ELSEVIER



# Docking-based generative approaches in the search for new drug candidates

**Tomasz Danel [1,*], Jan Łęski [1], Sabina Podlewska [2], Igor T. Podolak [1]**

[1] Faculty of Mathematics and Computer Science, Jagiellonian University, 6 Łojasiewicza Street, 30-348 Kraków, Poland
[2] Maj Institute of Pharmacology, Polish Academy of Sciences, Department of Medicinal Chemistry, 31-343 Kraków, Smętna Street 12, Poland

**Despite the popularity of virtual screening (VS) of existing compound libraries, the search for new potential drug candidates also takes advantage of generative protocols, where new compound suggestions are enumerated using various algorithms. To increase the activity potency of generative approaches, they have recently been coupled with molecular docking, a leading methodology of structure-based drug design (SBDD). In this review, we summarize progress since docking-based generative models emerged. We propose a new taxonomy for these methods and discuss their importance for the field of computer-aided drug design (CADD). In addition, we discuss the most promising directions for the further development of generative protocols coupled with docking.**

**Keywords: molecular docking; generative models; deep learning; evolutionary algorithms; computer-aided drug design; fragment-based drug design**

## Introduction

In many cases, drug discovery pipelines are now connected with the application of various *in silico* techniques, including machine learning (ML) methods, which can perform fast and effective analysis of huge amount of data.[1] Recently, ML was revolutionized by the rapid development of deep learning (DL) approaches, the main characteristic of which is the ability to extract higher-level features from raw input data. Considering its huge potential, DL has also entered the field of CADD. It has already been used for the development of quantitative structure–activity relationship (QSAR) models and in VS campaigns to improve the docking-based scoring of compound libraries.[2] In addition, DL methods have been used to predict ligand–protein contacts[3] and for computational assessment of the physicochemical and absorption, distribution, metabolism, excretion, and toxicity (ADMET) properties of compounds.[4]

DL bridges two fundamental approaches used in CADD: docking and generative methods. Structure-based approaches, with docking in particular, have already been the subject of several reviews.[5,6] In addition, generative approaches in general have already been well characterized in the context of their potential usage in the drug design process.[7] Here, we review the combination of generative models and docking for use in CADD. Its potential benefits for the design of new drug candidates include the ability to explore novel chemical space on the one hand (through generative methods) and the instant assessment with reliable methods on the other (through docking algorithms).

## Background

Here, we provide a technical background to the tools used in generative modeling with molecular docking.

### *Generative models for molecules*

In molecular design, the term 'generative model' describes a computer system that is able to yield new molecules, often with a set of predefined characteristics. Generative models in drug discovery are implemented to accelerate the design of novel therapeutic compounds.[8]

* *Corresponding author: Danel, T. (tomasz.danel@uj.edu.pl)*


https://doi.org/10.1016/j.drudis.2022.103439

Two molecular representations are prevalent in generative modeling: linearized text representations and molecular graphs. Linearized text representations, such as SMILES or SELFIES, can leverage advances in natural language processing, but they poorly describe geometrical relations between atoms. Molecular graphs can be constructed at different levels: 2D or 3D. They capture the concept of atom neighborhood (2D) or exploit spatial relations as a result of the recent development of roto-translationally equivariant neural networks (3D).[9]

One of the most widely used generative models in CADD are autoregressive models.[10] They build molecules out of atoms or fragments by selecting the most promising modifications in each step of the generation. The next modification can be picked with the use of an oracle, such as a predictive model or molecular docking. Otherwise, the model can be trained in a supervised manner by reconstructing the molecules in the input data set.

Another class of generative models used in chemistry are latent-based models. In this approach, molecules are decoded from a continuous latent vector space, which is a vectorized molecular representation artificially constructed by data enumeration and model training. Autoencoders are a typical example of these models, including variational autoencoders (VAE)[11] and adversarial autoencoders.[12] First, they encode input molecules into a low-dimensional vector representation that follows one of the known probability distributions; they then decode compounds from this representation to match the input molecules. Subsequently, if the decoder is an autoregressive network, it can be fine-tuned as described above or the latent space can be searched to identify potential ligands in a more manageable and structured data space. Generative adversarial networks (GANs)[8] are another example of latent-based models, but instead of using encoders, they use discriminator networks to evaluate compounds decoded from the sampled latent vectors.

### *Molecular docking software used in docking-based generative approaches*

Generative models started to leverage molecular docking to propose new drug candidates. The docking software most widely used for tasks related to new compounds generation is AutoDock Vina and its derivatives. It is an open-source program, under a permissive Apache license, widely used with docking-based generative models. Other include: smina,[13] a branch of AutoDock Vina with an improved scoring function; QuickVina,[14] which uses heuristics to accelerate docking; Glide,[15] an extensive and accurate tool; and gnina,[16] using neural networks (NNs) as a scoring function. NNs are sometimes trained on docking scores and used instead of docking to accelerate the process of docking-based compound evaluation.[17,18]

### *Model evaluation metrics*

The fundamental metrics used to evaluate generative models are validity, uniqueness, and novelty of the generated compounds.[19] Additionally, measures of drug-likeness, such as quantitative estimate of drug likeness (QED) or synthetic accessibility (SA) (e.g., estimated by the molecule.one tool[20]) are used to assess the quality of enumerated molecules. In most drug discovery projects, it is also important to have a varied set of drug candidates, the diversity of which can be measured e.g. by internal diversity or sphere exclusion diversity (SEDiv).[21] Finally, the most important metrics for the targeted generative models are docking-based measures.[18]

For 3D generative models, additional measures verify the correct conformation of the generated molecules, such as 3D maximum mean discrepancy value or root mean square deviation (RMSD) between the generated and reference conformations.[22] For some models, the shape and pharmacophoric constraints are checked using metrics such as the shape and color similarity score proposed by Imrie *et al.*[23]

## Docking-based generative models in drug design

A docking-based generative model creates several drug-like molecules by enhancing their binding affinity via utilization of a computational docking model. Docking scores can be used directly, such as a component of an optimized reward value, or indirectly as a filtering method, in which only compounds with a favorable docking score are retained from several generated structures.

We partitioned the docking-based generative models into two categories: the pocket- and target-based models (Fig. 1). The former construct a description of the binding pocket and create compounds that best fit the described binding site, using either a 2D representation or building 3D molecular graphs directly inside the pocket. The target-based models are trained specifically for the selected drug target. They can be guided by reinforcement learning, genetic algorithms, or different iterative methods. Other algorithms explore the latent representation of molecules to identify potential binders of the given target in pretrained generative models. Examples of generative approaches with their source code are gathered in Table 1.

### *Pocket-based models*

Pocket-based models use the shape and physicochemical properties of binding sites, either by encoding them in the model or by using docking scoring functions to assess the generated conformers (Fig. 1a,b). Binding pockets can be represented, such as 3D molecular graphs[9,22,24–26] or voxel grids,[27,28] and processed by graph (GNNs) or convolutional neural networks (CNNs), respectively. This way, the target protein can be replaced, in some cases, without model retraining, and the application of the model can be directly transferred to another target.[24,25,28]

#### ***SMILES models conditioned on binding pockets***

Xu *et al.*[29] encoded binding pockets with eigenvalues of the Coulomb matrix of coarse-grained atoms and generated compounds with a conditional RNN. In an extension of this approach, Zhang *et al.*[30] applied interaction fingerprints of the ligand–protein complexes obtained in docking, using an LSTM network to generate target-focused compounds. Both methods produce SMILES strings as an output. Zheng *et al.*[31] implemented a transformer architecture for scaffold hopping using protein encoding as a context for the optimization. Proteins were encoded from the strings of amino acids using TAPE, which is also a transformer architecture. Skalic *et al.*[27,32] used a semi-3D approach to generate compounds, whereby they encoded and decoded only the shape of a molecule, using a shape captioning network to decode SMILES strings. 3D CNNs encode the shape and pharmacophoric features of the molecule.[32] Proteins can be represented in a similar way to ensure that ligands match the given protein descrip-

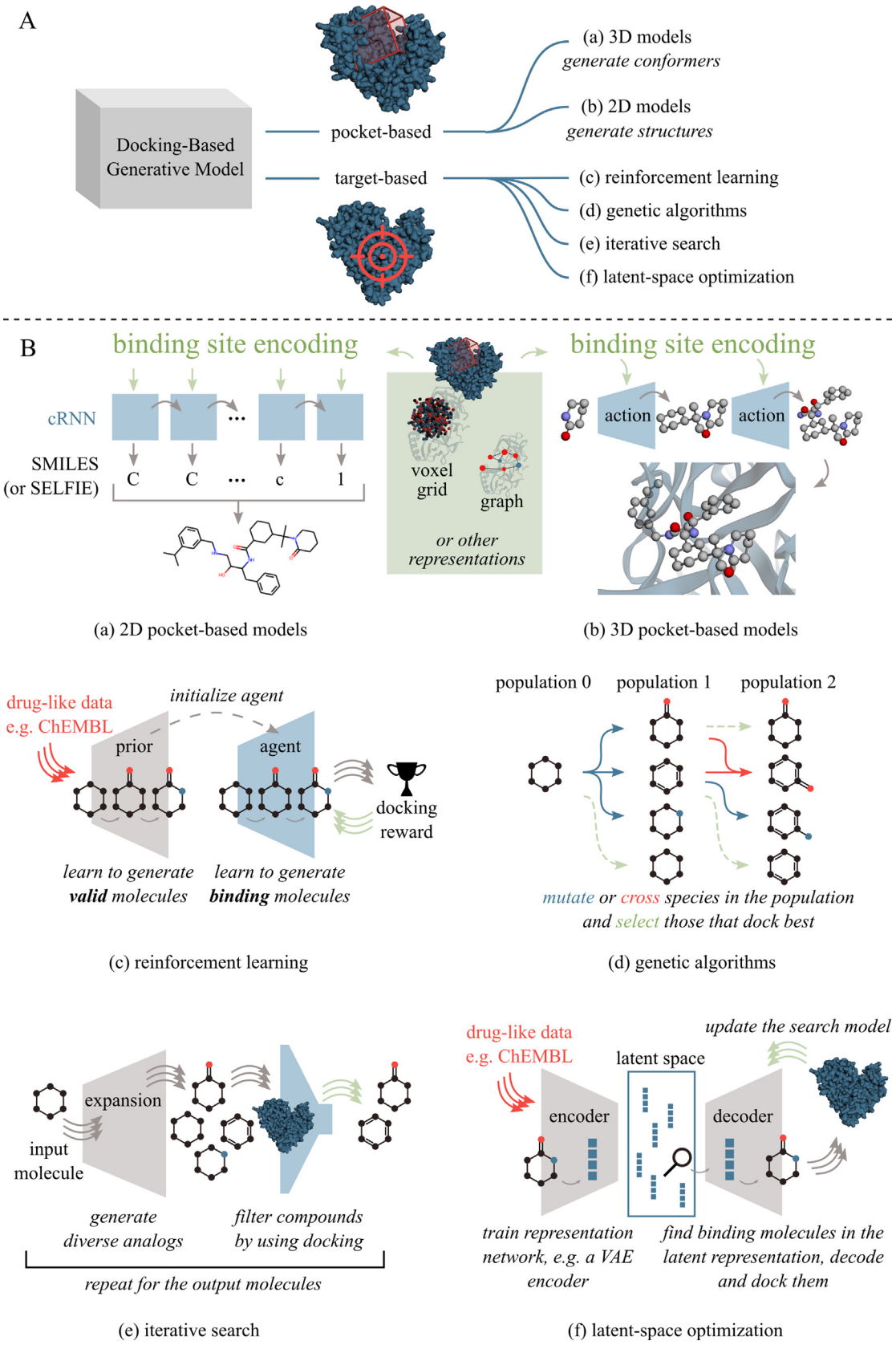


**FIG. 1**

The taxonomy of docking-based generative models proposed in the study. Abbreviation: VAE, variational autoencoder.

**TABLE 1**

**Docking-based generative models.**

| Name | URL | License | Refs |
|---|---|---|---|
| 3D SBDD model | github.com/luost26/3D-Generative-SBDD | MIT | 24 |
| Pocket2Mol | github.com/pengxingang/Pocket2Mol | MIT | 25 |
| LiGAN | github.com/mattragoza/liGAN | GNU GPL 2.0 | 28 |
| DeepHop | github.com/prokia/deepHops | MIT | 31 |
| LigDream | github.com/compsciencelab/ligdream | AGPL-3.0 | 32 |
| GB-GA | github.com/jensengroup/GB_GA | MIT | 20,34 |
| AutoGrow4 | durrantlab.com/autogrow4 | Apache 2.0 | 35 |
| JANUS | github.com/asparu-guzik-group/JANUS | Apache 2.0 | 39 |
| V-dock | github.com/knu-chem-lcbc/V-dock | Open source | 17 |
| MolFinder | github.com/duaibeom/MolFinder | BSD 3-Clause | 40 |
| LigBuilder V3 | https://www.pkumdl.cn/ligbuilder3/ | Academic | 42 |
| MORLD | github.com/wsjeon92/morld | Apache 2.0 | 43 |
| FREED | github.com/AITRICS/FREED | Apache 2.0 | 44 |
| MolScore | github.com/MorganCThomas/MolScore | MIT | 21 |
| REINVENT | github.com/MarcusOlivecrona/REINVENT | MIT | 38 |
| REINVENT 3.2 | github.com/MolecularAI/Reinvent | Apache 2.0 | 45,47,48 |
| LibINVENT | github.com/MolecularAI/Lib-INVENT | Apache 2.0 | 46 |
| Link-INVENT | github.com/MolecularAI/Reinvent | Apache 2.0 | 47 |
| DockStream | github.com/MolecularAI/DockStream | Apache 2.0 | 49 |
| STONED | github.com/aspuru-guzik-group/stoned-selfies | Apache 2.0 | 50 |
| JTVAE sample-and-dock | github.com/atfrank/SampleDock | GNU GPL 3.0 | 51 |
| OptiMol | github.com/jacquesboitreaud/OptiMol | Open source | 52 |
| Gradient Latent Search | github.com/cieplinski-tobiasz/smina-docking-benchmark | MIT | 18 |

tion.[27] In the publication, BicycleGAN is used to generate diverse ligands for a single input protein.

#### *3D generative models*

The first report of a deep generative model producing 3D compound structures conditioned on the receptor binding site was by Ragoza, Masuda and Koes.[28] As a starting point, the 3D bound molecular structure (including the target binding pocket) is encoded to a latent space using atomic density grids transformed by CNNs. Atoms are encoded as continuous, Gaussian-like densities in a 3D grid with separate channels for each atom type. Conditional variational autoencoder (CVAE) with the GAN loss function is used for training (the network is trained on the CrossDocked2020 data set). Ligand and receptor density grids form a CVAE input and ligand density grid is returned as an output. To obtain valid molecules from the generated grids, an algorithm combining beam search and gradient descent is used to return a set of atom types and coordinates with the best fit to a given atomic density grid; these are finally combined into a valid molecule by the respective bonds assignment.

Luo *et al.*[24] developed a 3D generative model, which predicts the probability of atom presence in the binding site region. The atoms of the protein and ligand are encoded as a graph of atoms connected by the k-NN algorithm, and rotationally invariant GNNs are used to determine an atom probability density in the 3D space. Finally, the autoregressive sampling algorithm is used for enumerating molecules from the model using the estimated density.

Instead of encoding binding pockets, Li *et al.*[22] directly incorporated docking to score the generated molecules. They developed an autoregressive method comprising two networks: a state encoder and a policy network. The state encoder is a GNN that creates representation of a partially built molecule, and the policy network decides which chemical moieties should be added to the molecule. MCTS with docking scores as the scoring function is used to sample molecules.

Peng *et al.* developed Pocket2Mol,[25] an efficient system based on an E(3)-equivariant generative network, which combines a GNN capturing chemical and geometrical constraints of the target binding pocket and a sampling algorithm leading to the generation of novel ligand candidates conditioned on the 3D pocket. Pocket2Mol adopts the autoregressive strategy to learn a probability distribution of particular atom or bond type for a given space fragment inside a pocket on the basis of the already existing atoms (part of the drug is randomly masked, and the model is trained to predict the remaining fragment).

### *Target-based models*

Here, we describe different approaches to docking-based compound generation, used by target- and, to some extent, pocket-based models (Fig. 1c–f).

#### *Genetic algorithms*

Genetic algorithms (GAs) are fit for use with docking score as the fitness function and have the advantage of not having to be trained. A well-known method is GANDI, a fragment-based model[33] in which predocked fragments are encoded by a genetic algorithm and tabu search is used to find optimal linkers for the formation of final compounds (the fitness is computed as a combination of local binding energy and similarity to pocket). Steinman and Jensen[20] extended the graph-based GB-GA model[34] to use the roulette selection and the fitness function to be a combination of docking score from Glide together with an index of synthesizability. The method was shown to outperform high throughput VS.

AutoGrow4[35] is a large open-source GA-based package, fit for both *de novo* and lead optimization tasks, each starting with a dif-

ferent population. AutoGrow4 implements all generation operators (elitism, mutation, and crossover) using Ranking, Roulette, and Tournament selections not to get trapped in local solutions. It uses docking scores (Vina and QVina[14] dockers are applied) together with diversity as its fitness score. Fu *et al.*[36] noted that GAs have high variance across runs because of the randomness in crossover and mutation. Therefore, based on the AutoGrow4 environment, they proposed to use separate equivariant NNs[37] to first choose parents for crossover and mutation, and then to select the second parent for crossover or a reaction from the SMARTS set for mutation. The networks are trained using reinforcement learning (RL) with the REINVENT scheme,[38] using docking scores as a reward. This approach was shown to give superior results, slightly better than AutoGrow4, REINVENT, and GB-GA.

The JANUS model[39] is interesting because it uses two populations: one for chemical space exploration controlled by DNNs with crossover and mutation, and one for exploitation controlled only by mutation. The DNN is trained for each population using molecules with known docking values. The populations share the best members. JANUS uses SELFIES representations and the STONED model (see below) for the definition of both of these operators, speeding up the generation of new members considerably. Using the docking value as a function of fitness, the model achieves significantly low docking scores. Importantly, the authors noted that such a GA model must also have the values responsible for the synthesizability and stability of the generated molecules built into the fitness function. The member generation operators should also take care of these characteristics.

Designing the V-dock model, Choi *et al.*[17] extended the MolFinder system[40] by adding an NN trained to evaluate docking scores. The entire system uses SMILES, along with the CSA algorithm,[41] and a combination of GA and simulated annealing. All of these elements greatly accelerate the model.

Another GA tool popular with medicinal chemists is LigBuilder.[42] LigBuilder V3 offers a polypharmacological approach to ligand enumeration. It is able to generate compounds with reference to their activity toward multiple targets using the Chemical Space Exploring Algorithm (CSEA). CSEA constructs potential ligands starting from the placement of an sp3 carbon atom in a randomly selected point of a binding site. Then, newly formed molecules are split into fragments, and those fragments that have the highest potential of desired biological activity are used as a starting point for the subsequent growing operations.

### Reinforcement learning models

RL has been used for more focused exploration of the chemical space in search of binding molecules. Docking scores can be used as rewards for the generated molecules to guide the generative process. Jeon and Kim[43] proposed an algorithm that builds a molecule by adding atoms and bonds sequentially. The intermediate steps are assessed in terms of SA and QED, and only the final compound is docked to evaluate its binding potential. By contrast, Yang *et al.*[44] proposed to build compounds out of chemically reasonable fragments, and docking scores are computed for each generation step.

Several target-based generative methods[21] are based on the REINVENT model proposed by Olivecrona *et al.*,[38] which uses RL for output optimization. First, a generative RNN model (the Prior) is trained on a subset of ChEMBL to generate valid SMILES strings. Then, an Agent, a copy of the Prior, is trained with RL to modify the proposition of the Prior toward some specific goal (e.g., improved docking scores). REINVENT 2.0[45] added diversity filters memorizing structures with similar scaffolds to drive the model toward higher diversity. Transfer learning introduced in the Prior training directs the creativity toward a given molecule subset. LibINVENT[46] added moieties driving the RL module toward best docking, diversity, or synthesizability.

Link-INVENT[47] was designed for the fragment-based drug discovery problem: on the basis of a batch of fragments, the encoder–decoder model proposes whole linked molecules to be optimized with RL toward best docking (Glide used). In the case of complex multiparameter optimization, curriculum learning might be favorable[48]: that is, divide optimization into several production objectives, such as molecules with a given target scaffold, that are drug-like, that optimize some given feature, realized by the Agent sequentially. Finally, DockStream[49] is a docking platform providing protein and ligand preparation tools and multiple docking backends that can be combined with REINVENT for structure-based drug design (SBDD).

### Iterative models with docking-based evaluation

Some models have the ability to produce new molecules, but do not have means, other than heuristic, to optimize them toward better docking. We call them iterative here because the generated molecules are evaluated toward some target and filtered accordingly.

Ghanokta *et al.*[53] propose a model where PathFinder[54] performs a retrosynthetic analysis starting from some molecule and enumerating all possible reactions, while keeping the core fixed. After filtering (SMARTS, physicochemical, and duplicate removal) and docking against a given target, a subset is used to train an ML model based on free energy perturbation (FEP) experiments, with input poses evaluated via Glide. This FEP prediction model is then used to filter a large number of generated molecules and select the top ones. This path might be used to train the REINVENT Prior model, giving high correct core and reaction-group molecules, speeding up the quest for the best one.

One possible improvement is to use the SELFIES representation.[50] STONED combines a sequence of point-wise modifications in a SELFIES to generate a large number of all valid new molecules (unlike when SMILES is used). This enables rapid generation of molecule subspaces by 'superimposition' of point modifications and extend quicker from the original one. The authors also show how to produce unambiguous paths between two given molecules. Given that the system is discrete, it is not possible to directly optimize some parameters (e.g., the docking score), other than by using some heuristics.

Another model is based on Monte-Carlo Tree Search (MCTS).[55] MCTS builds a search tree by adding a single level to a SMILES string in each selection-expansion stage. A high number or randomly generated complete strings are then evaluated with the Vina docker in the simulation stage. All nodes have their current score updated based on these evaluations (the backpropagation). The symbols that resulted in better scores are then chosen more frequently in later selection stages. The authors have identified a large fraction of molecules with better scores

for a given target than those from the US Food and Drug Administration (FDA) list.

A similar in operation sample-and-dock approach[51] uses a pretrained JTVAE model to map molecules to the latent space and then sample from its neighborhood. Sampled vectors are decoded, recoded as graphs and docked (RDKit and rDock were used) against the target. The best molecules are mapped again in a similar loop until no better molecules can be found. It is possible to apply transfer learning after pretraining using an additional subset of specific active molecules. This needs to be done with care, not to hinder JTVAE generativity.

***Latent-space optimization***

Conversely to most of the methods described in the previous sections, some models use the latent space of already trained generators to discover new binders without changing the generative process itself. Any generative model that uses a continuous latent representation to generate molecules, such as an autoencoder, can be used as a prior generative model and can be trained on a big data set of drug-like molecules (e.g., ChEMBL or ZINC). Next, a latent-space optimization algorithm is used to effectively sample latent variables that map to well-binding ligands.

An example of this approach is OptiMol,[52] an algorithm that uses a VAE as the prior generative model. First, the VAE is trained to encode molecular graphs and decode SELFIES representation. Next, two latent-space exploration schemes are used: Bayesian Optimization (BO) and Conditioning by Adaptive Sampling (CbAS). In both schemes, molecular docking is used to evaluate compounds decoded from the latent space and change the sampling strategy. BO uses Gaussian processes to find the most promising regions of the latent space, evaluate compounds sampled from these regions, and adjust the Gaussian process. By contrast, CbAS shifts the initial distribution of the prior generative model to maximize the docking-based objective function, which makes the search more effective.

Another approach to the latent-space optimization is presented by Ciepliński *et al.*,[18] who trained two VAE models, SMILES VAE, and Grammar VAE, as prior generators and used a predictive model trained to map latent vectors to docking scores as a surrogate function. This way, new binders can be found in the latent space by gradient ascent maximizing model predictions starting from a random initial point in the latent space.

## Other models and outlook

There are still several models that do not address the problem of docking score optimization directly, but are able to generate novel molecules and optimize their features, which might include docking objectives. For example, MOLUCINATE[56] is a VAE model that is able to produce a 3D molecule grid representation using graph spatial convolutions. The generated 3D positions could be paired with docking to provide active conformations. A deep 3D linker model[23,57] is designed to create linkers between two small molecular graphs by adding bonds in a breadth-first manner. This model is evaluated using docking scores, which could be integrated in the training procedure. Yet another example is a conditional 3D generative model by Gebauer *et al.*[58] For a given atom, the model generates a new one based on a factorized conditional probability. As conditions, the model uses isotropic polarizability, molecular fingerprints, and atomic composition, all embedded with an MLP network. Docking scores of the conformers could be added as another condition. The following idea of generating synthesizable molecules was suggested[59,60]: a SMILES representation is generated, all reactions needed are then predicted, molecules are filtered with those available, and these are then synthesized on a microfluidics platform. This model can easily be extended to optimize docking scores.

Other possible extensions of the current methods are combining generative models with DL solutions for affinity prediction or the prediction of docking poses. This would open a path to fully differentiable optimization of chemical structures.

## Concluding remarks

Generative ML methods constitute an important component of the set of the CADD tools. They enable exploration of novel chemical space (compared with commercially available compound libraries, which usually undergo VS) and, therefore, their popularity in the protocols for searching for new drugs is constantly expanding. To increase the effectiveness of generative methods to propose compounds that are active toward a particular target, their combination with molecular docking has recently been intensively explored. Here, we summarized docking-based generative methods and proposed their taxonomy. Moreover, we propose ways in which the docking-based generative models can be further developed.

## Data availability

No data was used for the research described in the article.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This work was supported by the National Science Centre (Poland) grant no. 2020/37/N/ST6/02728.

## References

1. Mitchell JBO. Machine learning methods in chemoinformatics. *Wiley Interdiscip Rev Comput Mol Sci*. 2014;4:468–481.
2. Pereira JC, Caffarena ER, Santos CN. Boosting docking-based virtual screening with deep learning. *J Chem Inf Model*. 2016;56:2495–2506.
3. Skwark MJ, Raimondi D, Michel M, Elofsson A. Improved contact predictions using the recognition of protein like contact patterns. *PLoS Comput Biol*. 2014;10:e1003889.
4. Lusci A, Pollastri G, Baldi P. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *J Chem Inf Model*. 2013;53:1563–1575.
5. Sulimov VB, Kutov DC, Taschilova AS, Ilin IS, Tyrtyshnikov EE, Sulimov AV. Docking paradigm in drug design. *Curr Top Med Chem*. 2021;21:507–546.
6. Pagadala NS, Syed K, Tuszynski J. Software for molecular docking: a review. *Biophys Rev*. 2017;9:91–102.

7. Meyers J, Fabian B, Brown N. De novo molecular design and generative models. *Drug Discov Today*. 2021;26:2707–2715.
8. Sousa T, Correia J, Pereira V, Rocha M. Generative deep learning for targeted compound design. *J Chem Inf Model*. 2021;61:5343–5361.
9. Han J, Rong Y, Xu T, Huang W. Geometrically equivariant graph neural networks: a survey. *arXiv*. 2022: arXiv:2202.07230v3.
10. Cheng Y, Gong Y, Liu Y, Song B, Zou Q. Molecular design in drug discovery: a comprehensive review of deep generative models. *Brief Bioinform*. 2021;22: bbab344.
11. Kingma DP, Welling M. Auto-encoding variational Bayes. *arXiv*. 2014: arXiv:1312.6114v10.
12. Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B. Adversarial autoencoders. *arXiv*. 2015: *arXiv:1511.05644*.
13. Koes DR, Baumgartner MP, Camacho CJ. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *J Chem Inf Model*. 2013;53:1893–1904.
14. Alhossary A, Handoko SD, Mu Y, Kwoh CK. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*. 2015;31:2214–2216.
15. Friesner RA, Banks JL, Murphy RB, Halgren TA, Klicic JJ, Mainz DT, et al.. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *J Med Chem*. 2004;47:1739–1749.
16. McNutt AT, Francoeur P, Aggarwal R, Masuda T, Meli R, Ragoza M, et al.. GNINA 1.0: molecular docking with deep learning. *J Cheminform*. 2021;13:1–20.
17. Choi J, Lee J. V-dock: fast generation of novel drug-like molecules using machine-learning-based docking score and molecular optimization. *Int J Mol Sci*. 2021;22:11635.
18. Cieplinski T, Danel T, Podlewska S, Jastrzebski S. We should at least be able to design molecules that dock well. *arXiv*. 2020: *arXiv:2006.16955*.
19. Brown N, Fiscato M, Segler MHS, Vaucher AC. GuacaMol: benchmarking models for de novo molecular design. *J Chem Inf Model*. 2019;59:1096–1108.
20. Steinmann C, Jensen JH. Using a genetic algorithm to find molecules with good docking scores. *PeerJ Phys Chem*. 2021;3:e18.
21. Thomas M, Smith RT, O'Boyle NM, Graaf C, Bender A. Comparison of structure- and ligand-based scoring functions for deep generative models: a GPCR case study. *J Cheminform*. 2021;13:1–20.
22. Li Y, Pei J, Lai L. Structure-based de novo drug design using 3D deep generative models. *Chem Sci*. 2021;12:13664–13675.
23. Imrie F, Hadfield TE, Bradley AR, Deane CM. Deep generative design with 3D pharmacophoric constraints. *Chem Sci*. 2021;12:14577–14589.
24. Luo S, Guan J, Ma J, Peng J. A 3D generative model for structure-based drug design. *Adv Neural Inf Process Syst*. 2021;34:1–11.
25. Peng X, Luo S, Guan J, Xie Q, Peng J, Ma J. Pocket2Mol: efficient molecular sampling based on 3D protein pockets. *Proc Mach Learn Res*. 2022;162:17644–17655.
26. Xia X, Hu J, Wang Y, Zhang L, Liu Z. Graph-based generative models for de novo drug design. *Drug Discov Today*. 2019;32–33:45–53.
27. Skalic M, Sabbadin D, Sattarov B, Sciabola S, De Fabritiis G. From target to drug: generative modeling for the multimodal structure-based ligand design. *Mol Pharm*. 2019;16:4282–4291.
28. Ragoza M, Masuda T, Koes DR. Generating 3D molecules conditional on receptor binding sites with deep generative models. *Chem Sci*. 2022;13:2701–2713.
29. Xu M, Ran T, Chen H. De novo molecule design through the molecular generative model conditioned by 3D information of protein binding sites. *J Chem Inf Model*. 2021;61:3240–3254.
30. Zhang J, Chen H. De novo molecule design using molecular generative models constrained by ligand–protein interactions. *J Chem Inf Model*. 2022;62:3291–3306.
31. Zheng S, Lei Z, Ai H, Chen H, Deng D, Yang Y. Deep scaffold hopping with multimodal transformer neural networks. *J Cheminform*. 2021;13:1–15.
32. Skalic M, Jiménez J, Sabbadin D, Fabritiis G. Shape-based generative modeling for de novo drug design. *J Chem Inf Model*. 2019;59:1205–1214.
33. Dey F, Caflisch A. Fragment-based de novo ligand design by multiobjective evolutionary optimization. *J Chem Inf Model*. 2008;48:679–690.
34. Jensen JH. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem Sci*. 2019;10:3567–3572.
35. Spiegel JO, Durrant JD. AutoGrow4: an open-source genetic algorithm for de novo drug design and lead optimization. *J Cheminform*. 2020;12:25.
36. Fu T, Gao W, Coley CW, Sun J. Reinforced genetic algorithm for structure-based drug design. https://openreview.net/pdf?id=_Sfd-icezCa [Accessed October 24, 2022].
37. Satorras VG, Hoogeboom E, Welling M. E(n) equivariant graph neural networks. *arXiv*. 2021: *arXiv:2102.09844*.
38. Olivecrona M, Blaschke T, Engkvist O, Chen H. Molecular de-novo design through deep reinforcement learning. *J Cheminform*. 2017;9:1–14.
39. Nigam A, Pollice R, Aspuru-Guzik A. Parallel tempered genetic algorithm guided by deep neural networks for inverse molecular design. *Digit Discov*. 2022;1:390–404.
40. Kwon Y, Lee J. MolFinder: an evolutionary algorithm for the global optimization of molecular properties and the extensive exploration of chemical space using SMILES. *J Cheminform*. 2021;13:1–14.
41. Lee J, Scheraga HA, Rackovsky S. New optimization method for conformational energy calculations on polypeptides: conformational space annealing. *J Comput Chem*. 1997;18:1222–1232.
42. Yuan Y, Pei J, Lai L. Ligbuilder V3: a multi-target de novo drug design approach. *Front Chem*. 2020;8:142.
43. Jeon W, Kim D. Autonomous molecule generation using reinforcement learning and docking to develop potential novel inhibitors. *Sci Rep*. 2020;10:1–11.
44. Yang S, Hwang D, Lee S, Ryu S, Hwang SJ. Hit and lead discovery with explorative RL and fragment-based molecule generation. *Adv Neural Inf Process Syst*. 2021;34:1–13.
45. Blaschke T, Arús-Pous J, Chen H, Margreitter C, Tyrchan C, Engkvist O, et al.. REINVENT 2.0: an AI tool for de novo drug design. *J Chem Inf Model*. 2020;60:5918–5922.
46. Fialková V, Zhao J, Papadopoulos K, Engkvist O, Bjerrum EJ, Kogej T, et al.. LibINVENT: Reaction-based generative scaffold decoration for in silico library design. *J Chem Inf Model*. 2022;62:2046–2063.
47. Guo J, Knuth F, Margreitter C, et al. Link-INVENT: generative linker design with reinforcement learning. *ChemRxiv*. Published online April 25, 2022. https://doi.org/10.26434/chemrxiv-2022-qkx9f.
48. Guo J, Fialková V, Arango JD, Margreitter C, Janet JP, Papadopoulos K, et al.. Improving de novo molecular design with curriculum learning. *Nat Mach Intell*. 2022;4:555–563.
49. Guo J, Janet JP, Bauer MR, Nittinger E, Giblin KA, Papadopoulos K, et al.. DockStream: a docking wrapper to enhance de novo molecular design. *J Cheminform*. 2021;13:1–21.
50. Nigam A, Pollice R, Krenn M, Passos Gomes G, Aspuru-Guzik A. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. *Chem Sci*. 2021;12:7079–7090.
51. Xu Z, Wauchope OR, Frank AT. Navigating chemical space by interfacing generative artificial intelligence and molecular docking. *J Chem Inf Model*. 2021;61:5589–5600.
52. Boitreaud J, Mallet V, Oliver C, Waldispuhl J. OptiMol: optimization of binding affinities in chemical space for drug discovery. *J Chem Inf Model*. 2020;60:5658–5666.
53. Ghanakota P, Bos PH, Konze KD, Staker J, Marques G, Marshall K, et al.. Combining cloud-based free-energy calculations, synthetically aware enumerations, and goal-directed generative machine learning for rapid large-scale chemical exploration and optimization. *J Chem Inf Model*. 2020;60:4311–4325.
54. Konze KD, Bos PH, Dahlgren MK, Leswing K, Tubert-Brohman I, Bortolato A, et al.. Reaction-based enumeration, active learning, and free energy calculations to rapidly explore synthetically tractable chemical space and optimize potency of cyclin- dependent kinase 2 inhibitors. *J Chem Inf Model*. 2019;59:3782–3793.
55. Srinivasan S, Batra R, Chan H, Kamath G, Cherukara MJ, Sankaranarayanan SKRS. Artificial intelligence-guided de novo molecular design targeting COVID-19. *ACS Omega*. 2021;6:12557–12566.
56. Arcidiacono M, Koes DR. MOLUCINATE: a generative model for molecules in 3D space. *arXiv*. 2019, arXiv:2109.15308v2.
57. Imrie F, Bradley AR, Schaar M, Deane CM. Deep generative models for 3D linker design. *J Chem Inf Model*. 2020;60:1983–1995.
58. Gebauer NWA, Gastegger M, Hessmann SSP, Müller KR, Schütt KT. Inverse design of 3D molecular structures with conditional generative neural networks. *Nat Commun*. 2022;13:1–11.
59. Moret M, Friedrich L, Grisoni F, Merk D, Schneider G. Generative molecular design in low data regimes. *Nat Mach Int*. 2020;2:171–180.
60. Grisoni F, Huisman BJH, Button AL, Moret M, Atz K, Merk D, et al.. Combining generative artificial intelligence and on-chip synthesis for de novo drug design. *Sci Adv*. 2021;7:eabg3338.