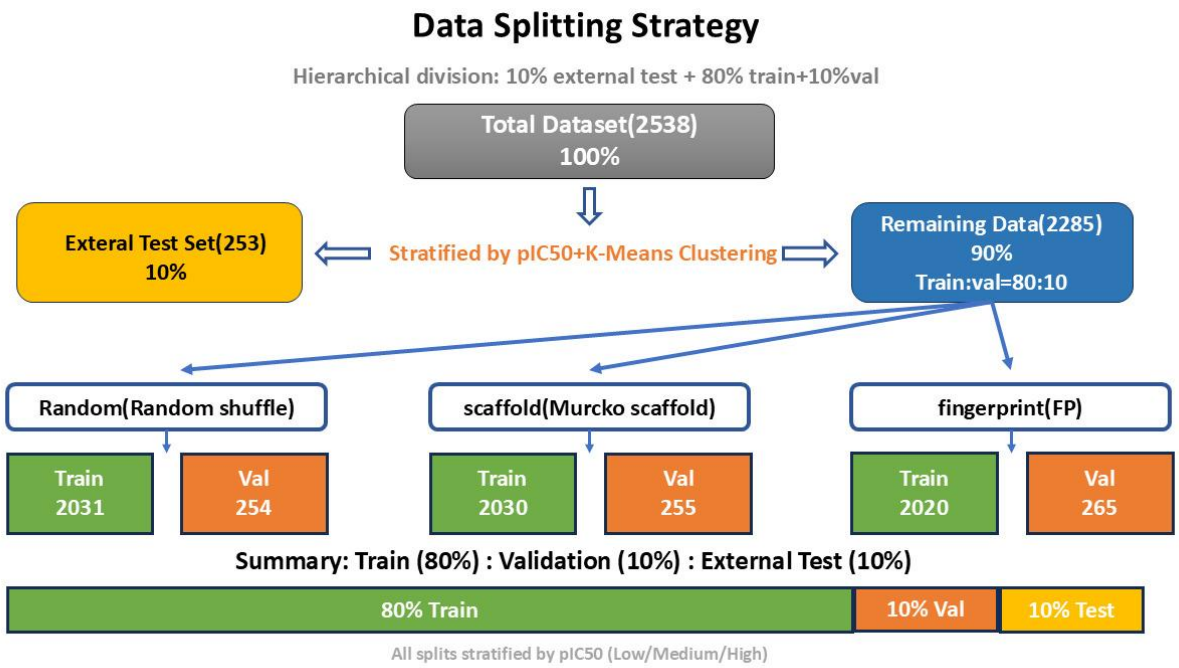
Supplementary Information

Figure S1. Data splitting strategies.



Data splitting was conducted on a curated dataset comprising 2538 compounds with associated pIC_{50} values. Four distinct splitting strategies were employed, as outlined below:

External Test Set Selection (Clustering-Based Stratified Extraction):

A 10% external test set was initially extracted from the complete dataset using a clustering-based methodology. The procedure entailed the following steps: - Molecular Fingerprint Generation: Morgan fingerprints (ECFP4, with a radius of 2 and 1024 bits) were computed for all compounds utilizing RDKit. - Initial K-Means Clustering: The fingerprint matrix was subjected to K-means clustering (with the number of clusters set to the maximum of 5 or n_samples/100, and a random state of 42). The number of clusters was automatically determined based on the dataset size. - Stratification Binning: Compounds were categorized into three activity strata (low, medium, high) according to their pIC_{50} values, using tertiles for stratification.

Cluster-Strata Merging: Clusters containing fewer than two compounds within any given activity stratum were consolidated into the largest cluster of the same stratum to ensure balanced representation. Iterative Stratified Sampling: For a maximum of 50 trials, a stratified random sample comprising 10% of the compounds was drawn, maintaining the distribution of both activity strata and cluster assignments. The Kolmogorov–Smirnov test ($p > 0.05$) was employed to confirm that the extracted external test set did not significantly differ from the remaining 90% of compounds in terms of pIC_{50}, molecular weight (MW), logP, and topological polar surface area (TPSA). The first successful trial, initiated with a seed value of 42 and incremented thereafter, was accepted.Output: The external test set (n = 253) was saved and excluded from all subsequent training and validation steps.

Random split

The remaining 90% of the compounds (n = 2285) were randomly divided into a training set, comprising 80% (n = 2,030), and a validation set, comprising 20% (n = 255), utilizing an 80/20 random partitioning strategy. To achieve a validation set that represents

20% of the original 90% subset, the test size was set to 0.111 relative to the remaining set. A random seed of 42 was employed to ensure reproducibility of the results.

Scaffold split

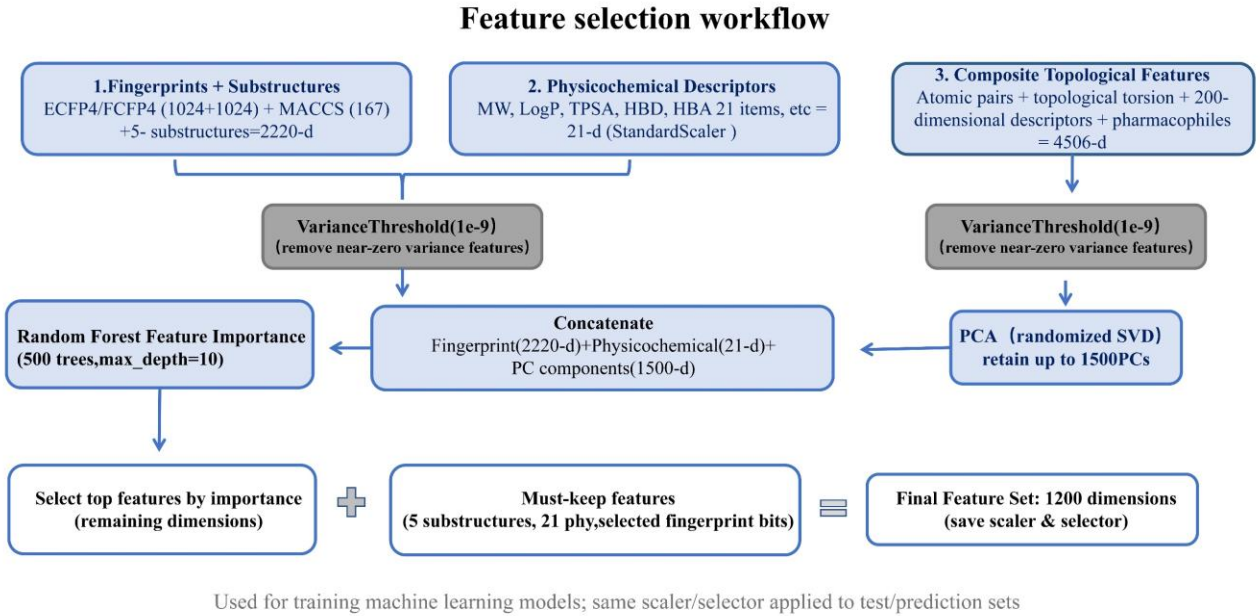
The compounds were categorized according to their Bemis-Murcko scaffolds to evaluate generalization to novel chemical cores. Scaffold generation was performed by computing the Murcko scaffold for each compound using RDKit's MurckoScaffold module. Subsequently, compounds sharing identical scaffolds were grouped together. These scaffold groups were then organized by size in descending order and sequentially allocated to the training set until it comprised approximately 80% of the remaining data (equivalent to 88.8% of the 90%, which corresponds to 80% of the original total). It was ensured that all compounds associated with a particular scaffold were entirely placed within either the training or validation set, avoiding any splitting. Finally, the training and validation sets were saved.

Fingerprint split

To prevent information leakage from structurally similar analogues, compounds were partitioned based on fingerprint similarity. Fingerprint generation involved computing Morgan fingerprints (ECFP4) for each compound, with a radius of 2 and a length of 1024 bits. For K-means clustering, the fingerprint matrix was organized into clusters, matching the number of clusters used for external test set extraction (n_clusters = max(5, n_samples/100), random_state = 42). Stratified sampling was then conducted within each cluster,

where compounds were randomly divided into training (80%) and validation (20%) sets using the train_test_split function (test_size = 0.111 relative to cluster size, random_state = 42). This approach ensured that structurally similar compounds remained within the same set. The final training set comprised all training samples from the clusters (n = 2030), while the validation set consisted of combined validation samples (n = 255). All data partitioning was executed using Python 3.10 with RDKit version 2025.09.01. Importantly, the external test set was excluded from model training and hyperparameter tuning. The scripts used for data splitting are available in the accompanying source code repository, as detailed in the Data Availability Statement.

Figure S2. Feature selection workflow



The feature selection flowchart elaborately demonstrates the process and methods of our feature screening. The final features used are the 1200-dimensional features that have been screened out. scripts are provided in the accompanying source code repository (see Data Availability Statement).

Table S1. 74-dimensional atomic features and 12-dimensional bond features used for graph neural network input.

Atomic features (74 dimensions)

Feature category	Description	Encoding	Dimensions
Atom type	One-hot encoding of element type (44 supported types: C, N, O, S, F, Si, P, Cl, Br, Mg, Na, Ca, Fe, As, Al, I, B, V, K, Tl, Yb, Sb, Sn, Ag, Pd, Co, Se, Ti, Zn, H, Li, Ge, Cu, Au, Ni, Cd, In, Mn, Zr, Cr, Pt, Hg, Pb)	One-hot	44
Atom degree	Degree of the atom (number of bonded neighbors, values 0–10)	One-hot	11
Number of implicit hydrogens	Implicit hydrogen count (0–6)	One-hot	7
Formal charge	Formal charge of the atom	Integer	1
Number of radical electrons	Radical electron count	Integer	1
Hybridization	Hybridization type (SP, SP^2, SP^3, SP^3D, SP^3D^2)	One-hot	5
Aromaticity	Whether the atom is part of an aromatic ring	Boolean	1
Total number of hydrogens	Total hydrogen count (implicit + explicit, 0–4)	One-hot	5

Total atomic features			74

Bond features (12 dimensions)

Feature category	Description	Encoding	Dimensions
Bond type	Bond order (SINGLE, DOUBLE, TRIPLE, AROMATIC)	One-hot	4
Conjugation	Whether the bond is conjugated	Boolean	1
In ring	Whether the bond belongs to a ring of any size	Boolean	1
Stereo configuration	Stereo configuration (STEREONONE, STEREOANY, STEREOZ,	One-hot	6
Total bond features			12

Atomic features are computed using RDKit's default atom featurizer (based on CanonicalAtomFeaturizer).Bond features are generated by CanonicalBondFeaturizer.All features are normalized or one-hot encoded as indicated.The total dimension of atomic features is 44 + 11 + 7 + 1 + 1 + 5 + 1 + 5 = 74; bond dimension is 4 + 1 + 1 + 6 = 12.For details, please refer to dgl-lifesci/python/dgllife/utils/featurizers.py at master · awslabs/dgl-lifesci · GitHub

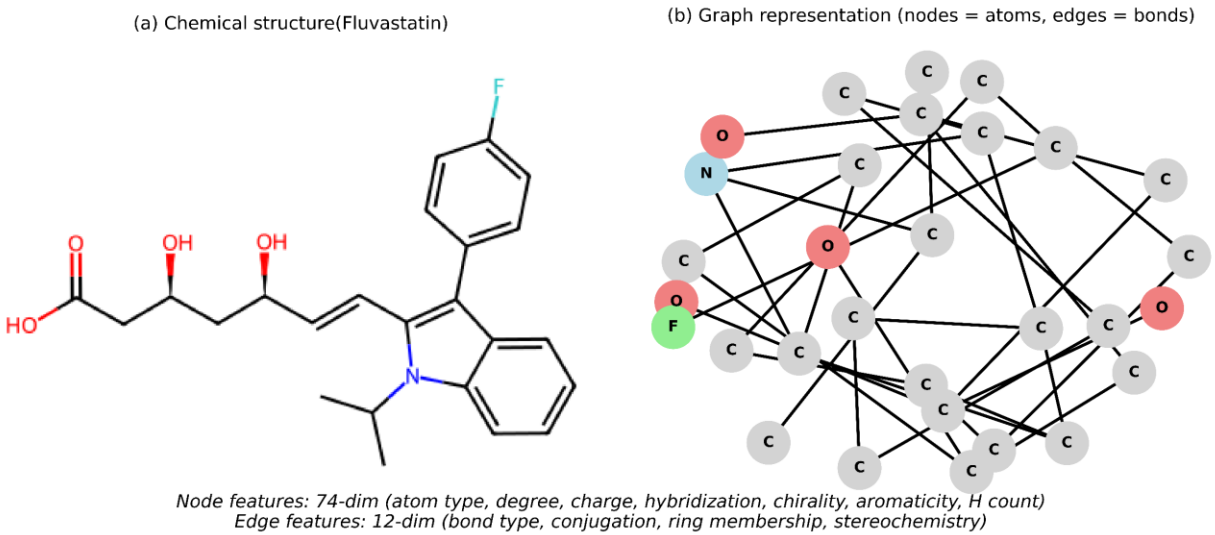
Table S2. Hyperparameter search spaces for the five machine learning models.

Model	Hyperparameter	Type	Search space / Values
LightGBM	n_estimators	Integer	[200, 3000]
	learning_rate	Float (log)	[0.001, 0.3]
	num_leaves	Integer	[20, 300] step 10
	max_depth	Integer	[3, 12]
	min_child_samples	Integer	[5, 100]
	subsample	Float	[0.5, 1.0]
	subsample_freq	Integer	[0, 5]
	colsample_bytree	Float	[0.5, 1.0]
	reg_alpha	Float	[0, 10]
	reg_lambda	Float	[0, 10]
	min_split_gain	Float	[0, 5]
NuSVR	nu	Float	[0.2, 0.8]
	C	Float (log)	[0.01, 100]
	gamma	Float (log)	[1e-4, 1.0]
	shrinking	Categorical	[True, False]
	tol	Float (log)	[1e-5, 1e-3]
GradientBoostin	n_estimators	Integer	[200, 2000]
	learning_rate	Float (log)	[0.005, 0.2]
	max_depth	Integer	[3, 10]
	min_samples_split	Integer	[5, 30]

	min_samples_leaf	Integer	[2, 15]
	subsample	Float	[0.6, 1.0]
	max_features	Categorical	["sqrt", "log2", 0.7, 0.9]
RandomForest	n_estimators	Integer	[100, 3000]
	max_depth	Integer	[5, 20]
	min_samples_split	Integer	[2, 20]
	min_samples_leaf	Integer	[1, 10]
	max_features	Categorical	["sqrt", "log2", 0.7, 0.9]
	bootstrap	Categorical	[True, False]
	oob_score	Conditional	[False, True] if bootstrap=True else
XGBoost	max_depth	Integer	[2, 14]
	learning_rate	Float (log)	[0.001, 0.3]
	n_estimators	Integer	[200, 3000]
	subsample	Float	[0.5, 1.0]
	colsample_bytree	Float	[0.5, 1.0]
	gamma	Float	[0, 10]
	min_child_weight	Integer	[1, 15]
	reg_alpha	Float	[0, 10]
	reg_lambda	Float	[0, 10]
	early_stopping_rounds	Fixed	80 (not searched)

“Float (log)” indicates the parameter was sampled on a logarithmic scale.All categorical parameters were chosen by trial.suggest_categorical().For RandomForest, oob_score is only considered when bootstrap=True; otherwise it is fixed to False.The number of Optuna trials for each model was 150 (not shown in this table).

Figure S3. Molecular graphs.



Molecular graph representation of fluvastatin as an example. Each node (atom) encodes 74 atomic features, and each edge (bond) encodes 12 bond features (see Table S1 for feature details). Graph neural networks (GCN, GAT, AttentiveFP) take such featurized graphs as input for pIC_{50} prediction.

Table S3. Architectural specifications of the three graph neural network models.

Model	Input features	Hidden size	Number of layers	Attention heads	Dropout	Activation	Output constraint	Edge feature usage	Pooling / Readout
GCN (simpleGCN)	74 (atom)	128	3 (layer-wise)	–	0.3	ReLU	Sigmoid	Not used	Global mean pooling (built
GAT (EnhancedGAT)	74 (atom)	128 per head (×8 heads)	3	8	0.3	ReLU	Sigmoid	Not used	Global mean pooling (built
AttentiveFP (EnhancedAFP)	74 (atom)	128 (graph-level)	3 (GAT layers) +	6 (num_timesteps)	0.3	ReLU (GAT) /	Sigmoid	Yes (edge features passed to	Attentive readout

All models were implemented using PyTorch and DGL Life, with a consistent random seed set to 42. The input node feature dimension (74) and edge feature dimension (12) were specified by the CanonicalAtomFeaturizer and CanonicalBondFeaturizer, respectively, as detailed in Table S1. The labels, represented as pIC_{50} values, were normalized to a [0,1] range using a fixed MinMax scaler with an original range of [3,11] prior to training. The final Sigmoid layer ensures that the outputs remain within this normalized range. It is noteworthy that the GCN and GAT models from dgllife do not inherently utilize edge features; only the AttentiveFP model incorporates bond information. Dropout was applied after each hidden layer, with a specified dropout rate of 0.3. For the GAT model, each attention head has a hidden dimension of 128, and the outputs from the 8 heads are concatenated, resulting in a 1024-dimensional representation before the final classification layer.

Methods S1. Dynamic Weighted Ensemble (DWE)

Overview

The DWE combines multiple base models (ML and GNNs) to predict pIC_{50}. For each compound, a sample-specific weight is assigned to each base model based on:

Cross-validation mean squared error (MSE) and prediction variance

Feature importance

Pairwise prediction correlations

Deviation of each model's prediction from the ensemble mean

A boost for high-activity compounds ($pIC_{50} \geq 8.0$)

All parameters, formulas, and steps are directly taken from the source code.

S2. Key Parameters (as defined in the code)

Parameter name in code	Value	Description
KFOLD_SPLITS	5	Number of cross-validation folds

RANDOM_SEED	42	Random seed for reproducibility
CONFIDENCE_POWER	2.5	Exponent for deviation penalty
UNCERTAINTY_WEIGHT	0.4	Weight for variance vs. MSE
CORRELATION_PENALTY	0.15	Penalty factor for correlation
ACTIVITY_THRESHOLD	8.0	pIC_{50} threshold for “high activity”
HIGH_ACTIVE_TOLERANCE	0.3	Allowed absolute error for high-active accuracy
HIGH_ACTIVE_WEIGHT_BOOST	1.4	Boost factor for high-active samples
PIC50_MIN, PIC50_MAX	3.0, 11.0	Clipping range for final predictions

Algorithm Steps (Exact as in the Code)

Step 1 – Training and cross-validation of base models

Each base model is trained on the full training set. Then, to estimate reliability, 5-fold cross-validation is performed on the training set:

For each fold, train on 4 folds, validate on the held-out fold.

Collect predictions for the validation fold.

After all folds, for each model compute:

MSE (errors[name]) = average squared error across all validation samples.

Variance (variances[name]) = variance of the predictions across validation samples.

Implementation detail:

For ML models (XGB, RF, GB, SVM), the model is cloned and retrained on each training fold (except LGB, which uses the pre-trained model directly).

For GNNs (AFP, GCN, GAT), the pre-trained model is used only for forward passes (no retraining during CV).

Step 2 – Feature importance

For tree-based models (XGB, RF, GB): use model.feature_importances_ (mean across features).

For LGB: use model.feature_importance(importance_type='gain').

For SVM: set to 1.0 (no native feature importance).

For GNNs (AFP, GCN, GAT): set to the average feature importance of all tree-based models.

Normalize so that the sum of feature importances over all models equals 1.

Step 3 – Base confidence score

For each model k:

$$\text{error\_w}_k = \frac{1}{\text{MSE}_k + 10^{-8}}$$

$$\text{var\_w}_k = \frac{1}{\text{Var}_k + 10^{-8}}$$

$$\text{base\_conf}_k = \left[(1 - \text{UNCERTAINTY\_WEIGHT}) \cdot \text{error\_w}_k + \text{UNCERTAINTY\_WEIGHT} \cdot \text{var\_w}_k\right] \cdot \text{feat\_imp}_k$$

$$\text{base\_conf}_k = \frac{\text{base\_conf}_k}{\sum_j \text{base\_conf}_j}$$

where UNCERTAINTY_WEIGHT = 0.4.

Step 4 – High-activity accuracy (high_acc)

On the training set, identify compounds with true $pIC_{50} \geq$ ACTIVITY_THRESHOLD (8.0). For each model, compute the proportion of these compounds for which the absolute prediction error ≤ HIGH_ACTIVE_TOLERANCE (0.3).

Step 5 – Pairwise prediction correlation (corrs)

Using all predictions from the training set, compute the Pearson correlation coefficient between every pair of models. If the coefficient is NaN, treat it as 0.0.

Step 6 – Dynamic weight calculation for a single compound (sample i)

For each validation or test compound i:

1. Collect predictions from all base models: pred_vals = [p1, p2, ..., pM].

2. Compute the mean prediction mean_pred and squared deviations dev = (pred_vals - mean_pred)^2.

3. Initial deviation-based weight:

$$\text{sample\_w} = \text{base\_conf} \cdot \left(\frac{1}{\text{dev} + 10^{-8}}\right)^{\text{CONFIDENCE\_POWER}}$$

where CONFIDENCE_POWER = 2.5.

4. Correlation penalty

   For each model j, calculate the average correlation with other models, weighted by model type similarity:

   - If two models are of the same type (tree–tree or graph–graph), the multiplicative factor is 1.2.
   - If they are of different types (e.g., tree–svm, graph–tree), the factor is 0.8.

     Then:

$$\text{penalty}_j = \sum_{l \neq j} \text{corrs}[j][l] \times \text{type\_factor}$$

$$\text{sample\_w}[j] \leftarrow \text{sample\_w}[j] \times \max\left(0.1,\ 1 - \text{CORRELATION\_PENALTY} \cdot \frac{\text{penalty}_j}{M-1}\right)$$

where CORRELATION_PENALTY = 0.15.

5. High-activity boost

   Determine whether the compound is considered "high-activity relevant":

   In training mode: is_high = (mean_pred ≥ 90th percentile of validation set pIC_{50}) or (true $pIC_{50} \geq 8.0$).

   In testing mode: is_high = (mean_pred ≥ 8.0) or (true $pIC_{50} \geq 8.0$) (the test set's true labels are available during evaluation, but in real deployment only the prediction condition is used).

   If is_high is True:

   Multiply sample_w by the model's high-activity accuracy: sample_w = sample_w * high_acc (element-wise).

   Compute std_pred = standard deviation of pred_vals.

   Set power = 0.5 if std_pred < 0.3 else 0.3.

Apply an additional power transform: sample_w = sample_w ** (CONFIDENCE_POWER * power).

Multiply by the boost factor: sample_w = sample_w * HIGH_ACTIVE_WEIGHT_BOOST (1.4).

6. Normalization

   Clip sample_w to a minimum of 1e-8, then divide by the sum to obtain final weights w_i for the sample.

Step 7 – Ensemble prediction

$$\hat{y}_i = \sum_{k=1}^{M} w_{k,i} \cdot p_{k,i}$$

$$\hat{y}_i = \text{clip}(\hat{y}_i, \text{PIC50\_MIN}, \text{PIC50\_MAX})$$

where PIC50_MIN = 3.0, PIC50_MAX = 11.0.

Important Implementation Notes

Label scaling for GNNs: GNNs output normalized predictions. A pre-fitted MinMax scaler (loaded from graph_label_minmax_{split}.pkl) is used to inverse-transform to the pIC_{50} scale, then clipped to [3,11].

Tree model predictions: All tree-based models (XGB, LGB, GB, RF) directly output pIC_{50} values (no scaling).

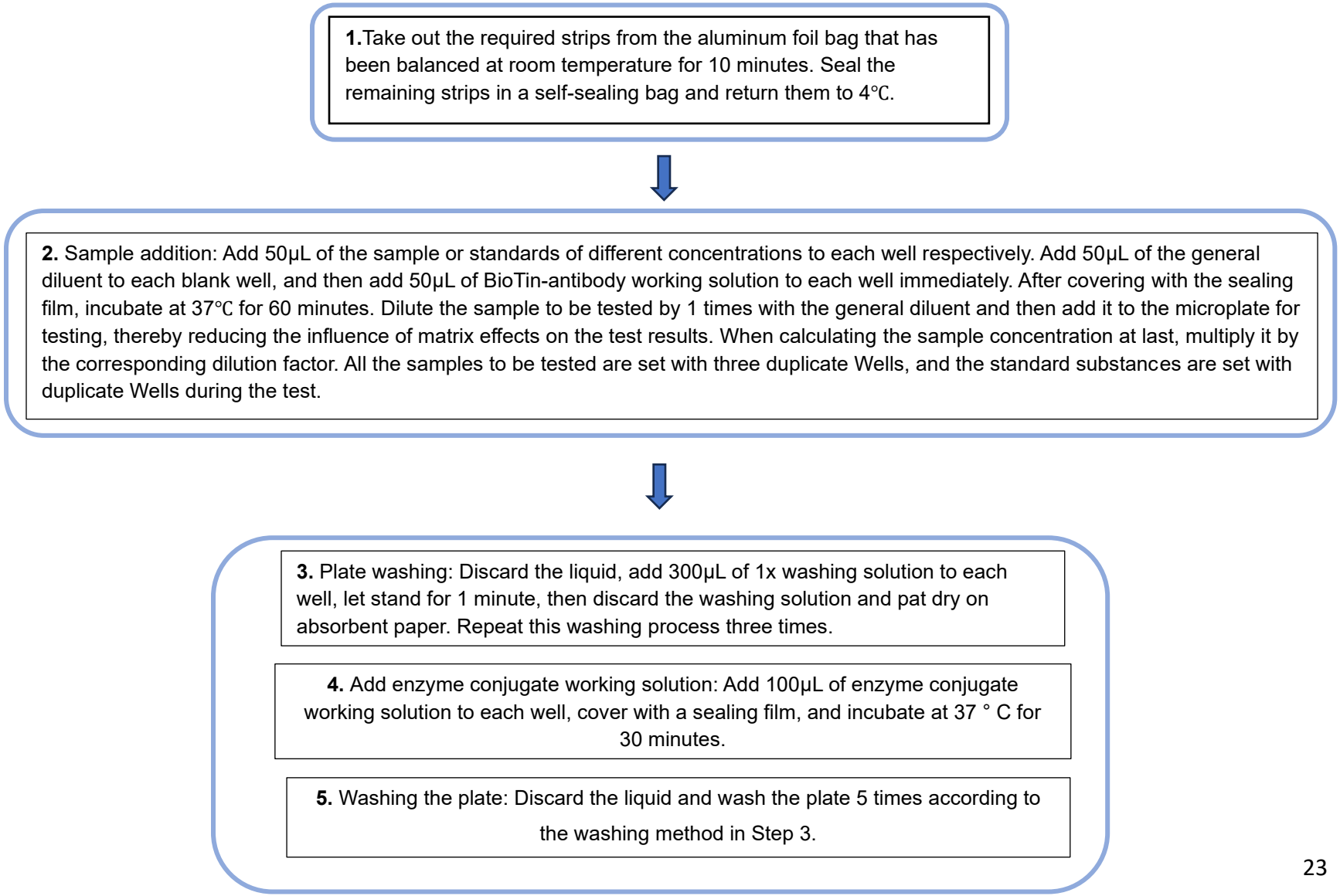
Weight saving: After training, the ensemble configuration (including errors, variances, feat_imp, high_acc, etc.) is saved to disk for later use in testing.

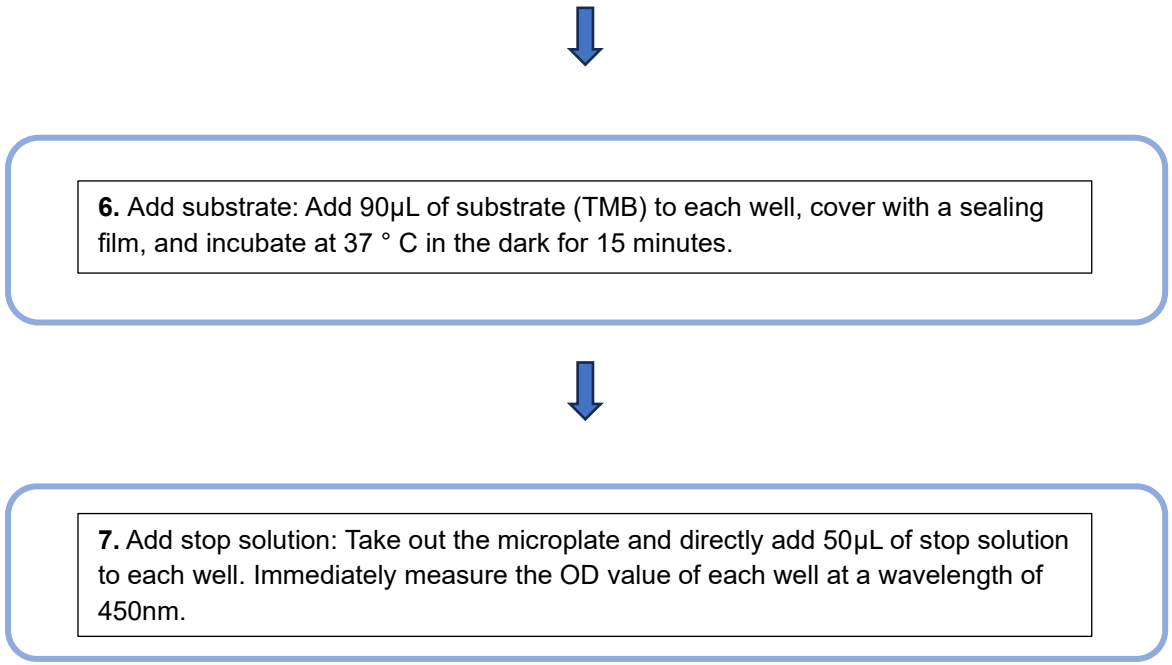
The specific combination of base models used for each data split (e.g., fingerprint split uses XGB, SVM, GCN) is reported in the main text (Table 3.3).

Parameters Quick Reference

Symbol in doc	Code variable	Value
α	CONFIDENCE_POWER	2.5
β_{unc}	UNCERTAINTY_WEIGHT	0.4
β_{corr}	CORRELATION_PENALTY	0.15
γ	HIGH_ACTIVE_WEIGHT_BOOST	1.4
High-active threshold	ACTIVITY_THRESHOLD	8.0
High-active tolerance	HIGH_ACTIVE_TOLERANCE	0.3

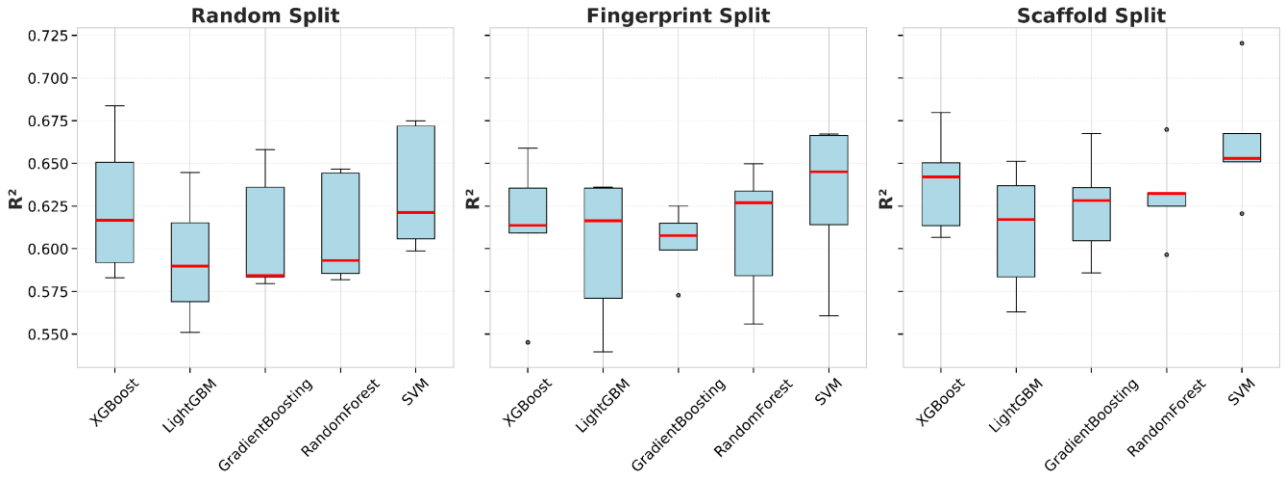
Figure S4.The operation process and method of cAMP inhibition experiment.





The above content is from the information of JONLNBIO (JL13253 96T) ELISA kit. For detailed operation manual, please refer to the following website: http://www.jonln.com/cus_human_camp/

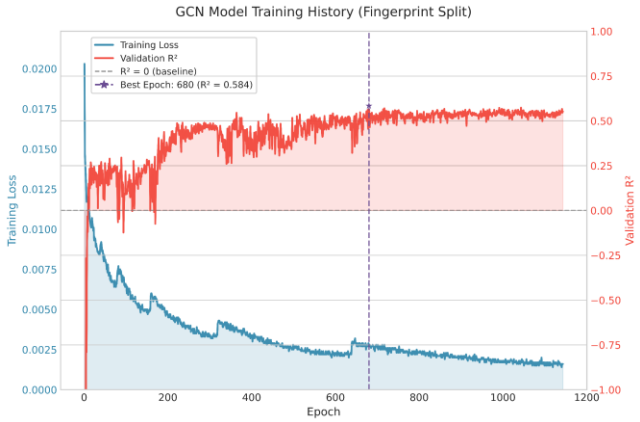
Figure S5. Boxplot of 5-fold cross-validation R^2 for machine learning models.



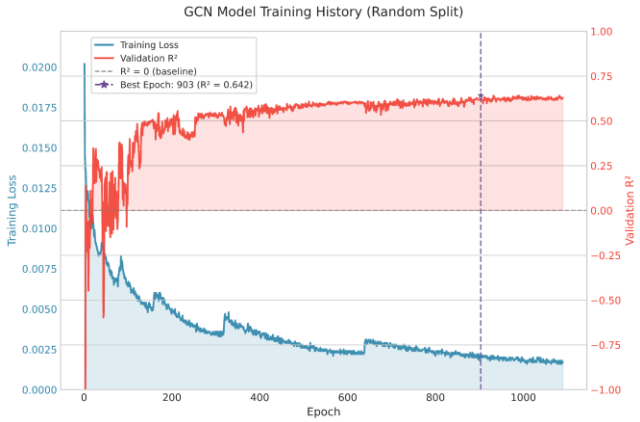
Boxplots illustrate the distribution of R^2 values derived from five independent folds for each model across three distinct splitting strategies: random, fingerprint, and scaffold. The median is denoted by a red line, while the boxes encompass the interquartile range (IQR), and the whiskers extend to 1.5 times the IQR. These results underscore the robustness of the machine learning model.

Figure S6. Training history of graph neural networks(GCN).

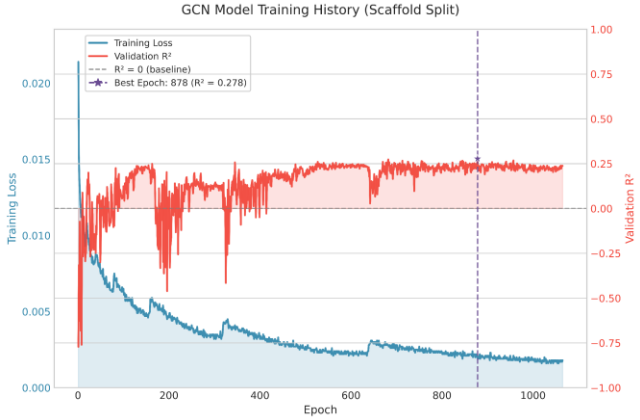
A.Fingerprint split GCN model.



B.Random split GCN model.

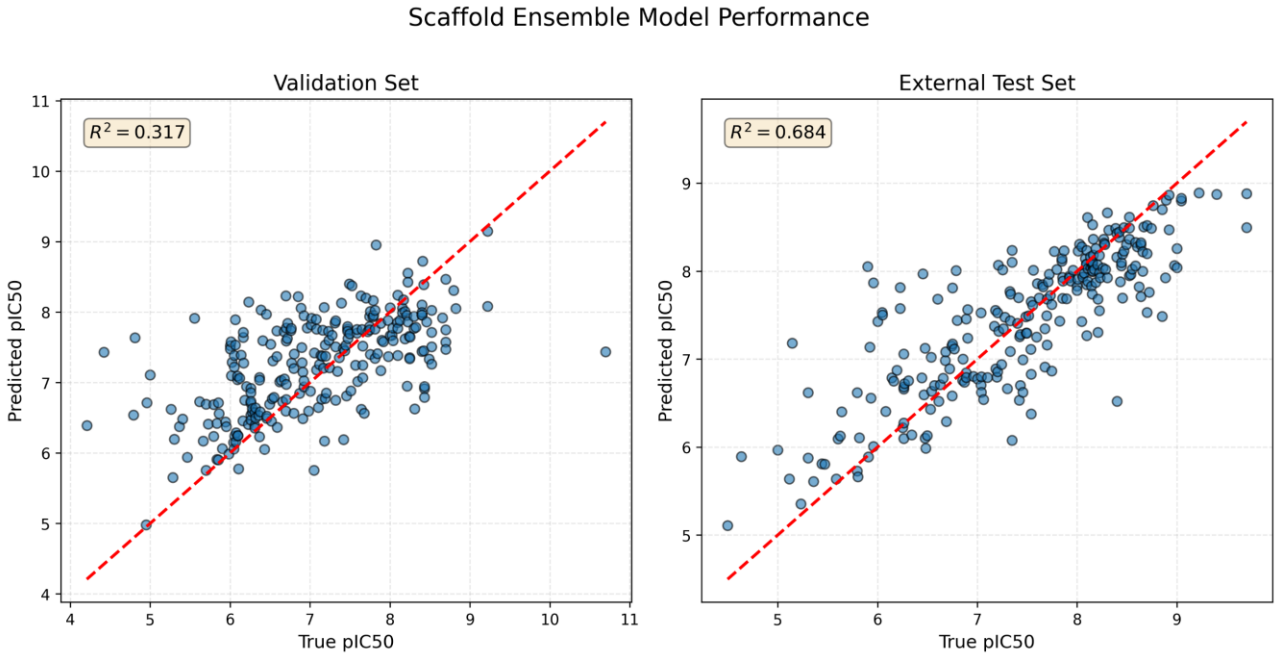


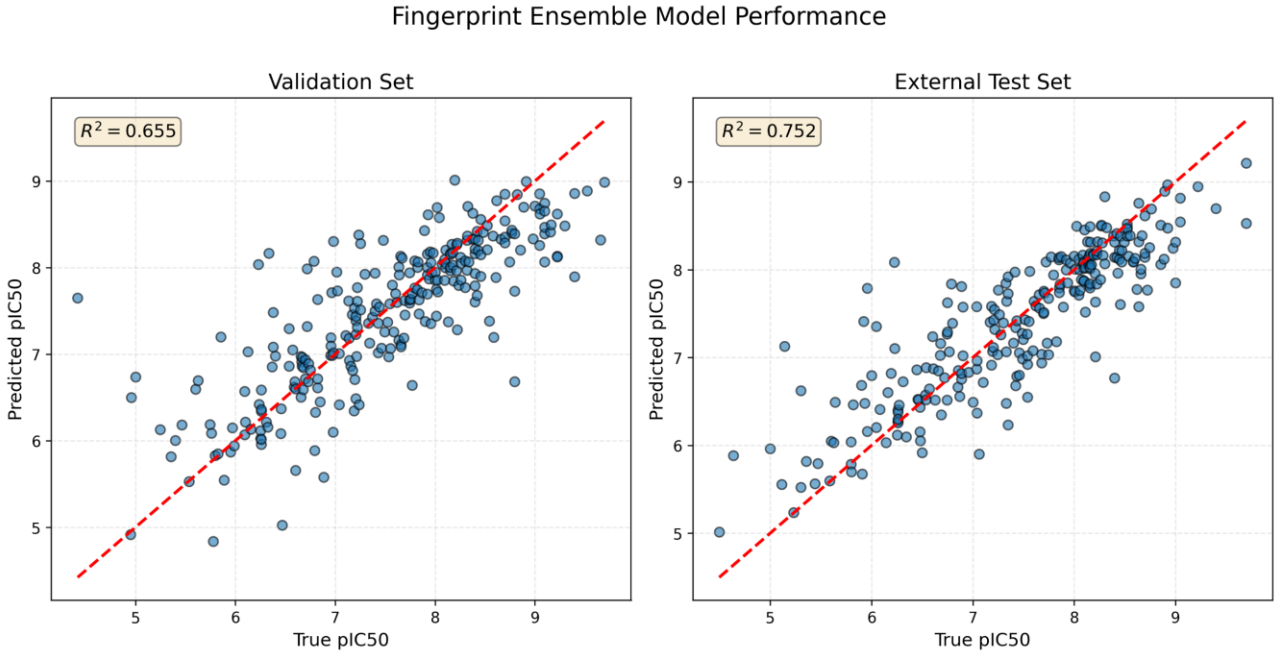
C.Scaffold split GCN model.

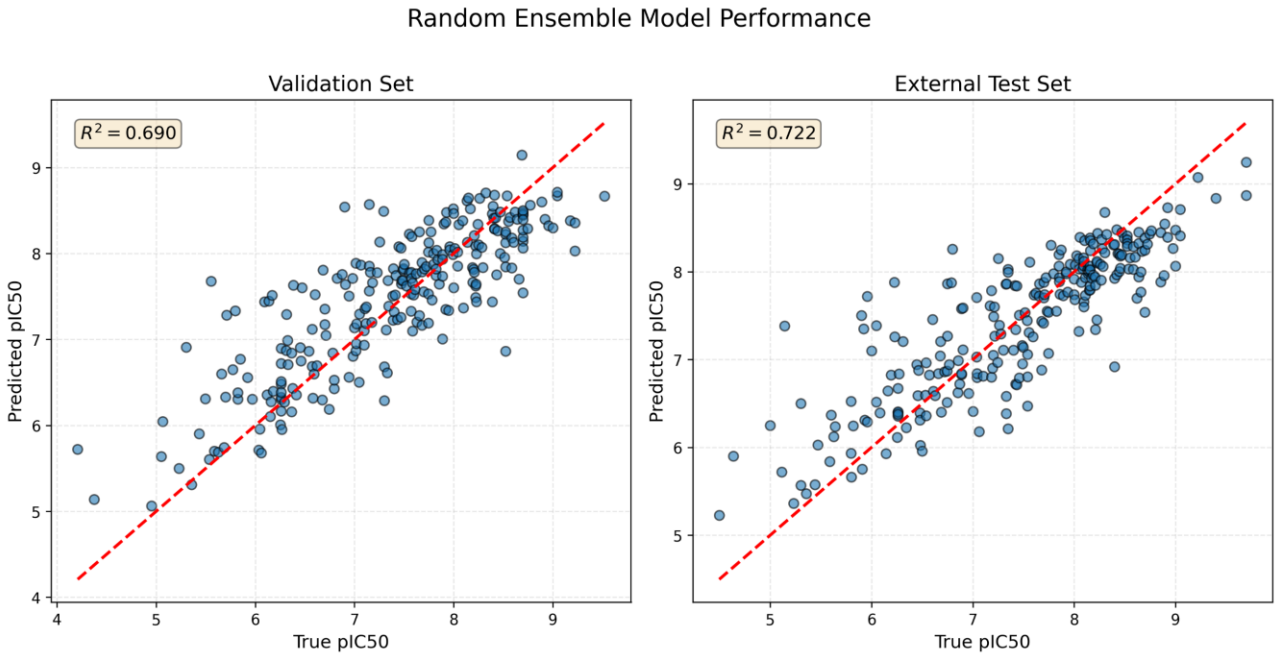


The graph presents representative learning curves for Graph Convolutional Network (GCN) models across three data partitioning strategies: Fingerprint (A), Random (B), and Scaffold (C). The left y-axis denotes the training loss, measured as mean squared error, while the right y-axis represents the validation R^2. The gray dotted line marks the epoch at which the highest validation R^2 was achieved, indicating the point of early stopping. Training was conducted using the AdamW optimizer in conjunction with a cosine annealing learning rate scheduler. Detailed parameter settings are available within the graph model training code. It is important to note that the optimal validation R^2 observed at a specific epoch may not correspond to the final validation R^2. This graph is intended solely to illustrate the training process.

FigureS7. Scatter plots of the predicted values and true values of the validation set/external test set in three different partitioned integrated models.




Fingerprint Ensemble Model Performance
Validation Set
$R^2 = 0.655$
Predicted pIC50
True pIC50
External Test Set
$R^2 = 0.752$
Predicted pIC50
True pIC50



The figure presents scatter plots illustrating the relationship between predicted and actual values for the integrated model applied to both the validation and external test datasets. The upper section of the figure corresponds to results obtained under the Scaffold division, the middle section pertains to the Fingerprint division, and the lower section represents the Random division. Within each section, the left panel displays results from the validation set, while the right panel shows results from the external test set. The Random integrated models utilized include GCN, GB, and SVM; the Fingerprint division employs GCN, XGB, and SVM; and the Scaffold division incorporates GAT, XGB, and SVM.

Table S4. Detailed model performance.

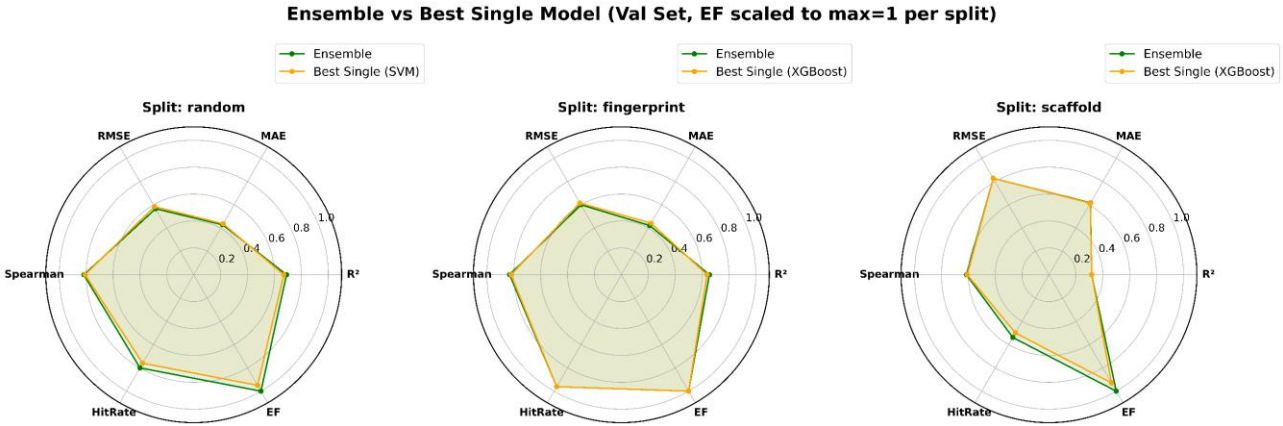
Split	Model	R2_val	MAE_val	RMSE_val	Spearman_val	HitRate @10%_val	EF_val	R2_cv_mean	R2_cv_std	R2_test	MAE_test	RMSE_test	Spearman_test	HitRate @10%_test	EF_test
random	XGBoost	0.6434	0.4500	0.6082	0.7972	0.8000	2.5722	0.6252	0.0420	0.6847	0.4086	0.5732	0.8120	0.9600	2.4048
random	LightGBM	0.6190	0.4770	0.6287	0.7861	0.8000	2.5722	0.5939	0.0371	0.6438	0.4453	0.6092	0.7871	0.9600	2.4048
random	GradientBoosting	0.6608	0.4545	0.5932	0.8068	0.9200	2.9580	0.6083	0.0363	0.6924	0.4262	0.5661	0.8389	0.9600	2.4048
random	RandomForest	0.6483	0.4539	0.6040	0.8094	0.8800	2.8294	0.6103	0.0324	0.7071	0.4128	0.5524	0.8393	0.9600	2.4048
random	SVM	0.6700	0.4375	0.5851	0.8078	0.7600	2.4435	0.6344	0.0365	0.7123	0.3906	0.5475	0.8453	1.0000	2.5050
random	GAT	0.6155	0.4747	0.6316	0.7761	0.8400	2.7008			0.6540	0.4370	0.6010	0.8190	0.8800	2.2044
random	GCN	0.6410	0.4538	0.6103	0.7825	0.7600	2.4435			0.6420	0.4670	0.6110	0.8010	0.8800	2.2044
random	AttentiveFP	0.5393	0.4793	0.6914	0.7529	0.8400	2.7008			0.5870	0.4430	0.6560	0.7670	0.7600	1.9038
random	ensemble(GB+SVM+GCN)	0.6904	0.4294	0.5668	0.8177	0.8000	2.5722			0.7224	0.4004	0.5378	0.8498	1.0000	2.5050
fingerprint	XGBoost	0.6384	0.4423	0.6157	0.8210	0.9615	2.5738	0.6125	0.0425	0.7184	0.3878	0.5417	0.8465	1.0000	2.5050
fingerprint	LightGBM	0.6229	0.4499	0.6287	0.8142	0.9231	2.4709	0.5997	0.0428	0.7013	0.4103	0.5578	0.8422	1.0000	2.5050
fingerprint	GradientBoosting	0.5961	0.4720	0.6507	0.7989	0.9231	2.4709	0.6040	0.0199	0.6951	0.4125	0.5636	0.8476	0.9600	2.4048
fingerprint	RandomForest	0.6053	0.4630	0.6432	0.8083	0.9615	2.5738	0.6101	0.0388	0.7217	0.3956	0.5385	0.8532	0.9600	2.4048
fingerprint	SVM	0.6226	0.4332	0.6290	0.8097	0.9231	2.4709	0.6306	0.0446	0.7379	0.3693	0.5226	0.8621	1.0000	2.5050
fingerprint	GAT	0.5983	0.4716	0.6489	0.8008	0.9231	2.4709			0.7100	0.3990	0.5490	0.8490	0.9200	2.3046
fingerprint	GCN	0.5737	0.4859	0.6685	0.7865	1.0000	2.6768			0.6720	0.4390	0.5850	0.8350	1.0000	2.5050
fingerprint	AttentiveFP	0.4890	0.5353	0.7319	0.7460	0.9615	2.5738			0.6220	0.4500	0.6270	0.8050	0.8800	2.2044

fingerprint	ensemble(XGB+SVM+GCN)	0.6552	0.4225	0.6012	0.8311	0.9615	2.5738			0.7520	0.3654	0.5083	0.8683	1.0000	2.5050
scaffold	XGBoost	0.3176	0.6148	0.8280	0.6080	0.5000	2.3182	0.6384	0.0296	0.6900	0.4036	0.5683	0.8282	1.0000	2.5050
scaffold	LightGBM	0.3033	0.6237	0.8366	0.6111	0.4615	2.1399	0.6103	0.0367	0.6668	0.4317	0.5892	0.8032	1.0000	2.5050
scaffold	GradientBoosting	0.2928	0.6281	0.8429	0.5915	0.5385	2.4965	0.6244	0.0312	0.6797	0.4254	0.5777	0.8272	0.9600	2.4048
scaffold	RandomForest	0.2852	0.6385	0.8474	0.5928	0.4615	2.1399	0.6312	0.0262	0.6766	0.4262	0.5805	0.8312	0.9600	2.4048
scaffold	SVM	0.2592	0.6366	0.8627	0.5776	0.5000	2.3182	0.6624	0.0366	0.6918	0.4105	0.5667	0.8341	1.0000	2.5050
scaffold	GAT	0.3048	0.5975	0.8358	0.6018	0.6538	3.0315			0.6290	0.4770	0.6220	0.8110	0.9600	2.4048
scaffold	GCN	0.2516	0.6429	0.8671	0.5884	0.6154	2.8531			0.5940	0.4870	0.6500	0.7760	0.9200	2.3046
scaffold	AttentiveFP	0.1134	0.6988	0.9438	0.4955	0.6538	3.0315			0.5210	0.4820	0.7070	0.7230	0.8800	2.2044
scaffold	ensemble(XGB+LGB+GAT)	0.3175	0.6176	0.8281	0.6141	0.5385	2.4965			0.6838	0.4115	0.5740	0.8179	1.0000	2.5050

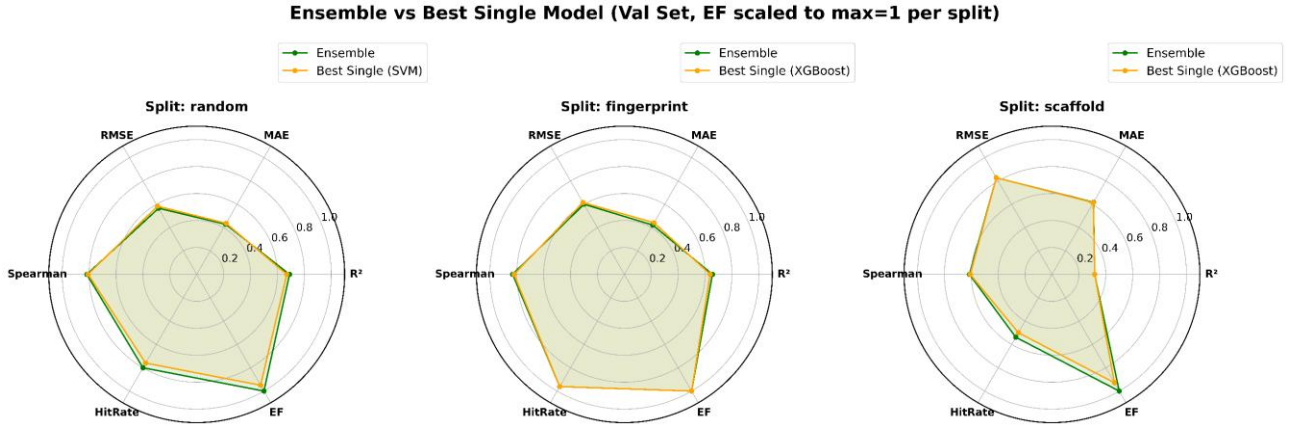
The performance metrics (R^2, MAE, RMSE, Spearman, HitRate@10%, EF) of both individual and integrated models were evaluated on the validation and external test sets across three different splitting strategies. For the machine learning models, the results are presented as the mean ± standard deviation from five-fold cross-validation. In the integrated model employing fingerprint division, the GCN model was selected for the graph-based component due to its stability. Although the GAT model demonstrated superior performance compared to the GCN model on both the validation and test sets, the R^2 value for the integrated model combining GAT, XGB, and SVM on the validation set was 0.646, which is lower than the 0.655 achieved by the GCN+XGB+SVM combination. On the test set, the R^2 values were 0.754 and 0.752, respectively, indicating negligible differences between the two. Consequently, to ensure more robust outcomes, the GCN+XGB+SVM integrated model strategy was selected under the fingerprint partitioning approach and subsequently applied to virtual screening.

FigureS8. Radar charts comparing the ensemble model with the best single.

A. Val Set



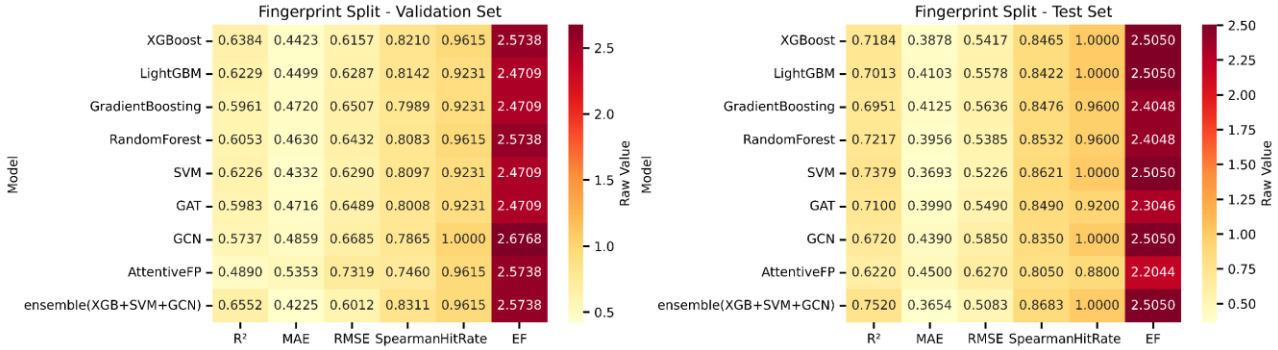
B. Test Set



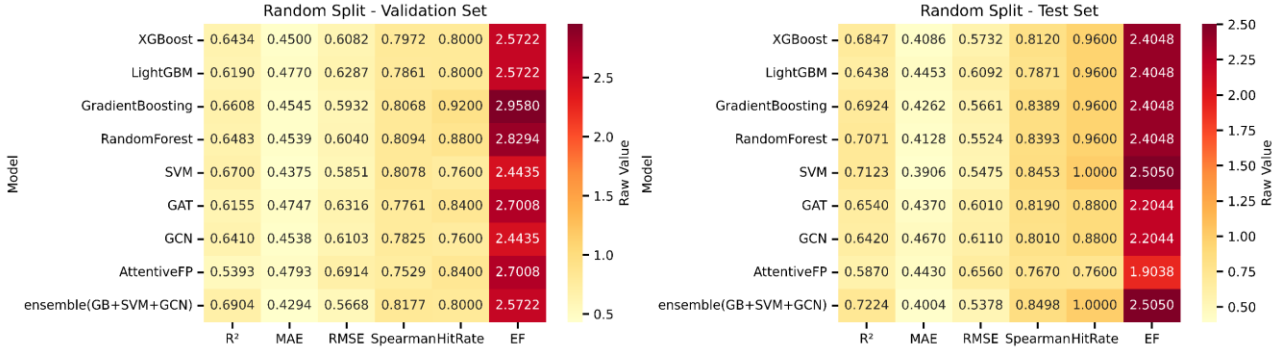
A – Performance on the validation set. B – Performance on the external test set. Six metrics are presented: R^2, MAE, RMSE, Spearman's ρ, HitRate@10%, and enrichment factor (EF@10%). For R^2, Spearman's ρ, HitRate, and EF, higher values signify superior performance, whereas for MAE and RMSE, lower values are preferable. Under both random and fingerprint splits, the ensemble model (depicted in blue) consistently surpasses the best single model (depicted in orange) in performance on both validation and test sets, particularly excelling in R^2 and EF metrics. In the more challenging scaffold split, the ensemble model performs comparably to the best single model on the validation set but demonstrates significantly enhanced performance on the external test set (e.g., $R^2 = 0.752$ for the ensemble compared to 0.718 for the best single model in the fingerprint split, as detailed in the main text but not shown in the radar plot). The close alignment between validation and test radar profiles under random and fingerprint splits indicates excellent generalizability. Conversely, the scaffold split reveals a larger discrepancy, which is anticipated given that the validation set includes intentionally novel scaffolds.

FigureS9. Heatmaps of model performance for all five ML models and three GNNs under three data splitting strategies.

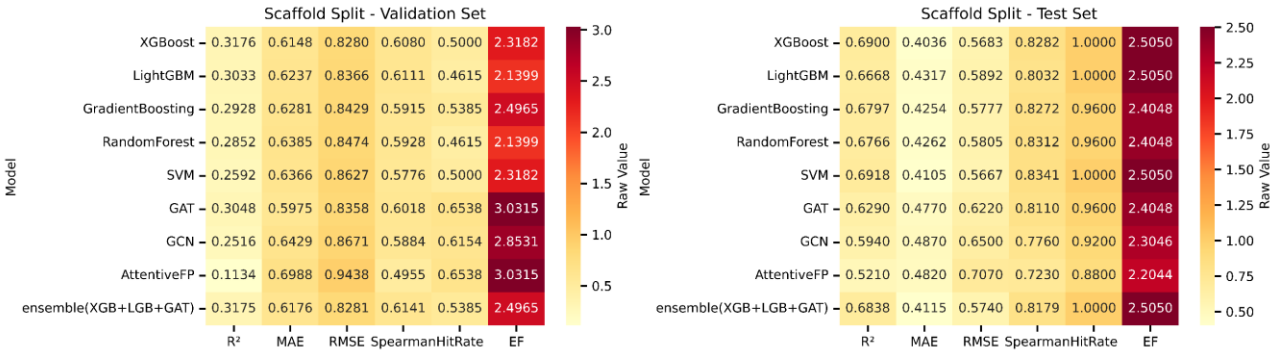
A. Fingerprint



B. Random



C. Scaffold



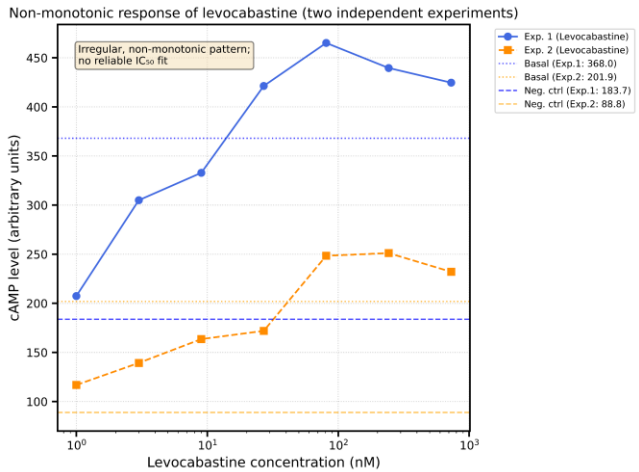
A – Fingerprint split. B – Random split. C – Scaffold split. In the context of the random split (B), Support Vector Machines (SVM) and Gradient Boosting exhibit the highest R^2 values, exceeding 0.66, while Graph Neural Networks (GNNs) demonstrate competitive performance. In the fingerprint split scenario (A), XGBoost achieves an R^2 of 0.638, and LightGBM also performs well; GNNs experience a slight decline in performance but maintain robustness. Under the scaffold split (C), all models experience a reduction in performance, with R^2 values ranging from approximately 0.3 to 0.35, although XGBoost continues to lead. GNNs display a more pronounced decline, suggesting increased sensitivity to novel scaffolds. These heatmaps corroborate the assertion that the ensemble model surpasses the performance of any individual model, as elaborated in the main text.

Table S5. Detailed information on the Top 20 candidate drugs.

Rank	Drug Name	Mechanism	MW (g/mol)	Matching	ensemble_pred (pIC50)
1	Pitavastatin	HMG-CoA reductase inhibitor	421.46	High	7.563
2	Fluvastatin	HMG-CoA reductase inhibitor	411.47	High	7.336
3	Levocabastine	Histamine H1 receptor antagonist	420.53	High	7.285
4	Astemizole	a second generation antihistamine	458.58	High	7.323
5	Seladelpar	PPAR-delta agonist	444.47	Medium	7.661
6	Posaconazole	Triazole antifungal	700.78	Medium	7.472
7	Bazedoxifene	SERM (estrogen receptor modulator)	470.61	Medium	7.351
8	Rimegepant	CGRP receptor antagonist	534.57	Medium	7.45
9	Pimavanserin	5-HT2A receptor antagonist	427.56	Low	7.7
10	Etofamide	Anti-amoebic agent	427.28	Low	7.627
11	Maralixibat	Ileal bile acid transporter inhibitor	674.96	Low	7.285
12	Delamanid	Anti-tuberculosis antibiotic	534.49	Low	7.271
13	Silodosin	α1-adrenergic receptor antagonist	499.54	Low	7.439
14	Sertindole	Atypical antipsychotic	440.94	Low	7.254
15	Revumenib	Menin inhibitor (antineoplastic)	630.82	Low	7.353
16	Vemurafenib	BRAF inhibitor (antineoplastic)	489.92	Low	7.302
17	Entrectinib	TRK/ROS/ALK inhibitor (antineoplastic)	560.65	Low	7.273
18	Elacestrant	Estrogen receptor antagonist	458.65	Low	7.262
19	Bemotrizinol	UV filter (sunscreen)	627.83	Low	7.297
20	Indium In-111 oxyquinoline	Radiopharmaceutical	543.36	Low	7.317

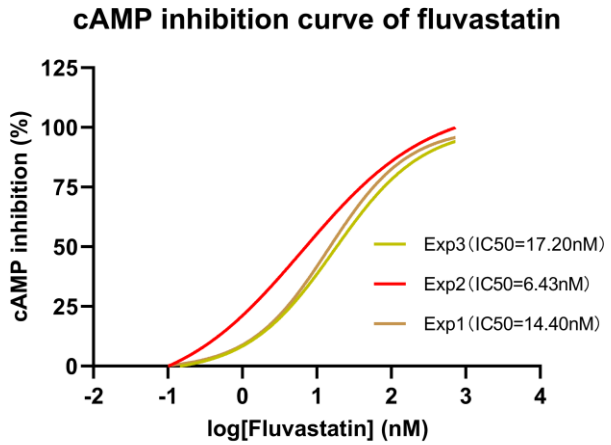
The ranking rule is determined based on whether there have been clinical or relevant experimental studies as described in the text, as well as the matching degree of the original indications for asthma.

Figure S10. Non-monotonic cAMP response of levocabastine in the DP2 inhibition assay.



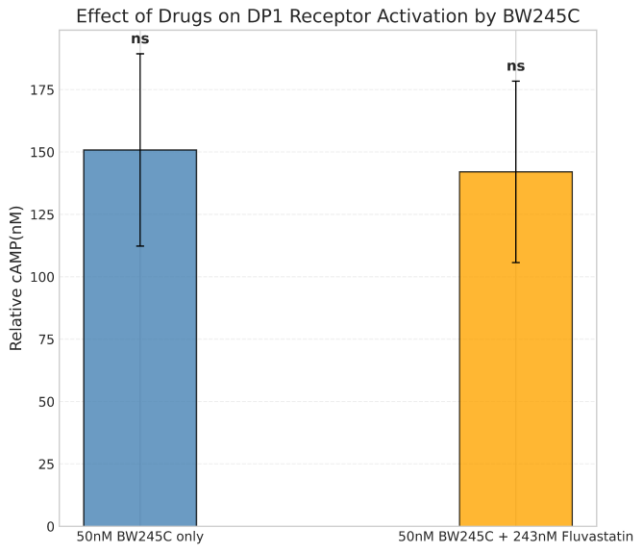
KU812 cells were exposed to escalating concentrations of levocabastine (1–729 nM) in the presence of 50 nM DK PGD_2 and 10 µM forskolin. In contrast to the typical concentration-dependent, monotonic inhibition pattern observed with fluvastatin, levocabastine exhibited irregular, non-monotonic fluctuations across the tested concentration range. This pattern was not reproducible in a dose-dependent manner and could not be adequately modeled using a four-parameter logistic model. As a result, levocabastine was excluded from further quantitative analysis, including IC_{50} estimation, and from subsequent functional assays. The observed irregularity suggests potential off-target interactions, compound instability, or non-specific assay interference, which are beyond the scope of the current DP2-focused validation study.

Figure S11. Individual dose-response curves of fluvastatin from three independent experiments in the cAMP inhibition assay.



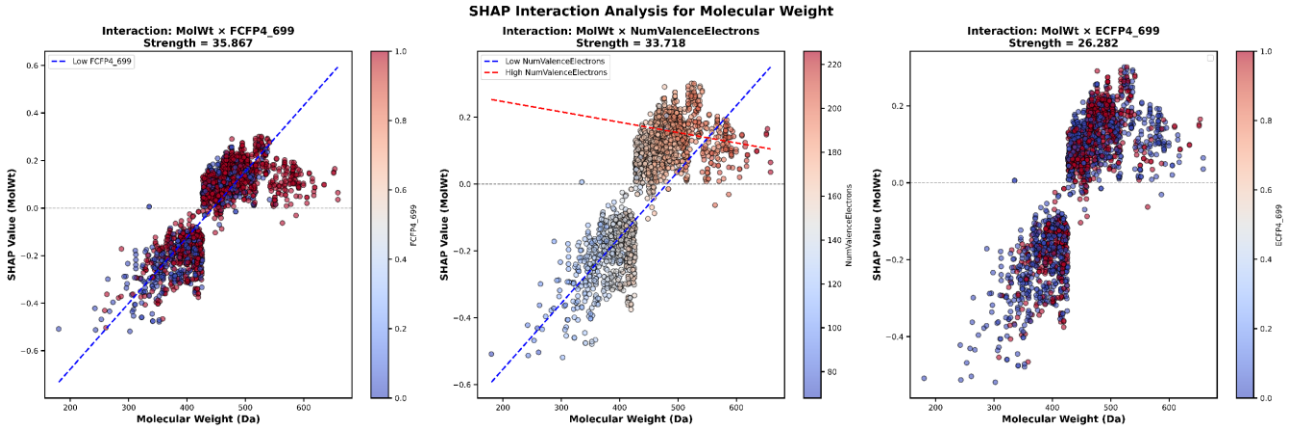
The three individual curves display significant overlap across the entire concentration range, exhibiting nearly superimposable sigmoidal shapes with comparable bottom, top, and slope parameters. The IC_{50} values calculated from each experiment were as follows: Experiment 1 = 14.4 nM, Experiment 2 = 6.433 nM, and Experiment 3 = 17.2 nM. The close concordance among replicates indicates high experimental reproducibility and substantiates the reliability of the pooled IC_{50} estimate (12.72 nM, 95% CI: 8.05–22.41 nM) reported in the main text (Figure 10). The minimal inter-experiment variability further corroborates that fluvastatin consistently induces concentration-dependent inhibition of DP2-mediated cAMP signaling, thereby supporting its identification as a potent and reproducible DP2 antagonist.

Figure S12.DP1 significance analysis.



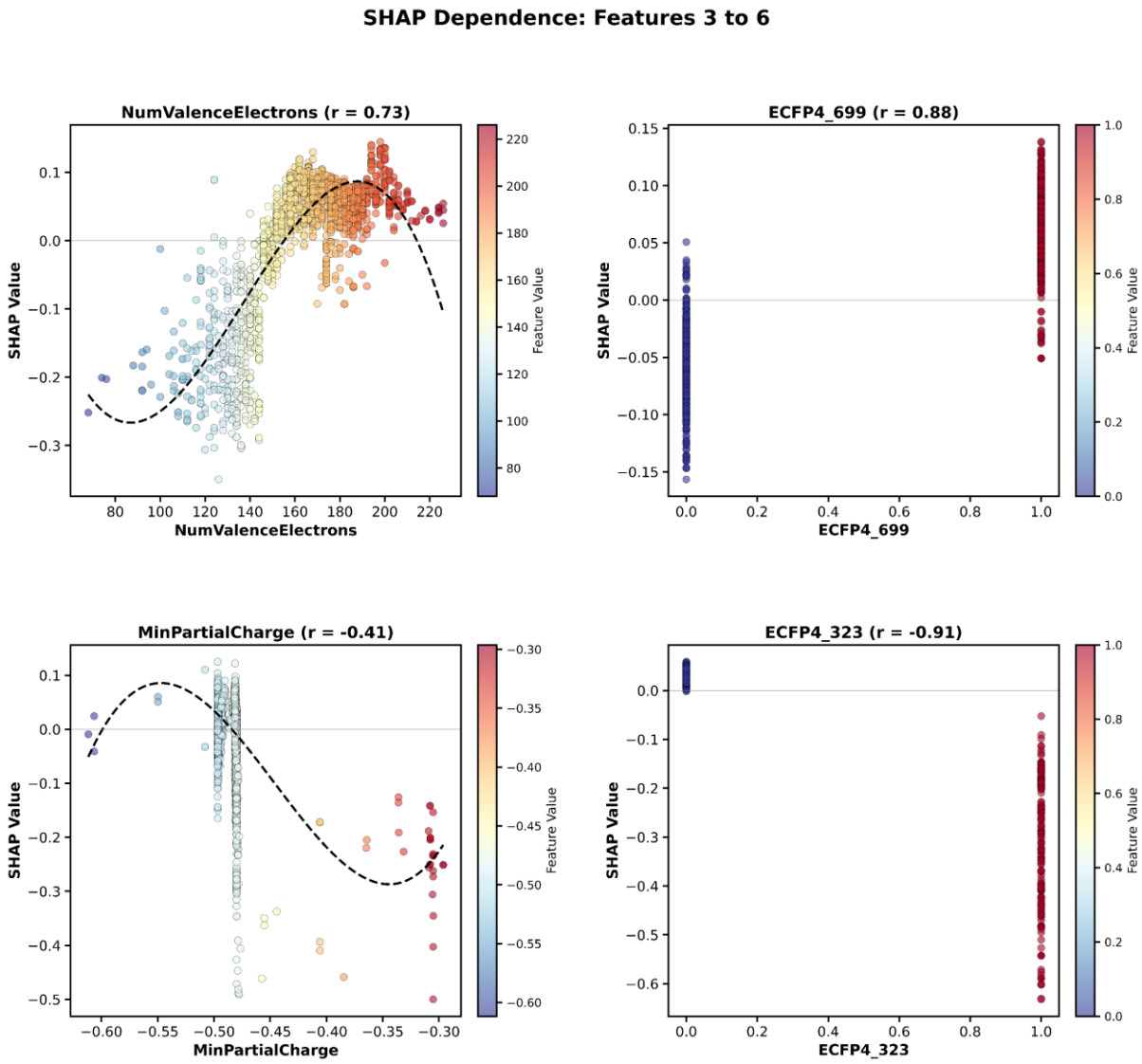
In the camp detection experiment, we incorporated a cohort of DP1 agonists to preliminarily assess the specific activating effect of the screened fluvastatin on DP2. Separate control groups for DP1 (comprising only 50 nM BW245C) and DP1 drug groups (comprising 50 nM BW245C and 243 nM fluvastatin) were established. Data analysis was conducted using the independent sample t-test (two-tailed, unequal variance). The results indicated no significant difference ($p > 0.05$), suggesting that the screened fluvastatin exhibits a specific effect on DP2 without antagonizing DP1.

Figure S13. SHAP interaction analysis of molecular weight (MolWt) with three selected features: FCFP4_699, ECFP4_699, and NumValenceElectrons.



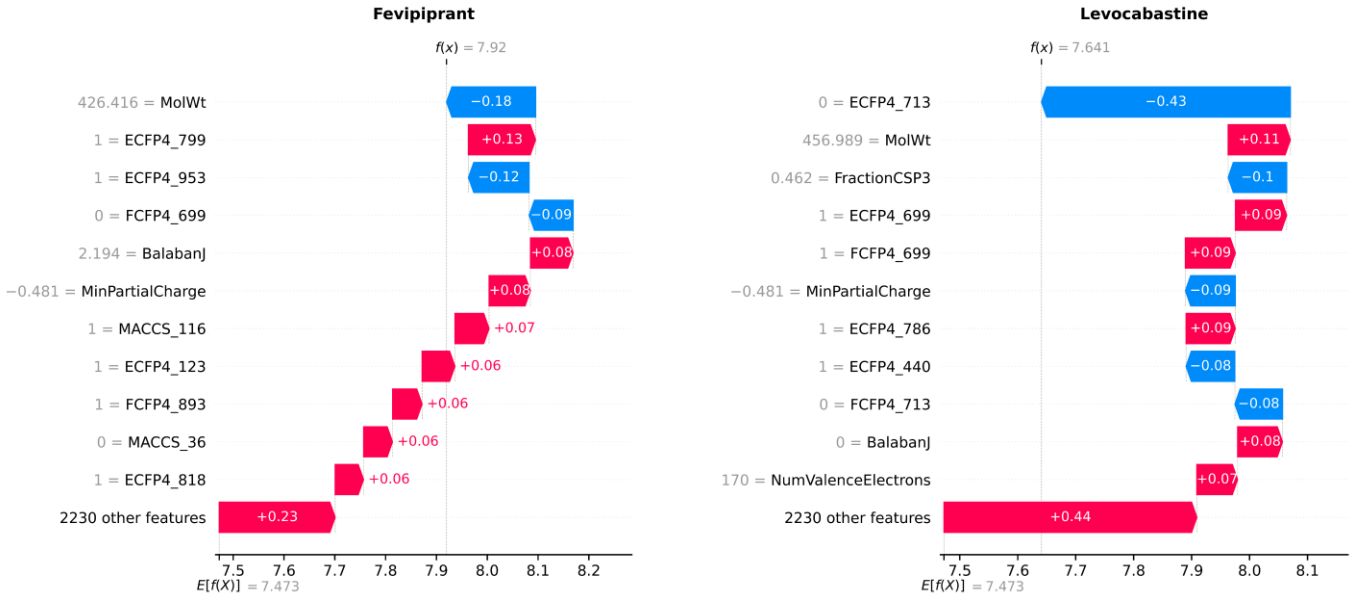
The interaction plots indicate that the optimal molecular weight range (420–550 Da) is influenced by concurrent structural characteristics. The presence of particular pharmacophore elements, such as FCFP4_699 and ECFP4_699, results in a leftward shift of this range, whereas an increased number of valence electrons enhances the advantageous effects associated with higher molecular weights. These insights offer practical guidance for the design of novel DP2 antagonists: integrating the FCFP4_699 substructure may permit the use of slightly smaller molecules, while maintaining extended conjugation could justify the selection of higher molecular weights within the active range.

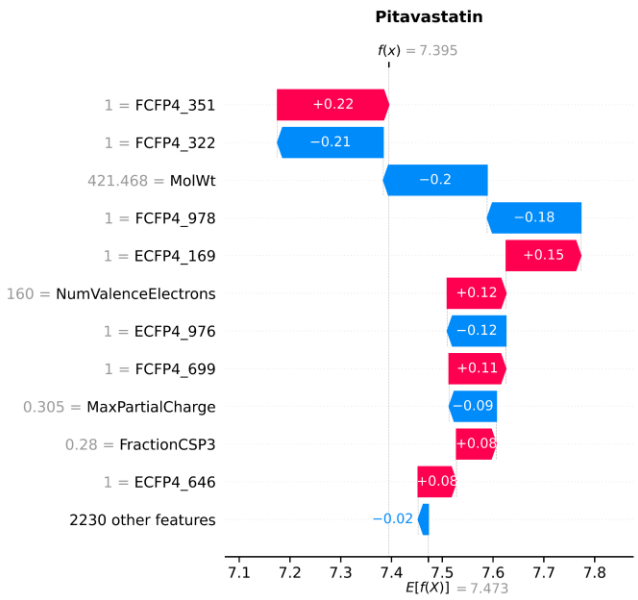
Figure S14. SHAP feature importance analysis(Top3-6)



The positive impact of NumValenceElectrons is associated with the extent of the molecular conjugated system, indicating potential π-π stacking interactions. ECFP4_699 and ECFP4_323 correspond to distinct local topological substructures and exhibit a modest yet positive influence on activity. A moderately negative value of MinPartialCharge facilitates electrostatic interactions, whereas an excessively negative value is counterproductive. Collectively, these features enhance the prediction of DP2 activity alongside molecular weight and essential pharmacophore fingerprints.

Figure S15. SHAP waterfall plots for fevipiprant (positive control), pitavastatin, and levocabastine.





Each graph highlights the feature with the most significant impact on the predicted pIC_{50} of compounds, with positive impacts in red and negative in blue. It also shows the base value (average prediction) and the final prediction, f(x). For Fevipiprant, a DP2 antagonist, the waterfall plot shows a strong positive contribution from features like FCFP4_699 and ECFP4_169, leading to a high predicted pIC_{50} that matches its known activity. Its molecular mass also positively influences the prediction. Contributions from pitavastine and levocabastine align with observed trends, reflecting consistency in molecular weight and fingerprint characteristics.

Table S6. Display of the top 20 SHAP values.

Rank	Feature	Mean_Abs_SHAP
1	MolWt	0.15595
2	FCFP4_699	0.082593
3	NumValenceElectrons	0.078563
4	ECFP4_699	0.056872
5	MinPartialCharge	0.05507
6	ECFP4_323	0.053852
7	MACCS_36	0.052722
8	BalabanJ	0.046794
9	MaxPartialCharge	0.045652
10	FractionCSP3	0.040754
11	Kappa1	0.039899
12	MolLogP	0.038003
13	FCFP4_893	0.037668
14	ECFP4_175	0.032904
15	ECFP4_713	0.032275
16	FCFP4_537	0.029724
17	ECFP4_834	0.029691
18	MolMR	0.025447
19	ECFP4_646	0.024388
20	TPSA	0.02218