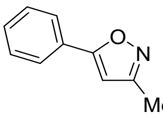
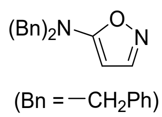
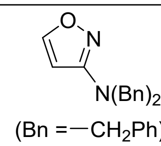
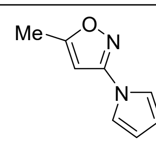
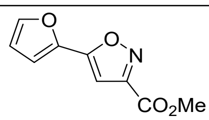
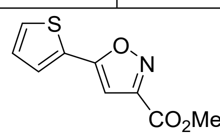
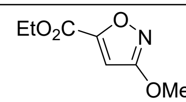
Aryl halides

					
AH1	**AH2**	**AH3**	**AH4**	**AH5**	**AH6**
					
AH7	**AH8**	**AH9**	**AH10**	**AH11**	**AH12**

		
AH13	**AH14**	**AH15**

Bases

		
B1	**B2**	**B3**

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

5.2 Target-task LM fine-tuning

The hyperparameter optimization is performed for fine-tuning the target-task LM. For this purpose, a randomized 80:20 train-test splits were used. The hyperparameters considered are listed in Table S2. In addition, effect of different number of augmented SMILES is also considered. The model is evaluated using accuracy as the error metric, as compiled in Table S2.

Table S2. Hyperparameter Optimization for the Target-task LM Fine-tuning

no. of augmented SMILES	dropout_rate	epoch[a]	learning rate[b]	train_loss	valid_loss	accuracy
varying the number of augmented SMILES						

0	0.0	[5,5]	[0.36, 0.01]	0.0700	0.0872	0.9601
25	0.0	[5,5]	[0.36, 0.01]	0.1942	0.1494	0.9496
50	0.0	[5,5]	[0.36, 0.01]	0.1965	0.1591	0.9486
varying the dropout rate						
0	0.0	[5,5]	[0.36, 0.01]	0.0700	0.0872	0.9601
0	0.1	[5,5]	[0.36, 0.01]	0.0868	0.0902	0.9599
0	0.2	[5,5]	[0.36, 0.01]	0.0846	0.0940	0.9598
0	0.3	[5,5]	[0.36, 0.01]	0.0934	0.0888	0.9601
0	0.4	[5,5]	[0.36, 0.01]	0.0839	0.0899	0.9599
0	0.5	[5,5]	[0.36, 0.01]	0.0839	0.0875	0.9610
0	0.6	[5,5]	[0.36, 0.01]	0.0833	0.0875	0.9608
0	0.7	[5,5]	[0.36, 0.01]	0.0807	0.0891	0.9606
0	0.8	[5,5]	[0.36, 0.01]	0.0791	0.0902	0.9600
0	0.9	[5,5]	[0.36, 0.01]	0.0820	0.0887	0.9599
0	1.0	[5,5]	[0.36, 0.01]	0.0799	0.0887	0.9606
varying the number of epochs						
0	0.5	[5,5]	[0.36, 0.01]	0.0839	0.0875	0.9610
0	0.5	[4,4]	[0.36, 0.01]	0.0874	0.0886	0.9606
0	0.5	[4,5]	[0.36, 0.01]	0.0898	0.0879	0.9605
0	0.5	[5,6]	[0.36, 0.01]	0.0804	0.0916	0.9604
0	0.5	[6,6]	[0.36, 0.01]	0.0839	0.0884	0.9603
0	0.5	[3,4]	[0.36, 0.01]	0.0903	0.0877	0.9603
varying the learning rate						
0	0.5	[5,5]	[0.36, 0.01]	0.0839	0.0875	0.9610
0	0.5	[5,5]	[1e-1,1e-2]	0.0838	0.0849	0.9607
0	0.5	[5,5]	[1e-1,1e-1]	0.3989	0.2013	0.9285
0	0.5	[5,5]	[1e-2,1e-2]	0.0763	0.0872	0.9605
0	0.5	[5,5]	[1e-2,1e-3]	0.0824	0.1060	0.9572
0	0.5	[5,5]	[1e-1,1e-3]	0.0809	0.0983	0.9590

[a]For the first step, the weights of the LSTM layers are kept frozen and the rest of the model is trained. In the second step, all layers are unfrozen so that the LSTM layers can be fine-tuned. [b]The notations such as [5,5] correspond to the number of epochs in each step and [0.36, 0.01] are the respective learning rates. One hyperparameter is varied at a time keeping others constant. The red color values and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

These optimal set of hyperparameters are considered for assessing the model performance on 10 independent runs on a set of randomly selected train-test splits. The model performance provided in Table S3 is reported in terms of the commonly recommended metrics such as accuracy and perplexity. An average accuracy of ~96% over 10 runs could be obtained.

Table S3. The Calculated Train and Test Accuracies for the Target-task LM Using the Optimal Set of Hyperparameters

sr. no. for runs	train_loss	test_loss	accuracy	perplexity
1	0.0960	0.0878	0.9605	1.0918
2	0.0912	0.0895	0.9602	1.0936
3	0.0943	0.0874	0.9603	1.0913
4	0.0870	0.0887	0.9609	1.0928
5	0.0894	0.0888	0.9609	1.0928
6	0.0897	0.0863	0.9613	1.0902
7	0.0873	0.0908	0.9601	1.0950
8	0.0854	0.0925	0.9602	1.0969
9	0.1006	0.0887	0.9610	1.0928
10	0.0886	0.0859	0.9611	1.0897
average over 10 runs			0.9606±0.0004	1.0927±0.0022

5.3 Target-task regressor fine-tuning

The hyperparameter optimization is performed for fine-tuning the target-task regressor. For this purpose, the full data is split into 60:10:30 train-validation-test sets. All the hyperparameters are tuned on the validation set. After hyperparameter tuning, the train and validation sets are merged for prediction on the test set. The models are evaluated using root mean squared error (RMSE) as the error metric (Table S4). In addition, the effect of SMILES augmentation and the gaussian noise is also considered for optimization.

Table S4. Hyperparameter Optimization for the Target-task Regressor Fine-tuning

No. of augmented SMILES	σ_{g_noise}	dropout_rate	epoch[a]	learning_rate[b]	train_rmse	val_rmse
varying the number of augmented SMILES						
0	n.a	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.8029	9.6842
10	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	8.1298	9.0649
15	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.5225	8.2353
20	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.2568	7.9294
25	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.6340	7.3684
30	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	8.2082	8.7325
varying the σ_{g_noise}						

25	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.6340	7.3684
25	0.1	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.5186	7.7146
25	0.2	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.5637	7.6486
25	0.3	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.6618	7.2703
25	0.4	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.7821	7.6305
25	0.5	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.3672	7.3831
25	0.6	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.5718	7.0591
25	0.7	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.4483	7.2374
25	0.8	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.5554	7.5259
varying the dropout rate						
25	0.6	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.5718	7.0591
25	0.6	0.1	[5,6,6,6]	[0.1,0.01,0.001,0.001]	8.4280	7.8635
25	0.6	0.2	[5,6,6,6]	[0.1,0.01,0.001,0.001]	9.2801	8.4921
25	0.6	0.3	[5,6,6,6]	[0.1,0.01,0.001,0.001]	9.7898	8.7552
25	0.6	0.4	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.2815	9.5502
25	0.6	0.5	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.0629	10.0120
varying the number of epochs						
25	0.6	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.5718	7.0591
25	0.6	0.0	[5,5,6,6]	[0.1,0.01,0.001,0.001]	7.7014	7.4396
25	0.6	0.0	[5,5,5,6]	[0.1,0.01,0.001,0.001]	7.8237	7.5924
25	0.6	0.0	[5,5,5,5]	[0.1,0.01,0.001,0.001]	7.6767	7.6799
varying the learning rate						
25	0.6	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	7.5718	7.0591
25	0.6	0.0	[5,6,6,6]	[0.001,0.001,0.001,0.001]	7.6009	7.2955
25	0.6	0.0	[5,6,6,6]	[0.1,0.1,0.01,0.01]	12.6780	12.0981
25	0.6	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.0001]	7.8873	7.3135

[a]The regressor is fine-tuned using gradual unfreezing method in four steps: (i) the regressor, (ii) the regressor and the final LSTM layer, (iii) the regressor and the last two LSTM layers, and (iv) the full model. [b]A notations such as [5,6,6,6] and [0.1,0.01,0.001,0.001] respectively corresponds to the number of epochs used in each of these steps and the respective learning rates. The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

The target-task regressor can be fine-tuned on both the general-domain and target-task LM. The same set of hyperparameters is used in both cases. We have considered 70:30 as well as 80:20 train-test splits. The final performance is reported in terms of RMSE, which is obtained as the average over 30 independent runs on randomized splits of the data. The results are shown in Tables S5 and S6. It is to be noted that the train-test splits for all models, **TL-m1/m2** (with and without gradual unfreezing) and **TL-m0** were maintained the same.

Table S5. Test and Train RMSEs in the Fine-tuning of the Target-task Regressor using a 70:30 Train-test Split[a]

	fine-tuning on general-domain LM			fine-tuning on target-task LM		
sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	6.4533	6.8233	7.0304	7.5313	7.5035	7.5217
2	6.1083	6.3448	6.9809	7.3694	6.8757	6.8035
3	6.0067	6.4718	6.603	8.0548	7.0367	7.0453
4	6.0313	6.9782	7.3341	7.3733	7.5256	7.6526
5	6.1772	6.4124	6.8022	7.4337	7.4487	7.5329
6	5.8693	6.4129	6.7227	7.4207	7.4089	7.4744
7	5.8526	6.6345	7.2806	6.8089	6.9890	6.9129
8	5.8161	6.3125	6.5126	7.4027	6.4263	6.4965
9	6.1986	7.2967	7.8244	7.3881	7.9734	7.7509
10	6.0064	7.3771	8.1775	7.3081	7.9528	8.1098
avg.	6.05±0.19	6.71±0.40	7.13±0.54	7.41±0.30	7.31±0.48	7.33±0.50

[a] The detail on the canonical and TTA SMILES is provided in Section 3.2.

Table S6. Test and Train RMSEs in the Fine-tuning of the Target-task Regressor on a 80:20 Train-test Split[a]

	fine-tuning on general-domain LM			fine-tuning on target-task LM		
sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	6.2294	5.8601	6.2804	7.3763	6.5234	6.5134
2	6.0269	6.0522	6.4118	6.8260	6.8342	6.5356
3	6.3975	5.6400	5.9920	7.4568	6.7074	6.7297
4	5.8919	6.0474	6.7786	6.9291	6.9419	6.9588
5	6.7984	6.3203	6.7909	7.5181	6.6594	6.4822
6	5.6971	5.7310	6.0720	6.9437	6.8856	6.7541
7	5.7554	6.5289	7.0019	7.0235	7.2600	7.2970
8	6.1219	5.6889	5.9164	7.0827	6.4973	6.4020
9	5.9101	6.1526	6.3930	6.8993	7.0935	6.9493
10	5.6370	6.0044	6.4834	6.8524	6.8616	6.7792
11	5.8242	5.9243	6.1411	7.1424	6.4003	6.4171
12	6.1774	6.1630	6.3646	7.0587	6.6822	6.6822
13	6.1666	6.2677	6.5755	6.8645	6.4842	6.6441
14	6.2381	5.9397	6.3149	6.9154	6.2060	6.2769
15	6.2911	6.0899	6.5439	7.0129	6.6084	6.6265
16	6.6277	6.5562	6.7752	7.2204	7.0100	7.0320

17	5.8452	6.0921	6.6174	6.7978	6.8195	6.9653
18	6.1447	6.2053	6.2457	7.0745	6.7582	6.6712
19	6.2606	5.3860	5.7325	7.2278	6.1461	6.2084
20	6.1292	5.6448	5.9941	6.8922	6.427	6.3275
21	6.4961	5.7861	6.1623	7.2003	6.1036	6.1036
22	6.1829	6.1405	6.6096	6.7101	6.6112	6.3908
23	6.0820	6.1022	6.2885	7.0068	6.8911	6.8572
24	6.2985	6.0198	6.3249	7.0876	6.8032	6.8834
25	6.4458	6.2781	6.7553	7.0195	6.8335	6.8358
26	6.3390	6.2569	6.7481	7.0289	6.8586	6.8091
27	5.8508	5.4504	5.9414	7.1111	6.5193	6.3902
28	5.9434	5.7649	6.3735	6.8771	6.7835	6.6399
29	6.0857	6.2369	6.2361	7.0917	6.6127	6.5899
30	6.4922	6.4072	6.7045	7.3085	6.9906	7.0054
avg.	6.15±0.28	6.02±0.29	6.39±0.31	7.05±0.19	6.69±0.27	6.66±0.28

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

5.4 Training the target-task regressor from scratch

In order to assess the impact of transfer learning, the target-task regressor is trained from scratch. Since we are not using any pre-trained or fine-tuned weights, there are no weights to freeze and then use gradual unfreezing. Thus, gradual unfreezing method does not apply to the results of **TL-m0**. The details of separate tuning of the hyperparameters are given in Table S7.

Table S7. Hyperparameter Optimization for Training the Target-task Regressor from Scratch[a]

No. of augmented SMILES	σ_{g_noise}	dropout_rate	epoch	learning rate	train_rmse	val_rmse
varying the number of augmented SMILES						
0	na	0.0	10	0.001	39.0246	30.0918
10	0.0	0.0	10	0.001	10.6959	10.2909
15	0.0	0.0	10	0.001	10.8605	10.4168
20	0.0	0.0	10	0.001	7.8769	8.1942
25	0.0	0.0	10	0.001	7.6071	7.5361
30	0.0	0.0	10	0.001	6.8265	6.7222
35	0.0	0.0	10	0.001	6.3798	6.6649
40	0.0	0.0	10	0.001	6.0463	5.7198
45	0.0	0.0	10	0.001	5.3750	5.9869
varying the σ_{g_noise}						
40	0.0	0.0	10	0.001	6.0463	5.7198
40	0.1	0.0	10	0.001	6.0463	5.9628

40	0.2	0.0	10	0.001	6.0395	5.8264
40	0.4	0.0	10	0.001	6.0557	6.3966
40	0.6	0.0	10	0.001	6.0179	6.2987
varying the dropout rate						
40	0.0	0.0	10	0.001	6.0463	5.7198
40	0.0	0.1	10	0.001	6.6042	5.6645
40	0.0	0.2	10	0.001	6.8145	5.9371
40	0.0	0.3	10	0.001	7.1548	6.0101
varying the learning rate						
40	0.0	0.0	10	0.001	6.0463	5.7198
40	0.0	0.0	10	0.01	6.5501	6.3464
40	0.0	0.0	10	0.1	27.3110	25.7869
varying the number of epochs						
40	0.0	0.0	10	0.001	6.0463	5.7198
40	0.0	0.0	15	0.001	5.4928	5.6153
40	0.0	0.0	20	0.001	5.4433	5.8047

[a]The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

We have performed all the calculations on 70:30 as well as 80:20 train-test splits. The final performance is reported in terms of the average RMSE over 30 independent runs consisting of randomized split of samples. The results are shown in Table S8.

Table S8. Test and Train RMSEs for the Training of Target-task Regressor on 70:30 and 80:20 Train-test Splits[a]

	70:30 split			80:20 split		
sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	6.3381	5.8502	5.5977	7.0893	5.4984	5.3419
2	6.1036	5.8955	5.6760	6.4479	5.7267	5.6131
3	5.9529	5.2694	5.0250	7.2583	5.5663	5.4422
4	5.8306	5.8932	5.7422	6.6200	6.2168	5.9782
5	5.6882	5.8267	5.7434	6.6236	6.2188	6.1086
6	5.9004	5.8004	5.5501	6.5040	5.9542	5.8010
7	5.6354	5.6541	5.5672	6.4047	6.3889	6.0299
8	5.8518	5.0614	4.8203	6.1282	5.7422	5.3796
9	5.7785	6.1539	5.8603	6.7609	5.4653	5.2451
10	6.0280	6.2022	6.0928	6.6262	5.8288	5.5447
avg.	5.91±0.21	5.76±0.36	5.57±0.38	6.65±0.33	5.86±0.32	5.65±0.31

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

Table S9. Test and Train RMSEs for the Training of Target-task Regressor from Scratch on 80:20 Train-test Splits[a]

sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	7.0893	5.4984	5.3419
2	6.4479	5.7267	5.6131
3	7.2583	5.5663	5.4422
4	6.6200	6.2168	5.9782
5	6.6236	6.2188	6.1086
6	6.5040	5.9542	5.801
7	6.4047	6.3889	6.0299
8	6.1282	5.7422	5.3796
9	6.7609	5.4653	5.2451
10	6.6262	5.8288	5.5447
11	6.3694	5.6834	5.3749
12	6.4468	5.4161	5.3669
13	6.6337	6.1422	5.823
14	6.6236	6.4495	6.1944
15	6.6236	5.8336	5.6069
16	6.2396	5.7983	5.5315
17	6.9875	7.3926	7.224
18	7.0065	5.9912	5.8932
19	6.7311	5.7036	5.4764
20	7.0125	5.3726	5.2885
21	6.4365	5.5263	5.313
22	6.8117	5.637	5.542
23	6.8005	5.4132	5.3039
24	6.8355	5.8782	5.5746
25	6.6130	5.372	5.1057
26	7.3400	6.8711	6.6689
27	6.2629	4.9994	4.7606
28	6.8315	5.341	5.3356
29	6.7110	5.7549	5.3379
30	6.6449	6.1499	6.034
avg.	6.68±0.29	5.84±0.49	5.64±0.48

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

It can be noticed from Table S9 that the results are comparable to that obtained using the TL model involving the fine-tuning of the target-task regressor, as presented in Table S6. For reaction-1, with a rich and well-distributed data distribution, the model architecture even without TL seems to be sufficient.

5.5 Y-randomization

The output values are shuffled randomly between various rows (samples) in such a way that in the new dataset no sample is associated with its true output value. With these randomized target values, we fine-tuned the regressor on the general-domain LM. The results for both 70:30 and 80:20 train-test splits are shown in Table S10. The test and train RMSEs are found to be much inferior as compared to when original outputs were used. This is an important observation that assures that the representation of samples using their respective SMILES help algorithm learn well enough to perform the overall tasks, such as the desired regression.

Table S10. Test and Train RMSEs for the Training of the Target-task Regressor on 70:30 and 80:20 Train-test Splits in y-Randomization Runs[a]

	70:30 split			80:20 split		
sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	12.7212	25.1581	31.5939	12.2646	26.1920	24.7763
2	15.3640	25.4082	32.1831	15.2290	24.1319	23.5969
3	14.3583	25.4397	27.1138	15.1000	25.6805	24.7502
4	14.3202	25.4186	30.4887	18.7735	23.6468	24.2945
5	13.4355	24.391	29.6586	16.0224	24.5513	24.2439
6	12.7524	25.2093	25.2796	15.2915	24.4042	24.7140
7	13.2924	24.9921	25.1111	15.0952	25.4276	27.0062
8	15.2859	24.6682	24.4178	15.9007	25.3792	24.4830
9	14.8447	25.0929	27.2585	14.5821	25.2550	24.4093
10	14.7910	24.9346	29.6291	13.6610	25.2007	26.9785
avg.	14.12±1.00	25.07±0.34	28.27±2.81	15.19±1.68	24.99±0.78	24.93±1.14

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

5.6 Out-of-bag performance

In order to evaluate the predictive performance of our model on more challenging data splits, we have used the same out-of-bag splits as that in the original work (ref. 18a in the main manuscript). Here, 15 additives (**A1**-**A15**) are kept in the training set and rest of the 8 additives (**A16**-**A23**) are present only in the test set (Table S1). We could obtain an average RMSE of 10.0 over all additives. Performance of individual additives is provided in Table S11.

Table S11. Out-of-sample Performance of Various Additives

Additive	RMSE (ULMFiT)	RMSE (random forest)
A16	10.1	6.9
A17	6.8	10.5
A18	11.1	13.7
A19	12.1	14.8
A20	7.5	8.6
A21	12.3	11.8
A22	11.2	12.7
A23	9.0	9.2
avg.	10.0	11.3

5.7 Reaction SMILES for prediction

We have used the full reaction SMILES in the following form as an input to the model:

{catalyst}.{arylhalide}.{base}.{additive}>>{product}

The results for 10 different runs on 80:20 train-test split is provided in Table S12. No improvement in the performance is observed with the inclusion of product SMILES (see Table S6 for the comparison).

Table S12. Test and Train RMSEs on a 80:20 Train-test Split with Reaction SMILES and **TL-m1** Model[a]

sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	7.1377	6.2457	6.553
2	7.0245	7.0551	7.3024
3	7.4928	6.9168	7.1621
4	7.1362	7.2082	7.688
5	7.5346	6.4471	6.8722
6	6.8675	7.0831	7.2513
7	6.7266	7.2489	7.5919
8	6.7776	6.1775	6.4546
9	7.3730	6.6223	7.1474

10	6.7575	6.436	6.7206
avg.	7.08±0.31	6.74±0.41	7.07±0.42

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

5.8 Target-task regressor fine-tuning without gradual unfreezing and with a constant learning rate

The hyperparameter optimization is performed for fine-tuning the target-task regressor. For this purpose, the full data is split into 70:10:20 train-validation-test sets. All the hyperparameters are tuned on the validation set. After hyperparameter tuning, the train and validation sets are merged for prediction on the test set. The models are evaluated using root mean squared error (RMSE) as the error metric (Table S13).

Table S13. Hyperparameter Optimization for Fine-tuning the Target-task Regressor Without Gradual Unfreezing[a]

No. of augmented SMILES	σ_{g_noise}	dropout_rate	epoch	learning rate	train_rmse	val_rmse
varying the number of augmented SMILES						
0	na	0.0	10	0.001	36.5606	34.6588
5	0.0	0.0	10	0.001	9.7016	11.1785
10	0.0	0.0	10	0.001	6.7458	7.4287
15	0.0	0.0	10	0.001	6.5306	7.4907
20	0.0	0.0	10	0.001	6.2362	6.9913
25	0.0	0.0	10	0.001	5.7489	6.7347
35	0.0	0.0	10	0.001	5.7612	6.1485
40	0.0	0.0	10	0.001	5.8675	6.1029
45	0.0	0.0	10	0.001	6.2896	5.6366
varying the σ_{g_noise}						
40	0.0	0.0	10	0.001	5.8675	6.1029
40	0.2	0.0	10	0.001	5.8963	6.3416
40	0.4	0.0	10	0.001	5.8806	6.3349
varying the dropout rate						
40	0.0	0.0	10	0.001	5.8675	6.1029
40	0.0	0.1	10	0.001	7.0093	6.6168
40	0.0	0.2	10	0.001	7.6310	7.1578
varying the learning rate						
40	0.0	0.0	10	0.001	5.8675	6.1029
40	0.0	0.0	10	0.01	8.8085	9.0126
40	0.0	0.0	10	0.0001	22.3905	21.2704

varying the number of epochs						
40	0.0	0.0	10	0.001	5.8675	6.1029
40	0.0	0.0	15	0.001	5.5162	5.8514
40	0.0	0.0	20	0.001	4.9379	5.5462

[a]The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

The final performance is reported in terms of RMSE is obtained as the average over 30 independent runs on randomized splits of the data. The results obtained after fine-tuning the regressor on general-domain LM are shown in Table S14. During training, since the data size is large, batch gradient descent is used. In each epoch, the training error is reported as an average over all the batches. If the training error is high at the beginning of an epoch and reduces as the model parameters are updated, it is possible that this average train error remains higher than the test error.

Table S14. Test and Train RMSEs for the Training of Target-task Regressor on 70:30 and 80:20 Train-test Splits[a]

	70:30 split			80:20 split		
sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_R^2 (canonical)	train_RMSE	test_RMSE (canonical)	test_R^2 (canonical)
1	5.4112	4.8409	0.9693	5.3672	4.8194	0.9696
2	4.8020	5.601	0.9584	5.3392	5.2725	0.9629
3	5.5573	4.6125	0.9709	5.6224	4.6599	0.9700
4	5.3713	5.3914	0.9604	6.3896	4.8968	0.9676
5	5.4172	5.0444	0.9659	6.6677	5.2238	0.9629
6	5.2438	4.8443	0.969	6.4651	4.4971	0.9734
7	5.1243	5.5051	0.9597	5.8503	4.8345	0.9697
8	5.1806	4.439	0.9736	6.1494	4.3657	0.9748
9	6.0628	4.9096	0.9677	6.3450	4.7053	0.9709
10	5.4905	5.8752	0.955	6.2308	4.2878	0.9753
avg.	5.37±0.33	5.11±0.47	0.96±0.01	6.04±0.47	4.76±0.33	0.97±0.004

[a] The detail of canonical SMILES is provided in Section 3.2.

Table S15. Test and Train RMSEs for the Training of Target-task Regressor on 80:20 Train-test Splits[a]

sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_R^2 (canonical)
1	5.3672	4.8194	0.9696
2	5.3392	5.2725	0.9629
3	5.6224	4.6599	0.9700
4	6.3896	4.8968	0.9676
5	6.6677	5.2238	0.9629
6	6.4651	4.4971	0.9734
7	5.8503	4.8345	0.9697
8	6.1494	4.3657	0.9748
9	6.3450	4.7053	0.9709
10	6.2308	4.2878	0.9753
11	5.5435	5.0420	0.9655
12	5.4156	4.9344	0.9676
13	5.5457	5.2232	0.9644
14	5.3502	5.0500	0.9648
15	5.2839	5.1569	0.9642
16	5.2359	5.3411	0.9621
17	5.3501	4.7711	0.9684
18	5.5120	5.2757	0.9630
19	5.6664	4.9612	0.9650
20	5.4169	4.7538	0.9687
21	5.3129	4.7336	0.9702
22	5.2394	5.0722	0.9660
23	5.2054	4.9655	0.9674
24	5.9960	4.9805	0.9649
25	5.5912	4.6390	0.9701
26	5.8492	5.2046	0.9619
27	4.9498	4.0351	0.9776
28	5.4010	4.6169	0.9726
29	5.1603	4.8954	0.9684
30	5.5579	5.4164	0.9590
avg.	5.63±0.44	4.89±0.33	0.97±0.004

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

The result of fine-tuning the regressor on target-task LM is provided in Table S16.

Table S16. Test and Train RMSEs for the Training of Target-task Regressor on 80:20 Train-test Splits[a]

sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_R^2 (canonical)
1	6.7221	4.608	0.9722
2	5.6606	5.2294	0.9635

3	7.1952	4.826	0.9678
4	6.6770	4.9726	0.9666
5	6.9531	5.463	0.9594
6	6.6729	5.0676	0.9663
7	6.1477	5.26	0.9641
8	6.4539	4.5177	0.973
9	6.6296	5.2198	0.9641
10	6.5286	4.7361	0.9699
11	5.9328	5.6199	0.9572
12	5.8110	5.3207	0.9623
13	5.8903	5.4989	0.9605
14	5.7585	5.4681	0.9587
15	5.6358	5.3975	0.9608
16	5.6590	5.891	0.9539
17	5.7550	5.4654	0.9585
18	5.8413	5.3224	0.9624
19	6.0070	5.4397	0.9579
20	5.8478	5.3161	0.9608
21	5.7449	5.0797	0.9657
22	5.6083	5.0797	0.9657
23	5.7021	5.5592	0.9591
24	6.3217	5.5949	0.9557
25	5.9936	5.5113	0.9578
26	6.1687	5.705	0.9542
27	5.4529	4.7577	0.9689
28	5.8918	5.3009	0.9639
29	5.6110	5.1962	0.9644
30	5.9394	5.6935	0.9547
avg.	6.07±0.45	5.27±0.34	0.96±0.005

[a] The detail of canonical SMILES is provided in Section 3.2.

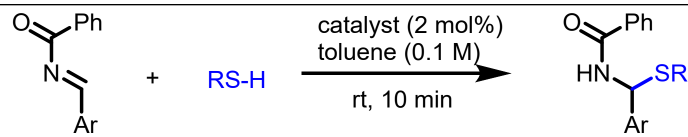
Reaction-2

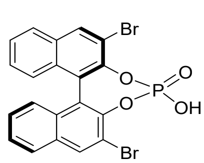
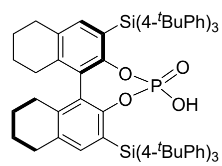
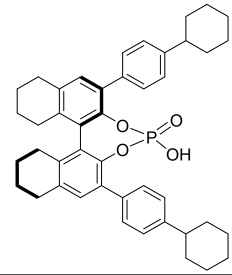
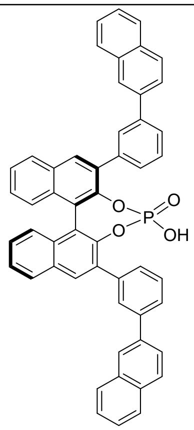
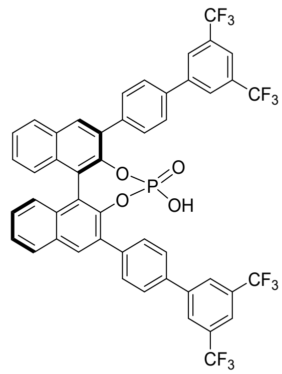
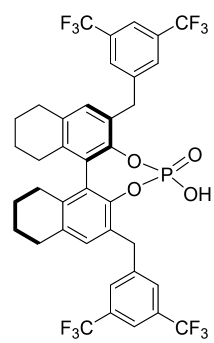
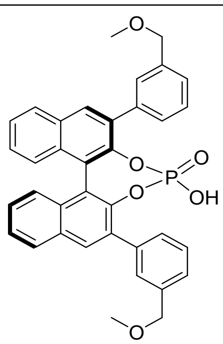
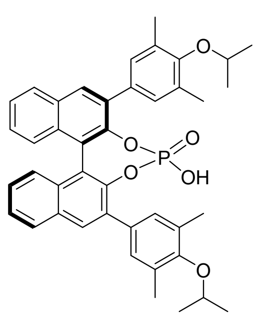
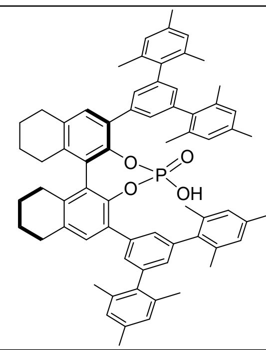
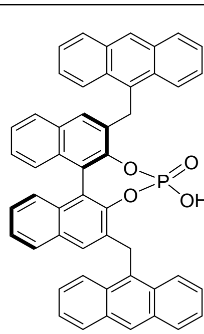
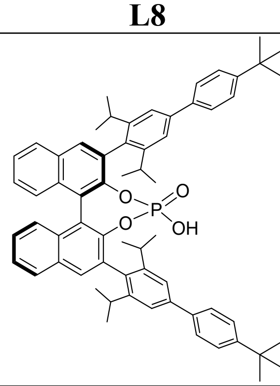
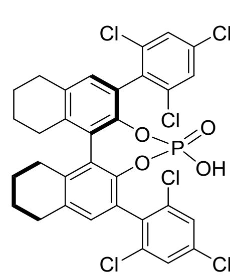
6. Enantioselective formation of N,S-acetals

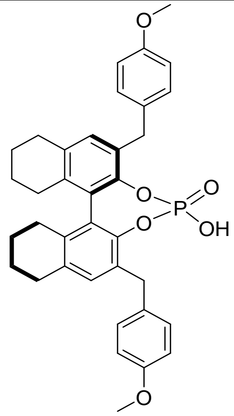
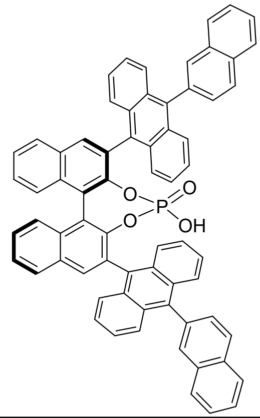
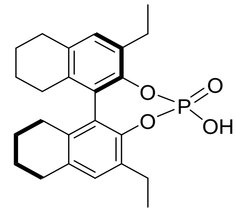
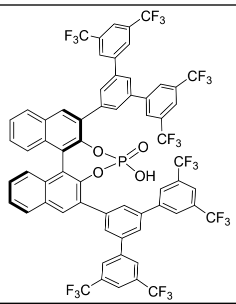
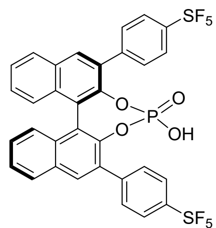
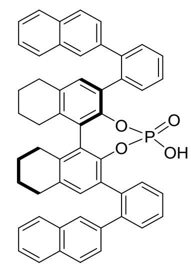
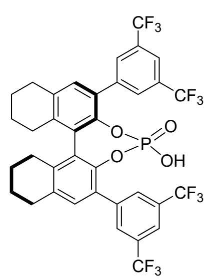
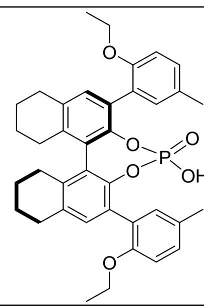
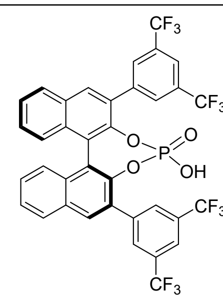
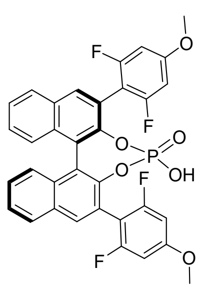
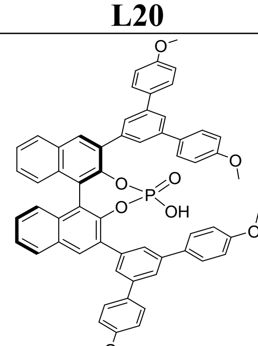
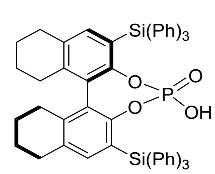
6.1 Summary of reactions

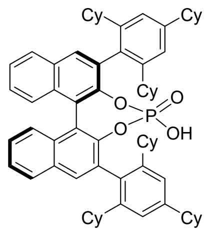
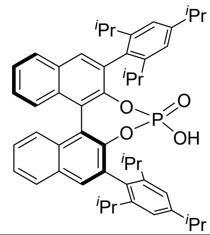
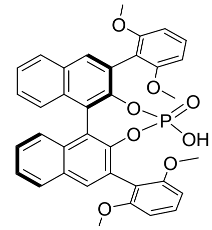
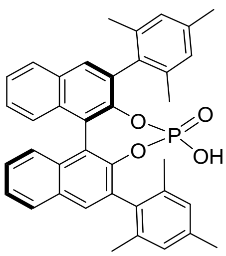
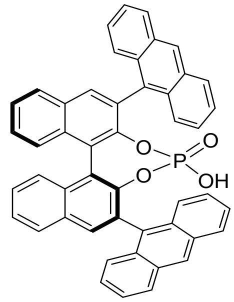
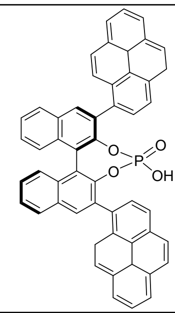
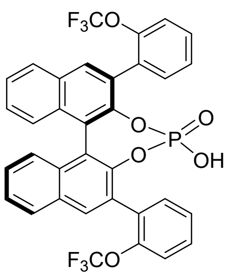
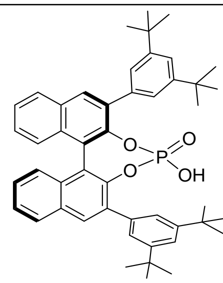
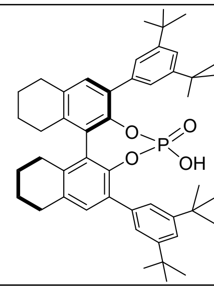
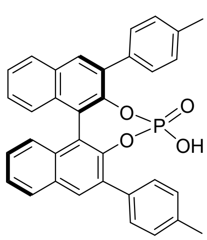
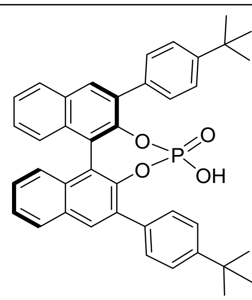
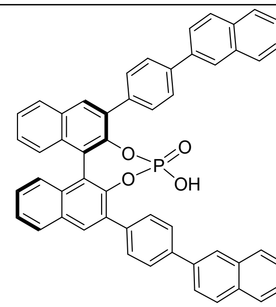
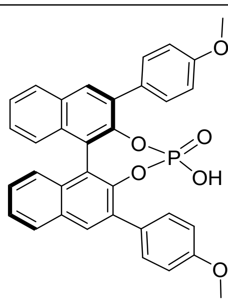
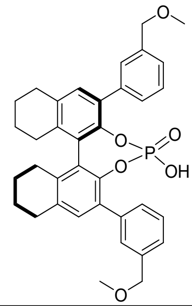
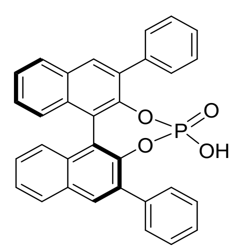
Table S17. Details of Reaction Components

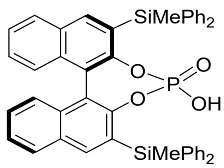
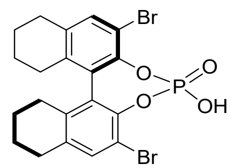
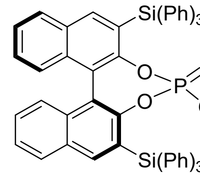
General Reaction Conditions
Reaction Components

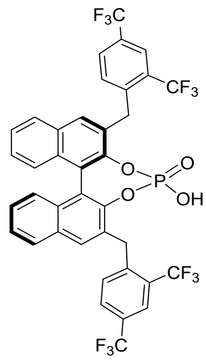


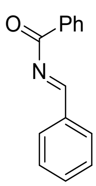
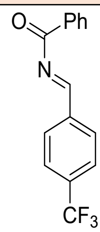
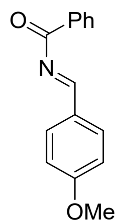
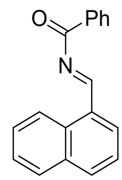
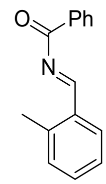
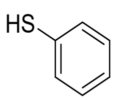
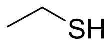
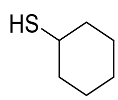
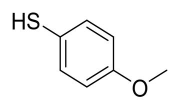
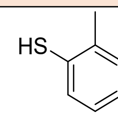
Ligands		
		
L1	**L2**	**L3**
		
L4	**L5**	**L6**
		
L7	**L8**	**L9**
		

L10	**L11**	**L12**
		
L13	**L14**	**L15**
		
L16	**L17**	**L18**
		
L19	**L20**	**L21**
		
L22	**L23**	**L24**

		
L25	**L26**	**L27**
		
L28	**L29**	**L30**
		
L31	**L32**	**L33**
		
L34	**L35**	**L36**
		
L37	**L38**	**L39**

		
L40	**L41**	**L42**


L43

Imines				
				
I1	**I2**	**I3**	**I4**	**I5**
Thiols				
				
T1	**T2**	**T3**	**T4**	**T5**

6.2 Target-task LM fine-tuning

The hyperparameter optimization is performed for fine-tuning the target-task LM. For this purpose, a randomized 80:20 train-test splits were used. The hyperparameters considered for fine-tuning the target-task LM are listed in Table S18. In addition, effect of different number of augmented SMILES is also considered. The model is evaluated using accuracy as the metric of performance, as compiled in Table S18.

Table S18. Hyperparameter Optimization for the Target-task LM Fine-tuning

no. of augmented SMILES	dropout_rate	epoch[a]	learning rate[b]	train_loss	val_loss	accuracy
varying the number of augmented SMILES						
0	0.0	[5,5]	[0.36, 0.01]	0.0946	0.1640	0.9500
25	0.0	[5,5]	[0.36, 0.01]	0.1541	0.1815	0.9228
50	0.0	[5,5]	[0.36, 0.01]	0.1558	0.1889	0.9215
varying the dropout rate						
0	0.0	[5,5]	[0.36, 0.01]	0.0946	0.1640	0.9500
0	0.1	[5,5]	[0.36, 0.01]	0.1302	0.1524	0.9537
0	0.2	[5,5]	[0.36, 0.01]	0.1261	0.1533	0.9535
0	0.3	[5,5]	[0.36, 0.01]	0.1261	0.1613	0.9510
0	0.4	[5,5]	[0.36, 0.01]	0.1334	0.1540	0.9528
0	0.5	[5,5]	[0.36, 0.01]	0.1322	0.1602	0.9503
0	0.6	[5,5]	[0.36, 0.01]	0.1414	0.1505	0.9525
0	0.7	[5,5]	[0.36, 0.01]	0.1451	0.1514	0.9535
0	0.8	[5,5]	[0.36, 0.01]	0.1645	0.1606	0.9502
0	0.9	[5,5]	[0.36, 0.01]	0.1796	0.1612	0.9493
varying the number of epochs						
0	0.2	[5,5]	[0.36, 0.01]	0.1261	0.1533	0.9535
0	0.2	[4,4]	[0.36, 0.01]	0.1538	0.1735	0.9490
0	0.2	[4,5]	[0.36, 0.01]	0.1396	0.1571	0.9531
0	0.2	[5,6]	[0.36, 0.01]	0.1065	0.1449	0.9540
0	0.2	[6,6]	[0.36, 0.01]	0.1133	0.1494	0.9525
varying the learning rate						
0	0.2	[5,6]	[0.36, 0.01]	0.1065	0.1449	0.9540
0	0.2	[5,6]	[1e-1,1e-2]	0.0998	0.1463	0.9526
0	0.2	[5,6]	[1e-1,1e-1]	1.6221	1.0114	0.6851
0	0.2	[5,6]	[1e-2,1e-2]	0.1399	0.1515	0.9530
0	0.2	[5,6]	[1e-2,1e-3]	0.2910	0.2517	0.9294
0	0.2	[5,6]	[1e-1,1e-3]	0.1457	0.2011	0.9431

[a]For the first step, the weights of the LSTM layers are kept frozen and the rest of the model is trained. In the second step, all layers are unfrozen so that the LSTM layers can be fine-tuned. [b]The notations such as [5,5] correspond to the number of epochs in each step and [0.36, 0.01] are the respective learning rates. One hyperparameter is varied at a time keeping others constant. The red color values and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

These optimal hyperparameter combinations are considered for assessing the model performance on 10 independent runs consisting of randomly distributed samples between the train-test splits. The model performance provided in Table S19 is reported in terms of the commonly recommended matrices such as accuracy and perplexity. An average accuracy of ~95% over 10 runs could be obtained.

Table S19. The Calculated Train and Test Accuracies for the Target-task LM Using the Optimal Set of Hyperparameters

sr. no. for runs	train_loss	test_loss	accuracy	perplexity
1	0.1218	0.1546	0.9520	1.1671
2	0.1124	0.1516	0.9542	1.1638
3	0.1074	0.1506	0.9525	1.1625
4	0.1168	0.1500	0.9538	1.1618
5	0.1028	0.1470	0.9532	1.1584
6	0.1090	0.1521	0.9514	1.1642
7	0.1200	0.1545	0.9507	1.1671
8	0.1279	0.1526	0.9524	1.1648
9	0.1157	0.1541	0.9519	1.1666
10	0.1036	0.1537	0.9522	1.1661
average over 10 runs			0.9524±0.0011	1.1642±0.0028

6.3 Target-task regressor fine-tuning

The hyperparameter optimization is performed for fine-tuning the target-task regressor. For this purpose, the full data is split into 70:10:20 train-validation-test sets. All the hyperparameters are tuned on the validation set. After hyperparameter tuning, the train and validation sets are merged for prediction on the test set. The models are evaluated using RMSE as the error metric (Table S20). In addition, the effect of SMILES augmentation with the inclusion of gaussian noise is also considered for optimization.

Table S20. Hyperparameter Optimization for the Target-task Regressor Fine-tuning

No. of augmented SMILES	σ_{g_noise}	dropout_rate	epoch[a]	learning_rate[b]	train_rmse	val_rmse
varying the number of augmented SMILES						
0	n.a	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.6088	9.1313
5	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	12.5853	9.4734
10	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	12.7097	9.1135
20	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	13.3047	8.4346
35	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	15.1743	11.9265
50	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	12.7090	9.3255
[5,1][c]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	12.2330	14.2425

[10,2]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	12.2893	10.8678
[15,3]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.6116	9.8993
[20,4]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.8338	9.8108
[25,5]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	12.6045	10.0374
[25,10]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	12.7084	9.4539
[30,10]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	14.4477	8.3259
[30,5]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.9840	10.1298
[40,10]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.5660	10.8672
[50,10]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4039	8.6911
[60,10]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.2207	11.4303
[50,15]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.1166	8.8644
[75,15]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.6366	10.1598
varying the $\sigma_{g\ noise}$						
[50,10]	0.0	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4039	8.6911
[50,10]	0.1	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4053	8.7040
[50,10]	0.2	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4347	8.4499
[50,10]	0.3	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4226	8.4635
[50,10]	0.5	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4177	9.4609
[50,10]	0.7	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4181	9.5155
varying the dropout rate						
[50,10]	0.3	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4226	8.4635
[50,10]	0.3	0.1	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.8354	9.0548
[50,10]	0.3	0.2	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.1981	10.0010
[50,10]	0.3	0.3	[5,6,6,6]	[0.1,0.01,0.001,0.001]	11.5677	10.3461
varying the number of epochs						
[50,10]	0.3	0.0	[5,5,5,5]	[0.1,0.01,0.001,0.001]	11.7875	8.4578
[50,10]	0.3	0.0	[5,5,5,6]	[0.1,0.01,0.001,0.001]	10.4921	8.9741
[50,10]	0.3	0.0	[5,5,6,6]	[0.1,0.01,0.001,0.001]	10.4651	8.9672
[50,10]	0.3	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4226	8.4635
[50,10]	0.3	0.0	[6,6,6,6]	[0.1,0.01,0.001,0.001]	10.4764	9.8866
varying the learning rate						
[50,10]	0.3	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.001]	10.4226	8.4635
[50,10]	0.3	0.0	[5,6,6,6]	[0.001,0.001,0.001,0.001]	10.5987	9.3813
[50,10]	0.3	0.0	[5,6,6,6]	[0.1,0.01,0.001,0.0001]	10.5584	8.4069
[50,10]	0.3	0.0	[5,6,6,6]	[0.1,0.01,0.01,0.001]	10.3653	9.5633
[50,10]	0.3	0.0	[5,6,6,6]	[0.1,0.01,0.01,0.01]	10.7131	9.6090

[a]The regressor is fine-tuned using gradual unfreezing method in four steps: (i) the regressor, (ii) the regressor and the final LSTM layer, (iii) the regressor and the last two LSTM layers, and (iv) the full model. [b]A notations such as [5,6,6,6] and [0.1,0.01,0.001,0.001] respectively corresponds to the number of epochs used in each of these steps and the respective learning rates. The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters. [c]A notation such as [n,m] refers to differential SMILES augmentation wherein the data with %*ee* ≤ 70 is augmented with [n] SMILES, while that with %*ee* > 70 is augmented with [m] SMILES.

The same set of hyperparameters is used for fine-tuning the target-task regressor on both the general-domain and target-task LM. We have considered 80:20 train-test splits. The final performance is reported in terms of RMSE, obtained as the average over 30 independent runs on randomized splits of the data. The results for individual runs are shown in Tables S21. The performance in terms of mean absolute error (MAE) is also reported in Table S22. It is to be noted that the train-test splits for all models, **TL-m1/m2** (with and without gradual unfreezing) and **TL-m0** were maintained the same.

Table S21. Test and Train RMSEs in the Fine-tuning of the Target-task Regressor[a]

sr. no. for runs	fine-tuning on general-domain LM			fine-tuning on target-task LM		
	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	10.8969	7.6220	7.8429	10.9327	7.9736	8.1443
2	11.4193	8.3091	8.3280	11.4087	9.0648	8.2838
3	10.2813	11.5042	11.7878	10.2634	12.4362	11.5708
4	12.5235	9.5899	9.7144	12.4624	8.7530	8.4618
5	11.7890	8.0276	8.4302	11.7831	7.6025	8.0163
6	12.4842	8.5522	8.9259	12.4499	8.6019	8.6372
7	10.4513	8.2827	7.9552	10.4934	7.9420	7.4382
8	12.9043	9.6445	9.9700	12.9097	9.3820	9.7876
9	11.4337	8.4755	9.1646	11.4744	8.9630	8.3900
10	12.8760	7.9736	8.1443	12.9172	8.4883	8.4687
11	13.4148	8.7315	9.1887	13.4408	8.8211	8.6722
12	10.5204	8.1384	8.5928	10.5875	8.8972	8.5362
13	10.8152	8.7813	8.3856	10.8929	8.1786	8.215
14	11.3905	10.0408	9.8717	11.4167	10.9905	10.0316
15	12.2543	9.2219	8.7812	12.2455	9.1431	8.8569
16	10.9649	8.8474	8.5042	11.0534	8.4428	8.3032
17	12.1229	8.1797	8.6658	12.1674	10.0357	8.9786
18	11.8143	7.7602	7.4142	11.8375	8.4599	7.8503
19	12.5671	7.9418	7.3004	12.6031	7.9604	7.4808
20	10.0408	9.0474	9.0732	10.1054	10.2004	9.2589
21	13.5604	8.9457	9.1877	13.5303	9.0303	9.2041
22	11.4916	9.1867	9.8506	11.4834	8.3005	8.5987
23	10.9593	7.9320	7.8641	10.9907	7.8052	7.7039
24	11.2306	8.6090	8.7527	11.2829	9.0837	9.0468
25	10.7972	9.3635	10.3277	10.8064	10.0026	10.3337
26	10.9956	11.1551	11.5735	11.0043	11.7584	11.502

27	10.6359	8.7473	9.2161	10.6643	9.2188	9.3065
28	11.5866	8.6359	8.5913	11.6264	9.6789	8.788
29	10.9376	8.5583	9.2763	10.9889	8.0363	8.2595
30	10.2055	10.6959	10.3535	10.2302	9.9971	10.2149
avg.	11.51±0.96	8.88±0.96	9.03±1.07	11.54±0.95	9.11±1.15	8.88±1.03

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

Table S22. Test MAEs in the Fine-tuning of the Target-task Regressor[a]

sr. no. for runs	fine-tuning on general-domain LM		fine-tuning on target-task LM	
	test_MAE (canonical)	test_MAE (TTA)	test_MAE (canonical)	test_MAE (TTA)
1	5.6329	5.9930	6.0960	5.9121
2	5.8218	6.2429	6.4515	6.1769
3	8.6536	8.9981	9.7720	9.0147
4	7.4414	7.6737	6.3012	6.1703
5	6.1805	6.3553	5.8992	5.9996
6	6.6447	6.7288	6.5335	6.4401
7	6.1748	6.0865	6.0802	5.628
8	7.0231	7.7940	6.9338	7.6341
9	5.9697	6.8378	6.6345	6.2840
10	6.0960	5.9121	6.2775	6.2262
avg.	6.56±0.92	6.86±1.00	6.70±1.12	6.55±1.02

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

6.4 Training the target-task regressor from scratch

In order to assess the impact of transfer learning, the target-task regressor is trained from scratch.

The hyperparameters are tuned separately, details of which are given in Table S23.

Table S23. Hyperparameter Optimization for Training the Target-task Regressor from Scratch[a]

No. of augmented SMILES	σ_{g_noise}	dropout_rate	epoch	learning rate	train_rmse	val_rmse
varying the number of augmented SMILES						
0	na	0.0	10	0.001	65.8805	64.2196
25	0.0	0.0	10	0.001	31.0083	28.9158
50	0.0	0.0	10	0.001	14.5160	9.6663
75	0.0	0.0	10	0.001	11.3552	8.5079
100	0.0	0.0	10	0.001	12.0342	8.6412
varying the σ_{g_noise}						
75	0.0	0.0	10	0.001	11.3552	8.5079
75	0.2	0.0	10	0.001	17.3483	25.1794
75	0.4	0.0	10	0.001	11.2871	9.1283

75	0.6	0.0	10	0.001	11.2255	8.8158
varying the dropout rate						
75	0.0	0.0	10	0.001	11.3552	8.5079
75	0.0	0.1	10	0.001	11.1855	8.6451
75	0.0	0.2	10	0.001	11.3013	8.9785
75	0.0	0.3	10	0.001	11.2939	9.5520
varying the learning rate						
75	0.0	0.0	10	0.001	11.3552	8.5079
75	0.0	0.0	10	0.01	12.1813	10.1768
varying the number of epochs						
75	0.0	0.0	10	0.001	11.3552	8.5079
75	0.0	0.0	15	0.001	12.5023	8.9775

[a]The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

The calculations are performed on randomized 80:20 train-test splits. The final performance is reported in terms of RMSE and MAE as the average over 30 independent runs on a randomized distribution of samples across test-train splits. The results are shown in Table S24.

Table S24. Test and Train RMSEs for the Training of Target-task Regressor[a]

sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	12.8888	10.1427	8.0141
2	12.5814	10.1091	7.8643
3	12.0354	10.9098	9.2555
4	12.5718	11.2784	10.4672
5	12.9339	9.3306	8.3209
6	12.7063	11.8261	9.8482
7	12.7670	11.0132	9.4511
8	12.5649	10.4499	9.2576
9	13.4493	11.309	9.2871
10	13.1673	9.1956	8.0819
11	12.2905	14.5161	12.2967
12	12.3890	10.7085	9.1438
13	13.0849	10.9115	9.4615
14	12.3712	12.7057	11.1367
15	12.7934	17.5182	16.4749
16	12.7762	12.316	10.679
17	12.4515	12.8179	10.4797
18	12.5317	12.9491	9.6972
19	12.9354	10.0093	8.4564
20	12.4512	12.6905	10.9547

21	12.7782	11.5631	10.3913
22	12.3783	11.8993	8.8372
23	12.8857	11.8810	9.4376
24	13.1704	12.5464	10.8570
25	12.4322	11.4361	10.0360
26	13.0497	11.7193	11.0650
27	12.6859	14.8841	12.1206
28	12.2237	10.2464	8.8790
29	13.0103	12.1166	10.1826
30	12.9494	13.8904	12.2743
avg.	12.71±0.32	11.83±1.75	10.09±1.71

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

6.5 Y-randomization

With the randomized target values, the regressor is fine-tuned on the general-domain LM. The results for are shown in Table S25. The test and train RMSEs are found to be much inferior as compared to when the original/true output values were used.

Table S25. Test and Train RMSEs for the Training of the Target-task Regressor on 80:20 Train-test Splits in y-Randomization Runs[a]

sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	10.2648	32.5373	32.7839
2	11.3377	28.3143	29.6992
3	11.2058	31.4305	31.5316
4	11.0272	32.2311	32.5435
5	11.2274	31.4421	31.8015
6	11.4057	33.4081	31.9833
7	11.1125	33.0163	33.1351
8	11.2239	33.3241	32.9844
9	10.9392	31.7564	32.8172
10	11.3970	33.7541	34.6366
avg.	11.11±0.33	32.12±1.57	32.39±1.28

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

6.6 Target-task regressor fine-tuning without gradual unfreezing and with a constant learning rate

The hyperparameter optimization is performed for fine-tuning the target-task regressor. For this purpose, the full data is split into 70:10:20 train-validation-test sets. All the hyperparameters are tuned on the validation set. After the hyperparameter tuning, the train and validation sets are merged for prediction on the test set. The models are evaluated using root mean square error (RMSE) as the error metric (Table S26).

Table S26. Hyperparameter Optimization for Fine-tuning the Target-task Regressor Without Gradual Unfreezing[a]

No. of augmented SMILES	σ_{g_noise}	dropout_rate	epoch	learning rate	train_rmse	val_rmse
varying the number of augmented SMILES						
0	na	0.0	10	0.001	66.7008	64.4411
25	0.0	0.0	10	0.001	31.3662	25.9997
50	0.0	0.0	10	0.001	13.8926	8.1117
75	0.0	0.0	10	0.001	12.2005	8.6147
100	0.0	0.0	10	0.001	12.3863	8.3888
varying the σ_{g_noise}						
75	0.2	0.0	10	0.001	12.2440	8.4933
75	0.4	0.0	10	0.001	12.1680	8.5478
75	0.6	0.0	10	0.001	12.2455	8.4124
75	0.8	0.0	10	0.001	12.2728	8.1331
varying the dropout rate						
75	0.8	0.0	10	0.001	12.2728	8.1331
75	0.8	0.1	10	0.001	12.5504	8.2329
75	0.8	0.2	10	0.001	12.7164	7.7244
varying the number of epochs						
75	0.8	0.0	10	0.001	12.2728	8.1331
75	0.8	0.0	15	0.001	13.6039	7.7908
75	0.8	0.0	20	0.001	13.1415	8.0967

[a]The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

The final performance is reported in terms of RMSE obtained as the average over 30 independent runs on randomized splits of the data. The results are compiled in Table S27.

Table S27. Test and Train RMSEs for the Training of Target-task Regressor on 80:20 Train-test Splits[a]

sr. no. for runs	fine-tuning on general-domain LM			fine-tuning on target-task LM		
	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	12.2091	7.9266	7.2346	12.2634	8.2011	7.9164
2	11.9734	7.9488	8.1814	11.9803	8.2714	8.6470
3	11.9175	9.8675	9.5886	11.9509	10.1322	9.6454
4	12.0145	8.2484	8.4984	12.0677	9.0014	8.7950
5	12.1635	7.7455	7.5260	12.2328	7.9868	8.0269
6	12.0559	9.1911	8.4049	12.0527	9.5428	9.4087
7	12.0410	9.4401	9.2092	12.0715	8.9392	8.9163
8	11.7743	8.9457	8.4240	11.8314	8.9550	8.8065
9	12.7562	7.2064	7.4976	12.7981	7.8450	7.8081
10	12.5685	7.7330	7.7639	12.5821	7.8648	7.7324
11	11.6808	9.1931	9.2079	11.7781	9.2996	9.5462
12	11.8210	8.3535	8.3877	11.7423	7.9743	8.2870
13	12.3879	9.0628	9.0958	12.4483	8.6132	8.8138
14	11.8599	9.6522	9.8919	11.9398	9.1257	9.3553
15	12.2167	9.1398	9.1603	12.1851	8.8423	8.8890
16	12.1004	8.7654	8.3788	12.1580	8.4331	8.3530
17	11.7919	9.1034	8.9343	11.8025	8.4948	8.3010
18	11.7502	8.4152	8.0256	11.7687	8.1299	8.2299
19	12.2091	8.5902	7.9692	12.2552	7.6532	7.5910
20	11.8720	7.2482	7.2482	11.8781	8.1642	7.9874
21	11.9459	9.8768	8.9874	11.9593	9.1635	8.9013
22	11.5858	7.2692	6.8580	11.6670	7.3880	7.4826
23	12.1025	8.2311	8.1505	12.1868	8.4504	8.3546
24	12.6071	8.8784	8.7314	12.1868	8.3910	9.0997
25	11.7280	9.2542	9.5449	11.8053	9.4989	9.7754
26	12.4865	9.6657	9.7561	12.4650	9.9115	10.0197
27	11.9653	8.1941	8.3417	11.8939	8.1557	8.1450
28	11.5001	8.0005	7.8504	11.7423	8.0908	8.1321
29	12.2241	8.3971	8.8237	12.2561	8.9865	8.8836
30	12.1439	9.8149	9.1986	12.2069	8.8589	8.5880
avg.	12.05±0.30	8.65±0.80	8.50±0.79	12.07±0.27	8.61±0.67	8.61±0.67

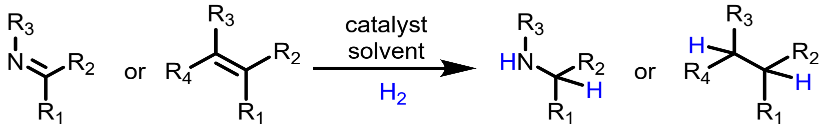
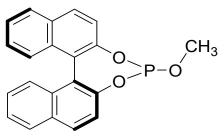
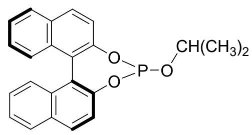
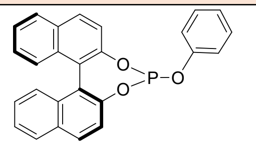
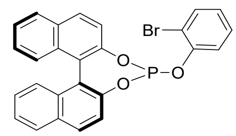
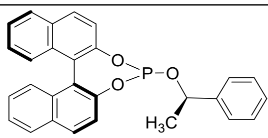
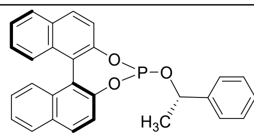
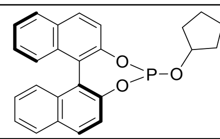
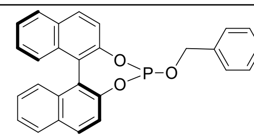
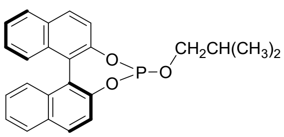
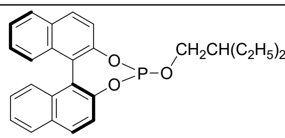
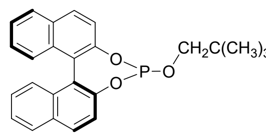
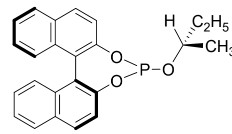
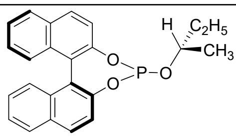
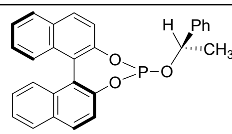
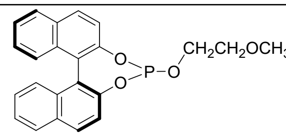
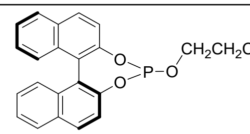
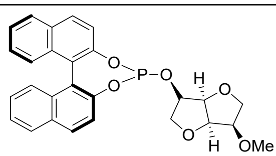
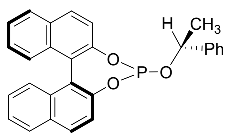
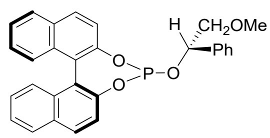
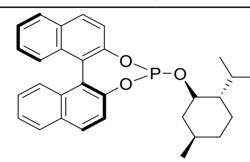
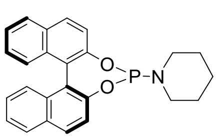
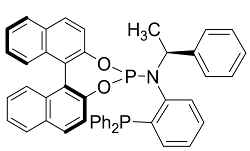
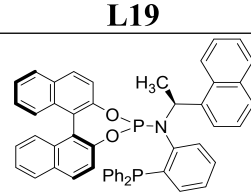
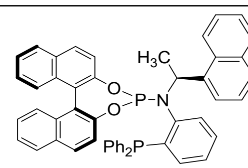
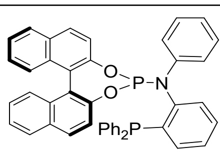
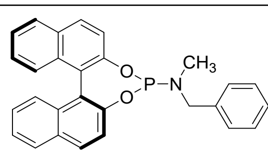
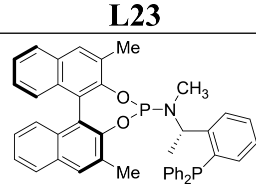
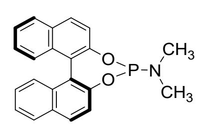
[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

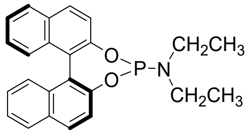
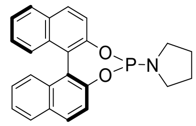
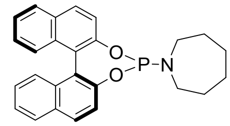
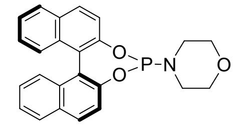
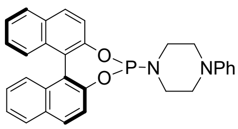
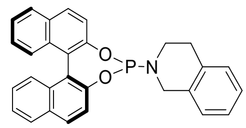
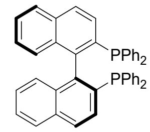
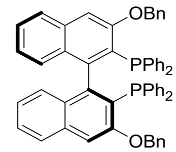
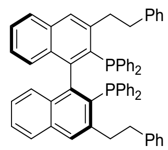
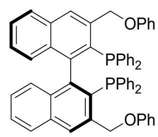
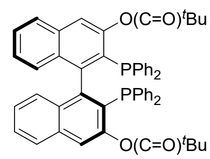
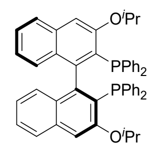
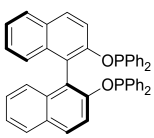
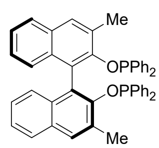
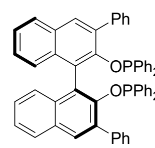
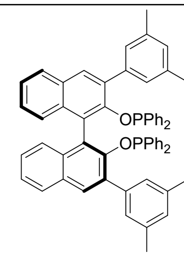
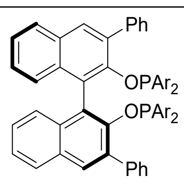
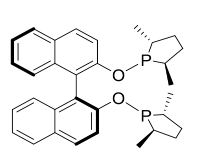
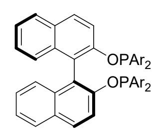
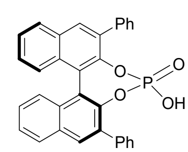
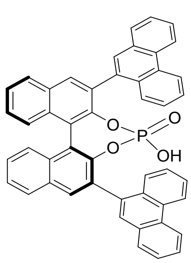
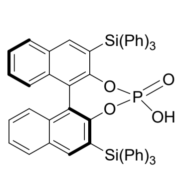
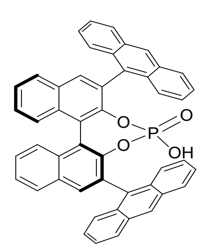
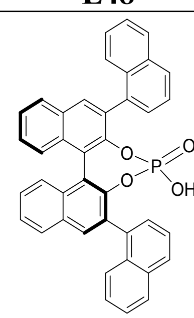
Reaction-3

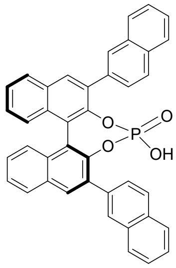
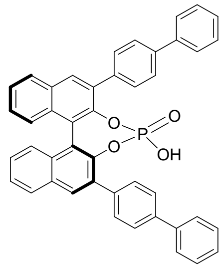
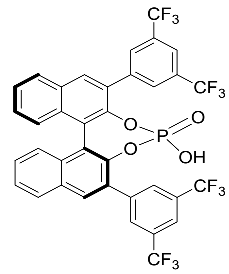
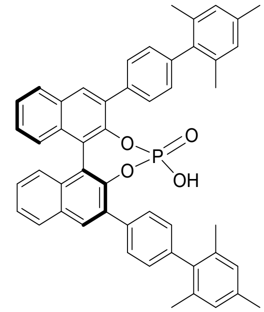
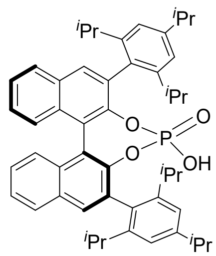
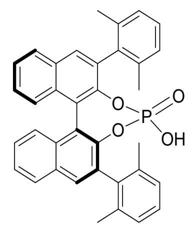
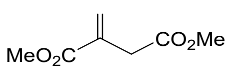
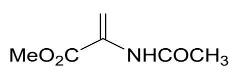
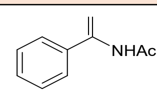
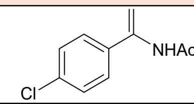
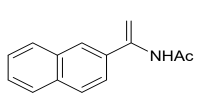
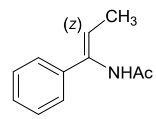
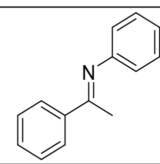
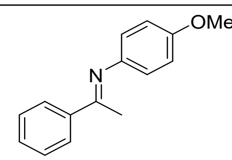
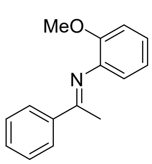
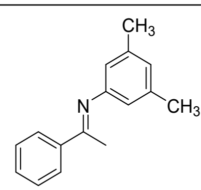
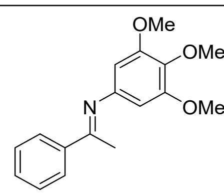
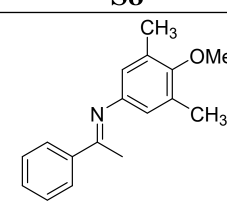
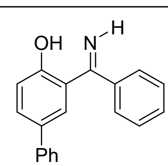
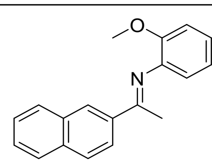
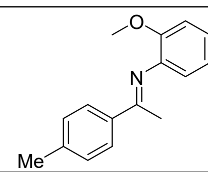
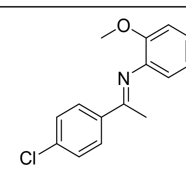
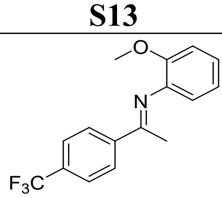
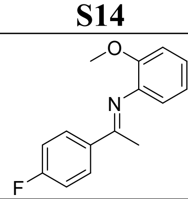
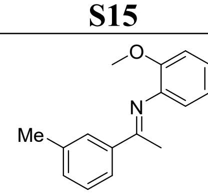
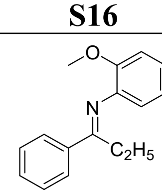
7. Asymmetric hydrogenation of alkenes and imines

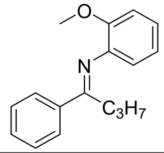
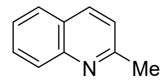
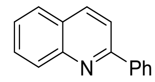
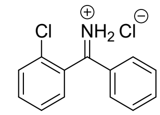
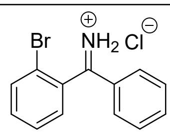
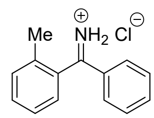
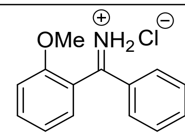
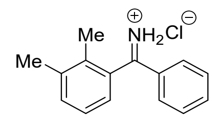
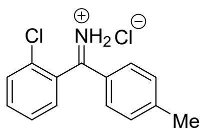
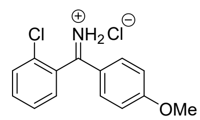
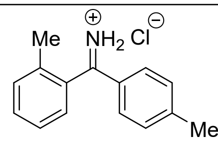
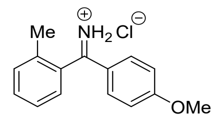
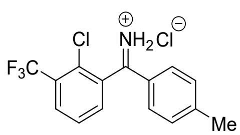
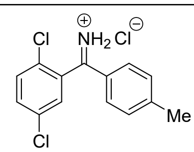
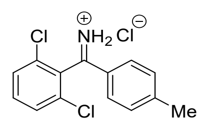
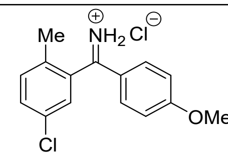
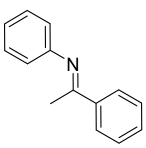
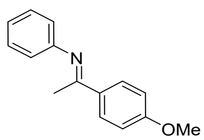
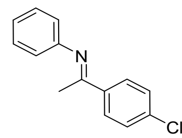
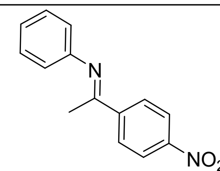
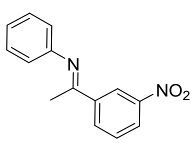
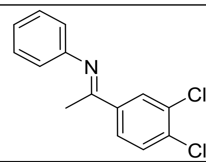
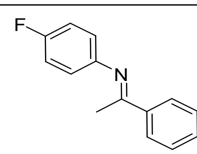
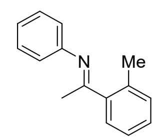
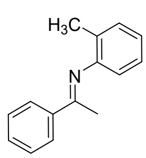
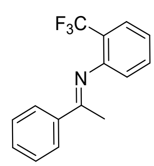
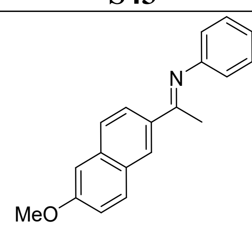
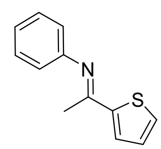
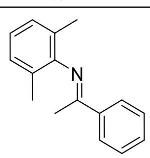
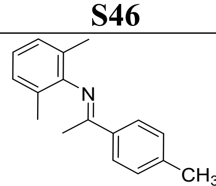
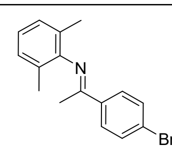
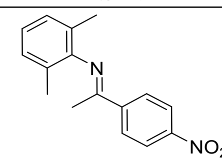
7.1 Summary of reactions

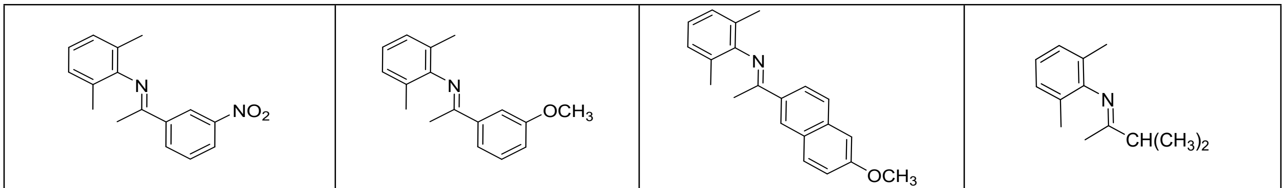
Table S28. Details of Reaction Components

General Reaction Conditions			
			
Reaction Components			
Ligands			
			
L1	**L2**	**L3**	**L4**
			
L5	**L6**	**L7**	**L8**
			
L9	**L10**	**L11**	**L12**
			
L13	**L14**	**L15**	**L16**
			
L17	**L18**	**L19**	**L20**
			
L21	**L22**	**L23**	**L24**
			
L25	**L26**	**L27**	**L28**

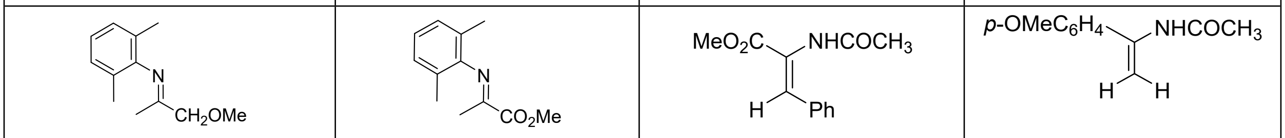
			
L29	**L30**	**L31**	**L32**
			
L33	**L34**	**L35**	**L36**
			
L37	**L38**	**L39**	**L40**
			
L41	**L42**	**L43**	**L44**
 Ar = 3,5-$Me_2C_6H_3$		 Ar = 3,5-$Me_2C_6H_3$	
L45	**L46**	**L47**	**L48**
			
L49	**L50**	**L51**	**L52**

			
L53	**L54**	**L55**	**L56**
			
L57		**L58**	
Substrates			
			
S1	**S2**	**S3**	**S4**
			
S5	**S6**	**S7**	**S8**
			
S9	**S10**	**S11**	**S12**
			
S13	**S14**	**S15**	**S16**
			
S17	**S18**	**S19**	**S20**

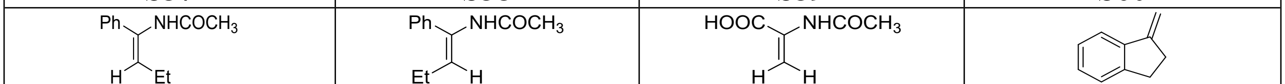
			
S21	**S22**	**S23**	**S24**
			
S25	**S26**	**S27**	**S28**
			
S29	**S30**	**S31**	**S32**
			
S33	**S34**	**S35**	**S36**
			
S37	**S38**	**S39**	**S40**
			
S41	**S42**	**S43**	**S44**
			
S45	**S46**	**S47**	**S48**
			
S49	**S50**	**S51**	**S52**



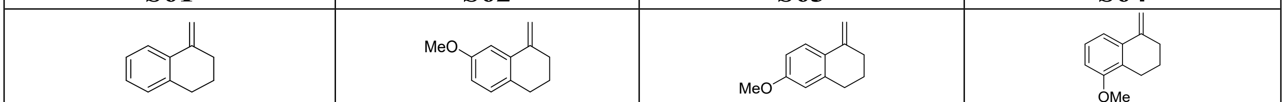
S53	**S54**	**S55**	**S56**



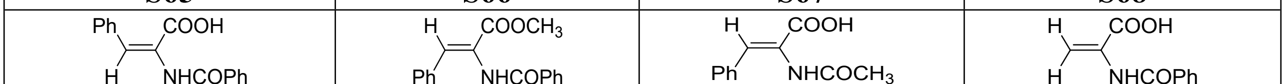
S57	**S58**	**S59**	**S60**



S61	**S62**	**S63**	**S64**

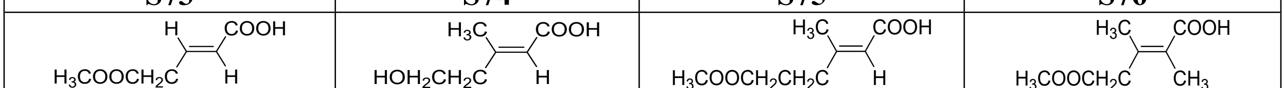


S65	**S66**	**S67**	**S68**

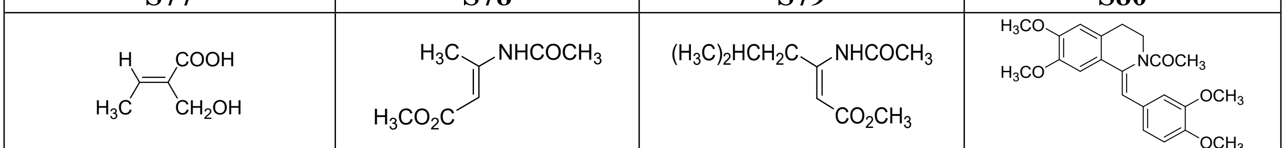


S69	**S70**	**S71**	**S72**

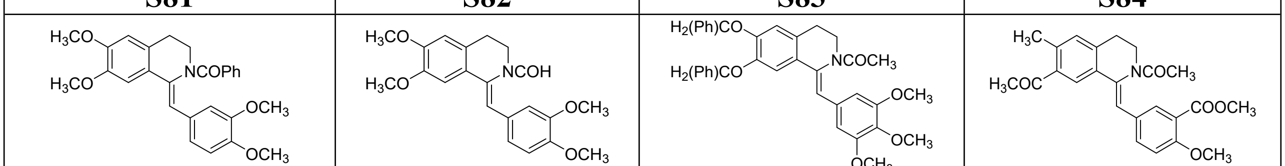
S73	**S74**	**S75**	**S76**



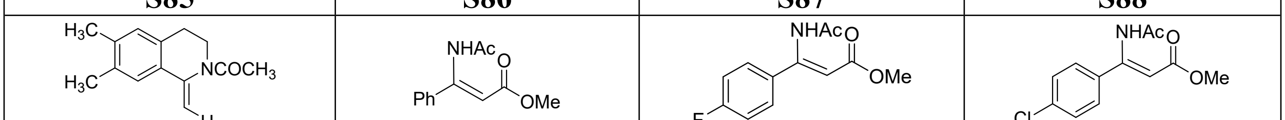
S77	**S78**	**S79**	**S80**



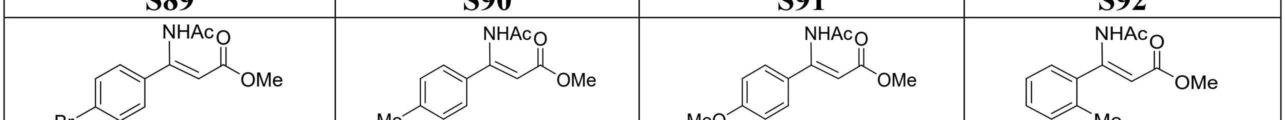
S81	**S82**	**S83**	**S84**



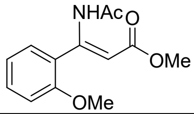
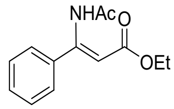
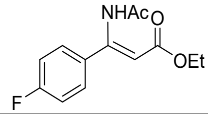
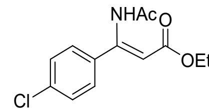
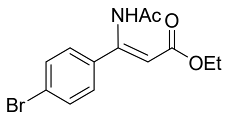
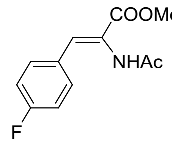
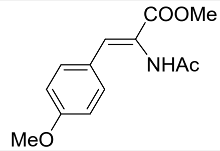
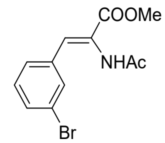
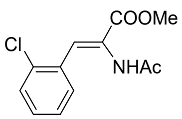
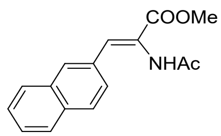
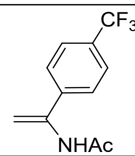
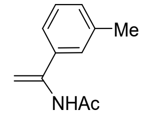
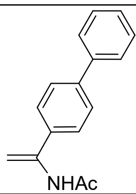
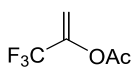
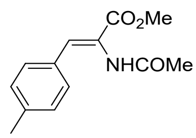
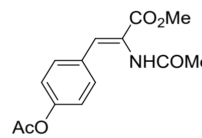
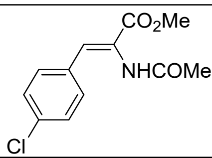
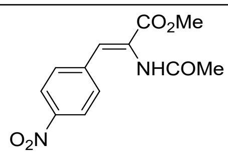
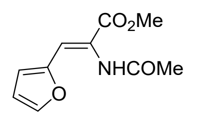
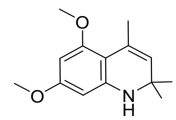
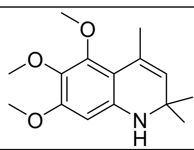
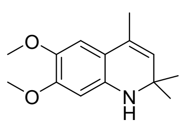
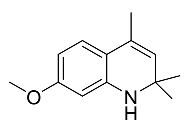
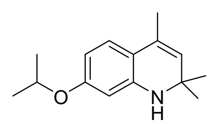
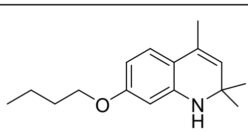
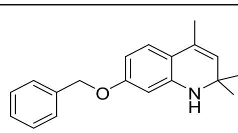
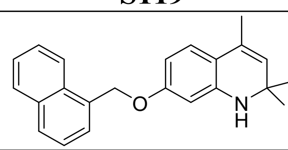
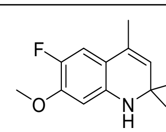
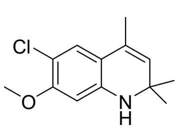
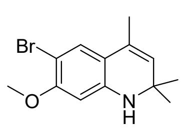
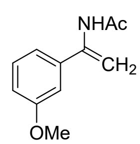
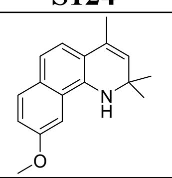
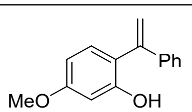
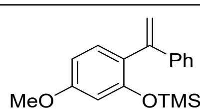
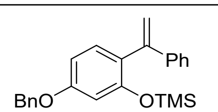
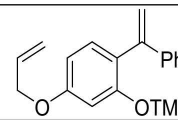
S85	**S86**	**S87**	**S88**

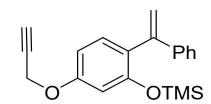
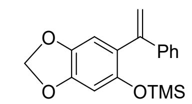
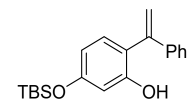
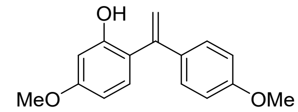
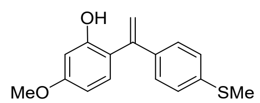
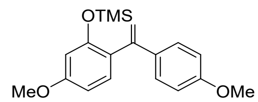
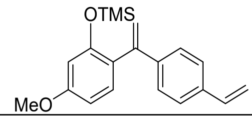
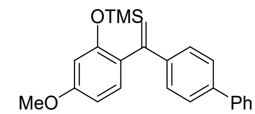
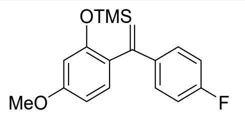
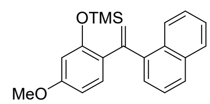
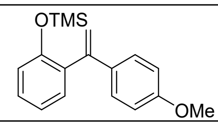
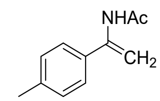
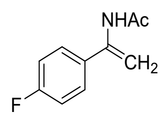
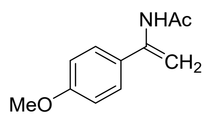
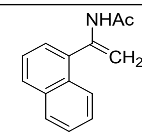
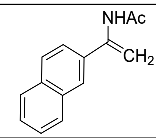
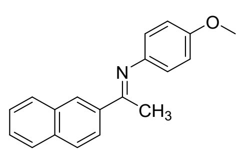
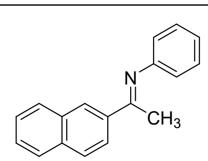
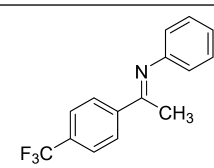
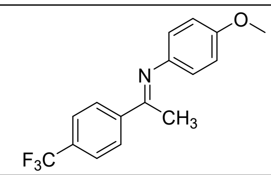
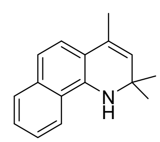
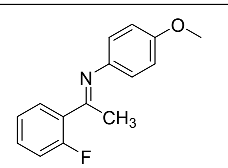
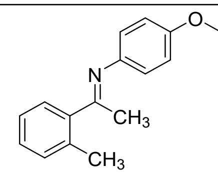
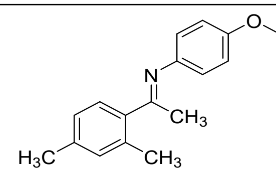
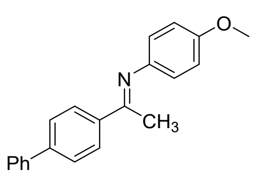
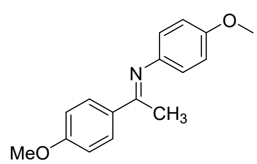
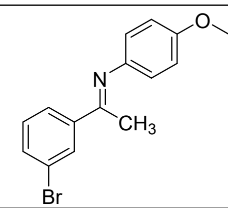
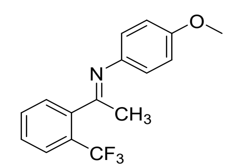
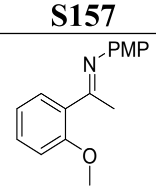
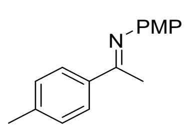
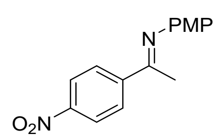
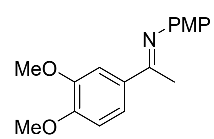
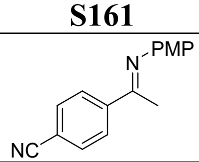
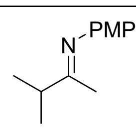
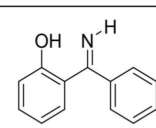
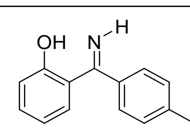


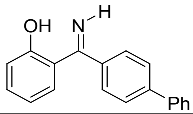
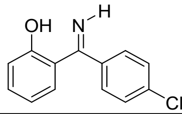
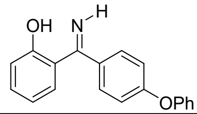
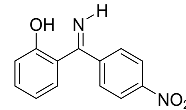
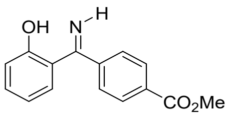
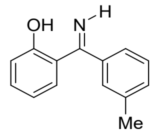
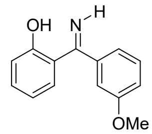
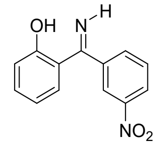
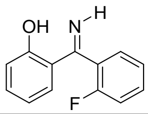
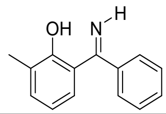
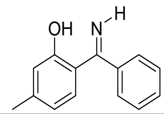
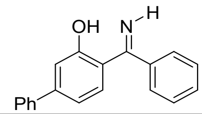
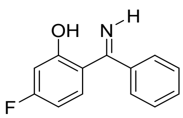
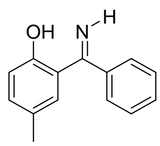
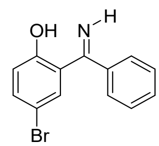
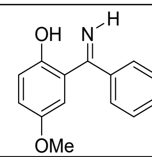
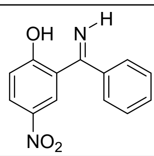
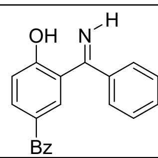
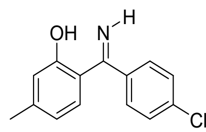
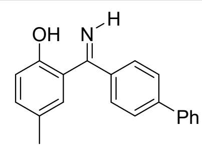
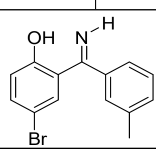
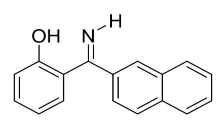
S89	**S90**	**S91**	**S92**



S93	**S94**	**S95**	**S96**

			
S97	**S98**	**S99**	**S100**
			
S101	**S102**	**S103**	**S104**
			
S105	**S106**	**S107**	**S108**
			
S109	**S110**	**S111**	**S112**
			
S113	**S114**	**S115**	**S116**
			
S117	**S118**	**S119**	**S120**
			
S121	**S122**	**S123**	**S124**
			
S125	**S126**	**S127**	**S128**
			
S129	**S130**	**S131**	**S132**

			
S133	**S134**	**S135**	**S136**
			
S137	**S138**	**S139**	**S140**
			
S141	**S142**	**S143**	**S144**
			
S145	**S146**	**S147**	**S148**
			
S149	**S150**	**S151**	**S152**
			
S153	**S154**	**S155**	**S156**
			
S157	**S158**	**S159**	**S160**
			
S161	**S162**	**S163**	**S164**
			
S165	**S166**	**S167**	**S168**

			
S169	**S170**	**S171**	**S172**
			
S173	**S174**	**S175**	**S176**
			
S177	**S178**	**S179**	**S180**
			
S181	**S182**	**S183**	**S184**
			
S185	**S186**	**S187**	**S188**
			
S189		**S190**	
Solvents			
MeOH	EtOH	toluene	*m*-xylene
DCM	THF	benzene	acetonitrile
methyl tert-butyl ether	$CHCl_3$	EtOAc	acetone

7.2 Target-task LM fine-tuning

The hyperparameter optimization is performed for fine-tuning the target-task LM. For this purpose, a randomized 80:20 train-test splits were used. The hyperparameters considered for fine-tuning the target-task LM are listed in Table S29. In addition, effect of different number of augmented SMILES is also considered. The model is evaluated using accuracy as the error metric, as compiled in Table S30.

Table S29. Hyperparameter Optimization for the Target-task LM Fine-tuning

no. of augmented SMILES	dropout_rate	epoch[a]	learning rate[b]	train_loss	val_loss	accuracy
varying the number of augmented SMILES						
0	0.0	[5,5]	[0.25, 0.01]	0.2594	0.5005	0.8760
25	0.0	[5,5]	[0.25, 0.01]	0.1817	0.3524	0.8845
50	0.0	[5,5]	[0.25, 0.01]	0.1775	0.3523	0.8858
75	0.0	[5,5]	[0.25, 0.01]	0.1792	0.3646	0.8859
varying the dropout rate						
25	0.0	[5,5]	[0.25, 0.01]	0.1817	0.3524	0.8845
25	0.1	[5,5]	[0.25, 0.01]	0.1978	0.3356	0.8862
25	0.2	[5,5]	[0.25, 0.01]	0.2008	0.3289	0.8858
25	0.3	[5,5]	[0.25, 0.01]	0.2081	0.3245	0.8870
25	0.4	[5,5]	[0.25, 0.01]	0.2140	0.3370	0.8838
25	0.5	[5,5]	[0.25, 0.01]	0.2132	0.3234	0.8867
25	0.6	[5,5]	[0.25, 0.01]	0.2185	0.3280	0.8852
varying the number of epochs						
25	0.5	[5,5]	[0.25, 0.01]	0.2132	0.3234	0.8867
25	0.5	[4,4]	[0.25, 0.01]	0.2165	0.3292	0.8850
25	0.5	[4,5]	[0.25, 0.01]	0.2131	0.3221	0.8865
25	0.5	[5,6]	[0.25, 0.01]	0.2036	0.3308	0.8860
25	0.5	[6,6]	[0.25, 0.01]	0.2019	0.3295	0.8871
varying the learning rate						
25	0.5	[5,5]	[0.25, 0.01]	0.2132	0.3234	0.8867
25	0.5	[5,5]	[1e-1,1e-2]	0.2061	0.3271	0.8870
25	0.5	[5,5]	[1e-1,1e-1]	1.0833	1.0731	0.6625
25	0.5	[5,5]	[1e-2,1e-2]	0.1983	0.3249	0.8875
25	0.5	[5,5]	[1e-2,1e-3]	0.2517	0.3705	0.8748
25	0.5	[5,5]	[1e-1,1e-3]	0.2616	0.3716	0.8727

[a]For the first step, the weights of the LSTM layers are kept frozen and the rest of the model is trained. In the second step, all layers are unfrozen so that the LSTM layers can be fine-tuned. [b]The notations such as [5,5] correspond to the number of epochs in each step and [0.25, 0.01] are the respective learning rates. One hyperparameter is varied at a time keeping others constant. The red color values and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

These optimal set of hyperparameters are considered for assessing the model performance on 30 independent runs on randomly selected train-test splits. The model performance provided in Table S30 is reported in terms of the commonly recommended metrics such as accuracy and perplexity. An average accuracy of ~88% over 10 runs could be obtained.

Table S30. The Calculated Train and Test Accuracies for the Target-task LM Using the Optimal Set of Hyperparameters

sr. no. for runs	train_loss	test_loss	accuracy	perplexity
1	0.2099	0.3316	0.8874	1.3931
2	0.2131	0.3192	0.8912	1.3761
3	0.2073	0.3397	0.8858	1.4045
4	0.2162	0.3407	0.8854	1.4060
5	0.2170	0.3514	0.8815	1.4211
6	0.2173	0.3323	0.8887	1.3942
7	0.2132	0.3402	0.8862	1.4053
8	0.2174	0.3272	0.8896	1.3871
9	0.2186	0.3475	0.8849	1.4155
10	0.2131	0.3323	0.8878	1.3942
average over 10 runs			0.8869±0.0027	1.3997±0.0134

7.3 Target-task regressor fine-tuning

The hyperparameter optimization is performed for fine-tuning the target-task regressor. For this purpose, the full data is split into 70:10:20 train-validation-test sets. All the hyperparameters are tuned on the validation set. After hyperparameter tuning, the train and validation sets are merged for prediction on the test set. The models are evaluated using root mean squared error (RMSE) as the error metric (Table S31). In addition, the effect of SMILES augmentation and the gaussian noise is also considered for optimization.

Table S31. Hyperparameter Optimization for the Target-task Regressor Fine-tuning

No. of augmented SMILES	σ_{g_noise}	dropout_rate	epoch[a]	learning_rate[b]	train_rmse	val_rmse
varying the number of augmented SMILES						
0	n.a	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	53.2583	55.3002
25	0.0	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	7.6757	10.1177
50	0.0	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	7.0685	10.5804
75	0.0	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	6.7385	11.3816
100	0.0	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	6.4796	9.8598
varying the σ_{g_noise}						

100	0.0	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	6.4796	9.8598
100	0.1	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	6.5274	9.9870
100	0.3	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	6.4147	10.0175
100	0.5	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	6.4875	8.6581
100	0.7	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	6.5236	10.5676
100	0.9	0.2	[4,4,4,6]	[0.1,0.01,0.001,0.001]	6.4714	10.0816
varying the dropout rate						
100	0.5	0.0	100	[0.1,0.01,0.001,0.001]	6.0390	8.9051
100	0.5	0.1	100	[0.1,0.01,0.001,0.001]	6.2551	8.7208
100	0.5	0.2	100	[0.1,0.01,0.001,0.001]	6.4875	8.6581
100	0.5	0.3	100	[0.1,0.01,0.001,0.001]	6.9044	9.5505
100	0.5	0.4	100	[0.1,0.01,0.001,0.001]	6.9461	12.3521
100	0.5	0.5	100	[0.1,0.01,0.001,0.001]	7.5880	11.9915
100	0.5	0.6	100	[0.1,0.01,0.001,0.001]	8.1788	11.2552
100	0.5	0.7	100	[0.1,0.01,0.001,0.001]	8.6819	12.1479
100	0.5	0.8	100	[0.1,0.01,0.001,0.001]	16.5154	18.3346
100	0.5	0.9	100	[0.1,0.01,0.001,0.001]	17.0877	18.8773
varying the number of epochs						
100	0.5	0.2	[2,2,2,4]	[0.1,0.01,0.001,0.001]	7.0960	12.2489
100	0.5	0.2	[2,2,3,4]	[0.1,0.01,0.001,0.001]	6.8996	10.7351
100	0.5	0.2	[2,3,3,4]	[0.1,0.01,0.001,0.001]	6.7606	10.9632
100	0.5	0.2	[3,3,3,4]	[0.1,0.01,0.001,0.001]	6.7643	10.9033
100	0.5	0.2	[3,3,4,4]	[0.1,0.01,0.001,0.001]	6.5815	10.5651
100	0.5	0.2	[3,4,4,4]	[0.1,0.01,0.001,0.001]	6.2720	9.6522
100	0.5	0.2	[4,5,5,5]	[0.1,0.01,0.001,0.001]	6.3624	9.5148
100	0.5	0.2	[5,5,5,5]	[0.1,0.01,0.001,0.001]	6.2547	9.0362
100	0.5	0.2	[5,5,5,6]	[0.1,0.01,0.001,0.001]	6.7054	8.6591
100	0.5	0.2	[3,4,5,6]	[0.1,0.01,0.001,0.001]	6.7926	8.8861
100	0.5	0.2	[5,5,6,6]	[0.1,0.01,0.001,0.001]	6.6360	8.2787
100	0.5	0.2	[5,5,5,7]	[0.1,0.01,0.001,0.001]	6.4233	10.3165
100	0.5	0.2	[5,6,6,6]	[0.1,0.01,0.001,0.001]	6.5439	7.9161
100	0.5	0.2	[6,6,6,6]	[0.1,0.01,0.001,0.001]	6.5653	8.8616
100	0.5	0.2	[5,6,6,6]	[0.1,0.01,0.001,0.0001]	6.7751	9.3255
100	0.5	0.2	[5,6,6,6]	[0.1,0.1,0.1,0.1]	17.9550	18.4490
100	0.5	0.2	[5,6,6,6]	[0.01,0.01,0.01,0.01]	12.0067	16.1513
100	0.5	0.2	[5,6,6,6]	[0.001,0.001,0.001,0.001]	7.1635	7.8248
100	0.5	0.2	[5,6,6,6]	[0.0001,0.0001,0.0001,0.0001]	82.7212	82.3532
100	0.5	0.2	[5,6,6,6]	[0.01,0.01,0.01,0.001]	6.3735	8.3869
100	0.5	0.2	[5,6,6,6]	[0.01,0.01,0.01,0.0001]	6.4117	8.1440
100	0.5	0.2	[5,6,6,6]	[0.001,0.001,0.001,0.0001]	7.3962	8.7910
100	0.5	0.2	[5,6,6,6]	[0.01,0.01,0.001,0.001]	6.6774	8.2106
100	0.5	0.2	[5,6,6,6]	[0.01,0.001,0.001,0.0001]	7.2627	8.2271
100	0.5	0.2	[5,6,6,6]	[0.01,0.01,0.001,0.0001]	6.6716	8.2195
100	0.5	0.2	[5,6,6,6]	[0.01,0.01,0.001,0.00001]	6.6843	8.4328
100	0.5	0.2	[5,6,6,6]	[0.145,0.001,0.001,0.001]	7.1950	7.6958

100	0.5	0.2	[5,6,6,6]	[0.145,0.01,0.001,0.001]	6.8491	7.6244
100	0.5	0.2	[5,6,6,6]	[0.145,0.01,0.001,0.0001]	6.7936	8.2566
100	0.5	0.2	[5,6,6,6]	[0.145,0.01,0.01,0.001]	6.3279	8.8561
100	0.5	0.2	[5,6,6,6]	[0.1,0.01,0.001,0.001]	6.7397	9.0314

[a]The regressor is fine-tuned using gradual unfreezing method in four steps: (i) the regressor, (ii) the regressor and the final LSTM layer, (iii) the regressor and the last two LSTM layers, and (iv) the full model. [b]A notations such as [5,6,6,6] and [0.1,0.01,0.001,0.001] respectively corresponds to the number of epochs used in each of these steps and the respective learning rates. The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

The same set of hyperparameters is used for fine-tuning the target-task regressor on both the general-domain and target-task LM. We have considered 80:20 train-test splits. The final performance is reported in terms of RMSE, which is obtained as the average over 30 independent runs on randomized splits of the data. The results for individual runs are shown in Tables S32. It is to be noted that the train-test splits for all models, **TL-m1/m2** (with and without gradual unfreezing) and **TL-m0** were maintained the same.

Table S32. Test and Train RMSEs in the Fine-tuning of the Target-task Regressor[a]

sr. no. for runs	fine-tuning on general-domain LM			fine-tuning on target-task LM		
	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	6.4831	7.1543	6.9134	6.3951	7.0065	7.0336
2	6.4823	7.7627	7.6061	6.3808	7.5288	7.6528
3	6.4352	7.9334	7.6228	6.4424	7.9988	7.4986
4	6.4070	7.2153	6.7302	6.2699	6.9781	6.2519
5	6.5837	8.8045	8.654	6.4534	8.7123	8.2751
6	6.5015	9.0709	8.3727	6.2509	8.6978	8.3931
7	6.2415	8.3253	8.3893	6.0646	8.8900	8.3231
8	6.4830	9.6592	9.2044	6.1776	8.1002	8.8305
9	6.3996	10.6512	8.951	6.3755	8.9474	9.7692
10	6.0019	12.2429	11.7417	5.9964	10.5252	10.2156
11	6.7729	10.8738	11.3783	6.5344	10.6799	10.49
12	6.4075	6.7824	6.3079	6.2999	6.5016	6.3817
13	6.7712	8.3610	9.1794	6.7755	9.5184	9.9077
14	6.6373	11.8554	11.7642	6.4330	11.8389	11.6335
15	6.0626	8.2104	7.4399	6.1445	10.2678	7.5643
16	6.6253	8.5903	8.4659	6.4368	8.888	9.0602
17	5.8803	9.1805	9.0947	5.8722	8.0000	8.3257
18	6.6342	8.2637	8.5163	6.5425	8.2447	8.1822

19	6.1751	9.1275	8.7403	6.1261	10.1153	9.2296
20	6.0375	7.7473	7.3700	6.1215	6.4382	5.922
21	6.2681	8.8771	9.5586	6.1895	8.5950	9.3903
22	6.9788	7.0352	7.1374	7.9520	8.0962	8.3329
23	6.5947	6.8375	7.1538	7.8181	7.8853	7.446
24	6.2237	9.7672	8.7818	7.1913	9.5189	7.6745
25	6.4295	9.3975	9.4610	7.3495	10.824	9.9727
26	6.3058	7.9112	8.0869	7.5445	8.7828	8.7828
27	6.4070	8.2829	6.4856	7.6023	7.8470	7.6094
28	6.4146	7.0626	6.4744	7.2400	7.5183	6.8904
29	6.3801	7.3265	8.2900	7.3724	7.3979	7.0316
30	6.4853	6.4660	6.8232	7.2683	8.1942	7.6386
avg.	6.42±0.24	8.56±1.46	8.36±1.46	6.65±0.60	8.61±1.34	8.32±1.35

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

7.4 Training the target-task regressor from scratch

To assess the impact of transfer learning, the target-task regressor is trained from scratch. The hyperparameters are tuned separately, details of which are given in Table S33.

Table S33. Hyperparameter Optimization for Training the Target-task Regressor from Scratch[a]

No. of augmented SMILES	σ_{g_noise}	dropout_rate	epoch	learning rate	val_rmse
varying the number of augmented SMILES					
0	na	0.2	10	0.001	86.2492
25	0.1	0.2	10	0.001	79.3839
50	0.1	0.2	10	0.001	65.3602
75	0.1	0.2	10	0.001	43.4010
100	0.1	0.2	10	0.001	28.5789
varying the σ_{g_noise}					
100	0.0	0.2	10	0.001	24.6558
100	0.1	0.2	10	0.001	28.5789
100	0.2	0.2	10	0.001	25.9034
100	0.3	0.2	10	0.001	28.7420
100	0.4	0.2	10	0.001	29.5739
100	0.5	0.2	10	0.001	27.1727
100	0.6	0.2	10	0.001	29.9812
100	0.7	0.2	10	0.001	29.5385
100	0.8	0.2	10	0.001	28.2456
100	0.9	0.2	10	0.001	30.1166
varying the dropout rate					
100	0.0	0.0	10	0.001	23.3387

100	0.0	0.1	10	0.001	21.9790
100	0.0	0.2	10	0.001	24.6558
100	0.0	0.3	10	0.001	25.6355
100	0.0	0.4	10	0.001	21.5097
100	0.0	0.5	10	0.001	28.0119
100	0.0	0.6	10	0.001	26.2109
100	0.0	0.7	10	0.001	28.2496
100	0.0	0.8	10	0.001	21.8387
100	0.0	0.9	10	0.001	23.4488
varying the learning rate					
100	0.0	0.4	10	0.001	21.5097
100	0.0	0.4	15	0.001	22.4838
100	0.0	0.4	20	0.001	15.8147
100	0.0	0.4	25	0.001	11.0299
100	0.0	0.4	30	0.001	12.8097
varying the number of epochs					
100	0.0	0.4	25	0.1737	18.4131
100	0.0	0.4	25	0.1	18.4209
100	0.0	0.4	25	0.01	17.1556
100	0.0	0.4	25	0.001	11.0299
100	0.0	0.4	25	0.0001	79.8296

[a]The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

The calculations are performed on randomized 80:20 train-test splits. The final performance is reported in terms of RMSE and MAE as average over 30 random runs. The results are shown in Table S34.

Table S34. Test and Train RMSEs for the Training of Target-task Regressor[a]

sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	7.0093	8.2416	7.4975
2	7.0247	8.4522	7.8052
3	6.9414	7.6541	7.5151
4	9.9757	11.9544	11.6347
5	6.8868	9.1058	9.1324
6	7.6996	12.3174	11.5632
7	6.6959	8.4980	8.2466
8	6.7101	11.2563	8.9672
9	6.9545	8.7429	9.6581
10	7.2055	12.7918	11.6887
11	7.0258	10.8638	11.1881

12	6.9004	8.5317	7.1883
13	7.1502	10.4378	9.7087
14	7.7765	13.1691	12.1839
15	6.7726	8.3223	7.3839
16	9.5526	14.8627	13.3703
17	7.5351	9.6446	9.6636
18	11.4004	12.3832	14.6339
19	7.1086	10.3574	9.9278
20	8.2092	11.6709	9.5863
21	6.8711	10.3343	10.4505
22	14.0025	15.8687	17.0431
23	6.9827	8.4816	7.7984
24	6.8561	11.1695	9.8039
25	7.7818	11.7818	11.7146
26	10.4114	11.4630	11.4880
27	6.9929	7.3369	7.2143
28	6.5493	7.5074	7.3504
29	8.5874	17.7851	12.3234
30	6.9736	9.0151	8.3124
avg.	7.82±1.68	10.67±2.54	10.07±2.40

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

7.5 Y-randomization

With the randomized target values, the regressor is fine-tuned on the general-domain LM. The results for are shown in Table S35. The test and train RMSEs are found to be much inferior as compared to when original output values were used.

Table S35. Test and Train RMSEs for the Training of the Target-task Regressor on 80:20 Train-test Splits in y-Randomization Runs[a]

sr. no. for runs	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	7.8551	16.9848	17.8943
2	7.9340	24.1535	24.2757
3	7.7546	19.6323	20.9865
4	7.9085	18.4658	18.7339
5	7.2827	19.4899	19.7476
6	7.5065	19.6614	23.3269
7	7.7162	17.9601	18.8415
8	7.5627	18.9467	20.8702

9	7.8764	22.8475	21.5071
10	8.2802	17.8417	18.7951
avg.	7.77±0.27	19.601±2.25	20.50±2.10

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

7.6 Target-task regressor fine-tuning without gradual unfreezing and with a constant learning rate

The hyperparameter optimization is performed for fine-tuning the target-task regressor. For this purpose, the full data is split into 70:10:20 train-validation-test sets. All the hyperparameters are tuned on the validation set. After hyperparameter tuning, the train and validation sets are merged for prediction on the test set. The models are evaluated using root mean squared error (RMSE) as the error metric (Table S36).

Table S36. Hyperparameter Optimization for Fine-tuning the Target-task Regressor Without Gradual Unfreezing[a]

No. of augmented SMILES	σ_{g_noise}	dropout_rate	epoch	learning rate	train_rmse	val_rmse
varying the number of augmented SMILES						
0	na	0.0	10	0.001	86.9343	90.0828
25	0.0	0.0	10	0.001	78.5605	78.4938
50	0.0	0.0	10	0.001	63.2845	63.1162
75	0.0	0.0	10	0.001	42.6581	46.6106
100	0.0	0.0	10	0.001	20.7015	25.0300
125	0.0	0.0	10	0.001	7.5218	8.1621
150	0.0	0.0	10	0.001	6.9791	7.4883
175	0.0	0.0	10	0.001	7.3050	7.1090
200	0.0	0.0	10	0.001	6.7834	7.76074
varying the σ_{g_noise}						
150	0.0	0.0	10	0.001	6.9791	7.4883
150	0.2	0.0	10	0.001	6.9550	7.3471
150	0.4	0.0	10	0.001	6.9596	7.2019
150	0.6	0.0	10	0.001	7.0340	7.3252
varying the dropout rate						
150	0.0	0.0	10	0.001	6.9791	7.4883
150	0.0	0.1	10	0.001	7.6973	6.9331
150	0.0	0.2	10	0.001	8.1398	7.75581
varying the number of epochs						
150	0.0	0.0	10	0.001	6.9791	7.4883

150	0.0	0.0	15	0.001	6.6557	6.70127
150	0.0	0.0	20	0.001	6.3491	7.31165

[a]The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

The final performance is reported in terms of RMSE is obtained as the average over 30 independent runs on randomized splits of the data. The results are shown in Table S37.

Table S37. Test and Train RMSEs for the Training of Target-task Regressor on 80:20 Train-test Splits[a]

sr. no. for runs	fine-tuning on general-domain LM			fine-tuning on target-task LM		
	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	6.1656	8.3282	7.4518	6.2941	7.5533	7.3147
2	5.9816	8.3008	7.6472	6.1454	7.6607	7.7940
3	6.3851	7.5690	7.3072	6.5154	7.9773	7.6173
4	6.2470	7.0483	6.4474	6.4059	7.3502	7.1574
5	6.0943	8.7158	8.6497	6.1612	8.9997	9.1463
6	6.0296	9.6768	9.0668	6.2411	10.3591	9.8667
7	5.9266	7.9520	8.2429	6.1065	9.8461	8.4521
8	5.8263	8.7675	8.3285	6.0228	8.6979	8.1588
9	6.0108	10.0582	10.7021	6.1464	10.3157	10.6121
10	5.6641	10.4968	9.7863	5.8417	9.9422	9.5187
11	6.2025	9.8821	9.9324	6.1775	9.2187	9.5673
12	5.9060	5.9633	5.5552	5.8713	5.8987	5.6992
13	6.1176	9.0206	9.1201	6.1145	9.0234	8.8914
14	5.9750	11.3824	11.481	5.9515	11.3757	11.4354
15	6.0170	7.2819	7.2342	6.0493	7.4005	7.2790
16	6.0186	9.132	9.1146	6.0066	9.2331	9.0435
17	5.6086	8.8846	8.2608	5.6314	7.9259	8.2685
18	6.0329	7.931	7.7475	6.0321	7.6410	7.8745
19	5.9597	9.8558	9.2248	5.9334	9.7983	9.0473
20	5.8749	6.2503	5.7559	5.8535	5.8362	5.6406
21	6.1665	8.7817	9.2328	6.1462	8.3732	8.884
22	6.4105	7.5811	6.9311	6.3992	7.6234	7.217
23	5.9079	7.3392	7.1642	6.0982	8.7617	8.0154
24	5.7256	10.0889	9.2731	5.8651	9.3102	7.8248
25	6.0126	10.1242	10.3381	6.1289	12.2371	10.6295
26	6.0494	7.532	7.9403	6.1748	8.3397	8.1284
27	6.2488	6.2098	6.0733	6.3963	7.0955	6.6839
28	5.9329	6.9747	6.8276	6.0717	7.6027	6.5536
29	5.8946	7.2685	6.6414	5.9899	7.4048	6.9557

30	6.0498	6.9438	7.0423	6.1637	7.4293	7.1086
avg.	6.01±0.18	8.38±1.40	8.15±1.49	6.10±0.19	8.54±1.46	8.21±1.40

[a]The detail on the canonical and TTA SMILES is provided in Section 3.2.

7.7 Importance of composite reaction representation

The input representation is a concatenation of SMILES of ligands and substrates (as described in Section 3). To examine the contribution from ligand and substrate, an additional analysis is performed on Reaction-3 data set. First, the SMILES of ligands are randomly shuffled across rows (keeping substrate SMILES as it is) such that in the new reaction representation, the ligands don't correspond to true output. Similar analysis is done by random shuffling of substrate SMILES (keeping ligand SMILES the same). As can be seen from the data provided in Table S38, the test and train RMSEs obtained using the SMILES of either random ligand or substrate, are found to be much inferior to when the original composite reaction representation was used. The **TL-m1** model without gradual unfreezing is used for these calculations.

Table S38. Test and Train RMSEs of Target-task Regressor with Randomized SMILES of Ligands and Substrates for Reaction-3[a]

sr. no. for runs	randomized ligand SMILES			randomized substrate SMILES		
	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)	train_RMSE	test_RMSE (canonical)	test_RMSE (TTA)
1	6.2999	9.5962	10.2658	6.0626	16.5713	15.9825
2	6.1268	14.0456	13.8714	6.1643	22.5128	22.1144
3	5.9165	11.3729	10.4956	6.5410	23.8279	24.3386
4	6.3746	9.8040	10.7601	6.3212	20.3056	21.0637
5	5.9490	11.1098	11.2667	6.0441	15.6052	14.6409
6	5.7531	12.7582	11.8772	5.8501	20.4670	21.1900
7	6.1662	11.6884	11.5273	6.1297	19.9021	19.6064
8	6.1942	11.8862	12.462	6.1523	20.2041	17.9354
9	5.9133	10.8669	10.5608	6.1880	23.1852	22.4933
10	5.6600	13.3108	13.4323	6.1920	18.8965	18.6219
avg.	6.04±0.23	11.64±1.43	11.65±1.26	6.16±0.18	20.15±2.66	19.80±3.03

[a]The details of the canonical and TTA SMILES is provided in Section 3.2.

8. Data augmentation

The randomized SMILES (generated through different starting atom) are used as a technique for data augmentation. The SMILES augmentation of the training data (details are provided in Section 3.1) is found to be very useful especially for small datasets. The results reflecting the impact of SMILES augmentation on all four reactions is provided in Table S39.

In the case of reaction-3, without any data augmentation, it is observed that the output values for all the samples got predicted in the range 18-20. Since this data dataset is for the 80-100 range, a large RMSE is obtained. As we increased the number of augmented SMILES, a significant improvement in the model performance could be noted as can be gleaned from the data presented in Table S39.

Table S39. Impact of Varying Number of Augmented SMILES on Test Set Performance

No. of augmented SMILES	Test RMSE		
	Reaction-1	Reaction-2	Reaction-3
0	9.0139	8.0897	67.6156
25	6.0044	7.5611	8.7541
50	6.2373	7.4954	8.0240
75	4.9759	7.4071	8.7551
100	-	9.5849	6.8348

9. Time economy

The extraction of chemically relevant molecular features using quantum chemical computations could be resource intensive and time consuming. For example, one of the commonly employed descriptors is vibrational frequencies and the corresponding intensities of the chosen normal mode of vibration. Although one could use relative atomic displacements for automatic identification of normal modes of interest, it is not always easy to ascertain whether a given mode is vibration or rotation, thus inducing a chance of error in judgment. Manually curating such data over thousands of samples can become tedious. There are other molecular features, which might not be amenable to an automated workflow, but would demand individual attention/extraction/decision. The use of SMILES as the molecular representation bypasses all these steps (as illustrated in Fig. S10) and thus provides a highly time economic tool, particularly for a larger samples space.

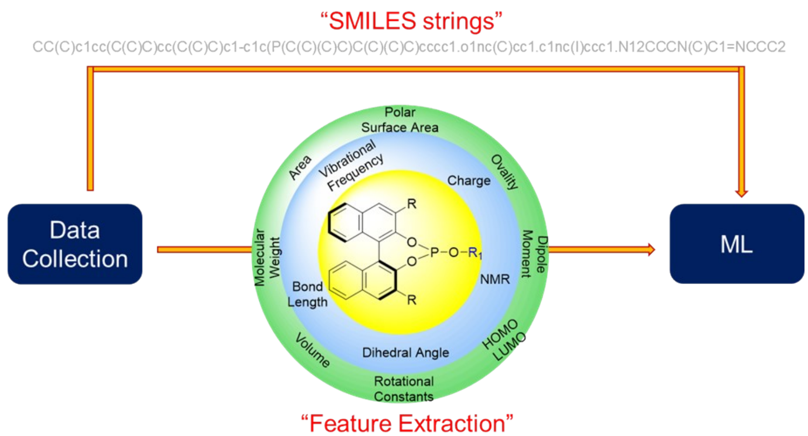


Fig. S10. A general comparison of the conventional workflow involving feature extraction and the one bypassing it by using the SMILES representation for molecules.

A representative case (reaction-3) is considered here for additional discussion. The minimum CPU time consumed for the optimization and frequency calculation of a typical ligand was more than 32 cpu hours, while it is close to 1 cpu hour for the optimization of a small substrate molecule (as collected from the respective output files of the quantum chemical program). There are 58 ligands and 190 substrates (as shown in Table S26), which need to be optimized for collecting the primary features. This would demand approximately ~2000 cpu hours. In addition to this, additional computations for the evaluation of NMR descriptors would demand additional cpu hours across all the above reaction partners. The molecular features like vibrational frequencies, sterimol etc., need extra human attention. On the other hand, the use of SMILES to build the data suitable for ML can bypasses the need for any tiresome feature extraction. Thus, countable (measured in terms of cpu hours), partially countable (codes that would help extract features), and uncountable (human time spent for cogent assessments of the large feature space) aspects that contribute to the overall time spent before one can get the first set of results are far lower in the representation learning method we have employed in this study. Thus, our approach is time-economic.

10. Analysis of the encoder output

In order to get an insight into what the model is actually learning; we extracted the output that the encoder passes to the decoder. The output size is same as the embedding size, i.e., 400 (see Fig. S3). For each reaction, 100 different samples are randomly selected for this analysis. The 100x400 matrix obtained from the encoder output is then processed using the principal component analysis (PCA). Next, a k-means clustering is performed on the first two principal components as obtained through the PCA. Interesting clusters were noticed for all four reaction, details of which are presented below.

10.1 Reaction-1

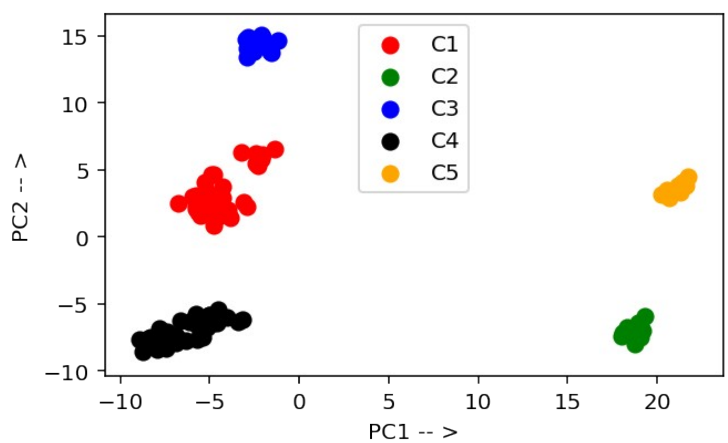


Fig. S11. K-means clustering on reaction-1.

For reaction-1, five distinct clusters were obtained on the basis of ligand and base (Fig. S11). In cluster 1 (denoted as C1, shown in red color), we noticed that **L2-L3** remains together with base **B3** while **L1** forms a group with **B1-B2**. The **L4** formed two separate clusters C2 (green) and C5 (orange), where C2 consists of **B1** and **B2** bases whereas C5 cluster has **B3** as the only base.

Cluster 3 (C3, blue) showed a combination between **L1** and **B3**. In C4 (black), we noticed a combination of **L2** and **L3** ligands. Further details of how various samples are distributed between the five clusters can be gathered from Table S40.

Table S40. Identities of Samples in Different Clusters for Reaction-1 (see Table S1 for the details of sample nomenclature)

Clusters				
C1	C2	C3	C4	C5
L3-B3-AH5-A2	**L4-B2-AH2-A12**	**L1-B3-AH4-A8**	**L3-B2-AH14-A2**	**L4-B3-AH2-A4**
L3-B3-AH3-A4	**L4-B2-AH3-A14**	**L1-B3-AH6-A8**	**L3-B1-AH6-A1**	**L4-B3-AH2-A6**
L3-B3-AH14-A1	**L4-B1-AH15-A9**	**L1-B3-AH2-A10**	**L3-B2-AH11-A3**	**L4-B3-AH13-A1**
L3-B3-AH13-A7	**L4-B2-AH2-A11**	**L1-B3-AH6-A12**	**L3-B1-AH15-A5**	**L4-B3-AH1-A3**
L3-B3-AH15-A12	**L4-B2-AH12-A15**	**L1-B3-AH14-A14**	**L3-B1-AH9-A8**	**L4-B3-AH14-A8**
L3-B3-AH4-A14	**L4-B2-AH10-A17**	**L1-B3-AH15-A9**	**L3-B2-AH4-A10**	**L4-B3-AH1-A10**
L3-B3-AH9-A15	**L4-B2-AH4-A21**	**L1-B3-AH3-A11**	**L3-B2-AH6-A15**	**L4-B3-AH6-A12**
L2-B3-AH8-A4	**L4-B1-AH7-A18**	**L1-B3-AH8-A11**	**L3-B1-AH15-A23**	**L4-B3-AH1-A11**
L2-B3-AH15-A10	**L4-B1-AH2-A22**	**L1-B3-AH3-A16**	**L3-B2-AH13-A23**	**L4-B3-AH15-A16**
L2-B3-AH6-A14		**L1-B3-AH13-A2**	**L3-B2-AH13-A17**	
L2-B3-AH13-A18			**L3-B2-AH15-A17**	
L1-B1-AH1-A4			**L3-B2-AH5-A21**	
L1-B2-AH13-A1			**L3-B1-AH4-A18**	
L1-B1-AH5-A5			**L3-B2-AH1-A20**	
L1-B2-AH15-A7			**L3-B1-AH8-A2**	
L1-B1-AH3-A8			**L2-B1-AH6-A4**	
L1-B2-AH7-A8			**L2-B1-AH12-A3**	
L1-B1-AH10-A12			**L2-B1-AH11-A7**	
L1-B2-AH1-A12			**L2-B2-AH1-A7**	
L1-B2-AH13-A14			**L2-B2-AH14-A8**	
L1-B1-AH12-A9			**L2-B1-AH13-A10**	
L1-B2-AH10-A11			**L2-B2-AH5-A10**	
L1-B2-AH13-A11			**L2-B2-AH13-A10**	
L1-B1-AH14-A13			**L2-B2-AH8-A12**	
L1-B1-AH15-A13			**L2-B2-AH5-A14**	
L1-B2-AH12-A13			**L2-B1-AH8-A23**	
L1-B2-AH10-A17			**L2-B2-AH4-A23**	
L1-B1-AH15-A19			**L2-B2-AH9-A17**	
L1-B1-AH3-A18			**L2-B2-AH14-A16**	
L1-B1-AH4-A18			**L2-B1-AH12-A18**	
L1-B2-AH2-A18			**L2-B1-AH1-A20**	
L1-B2-AH7-A18			**L2-B1-AH5-A20**	
L1-B2-AH7-A20			**L2-B1-AH15-A20**	
L1-B1-AH11-A22			**L2-B2-AH3-A20**	

10.2 Reaction-2

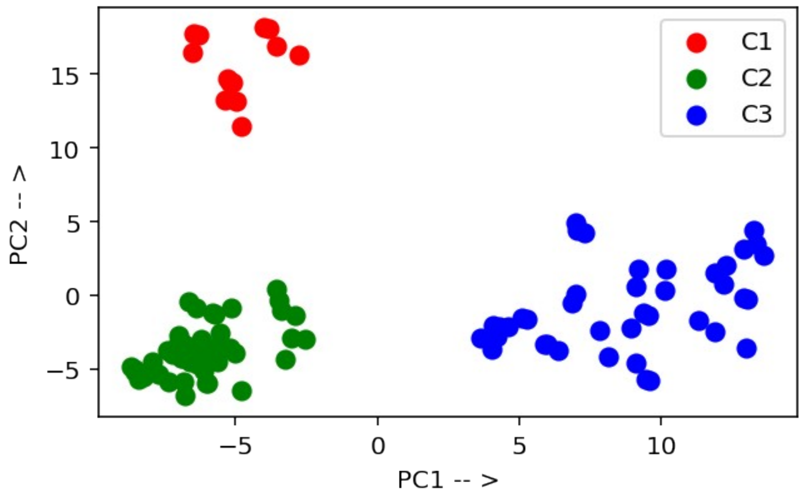


Fig. S12. K-means clustering on reaction-2.

Examination of Fig. S12, for reaction-2, reveals the formation of three distinct clusters on the basis of the ligand (Fig. S12). The reaction details are given in Table S15. In cluster 2 (shown as C2 in green color), shows the presence of ligands bearing relatively larger 3,3'-substituents with -CF_3, -$Si(Ph)_3$ and $C(Me)_3$ groups. On the other hand, in the case of C3 cluster (blue), the ligands have relatively smaller 3,3'-substituents with -OMe, -Me, -Br, -Cl groups on the aryl rings of those substituents. In C1 (red), the size of the 3,3'-substituents are approximately in between that of the ligands present in clusters C2 and C3. More details various samples can be found in Table S41.

Table S41. Details of Samples in Different Clusters for Reaction-2 (see Table S17 for the details of sample nomenclature)

C1	C2	C3
L39-I1-T2	**L25- I4-T3**	**L22- I1-T3**
L39- I1-T3	**L16- I1-T1**	**L22- I2-T2**
L39- I3-T5	**L16- I4-T1**	**L22- I5-T1**
L26- I1-T1	**L16- I4-T3**	**L28- I2-T1**
L26- I1-T2	**L16- I5-T2**	**L8- I1-T4**
L26- I3-T1	**L6- I5-T5**	**L8- I2-T4**
L26- I5-T3	**L10- I4-T1**	**L8- I5-T3**
L42- I4-T4	**L10- I4-T3**	**L13- I1-T5**
L42- I4-T4	**L43- I3-T4**	**L13- I5-T1**
L42- I1-T5	**L43- I5-T1**	**L13- I5-T4**

L29- I3-T5	L40- I2-T2	L23- I3-T5
L29- I4-T3	L40- I5-T3	L1- I2-T1
L29- I1-T2	L32- I1-T1	L1- I2-T4
	L32- I1-T2	L1- I3-T1
	L32- I3-T1	L41- I2-T4
	L32- I3-T3	L41- I2-T5
	L32- I3-T4	L41- I4-T1
	L32- I3-T5	L41- I4-T3
	L32- I5-T2	L41- I5-T1
	L33- I2-T3	L9- I2-T1
	L33- I2-T4	L9- I3-T3
	L33- I4-T5	L9- I4-T4
	L24- I4-T3	L9- I5-T3
	L2- I1-T2	L11- I1-T1
	L2- I3-T5	L11- I1-T3
	L2- I4-T5	L11- I4-T1
	L2- I5-T1	L15- I3-T5
	L5- I1-T1	L15- I4-T4
	L5- I2-T2	L12- I4-T2
	L5- I4-T4	L12- I4-T3
	L14- I4-T5	L37- I1-T1
	L35- I2-T3	L37- I1-T5
	L35- I3-T2	L20- I1-T4
	L35- I4-T1	L20- I1-T5
	L35- I4-T2	L38- I2-T1
	L30- I1-T4	L38- I1-T3
	L30- I4-T3	L38- I4-T1
	L18- I2-T2	L38- I5-T4
	L18- I3-T5	L34- I1-T3
	L4- I2-T3	L34- I4-T5
	L4- I3-T3	
	L4- I3-T4	
	L36- I5-T5	
	L3- I1-T2	
	L3- I4-T2	
	L19- I1-T2	
	L19- I5-T3	

10.3 Reaction-3

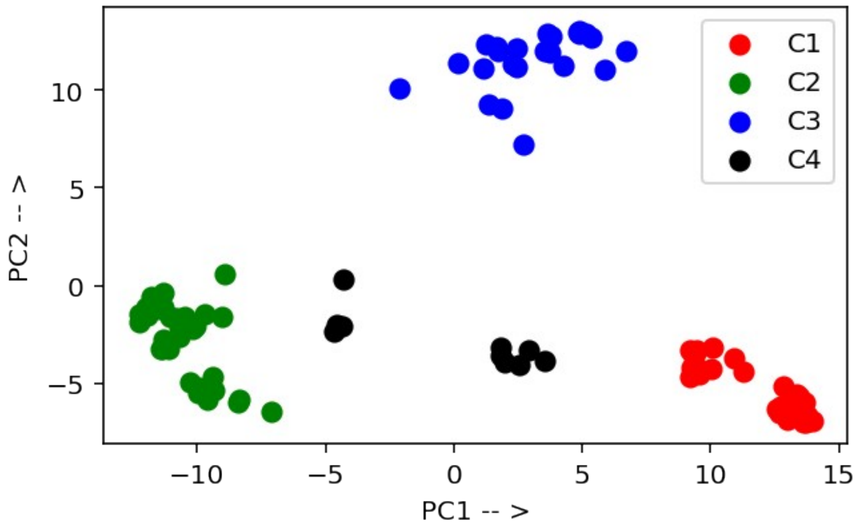


Fig. S13. K-means clustering on reaction-3.

Four distinct clusters were obtained in the case of rection-3, which is formed on the basis of ligand (Fig. S13). The reaction details are provided in Table S26. In cluster C2 (shown in green color), BINOL-phosphite and BINOL-phosphoramidite ligands get grouped together whereas BINOL-phosphoramidite appears exclusively in cluster C4 (black) as well. The similar ligands, BINAP and BINAP-O form cluster C3 (blue). The unique group of BINOL-phosphoric acid organocatalyst forms a distinct cluster C1 (red). More details of sample distribution can be found in Table S42.

Table S42. Details of Samples in Different Clusters for Reaction-3 (see Table S28 for the details of sample nomenclature)

C1	C2	C3	C3
L49-S116	**L6-S1**	**L37-S2**	**L25-S22**
L49-S116	**L1-S2**	**L36-S63**	**L22-S22**
L49-S118	**L2-S2**	**L39-S2**	**L23-S22**
L49-S122	**L1-S3**	**L39-S3**	**L23-S23**
L49-S124	**L8-S3**	**L35-S75**	**L27-S38**
L49-S126	**L13-S3**	**L35-S87**	**L27-S40**
L50-S129	**L14-S3**	**L35-S89**	**L27-S42**
L50-S129	**L15-S3**	**L44-S91**	**L27-S53**
L50-S134	**L9-S5**	**L44-S93**	**L27-S54**

L50-S138	L20-S1	L44-S97	L27-S56
L51-S4	L21-S9	L44-S100	
L51-S148	L21-S17	L44-S101	
L55-S149	L21-S20	L41-S2	
L55-S149	L26-S24	L45-S2	
L55-S149	L26-S27	L42-S3	
L55-S152	L26-S28	L43-S3	
L55-S7	L26-S29	L45-S3	
L55-S158	L26-S30	L43-S106	
L55-S159	L26-S32	L43-S108	
L55-S161	L28-S61	L46-S110	
L57-S155	L28-S62	L47-S59	
L57-S162	L28-S1	L47-S59	
L57-S149	L29-S4	L47-S111	
L57-S168	L30-S3	L47-S112	
L50-S172	L30-S4		
L50-S180	L30-S61		
L50-S182	L21-S59		
L50-S187	L21-S4		
	L21-S61		
	L31-S61		
	L31-S1		
	L32-S2		
	L32-S59		
	L32-S61		
	L32-S62		
	L32-S1		
	L33-S59		
	L33-S4		

11. Comparison of performance of target-task regressor fine-tuning with and without gradual unfreezing

The average performance over 30 runs for all three reactions is provided in Table S43. Here, **TL-m** denotes the fine-tuning without gradual unfreezing and **TL-m'** denotes the fine-tuning with gradual unfreezing.

Table S43. Performance Comparison of TL Models With and Without Gradual Unfreezing

	RMSE	**TL-m1**	**TL-m1'**	**TL-m2**	**TL-m2'**	**TL-m0**
Reaction-1	train	5.63±0.44	6.15±0.28	6.07±0.45	7.05±0.19	6.68±0.29
	test	4.89±0.33	6.02±0.29	5.27±0.34	6.69±0.27	5.84±0.49
Reaction-2	train	12.05±0.30	11.51±0.96	12.07±0.27	11.54±0.95	12.71±0.32
	test	8.65±0.80	8.88±0.96	8.61±0.67	9.11±1.15	11.83±1.75

Reaction-3	train	6.01±0.18	6.42±0.24	6.10±0.19	6.65±0.60	7.82±1.68
	test	8.38±1.40	8.56±1.46	8.54±1.46	8.61±1.34	10.67±2.54

12. Hyperparameter optimization procedure

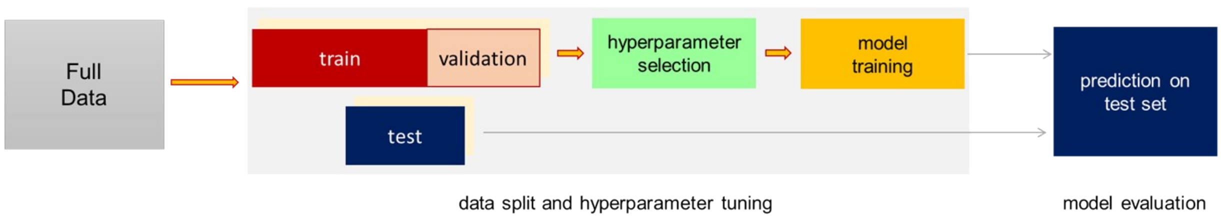


Fig. S14. The training and evaluation procedure employed in this study.

For the purpose of hyperparameter optimization, the full data is first split into 70:10:20 train-validation-test sets. The hyperparameters are tuned on the single train-validation set. After the hyperparameter tuning, the train and validation sets are merged to form a train set. The model is trained on this train set using the optimal values of the hyperparameters. The trained model is further used for prediction on the test set, which is not a part of the train or validation sets (Fig. S14).

In addition, we have performed hyperparameter tuning on 3 random train-validation splits for reaction-3 to examine whether the composition of the validation set has any notable impact. The result of hyperparameter optimization is provided in Table S44. The number of augmented SMILES is 150 with σ_{g_noise} of 0.0 (Table S34).

Table S44. Hyperparameter Optimization for Fine-tuning the Target-task Regressor Without Gradual Unfreezing for Three Random Train-Validation Splits[a]

dropout_rate	epoch	learning rate	train_rmse	val_rmse
split-1				
0.0	10	0.001	6.7209	7.8927
0.1	10	0.001	7.0252	8.4470
0.2	10	0.001	7.2419	8.8648
0.0	10	0.0001	83.5559	85.3966
0.0	10	0.01	9.0139	10.1404

0.0	15	0.001	6.4421	8.1865
0.0	20	0.001	6.0409	8.9503
split-2				
0.0	10	0.001	6.8216	12.5656
0.1	10	0.001	6.8225	12.6217
0.2	10	0.001	7.3789	12.8850
0.0	10	0.0001	83.8786	85.7206
0.0	10	0.01	6.7542	13.9754
0.0	15	0.001	6.3000	13.8625
0.0	20	0.001	5.6592	14.0624
split-3				
0.0	10	0.001	6.5698	11.3338
0.1	10	0.001	7.2143	11.1366
0.2	10	0.001	7.3890	12.7779
0.0	10	0.0001	85.0719	84.0886
0.0	10	0.01	9.3281	15.0218
0.0	15	0.001	6.2271	11.2415
0.0	20	0.001	6.0448	10.5989

[a]The values shown in red color and the highlighted rows respectively represent the best hyperparameter and optimal combination of the hyperparameters.

From Table S43, the optimal set of hyperparameters are: dropout_rate=0.0, learning rate=0.001, and number of epochs=15 (average of 3 runs). It is same as the hyperparameters obtained using a single train-validation split, shown in Table S36. The hyperparameters were chosen based on the balance between the train and validation losses.

13. Performance comparison for reactions 2 and 3

To compare the performance of reaction-2 with previously reported benchmarks, we considered 10 different 600:475 train-test splits. The results are reported in terms of mean absolute error (MAE in ($\Delta G_R^{\ddagger}-\Delta G_S^{\ddagger}$)). In the case of reaction-2, the support vector machine (SVM) gave a MAE of 0.1516±0.0050 kcal/mol (ref. 19a). With the structure-based multiple fingerprint features (MFF) as an alternative representation, provided a MAE of 0.144 kcal/mol (ref. 12a). With our ULMFiT model, we could obtain a MAE of 0.1554±0.0032 kcal/mol. The results with **TL-m1'** model are shown in Table S45.

Table S45. Test and Train MAEs for the Training of Target-task Regressor for Reaction-2

sr. no. for runs	train_MAE	test_MAE	test_R^2
1	0.1539	0.1575	0.8736
2	0.1820	0.1513	0.8887
3	0.1694	0.1580	0.8715
4	0.1859	0.1546	0.8841
5	0.1375	0.1563	0.8831
6	0.1585	0.1569	0.8925
7	0.1341	0.1599	0.8798
8	0.1549	0.1492	0.8904
9	0.1679	0.1552	0.8805
10	0.1729	0.1544	0.8723
avg.	0.1618±0.0173	0.1554±0.0032	0.8817±0.0076

For the performance comparison of reaction-3 with the previous study, we used 100 different 80:20 train-test splits and could obtain an RMSE of 8.59±0.84 and 8.65±1.08 with **TL-m1'** and **TL-m2'** respectively. With **TL-m0**, we observed an inferior RMSE of 10.93±2.59. The reported RMSE for reaction-3 was 8.4±1.8 with the best performing RF algorithm built on quantum mechanically derived descriptors (ref. 20a).

Table S46. Test and Train RMSEs for the Training of Target-task Regressor Using all Three **TL-m** Models for Reaction-3

TL-m2'		**TL-m1'**		**TL-m0**	
train_RMSE	test_RMSE	train_RMSE	test_RMSE	train_RMSE	test_RMSE
6.3596	7.7427	6.8491	7.7064	7.0852	8.7503
6.5294	9.1513	6.4895	8.3939	8.1773	10.1939
6.1404	8.3619	6.3862	9.126	9.0842	11.3878
6.3759	7.5982	6.5717	9.5639	7.2995	8.7129
6.2062	9.1337	6.7032	9.1481	7.2375	8.187
6.6893	7.0062	6.9175	8.0387	8.0346	9.3059
6.1535	8.8577	6.2992	8.6238	7.0534	9.4151
5.9580	9.5976	6.2202	9.6162	11.5595	16.2376
6.5224	9.0893	6.2558	9.4318	6.9502	10.0745
6.4311	11.2201	6.5286	9.6631	6.7083	10.9778
6.4355	7.0218	6.5722	7.8295	9.9927	11.5987
6.4314	12.2679	6.6748	8.8941	7.2373	9.4853
6.3260	6.5091	6.2180	7.6777	6.8573	8.0023
6.7349	7.8849	6.6193	8.4714	7.2625	7.963
6.1890	6.516	6.3976	7.5734	6.9548	10.4554

6.0877	9.2581	6.2552	9.4454	13.5728	16.7806
6.2251	9.9842	6.8797	9.3362	6.9862	9.7504
5.9650	9.4336	6.0561	9.1331	6.7675	10.4718
6.3416	9.9962	6.4241	11.6486	7.6271	10.9206
6.3507	8.5881	6.3522	8.023	9.5326	10.374
6.3001	7.9807	6.5690	8.251	6.8896	7.0424
6.4468	8.1955	6.7614	8.1124	7.1637	9.575
6.6906	9.4829	6.7647	9.8871	7.5059	9.2801
6.2544	8.9914	6.4846	8.4161	14.1811	14.5375
6.1977	9.0554	6.2578	9.4451	8.9318	11.0381
6.2862	8.2909	6.3901	7.7841	7.1652	7.6883
5.8617	9.2846	6.0491	9.119	6.6742	10.6165
6.2013	10.3585	6.3293	9.5902	7.0158	9.2255
6.1308	9.0587	6.3411	8.0687	13.6967	15.289
6.2158	11.0236	6.2269	9.285	8.4542	12.6772
6.7829	8.8889	7.1115	8.8663	11.4690	11.4645
6.1312	8.0224	6.4180	8.5354	11.0476	13.9708
6.7353	7.3326	6.8768	7.4632	7.1910	8.0613
6.5127	8.4573	6.5973	8.8352	9.2854	13.3683
6.3853	6.6789	6.6152	6.6183	12.6572	12.6163
6.2391	9.1461	6.5462	8.2436	7.3167	7.9583
5.8672	9.8927	6.2908	8.8918	13.9086	18.385
6.0173	7.9722	5.9652	8.2694	10.7509	10.6515
6.4434	9.2192	6.8351	8.4501	9.2224	10.5109
6.2421	8.0979	6.2992	8.2141	9.0252	11.5402
6.3599	8.7662	6.5050	8.1053	7.2776	8.8161
6.5503	6.5044	6.8114	7.2577	7.5576	8.1345
6.2893	9.4336	6.7561	10.0375	12.1090	15.622
6.2996	8.8133	6.4655	9.1437	7.3000	10.474
6.3114	8.5949	6.5654	8.6896	10.0342	12.0128
6.5639	10.2249	6.4797	10.6181	6.9237	10.9979
6.2443	7.8957	6.5294	8.013	9.4408	9.5573
5.8243	9.41	6.2933	9.5422	11.1334	12.2715
6.2066	8.9565	6.3285	8.8115	11.4307	12.7895
6.2711	7.6598	6.6107	8.0159	7.1710	9.135
6.2893	8.3606	6.5014	8.7875	9.4482	12.1648
6.0384	8.7015	6.1145	8.8447	9.1486	13.096
6.5600	8.9695	6.6905	9.3179	7.3386	9.6563
5.9977	8.9577	6.2510	9.7344	6.8937	9.3246
6.8244	8.2881	6.7104	9.2268	7.3475	9.04
7.4259	8.8708	6.8110	8.3893	7.6127	8.5317
6.4111	9.4384	6.4590	9.2353	11.7456	18.5653
6.0471	9.304	6.1417	8.5678	8.9360	13.8737
6.3961	9.9917	6.5433	9.2185	9.1878	10.3704
6.3638	7.4356	6.3992	8.5118	7.5263	8.771

6.1355	7.7803	6.3514	8.7419	9.7149	12.6023
6.0951	9.6213	6.5581	8.821	6.9828	8.7979
6.1106	8.7533	6.3890	8.6234	8.5163	14.3999
6.7023	8.5787	6.6716	8.1387	10.1176	11.7949
7.2177	9.1929	7.4639	8.4834	7.4836	9.0815
6.1682	9.0405	6.2492	8.688	6.7740	9.3034
6.4346	9.116	6.7505	9.1196	9.8552	11.3242
5.8264	10.0693	6.0451	9.3965	7.7437	10.6695
6.1219	7.8109	6.1169	7.8599	7.0236	9.97
6.4763	8.2535	6.5209	8.3271	8.0939	10.4769
6.8043	7.9624	7.0347	8.8768	8.0869	8.7809
6.2386	7.4138	6.5720	8.1629	7.7248	10.2693
7.0117	6.6545	7.0586	7.4466	11.1873	11.1965
6.3469	7.8556	6.6646	8.5469	7.1573	9.6236
6.3745	7.5514	6.4412	7.4115	13.2380	12.1121
6.1974	8.7832	6.4485	8.589	6.8861	9.0687
6.7206	7.6666	6.7989	6.9552	8.9228	10.617
6.4144	9.4531	6.5601	9.002	8.4073	9.6566
5.9893	10.0519	6.2179	8.895	7.0711	9.395
6.5541	7.2539	6.5892	7.6875	10.7602	13.6493
6.4159	8.3686	6.4218	7.8765	7.1064	7.7455
6.6771	7.3394	6.8865	7.7212	8.3441	9.8032
6.1469	8.6303	6.0920	7.5235	8.7299	12.024
6.4885	9.4807	6.6381	9.774	8.2087	9.72
6.3740	10.1924	6.4796	8.5694	7.0798	9.3893
6.7281	7.7974	6.9298	8.2473	10.5142	12.8748
6.0307	8.9875	6.5101	9.7896	6.5277	10.2947
6.3017	10.2741	6.6333	8.9043	7.0904	9.0501
6.6943	8.1125	6.9829	8.8701	7.5867	8.9449
6.6132	9.009	6.6996	8.6349	8.7305	10.0797
6.2134	7.0091	6.3858	6.6895	14.4157	14.379
6.0161	8.9351	6.2919	8.3468	12.1007	11.4311
6.4177	8.8755	6.5545	7.9828	7.6130	6.6777
5.9743	9.436	6.2408	8.6589	9.2285	16.1012
6.4007	7.355	6.5467	7.0128	7.3436	7.6384
6.4702	7.8351	6.7397	8.3495	11.2627	12.5806
6.2450	9.9494	6.3340	9.1437	10.4340	11.0897
6.4797	7.3515	6.7329	8.2351	7.5552	9.0015
6.1716	7.9234	6.2203	6.3909	11.9151	21.0517
6.6354	8.5171	6.6917	8.8732	12.3987	12.7537
6.35±0.28	8.65±1.08	6.52±0.26	8.59±0.84	8.83±2.06	10.93±2.59

14. Out-of-sample predictions for reactions 1 and 2

The model generalizability is evaluated on non-random splits similar to that employed in the previous studies. For reaction-1, the isoxazole additives were split into four different training and test sets (ref. 19).[9] For reaction-2, the data was divided into one common training set and three different test sets for (i) substrates, (ii) catalysts, (iii) both substrates-catalysts (refs. 19 and 33). We have used the same out-of-samples splits for prediction using our language model (**TL-m1**). The comparison of results for reactions 1 and 2 is provided in Tables S47 and S48.

Table S47. Performance Comparison on Out-of-Sample Splits for Reaction 1

R^2	additive test 1	additive test 2	additive test 3	additive test 4
Doyle	0.80	0.77	0.64	0.54
One-hot	0.69	0.67	0.49	0.49
MFF	0.85	0.71	0.64	0.18
ULMFiT	0.81	0.81	0.68	0.28

Table S48. Performance Comparison on Out-of-Sample Splits for Reaction 2

MAE (kcal/mol)	substrate test	catalyst test	catalyst-substrate test
Denmark	0.161	0.211	0.238
One-hot	0.178	0.447	0.507
MFF	0.137	0.254	0.282
ULMFiT	0.151	0.256	0.276

15. Comparison of model using paired t-test

The paired t-test is performed to analyze the same set of observations performed under different conditions to find out if the mean of the difference between paired samples is statistically significant. Here, we have used paired t-test to analyze different TL models for all three reactions. The hypothesis for paired t-test is as follows:

Null hypothesis H_0: mean_difference=0

Alternate hypothesis H_1: mean_difference≠0 with a 95% confidence interval and significance level (α) of 0.05. The p-value is compared to α to determine if the difference in means is significant.

If p-value ≤ α, difference between means is statistically significant.

If p-value > α, difference between means is not statistically significant.

The comparison of p-values for various models for all three reactions is provided in Table S49. **TL-m** denotes the fine-tuning without gradual unfreezing and **TL-m'** denotes the fine-tuning with gradual unfreezing.

Table S49. p-values as Obtained from Paired t-Test for All Three Reactions

Sr. no.	Models	Reaction-1	Reaction-2	Reaction-3
1	**TL-m1** and **TL-m1'**	0.000	0.135	0.216
2	**TL-m2** and **TL-m2'**	0.000	0.007	0.674
3	**TL-m1** and **TL-m2**	0.000	0.712	0.239
4	**TL-m1'** and **TL-m2'**	0.000	0.074	0.773
5	**TL-m1** and **TL-m0**	0.000	0.000	0.000
6	**TL-m2** and **TL-m0**	0.000	0.000	0.000

It can be noticed from Table S49 that for all three reactions, the model with TL is significantly different than the one without TL (rows 5 and 6). For reactions 2 and 3, no significant difference is observed either with fine-tuning or gradual unfreezing (rows 1-4).

16. References

(1) Merity, S., Keskar, N. S. & Socher, R. Regularizing and optimizing LSTM language models. *arXiv preprint arXiv*:1708.02182v1 (2017).

(2) Gulordava, K. et al. Colorless green recurrent networks dream hierarchically. *arXiv preprint arXiv*:1803.11138 (2018).

(3) (a) Gaulton, A. et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nuc. acids res.* **40**, D1100-D1107 (2011). (b) The already curated one million SMILES were taken

from https://doi.org/10.1186/s13321-020-00430-x for training the general-domain LM.

(4) Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1-learning rate, batch size, momentum, and weight decay, *arXiv preprint arXiv*:1803.09820 (2018).

(5) Kimber, T. B., Engelke, S., Tetko, I. V., Bruno, E. & Godin, G. Synergy effect between convolutional neural networks and the multiplicity of smiles for improvement of molecular prediction. *arXiv preprint arXiv*:1812.04439 (2018).

(6) O'Boyle, N. M. Towards a universal SMILES representation-a standard method to generate canonical SMILES based on the InChI. *J Cheminform.* **4**, 22 (2012).

(7) Adam, P. et al. Automatic differentiation in PyTorch. In: 31st Conf *Neural Inf Process Syst*. (2017).

(8) Howard, J. & Gugger, S. Fastai: a layered API for deep learning. *Information*. **11**, 108 (2020).

(9) Estrada, J. G., Ahneman, D. T., Sheridan, R. P., Dreher, S. D. & Doyle, A.G. Response to comment on ‘‘Predicting reaction performance in C–N cross-coupling using machine learning’’. *Science* **362**, eaat8763 (2018).