JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

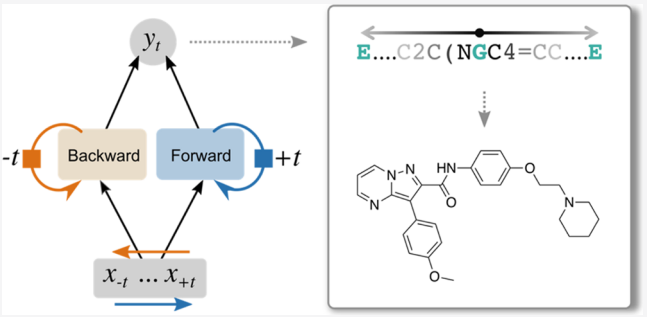
Bidirectional Molecule Generation with Recurrent Neural Networks

Francesca Grisoni,* Michael Moret, Robin Lingwood, and Gisbert Schneider*

Cite This: J. Chem. Inf. Model. 2020, 60, 1175−1183 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Recurrent neural networks (RNNs) are able to generate de novo molecular designs using simplified molecular input line entry systems (SMILES) string representations of the chemical structure. RNN-based structure generation is usually performed unidirectionally, by growing SMILES strings from left to right. However, there is no natural start or end of a small molecule, and SMILES strings are intrinsically nonunivocal representations of molecular graphs. These properties motivate bidirectional structure generation. Here, bidirectional generative RNNs for SMILES-based molecule design are introduced. To this end, two established bidirectional methods were implemented, and a new method for SMILES string generation and data augmentation is introduced—the bidirectional molecule design by alternate learning (BIMODAL). These three bidirectional strategies were compared to the unidirectional forward RNN approach for SMILES string generation, in terms of the (i) novelty, (ii) scaffold diversity, and (iii) chemical−biological relevance of the computer-generated molecules. The results positively advocate bidirectional strategies for SMILES-based molecular de novo design, with BIMODAL showing superior results to the unidirectional forward RNN for most of the criteria in the tested conditions. The code of the methods and the pretrained models can be found at URL https://github.com/ETHmodlab/BIMODAL.



INTRODUCTION

The chemical space of small organic molecules is estimated to contain 10^{60} to 10^{100} chemical structures.[1,2] Designing molecules with desired properties from scratch confronts chemists with a complex multivariate optimization task. Computational approaches have proved valuable to generate novel molecules,[3] for example, by extensive structure enumeration,[4,5] inversion of quantitative structure−activity relationship models,[6−8] evolutionary algorithms,[9,10] or rule-based design.[11,12] Most of these methods rely on a priori knowledge, for example, structure−activity relationships, aggregation rules, chemical transformation rules, fitness functions, and/or design constraints. Recently, generative deep learning methods—e.g., recurrent neural networks (RNNs),[13,14] adversarial autoencoders[15]—have emerged as potential alternatives to rule-based de novo molecular design methods.[16−21] Many of these generative machine learning methods build on text representations of molecules, such as simplified molecular input line entry systems (SMILES,[22] Figure 1a) strings, and directly sample new chemical entities without the need of explicit design rules, structure−activity relationship models, or molecular descriptors.

RNNs, in particular, have been applied to computational molecule generation.[21,23−26] In a recent benchmark study,[27] SMILES-based RNN with long-short term memory (LSTM) cells[28] resulted the best generative method among a selection of evolutionary, rule-based, and sequence-based methods (i.e.,

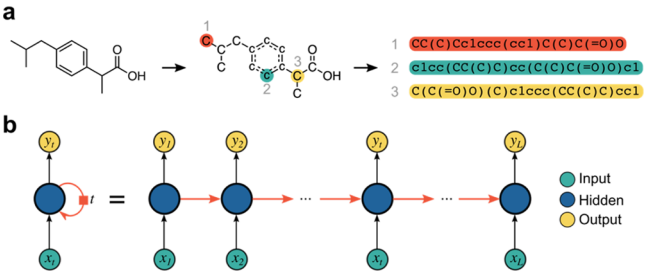


Figure 1. Overview of the basic concepts of this study. (a) SMILES strings, obtained from a molecular graph representation, where each atom is indicated by its element symbol, while branching and connectivity are indicated by symbols or lowercase letters (e.g., "( )", "=" and "c" for branching, double bonds, and aromatic carbons, respectively). Examples of three SMILES strings representing the drug ibuprofen are shown; the start atoms used for SMILES string production are indicated by gray numbers. (b) Simplified scheme of a forward RNN with one recurrent neuron layer. RNNs model a dynamic system, in which the network state at any t-th time point depends both on the current observation (x_t) and on the previous state (at $t - 1$) and is used to predict the output (y_t).

Received: October 9, 2019
Published: January 6, 2020

https://dx.doi.org/10.1021/acs.jcim.9b00943
J. Chem. Inf. Model. 2020, 60, 1175−1183

generative adversarial networks with reinforcement learning,[20] and adversarial autoencoders[19]), in the task of matching the training data distribution of structural and biological features. Moreover, RNNs proved applicable to generating new nuclear receptor modulators and bioactive natural-product inspired molecules in prospective studies.[29,30]

With sequences of SMILES characters ("tokens") as input, RNN models learn to predict one token at a time, based on the preceding portion of the sequence and a probability estimation (Figure 1b). From the learnt probability distribution, new SMILES strings can be sampled. RNNs are usually trained to read and generate SMILES strings in a "forward" manner, that is, from the left to the right. However, SMILES representations can be generated starting from any nonhydrogen atom and by proceeding in any direction (Figure 1a).[22] Different from natural languages, there is no uniquely defined start and end of a small molecule. The nonunivocity and nondirectionality offers the opportunity to explore bidirectional sequence generation, that is, methods that read and generate SMILES strings in both forward and backward directions. However, to the best of our knowledge, purely de novo ("end-to-end") bidirectional SMILES generation has not been explored to date.

The aim of the present study is to implement and evaluate bidirectional RNNs as SMILES string generators. To this end, two literature-based methods (synchronous forward backward RNN [FB-RNN],[31] neural autoregressive distribution estimator [NADE][32]), and the here newly developed method (bidirectional molecule design by alternate learning [BIMODAL]) were implemented and compared with a conventional forward RNN approach for SMILES string generation. The methods were evaluated with regard to their ability to construct chemically relevant molecules, in terms of the novelty of their scaffolds and selected biological and physicochemical properties. The results of this study highlight the advantages of bidirectional sequence generation methods and advocate their usage for prospective de novo molecular design.

THEORY

Forward RNNs. RNNs[33] can deal with sequence data as both input and output. Given an input vector at time step t, $\mathbf{x}(t) = \{x_1, x_2, ..., x_t\}$, and the corresponding output vector $\mathbf{y}(t) = \{y_1, y_2, ..., y_t\}$, RNNs aim to model the distribution $P(y_t|x_1, x_2, ..., x_t)$. RNNs model a dynamic system (Figure 1b), where the network state at any t-th time point (i.e., at any t-th position in the SMILES sequence) depends both on the current observation (i.e., the token x_t) and on the previous state of the system at $t-1$. Given an input sequence, generative RNNs are trained to extend this sequence by predicting the next sequence token, defined as $y_t = x_{t+1}$.[32] In this work, we use RNNs with LSTM cells,[28] which were proposed to solve the vanishing and exploding gradient problems caused by long sequences and a large network architecture.[34] At any given t-th time step, such a network is described by the following set of equations (eq 1)

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ g_t \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned} \tag{1}
$$

where h_t is the hidden state at time t, c_t is the memory cell state at time t, x_t is the input at time t, and h_{t-1} is the hidden state at the previous time step (or the initial hidden state at time $t = 0$); i_t, f_t, o_t are the input, forget, and output gates, respectively. The symbols σ and $\circ$ indicate the sigmoid function and the Hadamard product, respectively; W and b indicate the set of weights and bias values estimated by backpropagation-through-time gradient descent.[35] The memory cell (c_t) encodes the information of the inputs that have been observed until time t. The input gate (i_t) controls the extent to which the new input influences the cell state, the forget gate (f_t) controls what to keep from the previous cell state, and the output gate controls the extent to which the updated cell state is used to compute the new hidden state value (h_t). In this way, important sequence features can be carried along over long time spans, thereby capturing long-distance dependencies.[36] The hidden state value of the LSTM layer (or the last layer of multilayered LSTM networks, respectively) is the output of the RNN (y_t).

The most common version of RNNs for sequence generation proceeds from left to right (forward direction), that is, from $t = 1$ to $t = L$, where L is the length of the SMILES sequence. During training, the first position of the input is filled with a start-of-sequence token, and the last position of the input is filled with an end-of-sequence token. Once the RNN model is trained, the new sequences are generated by (i) inputting the starting token ("G"), and (ii) allowing the model to progressively select the next token, given the respective previous sequence of tokens, until the end token ("E") is generated (Figure 2a). At each step t, the probability for each k-th token to follow the preceding portion of the generated string is computed with a *softmax* function (eq 2)

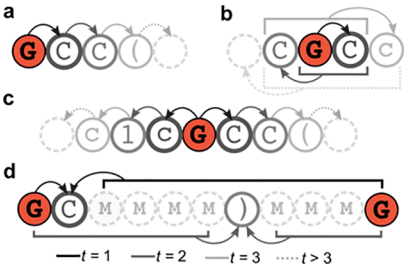


Figure 2. RNN-based methods for SMILES string generation. SMILES generation begins from the start token "G" and proceeds in a predetermined direction. (a) Forward RNN: From the start token "G", new tokens are added from left to right. (b) BIMODAL approach: tokens are generated in alternating directions at each time step (t). The model uses the whole sequence (forward and backward) to generate the next token. (c) Forward–backward model: Starting from the "G" token, two tokens are predicted per time step, one on each side. (d) NADE approach: missing "dummy" tokens ("M") are replaced with valid SMILES characters either toward the center of the string or in a random fashion.

https://dx.doi.org/10.1021/acs.jcim.9b00943
J. Chem. Inf. Model. 2020, 60, 1175−1183

$$P(x_{t+1} = k|x_1, \ldots, x_t) = \frac{\exp(y_t^k/T)}{\sum_{i=1}^{K}\exp(y_t^i/T)} \quad (2)$$

where y_t^k is the model output (logits) for the k-th token at time t (eq 1) and i runs over the set K of all tokens. The sampling of tokens is controlled by temperature parameter T. For high temperatures ($T \to \infty$), all tokens have nearly the same probability; the lower the temperature, the more the predicted y_t^k influences the probability of the k-th token. For $T \to 0$, the probability of the token with the highest y_t^k approximates 1. In this present work, "G" and "E" were used as start and end tokens of SMILES strings, respectively. In what follows, this method will be referred to as "forward RNN".

Bidirectional Methods. Bidirectional RNNs are usually composed of two unidirectional RNNs to allow for forward and backward prediction simultaneously.[37] Bidirectional RNNs are employed to solve supervised tasks, for example, regression and classification,[38,39] and as sequence encoders/decoders to/from fixed-length vectors.[39−43] However, using RNNs for bidirectional string generation is nontrivial, mostly due to the lack of "past" and "future" context information and the difficulty to combine the computed probabilities.[32] Only few RNN-based approaches have been explored for bidirectional string generation from a starting token.[31,32] In what follows, after describing two bidirectional methods borrowed from the field of natural language processing (synchronous FB-RNN[31] and NADE[32]), we introduce the new BIMODAL method.

Synchronous FB-RNN. The synchronous FB-RNN[31] was developed to predict previous and future words, given an arbitrary starting word within a sentence. We adapted this approach to generate preceding and subsequent portions of SMILES strings, starting from the token at an arbitrary position in the string. In particular, starting from token x_m, the FB-RNN model estimates the two conditional probability distributions in the forward and backward direction as follows (eq 3)

$$\vec{P}(x_{m+t+1} = k|x_{m-t}, \ldots, x_{m+t}) = \frac{\exp(\vec{y}_{m+t}^k/T)}{\sum_{i=1}^{K}\exp(\vec{y}_{m+t}^i/T)}$$

$$\overleftarrow{P}(x_{m-t-1} = k|x_{m-t}, \ldots, x_{m+t}) = \frac{\exp(\overleftarrow{y}_{m-t}^k/T)}{\sum_{i=1}^{K}\exp(\overleftarrow{y}_{m-t}^i/T)} \quad (3)$$

where $\vec{y}_{m+t}^k$ and $\overleftarrow{y}_{m-t}^k$ are the FB-RNN outputs (eq 1) for the k-th token at the t-th time interval in the forward and backward direction, respectively; T is the sampling temperature and i runs over the set K of all tokens. The estimated conditional probabilities are independent. Starting from token x_m, the FB-RNN simultaneously predicts x_{m+t} and x_{m-t} for any t-th time interval by elongating the sequence in the forward and backward direction, respectively (Figure 2c). In our implementation, SMILES string generation starts from the start token ("G") and proceeds in both directions until the end token ("E") is generated on both sides.

Neural Autoregressive Distribution Estimator. The NADE[32] was originally proposed to reconstruct missing values in sequences. Given a sequence of L tokens ($\mathbf{x} = \{x_1, x_2, ..., x_L\}$) with one missing token in the t-th position (x_t), the NADE model aims to reconstruct the gap by reading the preceding and next parts of the sequence in a forward and backward direction, respectively, and uses this information to replace the missing token. The conditional probability is estimated as follows (eq 4)

$$P(x_t = k|\{x_d\}_{\forall d \neq t}) = \frac{\exp(\hat{y}_t^k/T)}{\sum_{i=1}^{K}\exp(\hat{y}_t^i/T)}$$
$$\text{where } \hat{y}_t^i = \overleftarrow{y}_t^i + \vec{y}_t^i \quad (4)$$

where $\overleftarrow{y}_t^i$ is the prediction for the i-th token at the t-th time interval output by the backward RNN (from x_L to x_{t+1}), while $\vec{y}_t^i$ is the prediction of the forward RNN, which reads the sequence from x_1 to x_{t-1}. The conditional probability is estimated by using both forward and backward information of the sequence. For this present work, NADE was adapted to generating SMILES strings. In particular, a string of tokens representing a missing value ("dummy" token, "M") was used as the starting sequence, and the model was used to replace one "M" token at a time with valid SMILES tokens, either in a predefined or random order until the sequence has no more missing values (Figure 2d).

Bidirectional Molecule Design by Alternate Learning. The new BIMODAL algorithm is inspired by bidirectional RNNs for regression and classification[37] and combines features of both NADE and FB-RNN models. Like NADE, at any t-th time step, BIMODAL reads $\mathbf{x} = \{x_m, x_{m+1}, ..., x_t\}$ along the forward ($x_m \to x_t$) and backward ($x_t \leftarrow x_m$) direction; like FB-RNN, the SMILES sequence is generated in both directions. However, only one token per step is predicted alternatively on each side by using the left-to-right (forward) and right-to-left (backward) information simultaneously (Figure 2b). BIMODAL consists of two RNNs, one for reading the sequence in each direction (forward and backward), which are then combined to provide a joint prediction (y_t) (eq 5)

$$y_t = \vec{W}_{\text{hy}}\vec{h}_t + \overleftarrow{W}_{\text{hy}}\overleftarrow{h}_t + b_{\text{hy}} \quad (5)$$

where h_t is the hidden state, W_{hy} is the hidden-to-output weight matrix, and b_{hy} is bias vector, respectively (eq 1). In eq 5, the arrows indicate the network direction used for the token estimation, that is, forward ($\rightarrow$) and backward ($\leftarrow$).

In the SMILES generation setup, the BIMODAL reads the sequence in both forward and backward directions at each time step t (Figure 2b). Then, it generates a new token in either the forward direction ($x_{m+(t-1)/2+1}$, for odd t values) or in the backward direction ($x_{m-t/2-1}$ for even values of t, eq 6)

$$P(x_{m+t'+1} = k|x_{m-t'}, \ldots, x_{m+t'}) = \frac{\exp(y_{m+t'}^k/T)}{\sum_{i=1}^{K}\exp(y_{m+t'}^i/T)}$$
$$\text{where } t' = \frac{t-1}{2} \text{ (odd } t \text{ values)}$$

$$P(x_{m-t^*-1} = k|x_{m-t^*}, \ldots, x_{m+t^*}) = \frac{\exp(y_{m-t^*}^k/T)}{\sum_{i=1}^{K}\exp(y_{m-t^*}^i/T)}$$
$$\text{where } t^* = \frac{t}{2} \text{(even } t \text{ values)} \quad (6)$$

where $y_{m+t'}^k$ and $y_{m+t^*}^k$ are the model output for the k-th token at the considered time step ($m - t'$ and $m + t^*$, respectively), computed by combining backward and forward information according to eq 5. String generation starts from the "G" token and proceeds until the end token ("E") is produced in both directions.

https://dx.doi.org/10.1021/acs.jcim.9b00943
J. Chem. Inf. Model. 2020, 60, 1175−1183

MATERIALS AND METHODS

Data and Pre-treatment. All models were trained on canonical SMILES representations of 271,914 bioactive compounds. This training set was compiled from the ChEMBL22[44] database by retaining compounds with annotated $K_{d/I}/IC_{50}/EC_{50} < 1\ \mu M$, removing salts and stereochemical information. Pursuant to our earlier work,[24] nucleic acids and peptides were removed, SMILES strings with length between 34 and 74 tokens were retained and canonicalized[45] using RDKit (v. 2018.09.2.0)[46] prior to model training.

Model Implementation and Training. *Model Architecture.* For forward RNN, NADE, and FB-RNN, the network models consisted of five layers (BatchNormalization, LSTM layer 1, LSTM layer 2, BatchNormalization, linear). LSTM layers with either 256 or 512 hidden units were implemented and tested (corresponding to a total of 512 or 1024 hidden units, respectively). The BIMODAL network was composed of seven layers (BatchNormalization, LSTM layer 1—forward, LSTM layer 1—backward, LSTM layer 2—forward, LSTM layer 2—backward, BatchNormalization, linear). LSTM layers with 128 and 215 hidden units were tested (corresponding to a total of 512 and 1024 hidden units, respectively). A dropout value of 0.3 was used for the output weights in LSTM layer 1. Details can be found in Tables S1 and S2.

Model Training. Models were trained with the Adam optimization algorithm,[47] using cross-entropy loss (L) for performance optimization (eq S1). The cross-entropy loss was computed based on fivefold cross-validation (random partitioning protocol) and fitting (i.e., by using all of the training data). Models were trained for 10 epochs (one epoch = one pass of all of the data points through the network once).

Sequence Sampling. SMILES strings were sampled at a sampling temperature of $T = 0.7$, following a procedure for unidirectional RNN sampling from a previous study.[48] For NADE, a sequence of 74 "dummy" missing-value tokens ("M") was generated for further replacement.

Model Evaluation. *Evaluation Criteria.* De novo drug design is a multiparameter optimization problem.[49] As such, we set out to evaluate several properties and structural characteristics of the molecules produced by each generative method:

1. Structural uniqueness, validity, and novelty of the generated molecular graphs, these aspects constitute the primary design goal of any generative model, that is, ensuring that nonredundant, chemically valid, and novel molecules are generated. For the purpose of this study, structural novelty was defined as "not contained in the training set". We evaluated the models based on: (i) uniqueness, calculated as the percentage of unique SMILES strings generated (after canonicalization), (ii) validity, calculated as the percentage of chemically valid SMILES strings generated, (iii) novelty, calculated as the percentage of unique and valid molecules that were not included in the training set. The goal was to obtain models that exclusively generate unique, valid, and novel SMILES strings.
2. Scaffold diversity and novelty. Exploring novel and diverse scaffolds offers a choice in terms of chemical accessibility and prospects for lead optimization.[50] Thus, we evaluated the generative models for their ability to (i) generate a set of molecules with diverse scaffolds and (ii) produce novel scaffolds (defined in this study as "not present in the training set"). Bemis–Murcko scaffolds[51] were used for analyzing the scaffold diversity (ScaffDiv) and scaffold novelty (ScaffNov), defined as

$$\text{ScaffDiv} = n_{\text{scaff}}/n \times 100,$$

$$\text{ScaffNov} = n^{*}_{\text{scaff}}/n \times 100 \quad (7)$$

where n is the number of molecules, n_{scaff} is the number of unique scaffolds, and n^{*}_{scaff} is the number of unique scaffolds not present in the training set.
3. Biological and chemical relevance. Generative models should be able to sample molecules with desired chemical and biological properties, usually resembling the properties of the training set. Here, we compared the property distribution of the designs with the training molecules. This was performed by computing the Fréchet ChemNet distance (FCD).[52] FCD values are calculated on the activation of the penultimate layer of an LSTM model[53] trained to predict bioactivities. The obtained representation thus contains chemical and biological information about the molecular structures because it is obtained by considering structural (input layer) and biological information (output layer).[52] The lower the FCD between two sets, the closer they are in terms of structural and biological properties.[52]

Each model was trained on a set of 271,914 bioactive molecules from ChEMBL22 for 10 epochs, where the cross-validation loss converged for all cases.

Considered Model Variants. For bidirectional model training, the starting token can be, in principle, positioned at any position of the SMILES string. We tested two versions of each bidirectional method, based on the position of the starting token ("G") during model training:

1. Fixed starting position within the molecule. For BIMODAL and FB-RNN, the starting token was fixed in the center of each SMILES string during training (Figure 3a). For NADE, the generation was performed alternatively from the terminal left and terminal right

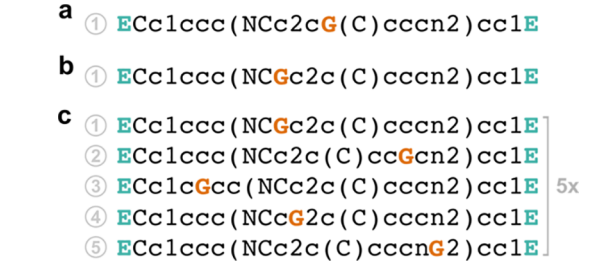


Figure 3. Schematic representation of starting position variants (fixed and random) and novel augmentation approach. (a) Fixed starting position within the SMILES string for FB-RNN and BIMODAL. During training, the starting token ("G") is always fixed in the center of each SMILES string. For NADE, the generation is performed alternatively from the terminal left and terminal right toward the center of the molecule. (b) Random starting position for FB-RNN and BIMODAL. The starting token ("G") is randomly placed within the SMILES string during training. For NADE, the replacement of dummy tokens is performed at random positions. (c) Augmentation (fivefold) for FB-RNN and BIMODAL. For each training molecule, n repetitions of the same SMILES string (here, $n = 5$) are produced, each containing the start token in a different (random) position. For NADE, the replacement of missing tokens starts in a different random position for each repeated SMILES string.

https://dx.doi.org/10.1021/acs.jcim.9b00943
J. Chem. Inf. Model. 2020, 60, 1175−1183

Table 1. Uniqueness, Validity, and Novelty Values (Mean ± Std. Dev.) after 10 Training Epochs (100 SMILES Sampled for Each Five-Fold Cross-Validation Run)[a]

model	starting point	no. hidden	data augmentation	unique (%)	valid (%)	novel (%)
forward	fixed	512		100 ± 0	93 ± 2	89 ± 2
BIMODAL	fixed	512		99.8 ± 0.5	79 ± 4	79 ± 4
FB-RNN	fixed	512		99.4 ± 0.8	51 ± 6	51 ± 6
NADE	fixed	512		100 ± 0	19 ± 4	18 ± 4
BIMODAL	random	512		100 ± 0	89 ± 3	89 ± 3
FB-RNN	random	512		100 ± 0	60 ± 8	60 ± 8
NADE	random	512		100 ± 0	7 ± 1	7 ± 1
forward	fixed	1024		100 ± 0	**95±1**	77 ± 4
BIMODAL	fixed	1024		99.4 ± 0.8	84 ± 4	81 ± 4
FB-RNN	fixed	1024		100 ± 0	53 ± 3	52 ± 3
NADE	fixed	1024		100 ± 0	21 ± 2	21 ± 2
BIMODAL	random	1024		100 ± 0	89 ± 5	88 ± 4
FB-RNN	random	1024		100 ± 0	63 ± 2	62 ± 2
NADE	random	1024		100 ± 0	6 ± 3	6 ± 3
BIMODAL	random	512	5×	100 ± 0	91 ± 1	90 ± 1
BIMODAL	random	512	10×	100 ± 0	94 ± 2	**94±1**
FB-RNN	random	512	5×	100 ± 0	63 ± 4	62 ± 4
FB-RNN	random	512	10×	100 ± 0	64 ± 4	64 ± 4
NADE	random	512	5×	100 ± 0	5 ± 2	5 ± 2
NADE	random	512	10×	100 ± 0	6 ± 3	6 ± 3

[a]The highest value for each considered metric is highlighted in boldface.

dummy tokens toward the center of the molecule. The fixed-start condition is comparable to the forward RNN, in which the starting token is always placed in the same position (at the beginning of the sequence).

2. Random starting position within the molecule, that is, by randomly placing the starting token within the SMILES string during training (Figure 3b). For NADE, the "randomness" was incorporated by performing the replacement of dummy tokens ("M") iteratively in random positions in the sequence. The random positioning of the start token may be considered an "unbiased" training procedure because this approach does not introduce any information about the length of a molecule nor does it enforce a symmetric positioning of the end tokens.

Additionally, two network sizes (512 and 1024 hidden units in total) were evaluated.

Data Augmentation. Data augmentation refers to the artificial incrementation of the training data volume, aimed to increase the model performance.[54] The possibility of placing the start token in an arbitrary position of the string during training allows performing a novel type of data augmentation. In particular, for each training molecule, we generated n repetitions of the same canonicalized SMILES string (n = 5 and n = 10); for each of these repetitions, the start token was placed in a different random position and used for training (Figure 3c). For NADE, the augmentation was performed by starting the replacement of "M" tokens in a different random position for each repeated SMILES string. This type of augmentation differs from the approach proposed by Bjerrum[55]—which uses multiple SMILES strings of each molecule—and it is, to the best of our knowledge, introduced for the first time in this present study.

Software and Code. All methods were implemented in Python (v3.6.8), using PyTorch (v1.0.0, https://pytorch.org/). Models were trained on trained on a NVIDIA GeForce GTX 1080 Ti—256 GB (NVIDIA Corporation, Santa Clara, California, United States). The FCD[52] index was computed using the code made available at URL https://github.com/bioinf-jku/FCD. Statistical tests were performed with MATLAB (vR2018a, The MathWorks, Inc., Natick, Massachusetts, United States). Scaffolds were calculated with RDKit[46] (v2019.03.1).

Code Availability. The code of the implemented methods and the pretrained models can be found in the GitHub repository at URL https://github.com/ETHmodlab/BIMODAL.

RESULTS AND DISCUSSION

Uniqueness, Validity, and Novelty. 500 SMILES strings (100 for each cross-validation run) were sampled with each method, and their uniqueness, validity, and novelty were calculated. We evaluated the effect of (i) the position of the starting token during the training (fixed vs random), (ii) the network size (512 vs 1024 hidden units), and (iii) data augmentation (5- and 10-fold augmentation).

Effect of Starting-Token Position. With 512 hidden units, all methods generated more than 99% unique molecules, and the forward RNN reached the highest number of valid and novel molecules (Table 1). In the case of fixed starting point, except for BIMODAL (79% valid and novel molecules), bidirectional generation methods performed only moderately, producing less than 52% valid and novel molecules. Using random starting points boosted the capability of two of the three bidirectional methods to generate valid and novel SMILES, with a 9-10% increase for the BIMODAL and FB-RNN (Figure 4). In the case of NADE, decreased validity and novelty were observed, potentially owing to the token replacement at random positions, which led to issues related to SMILES string correctness, for example, incorrect ring closure. With random starting points, the BIMODAL and forward RNN methods yielded similar novelty values.

Effect of Network Size. Most of the bidirectional methods did not reach the performance of the forward RNN with 512

https://dx.doi.org/10.1021/acs.jcim.9b00943
J. Chem. Inf. Model. 2020, 60, 1175−1183

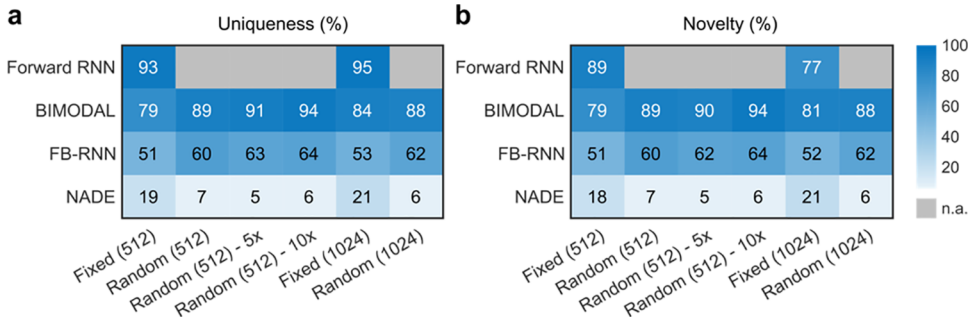


Figure 4. (a) Uniqueness and (b) novelty values (%) obtained by each investigated method (n.a. = not available). The generative models were tested with fixed and random starting points, two levels of augmentation (5- and 10-fold), and two network sizes (512 and 1024 hidden units).

Table 2. Scaffold Diversity and Novelty of the Generated Molecules (Eq 7)[a]

			scaffold diversity [%]		scaffold novelty [%]	
model	starting point	data augmentation	n_h = 512	n_h = 1024	n_h = 512	n_h = 1024
forward	fixed		75	72	64	55
BIMODAL	fixed		79	81	72	71
BIMODAL	random		79	79	72	70
BIMODAL	random	5×	80	79	72	68
FB-RNN	fixed		81	84	75	76
FB-RNN	random		81	88	75	77
FB-RNN	random	5×	84	89	77	75
NADE	fixed		72	73	67	68
NADE	random		88	90	85	86

[a]Different starting points (fixed and random starting point), data augmentation levels, and network sizes were tested (n_h = number of hidden units).

hidden units. Thus, we tested the effect of an increased net size (1024 hidden units) (Table 1). The increased network size had a positive effect on the uniqueness and/or novelty values of all of the tested bidirectional methods, with the exception of BIMODAL with random starting points. With 1024 hidden units, the forward RNN reproduced a higher percentage of the training data compared to the version with 512 hidden units.

Effect of Data Augmentation. To investigate the effect of the novel data augmentation strategy, we tested two augmentation levels (five- and ten-fold). With the exception of NADE, data augmentation led to increased uniqueness and novelty of the generated molecules (Table 1). BIMODAL showed the largest increment of performance, reaching 94% novel molecules with 10-fold augmentation, outperforming the forward RNN (89%).

Scaffold Diversity and Novelty. For each model, 30,000 unique and novel SMILES were sampled (5000 molecules generated in six independent sampling rounds) and analyzed for (i) scaffold diversity, that is, the percentage of unique scaffolds, and (ii) scaffold novelty, that is, the percentage of unique scaffolds that are not present in the training molecules (eq 7, Table 2). Five-fold data augmentation was tested because it resulted in the best compromise between the training data volume and compound novelty; NADE with data augmentation was not considered, due to the low validity and novelty of the generated molecules (Figure 4).

Except for NADE with fixed starting point, the bidirectional methods outperformed the forward RNN. It seems that the degree of scaffold novelty of the generated molecules is determined by the chosen method rather than by data augmentation, network size, or starting point position (Figure 5).

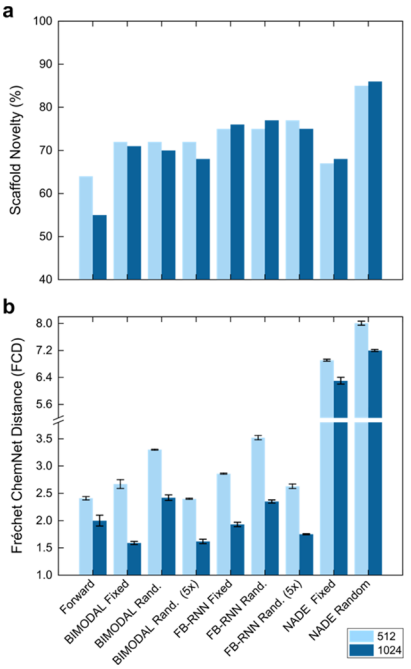


Figure 5. Comparison between generative methods in terms of (a) novelty of the scaffolds (compared to the training set scaffolds—the higher, the better) and (b) FCD to the training set molecules (the smaller, the better). Colors represent the net size (512 and 1024 total hidden neurons, corresponding to light blue and dark blue, respectively).

NADE with random replacement yielded the highest values of the investigated metrics. However, many of the scaffolds generated by NADE possess eight-membered rings (Figure 6),

https://dx.doi.org/10.1021/acs.jcim.9b00943
J. Chem. Inf. Model. 2020, 60, 1175−1183

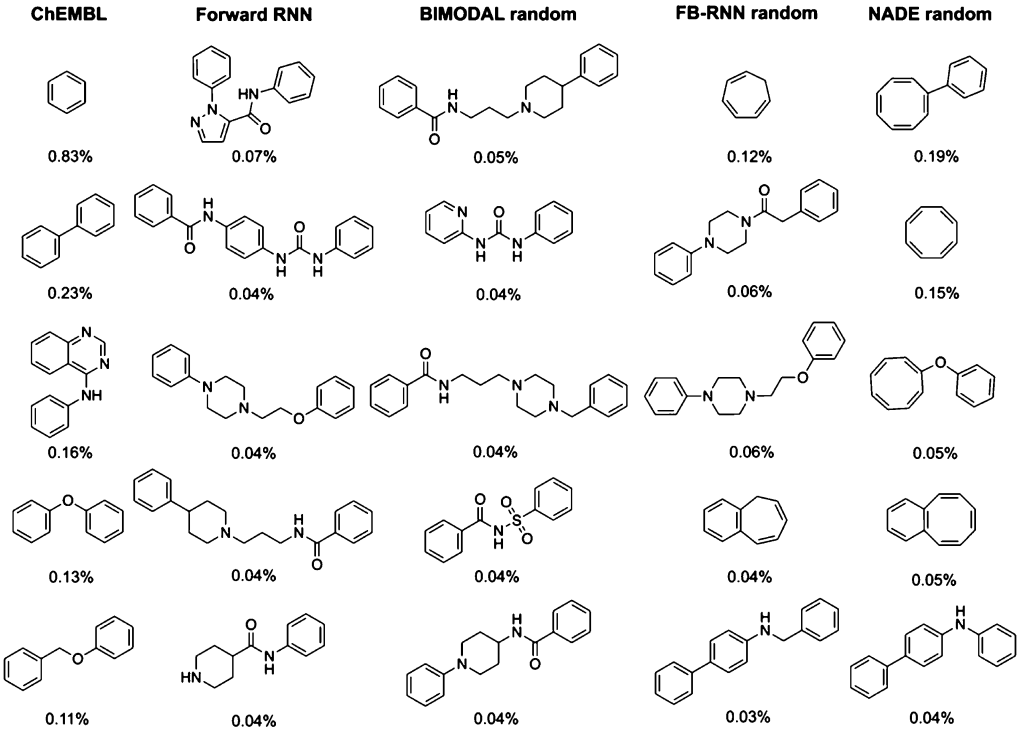


Figure 6. Most frequently generated scaffolds of each investigated model (512 hidden units and no data augmentation), in comparison with ChEMBL22 scaffolds. Only "novel" scaffolds (i.e., scaffolds not present in ChEMBL22) were reported. Numbers indicate the scaffold occurrence in 30,000 de novo designs. The complete list of frequently recurring scaffolds obtained with each method can be found in Table S3.

Table 3. FCD Values (Mean ± Std. Dev.) of the Generated Molecules for Each Model, with Different Network Size (512 and 1024 Hidden Units) and Different Augmentation Levels (No Augmentation and Fivefold Augmentation)[a]

			number of hidden units	
model	starting point	data augmentation	512	1024
forward	fixed		**2.41±0.03**	2.0 ± 0.1
BIMODAL	fixed		2.67 ± 0.08	**1.59±0.03**[b,c]
FB-RNN	fixed		2.86 ± 0.01	1.93 ± 0.04[b,d]
NADE	fixed		6.91 ± 0.03	6.3 ± 0.1
BIMODAL	random		3.30 ± 0.01	2.42 ± 0.05
FB-RNN	random		3.52 ± 0.04	2.35 ± 0.03
NADE	random		8.01 ± 0.06	7.20 ± 0.03
BIMODAL	random	5×	**2.40±0.01**[b,d]	**1.62±0.04**[b,c]
FB-RNN	random	5×	**2.63±0.04**[b,d]	**1.75±0.01**[b,c]

[a]For each tested number of hidden units, the smallest FCD value is highlighted in boldface. [b]Results of ANOVA with Tukey HSD post hoc test to the detect significant differences with the forward RNN (with the same no. of hidden units, $\alpha = 0.05$). [c]Significantly lower FCD compared to the forward RNN with the same no. of hidden units ($p < 0.001$). [d]No significant difference compared to the forward RNN model with the same no. of hidden units.

which may pose a challenge for synthesis. Of note, the pharmaceutical drugs in DrugBank[56] (v5.0.4) do not contain eight-membered cores. Compared to NADE, the other generative methods produced a more heterogeneous set of scaffolds (Figure 6).

Biological and Chemical Relevance. The sampled 30,000 unique and novel SMILES were used to calculate the FCD from the training set molecules. The lower the FCD, the closer the generated molecules are to the training set, in terms of structural and biological properties. Unlike previous studies,[27,57] we only used novel and unique molecules, to not reward models that reproduce the training set molecules.

With 512 hidden units and no data augmentation, the forward RNN produced the smallest FCD value, followed by BIMODAL, and FB-RNN with fixed starting points (Table 3). NADE showed the highest FCD values—more than twice the values reached by the other methods. The high FCD values indicate that the molecules generated by NADE lie far from the structural and biological domain of the training molecules. This aspect, together with the low validity and novelty of the produced SMILES, and the redundancy of the produced scaffolds, renders NADE suboptimal for molecular design.

Five-fold data augmentation on BIMODAL and FB-RNN led to FCD values comparable with those of the forward RNN (no statistically significant difference observed, analysis of variance (ANOVA) [$\alpha = 0.05$]). With increased network size (1024 hidden units), FCD values decreased for all methods and all of the tested cases (Figure 5b). Bidirectional methods experience the largest decrease of FCD values, with BIMODAL reaching significantly smaller FCD values than the forward RNN ($p <$

https://dx.doi.org/10.1021/acs.jcim.9b00943
J. Chem. Inf. Model. 2020, 60, 1175−1183

0.001, ANOVA with Tukey HSD post hoc analysis). This underscores the potential of BIMODAL to generate molecules closer in the chemical and bioactivity space of training data compared to forward RNN.

CONCLUSIONS AND OUTLOOK

The results of this study confirm the potential of bidirectional RNNs for de novo molecule design. The BIMODAL method seems to be particularly suited for prospective molecule design, judging from the features of the generated chemical entities, that is, their chemical and biological relevance (as measured by FCD) and their scaffold diversity. FB-RNN yielded moderate results, and NADE proved unsuitable in all of the cases tested. The introduced data augmentation strategy led to more accurate learning of the training data distribution and increased the novelty of the designs. Taken together, these results advocate further exploration of bidirectional strategies, in particular BIMODAL, for prospective de novo molecule design. Follow-up studies will be needed to determine the potential of diverse types of molecular representations (e.g., randomized SMILES[58]) and data augmentation[25] to further improve the performance of BIMODAL. The current implementation of BIMODAL has long runtimes (Table S4), owing to the two interacting RNNs. For this reason, alongside the method code, the pretrained model weights are made available in a GitHub repository (URL: https://github.com/ETHmodlab/BIMODAL), to be used for sampling novel molecules without the need of model re-training.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.9b00943.

Additional details on model architecture and training, frequent scaffolds, statistical tests and runtime measurements and methods' code, training data, and pretrained models are available as a GitHub repository (URL: https://github.com/ETHmodlab/BIMODAL) (PDF)

AUTHOR INFORMATION

Corresponding Authors

Francesca Grisoni − *Department of Chemistry and Applied Biosciences, RETHINK, ETH Zurich 8093 Zurich, Switzerland;* orcid.org/0000-0001-8552-6615; Email: francesca.grisoni@pharma.ethz.ch

Gisbert Schneider − *Department of Chemistry and Applied Biosciences, RETHINK, ETH Zurich 8093 Zurich, Switzerland;* orcid.org/0000-0001-6706-1084; Email: gisbert.schneider@pharma.ethz.ch

Authors

Michael Moret − *Department of Chemistry and Applied Biosciences, RETHINK, ETH Zurich 8093 Zurich, Switzerland*

Robin Lingwood − *Department of Chemistry and Applied Biosciences, RETHINK, ETH Zurich 8093 Zurich, Switzerland*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.9b00943

Author Contributions

F.G. and G.S. conceptualized the study. F.G. designed the study methodology with contribution from M.M. and R.L. R.L., and F.G. implemented the methods and performed the calculations. F.G. drafted the manuscript. All authors discussed the results and contributed to the manuscript.

Notes

The authors declare the following competing financial interest(s): G.S. is a cofounder of inSili.com GmbH, Zurich, and a consultant to the pharmaceutical industry.

ACKNOWLEDGMENTS

The authors thank Dr. P. Schneider and Dr. J. A. Hiss for the technical support. This research was supported by the Swiss National Science Foundation (SNSF, grant no. 205321_182176), the Novartis Forschungsstiftung (Free-Novation: AI in Drug Discovery), and the ETH RETHINK Initiative.

REFERENCES

(1) Dobson, C. M. Chemical Space and Biology. *Nature* **2004**, *432*, 824−828.

(2) Walters, W. P.; Stahl, M. T.; Murcko, M. A. Virtual screening-an overview. *Drug Discov. Today* **1998**, *3*, 160−178.

(3) Munk, M. E. Computer-Based Structure Determination: Then and Now. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 997−1009.

(4) Fink, T.; Reymond, J.-L. Virtual Exploration of the Chemical Universe up to 11 Atoms of C, N, O, F: Assembly of 26.4 Million Structures (110.9 Million Stereoisomers) and Analysis for New Ring Systems, Stereochemistry, Physicochemical Properties, Compound Classes, and Drug Discovery. *J. Chem. Inf. Model.* **2007**, *47*, 342−353.

(5) Wieland, T.; Kerber, A.; Laue, R. Principles of the Generation of Constitutional and Configurational Isomers. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 413−419.

(6) Miyao, T.; Arakawa, M.; Funatsu, K. Exhaustive Structure Generation for Inverse-QSPR/QSAR. *Mol. Inf.* **2010**, *29*, 111−125.

(7) Miyao, T.; Kaneko, H.; Funatsu, K. Inverse QSPR/QSAR Analysis for Chemical Structure Generation (from y to x). *J. Chem. Inf. Model.* **2016**, *56*, 286−299.

(8) Miyao, T.; Kaneko, H.; Funatsu, K. Ring-System-Based Exhaustive Structure Generation for Inverse-QSPR/QSAR. *Mol. Inf.* **2014**, *33*, 764−778.

(9) Fechner, U.; Schneider, G. Flux (2): Comparison of Molecular Mutation and Crossover Operators for Ligand-Based de Novo Design. *J. Chem. Inf. Model.* **2007**, *47*, 656−667.

(10) Devi, R. V.; Sathya, S. S.; Coumar, M. S. Evolutionary algorithms for de novo drug design—A survey. *Appl. Soft Comput.* **2015**, *27*, 543−552.

(11) Hartenfeller, M.; Zettl, H.; Walter, M.; Rupp, M.; Reisen, F.; Proschak, E.; Weggen, S.; Stark, H.; Schneider, G. DOGS: Reaction-Driven de Novo Design of Bioactive Compounds. *PLoS Comput. Biol.* **2012**, *8*, No. e1002380.

(12) Button, A.; Merk, D.; Hiss, J. A.; Schneider, G. Automated de Novo Molecular Design by Hybrid Machine Intelligence and Rule-Driven Chemical Synthesis. *Nat. Mach. Intell.* **2019**, *1*, 307−315.

(13) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Learning Internal Representations by Error Propagation*, ICS-8506; California University San Diego La Jolla, Institute for Cognitive Science, 1985.

(14) Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci. U.S.A.* **1982**, *79*, 2554−2558.

(15) Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial Autoencoders. **2015**, arXiv:1511.05644. arXiv preprint.

(16) 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 Cent. Sci.* **2018**, *4*, 268−276.

(17) Grisoni, F.; Neuhaus, C. S.; Gabernet, G.; Müller, A. T.; Hiss, J. A.; Schneider, G. Designing Anticancer Peptides by Constructive Machine Learning. *ChemMedChem* **2018**, *13*, 1300−1302.

(18) Putin, E.; Asadulaev, A.; Ivanenkov, Y.; Aladinskiy, V.; Sanchez-Lengeling, B.; Aspuru-Guzik, A.; Zhavoronkov, A. Reinforced Adversarial Neural Computer for de Novo Molecular Design. *J. Chem. Inf. Model.* **2018**, *58*, 1194−1204.
(19) Polykovskiy, D.; Zhebrak, A.; Vetrov, D.; Ivanenkov, Y.; Aladinskiy, V.; Mamoshina, P.; Bozdaganyan, M.; Aliper, A.; Zhavoronkov, A.; Kadurin, A. Entangled Conditional Adversarial Autoencoder for de Novo Drug Discovery. *Mol. Pharm.* **2018**, *15*, 4398−4405.
(20) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. **2017**, arXiv:1705.10843. Cs Stat.
(21) Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.* **2018**, *4*, 120−131.
(22) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.
(23) Ertl, P.; Lewis, R.; Martin, E.; Polyakov, V. In Silico Generation of Novel, Drug-like Chemical Matter Using the LSTM Neural Network. **2017**, arXiv:1712.07449. Cs Q-Bio.
(24) Gupta, A.; Müller, A. T.; Huisman, B. J. H.; Fuchs, J. A.; Schneider, P.; Schneider, G. Generative Recurrent Networks for De Novo Drug Design. *Mol. Inf.* **2018**, *37*, 1700111.
(25) Bjerrum, E. J.; Threlfall, R. Molecular Generation with Recurrent Neural Networks (RNNs). **2017**, arXiv:1705.04612. Cs Q-Bio.
(26) Yuan, W.; Jiang, D.; Nambiar, D. K.; Liew, L. P.; Hay, M. P.; Bloomstein, J.; Lu, P.; Turner, B.; Le, Q.-T.; Tibshirani, R.; Khatri, P.; Moloney, M. G.; Koong, A. C. Chemical Space Mimicry for Drug Discovery. *J. Chem. Inf. Model.* **2017**, *57*, 875−882.
(27) Brown, N.; Fiscato, M.; Segler, M. H. S.; Vaucher, A. C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.* **2019**, *59*, 1096−1108.
(28) Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735−1780.
(29) Merk, D.; Friedrich, L.; Grisoni, F.; Schneider, G. De Novo Design of Bioactive Small Molecules by Artificial Intelligence. *Mol. Inf.* **2018**, *37*, 1700153.
(30) Merk, D.; Grisoni, F.; Friedrich, L.; Schneider, G. Tuning Artificial Intelligence on the de Novo Design of Natural-Product-Inspired Retinoid X Receptor Modulators. *Commun. Chem.* **2018**, *1*, 68.
(31) Mou, L.; Yan, R.; Li, G.; Zhang, L.; Jin, Z. Backward and Forward Language Modeling for Constrained Sentence Generation. **2015**, arXiv:1512.06612. Cs.
(32) Berglund, M.; Raiko, T.; Honkala, M.; Kärkkäinen, L.; Vetek, A.; Karhunen, J. T. Bidirectional Recurrent Neural Networks as Generative Models. In *Advances in Neural Information Processing Systems 28*; Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc., 2015; pp 856−864.
(33) Jain, L. C.; Medsker, L. R. *Recurrent Neural Networks: Design and Applications*, 1st ed.; CRC Press, Inc.: Boca Raton, FL, USA, 1999.
(34) Hochreiter, S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *Int. J. Uncertain. Fuzziness Knowledge-Based Syst.* **1998**, *06*, 107−116.
(35) Werbos, P. J. Generalization of Backpropagation with Application to a Recurrent Gas Market Model. *Neural Netw.* **1988**, *1*, 339−356.
(36) Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. **2014**, arXiv:1412.3555. Cs.
(37) Schuster, M.; Paliwal, K. K. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673−2681.
(38) Graves, A.; Jaitly, N.; Mohamed, A. Hybrid Speech Recognition with Deep Bidirectional LSTM. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*; IEEE: Olomouc, Czech Republic, 2013; pp 273−278.
(39) Kang, S.; Cho, K. Conditional Molecular Design with Deep Generative Models. *J. Chem. Inf. Model.* **2019**, *59*, 43−52.
(40) Sattarov, B.; Baskin, I. I.; Horvath, D.; Marcou, G.; Bjerrum, E. J.; Varnek, A. De Novo Molecular Design by Combining Deep Autoencoder Recurrent Neural Networks with Generative Topographic Mapping. *J. Chem. Inf. Model.* **2019**, *59*, 1182−1196.
(41) Wang, C.; Yang, H.; Bartz, C.; Meinel, C. Image Captioning with Deep Bidirectional LSTMs. *Proceedings of the 24th ACM International Conference on Multimedia; MM '16*; ACM: New York, NY, USA, 2016; pp 988−997.
(42) Hamid, M.-N.; Friedberg, I. Identifying Antimicrobial Peptides Using Word Embedding with Deep Recurrent Neural Networks. *Bioinformatics* **2019**, *35*, 2009−2016.
(43) Goh, G. B.; Hodas, N. O.; Siegel, C.; Vishnu, A. SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties. **2017**, arXiv:1712.02034. Cs Stat.
(44) *ChEMBL22*, 2016. 10.6019/CHEMBL.Database.22.
(45) Schneider, N.; Sayle, R. A.; Landrum, G. A. Get Your Atoms in Order-An Open-Source Implementation of a Novel and Robust Molecular Canonicalization Algorithm. *J. Chem. Inf. Model.* **2015**, *55*, 2111−2120.
(46) *RDKit: Open-Source Cheminformatics*, 2019, http://www.rdkit.org.
(47) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. **2014**, arXiv:1412.6980. Cs.
(48) Moret, M.; Friedrich, L.; Grisoni, F.; Merk, D.; Schneider, G. Generating Customized Compound Libraries for Drug Discovery with Machine Intelligence. **2019**, ChemRxiv:10119299. https://doi.org/10.26434/chemrxiv.10119299.v1.
(49) Schneider, G. Mind and Machine in Drug Design. *Nat. Mach. Intell.* **2019**, *1*, 128−130.
(50) Schneider, G.; Schneider, P.; Renner, S. Scaffold-Hopping: How Far Can You Jump? *QSAR Comb. Sci.* **2006**, *25*, 1162−1171.
(51) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887−2893.
(52) Preuer, K.; Renz, P.; Unterthiner, T.; Hochreiter, S.; Klambauer, G. Fréchet ChemNet Distance: A Metric for Generative Models for Molecules in Drug Discovery. *J. Chem. Inf. Model.* **2018**, *58*, 1736−1741.
(53) Goh, G. B.; Siegel, C.; Vishnu, A.; Hodas, N. Using Rule-Based Labels for Weak Supervised Learning: A ChemNet for Transferable Chemical Property Prediction. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; KDD '18*; ACM: New York, NY, USA, 2018; pp 302−310.
(54) Simard, P.; Victorri, B.; LeCun, Y.; Denker, J. Tangent Prop-a Formalism for Specifying Selected Invariances in an Adaptive Network. *Advances in Neural Information Processing Systems*; NIPS, 1992; pp 895−903.
(55) Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. **2017**, arXiv:1703.07076. Cs.
(56) Law, V.; Knox, C.; Djoumbou, Y.; Jewison, T.; Guo, A. C.; Liu, Y.; Maciejewski, A.; Arndt, D.; Wilson, M.; Neveu, V.; Tang, A.; Gabriel, G.; Ly, C.; Adamjee, S.; Dame, Z. T.; Han, B.; Zhou, Y.; Wishart, D. S. DrugBank 4.0: Shedding New Light on Drug Metabolism. *Nucleic Acids Res.* **2014**, *42*, D1091−D1097.
(57) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; Kadurin, A.; Johansson, S.; Chen, H.; Nikolenko, S.; Aspuru-Guzik, A.; Zhavoronkov, A. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. **2018**, arXiv:1811.12823. Cs Stat.
(58) 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. Cheminf.* **2019**, *11*, 71.

NOTE ADDED AFTER ASAP PUBLICATION

Equations 2,3,4,6 were corrected on March 12, 2020.

https://dx.doi.org/10.1021/acs.jcim.9b00943
J. Chem. Inf. Model. 2020, 60, 1175−1183