Enhanced Sampling of Chemical Space for High Throughput Screening Applications using Machine Learning

Sarvesh Mehta,[†] Siddhartha Laghuvarapu,[†,§] Yashaswi Pathak,[†,§] Aaftaab Sethi,[‡] Mallika Alvala,[¶] and U. Deva Priyakumar[*,†]

†*Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500 032, India*

‡*Department of Medicinal Chemistry, National Institute of Pharmaceutical Education and Research, Hyderabad 500 037, India*

¶*School of Pharmacy and Technology Management, Narsee Monjee Institute of Management Sciences, Hyderabad*

§*Contributed equally to this work*

E-mail: deva@iiit.ac.in

Phone: +91 40 6653 1161. Fax: +91 40 6653 1413

Abstract

In drug discovery applications, high throughput virtual screening exercises are routinely performed to determine an initial set of candidate molecules referred to as "hits". In such an experiment, each molecule from large small-molecule drug library is evaluated for physical property such as the binding affinity (docking score) against a target receptor. In real-life drug discovery experiments, the drug libraries are extremely large but still a minor representation of the essentially infinite chemical space , and evaluation

of physical property for each molecule in the library is not computationally feasible. In the current study, a novel machine learning framework "MEMES" based on Bayesian optimization is proposed for efficient sampling of chemical space. The proposed framework is demonstrated to identify 90% of top-1000 molecules from a molecular library of size about 100 million, while calculating the docking score only for about 6% of the complete library. We believe that such a framework would tremendously help to reduce the computational hour and resources in not only drug-discovery but also areas that require such high-throughput experiments.

Introduction

Drug discovery process is an extremely laborious process and the pipeline involves several steps each of which is both expensive and time consuming. The first step in the process after target identification and validation is to identify hit molecules, where potential strong binding drug–like molecules against a drug target are identified using computational methods. Once the hit molecules are identified, they are experimentally evaluated typically using biochemical assays towards lead identification. Further processes involve lead optimization, *in vitro* and *in vivo* evaluation, pre-clinical studies and clinical trials before the drug can be approved for use. Structure based drug design (SBDD) method, docking, is routinely used for identification of lead molecules.[1–4] In SBDD method, large libraries of ligands[5–7] are virtually screened for their binding affinity against a drug target, which is a measure of the inter-molecular interaction between the target and the ligand.

Recently new methods that use modern deep/reinforcement learning have been proposed to tackle problems in molecular sciences such as physical property prediction,[8,9] and *de novo* molecule generation. Most of the deep learning models that tackles the problem of molecular generation are based on Variational autoencoders[10–13] and Generative Adversarial Networks.[14–16] These models typically map a lower dimensional continuous real number space z to a discrete chemical space and usually combined with Bayesian Optimization(BO),[16–19]

or Reinforcement Learning(RL) based algorithms[20–22] to bias the generation of compounds towards a desired property. SMILES based generative models[23,24] are combined with RL methods to generate new molecules but suffer from the problem of generating chemically invalid molecules.[22,25] Recently You et al. proposed a graph convolutional policy network that considers a molecule as a graph, and iteratively modifies the graph while optimizing for a target property and maintaining chemical validity at every step. Although these methods have been seen to perform really well on optimizing for tasks such as QED and LogP, they have been shown to perform inadequately while optimizing for objective functions involving docking calculations.[26] Moreover, in a recent study by Gao and Coley,[27] it was demonstrated that although the molecules generated by these methods are novel and diverse, they may be very difficult/infeasible to synthesize and hence cannot be of practical importance in a real-life drug discovery scenario.

On the contrary to the molecules generated by deep generative models, molecule library enumerated via simple reactions can be novel, diverse and at the same time practically be synthesized with a probability of $\sim 86\%$.[27–29] In a recent study performed by Lyu et al.,[28] 96 million docking calculations were performed against AmpC receptor. Among these the top ranked 1 million compounds (1 % of the initial set) were systematically examined to identify hit molecules, which were further validated experimentally. In the same study, 138 million docking calculations were performed for D_4 dopamine receptor, which was used to show that the hit-rates fell almost monotonically with the docking-score. Although, Lyu et al. docked compounds in the order of 10^8, it is still a small fraction when compared to the 1.6 billion molecules enumerated in ZINC Library. Moreover, their study also shows that hits for a target can be identified using only the top fraction of the ligands with respect to the docking score. Hence, a sampling method that can efficiently search the chemical space for high docking scores would speed up the process.

Recently, Gentile et al. proposed a deep learning based method "Deep Docking" to augment the process of SBDD.[30] In this work, iterative docking is done of a small portion of

large libraries. The obtained values are used to train ligand-based QSAR models, which are used to predict the scores of the remaining ligands in the library. A cut-off is set to identify the *hits* among these predicted molecules. Molecules are then randomly sampled from these hits to further train the QSAR model for the next iteration. In this manner, the author claim that with docking upto 50 times fewer molecules, 60% of the top scoring molecules can be retrieved. The problem with this approach is that, although the model performs well when it used to estimate the scores of molecules similar to the ones trained with, the prediction error is expected to be high for the molecules that have significant deviation from the ones trained with. This may result in poor selection of molecules after each iteration. Instead, if one can use a model that can also estimate the confidence in prediction, the confidence score can be incorporated for better iterative selection of molecules.

In this work, a novel Machine learning framework for Enhanced MolEcular Screening (MEMES) based on Bayesian optimization is proposed for efficient sampling of molecules during SBDD process. In the framework, the initial set of molecules are first featurized and represented as molecular vectors. These are then clustered using the K-means clustering algorithm. A small set of molecules are sampled from each cluster to build an initial diverse set of ligands, and their docking scores are calculated. A Gaussian process is trained as a surrogate function for the protein-ligand docking score. Two variants of the MEMES framework, ExactMEMES and DeepMEMES are introduced depending upon the choice of the surrogate function used (see Methods section). The initial training set is iteratively updated by sampling a small portion of molecules not previously sampled based on an acquisition function, and the process is repeated, until the maximum number of allowed docking calculation is reached. The proposed framework successfully samples a very high fraction of the top hits for a given protein and molecular library, while only calculating docking scores for 6% of the complete molecular library. Further, extensive analysis has been done to show the robustness of the framework on different proteins and molecular libraries with varying size.

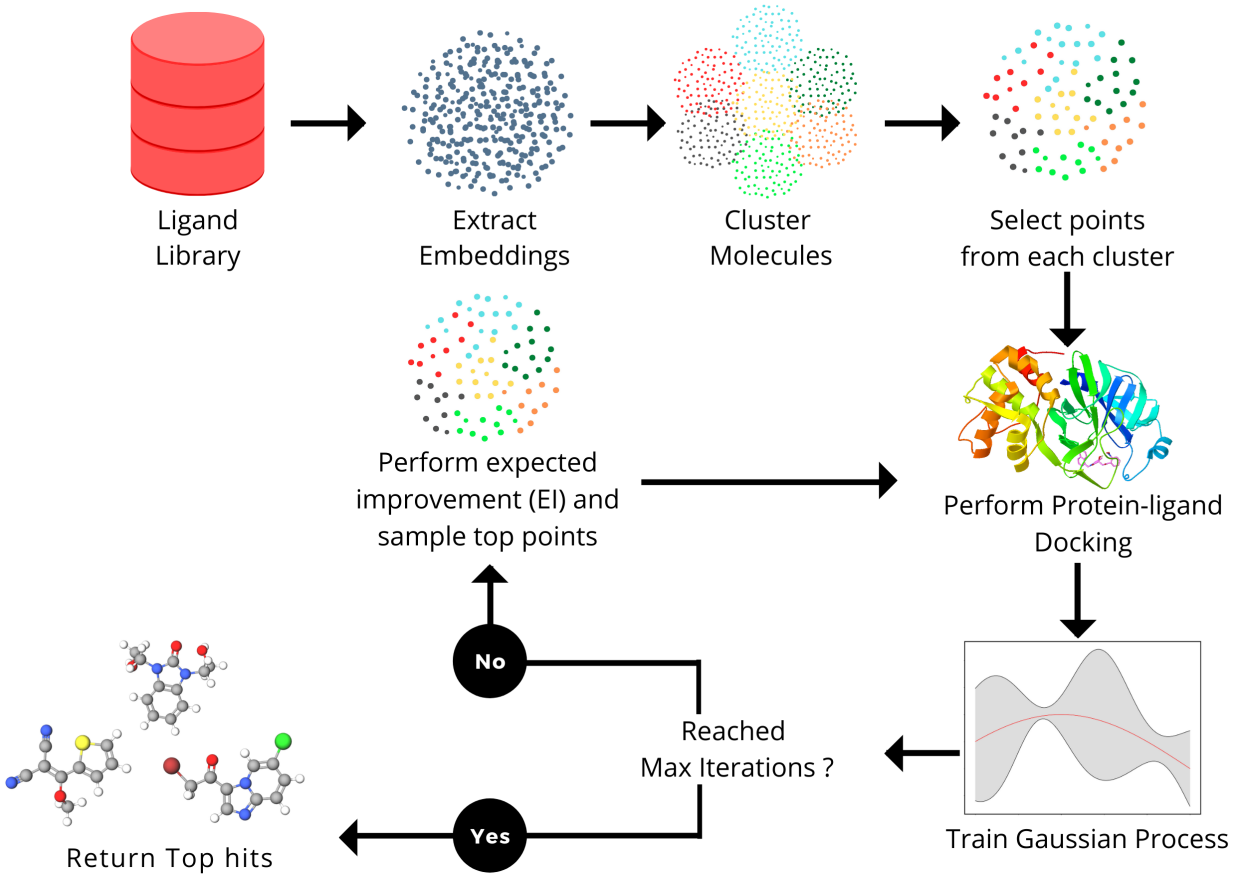


Figure 1: Overview of the proposed method, MEMES

Results and Discussion

The framework proposed in this work "MEMES" is based on Bayesian optimization (Figure 1). Firstly, in the MEMES method, all the ligands in the library are represented as fixed dimension feature vectors. Secondly, a small fraction of molecules are chosen to be the initial set. To ensure that this "inital set" is diverse and representative of the complete molecular library, a K-Means clustering[31] is performed on the pre computed feature vectors and molecules are uniformly sampled from each of the resulting clusters. Docking scores, for each of the molecule in the initial set is computed against the given target receptor. A Gaussian process[32–34] is then trained on this inital set. A new set of molecules is then picked from rest of the dataset based on the"Expected Improvement" values calculated using the trained gaussian process (see Methods). The docking score of these molecules

are computed and these are added to the initial training set and the gaussian process is retrained. The procedure is repeated iteratively, until the computational budget is reached or no improvement is observed.

In this section, the capability of the MEMES framework to sample set of molecules having high binding affinity and high overlap with the actual top hit molecules while only performing docking calculations on only 6% of the molecules in the complete library is demonstrated. Further, the capability of the proposed method to sample a diverse set of molecules is shown. In this work, performance of MEMES framework is evaluated on two different surrogate functions ExactGP and DeepGP. Although the ExactGP theoretically guarantees better performance than the DeepGP, it cannot be extended to be used on ultra large docking libraries due to computational constraints. Hence, in the subsequent subsection, the performances of ExactGP and DeepGP as the choice of surrogate function in the MEMES framework are compared to validate the performance of DeepMEMES against ExactMEMES. In the following subsection, the performance of the MEMES framework with DeepGP is demonstrated on large docking libraries. Finally, the robustness of the MEMES framework is demonstrated by applying it on molecular libraries with sizes ranging from 2 million to 96 million compounds.

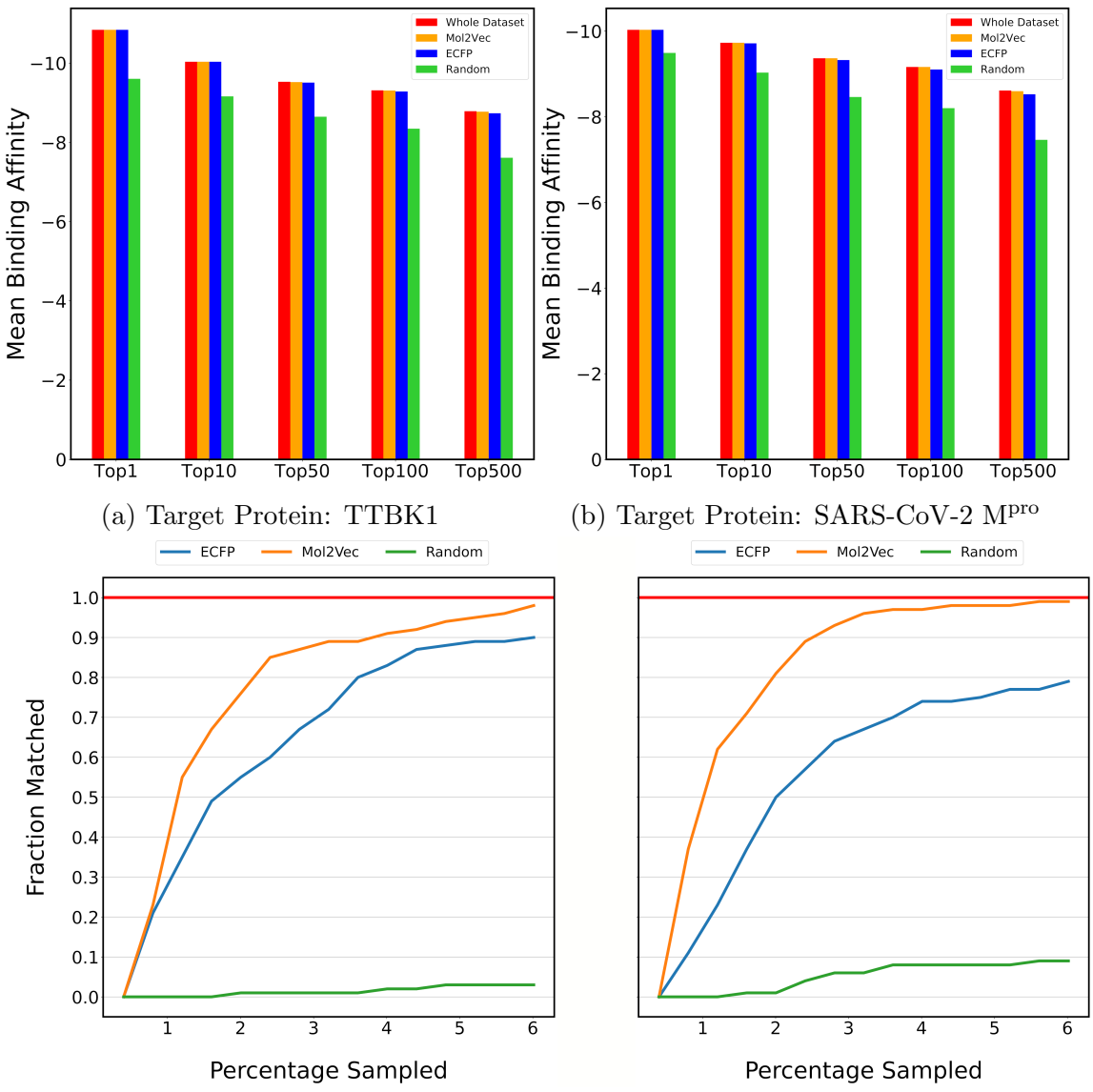
MEMES Identifies $95+\%$ of Top Candidates by sampling only 6% of the Dataset

Zinc-250K dataset contains 250,000 drug like molecules obtained from ZINC 15 database.[7] The ExactMEMES (MEMES framework with ExactGP) was applied on the Zinc-250K dataset against two protein receptors: Tau-Tubulin Kinase 1 (TTBK1) an attractive target protein to combat many neurodegenerative diseases such as Alzheimer's and main protease (M^{pro}) of SARS-CoV-2, responsible for the outbreak of COVID-19. As the ExactGP used in this framework cannot be applied to a very large molecular library, the ZINC-250K dataset was selected to assess the performance of ExactMEMES.

Virtual screening docking calculation were performed to identify molecules, that have high docking score against a target receptor, i.e. to find top hits. It is also desired that the top hits identified in this process are diverse and span the complete molecular library. Here, we show that ExactMEMES framework (with only 6% docking calculations) is able to sample molecules that have high binding affinity, high overlap with actual top hits and are spread across the chemical space of the given molecular library. This demonstrates that MEMES framework not only identifies molecules exhibiting high binding affinity but most of the top molecules in the complete library.

Figure 2a and 2b show the mean binding affinity of actual top molecules in the molecular library, top molecules sampled by ExactMEMES framework with Mol2Vec and ECFP as molecular featurizer technique and that by random sampling method, against TTBK1 and SARS-CoV-2 M^{pro} respectively. Top 20 docking hits in complete docking library for both target receptor are given in Supplementary Fig. S1 and S2. For, ExactMEMES and random sampling, 15000 ($\sim$ 6% of the complete molecular library) docking calculations were performed. From the Figure 2 its quite evident that the ExactMEMES methods significantly outperforms the random sampling baseline and matches the mean binding affinity of actual top compounds present in the molecular library. Figure 2 also shows that ExactMEMES with Mol2Vec featurization outperforms ExactMEMES with ECFP featurization. Distribution of binding affinity of top sampled molecules is given in the Supplementary Fig. S3.

Figure 2c and 2d shows the fraction of top 100 sampled molecules that are actual top hits for receptors TTBK1 and SARS-CoV-2 M^{pro} against the percentage of molecules sampled from the docking library using ExactMEMES and random sampling (see Supplementary Fig. S4 for similar analysis on top 500 sampled molecules). Figure 2c and 2d shows that ExactMEMES significantly outperforms random sampling and almost shows a complete overlap with the actual top hits when the percentage sampled is around 6%. Further intersection of top 100 molecules sampled by ExactMEMES framework, random sampling, and actual top hits for receptors TTBK1 and SARS-CoV-2 M^{pro} from the molecular library is shown in the



(a) Target Protein: TTBK1

(b) Target Protein: SARS-CoV-2 M^{pro}

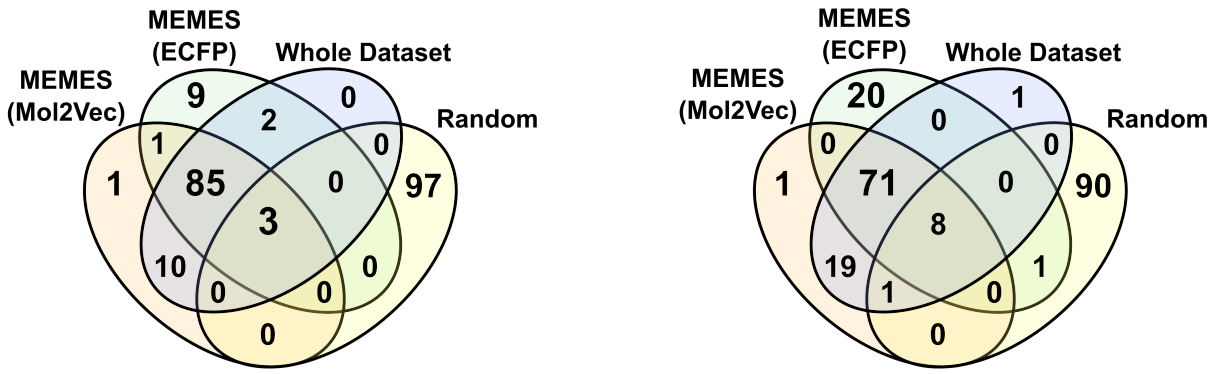
(c) Target Protein: TTBK1

(d) Target Protein: SARS-CoV-2 M^{pro}

Figure 2: Performance on Zinc250K using ExactMEMES against both target receptors. (a) and (b) compares the mean binding affinity of top hits sampled by MEMES and random sampling against mean binding of actual top hits in the library. (c) and (d) show the fraction of top 100 sampled molecules that are actual top hits against the percentage of dataset sampled.

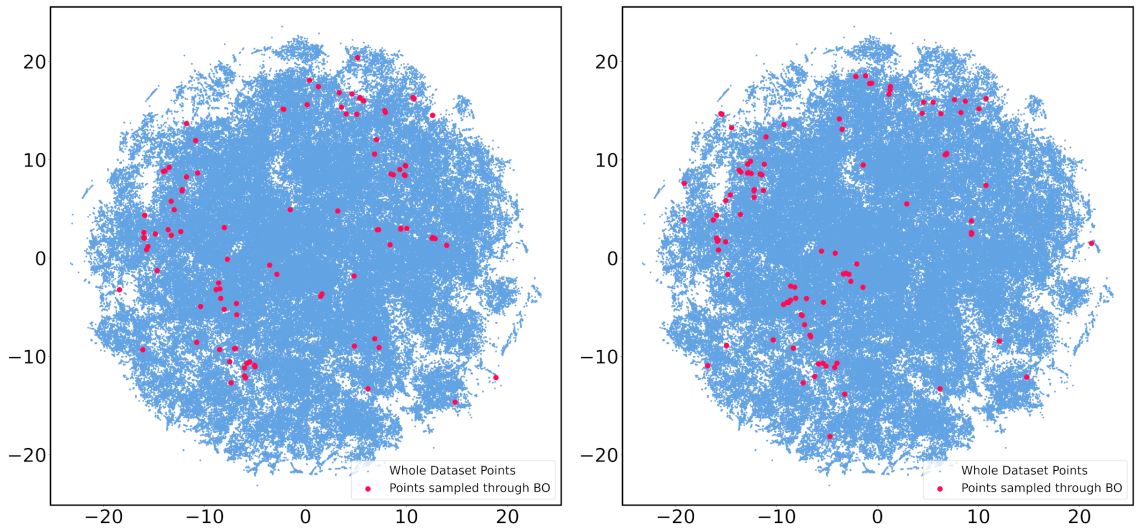
Figure 3 (See Supplementary Fig. S5 for overlap of top 20 and top 500 sampled molecules).

Apart from having a high binding affinity and high overlap with actual top hits, it is also desirable that the molecules sampled by the proposed framework are diverse and the method is not biased towards a certain distribution. To analyze the diversity of sampled



(a) Target Protein: TTBK1 (b) Target Protein: SARS-CoV-2 M^{pro}

Figure 3: Venn diagram showing the intersection of the top 100 molecules identified by different methods and actual top 100 hits of the docking library



(a) Target Protein: TTBK1 (b) Target Protein: SARS-CoV-2 M^{pro}

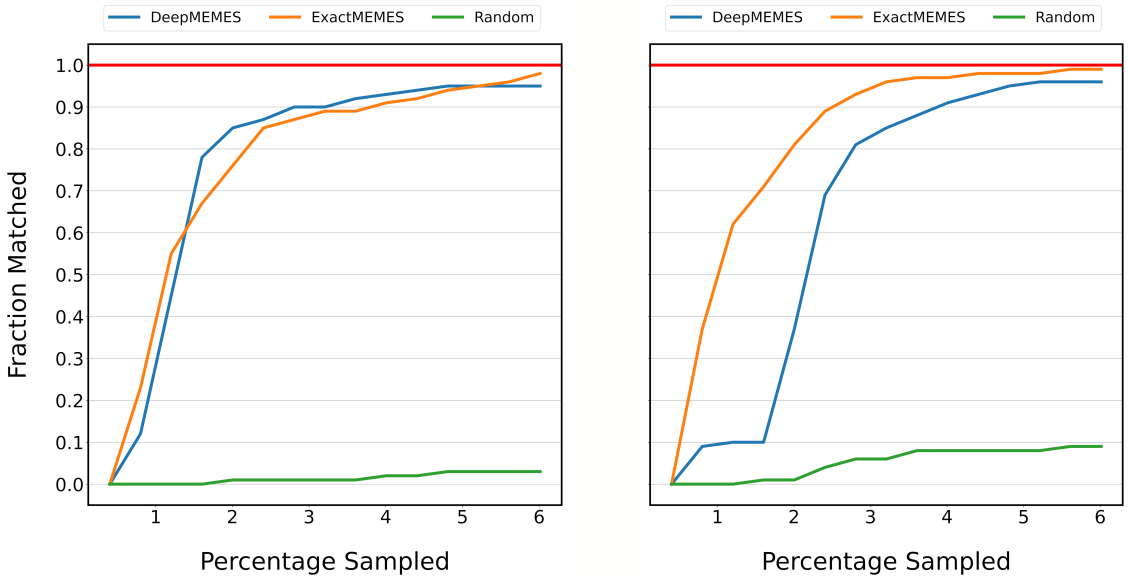
Figure 4: t-SNE plot of top 100 molecules sampled by MEMES framework (red) and complete dataset (blue), to demonstrate the diversity of sampled molecules

molecules, molecules in the dataset (blue) and top 100 sampled molecules (red) using the proposed framework are shown as a scattered plot. For reducing the dimensionality of the molecular embeddings, t-SNE[35] technique was used. From Figure 4 we can see that the sampled molecules marked in red are not confined to a particular region and spread

across the complete chemical space of molecular library. Hence, we can conclusively say that the sampled molecules are diverse. t-SNE plot for top 500 sampled molecules is given in Supplementary Fig. S6.

ExactMEMES vs DeepMEMES

Above experiments shows the ability of ExactMEMES framework to identify top hits only by performing docking less than 6% of complete docking library but it cannot be applied on large docking libraries due to computation constraints. Therefore to overcome this issue, DeepMEMES variant of proposed framework is introduced. In this section, the performance of DeepMEMES is compared against ExactMEMES. Zinc-250K is chosen as molecular library and Mol2Vec as molecular embedding for this comparision.



(a) Target Protein: TTBK1 (b) Target Protein: SARS-CoV-2 M^{pro}

Figure 5: To compare the performance of ExactMEMES and DeepMEMES, fraction of the top 100 molecules sampled that are actual top hits is plotted against the percentage of dataset sampled. Mol2Vec was chosen as featurization technique for comparison.

Figure 5 compares the fraction of the molecules matched with actual top hits of docking library between DeepMEMES and ExactMEMES. From the Figure 5, we can infer that DeepMEMES has comparable performance with ExactMEMES. See Supplementary Discus-

sion 2 for performance of DeepMEMES on Zinc-250K. Further sections show the application of DeepMEMES on different molecular libraries to assess its performance on large datasets.

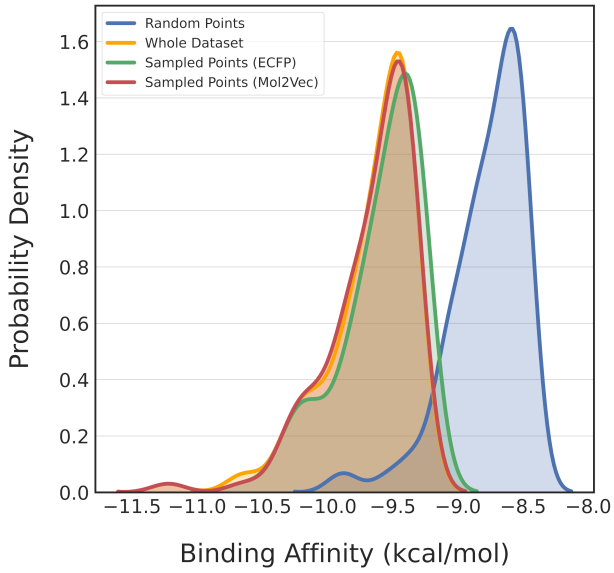
MEMES framework on large libraries

In real life drug discovery experiments, to find a hit against a target receptors, usually ultra large docking libraries are screened. Hence, it is essential to validate the performance of MEMES method on docking libraries that mimic real-life use cases. As, ExactMEMES cannot be applied on large docking libraries due to computational constrains and since both are comparable in performance, DeepMEMES framework performance was demonstrated on two large docking libraries Enamine[36] HTS Collection (2 million molecules) and an Ultra Large Docking Library[28] (96 million molecules).

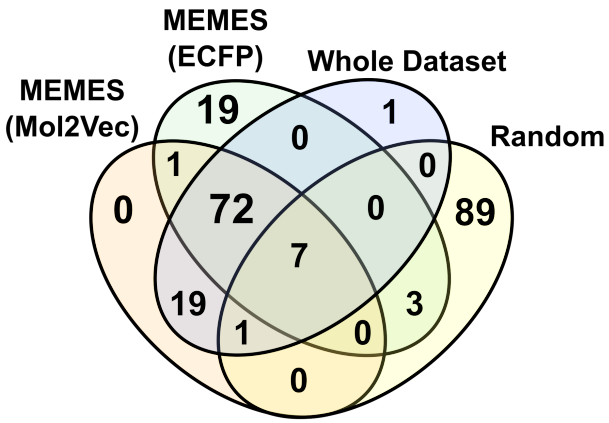
Enamine Dataset

Enamine dataset[36] consists of collections of compounds that are used in virtual screening. Enamine HTS Collection containing 2,106,952 screening compounds was chosen to illustrate the performance of DeepMEMES. DeepMEMES framework is applied on Enamine HTS Collection to demonstrate that the top docking hits can be identified only by docking a small fraction of the complete library against the target receptor TTBK1.

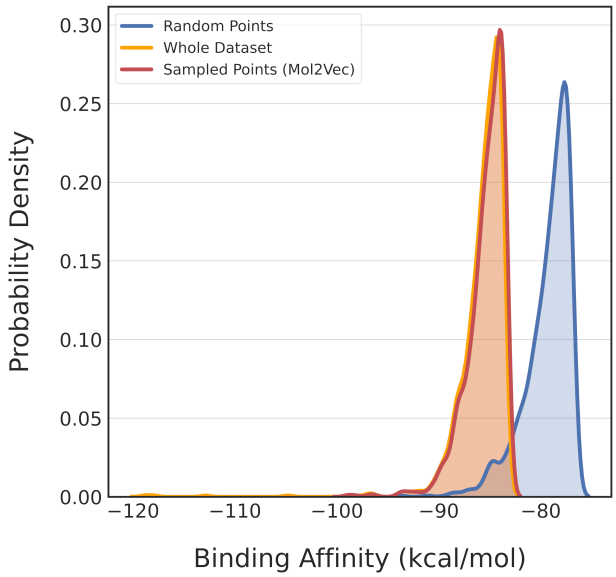
Figure 6a compares distribution of binding affinities and Figure 6b shows the overlap of top 100 molecules sampled using the DeepMEMES framework (using both Mol2Vec and ECFP embedding), random sampling and actual top hits for target protein TTBK1. Similar analysis for top 500 molecules is given in Supplementary Fig. S7. From the Figure 6a and 6b, we can infer that a high percentage of molecules sampled by DeepMEMES matches with the actual top hits by performing only 125,000 docking calculations, which is $\sim 6\%$ of chosen docking library (Supplementary Fig. S8 shows the fraction of top sampled molecules that are actual top hits against the percentage of molecules sampled).



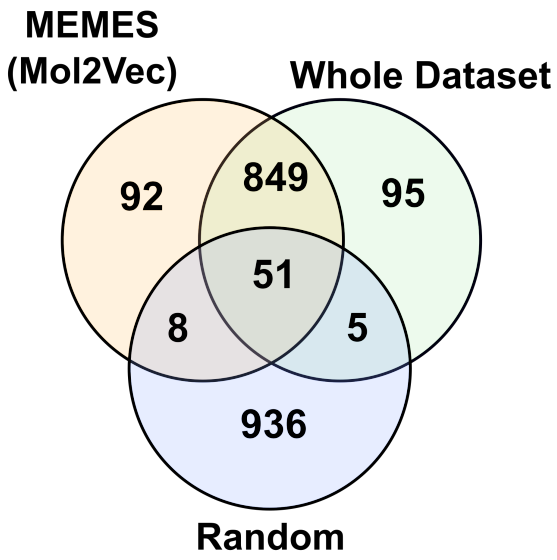
(a)



(b)



(c)



(d)

Figure 6: (a) and (b) show the performance of DeepMEMES on Enamine dataset against target protein TTBK1. (c) and (d) show the performance of DeepMEMES on Ultra Large Docking Library against target protein AmpC. (a) and (c) show the distribution of binding affinities for top 100 molecules and top 1000 molecules sampled by MEMES, respectively. Venn Diagram (b) and (d) demonstrate the overlap of top 100 hits and top 1000 hits identified by different methods, respectively.

Ultra Large Docking Library

In a recent study, Lyu et al.[28] introduced a large compound library containing 96 million molecules. The whole library was docked to find potential molecules against AmpC β-

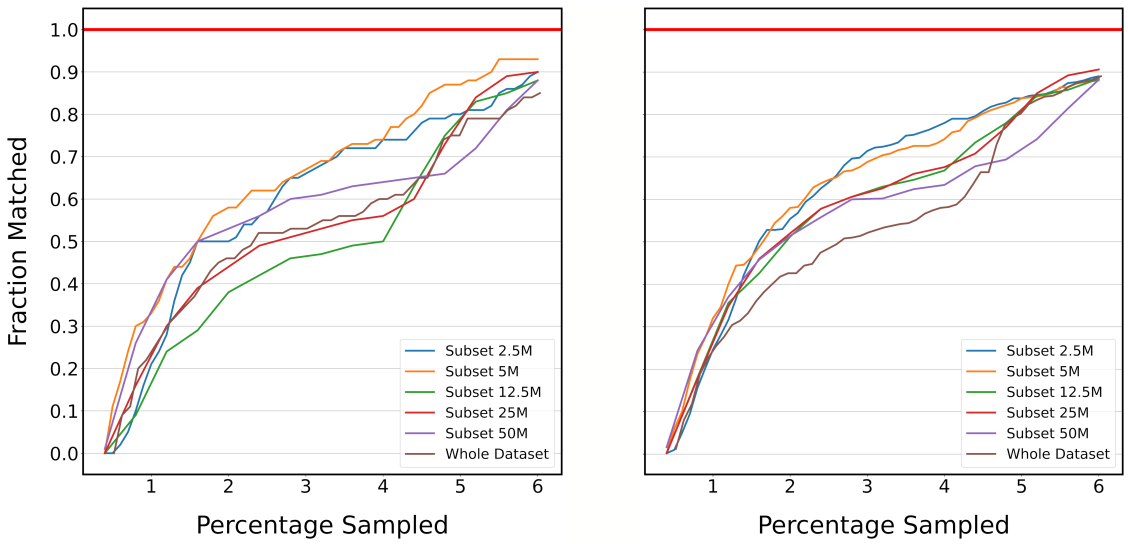
lactamase (AmpC) receptor. DeepMEMES framework is applied to this molecular library to show that top docking hits can be identified by performing docking calculations on a fraction of the complete library. Figure 6c compares distribution of binding affinities, and Figure 6d shows the overlap of top 1000 molecules sampled using the DeepMEMES framework (using Mol2Vec embedding), random sampling, and actual top hits for target protein AmpC. Similar analysis for top 500 and top 5000 molecules is given in Supplementary Fig. S9. From the Figure 6c and 6d we can infer that 90% of molecules sampled by the DeepMEMES framework matches the actual top hits only by performing 5,800,000 docking calculations, $\sim 6\%$ of the complete library (Supplementary Fig. S10 shows the fraction of top sampled molecules that are actual top hits against the percentage of molecules). It is a significant improvement over random sampling where only 5.5% of sampled molecules matches actual top hits.

Effect of docking library size on the performance of DeepMEMES

The previous section shows the application of DeepMEMES on Enamine HTS Collection[36] and Ultra Large docking library[28] for target protein TTBK1 and AmpC, respectively. The purpose of this experiment is to demonstrate the robustness of the proposed framework on docking libraries of varying sizes. K-Means clustering was performed on Ultra large docking library,[28] creating 1000 clusters, and subsets of different sizes ranging from 2 million to 96 million were created by uniformly sampling from each of the resulting clusters. Finally, DeepMEMES performance was assessed on each of the resulting subset.

Figure 7 shows the fraction match of the sampled molecules that matches actual top hits for different docking library sizes. $85\% - 95\%$ of the molecules sampled by the DeepMEMES framework with Mol2Vec featurization matches the actual top hits irrespective of the docking library's size, demonstrating the consistent performance of the proposed framework.

In summary, high throughput virtual screening requires exhaustive evaluation of each molecule in a complete docking library to find potential candidate molecules. In this study, MEMES framework based on Bayesian optimization for efficient sampling of chemical space



(a) Fraction match for top 100 molecules (b) Fraction match for top 500 molecules

Figure 7: Fraction of top molecules sampled by DeepMEMES (with Mol2Vec as featurization technique) that matches with actual top hits against the percentage of the dataset sampled.

for high throughput exercises is proposed. We showcase the MEMES framework application in hit identification, i.e., to sample molecules with high docking scores against target receptors. Two variants of the MEMES framework are introduced, ExactMEMES and DeepMEMES, depending on the choice of surrogate function. Various experiments were performed with Mol2Vec and ECFP as molecular embedding techniques, and with different sized molecular library ranging from 2 million to 96 million to find hit molecules against different target receptors to showcase the efficiency of the proposed framework. MEMES framework was able to identify more than 90% of the actual top hits while only calculating the docking score for about 6% of the complete molecular library showing the robustness of the proposed framework. In this work, MEMES framework application was demonstrated on virtual screening of molecular libraries, but it can also be applied on other screening applications where exhaustive evaluation is infeasible.

Method

In this section, the various components in the proposed framework (Figure 1) are explained. The docking methods, ligand libraries, and target receptors used for the experiments are described in the section Docking Methodology. In the section Molecular Representation, the choice of different molecular embedding techniques used in this work are explained in detail. Further, Bayesian Optimization, the techniques used to approximate protein-ligand scoring function and point selection methods are explained.

Docking Methodology

Molecular docking is useful in drug discovery projects to identify potential inhibitors against a protein receptor from small molecule libraries. The first step is ligand preparation, and protein preparation that was carried out using AutoDock 4.2 (AD 4)[37] in this study. Three different small-molecule libraries of varying sizes were used in this study. First is Zinc-250K dataset used earlier in molecular generation studies[18,21,38] which contains 250,000 drug-like molecules obtained from ZINC15 database.[7] Second is the Enamine dataset[36] containing screening compounds that are grouped into different collections. Enamine HTS Collection containing 2,106,952 molecules is used in this study. The last one is the Ultra Large Docking Library introduced by Lyu et al.,[28] which contains 96 million molecules docked against AmpC β-lactamase (AmpC) receptor. Target proteins, Tau-Tubulin Kinase 1 (PDB ID:4BTK) and SARS-CoV-2 Mpro complexed with N3 inhibitor (PDB ID:6LU7) used in the experiments were obtained from Research Collaboratory for Structural Bioinformatics – Protein Data Bank (RCSB – PDB).[39] The next step in molecular docking is grid map generation carried using AutoGrid 4 utility in AutoDock. Finally, docking calculation was done, keeping the protein active site rigid to get binding affinity. Detailed information about docking methodology is given in Supplementary Methods.

Molecular Representation

The first step in the pipeline is to encode molecules as fixed-dimensional vectors. It is essential to choose encoding methods that effectively encode molecular structures and are sensitive to small changes in configurations. In this work, we experimented with two molecular embedding techniques - ECFP and Mol2Vec.

Extended-connectivity fingerprints (ECFP)

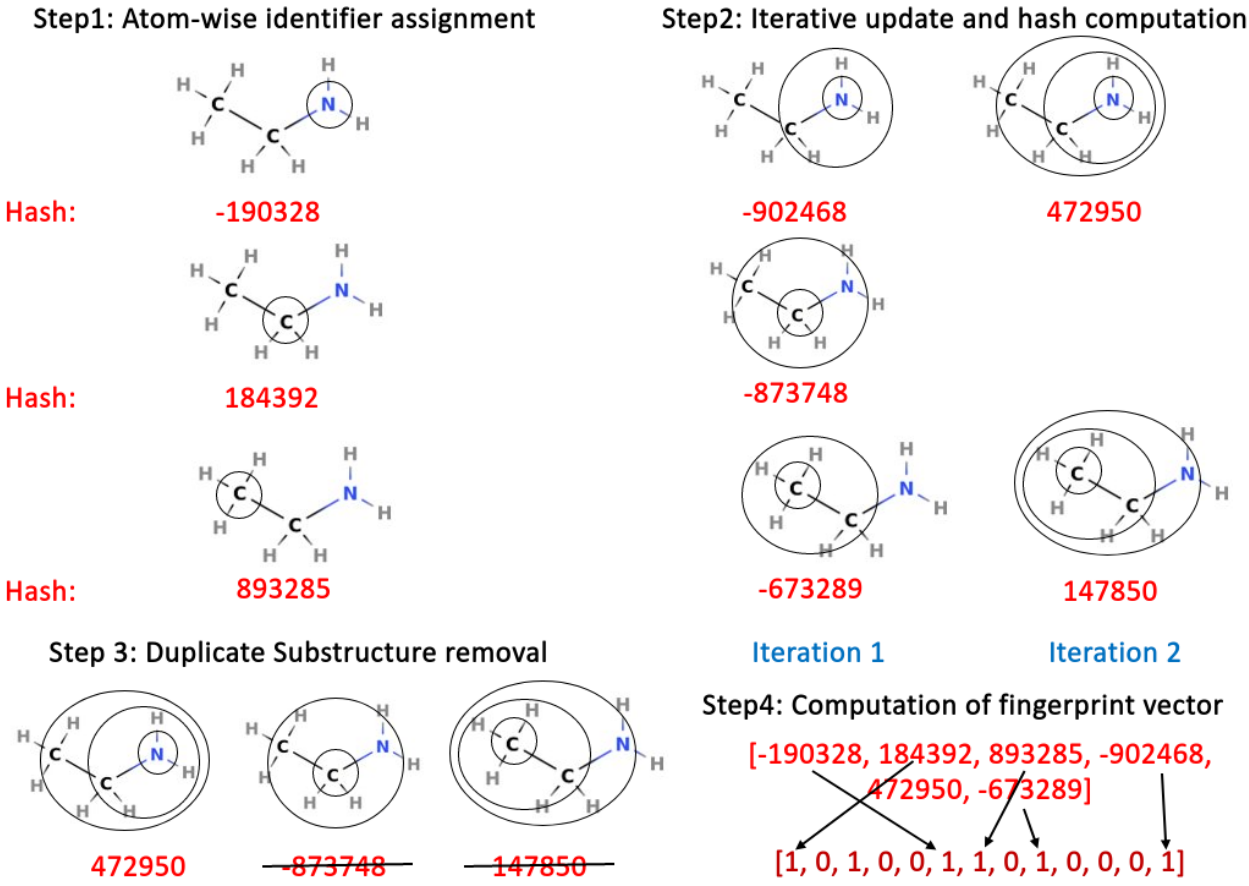


Figure 8: Overview of the ECFP algorithm. A unique integer value is assigned to each atom in Step 1. Step 2 involves iterative updating of the atom identifier. In Step 3 duplicate substructures are removed. Finally in Step 4 substructures are transformed into a bit vector.

Extended-connectivity fingerprints[40] encode molecules into a bit vector, each bit indicative of presence or absence of a specific substructure. A basic overview of the algorithm for fingerprinting is described here. First, each atom is assigned a unique integer value based

on the Morgan algorithm. The atom identifier is augmented with information gathered from neighboring atom and bond information and a unique identifier is obtained. This step is repeated for a desired number of iterations (defined by radius) indicating the depth of the information captured at each atom center. Duplicates are removed in case the same substructure multiple identifiers. The substructures are finally constructed into a bit vector. The algorithm is schematically described in Figure 8.

Mol2Vec

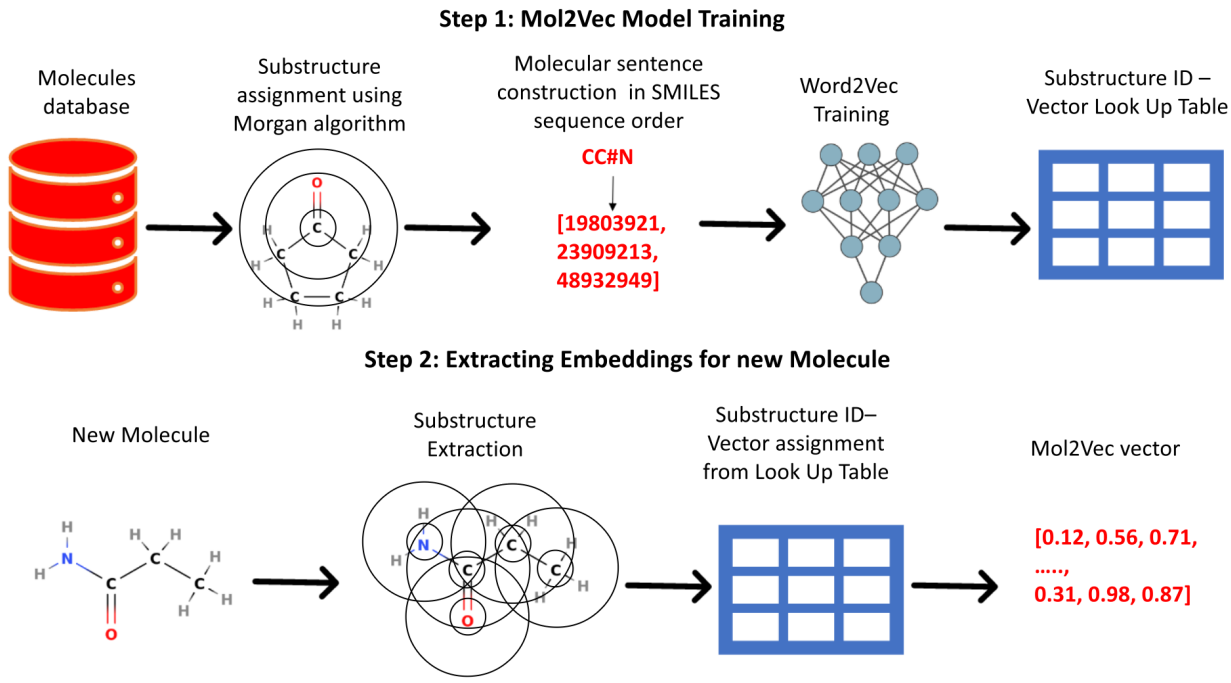


Figure 9: In this figure the procedure to obtain Mol2Vec vectors is depicted. Step 1: Mol2Vec model is pretrained on a corpus generated from a molecules database. Step 2: Mol2Vec vectors is generated for a new molecule using pretrained model.

Mol2Vec[41] is a molecular embedding technique inspired by Natural Language Processing technique, Word2Vec.[42] In the Word2Vec technique, words are encoded as vectors that are representative of semantics through unsupervised machine learning over a large text corpus. The Mol2Vec algorithm extends this methods for application to small molecules. In the Mol2Vec algorithm, substructures are first extracted using the Morgan algorithm at

radii 0 and 1 and a unique identifier is assigned to each of them. Using these identifiers, SMILES sequences of molecules are ordered as sentences, analogous to representing text sentences with words. The Word2Vec algorithm is then used for unsupervised training to construct an identifier-vector look up table. For a new molecule, the embedding is obtained by summing the vectors of all the identifiers in the sentence constructed. Training with Word2Vec algorithm helps tackle the sparse nature that encoding methods such as ECFP have, which makes it easier for their use with ML models. The Word2Vec training helps in contextualizing vectors that are representative of the structures, instead of a single bit value. The Mol2Vec algorithm is described in Figure 9. The Mol2Vec model is trained on ZINC 15. Mol2Vec descriptor has shown to have superior performance on regression tasks such as solubility prediction[43] and toxicity prediction.[44]

Bayesian Optimization

Bayesian Optimization is an optimization technique used to optimize black-box functions that are expensive to evaluate.[17,45] There are two main components in Bayesian optimization, a surrogate function which is a statistical model that can be used to approximate the black box, and acquisition function to determine the next points to sample. In this work, Gaussian Process Regression (ExactGP) and Deep Gaussian Process (DeepGP) are used as surrogate function in ExactMEMES and DeepMEMES variant, respectively, and Expected Improvement[33] is used as an acquisition function.

Gaussian Process Regression (GPR)

Gaussian process regression is a nonparametric Bayesian regression technique. Consider a data set of k points, $x_1, ..x_k$, whose function values are already known, are represented in a vector $[f(x_1), ...f(x_k)]$. In Bayesian statistics, the set of points is assumed to be drawn at random from a prior probability distribution. In a Gaussian process, the prior probability distribution is modelled as a multivariate Gaussian distribution with a mean and a covariance

vector. The prior distribution on the set of points $[f(x_1), ...f(x_k)]$ is given by -

$$f(x_{1:k}) \sim Normal(\mu_0(x_{1:k}), \Sigma_0(x_{1:k}, x_{1:k})) \tag{1}$$

In equation 1 the mean vector is obtained by evaluation of the mean function μ_0 at each point x_i and the covariance matrix is obtained by evaluation of covariance function or kernel Σ at each pair of points x_i and x_j. The kernel function should have a property that the points closer should have strong correlation and the resulting covariance matrix is positive semi-definite. Suppose the prior distribution is constructed for n points. For a point x at $k = n + 1$, the distribution is obtained from Baye's rule -

$$\begin{aligned} &f(x)|f(x_{1:n}) \sim Normal(\mu_n(x), \sigma_n^2(x)) \\ &\mu_n(x) = \Sigma_0(x, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}(f(x_{1:n}) - \mu_0(x_{1:n})) + \mu_0(x) \\ &\sigma_n^2(x) = \Sigma_0(x, x) - \Sigma_0(x, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}\Sigma_0(x_{1:n}, x) \end{aligned} \tag{2}$$

The conditional probability distribution is called the posterior probability distribution. For faster computations, the matrix inversions are obtained through Cholesky decompositions and solving a system of linear equations. In this work, the kernel function is chosen to be Radial Basis Function (RBF).[46] The implementation of Exact Gaussian Processes in GPyTorch[47] are used in this work.

Deep Gaussian Processes (DGPs)

Although Exact Gaussian processes help approximate black-box functions and provide a good estimate of uncertainty, the algorithm has a time complexity of the order, $O(n^3)$. As a result, Gaussian processes cannot be applied when the dataset is larger than a few hundred thousand points. Instead, Deep gaussian processes provide a scalable alternative.

Deep Gaussian Process is a type of Deep Belief Network where every hidden unit is a Gaussian Process. The output of the $l-1^{th}$ layer is used as the input to the l^{th} layer. It can be defined as composition of functions. Formally we can define DGP for training data set of k points $x_1, ..x_k$ whose function values are known represented in a vector y, as

$$\begin{aligned} \mathrm{f}^{(1:L)}(x_{1:k}) &= \mathrm{f}^{(L)}(\mathrm{f}^{(L-1)}(...\mathrm{f}^{(2)}(\mathrm{f}^{(1)}(x_{1:k}))...)) \\ \text{where} f_d^{(l)} &\sim GP(0, k_d^{(l)}(x, x')) \text{ for} f_d^{(l)} \in \mathrm{f}^{(l)} \end{aligned} \tag{3}$$

In Equation 3 L denotes the number of layers. Each layer has their own kernel and the noise between layers is assumed to be independent and identically distributed gaussian, which is absorbed into the kernel $k_{noisy}(x_i, x_j) = k(x_i, x_j) + \sigma_l^2 \delta_{ij}$ where δ_{ij} is the Kronecker Delta and σ_l^2 is the noise between layers.[34] The joint probability distribution for Deep Gaussian Process is given by

$$p(y, \{\mathrm{f}^{(l)}\}_{l=1}^{(L)}) = \prod_{i=1}^{N} p(y_i|\mathrm{f}_i^{(L)}) \prod_{i=1}^{L} p(\mathrm{f}_{(l)}|\mathrm{f}_{(l-1)}) \tag{4}$$

In equation 4 the first term corresponds to likelihood, and the second corresponds to the GP prior. Non linear transformation is applied on the output of every hidden layer due to which exact inference is not tractable.[34] To overcome this problem various number of approximations have been developed such as Expected Propagation,[48] Variational Auto-Encoded Deep Gaussian Processes,[49] and Doubly Stochastic Variational Inference for Deep Gaussian Processes.[50] In this work, Doubly Stochastic Variational Inference is used here. The implementation of Deep Gaussian Processes in Gpytorch[47] is used in this work.

Expected Improvement (EI)

As discussed, in Bayesian optimization, an acquisition function is necessary to determine the next points to be chosen. The acquisition function should be able to choose points that are estimated to have high binding affinity (exploitation), while also exploring unseen/uncertain regions. One such metric, Expected Improvement(EI), that can help balance exploration-

exploitation is used in this work and is described in this section.

Improvement at a point x is defined as -

$$I = max(0, f(x) - f^*) \tag{5}$$

In equation 5 $f*$ is the best function value found so far and $f(x)$ is the value of the function at x. When a Gaussian process is used, $f(x)$ is not a value, but a random variable $\sim N(\mu, \sigma^2)$, where μ and σ correspond to the mean and variance evaluated at point x. The expected improvement is defined as -

$$EI(x) = Exp[max(0, f(x) - f^*)] \tag{6}$$

Using the reparameterization trick, $x = \mu + \sigma\epsilon$ and integrating over the distribution, it can be shown that expected improvement can be obtained as:

$$EI(x) = (\mu(x) - f^* - \zeta)\Phi(Z) + \sigma(x)\phi(Z) \tag{7}$$

where

$$Z = \frac{(\mu(x) - f^* - \zeta)}{\sigma(x)} \tag{8}$$

Here Φ and ϕ are the cumulative distribution function(CDF) and the probability distribution function(PDF) of the standard normal distribution. In equation 7, the first term determines the exploration and second term determines the exploitation. The parameter ζ denotes the amount of exploration during optimization. In this work, ζ is chosen to be 0.01.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Code availability

The codes that support the findings of this study are available from the corresponding author upon reasonable request.

References

(1) Schmidt, H. R.; Betz, R. M.; Dror, R. O.; Kruse, A. C. Structural basis for σ 1 receptor ligand recognition. *Nat. Struct. Mol. Biol.* **2018**, *25*, 981–987.

(2) Lyne, P. D. Structure-based virtual screening: an overview. *Drug discovery today* **2002**, *7*, 1047–1055.

(3) Cheng, T.; Li, Q.; Zhou, Z.; Wang, Y.; Bryant, S. H. Structure-based virtual screening for drug discovery: a problem-centric review. *AAPS J.* **2012**, *14*, 133–141.

(4) McCorvy, J. D.; Butler, K. V.; Kelly, B.; Rechsteiner, K.; Karpiak, J.; Betz, R. M.; Kormos, B. L.; Shoichet, B. K.; Dror, R. O.; Jin, J., et al. Structure-inspired design of β-arrestin-biased ligands for aminergic GPCRs. *Nat. Chem. Biol.* **2018**, *14*, 126.

(5) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B., et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100–D1107.

(6) Blum, L. C.; Reymond, J.-L. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.* **2009**, *131*, 8732–8733.

(7) Sterling, T.; Irwin, J. J. ZINC 15–ligand discovery for everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324–2337.

(8) Pathak, Y.; Laghuvarapu, S.; Mehta, S.; Priyakumar, U. D. Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. Proceedings of the AAAI Conference on Artificial Intelligence. 2020; pp 873–880.

(9) Laghuvarapu, S.; Pathak, Y.; Priyakumar, U. D. Band nn: A deep learning framework for energy prediction and geometry optimization of organic small molecules. *J. Comput. Chem.* **2020**, *41*, 790–799.

(10) Kingma, D. P.; Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* **2013**,

(11) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **2018**, *4*, 268–276.

(12) Pathak, Y.; Juneja, K. S.; Varma, G.; Ehara, M.; Priyakumar, U. D. Deep learning enabled inorganic material generator. *Phys. Chem. Chem. Phys.* **2020**, *22*, 26935–26943.

(13) Dai, H.; Tian, Y.; Dai, B.; Skiena, S.; Song, L. Syntax-directed variational autoencoder for molecule generation. Proceedings of the International Conference on Learning Representations. 2018.

(14) Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. Advances in neural information processing systems. 2014; pp 2672–2680.

(15) De Cao, N.; Kipf, T. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973* **2018**,

(16) Blaschke, T.; Olivecrona, M.; Engkvist, O.; Bajorath, J.; Chen, H. Application of generative autoencoder in de novo molecular design. *Mol. Inf.* **2018**, *37*, 1700123.

(17) Frazier, P. I. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811* **2018**,

(18) Jin, W.; Barzilay, R.; Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364* **2018**,

(19) Griffiths, R.-R.; Hernández-Lobato, J. M. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.* **2020**, *11*, 577–586.

(20) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminf.* **2017**, *9*, 1–14.

(21) You, J.; Liu, B.; Ying, Z.; Pande, V.; Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. Advances in neural information processing systems. 2018; pp 6410–6421.

(22) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843* **2017**,

(23) Yang, X.; Zhang, J.; Yoshizoe, K.; Terayama, K.; Tsuda, K. ChemTS: an efficient python library for de novo molecular generation. *Sci. Technol. Adv. Mater.* **2017**, *18*, 972–976.

(24) Segler, M. H.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **2018**, *4*, 120–131.

(25) Popova, M.; Isayev, O.; Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **2018**, *4*, eaap7885.

(26) Cieplinski, T.; Danel, T.; Podlewska, S.; Jastrzebski, S. We should at least be able to Design Molecules that Dock Well. *arXiv preprint arXiv:2006.16955* **2020**,

(27) Gao, W.; Coley, C. W. The synthesizability of molecules proposed by generative models. *J. Chem. Inf. Model.*

(28) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K., et al. Ultra-large library docking for discovering new chemotypes. *Nature* **2019**, *566*, 224–229.

(29) Tomberg, A.; Boström, J. Can "easy" chemistry produce complex, diverse and novel molecules? **2020**,

(30) Gentile, F.; Agrawal, V.; Hsing, M.; Ton, A.-T.; Ban, F.; Norinder, U.; Gleave, M. E.; Cherkasov, A. Deep Docking: A Deep Learning Platform for Augmentation of Structure Based Drug Discovery. *ACS Cent. Sci.* **2020**,

(31) Hartigan, J. A.; Wong, M. A. Algorithm AS 136: A k-means clustering algorithm. *J R Stat. Soc. Ser. C Appl. Stat.* **1979**, *28*, 100–108.

(32) Rasmussen, C. E. Gaussian processes in machine learning. Summer School on Machine Learning. 2003; pp 63–71.

(33) Snoek, J.; Larochelle, H.; Adams, R. P. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems. 2012; pp 2951–2959.

(34) Damianou, A.; Lawrence, N. Deep gaussian processes. Artificial Intelligence and Statistics. 2013; pp 207–215.

(35) Maaten, L. v. d.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

(36) Enamine. `http://www.enamine.net/`.

(37) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; Olson, A. J. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *J. Comput. Chem.* **2009**, *30*, 2785–2791.

(38) Zhou, Z.; Kearnes, S.; Li, L.; Zare, R. N.; Riley, P. Optimization of molecules via deep reinforcement learning. *Sci. Rep.* **2019**, *9*, 1–10.

(39) RCSB. `https://www.rcsb.org.`, rcsb.

(40) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754.

(41) Jaeger, S.; Fulle, S.; Turk, S. Mol2vec: unsupervised machine learning approach with chemical intuition. *J. Chem. Inf. Model.* **2018**, *58*, 27–35.

(42) Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems. 2013; pp 3111–3119.

(43) Delaney, J. S. ESOL: estimating aqueous solubility directly from molecular structure. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1000–1005.

(44) Challenge, T. D. Tox21 data challenge 2014. 2014.

(45) Pelikan, M.; Goldberg, D. E.; Cantú-Paz, E., et al. BOA: The Bayesian optimization algorithm. Proceedings of the genetic and evolutionary computation conference GECCO-99. 1999; pp 525–532.

(46) Wilson, A.; Adams, R. Gaussian process kernels for pattern discovery and extrapolation. International conference on machine learning. 2013; pp 1067–1075.

(47) Gardner, J. R.; Pleiss, G.; Bindel, D.; Weinberger, K. Q.; Wilson, A. G. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Red Hook, NY, USA, 2018; p 7587–7597.

(48) Bui, T.; Hernández-Lobato, D.; Hernandez-Lobato, J.; Li, Y.; Turner, R. Deep Gaussian processes for regression using approximate expectation propagation. International conference on machine learning. 2016; pp 1472–1481.

(49) Dai, Z.; Damianou, A.; González, J.; Lawrence, N. Variational auto-encoded deep Gaussian processes. *arXiv preprint arXiv:1511.06455* **2015**,

(50) Salimbeni, H.; Deisenroth, M. Doubly stochastic variational inference for deep Gaussian processes. Advances in neural information processing systems. 2017; pp 4588–4599.

Acknowledgement

We thank Professor Brian Shoichet for making the data on docking calculations on AmpC protein available to us. We thank DST-SERB grant (no. CVD/2020/000343) for financial support. This work was partially funded by Intel Corp. as part of its Pandemic Response Technology Initiative (PRTI).

Author contributions

S.M., S.L., Y.P., and U.D.P. conceived the presented idea. A.S., M.A. and S.M. contributed to the process of docking calculation. S.M. and Y.P. performed all experiments and data analysis. S.M., S.L., Y.P., and U.D.P. wrote the manuscript. U.D.P. supervised the project.

All authors reviewed the manuscript. The funders did not have any role in the design, idea, data collection, analysis, interpretation, writing of the manuscript or decision to submit it for publication.

Competing interests

International Institute of Information Technology, Hyderabad has filed provisional patent application for the use of MEMES framework in high-throughput screening exercises, with U.D.P., S.M., S.L., and Y.P. listed as inventors. Provisional Patent Application No.: 202041050608. Application status: Awaiting Complete Specification (Provisional Patent Filed). The funders

Additional Information

Supplementary Information is available for this paper.