Improved Scaffold Hopping in Ligand-based Virtual Screening Using Neural Representation Learning

Luka Stojanović,[†,‡] Miloš Popović,[†,‡] Nebojša Tijanić,[†] Goran Rakočević,[†] and Marko Kalinić[*,†]

[†] *Totient, Inc., Sinđelićeva 9, 11000 Belgrade, Serbia*

[‡] *These authors contributed equally to this work*

E-mail: marko.kalinic@totient.bio

Abstract

Deep learning has demonstrated significant potential in advancing state of the art in many problem domains, especially those benefiting from automated feature extraction. Yet the methodology has seen limited adoption in the field of ligand-based virtual screening (LBVS), as traditional approaches typically require large, target-specific training sets, which limits their value in most prospective applications. Here, we report the development of a neural network architecture, and a learning framework designed to yield a generally applicable tool for LBVS. Our approach uses the molecular graph as input, and involves learning a representation that places compounds of similar biological profiles in close proximity within a hyperdimensional feature space; this is achieved by simultaneously leveraging historical screening data against a multitude of targets during training. Cosine distance between molecules in this space becomes a general similarity metric, and can readily be used to rank order database compounds in LBVS workflows. We demonstrate the resulting model generalizes exceptionally well to compounds and targets not used in its training. In three commonly employed LBVS benchmarks, our method outperforms popular fingerprinting algorithms without the need for any target-specific training. Moreover, we show the learned representation yields superior performance in scaffold hopping tasks, and is largely orthogonal to existing fingerprints. Summarily, we have developed and validated a framework for learning a molecular representation that is applicable to LBVS in a target-agnostic fashion, with as few as one query compound. Our approach can also enable organizations to generate additional value from large screening data repositories, and to this end we are making its implementation freely available at `https://github.com/totient-bio/gatnn-vs`

Introduction

Virtual screening (VS) has long been one of the most popular topics in cheminformatics, with a wealth of reports on novel VS method development, benchmarking, and prospective applications of the methodology.[1–3] Despite increasing commodification of high-throughput compound (HTS) screening, and emergence of novel hit discovery methodologies such as DNA-encoded library selections,[4] the promise of reliable computer-aided hit discovery has made interest in VS remain high. This interest currently seems to be undergoing expansive development, driven in no small part by advances in the field of deep learning.[5–9]

Deep learning has brought about considerable progress in several problem domains. Deep neural networks (DNNs) are often described as general pattern recognition engines, and excel in tasks facilitated by automated feature extraction,[10] where they are able to outperform expert judgment. Chemistry has been no exception to this trend; it has been shown that contemporary deep learning architectures can extract sufficient information even from image depictions of molecules – to yield robust predictors without any explicit information or rules pertaining to concepts such as valence, bonding, electronegativity, etc.[11] As such, deep learning potentially removes the need for any a priori feature engineering, allowing a suitably trained model to learn the best representation for a problem at hand.

This strength of DNNs seems to be ideally matched to one of the biggest challenges in ligand-based VS (LBVS), that being the optimal molecular representation to use.[12,13] In LBVS, one assumes that searching a database for compounds *similar* to a known active (the query) will afford structures that are also likely active. At its very foundation, LBVS thus leverages the "similarity-property principle" that postulates compounds similar in chemical structure ought to have similar biological and/or physicochemical profiles. Yet *similarity* in a chemical sense can have a multitude of meanings.[13] What practitioners typically seek are structurally unique and diverse starting points for further development, which means an effective LBVS method must be able to adopt a similarity definition that highly ranks not (only) close query analogues, but genuinely active compounds that are a "scaffold hop" away from the already known actives.[14]

The pursuit for scaffold hopping in LBVS has seen many molecular representations utilized, ranging from those that leverage only the information contained in the molecular graph (2D methods),[15–18] to more complex molecular embeddings derived from a ligand's three-dimensional structure (3D methods).[19–29] Methods centered around "fingerprinting" a molecule from its molecular graph have been particularly popular in LBVS, and can be seen as the canonical 2D LBVS method. These are simple to use, rely on common distance metrics in comparing two compounds (e.g. Tanimoto distance), they are agnostic to the bioactive conformation of the molecules being compared, and feature competitive performance in VS benchmarks when compared to their (more complex) 3D counterparts.[30–32] Furthermore, several studies have demonstrated that molecular fingerprints do feature scaffold hopping potential,[33–35] supporting their practical applicability in VS workflows.

Notwithstanding the solid performance of existing 2D LBVS methods, most currently utilized molecular fingerprints are the result of algorithms involving some degree of knowledge-guided, or manual feature engineering. The heuristics integrated in fingerprinting algorithms typically embody some high-level chemistry principles, attempting to maximize the information content of the resulting feature vector, so as to consequently boost its performance in a variety of chemical similarity-related tasks. While these approaches can clearly be successful, they always feature a trade-off in assigning importance to certain molecular features, while neglecting others, with this choice hard-coded in the algorithm, and not amenable to problem-specific tuning. As described above, this can ideally be addressed with DNNs, as a bespoke representation can be learned during model training, offering the prospect of improved performance over generic methods.

Here, we report the development of a deep learning architecture and a modeling framework designed to learn an optimal representation of a molecule for its application in LBVS and scaffold hopping. Our approach involves training a neural network once, utilizing large-scale historical HTS data, so that the learned molecular representation optimally discriminates compounds active against the same target – from those that are not. We then apply this network to "fingerprinting" any molecule from its graph representation, and use the resulting vectors for similarity-based searches leveraging common distance metrics. When evaluated on a variety of benchmark datasets, we show our method generalizes to novel targets and novel compounds exceptionally well, outperforming algorithms commonly used in LBVS both in terms of overall enrichment and scaffold hopping potential. The fact our method does not require any additional training to address novel targets makes it a generally applicable alternative to existing LBVS methods, and can be of particular utility in generating additional value from large public and private screening data repositories.

Related work

There have been several contributions published over the past 5 years that are related to the work reported here, either through the methodology employed, or the problem being addressed. We will provide a brief overview of the relevant publications in this section.

Two of the earliest reports of deep learning architectures applied to generating molecular representations from a molecular graph as input were published by Duvenaud et al.[36] and Kearnes et al.[37] Both groups introduced the concept of molecular graph convolutions as a means to learn a representation, tailoring it for property predictions and LBVS, respectively. Although the reported models did not significantly outperform standard fingerprinting approaches, they set the stage for several contributions that would leverage molecular graph convolutions in tackling cheminformatics problems.[38–41] Further frameworks for supervised learning from molecular graphs subsequently emerged, including a generalization of the aforementioned models in the form of message passing neural networks;[42,43] and additional refinements involving inclusion of adaptive attention weights,[44] and combining attention with edge message passing.[45] These models form a relatively diverse family to which our own contribution is closely related.

Although a variety of deep learning applications in cheminformatics have been reported, relatively few deal directly with LBVS. One major reason for this can be found in the fact that deep learning models typically require significant amounts of training data (thousands of data points, at least), which is incompatible with the overall LBVS paradigm. Namely, since any prospective application of LBVS would only be able to benefit from a few known actives, the requirement for a large training set presents a significant limitation. Still, some authors have described methods which involve training or refining a model for target-specific recall of novel hits leveraging datasets of known actives.[46] Others have attempted to address the large training set limitation and improve the applicability of deep learning models in several ways. Altae-Tran et al.[40] leveraged the "one-shot learning" framework, to carry out virtual screening supported by anywhere between 1 and 10 known actives and decoys, each, per target. Srinivas et al.[47] described an alternative approach based on collaborative filtering. Although technically not a deep learning model, but rather an extension of the HTS fingerprint methodology,[48,49] the described method was also reported to facilitate LBVS without the need for a large, target-specific training set. Jaeger et al.[50] reported Mol2vec, an unsupervised approach to generating molecular embeddings inspired by the field of natural language processing. Though Mol2vec requires no additional training to "fingerprint" compounds, every specific task, such as LBVS, requires additional machine learning, with the aforementioned limitations pertaining to size of available training sets. Finally, Winter et al.[51] reported an encoder-decoder architecture whose internal molecular representation was shown to be useful for a variety of cheminformatics tasks, including LBVS.

To the best of our knowledge, thus, our contribution is one of very few where a transferable molecular representation is learned during training to yield a model that can subsequently be utilized to fingerprint any molecule; and where the resulting fingerprints are directly comparable and applicable to LBVS in a target-agnostic fashion.

Results and discussion

Inspired by the successes of fingerprint-based LBVS, and strident advances of deep learning in problem domains involving automated feature extraction, we set out to design a framework and DNN architecture that would enable learning an optimal molecular representation for the problem of 2D LBVS. Our key requirement was that the trained model should serve as a generally applicable fingerprinting tool, such that no additional model training would be required to apply the model to novel targets, thus rendering it equally applicable to LBVS scenarios as the popular circular fingerprints.

The model we developed is founded on the graph attention network,[52] and will be referred to as GATNN in the remainder of the manuscript. The basic building block of the model (Fig. 1) are multi-head attention blocks, each of which yields a feature vector encoding properties of a node (i.e. atom), and its immediate neighborhood. This feature vector is produced by an aggregation of embeddings from all nodes of the respective substructure, whereby the aggregation is performed by weighing each node's "message" by an "attention coefficient", allowing the model to differentially treat every atom neighbor. To illustrate the utility of this approach, one could consider the atomic environment centered at an aromatic carbon substituted with a halogen. While e.g. chlorine and bromine substituents start with different atom embeddings, the attention mechanism assures that the learned node representation of the central aromatic carbon can end up being very similar for both substructures, provided this is the optimal representation given training data. From a high level perspective, this architecture thus provides prerequisites for learning isosteric replacements, without the need to define feature abstractions a priori (e.g. hydrogen bond donor or acceptor, positively ionizable center etc.). However, since attention coefficients are trained, it is pertinent to recognize one set of these weights would be insufficient to capture the complexity of substructures requiring differential weighing. Therefore, the attention blocks feature multiple "heads", each with its own set of attention weights and node embedding transformation functions; the final output of a multi-head attention block is subsequently generated by concatenation. Deviating from the original implementation of the graph attention network, our model also takes into account properties of edges (i.e. bonds), whose embeddings are treated by analogy to that described for nodes, and incorporated into the incrementally updated node feature vector (Fig. 1). For implementation details, the reader is referred to the Experimental section.

While a single multi-head attention block yields node feature embeddings that capture neighborhood information one bond away from every atom, embedding larger and more specific substructures can be accomplished by stacking multiple attention blocks. Similar in spirit to the incremental algorithm underlying extended connectivity circular fingerprints, GATNN features a stack of graph attention layers, with every higher layer integrating incrementally larger substructures in its output. To mitigate issues associated with diminishing gradients and loss of accuracy in such a deep architecture, the connections between layers are implemented as gated residual connections.[53] After the last attention block, the node features are pooled (Fig. 1), resulting in a final molecular embedding consisting of 2,048 floats.

To train the GATNN model, we chose to leverage multitask learning in order to expose the model to a variety of target classes and isoactive chemotypes, maximizing its transferability to novel targets and compounds. However, for this strategy to be efficient, one

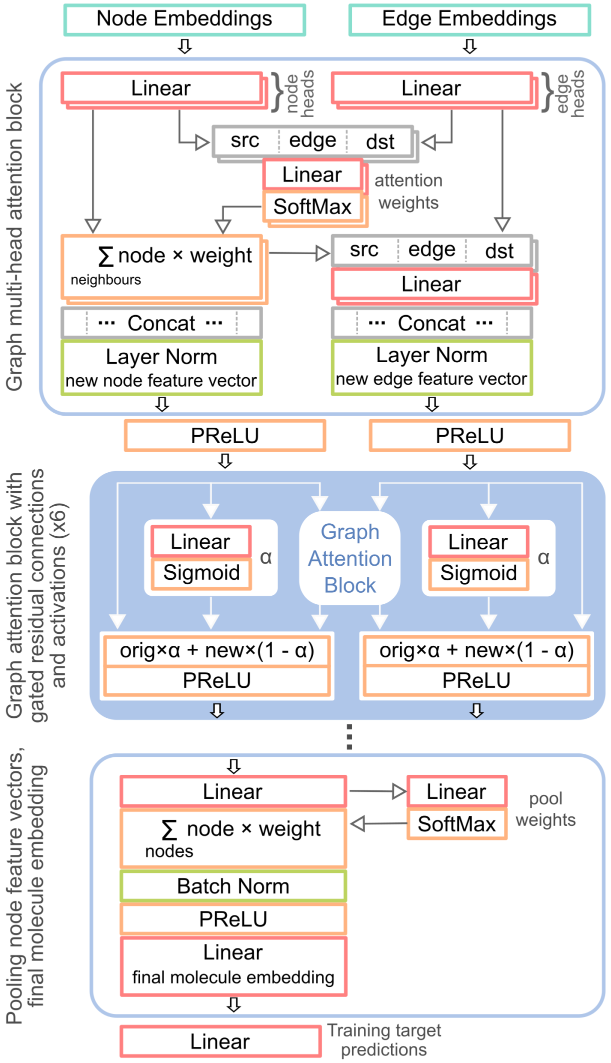


Figure 1: A summary representation of the neural network architecture underlying the model developed in this study.

typically needs to leverage a dense activity table where a given compound library has been evaluated across a range of different targets. Publicly available historical HTS data lend itself as the optimal choice, having been previously demonstrated as useful in engineering biological activity fingerprints.[48,49] In the training regime we used, the model was tasked with predicting a biological activity vector across 235 assays, equivalent to 235 classifier tasks. This approach essentially favored learning a molecular representations that embeds isoactive compounds (i.e. compounds active at the same target or targets) close in the hyperdimensional embedding space that we sought to use as the GATNN fingerprint. This is illustrated in Fig. 2, which shows a clear shift in similarity distribution of active compound pairs after training.

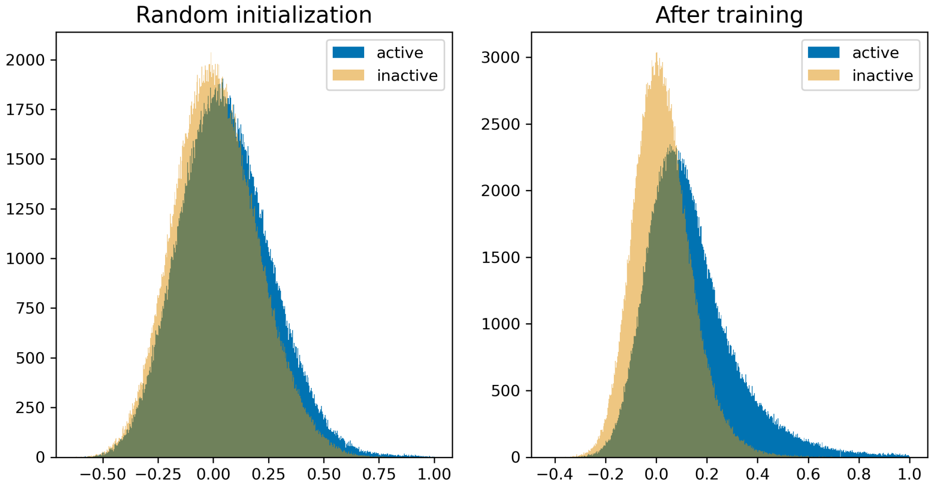


Figure 2: Distribution of cosine similarities between GATNN fingerprints of randomly sampled compound pairs that are, and are not active against the same target, respectively. The right-hand side plot shows a clear shift in distribution of active pair similarities, following 29 epochs of training.

After training, we removed the final classifier layer from the network, and used the molecular embedding layer as the GATNN fingerprint. This provides for a straightforward way to apply GATNN in LBVS, using a flow identical to one applied with other fingerprint-based approaches: fingerprints of database compounds can easily be generated by a simple forward pass of the network, subsequently stored and compared to queries, making the entire process highly efficient. The similarity comparisons of GATNN fingerprints – in all benchmark experiments – were performed using cosine similarity, as it was seen as the most suitable similarity metric for high-dimensional, real-valued vectors. We also attempted using generalized Tanimoto similarity,[48] and observed comparable, though slightly worse performance (results not shown).

We evaluated the performance of GATNN using three external datasets (MUV, ChEMBL, DUD-LBVS) that are commonly employed to benchmark LBVS methods, as proposed by Riniker and Landrum.[54] None of the targets or compounds represented in these datasets were used in model training. We compared the performance of GATNN to three fingerprint-based methods, namely an extended-connectivity circular fingerprint (ECFP4),[18] the topological torsion (TT) fingerprint,[16] and MinHash fingerprint (MHFP6);[55] the choice was motivated by the fact the first two consistently gave best results in previous studies utilizing the same benchmarking data,[54] while the MHFP6 is a more recent development with other useful properties such as fast approximate nearest neighbor searches. The feature count-based fingerprint (ECFC0) was included as a random enrichment baseline. We looked at three metrics: area under the receiver operating characteristic (ROC) curve (AUC); enrichment factor at 1% of the ranked dataset (EF 1%); and Boltzman-enhanced discriminator of ROC (BEDROC). As summarized in Fig. 3 and Table S1, although significant differences can be observed across the three datasets, GATNN consistently outperforms existing methods. The performance boost seen with GATNN is greatest in MUV and ChEMBL datasets, both in terms of AUC, and in metrics of early enrichment. Specifically, GATNN yields 36% and 13% improvement in BEDROC values in MUV and ChEMBL benchmarks, respectively, when compared to the best performing standard method. This finding was very encouraging as MUV is typically seen as a difficult benchmark, given that it features significant scaffold diversity, and thus intrinsically tests a method's scaffold hopping capability. Indeed, while ECFP4, TT, and MHFP performed close to random for several MUV targets (Fig. S1), GATNN exhibited robust enrichment across the benchmark. Conversely, GATNN was modestly outperformed by all three comparator fingerprints in terms of early enrichment in the DUD-LBVS benchmark, despite yielding a higher AUC. We postulated limited structural diversity of actives in DUD-LBVS (as illustrated in Fig. S2) was essentially erasing any advantages GATNN has over the other methods, the former being more attuned to scaffold hopping tasks.

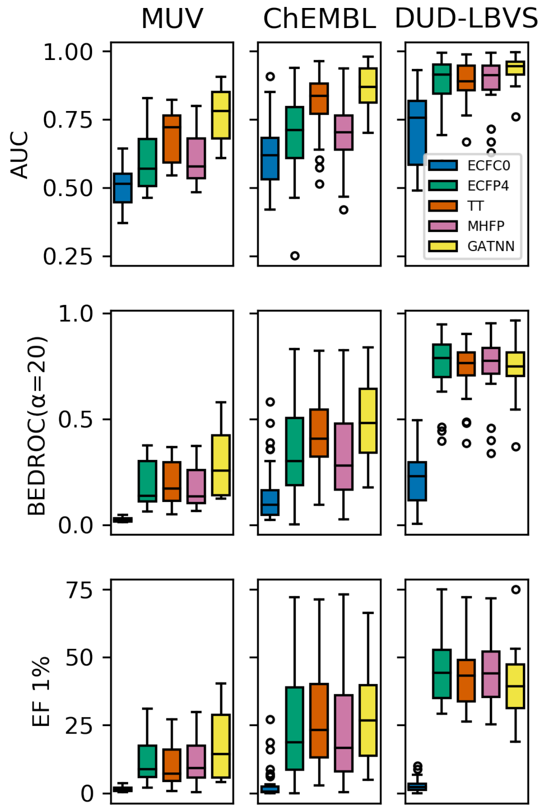


Figure 3: Summary performance of GATNN in three benchmarks (MUV, ChEMBL, DUD-LBVS), compared to well established fingerprinting methods (ECFP4, TT, MHFP), and a feature count-based fingerprint (ECFC0), included as a random enrichment baseline. Boxes edges correspond to 1st and 3rd quartile of the distribution of values obtained across all benchmark targets, with 10 benchmarking runs per target, and random selection of 5 query compounds on each run. Median value is denoted by a horizontal black dash, with minimum and maximum values given by the edges of whiskers. Outliers are shown as empty circles.

To investigate this further, we next looked at the performance of these three methods in scaffold enrichment. While there is no universally accepted scaffold definition to leverage in such an exercise, we opted to use Bemis-Murcko scaffolds (BMS),[56] and examine how varying the number of unique BMS in the query set affects the resulting BMS enrichment. The results are illustrated in Fig. 4. Unsurprisingly, increasing the number of query compounds from 1 to 5 consistently leads to doubling of scaffold enrichment, both across methods and benchmark sets. Likewise, GATNN outperforms the other two methods irrespective of query numbers, though differences are most pronounced in the MUV benchmark, and negligible in DUD-LBVS. Of practical relevance in low-data settings, with one query molecule GATNN yields a mean BMS enrichment factor of 7.33, whereas the second-best method, TT, gives a factor of 4.99 (MUV benchmark); the 47% increase can mean several additional scaffold hits in a prospective application. Notwithstanding the fact BMS enrichment might overestimate genuine scaffold hopping potential, overall, our results indicate GATNN features improved capabilities in this respect.

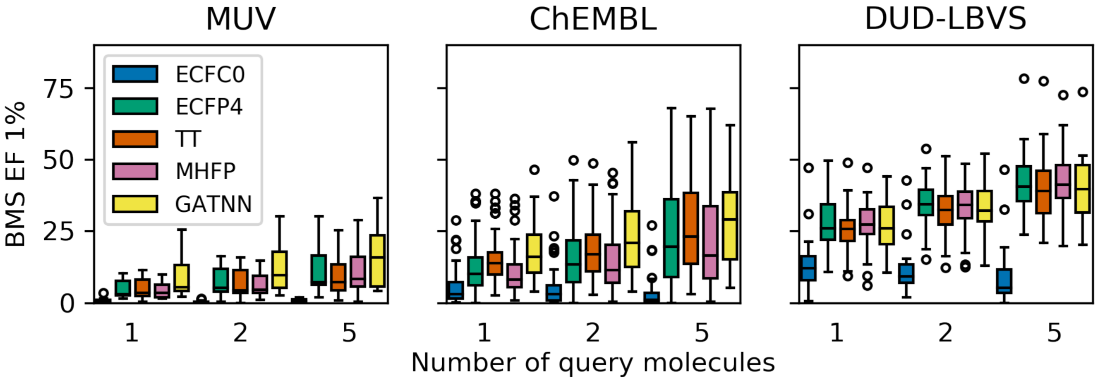


Figure 4: Scaffold hopping performance of GATNN, ECFP4, TT and MHFP, and its dependence on the number of distinct scaffolds in the query set. Performance was quantified by the enrichment of unique Bemis-Murcko scaffolds (BMS) in top 1% of the database, across three benchmark sets (MUV, ChEMBL, DUD-LBVS).

We also sought to understand to what degree are molecular embeddings generated by GATNN similar to existing fingerprints as, theoretically, the network could have largely been reproducing e.g. ECFP4, if that was already the optimal representation for LBVS tasks. We sampled 10,000 random pairs of molecules from the benchmark data and compared their similarities using GATNN and ECFP4, respectively. Interestingly, we observed virtually no correlation between molecular similarities quantified through these two representations (Pearson $r = 0.130$, Spearman $\rho = 0.148$), as illustrated in Fig. 5. This clearly indicates that, although both methods are able to enrich for actives in a LBVS campaign, they capture features translating to chemical similarity in a distinct fashion. This observation is not limited to random pairs of molecules, though we found the correlation to be stronger (Pearson $r = 0.701$, Spearman $\rho = 0.566$) when focusing only on similarities of molecules both active at the same target (Fig. S3). Summarily, this demonstrates GATNN yields a genuinely novel molecular representation, with significant orthogonality to ECFP4, at least.

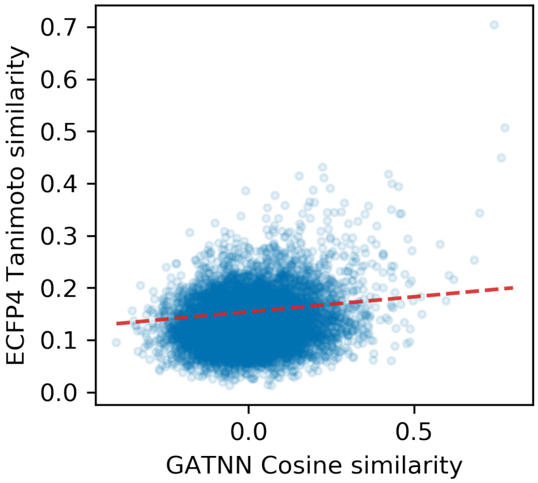
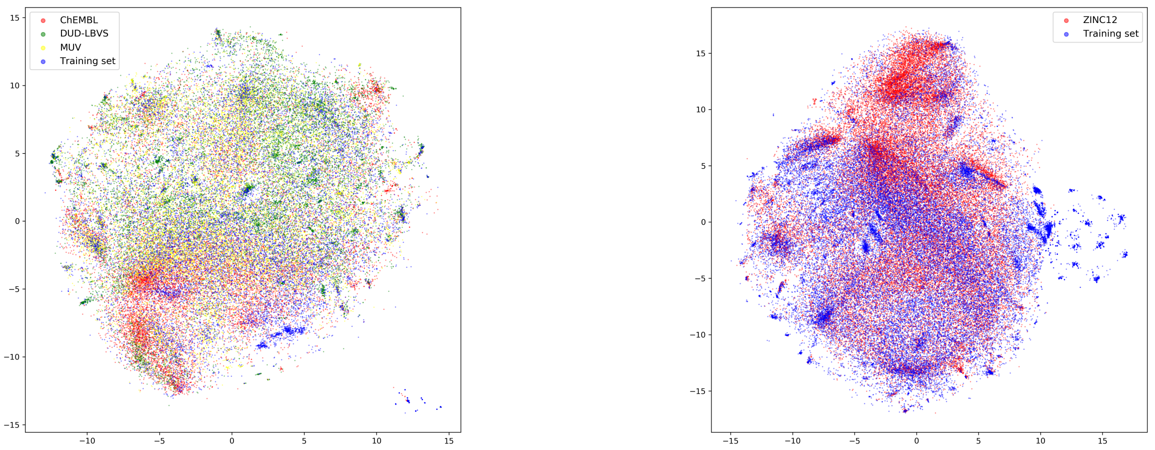


Figure 5: Correlation between Tanimoto similarity of ECFP4 fingerprints and cosine similarity of GATNN embeddings for 10,000 random pairs of molecules. Dashed red line represent best linear fit ($r = 0.130$).

As outlined in the Introduction section, the number of previously reported models that are readily comparable to the approach described in this manuscript is limited. The one-shot learning framework reported by Altae-Tran et al. features a similar degree of flexibility in terms of not requiring a large training set to support LBVS. The best performing model from this publication was reported by the authors to achieve a ROC-AUC of 0.663 for the MUV dataset in task hold-out validation with a support set of 5 active and 10 inactive examples; by contrast, GATNN scores an AUC of 0.763 with 5 active query compounds. Similarly, the model reported by Winter et al.,[51] leveraging an encoder-decoder architecture, achieves an AUC of 0.679 in the MUV benchmark. Looking beyond deep learning models, several newer fingerprinting algorithms have recently been described that provide a new baseline for comparing the performance of machine learning approaches. The best performing variations of connected subgraph fingerprints[57] were reported to achieve an AUC of 0.671, 0.814, and 0.900, in MUV, ChEMBL, and DUD benchmarks, respectively; GATNN scores 0.763, 0.866, and 0.931, although the last figure is not fully comparable, given our choice to use DUD-LBVS. Similarly improved performance was observed over MHFP6,[55] as illustrated in previous sections. The performance of the GATNN-based LBVS framework reported here, thus, compares favorably to related approaches, both conventional and those leveraging deep learning methodologies.

Although we determined GATNN features good comparative performance across relevant benchmarks, a prospective application of the method would typically involve LBVS of compound databases featuring a significant degree of structural diversity. We, therefore, looked at the distribution of training data in chemical space to garner a qualitative appreciation of the model's applicability domain (AD). We do note this analysis should be interpreted cautiously, given the AD of this type of model also needs to be considered in terms of target space coverage, as will be discussed later. As shown in Fig. 6, the breadth of chemical space captured by the training data roughly matches the diversity of the three benchmark sets, suggestive of the fact the estimated performance presents an interpolation scenario for the application of the model. Encouragingly, though, solid coverage of ZINC12[58] is also evident, albeit the model has seen limited exposure to some parts of the respective chemical space. Overall, these findings suggest our model, leveraging solely publicly available training data, would be well suited for screening a commonly employed VS database, from a chemical space AD perspective. It should be noted that while this analysis was based on GATNN embeddings, we made identical observations looking at projections derived from ECFP4 (Fig.

S4).

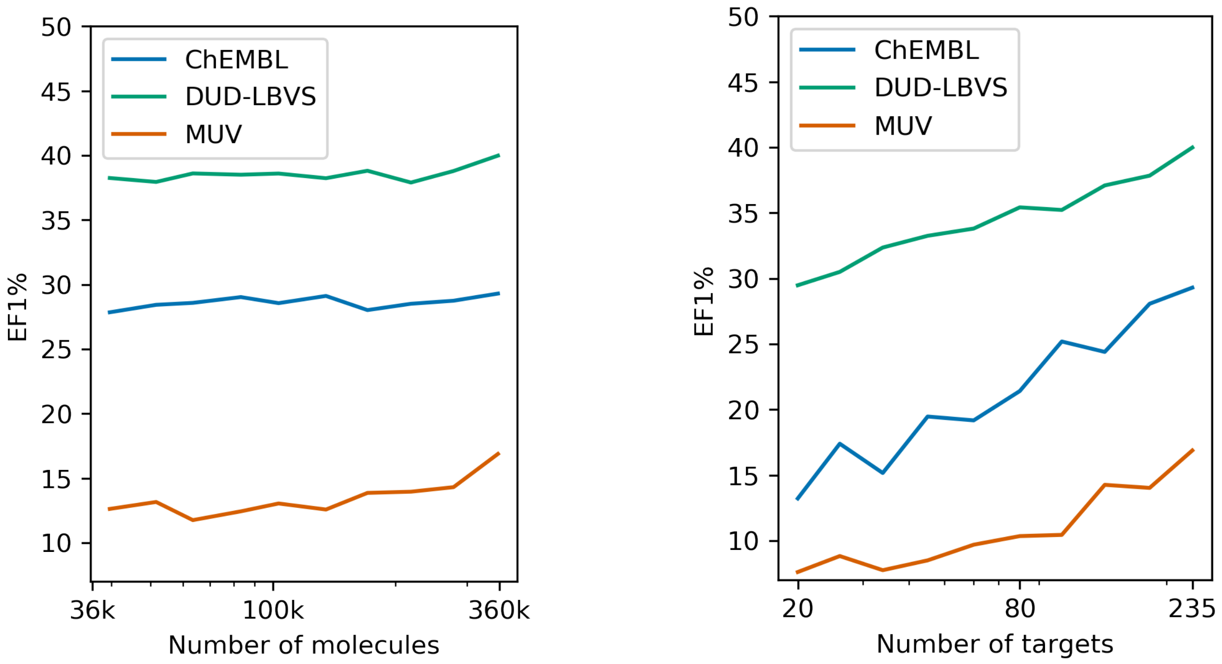


(a) GATNN training set vs. the three benchmark sets employed in this study, each randomly sampled to 12,000 compounds for visualization purposes.

(b) GATNN training set vs. ZINC12, each randomly sampled to 40,000 compounds.

Figure 6: Visualization of various datasets in chemical space, obtained by t-distributed Stochastic Neighbor Embedding (tSNE) of GATNN fingerprints. The relatively uniform distribution of training set compounds with respect to benchmark sets and ZINC12, respectively, suggests the model has been exposed to meaningfully diverse chemical structures, supporting its utility in prospective applications. tSNE embeddings were generated using the openTSNE library,[59] with perplexity set to 2000 and 500, for data presented in subfigure a) and b), respectively. Optimization was carried out for 1000 iterations, with cosine distance between GATNN fingeprints as the distance metric.

Finally, we sought to understand to what extent can the model's performance be improved by enhancing the training set with additional compounds, and more targets, respectively, while gaining a further perspective on its applicability domain. To do so, we performed random down-sampling of the original training data, and repeated the training process from a newly initialized network for each reduced training set. Surprisingly, we observed relatively modest performance gains with the inclusion of additional compounds, whereas more significant performance boosts were seen as the number of targets in the training set are increased (Fig. 7).



(a) EF1% scores on three benchmark datasets as function of training set size for a fixed number of 235 targets.

(b) EF1% scores on three benchmark datasets as function of the number of target labels for a fixed training dataset size of approximately 360,000 molecules.

Figure 7: Influence of training set size and diversity on model performance in external benchmarks. While only modest improvements are seen with inclusion of more compounds, adding data points from additional targets (assays) yields consistent and significant benefits.

In both scenarios of training dataset enlargement, performance in the MUV benchmark benefits the most, which seems to indicate scaffold hopping tasks are where the GATNN framework derives the largest advantage in leveraging historical screening data. Consistent with this, we also observed our model features a strong inductive bias,[60] as evidenced by the fact we can observe better-than-random enrichment across all 3 benchmarks, without a single epoch of training (Fig. S5). In essence, the architecture we developed, without any training, behaves not unlike a circular fingerprinting algorithm with a random hash function, providing baseline performance comparable to such methods; this is promising as it can be seen as a worse case scenario for the model's performance, when it's faced with unseen chemical matter or a novel target class. With training, on the other hand, the model learns a more nuanced representation that can differentially weigh feature importance, and reflect bioisostere-like equivalence in substructures, instead of simply accounting for presence of specific moieties. We hypothesize this also presents the basis for larger improvements seen with the addition of more targets to the training data (as opposed to more compounds), given that the model becomes exposed to more diverse examples of "isoactive" compounds when more targets are involved. From a practical perspective, these results also suggest the GATNN framework may be of significant utility in generating additional value from large corporate screening data repositories, as with additional training data – the performance reported herein could be easily surpassed.

The computational requirements of generating GATNN fingerprints are relatively modest: fingerprinting 1.7 million compounds from ChEMBL25 requires 9 minutes of GPU time on an Nvidia 2080Ti card. For comparison, generating ECFP4 fingerprints for the same dataset using RDKit takes 6 minutes on a single core of AMD Ryzen 9 3900X. GATNN fingerprints can also be generated using CPUs alone, though the resulting performance is far worse, with 5 hours required for fingerprinting ChEMBL25. Screening this dataset against a single query molecule takes approximately half a second when using RDKit's explicit bit vector representation of 2,048 bit ECFP4, and its C++ implemented Tanimoto similarity function

(*BulkTanimotoSimilarity*). On the other hand, performing the same screen with GATNN fingerprints and SciPy's[61] implementation of cosine distance takes approximately 3 seconds. Overall, while GATNN fingerprint-based flows do require more computational time, their overall performance is adequate for routine application in virtual screening flows.

Experimental

Datasets

Model training was carried out using publicly available HTS data consisting of 256 assays previously curated by Helal et al.,[49] and used in Riniker et al.[48] Of these, 14 assays were removed due to target overlap with the benchmark sets employed in our study. Additional 7 assays were eliminated because they were outliers in number of active molecules: 3 assays had more than 10,000 actives and 4 had less than 50. Full records for the remaining 235 assays were retrieved from PubChem[62] and processed to extract substance identifiers (SIDs) and the corresponding bioactivity outcomes; any outcome other than "active" or "inactive" (PubChem XML schema values of 2 and 1, respectively) was treated as ambiguous and not used in model training. Structural records (SMILES) for the full set of SIDs were then retrieved, desalted, and an InChI key was generated for each compound using RDKit.[63] Activity records of SIDs mapping to a single InChI key were merged using the following rules: (*i*) if conflicting outcomes were reported (i.e. both "active" and "inactive"), the outcome was treated as ambiguous; (*ii*) if all non-ambiguous records were concordant, the outcome was set to the corresponding value. Following deduplication and activity record merging, the dataset was filtered to remove: (*i*) compounds that were found to be active in more than 10% of assays in which they were tested; (*ii*) compounds that were represented in either one of the benchmark sets used in our study, as assessed by matching InChI keys. Finally, we observed that this dataset includes more than 45,000 close analogues (3-methyl-3,4,5,6-tetrahydro-2H-1,5-benzoxazocine and 3-methyl-2,3,4,5,6,7-hexahydro-1,5-

benzoxazonine derivatives, mainly) that we downsampled to 5,000 in order to avoid biasing the training dataset. The final set consisted of 353,390 compounds, totaling 56,931,860 data points, with the majority of included compounds evaluated in more than 200 assays (61.4%).

Model performance was evaluated using three external datasets, similarly to a previously proposed framework,[54] but with some modifications. The first benchmark set utilized was Maximum Unbiased Validation (MUV),[64] consisting of 17 targets, with 30 actives and 15,000 decoys each; no modifications were made to the original dataset, as distributed in ref.[54] The second benchmark set, referred throughout the manuscript as ChEMBL benchmark, was generated to include 50 targets proposed in Heikamp and Bajorath.[65] The procedure used to prepare the ChEMBL benchmark was identical to that described in ref.,[54] except that it was carried out using release 25, instead of release 14 of the ChEMBL database.[66] The resulting dataset consisted of 100 actives for each target, and 10,000 decoys. The third set included in the evaluation workflow was DUD-LBVS,[67] from which all the targets with less than 30 active examples were removed, as was done in.[54] The filtered dataset consisted of 21 targets, with between 31 and 365 actives, and between 1,058 and 15,560 decoys per target; the average number of decoys per active was 26. We opted to use DUD-LBVS instead of DUDe, given that DUDe features a significant target overlap with the ChEMBL benchmark, thus providing more orthogonality between the three benchmarks.

Model architecture and training

We developed the model based on the graph attention network[52] architecture. Our additions to that architecture are usage of gated residual connections[53] and feature vectors on edges as well as vertices. The model operates on molecular graphs, vertices being the atoms, and edges being the bonds. Size of the vertex and edge feature vectors are equal in each layer. The network was implemented in the PyTorch[68] framework, using the Deep Graph Library,[69] which requires all graph edges to be directed. We therefore represent each bond as a pair of graph edges (one in each direction).

The network consists of multiple graph attention blocks, each block consisting of multiple primitive neural network layers and operations. The result of every block is an updated vertex and edge vector representation. The final model is made up of an embedding layer, seven graph attention blocks, a pool block and, for training, fully connected layer for target activity prediction. Embedding size for both the nodes and the edges in each residual layer is 512, and the number of heads is 16 (each head accounting for 32 parameters). The output sizes of the two final fully connected output layers were 2048 and 235, for the molecular embedding and assay-specific classifiers, respectively.

In addition to the molecular graph structure, as represented in RDKit's molecule objects, the model uses a number of atom and bond features. For atoms we use: atom type, number of bonded hydrogen atoms, and chirality. For edges we only use bond type.

A graph attention block is implemented through following procedure:

1. Fully connected layers are applied to vertices and edges:

$$\beta^b = W^b_{vertex} v^{b-1}, \qquad \epsilon^b = W^b_{edge} e^{b-1}$$

   where W^b_{vertex} and W^b_{edge} are block-specific weight matrices multiplying the vertex vector v^{b-1} and edge vector e^{b-1} from the previous block, while β and ϵ are intermediate values used throughout the block b.

2. Attention based convolutional update applied to each node:

$$\alpha^b_{ij} = softmax_j(\sigma_A(A^b[\beta^b_i \parallel \epsilon^b_{ij} \parallel \beta^b_j]))$$

$$v^b_i = \sum_j \alpha^b_{ij} \beta^b_i \epsilon^b_{ij}$$

   where A^b is the block-specific attention weight matrix, $\beta^b_i \parallel \epsilon^b_{ij} \parallel \beta^b_j$ is concatenation of vertex-edge-vertex triplet of vectors and σ_A is a leaky rectified linear unit with the slope set to 0.2. We have also employed the multi-head extension to the attention mechanism, as described in the original graph attention paper,[52] so the final vertex feature vector is obtained by concatenating results of K independent attention mechanisms:

$$v_i^b = \Big\|_{k=1}^{K} \sum_j \alpha_{ijk}^b \beta_{ik}^b \epsilon_{ijk}^b$$

3. For the edge update, a fully connected layer is applied to a concatenated triplet consisting of feature vectors of the edge and the two adjacent vertices:

$$e_{ij}^b = W_{triplet}^b [v_i^b \parallel \epsilon_{ij}^b \parallel v_j^b]$$

   where $W_{triplet}^b$ is the block-specific weight matrix multiplying the concatenated vertex-edge-vertex vector.

4. At the end of each block we apply layer normalization[70] to vertex and edge feature vectors.

Blocks are followed by a gated (parametric) residual connection,[53] on both the vertex and the edge feature vectors. Since residual connections require that a dimensionality of input and output are equal, they are added from the block two onward. The outputs of the residual values are passed through parametric rectified linear units (PReLU).[71]

Finally, vertex feature vectors are pooled, in order to provide unified molecule representation. Weighted average pool is used, with weights derived in a process similar to graph attention:

1. Fully connected layers applied to vertices

$$\beta^m = W_\beta^m v^{last}$$

2. Weights are calculated using this vector and used for summation:

$$\alpha^m = softmax(\sigma_A(A\beta^m))$$

$$\mu' = \sum_i \alpha_i^m \beta_i^m$$

where σ_A is again a leaky rectified linear unit with the slope set to 0.2.

3. Batch normalization[72] and PReLU activation is applied

$$\mu = \sigma_P(batchnorm(\mu'))$$

4. Another fully connected layer is applied to create a final molecule embedding

$$m = W_\mu^m \mu$$

We trained the model using an AdamW optimizer.[73] The training was done using a mini-batch size of 256 molecules and a learning rate of 0.001.

The model was regularized using weight decay set to 0.02. Final model parameters are derived by running Stochastic Weight Averaging[74] over the models from the final 10 epochs of training.

The training procedure takes around 4 GPU hours on a system with 2080Ti cards.

Model evaluation

A GATNN fingerprint was generated for every compound from the benchmark datasets by a forward pass through the trained model, taking the final (embedding) layer as a suitable molecular representation. Cosine similarity was used as metric for comparing GATNN fingerprints. We found that the best results are achieved when feature vectors are standardized using the statistics calculated on benchmark dataset molecules before calculating cosine. Comparator fingerprints were generated using RDKit.[63] These included the RDKit implementation of extended connectivity circular fingerprints[18] with a radius of 2 bonds (ECFP4), and topological torsion fingerprints (TT).[16] MinHash fingerprints (MHFP) were generated using the implementation distributed by the authors,[55] with a radius of 3 bonds. ECFP4 was folded to 2048 bits, whereas TT and MHFP were represented by 2048 integers. Each fingerprint was compared using a different similarity function, as proposed to give best LBVS performance: ECFP4 was compared using Tanimoto, TT using Dice, and MHFP using Jaccard similarity.

For every target from each of the three benchmarks, 5 active compounds were randomly sampled without replacement, and were subsequently used as queries against the remainder of the database. To obtain a ranked list of database compounds, the MAX fusion approach[75] was used, meaning the maximum similarity to any one of the 5 queries was taken as the final similarity measure. This ranked list was used as input for calculating area under the receiver-operating characteristic (ROC) curve (AUC), Boltzman-enhanced discriminator of ROC (BEDROC), and enrichment factor at 1% of the rank-ordered list (EF1%), as previously elaborated.[54] This process was repeated 10 times for every target, and mean values were used to compute summary statistics across benchmark datasets.

For calculating scaffold enrichment, Bemis-Murcko scaffolds (BMS)[56] were constructed using the RDKit implementation of the method, facilitating grouping of all actives by scaffold, per target. Query compounds (1, 2, or 5) were drawn by randomly sampling BMS groups, and then randomly choosing one representative compounds when the corresponding group had more than one associated compound. From there on, the benchmarking process was identical to what was described above, with enrichment factors calculated on basis of BMS counts, rather than counts of distinct compounds.

Conclusion

We have presented a novel framework for leveraging deep learning in ligand-based virtual screening. Distinct from most previous contributions, our approach results in a target-agnostic model that can prospectively be applied with the same ease as conventional topological fingerprints. We have demonstrated that leveraging deep learning for automated feature extraction in conjunction with a multi-task training regime – results in a molecular representation with enhanced scaffold hopping potential, and with significant promise in low data hit discovery settings. Beyond the performance reported herein, we believe our framework also provides a foundation for generating significant additional value from organizational HTS data assets.

Acknowledgement

The authors thank Vladimir Savic (University of Belgrade – Faculty of Pharmacy) for numerous useful discussions during the course of this project. We thank Péter Kómár and Deniz Kural (Totient, Inc.) for reviewing the manuscript.

Supporting Information Available

The following files are available free of charge.

- GATNN VS SI Text.pdf: Figs. S1-S5, and Table S1

References

(1) Ripphausen, P.; Nisius, B.; Peltason, L.; Bajorath, J. Quo Vadis, Virtual Screening? A Comprehensive Survey of Prospective Applications. *J. Med. Chem.* **2010**, *53*, 8461–8467.

(2) Scior, T.; Bender, A.; Tresadern, G.; Medina-Franco, J.; Martínez-Mayorga, K.; Langer, T.; Cuanalo-Contreras, K.; Agrafiotis, D. K. Recognizing Pitfalls in Virtual Screening: A Critical Review. *J. Chem. Inf. Model.* **2012**, *52*, 867–881.

(3) Stumpfe, D.; Bajorath, J. Current Trends, Overlooked Issues, and Unmet Challenges in Virtual Screening. *J. Chem. Inf. Model.* **2020**,

(4) Goodnow, R. A.; Dumelin, C. E.; Keefe, A. D. DNA-encoded chemistry: enabling the deeper sampling of chemical space. *Nat. Rev. Drug Discovery* **2017**, *16*, 131–147.

(5) Baskin, I. I.; Winkler, D.; Tetko, I. V. A renaissance of neural networks in drug discovery. *Expert Opin. Drug Discovery* **2016**, *11*, 785–795.

(6) Goh, G. B.; Hodas, N. O.; Vishnu, A. Deep learning for computational chemistry. *J. Comput. Chem.* **2017**, *38*, 1291–1307.

(7) Carpenter, K. A.; Cohen, D. S.; Jarrell, J. T.; Huang, X. Deep learning and virtual drug screening. *Future Med. Chem.* **2018**, *10*, 2557–2567.

(8) Pérez-Sianes, J.; Pérez-Sánchez, H.; Díaz, F. Virtual Screening Meets Deep Learning. *Curr. Comput.-Aided Drug Des.* **2019**, *15*, 6–28.

(9) Mater, A. C.; Coote, M. L. Deep Learning in Chemistry. *J. Chem. Inf. Model.* **2019**, *59*, 2545–2559.

(10) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.

(11) Goh, G. B.; Siegel, C.; Vishnu, A.; Hodas, N. O.; Baker, N. Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models. *arXiv* **2017**, 1706.06689.

(12) Eckert, H.; Bajorath, J. Molecular similarity analysis in virtual screening: foundations, limitations and novel approaches. *Drug Discovery Today* **2007**, *12*, 225–233.

(13) Maggiora, G.; Vogt, M.; Stumpfe, D.; Bajorath, J. Molecular Similarity in Medicinal Chemistry. *J. Med. Chem.* **2014**, *57*, 3186–3204.

(14) Hu, Y.; Stumpfe, D.; Bajorath, J. Recent Advances in Scaffold Hopping. *J. Med. Chem.* **2017**, *60*, 1238–1246.

(15) Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. Atom pairs as molecular features in structure-activity studies: definition and applications. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 64–73.

(16) Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 82–85.

(17) Gedeck, P.; Rohde, B.; Bartels, C. QSAR – How Good Is It in Practice? Comparison of Descriptor Sets on an Unbiased Cross Section of Corporate Data Sets. *J. Chem. Inf. Model.* **2006**, *46*, 1924–1936.

(18) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754.

(19) Clark, M.; Cramer, R. D.; Jones, D. M.; Patterson, D. E.; Simeroth, P. E. Comparative molecular field analysis (CoMFA). 2. Toward its use with 3D-structural databases. *Tetrahedron Comput. Methodol.* **1990**, *3*, 47–59.

(20) Mason, J. S.; Morize, I.; Menard, P. R.; Cheney, D. L.; Hulme, C.; Labaudiniere, R. F. New 4-Point Pharmacophore Method for Molecular Similarity and Diversity Applications: Overview of the Method and Applications, Including a Novel Approach to the Design of Combinatorial Libraries Containing Privileged Substructures. *J. Med. Chem.* **1999**, *42*, 3251–3264.

(21) Kurogi, Y.; Guner, O. F. Pharmacophore Modeling and Three-dimensional Database Searching for Drug Design Using Catalyst. *Curr. Med. Chem.* **2001**, *8*, 1035–1055.

(22) Abrahamian, E.; Fox, P. C.; Nærum, L.; Christensen, I. T.; Thøgersen, H.; Clark, R. D. Efficient Generation, Storage, and Manipulation of Fully Flexible Pharmacophore Multiplets and Their Use in 3-D Similarity Searching. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 458–468.

(23) Wolber, G.; Langer, T. LigandScout: 3-D Pharmacophores Derived from Protein-Bound Ligands and Their Use as Virtual Screening Filters. *J. Chem. Inf. Model.* **2005**, *45*, 160–169.

(24) Haigh, J. A.; Pickup, B. T.; Grant, J. A.; Nicholls, A. Small Molecule Shape-Fingerprints. *J. Chem. Inf. Model.* **2005**, *45*, 673–684.

(25) Dixon, S. L.; Smondyrev, A. M.; Knoll, E. H.; Rao, S. N.; Shaw, D. E.; Friesner, R. A. PHASE: a new engine for pharmacophore perception, 3D QSAR model development, and 3D database screening: 1. Methodology and preliminary results. *J. Comput.-Aided Mol. Des.* **2006**, *20*, 647–671.

(26) Cheeseright, T. J.; Mackey, M. D.; Melville, J. L.; Vinter, J. G. FieldScreen: Virtual Screening Using Molecular Fields. Application to the DUD Data Set. *J. Chem. Inf. Model.* **2008**, *48*, 2108–2117.

(27) Ballester, P. J.; Finn, P. W.; Richards, W. G. Ultrafast shape recognition: Evaluating a new ligand-based virtual screening technology. *J. Mol. Graphics Modell.* **2009**, *27*, 836–845.

(28) Cross, S.; Baroni, M.; Goracci, L.; Cruciani, G. GRID-Based Three-Dimensional Pharmacophores I: FLAPpharm, a Novel Approach for Pharmacophore Elucidation. *J. Chem. Inf. Model.* **2012**, *52*, 2587–2598.

(29) Axen, S. D.; Huang, X.-P.; Cáceres, E. L.; Gendelev, L.; Roth, B. L.; Keiser, M. J. A Simple Representation of Three-Dimensional Molecular Structure. *J. Med. Chem.* **2017**, *60*, 7393–7409.

(30) Venkatraman, V.; Pérez-Nueno, V. I.; Mavridis, L.; Ritchie, D. W. Comprehensive Comparison of Ligand-Based Virtual Screening Tools Against the DUD Data set Reveals Limitations of Current 3D Methods. *J. Chem. Inf. Model.* **2010**, *50*, 2079–2093.

(31) Swann, S. L.; Brown, S. P.; Muchmore, S. W.; Patel, H.; Merta, P.; Locklear, J.; Hajduk, P. J. A Unified, Probabilistic Framework for Structure- and Ligand-Based Virtual Screening. *J. Med. Chem.* **2011**, *54*, 1223–1232.

(32) Hu, G.; Kuang, G.; Xiao, W.; Li, W.; Liu, G.; Tang, Y. Performance Evaluation of 2D Fingerprint and 3D Shape Similarity Methods in Virtual Screening. *J. Chem. Inf. Model.* **2012**, *52*, 1103–1113.

(33) Vogt, M.; Stumpfe, D.; Geppert, H.; Bajorath, J. Scaffold Hopping Using Two-Dimensional Fingerprints: True Potential, Black Magic, or a Hopeless Endeavor? Guidelines for Virtual Screening. *J. Med. Chem.* **2010**, *53*, 5707–5715.

(34) Ripphausen, P.; Nisius, B.; Bajorath, J. State-of-the-art in ligand-based virtual screening. *Drug Discovery Today* **2011**, *16*, 372–376.

(35) Gardiner, E. J.; Holliday, J. D.; O'Dowd, C.; Willett, P. Effectiveness of 2D fingerprints for scaffold hopping. *Future Med. Chem.* **2011**, *3*, 405–414.

(36) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. In *Advances in Neural Information Processing Systems 28*; Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc., 2015; pp 2224–2232.

(37) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 595–608.

(38) Xu, Y.; Pei, J.; Lai, L. Deep Learning Based Regression and Multiclass Models for Acute Oral Toxicity Prediction with Automatic Chemical Feature Extraction. *J. Chem. Inf. Model.* **2017**, *57*, 2672–2685.

(39) Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; Jensen, K. F. Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction. *J. Chem. Inf. Model.* **2017**, *57*, 1757–1772.

(40) Altae-Tran, H.; Ramsundar, B.; Pappu, A. S.; Pande, V. Low Data Drug Discovery with One-Shot Learning. *ACS Cent. Sci.* **2017**, *3*, 283–293.

(41) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.* **2019**, *10*, 370–377.

(42) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. Proceedings of the 34th International Conference on Machine Learning - Volume 70. 2017; pp 1263–1272.

(43) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370–3388.

(44) Ryu, S.; Lim, J.; Hong, S. H.; Kim, W. Y. Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network. *arXiv* **2018**, 1805.10988.

(45) Withnall, M.; Lindelöf, E.; Engkvist, O.; Chen, H. Building attention and edge message passing neural networks for bioactivity and physical–chemical property prediction. *J. Cheminf.* **2020**, *12*, 1.

(46) Xiao, T.; Qi, X.; Chen, Y.; Jiang, Y. Development of Ligand-based Big Data Deep Neural Network Models for Virtual Screening of Large Compound Libraries. *Mol. Inf.* **2018**, *37*, 1800031.

(47) Srinivas, R.; Klimovich, P. V.; Larson, E. C. Implicit-descriptor ligand-based virtual screening by means of collaborative filtering. *J. Cheminf.* **2018**, *10*, 56.

(48) Riniker, S.; Wang, Y.; Jenkins, J. L.; Landrum, G. A. Using Information from Historical High-Throughput Screens to Predict Active Compounds. *J. Chem. Inf. Model.* **2014**, *54*, 1880–1891.

(49) Helal, K. Y.; Maciejewski, M.; Gregori-Puigjané, E.; Glick, M.; Wassermann, A. M. Public Domain HTS Fingerprints: Design and Evaluation of Compound Bioactivity Profiles from PubChem's Bioassay Repository. *J. Chem. Inf. Model.* **2016**, *56*, 390–398.

(50) Jaeger, S.; Fulle, S.; Turk, S. Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition. *J. Chem. Inf. Model.* **2018**, *58*, 27–35.

(51) Winter, R.; Montanari, F.; Noé, F.; Clevert, D.-A. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.* **2019**, *10*, 1692–1701.

(52) Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2017**, 1710.10903.

(53) Bresson, X.; Laurent, T. Residual Gated Graph ConvNets. *arXiv* **2017**, 1711.07553.

(54) Riniker, S.; Landrum, G. A. Open-source platform to benchmark fingerprints for ligand-based virtual screening. *J. Cheminf.* **2013**, *5*, 26.

(55) Probst, D.; Reymond, J.-L. A probabilistic molecular fingerprint for big data settings. *J. Cheminf.* **2018**, *10*, 66.

(56) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887–2893.

(57) Bellmann, L.; Penner, P.; Rarey, M. Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration. *J. Chem. Inf. Model.* **2019**, *59*, 4625–4635.

(58) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757–1768.

(59) Poličar, P. G.; Stražar, M.; Zupan, B. openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. *bioRxiv* **2019**, 731877.

(60) Hüllermeier, E.; Fober, T.; Mernberger, M. In *Encyclopedia of Systems Biology*; Dubitzky, W., Wolkenhauer, O., Cho, K.-H., Yokota, H., Eds.; Springer New York: New York, NY, 2013; pp 1018–1018.

(61) Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272.

(62) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **2019**, *47*, D1102–D1109.

(63) Landrum, G. A. RDKit: Open-Source Cheminformatics Software. `https://www.rdkit.org/`, accessed on February 26, 2020.

(64) Rohrer, S. G.; Baumann, K. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. *J. Chem. Inf. Model.* **2009**, *49*, 169–184.

(65) Heikamp, K.; Bajorath, J. Large-Scale Similarity Search Profiling of ChEMBL Compound Data Sets. *J. Chem. Inf. Model.* **2011**, *51*, 1831–1839.

(66) Gaulton, A. et al. The ChEMBL database in 2017. *Nucleic Acids Res.* **2016**, *45*, D945–D954.

(67) Jahn, A.; Hinselmann, G.; Fechner, N.; Zell, A. Optimal assignment methods for ligand-based virtual screening. *J. Cheminf.* **2009**, *1*, 14.

(68) Paszke, A. et al. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc., 2019; pp 8024–8035.

(69) Wang, M. et al. Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs. *arXiv* **2019**, 1909.01315.

(70) Ba, J. L.; Kiros, J. R.; Hinton, G. E. Layer Normalization. *arXiv* **2016**, 1607.06450.

(71) He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). USA, 2015; pp 1026–1034.

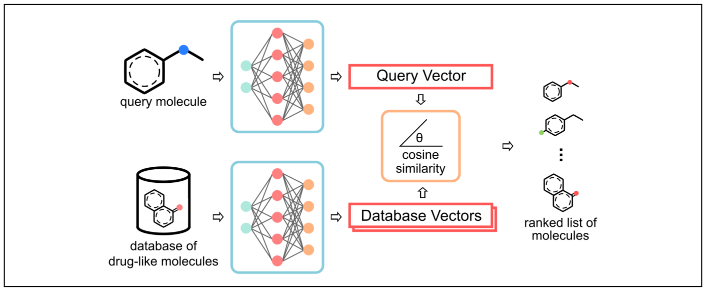
(72) Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, 11502.03167.

(73) Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, 1711.05101.

(74) Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; Wilson, A. G. Averaging Weights Leads to Wider Optima and Better Generalization. *arXiv* **2018**, 1803.05407.

(75) Willett, P. Fusing similarity rankings in ligand-based virtual screening. *Comput. Struct. Biotechnol. J.* **2013**, *5*, e201302002.

Graphical TOC Entry


query molecule
database of
drug-like molecules
Query Vector
θ
cosine
similarity
Database Vectors
⋮
ranked list of
molecules