Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models

Daniil Polykovskiy[1], **Alexander Zhebrak**[1], **Benjamin Sanchez-Lengeling**[2], **Sergey Golovanov**[3], **Oktai Tatanov**[3], **Stanislav Belyaev**[3], **Rauf Kurbanov**[3], **Aleksey Artamonov**[3], **Vladimir Aladinskiy**[1], **Mark Veselov**[1], **Artur Kadurin**[1], **Simon Johansson**[4], **Hongming Chen**[4], **Sergey Nikolenko**[1,3,5], **Alán Aspuru-Guzik**[6,7,8] **and Alex Zhavoronkov**[1]

[1]Insilico Medicine Hong Kong Ltd, Pak Shek Kok, New Territories, Hong Kong
[2]Chemistry and Chemical Biology Department, Harvard University, Cambridge, MA 02143 USA
[3]Neuromation OU, Tallinn, 10111 Estonia
[4]Hit discovery, Discovery Sciences, Biopharmaceutics R&D, AstraZeneca Gothenburg, Sweden
[5]National Research University Higher School of Economics, St. Petersburg, 190008, Russia
[6]Department of Chemistry and Department of Computer Science, University of Toronto, Toronto, Ontario M5S 3H6, Canada
[7]Vector Institute for Artificial Intelligence, Toronto, Ontario M5S 1M1, Canada
[8]Biologically-Inspired Solar Energy Program, Canadian Institute for Advanced Research (CIFAR), Toronto, Ontario M5S 1M1, Canada

Abstract

Generative models are becoming the tools of choice for the discovery of new molecules and materials. Such models are able to learn on a large collection of molecular structures and produce novel compounds. In this work, we introduce a benchmarking platform called Molecular Sets (MOSES) to support research on generative models for drug discovery. MOSES provides training and testing datasets and a set of metrics to evaluate the quality and diversity of generated structures. We have implemented and compared several molecular generation models and suggest to use our results as reference points for further advancements in generative chemistry research. The platform and source code are available at https://github.com/molecularsets/moses.

1 Introduction

The discovery of new molecules for drugs and materials can bring enormous societal and technological progress, potentially curing rare diseases and providing a pathway for personalized precision medicine [1]. However, complete exploration of the huge space of potential chemicals is computationally intractable; it has been estimated that the number of pharmacologically-sensible molecules is in the order of 10^{23} to 10^{80} compounds [2, 3]. Often, this search is constrained based on already discovered structures and desired qualities such as solubility or toxicity. There have been many approaches to exploring the chemical space *in silico* and *in vitro*, including high throughput screening, combinatorial libraries, and evolutionary algorithms [4–7]. Recent works demonstrated that machine learning methods can produce new small molecules [8–11] and peptides [12] showing biological activity.

Over the last few years, advances in machine learning, and especially in deep learning, have driven the design of new computational systems for modeling increasingly complex phenomena. One approach that has been proven fruitful for the modeling molecular data has been deep generative models. Deep generative models have found applications in a wide range of settings, from generating synthetic images [13] and natural language texts [14], to the applications in biomedicine, including

the design of DNA sequences [15], and aging research [16]. One important field of application for deep generative models lies in the inverse design of drug compounds [17] for a given functionality (solubility, ease of synthesis, toxicity). Deep learning also found other applications in biomedicine [18, 19], including target identification [20], antibacterial drug discovery [21], and drug repurposing [22, 23].
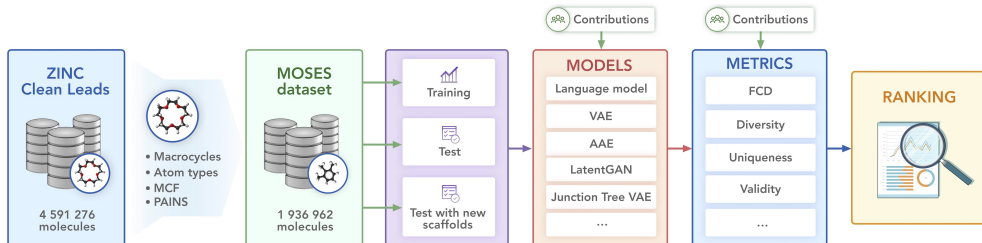

Figure 1: Molecular Sets (MOSES) pipeline. The open-source library provides a dataset, baseline models, and evaluation metrics.

Part of the success of deep learning in different fields has been driven by ever-growing availability of large datasets and standard benchmark sets. These sets serve as a common measuring stick for newly developed models and optimization strategies [24, 25]. In the context of organic molecules, MoleculeNet [26] was introduced as a standardized benchmark suite for regression and classification tasks. Brown et al. [27] proposed to evaluate generative models on goal-oriented and distribution learning tasks with a focus on the former. We focus on standardizing metrics and data for the distribution learning problem.

It is not immediately clear how to compare generative models and assess the quality of generated samples; in particular, the loss function of deep generative models is seldom a good choice for the final evaluation. Often the real validation of a candidate molecule requires experimental characterization either via multiple *in-vitro* experiments or a device prototype. Computational approaches for scoring molecules tend to suffer from a lack of accuracy (e.g., simulating target-ligand interaction) or are too time-consuming (e.g., simulating the electronic structure). In this work, we provide a benchmark suite for molecular generation, including a standardized dataset, data preprocessing utilities, evaluation metrics, and molecular generation models. We hope that this platform will serve as a clear and unified testbed for current and future generative models. We illustrate the main components our the platform—Molecular Sets (MOSES)—in Figure 1.

2 Distribution learning

In MOSES, we study distribution learning models. Formally, given a set of training samples $X_{\mathrm{tr}} = \{x_1^{\mathrm{tr}}, \ldots, x_N^{\mathrm{tr}}\}$ from an unknown distribution $p(x)$, distribution learning models approximate $p(x)$ with some distribution $q(x)$. The quality of a distribution learning model is some measure of deviation between $p(x)$ and $q(x)$. The model can either define a probability mass function $q(x)$ implicitly or explicitly. Explicit models such as Hidden Markov Models, n-gram language models, or normalizing flows [28, 29] can compute $q(x)$ and sample from it. Implicit models, such as variational autoencoders, adversarial autoencoders, or generative adversarial networks [30–32] can sample from $q(x)$, but can not compute the exact values of $q(x)$. To compare both kinds of models, evaluation metrics that we considered rely solely on samples from $q(x)$.

Generative models can represent molecular structures in different forms, including graphs, strings, or point clouds. For the review on representations of molecular structures, please refer to Appendix A.

3 Metrics

We compare generative models using metrics that depend on a generated set G and a test (reference) set R: fragment similarity, scaffold similarity, nearest neighbor similarity, internal diversity, and

2

Fréchet ChemNet Distance. We compute these metrics (except for validity) only for valid molecules from the generated set. We define the validity using RDKit's molecular structure parser that considers rules such as proper atom's valency and constraints on aromatic structures. We suggest generating 30,000 molecules and obtaining G as valid molecules from this set, since some metrics depend on the size of G.

Fraction of valid (Valid) and unique (Unique@k) molecules report validity of uniqueness of the generated SMILES strings. In the experiments, we compute Unique@K and for the first $K = 1{,}000$ and $K = 10{,}000$ valid molecules in the generated set. If the number of valid molecules is less than K, we compute uniqueness on all valid molecules.

Novelty is the fraction of the generated molecules that are not present in the training set. Low novelty indicates overfitting.

Fragment similarity (Frag) compares distributions of BRICS fragments [33] in generated and reference sets. Denoting $c_f(A)$ a number of times a substructure f appears in molecules from set A, and a set of fragments that appear in either G or R as F, the metric is defined as a cosine similarity:

$$\text{Frag}(G, R) = \frac{\sum\limits_{f \in F} \Big(c_f(G) \cdot c_f(R) \Big)}{\sqrt{\sum\limits_{f \in F} c_f^2(G)} \sqrt{\sum\limits_{f \in F} c_f^2(R)}}. \tag{1}$$

If molecules in both sets have similar fragments, Frag metric is large. If some fragments are over- or underrepresented (or never appear) in the generated set, the metric will be lower. Limits of this metric are $[0, 1]$.

Scaffold similarity (Scaff) is similar to fragment similarity metric, but instead of fragments we compare frequencies of Bemis–Murcko scaffolds [34]. Bemis–Murcko scaffold contains all molecule's ring structures and linker fragments connecting rings. We use RDKit implementation of this algorithm which additionally considers carbonyl groups attached to rings as parts of a scaffold. Denoting $c_s(A)$ a number of times a scaffold s appears in molecules from set A, and a set of fragments that appear in either G or R as S, the metric is defined as a cosine similarity:

$$\text{Frag}(G, R) = \frac{\sum\limits_{s \in S} \Big(c_s(G) \cdot c_s(R) \Big)}{\sqrt{\sum\limits_{s \in S} c_s^2(G)} \sqrt{\sum\limits_{s \in S} c_s^2(R)}}. \tag{2}$$

The purpose of this metric is to show how similar are the distributions of scaffolds present in generated and reference datasets. For example, if the model rarely produces molecules of a certain chemotype present in a reference set, the metric will be low. Limits of this metric are $[0, 1]$.

Note that both fragment and scaffold similarities compare molecules at a substructure level. Hence, it is possible to have a similarity 1 even when G and R contain different molecules.

Similarity to a nearest neighbor (SNN) is an average Tanimoto similarity $T(m_G, m_R)$ (also known as the Jaccard index) between fingerprints of a molecule m_G from the generated set G and its nearest neighbor molecule m_R in the reference dataset R:

$$\text{SNN}(G, R) = \frac{1}{|G|} \sum_{m_G \in G} \max_{m_R \in R} T(m_G, m_R), \tag{3}$$

In this work, we used standard Morgan fingerprints [35] with radius 2 and 1024 bits computed using RDKit library [36]. The resulting similarity metric can be interpreted as precision: if generated molecules are far from the manifold of the reference set, similarity to the nearest neighbor will be low. Limits of this metric are $[0, 1]$.

Internal diversity (IntDiv$_p$) [37] assesses the chemical diversity within the generated set of molecules G.

$$\text{IntDiv}_p(G) = 1 - \sqrt[p]{\frac{1}{|G|^2} \sum_{m_1, m_2 \in G} T(m_1, m_2)^p}. \tag{4}$$

This metric detects a common failure case of generative models—mode collapse. With mode collapse, the model produces a limited variety of samples, ignoring some areas of the chemical space. A higher

value of this metric corresponds to higher diversity in the generated set. In the experiments, we report $\text{IntDiv}_1(G)$ and $\text{IntDiv}_2(G)$. Limits of this metric are $[0, 1]$.

Fréchet ChemNet Distance (FCD) [38] is calculated using activations of the penultimate layer of a deep neural network ChemNet trained to predict biological activities of drugs. We compute activations for canonic SMILES (simplified molecular line entry system) [39] string representations of molecules. These activations capture both chemical and biological properties of the compounds. For two sets of molecules G and R, FCD is defined as

$$\text{FCD}(G, R) = \|\mu_G - \mu_R\|^2 + \text{Tr}\left(\Sigma_G + \Sigma_R - 2(\Sigma_G \Sigma_R)^{1/2}\right) \tag{5}$$

where μ_G, μ_R are mean vectors and Σ_G, Σ_R are full covariance matrices of activations for molecules from sets G and R respectively. FCD correlates well with other metrics. For example, if the generated molecules are not diverse enough (low IntDiv_p) or the model produces too many duplicates (low uniqueness), FCD will likely decrease, since Σ_G is smaller. We suggest to use FCD for hyperparameter tuning. Values of this metric are non-negative.

Properties distribution is a useful tool for visually assessing the generated set of molecules. To quantitatively compare the distributions in the generated and test sets, we compute a 1D Wasserstein-1 distance between property distributions of generated and test sets. We also visualize a kernel density estimation of these distributions in the Experiments section. We use the following four properties:

- **Molecular weight (MW)**: the sum of atomic weights in a molecule. By plotting histograms of molecular weight for the generated and test sets, one can judge if a generated set is biased towards lighter or heavier molecules.

- **LogP**: the water-octanol partition coefficient, a ratio of a chemical's concentration in the octanol phase to its concentration in the aqueous phase of a two-phase octanol/water system; computed with RDKit's Crippen [40] estimation.

- **Synthetic Accessibility Score (SA)**: a heuristic estimate of how hard (10) or how easy (1) it is to synthesize a given molecule. SA score is based on a combination of the molecule's fragments contributions [41]. Note that SA score does not adequately assess up-to-date chemical structures, but it useful for assessing distribution learning models.

- **Quantitative Estimation of Drug-likeness (QED)**: a 0 to 1 float value estimating how likely a molecule is a viable candidate for a drug. QED is meant to capture the abstract notion of aesthetics in medicinal chemistry [42]. Similar to SA, descriptor limits in QED have been changing during the last decade and current limits may not cover latest drugs [43].

4 Dataset

The proposed dataset used for training and testing the models is based on the ZINC Clean Leads collection which contains $4{,}591{,}276$ molecules with molecular weight in the range from 250 to 350 Daltons, a number of rotatable bonds not greater than 7, and XlogP [44] not greater then 3.5. We removed molecules containing charged atoms, atoms besides C, N, S, O, F, Cl, Br, H, or cycles larger than 8 atoms. The molecules were filtered via custom medicinal chemistry filters (MCFs) and PAINS filters [45]. We describe MCF filters and discuss PAINS in Appendix B. We removed charged molecules to avoid ambiguity with tautomers and pH conditions. Note that in the initial set of molecules, functional groups were present in both ionized and unionized forms.

The final dataset contains $1{,}936{,}963$ molecules, with internal diversity $\text{IntDiv}_1 = 0.857$; it contains $448{,}854$ unique Bemis-Murcko [34] scaffolds and $58{,}315$ unique BRICS [33] fragments. We show examples of obtained molecules in Figure 2 and a representative diverse subset in Appendix C. We provide recommended split into three non-intersecting parts: train ($1{,}584{,}664$ molecules), test ($176{,}075$ molecules) and scaffold test ($176{,}226$ molecules). The scaffold test set has all molecules containing a Bemis-Murcko scaffold from a random subset of scaffolds. Hence, scaffolds from the scaffold test set differ from scaffolds in both train and test sets. We use scaffold test split to assess whether a model can produce novel scaffolds absent in the training set. The test set is a random subset of the remaining molecules in the dataset.
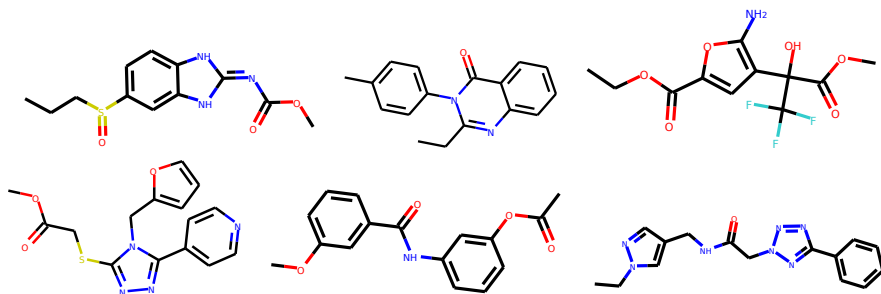
Figure 2: Examples of molecules from MOSES dataset.

5 Baselines

In the current version of MOSES, we implemented several models that cover different approaches to molecular generation, such as character-level recurrent neural networks (CharRNN) [46, 38], Variational Autoencoders (VAE) [30, 31, 47], Adversarial Autoencoders (AAE) [31, 11], Junction Tree Variational Autoencoders (JTN-VAE) [48], LatentGAN [49], and non-neural baselines.

Model comparison can be challenging since different training parameters (number of epochs, batch size, learning rate, initial state, optimizer) and architecture hyperparameters (hidden layer dimension, number of layers, etc.) can significantly alter their performance. For each model, we attempted to preserve its original architecture as published and tuned the hyperparameters to improve the performance. We used random search over multiple architectures for every model and selected the architecture that produced the best value of FCD. Models are implemented in Python 3 utilizing PyTorch [50] framework. We briefly introduce these models below. Please refer to the D for the training details and hyperparameters.

Character-level recurrent neural network (CharRNN) is a language model for learn the distribution over SMILES strings. The model learns SMILES syntax implicitly from the training set [46].

Variational autoencoder (VAE) [51] consists of two neural networks—an encoder and a decoder—that infer a mapping from high-dimensional data representation into a lower-dimensional space and back. The lower-dimensional space is called the latent space, which is often a continuous vector space with normal prior distribution. VAE parameters are optimized to encode and decode data by minimizing reconstruction loss and regularization term in a form of Kullback-Leibler divergence. VAE-based architecture for the molecular generation was studied in multiple previous works Kadurin et al. [31], Blaschke et al. [47], Gómez-Bombarelli et al. [30]. We combine aspects from these implementations in MOSES.

Adversarial Autoencoder (AAE) [52] replaces the Kullback-Leibler divergence with an adversarial objective. An auxiliary discriminator network is trained to distinguish samples from a prior distribution and model's latent codes. The encoder then adapts its latent codes to minimize discriminator's predictive accuracy. The training process oscillates between training the encoder-decoder pair and the discriminator. Kullback-Leibler divergence has a closed-form analytical solution only for a handful of distributions, while a discriminator can be used for any prior distribution. AAE-based models for molecular design were studied in [53, 31, 11].

Junction Tree VAE (JTN-VAE) [48] generates molecules in two phases by exploiting valid subgraphs as components. In the first phase, it generates a tree-structured object (a junction tree) whose role is to represent the scaffold of subgraph components and their coarse relative arrangements. The components are valid chemical substructures automatically extracted from the training set using tree decomposition and are then used as building blocks. In the second phase, the subgraphs (nodes of the tree) are assembled together into a coherent molecular graph.

Latent Vector Based Generative Adversarial Network (LatentGAN) [49] combines an autoencoder and a generative adversarial network architectures. This approach first pretrains an autoencoder to map SMILES structures onto latent vectors. Then a generative adversarial network is trained to produce latent vectors for the pre-trained decoder.

Table 1: Performance metrics for baseline models: fraction of valid molecules, fraction of unique molecules from 1,000 and 10,000 molecules. Reported (mean ± std) over three independent model initializations.

Model	Valid (↑)	Unique@1k (↑)	Unique@10k (↑)
Train	*1.0*	*1.0*	*1.0*
HMM	0.076 ± 0.0322	0.623 ± 0.1224	0.5671 ± 0.1424
NGram	0.2376 ± 0.0025	0.974 ± 0.0108	0.9217 ± 0.0019
Combinatorial	**1.0**	0.9983 ± 0.0015	0.9909 ± 0.0009
CharRNN	0.975 ± 0.026	**1.0** ± 0.0	**0.999** ± 0.0
VAE	0.977 ± 0.001	**1.0** ± 0.0	0.998 ± 0.001
AAE	0.937 ± 0.034	**1.0** ± 0.0	0.997 ± 0.002
JTN-VAE	**1.0**	**1.0**	**0.999**
LatentGAN	0.897 ± 0.002	**1.0** ± 0.0	0.997 ± 0.005

Non-neural baselines implemented in MOSES are n-gram generative model, Hidden Markov Model (HMM), and a combinatorial generator. N-gram model collects statistics of n-grams frequencies in the training set and uses such distribution to sequentially sample new strings. Hidden Markov models utilize Baum-Welch algorithm to learn a probabilistic distribution over the SMILES strings. The model consists of several states $(s_1, \ldots, s_K)$, transition probabilities between states $p(s_{i+1} \mid s_i)$, and token emission probabilities $p(x_i \mid s_i)$. Beginning from a "start" state, at each iteration the model samples a next token and state from emission and transition probabilities correspondingly. A combinatorial generator separates molecules of the training set into BRICS fragments and randomly connects them to form a new molecule. We sample fragments according to their frequencies in the training set to model the distribution better.

5.1 Platform

The dataset, metrics and baseline models are provided in a GitHub repository `https://github.com/molecularsets/moses` and as a PyPI package `molsets`. To contribute a new model, one should train a model on MOSES train set, generate 30,000 samples and compute metrics using the provided utilities. We recommend running the experiment at least three times with different random seeds to estimate sensitivity of the model to random parameter initialization.

6 Results

We trained the baseline models on MOSES train set and provide results in this section. In Table 1 we compare models with respect to the validity and uniqueness metrics. Hidden Markov Model and NGram models fail to produce valid molecules. Combinatorial generator and LatentGAN have a built-in validity constraints, so their validity is 100%.

Table 2 reports additional properties of the generated set: fraction of molecules passing filters, fraction of molecules not present in the training set, and internal diversity. Combinatorial generator has higher diversity than the training dataset, which might be favorable for discovering new chemical structures. Autoencoder-based models show low novelty, indicating that these models could overfit on the training set.

Table 3 reports Fréchet ChemNet Distance (FCD) and similarity to a nearest neighbor (SNN). All neural network-based models show low FCD, indicating that the models successfully captured the statistics of the dataset. Surprisingly, a simple language model, character level RNN, shows the best results in terms of the FCD measure. Variational autoencoder (VAE) showed best results in terms of SNN, although combined with low novelty we suppose that the model significantly overfitted on the training set.

In Table 4 we report similarities of substructure distributions—fragments and scaffolds. Scaffold similarity from the training set to the scaffold test set (TestSF) is zero by design. However, CharRNN successfully discovered many novel scaffolds (11%), suggesting that the model generalizes well.

Table 2: Performance metrics for baseline models: fraction of molecules passing filters (MCF, PAINS, ring sizes, charge, atom types), novelty, and internal diversity. Reported (mean $\pm$ std) over three independent model initializations.

Model	Filters ($\uparrow$)	Novelty ($\uparrow$)	IntDiv$_1$	IntDiv$_2$
Train	*1.0*	*0.0*	*0.857*	*0.851*
HMM	0.9024 ± 0.0489	$\mathbf{0.9994} \pm \mathbf{0.001}$	0.8466 ± 0.0403	0.8104 ± 0.0507
NGram	0.9582 ± 0.001	0.9694 ± 0.001	$\mathbf{0.8738} \pm \mathbf{0.0002}$	0.8644 ± 0.0002
Combinatorial	0.9557 ± 0.0018	0.9878 ± 0.0008	0.8732 ± 0.0002	$\mathbf{0.8666} \pm \mathbf{0.0002}$
CharRNN	0.994 ± 0.003	0.842 ± 0.051	0.856 ± 0.0	0.85 ± 0.0
VAE	$\mathbf{0.997} \pm \mathbf{0.0}$	0.695 ± 0.007	0.856 ± 0.0	0.85 ± 0.0
AAE	0.996 ± 0.001	0.793 ± 0.028	0.856 ± 0.003	0.85 ± 0.003
JTN-VAE	0.978	0.915	0.851	0.845
LatentGAN	0.973 ± 0.001	0.949 ± 0.001	0.857 ± 0.0	0.85 ± 0.0

Table 3: Performance metrics for baseline models: Fréchet ChemNet Distance (FCD) and Similarity to a nearest neighbor (SNN); Reported (mean $\pm$ std) over three independent model initializations. Results for random test set (Test) and scaffold split test set (TestSF).

Model	FCD ($\downarrow$)		SNN ($\uparrow$)	
	Test	TestSF	Test	TestSF
Train	*0.008*	*0.476*	*0.642*	*0.586*
HMM	24.4661 ± 2.5251	25.4312 ± 2.5599	0.3876 ± 0.0107	0.3795 ± 0.0107
NGram	5.5069 ± 0.1027	6.2306 ± 0.0966	0.5209 ± 0.001	0.4997 ± 0.0005
Combinatorial	4.2375 ± 0.037	4.5113 ± 0.0274	0.4514 ± 0.0003	0.4388 ± 0.0002
CharRNN	$\mathbf{0.073} \pm \mathbf{0.025}$	$\mathbf{0.52} \pm \mathbf{0.038}$	0.601 ± 0.021	0.565 ± 0.014
VAE	0.099 ± 0.013	0.567 ± 0.034	$\mathbf{0.626} \pm \mathbf{0.0}$	$\mathbf{0.578} \pm \mathbf{0.001}$
AAE	0.556 ± 0.203	1.057 ± 0.237	0.608 ± 0.004	0.568 ± 0.005
JTN-VAE	0.422	0.996	0.556	0.527
LatentGAN	0.296 ± 0.021	0.824 ± 0.030	0.538 ± 0.001	0.514 ± 0.009

Finally, we compared distributions of four molecular properties in generated and test sets (Figure 3): molecular weight (MW), water-octanol partition coefficient (logP), quantitative estimation of drug-likeness (QED) and synthetic accessibility score (SA). Deep generative models closely match the data distribution; hidden Markov Model is biased towards lighter molecules, which is consistent with low validity: larger molecules impose more validity constraints. A combinatorial generator has higher variance in molecular weight, producing larger and smaller molecules than those present in the training set.

7 Discussion

With MOSES, we have designed a molecular generation benchmark platform that provides a dataset with molecular structures, an implementation of baseline models, and metrics for their evaluation. While standardized comparative studies and test sets are essential for the progress of machine learning applications, the current field of de-novo drug design lacks evaluation protocols for generative machine learning models. Being on the intersection of mathematics, computer science, and chemistry, these applications are often too challenging to explore for research scientists starting in the field. Hence, it is necessary to develop a transparent approach to implementing new models and assessing their performance. We presented a benchmark suite with unified and extendable programming interfaces for generative models and evaluation metrics.

This platform should allow for a fair and comprehensive comparison of new generative models. For future work on this project, we will keep extending the MOSES repository with new baseline models and new evaluation metrics. We hope this work will attract researchers interested in tackling drug discovery challenges.

Table 4: Fragment similarity (Frag), Scaffold similarity (Scaff). Reported (mean ± std) over three independent model initializations. Results for random test set (Test) and scaffold split test set (TestSF).

| Model	Frag (↑)		Scaf (↑)	
	Test	TestSF	Test	TestSF
Train	*1.0*	*0.999*	*0.991*	*0.0*
HMM	$0.5754_{\pm 0.1224}$	$0.5681_{\pm 0.1218}$	$0.2065_{\pm 0.0481}$	$0.049_{\pm 0.018}$
NGram	$0.9846_{\pm 0.0012}$	$0.9815_{\pm 0.0012}$	$0.5302_{\pm 0.0163}$	$0.0977_{\pm 0.0142}$
Combinatorial	$0.9912_{\pm 0.0004}$	$0.9904_{\pm 0.0003}$	$0.4445_{\pm 0.0056}$	$0.0865_{\pm 0.0027}$
CharRNN	$\mathbf{1.0}_{\pm \mathbf{0.0}}$	$\mathbf{0.998}_{\pm \mathbf{0.0}}$	$0.924_{\pm 0.006}$	$\mathbf{0.11}_{\pm \mathbf{0.008}}$
VAE	$0.999_{\pm 0.0}$	$\mathbf{0.998}_{\pm \mathbf{0.0}}$	$\mathbf{0.939}_{\pm \mathbf{0.002}}$	$0.059_{\pm 0.01}$
AAE	$0.991_{\pm 0.005}$	$0.99_{\pm 0.004}$	$0.902_{\pm 0.037}$	$0.079_{\pm 0.009}$
JTN-VAE	0.996	0.995	0.892	0.1
LatentGAN	$0.999_{\pm 0.003}$	$\mathbf{0.998}_{\pm \mathbf{0.003}}$	$0.886_{\pm 0.015}$	$0.1_{\pm 0.006}$
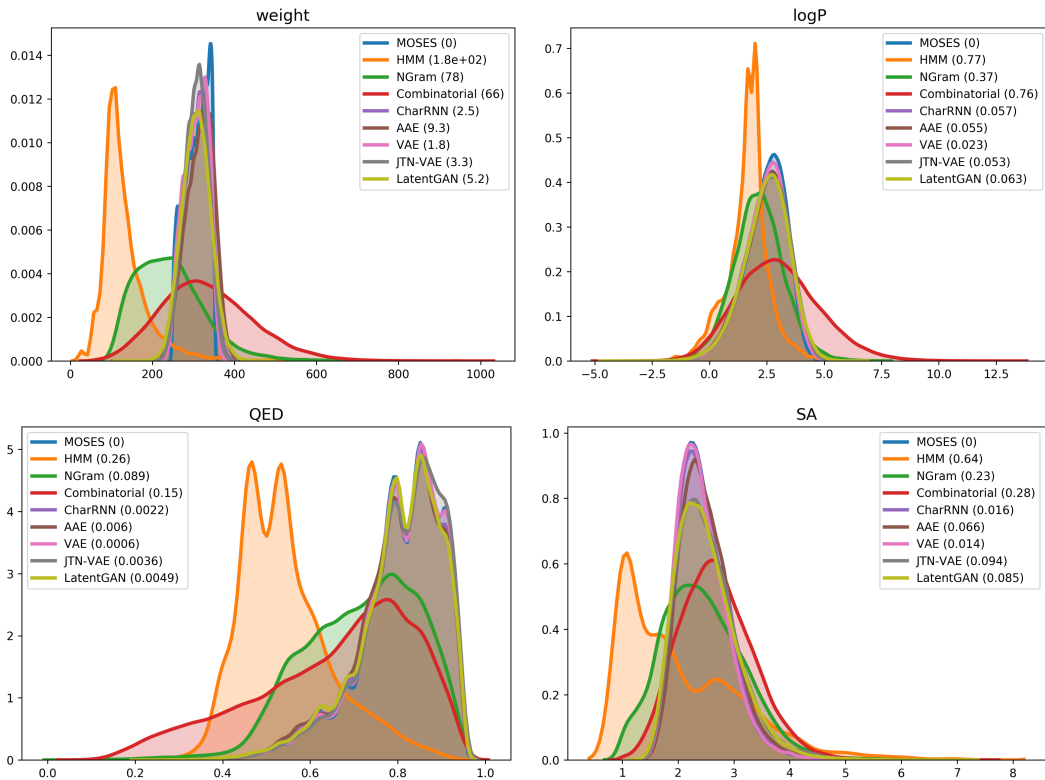
Figure 3: Distribution of chemical properties for MOSES dataset and sets of generated molecules. In brackets—Wasserstein-1 distance to MOSES test set. Parameters: molecular weight, water-octanol partition coefficient (logP), quantitative estimation of drug-likeness (QED) and synthetic accessibility score (SA).

Conflicts of interest

for, Kebotix, an artificial intelligence-driven molecular discovery company and a member of the science advisory board of Insilico Medicine.

Author Contributions

D.P., A.Zhebrak, S.G., O.T., S.B., R.K., A.A., A.K., S.J., H.C. designed and conducted the experiments; D.P., A.Zhebrak, B.S-L., V.A., M.V., S.J., H.C., S.N., A.A.-G., A.Zhavoronkov wrote the manuscript.

Correspondence

Correspondence to: daniil@insilico.com,   alex@insilico.com,   alan@aspuru.com, snikolenko@gmail.com.

References

[1] Su-In Lee, Safiye Celik, Benjamin A Logsdon, Scott M Lundberg, Timothy J Martins, Vivian G Oehler, Elihu H Estey, Chris P Miller, Sylvia Chien, Jin Dai, Akanksha Saxena, C Anthony Blau, and Pamela S Becker. A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia. *Nat. Commun.*, 9(1):42, January 2018.

[2] Jean-Louis Reymond. The chemical space project. *Acc. Chem. Res.*, 48(3):722–730, 2015.

[3] Peter Kirkpatrick and Clare Ellis. Chemical space. *Nature*, 432(7019):823–823, December 2004.

[4] Stefano Curtarolo, Gus L W Hart, Marco Buongiorno Nardelli, Natalio Mingo, Stefano Sanvito, and Ohad Levy. The high-throughput highway to computational materials design. *Nat. Mater.*, 12(3):191–201, March 2013.

[5] Xiangqian Hu, David N Beratan, and Weitao Yang. Emergent strategies for inverse molecular design. *Sci. China B*, 52(11):1769–1776, November 2009.

[6] Tu C Le and David A Winkler. Discovery and optimization of materials using evolutionary approaches. *Chem. Rev.*, 116(10):6107–6132, May 2016.

[7] Edward O Pyzer-Knapp, Changwon Suh, Rafael Gómez-Bombarelli, Jorge Aguilera-Iparraguirre, and Alán Aspuru-Guzik. What is High-Throughput virtual screening? a perspective from organic materials discovery. *Annu. Rev. Mater. Res.*, 45(1):195–216, 2015.

[8] Alex Zhavoronkov, Yan A Ivanenkov, Alex Aliper, Mark S Veselov, Vladimir A Aladinskiy, Anastasiya V Aladinskaya, Victor A Terentiev, Daniil A Polykovskiy, Maksim D Kuznetsov, Arip Asadulaev, et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature biotechnology*, pages 1–4, 2019.

[9] Daniel Merk, Lukas Friedrich, Francesca Grisoni, and Gisbert Schneider. De novo design of bioactive small molecules by artificial intelligence. *Molecular informatics*, 37(1-2):1700153, 2018.

[10] Daniel Merk, Francesca Grisoni, Lukas Friedrich, and Gisbert Schneider. Tuning artificial intelligence on the de novo design of natural-product-inspired retinoid x receptor modulators. *Communications Chemistry*, 1(1):68, 2018.

[11] Daniil Polykovskiy, Alexander Zhebrak, Dmitry Vetrov, Yan Ivanenkov, Vladimir Aladinskiy, Polina Mamoshina, Marine Bozdaganyan, Alexander Aliper, Alex Zhavoronkov, and Artur Kadurin. Entangled conditional adversarial autoencoder for de novo drug discovery. *Mol. Pharm.*, September 2018.

[12] Francesca Grisoni, Claudia S Neuhaus, Gisela Gabernet, Alex T Müller, Jan A Hiss, and Gisbert Schneider. Designing anticancer peptides by constructive machine learning. *ChemMed-Chem*, 13(13):1300–1302, 2018.

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations*, 2018.

[14] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[15] Nathan Killoran, Leo J Lee, Andrew Delong, David Duvenaud, and Brendan J Frey. Generating and designing dna with deep generative models. *Neural Information Processing Systems 2017 Computational Biology Workshop*, 2017.

[16] Alex Zhavoronkov, Polina Mamoshina, Quentin Vanhaelen, Morten Scheibye-Knudsen, Alexey Moskalev, and Alex Aliper. Artificial intelligence for aging and longevity research: Recent advances and perspectives. *Ageing Research Reviews*, 49:49–66, jan 2019. doi: 10.1016/j.arr. 2018.11.003. URL https://doi.org/10.1016/j.arr.2018.11.003.

[17] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, July 2018.

[18] Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, Wei Xie, Gail L. Rosen, Benjamin J. Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne E. Carpenter, Avanti Shrikumar, Jinbo Xu, Evan M. Cofer, Christopher A. Lavender, Srinivas C. Turaga, Amr M. Alexandari, Zhiyong Lu, David J. Harris, Dave DeCaprio, Yanjun Qi, Anshul Kundaje, Yifan Peng, Laura K. Wiley, Marwin H. S. Segler, Simina M. Boca, S. Joshua Swamidass, Austin Huang, Anthony Gitter, and Casey S. Greene. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, apr 2018. doi: 10.1098/rsif.2017.0387. URL https://doi.org/10.1098/rsif.2017.0387.

[19] Polina Mamoshina, Armando Vieira, Evgeny Putin, and Alex Zhavoronkov. Applications of deep learning in biomedicine. *Molecular Pharmaceutics*, 13(5):1445–1454, mar 2016.

[20] Polina Mamoshina, Marina Volosnikova, Ivan V. Ozerov, Evgeny Putin, Ekaterina Skibina, Franco Cortese, and Alex Zhavoronkov. Machine learning on human muscle transcriptomic data for biomarker discovery and tissue-specific drug target identification. *Frontiers in Genetics*, 9:242, 2018. ISSN 1664-8021. doi: 10.3389/fgene.2018.00242. URL https://www.frontiersin.org/article/10.3389/fgene.2018.00242.

[21] Yan A Ivanenkov, Alex Zhavoronkov, Renat S Yamidanov, Ilya A Osterman, Petr V Sergiev, Vladimir A Aladinskiy, Anastasia V Aladinskaya, Victor A Terentiev, Mark S Veselov, Andrey A Ayginin, et al. Identification of novel antibacterials using machine learning techniques. *Frontiers in pharmacology*, 10, 2019.

[22] Quentin Vanhaelen, Polina Mamoshina, Alexander M Aliper, Artem Artemov, Ksenia Lezhnina, Ivan Ozerov, Ivan Labat, and Alex Zhavoronkov. Design of efficient computational workflows for in silico drug repurposing. *Drug Discovery Today*, 22(2):210–222, 2017.

[23] A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina, and A. Zhavoronkov. Deep Learning Applications for Predicting Pharmacological Properties of Drugs and Drug Repurposing Using Transcriptomic Data. *Mol. Pharm.*, 13(7):2524–2530, 07 2016.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] J Deng, W Dong, R Socher, L-J Li, K Li, and L Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[26] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.*, 9(2):513–530, January 2018.

[27] Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

[28] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *International Conference on Learning Representations*, 2017.

[29] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *International Conference on Learning Representations*, 2019.

[30] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a Data-Driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, February 2018.

[31] Artur Kadurin, Alexander Aliper, Andrey Kazennov, Polina Mamoshina, Quentin Vanhaelen, Kuzma Khrabrov, and Alex Zhavoronkov. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8(7): 10883–10890, 2016. ISSN 1949-2553. doi: https://doi.org/10.18632/oncotarget.14073. URL https://www.oncotarget.com/article/14073/.

[32] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[33] J Degen, C Wegscheid-Gerlach, A Zaliani, and M Rarey. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem*, 3(10):1503–1507, 2008.

[34] G W Bemis and M A Murcko. The properties of known drugs. 1. molecular frameworks. *J. Med. Chem.*, 39(15):2887–2893, July 1996.

[35] H L Morgan. The generation of a unique machine description for chemical Structures-A technique developed at chemical abstracts service. *J. Chem. Doc.*, 5(2):107–113, 1965.

[36] Greg Landrum. RDKit: Open-source cheminformatics. http://www.rdkit.org/, 2006.

[37] Mostapha Benhenda. ChemGAN challenge for drug discovery: can AI reproduce natural chemical diversity? *CoRR*, abs/1708.08227, 2017.

[38] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. *J. Chem. Inf. Model.*, 58(9):1736–1741, September 2018.

[39] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28 (1):31–36, 1988.

[40] Scott A Wildman and Gordon M Crippen. Prediction of physicochemical parameters by atomic contributions. *J. Chem. Inf. Comput. Sci.*, 39(5):868–873, 1999.

[41] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.*, 1(1):8, June 2009.

[42] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nat. Chem.*, 4(2):90–98, January 2012.

[43] Michael D Shultz. Two decades under the influence of the rule of five and the changing properties of approved oral drugs: Miniperspective. *Journal of medicinal chemistry*, 62(4): 1701–1714, 2018.

[44] Renxiao Wang, Ying Fu, and Luhua Lai. A new atom-additive method for calculating partition coefficients. *Journal of chemical information and computer sciences*, 37(3):615–621, 1997.

[45] Jonathan B Baell and Georgina A Holloway. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.*, 53(7):2719–2740, April 2010.

[46] Marwin H S Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent Sci*, 4(1):120–131, January 2018.

[47] Thomas Blaschke, Marcus Olivecrona, Ola Engkvist, Jürgen Bajorath, and Hongming Chen. Application of generative autoencoder in de novo molecular design. *Mol. Inform.*, 37(1-2), January 2018.

[48] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2323–2332, Stockholmsmässan, Stockholm Sweden, 2018. PMLR.

[49] Oleksii Prykhodko, Simon Viet Johansson, Panagiotis-Christos Kotsias, Josep Arús-Pous, Esben Jannik Bjerrum, Ola Engkvist, and Hongming Chen. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics*, 11(1):74, 2019.

[50] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *NIPS workshop*, 2017.

[51] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*, 2013.

[52] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. *International Conference on Learning Representations*, 2016.

[53] Artur Kadurin, Sergey Nikolenko, Kuzma Khrabrov, Alex Aliper, and Alex Ahavoronkov. druGAN: An advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol. Pharm.*, 14(9):3098–3104, September 2017.

[54] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Sci Adv*, 4(7):eaap7885, July 2018.

[55] Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. Sequence tutor: Conservative Fine-Tuning of sequence generation models with KL-control. *arXiv*, November 2016.

[56] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-Reinforced generative adversarial networks (ORGAN) for sequence generation models. *arXiv*, May 2017.

[57] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *J. Cheminform.*, 9(1):48, December 2017.

[58] Seokho Kang and Kyunghyun Cho. Conditional molecular design with deep generative models. *J. Chem. Inf. Model.*, July 2018.

[59] Xiufeng Yang, Jinzhe Zhang, Kazuki Yoshizoe, Kei Terayama, and Koji Tsuda. ChemTS: an efficient python library for de novo molecular generation. *Sci. Technol. Adv. Mater.*, 18(1):972–976, November 2017.

[60] Evgeny Putin, Arip Asadulaev, Quentin Vanhaelen, Yan Ivanenkov, Anastasia V Aladinskaya, Alex Aliper, and Alex Zhavoronkov. Adversarial threshold neural computer for molecular de novo design. *Mol. Pharm.*, 15(10):4386–4397, October 2018.

[61] David Weininger, Arthur Weininger, and Joseph L Weininger. Smiles. 2. algorithm for generation of unique smiles notation. *Journal of chemical information and computer sciences*, 29(2):97–101, 1989.

[62] Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized smiles strings improve the quality of molecular generative models. *Journal of Cheminformatics*, 11 (1):1–13, 2019.

[63] Noel O'Boyle and Andrew Dalke. DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. *ChemRxiv*, 2018.

[64] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1945–1954, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/kusner17a.html.

[65] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. Syntax-directed variational autoencoder for structured data. *International Conference on Learning Representations*, 2018.

[66] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. Selfies: a robust representation of semantically constrained graphs with an example application in chemistry. *arXiv preprint arXiv:1905.13741*, 2019.

[67] Stephen E Stein, Stephen R Heller, and Dmitrii V Tchekhovskoi. An open standard for chemical structure representation: The iupac chemical identifier. In *International Chemical Information Conference*, 2003.

[68] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. Curran Associates, Inc., 2015.

[69] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[70] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022, August 2014.

[71] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[72] Anna Rybinska, Anita Sosnowska, Maciej Barycki, and Tomasz Puzyn. Geometry optimization method versus predictive ability in qspr modeling for ionic liquids. *Journal of computer-aided molecular design*, 30(2):165–176, 2016.

[73] Marc C Nicklaus, Shaomeng Wang, John S Driscoll, and George WA Milne. Conformational changes of small molecules binding to proteins. *Bioorganic & medicinal chemistry*, 3(4): 411–428, 1995.

[74] James H Nettles, Jeremy L Jenkins, Andreas Bender, Zhan Deng, John W Davies, and Meir Glick. Bridging chemical and biological space:"target fishing" using 2d and 3d molecular descriptors. *Journal of medicinal chemistry*, 49(23):6802–6810, 2006.

[75] Kunal Roy, Supratik Kar, and Rudra Narayan Das. *A primer on QSAR/QSPR modeling: fundamental concepts*. Springer, 2015.

[76] Fahimeh Ghasemi, Alireza Mehridehnavi, Alfonso Perez-Garrido, and Horacio Perez-Sanchez. Neural network and deep-learning algorithms used in qsar studies: merits and drawbacks. *Drug Discov. Today*, 23(10):1784–1790, 2018.

[77] Leonardo L.G. Ferreira and Adriano D. Andricopulo. Admet modeling approaches in drug discovery. *Drug Discovery Today*, 24(5):1157 – 1165, 2019. ISSN 1359-6446. doi: https://doi.org/10.1016/j.drudis.2019.03.015. URL http://www.sciencedirect.com/science/article/pii/S1359644618303301.

[78] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63, 2015.

[79] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.

[80] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, May 2010.

[81] A Gobbi and D Poppinger. Genetic optimization of combinatorial libraries. *Biotechnol. Bioeng.*, 61(1):47–54, 1998.

[82] Peter Willett. Similarity-based virtual screening using 2D fingerprints. *Drug discovery today*, 11(23-24):1046–1053, December 2006. ISSN 1359-6446. doi: 10.1016/j.drudis.2006.10.005. URL http://dx.doi.org/10.1016/j.drudis.2006.10.005.

[83] Milan Randić, Gregory M Brissey, Robert B Spencer, and Charles L Wilkins. Search for all self-avoiding paths for molecular graphs. *Computers & Chemistry*, 3(1):5–13, 1979.

[84] Kinkar Ch Das, Kexiang Xu, and Junki Nam. Zagreb indices of graphs. *Frontiers of Mathematics in China*, 10(3):567–582, 2015.

[85] Paola Gramatica. Whim descriptors of shape. *QSAR & Combinatorial Science*, 25(4):327–332, 2006.

[86] Andrea Mauri, Viviana Consonni, Manuela Pavan, and Roberto Todeschini. Dragon software: An easy approach to molecular descriptor calculations. *Match*, 56(2):237–248, 2006. ISSN 0340-6253. URL http://www.academia.edu/download/40014150/DRAGON_software_An_easy_approach_to_mole20151114-25260-1wp13db.pdf.

[87] Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. *Journal of cheminformatics*, 10(1):4, 2018.

[88] Amit S Kalgutkar, Iain Gardner, R Scott Obach, Christopher L Shaffer, Ernesto Callegari, Kirk R Henne, Abdul E Mutlib, Deepak K Dalvie, Jae S Lee, Yasuhiro Nakai, John P O'Donnell, Jason Boer, and Shawn P Harriman. A comprehensive listing of bioactivation pathways of organic functional groups. *Curr. Drug Metab.*, 6(3):161–225, June 2005.

[89] Amit S Kalgutkar and John R Soglia. Minimising the potential for metabolic activation in drug discovery. *Expert Opin. Drug Metab. Toxicol.*, 1(1):91–142, June 2005.

[90] Stephen J Capuzzi, Eugene N Muratov, and Alexander Tropsha. Phantom pains: Problems with the utility of alerts for p an-a ssay in terference compound s. *Journal of chemical information and modeling*, 57(3):417–427, 2017.

[91] Mario R Senger, Carlos AM Fraga, Rafael F Dantas, and Floriano P Silva Jr. Filtering promiscuous compounds in early drug discovery: is it a good idea? *Drug Discovery Today*, 21(6):868–872, 2016.

[92] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[93] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[94] Diederik P Kingma and Jimmy Lei Ba. Adam: A method of stochastic optimization. *International Conference on Learning Representations*, pages 1–15, 2015.

[95] M Schuster and K K Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681, November 1997.

[96] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL `https://www.aclweb.org/anthology/D14-1179`.

[97] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *International Conference on Learning Representations*, 2016.

[98] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *International Conference on Learning Representations*, 2016.

[99] Wengong Jin. Accelerated training of junction tree vae. *GitHub*, 2019. URL `https://github.com/wengong-jin/icml18-jtnn/tree/master/fast_molvae`.

[100] Esben Bjerrum and Boris Sattarov. Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders. *Biomolecules*, 8(4):131, 2018.

[101] Esben Jannik Bjerrum. Smiles enumeration as data augmentation for neural network modeling of molecules. *arXiv preprint arXiv:1703.07076*, 2017.

[102] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[103] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means+. *Proceedings of the VLDB Endowment*, 5(7), 2012.

A   Representations and descriptors of molecular structures

In this section, we discuss different approaches to representing a molecule in a machine learning-friendly way (Figure 4). This section does not intend to be a comprehensive overview of the field, but rather a brief introduction to main representations used in recent works. We consider three representations: string representations, graph representations and 3D representations. We also discuss different approaches to extracting features from molecules at the end of the section.
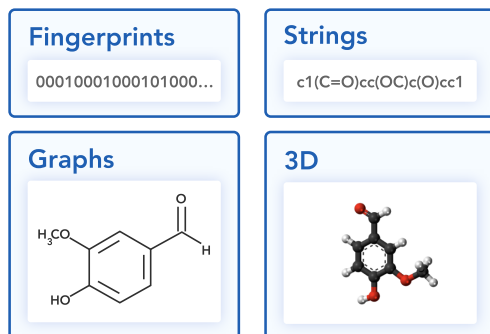

Figure 4: Different views on a vanillin molecule.

String representations. Representing a molecular structure as a string have been quickly adopted [54, 46, 55–59, 53, 60] for generative models due to the abundance of sequence modeling tools such

as recurrent neural networks, attention mechanisms, and dilated convolutions. Simplified molecular input line system (SMILES) [39] is the most widely used string representation for generative machine learning models. SMILES algorithm traverses a spanning tree of a molecular graph in depth-first order and stores atom and edge tokens. SMILES also uses special tokens for branching and edges not covered with a spanning tree. Note that since a molecule can have multiple spanning trees, different SMILES strings can represent a single molecule. While there is a canonicalization procedure to uniquely construct a SMILES string from a molecule [61], ambiguity of SMILES can also serve as augmentation and hence improve generative models [62].

DeepSMILES [63] was introduced as an extension of SMILES that seeks to reduce invalid sequences by altering syntax for branches and ring closures. Some methods try to incorporate SMILES syntax into a network architecture to increase the fraction of valid molecules [64, 65]. SELFIES [66] defines a new syntax based on a Chomsky type-2 grammar augmented with self-referencing functions. International Chemical Identifier (InChI) [67] is a more verbose string representation which explicitly specifies a chemical formula, atoms' charges, hydrogens, and isotopes. However, Gómez-Bombarelli et al. [30] reported that InChI-based models perform substantially worse than SMILES-based models in generative modeling—presumably due to a more complex syntax.

Molecular graphs. Graph representations have long been used in chemoinformatics for storing and processing molecular data. In a molecular graph each node corresponds to an atom and each edge corresponds to a bond. Such graph can specify hydrogens either explicitly or implicitly. In the latter case, the number of hydrogens can be deduced from atoms' valencies.

Classical machine learning methods mostly utilize molecular descriptors extracted from such graphs. Deep learning models, however, can learn from graphs directly with models such as Graph Convolutional Networks [68], Weave Networks [26], and Message Passing Networks [69]. Molecular graph can also be represented as adjacency matrix and node feature matrix; this approach has been successfully employed in the MolGAN model [32] for the QM9 dataset [70]. Other approaches such as Junction Tree VAE [48] process molecules in terms of their subgraphs.

3D geometry. In drug discovery, it is often sufficient to specify a graph structure of a molecule. However, since molecules exist and interact in 3D world, it could be beneficial to represent them in 3D. A common way to represent a molecule in 3D is to use voxel models or point clouds [71] (unordered sets of atom coordinates). Such methods are useful for modeling protein binding since they can encode relative positions of small molecules in a binding site. However, this representation introduces multiple challenges: optimization method significantly influences the performance of down-stream tasks [72]; minimum-energy conformation is usually different from in-pocket conformation [73]. Finally, many features including activity can be estimated from graph structure directly, since this information is already implicitly ingrained in the structure [74].

Molecular descriptors. We conclude this section by discussing different ways to extract features from molecular structures. These features are useful for multiple downstream tasks such as activity prediction [75, 76], ADME-Tox property prediction [77], and similarity search [78].

The first type of molecular descriptors are molecular fingerprints that are typically binary or count vectors of a fixed length. Molecular Access System (MACCS) fingerprints [79] are 166-bit vectors, each bit indicates presence or absence of a certain substructure. Path-based, circular [80, 35] and pharmacophore [81] fingerprints hash particular subgraphs into a fixed-length vector. Each hash number serves as an index for the resulting binary fingerprint, with bits set to one for every found subgraph. Path-based fingerprints find all paths of certain length in a graph; circular fingerprints find all substructures of given radius; pharmacophore fingerprints find all combinations of at most n pharmacophore points. Fingerprints can be used for virtual screening via fast similarity lookup [82]. Common similarity functions between binary vectors A and B are:

$$T(A, B) = \frac{\sum_i A_i \cdot B_i}{\sum_i (A_i + B_i - A_i \cdot B_i)} \qquad \text{Tanimoto similarity} \qquad (6)$$

$$C(A, B) = \frac{\sum_i A_i \cdot B_i}{\sqrt{\sum_i A_i^2} \sqrt{\sum_i B_i^2}} \qquad \text{Cosine similarity} \qquad (7)$$

$$D(A, B) = \frac{2 \sum_i A_i \cdot B_i}{\sum_i A_i + \sum_i B_i} \qquad \text{Dice similarity} \qquad (8)$$

The next group of molecular descriptors summarize molecular structure to produce a single number: molecular weight, number of atoms, water-octanol partition coefficient (logP), etc. Graph descriptors also report topological graph features such as graph diameter, a number of self-avoiding paths [83], or Zagreb indices [84]. Molecular descriptors can use statistics from a 3D structure such as gyration radius or WHIM descriptors [85]. Many of these molecular descriptors can be found in RDKit [36], Dragon descriptors [86] and Mordred descriptors [87] packages.

B Medicinal chemistry filters and PAINS filters

Medicinal chemistry filters are used to discard compounds containing so-called "structural alerts". Molecules containing such moieties either bear unstable or reactive groups or undergo biotransformations resulting in the formation of toxic metabolites or intermediates.

We filtered the dataset with medicinal chemistry filters (MCFs) that we explain in this section. We used MCFs for rational pre-selection of compounds more appropriate for modern drug design and development. These include some electrophilic alkylating groups, such as Michael acceptors (MCF1-3), alkyl halides (MCF4), epoxide (MCF5), isocyanate (MCF6), aldehyde (MCF7), imine (Schiff base, MCF8), aziridine (MCF9) which are very liable for nucleophilic attack. In many cases, it leads to unselective protein and/or DNA damage. Metabolism of hydrazine (MCF10) furnishes diazene intermediates (MCF11), which are also alkylating warheads. Monosubstituted furans (MCF12) and thiophenes (MCF13) are transformed into reactive intermediates via epoxidation. Their active metabolites irreversibly bind nucleophilic groups and modify proteins. Electrophilic aromatics (e.g. halopyridine, MCF14), oxidized anilines (MCF15) and disulfides (MCF16) are also highly reactive. In vivo, alkylators are trapped and inactivated by the thiol group of glutathione, which is a key natural antioxidant. Azides (MCF17) are highly toxic; compounds containing this functional group particularly cause genotoxicity. Aminals (MCF18) and acetals (MCF19) are frequently unstable and inappropriate in generated structures. In addition, virtual structures containing a large number of halogens (MCF20-22) should be excluded due to increased molecular weight and lipophilicity (insufficient solubility for oral administration), metabolic stability, and toxicity. The detailed mechanism of toxicity for structure alerts mentioned above has been comprehensively described in [88, 89].

PAINS (pan-assay interfering compounds) filters are the set of substructure filters proposed to use for reducing the number of false positives, assay artifacts and unspecific bioactive molecules in the screening libraries. It was stated that the presence of certain fragments in a structure could lead to undesirable properties (reactivity, chelation, the formation of colloidal aggregates, dyes) affecting assay results. It should be noted that the analysis of available data from the PubChem database clearly demonstrated the limitations of PAINS filters [90, 91]. Indeed, PAINS were observed among the molecules inactive in at least 100 bioassays (the dark chemical matter). Interestingly, structural analysis of well-known drugs revealed PAINS among them. For instance, quinone-based compounds were classified as PAINS, however there are quinone-based drugs approved by the FDA in the market. Despite mentioned above, this approach can be considered as a viable tool for narrowing down the large virtual chemical spaces produced by generative models to drug-like chemical matter.

C Diverse set of molecules from MOSES

In Figure 5, we show a diverse set of molecules of MOSES dataset. We obtained these molecules by iteratively adding structures with the lowest cosine similarity to the nearest compound in the currently selected set.

D Hyperparameters and training details

Character-level recurrent neural networks (CharRNN) used Long Short-Term Memory [92] cells stacked into 3 layers with hidden dimension 768 each. We used a dropout [93] layer with dropout rate 0.2. Softmax was utilized as an output layer. Training was done with a batch size of 64, using the Adam [94] optimizer for 80 epochs with a learning rate of 10^{-3} that halved after each 10 epochs. We display CharRNN model in Figure 6.

Variational autoencoder (VAE) used a bidirectional [95] Gated Recurrent Unit (GRU) [96] with a linear output layer as an encoder. The decoder was a 3-layer GRU of 512 hidden dimensions with intermediate dropout layers with dropout probability 0.2. Training was done with a batch size of 128, utilizing a gradient clipping of 50, KL-term weight linearly increased from 0 to 1 during training. We optimized the model using Adam optimizer with a learning rate of $3 \cdot 10^{-4}$. We trained the model for 100 epochs. We display an autoencoder model in Figure 7.

Adversarial Autoencoders (AAE) consisted of an encoder with a single layer bidirectional LSTM with 512 hidden dimensions, a decoder with a 2-layer LSTM with 512 hidden dimensions and a shared embedding of size 128. The discriminator network was a 2-layer fully connected neural network with 640 and 256 nodes respectively with exponential linear unit (ELU) [97] activation function [98]. We trained a model with a batch size of 512, with the Adam optimizer using a learning rate of 10^{-3} for 120 epochs. We halved the learning rate after each 20 epochs.

Junction Tree VAE (JT-VAE) We report the experimental results from the official JT-VAE repository [99].

Latent Vector Based Generative Adversarial Network (LatentGAN) pretrained an aused a heteroencoder [100] containing a two-layer bidirectional encoder with 512 LSTM units per layer. Authors added a Gaussian noise with a zero mean standard deviation of 0.1 to the latent codes, resembling VAE with a fixed variance of proposal distributions. The LSTM decoder had 4 layers. The neural network was trained on pairs of randomly chosen non-canonical SMILES strings [101]. The autoencoder network was trained for 100 epochs with a batch size of 128 sequences, using Adam optimizer with a learning rate 10^{-3} for first 50 epochs and with an exponential learning rate decay reaching a value of 10^{-6} in the final epoch. LatentGAN uses Wasserstein GAN with gradient penalty (WGAN-GP) [102] with a fully connected discriminator with 3 layers of which the first two used the leaky ReLU activation function, and the last layer no activation function. The generator consisted of five fully connected layers with batch normalization and leaky ReLU activation. The GAN was trained for 2,000 epochs using a learning rate of $2 \cdot 10^{-4}$ with Adam parameters $\beta_1 = 0.5, \beta_2 = 0.9$. We display LatentGAN model in Figure 8.

Combinatorial generator randomly joins BRICS fragments. We first cut all molecules from the training set into fragments and compute the frequency of each fragment. We also compute a distribution of the number of fragments in the training set. To produce a molecule, we first randomly sample a total number of fragments that we will use in the molecule. We then iteratively sample fragments according to their frequencies. We omit fragments that will lead to invalid final molecules. For example, if there are currently two free attachment points in the molecule and two fragments left to attach, we cannot attach fragments with more than one attachment points. We also experimented with randomly sampling fragments until there are no more connection points. However, such method performed worse.

N-gram model used 11-gram count statistics with pseudo-count of 0.01. During generation, when there were no statistics available for the current (n-1)-gram, we reduced the context length until some statistics were available. In extreme cases, the context reduced to a single token which is equivalent to a bigram model.

Hidden Markov Model uses Baum-Welch algorithm for training the model. We used HMM with 200 states and trained the model for 100 epochs on a subset of MOSES train set with first 100,000 molecules. Note that HMM uses batch training which leads to high computational costs. To speedup learning, we used K-means|| algorithm [103] to initialize parameters of the model.
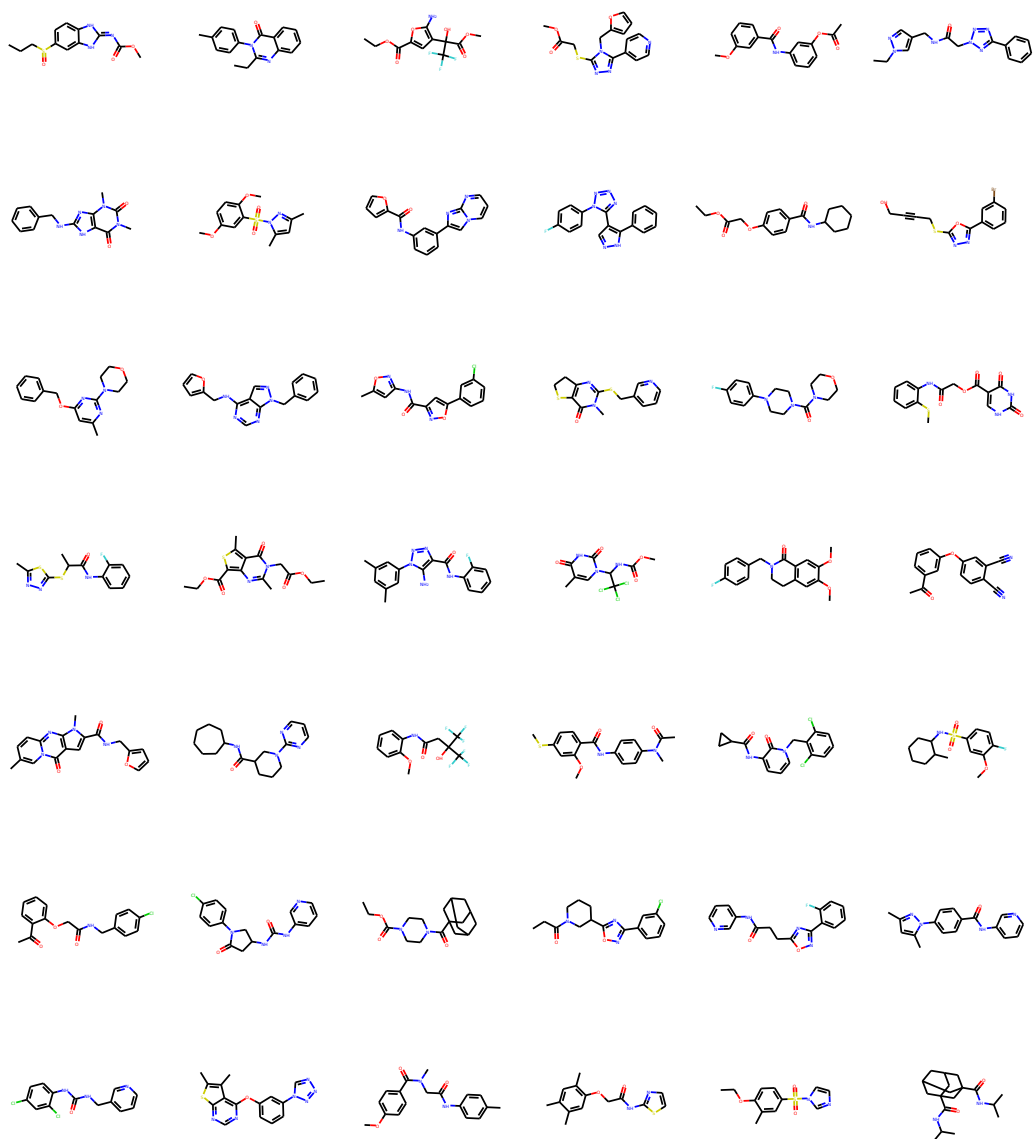
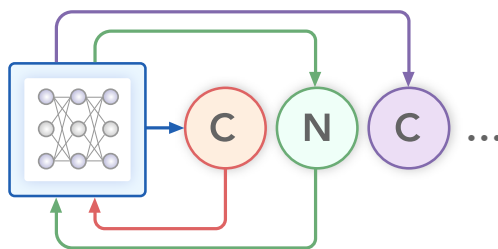Figure 5: A diverse subset of molecules from MOSES dataset.

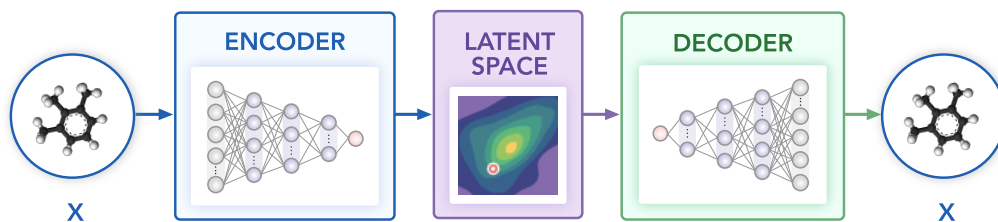Figure 6: CharRNN model. A model is trained by maximizing the likelihood of known molecules.

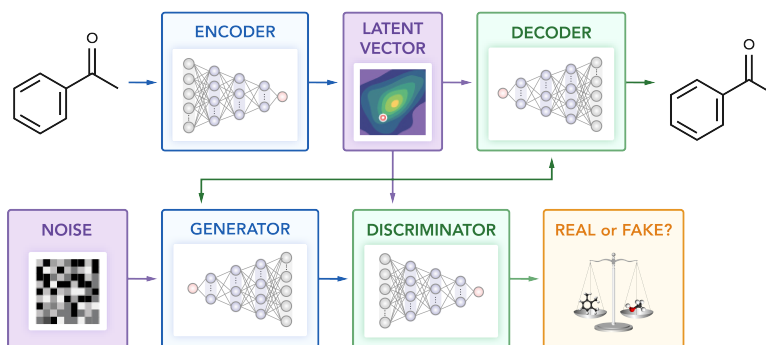Figure 7: Autoencoder-based models. VAE/AAE forms a specific distribution in the latent space.

Figure 8: LatentGAN model. A model combines an autoencoder and generative adversarial networks.