VNFlow: Integration of Variational Autoencoders and Normalizing Flows for Novel Molecular Design

Jiří Hostaš

jiri.hostas@nrc-cnrc.gc.ca

National Research Council Canada

Mohammad Sajjad Ghaemi

National Research Council Canada

Hang Hu

National Research Council Canada

Junan Lin

National Research Council Canada

Anguang Hu

Defence Research and Development Canada

Hsu Kiang Ooi

National Research Council Canada

Research Article

Keywords: normalizing flows, molecular design, generative AI, chemistry

Posted Date: July 9th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-6649856/v1

Additional Declarations: No competing interests reported.

VNFlow: Integration of Variational Autoencoders and Normalizing Flows for Novel Molecular Design

Jiří Hostaš[1,*], Mohammad S. Ghaemi[1], Hang Hu[1], Junan Lin[1], Anguang Hu[2], Hsu K. Ooi[1]

[1]Digital Technologies Research Centre, National Research Council Canada, Toronto, ON, Canada.
[2]Suffield Research Centre, Defence Research and Development Canada, Medicine Hat, AB, Canada.

*Corresponding author(s). E-mail(s): jiri.hostas@nrc-cnrc.gc.ca, jiri.hostas@gmail.com;

Abstract

Generative Artificial Intelligence is transforming the molecular discovery by enabling exploration of the vast, largely unexplored chemical space. However, current methods, including normalizing flows, struggle to balance the optimization of complex objectives and sampling speed, particularly when generating specific compound classes and more intricate scaffolds, such as aromatic rings. This work developed a generative model that efficiently samples novel molecules while optimizing their drug-likeness, ease of synthesis or chemical reactivity. To achieve this, we employed normalizing flows combined with variational autoencoders to generate samples which were evaluated for the Quantitative Estimate of Drug-likeness, the Synthetic Accessibility scores and, in case of organofluorine-phosphates, electronic density on the central phosphorus atom, approximated by Hirschfeld charges calculated with density functional theory. Our framework efficiently generated a diverse range of organofluorine-phosphates, demonstrating that combining normalizing flows directly with SELFIES or group-SELFIES can address key limitations in inverse molecular design, particularly when variational autoencoders cannot be applied due to a lack of available training data. Normalizing flows capture the chemical structures in a holistic way which paves the way towards targeted therapies that enable the optimization of complex molecular objectives.

Scientific contribution: To the best of our knowledge, this is the first report of integrating variational autoencoders with normalizing flows in a comprehensive

molecular design workflow. It is also the first application of conditional normalizing flows to molecular design. Our method yielded novel molecules with performance metrics surpassing those in the 2.4 million-compound ChEMBL database, highlighting its potential for identifying promising drug candidates.

Keywords: normalizing flows, molecular design, generative AI, chemistry

1 Introduction

The rapid growth in the amount of data, computing power, and advancements in artificial intelligence (AI) algorithms has led to the transformation of many scientific fields, including drug discovery.[1] This momentum was formally acknowledged in October 2024, when the Nobel Prizes in both Physics and Chemistry were awarded for the breakthroughs in training artificial neural networks and for solving the protein structure prediction challenge, respectively. However, studies estimate that the cost of developing a new drug remains in the range of a few billion dollars and can readily take more than 15 years to complete.[2, 3] The AI has the potential to accelerate rational design of new potential drug candidates, streamline clinical trial processes or democratize access to protein structure information which could significantly reduce the time and labor costs associated with drug development.[4, 5] In these and other applications, the early drug discovery methods continue to receive significant attention and efforts to address the challenges of exploring the vast, largely unknown, and yet to be thoroughly probed chemical space.[6]

The initial stages of drug discovery, namely lead discovery and lead optimization were among the first implementations of AI models in chemistry. The generation of new drug leads without a deep prior knowledge is a focus of the so-called *de novo* molecular generation methods, which aim to improve molecular metrics such as drug-likeness, target specificity, and ADMET (Absorption, Distribution, Metabolism, Excretion, and Toxicity).[7] Numerous AI models and databases have been developed to speed up the search through the chemical space which is mostly unexplored, large, discrete, and unstructured.[8] For a long time, the state-of-the-art approach has been to exhaustively scan and filter through fixed libraries or use discrete local search methods such as genetic algorithms.[9] This is highly impractical because it is not feasible in the near future to store or even search through the chemical space containing 10^{60} of all small molecules. Encouragingly, there have been methods which address some of these challenges by the conversion to continuous, data-driven, and machine-readable molecule representations and by generating new drug candidate molecules on demand. A good overview of the available methods can be found elsewhere, so let us point readers to the most popular machine learning solutions and highlight some of their differences.[10]

First of the rapidly developing methods are generative adversarial networks (GANs). GANs operate through a competition between two models: one deep neural network generates new data with the same characteristics as the training set, while the second model, the discriminator, is trained to distinguish whether the generated data is part of the training set or not. A successful application for small molecules,

MolGAN, was published by De Cao *et al.* where they focused on a direct generation of graph-structured molecular representations.[11] However, GANs are known to struggle with training instability and with generating larger and more complex molecules.

Since molecules can be viewed as sequences of strings and parentheses (representing atom types, bond types, and the starting and stopping points of the rings), the methods which proved their efficiency for the text generation, such as recurrent neural networks (RNNs), also found applications in the chemistry domain.[12, 13] The RNN models are known for an impressively high molecular validity rates reaching 98% for a long-short term memory (LSTM) models developed by Bjerrum *et al.* or over 90% for a gated recurrent unit (GRU) models, both examples of popular RNN architectures.[14–16]

Gómez-Bombarelli *et al.* used one-hot encoded Simplified Molecular Input Line Entry System (SMILES) representation as a starting point for their variational autoencoder (VAE).[17] The encoder part of the model was built using convolutional neural network layers while the decoder was the popular GRU model. A wide range of valid SMILES generation rates has been reported ranging between $\sim$ 75% and 0.7% depending on which part of the chemical space was sampled.[18] Although there exist other approaches with much higher decoding rates, such as the pure RNNs models with over 90% decoding rates, a simple validity check can screen out invalid samples during the generation process and the latent representation can be leveraged for a straightforward visualization or as input for other models. Thus, the success rate of generating valid SMILES sequences is only one of the metrics in the model comparison and moreover, the goal of the molecular design is often an improvement across several competing and chemically relevant molecular properties.

Another family of deep generative models are normalizing flows and they are gaining popularity due to their applicability in a wide range of generative tasks.[19]. These models provide competitive advantages over their GAN and VAE counterparts because they offer deterministic objective functions, efficient generative sampling and exact likelihood calculations. To the contrary, VAE models give only a lower-bound on log-likelihood while GANs do not provide any estimate of log-likelihood for its samples, which complicates their training and evaluation. Therefore, masked autoregressive flows, real non volume preserving flows and neural spline flows are some of the architectures designed to enable fast training or efficient sampling depending on the application and found their applications in image and graph generation applications.[20, 21]

In the molecular design, one of the pioneering applications of normalizing flows is a flow-based graph generative model, MoFlow.[22] To generate a full molecular graphs in a one-shot manner, MoFlow starts by generating bonds (edges) with a Glow-based model, follows by atoms (nodes) attached using a graph conditional flow, and finalizes with posthoc validity correction checking for the valency constraints.[23] There are also hierarchical models, such as MolGrow, which generates molecular graphs iteratively via generating molecular graphs from a single-node graph by recursively splitting every node into two.[24] Chemical rules are also incorporated in the model called, GraphAF, which at the time outperformed other VAE based methods on molecular optimization tasks.[25]

Another application of normalizing flows for rapid molecular generation of small molecules was published by Frey *et al.* in 2022.[26] Its simple architecture delivered

an impressive efficiency in generating thousands of molecules per second, however, it was not optimized to address the molecular optimization of larger molecules or to systematically improve the studied molecular metrics. Instead, the authors lean on post-hoc filters and multi-objective optimization through high-throughput virtual screening.

Despite the improvements of the state-of-the-art normalizing flows architectures, they tend to be restrictive which leads to excessively large models and a high number of model parametres which hinders their training. In image generation, it has been shown that this can be partially addressed by VAE encoding through feature reduction.[20, 27, 28] In the molecular design domain, normalizing flows might improve valid SMILES generation rates of pseudo-randomly chosen points in the latent space of other models or help the VAE decoder to sample from the part of the space with novel molecules that have desired properties.

This work presents several novel applications of normalizing flows for molecular discovery. Firstly, we combine the normalizing flows model with VAE chemical encoding to improve sampling of new molecules and to reduce the number of molecular features and parameters needed to train our normalizing flows. The feature reduction increases sampling and training efficiency, which opens the door for testing a larger variety of normalizing flows. Secondly, we test the conditional normalizing flows for molecular design for the first time. Finally, we showcase an iterative design workflow in a low data regime where we generate organofluorine-phosphate molecules using normalizing flows. For the generated organofluorine-phosphate molecules, we optimize the relative atomic coordinates of their atoms and calculate their atomic charges with density functional theory (DFT).

2 Methods

2.1 Molecular Representations, Datasets and Chemical Metrics

The choice of input representation plays a key a role in the generative design and it is an ongoing area of research.[29] Here, we start with the most popular way to represent molecules as 1D SMILES (Simplified Molecular Input Line Entry System) strings which are sequences of ASCII characters that use a depth-first graph traversal.[13] This sequence of characters is converted into a one-hot-encoded vector, a representation eliminating ordinality (inherent order present in categorical representation) and enabling a straightforward application of various deep neural network architectures (RNN, VAE, GAN *etc.*) popular in natural language processing. Despite the popularity of SMILES, the methods built using this representation are prone to generating invalid SMILES strings that do not adhere to the rather strict SMILES notation.

There have been attempts to address these issues caused by the complex SMILES grammar. Krenn *et al.* introduced a new string representation called SELFIES (Self-Referencing Embedded Strings) which are completely robust while being less efficient for certain tasks.[30] Typically, the length of a SELFIES string is noticeably larger compared to SMILES which can inconvenience the applications of normalizing flow-based models. This has been partially addressed by group-SELFIES representation where a single SELFIES string can define a group of atoms chosen a priori.[31] We

analyzed ZINC250k database using RDKit automatic fragmentation algorithms and identified 10 fragments with aromatic ring which were used to enrich the molecular generation of our tested models.[32]

Most of the publicly available datasets store molecules as SMILES (ChEMBL, ZINC250K, QM9 etc.).[8, 32–34] We trained several generative models on ChEMBL dataset of curated bioactive molecules.[8] We used a moderate-size dataset of 50,000 random ChEMBL22 structures (see analysis of QED and SA score in Table 1) for several initial tests and the latest (ChEMBL35) version of the entire dataset with $\sim$ 2,400,000 structures for further comparisons.

To assess the usefulness of the generated molecules, we evaluated the following metrics, which generally give an indication of which generated molecules might be interesting to study further: the quantitative estimate of drug-likeness (QED), Synthetic Accessibility (SA) score, Morgan fingerprint, the number of heavy atoms or aromatic rings; all of which were calculated using the RDKit package.[35]

Although the molecules reside in 3D space and molecule's conformations may be important for predictions of some properties such as binding strength to a specific receptor[36], most generative modelling methods are being developed primarily for the use with simpler graph and 1d representations, especially in the case of modelling of large and flexible molecules. Typically, the bond length and 3D information are discarded, as is the case with our models. However, information about the energetically stable conformation can be recovered by energy minimization with quantum mechanical methods or empirical force fields. We have done this using density functional theory (DFT) in the case of organofluorine-phosphates (see Sections 2.2 and 3.2).

2.2 Density Functional theory

Density Functional theory (DFT) has been applied in many applications from the description of electronic structure in materials science, protein-protein interactions, to drug–receptor interactions in drug discovery.[37–39] Here, DFT was used to test the stability and to probe the electronic-structure properties of the generated organophosphate molecules.

The geometry optimization, analytical harmonic vibrational frequencies and atomic Hirschfeld charges were calculated using Orca 5.0.4 software.[40] There are numerous ways and formulations which one can choose when applying DFT and we used B3LYP-D4 method with a medium-size (def-TZVP) basis set as a compromise between speed and accuracy.[41–43] The starting geometries were generated using RDKit[35] package.

2.3 Variational Autoencoders

Variational autoencoders (VAEs) are one of the most succesful and popular generative models in which the encoder - decoder pair is used to establish a discrete data mapping of $\mathbf{x}$ (e.g. one-hot-encoded SMILES, SELFIES or group-SELFIES) to and from a latent continuous representation ($\mathbf{z}$, see the top panel of the Figure 1). The encoder learns the underlying lower-dimensional distribution of features creating the data while the decoder generates new samples from this learned distribution. Alternatively, VAE

can be understood as a collection of two coupled but independently parameterized models trained using stochastic gradient descent and maximizing the Evidence Lower Bound (ELBO).[44] The ELBO consists of two main components: the reconstruction term ensuring that the latent space follows the original distribution and captures enough information to accurately reconstruct the input; and the regularization term (Kullback–Leibler divergence), which nudges the latent distribution to stay close to a predefined prior, typically a Gaussian distribution. A good introduction and discussion about the underlying model assumptions (*e.g.* about the latent spaces being close to normal) can be found elsewhere.[45]

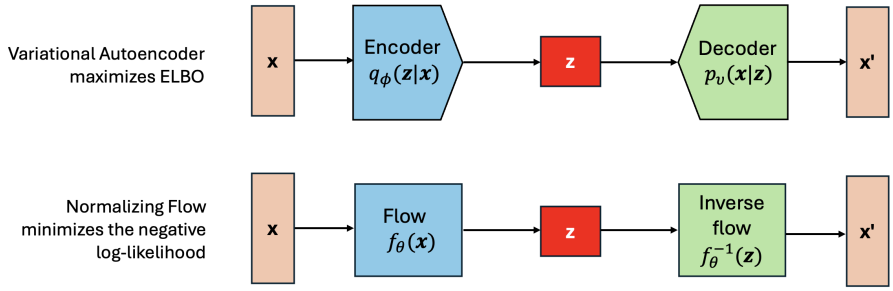

Fig. 1 Comparison of the training of a variational autoencoder and normalizing flow models.

Our VAE models were derived from the architecture introduced by Gómez-Bombarelli *et al.* which was in turn based on the previous applications for English texts.[17, 46] Their architecture consists of three 1D convolution layers (having filter sizes of 9, 9, and 10 with the respective kernel sizes of 9, 9, and 11) and a fully connected linear layer. The latent space has 292 dimensions and the decoder uses three stacked layers of gated recurrent unit (GRU) networks each with a hidden dimension of 501.[47] We used python package *pytorch* to implement it.[48] An example of our model convergence with respect to training and validation errors is depicted in the Figure A5, while the performed hyperparameter search for the encoder architecture trained on SELFIES is in the Table A1. Several models were trained on 50,000 randomly sampled molecules (using 80:20 training/validation split) from the ChEMBL22 data set and 2.4 million molecules from the ChEMBL35 data set were converted into one-hot-encoded SMILES and SELFIES representations. These VAE models were used as our baseline models for generating novel molecules via decoding random vectors and also as a dimension reduction tools of the feature vectors for the normalizing flows.

2.4 Normalizing Flows

While most machine learning and AI tasks can be understood as learning the distributions of training data, only a minor subset of methods can, in principle, construct these distributions explicitly. Normalizing flows take a simple multivariate normal distribution (with the diagonal Gaussian density being the most popular choice) and apply a sequence of differentiable and invertible functions, also referred to as bijections or

mappings:

$$f_\theta^{-1} = f_{\theta_1}^{-1} \circ f_{\theta_2}^{-1} \circ \cdots \circ f_{\theta_K}^{-1} \tag{1}$$

that are used to map it to the original, often complex, data distribution. These bijections have to be easy to compute and invert. Also, the determinant of their Jacobian should be easy to calculate, so one can compute the exact log-likelihood for each data point x by repeatedly applying the rule for change of variables as follows:

$$\log p_\theta(x) = \log p_\iota(z) + \sum_{i=1}^{K} \log \left| \det \frac{\partial f_{\theta_i}^{-1}(x_i)}{\partial x_i} \right|, \tag{2}$$

where $p_\theta(x)$ is the data distribution, $p_\iota(z)$ is the known diagonal Gaussian distribution and $\left| \det \frac{\partial f_{\theta_i}^{-1}(x_i)}{\partial x_i} \right|$ is the determinant of the Jacobian matrix of the function mappings $f_{\theta_i}^{-1}(x_i)$. The mappings are typically implemented in a form of invertible deep neural networks and they can be trained by minimizing Kullback–Leibler divergence via minimizing the negative log-likelihood. One of the main limitations of normalizing flows is that they can be hard to train and the invertability criteria can limit the expressivity of the used transformations.

Several popular types of non-linear flows are derived from the coupling flow framework which repeatably splits the input into parts, where the first part stays the same while the second part undergoes a scale-and-shift (affine) transformation. Affine flows are defined by shifting the input by a bias value μ and scaled by a parameter σ, where μ and σ are functions of the first part of the unchanged input and implemented as deep neural networks. The sums and multiplications are performed element-wise. The first type of flow tested here, based on the coupling of several of these affine transformations, is called Real Non Volume Preserving flow (Real NVP) and it is implemented in the python package *glasflow*.[49] Batch normalization was found to help with training models with a large number of coupling layers.[49] Since a subset of the input dimensions stay unmodified in each affine coupling layer, reverse or random permutations are applied (see more technical details in the *glasflow* documentation).

Affine flows have also been used within the autoregressive framework, which models data sequentially such that each dimension depends solely on the previously calculated dimensions. This has implications for sampling efficiency, as the sequential nature of the workflow prevents parallelization. In 2017, Masked Affine Autoregressive Flow (MAF) were introduced by Papamakarios *et al.* who introduced checkerboard masking which was tested it in its original *nflows* implementation.[20, 50]

In 2019, the monotonic rational quadratic splines were introduced to enhance the flexibility of both coupling and autoregressive transforms while keeping the invertibility of the transforms.[51] We have used the implementation available in the package *normflows*, where the autoregressive rational quadratic neural spline transform is combined with lower-upper (LU) linear permute transformation.[52] For a comprehensive and more formal overview of normalizing flows, we refer readers to several in-depth review articles on the subject.[53, 54]

To generate novel molecules, we begin by sampling a random vector from a multivariate diagonal Gaussian distribution ($\mathbf{z}$ at the bottom of Figure 2). This vector is then passed through the inverse flow, which consists of a previously trained sequence

of bijective and differentiable functions. The output is decoded using the VAE decoder and the SMILES, SELFIES or group-SELFIES string is obtained by inverse transformation of the one-hot-encoded vector. Finally, the molecular string is validated and the desired properties of the novel molecules calculated.

Papamakarios *et al.* showed that incorporating additional information provided by labels can benefit the generative models.[20] We used QED scores or atomic charges as labels, allowing the normalizing flows to learn the conditional probability distributions, effectively doubling the number of models tested.

Our two phase training procedure is shown in the Figure 2. All the components of the underlying VAE are fixed in the normalizing flow (second) training phase. It is possible to optimize the entire model, both VAE and normalizing flow, jointly in one phase, however, it was previously reported that it results in a poorer performance.[27]

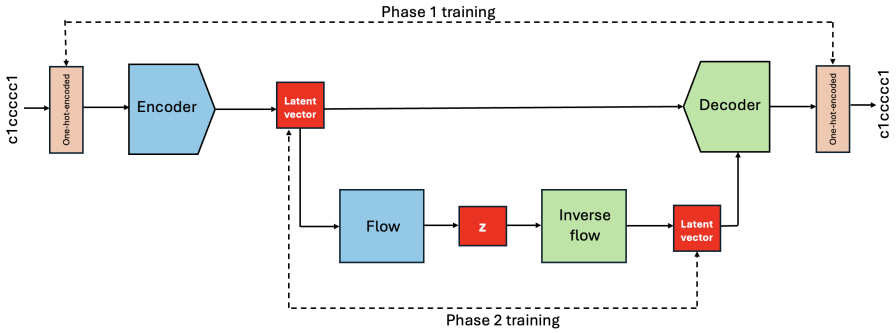

Fig. 2 The integration of variational autoencoder and normalizing flow models. First, the variational autoencoder model was trained on the entire database (50,000 or 2.4 million examples) by maximizing the ELBO loss function. Subsequently, the normalizing flow model was trained on a subset of this database (the latent vectors of the SMILES strings with the top 1% or, in case of the larger training dataset, the top 0.1% QED score) by minimizing the negative log-likelihood.

When there is a lack of training data for molecules with a specific desired functional group (*e.g.* organofluorine-phosphates, see Section 3.2), it is not possible to train a specialized VAE autoencoder encoding. Instead, we use SELFIES or group-SELFIES representations directly which proved to be more robust than using SMILES for a direct generation.[30] We apply a uniform noise in the range between 0 and 1 to dequantize the one-hot-encoded molecular vectors since the normalizing flows do not perform well with unprocessed binary data or discrete distributions.[26]

Unless stated otherwise, we keep the number of coupling layers (4) and hidden features (32) fixed. The models are trained with the Adam optimizer, a batch size of 100, a step size between 10^{-3} and 10^{-5}. Each model is trained using early stopping until no improvement is observed for 1,000 epochs. We train normalizing flows on distributions of all available molecular samples or only on the target distribution of molecules (such as molecules with the top 0.1 or 1% QED score, see Section 3.1) when stated. The train/test split was kept 80/20. In the case of the tests without VAE encoding and, therefore, the longer input lengths for normalizing flows, we used higher number of coupling layers (8) and hidden features (128).

2.5 Computational Resources and Software

We performed all the simulations on a single machine with an Intel Xeon Silver 4214 2.20GHz CPU and a single NVIDIA GP107GL Quadro P1000 GPU with 4GB GDDR5 on-board memory. We used following python packages: *nflows*[50], *glasflow*[49], and *normflows*[52], *sci-kit learn*[55], *pytorch*[48], *RDKit*[35], *pandas*[56], and *numpy*[57]. The code will be made available upon publication.

Table 1 Molecular metrics for sets of molecules sampled by the VAE decoder, compared with samples from the ChEMBL database. The best QED and SA scores among the normalizing flow models are highlighted in bold. A total of 100,000 samples were generated by each method.

Sampling method	Conditioned	Novel, valid. unique [%]	QED mean / max	SA score mean	Heavy atoms mean
VAE trained on: SMILES (50,000 examples)					
Real NVP	No	1.0%	**0.73** / 0.9467	2.61	22
	Yes	1.1%	**0.73** / **0.9478**	2.64	22
Masked Affine Autoregressive	No	1.1%	0.70 / 0.9468	2.67	19
	Yes	0.9%	0.72 / 0.9472	**2.49**	22
Autoreg. Rat. Quadr. Spline	No	0.9%	0.46 / 0.9466	3.95	18
Random[1]		0.8%	0.33 / 0.8852	4.58	18
VAE trained on: SELFIES (50,000 examples)					
Real NVP	No	92.0%	0.55 / 0.9468	4.14	17
	Yes	92.5%	0.54 / 0.9470	4.18	17
Masked Affine Autoregressive	No	90.4%	0.55 / 0.9475	4.08	17
	Yes	91.5%	0.54 / 0.9473	4.13	17
Autoreg. Rat. Quadr. Spline	No	88.6%	0.46 / 0.9468	4.51	20
Random[1]		86.1%	0.37 / 0.9450	4.85	19
VAE trained on: SMILES (2,4M examples)					
Real NVP	No	0.11%	0.64 / 0.9077	5.14	24
	Yes	0.15%	0.64 / 0.9276	5.01	24
Masked Affine Autoregressive	No	0.16%	0.60 / 0.9302	5.03	26
Autoreg. Rat. Quadr. Spline	No	0.5%	0.22 / 0.9058	4.91	48
Random[1]		2.72%	0.13 / 0.6881	5.22	68
ChEMBL22 (50,000 random samples)			0.76 / 0.9481	2.44	20
ChEMBL35 (2.4M samples)			0.56 / 0.9484	2.93	28

[1]Random refers to random latent vectors decoded by the VAE.

3 Results and Discussion

3.1 Normalizing flows combined with variational autoencoder for the design of novel bio-active molecules

A major challenge with normalizing flows is their generally low expressibility, which limits their applicability to high-dimensional inputs. We addressed this issue through feature reduction in the molecular representation by using a VAE. It was trained in two phases: the VAE model was trained first on 50,000 random samples from the

ChEMBL22 database, followed by the training of normalizing flows, which was performed only on the structures with the highest 1% QED score. This approach improves the efficiency of training normalizing flows by reducing both the number of features and the number of samples on which they are trained.

Essentially, our workflow aims to enhance the sampling efficiency of the VAE generative model by generating promising latent vectors using normalizing flows. To this end, we test several popular normalizing flows. The analysis of the training dataset is presented in Figures 3, 4, A4, and Table 1. Most molecules contain between 15 and 25 heavy atoms, have QED scores ranging from 0.6 to 0.9, logP values between 0 and 5, and feature 1 to 3 aromatic rings.

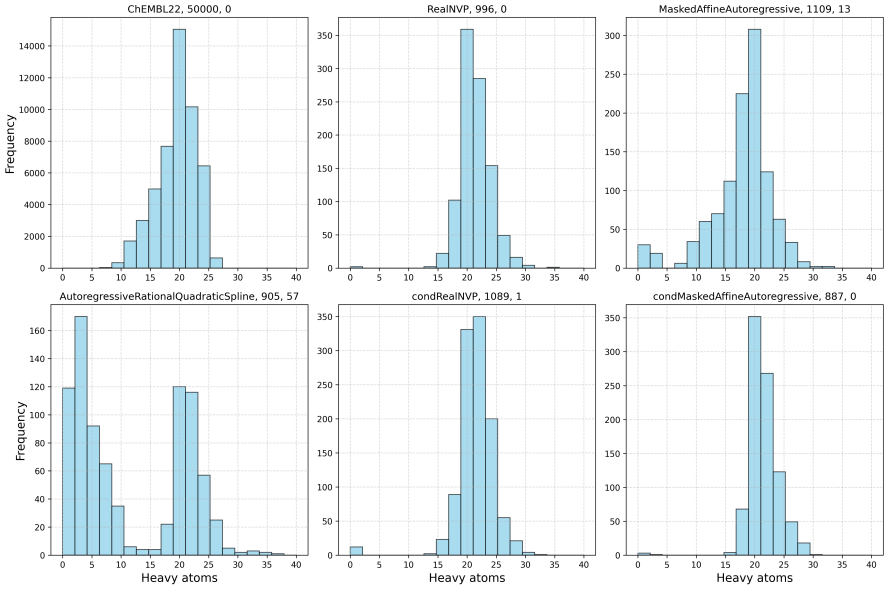

Fig. 3 Histograms of the number of heavy atoms for the starting dataset (50,000 molecules from ChEMBL22 dataset, top left panel) compared with the histograms of the molecules generated by different normalizing flows (from the left middle to the bottom right: Real NVP, masked affine autoregressive, autoregressive rational quadratic spline, conditional Real NVP, and conditional masked affine autoregressive flows). Please note that the distributions are plotted only in the range from 0 to 40. The numbers on top of the graphs stand for the total number of structures in the set and the number of structures with more than 40 heavy atoms, respectively.

Let us begin the analysis by comparing the performance of our VAE models with those of trained previously by Bombarelli *et al.* and Kusner *et al.*[17, 18] Table 1 shows that our VAE, using one-hot-encoded SMILES, achieved a sample validity rate of 0.8%, similar to the rates reported in previous studies. Higher rates were observed when interpolating between training data points or the sampled points are closer to the training data. Additionally, the more robust molecular representation (such as SELFIES, as shown in panel 2 of Table 1) resulted in a significantly higher validity rate of nearly 90%. However, in case of SELFIES, we observed a general decline in the quality

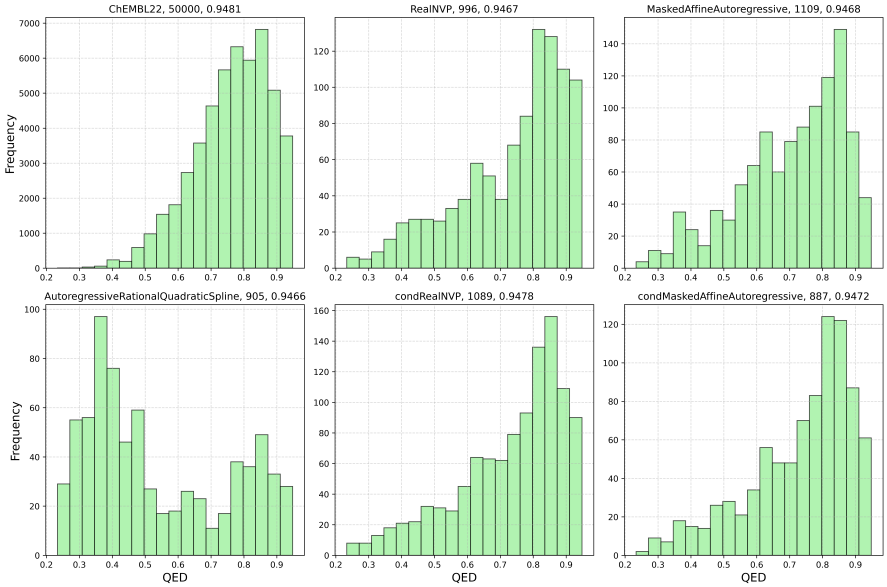

Fig. 4 Histograms of the QED score calculated for the starting dataset (50,000 structures from ChEMBL22 daset) compared with the histograms of the molecules generated by different normalizing flows (from the left middle to the bottom right: Real NVP, masked affine autoregressive, autoregressive rational quadratic spline, conditional Real NVP, and conditional masked affine autoregressive flows). The numbers on top of the graphs stand for the total number of structures plotted and the maximum QED value in the set, respectively.

of the samples documented by lowered QED scores and increased SA scores. This is because a SELFIES string gets truncated when a part of the SELFIES sequence or a single token does not result in a valid part of a molecule. On the other hand, SMILES string would simply be discarded. As a result, the average length of a SELFIES string is shorter, and the molecules are smaller, as evidenced by the lower mean number of heavy atoms (in our case ranging from 18-22 for samples generated with SMILES to 17-20 for SELFIES). Furthermore, the low validity rate is generally not a major issue, as filters and validity checks can quickly discard invalid SMILES strings. Thus, generating invalid strings can actually be beneficial for model development.[58] We trained the VAE model also on the entire ChEMBL35 dataset, containing 2.4 million SMILES. Notably, we observed an increase in the number of heavy atoms, resulting in molecules with a high count of heavy atoms, unremarkable QED scores, and exceptionally low validity rates. Further hyperparameter tuning may be required to effectively utilize a dataset of this size and diversity, however, this is beyond the scope of the current work.

Next, we focus on the results of the model trained on 50,000 SMILES examples, along with the overall performance of the normalizing flow architectures trained to generate VAE latent vectors. The corresponding results are summarized in Table 1. It is evident that the use of normalizing flows offers a substantial improvement over random sampling of the VAE latent space. While the increase in the percentage of valid and unique samples is relatively modest, the mean QED score is more than doubled, the maximum QED score achieved by any of the tested flows exceeds that obtained

via random sampling, and the mean SA score is improved by approximately 40%. Figure 3 shows that all models, except for the autoregressive rational quadratic splines, generated molecules of similar size to those in the training dataset, with the mean number of heavy atoms around 20. In fact, all normalizing flow models, except for the autoregressive rational quadratic spline flow, generated molecules with distributions of QED scores similar to those in the training dataset (see the kernel density estimation of QED score in Figure A1). Figure 4 also shows that the training dataset has a higher relative proportion of QED scores around the median value, compared to the normalizing flows, which exhibit extended tails in their QED score distributions, both in the lower and higher end. This is generally desired because we are typically looking for outliers with extremely high QED scores.

Clearly, Real NVP normalizing flow outperformed the masked affine autoregressive flow, and the same holds for their conditioned versions. The respective distributions are in the Figure 4. Real NVP achieved also the highest number of structures with the high QED scores (the bin furthest to the right), which, even at a relative scale (10% of the total sampled structures), surpassed the 7% of the ChEMBL training dataset. Additionally, among all the normalizing flows tested, Real NVP generated the molecule with the highest mean and maximum QED values, as shown in Table 1.

The autoregressive rational quadratic spline flows delivered the lowest average QED and the least promissing results (see Table 1). This can also be easily documented by the maximum of distribution of number of heavy atoms being at only 5 heavy atoms and with the largest number of samples with more than 40 heavy atoms (57 samples, see Figure 3). This means that the autoregressive rational quadratic spline flows generated both, a large number of very short sequences and also excessively long strings. Notice that this behaviour is not apparent when only the averaged metrics in the Table 1 are considered. Furthermore, autoregressive rational spline flows were nearly two orders of magnitude slower producing only 3 molecules per second when compared to other normalizing flows. As a result, we do not recommend this type of flow for applications similar to those studied in this work.

Finally, we can conclude that the Real NVP bijections showed a great promise for sampling the latent space. Conditioning the normalizing flows on the QED score resulted in only a minor improvement for the masked affine autoregressive flow, while no clear improvement was observed for the conditional Real NVP flows. Thus, it remains unclear how to optimally leverage conditional normalizing flows to achieve an overall improvement. Normalizing flows when combined with a trained VAE decoder can play a role of samplers of distributions very close to the one that they were trained on. The behavior of different normalizing flows is obviously very empirical here, as it is not clear what is the optimal size of the training dataset and how tightly the normalizing flows should try to approximate the trained distributions in various applications. More tests and development will need to be done to explore the combinations of molecular representations and flows.

Next, we will address whether these methods can be applied for exploration outside the training domain. We will begin by using the model trained on 50,000 samples from the ChEMBL22 database and evaluate how many samples are needed to improve metrics such as QED scores.

3.1.1 Search for bio-active molecules with properties outside of the distribution of the training dataset

In the previous section 3.1, we demonstrated that normalizing flows can effectively model the distributions of the training dataset. However, we did not observe any improvement in the metrics of the generated molecules relative to those in the training dataset. This outcome was expected, given that the generation process started with a training dataset of 50,000 molecules, while the number of novel, valid, and unique molecules was limited to around 1,000 per normalizing flow during these initial experiments with one-hot-encoded SMILES representation.

Therefore, we continued generating molecules with the Real NVP normalizing flow combined with our VAE model. Among the newly generated samples, 0.98% were valid and 0.75% were unique. The vast majority of these samples were novel, with only a small fraction (1,108 samples out of the first 250,000) overlapping with the original dataset. Furthermore, only 21 of these samples were re-invented and found to match entries in the considerably larger ChEMBL35 dataset, which contains 2.4 million molecules. These overlapping samples were excluded from further analysis, as they were not considered novel.

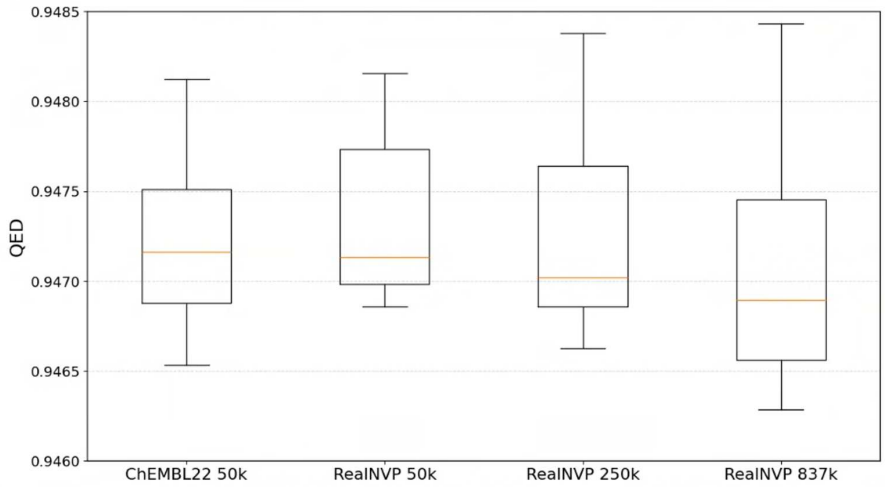

Fig. 5 Distributions of QED values for molecules with the highest 0.1% QED score in: 50,000 random samples from ChEMBL22 (training dataset), and the first 50,000, 250,000, and 837,000 generated molecules. These molecules were sampled using Real NVP normalizing flows with VAE encoding defined over one-hot-encoded SMILES. The number of data points visualized from left to right is 50, 50, 250, and 837, respectively. The mean QED values are in yellow.

Surprisingly, there was only minor degradation in performance as more than 50,000 samples were generated. This is evident in the distributions of the top 0.1% QED scores and their means for the training and generated datasets, shown in Figure 5. We did not observe an improvement in the maximum QED score in the previous tests, but it improved with the first 50,000 samples, followed by a larger improvement in the subsequent samples.

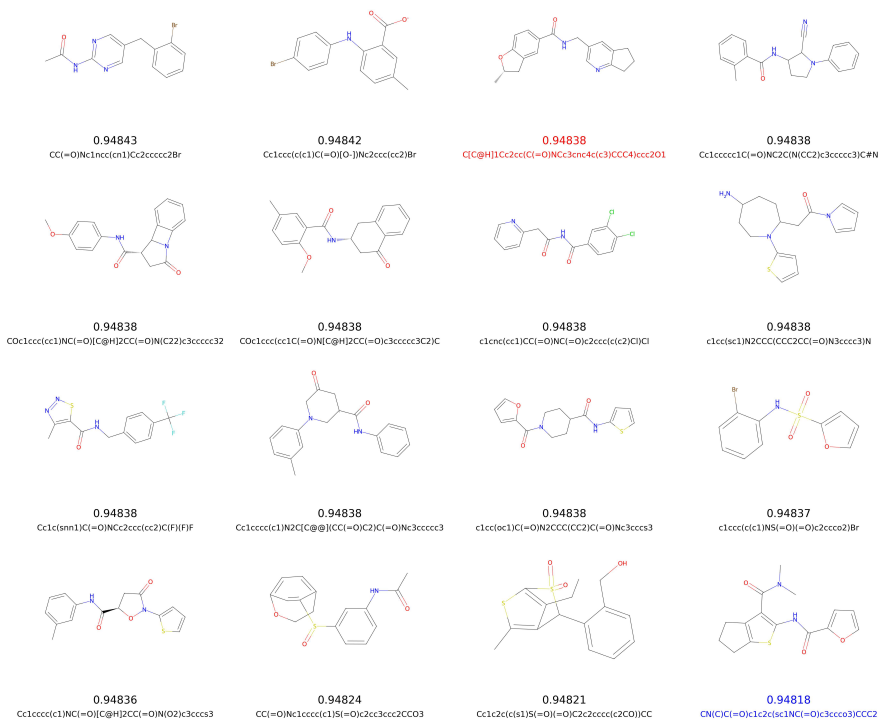

Fig. 6 The highest QED scores, the respective molecules, and their SMILES strings in the generated dataset (shown in black) and the ChEMBL dataset (for the 50,000 training dataset and the entire 2.4 million database, shown in blue and red, respectively). 17 generated molecules with the highest similarity index, calculated using Morgan fingerprint, were omitted from this Figure. The mean similarity index for the displayed molecules is 0.26 while the maximum is 0.33.

Finally, we analyzed the generated molecules that exhibited higher QED scores than any found in the training dataset. Molecules with similar molecular formulas, identical functional groups, or high structural similarity were excluded. The remaining 14 molecules, along with their QED scores, are presented in Figure 6. The highest QED score in the current version of ChEMBL35 is 0.94837, which was exceeded by four of the generated molecules. Among these, two differed by the presence of a negatively charged carboxyl group, a bromine atom, or a pyrimidine ring. Although we did not observe any significant degradation in sample quality, it is likely that retraining the VAE on the newly generated molecules could lead to further improvements.

3.2 Generation of novel organofluorine-phosphates using normalizing flows without VAE encoding

In this section, we focus on a specific class of molecules that are important in agriculture, industry, and as chemical warfare agent simulants, organofluorine-phosphates. While numerous publicly available datasets exist for bioactive molecules, none are

available for this group of compounds. This is one of the reasons why many of their properties, including toxicity patterns and reactivity, remain poorly understood.[59] There have been efforts to establish empirical guidelines for the assessment of reactivity, and the electronic density on the central phosphorus atom has been identified as a good first approximation.[60] It has been proposed that a higher electronic density at this center enhances the ability of the phosphorus atom to undergo nucleophilic attack, owing to increased electrostatic interactions with the negatively charged hydroxide nucleophile.[60] The electronic density can be approximated by an atomic electronic charge and computed using electronic structure methods based on DFT.

Since there is no comprehensive database of organofluorine-phosphates, training a specialized VAE model for this compound class is not feasible. While one could apply a VAE pre-trained on general bioactive molecules or append the desired functional group to structures from an existing database, such approaches are likely to result in a highly biased training set and low validity rates. The latter arises from the fact that the underlying molecular scaffolds may not readily accommodate the desired functional groups. To address this challenge and to demonstrate that normalizing flows can be effectively utilized in low-data regimes, we adopted molecular representations that are inherently robust and do not rely on VAEs to achieve high validity rates of generated molecules. In particular, we employed SELFIES and group-SELFIES, both of which are designed to produce syntactically valid molecular strings by construction.[31]

We initiated our search with a dataset of only 175 molecular examples published in a previous study that employed recurrent neural networks (RNNs), so we adopted an iterative architecture designed to improve and expand the training dataset over the course of the simulation.[61] The training dataset consists primarily of molecules composed of carbon atom chains and rings, containing only a limited number of functional groups or aromatic systems. The distribution of their phosphorus atomic charges is shown in blue in the top-left panel of Figure 7. Using this limited dataset, we were not successful in directly generating organofluorine-phosphates using normalizing flows; instead, we trained the model to generate the scaffolds capable of accommodating the attachment of the organofluorine-phosphate functional group, *e.g.*, -OP(C)(=O)F (see Figure 8 for examples). These scaffolds were subsequently used in the construction of organofluorine-phosphates.

We used conditional Real NVP flows with eight coupling layers and three hidden layers with 128 features each. Note the increase in model complexity compared to the previous application using a VAE, where only four coupling layers and 32 features were sufficient. This increase in complexity was driven by the higher feature dimensionality, which no longer depends solely on the VAE latent space dimension (292), but instead on the maximum length of the SELFIES encoding (equal to 16, the longest SELFIES string in the initial training dataset) and the total number of unique SELFIES tokens (17). After adding a capping empty token to make all the samples of the same length, this resulted in one-hot-encoded molecular representation with a length of 288. This dimension can rapidly increase when longer sequences and/or more tokens are considered to the point when it would no longer be feasible to effective use normalizing flows without application of a feature reduction technique. Finally, since our goal is to improve and expand the training dataset over the course of the simulation and

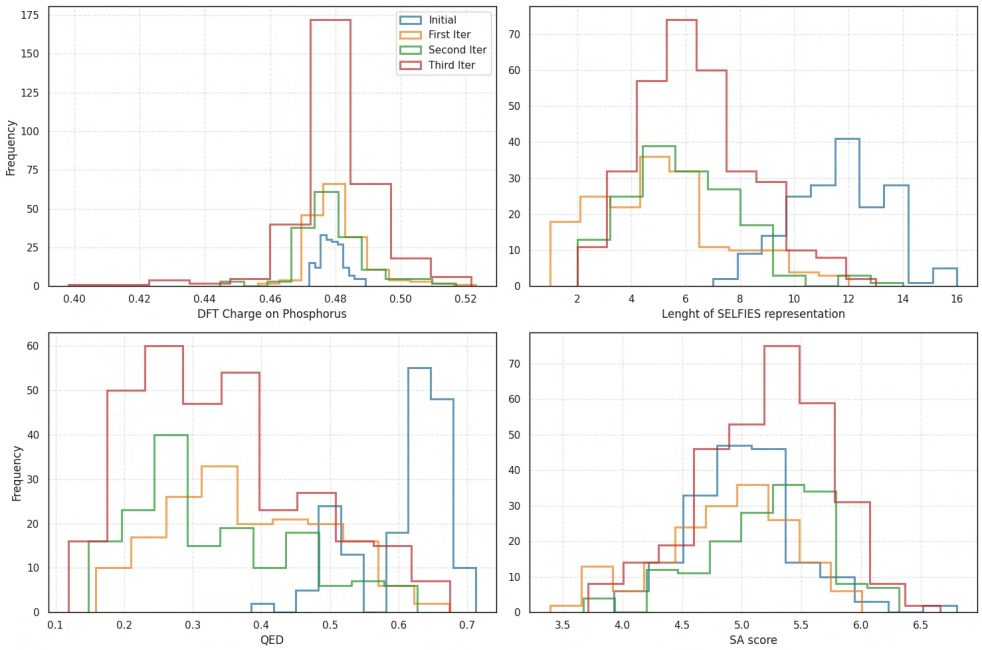

Fig. 7 Histograms of Hirshfeld charges on phosphorus atoms, length of SELFIES encoding (excluding the organofluorine-phosphate functional group), QED and SA score for the intial dataset and compared to those from the datasets generated in the first, second and third iterations.

to learn scaffolds rather than complete molecules, we opted to over-train rather than under-train the model in the beginning of the simulation.

We generated new samples in three successive iterations (500, 500, and 1,000, respectively). Out of the 2,000 generated samples, 61% (1,224) were valid and unique scaffolds. This represents a notable decline compared to previously reported SELFIES validity rates, primarily due to the added requirement that the generated scaffolds must be compatible with the attachment of the –OP(C)(=O)F functional group at the end of the string. From these 1,224 valid and unique organofluorine-phosphate molecules, we successfully determined 3D coordinates and Hirshfeld atomic charges for 55% (673 molecules) using the DFT method. These structures also exhibited positive normal modes which is indicative of the stability of the optimized geometries. Computational details are provided in Section 2.2 and examples of the generated structures can be found in columns 2, 3, and 4 of Figure 8.

The distributions of charges, SELFIES string lengths, QED scores, and SA scores are shown in Figure 7 and can be summarized as follows. No significant shift was observed in the distributions of charges on phosphorus atoms (as shown in the top-left panel). The median value remained at approximately 0.48 a.u. across the newly generated molecules over all three iterations. Nevertheless, we successfully produced examples with both higher and lower phosphorus charges than those found in the original dataset (shown in blue), indicating increased diversity. This suggests that the normalizing flow model was able to generate out-of-distribution samples consistently

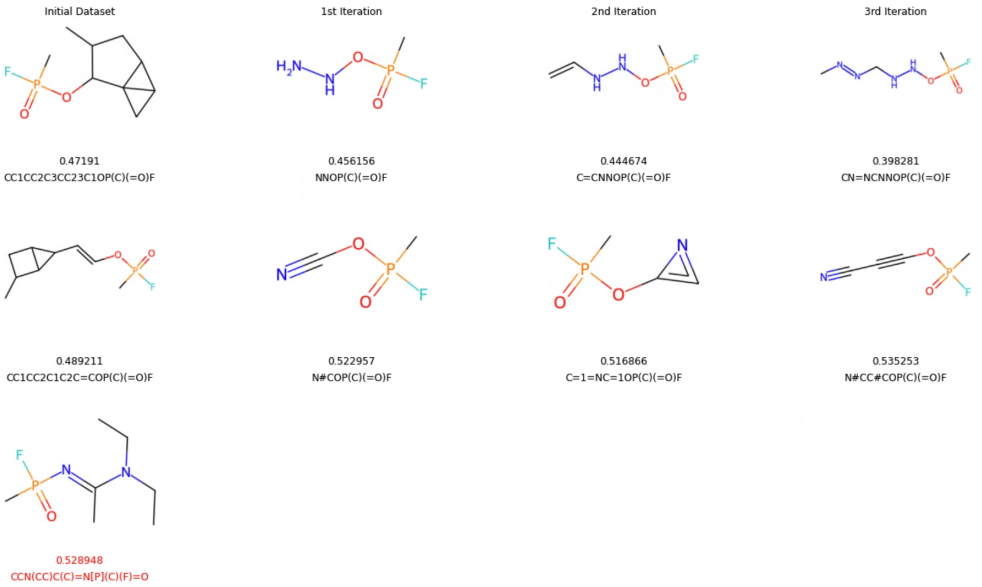

Fig. 8 The structures of organofluorine-phosphate molecules with the lowest and highest scores from the initial dataset are compared to those in the first, second and third iterations. For reference, the structure and analysis of the A230 molecule (also known as the Novichok agent) are shown in red at the bottom.

across all three iterative loops, rather than only in the initial iteration, as it was trained on both the original and previously generated data. Thus, we were successful in generating samples outside of the distribution.

After discarding token strings that do not form a valid part of a molecule (which are skipped in the SELFIES translation process), most of the resulting SELFIES strings and the resulting molecules are relatively short (see the top-right panel for the distribution of SELFIES lengths). This leads to a substantial decrease in the QED score, which can be attributed to several factors. First, we restricted the length of the SELFIES strings to match the maximum length in the initial dataset (16 tokens). In practice, we generated shorter sequences that included the capping token, resulting in considerably shorter SELFIES compared to those in the initial dataset. This issue is exacerbated by the SELFIES translation process, which skips tokens that do not form a valid part of a molecule. As a result, the length of the SELFIES is effectively shortened and the generated molecules are smaller.

We believe that using normalizing flows to generate a specific molecular type is not straightforward, as the normalizing flows approach learns molecular structures in a holistic way and does not learn shorter sequences within the molecules. This challenge may be further exacerbated by the large number of branch and ring tokens present in the training data, which complicates the generation of new samples. Additionally, we observed that our protocol was unable to generate molecules featuring aromatic rings. Therefore, a richer molecular representation, a more expressive normalizing flow model, or an expanded training dataset would be necessary to effectively learn such

molecular structures. To address some of the deficiencies, we test a novel molecular representation called group-SELFIES next.[31]

3.2.1 Performance of the group-SELFIES

To address several of the limitations observed in the previous section (such as the small molecular size, low QED scores, and the low occurrence of aromatic bonds), we now turn to a novel molecular representation known as group-SELFIES.[31] This approach builds upon standard SELFIES by incorporating an additional step: the definition of molecular fragments. By grouping atoms into chemically meaningful substructures, group-SELFIES offers a more compact and expressive representation, thereby improving the effectiveness of molecular generation. To keep the vocabulary limited while promoting diversity of the generated samples, we extracted 10 aromatic ring fragments from the ZINC250k database using RDKit fragmentation tool. To inspect these fragments, see Figure A2. Furthermore, to allow the generation of larger molecules, we increased the maximum length of the generated strings from 16 tokens to 30 tokens.

Table 2 compares the initial dataset with the generated molecules using the two representations, SELFIES and group-SELFIES. The mean QED score, SA score and number of heavy atoms were the same for both models: 0.40, 5, and 11, respectively. The SELFIES had a higher validity rate compared to the group-SELFIES model, resulting in 7,329 and 3,212 valid and unique molecules. This also resulted in almost twice the number of SELFIES samples with QED higher than any of the molecules in the training dataset: 70 compared to 39 for group-SELFIS (see the top examples in Figure 10).

Out of the 39 samples, all contained a ZINC250k aromatic ring fragment. While the SELFIES model generated several molecules with aromatic rings, these did not achieve high QED scores. Interestingly, the generated organofluorine-phosphate molecule with the second-highest QED score contains a cyclobutadiene, which has well-known antiaromatic properties. Although sterically bulky substituents or metal ligand binding can stabilize this moiety, it is not expected to be the case here.[62] This serves as a reminder that our molecular representation is not chemistry-aware, and generation under synthetic constraints is one of the possible future directions of this work.[63]

Both models were able to generate samples with higher QED score than any in the training dataset. However, group-SELFIES generated samples with a higher maximum QED score of 0.83, compared to 0.76 generated using the SELFIES encoding (see Table 2). Finally, while we observed a decline in the high QED scores in the later iterations using the SELFIES encoding, molecules generated using group-SELFIES continued to improve. This could be due to the small size of the training dataset, which lacks the fragments present in the ZINC250k database. These missing examples are only added as the simulation progresses.

Finally, let us briefly comment on the use of group-SELFIES.[31] This representation offers a promising improvement over traditional molecular encodings, as it allows longer sequences to be compressed by defining chemically meaningful fragments. However, the number of unique tokens grows rapidly with dataset size, as each fragment can have multiple variants depending on how it is connected to the rest of

Table 2 Test of the group-SELFIES and SELFIES representations in a generative task using Real NVP normalizing flows with 8 layers, each containing 128 neurons. The training set consisted of 175 organofluorine-phosphate molecules. Three iterations were performed, with each iteration enriching the training set with all the generated examples from the previous iterations.

Representation	Generated structures	QED mean/max	SA score mean	Heavy atoms mean	Aromatic rings*
SELFIES	7329	0.40 / 0.76	5.3	11.6	0%
group-SELFIES	3212	0.40 / 0.83	5.7	11.7	8% (5%)
Initial dataset	175	0.61 / 0.71	5.0	12.7	0%

* The percentage in the bracket for group-SELFIES is the occurance of the samples containing any group-SELFIES token in the generated valid string.

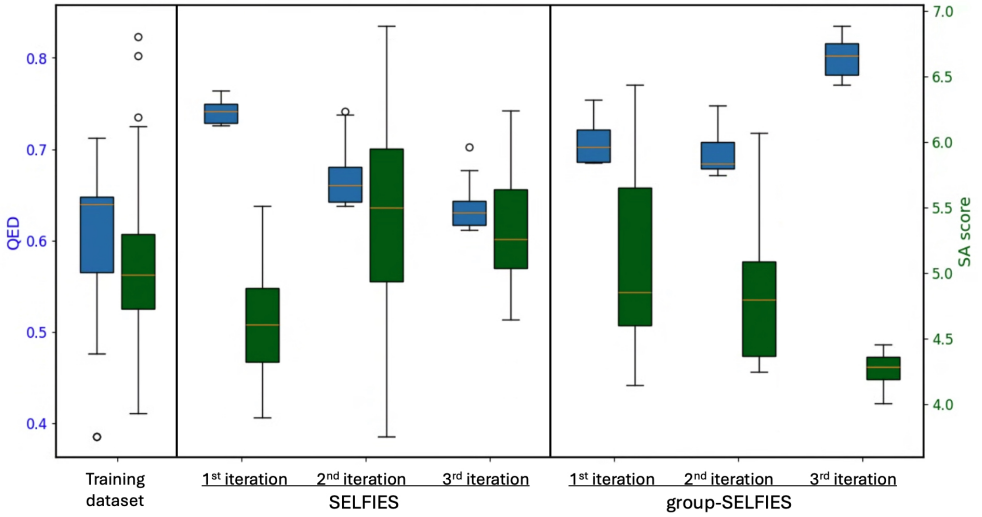

Fig. 9 Distributions of QED values for molecules in the training dataset and the top 1% of generated molecules (ranked by QED) across three iterations. Generation was performed using Real NVP normalizing flows with one-hot-encoded SELFIES and group-SELFIES representations.

the molecule. This leads to an increasingly large alphabet. As a result, even with a fixed maximum string length, the dimensionality of the one-hot-encoded representation increases, complicating its application in normalizing flows. This limits the model performance and requires more complex model architectures, as normalizing flows rely on relatively simple transformations and often struggle with high-dimensional, complex distributions.

4 Conclusions

In this paper, we presented the applications of normalizing flows to molecular design and integrated variational autoencoders with normalizing flows into a comprehensive workflow. We showed that normalizing flows can be effectively combined with one-hot-encoded SMILES representations compressed and transformed using variational

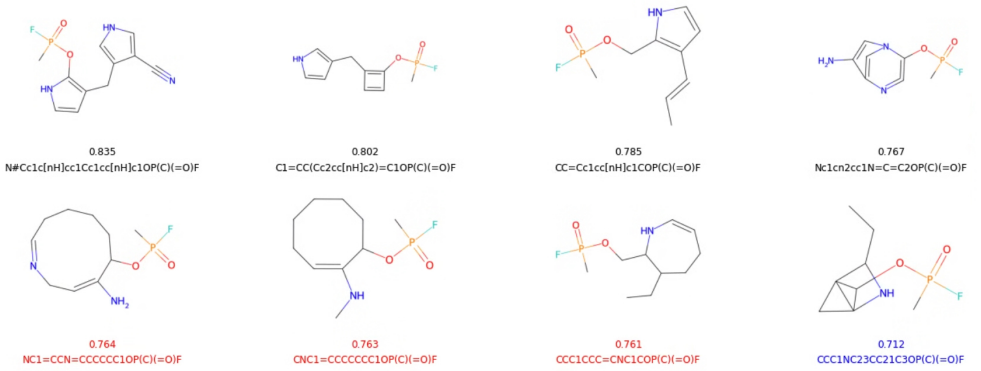

Fig. 10 The molecule with the highest QED in the training dataset, molecules generated using one-hot-encoded SELFIES and group-SELFIES are in blue, red, and black, respectively. Their QED scores and SMILES strings are included under the corresponding molecular structures for reference.

autoencoders, as well as directly with the one-hot-encoded SELFIES and group-SELFIES representations. These approaches yield higher validity scores compared to the SMILES representation, as expected.

Additionally, we found that normalizing flows can improve the sampling efficiency of other models, such as variational autoencoders. While the increase in the percentage of valid and unique samples was relatively modest, the mean QED score more than doubled, the maximum QED score increased, and the mean SA score improved by approximately 40%. We identified several novel molecules with QED scores exceeding those found in the 2.4 million-compound ChEMBL database, highlighting the potential of our approach for discovering highly promising drug candidates.

Normalizing flows were capable of learning highly non-Gaussian posterior densities, including those found in discrete chemical space in the molecular design. Furthermore, normalizing flow-based models can be conditioned on additional variables (for instance a costly target metrics such as binding energies or electronic charges) which could be leveraged to develop goal-oriented algorithms, targeted therapies and optimization of complex molecular objectives.

Further work is needed to develop suitable molecular representations that are both sufficiently expressive and efficient in order to leverage normalizing flows. Currently, the field of normalizing flows is mature enough to both support the development of improved sampling strategies for a wide range of models and serve as a promising competitive alternative to methods like Monte Carlo when domain knowledge has a potential to improve sampling. Normalizing flows are particularly well-suited for domains with sparse data such as material design for high-entropy alloys. It remains to be seen whether our results continue to hold in these and other domains, which we plan to explore in our future research.

Declarations

Funding. This project is supported by the National Research Council Canada (NRC) and the Defence Research and Development Canada (DRDC).

Author Contributions. Conceptualization and study design: MSG and JH. Study implementation: JH. Manuscript writing: JH. Administration, supervision and funding: AH and HKO. All authors reviewed manuscript.

Data Availability. The code and data are available at http://github.com/nrc-cnrc/VNFlow.

Competing interests. All authors declare that they have nothing to disclose.

Appendix A Additional Supporting Material

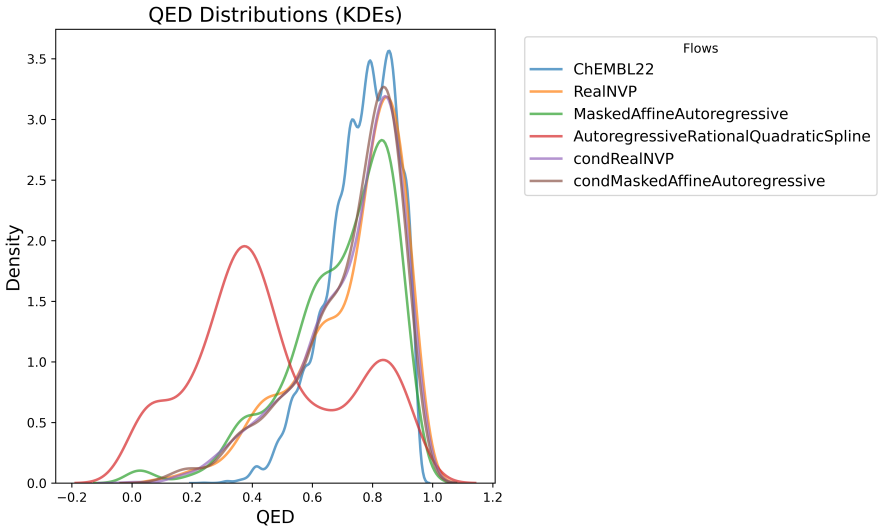

Fig. A1 Kernel Density Estimation (KDE) of QED distributions for the starting dataset (50,000 structres from ChEMBL daset) compared with those generated by different normalizing flows.

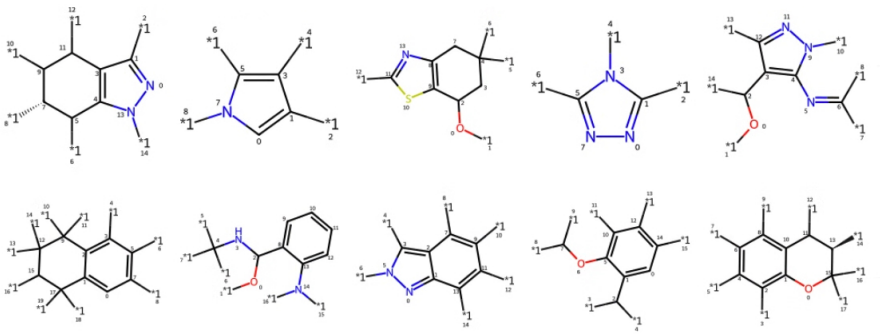

Fig. A2 Fragments containing aromatic rings and generated from ZINC250k database.

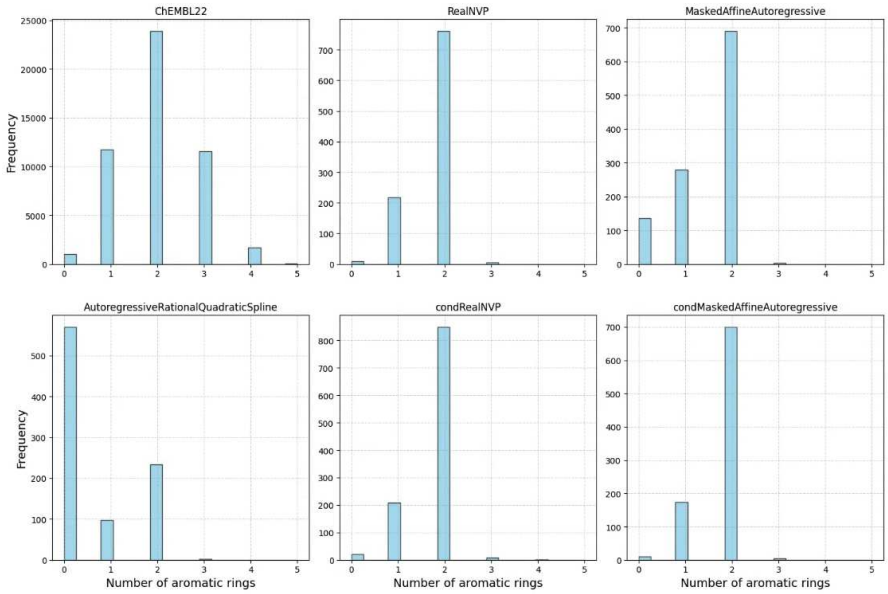

Fig. A3 The number of aromatic rings in the starting dataset (50,000 structres from ChEMBL daset) compared with those generated by different normalizing flows. For comparison, the random latent vectors decoded by VAE resulted in 735 non-aromatic molecules, 16 molecules with one aromatic ring and 1 molecule with two.

Table A1 Hyperparameter search for VAE model trained on one-hot-encoded SELFIES. Decoder architecture was kept fixed while number of convolutional layers used in the encoder was varied.

Model	Convolutional layers and their kernel size	Training Loss	Validation Loss
Chosen model	(9, 9, 9)	17.1	27.4
Setting #2	(9, 9, 9, 9)	39.8	46.4
Setting #3	(12, 12, 12)	17.0	27.8
Setting #4	(12, 12, 12, 12)	27.0	37.1

Note: The progression of training and validation losses of the chosen model is depicted in the Figure A5.

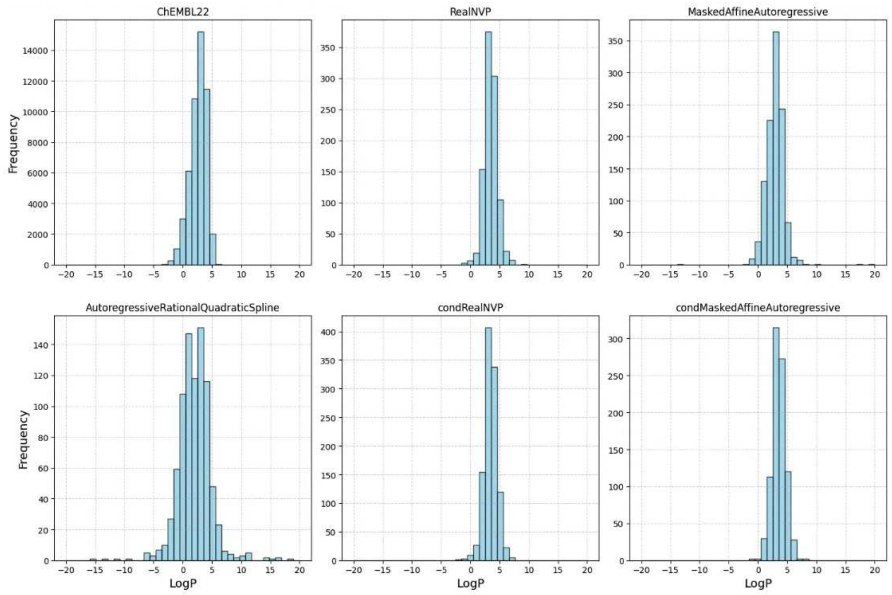

Fig. A4 The logP values in the starting dataset (50,000 structres from ChEMBL daset) compared with those generated by different normalizing flows.

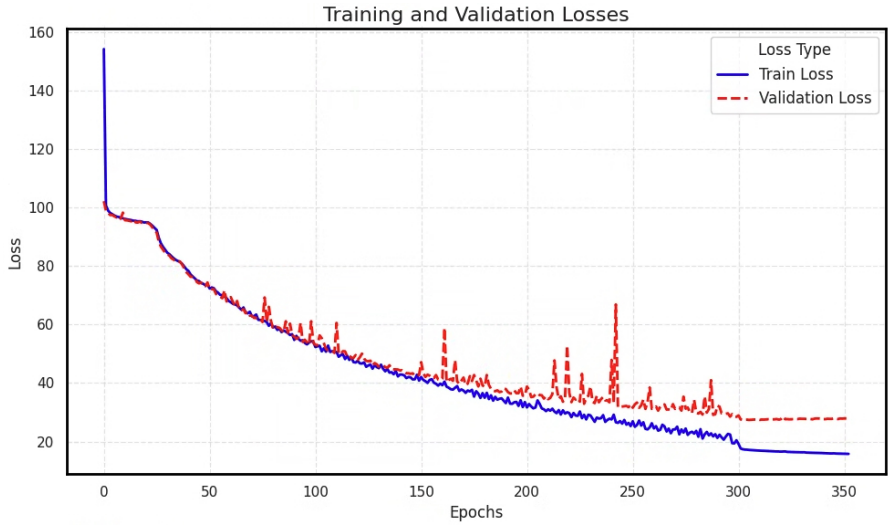

Fig. A5 Training and Validation losses during the optimization of VAE model using one-hot-encoded SELFIES as input.

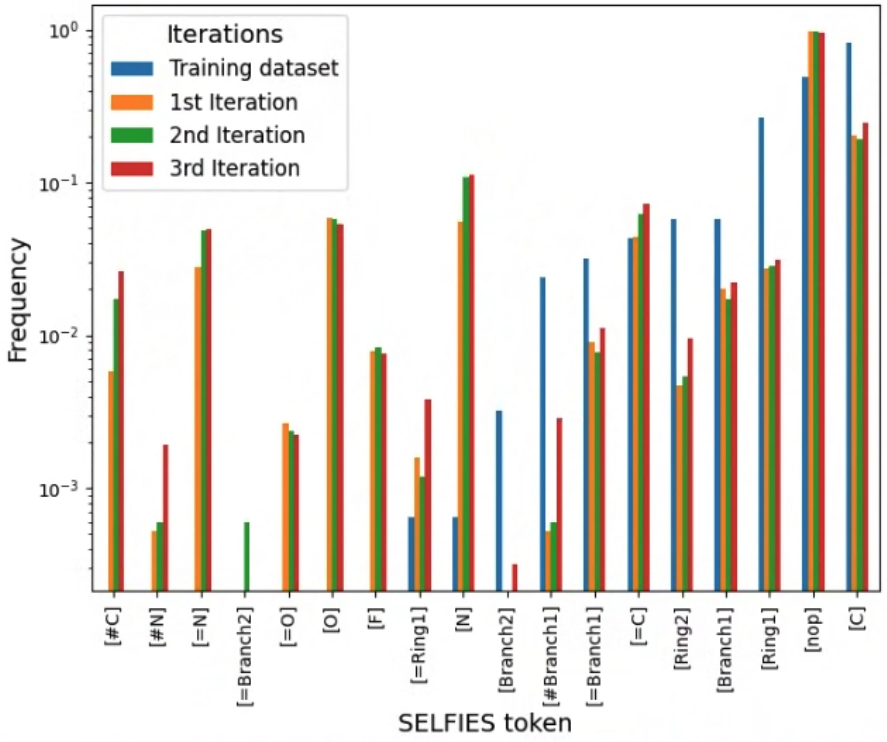

Fig. A6 Visualization of the occurrence of SELFIES tokens for the first three iterations (see details in section 3.2). Tokens are ordered based on their frequency in the training dataset. The frequency, plotted on a logarithmic scale, is calculated as the ratio of valid samples containing each token in the dataset.

References

[1] Bienstock, R.J.: Ai/ml methodologies and the future-will they be successful in designing the next generation of new chemical entities? Journal of Cheminformatics **17**(1), 46 (2025)

[2] Hinkson, I.V., Madej, B., Stahlberg, E.A.: Accelerating therapeutics for opportunities in medicine: A paradigm shift in drug discovery. Frontiers in Pharmacology **11** (2020) https://doi.org/10.3389/fphar.2020.00770

[3] Avorn, J.: The $2.6 billion pill — methodologic and policy considerations. New England Journal of Medicine **372**(20), 1877–1879 (2015) https://doi.org/10.1056/NEJMp1500848 https://www.nejm.org/doi/pdf/10.1056/NEJMp1500848

[4] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., *et al.*: Highly accurate protein structure prediction with alphafold. Nature **596**(7873), 583–589 (2021)

[5] Varadi, M., Bertoni, D., Magana, P., Paramval, U., Pidruchna, I., Radhakrishnan, M., Tsenkov, M., Nair, S., Mirdita, M., Yeo, J., Kovalevskiy, O., Tunyasuvunakool, K., Laydon, A., Žídek, A., Tomlinson, H., Hariharan, D., Abrahamson, J., Green, T., Jumper, J., Birney, E., Steinegger, M., Hassabis, D., Velankar, S.: Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences. Nucleic Acids Research **52**(D1), 368–375 (2023) https://doi.org/10.1093/nar/gkad1011 https://academic.oup.com/nar/article-pdf/52/D1/D368/55039845/gkad1011.pdf

[6] Bohacek, R.S., McMartin, C., Guida, W.C.: The art and practice of structure-based drug design: a molecular modeling perspective. Medicinal research reviews **16**(1), 3–50 (1996)

[7] Lyne, P.D.: Structure-based virtual screening: an overview. Drug discovery today **7**(20), 1047–1055 (2002)

[8] Mendez, D., Gaulton, A., Bento, A.P., Chambers, J., De Veij, M., Félix, E., Magariños, M., Mosquera, J., Mutowo, P., Nowotka, M., Gordillo-Marañón, M., Hunter, F., Junco, L., Mugumbate, G., Rodriguez-Lopez, M., Atkinson, F., Bosc, N., Radoux, C., Segura-Cabrera, A., Hersey, A., Leach, A.: Chembl: towards direct deposition of bioassay data. Nucleic Acids Research **47**(D1), 930–940 (2018) https://doi.org/10.1093/nar/gky1075 https://academic.oup.com/nar/article-pdf/47/D1/D930/27437436/gky1075.pdf

[9] Reymond, J.-L.: The chemical space project. Accounts of Chemical Research **48**(3), 722–730 (2015) https://doi.org/10.1021/ar500432k . PMID: 25687211

[10] Elton, D.C., Boukouvalas, Z., Fuge, M.D., Chung, P.W.: Deep learning for molecular design—a review of the state of the art. Mol. Syst. Des. Eng. **4**, 828–849 (2019) https://doi.org/10.1039/C9ME00039A

[11] De Cao, N., Kipf, T.: MolGAN: An implicit generative model for small molecular graphs. ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models (2018)

[12] Mikolov, T., Kombrink, S., Burget, L., Černocký, J., Khudanpur, S.: Extensions of recurrent neural network language model. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5528–5531 (2011). https://doi.org/10.1109/ICASSP.2011.5947611

[13] Weininger, D.: 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) https://doi.org/10.1021/ci00057a005

[14] Bjerrum, E.J., Threlfall, R.: Molecular generation with recurrent neural networks (rnns). CoRR **abs/1705.04612** (2017) 1705.04612

[15] Deursen, I.V.T. Peter Ertl, Godin, G.: Gen: highly efficient smiles explorer using autodidactic generative examination networks. J. Cheminform. **12**, 22 (2020)

[16] Arús-Pous, J., Johansson, S.V., Prykhodko, O., Bjerrum, E.J., Tyrchan, C., Reymond, J.L., Chen, H., Engkvist, O.: Randomized smiles strings improve the quality of molecular generative models. J. Cheminform. **11**, 71 (2019)

[17] Gómez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hernández-Lobato, J.M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., Aspuru-Guzik, A.: Automatic chemical design using a data-driven continuous representation of molecules. ACS Central Science **4**(2), 268–276 (2018) https://doi.org/10.1021/acscentsci.7b00572 . PMID: 29532027

[18] Kusner, M.J., Paige, B., Hernández-Lobato, J.M.: Grammar Variational Autoencoder (2017). https://arxiv.org/abs/1703.01925

[19] Dinh, L., Krueger, D., Bengio, Y.: NICE: Non-linear Independent Components Estimation (2015). https://arxiv.org/abs/1410.8516

[20] Papamakarios, G., Pavlakou, T., Murray, I.: Masked Autoregressive Flow for Density Estimation (2018). https://arxiv.org/abs/1705.07057

[21] Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real NVP. In: International Conference on Learning Representations (2017). https://openreview.net/forum?id=HkpbnH9lx

[22] Zang, C., Wang, F.: Moflow: An invertible flow model for generating molecular

graphs. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery. KDD '20, pp. 617–626 (2020). https://doi.org/10.1145/3394486.3403104 . ACM

[23] Kingma, D.P., Dhariwal, P.: Glow: Generative Flow with Invertible 1x1 Convolutions (2018). https://arxiv.org/abs/1807.03039

[24] Kuznetsov, M., Polykovskiy, D.: MolGrow: A Graph Normalizing Flow for Hierarchical Molecular Generation (2021). https://arxiv.org/abs/2106.05856

[25] Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., Tang, J.: GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation (2020). https://arxiv.org/abs/2001.09382

[26] Frey, N.C., Gadepally, V., Ramsundar, B.: FastFlows: Flow-Based Models for Molecular Graph Generation (2022). https://arxiv.org/abs/2201.12419

[27] Morrow, R., Chiu, W.-C.: Variational Autoencoders with Normalizing Flow Decoders (2020). https://arxiv.org/abs/2004.05617

[28] Rezende, D.J., Mohamed, S.: Variational Inference with Normalizing Flows (2016). https://arxiv.org/abs/1505.05770

[29] Wu, J.-N., Wang, T., Chen, Y., Tang, L.-J., Wu, H.-L., Yu, R.-Q.: t-smiles: a fragment-based molecular representation framework for de novo ligand design. Nature Communications **15**(1), 4993 (2024)

[30] Krenn, M., Häse, F., Nigam, A., Friederich, P., Aspuru-Guzik, A.: Self-referencing embedded strings (selfies): A 100% robust molecular string representation. Machine Learning: Science and Technology **1**(4), 045024 (2020) https://doi.org/10.1088/2632-2153/aba947

[31] Cheng, A.H., Cai, A., Miret, S., Malkomes, G., Phielipp, M., Aspuru-Guzik, A.: Group selfies: a robust fragment-based molecular string representation. Digital Discovery **2**, 748–758 (2023) https://doi.org/10.1039/D3DD00012E

[32] Irwin, J.J., Sterling, T., Mysinger, M.M., Bolstad, E.S., Coleman, R.G.: Zinc: a free tool to discover chemistry for biology. Journal of chemical information and modeling **52**(7), 1757–1768 (2012)

[33] Wigh, D.S., Goodman, J.M., Lapkin, A.A.: A review of molecular representation in the age of machine learning. WIREs Computational Molecular Science **12**(5), 1603 (2022) https://doi.org/10.1002/wcms.1603 https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1603

[34] Ramakrishnan, R., Dral, P.O., Rupp, M., Von Lilienfeld, O.A.: Quantum chemistry structures and properties of 134 kilo molecules. Scientific data **1**(1), 1–7

(2014)

[35] RDKit: Open-source Cheminformatics. https://doi.org/10.5281/zenodo.59163 . https://www.rdkit.org

[36] Jin, W., Barzilay, R., Jaakkola, T.: Hierarchical generation of molecular graphs using structural motifs. In: International Conference on Machine Learning, pp. 4839–4848 (2020). PMLR

[37] Teale, A.M., Helgaker, T., Savin, A., Adamo, C., Aradi, B., Arbuznikov, A.V., Ayers, P.W., Baerends, E.J., Barone, V., Calaminici, P., Cancès, E., Carter, E.A., Chattaraj, P.K., Chermette, H., Ciofini, I., Crawford, T.D., De Proft, F., Dobson, J.F., Draxl, C., Frauenheim, T., Fromager, E., Fuentealba, P., Gagliardi, L., Galli, G., Gao, J., Geerlings, P., Gidopoulos, N., Gill, P.M.W., Gori-Giorgi, P., Görling, A., Gould, T., Grimme, S., Gritsenko, O., Jensen, H.J.A., Johnson, E.R., Jones, R.O., Kaupp, M., Köster, A.M., Kronik, L., Krylov, A.I., Kvaal, S., Laestadius, A., Levy, M., Lewin, M., Liu, S., Loos, P.-F., Maitra, N.T., Neese, F., Perdew, J.P., Pernal, K., Pernot, P., Piecuch, P., Rebolini, E., Reining, L., Romaniello, P., Ruzsinszky, A., Salahub, D.R., Scheffler, M., Schwerdtfeger, P., Staroverov, V.N., Sun, J., Tellgren, E., Tozer, D.J., Trickey, S.B., Ullrich, C.A., Vela, A., Vignale, G., Wesolowski, T.A., Xu, X., Yang, W.: Dft exchange: sharing perspectives on the workhorse of quantum chemistry and materials science. Phys. Chem. Chem. Phys. **24**, 28700–28781 (2022) https://doi.org/10.1039/D2CP02827A

[38] Ye, N., Yang, Z., Liu, Y.: Applications of density functional theory in covid-19 drug modeling. Drug Discovery Today **27**(5), 1411–1419 (2022) https://doi.org/10.1016/j.drudis.2021.12.017

[39] Jakubec, D., Hostaš, J., Laskowski, R.A., Hobza, P., Vondrášek, J.: Large-scale quantitative assessment of binding preferences in protein–nucleic acid complexes. Journal of Chemical Theory and Computation **11**(4), 1939–1948 (2015)

[40] Neese, F.: The orca program system. WIRES Comput. Molec. Sci. **2**(1), 73–78 (2012) https://doi.org/10.1002/wcms.81

[41] Becke, A.D.: Density-functional thermochemistry. i. the effect of the exchange-only gradient correction. The Journal of chemical physics **96**(3), 2155–2160 (1992)

[42] Caldeweyher, E., Bannwarth, C., Grimme, S.: Extension of the d3 dispersion coefficient model. The Journal of Chemical Physics **147**(3), 034112 (2017) https://doi.org/10.1063/1.4993215

[43] Weigend, F., Ahlrichs, R.: Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for h to rn: Design and assessment of accuracy. Phys. Chem. Chem. Phys. **7**, 3297–3305 (2005) https://doi.org/10.1039/B508541A

[44] Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes (2022). https://arxiv.org/abs/1312.6114

[45] Kingma, D.P., Welling, M.: An introduction to variational autoencoders. Foundations and Trends® in Machine Learning **12**(4), 307–392 (2019) https://doi.org/10.1561/2200000056

[46] Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. In: Riezler, S., Goldberg, Y. (eds.) Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, pp. 10–21. Association for Computational Linguistics, Berlin, Germany (2016). https://doi.org/10.18653/v1/K16-1002 . https://aclanthology.org/K16-1002/

[47] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling (2014). https://arxiv.org/abs/1412.3555

[48] Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M.Y., Suk, H., Suo, M., Tillet, P., Wang, E., Wang, X., Wen, W., Zhang, S., Zhao, X., Zhou, K., Zou, R., Mathews, A., Chanan, G., Wu, P., Chintala, S.: PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In: 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24), vol. 2, pp. 929–947 (2024). https://doi.org/10.1145/3620665.3640366 . ACM

[49] Williams, M.J., jmcginn, federicostak, Veitch, J.: Uofgravity/glasflow: V0.4.1. https://doi.org/10.5281/zenodo.13914483 . https://doi.org/10.5281/zenodo.13914483

[50] Durkan, C., Bekasov, A., Murray, I., Papamakarios, G.: nflows: Normalizing Flows in PyTorch. https://doi.org/10.5281/zenodo.4296287 . https://doi.org/10.5281/zenodo.4296287

[51] Durkan, C., Bekasov, A., Murray, I., Papamakarios, G.: Neural spline flows. In: Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32 (2019)

[52] Stimper, V., Liu, D., Campbell, A., Berenz, V., Ryll, L., Schölkopf, B., Hernández-Lobato, J.M.: normflows: A pytorch package for normalizing flows. Journal of Open Source Software **8**(86), 5361 (2023) https://doi.org/10.21105/joss.05361

[53] Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing Flows for Probabilistic Modeling and Inference (2021). https://arxiv.org/abs/1912.02762

[54] Weng, L.: Flow-based deep generative models. lilianweng.github.io (2018)

[55] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[56] The pandas development team: Pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3509134

[57] Harris, C.R., Millman, K.J., Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M.H., Brett, M., Haldane, A., Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E.: Array programming with NumPy. Nature **585**(7825), 357–362 (2020) https://doi.org/10.1038/s41586-020-2649-2

[58] Skinnider, M.A.: Invalid SMILES are beneficial rather than detrimental to chemical language models. Nature Machine Intelligence **6**(4), 437–448 (2024) https://doi.org/10.1038/s42256-024-00821-x

[59] Karaboga, S., Severac, F., Collins, E.-M.S., Stab, A., Davis, A., Souchet, M., Hervé, G.: Organophosphate toxicity patterns: A new approach for assessing organophosphate neurotoxicity. Journal of Hazardous Materials **470**, 134236 (2024) https://doi.org/10.1016/j.jhazmat.2024.134236

[60] Jeong, K., Choi, J.: Theoretical study on the toxicity of 'novichok' agent candidates. Royal Society Open Science **6**(8), 190414 (2019) https://doi.org/10.1098/rsos.190414

[61] Hu, H., Ooi, H.K., Ghaemi, M.S., Hu, A.: Machine learning for the prediction of safe and biologically active organophosphorus molecules. Canadian Artificial Intelligence Association (CAIAC) (2023). https://caiac.pubpub.org/pub/ubvwd1py

[62] Maier, G., Pfriem, S., Schäfer, U., Matusch, R.: Tetra-tert-butyltetrahedrane. Angewandte Chemie International Edition in English **17**(7), 520–521 (1978) https://doi.org/10.1002/anie.197805201

[63] Parrot, M., Tajmouati, H., Silva, V.B.R., Atwood, B.R., Fourcade, R., Gaston-Mathé, Y., Do Huu, N., Perron, Q.: Integrating synthetic accessibility with ai-based generative drug design. Journal of Cheminformatics **15**(1), 83 (2023)