SPADE: Fast Drug Discovery by Learning from Sparse Data

Rahul Nandakumar
McCombs School of Business
University of Texas at Austin
Austin, TX
rahul.nandakumar@utexas.edu

Ben Fauber
NVIDIA
bfauber@nvidia.com

Deepayan Chakrabarti
McCombs School of Business
University of Texas at Austin
Austin, TX
deepay@utexas.edu

Abstract

Drug discovery seeks molecules (ligands) that bind strongly and selectively to a target protein. However, fewer than 5% of candidate ligands pass the bar for even the early stages of drug discovery. Furthermore, we want methods that work for novel proteins for which we have no prior data. Starting from scratch, we have to iteratively select and test candidate ligands such that we find enough ligands of the desired quality in as few tests as possible. Our proposed algorithm, named SPADE, introduces a novel approach to ligand selection that requires only 40 tests on average to find 10 high-quality ligands. In one-vs-one comparisons, SPADE outperforms deep learning and Bayesian optimization methods on more proteins, achieving median improvements of $7\% - 32\%$ in sample efficiency. SPADE is also 10x faster than its closest competitor at scoring candidate drugs. Dataset and code is available at https://anonymous.4open.science/r/SPADE_Fast_Drug_Discovery_by_Learning_from_Sparse_Data-F028/README.md

1 Introduction

Proteins regulate nearly all biological processes in the human body, and disruptions to their activity can lead to disease. A central goal of drug discovery is therefore to design small molecules (ligands) that bind strongly and selectively to a target protein [18, 44]. Despite decades of progress, this process remains highly inefficient: fewer than 5% of candidate ligands succeed even in early-stage screening. The challenge is particularly acute for *novel* proteins, where little or no prior data is available.

In early-stage drug discovery, researchers iteratively run design-make-test-analyze (DMTA) cycles [31]. In each cycle, a small number of ligands are selected, synthesized, and experimentally evaluated. The outcome of interest is the ligand's binding affinity, typically measured via the pIC50 (PIC) value [40], where higher values indicate stronger binding. Since experimental tests are slow and expensive, the key objective is to identify a small number of (say, 5) high-quality ligands (e.g., PIC ≥ 8) using as few tests as possible. We refer to this objective as a *race-to-k* problem, that is:

find k ligands with PIC above a desired threshold in as few DMTA cycles as possible.

A PIC of 8 or 8.5 is often a favorable starting point for the later stages of drug discovery [22]. Hence, we focus on the *race-to-8* and *race-to-8.5* problems.

This setting presents several fundamental challenges. First, the search space is vast, consisting of millions of candidate ligands represented by high-dimensional embeddings. Second, the signal is extremely sparse: only a small fraction of ligands meet the desired affinity threshold (e.g., only 7% of candidates have PIC ≥ 8, and just 2.7% have PIC ≥ 8.5). Third, we start with no labeled data and must learn entirely from sequential, adaptively collected observations. These properties make standard machine learning approaches difficult to apply effectively.

Preprint.

Recent work approaches this problem using active learning and Bayesian optimization. These methods estimate the PIC of each candidate ligand and combine it with an uncertainty estimate (e.g., from a Gaussian process) to guide selection [11]. However, in our setting, accurate estimation is difficult due to extreme data sparsity and high dimensionality: we begin with no data, and each ligand is represented by a $2{,}048$-dimensional embedding. Moreover, most ligands fail to meet the desired threshold. As a result, estimating PICs is both difficult and unnecessary for the race-to-k objective.

Another line of work constructs protein embeddings [30, 33]. However, directly predicting affinity using such embeddings achieves limited accuracy [6, 9]. Since we focus on novel proteins, we assume that no similar proteins are available, and thus do not rely on protein embeddings. If such information is available, it can be incorporated as a preprocessing step to filter the candidate ligand set (we show such an experiment in Appendix B).

Finally, we emphasize that our work focuses on **ligand affinity, which is fundamentally distinct from molecular docking** [38]. Our approach is binding-mode agnostic and focuses on continuous affinity (PIC) values. In contrast, datasets such as DUD-E [26] provide only docking scores, while LIT-PCBA [42] offers binary active/inactive labels. Neither dataset captures the continuous affinity information essential for ranking and prioritizing ligands, rendering them unsuitable for our objective.

Our contributions:

- **Problem formulation:** We cast early-stage drug discovery for novel proteins as a sparse, sequential *race-to-k* problem, where the objective is to identify k high-affinity ligands using as few DMTA cycles as possible.
- **Classification-based selection (Sec. 3):** We propose SPADE (**S**parse-data **P**redictions for **A**ccelerating **D**rug **E**xploration), which replaces full PIC estimation with the simpler task of predicting whether a ligand can outperform the current k^{th}-best ligand. Thus, SPADE focuses directly on improving the top-k set.
- **Robust learning under extreme sparsity (Sec. 3):** Since the positive class contains only $k-1$ ligands, standard classifiers struggle. Instead, we introduce a robust classifier that minimizes the expected loss over a Gaussian centered at each positive example, reducing overfitting to noise. We derive a closed-form expression for the expected loss to avoid sampling. We then combine these classifiers to efficiently identify the best candidate ligands.
- **Large-scale dataset (Sec. 4):** We built a new 1.5M-entry PubChem-derived dataset, complementary to existing datasets, to test sequential ligand discovery under realistic sparsity.
- **Empirical evaluation (Sec. 4):** Across 100 proteins, SPADE consistently outperforms state-of-the-art baselines, requiring $7\%-32\%$ fewer ligand tests to reach target PICs and achieving a $10\times$ speedup in scoring ligands.

2 Related Work

We review prior work on Ligand-Protein Interaction affinity (LPI) prediction, protein and ligand embeddings, and active-learning for drug discovery.

LPI prediction: Several papers cast LPI prediction as a binary classification task. Then, they apply machine learning models to learn affinities [7, 17, 23–25, 27, 47]. Deep learning models have also been developed for this task [13, 14, 19–21, 28, 45, 46]. However, simple binary labels, such as active/inactive, oversimplifies the continuous nature of binding affinities. Hence, these methods have been less useful for ranking ligands for drug screening [36]. In contrast, SPADE considers only the current top-k ligands as the positive class, so the labels shift over the DMTA cycles.

Another approach is to use physics-based methods like free-energy perturbation. These methods attempt to mimic the protein-ligand binding interactions via computational simulations. They can offer more precise results but are computationally intensive [35, 37, 43]. Hence, even if these methods are used instead of lab tests in DMTA cycles, we still need to minimize the number of cycles.

Protein and ligand embeddings: Existing approaches try to predict ligand-protein interaction affinity using vector embeddings [15, 17] and graph-based models [2, 17, 39, 41] for protein and ligand embeddings. Several specialized embeddings also exist for proteins [30, 33] and for ligands [5, 34]. However, embedding methods have limited impact on prediction accuracy [6, 9].

We focus on drug discovery for novel proteins. So, we assume there is no side information available about similar proteins. In our setting, we only have one protein (the target protein), so there is no need for protein embeddings. Our work is agnostic to the choice of ligand embeddings, and we show results using the popular ECFP [34], MACCS [5], and ChemBERTa [3] ligand embeddings.

Active learning and Bayesian optimization: Since most ligands have poor LPI, existing datasets are imbalanced. Active learning has been used to develop balanced training datasets for LPI prediction via explore-exploit strategies. However, the best way to find good interacting pairs is to only exploit [16]. Also, active learning is often used to learn the entire protein-ligand interaction landscape. In contrast, the goal of early-stage drug discovery is to quickly find a few high-quality ligands. Learning about the entire landscape is not necessary.

Bayesian optimization is another approach for selecting the ligands to test in each DMTA cycle. Here, we rank candidate ligands by combining their estimated PIC with uncertainty scores obtained via Gaussian processes [11]. The top-ranked ligands are tested in the next DMTA cycle, and all estimates are recomputed using the new data. But in our setting, there are very few ligands with known PICs, and the ligand embedding is high-dimensional. This leads to large uncertainty for ligand PIC estimates. SPADE avoids estimation and instead uses robust classification. We show that we outperform Bayesian optimization methods, and are also 10x faster than the closest competitor.

Structure-based and deep virtual screening: These methods analyze the ligand and protein structures to predict docking. However, docking is different from affinity prediction. Also, the best-performing methods typically require seconds of compute per ligand even with GPU acceleration [30], while SPADE is 3-4 orders of magnitude faster. Finally, such methods do not learn from DMTA cycles, unlike SPADE. We note that screening methods can filter the set of ligands as a preprocessing step for SPADE. This can improve accuracy by 18%-31% (Appendix B).

3 Proposed Method

Our goal is to identify several high-affinity ligands using as few DMTA cycles as possible. This setting presents several key challenges:

(1) Rare targets in a vast search space. We must search among 10^5–10^6 candidate ligands, with a median PIC of 5.9. The distribution is highly skewed: about 75% of ligands have PIC < 7, only 7% have PIC ≥ 8, and just 2.7% exceed 8.5.

(2) Sparse data. For novel proteins, we begin with no labeled data. Each data point must be obtained via costly and time-consuming ligand tests, so the model must learn effectively from little data. Furthermore, since our goal is to improve the top-k ligands in each DMTA cycle, the "positive class" remains extremely small (namely, the $k-1$ ligands that are currently the best).

(3) High dimensionality. Each ligand is represented by a $2{,}048$-dimensional ECFP vector [34]. This high dimensionality, combined with data sparsity, makes accurate PIC estimation difficult.

(4) High throughput requirements. In each DMTA cycle, we must score $10^5 - 10^6$ candidate ligands. Hence, the scoring needs to be fast and efficient.

3.1 Overview of SPADE

SPADE departs from standard approaches in two fundamental ways: it replaces global affinity estimation with a targeted classification objective, and it incorporates robustness directly into the learning process. Together, these design choices yield a method that is well-suited to extreme data sparsity, requires minimal tuning, and is significantly faster at scoring candidate ligands.

Classification instead of estimation. Most existing methods attempt to estimate the PIC for every candidate ligand. In our low-data setting, this is both difficult and unnecessary. Instead, SPADE focuses on a simpler and more targeted task: predicting whether a ligand can outperform the current k^{th}-best ligand. This directly optimizes for our goal of improving the top-k set in each DMTA cycle.

Robustness under extreme sparsity. The positive class consists of only the top-k ligands seen so far. To be robust against noise, SPADE optimizes an expected loss over a distribution centered at each positive ligand. This acts as a principled regularizer, encouraging the model to learn stable patterns

rather than overfitting to a handful of points. We further design the loss so that this expectation admits a closed-form expression, avoiding the need for sampling.

In each DMTA cycle, SPADE trains one robust classifier for each of the current top-k ligands. It then scores all untested ligands using a weighted combination of these classifiers, and selects the highest-scoring candidates for testing in the next cycle. This process iteratively refines the top-k set. Next, we discuss these steps in detail.

3.2 Classifier for a Single Top Ligand

After every DMTA cycle, we have a dataset $\mathcal{D}_{seen}$ of the ligands tested so far, along with their PICs. Let $\mathcal{S}^+ \subset \mathcal{D}_{seen}$ be the set of the best few ligands seen so far, and let $\mathcal{S}^- := \mathcal{D}_{seen} \setminus \mathcal{S}^+$. For each ligand $i \in \mathcal{S}^+$, we build a classifier $\mathcal{C}_i$ that separates i from all the ligands in $\mathcal{S}^-$:

$$\mathcal{C}_i := \arg\min_{\mathcal{C} \in \mathcal{F}} \left(\frac{\sum_{j \in \mathcal{S}^-} \ell(\mathcal{C}(\boldsymbol{x}_j), y = -1)}{|\mathcal{S}^-|} + E_{\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}_i, \sigma^2 I)} \left[\ell(\mathcal{C}(\boldsymbol{x}), y = 1) \right] \right) \tag{1}$$

where $\boldsymbol{x}_j \in \mathbb{R}^d$ denotes the embedding of ligand j, $\mathcal{F}$ is the search space over classifiers, $\mathcal{C}$ is any classifier in $\mathcal{F}$, $C(\boldsymbol{x})$ is the predicted score of C on a ligand with embedding $\boldsymbol{x}$, and $\ell(s, y)$ is the loss if a score s is predicted for a ligand with class $y \in \{+1, -1\}$.

Difference from empirical risk minimization: Instead of the empirical loss on $\mathcal{S}^+$, we use the *expected loss* over a Gaussian distribution. This distribution is centered at $\boldsymbol{x}_i$ for each $i \in \mathcal{S}^+$. Intuitively, $\boldsymbol{x}_i$ is a sample from a distribution of similar-PIC ligands. Since we do not have enough data to reconstruct that distribution, we use a Gaussian distribution as an approximation. The width σ of the Gaussian reflects our uncertainty about the distribution. Larger σ means greater uncertainty and a more robust classifier $\mathcal{C}_i$. A well-chosen $\sigma > 0$ ensures that $\mathcal{C}_i$ identifies the specific features of $\boldsymbol{x}_i$ that (a) set it apart from the low-PIC ligands in $\mathcal{S}^-$ and (b) are not due to randomness.

Choice of loss function: We choose $\ell(\cdot, \cdot)$ so that Equation 1 has a closed-form formula. Specifically, let any classifier $\mathcal{C} \in \mathcal{F}$ be parametrized by the pair $(c, \boldsymbol{w}) \in \mathbb{R} \times \mathbb{R}^d$. The score for $\mathcal{C}$ on a ligand $\boldsymbol{x} \in \mathbb{R}^d$ is given by $\mathcal{C}(\boldsymbol{x}) := c + \boldsymbol{w}^T \boldsymbol{x}$. We use the loss $\ell(\mathcal{C}(\boldsymbol{x}), y) := \max\{0, 1 - y \cdot \mathcal{C}(\boldsymbol{x})\}$ for $y \in \{+1, -1\}$. These choices offer two main benefits. First, we can learn from all features without making the model too complex and unstable. This is important for robustness, as the data is both small and imbalanced in our problem. Also, we can calculate the expected-loss term in Equation 1 in closed form, *without the need for sampling*.

Theorem 3.1. *Let $s_i := 1 - (c + \boldsymbol{w}^T \boldsymbol{x}_i)$ with $\|\boldsymbol{w}\| > 0$, and let $\Phi(\cdot)$ and $\phi(\cdot)$ represent the cdf and pdf of a $\mathcal{N}(0, 1)$ distribution. Then, we have:*

$$E_{\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}_i, \sigma^2 I)} \left[\ell(C(\boldsymbol{x}), y = 1) \right] = s_i \cdot \Phi\left(\frac{s_i}{\sigma \|\boldsymbol{w}\|} \right) + \sigma \|\boldsymbol{w}\| \cdot \phi\left(\frac{s_i}{\sigma \|\boldsymbol{w}\|} \right).$$

The proof is shown in Appendix A. Using Theorem 3.1, the loss function in Equation 1 has a closed-form formula. Furthermore, the objective is convex since it is the expectation over a convex loss. Hence, we can minimize Equation 1 using any standard convex solver.

Remark 3.2. In Theorem 3.1, the Gaussian serves as a tractable regularizer that yields a closed-form expected loss. We do not sample from the high-dimensional Gaussian; thus, we avoid approximation errors. Also, the simplicity of $\mathcal{C}(\cdot)$ makes scoring ligands extremely fast.

3.3 Combining Classifiers for Multiple Top Ligands

Suppose we have trained the classifiers $\mathcal{C}_i$ for all $i \in \mathcal{S}^+$. Now, we score each untested ligand (the set $\mathcal{D}_{rest}$) via

$$\text{score}(\boldsymbol{x}_j) := \sum_{i \in \mathcal{S}^+} \alpha^{p_i} \cdot \mathcal{C}_i(\boldsymbol{x}_j), \tag{2}$$

where p_i is the PIC of ligand i, and $\alpha \geq 1$ is a model parameter. In other words, ligand $j \in \mathcal{D}_{rest}$ is predicted to have a high PIC if it is scored highly by one or more of the classifiers $\mathcal{C}_i$ for $i \in \mathcal{S}^+$. Intuitively, the classifiers identify patterns that differentiate the top ligands from the rest. Hence, the

Algorithm 1 SPADE

1: **function** SPADE($\mathcal{D}_{seen}, \mathcal{D}_{rest}, \sigma, n_{max}, \alpha, \beta, p^+$)
▷ $\mathcal{D}_{seen} = \{(\boldsymbol{x}_i, p_i)\}$ sorted in decreasing order of PICs p_i
2: $m \leftarrow |\{p_i \geq p^+ \mid (\boldsymbol{x}_i, p_i) \in \mathcal{D}_{seen}\}|$
3: $n^+ \leftarrow \min(n_{max}, m)$
4: $\mathcal{S}^- \leftarrow \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, p_i) \in \mathcal{D}_{seen}, i > \max(\lceil \beta |\mathcal{D}_{seen}| \rceil, m)\}$
5: **for all** $i \in \{1 \dots n^+\}$ **do**
6: $\mathcal{C}_i \leftarrow$ Classify($\{\boldsymbol{x}_i\}$ versus $\mathcal{S}^- \mid \sigma$) ▷ (Eq. 1)
7: **end for**
8: $T \leftarrow \left\{ \sum_{i \leq n^+} \alpha^{p_i} \cdot \mathcal{C}_i(\boldsymbol{x}_j) \mid \boldsymbol{x}_j \in \mathcal{D}_{rest} \right\}$ ▷ (Eq. 2)
9: **return** top scoring ligands from T
10: **end function**

weighted sum selects ligands that possess several of these differentiating factors. The weights α^{p_i} ensure that better ligands in $\mathcal{S}^+$ are given more importance. The ligands in $\mathcal{D}_{rest}$ with the highest scores are selected for testing in the next DMTA cycle.

3.4 Overall Algorithm

Algorithm 1 shows how SPADE selects ligands for one DMTA cycle. We have a set $\mathcal{D}_{seen}$ of previously tested ligands (whose PICs are known) and the untested set $\mathcal{D}_{rest}$. From $\mathcal{D}_{seen}$, we select the top ligands with PIC$\geq p^+$, keeping at most n_{max} of these for the positive class. For the negative class, we keep all ligands except those with PIC above p^+ or those among the top-β fraction of ligands. Then, we train the classifiers $\mathcal{C}_i$ ($i \in \{1, \dots, n^+\}$) in step 6, and score the ligands in $\mathcal{D}_{rest}$ in step 8. The top-scoring ligands from $\mathcal{D}_{rest}$ are then selected for testing in the next DMTA cycle. Once lab tests reveal their PICs, we add them to $\mathcal{D}_{seen}$. These become inputs for the next DMTA cycle. Our novel combination of Gaussian regularization for robustness and the simple form of the classifier and scoring function enables SPADE to work even with limited data while scoring candidate ligands $10x$ faster than competing methods.

Implementation details: We standardize the scores output by each $\mathcal{C}_i$ for all ligands in $\mathcal{D}_{rest}$ to zero mean and unit variance (step 6). Finally, we noticed that if one ligand with a very high PIC is found in the early cycles, it can dictate the choice of ligands for several future cycles. To improve reliability, we limit each ligand $i \in \mathcal{D}_{seen}$ to help select at most 10 other ligands across all cycles. In our experiments, we use $\alpha = 5, \sigma = 1, \beta = 0.05, n_{max} = 20$, and $p^+ = 7$. In Section 4.4, we show that apart from σ, which controls robustness, our results are insensitive to all hyper-parameters.

4 Experiments

We ran experiments to answer the following questions: (a) How quickly do SPADE and competing methods find enough ligands with the desired PIC? (b) How much computation do they need to score candidate ligands? (c) How robust is SPADE to its model parameters and choice of ligand embedding? (d) How much does each component of SPADE contribute to its performance?

4.1 Experimental Setup

We constructed a new 1.5M-entry ligand–protein interaction (LPI) dataset from PubChem, yielding $\sim$3.5M datapoints when combined with BindingDB and Davis datasets [4, 8]. We excluded compounds with PIC < 5 across all proteins, as they provide little signal for learning. Consequently, our dataset is not directly comparable to high-throughput screening (HTS) hit rates, which include many uniformly inactive compounds and are designed for single-shot screening. In contrast, iterative screening methods like SPADE achieve higher hit rates [29]. Our curation provides a more informative and balanced benchmark for fair comparison across algorithms.

We represented ligands as $2,048$-dimensional ECFP, 167-dimensional MACCS, and 600-dimensional ChemBERTa embeddings [3, 5, 34]. We use ECFP as our primary molecular representation throughout the main experiments, since it performs best on downstream property-prediction tasks [32].

Method	Average (Top-10) Target PIC 7.0	7.5	8.0	8.5	9.0	Min (Top-3) Target PIC 7.0	7.5	8.0	8.5	9.0
Random	21	35	76	215	397	13	21	46	140	348
XGBoost	21	33	55	104	249	13	22	39	82	186
MLP	**17**	25	45	108	288	**12**	**18**	31	78	199
XGBRegressor	18	27	46	103	265	**12**	20	34	78	201
TabPFN	18	26	48	122	276	**12**	19	35	94	217
TabM	**17**	27	59	141	294	**12**	20	41	115	249
GP-M	**17**	27	53	136	290	**12**	21	41	112	235
GP-UCB	**17**	28	51	126	277	**12**	22	38	99	217
GP-EI	18	27	44	**92**	**228**	13	20	33	**70**	176
GP-PI	**17**	25	41	99	251	**12**	19	33	80	184
SPADE	**17**	**24**	**40**	**92**	257	**12**	**18**	**29**	71	**168**

Table 1: **Median value of mean-ligands-to-target (MLT) over** 100 **proteins (lower is better):** SPADE requires the fewest or nearly the fewest ligand tests to reach the target PICs in most cases.

We simulated early-stage drug discovery on 100 proteins with the most data. For each protein, we ran the following experiment. Let A be all the ligands associated with that protein. In other words, our dataset contains the PICs for all ligands in A. We keep these PICs hidden from the algorithm until it specifically tests for ligands. In the first iteration, SPADE randomly selects b ligands from A. The PICs of these selected ligands are revealed to SPADE. With this information, SPADE selects another b ligands (not at random). These are tested in the next iteration, and so on. In this way, each iteration represents a simulated DMTA cycle consisting of b ligand tests (we use $b = 10$, but other values yield similar results).

The iterations end when the top-10 ligands found so far have an average PIC $\geq t$ for $t \in \{7.5, 8, 8.5, 9\}$. Alternatively, we stop once the top-3 ligands each have PIC $\geq t$. We call these two endpoints *average top-10* and *min top-3* respectively. We note that ligands with PIC≥ 9.5 are too rare ($< 0.07\%$ of the ligands) to get reliable results.

Metrics: For each experiment, we calculated the **ligands-to-target**(t), which denotes the number of ligand tests needed to hit an endpoint at PIC t. If the endpoints are not reached within 400 ligand tests, we declare failure and set the ligands-to-target$= 400$. To remove the effect of the random initialization in the first iteration, we averaged this value over 50 repetitions. We call the result mean-ligands-to-target, or MLT(t). Smaller values of MLT(t) imply faster drug discovery.

Competing methods: We consider the following types of competing methods.

(a) Bayesian optimization: We use a Gaussian Process (GP) with a Tanimoto kernel, which has been found to work best for molecules [11]. The GP gives mean and variance estimates for our current belief about each ligand's PIC. In each DMTA cycle, we select ligands with the highest mean (GP-M), or mean plus standard deviation (GP-UCB), or the expected improvement in PIC (GP-EI), or the probability of improvement (GP-PI).

(b) Deep learning: Since we have limited training data, we consider two recent state-of-the-art methods for such settings, namely TabM [10] and TabPFN [12].

(c) Standard tools: We also tested XGBoost, Multilayer Perceptrons (MLP), and XGBRegressor. As a control, we added a method, named Random, that randomly selects ligands in each DMTA cycle.

4.2 Comparisons Between Methods

Overall comparison of mean ligands-to-target (MLT): Table 1 shows, for each method, the median MLT over 100 proteins. **SPADE is the fastest, or nearly the fastest, to the target PIC in almost all cases.** SPADE needs only $\mathbf{29 - 40}$ ligand tests to reach a PIC of $\mathbf{8}$, for both the *average top-10* and *min top-3* endpoints. For a target PIC of 8.5, we only need $71 - 92$ ligand tests.

Head-to-head comparisons: For each protein, we tracked which method reached the endpoint first over the 50 trials. If two methods are similar, each should finish first about equally often. A method

How often is a method significantly better?	Average (Top-10) Target PIC 7.0	7.5	8.0	8.5	9.0	Min (Top-3) Target PIC 7.0	7.5	8.0	8.5	9.0
SPADE	**95%**	**94%**	**91%**	**68%**	**24%**	**20%**	**61%**	**80%**	**75%**	**34%**
vs. XGBoost	1%	1%	2%	1%	13%	0%	1%	0%	1%	2%
SPADE	**18%**	**33%**	**60%**	**54%**	41%	0%	3%	**7%**	**27%**	**33%**
vs. MLP	3%	5%	6%	6%	15%	0%	3%	2%	5%	10%
SPADE	**78%**	**79%**	**69%**	**43%**	**28%**	**2%**	**18%**	**36%**	**28%**	**24%**
vs. XGBRegressor	1%	0%	2%	4%	10%	0%	0%	1%	2%	7%
SPADE	**23%**	**43%**	**46%**	**56%**	**56%**	**3%**	**6%**	**10%**	**30%**	**51%**
vs TabM	4%	2%	2%	2%	4%	1%	2%	1%	2%	1%
SPADE	**9%**	**16%**	**17%**	**44%**	**43%**	**2%**	**7%**	**8%**	**24%**	**34%**
vs. GP-M	8%	5%	3%	6%	6%	1%	1%	1%	2%	3%
SPADE	6%	**12%**	**13%**	**26%**	**32%**	1%	**5%**	**6%**	**13%**	**22%**
vs GP-UCB	**8%**	8%	11%	8%	9%	**2%**	1%	1%	2%	4%
SPADE	**54%**	**59%**	**42%**	**23%**	15%	2%	**10%**	**16%**	**17%**	11%
vs GP-EI	2%	3%	7%	8%	**22%**	2%	3%	2%	5%	**15%**
SPADE	2%	**9%**	**13%**	**18%**	21%	1%	**2%**	**6%**	**13%**	13%
vs GP-PI	**9%**	7%	9%	9%	**23%**	**2%**	1%	3%	2%	13%
SPADE (MACCS)	**8%**	**9%**	**12%**	**14%**	**24%**	1%	2%	2%	**8%**	**18%**
vs TabPFN (MACCS)	5%	5%	8%	9%	4%	1%	2%	2%	4%	0%
SPADE (ECFP)	**42%**	**46%**	**49%**	**45%**	**43%**	1%	**8%**	**11%**	**28%**	**36%**
vs TabPFN (MACCS)	2%	1%	1%	3%	1%	1%	1%	1%	3%	1%

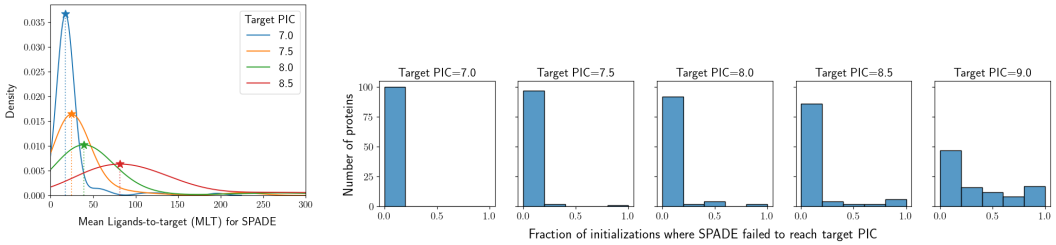
Table 2: **Head-to-head comparisons:** We report the percentage of proteins for which one method reaches the endpoint earlier than the other method, significantly more often than by chance. TabPFN runs only with MACCS, not ECFP, so we compare against SPADE run with both embeddings. For our main target PICs of 8 and 8.5, SPADE is reliably better for more proteins than any other method.

Median improvement of SPADE over	Average (Top-10) Target PIC 7.0	7.5	8.0	8.5	9.0	Min (Top-3) Target PIC 7.0	7.5	8.0	8.5	9.0
XGBoost	18%	24%	25%	16%	3%	22%	21%	23%	16%	14%
MLP	6%	10%	18%	22%	12%	0%	8%	17%	18%	22%
XGBRegressor	7%	11%	13%	18%	11%	10%	19%	13%	20%	24%
TabPFN	4%	13%	7%	12%	8%	-9%	-15%	-19%	6%	12%
TabM	4%	17%	32%	32%	19%	42%	43%	39%	40%	27%
GP-M	2%	26%	20%	31%	12%	45%	45%	19%	38%	28%
GP-UCB	0%	14%	16%	24%	11%	-8%	44%	17%	28%	26%
GP-EI	11%	13%	8%	13%	-7%	-14%	14%	14%	12%	-1%
GP-PI	-5%	12%	8%	17%	-2%	-59%	43%	22%	19%	4%

Table 3: **Percent lift of SPADE over competing methods (higher is better):** For proteins where one method is reliably better, we compute the improvement in SPADE's MLT over the competing method. SPADE has a median improvement of $8\% - 32\%$ over competing methods for our primary target PICs of 8 and 8.5 under the *average top-10* metric.

is reliably better for a protein if it finishes first significantly more than half the time ($p < 0.1$; results are similar for $p < 0.05$). Table 2 shows the fraction of proteins where SPADE or a competitor is reliably better. **SPADE beats every competitor for the main target PICs of 8 and 8.5.** GP-PI is the closest competitor, but SPADE is 10x faster (as shown later).

Table 3 compares the MLT for proteins where one method is reliably better than another. For the *average top-10* metric with target PICs 8 or 8.5, SPADE outperforms Bayesian optimization by $8\% - 31\%$, deep learning by $7\% - 32\%$, and standard classifiers and regressors by $13\% - 25\%$. Thus, **SPADE needs $7\% - 32\%$ fewer ligand tests than its competitors.**



(a) SPADE's MLT distribution (b) Out-of-reach target PICs

Figure 1: **Detailed analysis of SPADE's performance:** (a) As the target PIC increases, the distribution of SPADE's mean ligands-to-target (MLT) shifts to the right and has higher variance. (b) SPADE's failures to reach a PIC occur most for target PIC$= 9$, which are very rare (less than 0.5% of the ligand for the median protein). Detailed explanations are in the text.

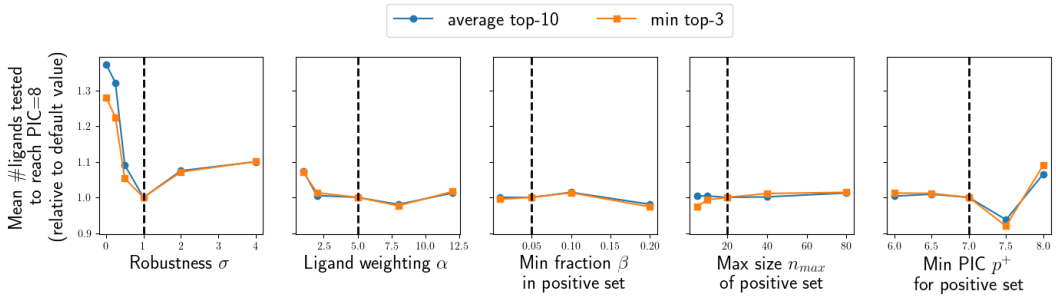


Figure 2: **Sensitivity analysis:** SPADE's performance is only sensitive to the robustness parameter σ used in Equation 1.

4.3 Detailed Analysis of SPADE

Fast ligand discovery, with wider tails for harder targets: Figure 1a shows that SPADE's MLT distribution shifts to the right as the target PIC increases, since we need more ligand tests to achieve the target. The variance also increases for the higher PICs, due to correlated ligand selection across the DMTA cycles. The effect of such correlations compounds as the number of DMTA cycles increases. Since we need more cycles for higher target PICs, we see higher variance for them.

SPADE only fails when the target PICs are especially rare: For each protein, we count the fraction of times SPADE does not reach the target within 400 ligand tests. Figure 1b shows the histogram of this fraction across all proteins. Almost all observed failures occur when the target PIC is 9, which is extremely rare in our data ($< 0.5\%$ of ligands). For the target PIC of 8.5, failures only occur for proteins where fewer than 0.1% of the ligands have PIC≥ 8.5 (the typical rate is 2.7%). GP-EI and GP-PI fail at the same rate as SPADE, and GP-UCB, GP-M, and TabM fail at about 45% higher rate.

Wall-clock time: The main bottleneck is scoring millions of untested ligands to select the best ones for the next DMTA cycle. Table 4a shows the wall-clock times for scoring 1,000 ligands. **SPADE is 10x faster** than its closest competitor (GP-PI).

4.4 Sensitivity Analysis and Ablation Study

Ligand embedding: We repeated our experiments using the MACCS and ChemBERTa ligand embeddings. Both are lower-dimensional than the ECFP embedding. Table 5 compares SPADE against GP-PI, its closest competitor, under the ChemBERTa and MACCS embeddings. At our primary target PIC of 8, SPADE is reliably better than GP-PI on roughly twice as many proteins under both embeddings. (17% vs. 8% for ChemBERTa; 16% vs. 8% for MACCS). At PIC 8.5, SPADE is also comparable or better. Thus, SPADE consistently outperforms other methods in the race-to-8, for all embeddings.

Model hyper-parameters: Figure 2 shows how SPADE's average MLT changes as we vary its hyperparameters. All results are for our primary target PIC of 8, and are normalized with respect to the default hyperparameter settings. Only the robustness parameter σ significantly affects performance

Method	Time (s)
SPADE	$\mathbf{4 \pm 0.1}$
GP-(any)	39 ± 0.2
TabM	97 ± 8.6
TabPFN	6096 ± 548

(a) Time to score 10^6 ligands

# ligand tests in a DMTA cycle (b)	Relative diff. from default
$b = 5$	$0.3\% \pm 1.05\%$
$b = 10$	(default)
$b = 20$	$0.4\% \pm 1.8\%$

(b) Sensitivity to batch size

Condition	Effect
No robustness	37% worse MLT
No exp. weight	10% worse MLT
No 10-ligand limit	PIC diff. < 0.04

(c) Ablation study

Table 4: **Summary of additional analyses for SPADE.**

	ChemBERTa (600 dim.) Target PIC					MACCS (167 dim.) Target PIC				
Method	7.0	7.5	8.0	8.5	9.0	7.0	7.5	8.0	8.5	9.0
SPADE is better	11%	16%	**17%**	**20%**	14%	**10%**	**10%**	**16%**	13%	14%
GP-PI is better	2%	5%	8%	15%	**29%**	3%	6%	8%	**16%**	**26%**

Table 5: **SPADE vs. GP-PI under ChemBERTa and MACCS.** Percentage of proteins where each method is significantly better ($p < 0.1$). SPADE dominates at lower-to-mid target PICs and is comparable or better at PIC 8.5.

(Equation 1). We note that we use the same default value of $\sigma = 1$ for all embeddings (ECFP, MACCS, and ChemBERTa) and all proteins, without per-protein retuning. SPADE *does not need cross-validation.* Since we need fewer than 40 total tests before reaching target PIC=8, any cross-validation based selection strategy would be dominated by sampling noise in this low-data setting.

Number of ligand tests in each DMTA cycle: Table 4b shows the relative difference in SPADE's performance as we vary the number of ligands b tested per DMTA cycle. The average top-10 PIC found by SPADE varies only slightly with b, confirming that the method is robust to this hyperparameter. Values are mean $\pm$ standard deviation of the relative difference vs. b=10.

Ablation Study: We removed three aspects of SPADE: its robustness, the exponential weighting scheme, and the condition that any ligand can be used to help select at most 10 other ligands. These correspond to Steps 6 and 8 of Algorithm 1, and the implementation details in Section 3. Table 4c shows that the first two aspects are important to SPADE's performance, while the last one is less important.

Filtering ligands using data for similar proteins: Finally, we simulated an experiment where we know of proteins similar to the target protein. Here, the data from similar proteins is used to filter the set of available ligands before running SPADE. This improves MLT by $18\% - 31\%$ (Table 6 in Appendix B).

5 Conclusions

We set out to rapidly identify ligands with PIC≥ 8 for a specific target protein. These high-quality ligands, however, are the proverbial needles in the vast haystack of all possible ligands. We also assume no prior knowledge about the protein. Remarkably, just 40 or so ligand tests suffice for SPADE to discover 10 high-quality ligands in this challenging setting. SPADE needs to test $7\% - 32\%$ fewer ligands than its competitors. Moreover, it is also three times faster at scoring ligands than its closest competitor. This translates to significant cost savings for early-stage drug discovery.

SPADE succeeds for two reasons. First, it focuses only on improving the top-k ligands seen so far. It does not estimate the PIC distribution across all ligands. Second, SPADE is designed to be robust. Specifically, in optimizing the model parameters, we minimize the expected loss over a broad distribution rather than the empirical loss on just k top ligands. This approach helps SPADE learn reliable signals from such limited and imbalanced data.

References

[1] Deepayan Chakrabarti and Benjamin Fauber. Robust High-Dimensional Classification From Few Positive Examples. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 1952–1958, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3. doi: 10.24963/ijcai.2022/271. URL https://www.ijcai.org/proceedings/2022/271.

[2] Ayan Chatterjee, Robin Walters, Zohair Shafi, Omair Shafi Ahmed, Michael Sebek, Deisy Morselli Gysi, Rose Yu, Tina Eliassi-Rad, Albert-László Barabási, and Giulia Menichetti. Improving the Generalizability of Protein-Ligand Binding Predictions with AI-Bind. *Nat. Commun.*, 14:1989, 2023.

[3] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. ChemBERTa: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[4] Mindy I. Davis, Jeremy P Hunt, Sanna Herrgård, Pietro Ciceri, Lisa M. Wodicka, Gabriel Pallares, Michael Hocker, Daniel K. Treiber, and Patrick P. Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, 29:1046–1051, 2011. URL https://api.semanticscholar.org/CorpusID:32070305.

[5] Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. Reoptimization of MDL Keys for Use in Drug Discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280, November 2002. ISSN 0095-2338. doi: 10.1021/ci010132r. URL https://doi.org/10.1021/ci010132r. Publisher: American Chemical Society.

[6] Ben Fauber. Accurate Prediction of Ligand-Protein Interaction Affinities with Fine-Tuned Small Language Models. *ArXiv*, abs/2407.00111v1, 2024.

[7] Jean-Loup Faulon, Milind Misra, Shawn Martin, Ken Sale, and Rajat Sapra. Genome Scale Enzyme–Metabolite and Drug–Target Interaction Predictions Using the Signature Molecular Descriptor. *Bioinformatics*, 24(2):225–233, 2007. ISSN 1367-4803.

[8] Michael K. Gilson, Tiqing Liu, Michael Baitaluk, George Nicola, Linda Hwang, and Jenny Chong. Bindingdb in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Research*, 44:D1045 – D1053, 2015. URL https://api.semanticscholar.org/CorpusID:8843610.

[9] Rohan Gorantla, Alžbeta Kubincová, Andrea Y. Weiße, and Antonia S. J. S. Mey. From Proteins to Ligands: Decoding Deep Learning Methods for Binding Affinity Prediction. *J. Chem. Inf. Model.*, 64(7):2496–2507, 2024.

[10] Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TABM: Advancing Tabular Deep Learning With Parameter-Efficient Ensembling. 2025.

[11] Ryan-Rhys Griffiths, Leo Klarner, Henry Moss, Aditya Ravuri, Sang Truong, Yuanqi Du, Samuel Stanton, Gary Tom, Bojana Rankovic, Arian Jamasb, et al. GAUCHE: A library for Gaussian processes in chemistry. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 01 2025. doi: 10.1038/s41586-024-08328-6. URL https://www.nature.com/articles/s41586-024-08328-6.

[13] Kexin Huang, Tianfan Fu, Lucas Glass, Marinka Zitnik, Cao Xiao, and Jimeng Sun. DeepPurpose: A Deep Learning Library for Drug–Target Interaction Prediction. *Bioinformatics*, 36: 5545 – 5547, 2020.

[14] Kexin Huang, Cao Xiao, Lucas Glass, and Jimeng Sun. MolTrans: Molecular Interaction Transformer for Drug–Target Interaction Prediction. *Bioinformatics*, 37:830 – 836, 2020.

[15] Yogesh Kalakoti, Shashank Yadav, and Durai Sundar. TransDTI: Transformer-Based Language Models for Estimating DTIs and Building a Drug Recommendation Workflow. *ACS Omega*, 7: 2706 – 2717, 2022.

[16] Joshua D Kangas, Armaghan W Naik, and Robert F Murphy. Efficient discovery of responses of proteins to compounds using active learning. *BMC Bioinformatics*, 15(1), December 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-143. URL `https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-15-143`. Publisher: Springer Science and Business Media LLC.

[17] Talia B. Kimber, Yonghui Chen, and Andrea Volkamer. Deep Learning in Virtual Screening: Recent Applications and Developments. *Int. J. Mol. Sci.*, 22:4435, 2021.

[18] Jonathan Knowles and Gianni Gromo. Target Selection in Drug Discovery. *Nat. Rev. Drug Discov.*, 2:63–69, 2003.

[19] Ingoo Lee, Jongsoo Keum, and Hojung Nam. DeepConv-DTI: Prediction of Drug-Target Interactions via Deep Learning with Convolution on Protein sequences. *PLOS Comput. Biol.*, 15(6):e1007129, 06 2019.

[20] Eelke B. Lenselink, Niels ten Dijke, Brandon Bongers, George Papadatos, Herman W. T. van Vlijmen, Wojtek Kowalczyk, Adriaan P. IJzerman, and Gerard J. P. van Westen. Beyond the Hype: Deep Neural Networks Outperform Established Methods Using a ChEMBL Bioactivity Benchmark Set. *J. Cheminform.*, 9:45, 2017.

[21] Shuya Li, Fangping Wan, Hantao Shu, Tao Jiang, Dan Zhao, and Jianyang Zeng. MONN: A Multi-objective Neural Network for Predicting Compound-Protein Interactions and Affinities. *Cell Syst.*, pages 308–322.e11, 2020.

[22] Joseph G. Lombardino and John A. Lowe III. The Role of the Medicinal Chemist in Drug Discovery — Then and Now. *Nat. Rev. Drug Discov.*, 3:853–862, 2004.

[23] Eric J. Martin, Prasenjit Mukherjee, David C. Sullivan, and Johanna M. Jansen. Profile-QSAR: A Novel meta-QSAR Method that Combines Activities across the Kinase Family To Accurately Predict Affinity, Selectivity, and Cellular Activity. *J. Chem. Inf. Model.*, 51(8):1942–1956, 2011.

[24] Eric J. Martin, Valery R. Polyakov, Xiang-Wei Zhu, Li Tian, Prasenjit Mukherjee, and Xin Liu. All-Assay-Max2 pQSAR: Activity Predictions as Accurate as Four-Concentration IC50s for 8558 Novartis Assays. *J. Chem. Inf. Model.*, 59(10):4450–4459, 2019.

[25] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin N. Steijaert, Jörg Kurt Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-Scale Comparison of Machine Learning Methods for Drug Target Prediction on ChEMBL. *Chem. Sci.*, 9:5441–5451, 2018.

[26] Michael M Mysinger, Michael Carchia, John J Irwin, and Brian K Shoichet. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594, 2012.

[27] Pedro F. Oliveira, Rita C Guedes, and Andre O Falcao. Inferring Molecular Inhibition Potency with AlphaFold Predicted Structures. *Sci. Rep.*, 14:8252, 2024.

[28] Hakime Öztürk, Elif Ozkirimli Olmez, and Arzucan Özgür. WideDTA: Prediction of Drug-Target Binding Affinity. *ArXiv*, abs/1902.04166, 2019.

[29] Shardul Paricharak, Adriaan P. IJzerman, Andreas Bender, and Florian Nigsch. Analysis of Iterative Screening with Stepwise Compound Selection Based on Novartis In-house HTS Data. *ACS Chemical Biology*, 11(5):1255–1264, May 2016. ISSN 1554-8929, 1554-8937. doi: 10.1021/acschembio.6b00029. URL `https://pubs.acs.org/doi/10.1021/acschembio.6b00029`.

[30] 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. doi: 10.1101/2025.06.14.659707.

[31] Alleyn T. Plowright, Craig Johnstone, Jan Kihlberg, Jonas Pettersson, Graeme Robb, and Richard A. Thompson. Hypothesis Driven Drug Design: Improving Quality and Effectiveness of the Design-Make-Test-Analyse Cycle. *Drug Discov. Today*, 17(1):56–62, 2012.

[32] Mateusz Praski, Jakub Adamczyk, and Wojciech Czech. Benchmarking pretrained molecular embedding models for molecular representation learning. *arXiv preprint arXiv:2508.06199*, 2025.

[33] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019. doi: 10.1101/622803. URL `https://www.biorxiv.org/content/10.1101/622803v4`.

[34] David Rogers and Mathew Hahn. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, May 2010. ISSN 1549-9596. doi: 10.1021/ci100050t. URL `https://doi.org/10.1021/ci100050t`. Publisher: American Chemical Society.

[35] 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. *Commun. Chem.*, 6:222, 2023.

[36] Arman A. Sadybekov, Anastasiia V. Sadybekov, Yongfeng Liu, Christos Iliopoulos-Tsoutsouvas, Xi-Ping Huang, Julie E. Pickett, Blake Houser, Nilkanth Patel, Ngan K. Tran, Fei Tong, Nikolai Zvonok, M. K. Jain, Olena V. Savych, Dmytro S. Radchenko, Spyros P. Nikas, Nicos A. Petasis, Yurii S. Moroz, Bryan L. Roth, Alexandros Makriyannis, and Vsevolod Katritch. Synthon-Based Ligand Discovery in Virtual Libraries of over 11 Billion Compounds. *Nature*, 601:452 – 459, 2021.

[37] Christina E. M. Schindler, Hannah Baumann, Andreas Blum, Dietrich Böse, Hans-Peter Buchstaller, Lars Burgdorf, Daniel Cappel, Eugene Chekler, Paul Czodrowski, Dieter Dorsch, Merveille K. I. Eguida, Bruce Follows, Thomas Fuchß, Ulrich Grädler, Jakub Gunera, Theresa Johnson, Lebrun Catherine Jorand, Srinivasa Karra, Markus Klein, Tim Knehans, Lisa Koetzner, Mireille Krier, Matthias Leiendecker, Birgitta Leuthner, Liwei Li, Igor Mochalkin, Djordje Musil, Constantin Neagu, Friedrich Rippmann, Kai Schiemann, Robert Schulz, Thomas Steinbrecher, Eva-Maria Tanzer, Andrea Unzue Lopez, Follis Ariele Viacava, Ansgar Wegener, and Daniel Kuhn. Large-Scale Assessment of Binding Free Energy Calculations in Active Drug Discovery Projects. *J. Chem. Inf. Model.*, 60(11):5457–5474, 2020.

[38] Reed M. Stein, Ying Yang, Trent E. Balius, Matt J. O'Meara, Jiankun Lyu, Jennifer Young, Khanh Tang, Brian K. Shoichet, and John J. Irwin. Property-Unmatched Decoys in Docking Benchmarks. *Journal of Chemical Information and Modeling*, 61(2):699–714, February 2021. ISSN 1549-9596, 1549-960X. doi: 10.1021/acs.jcim.0c00598. URL `https://pubs.acs.org/doi/10.1021/acs.jcim.0c00598`.

[39] Emma Svensson, Pieter-Jan Hoedt, Sepp Hochreiter, and Günter Klambauer. HyperPCM: Robust Task-Conditioned Modeling of Drug–Target Interactions. *J. Chem. Inf. Model.*, 64:2539 – 2553, 2024.

[40] David C. Swinney and Jason Anthony. How Were New Medicines Discovered? *Nat. Rev. Drug Discov.*, 10:507–519, 2011.

[41] Maha A. Thafar, Mona Alshahrani, Somayah Albaradei, Takashi Gojobori, Magbubah Essack, and Xin Gao. Affinity2Vec: Drug-Target Binding Affinity Prediction Through Representation Learning, Graph Mining, and Machine Learning. *Sci. Rep.*, 12:4751, 2022.

[42] Viet-Khoa Tran-Nguyen, Célien Jacquemard, and Didier Rognan. LIT-PCBA: an unbiased data set for machine learning and virtual screening. *Journal of chemical information and modeling*, 60(9):4263–4273, 2020.

[43] Lingle Wang, Yujie Wu, Yuqing Deng, Byungchan Kim, Levi Pierce, Goran Krilov, Dmitry Lupyan, Shaughnessy Robinson, Markus K. Dahlgren, Jeremy R. Greenwood, Donna Lee Romero, Craig E. Masse, Jennifer L. Knight, Thomas Steinbrecher, Thijs Beuming, Wolfgang Damm, Edward D Harder, Woody Sherman, Mark L. Brewer, Ron Wester, Mark A. Murcko, Leah L. 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. *J. Am. Chem. Soc.*, 137:2695–2703, 2015.

[44] Michael J. Waring, John Edmund Arrowsmith, Andrew R. Leach, Paul D. Leeson, Sam Mandrell, Robert M. Owen, Garry Pairaudeau, William D. Pennie, Stephen D. Pickett, Jibo Wang, Owen Wallace, and Alexander Weir. An Analysis of the Attrition of Drug Candidates from Four Major Pharmaceutical Companies. *Nat. Rev. Drug Discov.*, 14:475–486, 2015.

[45] Ming Wen, Zhimin Zhang, Shaoyu Niu, Haozhi Sha, Rui Yang, Yong-Huan Yun, and Hongmei Lu. Deep-Learning-Based Drug-Target Interaction Prediction. *J. Proteome Res.*, 16:1401–1409, 2017.

[46] Thomas M. Whitehead, Benedict W J Irwin, Peter A. Hunt, Matthew D. Segall, and Gareth John Conduit. Imputation of Assay Bioactivity Data Using Deep Learning. *J. Chem. Inf. Model.*, 59: 1197–1204, 2019.

[47] Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. Prediction of Drug–Target Interaction Networks from the Integration of Chemical and Genomic Spaces. *Bioinformatics*, 24:i232 – i240, 2008.

A Proofs

Proof of Theorem 3.1. The proof is similar to Theorem 1 of [1]. We have

$$\begin{aligned}
&E_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{x}_i,\sigma^2 I)}\left[\ell(C(\boldsymbol{x}),y=1)\right] \\
&\quad = E_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{x}_i,\sigma^2 I)} \max\left(0, 1-(c+\boldsymbol{w}^T\boldsymbol{x})\right) \\
&\quad = E_{\boldsymbol{v}\sim\mathcal{N}(1-(c+\boldsymbol{w}^T\boldsymbol{x}_i),\sigma^2\boldsymbol{w}^T\boldsymbol{w})} \max\left(0,\boldsymbol{v}\right) \\
&\quad = s_i\cdot\Phi\left(\frac{s_i}{\sigma\|\boldsymbol{w}\|}\right)+\sigma\|\boldsymbol{w}\|\cdot\phi\left(\frac{s_i}{\sigma\|\boldsymbol{w}\|}\right),
\end{aligned}$$

where $s_i = 1-(c+\boldsymbol{w}^T\boldsymbol{x}_i)$. The first equality uses the form of the loss function. The second equality follows from a change of variables. The third equality comes from the formula for expectations of truncated normals. □

B Experimental Details

Hyperparameters: In each DMTA cycle, we demeaned the ligand embedding vectors, but did not scale them. For the classification-based methods (XGBoost and MLP), we set the top 10% of the ligands in $\mathcal{D}_{seen}$ as positive. For XGBoost and XGBRegressor, we set the trees to have a maximum depth of 6 and an L_2 regularization of 1. For the MLP, we used one hidden layer with 32 units. For all methods (including SPADE), the best settings were chosen by optimizing the *average top-10* metric for a target PIC of 8 for < 10 proteins. We also observed that the metrics were not very sensitive to choice of parameters. Hence, we believe that the results reflect the intrinsic abilities of these methods.

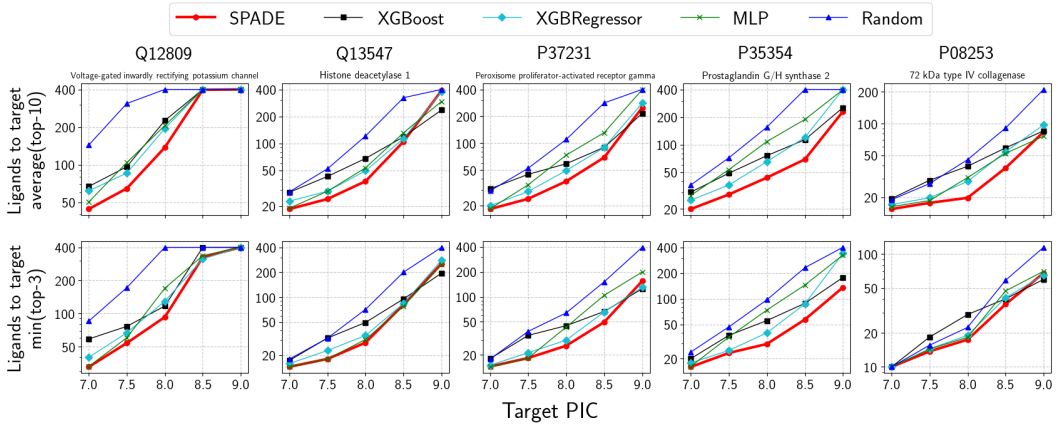


Figure 3: *Ligands-to-target for five example proteins (lower is better):* We show the number of ligand tests needed to reach a target PIC for the *average top-10* metric (top panel) and the *min top-3* metric (bottom panel). The UniProt IDs and names of the proteins are shown at the top. SPADE (red circles) is almost always the fastest to any target PIC. XGBoost (black squares) tends to be close to Random (blue triangles) initially, but improves later. In contrast, MLP (green crosses) is similar to SPADE initially, but underperforms later. XGBRegressor (cyan diamonds) is in the middle.

Compute Resources: All experiments were run on a single workstation with an Intel Core i9-10980XE CPU (18 cores / 36 threads), 256 GB of RAM, and 2× NVIDIA RTX 3090 GPUs (24 GB each). The 100-protein benchmark is parallel across proteins and across the 50 random initialisations per protein, so we ran these as independent processes. Wall-clock scoring times reported in Table 4a are per-process (not summed across parallel workers).

	Median PIC	Mean ligand tests for race to 8	Mean ligand tests for race to 8.5	Mean ligand tests for race to 9
All ligands	6.0	33.1	83.4	273.5
Only top 1000 ligands	6.4	22.9	54.6	231.1
Improvement	**7.1%**	**25.8%**	**31.4%**	**18.4%**

Table 6: SPADE *after filtering using similar proteins:* We trained an XGBoost classifier to predict if a ligand has a PIC above 8. Next, we applied this classifier to rank-order ligands for 10 unseen proteins. For each protein, we selected the top 1,000 ligands from around 16,000 candidates. Then, we ran SPADE on this filtered set of 1,000 ligands. We report the trimmed mean of the median PIC for both full and filtered sets. In addition, we also report the trimmed mean of the mean-ligands-to-target (MLT) for target PICs of 8, 8.5, and 9. Overall, the filtering improves the PIC distribution: the median PIC increases by 7%. This filtering also enhances SPADE's performance for all target PICs, yielding up to a 31% reduction in MLT at PIC 8.5.