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.

JURY: A Comprehensive Training-Free to Fully-Supervised Framework for Evaluating Chemical Language Models

Daanish Uddin Khan
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Naafey Aamer
naafey.aamer@cs.rptu.de
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Muhammad Sajjad
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Sebastian Vollmer
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Andreas Dengel
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Muhammad Nabeel Asim
German Research Centre for Artificial Intelligence

Research Article

Keywords: Molecular property prediction, ADMET, Molecular language models, Frozen representations, Representation probing, Evaluation Framework

Posted Date: August 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10420771/v1

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

Additional Declarations: No competing interests reported.

JURY: A Comprehensive Training-Free to Fully-Supervised Framework for Evaluating Chemical Language Models

Daanish Uddin Khan[1], Naafey Aamer[1*], Muhammad Sajjad[1], Sebastian Vollmer[1], Andreas Dengel[1,2,3], Muhammad Nabeel Asim[2,3]

[1]Department of Computer Science, University of Kaiserslautern-Landau, Gottlieb-Daimler-Straße, Kaiserslautern, 67663.
[2]DFKI, Trippstadter Str. 122, Kaiserslautern, 67663, Rhineland-Palatinate, Germany.
[3]Intelligentx GmbH (intelligentx.com), Kaiserslautern, 67663, Rhineland-Palatinate, Germany.

*Corresponding author(s). E-mail(s): naafey.aamer@cs.rptu.de;
Contributing authors: daanish.uddin_khan@dfki.de; muhammad.sajjad@dfki.de; sebastian.vollmer@dfki.de; andreas.dengel@dfki.de; muhammad_nabeel.asim@dfki.de;

Abstract

Language models have reshaped computational chemistry by learning from SMILES strings, a compact textual notation for a molecule's atoms and bonds. Trained on millions of such strings, these models compress each molecule into a fixed-length embedding usable for tasks like property prediction and de novo generation. To predict a property, researchers attach a prediction head and finetune the model, but a capable head can learn much of a property from labels alone, reaching strong results even on embeddings that carry little relevant chemistry. Since most models in the literature are evaluated only through such finetuned heads, reported performance conflates the model with the head, its training, and the labelled dataset's size and quality, leaving what the model itself contributes unclear. To close this gap, we present JURY, a stringent evaluation framework

that probes chemical language models directly. Two probes require no training: the Neighbourhood Probe predicts from a molecule's nearest neighbours in embedding space, and the Cluster Probe groups molecules by that space alone and predicts from each group's labels. Two probes are trained: the Linear Probe fits a single linear map from embedding to property, and the Nonlinear Probe trains a small network on as few as two labelled molecules up to the full training set. Together the four span zero-shot to fully supervised prediction. We apply each probe to 10 chemical language models across 3 architecture families on 73 ADMET tasks, we find these models are far more alike than prior evaluation suggests: no model dominates across probes, and ECFP fingerprints, used as a baseline, sit comfortably among them.
Scientific Contribution: A powerful prediction head recovers what a weak one cannot, and in doing so hides what the model failed to supply. JURY makes this distinction visible, replacing a single performance value with a profile across four probes, where agreement marks what a model encodes. Across ten chemical language models and ECFP, this profile shows an embedding space is defined by three properties, how much chemistry it holds, in what form, and how readily it can be recovered, none conveyed by a single prediction head.

Keywords: Molecular property prediction, ADMET, Molecular language models, Frozen representations, Representation probing, Evaluation Framework,

1 Introduction

Predicting how a molecule behaves in the body is one of the central problems of drug discovery: before a candidate compound can become a medicine, its developers must know whether it is absorbed into the bloodstream, how fast the body clears it, and whether it is toxic. These are its absorption, distribution, metabolism, excretion, and toxicity (ADMET) properties [1]. Measuring these in the laboratory is slow and expensive, so the field increasingly predicts them from structure alone [2], and much of that prediction now rests on molecular language models: transformers [3] trained on millions of molecules written as SMILES strings, a notation that records a molecule's atoms and bonds as a line of text [4]. Such a model turns each molecule into an embedding, a single fixed-length vector meant to distil its chemistry into a form a downstream predictor can use [5, 6]. To judge how good a model is, researchers finetune it by attaching a prediction head to the pretrained model, the model is trained on a

labelled dataset for one property, and how well it then predicts that property is taken as the model's quality [7].

These labelled datasets are built on experimental measurements that are costly to obtain, so the assays remain modest in size. Across one the most comprehensive collections of these property datasets, DeepPK [8], the training sets run from barely a hundred molecules to tens of thousands, and many test sets hold only a few dozen compounds (Appendix Table A1). Finetuning is demanding in its own right, training an entire network afresh for every assay and every model, a cost that climbs when a single model carries three billion parameters [9]. Since, the performance that is reported on these datasets is produced by training a head on top of the embedding, it evaluates the head and the model together, and how much of it belongs to the model and how much to the head that read it stays entangled. Given enough capacity, a prediction head can learn much of a property from the labels alone, reaching a strong result on an embedding that holds little of the relevant chemistry [10], so the head's fitting power drives the final number while the model beneath might do little of the work [11]. While the head can supply what the embedding lacks, the current pipelines evaluate finetuned head-and-model pairs, and do not evaluate a model independently [12]. One way to see a model on its own is to freeze it and train only a small predictor, a probe, on the fixed embeddings it produces [12]. No gradient flows back into the model, so whatever the probe predicts is drawn from the embedding as the model already learned. This way, most pretrained molecular models have been found to rank no higher than the extended-connectivity fingerprint (ECFP) [13], a hand-built descriptor that predates them and records which substructures a molecule contains [14]. A probe, though, has a capacity of its own, and what it recovers from an embedding depends on that capacity [10]. A separate line of work addresses the scarcity of labelled data directly, adapting a model to a new property from only a handful of labelled molecules, through few-shot and meta-learning methods standardised on the FS-Mol benchmark [15–18]

and through variants steered by a natural-language description of the task [19, 20]. Like a finetuned head, each of these still evaluates a model through an adaptation trained on top of it.

To address this gap in the domain, we present JURY, an evaluation framework that puts a chemical language model in front of four probes. Every model is held frozen, so nothing it learned can change and no gradient reaches it, and each probe must work from the embeddings exactly as the model learned them. Because all four read the same fixed vectors, any difference in what they report is attributable to the probe alone. The four span the full range of supervision a probe can carry. Two are training-free. (1) The Neighbourhood Probe predicts a molecule from the labels of its nearest neighbours in the embedding space, and (2) the Cluster Probe groups molecules by that space alone and predicts each from the labels of its group. Two are trained. (3) The Linear Probe fits a single linear map from the embedding to the property, and (4) the Nonlinear Probe trains a small neural network to fit a nonlinear one, from as few as two labelled molecules up to the entire training set. We apply JURY to ten pretrained chemical language models spanning the three transformer families (masked encoders, causal decoders, and encoder–decoders) together with the ECFP fingerprint as a baseline, across all 73 ADMET tasks of DeepPK, 49 classification and 24 regression. Every probe is applied as a fixed instrument, at the same configuration on every model, dataset and task, with no tuning to any one of them.

Our results reveal that these models are far more alike than previous evaluation shows. Under every probe, none of the 10 chemical language models dominate, and ECFP Fingerprints (used as a baseline) sit comfortably among them, ahead of all ten on the regression tasks whenever its embedding is used as written. Across the 73 tasks the four probes agreed only weakly on how the models compare (mean rank correlation 0.36). Chemical language models differ less in how much chemistry they hold than in the form they hold it in, and a score reported from a single predictor cannot

distinguish the two. JURY makes the distinction visible, replacing a performance value with a profile across four probes in which agreement marks what a model encodes and disagreement marks how it encodes it.

Two models tested divide almost perfectly along this line. MoLFormer-XL [5] holds its chemistry in the layout of the space itself: on the classification tasks it ranks first of the eleven under both the Neighbourhood and Cluster Probes, the two instruments that lean on the space's own organisation rather than a trained predictor, and it leads under the Cluster Probe on the regression tasks as well, so molecules sharing a property already sit close together in its space, before any predictor is trained. MolEncoder [21] is the reverse. On its geometry alone it sits mid-pack under the Neighbourhood Probe on classification and drops to last of all eleven on regression, yet once a simple predictor is trained on that same space it ranks first under the Linear Probe on both tasks and first under the Nonlinear Probe on regression. Neither model holds more chemistry than the other; they differ in where that chemistry sits, plainly in the geometry for MoLFormer-XL, hidden until training for MolEncoder. MoLFormer-XL is the model to reach for when a prediction must come from the space directly, without training a predictor, screening a library for compounds similar to ones already known to work, using their measured values, or grouping the library into families before most of it is tested. MolEncoder is the one to train when labelled data is on hand and fast, low-cost prediction across many endpoints is the goal, fitting a quick linear predictor per property across dozens of ADMET assays at once.

2 Methodology

This section discusses JURY's implementation and experimental settings in detail. Figure 1 illustrates the overall flow of the framework

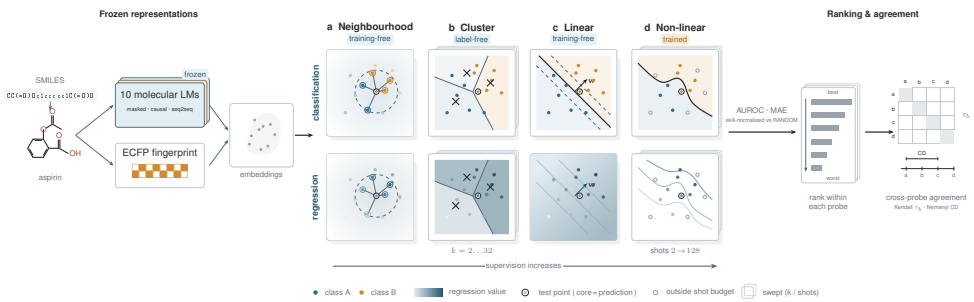
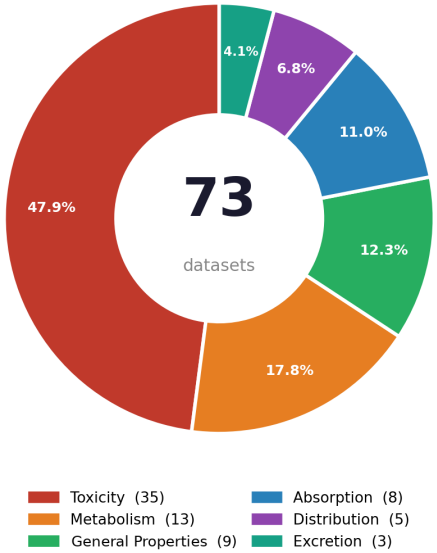


Fig. 1: Overview of the JURY evaluation framework.



Total: 73 datasets | 49 classification · 24 regression | Source: DeepPK (Myung et al., 2024)

Fig. 2: Distribution of the 73 DeepPK datasets.

2.1 Datasets

We evaluated every one of the 73 ADMET datasets in DeepPK [8], each dataset covers a single molecular property to be predicted. The endpoints span six categories (Figure 2): 8 Absorption, 5 Distribution, 13 Metabolism, 3 Excretion, and 35 Toxicity assays, together with 9 General Properties, the physicochemical measurements such as

logP, solubility, and pKa. DeepPK draws these from four earlier collections (ADMETlab 2.0 [22], Interpretable-ADMET [23], toxCSM [24], and pkCSM [25]), merging and standardising each dataset across its sources and partitioning it by a random 8:1:1 train/validation/test split. We took these curated molecules and published partitions as released, adding no filtering, re-standardisation, or re-splitting, and drew only on the train and test partitions, since the probes tune nothing on a held-out split. Within a dataset a molecule reaches the pipeline as one standardised SMILES [4] string paired with one label, giving 49 classification and 24 regression tasks. Per-dataset sizes, categories, and task assignments are listed in Appendix Table A1.

2.2 Frozen Models and Embeddings

Each molecule's SMILES was encoded into a single fixed-length vector, its embedding, by eleven sources: ten pretrained neural language models and the ECFP fingerprint. The ten chemical language models cover three transformer families: masked encoders (MoLFormer [5], ChemBERTa [6], bert-base-smiles [26], ChemBERT-ChEMBL [27], and MolEncoder [21]), causal decoders (GP-MoLFormer [28], MolGPT [29], and ChemFM-3B [9]), and encoder–decoder models (ZINC-T5 [30] and smole-bart [31]). ECFP is the Morgan fingerprint at radius 2 folded to a 2,048-bit vector [14], carried as a baseline.

Each model output one contextual vector per SMILES token, so each molecule was reduced to a single embedding by the model's own pooling: where a model exposes a dedicated pooled vector we took it, and otherwise we averaged the final-layer token vectors under the attention mask, drawing that final layer from the encoder for the encoder–decoder models. Every embedding was extracted once and stored, and the probes use only these frozen embeddings, so all four probes read the identical geometry for a given molecule. Per-model pretraining data, objectives, and parameter counts are given in Table 1.

Table 1: Pretrained molecular language models used as frozen representation sources, grouped by training objective.

Model	Pre-Training Data	Objective	Params
MoLFormer [5]	ZINC + PubChem (∼100M)	Masked token prediction	44.3M
ChemBERTa [6]	PubChem (77M)	Masked token prediction	3.4M
bert-base-smiles [26]	ChEMBL	Masked token prediction	109M
MolEncoder [21]	ChEMBL (∼1M)	Masked token prediction	15M
ChemBERT-ChEMBL [27]	ChEMBL v33	Masked token prediction	6.4M
MolGPT [29]	ZINC15	Next-token prediction	6.9M
GP-MoLFormer [28]	ZINC + PubChem (unique)	Next-token prediction	46.7M
ChemFM-3B [9]	UniChem (178M)	Next-token prediction	3.0B
ZINC-t5 [30]	ZINC	Denoising autoencoder	198M
smole-bart [31]	GuacaMol	Denoising autoencoder	5.4M

2.3 The JURY Evaluation Framework

JURY evaluates each frozen embedding with four probes, applied as fixed instruments at the same configuration on every source, dataset, and task, with no tuning to any one of them [12]. The four span the full range of supervision a probe can carry, from a rule that learns nothing to a fully trained network. The first takes the embedding geometry exactly as the model produced it. The other three begin with one shared preparatory step, a per-dimension z-score that centres each embedding dimension and puts it on a common scale, fit on the probe's own training molecules and then applied to the test molecules. For classification each probe returns a decision value or probability, and for regression a predicted value, both feeding the same downstream metrics.

2.3.1 Neighbourhood Probe

The Neighbourhood Probe measures whether molecules that sit close together in the embedding space tend to share a property. It trains nothing. To predict a test molecule, it looks at every training molecule and takes an average of their known property values, weighting each one by how similar it is to the test molecule so that closer

molecules count for more. Similarity is measured by the angle between two embedding vectors: each vector is first scaled to unit length, which compares molecules by direction alone and ignores how long the vector happens to be, and the similarity is then how closely the two directions align. Molecules pointing in opposite directions would count against the prediction, so their similarity is set to zero and they simply drop out of the average. If no training molecule is similar to a given test molecule, the prediction falls back to the average property value across the training set. Every source uses this same similarity measure, including the ECFP fingerprint, whose own native similarity coefficient is tested separately in a supplementary analysis.

2.3.2 Cluster Probe

The Cluster Probe measures whether molecules that share a property fall into the same region of the embedding space, when those regions are drawn without any knowledge of the property. After the shared standardisation, it sorts the training molecules into a fixed number of groups, chosen so that molecules placed in the same group sit close together in the space; the number of groups, k, is set in advance, and the property values play no part in forming them. Each group is then labelled with the average property value of its own training molecules, the fraction that are positive for classification or the mean value for regression. To predict a test molecule, the probe finds the group whose centre it lands closest to, by straight-line distance, and assigns it that group's average. If a group happened to contain no training molecules, a test molecule landing there instead takes the average property value across the whole training set.

2.3.3 Linear Probe

The Linear Probe measures how well a single straight-line relationship connects the embedding to the property. After the shared standardisation, the probe finds the best simple rule that combines a molecule's embedding values into one number approximating its property: each value is given a multiplier, the results are added, and the

total is the prediction. The multipliers are solved for in a single step across all training molecules at once, without the repeated adjustment that training a network needs. The calculation carries a fixed penalty, $\lambda = 1.0$, that keeps the multipliers from growing too large; the same value is used for every source and dataset, and its effect is examined separately in a supplementary analysis. This penalty also keeps the calculation stable and its solution unique, even when a dataset holds fewer molecules than the embedding has dimensions. For classification the two classes are written as $+1$ and -1, and the probe's output is the resulting value, which runs positive or negative, with higher values indicating the $+1$ class; for regression the output is the predicted value directly.

2.3.4 Nonlinear Probe

The Nonlinear Probe measures how well the property can be recovered from the embedding when the relationship between the two is allowed to bend rather than follow a straight line, and how much this depends on the number of labelled molecules available. It trains a small network that can fit these more complex patterns. After the shared standardisation, a fresh two-layer network (hidden width 128, trained for 500 epochs with the Adam optimiser at learning rate 10^{-3} and weight decay 10^{-4} [32]) is fit to the labelled molecules, with the standardisation fit on the same molecules the network trains on and then applied to the test molecules. For classification the network's output is passed through a function that maps it to a probability between 0 and 1; for regression the output is the predicted value directly. The probe runs at two operating points: a few-shot point that trains on a small labelled subset, standardised on that subset alone, and a full-supervision point that trains on the entire training set.

Two of the four probes are run across a range of settings rather than at a single one, so their performance can be traced as that setting changes. The Cluster Probe is run at group counts of $k \in \{2, 4, 8, 16, 32\}$, stepping from coarse to fine; going beyond two groups tests whether dividing the molecules more finely than the property's own

categories still tracks the property. The same values are used for every dataset, and the largest is capped so that even the smallest dataset has enough molecules to fill its groups. Rather than discarding small groups in advance, the pipeline records at each k how many groups come out empty or sparsely filled. The Nonlinear Probe's few-shot point is trained on a growing number of labelled molecules, up to a ceiling set by the data and differing by task. For regression, the molecules are drawn at random, up to the size of the smallest regression training set (122), giving $\{2, 4, 8, 16, 32, 64\}$; for classification they are drawn so both classes are represented evenly, up to twice the smallest minority class (73), giving $\{2, 4, 8, 16, 32, 64, 128\}$. A shared range up to 64 carries any comparison made across the two task types at a single count. The two probes that involve randomness, the Cluster Probe in how it first forms its groups and the Nonlinear Probe in how it starts its network and samples its few-shot molecules, were each run at five seeds (0–4) and averaged; the two deterministic probes were run once. Probe pseudocode and fallback rules are given in Appendix B.

2.4 Evaluation

Every probe turns a molecule's frozen embedding into a prediction, which was graded against the molecule's known label by a fixed metric (Table 2). Each dataset uses one primary metric for the main analysis and one secondary metric as a check. For classification the primary is the area under the receiver operating characteristic curve (AUROC), which measures how well a probe separates the two classes, and the secondary is the area under the precision–recall curve (AUPRC), which focuses on the positive class when positives are rare. For regression the primary is the mean absolute error (MAE), the average error in the property's own units, and the secondary is Spearman's ρ, which asks only whether molecules are ordered correctly. Since MAE is in each property's own units and cannot be compared across datasets, Spearman's ρ accompanies it wherever regression is shown on one common axis. To give each metric

Table 2: Evaluation metrics and their definitions

Quantity	**Definition**
Performance metrics	
AUROC ↑	$\dfrac{\sum_{i:\, y_i=1} R_i - n_+(n_+ + 1)/2}{n_+\, n_-}$
AUPRC ↑	$\dfrac{1}{n_+}\sum_{\ell=1}^{n} \mathbb{1}[y_{(\ell)}=1]\, P_\ell, \qquad P_\ell = \frac{1}{\ell}\sum_{j\le\ell} \mathbb{1}[y_{(j)}=1]$
MAE ↓	$\dfrac{1}{n}\sum_{i=1}^{n} \lvert s_i - y_i \rvert$
Spearman ρ ↑	$\mathrm{corr}\big(R^s, R^y\big)$
Chance floor (analytic permutation expectation, per dataset and metric)	
Chance floor, AUROC	0.5
Chance floor, MAE	$\dfrac{1}{n^2}\sum_{i,j} \lvert y_i - y_j \rvert$
Aggregation and ranking	
Skill score ↑	$\dfrac{x - \text{floor}}{\text{perfect} - \text{floor}}, \qquad \text{perfect} = 1 \text{ (higher-better)},\ 0 \text{ (lower-better)}$
Average rank	$\bar{r}_s = \dfrac{1}{N}\sum_{d=1}^{N} r_{s,d}$
Statistical analysis	
Friedman χ^2	$\dfrac{12N}{k(k+1)}\left(\sum_{s=1}^{k} \bar{r}_s^2 - \dfrac{k(k+1)^2}{4}\right)$
Nemenyi CD	$q_\alpha \sqrt{\dfrac{k(k+1)}{6N}}, \qquad q_\alpha = \text{studentized range}(\alpha, k)/\sqrt{2}$
Kendall τ-b	$\dfrac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}}, \qquad n_0 = \frac{k(k-1)}{2}$
Kendall's W	$\dfrac{12\, S}{J^2(k^3 - k) - J\, T}, \qquad S = \sum_s (R_s - \bar{R})^2$

Within a dataset: n test molecules, label y_i, prediction s_i; n_+, n_- positive and negative class counts; R_i the rank of s_i among the n predictions (ties averaged); R^s, R^y the rank vectors of predictions and labels.
Across datasets: N datasets, k sources; x a raw metric value; $r_{s,d}$ the within-dataset rank of source s (1 = best, ties averaged), $\bar{r}_s$ its average; J probes, n_c, n_d concordant and discordant source pairs, n_1, n_2 tie terms, R_s the rank-sum of source s, T its tie term.

a fixed reference, every dataset was also anchored at chance, the level reached by a predictor with no access to molecular structure, obtained by shuffling the labels. This lets a result described as near chance mean so literally, with the random predictor treated as a twelfth source that is graded but never probed. Two forms of variation were absorbed into every result. The two probes that involve randomness, the Cluster and Nonlinear, were each run at five seeds and their predictions averaged before grading, while the deterministic Neighbourhood and Linear Probes use none. Every estimate then received a 95% confidence interval from a bootstrap that resamples the test molecules $B = 1{,}000$ times, seeded so the interval never depends on run order, and the same implementation produces both the estimate and its interval.

Raw metric values cannot be compared across datasets of differing difficulty, so each was placed on a common footing with a skill score (Table 2) running from 0 at the dataset's chance floor to 1 at the metric's best attainable value; a worse-than-chance result was kept negative rather than clipped, and each per-dataset confidence interval was carried through the same mapping. The eleven sources were then summarised in two ways. Within each (dataset, probe, metric, sweep) slice they were ranked from best to worst, ties shared and only finite values counted, with the random predictor setting the floor but never ranked. Across datasets each source was reported both by its average rank and by its median skill, since the average rank alone would turn statistically tied sources into a firm order while the skill keeps the small margin between them. Every summary was formed overall and within each of the six property categories, with each row carrying its dataset count and categories of fewer than three datasets flagged as small. For the budget curves and any summary in a metric's own units, the median across datasets was used with a percentile bootstrap, since low-budget regression produces large negative outliers that would distort a mean. To compare the probes on equal terms, each was fixed at one operating point: the Neighbourhood, Linear, and Nonlinear Probes use the full training set, and the Cluster Probe fits on it at a single

cluster count k, so all four draw on the whole training set and differ only in what they do with it. That k was chosen without labels, as the largest value at which at least 90% of datasets stay acceptable, holding no empty clusters and a median under-filled fraction of at most 0.25; the best k per dataset was also recorded but kept separate as a clearly marked upper limit, never used in the main comparison.

To check whether our findings hold up, we applied three statistical tests to the per-source rankings, all defined in Table 2. The Friedman test asks whether the eleven sources really differ in how they rank across datasets, or whether their ordering is just noise. When it finds a real difference, the Nemenyi test says which sources stand apart from which. The Kendall τ-b, together with Kendall's W, then measures our central question: how far the four probes agree on the ordering of the sources, one pair of probes at a time for τ-b and all four together for W. Each result carries a confidence interval from a bootstrap that resamples the datasets, so agreement is called stable only when its interval stays clear of chance.

3 Results

Every model in this study turns a molecule into an embedding, a single list of numbers that fixes the molecule as one point in a space it shares with all the others. Whatever a model has learned about chemistry, it has to express through where these points sit. What we want to know is how much real chemistry each space holds, in what form it is held, and how easily that chemistry can be recovered. To find out, we place each frozen space in front of the four probes, using them as four different ways of checking what the space contains.

3.1 The Embedding Space Already Holds Chemistry

We begin with the simplest of the four probes, the Neighbourhood Probe, because it trains nothing at all and so shows what a space holds on its own. To predict a new

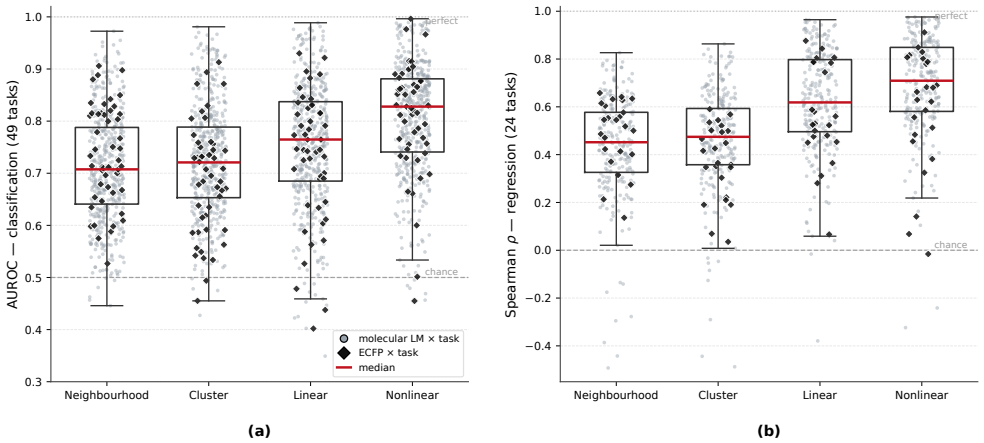


Fig. 3: How well each probe predicts, across all eleven models and every task. Each box shows the spread of performance for one probe, with one point per model and task. (a) Classification (49 tasks). (b) Regression (24 tasks).

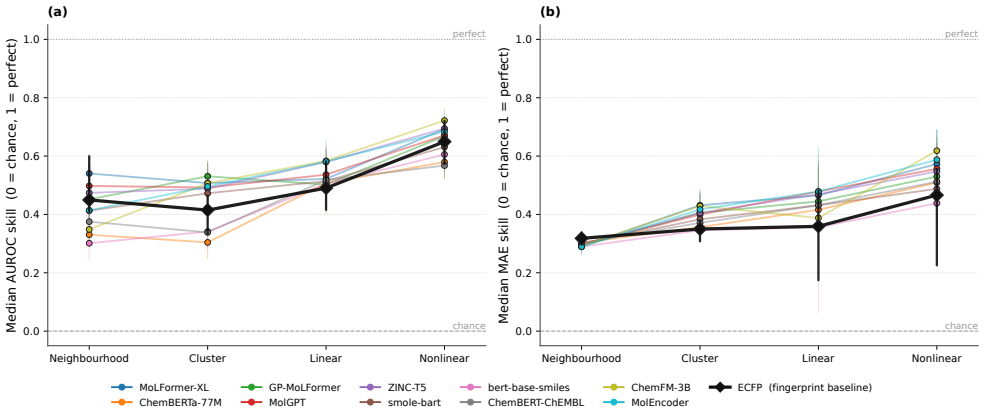


Fig. 4: The eleven models placed on a common footing, where 0 is chance and 1 is perfect. Each line is one model, showing its median skill across tasks under each probe. (a) Classification (49 tasks). (b) Regression (24 tasks).

molecule, it finds the labelled molecules sitting closest to it in the embedding space and averages what those molecules are known to do.

This matters because none of the models were ever trained on the ADMET properties we test. Whatever a model has placed in its embedding space, it placed there while learning something else entirely. Yet prediction from nearby molecules alone

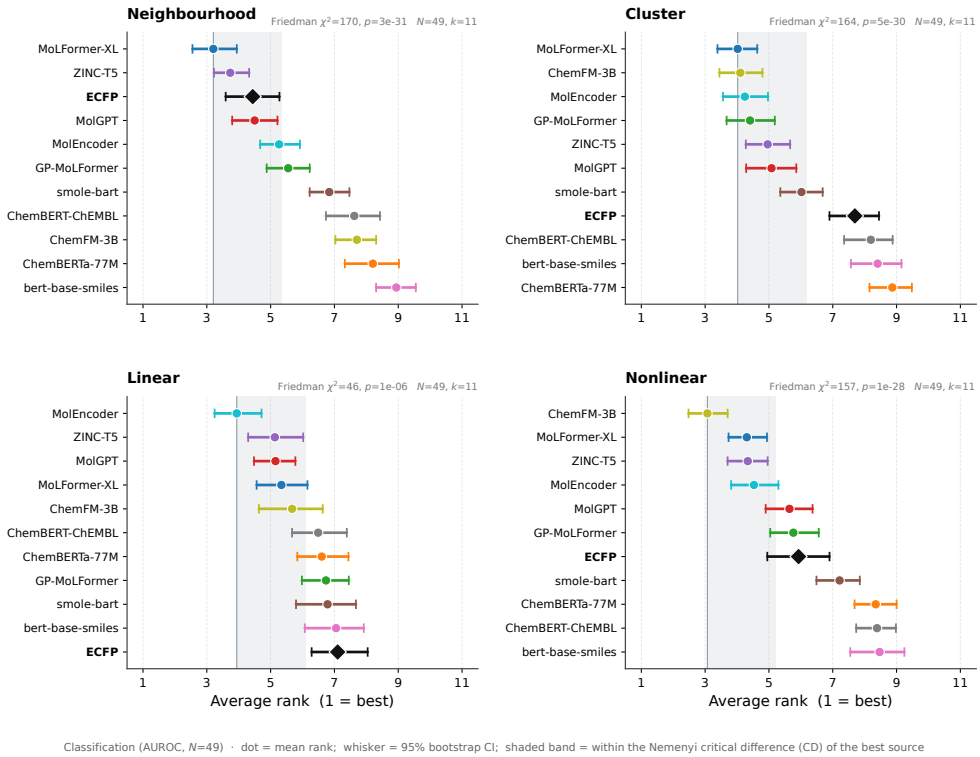


Fig. 5: How the eleven (ten models and the ECFP baseline) rank under each probe on the 49 classification tasks, one probe per panel. A dot is a model's average rank (1 is best) and the bar its uncertainty. Each panel gives its Friedman χ^2 and p.

lands well above chance: across all 73 tasks it reaches a median AUROC of 0.71 on the 49 classification tasks, where 0.5 is a coin flip and 1.0 is perfect, and a median Spearman of 0.45 on the 24 regression tasks, where 0 would mean the ordering carries no information (Figure 3). Molecules that share a property have, without ever being told to, landed close together. Some models arrange molecules this way more plainly than others. On the classification tasks MoLFormer-XL groups molecules by their behaviour better than any of the eleven. The most striking case is the ECFP baseline: on the regression tasks this hand-built fingerprint, which predates every transformer model here, predicts a molecule from its neighbours better than all ten of them, ranking best of the eleven on 18 of the 24 tasks. When prediction comes straight from the

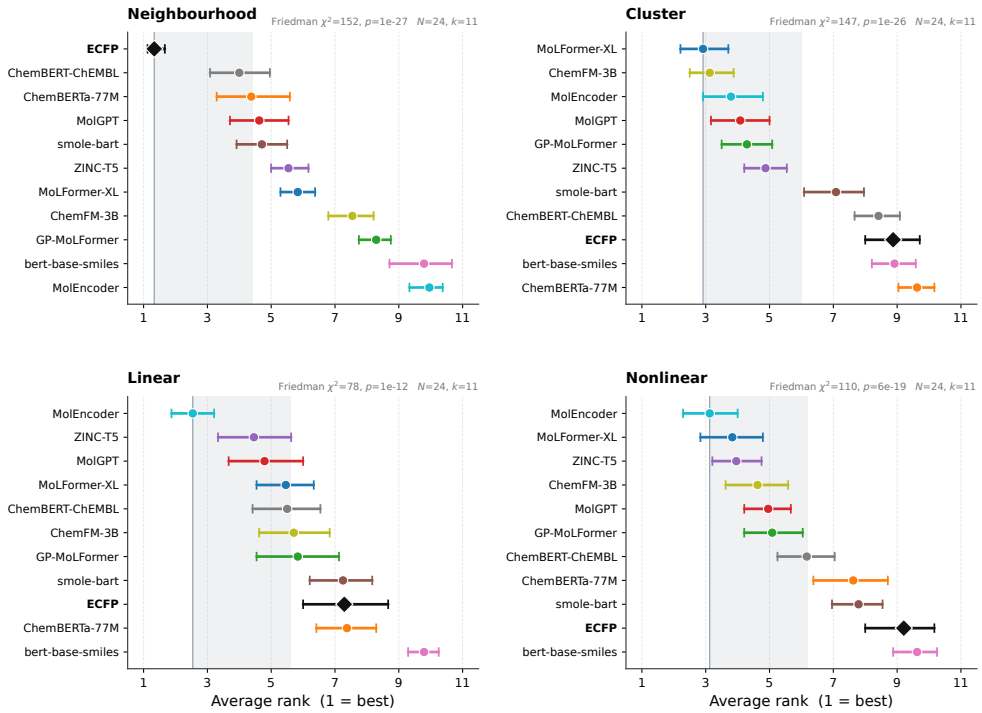


Fig. 6: How the eleven (ten models and the ECFP baseline) rank under each probe on the 24 regression tasks, one probe per panel, with models ranked by their error within each task. Layout as in Figure 5.

arrangement of the space, decades-old chemistry laid out by hand outdoes ten models trained on many millions of molecules.

This holds for every one of the eleven embeddings. Each model arranges molecules so that nearness lines up with real behaviour, and ECFP, a hand-built description of molecular structure that predates these models and serves as our baseline, does the same (Figure C1). The chemistry is already present in the arrangement itself, before a single predictor is trained. Simply by settling where each molecule sits, a model has already done a large part of the work.

The probes that do train, adjusting the space before they predict, reach higher still (Figure 3), so training can add to what the arrangement already supplies. This sets

up the question the rest of the results follow: if every model already holds chemistry in its space, how do the eleven actually differ, and what is a comparison between them measuring?

3.2 Models Hold Similar Amounts of Chemistry, in Different Forms

If every model already holds chemistry in its space, the natural question is how much, and whether some hold far more than others. Placed on a common footing, so that an easy property and a hard one count the same, the eleven crowd into a narrow band, and ECFP crowds in with them (Figure 4, with each model shown property by property in Figure C2). Whichever probe checks the space, the model in first place sits only a little above the middle of the pack. The eleven hold comparable amounts of chemistry, and the leader's advantage over an ordinary model is slim.

They are close in quality, yet a probe can still tell them apart. Ranking the eleven on each property and asking whether that order stays steady across all 73 tasks, or reshuffles by chance, gives a clear answer under every probe: the order is steady, comfortably past the usual bar for significance (Friedman test; Figures 5 and 6). The gaps between the models are genuine but small.

Small though they are, these gaps point somewhere specific, because the model that comes first changes depending on which probe does the checking. On the classification tasks, MoLFormer-XL leads when molecules are judged by their nearest neighbours and again when the space is sorted into clusters, MolEncoder leads once a straight-line predictor is fitted, and ChemFM-3B leads once a small network is trained (Figure 5). Three different models take first place across four ways of checking the same eleven spaces. The regression tasks tell the same story, and there it is ECFP whose standing swings the most, first under one probe and near the bottom under the others

(Figure 6), a case we come back to shortly. A model that looks strongest to one probe looks ordinary to another.

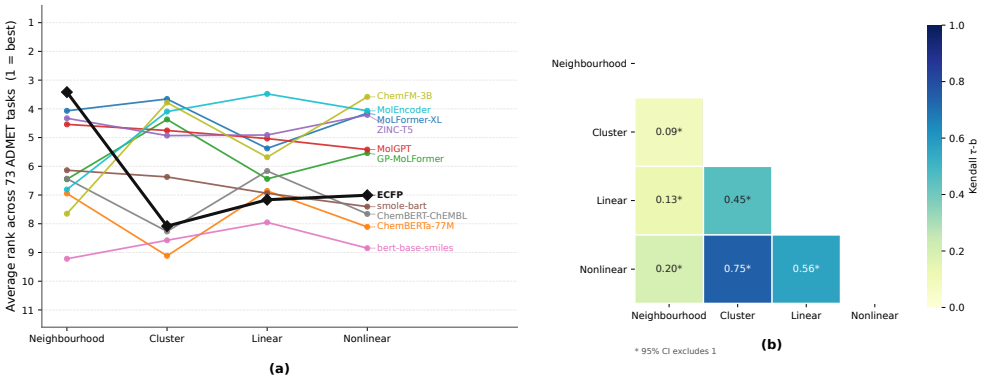


Fig. 7: Agreement between the probes, pooled over all 73 tasks. (a) Each model's average rank under the four probes, one line per model. (b) How far each pair of probes agrees on the ordering, measured by Kendall's τ-b.

Taken across all four probes at once, this movement is plain. We can measure how far the probes agree on a simple scale where 0 means their orderings have nothing in common and 1 means they match exactly. Comparing the probes two at a time, using a standard rank-agreement measure called Kendall's τ-b, and averaging over every pairing of the four, gives only 0.36, and no two probes ever reach the same ranking (Figure 7). Asking a second way, how tightly all four orderings line up at once, using a related measure called Kendall's W, gives only 0.62, when a single ranking shared by all four would give 1. Both checks say the same thing: the probes fall well short of agreement. The eleven models hold no fixed standing. Each probe examines the space a different way, and the ranking moves with it. That movement is the clue we follow: when two probes disagree about which model is best, each is recovering a model's chemistry from a different place, which can happen only if the chemistry is held in a different form. Where each form sits, and how to tell one from another, is what we turn to next.

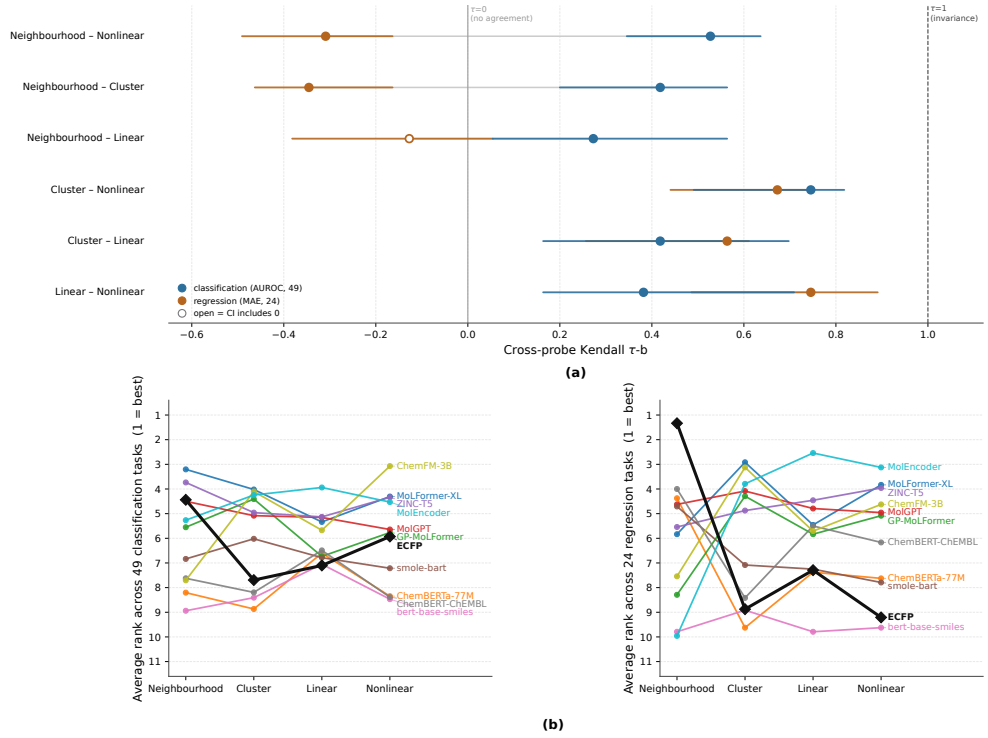


Fig. 8: Agreement between the probes, split by task type. (a) Kendall's τ-b for each pair of probes, shown separately for the 49 classification and 24 regression tasks. (b) Each model's average rank under the four probes, for classification and for regression.

3.3 Where the Chemistry Sits, Plain Geometry or Recovered by Training

The four probes divide cleanly along a single step in how each handles the space. The Neighbourhood Probe takes molecules where the model placed them and compares them by their positions directly. The Cluster, Linear, and Nonlinear Probes each begin with one shared preparatory step, stretching every direction of the space to a common width so that no direction counts for more simply because the model spread molecules further along it, and only then predict. This step, present in three of the probes and absent in the fourth, is the line along which the models' standings move. It separates

the chemistry a model lays out plainly in its spacing from the chemistry a model keeps in directions that surface only once the space is levelled.

ECFP shows this most sharply. On the regression tasks it is the best of the eleven when molecules are judged by their raw nearness, at an average rank of 1.3, and it falls to near the bottom, ranks of 7.3, 8.9, and 9.2, once the space is levelled and a cluster, a straight line, or a trained network predicts from it (Figure 6). Classification moves the same way, from 4.4 by nearest neighbours to 7.1 under the straight-line fit (Figure 5). ECFP keeps its chemistry in its plain spacing, so it leads when that spacing is used directly and fades when every direction is put on equal terms. To confirm the swing belongs to the spacing and not to the particular similarity we used, we repeated the neighbour lookup with ECFP's own native similarity in place of the shared one. The overall ordering of the eleven stayed the same, and ECFP's own standing only rose (Figure C3). Where ECFP holds its chemistry is what moves it.

Whole models divide along the same line. MoLFormer-XL holds its chemistry plainly in the layout of its space: it is first of the eleven under both the Neighbourhood and Cluster Probes on classification, and first under the Cluster Probe on regression, so molecules that behave alike already sit together before anything is trained (Figures 5 and 6). MolEncoder keeps its chemistry where a trained predictor has to recover it: it sits mid-pack under the Neighbourhood Probe on classification and last of all eleven on regression by nearest neighbours, yet rises to first once a straight-line or a trained predictor is fitted, on both tasks. Two models of comparable chemistry, holding it in opposite forms.

What draws this line is the levelling step, and it does its work whether or not labels guide the prediction. Neither the Neighbourhood Probe nor the Cluster Probe uses labels to build or arrange the space: the Neighbourhood Probe compares molecules by their given positions, and the Cluster Probe forms its groups from those positions alone. Yet these two agree almost not at all, at 0.09, while the Cluster Probe agrees

most closely of any pair with the fully trained Nonlinear Probe, at 0.75 (Figure 7). What the Cluster Probe holds in common with the trained probes is the levelling step, and that is what aligns them. The dividing line is the preparation of the space, and how much supervision a probe carries plays no part in drawing it.

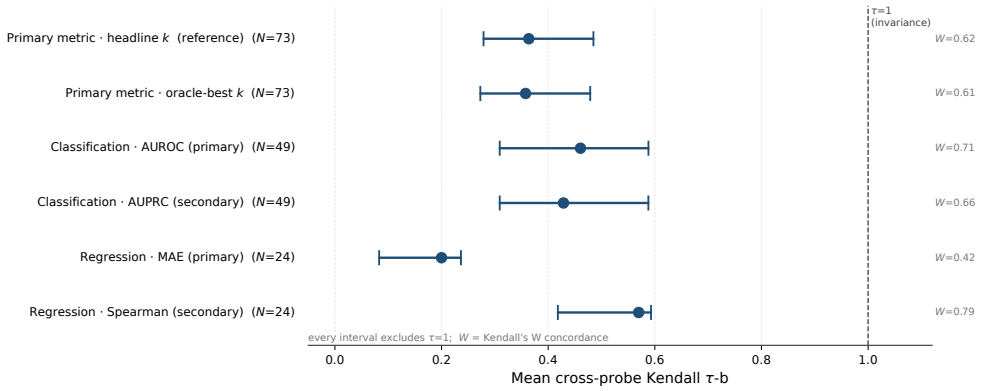


Fig. 9: Cross-probe agreement under alternative scoring and grouping choices. Each row gives the mean Kendall's τ-b across the probes, with the number of tasks it draws on (N).

This division is a real feature of the models rather than an accident of what we happened to measure. Split by task type, the four probes agree moderately on classification and barely at all on regression, and on regression the plain-geometry Neighbourhood Probe orders the models close to the reverse of the levelling probes, its agreement with them turning negative (Figure 8). That regression gap traces to the error measure the tasks were graded by. Graded by mean absolute error, which scores each prediction by how far it lands from the true value, the four probes agree at only 0.20. Graded instead by Spearman, which asks only whether the molecules are ordered correctly from lowest to highest, the same tasks lift to 0.57, back into the range classification occupies, while classification under a second metric holds steady at 0.43 (Figure 9). The same picture survives each probe's own free settings, the number of clusters chosen per property and the strength of the penalty on the straight-line fit (Figures 9 and C4), and

it holds inside every family of chemistry taken on its own, with no category reaching agreement, moderate where the tasks are plentiful, at 0.50 for Metabolism and 0.45 for Toxicity, and lower in the four smaller families (Figure 10). Wherever we look, the same two forms of chemistry pull apart.

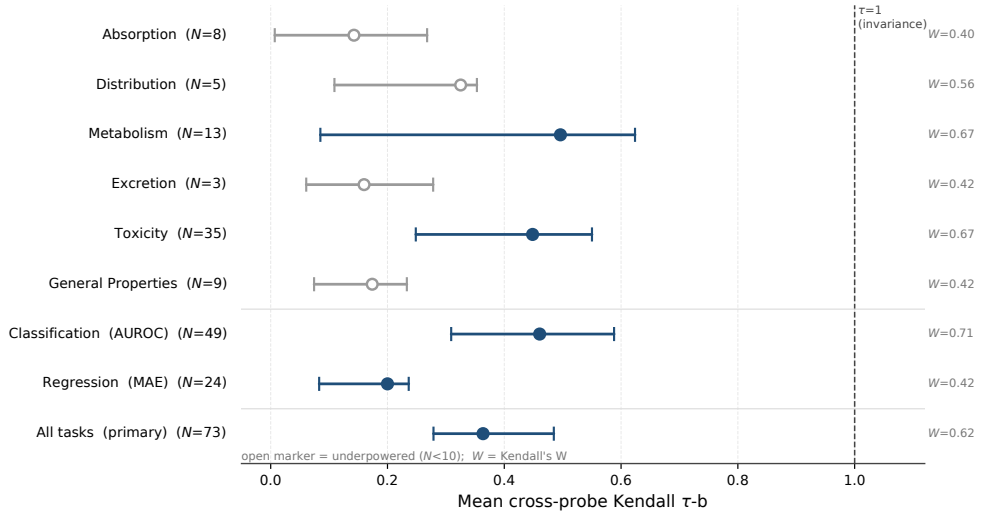


Fig. 10: Cross-probe agreement within each property family and each task type. Each row gives the mean Kendall's τ-b across the probes, with the number of tasks it draws on (N).

Everything to this point compares the probes at a single fixed strength, which leaves one possibility open: perhaps a stronger predictor, given more to learn from, would draw the same chemistry out of every model and close the gaps between them. Testing that means varying how much each predictor is allowed to learn, and doing so uncovers something the fixed comparison cannot, how much of a model's chemistry was already present in its space and how much only training brings out. That is what we measure next.

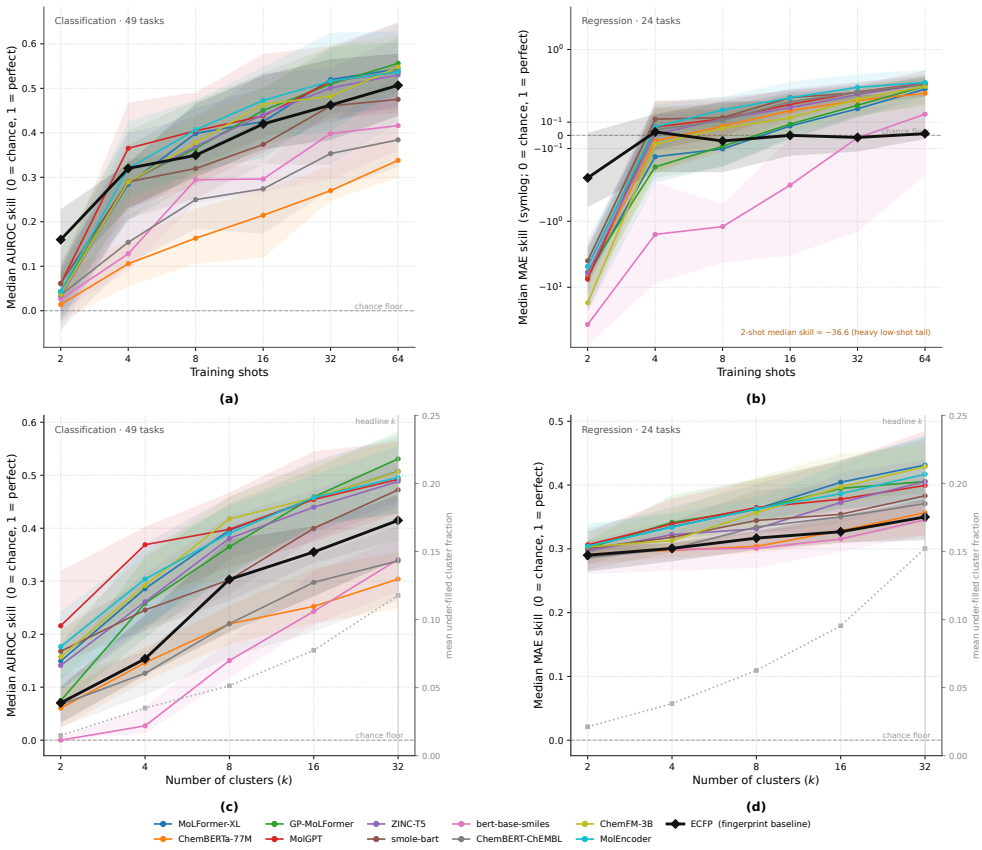


Fig. 11: Each curve is one model's median skill (0 is chance, 1 is perfect). (a, b) The Nonlinear Probe as labelled molecules grow from 2 to 64. (c, d) The Cluster Probe as the number of groups k grows from 2 to 32

3.4 What a Model Already Learned, and What Training Adds

To test whether more predictive power closes the gaps, we vary how much each predictor is given to learn from. The clearest way to do this is to train the small network of the Nonlinear Probe on a growing number of labelled molecules, starting from just 2, then 4, and on up to 64, and to watch how prediction improves as the labels accumulate (Figure 11). A curve like this carries two signals at once. Where it starts, with only a handful of labels, shows how much chemistry the space already supplies on its

own. How steeply it climbs shows how much of the chemistry has to be assembled from the labels instead.

The two task types answer this very differently. On the classification tasks, prediction sits above chance from the smallest budget of all and climbs steadily, from a skill score of 0.05 at 2 labels to 0.51 at 64, on a scale where 0 is chance and 1 is perfect (Figure 11). With almost nothing to learn from, the space already supports prediction, and adding labels sharpens what is largely there from the outset. ECFP starts highest of the eleven at 2 labels, exactly as its plainly held chemistry in the earlier sections would lead us to expect: a description that lays its chemistry out in the open needs the fewest labels to put it to use.

Regression begins in the opposite place. At 2 labels its skill sits far below chance, at a median of -6 and as low as -37 for the weakest embedding, and it climbs across chance only once 4 to 8 labels have accumulated, reaching 0.36 by 64 (Figure 11). The chemistry a regression task needs is not laid out ready to use; it is mostly built up from the labels as they arrive. Between these two extremes the two signals separate cleanly: a model whose curve starts high came in with chemistry already usable, and one whose curve starts low and climbs supplied little on its own and depended on the labels to supply it.

More labels are not the only way to give a predictor more to work with. The Cluster Probe gains in the same way as it is allowed more groups to sort molecules into, its skill rising from 0.11 to 0.44 on classification and from 0.30 to 0.38 on regression as the group count grows to 32, and climbing slowly onward to 0.53 across all tasks pooled together by 512 groups, at which point the datasets large enough to support that many groups have thinned from 73 to 64 (Figures 11 and C5). Whichever way we grant a predictor more power, it recovers more chemistry.

That leaves the question we set out to answer: does a stronger predictor recover the same chemistry from every model, closing the gaps between them? To see this

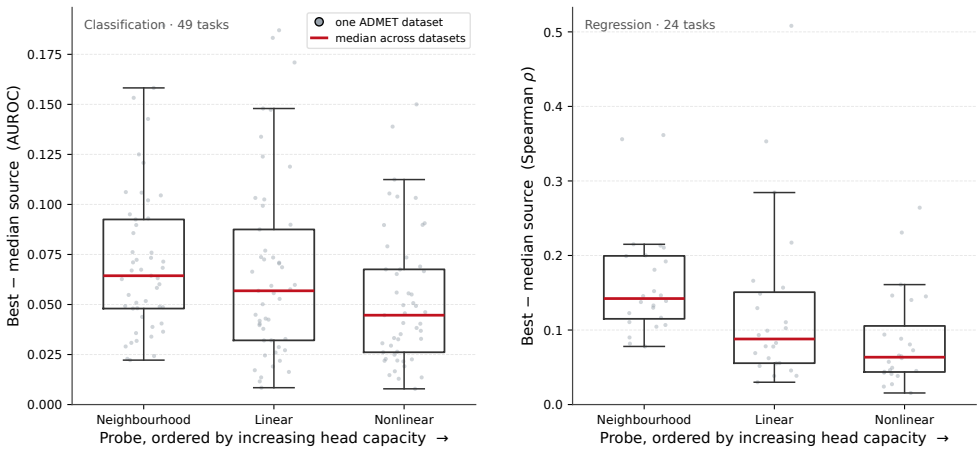


Fig. 12: The gap between the best model and the median one on each task, for the three probes ordered by increasing power (Neighbourhood, Linear, Nonlinear). Left: classification. Right: regression. The Cluster Probe is left out because it forms its groups without using the labels, so it has no place on an ordering by predictive power.

directly, we line up the three probes ordered by how much they can learn, from the least powerful to the most, the Neighbourhood Probe, then the Linear, then the Nonlinear, and measure on each how far the best model stands ahead of the middle of the pack (Figure 12). That lead narrows steadily as the predictor grows stronger, from 0.064 to 0.057 to 0.045 on classification and from 0.14 to 0.09 to 0.06 on regression. A stronger predictor recovers part of what a weaker one could not, and in doing so it draws the models closer together. It never draws them level. Even under the most powerful probe the best model keeps a clear lead, and the models stay ordered differently from one probe to the next, just as they were at every fixed strength before. Adding power compresses the distance between the models without ever settling on one ranking, so no amount of it turns the eleven into a single order. What a comparison between these models can and cannot claim is what the profile in the next section sets out.

4 Discussion

Taken together, the results of this study settle four points about what these embedding spaces hold and how they differ. Every model arranges molecules so that chemistry is already present in the space before any predictor is trained. The eleven hold comparable amounts of that chemistry, and because they differ in the form it takes, the model in front changes with the probe rather than standing fixed. What changes from model to model is the form the chemistry takes, plain in the spacing for some and recoverable only after a predictor levels the space for others. And the few-shot curve tells apart the chemistry a model brings with it from the chemistry that has to be built up from labels. Set side by side, these findings say that a single number can never describe one of these spaces on its own.

A model is described instead by three traits together: how much chemistry its space holds, in what form it holds it, and how readily that chemistry can be recovered. A single accuracy captures one of the three and stays silent on the rest. It cannot say whether a strong result came from chemistry lying open in the arrangement or from a trained predictor working out which directions to re-weight, and those are different models to reach for even when their accuracy is identical. MoLFormer-XL and MolEncoder make the point at its two extremes. They hold comparable chemistry, yet MoLFormer-XL lays its chemistry out plainly in how it spaces molecules, while MolEncoder keeps its chemistry where only a trained predictor recovers it. One accuracy would rate them alike and miss the very thing that separates them.

How a model places across the four probes therefore diagnoses where its chemistry sits, and that in turn points to the work it is fit for, settled before any labels are spent finding out. MoLFormer-XL reaches its best under the two untrained probes, coming first under the Neighbourhood Probe on the classification endpoints and under the Cluster Probe on both, which marks a model whose chemistry lies open in how it spaces molecules. It is fit for prediction that trains nothing at all: prioritising new

candidates that resemble already-measured actives, or sorting an untested library into property-coherent families before any assay budget is committed. ECFP, the baseline, places the same way, first under the Neighbourhood Probe on the regression endpoints, and is fit for that same untrained prediction there. MolEncoder places in the opposite pattern: out of reach of the raw-geometry Neighbourhood Probe, where it sits mid-pack on classification and last of the eleven on regression, and first once a predictor is fitted, leading the Linear Probe on both tasks. It is fit for the work where labelled data is on hand and a predictor can be trained, whether a single cheap linear map for fast, low-cost prediction across many endpoints at once, or a small network for the strongest fit a single endpoint allows. ChemFM-3B places this same way on the classification endpoints, reaching its best only under the trained network, and is fit for the case where accuracy on one endpoint is worth its training cost. Characterising a model by this full profile across the probes, rather than by one accuracy, is what turns an evaluation into a guide to how the model can be used.

The scope of these conclusions follows from the study's design. They rest on the 73 ADMET assays of DeepPK, a broad and widely used benchmark, and they hold most firmly in the property families with many tasks, where the measured agreement and disagreement are best supported. The four probes were chosen to span the range that matters, supervision from none to a trained network and geometry from untouched to fully reshaped, and they map that range well; probes placed elsewhere along it could reveal further structure a space holds. The eleven embeddings cover all three transformer families alongside a fingerprint baseline, a representative cross-section of the models in current use, and the profile extends readily to any further model by examining it the same way. And by holding every model frozen, the study isolates what a model supplies on its own, which is precisely what a downstream user inherits before spending any labels; how far finetuning can draw out a hidden-form model's chemistry is a natural next question the same framework is built to answer.

A single position on a leaderboard belongs as much to the probe that produced it as to the model it names. The same eleven models, examined the same way on the same tasks, fall into different orders depending on how each probe treats the space it is given, and none of those orders is the true one. What holds steady across the probes is a division between two questions: how much chemistry a model encodes, which the probes broadly agree on, and in what form it encodes that chemistry, which is where they part. A profile across the four probes carries both, while a single accuracy settles neither, since one number cannot say whether a strong result came from chemistry lying open in the space or from a predictor trained to recover it.

This reframes how a chemical language model should be judged and chosen. A model earns its place by where its chemistry sits and how well that matches the task in front of it, so the same model can be the right choice for prediction straight from the space and the wrong one when a predictor must be trained, or the reverse. Seen this way, the long-standing finding that these models rank no better than a hand-built fingerprint that predates them becomes a statement about one way of examining them, and it opens onto a more useful question: for a given task, which model holds its chemistry in the form that task can use. Answering that calls for examining a model across a panel of probes and reporting the profile they produce, which is what JURY provides, and what a single benchmark number leaves hidden.

List of abbreviations

ADMET Absorption, distribution, metabolism, excretion, and toxicity

AUPRC Area under the precision–recall curve

AUROC Area under the receiver operating characteristic curve

BCE Binary cross-entropy

CI Confidence interval

ECFP Extended-connectivity fingerprint

IQR Interquartile range

LM Language model (language modelling)

MAE Mean absolute error

MLM Masked language modelling

MSE Mean squared error

ReLU Rectified linear unit

SMILES Simplified molecular-input line-entry system

SPD Symmetric positive definite

W Kendall's coefficient of concordance

τ-b Kendall's τ-b rank correlation coefficient

Declarations

Availability of data and materials. All datasets supporting the conclusions of this article are available in the DeepPK repository, https://biosig.lab.uq.edu.au/deeppk/data [8]. No new datasets were generated; the study uses DeepPK's curated molecules and its published train/validation/test partitions without modification. All code required to reproduce the study is available at upon request.

Competing interests. The authors declare that they have no competing interests.

Funding. No funding was received for this work.

Authors' contributions.

- **Daanish Uddin Khan:** Conceptualization, Methodology, Implementation, Formal analysis, Manuscript Writing, Visualization
- **Naafey Aamer:** Methodology, Validation, Investigation, Manuscript Writing
- **Muhammad Sajjad:** Implementation, Methodology
- **Sebastian Vollmer:** Manuscript Editing, Methodology Validation
- **Andreas Dengel:** Supervision, Reviewed the article and performed final editing

- **Muhammad Nabeel Asim:** Conceptualization, review and editing manuscript, supervision.

Appendix A Datasets

Table A1: Per-dataset characteristics of the 73 DeepPK ADMET endpoints used in this study. Categories follow DeepPK's taxonomy.

Dataset	Category	n_{train}	n_{test}	Task
caco2_reg	Absorption	1978	247	Regression
f20	Absorption	804	100	Classification
hia	Absorption	1150	144	Classification
mdck	Absorption	908	114	Regression
ob	Absorption	795	99	Classification
pgp_inhibitor	Absorption	1795	224	Classification
pgp_substrate	Absorption	993	125	Classification
skin_permeability	Absorption	148	19	Regression
bbb_cns	Distribution	122	15	Regression
bbb_logbb	Distribution	2562	320	Classification
fu	Distribution	1901	238	Regression
ppb	Distribution	2889	361	Regression
vd	Distribution	866	108	Regression
bcrp	Metabolism	555	69	Classification
cyp1a2_inhibitor	Metabolism	15917	1986	Classification
cyp1a2_substrate	Metabolism	292	37	Classification
cyp2c19_inhibitor	Metabolism	15725	1963	Classification
cyp2c19_substrate	Metabolism	208	26	Classification

continued on next page

Table A1 – continued from previous page

Dataset	Category	n_{train}	n_{test}	Task
cyp2c9_inhibitor	Metabolism	15420	1930	Classification
cyp2c9_substrate	Metabolism	827	103	Classification
cyp2d6_inhibitor	Metabolism	15760	1970	Classification
cyp2d6_substrate	Metabolism	733	92	Classification
cyp3a4_inhibitor	Metabolism	18428	2307	Classification
cyp3a4_substrate	Metabolism	1241	155	Classification
oatp1b1	Metabolism	1069	134	Classification
oatp1b3	Metabolism	997	125	Classification
cl	Excretion	668	83	Regression
oct2	Excretion	714	90	Classification
t0.5	Excretion	602	75	Classification
ames	Toxicity	6121	766	Classification
avian_tox	Toxicity	463	58	Classification
bee_tox	Toxicity	146	18	Classification
bioconcF	Toxicity	540	68	Regression
biodegradation	Toxicity	1329	165	Classification
carcinogenicity	Toxicity	1307	162	Classification
crustacean	Toxicity	530	66	Classification
dili	Toxicity	971	121	Classification
eye_corrosion	Toxicity	1837	230	Classification
eye_irritation	Toxicity	4166	522	Classification
fdamdd_reg	Toxicity	960	120	Regression
fm_reg	Toxicity	693	87	Regression
h_ht	Toxicity	2324	290	Classification

continued on next page

Table A1 – continued from previous page

Dataset	Category	n_{train}	n_{test}	Task
herg	Toxicity	16582	2073	Classification
lc50dm	Toxicity	276	35	Regression
micronucleus_tox	Toxicity	512	64	Classification
nr_ahr	Toxicity	5743	719	Classification
nr_ar	Toxicity	6272	784	Classification
nr_ar_lbd	Toxicity	5598	700	Classification
nr_aromatase	Toxicity	4822	605	Classification
nr_er	Toxicity	5329	668	Classification
nr_er_lbd	Toxicity	5774	722	Classification
nr_gr	Toxicity	4838	605	Classification
nr_ppar_gamma	Toxicity	5372	671	Classification
nr_tr	Toxicity	4292	537	Classification
pyriformis_reg	Toxicity	1396	174	Regression
rat_acute_reg	Toxicity	8103	1014	Regression
rat_chronic	Toxicity	304	38	Regression
respiratory_tox	Toxicity	3081	382	Classification
skin_sens	Toxicity	432	54	Classification
sr_are	Toxicity	4583	572	Classification
sr_atad5	Toxicity	5846	732	Classification
sr_hse	Toxicity	5120	637	Classification
sr_mmp	Toxicity	5130	644	Classification
sr_p53	Toxicity	5971	746	Classification
bp	General Properties	4321	540	Regression
hydrationE	General Properties	512	64	Regression

continued on next page

Table A1 – continued from previous page

Dataset	Category	n_{train}	n_{test}	Task
logd	General Properties	8503	1063	Regression
logp	General Properties	11915	1489	Regression
logs	General Properties	7965	1000	Regression
logvp	General Properties	2152	269	Regression
mp	General Properties	6839	854	Regression
pka	General Properties	2554	319	Regression
pkb	General Properties	2928	366	Regression

Appendix B Probe definitions

All four probes are pure functions of $(X_{\text{tr}}, y_{\text{tr}}, X_{\text{te}}, \tau)$, where $\tau \in \{\text{classification}, \text{regression}\}$; they hold no global state and read no configuration. Classification labels are encoded in $\{0, 1\}$; the same probe code runs both tasks, with only the readout differing.

Probe Pseudocode

Algorithm 1 Neighbourhood Probe (cosine, training-free)

$\hat{X}_{\text{tr}} \leftarrow \ell_2\text{-normalise}(X_{\text{tr}})$; $\hat{X}_{\text{te}} \leftarrow \ell_2\text{-normalise}(X_{\text{te}})$ ▷ zero vectors left at zero
$S \leftarrow \hat{X}_{\text{te}}\, \hat{X}_{\text{tr}}^{\top}$ ▷ cosine similarity, $(n_{\text{te}} \times n_{\text{tr}})$
$S \leftarrow \max(S, 0)$ ▷ non-negative kernel
$w_i \leftarrow \sum_j S_{ij}$; $\nu_i \leftarrow \sum_j S_{ij}\, y_{\text{tr},j}$
for each test point i: $\text{score}_i \leftarrow \nu_i / w_i$ if $w_i > 0$, **else** $\bar{y}_{\text{tr}}$ ▷ base-rate fallback

Fallback rules.

Each probe's fallback substitutes the global training base rate (classification) or mean $\bar{y}_{\text{tr}}$ (regression), counted as a diagnostic rather than applied silently. Under ECFP's

Algorithm 2 Cluster Probe (Euclidean, label-free fit, seeded)

1: z-score $X_{\text{tr}} \to X^s_{\text{tr}}$, apply to $X_{\text{te}} \to X^s_{\text{te}}$
2: fit k-means on X^s_{tr} *without labels* (`n_init`=10, `random_state`=seed)
3: **for** cluster c: $r_c \leftarrow$ mean $\{y_{\text{tr},j} : \text{assign}(j)=c\}$ if non-empty, **else** $\bar{y}_{\text{tr}}$ ▷ positive-rate (clf) / mean (reg) readout
4: assign each test point to nearest centroid (Euclidean); $\text{score}_i \leftarrow r_{c(i)}$
5: **record** empty- and under-filled-cluster fractions as diagnostics ▷ never gate k or scoring

Algorithm 3 Linear Probe (closed-form L_2, λ=1.0)

1: z-score $X_{\text{tr}} \to X^s_{\text{tr}}$, apply to $X_{\text{te}} \to X^s_{\text{te}}$
2: $y \leftarrow \{-1, +1\}$ encoding of y_{tr} (classification) **or** y_{tr} (regression)
3: $\bar{y} \leftarrow \text{mean}(y)$; $y_c \leftarrow y - \bar{y}$
4: $w \leftarrow \left(X^{s\top}_{\text{tr}} X^s_{\text{tr}} + \lambda I_D\right)^{-1} X^{s\top}_{\text{tr}} y_c$ ▷ primal $D \times D$ solve; SPD by construction
5: score $\leftarrow X^s_{\text{te}} w + \bar{y}$ ▷ signed decision value (classification) / prediction (regression)

Algorithm 4 Nonlinear Probe (2-layer, shot budget n, seeded)

1: draw n shots with seed: stratified even split across classes (classification) **or** uniform without replacement (regression)
2: z-score on the *shot subset only*; apply to X_{te}
3: train Linear$(D, 128) \to$ ReLU $\to$ Linear$(128, 1)$ for 500 epochs, Adam (lr=10^{-3}, weight decay 10^{-4}), BCE-with-logits (clf) / MSE (reg)
4: score $\leftarrow \sigma$(logit) (classification) / raw output (regression)

Tanimoto similarity an all-zero fingerprint pair (empty union) takes similarity 0 before this same fallback applies, and the ℓ_2 and similarity normalisations leave zero vectors at zero so they exert no pull.

Appendix C Supplementary figures and analyses

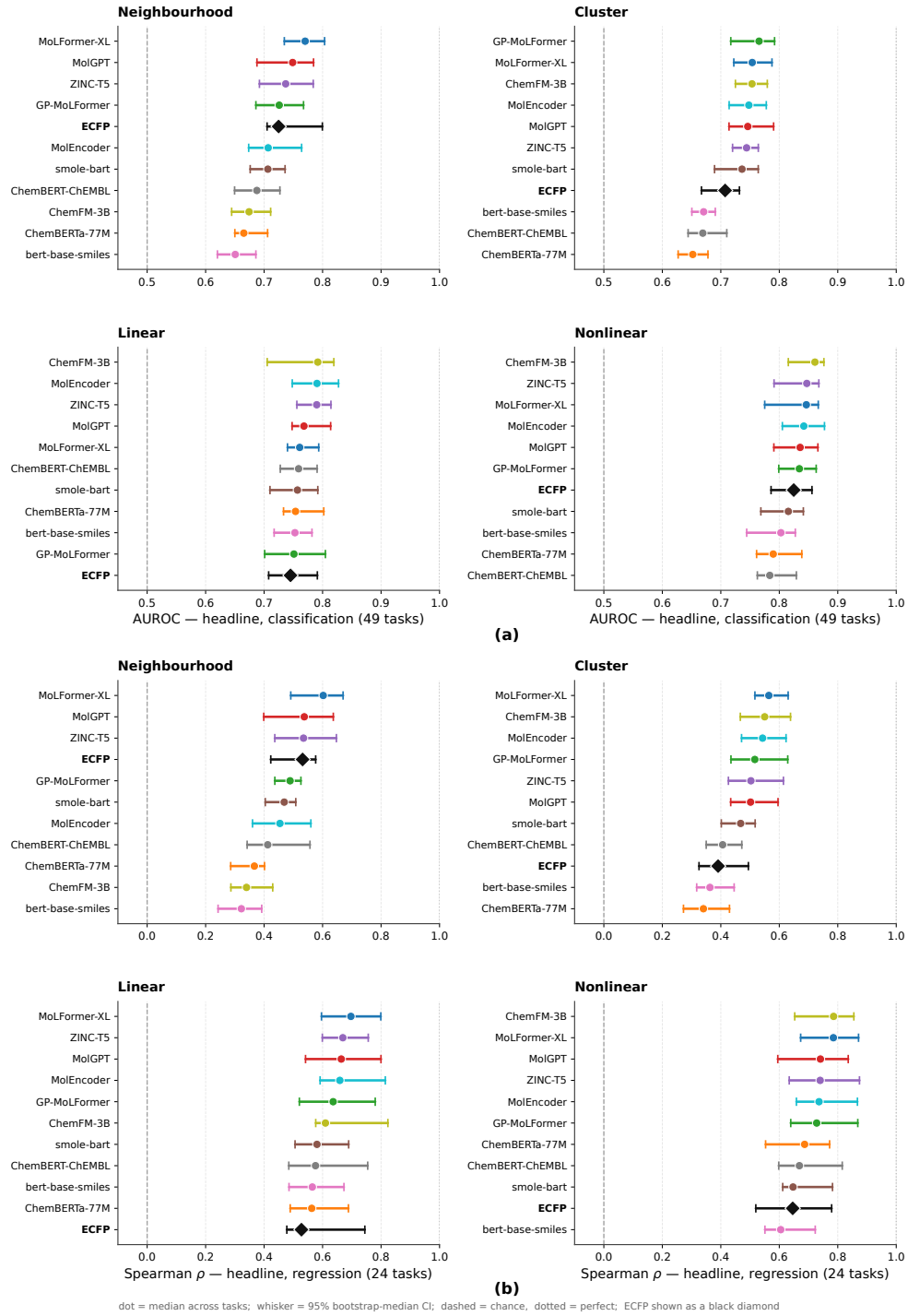


Fig. C1: How well each model predicts under each probe. (a) Classification, AUROC (49 tasks). (b) Regression, Spearman ρ (24 tasks).

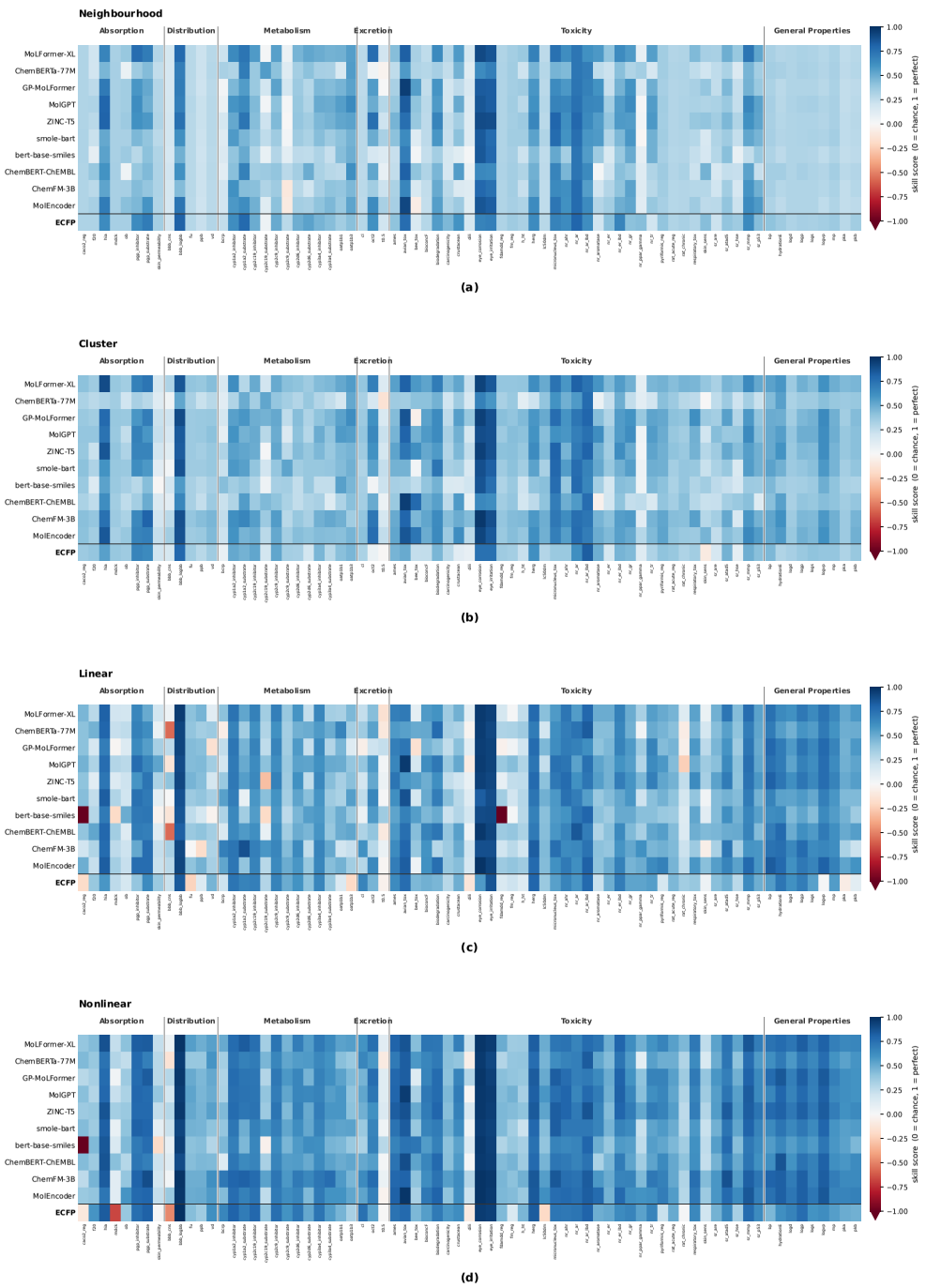


Fig. C2: Each model's skill on every task, one panel per probe, where 0 is chance and 1 is perfect. (a) Neighbourhood. (b) Cluster. (c) Linear. (d) Nonlinear.

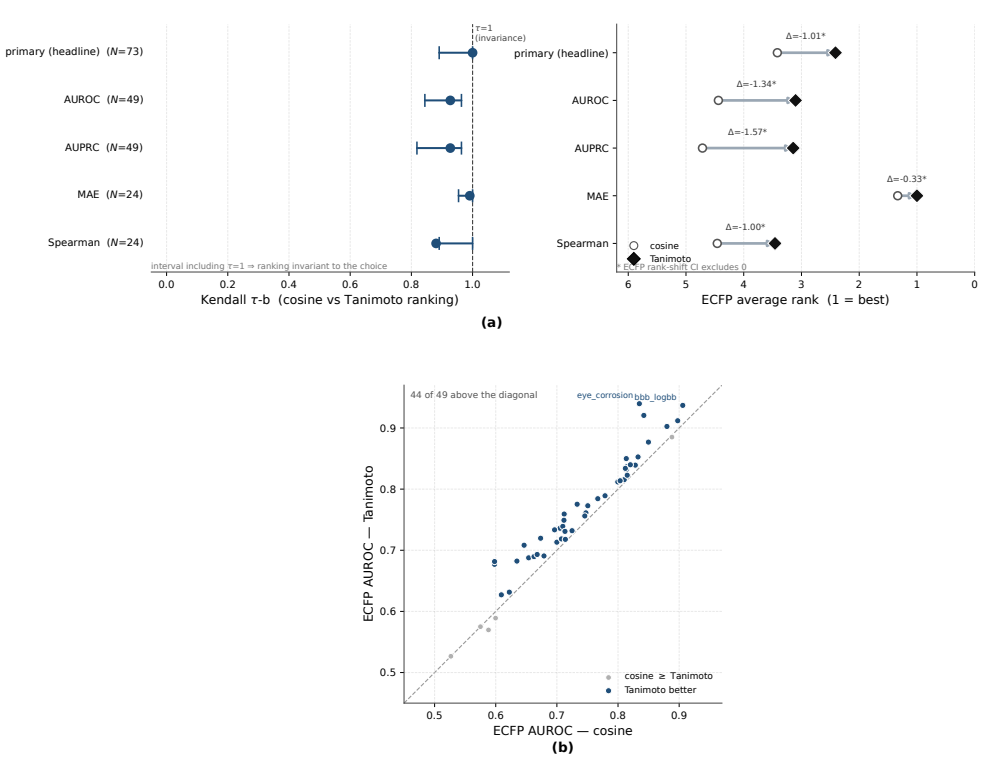


Fig. C3: The Neighbourhood Probe for ECFP using its own similarity (Tanimoto) in place of the shared one (cosine). (a) Left: how far the two orderings of the models agree, by Kendall's τ-b. Right: ECFP's change in average rank between them. (b) ECFP's classification performance on each task, the two similarities against the diagonal.

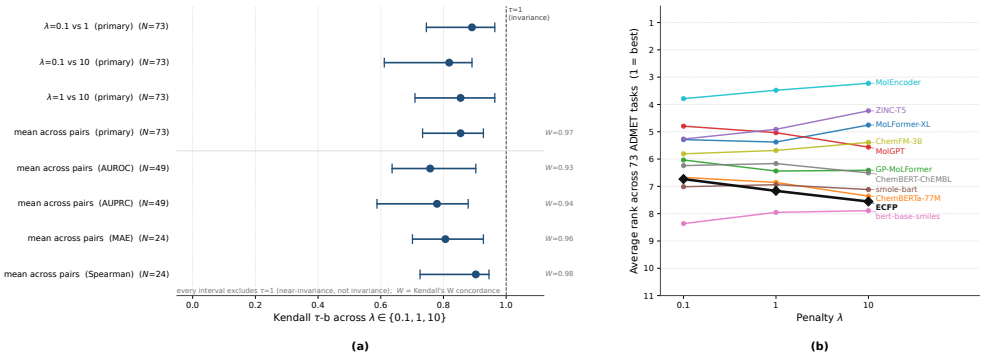


Fig. C4: The Linear Probe's ranking under three penalty strengths $\lambda \in \{0.1, 1, 10\}$. (a) How far the orderings agree across λ, by Kendall's τ-b. (b) Each model's average rank across λ.

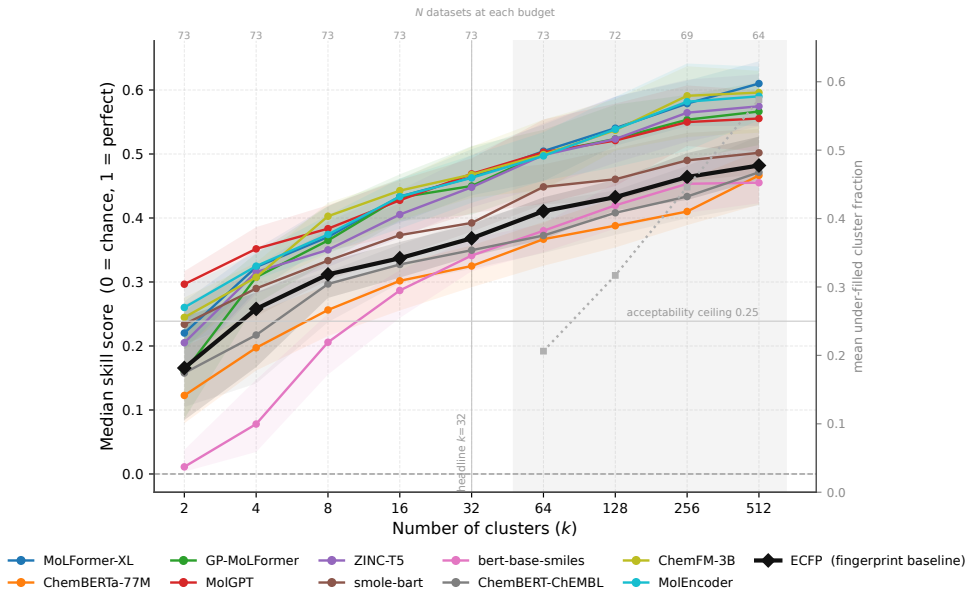


Fig. C5: The Cluster Probe as the number of groups k grows to 512. Each curve is one model's median skill across tasks; the right axis shows the share of groups left nearly empty, with $k = 32$ marked. Large k runs only on tasks big enough to support it, so the number of tasks per point is annotated.

References

[1] Kola I, Landis J (2004) Can the Pharmaceutical Industry Reduce Attrition Rates? Nature Reviews Drug Discovery 3(8):711–716. https://doi.org/10.1038/nrd1470

[2] Ferreira LLG, Andricopulo AD (2019) ADMET Modeling Approaches in Drug Discovery. Drug Discovery Today 24(5):1157–1165. https://doi.org/10.1016/j.drudis.2019.03.015

[3] Vaswani A, Shazeer N, Parmar N, et al (2017) Attention Is All You Need. In: Advances in Neural Information Processing Systems, pp 5998–6008

[4] Weininger D (1988) SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. Journal of Chemical Information and Computer Sciences 28(1):31–36. https://doi.org/10.1021/ci00057a005

[5] Ross J, Belgodere B, Chenthamarakshan V, et al (2022) Large-scale chemical language representations capture molecular structure and properties. Nature Machine Intelligence 4(12):1256–1264. https://doi.org/10.1038/s42256-022-00580-7

[6] Chithrananda S, Grand G, Ramsundar B (2020) ChemBERTa: Large-scale self-supervised pretraining for molecular property prediction. arXiv:2010.09885

[7] Sultan A, Sieg J, Mathea M, et al (2024) Transformers for molecular property prediction: Lessons learned from the past five years. Journal of Chemical Information and Modeling 64(16):6259–6280. https://doi.org/10.1021/acs.jcim.4c00747

[8] Myung Y, de Vries SJ, Nguyen TB, et al (2024) Deep-PK: deep learning for small molecule pharmacokinetic and toxicity prediction. Nucleic Acids Research 52(W1):W469–W475. https://doi.org/10.1093/nar/gkae310

[9] Cai F, Zacour K, Zhu T, et al (2026) ChemFM as a scaling law guided foundation model pre-trained on informative chemicals. Communications Chemistry 9(1). https://doi.org/10.1038/s42004-025-01793-8

[10] Hewitt J, Liang P (2019) Designing and Interpreting Probes with Control Tasks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pp 2733–2743, https://doi.org/10.18653/v1/D19-1275

[11] Voita E, Titov I (2020) Information-Theoretic Probing with Minimum Description Length. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, pp 183–196, https://doi.org/10.18653/v1/2020.emnlp-main.14

[12] Alain G, Bengio Y (2016) Understanding intermediate layers using linear classifier probes. arXiv preprint arXiv:161001644

[13] Praski M, Adamczyk J, Czech W (2025) Benchmarking pretrained molecular embedding models for molecular representation learning. arXiv preprint arXiv:250806199

[14] Rogers D, Hahn M (2010) Extended-Connectivity Fingerprints. Journal of Chemical Information and Modeling 50(5):742–754. https://doi.org/10.1021/ci100050t

[15] Stanley M, Bronskill JF, Maziarz K, et al (2021) FS-Mol: A Few-Shot Learning Dataset of Molecules. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks

[16] Wang Y, Abuduweili A, Yao Q, et al (2021) Property-Aware Relation Networks for Few-Shot Molecular Property Prediction. In: Advances in Neural Information Processing Systems, pp 17441–17454

[17] Schimunek J, Seidl P, Friedrich L, et al (2023) Context-Enriched Molecule Representations Improve Few-Shot Drug Discovery. In: International Conference on Learning Representations

[18] Li R, Li M, Liu W, et al (2025) UniMatch: Universal Matching from Atom to Task for Few-Shot Drug Discovery. In: International Conference on Learning Representations

[19] Seidl P, Vall A, Hochreiter S, et al (2023) Enhancing Activity Prediction Models in Drug Discovery with the Ability to Understand Human Language. In: Proceedings of the 40th International Conference on Machine Learning, pp 30458–30490

[20] Zhao H, Liu S, Chang M, et al (2023) GIMLET: A Unified Graph-Text Model for Instruction-Based Molecule Zero-Shot Learning. In: Advances in Neural Information Processing Systems

[21] Praski M, Adamczyk J, Czech W (2025) MolEncoder: towards optimal masked language modeling for molecules. Digital Discovery https://doi.org/10.1039/D5DD00369E

[22] Xiong G, Wu Z, Yi J, et al (2021) ADMETlab 2.0: an integrated online platform for accurate and comprehensive predictions of ADMET properties. Nucleic Acids Research 49(W1):W5–W14. https://doi.org/10.1093/nar/gkab255

[23] Wei Y, Li S, Li Z, et al (2022) Interpretable-ADMET: a web service for ADMET prediction and optimization based on deep neural representation. Bioinformatics 38(10):2863–2871. https://doi.org/10.1093/bioinformatics/btac192

[24] de Sá AGC, Long Y, Portelli S, et al (2022) toxCSM: comprehensive prediction of small molecule toxicity profiles. Briefings in Bioinformatics 23(5):bbac337. https://doi.org/10.1093/bib/bbac337

[25] Pires DEV, Blundell TL, Ascher DB (2015) pkCSM: Predicting Small-Molecule Pharmacokinetic and Toxicity Properties Using Graph-Based Signatures. Journal of Medicinal Chemistry 58(9):4066–4072. https://doi.org/10.1021/acs.jmedchem.5b00104

[26] Unikei (2022) bert-base-smiles: BERT Pretrained on SMILES with Masked Language Modeling and Molecular-Formula Validity Prediction. https://huggingface.co/unikei/bert-base-smiles, hugging Face model repository

[27] Lee J (2023) ChemBERT_ChEMBL_pretrained: Molecular BERT Pretrained on ChEMBL v33. https://huggingface.co/jonghyunlee/ChemBERT_ChEMBL_

pretrained, hugging Face model repository

[28] Ross J, Belgodere B, Hoffman SC, et al (2025) GP-MoLFormer: A Foundation Model For Molecular Generation. Digital Discovery https://doi.org/10.1039/D5DD00122F

[29] Bagal V, Aggarwal R, Vinod PK, et al (2022) MolGPT: Molecular Generation Using a Transformer-Decoder Model. Journal of Chemical Information and Modeling 62(9):2064–2076. https://doi.org/10.1021/acs.jcim.1c00600

[30] Sagawa T (2022) ZINC-t5-v2: T5 Pretrained on ZINC SMILES with Character-Level Tokenizer. https://huggingface.co/sagawa/ZINC-t5-v2, hugging Face model repository

[31] Khan S (2023) UdS-LSV/smole-bart: BART-based Molecular Language Model Pretrained on SMILES. https://huggingface.co/UdS-LSV/smole-bart, hugging Face model repository

[32] Kingma DP, Ba J (2015) Adam: A Method for Stochastic Optimization. In: International Conference on Learning Representations