# Learning Binding Affinities via Fine-tuning of Protein and Ligand Language Models

Rohan Gorantla, Aryo Pradipta Gema, Ian Xi Yang, Álvaro Serrano-Morrás, Benjamin Suutari, Jordi Juárez Jiménez, Antonia S. J. S. Mey

[1*]EaStCHEM School of Chemistry, University of Edinburgh, David Brewster Road, Edinburgh, EH9 3FJ, Midlothian, United Kingdom.
[2]School of Informatics, University of Edinburgh, David Brewster Road, Edinburgh, EH8 9AB, midlothian, United Kingdom.
[3]Exscientia, Schrödinger Building, Oxford, OX4 4GE, United Kingdom.
[4]Unitat de Fisicoquímica, Departament de Farmàcia i Tecnologia Farmacèutica, i Fisicoquímica, Facultat de Farmàcia i Ciències de l'Alimentació, Universitat de Barcelona (UB), Av. Joan XXIII, 27-31, 08028 Barcelona, Spain.
[5]Institut de Química Teòrica i Computacional (IQTC), Facultat de Química i Física, Universitat de Barcelona (UB), C. Martí i Franquès, 1, 08028 Barcelona, Spain .

Contributing authors: antonia.mey@ed.ac.uk;

**Abstract**

Accurate *in-silico* prediction of protein-ligand binding affinity is essential for efficient hit identification in large molecular libraries. Commonly used structure-based methods such as giga-docking often fail to rank compounds effectively, and free energy-based approaches, while accurate, are too computationally intensive for large-scale screening. Existing deep learning models struggle to generalize to new targets or drugs, and current evaluation methods do not reflect real-world performance accurately. We introduce **BALM**, a deep learning framework that predicts **b**inding **a**ffinity using pretrained protein and ligand **l**anguage **m**odels. BALM learns experimental binding affinities by optimizing cosine similarity in a shared embedding space. We also propose improved evaluation strategies with diverse data splits and metrics to better assess model performance. Using the

BindingDB dataset, BALM shows strong generalization to unseen drugs, scaffolds, and targets. It excels in few-shot scenarios for targets such as *USP7* and *Mpro*, outperforming traditional machine learning and docking methods, including AutoDock Vina. Adoption of the target-based evaluation methods proposed will allow for more stringent evaluation of machine learning-based scoring tools. Frameworks such as BALM show good performance, are computationally efficient, and are highly adaptable within this evaluation setting, making them practical tools for early-stage drug discovery screening.

**Keywords:** Binding affinity, Protein-ligand, Machine learning, LLMs, Docking

# Introduction

Identifying hit compounds from ultra-large libraries for accelerating target-based drug discovery hinges on the precision of *in-silico* methods for predicting protein-ligand binding affinity (BA). Structure-based approaches like giga-docking [1, 2] are commonly used for virtual screening [3, 4]. These docking-based approaches are not reliable for predicting binding affinities and typically fall short in rank-ordering hit compounds. Free energy-based methods [5–7] offer more reliable predictions but are computationally costly for hit discovery in ultra-large libraries and are restricted to lead optimization stages. Limited size and diversity of screening libraries have long been a bottleneck for the detection of novel potent ligands and for the whole process of drug discovery [8]. With recent advancements in computational libraries like Enamine [4, 9], scaling up the screening to several million to a billion compounds increases the chances of identification of more potent ligands with better physiochemical properties. This necessitates compute-efficient and accurate *in-silico* methods for effective hit identification in ultra-large libraries and identifying off-target effects of the hit compounds. Machine learning advancements, particularly deep learning (DL), are accelerating drug discovery stages, from virtual screening [10–14] for discovering hits to optimising lead compounds with active learning [15–17] and *de-novo* design [18–21] techniques.

Many deep learning strategies have been developed in recent years for binding affinity prediction [11, 12, 22–24], yet challenges remain at the model, data, and evaluation levels. At the model level, DL methods typically fall into two categories - complex-based and sequence-based approaches. Complex-based models leverage three-dimensional (3D) protein-ligand structural data, often derived from databases like PDBBind [25]. Although the use of 3D structures can potentially capture detailed interaction information, complex-based models often fail to generalize due to the limited availability of high-quality 3D complexes and biases within structural data [26]. On the sequence-based side, models are trained using 1D protein sequences and SMILES representations of ligands, making them more flexible in terms of data requirements. However, investigations into understanding what these models learn from input protein sequence and ligand SMILES data showed the dependency of these methods

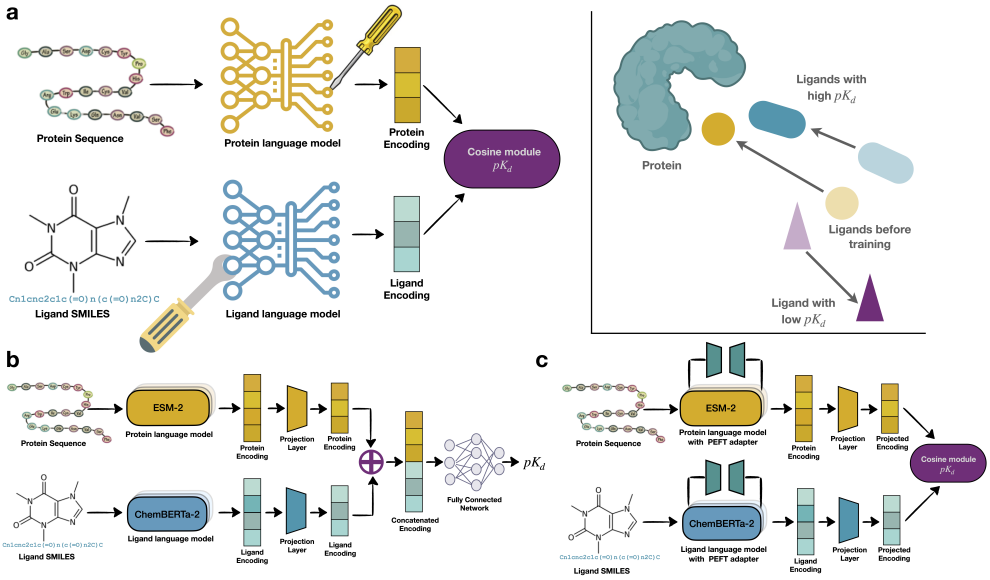


**Fig. 1 Overview of BALM architecture. (a)** BALM learns by optimizing the distance between protein and ligand encodings using a cosine similarity metric directly representing binding affinity (p$K_d$). Cosine similarity is maximized for binding interactions and minimized for non-binding interactions. Protein sequences and ligand SMILES strings are encoded using language models trained on extensive protein and ligand databases. ESM-2 [30] is used for encoding protein sequences, and ChemBERTa-2 [31] for extracting features from ligand SMILES. These encoded features are projected into a shared latent space via linear layers. The model then optimizes the cosine similarity between protein and ligand embeddings to learn binding affinity. **(b)** The baseline model is based on a pipeline from previous studies [12, 17, 23, 32, 33] where features are first extracted from protein sequences and SMILES strings using a deep learning architecture. Then these features are concatenated and passed through a linear layer to predict binding affinity. The model is trained using a Mean Squared Error (MSE) loss function to predict the binding affinity. **(c)** BALM with parameter-efficient fine-tuning (PEFT). PEFT adapters are added to the protein (LoKr) and ligand (LoHa) language models, allowing selective fine-tuning of small subsets of additional parameters while keeping the main model weights frozen. The fine-tuned embeddings are then projected and optimized using cosine similarity similar to the original BALM for affinity prediction.

on ligand information and not using both protein and ligand information for predicting [27]. Challenges at the data and evaluation levels further hinder DL model performance in binding affinity prediction. Commonly used datasets for training these DL models combine IC50 and $K_i$ data from various assays, introducing significant noise due to differences in how experiments are performed, which undermines the reliability of the model [28]. Furthermore, evaluating these models by splitting data randomly across train and test sets often results in data leakage, where similar compounds end up in both the training and test sets, leading to overestimating the model's generalizability [29].

In this work, we address these three key challenges at **model, data,** and **evaluation levels** to make the current DL models more useful for practical screening

purposes. At the model level, we introduce **BALM**, a novel sequence-based deep learning method for predicting protein-ligand **b**inding **a**ffinity using pretrained protein and ligand **l**anguage **m**odels. BALM operates on pairs of data (protein sequences, ligand SMILES) as shown in Fig. 1 to learn an embedding space that directly represents binding affinity $pK_d$. Inspired by advances in protein [30] and ligand [31] language models, which learn representations in an unsupervised manner from large-scale datasets such as UniRef [34] for proteins and PubChem [35] for ligands. BALM uses these protein (ESM-2) and ligand (ChemBERTa-2) language models to extract features from proteins and ligands and uses incorporate parameter-efficient fine-tuning (PEFT) [36–41] to adapt these embeddings specifically for binding affinity prediction, but optimizing only a small fraction of model parameters and significantly reducing computational costs. In terms of learning the feature space, BALM draws inspiration from metric learning, optimizing a distance metric (cosine similarity) to directly predict binding affinity ($pK_d$) by embedding protein-ligand pairs in a shared feature space that reflects interaction strength. While there are other metric learning-based methods for binding affinity prediction [42–44] proposed recently, which use a contrastive learning framework with triplet loss, inspired by approaches like contrastive language-image pre-training (CLIP) [45]. However, BALM differs from these approaches as it directly learns the binding strength ($pK_d$) as the distance between proteins and ligands in the feature space.

At the data and evaluation levels, we systematically investigate BALM's performance across challenging data splits within the BindingDB dataset [46, 47], including zero-shot (cold drug and cold target), scaffold, and random splits. Our experiments demonstrate that BALM consistently outperforms baseline models across all splits, showcasing robust generalization to unseen drugs, scaffolds, and targets. Incorporating PEFT further enhances BALM's performance, highlighting the benefits of parameter-efficient fine-tuning for producing reliable affinity models essential for discovery campaigns. However, our analysis reveals significant variability in zero-shot predictions across individual targets, particularly in cold target settings where protein-ligand data are sparse. This variability demonstrates the need for target-specific evaluation metrics, as aggregate metrics, such as Pearson or Spearman correlations, may obscure weak performance across diverse targets. By utilizing Fisher-transformed correlations to capture target-specific performance, we obtain a clearer view of BALM's practical applicability and limitations, offering a model evaluation approach that could inform the development and fair assessment of future deep learning methods.

To further assess BALM's adaptability, we evaluated its performance in practical few-shot learning scenarios on two challenging targets—*USP7*, a ubiquitin-specific protease [48], and *Mpro*, the main protease of SARS-CoV-2 [49, 50]—that were absent in the BindingDB dataset. Using a pretrained BALM model, we fine-tuned the embeddings for these targets, observing rapid and notable performance gains with only a small fraction of additional data, highlighting BALM's potential in real-world settings with limited experimental data. We benchmarked BALM against simple machine learning models, such as Gaussian Processes, which depend heavily on feature selection and kernel choice, impacting generalizability and performance. In contrast,

BALM's performance is robust across different targets without further tuning requirements, emphasizing its practicality in similar scenarios. We further compared BALM's performance with traditional structure-based docking methods, including AutoDock Vina [51] and rDock [52], on the Leak Proof PDBBind dataset [53], which was designed to eliminate test set data leakage. BALM outperformed these methods in predictive accuracy, showcasing its strength as a sequence-based model that circumvents the limitations of relying on 3D structural data. Lastly, we evaluated BALM on the Free Energy Benchmark [7], which includes data representative of the lead optimization stage, featuring congeneric series for three challenging targets (*MCL1*, *HIF2A*, *SYK*). On *MCL1*, BALM demonstrated competitive accuracy by few-shot fine-tuning but struggled on *HIF2A*, and *SYK*). For highly similar ligands, as in lead optimization datasets, BALM's predictions clustered within a narrower range, indicating some limitations in distinguishing fine variations in ligands. Through these contributions, BALM not only advances model robustness and efficiency but also reframes data and evaluation standards for deep learning models in protein-ligand binding affinity prediction.

# Results and Discussion

## BALM learns better across challenging data splits

We investigated the performance of the BALM model (Fig. 1a) on the BindingDB dataset [46, 47] and compared it with the Baseline model [12, 17, 23, 32, 33] using different data splits. We focused on the BindingDB dataset, specifically looking at $K_d$ binding affinities [47], which contains around 48,000 interactions involving 1,090 protein targets and 9,900 ligands. To ensure stable model training, we transformed the $K_d$ values into $pK_d$ values, following recommendations from previous studies [23, 27, 33]. To address significant bias caused by binding assay limits and the resulting skewed range of affinity predictions (Fig. A1), we cleaned the dataset by removing the top five most frequent limits. The highest assay limit at $pK_d$ of 4.99 encompassed nearly 21,000 interactions. After this cleaning step, the dataset was reduced to about 25,000 interactions, involving around 1,070 targets and 9,200 ligands. To evaluate the efficacy of each model, we consider the random, cold target, cold drug, and scaffold dataset splits, as discussed in the Data section. In the random split, interactions were randomly divided into train and test sets. The cold target split involved grouping protein targets into distinct sets to assess performance on unseen proteins, while the cold drug split assigned ligands similarly to test novel compounds. For the scaffold split, drugs were categorized by their two-dimensional murcko scaffolds [54] using RDKit's `MurckoScaffold` module [55], testing the model's ability to generalize to new chemical scaffolds. These splits provide insights into how each model performs under varying degrees of novelty and complexity in the test data. Target and drug-specific data splits are key for generalisability. We evaluated the models' ability to generalize using the concordance index (CI), Pearson correlation, Spearman rank correlation, and root mean squared error (RMSE). To compare the performance of two trained models on the test set, we assessed statistical significance using paired t-tests, with $p < 0.05$ considered significant.

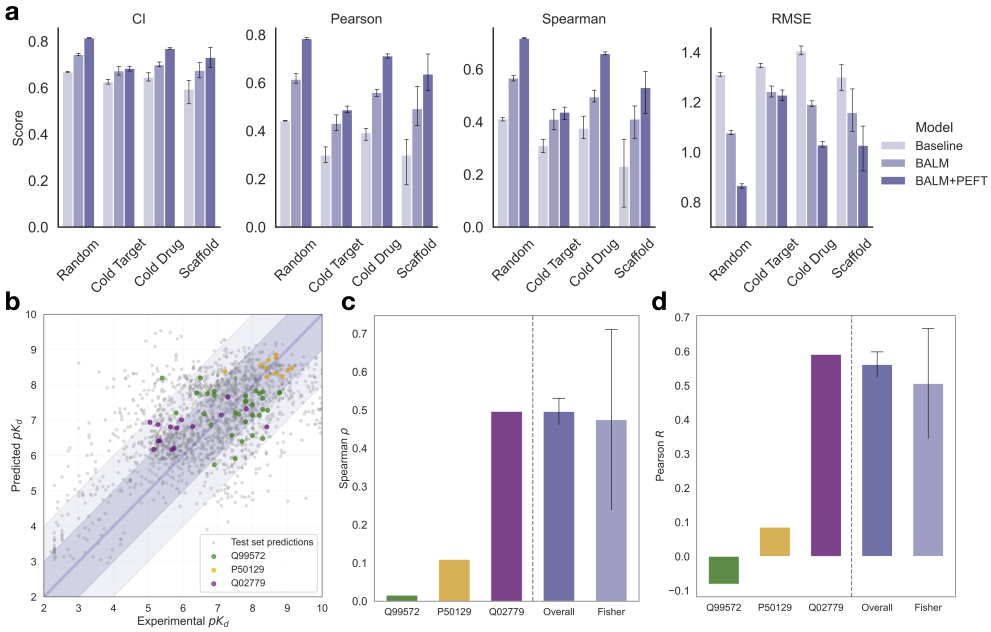


**Fig. 2 BALM performs better than the baseline model on challenging data splits, and parameter-efficient fine-tuning further improves its performance. (a)** Performance comparison of Baseline, BALM, and BALM+PEFT models on the filtered BindingDB dataset using four data splits - Random, Cold Target, Cold Drug, and Scaffold. Error bars indicate the standard deviation that was observed during three random variations in the data splits and model initializations. Across all splits, BALM+PEFT demonstrates the best performance, as reflected by increased Pearson correlation and reduced RMSE. **(b)** Scatter plot showing zero-shot predictions of the BALM+PEFT variant on three randomly selected targets from the test set in cold target split. Grey points represent predictions across the entire test set, while coloured points highlight specific targets: *Q99572* (green), *P50129* (yellow), and *Q02779* (purple). **(c)** Spearman correlation ($\rho$) bar plot for selected test targets compared to overall dataset performance and Fisher mean performance. *Overall* $\rho$ indicates the model's correlation calculated across all test set predictions, while the *Fisher* $\rho$ aggregates individual correlations for each target by averaging correlations at the target level. The *Fisher* $\rho$ provides a more realistic picture of model performance, accounting for target-specific variability and showing higher variance, as some targets may have better predictions (high Spearman) while others show worse performance (low Spearman). Larger standard deviations in *Fisher* reflect these differences across targets. **(d)** Pearson correlation ($R$) bar plot for selected test targets, similar to panel (c). The *Overall* $R$ reflects the model's global performance across the test set, while *Fisher* $R$ gives an aggregated view of target-specific correlations. *Fisher* aggregation helps highlight the performance distribution across targets, and large standard deviations for the *Fisher* suggest that some targets are predicted well while others are predicted poorly, indicating variability in the model's zero-shot performance across unseen targets.

The Baseline model (Fig. 1b) follows a pipeline from previous studies [12, 17, 23, 32, 33] where features are extracted from protein sequences and SMILES strings using a DL architecture. These features are then concatenated and passed through a linear layer to predict binding affinity, with the Mean Squared Error (MSE) loss function used to learn p$K_d$. For a fair comparison, we employ protein and ligand language models and utilize projected encodings from the projection layer similar to BALM. These encoded

representations are concatenated and passed through a fully connected layer to predict binding affinity. Both the baseline and BALM models are run with three random seeds to ensure the model performance is consistent with random variation in the data splits or initialization. The error bars on the plots represent the standard deviation observed during three random variations in the data splits or model initialization.

Fig. 2a provides an overview of the performance of the BALM and Baseline models across four data splits on each model evaluation metric used in our study. Across all four data splits, the BALM model consistently outperforms the Baseline model on each metric, demonstrating its robustness. A clear trend can be observed in the Random split, where both models exhibit significantly better performance than in the Cold and Scaffold splits. For the Random split, from the ranking metrics in Fig. 2a, we observe that BALM achieves a CI of $0.72 \pm 0.01$ and Spearman correlation of $0.61 \pm 0.01$, significantly outperforming the Baseline model (CI: $0.64 \pm 0.01$, Spearman: $0.40 \pm 0.02$) with $p = 0.007$ for CI and $p = 0.007$ for Spearman. Furthermore, BALM demonstrates a 44.6% improvement in Pearson correlation (BALM: $0.67 \pm 0.01$, Baseline: $0.46 \pm 0.02$, $p = 0.002$) and a significantly lower RMSE ($1.16 \pm 0.02$, $p = 0.007$) compared to the Baseline model ($1.46 \pm 0.03$).

To evaluate the generalisability of the models to unseen drugs or chemical scaffolds, we tested the models on cold drug and scaffold splits. In the cold drug split (Fig. 2a), BALM outperforms the Baseline model with a Pearson correlation of $0.62 \pm 0.07$ (Baseline: $0.35 \pm 0.10$, $p = 0.008$), representing a 79.8% improvement. Spearman correlation for BALM ($0.54 \pm 0.08$, Baseline: $0.30 \pm 0.09$, $p = 0.002$) also shows an 82% increase. Additionally, RMSE for BALM ($1.19 \pm 0.02$) is reduced by 18.8% compared to the Baseline model ($1.46 \pm 0.06$).

In the scaffold split, BALM exhibits strong performance, achieving a Pearson correlation of $0.55 \pm 0.07$ (Baseline: $0.27 \pm 0.05$, $p = 0.007$) and a Spearman correlation of $0.50 \pm 0.09$ (Baseline: $0.23 \pm 0.06$, $p = 0.024$). The RMSE for BALM is also lower ($1.31 \pm 0.09$) compared to the Baseline model ($1.53 \pm 0.13$). However, we observed that the variance in performance is higher in the Scaffold split than in the Cold Drug split, as evidenced by the error bars in Fig. 2a.

In the cold target split, BALM distinctly outperforms the Baseline model, demonstrating the improved ability to generalize to previously unseen protein targets. As shown in Fig. 2a, BALM achieves a CI of $0.66 \pm 0.01$ compared to the Baseline model's CI of $0.59 \pm 0.03$ ($p = 0.062$), and a Pearson correlation of $0.54 \pm 0.03$ compared to $0.26 \pm 0.10$ for the Baseline ($p = 0.023$). These results emphasize BALM's enhanced capability to generalize to new targets. However, it is important to note that the overall performance across both models is lower in the cold target split than in other splits, highlighting the inherent challenge of predicting interactions with unseen targets by these models.

Overall, the BALM model demonstrates superior performance over the Baseline model across all evaluated metrics and dataset splits. These experiments across different splits show the BALM model's enhanced reliability over the Baseline model, especially notable in its consistent ability to outperform across challenging cold and scaffold splits.

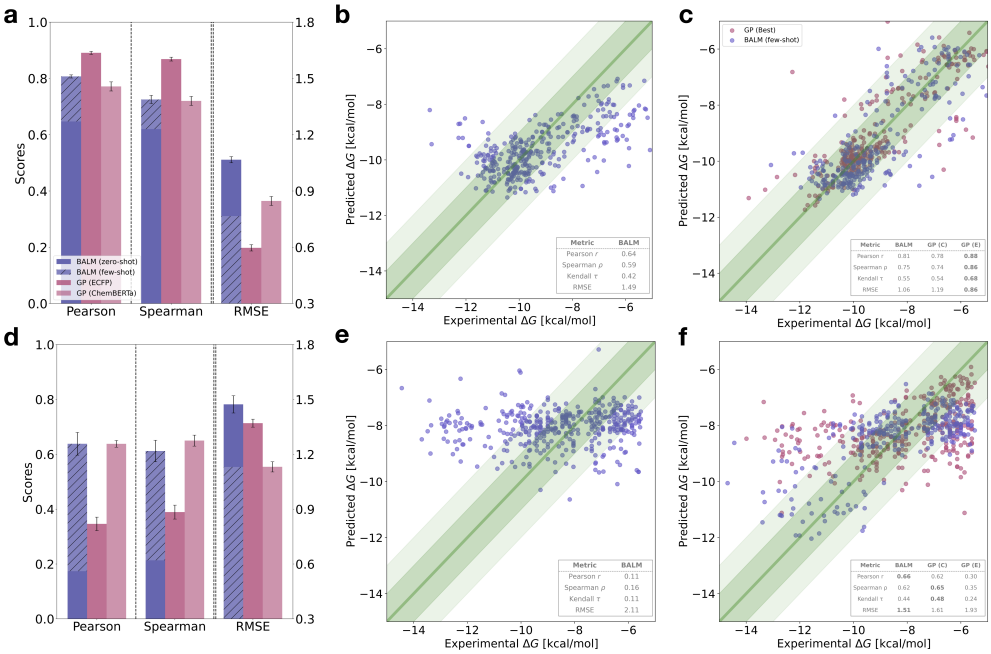


**Fig. 3 Zero-shot performance of pretrained BALM model and few-shot comparison of BALM with Gaussian Process (GP) models on *USP7* and *Mpro* targets. (a, d)** Performance metrics for Pearson correlation, Spearman correlation, and RMSE on *USP7* (a) and *Mpro* (d) targets, comparing BALM in zero-shot (solid) and few-shot (patterned) settings with GP models trained using ECFP8 fingerprints (GP (E), Tanimoto kernel) and ChemBERTa embeddings (GP (C), RBF kernel). The pretrained BALM+PEFT model is fine-tuned by retraining only the projection layer. Error bars indicate standard deviation over three random seeds. **(b, e)** Scatter plots showing zero-shot model predictions of pretrained BALM+PEFT model for *USP7* (b) and *Mpro* (e) targets. Experimental $\Delta G$ values (kcal/mol) are on the x-axis and predicted $\Delta G$ values are on the y-axis. Only 20% of the test set (selected randomly) is shown for readability. **(c, f)** Few-shot BALM predictions (blue) and GP predictions (GP (Best), red) for *USP7* (c) and *Mpro* (f) targets, highlighting BALM's robust performance across different GP configurations. We show 20% of the test set (selected randomly) for readability. GP (Best) refers to the optimal GP configuration for each target: GP (C) for *USP7* and GP (E) for *Mpro*, with metrics displayed for direct comparison.

## Parameter-efficient fine-tuning of BALM yields performance boosts

To systematically study the influence of fine-tuning methods on the BALM framework we used four different parameter-efficient adapter methods (LoRA, LoHa, LoKr, and IA3). The adapters are added to the language models as shown in Fig. 1c. We fine-tune protein and ligand language models of the BALM model, keeping their weights frozen and adding additional parameters ($< 1\%$ of pretrained parameters) via PEFT adapters. Furthermore, three of these PEFT methods (LoRA, LoHa, and LoKr) have a hyperparameter $r$, the rank of the matrix, that controls the number of additional parameters to be added during fine-tuning. Lower $r$ means fewer additional parameters. In the study, we evaluated LoRA, LoHa, and LoKr using three different ranks (8,

16, and 32) along with $IA^3$ on the BindingDB data with random splitting. We analyzed the performance of each PEFT method using the Pearson correlation metric to measure the impact of the fine-tuning protein and ligand language models. Fig. A2 shows the percentage change in Pearson correlation compared to the BALM model without fine-tuning for ligand-only, protein-only, and a combination of best fine-tuning methods for protein and ligand language models. After evaluating four PEFT methods across different ranks on the BindingDB dataset, LoHa and LoKr consistently provided the best results. LoHa (rank 16) showed the highest performance boost for ligand-only fine-tuning, improving Pearson correlation by 9.4%, while LoKr (rank 8) led protein-only fine-tuning with an 18.2% increase. Combining the best-performing methods for both ligand (LoHa) and protein (LoKr) yielded the BALM+PEFT variant, which demonstrated a substantial 23.4% improvement over the original BALM model. It highlighted the synergistic effect of fine-tuning both protein and ligand language models of the BALM framework.

From Fig. 2a, it is clear that the BALM+PEFT variant consistently outperforms the BALM model across all splits and metrics. In the random split, BALM+PEFT shows a significant 17.7% improvement in Pearson correlation (BALM+PEFT: $0.78 \pm 0.00^{+}$, BALM: $0.67 \pm 0.00^{+}$, $p = 0.0019$) and an 8.5% reduction in RMSE ($p = 0.002$). Spearman correlation improves by 17% ($p = 0.005$). In the cold target split, BALM+PEFT achieves a 21.4% increase in Pearson correlation (BALM+PEFT: $0.53 \pm 0.03$, BALM: $0.43 \pm 0.03$, $p = 0.045$) and a 6.6% reduction in RMSE. The Spearman correlation also improves by 19.1%. For the cold drug split, BALM+PEFT shows a 7.9% increase in Pearson correlation (BALM+PEFT: $0.61 \pm 0.02$, BALM: $0.56 \pm 0.02$, $p = 0.211$), a 4.4% reduction in RMSE and Spearman improves by 9.7%. Similarly, in the scaffold split, BALM+PEFT demonstrates a 15.2% improvement in Pearson correlation ($p = 0.037$) and a 16.4% increase in Spearman correlation ($p = 0.044$), with a reduction in RMSE by 7.4%.

Overall, parameter-efficient fine-tuning methods boost the performance of the BALM model, with combined fine-tuning of protein and ligand language models yielding the best results. The consistent improvements in both Pearson and Spearman correlations, along with reductions in RMSE, highlight the significant impact of parameter-efficient fine-tuning in optimizing the BALM framework.

## Zero-shot performance is not reliable for all targets, and commonly used cumulative metrics do not give the real picture in cold target-setting

We evaluated the zero-shot performance of the BALM+PEFT model in the cold target split using selected targets to understand its reliability on unseen proteins. Fig. 2b shows the zero-shot predictions for three representative targets (*Q99572*, *P50129*, and *Q02779*). The scatter plot illustrates the prediction accuracy for these targets, with significant variability in performance. For example, while the model performs well for *Q02779* (purple points), with predictions closely aligning with experimental values, it shows poorer performance for *Q99572* (green points) and *P50129* (yellow points), where predictions deviate substantially from the true values.

To further highlight the variability in target-specific performance, we use both the overall and Fisher aggregated correlations (Fig. 2c,d). The *Overall* metric, calculated across the entire test set, gives a broad view of model performance (Spearman $\rho$: 0.49, Pearson $R$: 0.51). However, as seen in Fig. 2c,d, the *Fisher* mean, which aggregates correlations for each individual target, reveals a more realistic picture by accounting for target-specific variability. The *Fisher* mean shows larger standard deviations, reflecting the high variance across targets—some targets show high correlation with experimental values, while others show poor predictions, contributing to substantial deviations in the *Fisher* correlation ($\rho$: 0.49, $R$: 0.50). These results indicate that cumulative metrics, such as overall Pearson or Spearman correlation, may overestimate the model's reliability in a cold target-setting by averaging out poor performance on difficult targets.

We further tested the BALM+PEFT model's zero-shot and few-shot performance on the *USP7* and *Mpro* targets (Fig. 3). These datasets represent challenging biological targets—*USP7*, a ubiquitin-specific protease [48], and *Mpro*, the SARS-CoV-2 main protease [49, 50]—both comprising inhibitors with experimental IC50 values converted into pIC50 values for uniformity. Both these targets were not present in the BindingDB dataset. In the zero-shot setting (Fig. 3b,e), the model shows moderate performance on these targets, as seen from the scatter plots and metrics. For *USP7*, zero-shot Pearson correlation is $r = 0.64$ with an RMSE of 1.49 kcal/mol (Fig. 3b), while for *Mpro*, Pearson correlation is notably lower at $r = 0.11$, and the RMSE is higher at 2.11 kcal/mol (Fig. 3e), showing the variability in performance across different targets.

## With few-shot fine-tuning, the BALM framework rapidly adapts to new targets and can be used for screening compound libraries

While zero-shot performance is highly variable depending on the target, we explored the adaptability of the BALM+PEFT model through few-shot fine-tuning. Using the pretrained BALM+PEFT model on BindingDB as the base model, we fine-tuned only the linear projection layer, which accounts for less than 1% of the model's parameters (262,656 out of 152.5 million total parameters, or approximately 0.17%). This makes fine-tuning computationally efficient while ensuring fast adaptation to new targets. Importantly, fine-tuning only a small fraction of the model's parameters provides a reasonable strategy for screening compound libraries in new drug discovery efforts, as it minimizes training time while achieving significant performance gains.

To benchmark BALM's few-shot performance, we compared it against Gaussian Process [56] (GP) models, commonly used in few-shot settings like active learning [15–17]. In scenarios where experimental data is limited, typical of real-world drug discovery campaigns—active learning allows a machine learning model to iteratively select informative data points labelled using computationally expensive methods, such as free energy calculations. GP models are often employed as surrogates in such workflows as they can make reliable predictions with limited data [16, 17]. For this comparison, we trained GP models using two different molecular representations-Extended-Connectivity Fingerprints (ECFP8) paired with a Tanimoto kernel and ChemBERTa embeddings paired with a radial basis function (RBF) kernel. The choice

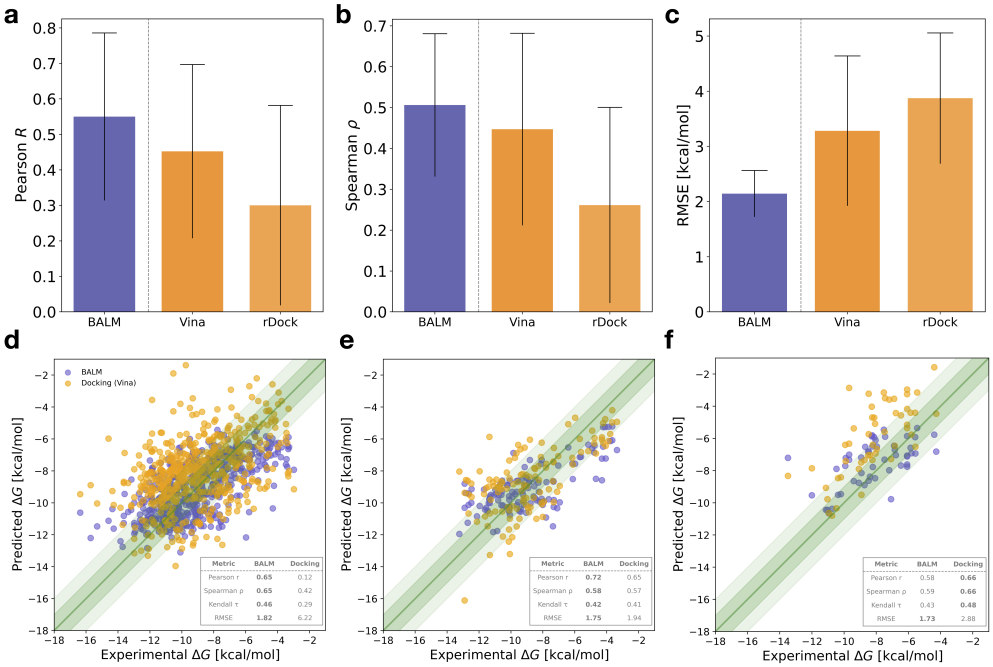


**Fig. 4 Evaluation of BALM versus docking methods, AutoDock Vina [51] and rDock [52], across various target families in the LP-PDBBind [53] test split.** The test set includes approximately 2,100 protein-ligand complexes spanning 12 diverse target families, with results shown here for representative families. Fisher-transformed metrics are used for comparing Pearson and Spearman correlations across target families, providing an accurate performance measure that accounts for variability within each family. AutoDock Vina consistently performs better than rDock across most target families. **(a)** Bar plot of Pearson correlation for each method across all families; error bars represent the standard deviation across target families. **(b)** Spearman correlation, showing the ranking accuracy of each method across families with similar error bars. **(c)** RMSE, directly comparing the predictive accuracy of each method, where lower values indicate higher precision in binding affinity predictions. **(d-f)** Scatter plots showing the performance of BALM (blue) versus AutoDock Vina (orange) for different target families. **(d)** For *Transferase* targets (545 PDB complexes), BALM achieves superior Pearson and Spearman correlations compared to Vina, demonstrating its effectiveness in predicting kinase-related binding affinities. **(e)** For *Chaperone* targets (119 PDB complexes), BALM outperforms Vina, especially in RMSE, which suggests higher prediction stability for this family. **(f)** In the case of *Oxidoreductase* targets (52 PDB complexes), docking methods show slightly better ranking performance, while BALM achieves a lower RMSE, indicating more precise binding affinity predictions for this target class.

of features and kernel substantially impacts GP model performance, but these choices are not known as *apriori*. Thus, the GP models provide a benchmark for BALM in few-shot scenarios, emphasizing BALM's advantage of achieving reliable performance without the need for such hyperparameter tuning.

We evaluated the few-shot performance on two challenging biological targets—*USP7* (Fig. 3a,b,c), a ubiquitin-specific protease, and *Mpro* (Fig. 3d,e,f), the main protease of SARS-CoV-2. As seen in Fig. 3a and d, BALM exhibits notable improvements in Pearson and Spearman correlations and a reduction in RMSE with

few-shot fine-tuning using 20% of the data for training and the rest for testing. All the models are trained and tested with three different random seeds. In the zero-shot setting, BALM's Pearson correlation is 0.64 for *USP7* (Fig. 3a) and 0.11 for *Mpro* (Fig. 3d) that improves when trained on a few labelled examples to a Pearson correlation of 0.81 for *USP7* and 0.66 for *Mpro*, with RMSE values decreasing from 1.49 kcal/mol to 1.03 kcal/mol for *USP7* and from 2.11 kcal/mol to 1.51 kcal/mol for *Mpro*. These results demonstrate BALM's ability to achieve robust performance with minimal additional data and fine-tuning on a small subset of parameters.

The GP models, while showing reasonable accuracy, display sensitivity to the choice of molecular representation. As shown in Fig. 3c and f, the GP model's performance varies depending on the selected fingerprint. For *USP7*, the GP model with ECFP8 fingerprints achieves a Pearson correlation of 0.88 and RMSE of 0.86 kcal/mol, while with ChemBERTa embeddings, the correlation drops slightly to 0.75 with an RMSE of 1.19 kcal/mol. For *Mpro*, the performance variation is even more pronounced, where the ECFP8-based GP model attains a Pearson correlation of 0.62, compared to 0.48 with ChemBERTa, while RMSE values range from 1.21 kcal/mol to 1.94 kcal/mol, respectively. This variability suggests that GP models may require careful feature selection to achieve optimal results, a process that may not generalize well across different targets. BALM's consistent performance across different targets and its independence from feature-specific tuning make it a reliable option for few-shot learning. The GP models, while competitive with specific features on each target with BALM, demonstrate the limitation of these models in reliably generalizing across diverse molecular landscapes without careful pre-selection of molecular representations.

In terms of computational efficiency, the time taken for zero-shot predictions on *USP7* and *Mpro* is approximately 1 minute 30 seconds (for around 2,000 ligands), while fine-tuning the model with 10% to 30% of training data takes between 14 and 25 minutes, depending on the training ratio on a Nvidia A100 GPU. BALM's combination of zero-shot capability and rapid improvement with few-shot training positions it as a robust tool for drug discovery screening scenarios, where rapid adaptation to novel targets is essential for high-throughput drug discovery with minimal dependency on feature engineering.

## BALM goes beyond Docking scores and generalizes well to various target families

We investigate the performance of the BALM model (with PEFT) on the LP-PDBBind dataset [53] and compare it with two docking methods, AutoDock Vina [51] and rDock [52]. LP-PDBBind dataset, derived from PDBBind v2020 [57], consists of protein-ligand complexes with experimentally measured binding affinities. LP-PDBBind is specifically designed to minimize similarities in sequence, structure, and ligand chemistry across training, validation, and test sets, allowing the testing of the model's generalizability. For our evaluation, we employ the Clean Level 2 (CL2) split, as suggested by Li et al. [53], due to its higher structural reliability. The test set comprises around 2,100 protein-ligand complexes from 12 protein families.

Docking methods, rDock [52] and AutoDock Vina [51] use 3D protein-ligand complex structures to provide binding affinity, whereas BALM requires only 1D sequence

and SMILES representations of proteins and ligands. This distinction is crucial, as BALM bypasses the need for precise 3D structural information, making it applicable in scenarios where such data is unavailable.

In Fig. 4, we compare BALM with docking methods using bar plots for fisher-transformed Pearson (Fig. 4a) and Spearman correlations (Fig. 4b), as well as RMSE (Fig. 4c) across target families. The results show that BALM consistently achieves higher Fisher-transformed Pearson and Spearman correlations compared to both docking methods, indicating stronger predictive and ranking performance across families. Among the docking methods, AutoDock Vina generally exhibits better performance than rDock in both correlations, suggesting more stable ranking accuracy. In terms of predictive accuracy, as measured by RMSE, BALM achieves the lowest RMSE across families, with Vina displaying higher variability but better consistency than rDock.

Next, we examine individual target families, comparing the performance of BALM (blue) versus Vina (orange) on three representative target families Fig. 4d-f, with further comparisons provided in Appendix Fig. A4 (BALM vs. Vina) and Fig. A5 (BALM vs. rDock) across all 12 families. For *Transferase* targets (Fig. 4d), BALM demonstrates strong predictive accuracy with a Pearson correlation of 0.65 and an RMSE of 1.82 kcal/mol, outperforming Vina, which has a Pearson correlation of 0.12 and an RMSE of 6.22 kcal/mol. In the case of *Chaperone* proteins (Fig. 4e), BALM outperforms Vina in ranking with a Pearson correlation of 0.72 versus Vina's 0.65, and has a significantly lower RMSE of 1.75 kcal/mol compared to Vina's 1.94 kcal/mol. However, the ranking metrics for BALM ($\rho = 0.58$, $\tau = 0.42$) and Vina ($\rho = 0.57$, $\tau = 0.41$) are similar. For *Oxidoreductase* targets ( Fig. 4f), Vina surpasses BALM in ranking metrics (Spearman $\rho = 0.66$ vs. 0.59 for BALM), but BALM achieves a lower RMSE of 1.73 kcal/mol versus Vina's 2.88 kcal/mol.

For other families (Fig. A4), in challenging *Metal-containing* targets, BALM achieves a Spearman $\rho$ of 0.58 with an RMSE of 2.33 kcal/mol, outperforming Vina, which has a Spearman $\rho$ of 0.17 and an RMSE of 4.71 kcal/mol. In the *Membrane* target family, BALM's Spearman $\rho$ is 0.40, higher than Vina's 0.35, with a notably lower RMSE of 2.11 kcal/mol compared to 2.84 kcal/mol for Vina, indicating a more stable predictive performance. A similar trend is observed for *viral* protein targets, with BALM having a higher ranking correlation and lower RMSE. BALM and Vina display similar ranking performances for families such as *Lyase* and *Transcription* factors. For instance, in the *Lyase* family, both methods show comparable Spearman $\rho$ values (0.38 for BALM and 0.40 for Vina), but BALM retains a significantly lower RMSE of 2.84 kcal/mol compared to 4.42 kcal/mol for Vina. This trend persists in the *Transcription* factor family, where BALM and Vina achieve similar Spearman $\rho$ values (0.41 and 0.38, respectively), yet BALM's RMSE is considerably lower at 1.49 kcal/mol versus 2.03 kcal/mol for Vina. In the *Isomerase* and *Ligase* families, Vina surpasses BALM in ranking correlations, although BALM consistently achieves a lower RMSE. For example, in the *Ligase* family, Vina attains a Spearman $\rho$ of 0.64 compared to BALM's 0.57, yet BALM achieves a more precise RMSE of 2.23 kcal/mol, contrasting with Vina's 2.10 kcal/mol.

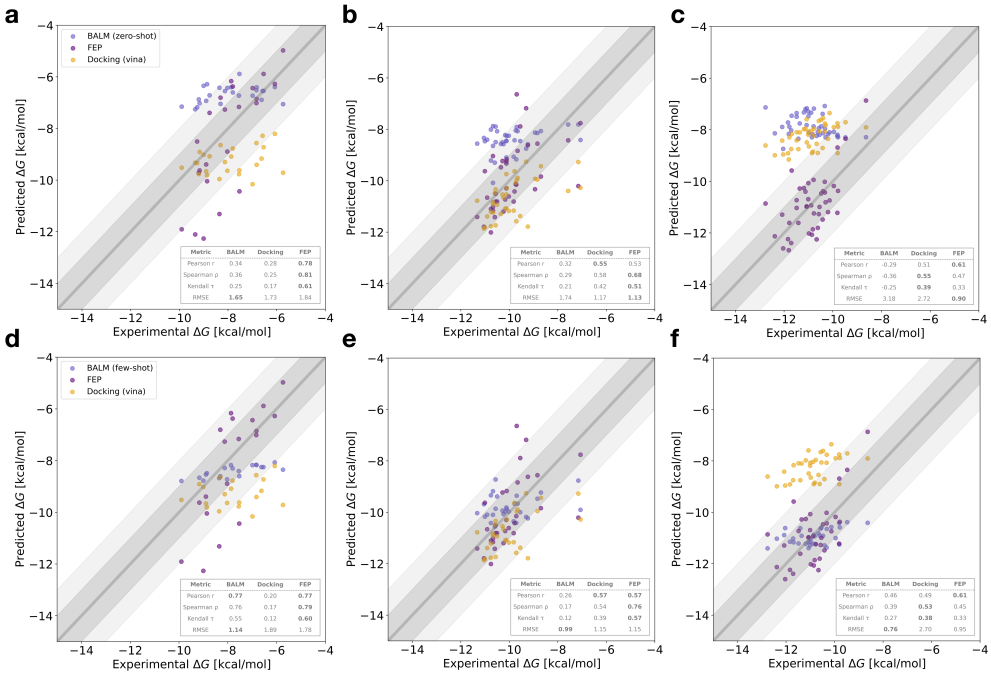


**Fig. 5 Comparing BALM, Alchemical Free Energy (AFE) predictions, and AutoDock Vina across three benchmark protein-ligand targets.** The benchmark includes *MCL1* (25 ligands), *HIF2A* (37 ligands), and *SYK* (43 ligands). Panels **(a, b, c)** present the zero-shot predictions from BALM (blue) alongside AFE (yellow) and docking (orange) for each target. **(d, e, f)** illustrate the few-shot performance of BALM, fine-tuned with approximately 20% of the data. Each plot displays metrics for Pearson correlation, Spearman $\rho$, Kendall $\tau$, and RMSE.

## BALM on free energy benchmark datasets

We evaluate the performance of the BALM model, both in zero-shot and few-shot settings, on a subset of targets from the protein-ligand free energy benchmark curated by Hahn et al. [7]. This benchmark is composed of a congeneric series of ligands, which are structurally similar compounds with systematic variations. Alchemical free energy methods, and in paritcular relative binding free energy (RBFE) methods based on molecular dynamics simulations and leveraging 3D structural information, is commonly applied in such contexts [6].

Specifically, we selected three targets, *MCL1*, *HIF2A*, and *SYK*, containing 25, 37, and 43 unique compounds, respectively, to compare BALM's performance against RBFE methods run as part of the OpenFE benchmark [58]. Additionally, we evaluated docking performance on these targets using AutoDock Vina, by rescoring the 3D poses. For BALM, we used a pretrained BALM+PEFT variant on the BindingDB dataset. Fig. 5 shows BALM's performance in zero-shot (a, b, c) and few-shot (d, e, f) modes for each target.

In the zero-shot setting (panels a, b, c), BALM's performance is mixed across the targets. On *MCL1* (Fig. 5a), BALM achieves a Spearman correlation of 0.36 and an

RMSE of 1.65 kcal/mol, which is competitive with Vina (Spearman $\rho = 0.25$, RMSE = 1.73 kcal/mol) but still below the precision of RBFE (Spearman $\rho = 0.58$, RMSE = 1.13 kcal/mol). This suggests BALM's capacity to predict within a moderate error range, even without fine-tuning, although models based on 3D structural information auch as alchemical free energy methods and docking capture binding nuances more effectively. For *HIF2A* (Fig. 5b), BALM achieves a Pearson correlation of 0.32 and a Spearman $\rho$ of 0.29, which is far below both RBFE and docking predictions, where free energy methods reach a Spearman $\rho$ of 0.63 and an RMSE of 1.13 kcal/mol while docking's Spearman $\rho$ is 0.58 and RMSE of 1.17 kcal/mol. Similarly, on the *SYK* target (Fig. 5c), BALM falls short of AFE's (Spearman $\rho = 0.47$) and Vina's (Spearman $\rho = 0.55$) ranking correlation. We can observe that BALM's predictions cluster within a narrow binding affinity range, indicating potential difficulty in distinguishing small R-group modifications in the congeneric series.

In the few-shot setting (panels d, e, f), where BALM is fine-tuned on approximately 20% of the data (around 5 ligands for *MCL1*, 7 for *HIF2A*, and 8 for *SYK*), BALM shows improvements in RMSE across all targets however the rank correlation trends are different across each target. For *MCL1* (Fig. 5d), BALM's few-shot fine-tuning increases its Spearman $\rho$ to 0.76 and Pearson correlation to 0.77, surpassing both RBFE and docking in ranking correlations, with a reduced RMSE of 1.14 kcal/mol. For *SYK* (Fig. 5f), BALM's few-shot performance yields a Spearman $\rho$ of around 0.4, RBFE at 0.45 and docking via rescoring of the pose at 0.53 still perform better. The RMSE for BALM also drops to 0.76 kcal/mol, however this may be due to the model trying to optimize on its loss function and as a result effectively 'cheats'. This can be seen by the very narrow band of free energy predictions. Interestingly, on *HIF2A* (Fig. 5e), few-shot fine-tuning worsens BALM's Pearson and Spearman correlations compared to its zero-shot performance, and the RMSE shows similar trends as the other targets. These results hint at BALM's limitations in the context of data which is similar during the lead optimization. While few-shot learning helps reduce prediction error, the model may still struggle with fine-grained differentiation in binding energy for closely related ligands in this dataset. The narrowing range of predictions in the few-shot setting, as shown in the scatter plots, indicates that BALM may optimize for RMSE at the cost of capturing the full dynamic range of binding energies. RBFE methods, that leverage both dynamics and 3D structure, still have a clear advantage over the SMILEs-based BALM approach for subtle ligand changes.

With clear strategies on how to assess the usefulness of ML or physics-based scoring functions introduced in this paper, future benchmarks should follow such strategies to truly assess the new scoring methods such as BALM. This will then enable the assessment of various scoring strategies from physics-based docking methods to 1D and 3D structure-based machine learning frameworks such as BALM.

# Methods

**Model overview and objective.** BALM is designed to predict protein-ligand binding affinity by taking ligand SMILES strings and protein sequences as the input. Our method begins by encoding the protein sequences and ligand SMILES by integrating

protein and ligand language models. These language models trained on extensive ligand [35] and protein databases [34] encapsulate the chemical properties inherent in SMILES strings and physicochemical, functional, and evolutionary patterns in protein sequences. The encodings obtained from the language models are then projected through a linear layer to a shared latent space. The core idea behind BALM is to maximize the cosine similarity between protein-ligand embeddings for a binding interaction and minimize the same for non-binding interactions, thus learning protein-ligand interaction features. The model learns to understand the relationship between protein and ligand embeddings and quantifies it through a cosine similarity score, a proxy for the binding affinity score ($\text{pK}_d$). The model learns by minimizing the mean-squared error (MSE) loss.

**Encoding and projecting protein and ligand features.** We use ESM-2 [30] model for encoding protein features and ChemBERTa-2 [31] for encoding ligands. It is important to note that our focus is not to evaluate the performance of protein or ligand language models but to leverage their learned representations for BA prediction. ESM-2 model is trained on sequences from the UniRef database [34] with more than 220 million (unaligned) sequences using bidirectional transformer architecture [59]. ESM-2 learns in an unsupervised manner through masked language modelling (MLM) objective, where 15% of amino acids are masked in the input sequence, and it is tasked with predicting these missing positions. We use the 150 million parameter model as it strikes a balance between computational efficiency and performance. ChemBERTa-2 is based on the RoBERTa [60] transformer model, using both MLM similar to ESM-2 by masking 15% tokens and multi-task regression (MTR) pretraining on 200 downstream molecular properties. In our work, we use the model trained on the largest dataset size using the MTR objective as it outperformed the other model configurations in the comparative study [31]. For a given input protein sequence, $I_P$, and ligand SMILES, $I_L$, we encode and project to get embeddings of the same dimensions as shown below:

$$\mathbf{x}_{\text{P}} = \mathbf{W}_{\text{P}} \cdot f(I_{\text{P}}) + \mathbf{b}_{\text{P}}, \tag{1}$$

$$\mathbf{x}_{\text{L}} = \mathbf{W}_{\text{L}} \cdot g(I_{\text{L}}) + \mathbf{b}_{\text{L}}. \tag{2}$$

Here, $f(I_{\text{P}}) \in \mathbb{R}^{d_{\text{P}}}$ extracts the protein features using ESM-2 and similarly, $g(I_{\text{L}}) \in \mathbb{R}^{d_{\text{L}}}$ extracts the ligand features using ChemBERTa-2. We then transform them separately into $\mathbf{x}_{\text{P}}, \mathbf{x}_{\text{L}} \in \mathbb{R}^K$ using a single fully connected layer (with a ReLU activation). These layers are parameterized with weight matrices $W_{\text{P}} \in \mathbb{R}^{K \times d_{\text{P}}}$ for targets and $W_{\text{L}} \in \mathbb{R}^{K \times d_{\text{L}}}$ for ligands, and bias vectors $b_{\text{P}}, b_{\text{L}} \in \mathbb{R}^K$. As the encodings derived from the ESM-2 and the ChemBERTa-2 model have different dimensions, 640 (150M model) and 384, respectively, we employ linear layers to map both sets of embeddings to a shared latent space of dimension $K$ (256) suitable for similarity computations. Both ESM-2 and ChemBERTa-2 models are accessible via the Hugging face transformers library [61], under the model names `facebook/esm2_t30_150M_UR50D`, and `DeepChem/ChemBERTa-77M-MTR`, respectively.

**Training with cosine similarity and MSE.** In the training phase of BALM, the cosine similarity metric is used to measure the affinity between the projected protein and ligand embeddings. BALM utilizes cosine similarity as the core metric for learning protein-ligand binding as a regression problem. BALM defines the binding score by

computing the cosine similarity of the protein and ligand embeddings:

$$\sigma(\mathbf{x}_{\mathrm{P}}, \mathbf{x}_{\mathrm{L}}) = \frac{\mathbf{x}_{\mathrm{P}} \cdot \mathbf{x}_{\mathrm{L}}}{\|\mathbf{x}_{\mathrm{P}}\| \|\mathbf{x}_{\mathrm{L}}\|}, \tag{3}$$

where $\mathbf{x}_{\mathrm{P}}$ and $\mathbf{x}_{\mathrm{L}}$ denote the embeddings of the protein and ligand, respectively. The cosine similarity $\sigma$ ranges from -1 to 1, with values closer to 1 indicating a stronger binding (high $\mathrm{pK}_d$). $\mathbf{x}_{\mathrm{P}}^i$ and $\mathbf{x}_{\mathrm{L}}^i$ are the projected embeddings of the protein and ligand, respectively. BALM is trained by minimizing the MSE between predicted binding scores given by $\sigma(\mathbf{x}_{\mathrm{P}}, \mathbf{x}_{\mathrm{L}})$ and experimental affinities ($y_i$) adjusted to the -1 to 1 scale :

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left( y_i - \sigma(\mathbf{x}_{\mathrm{P}}^i, \mathbf{x}_{\mathrm{L}}^i) \right)^2, \tag{4}$$

where $N$ is the number of samples.

**Parameter-efficient fine-tuning on BALM.** PEFT allows adapting pretrained language models to new domains by fine-tuning a small subset of additional parameters while keeping the initial model parameters fixed. This reduces computational requirements and prevents catastrophic forgetting. Given a pretrained model $T$ with parameters $\theta$, and a downstream task $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, where $(x_i, y_i)$ represents an input and ground-truth pair for task $\mathcal{D}$, PEFT introduces a small set of trainable parameters $\Delta\theta$, where $|\Delta\theta| \ll |\theta|$. The objective is to adapt $\theta$ to task $\mathcal{D}$ by optimizing:

$$\min_{\Delta\theta} \mathbb{E}_{(x_i, y_i) \in \mathcal{D}} \mathcal{L}(T_{\theta + \Delta\theta}(\hat{y}_i | x_i), y_i), \tag{5}$$

where $\hat{y}_i$ denotes the predicted affinities and $\mathcal{L}$ denotes the MSE loss function reflecting the model's performance on task $\mathcal{D}$. Various PEFT methods have been proposed, for more details refer to the survey papers [36, 37].

In this work, we use the reparameterised and additive PEFT methods [36, 37] to fine-tune BALM's protein (ESM-2) and ligand (ChemBERTa-2) language models for binding affinity prediction. The reparameterized PEFT methods we used in the study include Low-Rank Adaptation (LoRA) [62], LoRA with Hadamard product (LoHa) [63], and LoRA with Kronecker product (LoKr) [64], applied to the key, query, and value matrices. An additive PEFT method, Infused Adapter by Inhibiting and Amplifying Inner Activations ($\mathrm{IA}^3$) [65], is also studied for fine-tuning BALM. These methods are implemented using the Hugging face `PeftModel` class [61, 66]. We use four PEFT methods in our study, and these are discussed in detail in the Appendix B.

## Data for the study

To benchmark the performance of our models, we utilized several publicly available datasets, including BindingDB [46, 47], LP-PDBBind [53], and other datasets specific to protein-ligand systems such as *USP7* [48], *Mpro* [49, 50], and three targets from protein-ligand free energy benchmark [7] - *MCL1*, *HIF2A*, and *SYK*. These datasets encompass a wide range of binding affinity measurements and chemical diversity, enabling us to compare with docking and alchemical free energy methods. These

datasets provide a comprehensive benchmark for strategically evaluating the machine learning model's capability to predict binding affinity under various conditions, including zero-shot and few-shot scenarios. We release all the cleaned datasets used in our work as `BALM-Benchmark` on Hugging face [61].

**BindingDB** dataset provides experimentally measured binding affinities of protein-ligand interactions, we focus on the $K_d$ version [47] due to inconsistencies in IC50 and $K_i$ measures [28]. This version contains 52,284 interactions with 10,665 ligand SMILES and 1,413 protein sequences. We filter out sequences with more than 1024 residues for computational efficiency with the ESM-2 model [30]. The $K_d$ values were transformed into $pK_d$ for stable training, following previous works [23, 27, 33]. We were left with approximately 1100 targets and 48,000 interaction data after the filtering. Additionally, to avoid bias from assay limits (Fig. A1), we removed the top five most frequent limits. This reduced the dataset to about 25,000 interactions involving approximately 1,070 targets and 9,200 ligands. We used around 70% of these interaction data for training, 10% interactions for validation, and 20% interactions for testing. Four data splits (Random, Cold Target, Cold Drug, and Scaffold) were used to evaluate model generalizability. In the *random split*, commonly used in prior works [23, 27, 33], protein-ligand interaction pairs are randomly distributed across training, validation, and test sets, ensuring each subset is a statistically representative sample. The *cold target split* segregates protein targets into distinct groups for training, validation, and testing, allowing evaluation on unseen protein targets. Similarly, the *cold drug split* randomly allocates drugs to training, validation, and test sets, assigning all protein-ligand pairs linked to each ligand to the corresponding set, enabling performance evaluation on unseen drugs. Lastly, the *scaffold split* employs the Murcko scaffolds [54] concept, using the `MurckoScaffold` module from RDkit [55] to bin drugs by their core scaffolds, ensuring no structural overlap across sets. Approximately 4570 scaffolds were used from the BindingDB dataset for this purpose.

**LP-PDBBind** dataset is derived from the publicly available PDBBind v2020 [57] dataset, a curated set of approximately 20,000 protein-ligand complex structures (3D) with experimentally measured binding affinities. The dataset has been reorganized to minimize sequence, structural interaction patterns, and chemical similarity across training, validation, and test splits [53]. It has also been cleaned to remove covalent-bound ligand-protein complexes, ligands with rare atomic elements, and structures with steric clashes while maintaining consistency in reported binding free energies. Recently, [67] provided more detailed analysis of the LP-PDBBind's data splits. Furthermore, the dataset has been categorized based on the quality of structures into Clean Levels 1, 2, and 3, consisting of 14,324, 7,985, and 4,404 entries, respectively. For our purposes, we utilize CL1 for training, while CL2 is employed for validation and testing due to its higher reliability, as suggested by Li et al. [53].

**USP7** dataset is curated by Shen et al. [48], evaluates inhibitors targeting the ubiquitin-specific protease 7 (*USP7*). The dataset consists of over 4000 ligands with associated binding affinities collected from ChEMBL [68]. After processing, the final dataset comprises 1,799 unique ligands with measured experimental affinities, represented as IC50 values. These values were then transformed into pIC50 values for uniformity and stability in training.

**Mpro** dataset is derived from the COVID Moonshot project focusing on inhibitors of the SARS-CoV-2 main protease (*Mpro*) [49, 50], includes data from multiple design sprints, capturing both racemic mixtures and enantiomers. We filtered to remove assay limits, and the cleaned dataset contains 2,062 unique ligands with experimentally determined IC50 values. We converted IC50 values to pIC50 values similar to the *USP7* target. Further details about the data curation can be found here [49, 50].

**Protein-ligand free energy benchmark** curated by Hahn et al. [7] contains about 21 targets for benchmarking alchemical free energy calculations. We selected three targets, *MCL1*, *HIF2A*, and *SYK*, to benchmark the machine learning model performance as compared to the alchemical energy calculations [58]. The *HIF2A* dataset contains 37 ligands, the *MCL1* dataset includes 25 ligands, and *SYK* dataset consists of 43 unique ligands.

**Supplementary information.** Supporting Information is provided with additional experimental results and methods studied in this work.

**Acknowledgements.** RG and APG were supported by the United Kingdom Research and Innovation (grant EP/S02431X/1), UKRI Centre for Doctoral Training in Biomedical AI at the University of Edinburgh, School of Informatics. RG is also supported by Exscientia Plc, Oxford. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY NC) licence to any author-accepted manuscript version arising. Experiments from this work are conducted mainly on the Edinburgh International Data Facility[1] and supported by the Data-Driven Innovation Programme at the University of Edinburgh.

**Data availability.** All datasets curated for this study are publicly available via the Hugging Face `BALM-Benchmark` collection. The datasets can be accessed at https://huggingface.co/datasets/BALM/BALM-benchmark, and the pretrained models are accessible on Hugging Face at https://huggingface.co/BALM.

**Code availability.** All the code containing scripts for data processing, model training, and evaluation is publicly accessible on GitHub at https://github.com/meyresearch/BALM

# References

[1] Lyu, J. *et al.* Ultra-large library docking for discovering new chemotypes. *Nature* **566**, 224–229 (2019).

[2] Stanzione, F., Giangreco, I. & Cole, J. C. Use of molecular docking computational tools in drug discovery. *Prog. Med. Chem.* **60**, 273–343 (2021).

[3] Shoichet, B. K. Virtual screening of chemical libraries. *Nature* **432**, 862–865 (2004).

[1] https://edinburgh-international-data-facility.ed.ac.uk/

[4] Beroza, P. *et al.* Chemical space docking enables large-scale structure-based virtual screening to discover rock1 kinase inhibitors. *Nat. Commun.* **13**, 6447 (2022).

[5] Robo, M. T., Hayes, R. L., Ding, X., Pulawski, B. & Vilseck, J. Z. Fast free energy estimates from $\lambda$-dynamics with bias-updated gibbs sampling. *Nat. Commun.* **14**, 8515 (2023).

[6] Mey, A. S. *et al.* Best practices for alchemical free energy calculations. *Living J. Mol. Sci.* **2**, 18378 (2020).

[7] Hahn, D. F. *et al.* Best practices for constructing, preparing, and evaluating protein-ligand binding affinity benchmarks. *Living J. Mol. Sci.* **4**, 1497–1497 (2022).

[8] Sadybekov, A. V. & Katritch, V. Computational approaches streamlining drug discovery. *Nature* **616**, 673–685 (2023).

[9] Grygorenko, O. O. *et al.* Generating multibillion chemical space of readily accessible screening compounds. *Iscience* **23** (2020).

[10] Hadfield, T. E., Scantlebury, J. & Deane, C. M. Exploring the ability of machine learning-based virtual screening models to identify the functional groups responsible for binding. *J. Cheminformatics* **15**, 84 (2023).

[11] Zhang, Y., Li, S., Meng, K. & Sun, S. Machine learning for sequence and structure-based protein–ligand interaction prediction. *J. Chem. Inf. Model.* (2024).

[12] Kimber, T. B., Chen, Y. & Volkamer, A. Deep learning in virtual screening: recent applications and developments. *Int. J. Mol. Sci.* **22**, 4435 (2021).

[13] Smer-Barreto, V. *et al.* Discovery of senolytics using machine learning. *Nat. Commun.* **14**, 3445 (2023).

[14] Rodriguez, S. *et al.* Machine learning identifies candidates for drug repurposing in alzheimer's disease. *Nat. Commun.* **12**, 1033 (2021).

[15] Gusev, F., Gutkin, E., Kurnikova, M. G. & Isayev, O. Active learning guided drug design lead optimization based on relative binding free energy modeling. *J. Chem. Inf. Model.* **63**, 583–594 (2023).

[16] Thompson, J. *et al.* Optimizing active learning for free energy calculations. *Artif. Intell. Life Sci.* **2**, 100050 (2022).

[17] Gorantla, R., Kubincova, A., Suutari, B., Cossins, B. P. & Mey, A. Benchmarking active learning protocols for ligand binding affinity prediction. *J. Chem. Inf. Model.* **64**, 1955–1965 (2024).

[18] Huang, L. *et al.* A dual diffusion model enables 3d molecule generation and lead optimization based on target pockets. *Nat. Commun.* **15**, 2657 (2024).

[19] Anstine, D. M. & Isayev, O. Generative models as an emerging paradigm in the chemical sciences. *J. Am. Chem. Soc.* **145**, 8736–8750 (2023).

[20] Sattari, K. *et al.* De novo molecule design towards biased properties via a deep generative framework and iterative transfer learning. *Digit. Discov.* (2024).

[21] Runcie, N. T. & Mey, A. S. Silvr: Guided diffusion for molecule generation. *J. Chem. Inf. Model.* **63**, 5996–6005 (2023).

[22] Chatterjee, A. *et al.* Improving the generalizability of protein-ligand binding predictions with ai-bind. *Nat. Commun.* **14**, 1989 (2023).

[23] Öztürk, H., Özgür, A. & Ozkirimli, E. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics* **34**, 821–829 (2018).

[24] Jiménez, J., Skalic, M., Martinez-Rosell, G. & De Fabritiis, G. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *J. Chem. Inf. Model.* **58**, 287–296 (2018).

[25] Wang, R., Fang, X., Lu, Y. & Wang, S. The pdbbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *J. Med. Chem.* **47**, 2977–2980 (2004).

[26] Volkov, M. *et al.* On the frustration to predict binding affinities from protein–ligand structures with deep neural networks. *J. Med. Chem.* **65**, 7946–7958 (2022).

[27] Gorantla, R., Kubincova, A., Weiße, A. Y. & Mey, A. S. From proteins to ligands: Decoding deep learning methods for binding affinity prediction. *J. Chem. Inf. Model.* **64**, 2496–2507 (2024).

[28] Landrum, G. A. & Riniker, S. Combining ic50 or k i values from different sources is a source of significant noise. *J. Chem. Inf. Model.* (2024).

[29] Backenköhler, M., Groß, J., Wolf, V. & Volkamer, A. Guided docking as a data generation approach facilitates structure-based machine learning on kinases. *J. Chem. Inf. Model.* **64**, 4009–4020 (2024).

[30] Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).

[31] Ahmad, W., Simon, E., Chithrananda, S., Grand, G. & Ramsundar, B. Chemberta-2: Towards chemical foundation models. *arXiv:2209.01712* (2022).

[32] Nguyen, T. *et al.* Graphdta: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics* **37**, 1140–1147 (2021).

[33] Jiang, M. *et al.* Drug–target affinity prediction using graph neural network and contact maps. *RSC Adv.* **10**, 20701–20712 (2020).

[34] Suzek, B. E. *et al.* Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932 (2015).

[35] Kim, S. *et al.* Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **47**, D1102–D1109 (2019).

[36] Han, Z., Gao, C., Liu, J. & Zhang, S. Q. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv:2403.14608* (2024).

[37] Xu, L., Xie, H., Qin, S.-Z. J., Tao, X. & Wang, F. L. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv:2312.12148* (2023).

[38] Sultan, A., Sieg, J., Mathea, M. & Volkamer, A. Transformers for molecular property prediction: Lessons learned from the past five years. *J. Chem. Inf. Model.* **64**, 6259–6280 (2024).

[39] Dutt, R., Ericsson, L., Sanchez, P., Tsaftaris, S. A. & Hospedales, T. Parameter-efficient fine-tuning for medical image analysis: The missed opportunity. *arXiv:2305.08252* (2023).

[40] Gema, A., Daines, L., Minervini, P. & Alex, B. Parameter-efficient fine-tuning of llama for the clinical domain. *arXiv:2307.03042* (2023).

[41] Gema, A. P., Hong, G., Minervini, P., Daines, L. & Alex, B. Edinburgh clinical nlp at semeval-2024 task 2: Fine-tune your model unless you have access to gpt-4. *arXiv:2404.00484* (2024).

[42] Luo, D., Liu, D., Qu, X., Dong, L. & Wang, B. Enhancing generalizability in protein–ligand binding affinity prediction with multimodal contrastive learning. *J. Chem. Inf. Model.* (2024).

[43] Kaufman, B. *et al.* Coati: Multimodal contrastive pretraining for representing and traversing chemical space. *J. Chem. Inf. Model.* **64**, 1145–1157 (2024).

[44] Singh, R., Sledzieski, S., Bryson, B., Cowen, L. & Berger, B. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proc. Natl. Acad. Sci. U.S.A.* **120**, e2220778120 (2023).

[45] Radford, A. *et al.* *Learning transferable visual models from natural language supervision*, Vol. 139 of *Proceedings of Machine Learning Research*, 8748–8763 (PMLR, 2021).

[46] Gilson, M. K. *et al.* Bindingdb in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res.*

**44**, D1045–D1053 (2016).

[47] Huang, K. *et al.* Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv:2102.09548* (2021).

[48] Shen, W.-f., Tang, H.-w., Li, J.-b., Li, X. & Chen, S. Multimodal data fusion for supervised learning-based identification of usp7 inhibitors: a systematic comparison. *J. Cheminform.* **15**, 1–16 (2023).

[49] Achdout, H., Aimon, A., Bar-David, E. & Morris, G. Covid moonshot: open science discovery of sars-cov-2 main protease inhibitors by combining crowdsourcing, high-throughput experiments, computational simulations, and machine learning. *BioRxiv* (2020).

[50] Boby, M. L. *et al.* Open science discovery of potent noncovalent sars-cov-2 main protease inhibitors. *Science* **382**, eabo7201 (2023).

[51] Trott, O. & Olson, A. J. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **31**, 455–461 (2010).

[52] Ruiz-Carmona, S. *et al.* rdock: A fast, versatile and open source program for docking ligands to proteins and nucleic acids. *PLoS Comput. Biol.* **10**, e1003571 (2014).

[53] Li, J. *et al.* Leak proof pdbbind: A reorganized dataset of protein-ligand complexes for more generalizable binding affinity prediction. *arXiv:2308.09639* (2023).

[54] Bemis, G. W. & Murcko, M. A. The properties of known drugs. 1. molecular frameworks. *J. Med. Chem.* **39**, 2887–2893 (1996).

[55] Bento, A. P. *et al.* An open source chemical structure curation pipeline using rdkit. *J. Cheminform.* **12**, 1–16 (2020).

[56] Deringer, V. L. *et al.* Gaussian process regression for materials and molecules. *Chem. Rev.* **121**, 10073–10141 (2021).

[57] Liu, Z. *et al.* Pdb-wide collection of binding data: current status of the pdbbind database. *Bioinformatics* **31**, 405–412 (2015).

[58] Baumann, H. M. *et al.* Openfe 1.0rc release: Benchmarking results (2024). URL https://doi.org/10.5281/zenodo.13959654. Accessed: October 21, 2024.

[59] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805* (2018).

[60] Liu, Y. *et al.* Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692* (2019).

[61] Wolf, T. *et al.* Huggingface's transformers: State-of-the-art natural language processing. *arXiv:1910.03771* (2019).

[62] Hu, E. J. *et al.* Lora: Low-rank adaptation of large language models. *arXiv:2106.09685* (2021).

[63] Yeh, S.-Y. *et al.* Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. *arXiv:2309.14859* (2023).

[64] He, X., Li, C., Zhang, P., Yang, J. & Wang, X. E. Williams, B., Chen, Y. & Neville, J. (eds) *Parameter-efficient model adaptation for vision transformers.* (eds Williams, B., Chen, Y. & Neville, J.) *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, 817–825 (AAAI Press, 2023). URL https://doi.org/10.1609/aaai.v37i1.25160.

[65] Liu, H. *et al.* Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Adv. Neural Inf. Process. Syst.* **35**, 1950–1965 (2022).

[66] Mangrulkar, S. *et al.* Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft (2022).

[67] Joeres, R., Blumenthal, D. B. & Kalinina, O. V. Datasail: Data splitting against information leakage. *bioRxiv* 2023–11 (2023).

[68] Mendez, D. *et al.* Chembl: towards direct deposition of bioassay data. *Nucleic Acids Res.* **47**, D930–D940 (2019).

[69] ULC, C. C. G. *Molecular Operating Environment (MOE).* Chemical Computing Group, Montreal, QC, Canada (2022). https://www.chemcomp.com.

[70] Eberhardt, J., Santos-Martins, D., Tillack, A. & Forli, S. Autodock vina 1.2.0: New docking methods, expanded force field, and python bindings. *J. Chem. Inf. Model.* **61**, 3891–3898 (2021).

[71] Morris, G. *et al.* Autodock4 and autodocktools4: automated docking with selective receptor flexibility. *J. Comput. Chem.* **30**, 2785–2791 (2009).

# Appendix A Results

### *BindingDB data distribution*

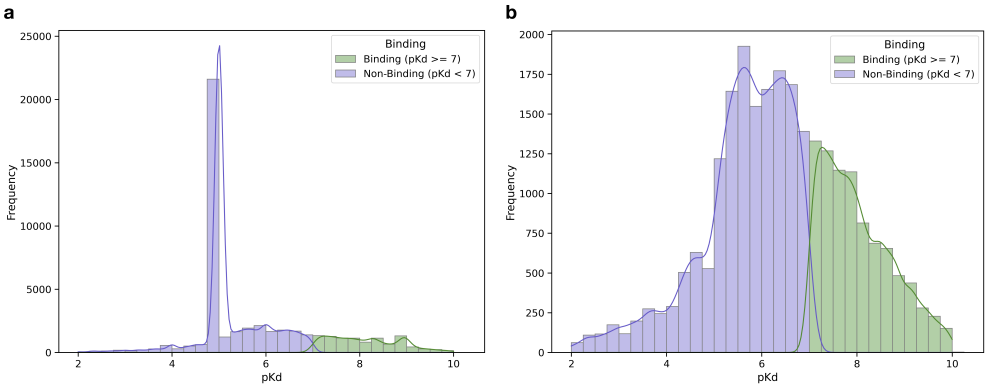


**Fig. A1 BindingDB distribution of pK$_d$ values before and after removing assay limits. (a)** The distribution of binding affinities (pK$_d$) in the original BindingDB dataset shows significant skewness, primarily caused by the dominance of interactions with assay limits, where the most frequent value (pK$_d$ = 4.99) accounts for a large portion of the data. **(b)** After removing the top five most frequent assay limits, the dataset demonstrates a more balanced distribution of binding affinities, providing a better foundation for model training and reducing prediction bias.

### *Parameter-efficient fine-tuning significantly improves BALM's performance*

For both ligand and protein-only fine-tuning, we tested various PEFT methods and ranks. LoHa and LoKr showed improvement in performance across all ranks as compared to the no fine-tuning model, while LoRA showed improvement with rank 8. However, the performance dropped at higher ranks. LoHa seemed to be stable, with rank 16 giving the best improvement of approximately 9.4%, showing a significant improvement over the BALM model without fine-tuning. Other ranks for LoHa also provided notable improvements, achieving around 8.7% (rank 8) and 8.3% (rank 32). IA3, while beneficial, offered a moderate gain of about 5.0%. In protein-only fine-tuning, LoKr seemed to be consistent across all ranks, with the highest improvement of approximately 18.2% at rank 8. Other ranks for LoKr also showed significant improvements, achieving gains of approximately 17.4% (ranks 16 and 32). IA3 and LoHa also demonstrated substantial improvements, with LoHa (rank 8) achieving around 16.8% gain and LoHa (rank 16) showing a gain of 16.1%. In contrast, LoRA showed a drop in performance. Finally, we combined the best-performing fine-tuning methods, LoHa for ligands and LoKr for proteins, and chose rank 16 for both to keep it consistent to create the BALM+PEFT variant. This combination yielded a performance gain of around 23.4% compared to BALM with no fine-tuning.

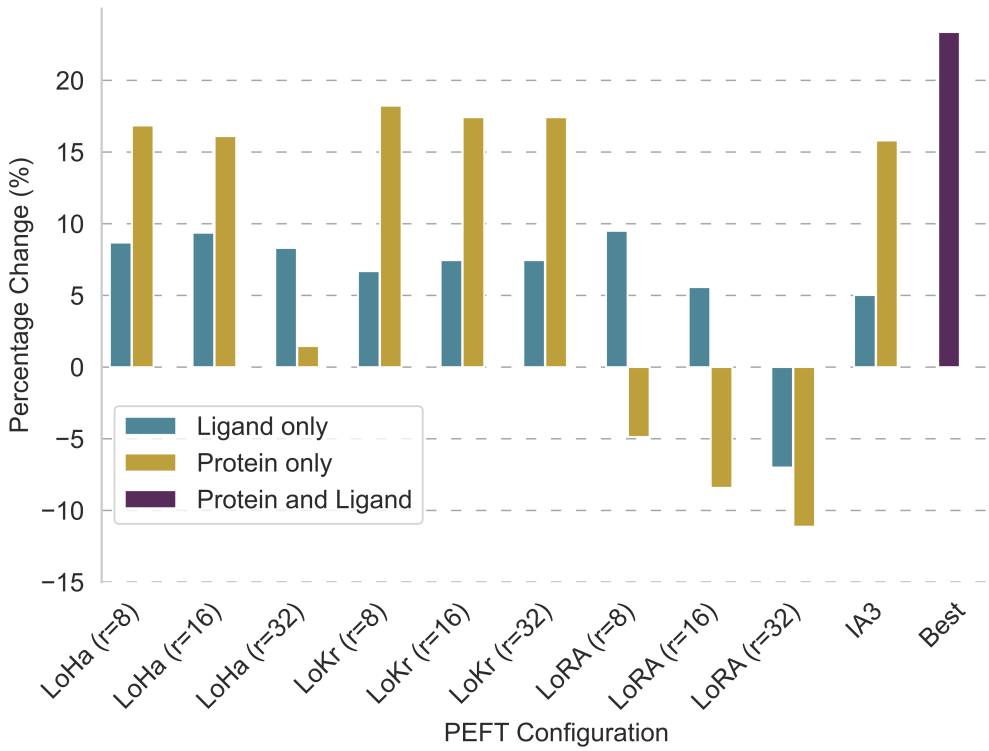


**Fig. A2 Comparative analysis of parameter-efficient fine-tuning (PEFT) methods for protein and ligand language models.** This study evaluates the performance of various PEFT methods, specifically LoRA, LoHa, LoKr, and IA$^3$, in fine-tuning protein and ligand language models within the BALM framework. The performance of these methods is measured using Pearson correlation and compared against BALM without fine-tuning. Initially, we fine-tune protein and ligand language models separately in the BALM framework. Subsequently, the most effective fine-tuning methods (denoted as Best in the plot) for protein (LoKr) and ligand (LoHa) models are applied to assess their combined impact on the performance of BALM+PEFT.

# Appendix B Methods

## PEFT techniques

The strategies for fine-tuning the protein and ligand language models used in the study are discussed below.

- **LoRA** introduces $\Delta W = B \cdot A$, where $B \in \mathbb{R}^{p \times r}$ and $A \in \mathbb{R}^{r \times q}$ are trainable low-rank matrices, with $r \ll \min(p, q)$. This modification is added to the pretrained weight $W_0$, yielding an updated forward pass,

$$h' = W_0 h + b + \gamma(BA)h, \tag{B1}$$

  where $h$ represents the input and $h'$ represents the output after forward pass using the updated weights. $\gamma$ is a scaling factor to control the initialization of B and A

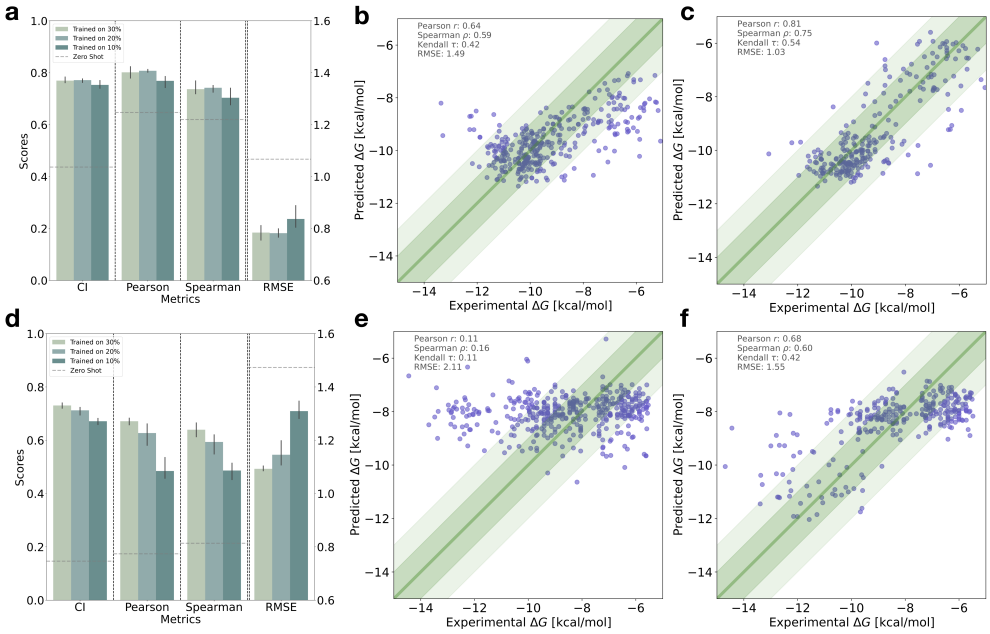


**Fig. A3 Zero-shot and few-shot performance of the BALM+PEFT model on *Mpro* and *USP7* targets. (a, d)** Performance comparison of the pretrained BALM+PEFT model (zero-shot) and few-shot fine-tuning using 10%, 20%, and 30% of experimental data. Testing is performed on the remaining *USP7* (a) and *Mpro* (d) data. The model is fine-tuned by retraining only the projection layer, and the data is split randomly with three different seeds. Error bars indicate the standard deviation across different splits, and performance metrics are reported for Concordance Index (CI), Pearson correlation, Spearman rank correlation, and Root Mean Squared Error (RMSE). **(b, e)** Scatter plots showing zero-shot model predictions for *USP7* (b) and *Mpro* (e) targets. Experimental $\Delta G$ values (kcal/mol) are on the x-axis and predicted $\Delta G$ values are on the y-axis. Only 20% of the test set (selected randomly) is shown for readability. **(c, f)** Scatter plots showing few-shot model performance on 20% of the training data for *USP7* (c) and *Mpro* (f) targets, with experimental $\Delta G$ on the x-axis and predicted $\Delta G$ on the y-axis. Metrics for Pearson $R$, Spearman $\rho$, Kendall $\tau$, and RMSE are displayed in the top-left corner of each scatter plot. Only 20% of the test set (selected randomly) is shown for readability.

for balancing pretrained knowledge and new adaptations, with $\gamma = \alpha/r$ for some defined $\alpha$ and $r$ is the rank of the matrix.

- **LoHa** uses $\Delta W = (B_1A_1) \odot (B_2A_2)$, enhancing the rank of the update matrix and thus the fine-tuning capacity without significantly increasing the parameter count. The forward pass becomes

$$h' = W_0h + b + \gamma[(B_1A_1) \odot (B_2A_2)]h, \tag{B2}$$

  where $\odot$ denotes the Hadamard (element-wise) product.
- **LoKr** employs $\Delta W = C \otimes (BA)$, leveraging the multiplicative rank property of Kronecker products to extend beyond low-rank constraints while maintaining parameter

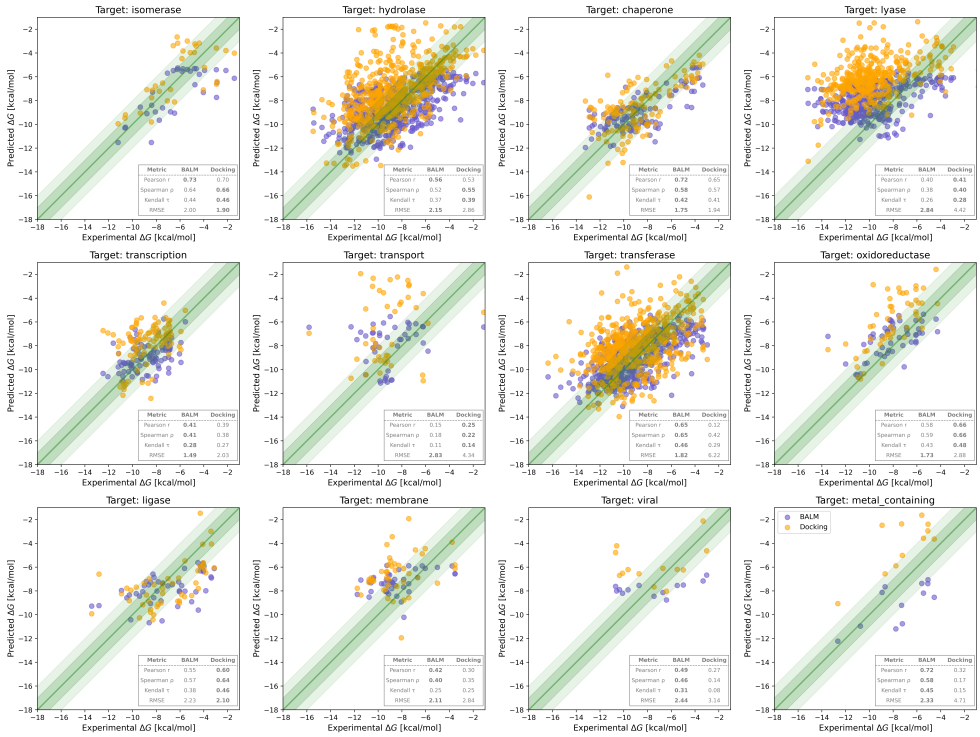


**Fig. A4 Comparing BALM's and Autodock Vina's performance across different target families in the LP-PDBBind dataset.** Scatter plots show the predicted versus experimental binding affinities ($\Delta G$) for various target types in the zero-shot setting using the BALM+PEFT model. Target families include a wide range of protein classes, such as isomerases, hydrolases, and membrane proteins. Pearson correlation ($r$), Kendall $\tau$ Spearman rank correlation ($\rho$), and RMSE are shown for each target type.

efficiency. The forward pass is updated as,

$$h' = W_0 h + b + \gamma(C \otimes (BA))h, \tag{B3}$$

where $\otimes$ denotes the Kronecker product.

- **IA$^3$** method works by rescaling the inner activations of the self-attention in the transformer layers with learned vectors. IA$^3$ introduces three learnable rescaling vectors $\mathbf{l}_k \in \mathbb{R}^{d_k}$, $\mathbf{l}_v \in \mathbb{R}^{d_v}$, and $\mathbf{l}_f \in \mathbb{R}^{d_{\text{ff}}}$, which are used to rescale the key, value, and feed-forward network (FFN) activations in the transformer layers compared to learned low-rank updates to key, query and value weight matrices like in LoRA, LoKr and LoHa. The self-attention mechanism is adjusted as follows

$$\text{Attn}(Q, K, V) = \left(\frac{Q(\mathbf{l}_k \odot K^T)}{\sqrt{d_k}}\right)(\mathbf{l}_v \odot V), \tag{B4}$$

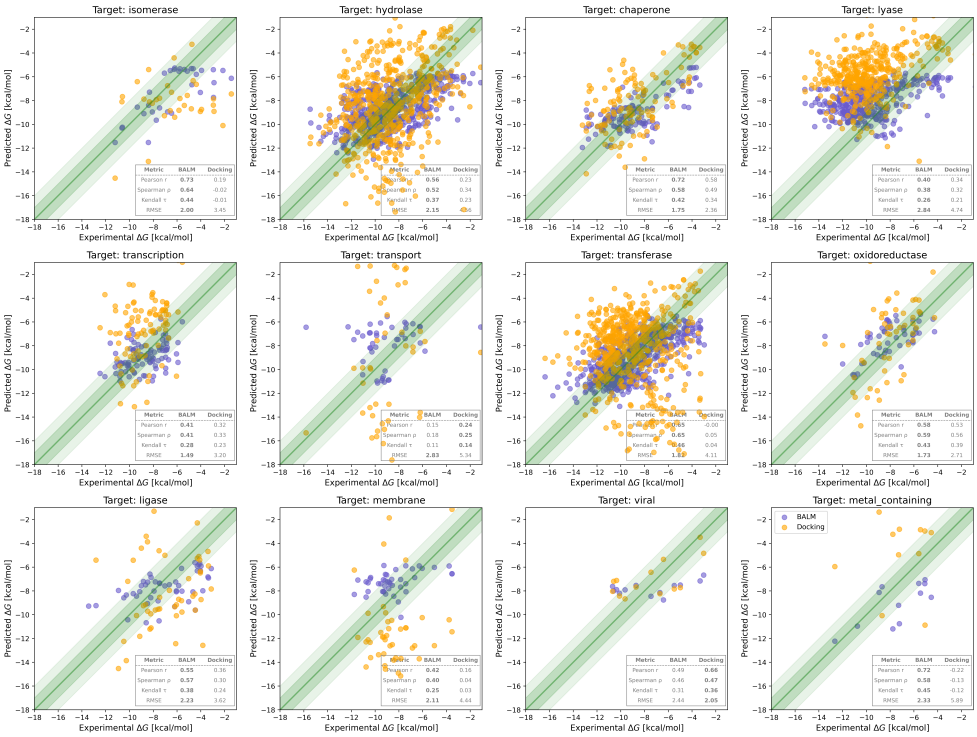


**Fig. A5 Comparing BALM's and rDock performance across different target families in the LP-PDBBind dataset.** Scatter plots show the predicted versus experimental binding affinities ($\Delta G$) for various target types in the zero-shot setting using the BALM+PEFT model. Target families include a wide range of protein classes, such as isomerases, hydrolases, and membrane proteins. Pearson correlation ($r$), Kendall $\tau$ Spearman rank correlation ($\rho$), and RMSE are shown for each target type.

where $Q, K, V$ are the query, key, and value matrices, respectively, and $d_k$ is the dimension of the key vectors. In the FFN, the adaptation is represented as $\text{FFN}(h') = (\mathbf{l}_{\text{ff}} \odot \theta(W_1 h))W_2$, where $\mathbf{l}_{\text{ff}}$ is the learned scaling vector applied to the output of the first layer of the FFN, $W_1$ and $W_2$ are the weight matrices of the FFN, and $\theta$ is the FFN's activation function.

## Docking approach for LP-PDBBind

The LP-PDBBind [53] database provided the native receptor conformation and ligand binding poses extracted from their protein data bank (PDB) entries. When multiple alternative conformations were reported in the crystallographic data, the first alternative conformation was selected. The tautomers and protonation states of the protein-ligand complexes were obtained using the Molecular Operating Environment suite v2022-2 [69] while maintaining the structural waters and cosolvents. Non-standard residues, isotopes, and cosolvents (such as crystallization buffer molecules

like polyethylene glycol, ethanediol, phosphates, etc.) were appropriately removed or replaced with standard elements or residues. Coordination ions within a 6 Å radius of the ligand were retained.

For the evaluation with rDock [52], a cavity was generated for each system using the reference ligand method. Then, the ligands underwent a single-step simplex minimization, and the `SCORE` and `SCORE.INTER` were reported. For the evaluation with AutoDock Vina v1.2.5 [70], the charges of the receptors and ligands were estimated using the Gasteiger model with MGLtools [71] and then re-scored using Vina within the automatically detected cavity.