Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Empty scaffold allocation bias in molecular distribution shift evaluation

Tao Song

tsong@upc.edu.cn

China University of Petroleum (East China)

Yong Wang

China University of Petroleum (East China)

Shuang Wang

China University of Petroleum (East China)

Xun Wang

Tiangong University

Article

Keywords:

Posted Date: July 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9951586/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: There is **NO** Competing Interest.

Empty scaffold allocation bias in molecular distribution shift evaluation

Yong Wang[1], Shuang Wang[1], Xun Wang[1] & Tao Song[1*]

[1]Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China.

[*]email: tsong@upc.edu.cn

Abstract

Scaffold-based evaluation is widely used to estimate molecular machine-learning generalization to unseen scaffolds, but its validity depends on each split unit representing a chemically meaningful shared core. Here we show that this principle breaks down for molecules with empty Bemis–Murcko scaffold representations. These no-scaffold molecules are not an ordinary scaffold class: they lack a common ring–linker framework, are internally heterogeneous, show much lower within-unit similarity than true scaffold groups, and are weakly supported by the scaffold-bearing chemical space. Nevertheless, standard scaffold splitting collapses them into a single empty-scaffold bucket and treats them as one indivisible split unit. Across MoleculeNet and Deep-PK/ADMET benchmarks, this is not rare: 8 of 16 MoleculeNet datasets and 40 of 73 Deep-PK/ADMET tasks contain more than 10% no-scaffold molecules, with several physicochemical endpoints approaching 50%. We further show that Scaffold Split, UMAP-based clustering, DataSAIL and LoHi can all deplete no-scaffold molecules from training while enriching them in validation or test sets. This hidden allocation bias confounds scaffold-level evaluation with residual-subpopulation underrepresentation, compressing classification ranking toward the prior baseline and enriching high-multiple regression error inflation. Empty-Scaffold-Aware repair restores training-side representation while retaining evaluation exposure and controlling train-facing similarity.

Introduction

Machine learning models are increasingly used for molecular property prediction, drug discovery, toxicity screening and ADMET assessment[1-6]. Their apparent utility is often judged by benchmark performance, yet such estimates depend critically on how molecules are split into training and evaluation sets[4, 7, 8]. Random splits can place close structural analogues on both sides of the split, yielding overly optimistic estimates of generalization[7, 9, 10]. Scaffold-based evaluation is therefore widely used to approximate extrapolation to unseen structural series or new chemical space by assigning molecules with the same scaffold representation to the same split branch[7, 9, 11].

This strategy relies on a basic but often implicit premise: each scaffold-defined split unit should correspond to a chemically meaningful shared core11. For scaffold-bearing molecules, Bemis–Murcko scaffolds provide such a representation through ring systems and linkers[11]. However, for molecules lacking a retained ring–linker framework, the Bemis–Murcko procedure can return an empty scaffold representation. In standard scaffold splitting, these molecules are not separated according to their structural differences. Instead, they are collapsed into a single empty-scaffold bucket and treated as one indivisible split unit. This creates a conceptual failure of scaffold splitting: an absence-defined residual bucket is treated as if it were a chemically coherent scaffold class.

The consequences of this representation-induced grouping have not been systematically examined. If no-scaffold molecules were rare, internally coherent or well supported by scaffold-bearing chemical space, their allocation would likely have limited impact on evaluation. A more problematic scenario arises when they are common, structurally heterogeneous, weakly cohesive and poorly supported by molecules retained in the training set. Under this condition, scaffold splitting can unintentionally move a diverse residual population away from training and concentrate it in validation or test branches. The resulting evaluation is no longer a clean scaffold holdout; it can confound scaffold-level generalization with training-side underrepresentation of an absence-defined molecular subpopulation[10-13]. Conversely, if

no-scaffold molecules are absent from validation or test sets, model selection and final evaluation may miss failures on this population altogether.

Existing work has shown that molecular benchmark estimates are highly sensitive to split design[4, 7, 9, 10]. Scaffold splits reduce analogue leakage relative to random splits, but they do not fully capture the chemical diversity encountered in prospective screening or external compound libraries[10, 14]. Beyond scaffold grouping, UMAP-based clustering splits have been proposed to create more structurally separated evaluation sets and to provide a more challenging benchmark for virtual-screening settings[14-16]. DataSAIL and LoHi address related limitations by reducing similarity-induced leakage or by aligning evaluation with drug-discovery scenarios such as hit identification and lead optimization[17, 18]. However, these approaches primarily target global train–test similarity, leakage reduction or task-level realism, and do not explicitly address empty-scaffold allocation. It therefore remains unclear how common no-scaffold molecules are in standard benchmarks, whether they behave like ordinary scaffold groups, whether current splitters preserve their training-side representation, and how allocation imbalance affects downstream model assessment.

Here we identify empty-scaffold allocation as a previously overlooked failure mode in scaffold-based molecular evaluation. Across MoleculeNet and DeepPK/ADMET benchmarks, we show that no-scaffold molecules are frequent in many datasets and that the empty-scaffold bucket is chemically diffuse, weakly cohesive and weakly supported rather than scaffold-like. We then compare Scaffold Split, UMAP-based clustering split, DataSAIL and LoHi, showing that none explicitly guarantees no-scaffold training-side representation. This allocation imbalance creates a low-support evaluation axis that is distinct from scaffold novelty alone: in classification tasks, it is associated with compression of ranking performance toward the prior baseline; in regression tasks, it is associated with high-multiple relative mean absolute error (relMAE) inflation. Finally, we introduce Empty-Scaffold-Aware repair (ESA), a split-level repair strategy that preserves the scaffold-bearing leakage-aware backbone while reassigning no-scaffold molecules to restore training-side representation and control additional train-facing similarity exposure. Together, these results show that reliable

molecular benchmark design should diagnose not only analogue leakage and global train–test similarity, but also representation-induced underrepresentation of weakly supported residual subpopulations.

Results

Empty scaffold representations are common

Empty scaffold representations occur frequently in widely used molecular machine-learning benchmarks. Among the 16 MoleculeNet datasets, 8 contained more than 10% no-scaffold molecules, with FreeSolv and QM7 reaching 49.84% and 44.42%, respectively (**Fig. 1a, Supplementary Table 1**). In the Deep-PK/ADMET benchmark, 40 of 73 tasks contained more than 10% no-scaffold molecules, and several physicochemical endpoints, including Hydration energy and Boiling point, approached 50% (**Fig. 1a, Supplementary Table 2**). These results show that empty-scaffold representations are a recurrent feature of current molecular benchmark datasets.

In standard Scaffold Split, molecules with empty Bemis–Murcko scaffold representations are assigned to a single empty-scaffold bucket and are therefore treated as one split unit. Unlike scaffold-bearing groups, this bucket is defined by the absence of a retained scaffold representation rather than by a shared scaffold core. When no-scaffold molecules account for a substantial fraction of a dataset, this grouping can concentrate them in a single split branch and reduce their representation in the others (**Fig. 1b**). These observations identify empty-scaffold allocation as a potential source of split imbalance in scaffold-based evaluation.

.

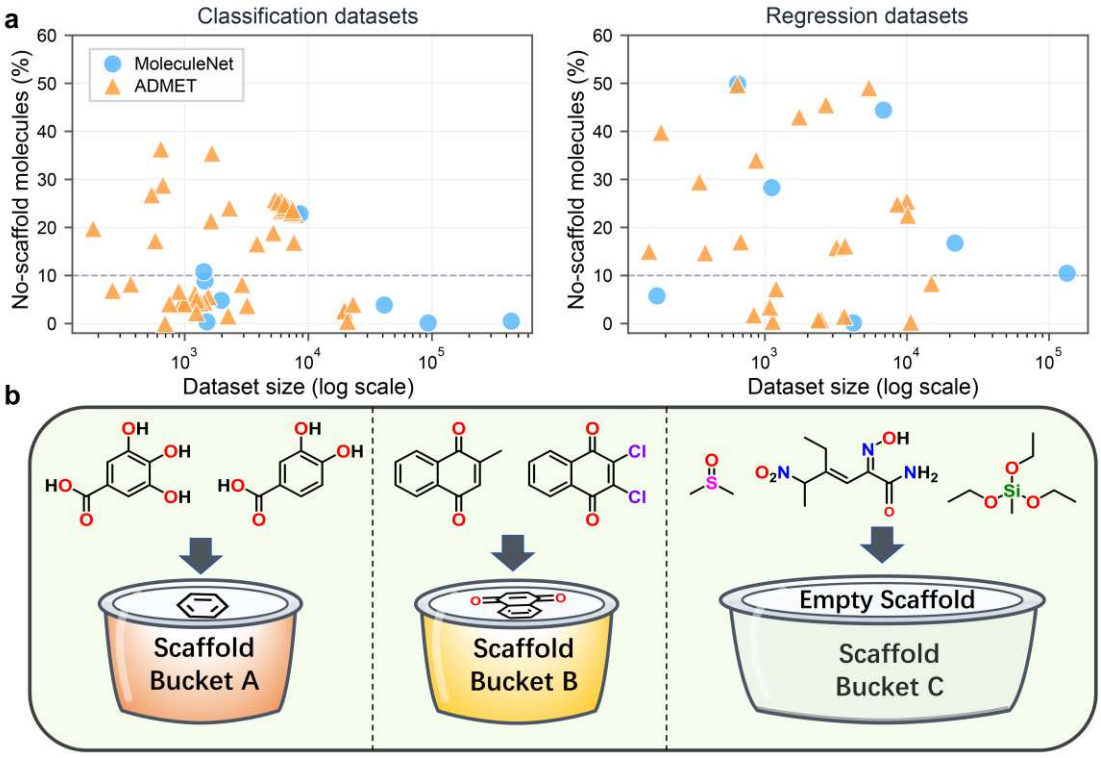


Figure 1. No-scaffold prevalence and empty-scaffold grouping in molecular benchmarks. a, Proportion of no-scaffold molecules across MoleculeNet and ADMET datasets. **b,** Schematic of empty-scaffold grouping under standard Scaffold Split. Scaffold-bearing molecules form groups based on non-empty Bemis–Murcko scaffolds. In contrast, molecules with empty scaffold representations are assigned to a single empty-scaffold bucket and treated as one split unit.

No-scaffold molecules are diffuse and weakly supported

We next examined whether no-scaffold molecules form a structurally coherent group. This analysis retained 52 datasets with at least 8% no-scaffold molecules and covered 110 tasks after expanding multi-endpoint datasets, including 73 classification tasks and 37 regression tasks.

Across these datasets, the median within-group pairwise Tanimoto similarity of no-scaffold molecules was 0.110, close to the background level of all scaffold-bearing molecules (0.098) and markedly lower than the within-class similarity of true scaffold classes (0.303; **Fig. 2a**). Cohesion Gini showed a consistent pattern. The median Cohesion Gini of no-scaffold molecules was 0.208, higher than that of the scaffold-

bearing background (0.137) and true scaffold classes (0.083; **Fig. 2a**). This indicates that similarity within the no-scaffold set was unevenly distributed rather than uniformly shared across the group. Thus, the empty-scaffold bucket was structurally diffuse rather than scaffold-like.

No-scaffold molecules were also weakly supported by the scaffold-bearing chemical space. The median nearest-neighbor similarity was 0.573 within true scaffold classes and 0.477 between different scaffold classes, whereas the nearest-neighbor similarity from no-scaffold molecules to the scaffold-bearing space was 0.300 (**Fig. 2a**). Thus, no-scaffold molecules showed lower nearest-neighbor support from scaffold-bearing molecules than did ordinary scaffold-bearing reference groups.

The no-scaffold subset also showed task-dependent differences in label and target space. Across 37 regression tasks, the median target-distribution shift was |Hedges' g| = 0.312, with a median KS statistic of 0.173. Nearly half of the regression tasks had |Hedges' g| ≥ 0.5, and eight reached |Hedges' g| ≥ 0.8 (**Fig. 2b, Supplementary Table 3**). Among 73 classification tasks, more than half had an absolute positive-rate difference greater than 10%, and approximately one quarter exceeded 20%. Bernoulli Jensen–Shannon divergence increased with |Δpositive rate| (**Fig. 2b, Supplementary Table 4**). Fingerprint-based UMAP visualizations further showed that no-scaffold molecules frequently occupy regions separated from scaffold-bearing molecules in representation space (**Supplementary Fig. 7**). Together, these analyses show that no-scaffold molecules differ from scaffold-bearing molecules in structural cohesion, chemical-space support and task-dependent label or target distributions.

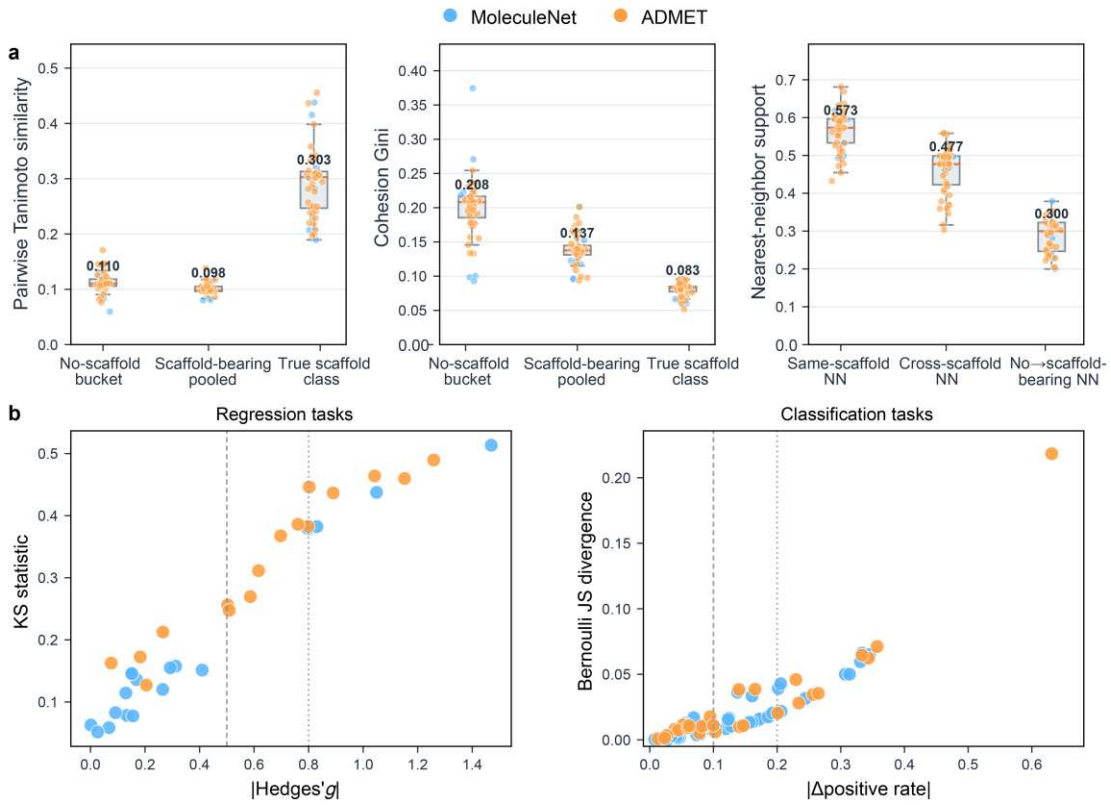


Figure 2. Chemical-space properties and task-dependent distribution shifts of no-scaffold molecules. a, Structural-space characteristics of no-scaffold molecules and reference groups, including within-group pairwise Tanimoto similarity, Cohesion Gini, and nearest-neighbor similarity. **b,** Distribution shifts of the no-scaffold subset in target and label space. The left panel shows |Hedges' g| and KS statistics for regression tasks; the right panel shows |Δpositive rate| and Bernoulli Jensen–Shannon divergence for classification tasks.

Splitters vary in no-scaffold training coverage

We next compared four representative splitting strategies: Bemis–Murcko Scaffold Split and three similarity- or scenario-aware alternatives, namely UMAP-based clustering split, DataSAIL and LoHi. These alternatives were proposed to address limitations of conventional splitting, including similarity-induced leakage and mismatch with practical drug-discovery scenarios. Here, we asked whether they also preserve no-scaffold training-side representation.

For multi-fold methods, including Scaffold Split and UMAP-based clustering split, we assessed minimum training-side coverage. For each dataset, we selected the fold with the lowest no-scaffold proportion in the training set and defined it as the train-depleted representative fold. This fold was used to assess worst-case training-side coverage rather than average fold composition, because a splitter can appear balanced on average while still producing a valid fold in which no-scaffold molecules are depleted from training and concentrated in evaluation branches.

Under this criterion, standard Scaffold Split produced the strongest depletion. In all 52 datasets, the train-depleted representative fold contained zero no-scaffold molecules in the training set, whereas the maximum no-scaffold proportion in validation or test reached a median of 78.2%. Thus, standard Scaffold Split may allocate the empty-scaffold bucket to evaluation while leaving the training set without no-scaffold coverage (**Fig. 3a**).

UMAP-based clustering split does not use Bemis–Murcko scaffold labels, but it also did not provide stable minimum coverage. Among the 48 datasets for which the procedure successfully ran, the median no-scaffold proportion in the train-depleted representative fold was 7.9%, below the median overall no-scaffold proportion of 23.3%. The median train-to-overall no-scaffold ratio was 0.32. In 7 of 48 datasets, the training set contained zero no-scaffold molecules; in 23 of 48 datasets, the training no-scaffold proportion was below 5%; and in 33 of 48 datasets, the train-to-overall ratio was below 0.5. Meanwhile, the median maximum no-scaffold proportion across validation and test was 94.2%, with 40 of 48 datasets exceeding 80%. Thus, continuous-representation clustering may still generate valid folds with low no-scaffold training coverage and high evaluation-side exposure (**Fig. 3a**).

For single-split methods, DataSAIL and LoHi, we evaluated their original train/validation/test outputs. DataSAIL also failed to consistently preserve no-scaffold training-side representation. The median overall no-scaffold proportion was 23.6%, whereas the median training proportion was 14.7%. In contrast, the median maximum no-scaffold proportion across validation and test was 81.6%, and the median train-to-overall ratio was 0.64. Although DataSAIL avoided the systematic training-side

clearance observed in Scaffold Split, 17 of 52 datasets had a train-to-overall ratio below 0.5, and 9 of 52 had a training no-scaffold proportion below 5%. Several datasets showed pronounced depletion: Bioconcentration factor had 17.0% no-scaffold molecules overall but only 2.1% in training and 82.0% in test; pK_b had 16.1% overall, 2.8% in training, and 95.4% in validation. Ames, Respiratory tox, and Eye irritation showed similar patterns, with training no-scaffold proportions below 4% and evaluation-side proportions of 83.0–95.5%. These results show that DataSAIL does not reliably maintain no-scaffold coverage proportional to the full dataset (**Fig. 3b, Supplementary Table 5**).

LoHi produced valid splits for only 24 of the 52 analyzed datasets. In these successful runs, its global no-scaffold allocation was more training-preserving: the median overall no-scaffold proportion was 25.0%, the median training proportion was 27.3%, and the median maximum evaluation-side proportion was 16.1%. None of the 24 datasets had a training no-scaffold proportion below 5%, and only one had a train-to-overall ratio below 0.5. However, local depletion remained. In Biodegradation, the overall no-scaffold proportion was 33.3%, but the training proportion dropped to 13.2%, while validation and test increased to 88.8% and 86.4%. In Hydration energy, the overall proportion was 48.7%, the training proportion was 25.9%, and the test proportion reached 87.7%. In pK_a, pK_b, and Marine tox (Fathead minnow), the maximum evaluation-side no-scaffold proportion ranged from 42.9% to 73.2%, exceeding training proportions of 8.5–20.1%. Thus, LoHi mitigated no-scaffold training-side depletion in most successful runs, but did not fully eliminate task-level imbalance (**Fig. 3b, Supplementary Table 5**).

Together, these results show that alternative splitting strategies differed in their handling of no-scaffold allocation, but none consistently guaranteed proportional no-scaffold training-side representation.

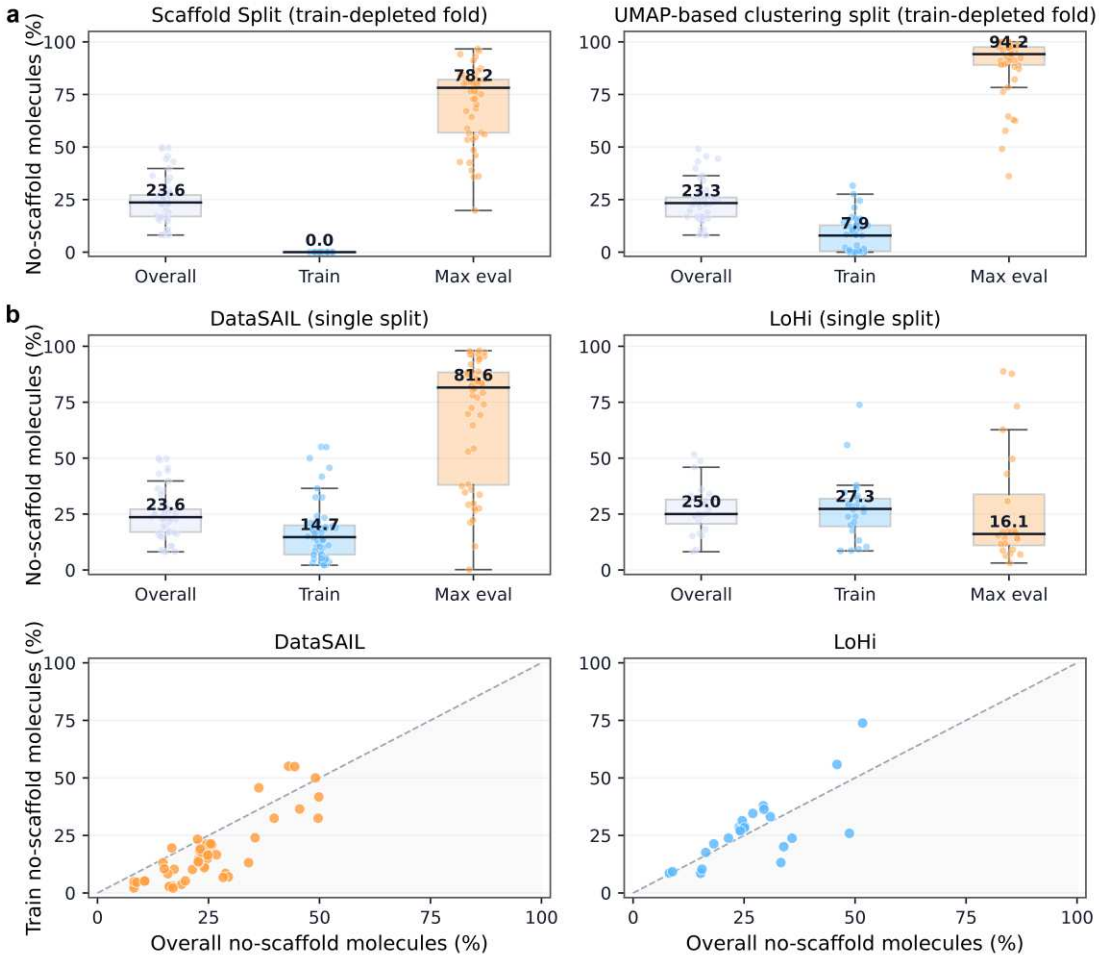


Figure 3. No-scaffold training-side underrepresentation under four splitting strategies. a, No-scaffold proportions in the full dataset, training set, and Max eval branch for the train-depleted representative folds of Scaffold Split and UMAP-based clustering split. Max eval denotes the higher no-scaffold proportion between validation and test. These folds quantify minimum-coverage allocation risk rather than average fold composition. **b,** Single-split results for DataSAIL and LoHi. The top row shows no-scaffold proportions in the full dataset, training set, and max eval branch. The bottom row compares no-scaffold proportions in the full dataset and training set; points below the diagonal indicate lower training-side representation than expected from the full dataset.

No-scaffold subsets reduce classification performance

We used Bemis–Murcko Scaffold Split for model training and evaluation because it produced a train-depleted fold in which the training set contained no no-scaffold

molecules while the test set contained the no-scaffold bucket. This fold directly tested model performance on no-scaffold molecules when this subgroup was absent from training. The remaining folds served as ordinary-fold references, enabling comparison within the same task, model and splitting protocol. We used normPR as the primary classification metric because it measures precision–recall ranking relative to the positive-rate baseline of each evaluated subset, enabling comparison across tasks and split branches with different class priors; metric-selection rationale and sensitivity analyses using AP and ROC-AUC are provided in **Supplementary Note 1**. The fold containing all no-scaffold molecules in the test set was defined as the no-scaffold fold, and performance was computed on the no-scaffold subset of that test set rather than on the entire test fold. The remaining folds were used as ordinary-fold references for the same model–task pair.

The analysis included seven representative baseline models. To reduce instability from small test subsets, we retained only tasks in which the no-scaffold test subset contained at least 10 positive and 10 negative samples. Ordinary folds were also included only when their test sets contained at least 10 samples from each class. This filtering retained 63 classification tasks, including 37 from MoleculeNet and 26 from ADMET, corresponding to 441 model–task evaluation points.

Using the median ordinary-fold normPR as the reference for each model–task pair, the no-scaffold test subset showed consistently lower classification performance. The median ordinary-fold normPR was 0.259 for MoleculeNet and 0.409 for ADMET. Relative to this baseline, the task-level mean ΔnormPR across the seven models ranged from −0.265 to −0.161 in MoleculeNet and from −0.273 to −0.218 in ADMET (**Fig. 4a**). These decreases were substantial relative to the ordinary-fold baselines.

The same pattern was evident at the evaluation-point level. In MoleculeNet, 89.2% of evaluation points (231/259) fell below their corresponding ordinary-fold median baseline; in ADMET, the proportion was 91.8% (167/182). A within-fold comparison gave the same conclusion. When no-scaffold and scaffold-bearing test subsets were compared directly within the no-scaffold fold, the no-scaffold subset had lower normPR in 86.5% of MoleculeNet evaluation points (224/259) and 87.4% of ADMET

evaluation points (159/182) (**Fig. 4b**). Thus, the degradation observed in the no-scaffold fold cannot be explained solely by fold-level difficulty, but reflects a subpopulation-specific performance loss.

Task-level results further showed that the performance decrease was not restricted to a single model family. Among the 63 classification tasks, 44 showed below-baseline performance for all seven models on the no-scaffold test subset, including 24 MoleculeNet tasks and 20 ADMET tasks. Task-level ΔnormPR values were also correlated across models, with mean pairwise Spearman correlations of 0.649 in MoleculeNet, 0.844 in ADMET and 0.735 after combining both benchmarks (**Fig. 4c**). These results show that reduced classification performance on no-scaffold subsets was recurrent across benchmarks, tasks and model types.

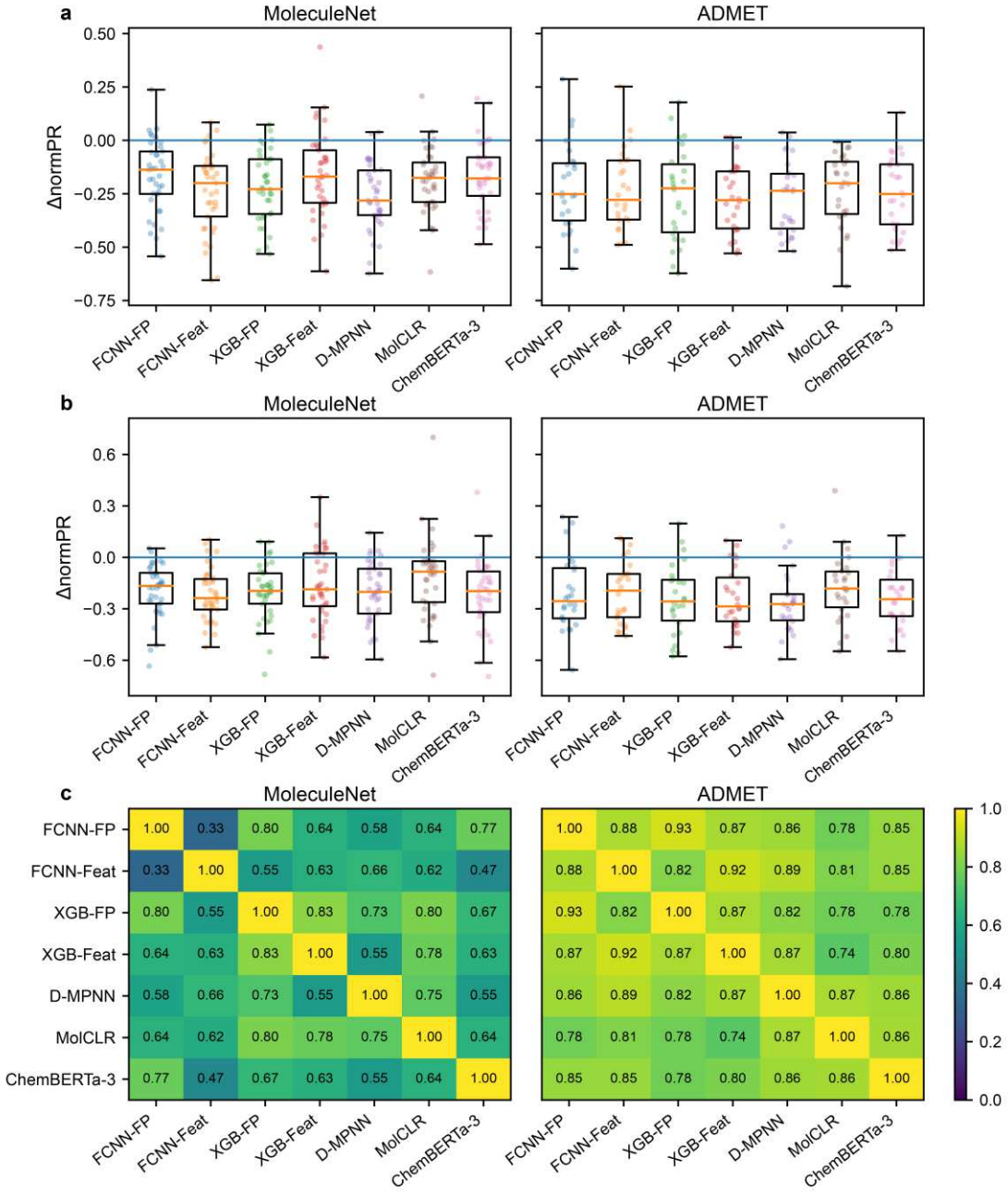


Figure 4. Classification performance on no-scaffold test subsets. a, Task-level ΔnormPR distributions for the no-scaffold test subset relative to ordinary folds. **b,** Within-fold normPR differences between no-scaffold and scaffold-bearing test subsets in the no-scaffold fold. **c,** Pairwise Spearman correlation matrix of task-level ΔnormPR across models.

Ranking performance shifts toward the prior baseline

We further examined the full distribution of normPR to assess whether the decrease in

classification performance was concentrated near the prior baseline. Because normPR is normalized against the test-set positive rate, values close to zero indicate little ranking information beyond the prior baseline.

Across 441 model–task evaluation points, the no-scaffold test subset showed a marked downward shift in normPR. In MoleculeNet, the median normPR decreased from 0.259 in ordinary folds to 0.035 on the no-scaffold test subset of the no-scaffold fold. In ADMET, the corresponding median decreased from 0.409 to 0.119 (**Fig. 5a**). The upper part of the no-scaffold distribution was also lower than the ordinary-fold reference. In MoleculeNet, the upper quartile of normPR on the no-scaffold test subset was 0.100, below the ordinary-fold median of 0.259. In ADMET, the upper quartile was 0.383, also below the ordinary-fold median of 0.409. Thus, even the better-performing no-scaffold evaluation points often remained below typical ordinary-fold performance.

Empirical cumulative distribution functions further summarized this distributional shift. Using $\tau = 0.10$ as a reference threshold, we calculated the normPR level that ordinary folds would need to reach to cover the same cumulative fraction observed for the no-scaffold test distribution at that threshold. The resulting scale factors were 3.92 for MoleculeNet and 3.70 for ADMET. When the threshold was tightened to $\tau = 0.05$, the factors increased to 5.84 and 5.17, respectively (**Fig. 5b**). These scale factors show that the no-scaffold distribution was more concentrated at low normPR values than the ordinary-fold distribution.

Near-baseline normPR values were also more frequent in the no-scaffold test subset. In MoleculeNet, 57.1% and 40.2% of no-scaffold evaluation points had normPR ≤ 0.05 and normPR ≤ 0.02, respectively, compared with 6.9% and 4.5% in ordinary folds. In ADMET, the corresponding proportions were 28.0% and 12.1% for the no-scaffold test subset, compared with 1.7% and 0.6% in ordinary folds (**Supplementary Fig. 1**). Relative to ordinary folds, these near-baseline regions were enriched by 8.2-fold and 8.9-fold in MoleculeNet, and by 16.1-fold and 20.0-fold in ADMET. Thus, the no-scaffold test subset was enriched for low-information ranking failures that were rare in ordinary folds.

The same pattern was especially pronounced in the Tox21 sub-benchmark. Across its 12 classification tasks, the median normPR on the no-scaffold test subset was 0.018, compared with 0.303 in ordinary folds. The ECDF scale factors were 5.81 at $\tau = 0.10$ and 9.32 at $\tau = 0.05$ (**Supplementary Fig. 2**), indicating pronounced concentration near the prior-baseline region.

Together, these results show that no-scaffold classification degradation reflects not only lower median normPR, but also compression of ranking performance toward the prior-baseline regime.

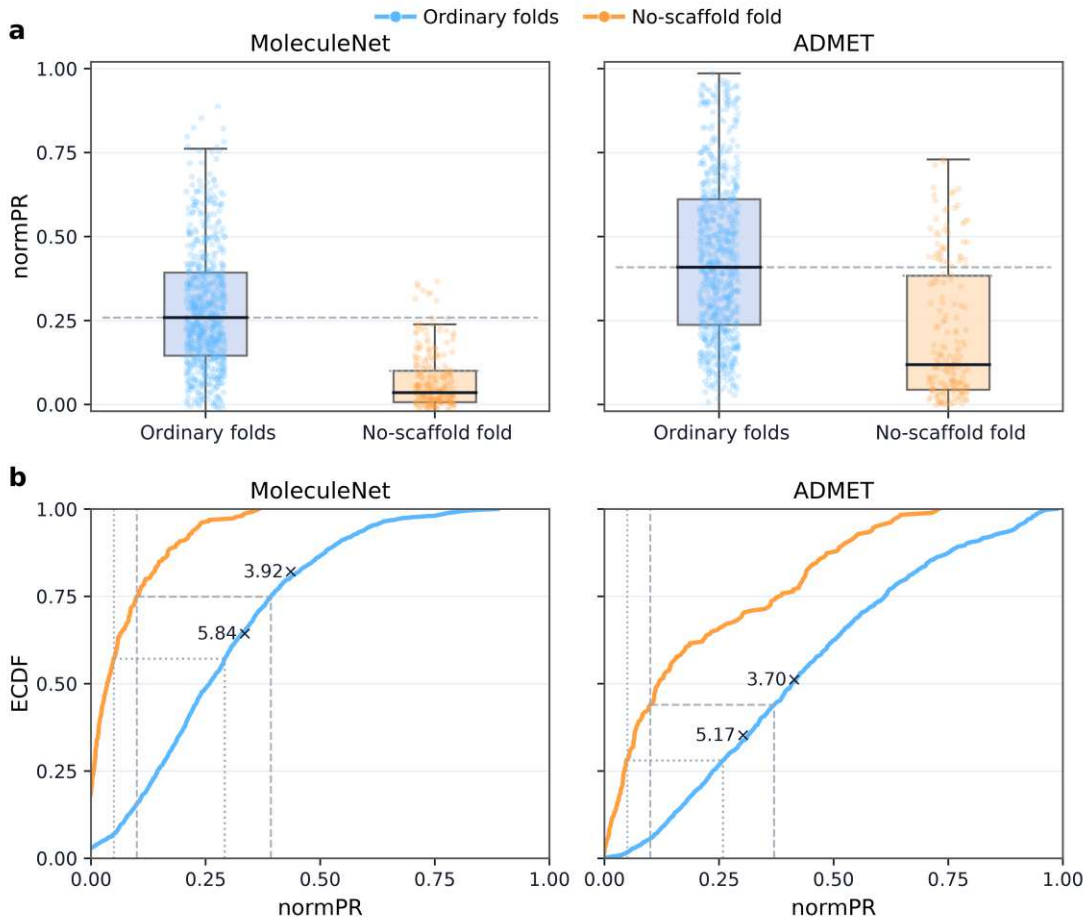


Figure 5. Ranking performance on no-scaffold test subsets. a, normPR distributions for ordinary folds and the no-scaffold test subset of the no-scaffold fold. **b,** Empirical cumulative distribution functions for the two test distributions. Dashed lines indicate the reference thresholds $\tau = 0.10$ and $\tau = 0.05$.

Regression errors increase on no-scaffold subsets

We used the same Scaffold Split setting for regression tasks, comparing the no-scaffold test subset in the train-depleted fold with ordinary-fold references under the same model–task pair. Because regression tasks differ in units and target magnitudes, we used relMAE as the primary regression metric to quantify numerical error relative to the target scale of each evaluated subset; metric-selection rationale and sensitivity analyses using MAE, RMSE, R^2, Pearson r and Spearman ρ are provided in **Supplementary Note 1**. Scale differences can also occur within the same task, because no-scaffold molecules may differ from scaffold-bearing or ordinary-fold reference molecules in target-value distribution and target scale. We therefore interpreted relMAE inflation within each model–task pair and decomposed it into the absolute MAE ratio and target-scale ratio. A relMAE inflation value greater than 1 indicates higher scale-adjusted error on the no-scaffold test subset than on the corresponding ordinary-fold reference.

Across 259 model–task evaluation points, the no-scaffold test subset showed consistent relMAE elevation relative to the ordinary-fold reference. The median relMAE increased from 0.190 in ordinary folds to 0.256 on the no-scaffold test subset, corresponding to an overall 1.35-fold increase. By benchmark, the median relMAE increased from 0.042 to 0.076 in MoleculeNet and from 0.270 to 0.464 in ADMET, corresponding to 1.83-fold and 1.72-fold increases, respectively (**Fig. 6a**). The increase was accompanied by enrichment of high-multiple relMAE inflation: 44.8% of evaluation points reached at least 2-fold inflation, and 22.4% reached at least 3-fold inflation (**Fig. 6b**).

This high-multiple inflation was far more frequent than ordinary fold-to-fold variability. We used pairwise ordinary fold-to-fold relMAE ratios as a background reference and compared them with relMAE inflation on the no-scaffold test subset. At the 2-fold threshold, the tail probability for the no-scaffold test subset was 45.6% in MoleculeNet, compared with 1.76% for the ordinary-fold background. In ADMET, the corresponding values were 43.8% and 1.32%. At the 3-fold threshold, the no-scaffold and ordinary-fold background probabilities were 30.6% versus 0.08% in MoleculeNet

and 11.6% versus 0.09% in ADMET. Relative to the ordinary-fold background, the 2-fold tail probability was approximately 26-fold and 33-fold higher in MoleculeNet and ADMET, respectively; the 3-fold tail probability was approximately 378-fold and 132-fold higher (**Fig. 6b**).

High relMAE inflation was not uniformly distributed across regression endpoints. Among 37 regression tasks, 27 showed higher relMAE than the ordinary-fold baseline for all seven models on the no-scaffold test subset. When requiring a task-level median relMAE inflation of at least 2-fold, 17 tasks met the criterion (**Fig. 6c**).

Model-level summaries showed different degrees of relMAE inflation across models. FCNN-feature showed the highest median relMAE inflation, with values of 3.68 in MoleculeNet and 2.26 in ADMET. D-MPNN also showed substantial inflation, with corresponding values of 2.98 and 2.18. Pretrained models showed lower median inflation, but their median inflation ratios remained above 1: MolCLR had median inflation values of 1.19 and 1.53 in MoleculeNet and ADMET, respectively, while ChemBERTa-3 had values of 1.31 and 1.62 (**Supplementary Fig. 3**). After excluding FCNN-feature, the remaining six models still showed a combined median inflation ratio of 1.71; 89.2% of evaluation points had higher relMAE on the no-scaffold test subset, 40.1% reached at least 2-fold inflation and 17.1% reached at least 3-fold inflation.

Error decomposition separated changes in absolute prediction error from changes in target scale. In MoleculeNet, the target-scale ratio for the no-scaffold test subset was approximately 1.01, whereas the absolute MAE ratio was 1.76. In ADMET, the target-scale ratio was approximately 0.76, while the absolute MAE ratio was 1.32 (**Fig. 6d**). Thus, MoleculeNet inflation was mainly associated with increased absolute prediction error, whereas ADMET inflation reflected both absolute error growth and target-scale contraction.

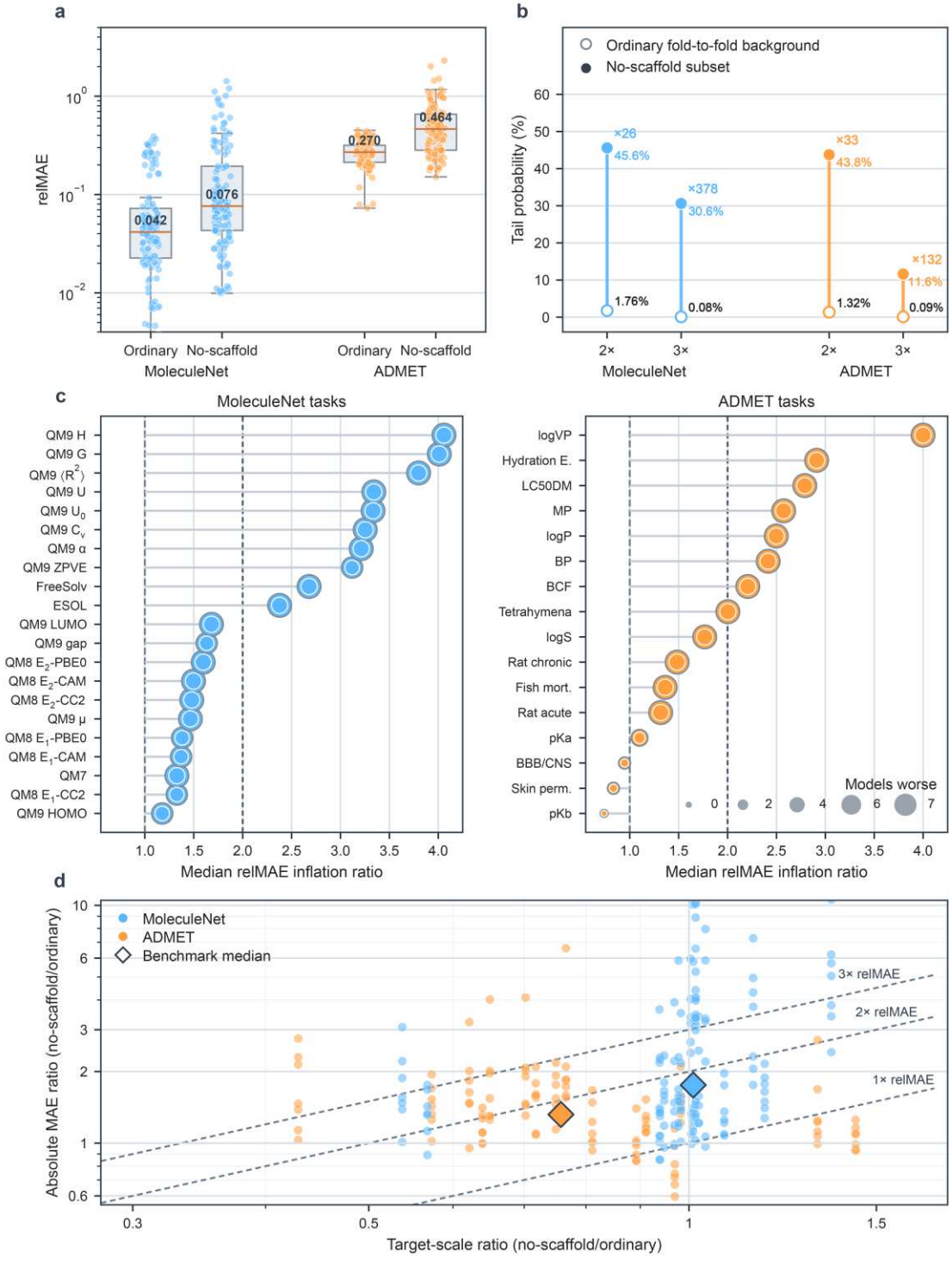


Figure 6. Regression error on no-scaffold test subsets. a, relMAE distributions for ordinary folds and the no-scaffold test subset. **b,** Tail probabilities of 2-fold and 3-fold relMAE inflation and their enrichment relative to ordinary fold-to-fold background variability. **c,** Task-level median relMAE inflation. d, Decomposition of relMAE inflation into absolute MAE ratio and target-scale ratio.

Prediction-scale compression tracks classification decline

We next examined task-level associations between classification performance changes and candidate diagnostic variables, including training-support loss, prior shift, calibration drift and prediction-scale compression. Analyses were performed at the task-level median scale, with ΔnormPR used as the classification failure metric.

Among the 63 retained classification tasks, the no-scaffold fold was often accompanied by stronger class imbalance. In MoleculeNet, 62.2% of the 37 tasks became more imbalanced in the no-scaffold fold, with a median imbalance amplification ratio of 1.91. In ADMET, the corresponding proportion was 73.1%, with a median ratio of 2.70 (**Supplementary Table 12**). However, this prior shift was weakly associated with task-level ΔnormPR. The Spearman correlation between |Δpositive rate| and ΔnormPR was close to zero in MoleculeNet ($\rho = -0.007$) and remained weak in ADMET ($\rho = -0.072$). Calibration metrics showed the same pattern: correlations of ΔBrier and ΔECE with ΔnormPR were weak, with all four absolute correlations across the two benchmarks no greater than 0.153 (**Fig. 7a; Supplementary Figs. 4b–d; Supplementary Table 13**).

In contrast, prediction-scale compression was more strongly associated with ΔnormPR. The task-level median collapse strength was 1.48 in MoleculeNet and 1.84 in ADMET (**Supplementary Table 12**). Collapse strength correlated negatively with ΔnormPR, with $\rho = -0.311$ in MoleculeNet ($p = 0.061$) and $\rho = -0.558$ in ADMET ($p = 0.003$; **Fig. 7a, b; Supplementary Table 13**). After controlling for |Δpositive rate|, ΔBrier and ΔECE, the partial Spearman correlations were −0.475 in MoleculeNet ($p = 0.0045$) and −0.782 in ADMET ($p = 1.04 \times 10^{-5}$; **Fig. 7b; Supplementary Fig. 4e; Supplementary Table 13**).

Training-support similarity also decreased in the no-scaffold fold, from 0.435 to 0.263 in MoleculeNet and from 0.445 to 0.308 in ADMET (**Fig. 7b**, top; **Supplementary Table 12**). Using training-support similarity below 0.30 as a low-support threshold, the fraction of low-support tasks increased from 0 to 78.4% in MoleculeNet and from 11.5% to 46.2% in ADMET (**Supplementary Table 12**). Support loss was weakly associated with ΔnormPR in MoleculeNet ($\rho = -0.085$, $p =$

0.617) and more strongly associated in ADMET (ρ = −0.464, p = 0.017; **Fig. 7a; Supplementary Fig. 4a; Supplementary Table 13**). Together, these results indicate that classification degradation is better aligned with prediction-scale compression than with prior shift or calibration drift, against a background of reduced training support.

Support loss associates with regression error inflation

We then examined task-level associations between regression error inflation and candidate diagnostic variables, including training-support loss, target-distribution shift, absolute MAE growth, target-scale contraction and output-scale change. We used relMAE inflation as the regression failure metric.

For regression tasks, training-support similarity decreased in the no-scaffold fold. At the task-level median scale, training-support similarity decreased from 0.533 to 0.393 in MoleculeNet and from 0.429 to 0.261 in ADMET (**Supplementary Table 14**). Support loss was positively associated with relMAE inflation, with ρ = 0.351 in MoleculeNet (p = 0.119) and ρ = 0.782 in ADMET (p = 3.4 × 10−4). Support loss was also positively associated with ΔMAE in both benchmarks, with ρ = 0.538 in MoleculeNet (p = 0.0119) and ρ = 0.732 in ADMET (p = 0.0013; **Fig. 7c, d; Supplementary Fig. 5a, b; Supplementary Table 15**).

Across all 259 model–task evaluation points, evaluation points with more severe support loss showed higher relMAE inflation than those with lighter support loss, with median relMAE inflation of 2.305 versus 1.628. The corresponding 2-fold inflation rate was also higher in the more severe support-loss group (62.6% versus 35.1%), whereas the 3-fold inflation rate was similar between groups (20.9% versus 23.2%; **Fig. 7d; Supplementary Fig. 5e; Supplementary Table 14**).

Associations with target-distribution shifts were benchmark-dependent. In MoleculeNet, |Hedges' g| and KS statistic were weakly associated with relMAE inflation, with correlations of ρ = 0.023 (p = 0.920) and ρ = 0.194 (p = 0.399), respectively (**Fig. 7c; Supplementary Fig. 5c; Supplementary Table 15**). In ADMET, the corresponding correlations were stronger: ρ = 0.609 for |Hedges' g| (p = 0.012) and ρ = 0.544 for KS statistic (p = 0.029; **Fig. 7c; Supplementary Fig. 5d; Supplementary Table 15**). Consistent with this benchmark difference, the target-scale

ratio was approximately 1.01 in MoleculeNet and 0.758 in ADMET, whereas the MAE ratio was 1.59 in MoleculeNet and 1.45 in ADMET (**Supplementary Table 14**).

Output-scale change showed different patterns across benchmarks. The task-level median output-scale change was approximately 1.10 in MoleculeNet and 0.95 in ADMET (**Supplementary Table 14**). Its association with relMAE inflation was positive in MoleculeNet ($\rho = 0.548$, $p = 0.010$), but weak in ADMET ($\rho = 0.041$, $p = 0.880$; **Fig. 7c; Supplementary Table 15**). Thus, regression error inflation was more consistently associated with support loss than with output-scale change across benchmarks.

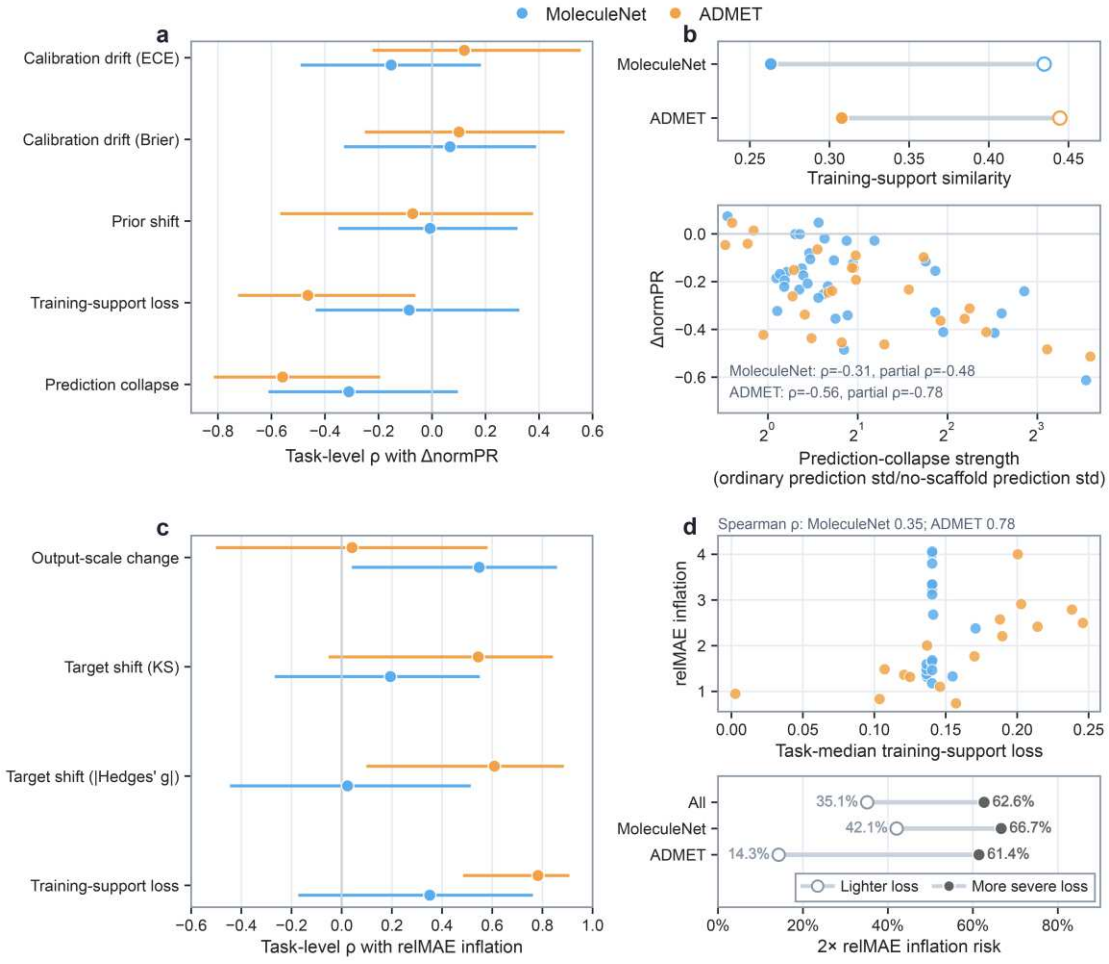


Figure 7. Diagnostic associations with no-scaffold performance changes. a, Task-level correlations between candidate diagnostic variables and ΔnormPR in classification tasks. **b,** Top: nearest-neighbor training-support similarity in ordinary folds and the no-scaffold fold. Bottom: association between prediction-scale compression and ΔnormPR. **c,** Task-level correlations between candidate diagnostic

variables and relMAE inflation in regression tasks. **d,** Top: association between support loss and relMAE inflation. Bottom: 2-fold relMAE inflation rate after stratification by support-loss severity.

ESA restores no-scaffold training representation

We used DataSAIL as the backbone for ESA to test whether no-scaffold allocation could be repaired after a leakage-aware split had already been generated, rather than merely correcting the empty-scaffold bucket artifact of standard Scaffold Split. DataSAIL was suitable because it provides a similarity-aware split with leakage-oriented diagnostics and generated valid splits for all analyzed datasets. However, as shown above, it still left no-scaffold molecules underrepresented in training and enriched in validation or test branches. ESA therefore kept the DataSAIL assignments of scaffold-bearing molecules fixed, partitioned no-scaffold molecules into local structural micro-clusters, and reassigned these micro-clusters across training, validation and test branches.

A preliminary sweep over five predefined ESA configurations was performed on a fixed tuning subset using only split-composition and train-facing similarity-exposure diagnostics; the main-text analysis used the selected ESA-TrainSupport configuration (**Supplementary Table 16 and 17**).

Compared with raw DataSAIL, ESA increased no-scaffold representation in the training set. Support recovery increased from 0.6445 to 1.1141 (**Fig. 8a**). The no-scaffold proportion in the training set increased from 0.1469 to 0.2602, whereas the corresponding proportions in validation and test decreased from 0.7323 and 0.3394 to 0.1148 and 0.1147(**Supplementary Fig. 6a**). ESA increased support recovery in 48 of 52 datasets.

ESA also reduced the concentration of no-scaffold molecules in validation and test relative to training. The sink ratio decreased from 5.2764 under raw DataSAIL to 0.4335 after ESA, with reductions observed in 50 of 52 datasets (**Fig. 8b**). These changes indicate that ESA redistributed no-scaffold molecules across split branches rather than simply moving them into the training set.

ESA did not restore training-side representation by removing no-scaffold molecules from evaluation. In the final repaired configuration, all 52 datasets retained no-scaffold molecules in both validation and test sets. At the same time, train-facing similarity exposure decreased. The median L_{norm} from the no-scaffold validation branch to the full training set decreased from 0.0771 to 0.00935, and the corresponding value for the no-scaffold test branch decreased from 0.0313 to 0.00408 (**Fig. 8c**). Leak rate at the 0.4 similarity threshold also decreased, from 0.3448 to 0.2609 for validation and from 0.1921 to 0.0909 for test (**Supplementary Fig. 6d**).

After merging the repaired no-scaffold allocation with the fixed scaffold-bearing backbone, repair-induced similarity exposure remained lower for no-scaffold validation/test branches than for scaffold-bearing validation/test branches. The median ΔL_{norm} values for scaffold-bearing validation and test branches were 0.0183 and 0.0189, respectively, whereas the corresponding values for no-scaffold validation and test branches were 0.0053 and 0.0025 (**Fig. 8d**). Across all 52 datasets, ΔL_{norm} for the no-scaffold validation/test branches was lower than that for the corresponding scaffold-bearing branches.

ESA had minimal impact on sample retention and global split quality. Only QM9 lost two molecules during repair, while the other 51 datasets retained all molecules. This corresponded to approximately 7.0×10^{-6} of all molecules and 3.4×10^{-5} of all no-scaffold molecules. Relative to raw DataSAIL, the median full-split L_{norm} decreased from 0.2538 to 0.2416 (**Supplementary Fig. 6b**). Together, these results show that ESA repaired no-scaffold allocation imbalance while retaining evaluation exposure and controlling train-facing similarity exposure.

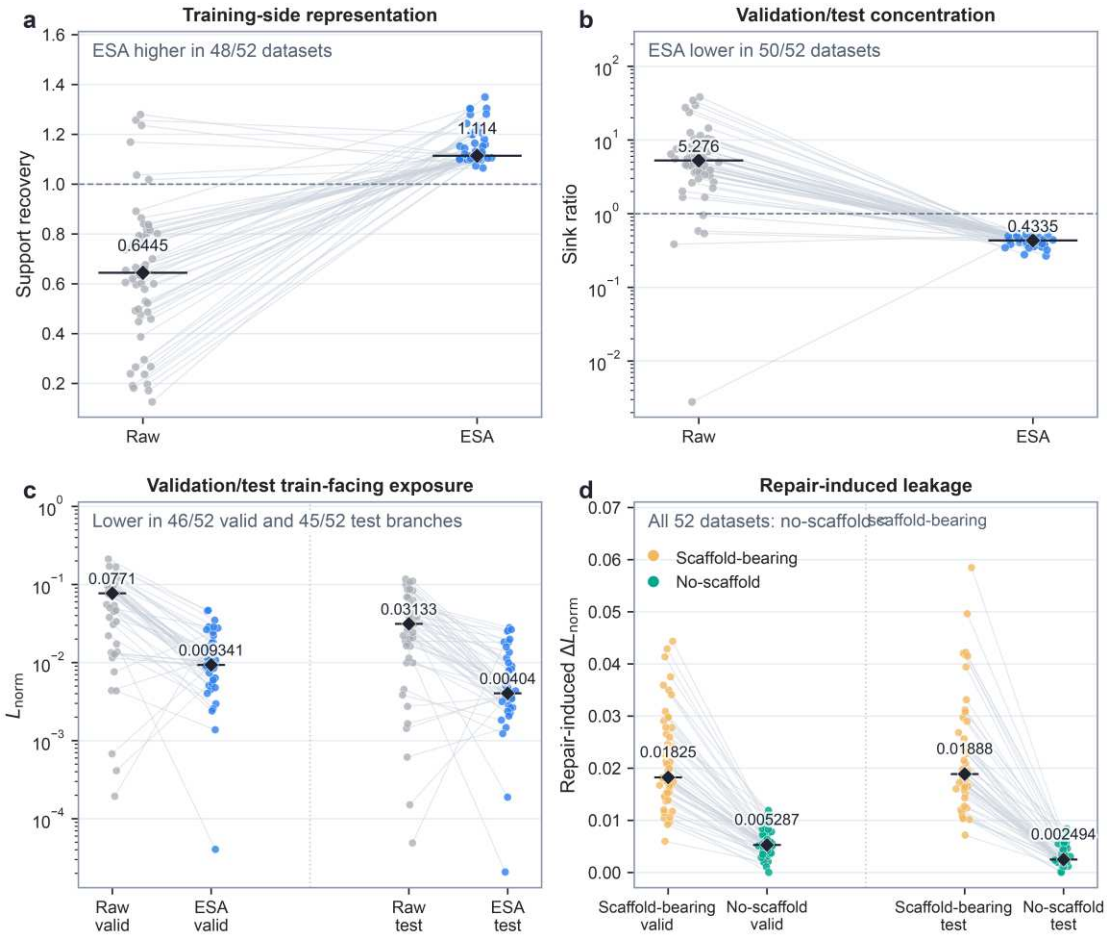


Figure 8. ESA restores no-scaffold training-side representation under controlled similarity exposure. a, Support recovery under raw DataSAIL and ESA. **b,** Sink ratio under raw DataSAIL and ESA. **c,** L_{norm} from no-scaffold validation and test branches to the full training set. **d,** Repair-induced ΔL_{norm} for scaffold-bearing and no-scaffold validation/test branches.

Discussion

This study shows that empty-scaffold allocation can confound the interpretation of scaffold-based molecular evaluation. Scaffold splits are commonly treated as tests of generalization to unseen structural series, but this interpretation assumes that each split unit corresponds to a chemically coherent molecular framework. Molecules with empty Bemis–Murcko scaffold representations violate this assumption. They are grouped not because they share a ring–linker core, but because scaffold extraction returns no retained scaffold. The empty-scaffold bucket is therefore an absence-defined residual

group rather than an ordinary scaffold class. Its allocation can cause an apparent scaffold holdout to reflect not only scaffold novelty, but also training-side underrepresentation of a structurally diffuse and weakly supported residual population. This finding does not invalidate scaffold-based evaluation. Scaffold splits remain an important alternative to random splits, which can place close structural analogues on both sides of the split and produce overly optimistic estimates of generalization. The issue identified here is more specific: scaffold-based evaluation can be confounded when a common, weakly cohesive residual subgroup is largely absent from training but enriched in validation or test sets. The central benchmark question is therefore not whether no-scaffold molecules should appear in evaluation, but whether their evaluation exposure is supported by adequate training-side representation.

The comparison across splitting strategies shows that leakage control and support coverage are distinct benchmark properties. Scaffold Split exposes the most direct form of the problem, because empty-scaffold molecules are compressed into a single split unit and can be moved away from training as a group. UMAP-based clustering, DataSAIL and LoHi do not use the same scaffold grouping rule, but still do not guarantee adequate no-scaffold training-side representation, consistent with the distinct chemical-space localization and weak training support of no-scaffold molecules (**Fig. 2a and Supplementary Fig. 7**). A split can therefore reduce train–test similarity or create a more challenging evaluation set while still amplifying training-side underrepresentation of a weakly supported subgroup.

The downstream effects of this imbalance depended on the prediction task. In classification, no-scaffold-enriched evaluation under depleted training support was associated with compression of ranking performance toward the prior-only regime, indicating loss of discriminative ordering rather than a simple class-imbalance effect. In screening or prioritization settings, this means that model scores may carry limited ranking information beyond class prevalence within the affected subset. In regression, the same allocation imbalance was associated with enrichment of high-multiple relMAE inflation, indicating numerical extrapolation instability under reduced support. Aggregate scaffold-split metrics can therefore obscure subpopulation-specific failures,

including near-prior ranking performance in classification and severe relative error inflation in regression.

The mechanistic analyses separate the evaluation background from the immediate failure phenotype. Reduced training support defines the background created by no-scaffold allocation, whereas prediction-score compression and error inflation describe how models respond to that background. In classification, prediction-score compression tracked performance degradation more closely than prior shift or calibration drift. In regression, support loss was associated with absolute and relative error inflation, with additional amplification from target-scale and target-distribution differences in some ADMET endpoints. These analyses are diagnostic rather than causal, because they are based on associations across tasks and model–task pairs rather than controlled interventions on training support. Nevertheless, they show that no-scaffold allocation is not merely a bookkeeping artifact; it can alter the support structure of the evaluation problem.

ESA should be understood in this framework. It is an allocation-level diagnostic and repair procedure, not evidence that downstream predictive performance will necessarily improve in all settings. By preserving the scaffold-bearing backbone and reallocating only no-scaffold molecules, ESA restores training-side representation while retaining no-scaffold molecules in validation and test sets. Its purpose is to separate scaffold-level generalization from representation-driven loss of training support. Thus, ESA illustrates a broader principle: evaluation difficulty and training-side representation should be reported as distinct properties of a benchmark split.

Several aspects of this work define directions for future study. The performance analyses use no-scaffold-enriched folds as a diagnostic stress condition to characterize the consequences of training-side depletion. Future benchmark designs that independently vary training support, scaffold novelty, label or target shift, and evaluation enrichment would help separate the contribution of each factor. More broadly, molecular benchmark evaluation should characterize not only similarity between evaluation and training molecules, but also whether weakly supported molecular subgroups have been systematically excluded from the training side.

At minimum, benchmark reports should separate three quantities: how often no-scaffold molecules occur, how much training-side support they receive, and how strongly they are enriched in validation or test branches. Nearest-neighbor training support and train-facing similarity exposure provide complementary diagnostics for determining whether this enrichment reflects leakage, under-support or both. Making these quantities explicit would help distinguish scaffold-level generalization from support-driven extrapolation failure and improve the interpretability of molecular benchmark evaluation.

Methods

Molecular standardization and scaffold extraction

We used molecular machine-learning benchmarks from MoleculeNet and DeepPK/ADMET[6, 7]. Molecules were provided as SMILES strings and processed with RDKit for parsing and standardization. Invalid SMILES were excluded. Duplicate molecules were resolved based on standardized canonical SMILES[19, 20, 21, 22].

Bemis–Murcko scaffolds were extracted using RDKit. For molecule m_i, the scaffold representation was denoted as

$$S_i = \mathrm{BM}(m_i) \tag{1}$$

A molecule was defined as no-scaffold if $S_i = \emptyset$ and scaffold-bearing otherwise. In standard Scaffold Split, all molecules with empty Bemis–Murcko scaffold representations were assigned to the same empty-scaffold bucket.

Molecular fingerprints and similarity analysis

Molecular structures were represented by Morgan fingerprints[23, 24]. Unless otherwise specified, fingerprints were generated with radius 2, bit length 2048, and chirality retained. The fingerprint of molecule m_i was denoted as f_i. The Tanimoto similarity[25, 26] between molecules m_i and m_j was defined as

$$T(i,j) = \frac{|f_i \cap f_j|}{|f_i \cup f_j|} \tag{2}$$

For a molecular set G, the within-group pairwise similarity was defined as

$$T_G = \{T(i,j): m_i, m_j \in G,\ i < j\} \tag{3}$$

For true scaffold classes, within-class similarity was first averaged within each dataset with scaffold class size as the weight, and then aggregated across datasets.

Cohesion Gini[27] was used to quantify the unevenness of pairwise similarity contributions within a group. Given $T_G = \{s_1, \ldots, s_n\}$,

$$\mathrm{Gini}(G) = \frac{\sum_{a=1}^{n}\sum_{b=1}^{n}|s_a - s_b|}{2n\sum_{a=1}^{n} s_a} \tag{4}$$

For molecule m_i and reference set R, nearest-neighbor similarity was defined as

$$\mathrm{NN}(m_i, R) = \max_{m_j \in R} T(i,j) \tag{5}$$

For a molecular set A, training-support similarity relative to R was defined as

$$\mathrm{Support}(A,R) = \mathrm{median}_{m_i \in A}\,\mathrm{NN}(m_i, R) \tag{6}$$

In model diagnostic analyses, support loss was defined as the difference in training-support similarity between ordinary folds and the no-scaffold fold:

$$\text{Support loss} = \mathrm{Support}_{\mathrm{ordinary}} - \mathrm{Support}_{\mathrm{no-scaffold}} \tag{7}$$

Distribution-shift analysis

For regression tasks, distributional differences between no-scaffold and scaffold-bearing molecules were quantified using |Hedges' g|[28, 29] and the two-sample Kolmogorov–Smirnov statistic (KS)[30, 31]:

$$g = J \cdot \frac{\bar{y}_{\mathrm{ns}} - \bar{y}_{\mathrm{sb}}}{s_{\mathrm{pooled}}} \tag{8}$$

	$D_{KS} = \sup_y \lvert F_{ns}(y) - F_{sb}(y) \rvert$	(9)

where $\bar{y}_{\mathrm{ns}}$ and $\bar{y}_{\mathrm{sb}}$ are the mean target values of no-scaffold and scaffold-bearing molecules, respectively, s_{pooled} is the pooled standard deviation, and J is the small-sample correction factor. $F_{\mathrm{ns}}(y)$ and $F_{\mathrm{sb}}(y)$ denote the empirical cumulative distribution functions of target values for no-scaffold and scaffold-bearing molecules, respectively.

For classification tasks, label-distribution shift was measured by positive-rate difference and Bernoulli Jensen–Shannon divergence[32]:

$$\Delta p = p_{\mathrm{ns}} - p_{\mathrm{sb}} \tag{10}$$

$$\mathrm{JSD}(P_{\mathrm{ns}}, P_{\mathrm{sb}}) = \frac{1}{2} D_{\mathrm{KL}}(P_{\mathrm{ns}} || M) + \frac{1}{2} D_{\mathrm{KL}}(P_{\mathrm{sb}} || M) \quad (11)$$

where p_{ns} and p_{sb} are the positive rates of no-scaffold and scaffold-bearing molecules, respectively; P_{ns} and P_{sb} are the corresponding Bernoulli label distributions; and $M = (P_{\mathrm{ns}} + P_{\mathrm{sb}})/2$.

Splitter implementation and split-composition analysis

We compared four molecular splitting strategies: Bemis–Murcko Scaffold Split, UMAP-based clustering split, DataSAIL, and LoHi. Train/validation/test split weights were adjusted according to dataset size. For multi-fold splitters, namely Bemis–Murcko scaffold split and UMAP-based clustering split, datasets with $n \leq 1000$ used 3:1:1 split weights and generated 5 folds; datasets with $1000 < n \leq 3000$ used 5:1:1 split weights and generated 7 folds; datasets with $n > 3000$ used 8:1:1 split weights and generated 10 folds. For single-split methods, DataSAIL and LoHi used the same size-dependent train/validation/test weights.

For multi-fold methods, split-composition analyses used the train-depleted representative fold, defined as the fold with the lowest no-scaffold proportion in the training set.

Bemis–Murcko Scaffold Split grouped molecules by their Bemis–Murcko scaffold representation and assigned molecules sharing the same scaffold as an indivisible unit to train, validation, or test. Molecules with empty Bemis–Murcko scaffolds formed a single empty-scaffold group. In cross-validation analyses, the fold in which all no-scaffold molecules appeared in the test set was defined as the no-scaffold fold; the remaining folds, whose test sets contained only scaffold-bearing molecules, were defined as ordinary folds. Unless otherwise stated, no-scaffold performance and mechanistic diagnostics were computed on the no-scaffold subset of the test set in the no-scaffold fold, rather than on the entire test set of that fold.

The UMAP-based clustering split represented molecules with Morgan fingerprints before dimensionality reduction and clustering. Fingerprints were generated with radius 2 and bit length 1024. UMAP was run using 100 nearest neighbors, a minimum distance of 0.0, two embedding dimensions, and the Jaccard distance metric. The embedded

molecules were clustered into N clusters, where N matched the number of folds determined by dataset size. Agglomerative clustering with Ward linkage was used[33, 34]. The resulting clusters were treated as split units and combined with a sliding-window scheme to generate train, validation, and test folds.

For DataSAIL, normalized SMILES were assigned unique molecular IDs. We used the C1e technique, molecule mode M, the SCIP solver[35], and integer split weights matched to dataset size. Output split labels were fixed as train, validation, and test.

For LoHi, molecules were first partitioned into a training set and a remaining subset, after which the remaining molecules were further divided into validation and test sets. The similarity threshold was set to 0.4. Datasets for which LoHi failed to return a valid split were excluded from LoHi-specific summaries.

For each splitter, we calculated the no-scaffold proportion in the full dataset, training set, validation set, and test set. Training-side representation was measured by support recovery:

$$\text{Support recovery} = \frac{q_{\text{train}}}{q_{\text{all}}} \tag{12}$$

where q_{all} and q_{train} are the no-scaffold proportions in the full dataset and training set, respectively. Evaluation-side enrichment was measured by sink ratio:

$$\text{Sink ratio} = \frac{\max(q_{\text{valid}}, q_{\text{test}})}{q_{\text{train}}} \tag{13}$$

where q_{valid} and q_{test} are the no-scaffold proportions in the validation and test sets, respectively.

Model training and evaluation

We evaluated seven representative baseline models, including FCNN[36] and XGBoost[37] models based on molecular fingerprints or molecular descriptors, the graph neural network D-MPNN[9], the pretrained graph model MolCLR[38], and the pretrained molecular language model ChemBERTa-3[39]. All models were trained and evaluated independently on the same split indices.

For fingerprint-based FCNN and XGBoost models, molecules were represented by Morgan fingerprints with radius 2 and bit length 1024. For descriptor-based FCNN and XGBoost models, molecular descriptors computed by RDKit were used as input

features. The descriptor vector contained 43 two-dimensional RDKit descriptors covering molecular weight, hydrogen-bond donor and acceptor counts, rotatable bonds, heavy-atom and heteroatom counts, amide bonds, fraction of sp^3 carbons, ring and heterocycle counts, stereocenter counts, Labute ASA, TPSA, Crippen logP, Crippen MR, Kier–Hall connectivity indices, kappa shape indices and molecular flexibility descriptors[40-42].

D-MPNN used molecular graphs as input. MolCLR was fine-tuned from pretrained graph representations for each prediction task. ChemBERTa-3 used SMILES strings as input and was fine-tuned from a pretrained molecular language model. All models were fitted on the training set, with the validation set used for model selection or early stopping.

Task inclusion criteria

Datasets with fewer than 8% no-scaffold molecules were excluded from structural-cohesion and distribution-shift analyses. Multi-endpoint datasets were expanded into task-level units before classification and regression analyses.

For classification performance analyses, tasks were retained only when the no-scaffold test subset contained at least 10 positive and 10 negative samples. Ordinary-fold reference subsets were also required to contain at least 10 samples from each class. Tasks without any eligible ordinary-fold reference were excluded from the main classification analysis.

Classification performance metrics

For classification tasks, the primary performance metric was normalized PR (normPR) performance. Because average precision (AP) depends on the positive rate of the evaluation set, raw AP can be difficult to compare across subsets or tasks with different class priors[43-45]. We therefore normalized AP against the random-prior baseline defined by the test-set positive rate. For a given test set, let AP be the average precision and π be the positive rate:

$$\mathrm{normPR} = \frac{AP - \pi}{1 - \pi} \tag{14}$$

Under this definition, normPR = 0 corresponds to prior-only ranking performance, whereas normPR > 0 indicates ranking information beyond the test-set prior. Accordingly, normPR values close to zero indicate ranking performance close to the prior baseline.

For each model–task pair, the median normPR across ordinary folds was used as the reference baseline. Classification degradation on the no-scaffold test subset was defined as

$$\Delta\text{normPR} = \text{normPR}_{\text{no-scaffold fold, no-scaffold}} - \text{median}\left(\text{normPR}_{\text{ordinary folds}}\right) \quad (15)$$

Within the no-scaffold fold, the performance difference between no-scaffold and scaffold-bearing test subsets was defined as

$$\Delta\text{normPR}_{\text{within}} = \text{normPR}_{\text{no-scaffold}} - \text{normPR}_{\text{scaffold-bearing}} \quad (16)$$

To quantify the fraction of evaluation points with performance close to the prior-only regime, we defined the near-baseline rate at threshold τ as

$$R_{\tau} = \frac{|e \in \mathcal{E}:\text{normPR}_{e} \leq \tau|}{|\mathcal{E}|} \quad (17)$$

where $\mathcal{E}$ denotes the set of model–task evaluation points.

Regression performance metrics

For regression tasks, the primary error metric was relMAE. For a test set A, mean absolute error was defined as

$$\text{MAE}(A) = \frac{1}{|A|}\sum_{i \in A} |\hat{y}_i - y_i| \quad (18)$$

where $\hat{y}_i$ and y_i denote the predicted and observed target values for molecule m_i, respectively., relMAE was defined as

$$\text{relMAE}(A) = \frac{\text{MAE}(A)}{S_y(A)} \quad (19)$$

where $S_y(A)$ is the target-scale factor, defined as the mean absolute target value in test set A. relMAE was used as a scale-adjusted error metric for regression endpoints with different target units and magnitudes. No-scaffold degradation was interpreted primarily through within-model–task comparisons against the corresponding ordinary-

fold reference, rather than as an absolute cross-task error scale. Error inflation of the no-scaffold test subset relative to ordinary folds was defined as

$$\text{relMAE inflation} = \frac{\text{relMAE}_{\text{no-scaffold fold,no-scaffold}}}{\text{median}(\text{relMAE}_{\text{ordinary folds}})} \tag{20}$$

To separate the sources of relative error inflation, we further computed the absolute MAE ratio and target-scale ratio:

$$\text{MAE ratio} = \frac{\text{MAE}_{\text{no-scaffold}}}{\text{median}\left(\text{MAE}_{\text{ordinary folds}}\right)} \tag{21}$$

$$\text{Target-scale ratio} = \frac{S_y(\text{no-scaffold})}{\text{median}\left[S_y(\text{ordinary folds})\right]} \tag{22}$$

Here, $\text{MAE}_{\text{no-scaffold}}$ and $S_y(\text{no-scaffold})$ were computed on the no-scaffold test subset of the no-scaffold fold. Ordinary-fold quantities were computed separately for each ordinary fold and summarized by the median as the reference baseline. To estimate ordinary-fold background variability, pairwise relMAE ratios were computed among ordinary folds within each model–task pair. The probabilities of 2-fold and 3-fold differences were then compared with relMAE inflation on the no-scaffold test subset.

Mechanistic analysis

Mechanistic analyses were performed at the task-level median scale. For classification tasks, failure severity was measured by ΔnormPR, and candidate mechanisms included training-support loss, prior shift, calibration drift, and prediction collapse. For regression tasks, failure severity was measured by relMAE inflation, and candidate mechanisms included training-support loss, target shift, absolute MAE growth, target-scale contraction, and output-scale change.

For classification tasks, training-support loss was defined as the difference in training-support similarity between ordinary folds and the no-scaffold fold, as described above. Training-support similarity below 0.30 was treated as a low-support condition in threshold-based analyses. Prior shift was quantified as the absolute positive-rate difference between the no-scaffold test subset and the ordinary-fold reference:

$$\left|\Delta\pi\right| = \left|\pi_{\text{ns}} - \pi_{\text{ord}}\right| \tag{23}$$

Calibration drift was measured by changes in Brier score[46] and expected calibration error (ECE)[47, 48]. Brier score was defined as

$$\text{Brier} = \frac{1}{n}\sum_{i=1}^{n}(\hat{p}_i - y_i)^2 \tag{24}$$

where $\hat{p}_i$ denotes the predicted probability for the positive class, $y_i \in \{0,1\}$ denotes the corresponding binary label, and n is the number of samples in the evaluated subset. ECE was defined as

$$\text{ECE} = \sum_{b=1}^{B}\frac{|B_b|}{n}|\text{acc}(B_b) - \text{conf}(B_b)| \tag{25}$$

where B_b denotes the set of samples in confidence bin b, $\text{acc}(B_b)$ and $\text{conf}(B_b)$ denote the empirical accuracy and mean predicted confidence in that bin, respectively. In mechanistic analyses, ΔBrier and ΔECE were used to represent calibration changes in the no-scaffold fold relative to the ordinary-fold reference.

Prediction collapse was quantified by the ratio of the predicted-probability standard deviation in ordinary folds to that in the no-scaffold test subset:

$$\text{Collapse strength} = \frac{\text{SD}(\hat{p}_{\text{ordinary}})}{\text{SD}(\hat{p}_{\text{no-scaffold}})} \tag{26}$$

where $\hat{p}_{\text{ordinary}}$ and $\hat{p}_{\text{no-scaffold}}$ denote predicted probabilities in ordinary folds and in the no-scaffold test subset, respectively. Larger values indicate stronger compression of the prediction-score scale under the no-scaffold test distribution.

To assess whether prediction collapse was independent of prior shift and calibration drift, we computed partial Spearman correlations. Specifically, ΔnormPR, the candidate mechanism, and the control variables were first converted to ranks. The candidate mechanism and ΔnormPR were then separately residualized against $|\Delta\pi|$, ΔBrier, and ΔECE, and the correlation between the two residuals was computed. p-values were calculated from the corresponding t-distribution[49].

For regression tasks, training-support loss was defined as the difference in nearest-neighbor training-support similarity between ordinary folds and the no-scaffold fold. Target shift was quantified by |Hedges' g| and the KS statistic. Absolute MAE growth

was measured by MAE ratio, and target-scale contraction was measured by target-scale ratio. Output-scale change was defined as

$$\text{Output-scale change} = \frac{\mathrm{SD}(\hat{y}_{\text{no-scaffold}})}{\mathrm{median}\left[\mathrm{SD}(\hat{y}_{\text{ordinary}})\right]} \tag{27}$$

where $\hat{y}_{\text{no-scaffold}}$ denotes regression predictions for the no-scaffold test subset in the no-scaffold fold, and $\hat{y}_{\text{ordinary}}$ denotes predictions in ordinary folds.

For regression tasks, evaluation points were stratified into lighter and more severe support-loss groups according to the median point-level support loss. Within each group, we summarized the median relMAE inflation and the risks of 2× and 3× inflation.

Associations between candidate mechanisms and failure severity were assessed using Spearman rank correlation.

Empty-scaffold-aware repair procedure

ESA was implemented as a post-splitting allocation repair applied to raw DataSAIL splits. Raw DataSAIL splits were used as the leakage-aware scaffold-bearing backbone. For each dataset, the assignments of scaffold-bearing molecules were kept fixed, and only no-scaffold molecules were reassigned. The analysis evaluated whether no-scaffold molecules could be redistributed to restore training-side representation while preserving no-scaffold exposure in validation and test branches and controlling additional train-facing similarity exposure.

Let D_{ns} denote the set of no-scaffold molecules, and let T_{sb} be the scaffold-bearing molecules assigned to training by the original DataSAIL split. ESA partitions no-scaffold molecules into local structural micro-clusters and assigns these micro-clusters to the no-scaffold training branch T_{ns}, validation branch V_{ns}, test branch Te_{ns}, or a dropped set D_{drop}. The full training set after repair is $T_{\text{all}} = T_{\text{sb}} \cup T_{\text{ns}}$

No-scaffold micro-clusters were constructed from Morgan fingerprints using Tanimoto similarity. No-scaffold micro-clustering used a Tanimoto similarity threshold of 0.40. To avoid assigning an overly large no-scaffold group as an indivisible unit, the maximum micro-cluster size was capped at 0.80 times the smallest no-scaffold branch

target in the primary ESA-TrainSupport configuration. Molecules without valid fingerprints were assigned as singleton groups.

For candidate assignments, ESA used a coupled greedy score that combined sampled similarity costs and split-composition penalties:

$$J = \lambda_{\text{int}} L_{\text{internal}}(D_{\text{ns}}) + \lambda_{\text{val}} L(V_{\text{ns}} \to T_{\text{all}}) + \lambda_{\text{test}} L(Te_{\text{ns}} \to T_{\text{all}}) + \lambda_{\text{imb}} I + \lambda_{\text{drop}} |D_{\text{drop}}| \quad (28)$$

Here, $L_{\text{internal}}(D_{\text{ns}})$ penalizes undesirable similarity coupling among no-scaffold micro-clusters across split branches. $L(V_{\text{ns}} \to T_{\text{all}})$ and $L(Te_{\text{ns}} \to T_{\text{all}})$ penalize train-facing similarity exposure from no-scaffold validation and test micro-clusters to the full training set, respectively. I denotes split-size imbalance relative to the target train/validation/test proportions, and $|D_{\text{drop}}|$ is the number of dropped no-scaffold molecules. Similarity costs used in greedy assignment were estimated from sampled pairwise Tanimoto similarities, with a maximum sampling cap of 16 molecules per group. These greedy-score similarity terms were used for assignment and are distinct from the post hoc normalized leakage metric L_{norm} defined below.

For the primary ESA-TrainSupport configuration, used $(\lambda_{\text{int}}, \lambda_{\text{val}}, \lambda_{\text{test}}, \lambda_{\text{imb}})$ = (0.5, 1.0, 1.5, 0.03), with $\lambda_{\text{drop}} = 2.0$. The larger weight on $L(Te_{\text{ns}} \to T_{\text{all}})$ reflects the priority of limiting train-facing similarity exposure for the no-scaffold test branch.

The assignment was subject to branch-level lower bounds on no-scaffold molecule counts. If $n_{\text{ns}} = |D_{\text{ns}}|$, and (r_T, r_V, r_{Te}) are the dataset-size-dependent target split ratios, the no-scaffold branches were constrained as

$$|T_{\text{ns}}| \geq 0.90 r_T n_{\text{ns}}, |V_{\text{ns}}| \geq 0.40 r_V n_{\text{ns}}, |Te_{\text{ns}}| \geq 0.40 r_{Te} n_{\text{ns}} \quad (29)$$

The dropped no-scaffold fraction was constrained by $|D_{\text{drop}}| \leq 0.05 n_{\text{ns}}$.

If validation or test lower bounds remained unmet after greedy assignment, small low-train-similarity no-scaffold micro-clusters were moved from the no-scaffold training branch to the underfilled branch without violating the training lower bound. This step ensured that ESA did not restore training-side representation by moving all no-scaffold molecules into training.

For each dataset, we recorded the no-scaffold proportions in the training, validation and test sets before and after ESA repair. Training-side representation was summarized by support recovery, and evaluation-side over-concentration was summarized by sink ratio, as defined above.

To assess train-facing similarity exposure after repair, we calculated a dataset-normalized leakage contribution. For each dataset D, let D_{valid} denote molecules with valid fingerprints. The dataset-level denominator was defined as the total pairwise Tanimoto similarity within the dataset:

$$Z(D) = \sum_{m_i, m_j \in D_{\text{valid}}, i<j} T(i,j) \tag{30}$$

For an evaluation branch C and reference set R, the total cross-branch similarity was defined as

$$L_{\text{total}}(C \to R) = \sum_{m_i \in C_{\text{valid}}} \sum_{m_j \in R_{\text{valid}}} T(i,j) \tag{31}$$

The normalized leakage contribution was then defined as

$$L_{\text{norm}}(C \to R) = \frac{L_{\text{total}}(C \to R)}{Z(D)} \tag{32}$$

We computed L_{norm} separately for the no-scaffold validation and test branches against the full training set, $L_{\text{norm}}(V_{\text{ns}} \to T_{\text{all}})$ and $L_{\text{norm}}(Te_{\text{ns}} \to T_{\text{all}})$, respectively. We also computed the mean maximum similarity from each query molecule to the reference set:

$$\text{MeanMaxSim}(C \to R) = \frac{1}{|C_{\text{valid}}|} \sum_{m_i \in C_{\text{valid}}} \max_{m_j \in R_{\text{valid}}} T(i,j) \tag{33}$$

The leak rate at a Tanimoto similarity threshold of 0.40 was defined as

$$\text{LeakRate}_{0.4}(C \to R) = \frac{|\{m_i \in C_{\text{valid}} : \max_{m_j \in R_{\text{valid}}} T(i,j) \geq 0.40\}|}{|C_{\text{valid}}|} \tag{34}$$

After merging the repaired no-scaffold allocation with the fixed scaffold-bearing backbone, we further evaluated repair-induced similarity exposure. For scaffold-bearing and no-scaffold validation/test branches, we calculated the change in normalized leakage before and after repair:

$$\Delta L_{\mathrm{norm}} = L_{\mathrm{norm}}^{\mathrm{post}} - L_{\mathrm{norm}}^{\mathrm{pre}} \quad (35)$$

This analysis was reported separately for four branches: scaffold-bearing validation, scaffold-bearing test, no-scaffold validation and no-scaffold test. The full algorithm, preliminary condition sweep and comparison between ESA-Balanced and ESA-TrainSupport are provided in **Supplementary Notes 3 and 4, Supplementary Algorithm 1, and Supplementary Tables 16 and 17**.

Data availability

The original MoleculeNet and Deep-PK/ADMET datasets used in this study are publicly available from the MoleculeNet data portal (https://moleculenet.org/datasets-1) and the Deep-PK data portal (https://biosig.lab.uq.edu.au/deeppk/data), respectively. The processed datasets, train/validation/test split files, and model-prediction tables generated in this study are available at Zenodo: https://doi.org/10.5281/zenodo.20337102. These archived files include the processed, splits and predictions folders used by the analysis scripts.

Code availability

The custom code used for molecular preprocessing, scaffold extraction, split generation, no-scaffold allocation diagnostics, Empty-Scaffold-Aware repair, metric calculation and numerical analysis is available at GitHub: https://github.com/wangyingyu1993/empty-scaffold-allocation. The repository includes the analysis scripts, environment file and ESA configuration files required to reproduce the main analyses from the processed datasets, split files and model-prediction tables.

Supporting Information

Supplementary Notes 1-4, Supplementary Figs. 1-7, Supplementary Tables 1-17 and Supplementary Algorithm 1.

Corresponding Author:

Tao Song,

E-mail: tsong@upc.edu.cn

Author Contributions

The manuscript was written through contributions of all authors. Yong Wang conceived and designed the study, conducted experiments, performed data analysis, and wrote the initial draft of the manuscript. Shuang Wang was responsible for software development, data analysis and provided funding support. Xun Wang provided research supervision and funding support. Tao Song supervised the research, revised and edited the manuscript, managed the project, and provided funding support. All authors have given approval to the final version of the manuscript.

Funding Sources

This work was supported by National Key Research and Development Project of China (2021YFA1000103, 2021YFA1000102), National Natural Science Foundation of China (Grant Nos.62272479, 62372469, 62202498). Taishan Scholarship (tstp20240506), Shandong Provincial Natural Science Foundation (ZR2021QF023).

Competing interests

The authors declare no competing interest.

References

1. Catacutan, D. B., Alexander, J., Arnold, A. & Stokes, J. M. Machine learning in preclinical drug discovery. *Nat. Chem. Biol.* **20**, 960–973 (2024).
2. Zhang, K. et al. Artificial intelligence in drug development. *Nat. Med.* 31, 45–59 (2025).
3. Buterez, D., Janet, J. P., Kiddle, S. J., Oglic, D. & Lió, P. Transfer learning with graph neural networks for improved molecular property prediction in the multi-fidelity setting. *Nat. Commun.* **15**, 1517 (2024).

4. Durant, G., Boyles, F., Birchall, K. & Deane, C. M. The future of machine learning for small-molecule drug discovery will be driven by data. *Nat. Comput. Sci.* **4**, 735–743 (2024).
5. Swanson, K. et al. ADMET-AI: a machine learning ADMET platform for evaluation of large-scale chemical libraries. *Bioinformatics* **40**, btae416 (2024).
6. Myung, Y., de Sá, A. G. & Ascher, D. B. Deep-PK: deep learning for small molecule pharmacokinetic and toxicity prediction. *Nucleic Acids Res.* **52**, W469–W475 (2024).
7. Wu, Z. et al. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).
8. Huang, K. et al. Therapeutics Data Commons: machine learning datasets and tasks for drug discovery and development. *Proc. 35th Conf. Neural Inf. Process. Syst. Track Datasets Benchmarks* (2021).
9. Yang, K. et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **59**, 3370–3388 (2019).
10. Tossou, P., Wognum, C., Craig, M., Mary, H. & Noutahi, E. Real-world molecular out-of-distribution: specification and investigation. *J. Chem. Inf. Model.* **64**, 697–711 (2024).
11. Bemis, G. W. & Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **39**, 2887–2893 (1996).
12. Sahigara, F. et al. Comparison of different approaches to define the applicability domain of QSAR models. *Molecules* **17**, 4791–4810 (2012).
13. Segal, N., Netanyahu, A., Greenman, K. P., Agrawal, P. & Gómez-Bombarelli, R. Known unknowns: out-of-distribution property prediction in materials and molecules. *npj Comput. Mater.* **11**, 345 (2025).
14. Guo, Q., Hernandez-Hernandez, S. & Ballester, P. J. Scaffold splits overestimate virtual screening performance. In *Artificial Neural Networks and Machine Learning – ICANN 2024*, Lecture Notes in Computer Science Vol. **15025**, 58–72 (Springer, 2024).
15. Guo, Q., Hernandez-Hernandez, S. & Ballester, P. J. UMAP-based clustering split

for rigorous evaluation of AI models for virtual screening on cancer cell lines. *J. Cheminform.* **17**, 94 (2025).

16. Healy, J. & McInnes, L. Uniform manifold approximation and projection. *Nat. Rev. Methods Primers* **4**, 82 (2024).
17. Joeres, R., Blumenthal, D. B. & Kalinina, O. V. Data splitting to avoid information leakage with DataSAIL. *Nat. Commun.* **16**, 3337 (2025).
18. Steshin, S. Lo-Hi: practical ML drug discovery benchmark. *Adv. Neural Inf. Process. Syst.* **36**, 64526–64554 (2023).
19. Bento, A. P. et al. An open source chemical structure curation pipeline using RDKit. *J. Cheminform.* **12**, 51 (2020).
20. Fourches, D., Muratov, E. & Tropsha, A. Trust, but verify: on the importance of chemical structure curation in cheminformatics and QSAR modeling research. *J. Chem. Inf. Model.* **50**, 1189–1204 (2010).
21. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).
22. Weininger, D., Weininger, A. & Weininger, J. L. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.* **29**, 97–101 (1989).
23. Morgan, H. L. The generation of a unique machine description for chemical structures—a technique developed at Chemical Abstracts Service. *J. Chem. Doc.* **5**, 107–113 (1965).
24. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**, 742–754 (2010).
25. Willett, P., Barnard, J. M. & Downs, G. M. Chemical similarity searching. *J. Chem. Inf. Comput. Sci.* **38**, 983–996 (1998).
26. Bajusz, D., Rácz, A. & Héberger, K. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J. Cheminform.* **7**, 20 (2015).
27. Ceriani, L. & Verme, P. The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini. *J. Econ. Inequal.* **10**, 421–443 (2012).

28. Lakens, D. Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and ANOVAs. *Front. Psychol.* **4**, 863 (2013).
29. Hedges, L. V. & Olkin, I. Statistical Methods for Meta-Analysis. (Academic Press, 1985).
30. Conover, W. J. Practical Nonparametric Statistics, 3rd edn. (John Wiley & Sons, 1999).
31. Hodges, J. L. Jr. The significance probability of the Smirnov two-sample test. *Ark. Mat.* **3**, 469–486 (1958).
32. Lin, J. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **37**, 145–151 (1991).
33. Ward, J. H. Jr. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **58**, 236–244 (1963).
34. Murtagh, F. & Legendre, P. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *J. Classif.* **31**, 274–295 (2014).
35. Achterberg, T. SCIP: solving constraint integer programs. *Math. Program. Comput.* **1**, 1–41 (2009).
36. Shao, J. et al. Prediction of maximum absorption wavelength using deep neural networks. *J. Chem. Inf. Model.* **62**, 1368–1375 (2022).
37. Chen, T. & Guestrin, C. XGBoost: a scalable tree boosting system. In *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 785–794 (ACM, 2016).
38. Wang, Y., Wang, J., Cao, Z. & Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **4**, 279–287 (2022).
39. Singh, R. et al. ChemBERTa-3: an open source training framework for chemical foundation models. *Digit. Discov.* **5**, 662–685 (2026).
40. Wildman, S. A. & Crippen, G. M. Prediction of physicochemical parameters by atomic contributions. *J. Chem. Inf. Comput. Sci.* **39**, 868–873 (1999).
41. Ertl, P., Rohde, B. & Selzer, P. Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. *J. Med. Chem.* **43**, 3714–3717 (2000).
42. Hall, L. H. & Kier, L. B. The molecular connectivity chi indexes and kappa shape

indexes in structure-property modeling. In *Reviews in Computational Chemistry* Vol. **2**, 367–422 (VCH Publishers, 1991).

43. Davis, J. & Goadrich, M. The relationship between precision-recall and ROC curves. In *Proc. 23rd Int. Conf. Mach. Learn.* 233–240 (ACM, 2006).
44. Saito, T. & Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One* **10**, e0118432 (2015).
45. Brabec, J., Komárek, T., Franc, V. & Machlica, L. On model evaluation under non-constant class imbalance. In *Computational Science – ICCS 2020*, Lecture Notes in Computer Science Vol. 12140, 74–87 (Springer, 2020).
46. Brier, G. W. Verification of forecasts expressed in terms of probability. *Mon. Weather Rev.* **78**, 1–3 (1950).
47. Naeini, M. P., Cooper, G. F. & Hauskrecht, M. Obtaining well calibrated probabilities using Bayesian binning. *Proc. AAAI Conf. Artif. Intell.* **29**, 2901–2907 (2015).
48. Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. On calibration of modern neural networks. In Proc. 34th Int. Conf. Mach. Learn., *Proc. Mach. Learn. Res*. **70**, 1321–1330 (PMLR, 2017).
49. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Sourcedata.xlsx
- Supplementaryinformation.docx