STORM: A LIGHTWEIGHT STRUCTURE-FREE MODEL FOR BINDING AFFINITY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurately predicting a small molecule's binding affinity to a protein, a critical task in drug discovery, remains challenging. Physics-based approaches like free-energy perturbation (FEP) have very high computational costs and require expert setup. Recent machine learning–based cofolding models have reported accuracy approaching that of FEP—and at least one of them, Boltz-2, has already been widely adopted for affinity prediction—but they remain expensive for large computational screens and even more expensive for retraining on newly collected experimental data. To address these challenges, we introduce **STORM**: **S**imilarity **T**ransfer from **O**bserved **R**eference **M**olecules. STORM is a fast, simple machine learning model that ensembles many small model fusions of diverse protein and molecule representations for binding affinity prediction. Trained on the same data, STORM matches or exceeds the accuracy of Boltz-2 on multiple evaluation datasets, while delivering a speed improvement of over five orders of magnitude for both inference and training. Our results also suggest opportunities to substantially improve affinity prediction accuracy by integrating historical experimental affinity data with structural and biophysical information.

1 INTRODUCTION

Finding molecules that bind tightly to a target protein is a critical early step in drug discovery (Paggi et al., 2024). Despite decades of research, however, accurately predicting binding affinities computationally remains an open challenge. As libraries of drug-like candidate molecules available to researchers grow into trillions of compounds (eMolecules, 2022; Enamine & Chemspace, 2024), having fast and accurate methods for predicting binding affinities has become even more important.

One major branch of research has focused on developing physics-based simulation methods for predicting binding affinities (Wang et al., 2015). Methods like free energy perturbation (FEP) can make predictions that correlate well with experimental measurements when provided a structure of the target protein bound to the small molecule. However, setting up FEP simulations requires careful expert calibration, high-quality structures, and significant computational resources—many hours of GPU time per prediction [CITE?]. Recently, machine learning methods such as Boltz-2 (Passaro et al., 2025) have reported performance approaching FEP on binding affinity prediction, at a far lower computational cost. These methods are cofolding foundation models, or machine learning architectures similar to AlphaFold3 (Abramson et al., 2024; Shen et al., 2026), which was designed for biomolecular structure prediction. Last year, Boltz-2 pioneered the approach of extending these models to also predict binding affinities (Passaro et al., 2025). This work has already had a major impact: Boltz-2 is already widely used for affinity prediction (Fry et al., 2026; Amini et al., 2026), and several other cofolding models have incorporated affinity prediction using the same approach (Shenoy et al., 2026; SandboxAQ, 2026).

Cofolding models are trained on large amounts of experimental biomolecular structure data and even larger amounts of biomolecular sequence data in the form of multiple sequence alignments (MSAs). To add affinity prediction capabilities, they are also trained on datasets of publicly reported experimental binding affinities for compounds at many different protein targets. These methods are substantially faster than FEP, but they still require extensive computational resources to complete large virtual compound screens for drug discovery, limiting the number of compounds that can be screened. Because these methods need to be trained on very large datasets with many different types

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

of data, refitting them or fine-tuning them on small amounts of additional experimental data as it becomes available can also be difficult.

Most of the data used to train cofolding models–in particular, experimental structures and MSAs–is critical for their structure-prediction capabilities. It is unclear, however, to what extent this data contributes to their affinity-prediction capabilities. Might a much simpler model that ignores three-dimensional structure, and that is trained only on experimental binding affinities, be able to achieve similar performance? Decades of research have shown that models trained only on affinity data at an individual target can achieve reasonable binding affinity predictions at that target by taking advantage of similar molecules being more likely to have similar binding affinities (Cherkasov et al., 2014; Zhang et al., 2024). Recent work has shown that these chemical trends in binding affinity be transferred between protein targets that share low sequence identity, suggesting that learning to transfer historical affinity relationships between protein targets could provide an alternative route to strong binding affinity prediction (Mattsson & Walters, 2026). Focusing on this signal would also remove much of the training and modeling complexity associated with MSA and structure data.

Inspired by these observations, we introduce **STORM**, or **S**imilarity **T**ransfer from **O**bserved **R**eference **M**olecules. STORM is a structure-free binding affinity predictor that focuses on learning relationships between small molecule affinities that can transfer between protein targets. We leverage the fact that different simple representations for small molecules, such as various chemical descriptors or different molecular fingerprint algorithms, can capture different activity relationships between molecules and provide complementary signals. We also apply this technique to build simple protein target representations, and then independently train and ensemble many small models on fusions of these target and molecule representations. We make **four main contributions**:

- ***(1)*** We introduce a structure-free method for combining simple, complementary protein and small-molecule representations that can be trained on historical affinity measurements.
- ***(2)*** We show that when trained on the same data sources, STORM is as accurate or more accurate than Boltz-2 and substantially outperforms previous structure-free predictors across multiple affinity benchmarks, including OpenFE and a ChEMBL 35 time split.
- ***(3)*** We show that STORM has dramatically lower training and inference costs compared to co-folding methods that achieve similar performance, making it much more practical to perform very large virtual screens and to update the model as new experimental data becomes available.
- ***(4)*** We show that, given a handful of experimental affinity measurements for a target not in the training set—a common scenario in drug discovery—we can substantially improve STORM's prediction accuracy by finetuning the model on a laptop in minutes.

In addition to these practical advances, our work has important longer-term implications, suggesting opportunities to substantially improve the accuracy of machine learning–based affinity prediction by combining the strengths of cofolding models with physics-based affinity predictors.

2 Related Work

Cofolding affinity predictors. Many machine learning methods have been developed that predict the affinity of a small molecule bound to a protein (Ballester & Mitchell, 2010; Stepniewska-Dziubinska et al., 2018; Jiménez et al., 2018). Other methods, such as those in Lu et al. (2022), Lu et al. (2024), and Morehead & Cheng (2025), predict both the affinity and the pose of the docked ligand in the pocket of a given input protein structure. However, Boltz-2 was the first model to integrate a base cofolding model with a small molecule binding affinity prediction module, leading to significantly improved accuracy in affinity prediction compared to previous machine learning methods (Passaro et al., 2025). This was an important advance, particularly given that Boltz-2 is dramatically faster than leading physics-based methods. Cofolding models use an AlphaFold 3–style trunk to integrate protein and small-molecule sequence information, MSA information, and, optionally, existing template structures to predict biomolecular complex structures. Boltz-2 pioneered the use of the latent representation of the Boltz-2 trunk in combination with their predicted protein–small molecule complex to train an affinity predictor module on historical binding affinity data. Subsequently, many others have adopted this core design, including the AQAffinity model on an

OpenFold3 base (SandboxAQ, 2026; The OpenFold3 Team, 2025), and IsoDDE (Isomorphic Labs Team, 2026), a proprietary method descended from AlphaFold 3. TerraBind also follows a similar architecture, but uses a course grained representation in the cofolding base, and makes affinity predictions using the trunk's latent representation alone (Rossi et al., 2026). All of these methods still require a large model trunk trained on large amounts of structural data and MSA information, in addition to historical binding affinity measurements. STORM instead uses simple representations and historical binding affinity measurements alone.

Structure-free affinity predictors. Substantial prior work has also been done on developing structure-free affinity predictors. Many early methods, including chemogenomic models, focused on learning from datasets of binding affinity values across many targets using matrix factorization or graph learning based approaches (Pahikkala et al., 2015; He et al., 2017). Methods such as DeepDTA and AttentionDTA use initial simple amino acid sequence encodings for proteins and SMILES string representations for molecules that are combined using attention, and then trained end-to-end for affinity prediction(Öztürk et al., 2018; Zhao et al., 2022). GraphDTA is similar, but it represents molecules as graphs based on their chemical connectivity before fusing to the learned target representation(Nguyen et al., 2021). Other approaches instead use large, pretrained encoders to generate embedding representations for proteins and molecules. BALM encodes protein sequences with ESM-2, a protein language model and small molecules with ChemBERTa-2, a small molecule language model, projects their representations together, and then trains end-to-end (Gorantla et al., 2025; Ahmad et al., 2022); Structure-free DTA is similar but uses the original ChEMBERTa (Bidgoli et al., 2026). Finally, PSICHIC uses a hybrid representation in which small molecules are represented as graphs based on their chemical connectivity, but the protein's ESM-2 embeddings and predicted residue contacts are used to build a protein graph, which then becomes fused with the molecule representation (Koh et al., 2024). Despite the many differences between these methods, each of them individually uses a single representation for the target protein and a single representation for the small molecule when making predictions, fed through a single model. STORM instead relies on making an ensemble of many different small models that use many different combinations of simple input representations.

3 METHODS

STORM is an ensemble composed of 15 different independently trained small models. Each model shares a core multi-layer perceptron (MLP) architecture, with a variably sized first layer depending on the input representation, and is trained with the same loss. Each model receives a different input representation, consisting of simple target protein representations combined with a simple molecule representation. To make a prediction for a single protein and small molecule pair, 5 different protein target representations are calculated, and 9 unique ligand representations are calculated, which are then processed by their respective models. Not all pairs of target and molecular representations are used to train models, and some models are weighted with two votes instead of one, based on an analysis of each model performance on held-out validation data. We describe each of these components in further detail below.

3.1 TASK

Given a target protein p, a small molecule m, and an experimental endpoint e, STORM predicts a continuous experimental activity value $\hat{y}(p, m, e)$. We train on measured concentrations on a $\mathrm{pActivity}$ scale. This means larger values correspond to lower measured concentrations and tighter binding affinities. The endpoint identifies the type of measurement made in an experimental assay. We encode K_i, K_d, IC_{50}, and EC_{50} separately; other endpoint types receive an all-zero endpoint vector. Each component model receives an endpoint encoding and a molecule representation. Most components also receive a target representation; target-blind components receive no target features.

3.2 INPUT REPRESENTATIONS

Across the different models in the ensemble, STORM uses 5 protein target representations and 9 unique molecule representations. None of these representations are derived from 3D structural data,

Table 1: Base target features used by STORM.

Feature	Description	Input size
Target ID	Code derived from a hash of the target identifier.	128
GO	Gene Ontology annotations (The Gene Ontology Consortium, 2026; Ashburner et al., 2000).	2,048
InterPro	Protein-family and domain annotations (Blum et al., 2025).	2,048
Species	Human, nonhuman, or unknown indicator.	3

Table 2: STORM Molecule representations.

Representation	Description	Input size
Morgan count	Circular atom neighborhoods of radius 2 (Rogers & Hahn, 2010).	2,048
RDKit path count	Atom and bond paths of length 1–7 (RDKit contributors).	2,048
Atom pair count	Atom-type pairs and their graph distances (Carhart et al., 1985).	2,048
Topological torsion count	Sequences of four connected atoms (Nilakantan et al., 1987).	2,048
Avalon binary	Avalon molecular fingerprint (RDKit contributors).	2,048
2D pharmacophore count	Pharmacophore feature pairs, triplets, and graph distances (scikit-fingerprints contributors).	2,048
Klekota–Roth count	Predefined substructures (Klekota & Roth, 2008).	4,860
PubChem count	PubChem chemical features (scikit-fingerprints contributors).	757
RDKit/Mordred descriptors	Calculated 2D physicochemical and topological descriptors (RDKit contributors; Moriwaki et al., 2018).	1,212

such as protein structures, small molecule structures, or structures of protein–small molecule complexes. Additionally, none of these representations are derived from pretrained protein language models or molecular foundation models. Each target representation is built from the features described in Table 1, with some component models receiving multiple target representations while others only receive a single individual representation. The molecule representations are listed in Table 2. The InterPro (Blum et al., 2025), Gene Ontology (The Gene Ontology Consortium, 2026), and UniProtKB (The UniProt Consortium, 2025) databases were used to obtain all protein target feature information. The RDKit (RDKit contributors) and scikit-fingerprints (Adamczyk & Ludynia, 2024) packages were used to calculate molecular representations. See the Appendix for more information.

3.3 Model Architecture

Every component model in STORM has a simple, early fusion architecture. The target representations, when present, are each projected to their referenced size and concatenated with the molecule representation in a shared MLP prediction head. The endpoint is encoded as a four-dimensional vector and projected into the first hidden layer. Table 3 summarizes the common prediction path. Here, d_m is the size of the molecule representation and d_p is the size of the encoded target representation; $d_p = 0$ for target-blind components. In total, across all 15 models, STORM has approximately 20 million parameters, or an average of 1.2 million parameters per component. Additional details are available in the Appendix.

3.4 Loss Function

We train each component using a combination of pointwise activity predictions and within-assay activity differences. Let y_i and $\hat{y}_i$ denote the measured and predicted pActivity for example i. We use the Huber loss

$$H_\delta(r) = \begin{cases} \frac{1}{2}r^2, & |r| \le \delta, \\ \delta\left(|r| - \frac{1}{2}\delta\right), & |r| > \delta, \end{cases} \tag{1}$$

with $\delta = 1$. For a sampled pair (i, j) from the same assay a, we normalize both the measured and predicted activity differences by $s_a = \max(\sigma_a, 0.5)$, where σ_a is the standard deviation of measured activities in that training assay. The training objective for a batch is

Table 3: Common architecture of each STORM component model.

Layer	Input size	Output size	Operation
Input fusion	$d_m + d_p$	384	Linear projection; add projected endpoint encoding
Hidden layer 1	384	384	Layer normalization, SiLU, dropout (0.20)
Hidden layer 2	384	192	Linear projection, layer normalization, SiLU, dropout (0.10)
Prediction	192	1	Linear projection to pActivity

$$\mathcal{L} = 0.25\,\frac{1}{|B|}\sum_{i\in B} H_1(\hat{y}_i - y_i) + \frac{1}{|P_B|}\sum_{(i,j)\in P_B} H_1\left(\frac{(\hat{y}_i - \hat{y}_j) - (y_i - y_j)}{s_{a(i)}}\right), \tag{2}$$

where B is the batch and P_B is its set of sampled within-assay pairs. This helps us avoid high errors for small differences between values in assays, while still using a relative scale. We sample pairs only from assays with at least five training compounds. For an assay fragment containing n examples in a batch, we sample up to $\min\{\binom{n}{2}, \mathrm{round}(8\sqrt{n}), 256\}$ pairs. The pointwise term learns absolute activity values, while the pairwise term directly trains the model to predict activity differences between molecules measured in the same assay.

3.5 Graph Smoothing

After each STORM component makes its prediction, and after all predictions from the ensemble have been weighted according to the vote share, we then smooth predictions for molecules in the same evaluation assay. We found that applying this smoothing helped reduce error from prediction noise on the validation data.

This step uses molecular similarity and model predictions; it does not use the assay's measured activities. We calculate Tanimoto similarity s_{ij} between the binary supports of the molecules' Klekota-Roth count fingerprints (Klekota & Roth, 2008). For molecule i, let N_i contain up to five other molecules from the same assay with the highest positive similarity. Starting from the ensemble prediction $\hat{y}_i^{(0)}$, we apply

$$\hat{y}_i^{(t+1)} = (1-\lambda)\hat{y}_i^{(t)} + \lambda\frac{\sum_{j\in N_i} s_{ij}\hat{y}_j^{(t)}}{\sum_{j\in N_i} s_{ij}}, \qquad \lambda = 0.10, \tag{3}$$

for three passes, $t = 0, 1, 2$. If N_i is empty, the prediction for molecule i remains unchanged. The final prediction is $\hat{y}_i^{(3)}$.

3.6 Target-Specific Finetuning

STORM can be easily finetuned on new experimental data, for new or existing targets. We begin with the base pretrained component models and finetune each independently on the new support affinities. We update the full model using a pointwise Huber loss and penalty for movement away from the weights' starting point (Li et al., 2018). See the Appendix for more details.

4 Experiments

4.1 Training Data

We train STORM and all structure-free comparison methods on binding affinity measurements curated from the ChEMBL 34 (ChEMBL, 2024a) relase and the April 1, 2024 snapshot of BindingDB (BindingDB, 2024; Liu et al., 2025). We then apply a data cleaning and standardization protocol adapted from the Boltz-2 model (Passaro et al., 2025) to remove low quality data, filter for specific experimentally measured endpoints, and standardize small molecules, protein targets, and affinity values. Full details are available in the Appendix.

We also filter our data to remove all overlap with pairs of protein target and small molecule measurements that appear in any of our evaluation sets across any of our experiments. Matching the setup of Passaro et al. (2025), we remove all affinity measurements for any protein targets that share ≥ 0.9 sequence identity with any protein targets in the OpenFE Benchmark set–which includes the FEP+4 subset–using MMseqs2 (Steinegger & Söding, 2017).

In total, our training set is composed of 6,775 unique protein IDs, 934,000 unique molecules, 175,000 chemical series, and 1.7 million unique experimental binding affinity measurements.

We note that while our data sources were chosen to match Boltz-2 and enable a fair comparison, they are a *subset* of the training data sources used to train Boltz-2's affinity module. In addition to these sources, Boltz-2 also uses data from PubChem small assays (Kim et al., 2023), PubChem high-throughput screens (Kim et al., 2023), CeMM fragment screens (Offensperger et al., 2024), MIDAS metabolite data (Hicks et al., 2023), and synthetic decoys (Passaro et al., 2025). These sources collectively represent millions of additional training examples and experimental data points, though many are binary values. Boltz-2's exact affinity training records have not been released, so we cannot complete an exact reproduction of the models' training data or confirm exact data overlap.

Importantly, we note that many of the choices that went into Boltz-2 data curation were designed specifically for the Boltz-2 model architecture, which jointly optimizes a continuous affinity value head and a binary binding likelihood head on a shared affinity module. Boltz-2 also filtered data based on the predicted quality of its generated protein-small molecule complex. These data curation rules do not transfer to structure-free binding affinity prediction models that output continuous values. For these reasons, we train STORM and all structure-free binding affinity prediction methods on only exact binding affinity measurements from ChEMBL 34 and BindingDB without matching all of the Boltz-2 filtering steps.

4.2 Comparisons

4.2.1 Structure-based methods

In Section 2, we referenced multiple cofolding models that have been trained for affinity prediction. However, many of them are ill-suited for benchmarking and comparison.

Notably, IsoDDE reports evaluations on multiple benchmarks and has the highest reported performance on several (Isomorphic Labs Team, 2026). However, although its technical report specifies a training date cutoff, the actual composition of its training data is not disclosed. Consequently, IsoDDE's results cannot be compared with other methods: differences in performance may simply reflect differences in training data.

While TerraBind has disclosed more information about its model architecture and training data (Rossi et al., 2026), it is also a proprietary, closed-source model with limited benchmark reports, making comparisons to our full suite of benchmarks impossible. AQAffinity does share code, weights, and details on training data (SandboxAQ, 2026). However, they train their model on later versions of ChEMBL and BindingDB, which overlap with our evaluation sets. Evaluating AQAffinity alongside Boltz-2 would require complete retraining, which is computationally infeasible.

Because of these challenges, we use Boltz-2 as a representative model for structure-based methods. As an open-source model with weights freely available, Boltz-2 is also an important practical comparison, as the most widely deployed cofolding model with affinity prediction today. It has been benchmarked on all of our evaluations, and pioneered the affinity prediction approach used by all subsequent models today.

4.2.2 Structure-free methods

As referenced in Section 2, many different structure-free methods exist. For our comparison, we chose published models with the best reported performance. We compare STORM against **BALM**, **Structure-free DTA**, **Attention DTA**, and **PSICHIC**. As an additional baseline, we also compare to the molecular fingerprint and descriptor method in Mattsson & Walters (2026), referred to as **Mattsson-Walters**. This method uses a molecular fingerprint, descriptors, and a one-hot protein representation, which are fed into an MLP to make binding affinity predictions. We retrain all

structure-free methods on our matched training data corpus and evaluate on the benchmarks described below.

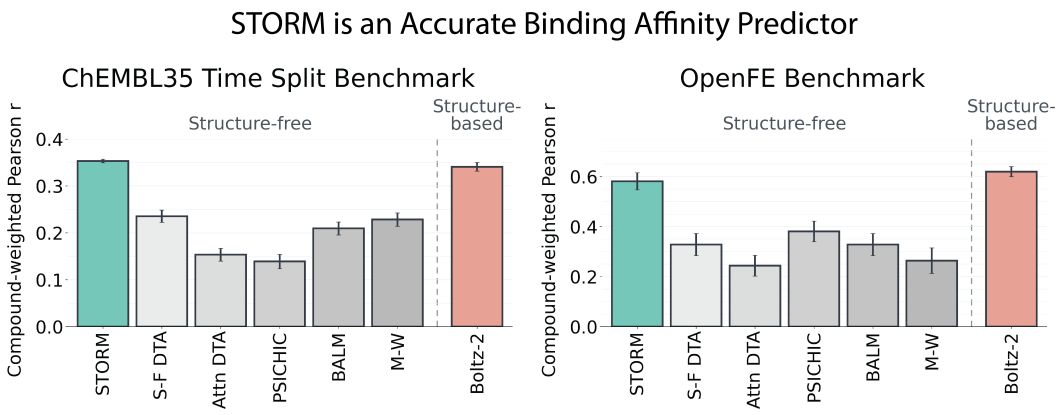

Figure 1: **STORM is competitive with Boltz-2 on ChEMBL35 Time Split and OpenFE binding affinity benchmarks and substantially exceeds accuracy of other structure-free methods.** We report compound-weighted Pearson correlation r. Abbreviations: S-F DTA: Structure-free DTA; Attn DTA: Attention DTA; M-W: Mattsson-Walters. Error bars denote bootstrapped standard errors. We estimate Boltz-2 error bars from the Boltz-2 and IsoDDE papers; see Appendix for details.

4.3 STORM Matches or Exceeds Boltz-2 on Four Binding Affinity Evaluation Sets

We evaluate on four held-out datasets that serve as de facto benchmarks in Boltz-2 and IsoDDE.

CASP16 (Gilson et al., 2026) CASP16 is a blind prediction challenge comprising 140 affinity measurements across two targets (17 measurements against Human chymase and 123 against Human autotaxin). Similar to the Boltz-2 evaluation, we treat this challenge as a retrospective evaluation.

OpenFE (Ross et al., 2023; Gowers et al., 2023) 873 experimental measurements across 39 targets drawn from the OpenFE Industry Benchmarks. The 39 targets include kinases and other enzymes, but excludes membrane proteins such as GPCRs.

FEP+4 (Ross et al., 2023; Hahn et al., 2022) A four target subset of the FEP+ benchmark. There are a total of 87 measurements across four kinase targets: CDK2, TYK2, JNK1, and p38α.

ChEMBL 35 (ChEMBL, 2024b; Isomorphic Labs Team, 2026) A new evaluation set introduced in the IsoDDE technical report. This evaluation consists of new assays in the ChEMBL35 release split across different classes of target. In Isomorphic Labs Team (2026), the authors show a chart for every class, comparing IsoDDE and Boltz-2. We aggregate the per-class data into one overall number; see Appendix for details.

OpenFE and FEP+4 are zero-shot evaluations of binding affinity predictors, as no targets appear in the training dataset. Both CASP16 targets and some ChEMBL35 targets appear in the training data. We report compound-weighted Pearson r correlation on all benchmarks, mirroring the Boltz-2 paper.

Results We benchmark STORM against Boltz-2 and the structure-free methods, measuring compound-weighted Pearson r correlation on ChEMBL35 Time Split, OpenFE, CASP16, and FEP+4 benchmarks (Figure 1 and Table 4). We emphasize ChEMBL35 Time Split and OpenFE Benchmarks in Figure 1 as they have a diverse set of 449 and 39 protein targets, respectively, as opposed to CASP16 and FEP+4, which only cover 2 and 4 protein targets, respectively. Correlations are calculated within assays and weighted by evaluated compound counts. For Boltz-2, we used the CASP16, FEP+4, and OpenFE numbers reported in the original manuscript. For the

Model	CASP16	FEP+4	OpenFE	ChEMBL35 Time Split
STORM	**0.667 ± 0.049**	**0.694 ± 0.031**	0.581 ± 0.034	**0.353 ± 0.004**
Boltz-2	0.650 ± 0.020	0.660 ± 0.030	**0.620 ± 0.020**	0.341 ± 0.009
Structure-free DTA	0.380 ± 0.103	0.441 ± 0.058	0.328 ± 0.044	0.235 ± 0.013
AttentionDTA	0.580 ± 0.124	0.062 ± 0.218	0.244 ± 0.041	0.154 ± 0.014
PSICHIC	0.565 ± 0.141	0.518 ± 0.071	0.381 ± 0.041	0.139 ± 0.015
BALM	0.387 ± 0.013	0.481 ± 0.081	0.328 ± 0.044	0.209 ± 0.014
Mattsson-Walters	0.252 ± 0.063	0.638 ± 0.101	0.264 ± 0.051	0.228 ± 0.014

Table 4: Model compound-weighted Pearson r across datasets. Best values are bolded; second-best values are underlined. Error bars denote standard error.

Boltz-2 ChEMBL35 Benchmark accuracy, we use the numbers reported in the IsoDDE technical report. Appendix Table 7 shows accuracies per model, stratified by protein class. Across all four benchmarks, STORM significantly outperforms all structure-free methods. On CASP16, FEP+4, and ChEMBL35, STORM also outperforms Boltz-2, despite using no 3D structure information or MSAs for training.

4.4 STORM IS EXTREMELY FAST FOR RUNNING INFERENCE

Setup To study the Pareto front of binding affinity models, we compare models' performance on ChEMBL35 Time Split versus their inference cost (Figure 2, left panel). We obtained a single NVIDIA L4 GPU, a cheap, energy-efficient GPU with 24 GB VRAM, readily available on cloud platforms. We swept batch sizes to identify the largest feasible batch and then benchmarked that size and the largest feasible power-of-two batch, selecting the batch size that yielded faster throughput. For each configuration, we repeated inference runs and computed elapsed time divided by the number of ligands processed. We report the lower median seconds per ligand. For Boltz-2, we did not use MSAs during inference.

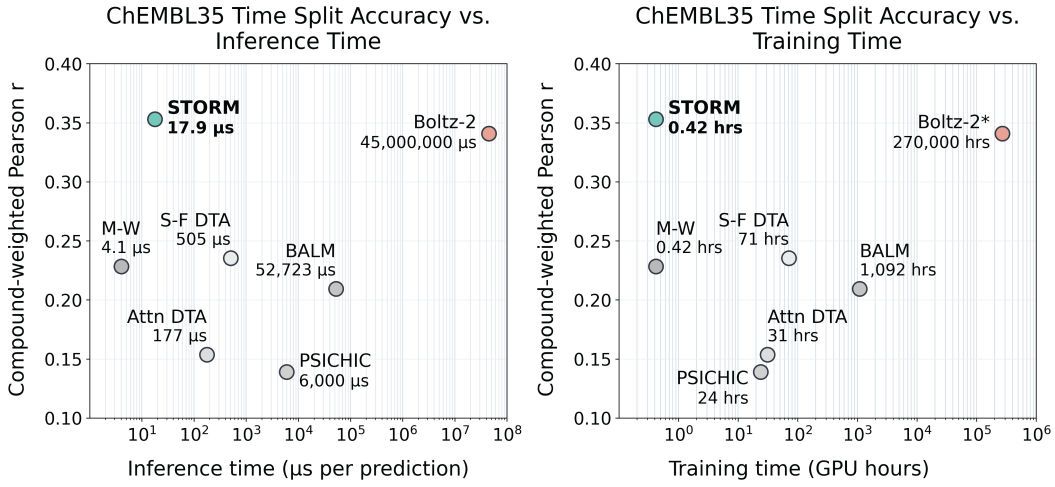

Figure 2: **STORM is fast while maintaining accurate binding affinity predictions.** We plot both inference cost-accuracy tradeoffs (left panel) and pretraining cost-accuracy tradeoffs (right panel). On the y axis, we show the ChEMBL35 compound weighted Pearson r correlation. We observe that STORM provides the optimal tradeoff for accuracy and speed (top left corners in both plots). STORM achieves similar accuracy to Boltz-2 while being over five orders of magnitude faster. Abbreviations are same as Figure 1.

Results STORM (Figure 2, left panel, top left) provides the fastest accuracy-to-inference time ratio. STORM runs at 17.9 microseconds per prediction, second only to Mattsson-Walters. All

other structure-free methods are slower and have lower accuracy, and Boltz-2 is the slowest method due to the cofolding component, running at about 45 seconds per ligand.

4.5 STORM IS EXTREMELY FAST TO TRAIN

Setup We compare models' performance on ChEMBL35 versus their training cost (Figure 2, right panel). We sweep batch sizes on an L4 GPU, in the same manner as Section 4.4. We then retrain all the structure-free methods and record the fastest training time. In Figure 2 we estimate 270,000 L4-equivalent GPU hours for retraining Boltz-2, with details in the Appendix.

Results STORM (Figure 2, right panel, top left) provides the fastest accuracy-to-training time ratio, training in under half an hour. All other structure-free methods train as fast as or slower than STORM, and Boltz-2 is the slowest method.

4.6 STORM IMPROVES WITH SMALL AMOUNTS OF DATA

Setup We also study STORM's ability to adapt to new data via finetuning. In practical drug discovery campaigns, it is common to collect limited amounts of experimental data against a protein target of interest. Because additional experimental measurements are expensive to collect, a model that learns efficiently from small amounts of data is practically highly valuable. We select three protein targets, **LpxH**, **A0A0B5JFF1**, and **ALPK1**. None of these targets or their corresponding small molecules were in the STORM training data. We split the compounds into train and test and finetune STORM. We compare the finetuned STORM against the base model STORM. Additional details are in the Appendix.

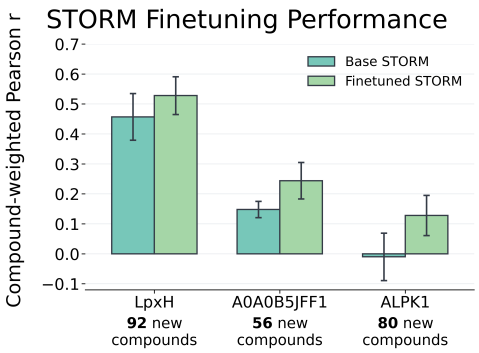

Figure 3: STORM finetuning results on three new targets. Error bars denote standard deviation.

Results Figure 3 shows compound-weighted Pearson r of the base model versus the finetuned model. In all three cases, the finetuned STORM model benefits from the additional data. We note that even with very small finetuning sets, STORM is able to learn patterns in the data and improve performance. For ALPK1, where the base STORM has effectively zero Pearson r, the 80 added compounds provide a large boost in performance.

5 CONCLUSION

We present STORM as a practical choice for machine learning–based binding affinity prediction. STORM achieves performance exceeding or comparable to the state-of-the-art machine learning methods at a fraction of the computational cost for training and inference. STORM achieves this without using embeddings generated by pretrained protein language models or molecular foundation models, and with a simple model architecture that is easy for researchers to adapt or modify. STORM's ability to be retrained or easily finetuned on new experimental data–a critical use case in drug discovery–with ordinary consumer hardware provides a compelling advantage over other methods and makes it highly accessible for researchers everywhere.

STORM also highlights an important gap between current machine learning methods and traditional physics-based simulation methods such as FEP. Unlike the machine learning methods discussed in this paper, physics-based methods are not trained on a large affinity dataset; they rely entirely on the provided structure of the protein target and biophysical principles to produce accurate predictions of binding affinity. On the contrary, STORM receives no signal from biomolecular structure or biophysical principles. Our results suggest opportunities for substantially improving cofolding methods: structural and biophysical information could complement historical affinity transfer, particularly in cases where reference information is sparse or similar molecules have sharply differing affinity measurements.

6 Reproducibility Statement

We describe details on training and model architecture in the Methods section of this paper. We provide additional details and ablation studies in the Appendix. We provide the datasets, source code, and weights needed to reproduce our results in a zip file uploaded to the Supplementary Information in the OpenReview submission.

7 AI Usage Disclosure

In this work, we used generative AI tools (namely OpenAI's GPT and Anthropic's Claude models) for assistance in coding, formatting, literature review, and writing clarity. We have not used generative AI tools to propose or refine hypotheses, clean and reformat dataset, and the following are not applicable: generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, design or provide feedback on research methodology or experiments, implement methods, assist with translation, support qualitative and thematic data analysis, interpret results. We have reviewed all such uses of AI. Generative AI was not used in the core development and ideation of this work, and we take responsibility for the contents in this manuscript.

8 Ethics Statement

We adhere to the ICLR 2027 Code of Ethics. To our best knowledge, there is minimal risk in our work. The data and models we use are publicly available, and we do not anticipate any potential misuse to warrant ethical concerns.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Jakub Adamczyk and Piotr Ludynia. Scikit-fingerprints: Easy and efficient computation of molecular fingerprints in Python. *SoftwareX*, 28:101944, 2024. doi: 10.1016/j.softx.2024.101944.

Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*, 2022.

Samad Amini, Simone Sciabola, and Ye Wang. Affinity fine-tuning of Boltz-2: An open framework for protein-ligand potency prediction in drug discovery. *bioRxiv*, 2026. doi: 10.64898/2026.05.26.727958. Preprint.

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000. doi: 10.1038/75556.

Pedro J. Ballester and John B. O. Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010. doi: 10.1093/bioinformatics/btq112.

Amir Hallaji Bidgoli, Morteza Mahdavi, and Hamed Malek. Structure-free drug–target affinity prediction using protein and molecule language models. *Journal of Cheminformatics*, 18(1):21, 2026.

BindingDB. BindingDB: April 2024 snapshot. Data set, April 2024. URL `https://www.bindingdb.org/`. April 2024 database snapshot.

Matthias Blum, Antonina Andreeva, Laise Cavalcanti Florentino, Sara Rocio Chuguransky, Tiago Grego, Emma Hobbs, Beatriz Lazaro Pinto, Ailsa Orr, Typhaine Paysan-Lafosse, Irina Ponamareva, Gustavo A. Salazar, Nicola Bordin, Peer Bork, Alan Bridge, Lucy Colwell, Julian Gough, Daniel H. Haft, Ivica Letunic, Felipe Llinares-López, Aron Marchler-Bauer, Laetitia Meng-Papaxanthos, Huaiyu Mi, Darren A. Natale, Christine A. Orengo, Arun P. Pandurangan, Damiano Piovesan, Catherine Rivoire, Christian J. A. Sigrist, Narmada Thanki, Françoise Thibaud-Nissen, Paul D. Thomas, Silvio C. E. Tosatto, Cathy H. Wu, and Alex Bateman. InterPro: the protein sequence classification resource in 2025. *Nucleic Acids Research*, 53(D1):D444–D456, 2025. doi: 10.1093/nar/gkae1082.

R. E. Carhart, D. H. Smith, and R. Venkataraghavan. Atom pairs as molecular features in structure-activity studies: definition and applications. *Journal of Chemical Information and Computer Sciences*, 25(2):64–73, 1985. doi: 10.1021/ci00046a002.

ChEMBL. ChEMBL database, release 34. Data set, March 2024a. URL `https://doi.org/10.6019/CHEMBL.database.34`.

ChEMBL. ChEMBL database release 35, December 2024b. URL `https://doi.org/10.6019/CHEMBL.database.35`.

Artem Cherkasov, Eugene N. Muratov, Denis Fourches, Alexandre Varnek, Igor I. Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C. Martin, Roberto Todeschini, Viviana Consonni, Victor E. Kuz'min, Richard Cramer, Romualdo Benigni, Chihae Yang, James Rathman, Lothar Terfloth, Johann Gasteiger, Ann Richard, and Alexander Tropsha. QSAR modeling: Where have you been? where are you going to? *Journal of Medicinal Chemistry*, 57(12):4977–5010, 2014. doi: 10.1021/jm4004285.

eMolecules. Launched: The world's largest and most actionable virtual chemical space, October 2022. URL `https://www.emolecules.com/news-announcement-explore`.

Enamine and Chemspace. Chemspace boosts Enamine's drug discovery platform, October 2024. URL https://enamine.net/public/press-release/Chemspace-boosts-Enamines-drug-discovery-platform-PR-October-9th-2024.pdf.

Benjamin Fry, Kaia Slaw, and Nicholas F. Polizzi. Zero-shot design of drug-binding proteins via neural iterative selection–expansion. *Nature*, 656:237–249, 2026. doi: 10.1038/s41586-026-10670-w.

Michael K. Gilson, Jerome Eberhardt, Peter Škrinjar, Janani Durairaj, Xavier Robin, and Andriy Kryshtafovych. Assessment of pharmaceutical protein–ligand pose and affinity predictions in CASP16. *Proteins: Structure, Function, and Bioinformatics*, 94(1):249–266, 2026. doi: 10.1002/prot.70061. URL https://doi.org/10.1002/prot.70061.

Rohan Gorantla, Aryo Pradipta Gema, Ian Xi Yang, Álvaro Serrano-Morrás, Benjamin Suutari, Jordi Juárez-Jiménez, and Antonia SJS Mey. Learning binding affinities via fine-tuning of protein and ligand language models. *Journal of Chemical Information and Modeling*, 65(22):12279–12291, 2025.

Richard J. Gowers, Irfan Alibay, David W. H. Swenson, Michael M. Henry, Benjamin Ries, Hannah M. Baumann, and James R. B. Eastwood. The open free energy library, 2023. URL https://doi.org/10.5281/zenodo.8344248.

David F Hahn, Christopher I Bayly, Melissa L Boby, Hannah E Bruce Macdonald, John D Chodera, Vytautas Gapsys, Antonia SJS Mey, David L Mobley, Laura Perez Benito, Christina EM Schindler, et al. Best practices for constructing, preparing, and evaluating protein-ligand binding affinity benchmarks [article v1. 0]. *Living journal of computational molecular science*, 4(1):1497, 2022.

Tong He, Marten Heidemeyer, Fuqiang Ban, Artem Cherkasov, and Martin Ester. SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of Cheminformatics*, 9:24, 2017. doi: 10.1186/s13321-017-0209-z.

Kevin G Hicks, Ahmad A Cluntun, Heidi L Schubert, Sean R Hackett, Jordan A Berg, Paul G Leonard, Mariana A Ajalla Aleixo, Youjia Zhou, Alex J Bott, Sonia R Salvatore, et al. Protein-metabolite interactomics of carbohydrate metabolism reveal regulation of lactate dehydrogenase. *Science*, 379(6636):996–1003, 2023.

Isomorphic Labs Team. *Accurate Predictions of Novel Biomolecular Interactions with IsoDDE*, February 2026.

José Jiménez, Miha Škalič, Gerard Martínez-Rosell, and Gianni De Fabritiis. KDEEP: Protein–ligand absolute binding affinity prediction via 3D-convolutional neural networks. *Journal of Chemical Information and Modeling*, 58(2):287–296, 2018. doi: 10.1021/acs.jcim.7b00650.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–D1380, 2023.

Justin Klekota and Frederick P. Roth. Chemical substructures that enrich for biological activity. *Bioinformatics*, 24(21):2518–2525, 2008. doi: 10.1093/bioinformatics/btn479.

Huan Yee Koh, Anh T. N. Nguyen, Shirui Pan, Lauren T. May, and Geoffrey I. Webb. Physicochemical graph neural network for learning protein-ligand interaction fingerprints from sequence data. *Nature Machine Intelligence*, 6:673–687, 2024. doi: 10.1038/s42256-024-00847-1.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2825–2834. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/li18a.html.

Tiqing Liu, Linda Hwang, Stephen K. Burley, Carmen I. Nitsche, Christopher Southan, W. Patrick Walters, and Michael K. Gilson. BindingDB in 2024: a FAIR knowledgebase of protein-small molecule binding data. *Nucleic Acids Research*, 53(D1):D1633–D1644, 2025. doi: 10.1093/nar/gkae1075. URL https://doi.org/10.1093/nar/gkae1075.

Wei Lu, Qifeng Wu, Jixian Zhang, Jiahua Rao, Chengtao Li, and Shuangjia Zheng. TANKBind: Trigonometry-aware neural networks for drug-protein binding structure prediction. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.neurips.cc/paper_files/paper/2022/hash/2f89a23a19d1617e7fb16d4f7a049ce2-Abstract-Conference.html.

Wei Lu, Jixian Zhang, Weifeng Huang, Ziqiao Zhang, Xiangyu Jia, Zhenyu Wang, Leilei Shi, Chengtao Li, Peter G. Wolynes, and Shuangjia Zheng. DynamicBind: predicting ligand-specific protein-ligand complex structure with a deep equivariant generative model. *Nature Communications*, 15:1071, 2024. doi: 10.1038/s41467-024-45461-2.

Björn Mattsson and W Patrick Walters. Identifying and addressing systematic data leakage in protein-ligand affinity benchmarks. *bioRxiv*, pp. 2026–06, 2026.

Alex Morehead and Jianlin Cheng. FlowDock: Geometric flow matching for generative protein–ligand docking and affinity prediction. *Bioinformatics*, 41(Supplement$_1$) : $i198 - -i206, 2025. doi$: .

Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. *Journal of Cheminformatics*, 10:4, 2018. 10.1186/s13321-018-0258-y.

Thin Nguyen, Hang Le, Thomas P. Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. GraphDTA: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37(8):1140–1147, 2021. 10.1093/bioinformatics/btaa921.

Ramaswamy Nilakantan, Norman Bauman, J. Scott Dixon, and R. Venkataraghavan. Topological torsion: a new molecular descriptor for SAR applications. comparison with other descriptors. *Journal of Chemical Information and Computer Sciences*, 27(2):82–85, 1987. 10.1021/ci00054a008.

NVIDIA. NVIDIA H100 Tensor Core GPU. https://www.nvidia.com/en-us/data-center/h100/, a. Product specifications, H100 SXM configuration. Accessed September 26, 2026.

NVIDIA. NVIDIA L4 Tensor Core GPU. https://www.nvidia.com/en-us/data-center/l4/, b. Product specifications. Accessed September 26, 2026.

Fabian Offensperger, Gary Tin, Miquel Duran-Frigola, Elisa Hahn, Sarah Dobner, Christopher W am Ende, Joseph W Strohbach, Andrea Rukavina, Vincenth Brennsteiner, Kevin Ogilvie, et al. Large-scale chemoproteomics expedites ligand discovery and predicts ligand behavior in cells. *Science*, 384(6694):eadk5864, 2024.

Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018. 10.1093/bioinformatics/bty593.

Joseph M. Paggi, Ayush Pandit, and Ron O. Dror. The art and science of molecular docking. *Annual Review of Biochemistry*, 93:389–410, 2024. 10.1146/annurev-biochem-030222-120000.

Tapio Pahikkala, Antti Airola, Sami Pietilä, Sushil Shakyawar, Agnieszka Szwajda, Jing Tang, and Tero Aittokallio. Toward more realistic drug–target interaction predictions. *Briefings in Bioinformatics*, 16(2):325–337, 2015. 10.1093/bib/bbu010.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025. 10.1101/2025.06.14.659707.

RDKit contributors. RDKit: Open-source cheminformatics. Computer software. URL https://www.rdkit.org/.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. 10.1021/ci100050t.

Gregory A. Ross, Chao Lu, Guido Scarabelli, Steven K. Albanese, Evelyne Houang, Robert Abel, Edward D. Harder, and Lingle Wang. The maximal and current accuracy of rigorous protein-ligand binding free energy calculations. *Communications Chemistry*, 6(1):222, 2023. 10.1038/s42004-023-01019-9. URL https://doi.org/10.1038/s42004-023-01019-9.

Matteo Rossi, Ryan Pederson, Miles Wang-Henderson, Ben Kaufman, Edward C. Williams, Carl Underkoffler, Owen Lewis Howell, Adrian Layer, Stephan Thaler, Narbe Mardirossian, and John Anthony Parkhill. TerraBind: Fast and accurate binding affinity prediction through coarse structural representations, 2026. URL https://arxiv.org/abs/2602.07735.

SandboxAQ. AQAffinity: Open-source structure-to-affinity built on OpenFold3 by SandboxAQ, January 2026. URL https://www.sandboxaq.com/post/aqaffinity-open-source-structure-to-affinity-built-on-openfold3-by-sandboxaq.

scikit-fingerprints contributors. PubChemFingerprint. scikit-fingerprints documentation. URL https://scikit-fingerprints.readthedocs.io/latest/modules/generated/skfp.fingerprints.PubChemFingerprint.html. Accessed September 25, 2026.

Chao Shen, Xujun Zhang, Shukai Gu, Odin Zhang, Qinghan Wang, Gang Du, Yihao Zhao, Linlong Jiang, Peichen Pan, Yu Kang, et al. Unlocking the application potential of alphafold3-like approaches in virtual screening. *Chemical Science*, 17(5):2858–2879, 2026.

Nikhil Shenoy, David Errington, Emmanuel Bengio, Kacper Kapuśniak, Kerstin Klaeser, Yui Tik Pang, Vladimir Radenkovic, Prudencio Tossou, Therence Bois, Andrew Wedlake, and Francesco Di Giovanni. Nesso-1: Accelerating open-source binding affinity predictions. *bioRxiv*, 2026. 10.64898/2026.08.01.742196. URL https://www.biorxiv.org/content/10.64898/2026.08.01.742196v1.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017. 10.1038/nbt.3988.

Marta M. Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018. 10.1093/bioinformatics/bty374.

The Gene Ontology Consortium. The Gene Ontology knowledgebase in 2026. *Nucleic Acids Research*, 54(D1):D1779–D1792, 2026. 10.1093/nar/gkaf1292.

The OpenFold3 Team. Openfold3-preview, 2025. URL https://github.com/aqlaboratory/openfold-3.

The UniProt Consortium. UniProt: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, 2025. 10.1093/nar/gkae1010.

Lingle Wang, Yujie Wu, Yuqing Deng, Byungchan Kim, Levi Pierce, Goran Krilov, Dmitry Lupyan, Shaughnessy Robinson, Markus K. Dahlgren, Jeremy Greenwood, Donna L. Romero, Craig Masse, Jennifer L. Knight, Thomas Steinbrecher, Thijs Beuming, Wolfgang Damm, Ed Harder, Woody Sherman, Mark Brewer, Ron Wester, Mark Murcko, Leah Frye, Ramy Farid, Teng Lin, David L. Mobley, William L. Jorgensen, Bruce J. Berne, Richard A. Friesner, and Robert Abel. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *Journal of the American Chemical Society*, 137(7):2695–2703, 2015. 10.1021/ja512751q.

Ruibo Zhang, Daniel Nolte, Cesar Sanchez-Villalobos, Souparno Ghosh, and Ranadip Pal. Topological regression as an interpretable and efficient tool for quantitative structure-activity relationship modeling. *Nature Communications*, 15:5072, 2024. 10.1038/s41467-024-49372-0.

Qichang Zhao, Guihua Duan, Mengyun Yang, Zhongjian Cheng, Yaohang Li, and Jianxin Wang. Attentiondta: Drug–target binding affinity prediction by sequence-based deep learning with attention mechanism. *IEEE/ACM transactions on computational biology and bioinformatics*, 20(2): 852–863, 2022.

A TRAINING DATA CURATION

B TRAINING DATA CURATION

We built one dataset of experimentally measured, equality-qualified binding affinities from ChEMBL 34 and the 1 April 2024 BindingDB snapshot. STORM and all structure-free comparison methods were trained using this dataset. No model in this comparison was trained on inequality-qualified measurements. We did not apply a publication-date cutoff. The steps below distinguish curation shared by the comparison methods from processing used for the canonical STORM fit.

B.1 GENERAL-PURPOSE CURATION

1. **Select measurements and assays.** We retained equality-qualified, positive concentrations in convertible units for K_i, K_d, IC_{50}, and EC_{50} from both sources, and AC_{50} and XC_{50} from ChEMBL. ChEMBL records required a single-protein target, one UniProt mapping, assay confidence score 9, direct target assignment, and a binding or functional assay. We excluded records flagged as potential duplicates or carrying validity annotations other than "Manually validated." ChEMBL mutant and variant assays were retained and flagged. BindingDB records required a single-chain target and a documented assay mapping.

2. **Resolve protein targets.** We assigned canonical UniProt sequences to retained targets and excluded records without an unambiguous mapping. BindingDB targets were resolved through trusted UniProt accessions or sequence matches to ChEMBL 34 and unambiguous BindingDB Swiss-Prot anchors. Sequence-based mappings required at least 90% identity and 90% coverage of both sequences. Canonical sequences were used as model inputs, including for records flagged as mutant or variant assays.

3. **Standardize ligands and values.** Ligands underwent RDKit cleanup, largest-fragment selection with an organic-fragment preference, functional-group normalization, metal disconnection, and reionization, followed by canonical isomeric SMILES generation. We excluded unparseable structures, molecules with more than 50 heavy atoms, and molecules with exact molecular weight above 1,000 Da. Concentrations were converted to $y = \log_{10}(c/1\,\mu\mathrm{M})$, with endpoint identities retained. We did not apply a PAINS filter, a model-derived structural-confidence filter, or a minimum assay-size threshold.

4. **Remove evaluation overlap.** Using MMseqs2, we removed training proteins in clusters containing OpenFE or FEP+4 targets at 90% sequence identity (`cov-mode=0`, `c=0.01`). Across both sources, we also removed records sharing a UniProt target and standardized ligand with CASP16 or IsoDDE evaluation records.

5. **Define series and remove duplicate assays.** We grouped measurements by source assay, target, and endpoint. BindingDB series used Entry DOI, canonical target, and endpoint, with Article DOI and then patent identifiers as fallbacks. Within- and cross-source assays were considered duplicates only when their complete standardized compound sets matched and paired concentrations agreed within 5%. ChEMBL was preferred for cross-source duplicates. Partially overlapping assays and assays with differing measurements remained separate. We preserved source record and assay identifiers for traceability.

B.2 CANONICAL STORM PROCESSING

6. **Consolidate repeated assay–compound measurements.** For the canonical STORM fit, repeated measurements of the same standardized molecule within a series were averaged when their sample standard deviation was at most 0.5 in $\log_{10}(\mu\mathrm{M})$ units. Groups exceeding this threshold were removed.

7. **Construct the regression objective.** The canonical model combines pointwise Huber loss on exact affinity values with pairwise Huber loss on differences between compounds in the same assay. Assays with fewer than five compounds can contribute to the pointwise term but do not supply sampled pairs. No bounded-value or binary classification loss is used.

C Protein Target and Molecule Representations

GO and InterPro representations. For the GO and InterPro representations, we only keep terms that occur in the training targets. Terms that always appear together at those targets are collapsed into a single feature. After this grouping, the features are additively hashed into separate 2,048-dimensional GO and InterPro vectors.

Target ID representation. ID128 represents each target's UniProt accession as a deterministic 128-dimensional binary vector with 64 active entries. We generate the vector by hashing the accession and assign distinct vectors to distinct accessions in the target codebook. This feature identifies the target but does not encode its sequence or annotations. For an unseen target, the combined target configuration uses the ID128 code of its closest training target identified by MMseqs2 sequence search; the ID-only configuration instead uses a code generated from the unseen target's own accession.

Target configurations. Our five target configurations are all four base features together; GO alone; InterPro alone; target ID alone; and a target-blind configuration with no target features. The species indicator is included only in the combined configuration.

Fingerprint Representations Our fingerprint representations consist of some count-based fingerprints, which have positive integer feature counts, and binary fingerprints that indicate the presence or absence of a feature. The 2D pharmacophore fingerprint is folded into 2,048 additive count bins. Our merged RDKit/Mordred descriptors are formed from 411 selected RDKit and 934 selected Mordred descriptors. These are a subset of the total features calculated by the RDKit package and the Mordred package that were chosen for having less than 1% of values missing, and for columns that had correlations of less than 98%. Then, we completed a second de-duplication pass merging the RDKit features and Mordred features at the same threshold, leaving the final 1,212 descriptors we used.

D Model Details

Canonical model inputs. For the canonical model, inequality-qualified measurements were excluded from fitting. Repeated assay–compound measurements were averaged in logarithmic units when their sample standard deviation was at most 0.5; groups exceeding this threshold were removed. This yielded 2,216,239 training rows across 174,946 series. These fit-time operations were separate from the assay deduplication described above.

Target representation	Ligand input	Votes
Canonical (ID128 + GO + InterPro + species)	Atom pair count	1
Canonical	Morgan count	1
Canonical	2D pharmacophore count	1
Canonical	RDKit/Mordred graph-collapsed 1,212 descriptors	2
Canonical	Topological torsion count	1
Target blind	Atom pair count	1
Target blind	Avalon binary	1
Target blind	Morgan count	1
Target blind	RDKit path count	2
GO only	2D pharmacophore count	1
GO only	RDKit path count	2
InterPro only	Klekota-Roth count	1
InterPro only	PubChem count	1
InterPro only	RDKit/Mordred graph-collapsed 1,212 descriptors	1
ID128 only	Atom pair count	1

Weighted Ensemble. STORM is a weighted ensemble model, with 15 component models that are independently trained and then weighted with 1 or 2 votes. The number of votes for each model was decided based on fitting to a held-out validation assay set. The full procedure is below.

Table 5: Greedy component-removal ablation of the selected STORM ensemble. At each step, the component whose removal maximizes the mean of OpenFE and ChEMBL35 Time Split performance is removed. The remaining predictions retain their original votes and receive the fixed graph-smoothing step. Values are compound-weighted within-assay Pearson r, reported as mean $\pm$ sample standard deviation across three independently trained seeds.

Models left	Votes left	Component removed	OpenFE	ChEMBL35
15	18	Full ensemble	0.5814 ± 0.0089	0.3529 ± 0.0040
14	17	Target-blind atom pair	0.5815 ± 0.0086	0.3540 ± 0.0034
13	16	Canonical 2D pharmacophore	0.5835 ± 0.0091	0.3517 ± 0.0049
12	15	ID-only atom pair	0.5816 ± 0.0077	0.3519 ± 0.0033
11	14	InterPro-only Klekota–Roth	0.5827 ± 0.0106	0.3477 ± 0.0030
10	13	InterPro-only merged descriptors	0.5798 ± 0.0103	0.3464 ± 0.0029
9	12	Target-blind Avalon	0.5750 ± 0.0104	0.3469 ± 0.0026
8	11	Canonical Morgan	0.5772 ± 0.0068	0.3396 ± 0.0043
7	10	GO-only 2D pharmacophore	0.5756 ± 0.0087	0.3348 ± 0.0040
6	9	InterPro-only PubChem	0.5708 ± 0.0115	0.3293 ± 0.0048
5	8	Target-blind Morgan	0.5639 ± 0.0103	0.3245 ± 0.0067
4	7	Canonical atom pair	0.5565 ± 0.0136	0.3149 ± 0.0056
3	6	Canonical topological torsion	0.5432 ± 0.0137	0.3025 ± 0.0086
2	4	GO-only RDKit path	0.5293 ± 0.0125	0.2824 ± 0.0117
1	2	Canonical merged descriptors	0.4949 ± 0.0057	0.2149 ± 0.0055

The component models are trained independently. For model k, let $\hat{y}_{ik}$ be its prediction for example i and let $v_k \in \{1, 2\}$ be its number of votes. The ensemble prediction is

$$\hat{y}_i^{(0)} = \frac{\sum_{k=1}^{15} v_k \hat{y}_{ik}}{\sum_{k=1}^{15} v_k}, \qquad \sum_{k=1}^{15} v_k = 18. \tag{4}$$

A two-vote component contributes twice to this average; it is still a single trained model.

E Ablations

Contribution of ensemble components. To examine how the components contribute to STORM, we start with the 15-model, 18-vote ensemble and remove one model at a time. At each step, we remove the model whose deletion gives the highest mean of OpenFE and ChEMBL35 Time Split performance among the remaining models. The surviving models retain their original vote weights, and we apply the same graph-smoothing rule after each removal. Table 5 shows that performance changes little when the first few components are removed, but falls as the ensemble becomes smaller. With five models remaining, OpenFE Pearson r decreases from 0.5814 to 0.5639, and ChEMBL35 Pearson r decreases from 0.3529 to 0.3245. The removal order is greedy and depends on the models already removed, so it should not be interpreted as an independent measure of each component's importance.

Diverse representations versus repeated fits of one model. We ask whether the gain from combining different representations could instead be obtained by training more instances of a single strong component. Among the 15 selected components, the target-blind RDKit path model has the highest mean standalone performance across OpenFE and ChEMBL35 Time Split. We train 15 independent instances of this component with different seeds, average their predictions, and apply the same graph-smoothing rule used for STORM. As shown in Table 6, this ensemble reaches 0.5055 on OpenFE and 0.2562 on ChEMBL35, compared with 0.5814 and 0.3529 for the diverse 15-component ensemble. The result suggests that combining complementary molecule and target representations contributes more than repeatedly fitting this single component.

F Per-class ChEMBL35 Time Split Results

Table 7 contains compound-weighted Pearson r correlations for the ChEMBL35 Time Split, broken down by the six classes–Enzyme, GPCR, Ion Channel, Kinase, Other, and Transcription Factor.

Table 6: A diverse 15-component ensemble compared with 15 independent fits of the strongest single component, target-blind RDKit path. All predictions receive the same graph-smoothing step. The repeated-component column is one pooled prediction from 15 independent fits; the STORM column reports mean ± sample standard deviation across three independently trained ensembles. Values are compound-weighted within-assay Pearson r.

Evaluation set	15 RDKit path fits	Diverse STORM
CASP16	0.6613	0.6674 ± 0.0086
FEP+4	0.6569	0.6944 ± 0.0230
OpenFE	0.5055	0.5814 ± 0.0089
ChEMBL35 Time Split	0.2562	0.3529 ± 0.0040

Model	Overall ($n = 449$)	Enzyme ($n = 135$)	GPCR ($n = 97$)	Ion Channel ($n = 17$)	Kinase ($n = 115$)	Other ($n = 51$)	Transcription Factor ($n = 34$)
STORM	$\mathbf{0.353 \pm 0.004}$	$\mathbf{0.318 \pm 0.012}$	$\mathbf{0.412 \pm 0.012}$	0.319 ± 0.019	0.352 ± 0.009	$\mathbf{0.337 \pm 0.013}$	$\mathbf{0.348 \pm 0.015}$
Boltz-2	0.341 ± 0.009	0.307 ± 0.017	0.298 ± 0.019	0.143 ± 0.047	$\mathbf{0.416 \pm 0.018}$	0.336 ± 0.028	0.347 ± 0.033
Structure-free DTA	0.235 ± 0.013	0.178 ± 0.027	0.292 ± 0.030	$\mathbf{0.331 \pm 0.061}$	0.217 ± 0.019	0.258 ± 0.046	0.284 ± 0.068
AttentionDTA	0.154 ± 0.014	0.108 ± 0.030	0.263 ± 0.027	0.098 ± 0.057	0.110 ± 0.020	0.187 ± 0.043	0.179 ± 0.059
PSICHIC	0.139 ± 0.015	0.132 ± 0.033	0.155 ± 0.038	0.198 ± 0.071	0.151 ± 0.020	0.104 ± 0.049	0.074 ± 0.071
BALM	0.209 ± 0.014	0.198 ± 0.029	0.254 ± 0.027	0.225 ± 0.066	0.186 ± 0.023	0.238 ± 0.043	0.178 ± 0.062
Mattsson-Walters	0.228 ± 0.014	0.227 ± 0.029	0.297 ± 0.033	0.113 ± 0.087	0.202 ± 0.022	0.207 ± 0.049	0.245 ± 0.056

Table 7: Compound-weighted within-assay Pearson correlation on the ChEMBL35 Time Split benchmark. Parenthetical n values count assays. Boltz-2 values are reported references.

There are 449 targets total, and we label each class with the number of assays measured against it. We use the Boltz-2 results reported in the IsoDDE technical report. We calculate an Overall score for Boltz-2 is calculated by weighting each class's reported Pearson r by its number of evaluated compounds, and by estimating the standard error from the 95% CI reported in Figure 8 of the IsoDDE paper. The IsoDDE ChEMBL35 Time Split benchmark figure does not specify whether the correlation is compound weighted; however, since the CASP16, FEP+4, and OpenFE figures show compound weighted Pearson r, we assume the ChEMBL35 benchmark is the same.

G Estimating L4-equivalent GPU hours needed to Pretrain Boltz-2

In Figure 2, we estimate the GPU hours it would take to train Boltz-2 from scratch on an NVIDIA L4 GPU. We use the following steps to reach this estimation.

We use OpenFold3-preview 2 (The OpenFold3 Team, 2025), a comparable cofolding only method, as our reference point, as some training details are available. The model was trained for 14 days on 256 NVIDIA H100 GPUs, leading to 86,016 H100 GPU hours. OpenFold3-preview 2 was trained on 13M structures, and Boltz-2 was trained on 5M structures (both include distilled structures). For simplicity, we assume a linear relationship between compute and the size of the distilled training dataset (this may not always be the case). Finally, H100s have a peak BF16 Tensor Core throughput of 1,979 TFLOP/s (NVIDIA, a), compared to the L4's 242 TFLOP/s (NVIDIA, b). Mutliplying the 86,016 H100 GPU hours by the TFLOP/s and training dataset sizes gives us

$$86,016 \times \frac{1,979}{242} \times \frac{5}{13} \approx 270,000 \text{ L4-equivalent GPU hours}$$

We once again emphasize that this is a very rough estimation, and many details about Boltz-2's runtime are not available publicly. We intend this estimate only to indicate the approximate order of magnitude of Boltz-2's training compute.

H Experimental Setup

H.1 Additional Details on Finetuning Study

We evaluated finetuning on three targets absent from the base model's training set. For LpxH, we used the larger of two nonoverlapping assays as the support set (92 compounds) and the smaller

assay as the held-out test set (42 compounds). A0A0B5JFF1 had one eligible assay with 80 compounds; a fixed random compound split assigned 56 to support and 24 to test. For ALPK1, we used a within-assay demonstration split rather than the available assay pair. We clustered compounds using 2,048-bit radius-2 Morgan fingerprints and RDKit Butina clustering with a Tanimoto distance cutoff of 0.5, then selected 80 support and 40 test compounds across the clusters. The remaining 318 eligible compounds were not used. This procedure sampled compounds using the clusters; it did not hold out entire clusters.

For each target, we selected finetuning settings using three independent 80/20 partitions of the support compounds. We trained on each partition's training portion and ranked settings by mean Pearson correlation on its held-out support portion, using the canonical Morgan component from pretrained seed 2023 for the search. The test compounds were excluded from this selection. We then restarted all 15 components from their epoch-45 pretrained checkpoints, finetuned each component on the complete support set for each of the three pretrained seeds, and combined their predictions using the original 18 votes. Full-model finetuning used a learning rate of 10^{-5} and an L2-SP penalty weight of 0.1 for all three targets, with 50 steps for LpxH, 100 for A0A0B5JFF1, and 200 for ALPK1. We compared each adapted ensemble with its corresponding untouched base ensemble on the same held-out test compounds.

H.2 Finetuning Loss

For each target, we initialized the model from its pretrained checkpoint and fine-tuned it on the support assay. We updated all model parameters, minimizing

$$\mathcal{L}_{\mathrm{FT}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{SmoothL1}(f_\theta(x_i), y_i) + \lambda \sum_{p\in\mathcal{P}} \left\|\theta_p - \theta_p^{(0)}\right\|_2^2, \tag{5}$$

where $\theta^{(0)}$ denotes the pretrained parameters and $\mathcal{P}$ contains the parameters updated in the chosen adaptation mode. The second term is L2-SP regularization, which penalizes movement away from the pretrained starting point (Li et al., 2018). The penalty is summed over parameters without normalization.

We selected the number of update steps and λ using validation partitions of the support assay only. We then restarted from the pretrained checkpoint, fine-tuned on all support compounds using the selected settings, and evaluated once on the held-out test assay. Test-assay labels were not used for adaptation or hyperparameter selection.