Enhancing selectivity using Wasserstein distance based reweighing: Supplementary Materials

Pratik Worah*

Abstract

Given two labeled data-sets $\mathcal{S}$ and $\mathcal{T}$, we design a simple and efficient greedy algorithm to reweigh the loss function such that the limiting distribution of the neural network weights that result from training on $\mathcal{S}$ approaches the limiting distribution that would have resulted by training on $\mathcal{T}$.

On the theoretical side, we prove that when the metric entropy of the input data-sets is bounded, our greedy algorithm outputs a close to optimal reweighing, i.e., the two invariant distributions of network weights will be provably close in total variation distance. Moreover, the algorithm is simple and scalable, and we prove bounds on the efficiency of the algorithm as well.

Our algorithm can deliberately introduce distribution shift to perform (soft) multi-criteria optimization. As a motivating application, we train a neural net to recognize small molecule binders to MNK2 (a MAP Kinase, responsible for cell signaling) which are non-binders to MNK1 (a highly similar protein). We tune the algorithm's parameter so that overall change in holdout loss is negligible, but the selectivity, i.e., the fraction of top 100 MNK2 binders that are MNK1 non-binders, increases from 54% to 95%, as a result of our reweighing. Of the 43 distinct small molecules predicted to be most selective from the enamine catalog, 2 small molecules were experimentally verified to be selective, i.e., they reduced the enzyme activity of MNK2 below 50% but not MNK1, at 10μM – a 5% success rate.

1 Introduction

Deep learning has found applications in diverse areas ranging from organic chemistry to computer generated art. Its applicability is limited by the availability of large amounts of labeled training data. A typical all-to-all neural net of width n (say $n = 10000$) and depth d (say $d = 10$) has $\Theta(n^2 d)$ ($\simeq 10^8$) weight parameters, and although many neural nets work well despite being somewhat over-parameterized, they still need the number of training examples to be of a similar or only slightly smaller order of magnitude.

This reliance on large amounts of training data leads to difficulties in training when we have multiple objectives. For example, suppose we separately gather training data for a neural net to perform classification tasks into a set of classes A using labeled data-set $\mathcal{S}$, and into a set of classes B using labeled data-set $\mathcal{T}$. Then, unless $\mathcal{T}$ and $\mathcal{S}$ have many data-points in common, we will not have enough examples to train a neural net for the classification into classes of $A \times B$. In this paper, we address this problem by designing a scalable algorithm that reweighs dataset $\mathcal{S}$ (using $\mathcal{T}$), so that training on the reweighted $\mathcal{S}$ leads to network that is close to one obtained by training on $\mathcal{T}$. Moreover, we apply it to a drug discovery application and obtain wet-lab verified results.

Our main contribution is theoretical. Algorithm 1 deliberately introduces distribution skew and reweighs the labeled training data-set $\mathcal{S}$ using the data-set $\mathcal{T}$ so that if we train a neural network on the reweighed $\mathcal{S}$ for a long enough period of time then its weights will be "tilted" so that the

*Google Research, pworah@google.com

Preprint. Under review.

classification error on classes B in $\mathcal{T}$ will also be reduced. The amount of reduction is determined by the choice of tilt parameter α in Algorithm 1. Theorems 5.4, 5.13, 5.15, and 5.16 formally show correctness and efficiency of Algorithm 1. In particular, Theorem 5.4 proves correctness and also justifies the choice of Wasserstein metric in Algorithm 1. Theorems 5.13, 5.15 and 5.16 provide efficiency guarantees for Algorithm 1.

It has been known since the 1980s that greedy algorithms, like Algorithm 1, have poor approximation guarantees for computing Wasserstein distance [RT81]. Therefore, we need to assume and exploit some property of about our input instances to get around the lower bound in [RT81]. This critical property turn out to be: input instances must admit a small size covering. This small covering assumption is used in both the random sampling and the greedy reweighing steps of our algorithm. The connection to random sampling is not surprising and it comes from known techniques – the union bound in the large deviations proof. However, the connection between small coverings and greedy matching algorithms (at its core computing Wasserstein distance is equivalent to computing minimum weight matchings) is somewhat surprising, since we are not aware of results obtaining sharper guarantees on approximate minimum weight matching, based on the covering properties of the input data-set.

Organization: In Section 2, we describe an example application to drug discovery that will serve as our running example, whenever needed. More theoretically inclined readers may skip this section and any references to it. In Section 3, we discuss prior work from the areas of machine learning theory, algorithms and computational drug discovery, relevant to our paper. In Section 4, we present Algorithm 1 and provide a technical overview of our paper – how our various theoretical results fit together. Section 5 presents the theorem statements. In particular, Theorem 5.4 explains why the Wasserstein metric is an intuitive and appropriate choice of metric for Algorithm 1; Theorems 5.13, 5.15 and 5.16 show that the greedy random sampling based algorithm can compute the minimum weight bipartite matching, and hence Wasserstein distance, in near linear time; and they provide an upper-bound on the approximation error under our low metric entropy assumption. Thus showing that Algorithm 1 is scalable. Finally, the supplementary material contains proofs, and further figures and results from our drug discovery application.

2 Running example: drug discovery

Multicriteria optimization problems involving data-sets that admit small coverings arise naturally in drug discovery (see Section 4 for details). Therefore, as a concrete motivating example, we illustrate an application of Algorithm 1 to a toy problem in this area, which will also serve as our running example, when needed.

In drug discovery, typically, one wants to isolate small molecules (inhibitors) that bind strongly to a given enzyme, but often we want to exclude small molecules that bind to another similar enzyme. For example, MNK1 and MNK2 are two structurally similar kinases (a kinase is an enzyme for phosphorylation or de-phosphorylation of proteins). We want to identify small molecules that bind strongly to MNK2 (MNK2 hits), but we also prefer that the identified small molecules not bind to MNK1 (MNK1 non-hits). In other words, we want to isolate molecules that are selective for MNK2 over MNK1.

In the in-silico experiments, we were able to increase the percentage of MNK1 non-hits in our set of top predicted MNK2 hits – the selectivity – from 54% to 95% on holdout data, using the reweighing procedure in Algorithm 1. We used a relatively small training set of about 250K small molecules in total; labeled as MNK1 non-binders, and MNK2 binders and non-binders; and a small holdout set of 7K small molecules that consists of molecules which are labeled as: MNK2 hits (binders), and MNK1 hits (binders) or MNK1 non-hits (non-binders). Further details are given in the supplement section. In Figure 1, for the neural network models with and without reweighing, we plot the cumulative number of MNK1 non-hits on the y-axis; and on the x-axis any given point, say k, represents the top k predicted MNK2 hits from the examples in the holdout set. While we can not make our training data-sets and code public for proprietary reasons, we were able to experimentally (in wet-lab) verify that two out of the top fifty (actually 43, since 7 out of 50 molecules could not be synthesized and tested) predicted selective small molecules, obtained by running our neural network model on the Enamine 1.9B molecules catalog (`https://enamine.net`), were indeed selective for MNK2 over

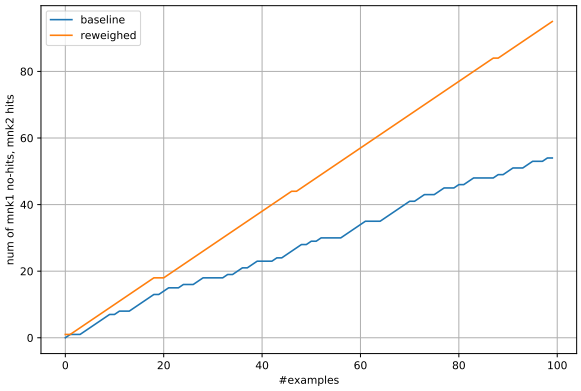
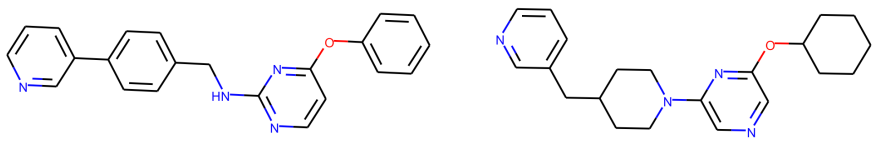


Figure 1: Selectivity of reweighed (using Algorithm 1) and baseline (without reweighing) neural nets. Note that this increase in selectivity from 54% to 95% came without any significant change in the validation loss – the AUC for the classification of MNK2 binders vs non-binders remained around 0.6 in both cases. See supplement for details.

MNK1. That is a success rate of roughly 5% on this admittedly small sample set. We do note that the results are from a single point concentration assay and can be noisy.

We are not aware of other such multi-target prediction results in DNA encoded library (DEL) space (see [SBF+22] for background), where one simultaneously predicts hits/non-hits against two or more proteins. However, the success rates for single target experiments with traditional high-throughput screening is $\sim 1\%$ (see for example the discussion in [MSK+20]) and it is generally accepted that multi-target prediction is a harder problem.

More importantly, the two predicted and tested molecules provide a degree of verification for our experimental application (which has been the motivation, but is not the main result of this paper). The supplement summarizes the results of the experiments and provides further context about them.



C(NC=1N=CC=C(OC=2C=CC=CC2)N1)C=3C=CC(=CC3)C=4C=CC=NC4

C(C1CCN(CC1)C=2C=NC=C(OC3CCCCC3)N2)C=4C=CC=NC4

Figure 2: Two predicted and verified selective MNK1 non-hits and MNK2 hits from the Enamine catalog. The enzyme activity was found to be above 50% for MNK1 but below 50% for MNK2 at 10μM concentration for each of the two small molecules: $\sim 20\%$ vs 70% and $\sim 39\%$ vs 59%. See supplement for all activities. Note that these values are from single point concentration assay and can be noisy.

3 Related work

The question of learning with differing test and train distributions has been well investigated in the machine learning community under different names: ditribution shift and covariate shift, see for

example the book [QCSSL09], the papers [Shi00, RR83, DSP05, BS07, HSG$^+$07] and [CMRR08], to name just a few. The question is also relevant to our paper since our algorithm can be used to reweigh the train data-set to bring the post training neural network weights closer to what they would have been, had we trained based on the test distribution. However, prior results rely on estimating the train and test distributions. For example, parameter estimation of the densities followed by change of variable using the Jacobian. Assuming a logarithmic number of features, that leads to a $\tilde{O}(n^2)$ algorithm for distribution skew correction (n being the training and test data-set size) – much more efficient than Wasserstein distance computation that requires solving a $\Theta(n^2)$ sized linear program. That explains why the Wasserstein distance based ideas are less explored in this context, so far. Distribution shift correction has also found new applications in domain adaptation literature. However, the only prior theoretical works involving Wasserstein distance computation that we found in this area were [CFHR17] and [LDN$^+$21], which focus on exact solution of the Wasserstein distance problem.

In a high dimensional feature space, the density estimation can be inaccurate since the number of samples required increases exponentially in number of features for any formal guarantee (as can be seen from large deviation bounds [DZ10]). Moreover, if we are only interested in partial tilting of one distribution towards another, as in Algorithm 1 (where α controls the amount of tilt), then it is reasonable to look for approximate but efficient computation of Wasserstein distance. That is what we do in this paper using Algorithm 1, which is a $\tilde{O}(n^2)$ time algorithm as well. We now have the added advantage of a provable upper bound on the test-train prediction error, before and after tilting using Algorithm 1, from Theorem 5.4.

The problem of efficient Wasserstein distance computation has also received much attention in the algorithms community. The paper [SA12] studies the equivalence between Wasserstein distance computation and matching algorithms in the metric space setting. Efficient matching algorithms have been well studied in literature for five decades. The optimal algorithm for computing weighted matchings is due to Gabow and Tarjan [GT91] and runs in time $O(m\sqrt{n})$, where m is the number of edges and n the number of vertices in the graph. Since then more sophisticated algorithms have been designed, see for example [Vai89, IK93, SA12] and [ABIW09] to name a few. However, under our assumptions even the simple greedy algorithm performs remarkably well, and it scales efficiently for large training data-sets.

Reingold and Tarjan [RT81] showed that the greedy algorithm has an abysmal approximation ratio of $n^{\log_2 3/2}$ for bipartite graphs. In this paper, we show in Theorem 5.13 that the approximation ratio of the greedy algorithm is much better under our bounded metric entropy assumption than the lower bound in [RT81]. Hence, an assumption about a covering property of the input leads to more optimal matchings – a somewhat surprising algorithmic result that may be of independent interest.

Finally, the idea of using deep learning for drug discovery has gained popularity in pharmaceutical research over the last few years, especially given the amount of data now available [Mul16]. The paper [MSK$^+$20] shows that neural nets can be trained on DNA encoded chemical libraries to identify new small molecules that bind to a given protein target. It is particularly relevant to this work, as we build upon that. Our work extends their work by allowing us to select molecules that bind to one protein target and not to another. Other papers in this rapidly growing area include [ZKL$^+$19, KMB$^+$16] and [GSR$^+$17].

4 Problem statement and overview of results

Suppose we are given two labeled training data sets, say $\mathcal{S}$ and $\mathcal{T}$ for two different classification tasks. Moreover, let's assume that the points in the data-sets are weighted according to two different probability distributions, say $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$ respectively. Our goal is to reweigh $\mathcal{S}$, i.e., "tilt" $\mathbb{P}_{\mathcal{S}}$ towards $\mathbb{P}_{\mathcal{T}}$, and train a neural net classifier so that the limiting distribution of network weights is closer to the one that would be obtained from training using $\mathbb{P}_{\mathcal{T}}$. We assume labels of $\mathcal{S}$ are known, and the labels of $\mathcal{T}$ may be unknown or they may be known but $|\mathcal{T} \cap \mathcal{S}|$ may be small. Our reweighing algorithm handles both cases. Furthermore, our reweighing procedure (Algorithm 1) works efficiently on very large data-sets, with provable guarantees.

For our running example, the drug discovery application, the set $\mathcal{T}$ consists of a subset of small molecules labeled non-binders (non-hits) for the protein MNK1, and the full labeled training set consists of molecules that are binders and non-binders for the protein MNK2, while the set $\mathcal{S}$ is

not the full training set but just the set of binders to MNK2[2]. Here the labels of the molecules in $\mathcal{T}$ are known but not necessarily on the same molecules as $\mathcal{S}$. Our goal: given a new small molecule, we want to compute the likelihood that it is a binder for MNK2 and a non-binder for MNK1. Such models can allow us to make predictions on large commercially available catalogs and enrich compounds that have high likelihood to bind to MNK2 but not MNK1. The results of our experimental implementation of a version of Algorithm 1 to predict selective binders for MNK2 over MNK1 is given in the supplement.

The rest of the paper concentrates on providing a theoretical explanation for why Algorithm 1 should work as intended and scale well in general; beyond the experiments with MNK1-MNK2.

Algorithm 1 Reweigh Distribution and Train

Input: Two data-sets: $\mathcal{S}$ and $\mathcal{T}$ of size n each, weighed according to $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$ respectively, and a tilt factor $\alpha \in [0, 1]$.
Output: A neural net model trained using a reweighed version of $\mathbb{P}_{\mathcal{S}}$ such that its weight parameters behave as if trained on distribution closer (in Wasserstein metric) to $\mathbb{P}_{\mathcal{T}}$.
▷ RandomSample returns an empirical probability distribution computed from sample size m.
▷ $R_{\mathcal{S}} \subseteq \mathcal{S}$ and $R_{\mathcal{T}} \subseteq \mathcal{T}$ denote the random sample of points from their respective ground sets.
$\mathbb{P}_{R_{\mathcal{S}}} := \mathrm{RandomSample}_m(\mathcal{S}, \mathbb{P}_{\mathcal{S}})$
$\mathbb{P}_{R_{\mathcal{T}}} := \mathrm{RandomSample}_m(\mathcal{T}, \mathbb{P}_{\mathcal{T}})$
▷ Obtain a α-tilted version of $\mathbb{P}_{R_{\mathcal{S}}}$ that's close to $\mathbb{P}_{R_{\mathcal{T}}}$ using greedy minimum weight metric bipartite matching algorithm (ScaledGreedyReweight in supplement)
$\mathbb{P}'_{R_{\mathcal{S}}} := \mathrm{ScaledGreedyReweight}(\mathbb{P}_{R_{\mathcal{S}}}, \mathbb{P}_{R_{\mathcal{T}}}, \alpha)$
▷ Obtain a reweighted version of $\mathcal{S}$
$\mathbb{P}'_{\mathcal{S}} = (1 - \alpha)\mathbb{P}_{\mathcal{S}} + \alpha\mathbb{P}'_{R_{\mathcal{S}}}$.
▷ Train neural net on $\mathbb{P}'_{\mathcal{S}}$.
Use stochastic gradient descent (SGD) to train the neural net using $\mathbb{P}'_{\mathcal{S}}$.

A difference in $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$ results in a difference in the convergence point of the weights in any neural net training procedure, like SGD. This is because the loss functions in the SGD algorithm will differ in their weights, though they may have the same form. Therefore, given two mean squared error loss functions weighted with different probability distributions, say $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$, on each of their terms, a natural question is: what is the relation between the output of two neural nets that are trained using the two different loss functions?

The expected output of any neural net, on any given input, at the end of a long enough training period, depends on the invariant measure of the weights from the SGD training algorithm. The 1-Wasserstein distance[3] between the weights in the loss functions is the same as the 1-Wasserstein distance between the occurrence frequencies of data-points in the two underlying data-sets $\mathcal{S}$ and $\mathcal{T}$. Our first theoretical contribution, Theorem 5.4, shows that $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})$, the 1-Wasserstein distance between the loss function weight distributions $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$, upper-bounds the total variation distance between the invariant measures of two such neural nets. Therefore, if the data-sets $\mathcal{S}$ and $\mathcal{T}$ are not identical, i.e., $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$ do not have the same support set, then the best we can hope for is a small $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})$, in order to obtain similar neural net models for interchangeability. The above explains our choice of the 1-Wassertein metric in Algorithm 1 (see also Remark 5.5 for comparing with the Levy metric).

Let's assume $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})$ is large, so that we need to reweigh $\mathcal{S}$. For reweighing, we do not remove any examples from $\mathcal{S}$, instead we prefer to increase the weight of some already present examples in $\mathcal{S}$. Therefore, we compute a distribution $\mathbb{P}'_{\mathcal{S}}$ with the same set of support as $\mathbb{P}_{\mathcal{S}}$, such that it minimizes 1-Wasserstein distance between $(1 - \alpha)\mathbb{P}_{\mathcal{S}} + \alpha\mathbb{P}_{\mathcal{T}}$ and $\mathbb{P}'_{\mathcal{S}}$, for some fixed choice of tilt factor $\alpha \in [0, 1]$.[4] We use $\mathbb{P}'_{\mathcal{S}}$ as the new set of weights for neural net training. Note that, for α close to 1, the distribution will be closest to $\mathbb{P}_{\mathcal{T}}$ while being supported on $\mathcal{S}$.

Note that the optimal $\mathbb{P}'_{\mathcal{S}}$ mentioned above can be computed by solving a linear program that closely resembles the 1-Wasserstein distance computation linear program. However, the size of the linear

[2]Note that we have the labeled set of non-binders to MNK2 in our training examples as well, but they remain unaffected by the reweighing, though they are used in training as well.

[3]See, for example [BGV07] and the references therein for background on the Wasserstein distances.

[4]The optimum value of α can be chosen by binary search after running multiple training evaluations.

program would be quadratic in the size of the data-sets, making the computation intractable for most practical data-sets.[5] Therefore, we look for inaccurate but efficient algorithms and a natural candidate is the randomized greedy algorithm below.

The 1-Wasserstein metric has an equivalent interpretation as a optimal transport problem. See Theorem 5.6 (essentially repeated from [SA12]) for a formal statement that reduces it to the metric minimum weight bipartite matching problem [RT81].

Given a bipartite graph with vertices embedded in a metric space, the *metric minimum weight bipartite matching problem* asks to compute a minimum weight matching, where the weight of a matching is the sum of the lengths of edges in the matching.

One tractable way to compute a minimum weight bipartite matching is to use a faster but sub-optimal algorithm. The greedy algorithm, formally studied by [RT81] in this context, is a natural contender. It is almost linear time, and easy to implement. However, [RT81] showed that such greedy algorithms can be really inaccurate. Moreover, even the greedy algorithm requires linear space and given the size of our data-sets, that can also become a constraint.

However, if our input instances admit a small sized covering then we show that the greedy algorithm run on a large enough random sample of data, i.e., Algorithm 1 for a large enough choice of m, performs reasonably well. Our main contribution here is Theorem 5.16. Theorem 5.16 states that the greedy algorithm on a small fraction random sample of an $\Theta(n)$ point data-set can be used to approximate the 1-Wasserstein distance with a poly-logarithmic factor approximation.

So, the question arises: What precisely does a small covering assumption above mean and what for kind of natural problems does Algorithm 1 scales efficiently without deterioration in the approximation guarantee?

The *metric entropy* of a point set (see Definition 5.10) is the minimum number of balls of a given radius required to cover the point set. So, in high dimensions, data-sets with low metric entropy can be characterized from their values on (relatively) small balls spread through space. This is indeed the case with DEL data-sets like MNK1-MNK2. The combinatorial synthesis process utilized in DELs often results in local chemical similarity among compounds that share common building blocks. Since similar molecules likely have the same binding behavior, synthesized molecules form a small ball around a parent molecule in the molecule fingerprint space. Therefore, molecule binding vs non-binding data-sets likely have low metric entropy. It turns out that for training data-sets with low metric entropy the greedy algorithm of [RT81] performs provably well (cf. Theorem 5.13).

The proof of Theorem 5.16 relies on Theorems 5.13 and 5.15.

Theorem 5.13 show that greedy minimum weight matching on bipartite graphs for vertex sets with low metric entropy has a much better (poly-logarithmic) approximation guarantee in our case, as opposed to the polynomial approximation guarantee from [RT81]. This argument, especially the connection between covering and matching in Lemma 5.12 may be of independent interest.

Theorem 5.15 essentially shows that random samples on data-sets with bounded metric entropy preserves 1-Wasserstein distances. This allows us to work with small samples of large data-sets. For the proof, we need large deviation bounds for the 1-Wasserstein distance between the theoretical distribution and its empirical distribution. Such results have been explored previously with tight Sanov's theorem type bounds in low dimensional spaces (see for example [BGV07]), but in high dimensions, we need the assumption of low metric entropy for the same results to go through (see chapter 6 in [DZ10]). Coincidentally, that is precisely our assumption in Theorem 5.13!

5 Theorems and Proofs

5.1 Bounding 1-Wasserstein distance suffices

In this section, in Theorem 5.4, we show that the Wasserstein distance between two measures, corresponding in the sum of squares loss function, upper bounds the total variation distance between the invariant measures underlying the stochastic gradient descent (SGD) algorithms.

[5] A typical large data-set has 10-100M examples, and computing W_1 over two such data-sets requires solving a linear program – a $\Theta(n^3)$ time procedure, resulting in the order of 10^{24} computational operations!

Assumption 5.1. *We assume that our input consists of $2n$ points among the vertices of the hypercube: $Q_{d(n)} := \{0,1\}^{d(n)}$, where $d(n)$ is $\Theta(\log n)$.*

Restricting the state-space to the hypercube is fairly standard in algorithms literature. For the drug discovery example, the state-space is just the binary molecule fingerprint vectors i.e., roughly $d(n) = 2000$. Note that assuming training and test data-sets of equal size is more for clarity of presentation, as one can add dummy example points of 0 weight, if needed.

Let $X \times Y$ denote the usual space of labeled examples i.e, in our case $X \subseteq \{0,1\}^{d(n)}$ is the set of feature values and $Y := \{0,1\}$ is the set of labels. Our object of interest in this section is a neural network with smooth bounded activation functions. Let $y = f(w,x)$ denote the abstraction of our neural network, where w denotes the real valued vector of weights. For a depth p neural-net with polynomial activation functions of degree q, $f(w,x)$ is a polynomial in x with degree at most pq.

Let $\ell(\cdot)$ denote the loss function, which we will assume to be the sum of square loss, for the sake of concreteness. The ideas easily extend to any low degree loss function. The training loss can be written as:

$$\ell_w(\mathbb{P}_{\mathcal{S}}) := \mathbb{E}_{(x,y)\sim\mathbb{P}_{\mathcal{S}}}[(y - f(w,x))^2]. \tag{1}$$

Recall that (see for example [CS18]), a stochastic gradient descent algorithm with loss function ℓ can be abstracted as the Itó diffusion in the limit of small step size:

$$\mathrm{d}w_{\mathcal{S}}(t) = \nabla_w \ell_w(\mathbb{P}_{\mathcal{S}})\mathrm{dt} + \sigma_{\mathcal{S}}\mathrm{dB(t)}, \tag{2}$$

where ∇_w denotes gradient with respect to w, $B(t)$ denotes Standard Brownian Motion in $|w|$-dimensions and the matrix $\sigma_{\mathcal{S}}$ depends upon the variance of the loss function for the mini-batch, mini-batch size and the learning rate.

Assumption 5.2. *We assume that the diffusion corresponds to an uniformly elliptic generator, since that ensures the existence of a unique limiting (invariant) measure [BD17]. Furthermore, we assume $\sigma_{\mathcal{S}}$ is isotropic i.e, it's a scalar multiple of the identity $\sigma \cdot \mathrm{Id}$ and that $\sigma_{\mathcal{S}} = \sigma_{\mathcal{T}}$, in Theorem 5.4.*

We relax the isotropy assumption somewhat in a corollary (see supplement). Finally, for our situation of interest, i.e., $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})$ large, we make a *covariate shift type assumption*.

Assumption 5.3. *We assume that $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}}) = \Omega(1)$. Furthermore, f and y are bounded, say $y, f \in [0,1]$ and*

$$\left|\mathbb{E}_{y\sim\mathbb{P}_{\mathcal{S}}(\cdot|x)}[(y-f(w,x))^2] - \mathbb{E}_{y\sim\mathbb{P}_{\mathcal{T}}(\cdot|x)}[(y-f(w,x))^2]\right| = O(1). \tag{3}$$

Essentially, it says the data-sets have similar average loss in the same neighborhood for a given set of weights.

Theorem 5.4. *Suppose we train two neural networks, under the assumptions 5.2 and 5.3 above, on different input distributions, $\mathbb{P}_{\mathcal{T}}$ and $\mathbb{P}_{\mathcal{S}}$, using the stochastic gradient descent (SGD) algorithm. Then, the total variation distance between their invariant measures can be bounded by $O(W_1(\mathbb{P}_{\mathcal{T}}, \mathbb{P}_{\mathcal{S}}))$, in the limit as SGD step-size goes to 0.*

Proof deferred to supplement.

Remark 5.5. *For dimension $d(n)$ large, the Levy-Prokhorov distance $L(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})$ between two distributions can be $\omega(1)$ times the Wasserstein distance $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})$, so a Levy-Prokhorov metric based algorithm and guarantee can be weaker than the above.*

5.2 The Greedy Algorithm

5.2.1 Reduction to bipartite matching

So far, we have established that 1-Wasserstein metric is a sufficient topology to work with. This leads to the problem of computing the 1-Wasserstein distance on two large datasets. That problem is equivalent to the minimum weight bipartite matching problem. In particular, we have the following lemma from [SA12].

Theorem 5.6. *[SA12] Given an instance of the optimal transport problem with supply and demands on two sets of points (R, B), i.e, equivalently the 1-Wasserstein distance computation problem in our case; we can construct an instance of the minimum weight bipartite matching problem such that solving the latter up to an approximation factor α will solve the former up to the same approximation factor α.*

The Algorithm ScaledGreedyWeight (below) carries out the reduction in Theorem 5.6 and calls GreedyMatch which matches two multisets embedded in a metric space using greedy algorithm on the edge lengths.

Notation: Scaling a discrete probability distribution $\mathbb{P}$ up by an integer factor of C leads to a numerical rounding error of $\frac{1}{C \min\{\mathbb{P}\}}$, where $\min\{\mathbb{P}\}$ denotes the minimum positive value of density $\mathbb{P}$. Assume that we pick a large enough constant C below, so that we can ignore the rounding error for the purposes of Theorem 5.13.

Algorithm 2 ScaledGreedyReweight (scale distributions and call bipartite matching)

Input: Two probability distributions $\mathbb{P}_B, \mathbb{P}_R$ supported on $B, R \subset Q_d$, and a tilt factor $\alpha \in (0, 1)$.
Output: Probability distribution $\mathbb{P}'_B$ supported on B. $\mathbb{P}'_B$ is close to $\alpha\mathbb{P}_R + (1-\alpha)\mathbb{P}_B$ in W_1, under assumptions of Theorem 5.13.
for $r \in R$ **do**
    $\text{Supply}(r) \leftarrow C \cdot \alpha\mathbb{P}_R(r)$
for $b \in B$ **do**
    $\text{Demand(b)} \leftarrow C - C \cdot (1-\alpha)\mathbb{P}_B(r)$
    if $\text{Demand}(b) < 0$ **then**
        $\text{Demand(b)} \leftarrow 0$
Create multi-set B', R' with multiplicities of each element being equal to their Demand and Supply respectively.
Use $\text{GreedyMatch}(R', B')$ to compute the met (matched) demands, i.e., the extent to which the demands of B that are actually fulfilled by R.
Normalize the weights of met demands to obtain a probability distribution $\mathbb{P}'_B$ supported on B.
return $\mathbb{P}'_B$.

5.2.2 Greedy algorithm and metric entropy

Recall that the data-points are set in the d-dimensional hypercube Q_d with ℓ_1 metric, where $d = \log^{O(1)} n$. The minimum weight bipartite matching problem is known to be harder than its non bipartite version. For example, the greedy algorithm is known to have a lower bound of $\Theta(n^{\log_2 3/2})$ [RT81] for the bipartite version with n data-set $\mathcal{T}$ and n data-set $\mathcal{S}$ vertices. As an aside, a variant of the greedy algorithm (the hyper-greedy algorithm) provides a $\log n$ approximation in the non-bipartite case i.e, when any vertex can be matched to any other vertex. For the bipartite case, we obtain better approximation guarantees via the greedy algorithm, assuming small metric entropy of the input point sets.

Definition 5.7. *Given a perfect matching M over a subset of vertices C in a graph G, an* alternating cycle γ *is a cycle in G such that each alternate edge in the cycle belongs to M. Note that any such M corresponds to a set of vertex disjoint alternating cycles.*

In particular, Reingold and Tarjan [RT81] essentially show the following theorem.

Theorem 5.8. *[RT81] Given a set of n data-set $\mathcal{T}$ and data-set n $\mathcal{S}$ points in a metric space, the greedy algorithm returns a matching with weight that is within a factor of $\gamma^{\log_2 \frac{3}{2}}$ of the minimum weight matching, where γ is the length of the longest alternating cycle in the set (which can be $\Theta(n)$ for $Q_{\log n}$).*

In order to improve upon their guarantee, we will exploit the following assumption for our input instance.

Assumption 5.9. *We assume that all input $\mathcal{T}$ and $\mathcal{S}$ points can be covered by η balls of radius ζ lying within Q_d. We call such an input instance* (η, ζ)-bounded. *The parameters η and ζ will determine the approximation guarantee of our algorithm.*

Definition 5.10. *Given a metric space, say (Q, d) and $E \subset Q$,, the* metric entropy $N_r^{\text{ent}}(E)$ *is the largest number of points $\{x_1, \ldots, x_n\}$ one can find in E that are r-separated, i.e., $d(x_i, x_j) \geq r$ for all $i \neq j$.*

Definition 5.11. *Given a metric space, say (Q, d) and $E \subset Q$, the (external)* covering number $N_r^{\text{cov}}(E)$ *is the fewest number of points $\{x_1, \ldots, x_n \in Q\}$ such that the d-balls $\{B(x_1, r), \ldots, B(x_n, r)\}$ cover E.*

Lemma 5.12 (Structural Lemma)**.** *For an alternating cycle γ induced by the greedy matching, if the weight of edges in the alternating cycle coming from the greedy matching is at least α times the weight of edges in the alternating cycle coming from the minimum weight matching then the metric entropy of γ is large i.e, more precisely,*

$$\left(N^{\mathrm{ent}}_{\alpha/2}(\gamma)\cdot\frac{2d-\alpha}{\alpha}\right)^{\log_2 3/2}\geq\alpha. \tag{4}$$

Proof deferred to supplement. Note that an approximation factor of d is trivial on Q_d or on any set with $d_{\min}=1$ and $d_{\max}=d(n)$. The following corollary shows that the above indeed helps to improve upon the trivial bound for appropriately bounded instances.

Corollary 5.12.1. *Lemma 5.12 implies that the greedy algorithm achieves an approximation factor of $o(d^{3/4})$ on a $(d^{3/4},d^{3/4})$-bounded instance.*

Proof. We know that $N^{\mathrm{cov}}_r(E)\geq N^{\mathrm{ent}}_r(E)$ (see for example [DZ10]). Therefore, Lemma 5.12 implies

$$\alpha\leq\left(N^{\mathrm{cov}}_{\alpha/2}(\gamma)\cdot\frac{2d-\alpha}{\alpha}\right)^{\log_2 3/2}. \tag{5}$$

For $\alpha=d^{3/4}$, the right side of Equation 5 is $d^{log_2 3/2}$, while the left side is $d^{3/4}$. Since $\log_2 3/2<3/4$ we have a contradiction. Therefore, $\alpha=o(d^{3/4})$. □

Of course, as the metric entropy decreases, the approximation factor improves, see for example the theorem below.

Theorem 5.13. *For $\eta=O(d^{\frac{1}{\xi\log_2 3/2}})$, ($\xi>1$), Lemma 5.12 implies that the greedy algorithm achieves an approximation factor of $\max\{2\zeta,O\left(d^{\frac{1+\xi\log_2(3/2)}{\xi(1+\log_2(3/2))}}\right)\}$ on a (η,ζ)-bounded minimum weight matching instance.*

Together with Theorem 5.6, Theorem 5.13 implies that the greedy algorithm obtains the approximation factor on a (η,ζ)-bounded Wasserstein distance computation instance. Proof deferred to supplement.

5.2.3 Small random samples suffice

In this subsection, we show that if the metric entropy is small, and so is the spread (see Definition 5.14) of the underlying distribution, then the empirical distribution of a much (polynomially) smaller sample is close to the actual distribution, in the 1-Wasserstein metric, with high probability.

Definition 5.14. *Let μ be the uniform distribution supported on a subset of vertices Q of $Q_{d(n)}$. The* spread *of μ, $S(\mu)$, is defined as:*

$$S(\mu):=\inf_{x_0\in Q}\left(1+\ln\left(\int_Q e^{d(x_0,x)^2}d\mu(x)\right)\right)^{1/2}, \tag{6}$$

where $d(\cdot,\cdot)$ denotes the ℓ_1 distance on $Q_{d(n)}$.

Note that the spread is positive and greater than 1, for any distribution defined on the hypercube, since the minimum value of $d(\cdot,\cdot)$ is 1. In general, $S(\mu)$ can be a function of $d(n)$.

Theorem 5.15. *For a (η,ζ) coverable point-set, with $m=\alpha(n)\left(\eta 2^{\zeta}\right)$ and $\alpha(n)\in(0,1)$, the 1-Wasserstein distance between the empirical distribution and the true distribution of data-sets with bounded metric entropy obeys the following Sanov type concentration bound:*

$$\exists\ \alpha(n)\to 0,\ \lim_{n\to\infty}\tfrac{1}{\eta 2^{\zeta}}\ln\mathbb{P}(W_1(\hat{\mu}_m,\mu)\geq\log\log n+o(1)S(\mu))\leq-\Omega(1). \tag{7}$$

The proof is in the supplement. It closely follows the covering based proof of multidimensional Cramer's theorem in its metric entropy version (exercise 6.2.19 in [DZ10]) with two main differences: (1) we need the topology induced by the Wasserstein metric instead of the Levy metric, and (2) our

space has large dimension, i.e., say $\log n$, which depends upon n[6]. The second point requires us to be more careful with the covering argument, and so we only prove a relatively weaker result, with the help of the transportation inequality from [BV05].

5.2.4 Efficiency guarantee: Greedy with random sampling

Theorems 5.15, 5.13, and 5.6 imply the following efficiency guarantee about the greedy minimum weight bipartite matching algorithm (GreedyMatch) on a random sample, and therefore Algorithm 1 as well.

Theorem 5.16. *Suppose we are given two data-sets with $\mathcal{S}$ and $\mathcal{T}$ that are weighted according to distributions $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$. If,*

1. *$\mathcal{S} \cup \mathcal{T}$ admits a small covering: an (η, ζ) covering with $\eta, \zeta = O(\log^c n)$ and $\eta = O(\log^c n)$ for some constant $c \leq \frac{1}{1+\log_2(3/2)}$; and*
2. *$\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$ are sufficiently far apart: $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}}) \geq \log\log n + o(1) \max\{S(\mathbb{P}_{\mathcal{S}}), S(\mathbb{P}_{\mathcal{T}})\}$*

then the greedy algorithm achieves an approximation ratio of $\max\left(2\zeta, O\left(d^{\frac{1+\xi\log_2(3/2)}{1+\log_2(3/2)}}\right)\right)$ with probability $1 - o(1)$, when computed on a small random sample of $r(n)$ fraction of data-points and $r(n) \to 0$.

More succinctly, Theorem 5.16 states that the greedy algorithm on a small random sample can be used to approximate $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})$ on our data-sets $\mathcal{S}$ and $\mathcal{T}$, as long as the data-sets admit a small size covering using balls of small radius, and the two training weight distributions $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$ are sufficiently different, which is the interesting case.

6 Acknowledgements

The author is grateful to Wen Torng, JW Feng, other former members of the computational drug discovery team including Jin Xu and Partick Riley for their help and advise with this paper.

References

[ABIW09] Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David Woodruff. Efficient sketches for earth-mover distance, with applications. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 324–330, 2009.

[BD17] C. Bianca and Christian Dogbe. On the existence and uniqueness of invariant measure for multidimensional stochastic processes. *Nonlinear Studies - The International Journal*, 2017.

[BGV07] François Bolley, Arnaud Guillin, and Cédric Villani. Quantitative concentration inequalities for empirical measures on non-compact spaces. *Probability Theory and Related Fields*, 137:541–593, 2007.

[BS07] Steffen Bickel and Tobias Scheffer. Dirichlet-enhanced spam filtering based on biased samples. In *Advances in Neural Information Processing Systems*, 2007.

[BV05] François Bolley and Cédric Villani. Weighted csiszár-kullback-pinsker inequalities and applications to transportation inequalities. *Annales de la Faculté des sciences de Toulouse : Mathématiques*, 14(3):331–352, 2005.

[CFHR17] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[6] This is one reason why we can't directly apply the result from [BGV07]. The constants in the exponential in their theorems will depend on n, and it's not immediately clear whether the dependence will lead to a non-trivial result.

[CMRR08] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. Sample selection bias correction theory. In *Proceedings of the International Conference on Algorithmic Learning Theory*, 2008.

[CS18] Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *International Conference on Learning Representations*, 2018.

[DMM+17] Agnieszka Dreas, Maciej Mikulski, Mariusz Milik, Charles-Henry Fabritius, Krzysztof Brzózka, and Tomasz Rzymski. Mitogen-activated Protein Kinase (MAPK) Interacting Kinases 1 and 2 (MNK1 and MNK2) as Targets for Cancer Therapy: Recent Progress in the Development of MNK Inhibitors. *Current Medicinal Chemistry*, 24:3025 – 3053, 2017.

[DSP05] Miroslav Dudik, Robert Schapire, and Steven Phillips. Correcting sample selection bias in maximum entropy density estimation. In *Advances in Neural Information Processing Systems*, 2005.

[DZ10] Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*. Springer-Verlag Berlin Heidelberg, 2010.

[GSR+17] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017.

[GT91] Harold N. Gabow and Robert E. Tarjan. Faster scaling algorithms for general graph matching problems. *Journal of the ACM*, 38(4):815–853, 1991.

[HSG+07] Jiayuan Huang, Alexander Smola, Arthur Gretton, Karsten Borgwardt, and Bernhard Scholkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, 2007.

[IK93] Celina Imielinska and Bahman Kalantari. A generalized hypergreedy algorithm for weighted perfect matching. In *BIT*, 1993.

[KMB+16] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. In *Journal of Computer-Aided Molecular Design*, volume 30, page 595–608, 2016.

[LDN+21] Trung Le, Dat Do, Tuan Nguyen, Huy Nguyen, Hung Bui, Nhat Ho, and Dinh Q. Phung. On label shift in domain adaptation via wasserstein distance. *CoRR*, abs/2110.15520, 2021.

[MSK+20] Kevin McCloskey, Eric A. Sigel, Steven Kearnes, Ling Xue, Xia Tian, Dennis Moccia, Diana Gikunju, Sana Bazzaz, Betty Chan, Matthew A. Clark, John W. Cuozzo, Marie-Aude Guié, John P. Guilinger, Christelle Huguet, Christopher D. Hupp, Anthony D. Keefe, Christopher J. Mulhern, Ying Zhang, and Patrick Riley. Machine learning on dna-encoded libraries: A new paradigm for hit finding. *Journal of Medicinal Chemistry*, 63(16):8857–8866, 2020.

[Mul16] Asher Mullard. DNA tags help the hunt for drugs. *Nature*, 530:367–369, 2016.

[QCSSL09] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[RR83] Paul Rosenbaum and Donald Rubin. The central role of the propensity score in observational studies for causal effects. In *Biometrika*, 1983.

[RT81] Edward Reingold and Robert Tarjan. On a greedy heuristic for complete matching. In *Siam Journal of Computing*, pages 676–681, 1981.

[SA12] R. Sharathkumar and Pankaj K. Agarwal. Algorithms for the transportation problem in geometric settings. In *Proceedings of the 2012 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 306–317, 2012.

[SBF+22] Alexander Satz, Andreas Brunschweiger, Mark Flanagan, Andreas Gloger, Nils Hansen, Letian Kuai, Verena Kunig, Xiaojie Lu, Daniel Madsen, Lisa Marcaurelle, Carol Mulrooney, Gary O'Donovan, Sylvia Sakata, and Jorg Scheuermann. Dna-encoded chemical libraries. *Nature Review Methods Primers*, 2(3), 2022.

[Shi00] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[Vai89] Pravin Vaidya. Geometry helps in matching. *Siam J. of Computing*, 18(6):1201–1225, 1989.

[ZKL+19] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N. Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific Reports*, 9(10752), 2019.

7 Supplement

7.1 Further details about the greedy bipartite matching algorithm

For the sake of completeness, we outline below a way to implement the greedy matching algorithm.

Algorithm 3 GreedyMatch (Greedy metric bipartite matching)

Input: Two multi-sets of n points R, B in Q_d.
Output: A matching from R to B.
▷ The set B is shared across all threads
procedure WEIGHTEDMATCH(R, B)
    for $r \in R$ **do** ▷ All for loop statements run in parallel
        $b \leftarrow \text{BreadthFirstSearch}(r, B)$
        $M \leftarrow M \cup \{r \to b\}$
    return M ▷ M is the matching
procedure BREADTHFIRSTSEARCH(r, B)
    for $i = 1, ..., d$ **do**
        for $v \in Q_d, \|v - r\|_1 = i$ **do**
            if $v \in B$ **then**
                $B \leftarrow B \setminus v$
                return v ▷ r matches to v

7.2 Proof of Theorem 5.4

Proof. Let $\mathcal{L}_\mathcal{T}, \mathcal{L}_\mathcal{S}$ be the infinitesimal generators, and let $\rho_\mathcal{T}(w), \rho_\mathcal{S}(w)$ be the invariant measures, corresponding to the SGD for $\mathcal{T}, \mathcal{S}$ respectively. Then, from our ergodicity assumption about the SGD, and the definition of invariant measures, we have:

$$\mathcal{L}^*_\mathcal{T}\rho_\mathcal{T}(w) = 0, \tag{8}$$
$$\mathcal{L}^*_\mathcal{S}\rho_\mathcal{S}(w) = 0. \tag{9}$$

We know that $\mathcal{L}_\mathcal{S}$ is a perturbation of $\mathcal{L}_\mathcal{T}$. So, let

$$\mathcal{L}^*_\mathcal{S}\rho_\mathcal{T}(w) = \varepsilon(w). \tag{10}$$

Therefore,

$$\mathcal{L}^*_\mathcal{S}\rho_\mathcal{S}(w) - \mathcal{L}^*_\mathcal{S}\rho_\mathcal{T}(w) = \varepsilon(w), \tag{11}$$

and

$$\mathcal{L}^*_\mathcal{T}\rho_T(w) - \mathcal{L}^*_\mathcal{S}\rho_\mathcal{T}(w) = \varepsilon(w). \tag{12}$$

Putting Equations 11 and 12 together, we have:

$$\begin{aligned} \mathcal{L}^*_\mathcal{S}(\rho_\mathcal{S}(w) - \rho_\mathcal{T}(w)) &= (\mathcal{L}^*_\mathcal{T} - \mathcal{L}^*_\mathcal{S})\rho_\mathcal{T}(w) \\ (\rho_\mathcal{S}(w) - \rho_\mathcal{T}(w)) &= (\mathcal{L}^*_\mathcal{S})^{-1}(\mathcal{L}^*_\mathcal{T} - \mathcal{L}^*_\mathcal{S})\rho_\mathcal{T}(w), \end{aligned} \tag{13}$$

where we have used the uniform ellipticity assumption in the last step to ensure the inverse exists. Taking 1-norm on both sides and using the sub-additivity of operator norms, we have:

$$\|\rho_\mathcal{S}(w) - \rho_\mathcal{T}(w)\|_1 \leq \|(\mathcal{L}^*_\mathcal{S})^{-1}\|_1 \|(\mathcal{L}^*_\mathcal{T} - \mathcal{L}^*_\mathcal{S})\|_1. \tag{14}$$

We will upper-bound $\|(\mathcal{L}^*_\mathcal{T} - \mathcal{L}^*_\mathcal{S})\|_1$ in terms of $W_1(\mathbb{P}_\mathcal{T}, \mathbb{P}_\mathcal{S})$. The essential idea is to simply write down the adjoints of the elliptic operators, group like terms together and use Kantorovich-Rubenstein duality to upper-bound each of the resulting terms in terms of $W_1(\mathbb{P}_\mathcal{T}, \mathbb{P}_\mathcal{S})$.

Recall that,

$$\mathcal{L}_\mathcal{T}\psi \equiv \nabla_w \ell_w(\mathbb{P}_\mathcal{T}) \frac{\partial \psi}{\partial w_i} + \epsilon^2 D \frac{\partial \psi}{\partial w_i \partial w_j}, \tag{15}$$

$$\mathcal{L}^*_\mathcal{T}\psi \equiv \frac{\partial \nabla_w \ell_w(\mathbb{P}_\mathcal{T})}{\partial w_i}\psi + \nabla_w \ell_w(\mathbb{P}_\mathcal{T}) \frac{\partial \psi}{\partial w_i} - \epsilon^2 D \frac{\partial \psi}{\partial w_i \partial w_j}, \tag{16}$$

where we have used the Einstein summation notation on the partial derivatives for the sake of brevity in expressing the last two equations. Similarly, we can write out $\mathcal{L}_{\mathcal{S}}$ and $\mathcal{L}^*_{\mathcal{S}}$.

Note that:

$$(\mathcal{L}^*_{\mathcal{T}} - \mathcal{L}^*_{\mathcal{S}})\psi \equiv \left(\frac{\partial \nabla_w \ell_w(\mathbb{P}_{\mathcal{T}})}{\partial w_i} - \frac{\partial \nabla_w \ell_w(\mathbb{P}_{\mathcal{T}})}{\partial w_i}\right)\psi + (\nabla_w \ell_w(\mathbb{P}_{\mathcal{T}}) - \nabla_w \ell_w(\mathbb{P}_{\mathcal{S}}))\frac{\partial \psi}{\partial w_i}. \quad (17)$$

One can choose ψ as any Lipschitz function of unit ℓ_1 norm, so that if we upper bound the coefficients of the two partial terms on the RHS of Equation 17 for every co-ordinate i by $W_1(\mathbb{P}_{\mathcal{T}}, \mathbb{P}_{\mathcal{S}})$, then we will have bounded $\|\mathcal{L}^*_{\mathcal{T}} - \mathcal{L}^*_{\mathcal{S}}\|_1$ by $W_1(\mathbb{P}_{\mathcal{T}}, \mathbb{P}_{\mathcal{S}})$. The first term can be upper-bounded as:

$$\begin{aligned}
\nabla_w \ell_w(\mathbb{P}_{\mathcal{T}}) - \nabla_w \ell_w(\mathbb{P}_{\mathcal{S}}) &= \mathbb{E}_{x\sim\mathbb{P}_{\mathcal{S}}}\mathbb{E}_{y\sim\mathbb{P}_{\mathcal{S}}(\cdot|x)}[(y - f(w,x))^2] - \mathbb{E}_{x\sim\mathbb{P}_{\mathcal{T}}}\mathbb{E}_{y\sim\mathbb{P}_{\mathcal{T}}(\cdot|x)}[(y - f(w,x))^2] \\
&\simeq \mathbb{E}_{x\sim\mathbb{P}_{\mathcal{S}}}\mathbb{E}_{y\sim\mathbb{P}_{\mathcal{S}}(\cdot|x)}[(y - f(w,x))^2] - \mathbb{E}_{x\sim\mathbb{P}_{\mathcal{T}}}\mathbb{E}_{y\sim\mathbb{P}_{\mathcal{S}}(\cdot|x)}[(y - f(w,x))^2] \\
&\leq O(W_1(\mathbb{P}_{\mathcal{T}}, \mathbb{P}_{\mathcal{S}})), \quad (18)
\end{aligned}$$

where we have used:

1. The covariate shift type assumption in deriving the second equality (Equation 3).
2. The Kantorovich-Rubenstein duality together with the assumption that $\mathbb{E}_{y\sim\mathbb{P}(\cdot|x)}[(y - f(w,x))^2]$ is O(1)-Lipschitz in deriving the last inequality.

Similarly, one can show the same upper-bound for $\left(\frac{\partial \nabla_w \ell_w(\mathbb{P}_{\mathcal{T}})}{\partial w_i} - \frac{\partial \nabla_w \ell_w(\mathbb{P}_{\mathcal{T}})}{\partial w_i}\right)$. Therefore, $\|\mathcal{L}^*_{\mathcal{T}} - \mathcal{L}^*_{\mathcal{S}}\|_1 \leq O(1)W_1(\mathbb{P}_{\mathcal{T}}, \mathbb{P}_{\mathcal{S}})$. □

Corollary 7.0.1. *The anisotropic diffusivity case: The upper-bound holds when the diffusivity is anisotropic as well, with the caveat that $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})$ be replaced by $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})^2$.*

Proof. The proof of Theorem 5.4 uses isotropic diffusivity in one place only – when computing the difference $\mathcal{L}^*_{\mathcal{T}} - \mathcal{L}^*_{\mathcal{S}}$. Note that, the diffusivity may be written as (see for example [CS18]):

$$D(\mathbb{P}) := \mathbb{E}\left[\nabla \ell_w(\mathbb{P}) \cdot \nabla \ell_w(\mathbb{P})^T\right] - \mathbb{E}\left[\nabla \ell_w(\mathbb{P})\right] \cdot \mathbb{E}\left[\nabla \ell_w(\mathbb{P})^T\right]. \quad (19)$$

Then, $D(\mathbb{P}_{\mathcal{S}}) - D(\mathbb{P}_{\mathcal{T}})$ can be upper-bounded in terms of $O(W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})) + O(W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})^2)$ under the assumption that we have $O(1)$-Lipschitz gradients. The argument is similar to that used for the drift term in the isotropic case, albeit with one new observation, the term

$$\mathbb{E}_{\mathbb{P}_{\mathcal{S}}\times\mathbb{P}_{\mathcal{S}}}\left[f(w)\right] - \mathbb{E}_{\mathbb{P}_{\mathcal{T}}\times\mathbb{P}_{\mathcal{T}}}\left[f(w)\right]$$

can be upper-bounded by $W_1(\mathbb{P}_{\mathcal{S}}, \mathbb{P}_{\mathcal{T}})^2$ using Kantorovich-Rubenstein duality and the definition of Wasserstein distance. □

7.3 Proof of Lemma 5.12

Proof. One way to write the greedy matching algorithm is to imagine it as a set of parallel breadth first searches (BFS), as in Algorithm 3. For any two vertices $x, y \in \mathrm{R}$ in an alternating cycle γ, suppose their neighbors from the greedy matching algorithm are x' and y' respectively. Think of the last step in Algorithm 3 before either x or y was matched, so at that time-point the BFS from x' and y' hadn't reached either x or y. Therefore, we have the following relationship between their mutual distances:

$$\min\{d(x', y), d(y', x)\} \geq \min\{d(x, x'), d(y, y')\}, \quad (20)$$

where d denotes the distance metric, which in our case is the underlying cost in W_1 i.e, the ℓ_1 distance. The situation is illustrated in Figure 3.

Now suppose that the weight of the greedy matching edges in γ is α times the weight of the minimum weight matching. Then we show below that a significant fraction of the greedy edges in the cycle γ must be at a distance at least $\alpha/2$ from their neighbors.

Let G be the set of greedy edges in γ and M be the set of optimal matching edges. Then we have,

$$\sum_{xy\in G} d(x, y) \geq \alpha \sum_{xy\in M} d(x, y). \quad (21)$$

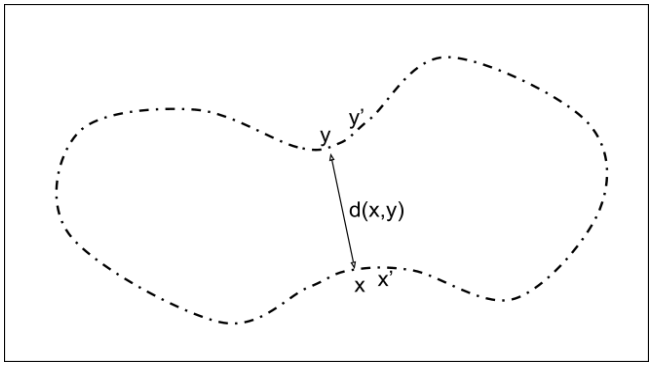


Figure 3: Alternate edges in an alternating cycle γ belong to greedy and optimal matching.

Let f be the fraction of edges in G with weight at least $\alpha/2$. Let's call that set $G_{\alpha/2}$. Recall that, in the setting of Q_d, $d_{\min} = 1$ and $d_{\max} = d$. Therefore, we have

$$d \cdot f + \frac{\alpha}{2} \cdot (1 - f) \geq \alpha. \tag{22}$$

Therefore, $f \geq \frac{\alpha}{2d-\alpha}$.

By the definition of metric entropy and Equation 20, we know that

$$|\gamma| f \leq N^{\text{ent}}_{\alpha/2}(G_{\alpha/2}) \leq N^{\text{ent}}_{\alpha/2}(\gamma). \tag{23}$$

By Theorem 5.8 we know that $\alpha \leq |\gamma|^{\log_2 3/2}$. Putting that together with Equation 23 gives:

$$\alpha \leq \left(N^{\text{ent}}_{\alpha/2}(\gamma) \cdot \frac{2d - \alpha}{\alpha} \right)^{\log_2 3/2}. \tag{24}$$

□

7.4 Proof of Theorem 5.13

Proof. We have two cases:

1. $\alpha \leq 2\zeta$: In this case, there's nothing to prove.
2. $\alpha \geq 2\zeta$: In this case, since $N^{\text{cov}}_{\alpha/2}(\gamma) \geq N^{\text{cov}}_{\zeta}(\gamma) = \eta$, we have

$$\alpha \leq \left(\eta \cdot \frac{2d - \alpha}{\alpha} \right)^{\log_2 3/2}. \tag{25}$$

Therefore, we have two sub-cases:

(a) $\alpha = \Omega(d)$: In this case, we obtain from Equation 25 that $\alpha = O(\eta^{\log_2 3/2})$, which is $o(d)$ for $\eta = O(d^{\frac{1}{\xi \log_2 3/2}})$ – a contradiction for $\xi > 1$. Hence $\alpha = o(d)$.

(b) $\alpha = o(d)$: In this case we obtain:

$$\begin{aligned} \alpha^{1+\log_2 3/2} &\leq (\eta \cdot 2d)^{\log_2 3/2} \\ \alpha &\leq O\left(d^{\frac{1+\xi \log_2(3/2)}{\xi(1+\log_2(3/2))}} \right), \end{aligned} \tag{26}$$

where we have used $\eta = O(d^{\frac{1}{\xi \log_2 3/2}})$ in the last inequality.

□

7.5 Proof of Theorem 5.15

Proof. Recall that, we have an $(\eta, \zeta) = (\log^{c_1} n, \log^{c_2} n)$ instance, for some small constants c_1 and c_2. So the covering number of the support set for μ, denoted $\mathcal{S}_\mu$, with balls of radius δ ($\delta \in [1, \zeta)$), denoted $m(\mathcal{S}_\mu, \delta)$, is upper bounded by $\eta \cdot \frac{\mathrm{Vol}(\zeta, Q_{d(n)})}{\mathrm{Vol}(\delta, Q_{d(n)})}$. Replacing the asymptotic value for the volume, we get

$$\eta \cdot \frac{\mathrm{Vol}(\zeta, Q_{d(n)})}{\mathrm{Vol}(\delta, Q_{d(n)})} \lesssim \eta \cdot 2^{-d(n)(H(\zeta/d(n)) - H(\delta/d(n)))}, \tag{27}$$

where $H(x) := x \log_2 x + (1-x)\log_2(1-x)$ is the entropy function and is negative for $x \in (0,1)$.

Since $\mathcal{S}_\mu$ of μ is finite, the set of set of probability measures M_1 that are supported on $\mathcal{S}_\mu$ is compact in the 1-Wasserstein metric topology. Therefore, there exists a finite covering of M_1, i.e., using elementary measures that are constant on the atoms of a finite covering of $\mathcal{S}_\mu$, we can approximate any given probability measure in M_1 up to an additive constant $\varepsilon + \delta$, in the 1-Wasserstein metric. The value of the constant for each ball in the covering ranging in $[0, 1]$ in steps of ε. We will fix the values of $\varepsilon \in (0,1)$ and $\delta \in [1, \zeta)$ later in the proof.

Therefore, as in exercise 6.2.19 in [DZ10], we can bound the covering number of M_1, i.e., $m(M_1, \delta, \varepsilon)$ by

$$m\left(M_1, \delta, \varepsilon\right) \leq \binom{m(\mathcal{S}_\mu, \delta)(1 + \frac{1}{\varepsilon})}{m(\mathcal{S}_\mu, \delta)} \leq \left(\frac{4}{\varepsilon}\right)^{m(\mathcal{S}_\mu, \delta)}. \tag{28}$$

Therefore, we have by the standard covering argument for the proof of multidimensional version of Cramer's large deviation bound (equivalently Sanov's theorem for finite spaces, see exercise 6.2.19 in [DZ10]):

$$\exists m_0 \; \forall m > m_0, \; \mathbb{P}(\hat{\mu}_m \in A) \leq m\left(M_1, \delta, \varepsilon\right) \cdot e^{-m \cdot \inf_{\nu \in A^{\varepsilon+\delta}} H(\nu, \mu)}, \tag{29}$$

where $H(\nu, \mu)$ is the relative entropy (KL divergence), and A^δ is the δ blow-up of $A \subset M_1$ with respect to the 1-Wasserstein metric.

Note that $\inf_{\nu \in A^{\varepsilon+\delta}} H(\nu, \mu)$ can be lower bounded in terms of the 1-Wasserstein distance using the following transportation inequality from [BV05].

Theorem 7.1. *[BV05] For distribution μ, ν supported on any polish space, we have:*

$$H(\mu, \nu) S(\mu) \geq W_1(\mu, \nu). \tag{30}$$

Essentially,

$$\inf_{\nu \in A^{\varepsilon+\delta}} H(\nu, \mu) \geq \inf_{\nu \in A^{\varepsilon+\delta}} \frac{W(\nu, \mu)}{S(\mu)} \geq \frac{W(\nu, \mu) - \delta - \varepsilon}{S(\mu)}. \tag{31}$$

For $\varepsilon \ll 1$, we have $\delta + \varepsilon \simeq \delta$. Therefore, the exponent on the RHS of Equation 29 can be lower bounded as

$$\inf_{\nu \in A^{\varepsilon+\delta}} H(\nu, \mu) \geq \frac{W(\nu, \mu) - \delta}{S(\mu)}. \tag{32}$$

Furthermore, for $m = \alpha(n)|Q| = \alpha(n)\left(\eta 2^{\zeta}\right)$, the RHS of Equation 29 can be upper bounded as

$$\begin{aligned} \mathbb{P}(\hat{\mu}_m \in A) &\leq \left(\frac{4}{\varepsilon}\right)^{m(\mathcal{S}_\mu, \delta)} \cdot e^{-\left(\eta 2^{-d(n)H(\zeta/d(n))}\right)\alpha(n) \cdot \left(\frac{W(\nu,\mu)-\delta}{S(\mu)}\right)} \\ &\leq e^{\eta \cdot 2^{-d(n)(H(\zeta/d(n)) - H(\delta/d(n)))} \ln\left(\frac{4}{\varepsilon}\right)} \cdot e^{-\left(\eta 2^{-d(n)H(\zeta/d(n))}\right)\alpha(n) \cdot \left(\frac{W(\nu,\mu)-\delta}{S(\mu)}\right)}, \end{aligned} \tag{33}$$

where we have used Equation 27 in the last inequality. If we choose δ and ε such that

$$\begin{aligned} \left(\frac{W(\nu, \mu) - \delta}{S(\mu)}\right) &\geq \frac{2^{d(n)H(\delta/d(n))} \ln\left(\frac{4}{\varepsilon}\right)}{\alpha(n)} \\ W(\nu, \mu) &\geq \delta + \frac{2^{d(n)H(\delta/d(n))} \ln\left(\frac{4}{\varepsilon}\right)}{\alpha(n)} S(\mu), \end{aligned} \tag{34}$$

equivalently for a small enough $\alpha(n)$, say $\alpha(n) = \log\log n$, choose $\delta(n) = \log\log n$ and $\varepsilon > 0$ then $W(\nu, \mu) \geq \log\log n + o(1) S(\mu)$ and the exponent in Equation 33 is negative. Thus $\mathbb{P}(\hat{\mu}_n \in A) \to 0$ as $n \to \infty$. □

7.6 Experimental results

MNK1 and MNK2 are two structurally similar kinases responsible for cell signalling. Their inhibition has been explored for certain cancer therapies (see for example, the survey [DMM+17]). In this section, we describe an application of Algorithm 1 that selects for MNK2 binders which are MNK1 non-binders.

We use (roughly) the same set-up as in [MSK+20] (the exact details are elaborated in Supplementary subsection 7.7). But, at a high level, we start with a train data-set $\mathcal{S}$, labeled as hits (binders) and non-hits (non-binders) for MNK2, and another labeled set $\mathcal{T}$ which consists of just MNK1 non-hits (non-binders)[7]. Let $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$ denote the distribution of weights on the examples in the two data-sets. We assume $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$ are uniform in our experiment section, but their set of support is different. Next, we used $\alpha = 0.95$ in Algorithm 1 to reweigh the MNK2 binders in $\mathcal{S}$ for training using $\mathcal{T}$ (the MNK1 non-binders). In theory, this should bring the limiting distribution of network weights, that results from training on reweighed $\mathbb{P}_{\mathcal{S}}$, "closer" to that which can be obtained from training on $\mathbb{P}_{\mathcal{T}}$. That's our experimental model. For the baseline model, we simply skip the reweighing step in Algorithm 1 and train the network using SGD.

7.7 Further details about the experimental setup

Below we provide some more details of our experiments beyond the high level description in the main paper. We use a scaled down but otherwise same setup as in [MSK+20]. The experiment consists of two sets of training data:

1. Baseline data: The baseline training set consists of labeled disynthon examples divided into five classes: (1) Non-hit, (2) Matrix binder, (3) Promiscous hit, (4) Non-competitive hit (5) Competitive hit. Of these five, the class of interest is competitive hit and has about 1,200 molecules. The overall training set size is approximately 250K labeled disynthons (similar to, but a scaled down version of [MSK+20]) for MNK1 and MNK2 combined.[8]
2. Experiment data: This is exactly the same as baseline data, except for one caveat: molecules that are competitive binders to MNK2 and close to non-hits to MNK1 in the molecular fingerprint space, have their weights increased in the loss function, using the transportation algorithm (Algorithm 1) described in the previous section. In other words, small molecules close to non-hits to MNK1 are weighted relatively higher amongst the competitive binders to MNK2.
3. We use a holdout of our labeled data-set, that consists of about 7K small molecules that belong to the labeled set of: MNK2 binders, and MNK1 binders or MNK1 non-binders; to compute the selectivity of our algorithm. About 40% of this set was labeled MNK2 binders and MNK1 non-binders. For both models we obtain a set of top 100 molecules in the holdout data that are predicted to be the strongest binders to MNK2.

7.8 Neural network predictions

In this subsection we discuss the accuracy and predictions of our neural network.

Definition 7.2. *Given a model that predicts MNK2 binders vs non-binders and an inference data-set of known MNK2 binders, predict the top 100 molecules that are the most likely to be selective for MNK2 over MNK1. Define* selectivity *as the fraction of those top 100 molecules that don't bind to MNK1.*

Figure 4, reproduced from the introduction, compares the reweighed and baseline model selectivity.

We measure the success of our reweighing experiment using two metrics:

[7] These labeled training data-sets are proprietary and can't be disclosed, as was the case in [MSK+20]. Hence we train on proprietary data-sets, but perform inference on publicly available data-sets (Enamine and MCULE catalogs). to validate the model prospectively, we experimentally tested our top predicted selective molecules from the publicly available Enamine catalog, which results in 2 molecules (of 43 tested) verified to be selective by single point of concentration assays at 10μM. See Subsection 7.9 for more details.

[8] Because of our reliance on proprietary disynthon libraries, the training data-sets can't be open sourced, as was the case in [MSK+20].

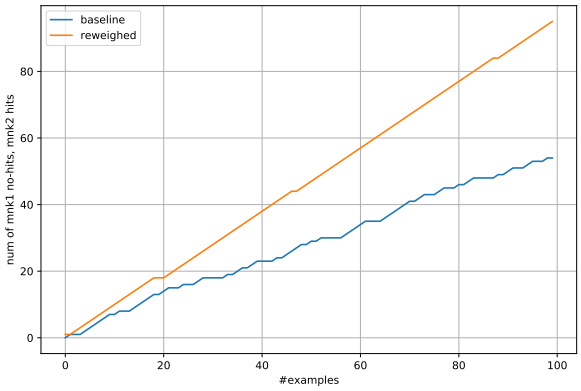


Figure 4: Selectivity of reweighed (using Algorithm 1) and baseline (without reweighing) neural nets. Note that this increase in selectivity from 54% to 95% came without any significant change in the MNK2 hit vs no-hit validation loss.

1. The change in the AUC for MNK2 binder vs non-binders: Increasing the weight of a subset of non-competitive binders means that we deliberately introduce training-test distribution skew in our data. This has the potential to impact AUC negatively. However, we found that our AUC stayed around 0.6 for the MNK2 binder vs non-binder classification, and did not change significantly between baseline and experiment.

2. The increase in selectivity of MNK1 non-hits: We measure this as the proportion of MNK1 non-hits and MNK2 competitive hits in our top 100 molecules ranked by model scores from a labeled set of 7K molecules that were known to be MNK2 competitive hits. The result is shown in Figure 4.

   The reweighed data was 95% selective as opposed to 54% for baseline. This means that 95 of the top 100 MNK2 competitive hits predicted by our model were MNK1 non-hits, and the remaining 5 were MNK2 hits and either MNK1 hits or matrix binders. A similar interpretation holds for the baseline evaluation, which was 54% selective.

7.8.1 Model properties

Next, we want to investigate any clustering behavior of small molecules in the holdout set, i.e., how higher weighted molecules, by Algorithm 1, are distributed relative to each other in the chemical fingerprint space. In order to visualize the effect, we compute and plot the histogram of mean Tanimoto similarity[9] for each of the top 100 binders, in the baseline model, with the remaining 99 of the top 100 molecules in the baseline model, in Figure 5. Similarly, we compute and plot the histogram of mean Tanimoto similarity between each of the top 100 binders in the reweighed model, with the remaining 99 of the top 100 molecules in the reweighed model, in Figure 6.

We find that the molecules in the experimental model are more similar to each other than the baseline model (mean similarity increases from 0.32 to 0.53), and hence are "packed" more closely together.

One potential reason for the top binders in the experimental model to be clustered together could be that Algorithm 1 is increasing the weight on the (likely small) portion of chemical space of MNK2 binders which are in the vicinity of MNK1 non-binders. In order to support this assertion, we compute the mean ECFP6 Tanimoto similarity for each of the top 100 binders in the baseline model with the top 100 binders of the reweighed model, and vice versa (Figures 8 and 9 respectively). We observe that the mean similarity score for the former, in Figure 8, is clearly bifurcated into two parts with means around 0.2 and 0.5 respectively. The right peak of 30 small molecules mostly comes from the 54 molecules that were MNK1 non-binders. This means that a subset of the top 100 MNK2 binders in baseline are much more similar to the top 100 MNK2 binders in the reweighed model. This leads us to the following remark.

[9]Tanimoto similarity (Jaccard distance) of two molecules ranges in $[0, 1]$, with higher values indicating more similarity.

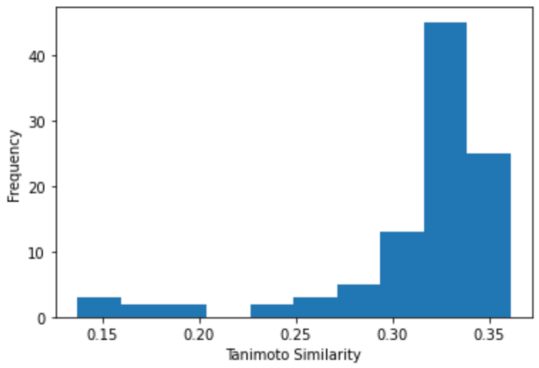


Figure 5: Clustering of top molecules in base model

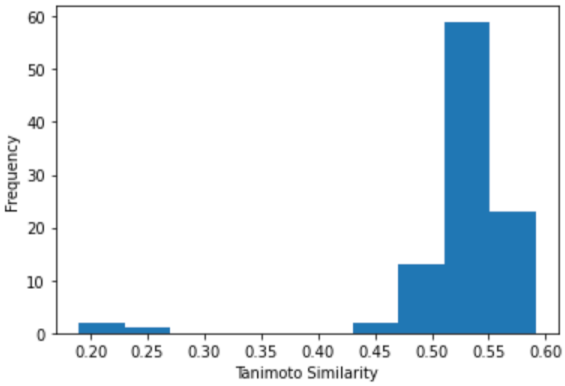


Figure 6: Clustering of top molecules in reweighed model

Figure 7: How spread out are top 100 molecules in baseline and reweighed models, by themselves

Remark 7.3. *Figures 5, 6, 8 and 9 are consistent with the following hypothesis: Algorithm 1 is successfully highlighting a closely packed portion of the chemical space of MNK2 binders that are non-binders to MNK1, while the top MNK2 binders in the baseline model are relatively more spread out in the chemical fingerprint space.*

7.9 Wet-lab results

Due to proprietary reasons we can not make our code and data public. However, to ensure a degree of verifiability of our results, about fifty of the top predicted selective small molecules from the enamine catalog were synthesized and experimentally tested for binding properties to MNK1 and MNK2. This subsection discusses those results.

We selected 50 of the top predicted small molecules that should bind to MNK2 but not to MNK1 from the enamine catalog of 1.9 Billion small molecules, for synthesis and experimental testing. The selection process essentially filtered out any small molecule that was above a ECFP6 Tanimoto similarity of 0.3 with respect to an already selected (higher score) small molecule in the top predicted selective molecules. At $10\mu M$ concentration, it was found that 2 out of 43[10] (roughly 5%) of predicted small molecules reduced the enzyme activity of MNK2 below 50% but not that of MNK1. The full table of activities for all 43 molecules is given in Section 7.10.

Remark 7.4. *Note that most small molecules that bind to MNK2 will also bind to MNK1 because of structural similarities. Admittedly, our training set is an order of magnitude smaller than that in [MSK$^+$20] and the number of molecules experimentally verified is also an order of magnitude smaller than [MSK$^+$20]. However, if the result scales up, then the 5% experimentally*

[10]Seven molecules could not be synthesized and tested.

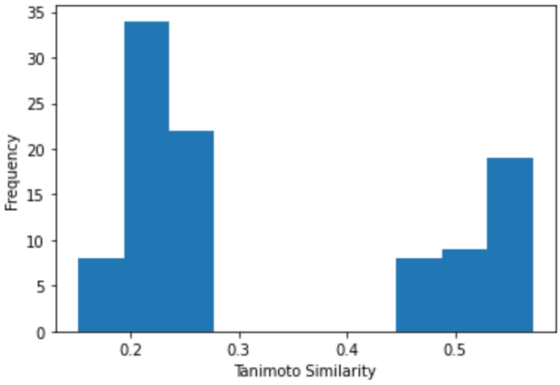


Figure 8: Mean Tanimoto Similarity for each top molecule in base with reweighed model (note the bifurcation).

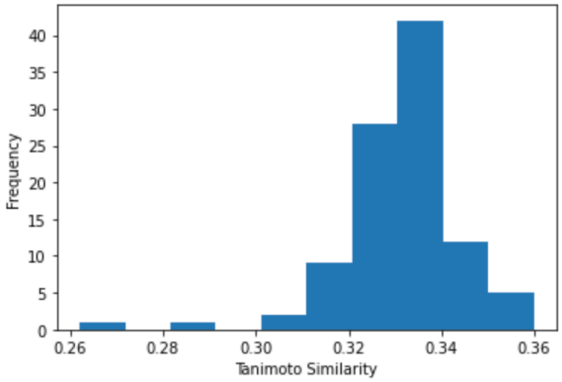


Figure 9: Mean Tanimoto Similarity for each top molecule in reweighed with base model

Figure 10: How spread out are top 100 molecules in the two models, relative to each other

verified success rate, in predicting selectivity against two targets simultaneously, may be further compared with the 30% *success rate for predicting binders against single targets using similar ML approaches [MSK*$^+$*20].*

The names of the two molecules in Figure 2 are:

1. For C(NC=1N=CC=C(OC=2C=CC=CC2)N1)C=3C=CC(=CC3)C=4C=CC=NC4, the IUPAC name is:
   4-phenoxy-N-[4-(pyridin-3-yl)phenyl]methylpyrimidin-2-amine,
2. For C(C1CCN(CC1)C=2C=NC=C(OC3CCCCC3)N2)C=4C=CC=NC4, the IUPAC name is:
   2-(cyclohexyloxy)-6-4-[(pyridin-3-yl)methyl]piperidin-1-ylpyrazine.

On average the model predicts molecules which are better binders to MNK2 than MNK1, as can be seen by the plot of the enzyme activity of the forty three small molecules below.

The full list of wet-lab tested molecules and the enzyme activities at $10\mu M$ concentration of the small molecule is given in the appendix Section 7.10.

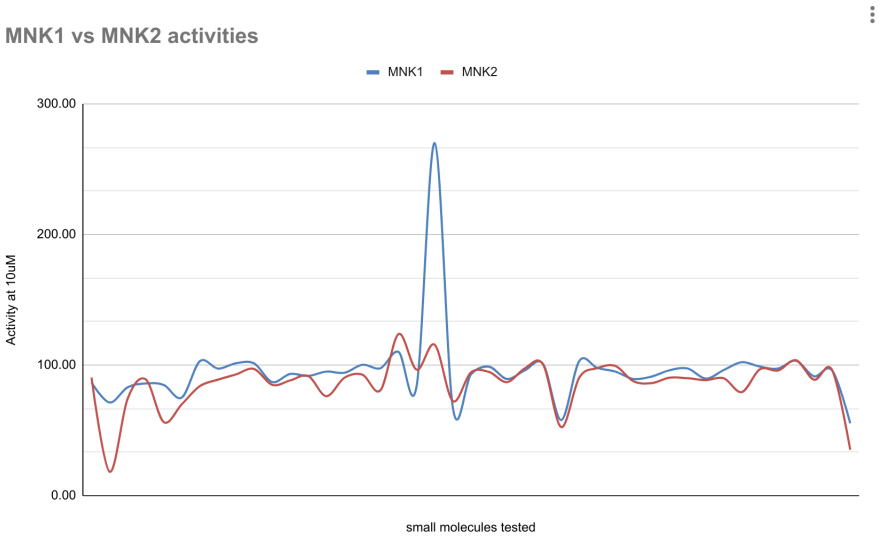


Figure 11: Average enzyme activities; lower y-values indicate better binders

7.10 Smiles from wet-lab experiment

We selected fifty molecules from the top predicted binders for MNK2 and non-binders for MNK1, from the enamine catalog. The selection ensured that the ECFP6 Tanimoto similarity between any two wet-lab tested molecules was at most 0.3. This was done in order to ensure sufficient diversity among the selected compounds for experimental testing.

The smiles with activities for all forty three (seven molecules could not be tested) wet-lab tested molecules are given in Figure 12. The mean activation value for the MNK1 columns was 96 and that for the MNK2 columns was 88, thus on average the molecules bind more to MNK2 than MNK1. Note that for single point of concentration assays the results can be noisy so that the difference is likely not statistically significant for sample of size 43.

	MNK1		MNK2	
smiles	Data 1	Data 2	Data 1	Data 2
CC1=NC(=CC(=N1)N2CCC(CC2)OC=3C=NC=CN3)N4CCC(CC4)OC=5C=NC=CN5	86.69	87.99	90.55	99.54
C(NC=1N=CC=C(OC=2C=CC=CC2)N1)C=3C=CC(=CC3)C=4C=CC=NC4	71.46	72.21	18.43	21.70
FC(F)(F)C=1C=CC(OC2CCN(CC2)C=3C=NC=C(N3)N4C=NC=N4)=NC1	83.11	84.59	74.39	72.46
CC1=NC=CN1C=2C=NC=C(OC=3C=CC(=CC3)N4C=NC=N4)N2	86.02	96.16	89.16	101.05
CC(NC(=O)CNC(=O)C=1C=CC(Cl)=CC1)C=2C=CC(=CC2)C=3C=CC=NC3	84.85	84.42	56.41	61.07
OCCOC=1C=NC=C(OC=2C=CC(=CC2)C(=O)C=3C=CC(F)=CC3)N1	75.24	75.47	70.12	80.63
CC=1C=NC(=NC1C=2C=CC=NC2)N3CCCC(CC=4C=CC=NC4)C3	103.08	106.18	84.04	82.69
COC=1C=CC=C2C(C)=C(OC12)C(=O)N3CCC=C(C3)C=4C=CN=CC4	97.37	95.45	88.92	93.10
CNC(=O)C=1C=CC(OC)=C(NC(=O)C=2C=CC=3C(=O)NCC(=O)NC3C2)C1	101.53	98.29	92.99	94.95
O=C(CN1C=NC=2SC=CC2C1=O)NCCC3CCC=CC3	101.38	100.77	97.11	99.23
O=C(OC1CCN(CC1)C=2N=NC=C3C=CC=CC23)C=4C=CC=C(OC=5C=NC=CN5)C4	87.11	87.20	84.94	88.91
CC(C)CNC(=O)CNC(=O)CCC=1C=CC=2C=CC=CC2N1	93.35	96.25	88.33	92.38
CNC(=O)C1(C)CCN(C1)C(=O)C2=CC(Cl)=CN2	91.75	101.76	91.70	94.38
FC(F)(F)C=1C=CC=C(C1)C2CCCN(C2)C(=O)CC=3C=CC(=CC3)N4C=CN=C4	95.11	102.67	76.26	80.86
O=C(CCN1C=NC=2SC=3CCCCC3C2C1=O)N4CCC(CC4)OC=5C=NC=CN5	94.20	109.93	90.41	88.41
FC(F)(F)C=1C=CC(=CC1)C2=CC=C(O2)C(=O)NC3CCN(CC3)C(=O)C=4C=NC=CN4	100.34	101.08	92.75	88.55
CC(NC=1N=CC=C(OC=2C=CC(F)=CC2)N1)C=3C=CC(=CC3)N4C=CN=C4	97.72	109.81	80.88	73.82
O=C(NC=1C=CC(OC=2C=NC=CN2)=CC1)C3CN(C(=O)C3)C=4C=CC=5CCCC5C4	110.17	118.25	123.91	132.32
O=C(CN1C=NC=2SC(=CC2C1=O)C=3C=CC=CC3)N4CCC=CC4	84.23	92.81	96.53	98.60
CCC=1C=C(OC=2C=CC=C(C2)N3C=CN=C3)N=C(N1)C=4C=CC(=CC4)[N+](=O)[O-]	270.12	297.77	115.62	167.71
CCN1C=C(OC=2C=NC(=CN2)C(=O)OC=3C=CC(=CC3)C4CCOC4)C=N1	67.62	72.29	72.54	71.02
CSC=1C=CC(CCC(=O)NCC(=O)NC=2C=CC(Cl)=CC2)=CC1	92.89	98.48	94.42	98.44
CC=1C=C(NC(=O)CN2C(=O)CC=3C=CC=CC23)C=CC1C=4C=CC=NC4	98.88	93.22	94.87	95.46
CCN1N=CC=C1C2CCN(CC2)C=3C=NC=C(OCC=4C=CC=C(OC(F)(F)F)C4)N3	89.45	85.68	87.02	86.01
CNC(=O)C=1C=CC=C(CCNC(=O)CN)C1	96.11	97.23	97.75	93.28
CNC(=O)C1CC=2C=CC=CC2N1C(=O)CN3C=CC=4C=CC=CC34	100.84	94.29	100.54	93.63
COC=1C=NC=C(CN2C=C(C=N2)C=3C=CC(Br)=CC3)N1	58.01	56.88	52.60	52.23
ClC=1C=NC=C(OC2CCN(CC2)C(=O)C=3C=NC=C(N3)N4C=CN=C4)C1	103.22	95.26	90.49	101.93
CC(NC(=O)C=1C=CC=C(C1)C2CCOC2)C=3C=CC(OC=4C=CC=NC4)=CC3	97.86	100.28	97.77	96.84
CC=1C=C(N=C(N1)C=2C=CC(Cl)=CC2)C(=O)OCC3CCN(CC3)C=4C=NC=CN4	95.10	97.59	99.38	94.90
CNC(=O)CN1C=CC=2C=CC(NC(=O)CN3C=CC(C)=CC3=O)=CC12	89.43	92.56	87.68	90.41
CC(C)(C)NC(=O)CNC(=O)CCCC=1C=CC=2NC(=O)CC2C1	91.29	92.65	86.27	94.26
CN(C1CCC=2C=CC=CC2C1)C(=O)C=3C=NN(C)C3N4C=CC=C4	96.13	96.01	90.44	89.75
CC=1C=C(O)N=C(C=CC=2C=CC(OC=3C=NC=CN3)=CC2)N1	97.46	97.73	90.02	88.79
CCN1C=C(C=N1)C=2C=CC(OC=3C=CC=NC3)=CC2	89.76	88.17	88.59	89.01
FC(F)(F)OC=1C=CC=C(CC(=O)NCC(=O)NC=2C=CC(=CC2)N3C=NC=N3)C1	96.28	92.74	90.01	88.07
CNC(=O)CC=1C=CC=C(NC(=O)CN2C=CC(=O)C=3C=CC=CC23)C1	102.30	99.49	79.40	82.48
CNC(=O)C=1C=CC(=CC1)C=2C=CC=CC2CN3CCCCC3	98.92	98.91	96.91	99.78
CC=1C=C(N=C(N1)C=2C=CC=NC2)N3CCC(CC4=CC=NN4)CC3	97.40	92.56	95.88	98.21
CNC(=O)C=1C=CC(CCNC(=O)[C@H](N)CC2CCCCC2)=CC1	103.43	95.16	103.79	92.31
ClC=1C=NC=C(OC2CCN(CC2)C(=O)C=3C=CC(OC=4C=CC=NC4)=CC3)N1	91.46	88.23	88.85	92.21
CNC(=O)C1CCCN1C(=O)CNC(=O)C2=CC(Cl)=CN2	95.95	92.44	96.12	88.42
C(C1CCN(CC1)C=2C=NC=C(OC3CCCCC3)N2)C=4C=CC=NC4	55.51	63.99	35.24	44.13

Figure 12: Columns denote % enzyme activities relative to DMSO control. Each compound was tested twice against each kinase, at a concentration of 10μM. Reactions were carried out at Km ATP according to the RBC Km binning structure. Control compound, Staurosporine, was tested in 10-dose IC50 mode with 4-fold serial dilution starting at 20μM with IC50 for MNK1: 1.19E-7 and MNK2: 1.8E-8. Note lower column values indicate a better binder.