Supplementary Information for

DeepReac+: Deep active learning for quantitative modeling of chemical reactions

Yukang Gong[1], Dongyu Xue[1], Guohui Chuai[1], Jing Yu[1, *], Qi Liu[1, *]

1 Department of Ophthalmology, Shanghai Tenth People's Hospital, Bioinformatics Department, School of Life Sciences and Technology, Tongji University, Shanghai, 200072, China.

* Correspondence: Jing Yu, Qi Liu

Email: dryujing@aliyun.com, qiliu@tongji.edu.cn

This PDF file includes:

Methods

Data preprocessing

1) Dataset A:
The original work contains 4140 experiments of Buchwald-Hartwig cross-coupling reaction.[1] However, when we examined the original data file, it was found that the experiments containing additive 7 were absent so there were only 3960 experiments. Among them, the yield of five experiments is recorded as “NA” and should be removed. Thus, Dataset A consists of the remaining 3955 experiments.

2) Dataset B:
The original work contains 5760 experiments of Suzuki-Miyaura coupling reaction.[2] However, when we examined the original data file, it was found that the solvent of some experiments containing substrate 2d is “THF_V2” which is confusing. In addition, Suzuki-Miyaura coupling reaction can also take place between substrate 2d and substrate 2a/2b/2c but this kind of data is absent. The resulting incomplete reaction space isn’t suitable for the proper simulation of active learning loop. For example, if a data point selected by some sampling strategy isn’t included in the dataset, it can’t be “labeled” in the simulation, which is not consistent with the fact. Thus, we decided to remove the experiments containing substrate 2d and Dataset B consists of the remaining 4608 experiments.

3) Dataset C:
The original work contains 1075 experiments of asymmetric N, S-acetal formation using CPA catalysts.[3] All data are available and unambiguous so Dataset C consists of the same 1075 experiments.

Design details of *DeepReac*

Figure 1A is the overview of our proposed predictive model *DeepReac*. The input can include organic and inorganic components. The former is encoded into dense vectors via *Molecule GAT module* while the latter is first mapped into sparse one-hot vectors and then embedded to dense vectors via an embedding layer. These vectors are then represented as a reaction graph, where each node corresponds to a component and different components can interact with each other through edges. The reaction graph is feed into *Reaction GAT module* to model node interactions. A capsule layer is applied on the output of *Reaction GAT module* to estimate the reaction performance such as yield and stereoselectivity. In the following, we will introduce the details of *DeepReac*.

1) Inputs of *DeepReac*
As for organic components, we utilize RDKit, an open-source cheminformatics software,[4] to convert molecule file into graph format designed by Deep Graph Library (DGL),[5] a Python package built for easy implementation of graph neural network model family. Therefore, any molecule file recognized by RDKit, including SMILES, MOL, SDF, etc., is suitable for input and we choose SDF file as input here. As for inorganic components, all we need is categorical information and they are represented as one-hot encoding vectors. Note that organic components can also be represented as one-hot encoding vectors and feed into the embedding layer depending on whether the detailed information of molecular structure is necessary.

2) Building blocks of *DeepReac*
A core building block of our proposed model is graph attention (GAT) layer[6]. Given a graph, the input to GAT layer is a set of node features, $h = \{h_1,h_2,\cdots,h_n\}$, where n is the number of nodes, and a corresponding adjacency matrix which shows how these nodes are connected to each other. Similar to the self-attention mechanism introduced in the milestone paper[7] in the field of natural language processing, a pair-wise attention score between two neighbors is computed as:

$$e_{ij} = LeakyReLU(\vec{a}^T[Wh_i \parallel Wh_j]) \qquad (1)$$

where $\cdot^T$ represents transposition and $\|$ is the concatenation operation. $\vec{a}$ and W are learnable weight vector and matrix respectively. Then, the attention scores are normalized across all choices of j using the softmax function:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k \in N_i} exp(e_{ik})} \tag{2}$$

where N_i is the neighboring nodes of i in the graph. Finally, the normalized attention scores are used to compute a linear combination of the features corresponding to them, to serve as the output features for every node. Moreover, multi-head attention, which is inspired by the idea of multi-channel applied in the convolutional neural network, is introduced to enrich the model capacity and to stabilize the learning process. Thus, the final output node features are computed as:

$$h_i^{'} = \|_{k=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^{k} W^{k} h_j\right) \tag{3}$$

where K is the number of attention heads, $\|$ represents concatenation and σ is a nonlinear activation function. After above message passing, every updated node features of a graph can be aggregated to obtain graph-level representation in various ways, which is also known as graph pooling or readout. Common readout operations include summation, average, maximum or minimum over all node features. As another crucial building block of *DeepReac*, capsule layer[8] can also work as graph readout operation. Each node feature can be viewed as a lower-level capsule. After dynamic routing-by-agreement which generalizes the static routing of max-pooling, a higher-level capsule can be a richer representation of data. Since a capsule's dynamic routing algorithm can be naturally expressed as a graph algorithm, the capsule layer as well as the GAT layer are implemented with DGL[5].

3) *Molecule GAT module* & *Reaction GAT module*

In *Molecule GAT module*, we apply a two-layer GAT and each layer consists of $K = 4$ attention heads computing hidden features whose dimensionality is 32, followed by an ELU nonlinearity. The input node features whose dimensionality is 74 are extracted from molecule file by *CanonicalAtomFeaturizer* function in DGL[5]. The output node features are aggregated by *WeightedSumAndMax* operation, which combines summation and maximum readout to retain as much molecular information as possible. If there are inorganic components which are represented as one-hot encoding vectors, an embedding layer is then applied upon the one-hot vectors to embed them to dense real-value embedding vectors whose dimension is equal to that of output of *Molecule GAT module*.

In *Reaction GAT module*, we also apply a two-layer GAT and each layer consists of $K = 4$ attention heads computing hidden features whose dimensionality is 32, followed by an ELU nonlinearity. As its input, a reaction graph is constructed by all reaction components which are treated as virtual nodes and connected to each other. The node features come from the output of *Molecule GAT module* and the embedding layer if there are inorganic components. The output node features are aggregated by a capsule layer to produce a final reaction feature which is then passed through one full connected layer to perform regression task including prediction of reaction yield or stereoselectivity.

To visualize the representation learning of *DeepReac*, the informative reaction features outputted by the capsule layer are projected to a 2D plot by PCA (principal components analysis) and t-SNE (t-distributed stochastic neighbor embedding)[9]. These algorithms are implemented using Scikit-Learn,[10] an established Python machine learning package.

4) Training & Hyperparameter optimization of *DeepReac*

Nested cross validation was performed with five-fold random splitting for both inner loop and outer loop on each dataset. In the inner loop, a grid search for optimizing the hyperparameters of *DeepReac* is performed. The hyperparameters we optimized and the corresponding best values on each split of the three datasets are shown in Table S3-5 respectively. In the outer loop, with the hyperparameters achieving the best performance, i.e. the lowest RMSE, in the inner loop, a model is then trained on the full training set and eventually used for the prediction of validation set. According to the original reports[1, 3, 11], RMSE and R^2 are used as evaluation metrics for Dataset A and B while MAE and R^2 are used as evaluation metrics for Dataset C. It should be noted that we didn't optimize the number of epoch and learning rate which were chosen by experience. The corresponding codes for hyperparameters optimization are provided in the GitHub repo (https://github.com/bm2-lab/DeepReac) in case users want to adjust the hyperparameter set to be optimized.

The model is initialized using Glorot initialization[12] and trained to minimize MSE loss using the Adam SGD optimizer[13] with an initial learning rate of 0.001 for 1000 epochs. Our architecture was built with Pytorch and trained on Nvidia GeForce RTX 1080 Ti GPUs.

Table S3 Overview of hyperparameters optimized in *DeepReac* model on Dataset A

Hyperparameter		**Range**	**Best Values**				
			CV-split 1	**CV-split 2**	**CV-split 3**	**CV-split 4**	**CV-split 5**
Molecule GAT module	Dimensionality of hidden features	{16, 32}	32	32	32	32	32
	Number of GAT layers	{2, 3}	2	2	2	2	2
	Number of attention heads	{2, 4}	4	2	4	4	4
Reaction GAT module	Dimensionality of hidden features	{16, 32}	32	32	32	32	32
	Number of GAT layers	{2, 3}	2	2	2	2	2
	Number of attention heads	{2, 4}	4	2	4	4	4
Capsule module	Dimensionality of hidden features	{16, 32, 128}	32	32	32	32	32
Dropout probability		{0, 0.1, 0.5}	0	0	0	0	0

Table S4 Overview of hyperparameters optimized in *DeepReac* model on Dataset B

Hyperparameter		**Range**	**Best Values**				
			CV-split 1	**CV-split 2**	**CV-split 3**	**CV-split 4**	**CV-split 5**
Molecule GAT module	Dimensionality of hidden features	{16, 32}	32	32	32	32	32
	Number of GAT layers	{2, 3}	2	2	2	2	2
	Number of attention heads	{2, 4}	4	4	4	4	4
Reaction GAT module	Dimensionality of hidden features	{16, 32}	32	32	32	32	32
	Number of GAT layers	{2, 3}	2	2	2	2	2
	Number of attention heads	{2, 4}	4	4	4	4	4
Capsule module	Dimensionality of hidden features	{16, 32, 128}	32	32	32	32	32
Dropout probability		{0, 0.1, 0.5}	0	0	0	0	0

Table S5 Overview of hyperparameters optimized in *DeepReac* model on Dataset C

Hyperparameter		Range	Best Values				
			CV-split 1	**CV-split 2**	**CV-split 3**	**CV-split 4**	**CV-split 5**
Molecule GAT module	Dimensionality of hidden features	{16, 32}	16	16	16	32	16
	Number of GAT layers	{2, 3}	2	2	2	2	2
	Number of attention heads	{2, 4}	2	4	2	2	2
Reaction GAT module	Dimensionality of hidden features	{16, 32}	16	16	16	32	16
	Number of GAT layers	{2, 3}	2	2	2	2	2
	Number of attention heads	{2, 4}	2	4	2	2	2
Capsule module	Dimensionality of hidden features	{16, 32, 128}	128	32	32	32	128
Dropout probability		{0, 0.1, 0.5}	0.5	0.5	0.5	0.5	0.5

Design details of the active learning strategies in *DeepReac+*

1) Sampling strategies

The effect of active learning depends extremely on sampling strategy which decides what kind of samples can boost the predictive performance of model most. The essence of sampling strategy is a ranking algorithm: rank the unlabeled set based on certain measure and select top 10 as candidates. Besides random and greed strategies, all the other strategies involve the calculation of similarity between data points. The representation of experiment, a feature vector which can be simple one-hot encoding, customized chemical descriptors or task-related representation learnt automatically by *DeepReac*, is used to calculate the similarity. And the way of similarity calculation adopted here is cosine similarity. The details of five sampling strategies are shown in Table S6.

Table S6 Details of five sampling strategies

Sampling strategy	**Calculation steps**
Random	Select 10 candidates randomly from the unlabeled set.
Diversity	1, Find the most similar labeled data point of each unlabeled data point and record the similarity between them; 2, Sort the unlabeled set in ascending order according to the above similarity; 3, Select the top 10 as candidates.
Adversary	1, Find the most similar labeled data point of each unlabeled data point; 2, Calculate the difference between the predicted value of each unlabeled data point and the label value of corresponding labeled data point; 3, Sort the unlabeled set in descending order according to the above difference; 4, Select the top 10 as candidates.
Greed	1, Sort the unlabeled set in descending order according to their predicted values; 2, Select the top 10 as candidates.
Balance	1, Select 5 candidates according to greed strategy; 2, Select 5 candidates according to adversary strategy.

2) Simulation procedure

In order to validate the active learning loop and test the effect of different sampling strategies, simulations under various scenarios were performed. Since the size of training set is too small in the early stage of simulation and repetitive retaining is performed, some training settings here are slightly different from the above-mentioned ones. Because simulation involves dozens or hundreds of rounds of retraining, it will consume huge amount of time if hyperparameter optimization is performed during each iteration. It won't be a problem in real-world applications, so we cancel the step of hyperparameter optimization for brevity. Considering the results of hyperparameter optimization on the three datasets, we chose the hyperparameters achieving the best performance in most of splits on the three datasets to do simulations for consistency, which is {*Molecule GAT module*: Dimensionality of hidden features = 32; Number of GAT layers = 2; Number of attention heads = 4. *Reaction GAT module*: Dimensionality of hidden features = 32; Number of GAT layers = 2; Number of attention heads = 4. *Capsule module*: Dimensionality of hidden features = 32. Dropout probability = 0}. The model is trained for 100 epochs in each iteration and early stopping technique[14] is used to avoid overfitting which is likely to be caused by too small amount of training data. A fifth of training set, which is called as labeled set in the procedures described below, is treated as test set. During training, predictive error of the model on the test set is monitored and the training is stopped when the test error does not improve for 10 consecutive epochs. The detailed procedures under different scenarios are as following:

a) For library mode (related to Figure 2-3 & S1-6, 11):
1, 10% of dataset are chosen randomly to work as labeled set and remaining 90% is unlabeled set which means their labels can't "seen". The former is then used to train an initial model;
2, Unlabeled set are predicted by the trained model and the corresponding RMSE or MAE is recorded according to different predictive tasks;
3, With certain sampling strategy, 10 (or 50, 96 for Figure S1) candidates are removed from unlabeled set and added to labeled set;
4, The updated labeled set is used to retrain the model;
5, Repeat 2-5 steps until labeled set exceeds 90% of dataset;
6, The above procedure is repeated 30 times for each sampling strategy.

b) For catalyst-unknown mode (related to Figure 4 & S7-9):
1, Each dataset is split to Group 1-5 according to reference compounds (Table S1). One of them is treated as validation set which can't be sampled during simulation. 10% of the remaining data are chosen randomly to work as labeled set and 90% is unlabeled set which means their labels can't be "seen". The former is then used to train an initial model;
2, Unlabeled set and validation set are predicted by the trained model and the corresponding RMSE or MAE of latter is recorded according to different predictive tasks;
3, With certain sampling strategy, 10 candidates are removed from unlabeled set and added to labeled set;
4, The updated labeled set is used to retrain the model;
5, Repeat 2-5 steps until labeled set exceeds 90% of dataset;
6, The above procedure is repeated 30 times for each sampling strategy.

c) For identification of the optimal reaction conditions on Dataset A (related to Figure 5A-E & S10A-E):
1, The dataset is split to Set 1-5 according to reference compounds (Table S2). One of them is chosen as labeled set and another one is chosen as unlabeled set which means their labels can't be "seen". The former is then used to train an initial model;
2, Unlabeled set are predicted by the trained model and the corresponding RMSE is recorded;
3, With certain sampling strategy, 10 candidates are removed from unlabeled set and added to labeled set. The labels of them are also recorded;
4, The updated labeled set is used to retrain the model;
5, Repeat 2-5 steps until labeled set exceeds 90% of dataset;
6, The above procedure is repeated on the remaining subsets for each sampling strategy.

d) For identification of the optimal reaction conditions on Dataset B (related to Figure 5F & S10F):
1, 96 data points are chosen randomly to work as labeled set and the remaining data is unlabeled set which means their labels can't be "seen". The former is then used to train an initial model;
2, Unlabeled set are predicted by the trained model and the corresponding RMSE is recorded;
3, With certain sampling strategy, 96 candidates are removed from unlabeled set and added to labeled set. The labels of them are also recorded;
4, The updated labeled set is used to retrain the model;

5, Repeat 2-5 steps until labeled set exceeds 90% of dataset;
6, The above procedure is repeated 30 times for each sampling strategy.

Supplementary figures & tables

Table S1 Details of data splitting of three datasets for simulation under catalyst-unknown mode

Groups		Reference compounds
Dataset A	Group 1	t-BuBrettPhos
	Group 2	t-BuXPhos
	Group 3	AdBrettPhos
	Group 4	XPhos
Dataset B	Group 1	CataCXium A, P(Cy)$_3$, XPhos
	Group 2	dppf, dtbpf, Xantphos
	Group 3	SPhos, P(o-Tol)$_3$, None
	Group 4	P(tBu)$_3$, AmPhos, P(Ph)$_3$
Dataset C*	Group 1	245_i, 202_vi, 166_i, 251_vi, 182_i, 5_i, 157_i, 73_i, 87_i, 365_i, 181_i
	Group 2	147_i, 365_vi, 99_vi, 229_i, 223_i, 262_vi, 29_i, 7_i, 61_i, 249_i, 246_vi
	Group 3	242_i, 382_i, 71_vi, 72_i, 286_vi, 76_vi, 328_vi, 144_i, 229_vi, 99_i, 276_i
	Group 4	242_vi, 253_i, 1_i, 371_i, 205_vi, 207_i, 145_i, 210_i, 230_i, 245_vi

*The code of reference compounds is the same as that in the original work.

Table S2 Details of data splitting of Dataset A for identification of the optimal reaction conditions

Sets	Reference compounds
Set 1	1-chloro-4-(trifluoromethyl) benzene, 1-bromo-4-(trifluoromethyl) benzene, 1-iodo-4-(trifluoromethyl) benzene
Set 2	1-chloro-4-methoxybenzene, 1-bromo-4-methoxybenzene, 1-iodo-4-methoxybenzene
Set 3	1-chloro-4-ethylbenzene, 1-bromo-4-ethylbenzene, 1-ethyl-4-iodobenzene
Set 4	2-chloropyridine, 2-bromopyridine, 2-iodopyridine
Set 5	3-chloropyridine, 3-bromopyridine, 3-iodopyridine

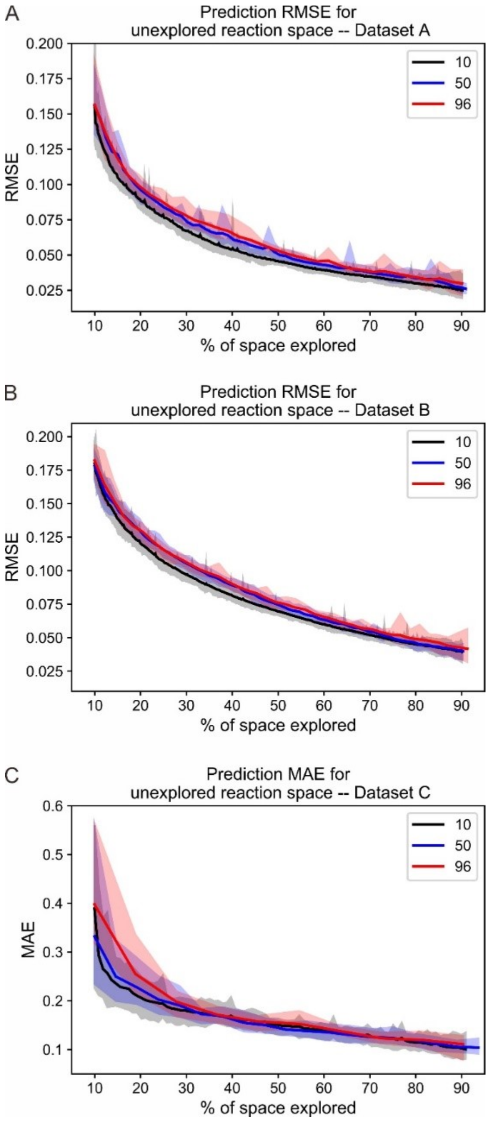

Figure S1. Simulation results of adversary-based strategy with *DeepReac* on three benchmark datasets using different numbers of candidates each iteration.

Aggregated results from 30 simulations show the average RMSE/MAE of *DeepReac* on Dataset A (A), Dataset B (B) and Dataset C (C) versus the fraction of the chemical space explored; the filled areas around the curves are defined by the maximum and minimum values. The black line indicates 10 candidates each iteration, the blue line indicates 50 candidates each iteration, and the red line indicates 95 candidates each iteration.

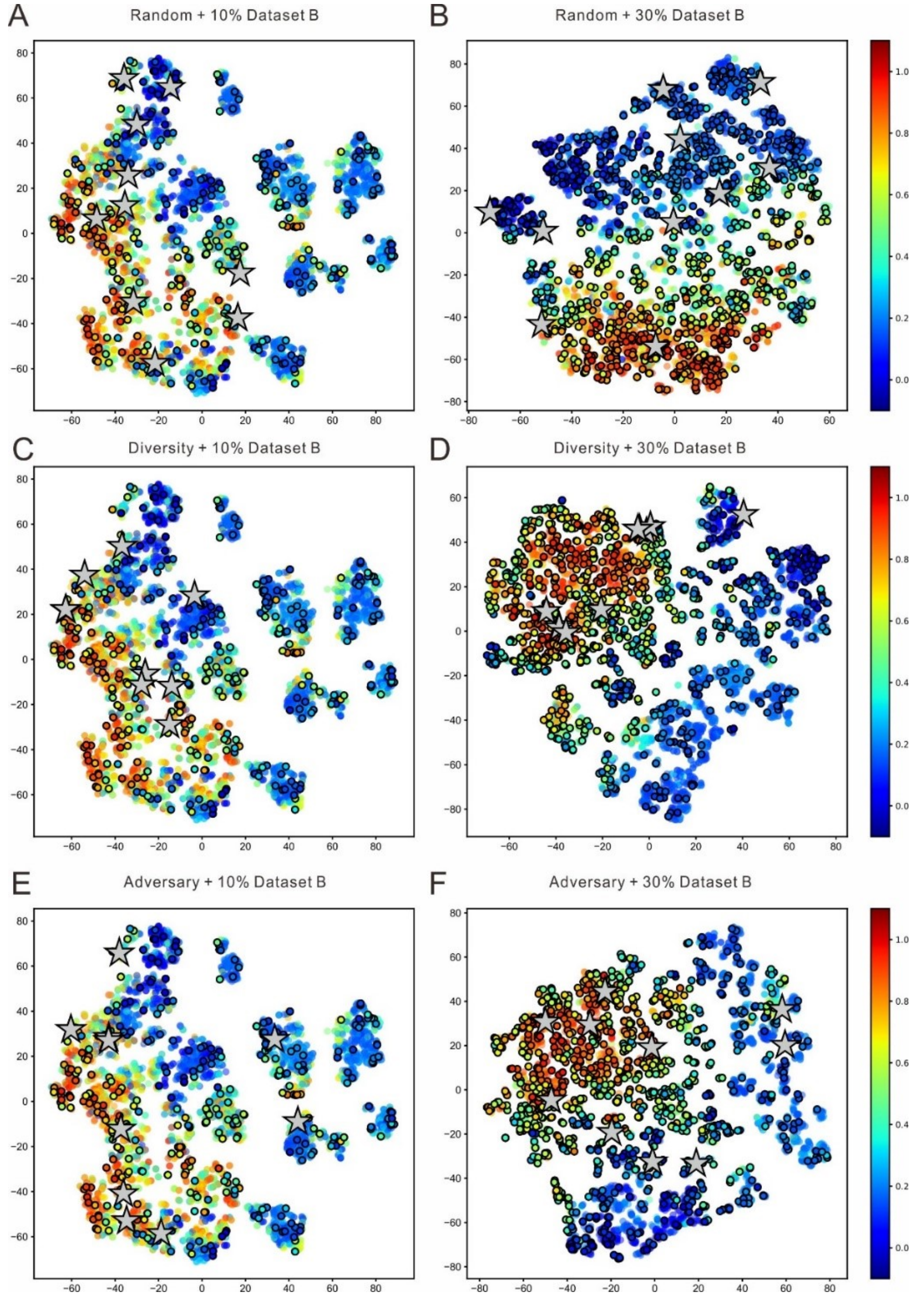
A
Random + 10% Dataset B
B
Random + 30% Dataset B
C
Diversity + 10% Dataset B
D
Diversity + 30% Dataset B
E
Adversary + 10% Dataset B
F
Adversary + 30% Dataset B
80
60
40
20
0
−20
−40
−60
−80
−60
−40
−20
0
20
40
60
80
1.0
0.8
0.6
0.4
0.2
0.0

Figure S2. Visualization of presentation-learning of *DeepReac* model with three sampling strategies on Dataset B, related to Figure 3.

t-Distributed stochastic neighbor embedding (t-SNE) of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 10% data. Data points are colored by true yield and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (A), diversity-based sampling strategy (C) and adversary-based sampling strategy (E) are marked as grey stars.t-SNE of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 30% data under active learning setting. Data points are colored by true yield and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (B), diversity-based sampling strategy (D) and adversary-based sampling strategy (F) are marked as grey stars.

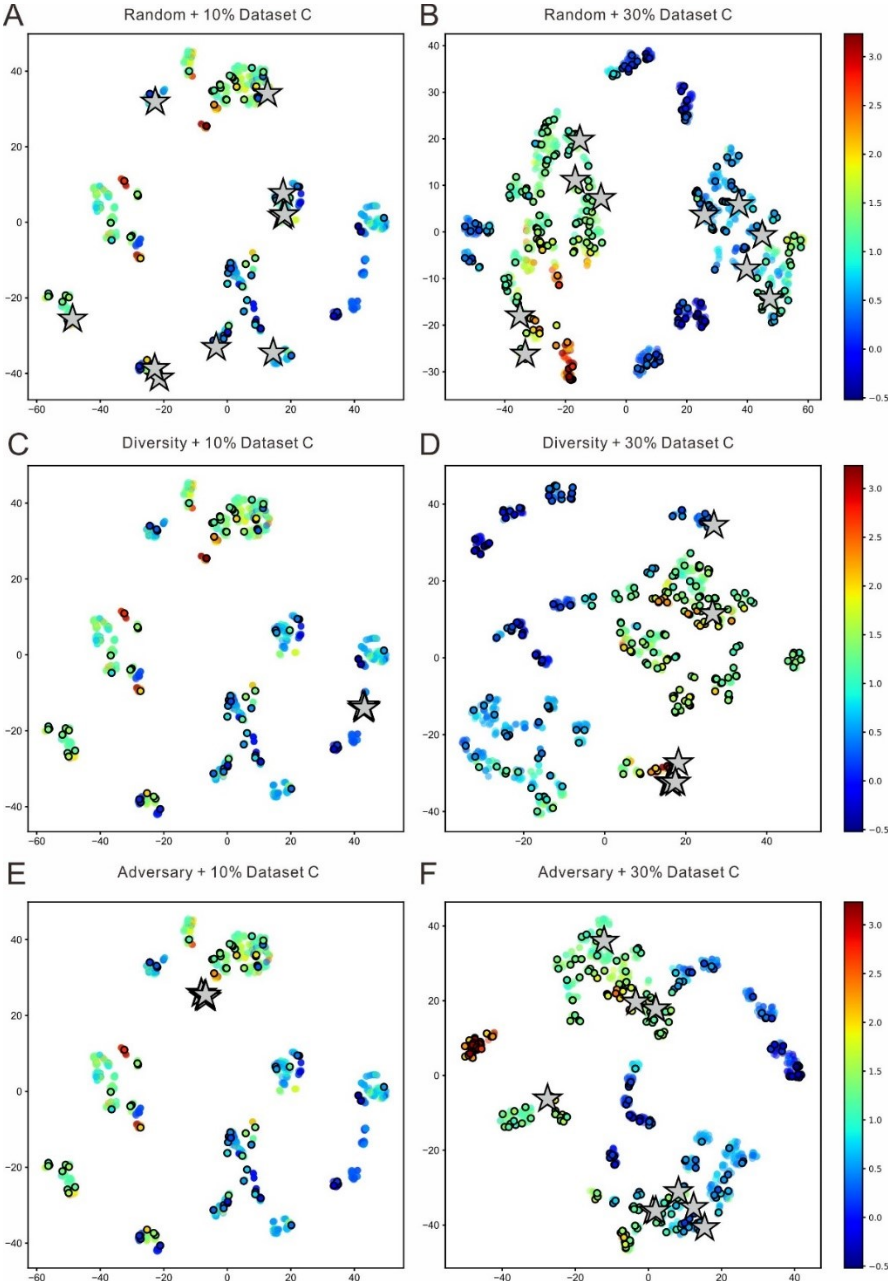

Figure S3. Visualization of presentation-learning of *DeepReac* model with three sampling strategies on Dataset C, related to Figure 3.

t-Distributed stochastic neighbor embedding (t-SNE) of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 10% data. Data points are colored by true △△G and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (A), diversity-based sampling strategy (C) and adversary-based sampling strategy (E) are marked as grey stars. t-SNE of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 30% data under active learning setting. Data points are colored by true yield and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (B), diversity-based sampling strategy (D) and adversary-based sampling strategy (F) are marked as grey stars.

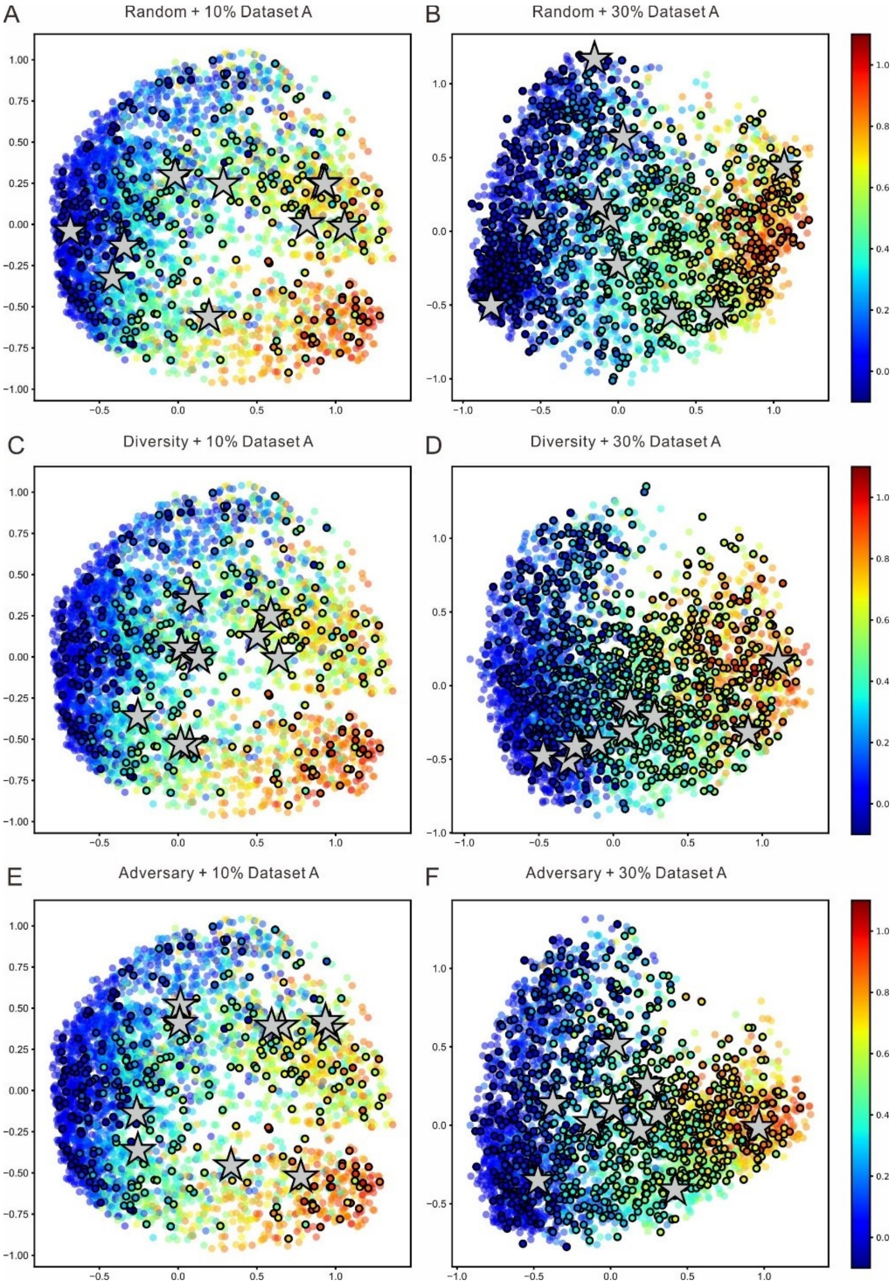

Figure S4. Visualization of presentation-learning of *DeepReac* model with three sampling strategies on Dataset A, related to Figure 3.

Principal components analysis (PCA) plots of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 10% data. Data points are colored by true ΔΔG and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (A), diversity-based sampling strategy (C) and adversary-based sampling strategy (E) are marked as grey stars. PCA plots of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 30% data under active learning setting. Data points are colored by true yield and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (B), diversity-based sampling strategy (D) and adversary-based sampling strategy (F) are marked as grey stars.

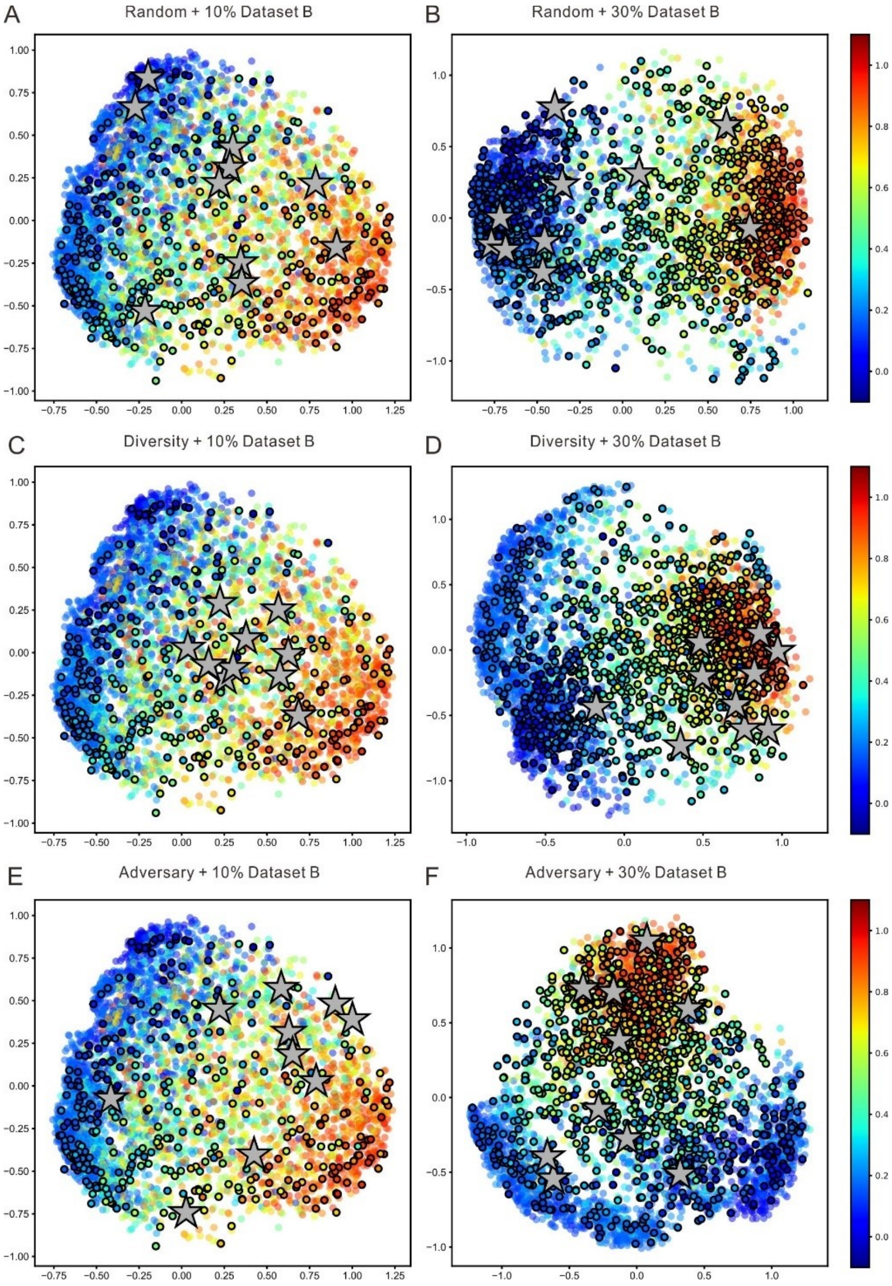

Figure S5. Visualization of presentation-learning of *DeepReac* model with three sampling strategies on Dataset B, related to Figure 3.

Principal components analysis (PCA) plots of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 10% data. Data points are colored by true ΔΔG and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (A), diversity-based sampling strategy (C) and adversary-based sampling strategy (E) are marked as grey stars. PCA plots of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 30% data under active learning setting. Data points are colored by true yield and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (B), diversity-based sampling strategy (D) and adversary-based sampling strategy (F) are marked as grey stars.

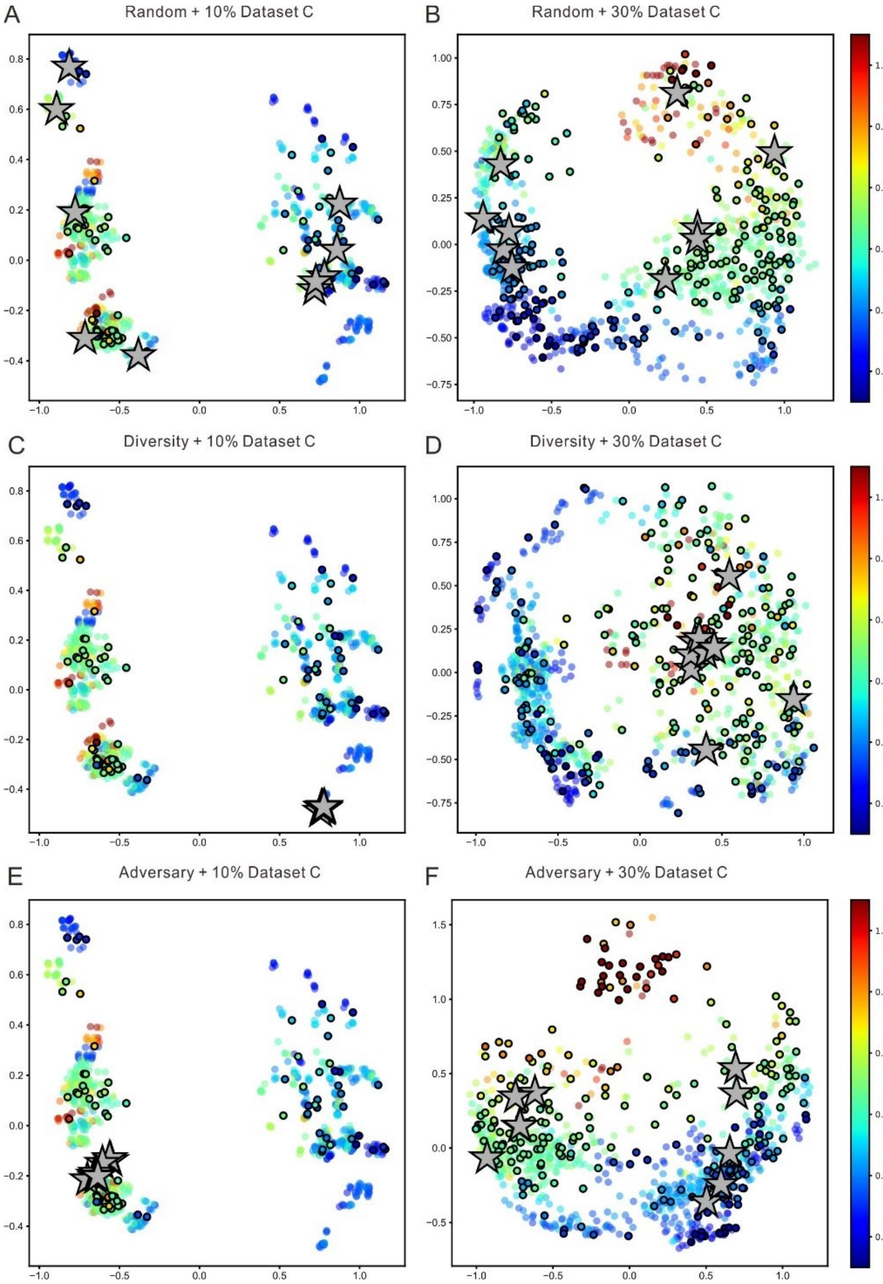

Figure S6. Visualization of presentation-learning of *DeepReac* model with three sampling strategies on Dataset C, related to Figure 3.

Principal components analysis (PCA) plots of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 10% data. Data points are colored by true ΔΔG and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (A), diversity-based sampling strategy (C) and adversary-based sampling strategy (E) are marked as grey stars. PCA plots of all reaction presentations outputted by *Capsule module* in *DeepReac* model trained with 30% data under active learning setting. Data points are colored by true yield and the labeled ones are indicated by black edge. Candidates selected by random sampling strategy (B), diversity-based sampling strategy (D) and adversary-based sampling strategy (F) are marked as grey stars.

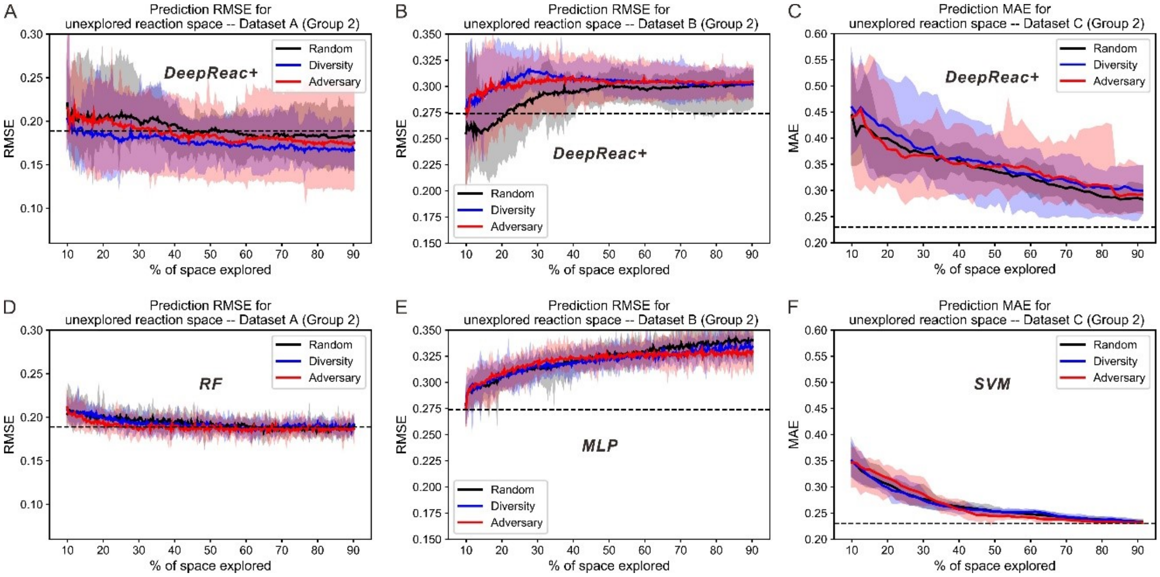

Figure S7. Simulation results of three sampling strategies with *DeepReac* and other models on Group 2 of three benchmark datasets under catalyst-unknown mode, related to Figure 4.

Aggregated results from 30 simulations show the average RMSE/MAE of *DeepReac* on Dataset A (A), Dataset B (B) and Dataset C (C) versus the fraction of the chemical space explored; the filled areas around the curves are defined by the maximum and minimum values. The black line indicates the random sampling strategy, the blue line indicates the diversity-based sampling strategy, and the red line indicates the adversary-based sampling strategy. The horizontal dashed black line indicates the best model performance achieved using all training data without active learning. Aggregated results from 30 simulations showing the average RMSE/MAE of RF/MLP/SVM on Dataset A (D), Dataset B (E) and Dataset C (F) versus the fraction of the chemical space explored; the filled areas around the curves are defined by the maximum and minimum values. The black line indicates the random sampling strategy, the blue line indicates the diversity-based sampling strategy, and the red line indicates the adversary-based sampling strategy. The horizontal dashed black line indicates the best model performance achieved using all training data without active learning. RMSE, root-mean-square error. MAE, mean absolute error, in kcal/mol. RF, random forest. MLP, multilayer perceptron. SVM, support vector machine.

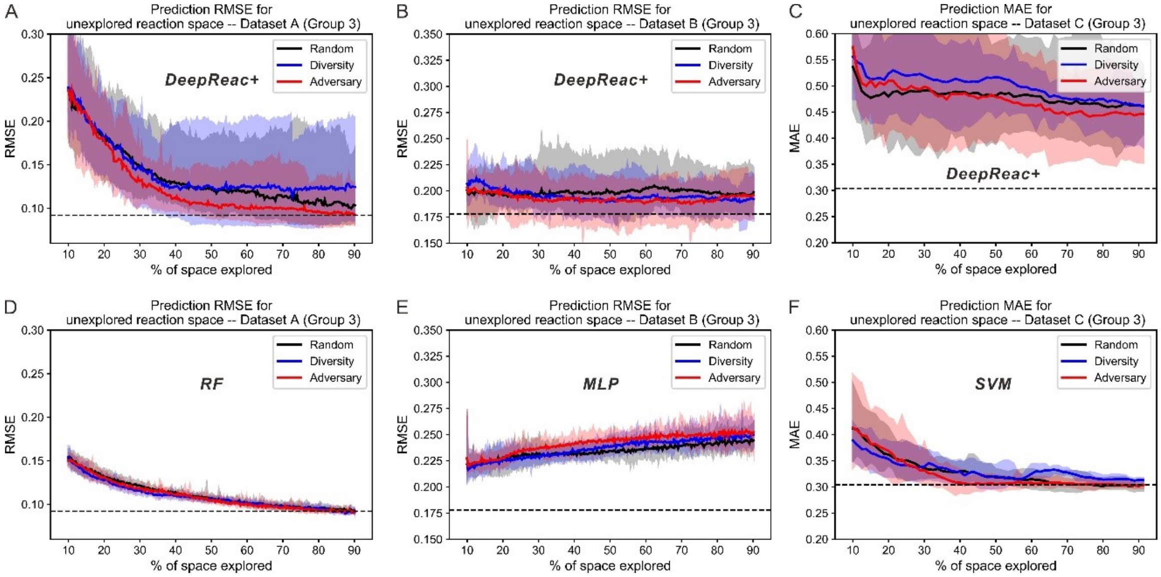

Figure S8. Simulation results of three sampling strategies with *DeepReac* and other models on Group 3 of three benchmark datasets under catalyst-unknown mode, related to Figure 4.

Aggregated results from 30 simulations show the average RMSE/MAE of *DeepReac* on Dataset A (A), Dataset B (B) and Dataset C (C) versus the fraction of the chemical space explored; the filled areas around the curves are defined by the maximum and minimum values. The black line indicates the random sampling strategy, the blue line indicates the diversity-based sampling strategy, and the red line indicates the adversary-based sampling strategy. The horizontal dashed black line indicates the best model performance achieved using all training data without active learning. Aggregated results from 30 simulations showing the average RMSE/MAE of RF/MLP/SVM on Dataset A (D), Dataset B (E) and Dataset C (F) versus the fraction of the chemical space explored; the filled areas around the curves are defined by the maximum and minimum values. The black line indicates the random sampling strategy, the blue line indicates the diversity-based sampling strategy, and the red line indicates the adversary-based sampling strategy. The horizontal dashed black line indicates the best model performance achieved using all training data without active learning. RMSE, root-mean-square error. MAE, mean absolute error, in kcal/mol. RF, random forest. MLP, multilayer perceptron. SVM, support vector machine.

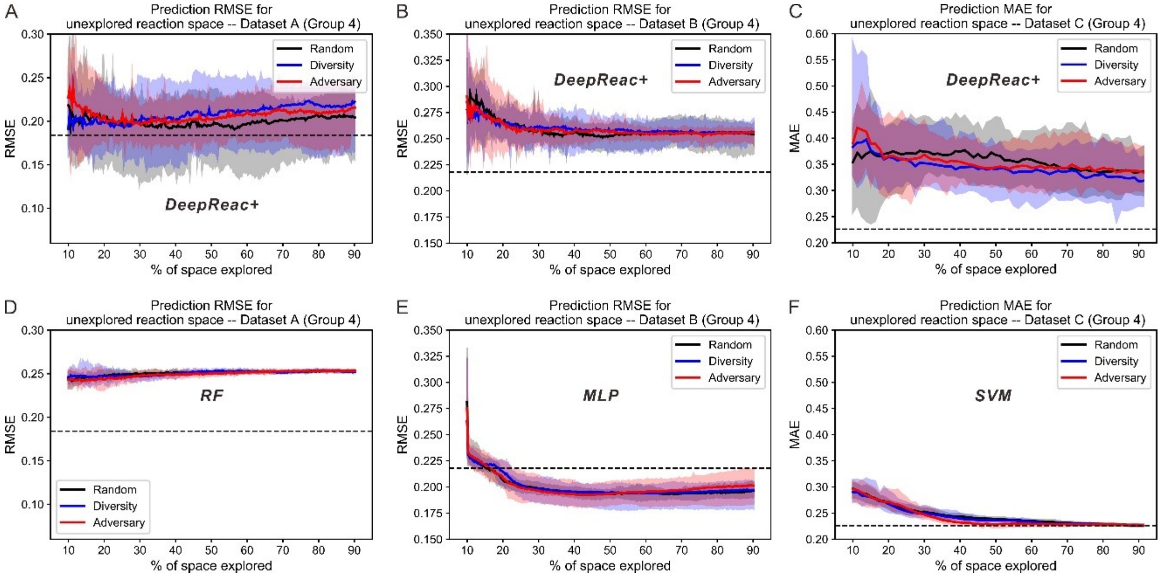

Figure S9. Simulation results of three sampling strategies with *DeepReac* and other models on Group 4 of three benchmark datasets under catalyst-unknown mode, related to Figure 4.

Aggregated results from 30 simulations show the average RMSE/MAE of *DeepReac* on Dataset A (A), Dataset B (B) and Dataset C (C) versus the fraction of the chemical space explored; the filled areas around the curves are defined by the maximum and minimum values. The black line indicates the random sampling strategy, the blue line indicates the diversity-based sampling strategy, and the red line indicates the adversary-based sampling strategy. The horizontal dashed black line indicates the best model performance achieved using all training data without active learning. Aggregated results from 30 simulations showing the average RMSE/MAE of RF/MLP/SVM on Dataset A (D), Dataset B (E) and Dataset C (F) versus the fraction of the chemical space explored; the filled areas around the curves are defined by the maximum and minimum values. The black line indicates the random sampling strategy, the blue line indicates the diversity-based sampling strategy, and the red line indicates the adversary-based sampling strategy. The horizontal dashed black line indicates the best model performance achieved using all training data without active learning. RMSE, root-mean-square error. MAE, mean absolute error, in kcal/mol. RF, random forest. MLP, multilayer perceptron. SVM, support vector machine.

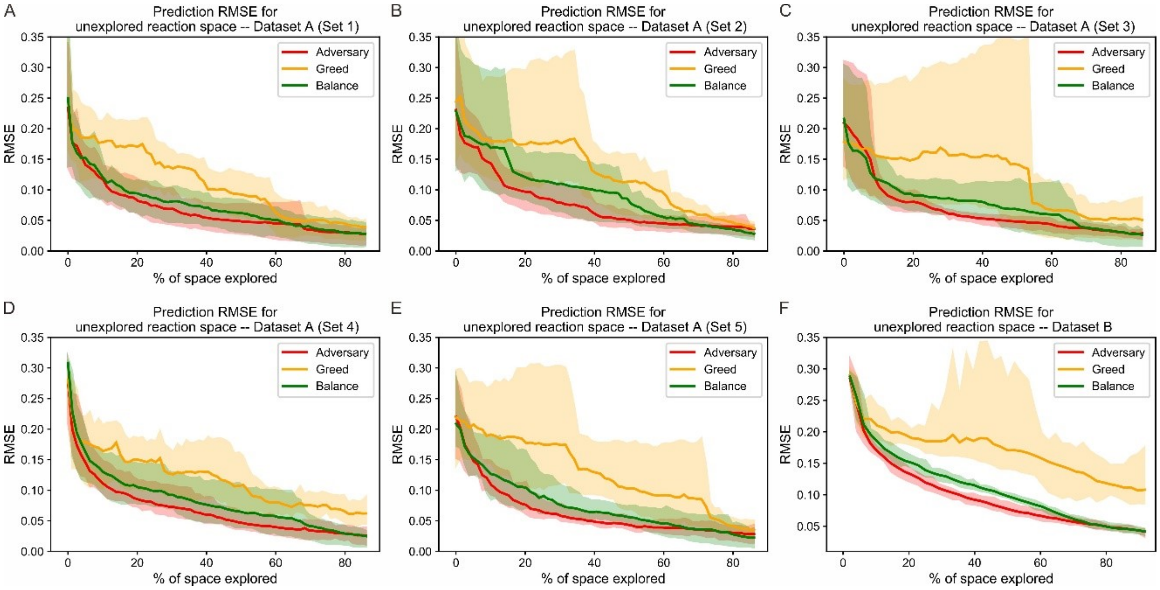

Figure S10. Simulation results of three sampling strategies with *DeepReac* on Dataset A & B, related to Figure 5.

Aggregated results from 30 simulations showing the average RMSE of *DeepReac* on 5 subsets of Dataset A (A-E) and Dataset B (F) versus the fraction of chemical space explored; the filled areas around the curves are defined by the maximum and minimum values. Red line indicates adversary-based sampling strategy, yellow line indicates greed sampling strategy and green line indicates balance sampling strategy. RMSE, root-mean-square error.

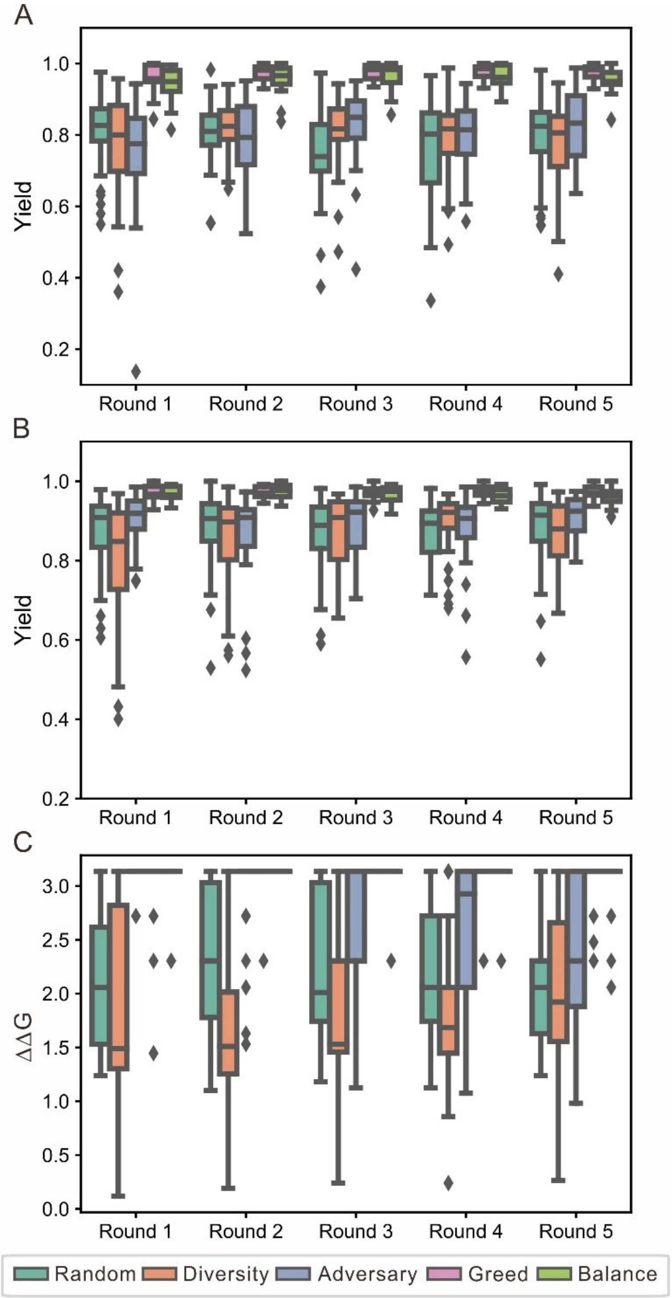

Figure S11. Statistical analysis of the ground-truth distribution of the candidates of five sampling strategies in the first 5 rounds of iteration on three benchmark datasets under library mode, related to Figure 5.

Each box plot shows the ground-truth distribution of the candidates of a certain sampling strategy at a round of iteration, which are colored according to the strategy type. The quartiles are shown, and the outliers are marked as square points. The results of the first 5 rounds of iteration are summarized for Dataset A (A), Dataset B (B) and Dataset C (C).

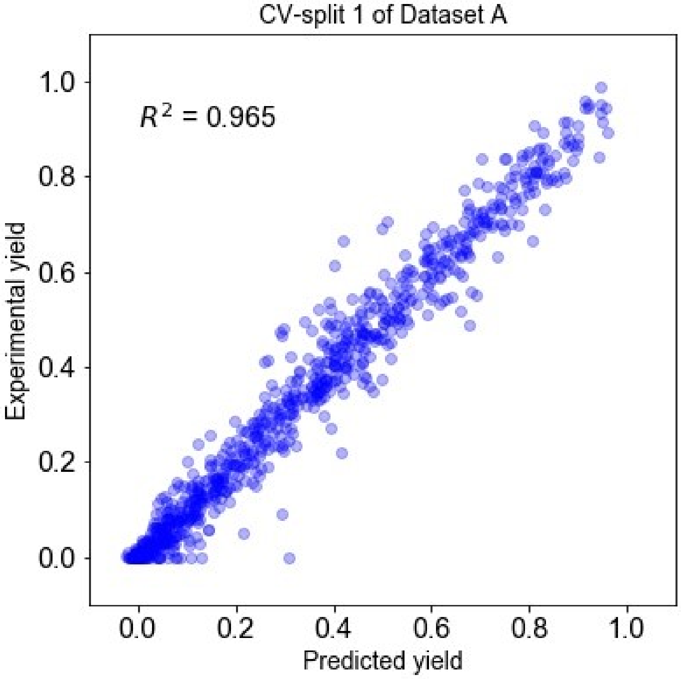

Figure S12. Plot for the prediction of CV-split 1 of Dataset A using the *DeepReac* model.

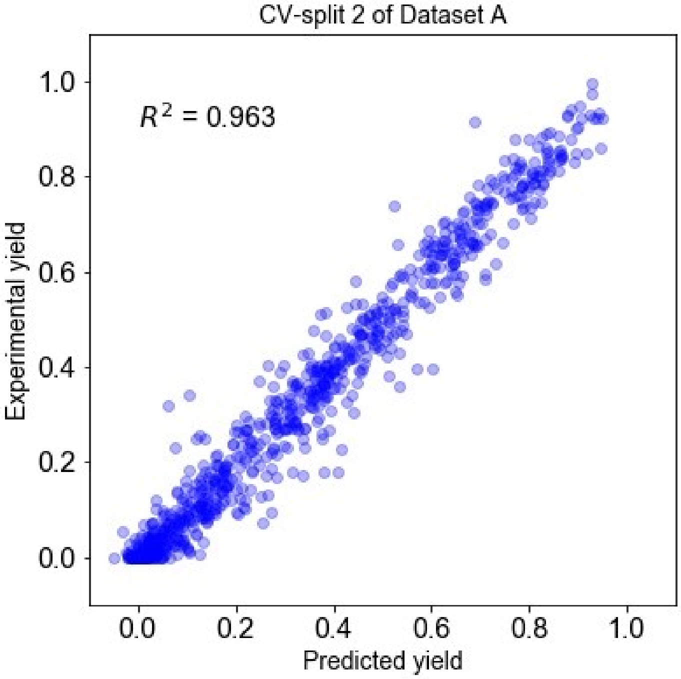

Figure S13. Plot for the prediction of CV-split 2 of Dataset A using the *DeepReac* model.

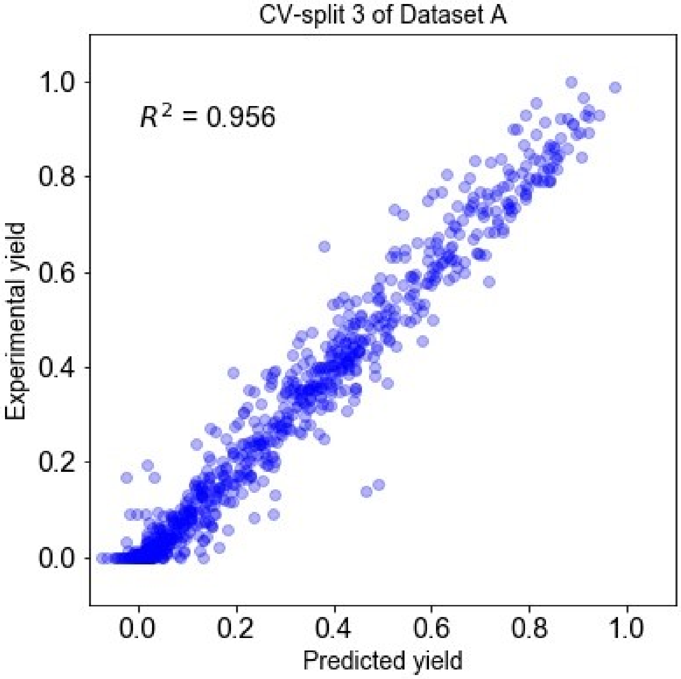

Figure S14. Plot for the prediction of CV-split 3 of Dataset A using the *DeepReac* model.

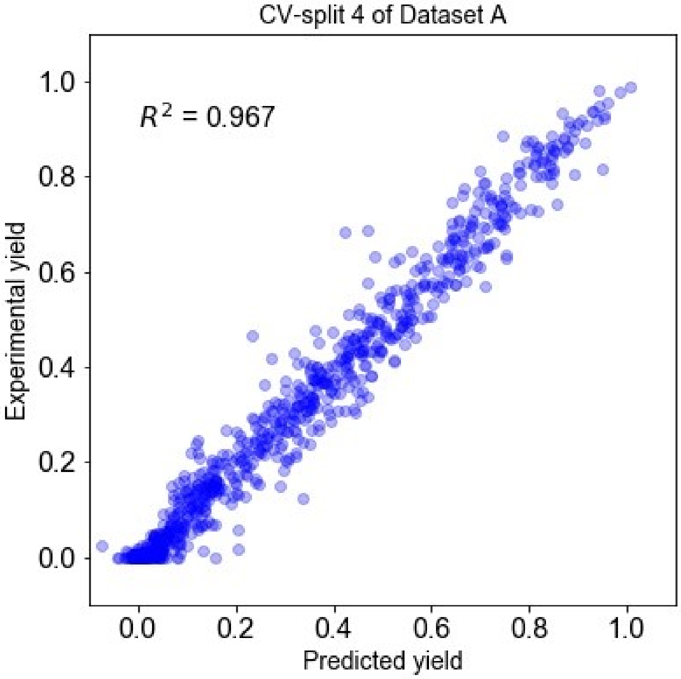

Figure S15. Plot for the prediction of CV-split 4 of Dataset A using the *DeepReac* model.

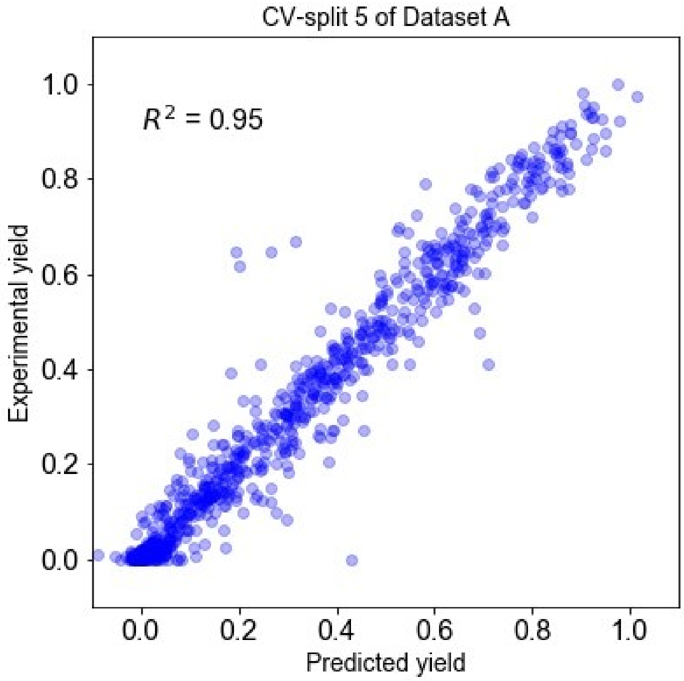

Figure S16. Plot for the prediction of CV-split 5 of Dataset A using the *DeepReac* model.

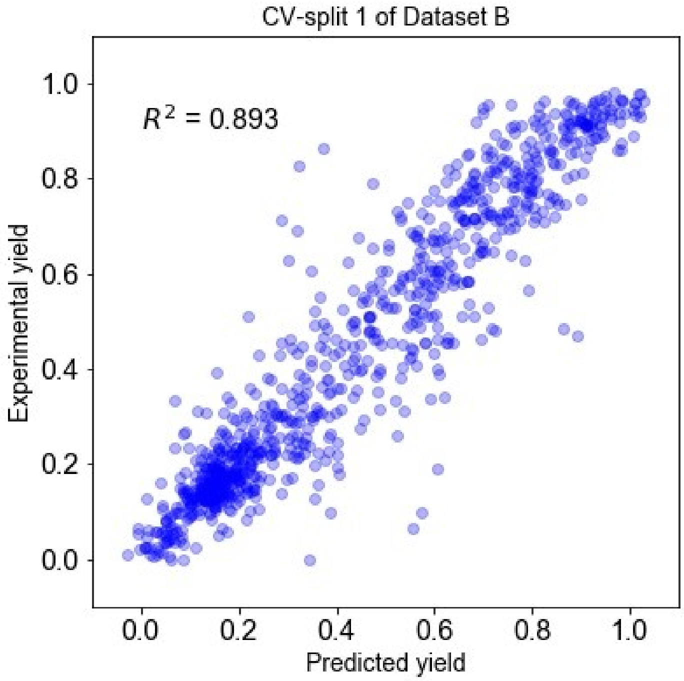

Figure S17. Plot for the prediction of CV-split 1 of Dataset B using the *DeepReac* model.

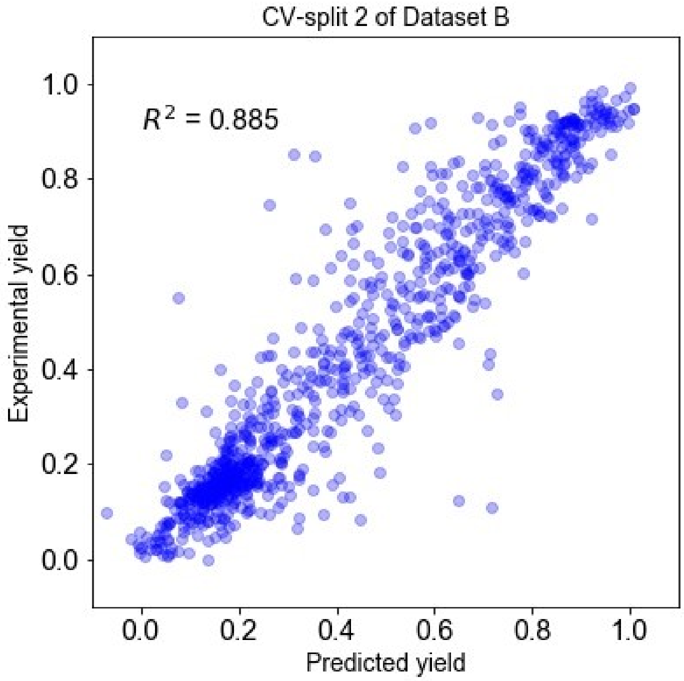

Figure S18. Plot for the prediction of CV-split 2 of Dataset B using the *DeepReac* model.

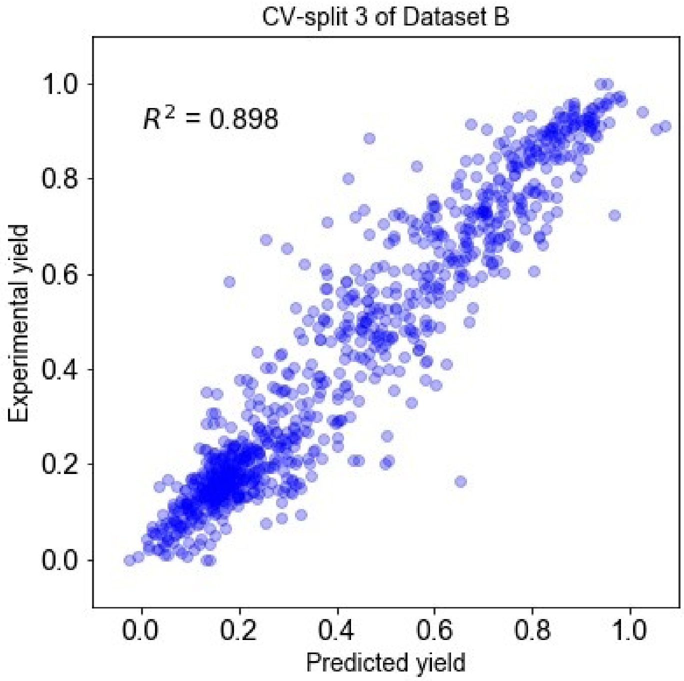

Figure S19. Plot for the prediction of CV-split 3 of Dataset B using the *DeepReac* model.

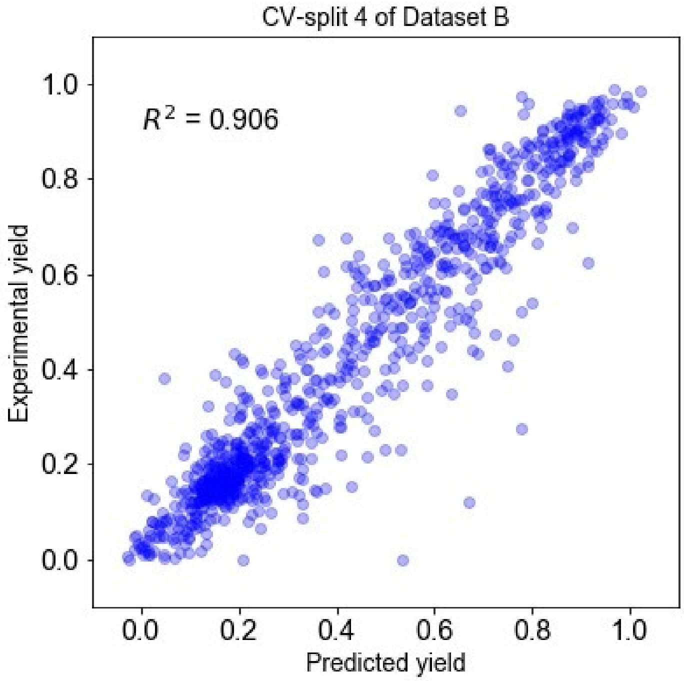

Figure S20. Plot for the prediction of CV-split 4 of Dataset B using the *DeepReac* model.

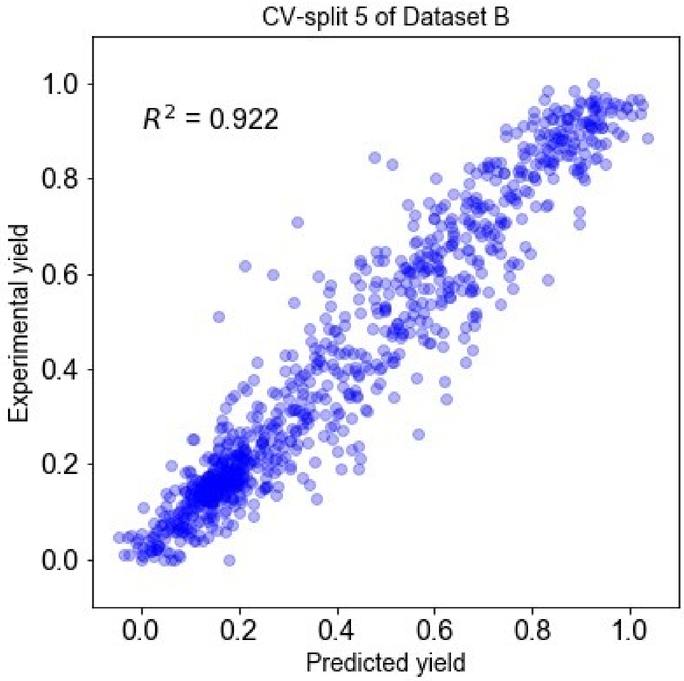

Figure S21. Plot for the prediction of CV-split 5 of Dataset B using the *DeepReac* model.

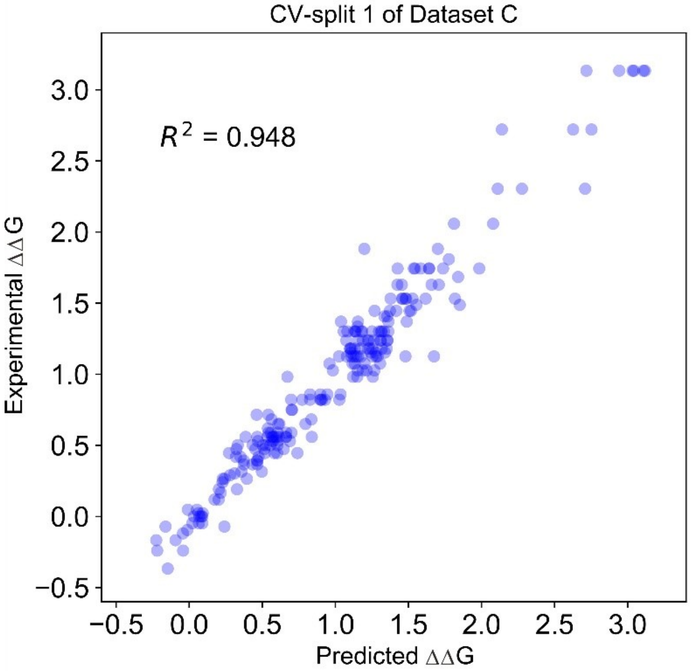

Figure S22. Plot for the prediction of CV-split 1 of Dataset C using the *DeepReac* model.

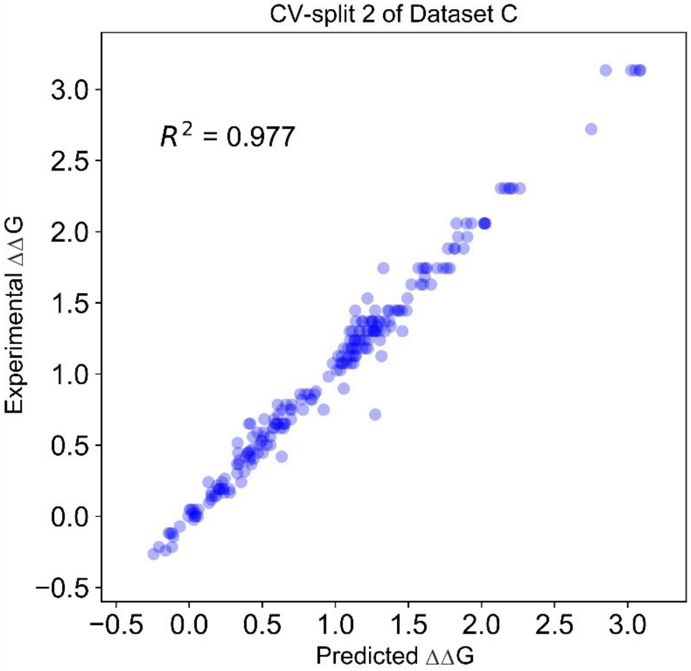

Figure S23. Plot for the prediction of CV-split 2 of Dataset C using the *DeepReac* model.

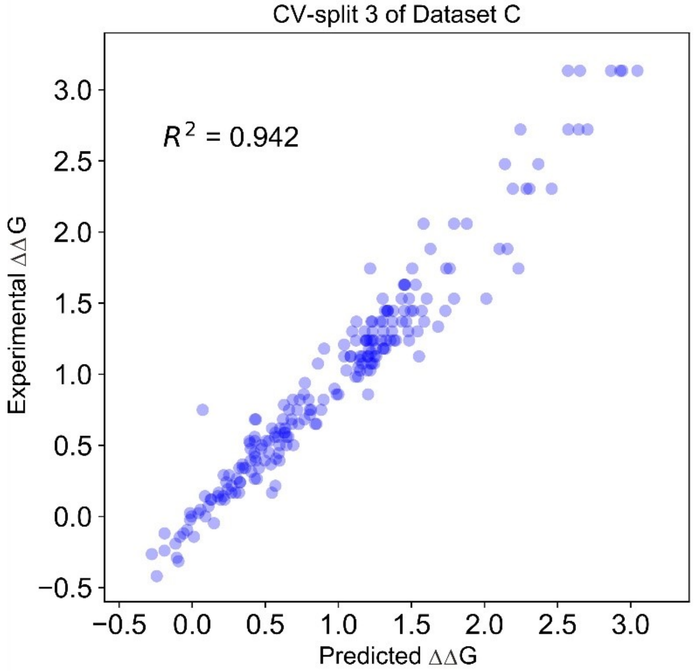

Figure S24. Plot for the prediction of CV-split 3 of Dataset C using the *DeepReac* model.

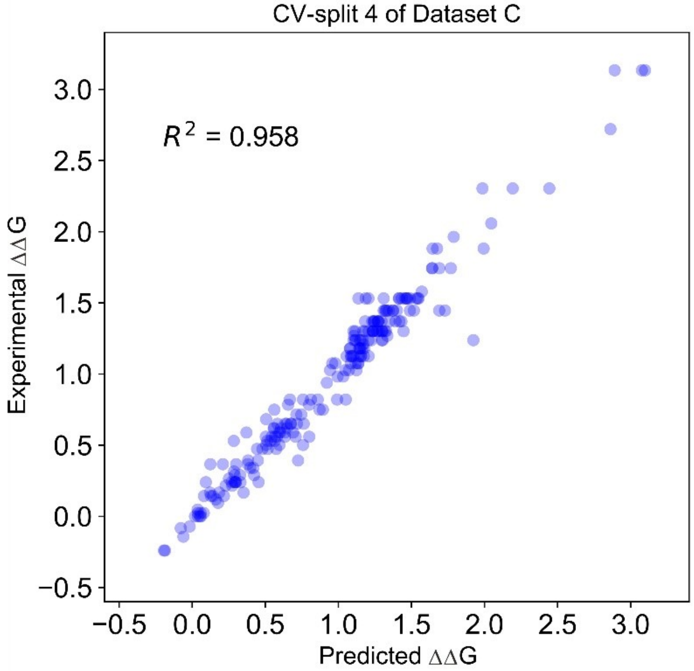

Figure S25. Plot for the prediction of CV-split 4 of Dataset C using the *DeepReac* model.

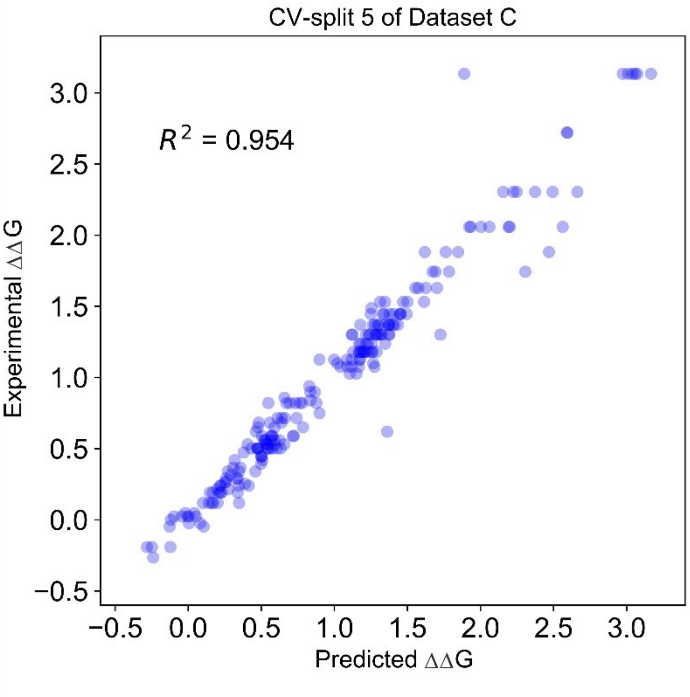

Figure S26. Plot for the prediction of CV-split 5 of Dataset C using the *DeepReac* model.

References

1. D. T. Ahneman, J. G. Estrada, S. Lin, S. D. Dreher and A. G. Doyle, *Science*, 2018, **360**, 186-190.
2. D. Perera, J. W. Tucker, S. Brahmbhatt, C. J. Helal, A. Chong, W. Farrell, P. Richardson and N. W. Sach, *Science*, 2018, **359**, 429-434.
3. A. F. Zahrt, J. J. Henle, B. T. Rose, Y. Wang, W. T. Darrow and S. E. Denmark, *Science*, 2019, **363**, eaau5631.
4. G. Landrum, RDKit: Open-source cheminformatics, https://rdkit.org/docs/index.html).
5. M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li and Z. Zhang, *Journal*, 2019, arXiv:1909.01315.
6. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, *Journal*, 2017, arXiv:1710.10903.
7. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, presented in part at the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017.
8. S. Sabour, N. Frosst and G. E Hinton, *Journal*, 2017, arXiv:1710.09829.
9. V. D. M. Laurens and G. Hinton, *Journal of Machine Learning Research*, 2008, **9**, 2579-2605.
10. A. Swami and R. Jain, *Journal of Machine Learning Research*, 2013, **12**, 2825-2830.
11. J. M. Granda, L. Donina, V. Dragone, D. L. Long and L. Cronin, *Nature*, 2018, **559**, 377-381.
12. X. Glorot and Y. Bengio, *Journal of Machine Learning Research*, 2010, **9**, 249-256.
13. D. P. Kingma and J. Ba, *Journal*, 2014, arXiv:1412.6980.
14. N. Morgan and H. Bourlard, *Generalization and parameter estimation in feedforward nets: some experiments*, Morgan Kaufmann Publishers Inc., 1990.