ACS central science

Research Article

Cite This: ACS Cent. Sci. XXXX, XXX, XXX−XXX

Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules

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*[,‡,⊥]

[†]Kyulux North America Inc., 10 Post Office Square, Suite 800, Boston, Massachusetts 02109, United States

[‡]Department of Chemistry and Chemical Biology, Harvard University, Cambridge, Massachusetts 02138, United States

[¶]Department of Computer Science, University of Toronto, 6 King's College Road, Toronto, Ontario M5S 3H5, Canada

[§]Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, U.K.

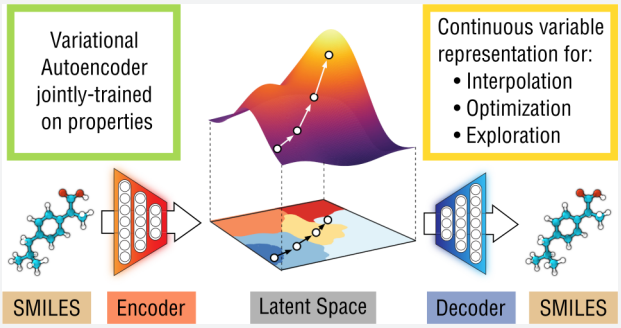
[∇]Google Brain, Mountain View, California, United States

[∥]Princeton University, Princeton, New Jersey, United States

[⊥]Biologically-Inspired Solar Energy Program, Canadian Institute for Advanced Research (CIFAR), Toronto, Ontario M5S 1M1, Canada

Supporting Information

ABSTRACT: We report a method to convert discrete representations of molecules to and from a multidimensional continuous representation. This model allows us to generate new molecules for efficient exploration and optimization through open-ended spaces of chemical compounds. A deep neural network was trained on hundreds of thousands of existing chemical structures to construct three coupled functions: an encoder, a decoder, and a predictor. The encoder converts the discrete representation of a molecule into a real-valued continuous vector, and the decoder converts these continuous vectors back to discrete molecular representations. The predictor estimates chemical properties from the latent continuous vector representation of the molecule. Continuous representations of molecules allow us to automatically generate novel chemical structures by performing simple operations in the latent space, such as decoding random vectors, perturbing known chemical structures, or interpolating between molecules. Continuous representations also allow the use of powerful gradient-based optimization to efficiently guide the search for optimized functional compounds. We demonstrate our method in the domain of drug-like molecules and also in a set of molecules with fewer that nine heavy atoms.



INTRODUCTION

The goal of drug and material design is to identify novel molecules that have certain desirable properties. We view this as an optimization problem, in which we are searching for the molecules that maximize our quantitative desiderata. However, optimization in molecular space is extremely challenging, because the search space is large, discrete, and unstructured. Making and testing new compounds are costly and time-consuming, and the number of potential candidates is overwhelming. Only about 10^8 substances have ever been synthesized,[1] whereas the range of potential drug-like molecules is estimated to be between 10^{23} and 10^{60}.[2]

Virtual screening can be used to speed up this search.[3−6] Virtual libraries containing thousands to hundreds of millions of candidates can be assayed with first-principles simulations or statistical predictions based on learned proxy models, and only the most promising leads are selected and tested experimentally.

However, even when accurate simulations are available,[7] computational molecular design is limited by the search strategy used to explore chemical space. Current methods either exhaustively search through a fixed library,[8,9] or use discrete local search methods such as genetic algorithms[10−15] or similar discrete interpolation techniques.[16−18] Although these techniques have led to useful new molecules, these approaches still face large challenges. Fixed libraries are monolithic, costly to fully explore, and require hand-crafted rules to avoid impractical chemistries. The genetic generation of compounds requires manual specification of heuristics for mutation and crossover rules. Discrete optimization methods have difficulty

Received: December 2, 2017

DOI: 10.1021/acscentsci.7b00572
ACS Cent. Sci. XXXX, XXX, XXX−XXX

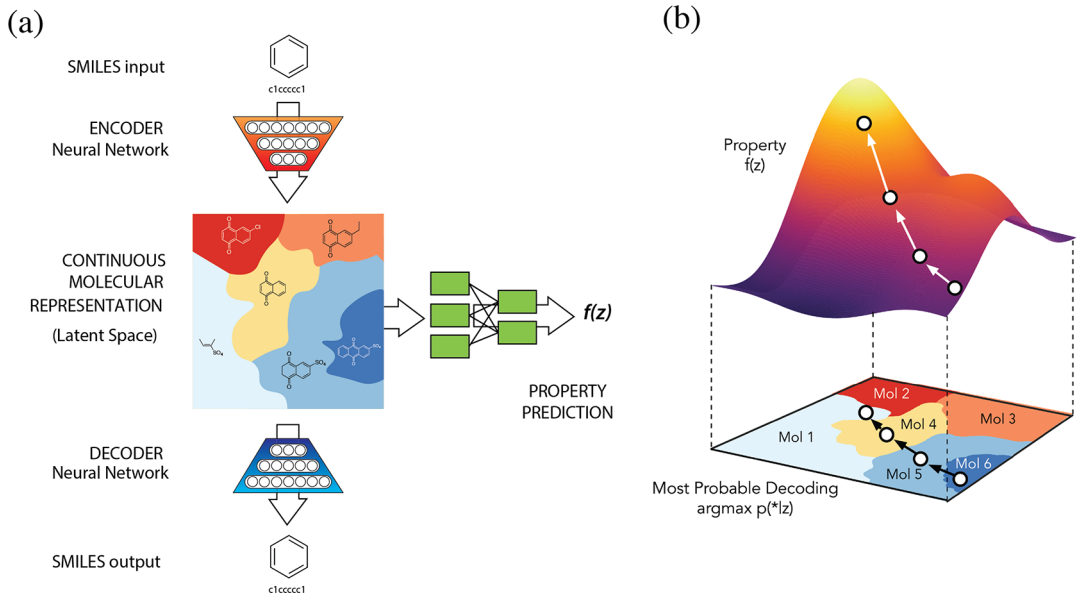


Figure 1. (a) A diagram of the autoencoder used for molecular design, including the joint property prediction model. Starting from a discrete molecular representation, such as a SMILES string, the encoder network converts each molecule into a vector in the latent space, which is effectively a continuous molecular representation. Given a point in the latent space, the decoder network produces a corresponding SMILES string. A mutilayer perceptron network estimates the value of target properties associated with each molecule. (b) Gradient-based optimization in continuous latent space. After training a surrogate model $f(z)$ to predict the properties of molecules based on their latent representation z, we can optimize $f(z)$ with respect to z to find new latent representations expected to have high values of desired properties. These new latent representations can then be decoded into SMILES strings, at which point their properties can be tested empirically.

effectively searching large areas of chemical space because it is not possible to guide the search with gradients.

A molecular representation method that is continuous, data-driven, and can easily be converted into a machine-readable molecule has several advantages. First, hand-specified mutation rules are unnecessary, as new compounds can be generated automatically by modifying the vector representation and then decoding. Second, if we develop a differentiable model that maps from molecular representations to desirable properties, we can enable the use of gradient-based optimization to make larger jumps in chemical space. Gradient-based optimization can be combined with Bayesian inference methods to select compounds that are likely to be informative about the global optimum. Third, a data-driven representation can leverage large sets of unlabeled chemical compounds to automatically build an even larger implicit library, and then use the smaller set of labeled examples to build a regression model from the continuous representation to the desired properties. This lets us take advantage of large chemical databases containing millions of molecules, even when many properties are unknown for most compounds.

Recent advances in machine learning have resulted in powerful probabilistic generative models that, after being trained on real examples, are able to produce realistic synthetic samples. Such models usually also produce low-dimensional continuous representations of the data being modeled, allowing interpolation or analogical reasoning for natural images,[19] text,[20] speech, and music.[21,22] We apply such generative models to chemical design, using a pair of deep networks trained as an autoencoder to convert molecules represented as SMILES strings into a continuous vector representation. In principle, this method of converting from a molecular representation to a continuous vector representation could be applied to any molecular representation, including chemical fingerprints,[23] convolutional neural networks on graphs,[24] similar graph-convolutions,[25] and Coulomb matrices.[26] We chose to use SMILES representation because this representation can be readily converted into a molecule.

Using this new continuous vector-valued representation, we experiment with the use of continuous optimization to produce novel compounds. We trained the autoencoder jointly on a property prediction task: we added a multilayer perceptron that predicts property values from the continuous representation generated by the encoder, and included the regression error in our loss function. We then examined the effects that joint training had on the latent space, and tested optimization in this latent space for new molecules that optimize our desired properties.

Representation and Autoencoder Framework. The autoencoder in comprised of two deep networks: an encoder network to convert each string into a fixed-dimensional vector, and a decoder network to convert vectors back into strings (Figure 1a). The autoencoder is trained to minimize error in reproducing the original string; i.e., it attempts to learn the identity function. Key to the design of the autoencoder is the mapping of strings through an *information bottleneck*. This bottleneck—here the fixed-length continuous vector—induces the network to learn a compressed representation that captures the most statistically salient information in the data. We call the vector-encoded molecule the *latent representation* of the molecule.

For unconstrained optimization in the latent space to work, points in the latent space must decode into valid SMILES strings that capture the chemical nature of the training data. Without this constraint, the latent space learned by the autoencoder may be sparse and may contain large "dead areas", which decode to invalid SMILES strings. To help ensure that points in the latent space correspond to valid realistic

DOI: 10.1021/acscentsci.7b00572
ACS Cent. Sci. XXXX, XXX, XXX−XXX

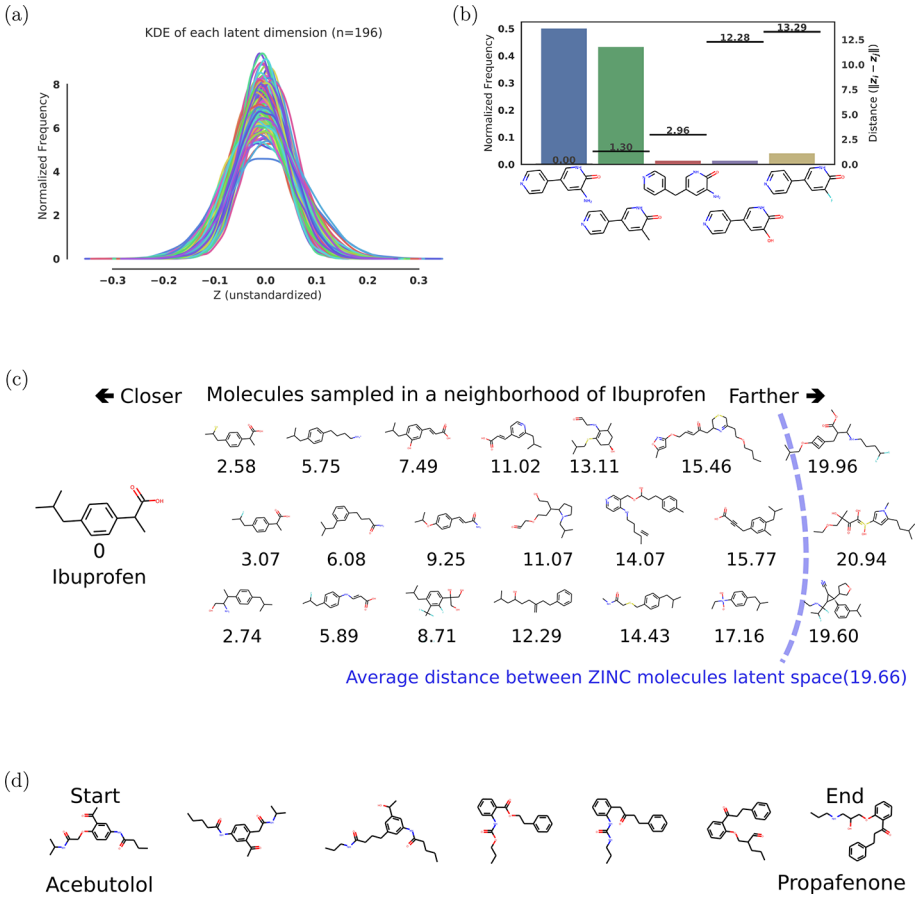


Figure 2. Representations of the sampling results from the variational autoencoder. (a) Kernel Density Estimation (KDE) of each latent dimension of the autoencoder, i.e., the distribution of encoded molecules along each dimension of our latent space representation; (b) histogram of sampled molecules for a single point in the latent space; the distances of the molecules from the original query are shown by the lines corresponding to the right axis; (c) molecules sampled near the location of ibuprofen in latent space. The values below the molecules are the distance in latent space from the decoded molecule to ibuprofen; (d) *slerp* interpolation between two molecules in latent space using six steps of equal distance.

molecules, we chose to use a *variational* autoencoder (VAE)[27] framework. VAEs were developed as a principled approximate-inference method for latent-variable models, in which each datum has a corresponding, but unknown, latent representation. VAEs generalize autoencoders, adding stochasticity to the encoder which combined with a penalty term encourages all areas of the latent space to correspond to a valid decoding. The intuition is that adding noise to the encoded molecules forces the decoder to learn how to decode a wider variety of latent points and find more robust representations. Variational autoencoders with recurrent neural network encoding/decoding were proposed by Bowman et al. in the context of written English sentences, and we followed their approach closely.[20] To leverage the power of recent advances in sequence-to-sequence autoencoders for modeling text, we used the SMILES[28] representation, a commonly used text encoding for organic molecules. We also tested InChI[29] as an alternative string representation, but found it to perform substantially worse than SMILES, presumably due to a more complex syntax that includes counting and arithmetic.

The character-by-character nature of the SMILES representation and the fragility of its internal syntax (opening and closing cycles and branches, allowed valences, etc.) can still result in the output of invalid molecules from the decoder, even with the variational constraint. When converting a molecule from a latent representation to a molecule, the decoder model samples a string from the probability distribution over characters in each position generated by its final layer. As such, multiple SMILES strings are possible from a single latent space representation. We employed the open source cheminformatics suite RDKit[30] to validate the chemical structures of output molecules and discard invalid ones. While it would be more efficient to limit the autoencoder to generate only valid strings, this postprocessing step is lightweight and allows for greater flexibility in the autoencoder to learn the architecture of the SMILES.

To enable molecular design, the chemical structures encoded in the continuous representation of the autoencoder need to be correlated with the target properties that we are seeking to optimize. Therefore, we added a model to the autoencoder that predicts the properties from the latent space representation. This autoencoder was then trained jointly on the reconstruction task and a property prediction task; an additional multilayer perceptron (MLP) was used to predict the property from the latent vector of the encoded molecule. To propose promising new candidate molecules, we can start from the latent vector of an encoded molecule and then move in the direction most likely to improve the desired attribute. The resulting new candidate vectors can then be decoded into corresponding molecules (Figure 1b).

Two autoencoder systems were trained: one with 108 000 molecules from the QM9 data set of molecules with fewer than

DOI: 10.1021/acscentsci.7b00572
ACS Cent. Sci. XXXX, XXX, XXX−XXX

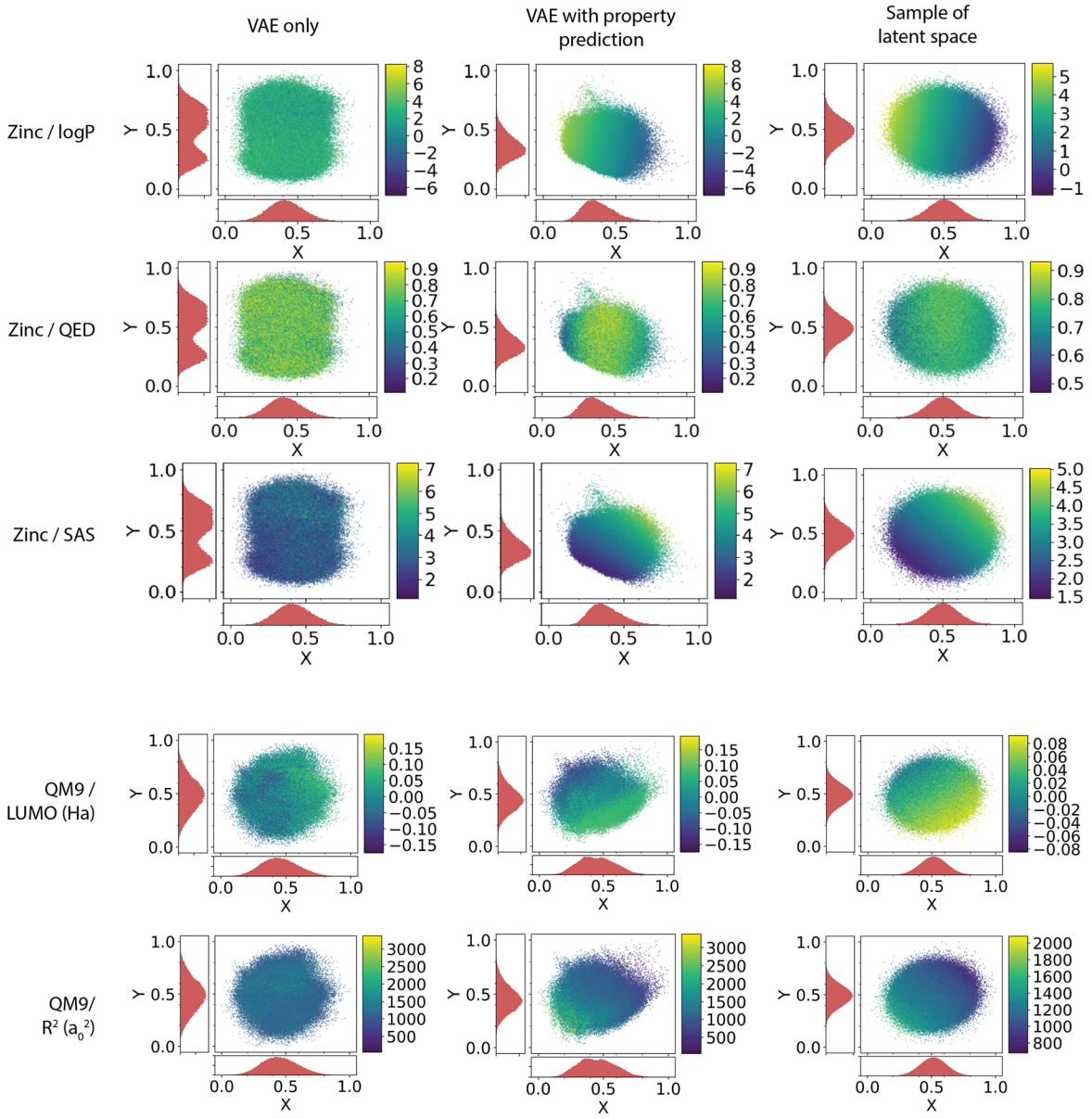


Figure 3. Two-dimensional PCA analysis of latent space for variational autoencoder. The two axis are the principle components selected from the PCA analysis; the color bar shows the value of the selected property. The first column shows the representation of all molecules from the listed data set using autoencoders trained without joint property prediction. The second column shows the representation of molecules using an autoencoder trained with joint property prediction. The third column shows a representation of random points in the latent space of the autoencoder trained with joint property prediction; the property values predicted for these points are predicted using the property predictor network. The first three rows show the results of training on molecules from the ZINC data set for the logP, QED, and SAS properties; the last two rows show the results of training on the QM9 data set for the LUMO energy and the electronic spatial extent (R^2).

9 heavy atoms[31] and another with 250 000 drug-like commercially available molecules extracted at random from the ZINC database.[32] We performed random optimization over hyperparameters specifying the deep autoencoder architecture and training, such as the choice between a recurrent or convolutional encoder, the number of hidden layers, layer sizes, regularization, and learning rates. The latent space representations for the QM9 and ZINC data sets had 156 dimensions and 196 dimensions, respectively.

RESULTS AND DISCUSSION

Representation of Molecules in Latent Space. First, we analyze the fidelity of the autoencoder and the ability of the latent space to capture structural molecular features.Figure 2a shows a kernel density estimate of each dimension when encoding a set of 5000 randomly selected ZINC molecules from outside the training set. The kernel density estimate shows the distribution of data points along each dimension of the latent space. Whereas the distribution of data point in each individual dimension shows a slightly different mean and standard deviation, all the distributions are normal as enforced by the variational regularizer.

The variational autoencoder is a doubly probabilistic model. In addition to the Gaussian noise added to the encoder, which can be turned off by simply sampling the mean of the encoding distribution, the decoding process is also nondeterministic, as the string output is sampled from the final layer of the decoder. This implies that decoding a single point in the latent space

DOI: 10.1021/acscentsci.7b00572
ACS Cent. Sci. XXXX, XXX, XXX−XXX

Table 1. Comparison of Molecule Generation Results to Original Datasets

source[a]	data set[b]	samples[c]	logP[d]	SAS[e]	QED[f]	% in ZINC[g]	% in emol[h]
Data	ZINC	249k	2.46 (1.43)	3.05 (0.83)	0.73 (0.14)	100	12.9
GA	ZINC	5303	2.84 (1.86)	3.80 (1.01)	0.57 (0.20)	6.5	4.8
VAE	ZINC	8728	2.67 (1.46)	3.18 (0.86)	0.70 (0.14)	5.8	7.0
Data	QM9	134k	0.30 (1.00)	4.25 (0.94)	0.48 (0.07)	0.0	8.6
GA	QM9	5470	0.96 (1.53)	4.47 (1.01)	0.53 (0.13)	0.018	3.8
VAE	QM9	2839	0.30 (0.97)	4.34 (0.98)	0.47 (0.08)	0.0	8.9

[a]Describes the source of the molecules: data refers to the original data set, GA refers to the genetic algorithm baseline, and VAE to our variational autoencoder trained without property prediction. [b]Shows the data set used, either ZINC or QM9. [c]Shows the number of samples generated for comparison, for data, this value simply reflects the size of the data set. Columns d−f show the mean and, in parentheses, the standard deviation of selected properties of the generated molecules and compares that to the mean and standard deviation of properties in the original data set. [d]Shows the water−octanal partition coefficient (logP).[36] [e]Shows the synthetic accessibility score (SAS).[37] [f]Shows the Qualitative Estimate of Drug-likeness (QED),[38] ranging from 0 to 1. We also examine how many of the molecules generated by each method are found in two major molecule databases: [g]ZINC; [h]E-molecules[39], and compare these values against the original data set.

back to a string representation is stochastic. Figure 2b shows the probability of decoding the latent representation of a sample FDA-approved drug molecule into several different molecules. For most latent points, a prominent molecule is decoded, and many other slight variations appear with lower frequencies. When these resulting SMILES are re-encoded into the latent space, the most frequent decoding also tends to be the one with the lowest Euclidean distance to the original point, indicating the latent space is indeed capturing features relevant to molecules.

Figure 2c shows some molecules in the latent space that are close to ibuprofen. These structures become less similar to increasing distance in the latent space. When the distance approaches the average distance of molecules in the training set, the changes are more pronounced, eventually resembling random molecules likely to be sampled from the training set. SI Figure 1d shows the distribution of distances in latent space between 50 000 random points from our ZINC training set. We estimate that we can find 30 such molecules in the locality of a molecule, i.e., 30 molecules closer to a given seed molecule from our data set than any other molecule in our data set. As such, we estimate that our autoencoder that was trained on 250 000 molecules from ZINC encodes approximately 7.5 million molecules. The probability of decoding from a point in latent space is dependent on how close this point is to the latent representations of other molecules; we observed a decoding rate of 73−79% for points that are close to known molecules, and 4% for randomly selected latent points.

A continuous latent space allows interpolation of molecules by following the shortest Euclidean path between their latent representations. When exploring high dimensional spaces, it is important to note that Euclidean distance might not map directly to notions of similarity of molecules.[33] In high dimensional spaces, most of the mass of independent normally distributed random variables is not near the mean, but in an annulus around the mean.[34] Interpolating linearly between two points might pass by an area of low probability, to keep the sampling on the areas of high probability we utilize spherical interpolation[35] (*slerp*). With *slerp*, the path between two points is a circular arc lying on the on the surface of a *N*-dimensional sphere. Figure 2d shows the spherical interpolation between two random drug molecules, showing smooth transitions in between. SI Figure 3 shows the difference between linear and spherical interpolation.

Table 1 compares the distribution of chemical properties in the training sets against molecules generated with a baseline genetic algorithm, and molecules generated from the variational autoencoder. In the genetic algorithm, molecules were generated with a list of hand-designed rules.[10−15] This process was seeded using 1000 random molecules from the ZINC data set and generated over 10 iterations. For molecules generated using the variational autoencoder, we collected the set of all molecules generated from 400 decoding attempts from the latent space points encoded from the same 1000 seed molecules. We compare the water−octanol partition coefficient (logP), the synthetic accessibility score (SAS),[37] and Quantitative Estimation of Drug-likeness (QED),[38] which ranges in value between 0 and 1, with higher values indicating that the molecule is more drug-like. SI3 Figure 2 shows histograms of the properties of the molecules generated by each of these approaches and compares them to the distribution of properties from the original data set. Despite the fact that the VAE is trained purely on the SMILES strings independently of chemical properties, it is able to generate realistic-looking molecules whose features follow the intrinsic distribution of the training data. The molecules generated using the VAE show chemical properties that are more similar to the original data set than the set of molecules generated by the genetic algorithm. The two rightmost columns in Table 1 report the fraction of molecules that belong to the 17 million drug-like compounds from which the training set was selected and how often they can be found in a library of existing organic compounds. In the case of drug-like molecules, the VAE generates molecules that follow the property distribution of the training data, but are new as the combinatorial space is extremely large and the training set is an arbitrary subsample. The hand-selected mutations are less able to generate new compounds while at the same time biasing the properties of the set to higher chemical complexity and decreased drug-likeness. In the case of the QM9 data set, since the combinatorial space is smaller, the training set has more coverage and the VAE generates essentially the same population statistics as the training data.

Property Prediction of Molecules. The interest in discovering new molecules and chemicals is most often in relation to maximizing some desirable property. For this reason, we extended the purely generative model to also predict property values from the latent representation. We trained a multilayer perceptron jointly with the autoencoder to predict properties from the latent representation of each molecule.

With joint training for property prediction, the distribution of molecules in the latent space is organized by property values. Figure 3 shows the mapping of property values to the latent

DOI: 10.1021/acscentsci.7b00572
ACS Cent. Sci. XXXX, XXX, XXX−XXX

Table 2. MAE Prediction Error for Properties Using Various Methods on the ZINC and QM9 Datasets

database/property	mean[a]	ECFP[b]	CM[b]	GC[b]	1-hot SMILES[c]	Encoder[d]	VAE[e]
ZINC250k/logP	1.14	0.38		0.05	0.16	0.13	0.15
ZINC250k/QED	0.112	0.045		0.017	0.041	0.037	0.054
QM9/HOMO, eV	0.44	0.20	0.16	0.12	0.12	0.13	0.16
QM9/LUMO, eV	1.05	0.20	0.16	0.15	0.11	0.14	0.16
QM9/Gap, eV	1.07	0.30	0.24	0.18	0.16	0.18	0.21

[a]Baseline, mean prediction. [b]As implemented in Deepchem benchmark (MoleculeNet),[40] ECFP-circular fingerprints, CM-coulomb matrix, GC-graph convolutions. [c]1-hot-encoding of SMILES used as input to property predictor. [d]The network trained without decoder loss. [e]Full variational autoencoder network trained for individual properties.

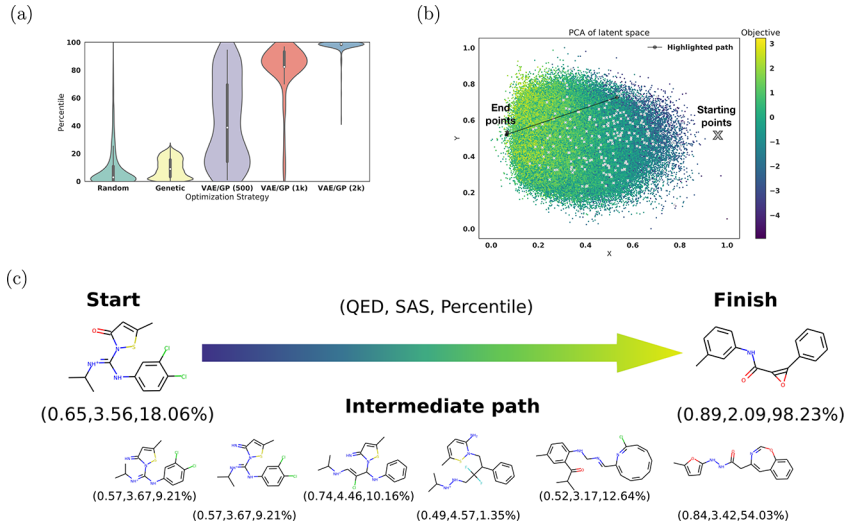


Figure 4. Optimization results for the jointly trained autoencoder using 5 × QED − SAS as the objective function. (a) shows a violin plot which compares the distribution of sampled molecules from normal random sampling, SMILES optimization via a common chemical transformation with a genetic algorithm, and from optimization on the trained Gaussian process model with varying amounts of training points. To offset differences in computational cost between the random search and the optimization on the Gaussian process model, the results of 400 iterations of random search were compared against the results of 200 iterations of optimization. This graph shows the combined results of four sets of trials. (b) shows the starting and ending points of several optimization runs on a PCA plot of latent space colored by the objective function. Highlighted in black is the path illustrated in part (c). (c) shows a spherical interpolation between the actual start and finish molecules using a constant step size. The QED, SAS, and percentile score are reported for each molecule.

space representation of molecules, compressed into two dimensions using PCA. The latent space generated by autoencoders jointly trained with the property prediction task shows in the distribution of molecules a gradient by property values; molecules with high values are located in one region, and molecules with low values are in another. Autoencoders that were trained without the property prediction task do not show a discernible pattern with respect to property values in the resulting latent representation distribution.

While the primary purpose of adding property prediction was to organize the latent space, it is interesting to observe how the property predictor model compares with other standard models for property prediction. For a more fair comparison against other methods, we increased the size of our perceptron to two layers of 1000 neurons. Table 2 compares the performance of commonly used molecular embeddings and models to the VAE. Our VAE model shows that property prediction performance for electronic properties (i.e., orbital energies) are similar to graph convolutions for some properties; prediction accuracy could be improved with further hyperparameter optimization.

Optimization of Molecules via Properties. We next optimized molecules in the latent space from the autoencoder which was jointly trained for property prediction. In order to create a smoother landscape to perform optimizations, we used a Gaussian process model to model the property predictor model. Gaussian processes can be used to predict any smooth continuous function[41] and are extremely lightweight, requiring only a few minutes to train on a dataset of a few thousand molecules. The Gaussian process was trained to predict target properties for molecules given the latent space representation of the molecules as an input.

The 2000 molecules used for training the Gaussian process were selected to be maximally diverse. Using this model, we optimized in the latent space to find a molecule that maximized our objective. As a baseline, we compared our optimization results against molecules found using a random Gaussian search and molecules optimized via a genetic algorithm.

The objective we chose to optimize was 5 × QED − SAS, where QED is the Quantitative Estimation of Drug-likeness,[38] and SAS is the Synthetic Accessibility score.[37] This objective represents a rough estimate of finding the most drug-like molecule that is also easy to synthesize. To provide the greatest challenge for our optimizer, we started with molecules from the ZINC data set that had an objective score in the bottom 10%, i.e., were in the 10th percentile.

From Figure 4a we can see that the optimization with the Gaussian process (GP) model on the latent space representation consistently results in molecules with a higher percentile

DOI: 10.1021/acscentsci.7b00572
ACS Cent. Sci. XXXX, XXX, XXX−XXX

score than the two baseline search methods. Figure 4b shows the path of one optimization from the starting molecule to the final molecule in the two-dimensional PCA representation, the final molecule ending up in the region of high objective value. Figure 4c shows molecules decoded along this optimization path using a Gaussian interpolation.

Performing this optimization on a GP model trained with 1000 molecules leads to a slightly wider range of molecules as shown in Figure 4a. Since the training set is smaller, the predictive power of the GP is lower which when optimizing in latent space and as a result optimizes to several local minima instead of a global optimization. In cases where it is difficult to define an objective that completely describes all the traits desired in a molecule, it may be better to use this localized optimization approach to reach a larger diversity of potential molecules.

CONCLUSION

We propose a new family of methods for exploring chemical space based on continuous encodings of molecules. These methods eliminate the need to hand-craft libraries of compounds and allow a new type of directed gradient-based search through chemical space. In our autoencoder model, we observed high fidelity in reconstruction of SMILES strings and the ability to capture characteristic features of a molecular training set. The autoencoder exhibited good predictive power when training jointly with a property prediction task, and the ability to perform gradient-based optimization of molecules in the resulting smoothed latent space.

There are several directions for further improvement of this approach to molecular design. In this work, we used a text-based molecular encoding, but using a graph-based autoencoder would have several advantages. Forcing the decoder to produce valid SMILES strings makes the learning problem unnecessarily hard since the decoder must also implicitly learn which strings are valid SMILES. An autoencoder that directly outputs molecular graphs is appealing since it could explicitly address issues of graph isomorphism and the problem of strings that do not correspond to valid molecular graphs. Building an encoder which takes in molecular graphs is straightforward through the use of off-the-shelf molecular fingerprinting methods, such as ECFP[23] or a continuously parametrized variant of ECFP such as neural molecular fingerprints.[24] However, building a neural network which can output arbitrary graphs is an open problem.

Further extensions of this work to use a explicitly defined grammar for SMILES instead of forcing the model to learn one[42] or to actively learn valid sequences[43,44] are underway, as is the application of adversarial networks for this task.[45−47] Several proceeding works have further explored the use of Long Short-Term Memory (LSTM) networks and recurrent networks applied to SMILES strings to generate new molecules[48,49] and predict the outcomes of organic chemistry reactions.[50]

The autoencoder sometimes produced molecules that are formally valid as graphs but contain moieties that are not desirable because of stability or synthetic constraints. Examples are acid chlorides, anhydrides, cyclopentadienes, aziridines, enamines, hemiaminals, enol ethers, cyclobutadiene, and cycloheptatriene. One option is to train the autoencoder to predict properties related to steric constraints of other structural constraints. In general, the objective function to be optimized needs to capture as many desirable traits as possible and balance them to ensure that the optimizer focuses on genuinely desirable compounds. This approach has also been tested in a few following works.[43,44]

The results reported in this work, and its application to optimizing objective functions of molecular properties, have already and will continue to influence new avenues for molecular design.

METHODS

Autoencoder Architecture. Strings of characters can be encoded into vectors using recurrent neural networks (RNNs). An encoder RNN can be paired with a decoder RNN to perform sequence-to-sequence learning.[51] We also experimented with convolutional networks for string encoding[52] and observed improved performance. This is explained by the presence of repetitive, translationally invariant substrings that correspond to chemical substructures, e.g., cycles and functional groups.

Our SMILES-based text encoding used a subset of 35 different characters for ZINC and 22 different characters for QM9. For ease of computation, we encoded strings up to a maximum length of 120 characters for ZINC and 34 characters for QM9, although in principle there is no hard limit to string length. Shorter strings were padded with spaces to this same length. We used only canonicalized SMILES for training to avoid dealing with equivalent SMILES representations. The structure of the VAE deep network was as follows: For the autoencoder used for the ZINC data set, the encoder used three 1D convolutional layers of filter sizes 9, 9, 10 and 9, 9, 11 convolution kernels, respectively, followed by one fully connected layer of width 196. The decoder fed into three layers of gated recurrent unit (GRU) networks[53] with hidden dimension of 488. For the model used for the QM9 data set, the encoder used three 1D convolutional layers of filter sizes 2, 2, 1 and 5, 5, 4 convolution kernels, respectively, followed by one fully connected layer of width 156. The three recurrent neural network layers each had a hidden dimension of 500 neurons.

The last layer of the RNN decoder defines a probability distribution over all possible characters at each position in the SMILES string. This means that the writeout operation is stochastic, and the same point in latent space may decode into different SMILES strings, depending on the random seed used to sample characters. The output GRU layer had one additional input, corresponding to the character sampled from the softmax output of the previous time step and was trained using teacher forcing.[54] This increased the accuracy of generated SMILES strings, which resulted in higher fractions of valid SMILES strings for latent points outside the training data, but also made training more difficult, since the decoder showed a tendency to ignore the (variational) encoding and rely solely on the input sequence. The variational loss was annealed according to sigmoid schedule after 29 epochs, running for a total 120 epochs.

For property prediction, two fully connected layers of 1000 neurons were used to predict properties from the latent representation, with a dropout rate of 0.20. To simply shape the latent space, a smaller perceptron of 3 layers of 67 neurons was used for the property predictor, trained with a dropout rate of 0.15. For the algorithm trained on the ZINC data set, the objective properties include logP, QED, SAS. For the algorithm trained on the QM9 data set, the objective properties include HOMO energies, LUMO energies, and the electronic spatial

DOI: 10.1021/acscentsci.7b00572
ACS Cent. Sci. XXXX, XXX, XXX−XXX

extent (R^2). The property prediction loss was annealed in at the same time as the variational loss. We used the Keras[55] and TensorFlow[56] packages to build and train this model and the RDKit package for cheminformatics.[30]

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acscentsci.7b00572.

Peripheral findings including statistics on the latent space, reconstruction accuracy, training robustness with respect to data set size, and more sampling interpolation examples (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: alan@aspuru.edu.

ORCID

Rafael Gómez-Bombarelli: 0000-0002-9495-8599
Jennifer N. Wei: 0000-0003-3567-9511
Dennis Sheberla: 0000-0002-5239-9151
Alán Aspuru-Guzik: 0000-0002-8277-4434

Author Contributions

#R.G.-B. J.N.W. J.M.H.-L. and D.D. contributed equally to this work.

Notes

The authors declare no competing financial interest.

The code and full training data sets are available at https://github.com/aspuru-guzik-group/chemical_vae.

■ ACKNOWLEDGMENTS

This work was supported financially by the Samsung Advanced Institute of Technology. The authors acknowledge the use of the Harvard FAS Odyssey Cluster and support from FAS Research Computing. J.N.W. acknowledges support from the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1144152. J.M.H.-L. acknowledges support from the Rafael del Pino Foundation. R.P.A. acknowledges support from the Alfred P. Sloan Foundation and NSF IIS-1421780. A.A.G. acknowledges support from The Department of Energy, Office of Basic Energy Sciences under Award DE-SC0015959. We thank Dr. Anders Frøseth for his generous support of this work.

■ REFERENCES

(1) Kim, S.; Thiessen, P. A.; Bolton, E. E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B. A.; Wang, J.; Yu, B.; Zhang, J.; Bryant, S. H. PubChem Substance and Compound databases. *Nucleic Acids Res.* **2016**, *44*, D1202−D1213.

(2) Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *J. Comput.-Aided Mol. Des.* **2013**, *27*, 675−679.

(3) Shoichet, B. K. Virtual screening of chemical libraries. *Nature* **2004**, *432*, 862−5.

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

(5) Cheng, T.; Li, Q.; Zhou, Z.; Wang, Y.; Bryant, S. H. Structure-Based Virtual Screening for Drug Discovery: a Problem-Centric Review. *AAPS J.* **2012**, *14*, 133−141.

(6) Pyzer-Knapp, E. O.; Suh, C.; Gómez-Bombarelli, R.; Aguilera-Iparraguirre, J.; Aspuru-Guzik, A. What Is High-Throughput Virtual Screening? A Perspective from Organic Materials Discovery. *Annu. Rev. Mater. Res.* **2015**, *45*, 195−216.

(7) Schneider, G. Virtual screening: an endless staircase? *Nat. Rev. Drug Discovery* **2010**, *9*, 273−276.

(8) Hachmann, J.; Olivares-Amaya, R.; Atahan-Evrenk, S.; Amador-Bedolla, C.; Sánchez-Carrera, R. S.; Gold-Parker, A.; Vogt, L.; Brockway, A. M.; Aspuru-Guzik, A. The Harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid. *J. Phys. Chem. Lett.* **2011**, *2*, 2241−2251.

(9) Gómez-Bombarelli, R.; et al. *Nat. Mater.* **2016**, *15*, 1120−1127.

(10) Virshup, A. M.; Contreras-García, J.; Wipf, P.; Yang, W.; Beratan, D. N. Stochastic Voyages into Uncharted Chemical Space Produce a Representative Library of All Possible Drug-Like Compounds. *J. Am. Chem. Soc.* **2013**, *135*, 7296−7303.

(11) Rupakheti, C.; Virshup, A.; Yang, W.; Beratan, D. N. Strategy To Discover Diverse Optimal Molecules in the Small Molecule Universe. *J. Chem. Inf. Model.* **2015**, *55*, 529−537.

(12) Reymond, J.-L. The Chemical Space Project. *Acc. Chem. Res.* **2015**, *48*, 722−730.

(13) Reymond, J.-L.; van Deursen, R.; Blum, L. C.; Ruddigkeit, L. Chemical space as a source for new drugs. *MedChemComm* **2010**, *1*, 30−38.

(14) Kanal, I. Y.; Owens, S. G.; Bechtel, J. S.; Hutchison, G. R. Efficient Computational Screening of Organic Polymer Photovoltaics. *J. Phys. Chem. Lett.* **2013**, *4*, 1613−1623.

(15) O'Boyle, N. M.; Campbell, C. M.; Hutchison, G. R. Computational Design and Selection of Optimal Organic Photovoltaic Materials. *J. Phys. Chem. C* **2011**, *115*, 16200−16210.

(16) van Deursen, R.; Reymond, J.-L. Chemical Space Travel. *ChemMedChem* **2007**, *2*, 636−640.

(17) Wang, M.; Hu, X.; Beratan, D. N.; Yang, W. Designing molecules by optimizing potentials. *J. Am. Chem. Soc.* **2006**, *128*, 3228−3232.

(18) Balamurugan, D.; Yang, W.; Beratan, D. N. Exploring chemical space with discrete, gradient, and hybrid optimization methods. *J. Chem. Phys.* **2008**, *129*, 174105.

(19) Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015; https://arxiv.org/abs/1511.06434.

(20) Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Jozefowicz, R.; Bengio, S. Generating Sentences from a Continuous Space, 2015; https://arxiv.org/abs/1511.06349.

(21) van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio, 2016; https://arxiv.org/abs/1609.03499.

(22) Engel, J.; Resnick, C.; Roberts, A.; Dieleman, S.; Eck, D.; Simonyan, K.; Norouzi, M. Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders, 2017; http://arxiv.org/abs/1704.01279.

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

(24) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Gómez-Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. *Adv. Neural Information Processing Syst.* **2015**, 2215−2223.

(25) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving Beyond Fingerprints, 2016; http://arxiv.org/abs/1603.00856.

(26) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **2012**, *108*, 058301.

(27) Kingma, D. P.; Welling, M. Auto-encoding Variational Bayes, 2013; https://arxiv.org/abs/1312.6114.

(28) Weininger, D. SMILES a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.* **1988**, *28*, 31−36.

DOI: 10.1021/acscentsci.7b00572
ACS Cent. Sci. XXXX, XXX, XXX−XXX

(29) Heller, S.; McNaught, A.; Stein, S.; Tchekhovskoi, D.; Pletnev, I. InChI - the worldwide chemical structure identifier standard. *J. Cheminf.* **2013**, *5*, 7.
(30) RDKit: Open-source cheminformatics; http://www.rdkit.org, [Online; accessed 11-April-2017].
(31) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 140022.
(32) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757−1768.
(33) Aggarwal, C. C.; Hinneburg, A.; Keim, D. A. *Database Theory − ICDT 2001:8th International Conference London, UK, January 4−6, 2001 Proceedings*; Springer: Berlin, Heidelberg, 2001; pp 420−434.
(34) Domingos, P. A few useful things to know about machine learning. *Commun. ACM* **2012**, *55*, 78.
(35) White, T. Sampling Generative Networks, 2016; http://arxiv.org/abs/1609.04468.
(36) Wildman, S. A.; Crippen, G. M. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 868−873.
(37) Ertl, P.; Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminf.* **2009**, *1*, 8.
(38) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **2012**, *4*, 90−98.
(39) E-molecules. https://www.emolecules.com/info/plus/download-database, [Online; accessed 22-July-2017].
(40) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning, 2017; https://arxiv.org/abs/1703.00564.
(41) Rasmussen, C. E.; Williams, C. K. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, 2006; Vol. *1*.
(42) Kusner, M. J.; Paige, B.; Hernández-Lobato, J. M. Grammar Variational Autoencoder, 2017; https://arxiv.org/abs/1703.01925.
(43) Janz, D.; van der Westhuizen, J.; Hernández-Lobato, J. M. Actively Learning what makes a Discrete Sequence Valid, 2017; http://arxiv.org/abs/1603.00856.
(44) Jaques, N.; Gu, S.; Bahdanau, D.; Hernández-Lobato, J. M.; Turner, R. E.; Eck, D. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. International Conference on Machine Learning, 2017; pp 1645−1654.
(45) Guimaraes, G. L.; Sanchez-Lengeling, B.; Farias, P. L. C.; Aspuru-Guzik, A. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. *arXiv:1705.10843*, 2017.
(46) Sanchez-Lengeling, B.; Outeiral, C.; Guimaraes, G. L.; Aspuru-Guzik, A. Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC), 2017; https://chemrxiv.org/articles/ORGANIC_1_pdf/5309668.
(47) Blaschke, T.; Olivecrona, M.; Engkvist, O.; Bajorath, J.; Chen, H. Application of generative autoencoder in de novo molecular design. *Mol. Inf.* **2017**, *36*, 1700123.
(48) Yang, X.; Zhang, J.; Yoshizoe, K.; Terayama, K.; Tsuda, K. ChemTS: an efficient python library for de novo molecular generation. *Sci. Technol. Adv. Mater.* **2017**, *18*, 972−976.
(49) Segler, M. H.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating focussed molecule libraries for drug discovery with recurrent neural networks, 2017; https://arxiv.org/abs/1701.01329.
(50) Liu, B.; Ramsundar, B.; Kawthekar, P.; Shi, J.; Gomes, J.; Luu Nguyen, Q.; Ho, S.; Sloane, J.; Wender, P.; Pande, V. Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models. *ACS Cent. Sci.* **2017**, *3*, 1103−1113.
(51) Sutskever, I.; Vinyals, O.; Le, Q. V. Sequence to sequence learning with neural networks. *Adv. Neural Information Processing Syst.* **2014**, 3104−3112.
(52) Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A Convolutional Neural Network for Modelling Sentences, 2014; https://arxiv.org/abs/1404.2188.
(53) Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, 2014; http://arxiv.org/abs/1412.3555.
(54) Williams, R. J.; Zipser, D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.* **1989**, *1*, 270−280.
(55) Chollet, F. K. https://github.com/fchollet/keras, 2015.
(56) Abadi, M. et al. TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016; pp 265−283.

DOI: 10.1021/acscentsci.7b00572
ACS Cent. Sci. XXXX, XXX, XXX−XXX