JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

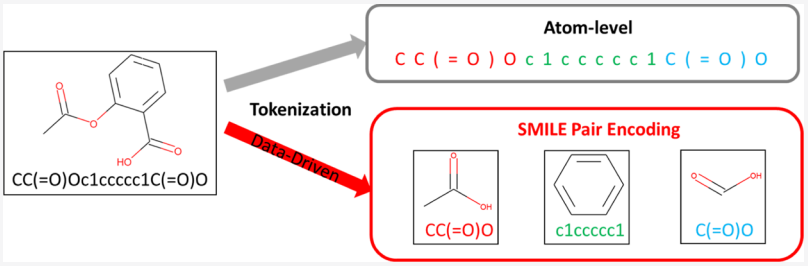
pubs.acs.org/jcim Article

SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning

Xinhao Li and Denis Fourches*

Cite This: J. Chem. Inf. Model. 2021, 61, 1560−1569 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



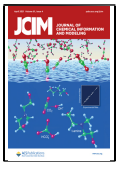
ABSTRACT: Simplified molecular input line entry system (SMILES)-based deep learning models are slowly emerging as an important research topic in cheminformatics. In this study, we introduce SMILES pair encoding (SPE), a data-driven tokenization algorithm. SPE first learns a vocabulary of high-frequency SMILES substrings from a large chemical dataset (e.g., ChEMBL) and then tokenizes SMILES based on the learned vocabulary for the actual training of deep learning models. SPE augments the widely used atom-level tokenization by adding human-readable and chemically explainable SMILES substrings as tokens. Case studies show that SPE can achieve superior performances on both molecular generation and quantitative structure−activity relationship (QSAR) prediction tasks. In particular, the SPE-based generative models outperformed the atom-level tokenization model in the aspects of novelty, diversity, and ability to resemble the training set distribution. The performance of SPE-based QSAR prediction models were evaluated using 24 benchmark datasets where SPE consistently either did match or outperform atom-level and *k*-mer tokenization. Therefore, SPE could be a promising tokenization method for SMILES-based deep learning models. An open-source Python package *SmilesPE* was developed to implement this algorithm and is now freely available at https://github.com/XinhaoLi74/SmilesPE.

1. INTRODUCTION

Over the past few years, the cheminformatics community has witnessed dramatic advances in using deep learning neural networks (DLNN) to tackle challenging tasks ranging from molecular property prediction[1−5] to *de novo* molecular generation and optimization.[6−10] The success of deep learning techniques in natural language processing (NLP) makes use of text-based molecular representations as an attractive research area.[11] Processing text-based chemical representations for deep learning models requires breaking those structures into a sequence of standard units (or "tokens"), a process called tokenization. The tokens are supposed to encode the essential structural features that are able to reliably and consistently characterize each compound. Through the specific type of neural network architectures such as recurrent neural network (RNN),[12] convolutional neural network,[13] or transformer,[14] these modeling techniques will process those string-based tokens and use them to learn the molecular representations for various modeling tasks.

In that context, SMILES[15,16] (simplified molecular input line entry system) is the most popular text representation of chemicals; it encodes a molecular graph as a fairly simple, human-readable sequence of characters. As a result, SMILES is typically used to store chemical structures, but one should underline they lack explicit 2D (and 3D) information on the atom/bond connectivity/coordinates and the overall molecular graph. Therefore, quantitative structure−activity relationship (QSAR) models based on SMILES strings are intuitively considered as less reliable as other models based on 2D (and/or 3D) molecular descriptors. The standard approach for SMILES tokenization is to simply break the SMILES string character by character. Such character-level tokenization has numerous issues such as the facts that some chemically meaningful information regarding a single atom can be represented by multiple characters and may thus result in ambiguous meanings. With character-level tokenization, "[C@@H]" is tokenized into six characters "[," "C," "@," "@," "H," and "]" even though it encodes for the stereochemistry information of a single carbon atom. The token "C" refers to the symbol of carbon but can also

Received: September 27, 2020
Published: March 15, 2021



https://dx.doi.org/10.1021/acs.jcim.0c01127
J. Chem. Inf. Model. 2021, 61, 1560−1569

Table 1. Summary of QSAR Benchmark Datasets

target names	target abbreviations	ChEMBL ID	number of molecules
α-2a adrenergic receptor	A2a	CHEMBL1867	199
dopamine D2 receptor	DRD2	CHEMBL217	469
dihydrofolate reductase	dihydrofolate	CHEMBL202	573
carbonic anhydrase II	carbonic	CHEMBL205	591
tyrosine-protein kinase ABL	ABL1	CHEMBL1862	755
μ opioid receptor	opioid	CHEMBL233	777
cannabinoid CB1 receptor	cannabinoid	CHEMBL218	1086
cyclooxygenase-1	COX-1	CHEMBL221	1306
monoamine oxidase A	monoamine	CHEMBL1951	1307
tyrosine-protein kinase LCK	LCK	CHEMBL258	1336
glucocorticoid receptor	glucocorticoid	CHEMBL2034	1387
Norepinephrine transporter	ephrin	CHEMBL222	1507
caspase-3	caspase	CHEMBL2334	1584
thrombin	coagulation	CHEMBL204	1591
estrogen receptor alpha	estrogen	CHEMBL206	1622
serine/threonine-protein kinase B-raf	B-raf	CHEMBL5145	1717
glycogen synthase kinase-3 beta	glycogen	CHEMBL262	1724
vanilloid receptor	vanilloid	CHEMBL4794	1761
serine/threonine-protein kinase Aurora-A	Aurora-A	CHEMBL4722	2084
tyrosine-protein kinase JAK2	JAK2	CHEMBL2971	2388
cyclooxygenase-2	COX-2	CHEMBL230	2759
acetylcholinesterase	acetylcholinesterase	CHEMBL220	2966
epidermal growth factor receptor erbB1	erbB1	CHEMBL203	4742
HERG	HERG	CHEMBL240	5010

be part of the symbol of chlorine ("C" and "l"). Atom-level tokenization is a more commonly used method that follows the character-level tokenization with some modifications to ensure atoms are extracted as tokens: (1) multicharacter element symbols such as "Cl" and "Br" are considered as individual tokens; (2) special characters encoded between brackets are considered as tokens (e.g., "[nH]," "[O-]," and "[C@]"). *k*-mers (also known as *n*-grams) are sequences of *k* consecutive overlapping characters in a string. 4-mers of the SMILES "CC(=O)Oc1ccccc1C(=O)O" are ["CC(=," "C(=O," "(=O)O," ..., "C(=O," "(=O)," and "=O)O"]. Vidal et al.[17] defined the LINGO profile as a vector of the occurrences of 4-mers in SMILES strings to represent the molecules. They demonstrated that the LINGO profile can be used to calculate molecular similarities and to predict molecular properties. Compared to atom-level tokens, *k*-mers naturally contains the atom connectivity information. However, the *k*-mer tokenization suffers from the so-called out-of-vocabulary issue: models can only learn the representation of the *k*-mers in the training set but fail to provide meaningful representation for new *k*-mers they have never seen before.

In this study, we present SMILES pair encoding (SPE), a data-driven substructural tokenization algorithm for deep learning applications. The SPE is inspired by the byte pair encoding (BPE) algorithm,[18] a major tokenization method in NLP. BPE was initially developed as a data compression algorithm and further adopted as a subword tokenization algorithm. BPE identifies common words and frequent subword units from a large text corpus and assigns them as unique tokens. During the tokenization process, the less common words will further be broken into frequent subword units (e.g., "goodness" is broken into "good" and "ness"). Similar to BPE, SPE identifies and keeps the frequent SMILES substrings as unique tokens. Starting from atom-level tokens, SPE generates the SMILES substring tokens by iteratively merging the high-frequency token pairs from a large chemical dataset. SPE enhances the widely used atom-level tokenization in two major aspects:

1. Chemically meaningful substructures: SPE ensures that the most common SMILES substrings are represented as unique tokens. The SMILES substrings encode molecular substructures that include richer information and better reflect the molecular functionalities compared to the atom-level tokens;
2. Shorter input for deep learning models: The input token sequences from SPE are shorter compared to those from atom-level tokenization. Shorter inputs can reduce the computational cost and accelerate DLNN model training.

Herein, we performed two case studies to showcase the potential of SPE when it comes to both molecular generation and predictive QSAR models. The goal of these case studies is to evaluate whether SPE tokenization could represent a valuable alternative to the widely used atom-level tokenization. This study demonstrates that for both generative and predictive QSAR tasks, the SPE tokenization led to superior performances compared to the atom-level tokenization. In the molecular generation case study, we trained RNN-based language models with SPE and atom-level tokenization. The SPE model is able to generate more diverse population of novel molecules but has a lower validity rate compared to the atom-level model. Further analysis shows that the SPE model is capable of generating molecules that better resemble the distribution of the training set. In the second case study, we compared the SPE, atom-level, and *k*-mer tokenization methods using 24 benchmark datasets for QSAR modeling purposes. SPE consistently showed strong performance compared to the other two tokenization methods.

2. METHOD

2.1. SMILES Pair Encoding. The SPE algorithm consists of two major steps: the vocabulary training step which learns the

https://dx.doi.org/10.1021/acs.jcim.0c01127
J. Chem. Inf. Model. 2021, 61, 1560−1569

high-frequency SMILES substrings from a large chemical dataset and the tokenization step which applies the trained vocabulary to a given dataset of SMILES, returning a sequence of tokens. In this section, we describe how to train an SPE vocabulary and how to use the trained vocabulary to tokenize SMILES for deep learning.

An SPE vocabulary is trained according to the following steps:

- Step 1: Tokenize SMILES from a large dataset (e.g., ChEMBL[19]) at the atom-level;
- Step 2: Initialize the vocabulary with all unique tokens;
- Step 3: Iteratively count the occurrence of all token pairs in the tokenized SMILES, merge the most frequent occurring token pair as a new token, and add it to the vocabulary. This step will stop when one of the conditions is met: (1) A desired vocabulary size is achieved or (2) no pair of tokens affords a frequency larger than a given frequency threshold (FT). The maximum vocabulary size (MVS) and FT are hyperparameters for training SPE.

After training the SPE vocabulary, we can then tokenize any given set of SMILES strings based on the trained vocabulary. Importantly, the SMILES substrings in the trained vocabulary are ordered by their frequency and can be used for the chemical analysis of that particular database. During the tokenization process, each SMILES string is first tokenized at the atom level. SPE can then iteratively check the frequency of all pairs of tokens and merge the pair of tokens that have the highest frequency count in the trained SPE vocabulary until no further merging operation can be conducted.

It is worth noting that the proposed algorithm can also be applied to other popular text-based representation of chemicals for DLNN applications such as DeepSMILES[20] and SELFIES.[21] DeepSMILES is a variant of SMILES with a different representation of branches and rings but still shares the same atom-level characters with SMILES. Moreover, SELFIES represents all information of a molecular graph (atoms, bonds, branches, and rings) as characters in brackets. These characters can be directly recognized as tokens by the atom-level tokenization. As a result, one could train a specific SPE vocabulary dedicated for DeepSMILES or SELFIES without any modification.

2.2. Dataset Preparation. ChEMBL25[19] was used to train the SPE vocabulary and benchmark the generative models. The QSAR benchmark datasets were directly taken from a previous study by Cortés-Ciriano et al.[22] that include the curated pIC_{50} values for 24 diverse protein targets. All molecules were standardized with the following steps using MolVS[23] and RDKit[24] packages in Python: (1) Sanitize with RDKit; (2) replace all atoms with the most abundant isotope for that element; (3) remove counterions in the salts and neutralize the molecules; and (4) remove the mixtures. The canonical SMILES were then generated for modeling. After curation, about 1.7 million ChEMBL25 SMILES did remain. The QSAR benchmark datasets are summarized in Table 1 and the distribution of pIC50 values is summarized in Figure S1 in the Supporting Information.

2.3. Machine Learning. The molecular generation was formulated as a language/text modeling task, which was first introduced by Waller et al.[25] for *de novo* molecular design. The RNN-based language models were trained using a large chemical data set to predict the next token t_{i+1}, given a sequence of tokens $\{t_1, t_2, ..., t_i\}$ preceding it. The models learn a probability distribution of the training molecules and can then sample from the learned distribution to generate new molecules.

The QSAR models were developed using the MolPMoFiT framework we developed recently.[26] MolPMoFiT is an effective transfer learning method for QSAR modeling, which uses the chemical language model pretraining + task-specific fine-tuning strategy.[27] Fine-tuning the pretrained language model on QSAR datasets enables the knowledge learned from the large scale unlabeled chemical data to be transferred to smaller supervised datasets. In order to evaluate whether the SMILES-based deep learning methods can provide robust and reliable predictions compared to traditional QSAR modeling methods, we also included random forest (RF) models trained with ECFP6 as a baseline comparison.

2.4. Evaluation Metrics. *2.4.1. Evaluation Metrics for Generative Models.* We focus on comparing SPE and atom-level tokenization on the distribution learning problem.[8] Distribution learning models aim to learn a probability distribution on the training set and generate new molecules with similar structures and properties by sampling from the learned distribution, which are mainly used for building large-scale virtual libraries. We used a set of metrics to evaluate the quality, diversity, and chemical space coverage of the generated molecular structures.

- *Validity*: the percentage of generated SMILES that can be converted to valid molecules. Validity evaluates how good a generative model has learned the SMILES syntax and the chemical rules for constructing molecules.
- *Novelty*: the percentage of valid molecules that are not included in the training set. A low novelty may indicate that the generative model overfits the training set.
- *Uniqueness*: the percentage of valid molecules that are unique. A low uniqueness would be an indication of mode collapse in which the model samples a limited variety of molecules from a few specific areas of the chemical space.
- Internal Diversity: The internal diversity measures the chemical diversity of a set of generated molecules. Internal diversity is also a good indicator of mode collapse. The internal diversity of a generated set of molecules G can be defined by eq 1. A larger value means a higher chemical diversity of the generated set. The $T(x_1, x_2)$ is the Tanimoto similarity between molecules x_1 and x_2 which are represented by 1024-bit ECFP6.

$$\text{Internal diversity} = 1 - \frac{1}{|G|} \sum_{(x_1, x_2) \in G \times G} T(x_1, x_2) \quad (1)$$

- Nearest Neighbor Similarity (SNN): SNN is the average Tanimoto similarity between the molecules in the generated set G and their corresponding nearest neighbors in the reference set R (eq 2). A larger value of this metric corresponds to a higher similarity between the generated set and the reference set.

$$\text{SNN} = \frac{1}{|G|} \sum_{x_G \in G} \max_{x_R \in R} T(x_G, x_R) \quad (2)$$

- Substructure Coverage: This series of metrics evaluates the generative models on the chemical space coverage of a reference set on the substructure level. The coverage is defined as the percentage of substructures in the reference set R that are also in the generative set G. We computed the coverage (full), coverage (top 1000), and coverage

https://dx.doi.org/10.1021/acs.jcim.0c01127
J. Chem. Inf. Model. 2021, 61, 1560−1569

(top 5000) for each model (eqs 3−5). In eq 3, $N_{G\cap R}$ is the number of unique substructures in both a generated set G and a reference R. N_R is the number of unique substructures in the reference set R. We evaluated the coverage of four type of substructures: (1) BRICS fragments;[28] (2) functional groups; (3) Bemis−Murcko scaffolds;[29] and (4) ring systems. We selected the algorithm proposed by Ertl[30] to automatically identify functional groups. This method allows us to identify the functional groups without a predefined list of substructures, which is more suitable for the analysis of large chemical datasets. For the extraction of ring systems, we followed the Zhang et al.'s implementation:[31] First, the monocyclic rings were identified, and then monocyclic rings with shared atom(s) were merged. The RDKit package was used for identification and extraction of the substructures. The RDKit implementation of Ertl's algorithm can be accessed at https://github.com/rdkit/rdkit/tree/master/Contrib/IFG.

- Coverage (full): the percentage of the substructures in the reference set R that are also in the generative set G.

$$\text{Coverage} = \frac{N_{G\cap R}}{N_R} \tag{3}$$

- Coverage (top 1000): the percentage of the 1000 most common substructures in the reference set R that are also in the generative set G.

$$\text{Coverage} = \frac{N_{G\cap R_{1000}}}{1000} \tag{4}$$

- Coverage (top 5000): the percentage of the 5000 most common substructures in the reference set R that are also in the generative set G.

$$\text{Coverage} = \frac{N_{G\cap R_{5000}}}{5000} \tag{5}$$

2.4.2. Evaluation Metrics for QSAR Models. All 24 QSAR benchmark datasets correspond to regression tasks. The root-mean-square-error (RMSE), coefficient of determination (R^2), and mean absolute error (MAE) were used as evaluation metrics for the regression models. Cohen's d[32] (eq 6) measures the relative performances of two methods. $\bar{x}_1$ and $\bar{x}_2$ are the mean values for each group of results. SD_1 and SD_2 are the standard deviations for each group of results. A positive d value means method 1 has a larger mean than method 2, while a negative d value means method 1 has a smaller mean than method 2. The thresholds of small, medium, and large effects are set to 0.2, 0.5, and 0.8 as recommended.[32,33] The effect with a $|d|$ (absolute value of d) less than 0.2 means no difference, between 0.2 and 0.5 means minor difference, between 0.5 and 0.8 means medium difference, and greater than 0.8 means large difference.

$$\text{Cohen's } d = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{(SD_1^{\,2} + SD_2^{\,2})/2}} \tag{6}$$

2.5. Experimental Section. *2.5.1. Training an SPE Vocabulary.* SPE is a data-driven algorithm, and therefore both data quality and quantity are crucial. SMILES augmentation[26,34−37] is widely used as a data augmentation technique in deep learning applications. In order to capture common SMILES substrings in both canonical and noncanonical SMILES, we generated one noncanonical SMILES for each canonical SMILES in the curated ChEMBL dataset. As a result, 3.4 M SMILES were obtained for training the actual SPE vocabulary. The MVS was set to 30,000 and the FT was set to 2000 to ensure the common SMILES substrings can be included in the vocabulary.

2.5.2. Tokenization. The implementation of the SPE, atom-level, and k-mer tokenization methods can be accessed at https://github.com/XinhaoLi74/SmilesPE. Following the choice of LINGO,[17] we use $k = 4$ for the k-mer tokenization in this study. The 4-mers of a SMILES string is collected by using a sliding window to walk over the sequence of atom-level characters. From a string of length n, $(n - k) + 1$ k-mers can be collected. Herein, the length of 4-mers sequence of a SMILES is $(n - 3)$, where n is the number of atom-level characters in the SMILES.

2.5.3. Language Models. We trained three language models with SPE tokenization, atom-level tokenization, and k-mer tokenization. A high-quality language model requires a large training corpus. The SMILES augmentation allows the language model to learn the chemical rules while not overfitting the syntax rules of SMILES canonicalization. Herein, 9 million SMILES (1 canonical + 5 noncanonical SMILES for each compound) generated from the curated ChEMBL25 dataset are used for the model training. Among the three tokenization methods for SMILES-based deep learning, the k-mer tokenization suffers from the out-of-vocabulary problem; herein, we defined the vocabulary of k-mer with the 30,000 most common 4-mers extracted from the training set, and all other 4-mers were replaced with "[UNK]" (unknown). The model architecture we choose for language modeling is AWD-LSTM[38] (ASGD weight-dropped LSTM), a variant of LSTM (long short-term memory) models that are enhanced with various kinds of dropouts and regularizations. Specially, dropouts are applied to embedding layer, input layer, weights, and hidden layers. It has been shown as resulting in strong performances for language modeling in NLP. We chose the same model hyperparameters used in our previous MolPMoFiT study:[26] the models have an embedding layer with a size of 400, three LSTM layers with 1152 hidden units per layer, and a SoftMax layer. We apply embedding dropout of 0.1, input dropout of 0.6, weight dropout of 0.5, and hidden dropout of 0.2. Models are trained with a base learning rate of 0.008 for 10 epochs using one cycle policy.[39]

2.5.4. Molecular Generation Benchmark. We compared SPE and atom-level tokenization on the distribution-based molecular generation task. For each language model, 1 million SMILES strings were sampled and evaluated on the evolution metrics mentioned in Section 2.4.1. The validation of generated SMILES is evaluated by RDKit. The valid SMILES were then canonicalized for computing the other evaluation metrics. Because of the high computational cost, the nearest neighbor similarity was computed using 100,000 molecules randomly selected from the generated set and the training set.

2.5.5. QSAR Models. (1) Language model fine-tuning: The QSAR models were fine-tuned on the pretrained language models following the procedure of MolPMoFiT.[26] All models were tuned with base learning rates and training epochs on the validation sets and evaluated on the test sets on 10 random 80:10:10 splits. SMILES augmentation was applied as descripted in our previous study.[26] During training, the SMILES of training sets were augmented 25 times and the SMILES of validation sets were augmented 15 times. Test time augmentation was applied to compute the final predictions: for each compound, the final prediction is generated by

https://dx.doi.org/10.1021/acs.jcim.0c01127
J. Chem. Inf. Model. 2021, 61, 1560−1569

averaging predictions of the canonical SMILES and four augmented SMILES. These SMILES augmentation settings were found to perform well on a variety of datasets. (2) RF baseline models: The RF models were trained using the Scikit-learn[40] package in Python. The number of tress was set to 500, and other parameters were set to the default values. 1024-bis version of ECFP6 was computed using the RDKit package.

2.6. Implementation. The machine learning models were implemented using PyTorch,[41] fastai,[42] Scikit-learn,[40] and MolPMoFiT. The MolPMoFiT code is available at https://github.com/XinhaoLi74/MolPMoFiT.

3. RESULTS AND DISCUSSION

3.1. SPE on ChEMBL. A dataset with ~3.4 million SMILES generated from the curated ChEMBL dataset, containing both canonical and noncanonical SMILES, was used to train an SPE vocabulary. The trained SPE vocabulary contained 3002 unique SMILES substrings with length ranging from 1 to 22 (Figure 1).

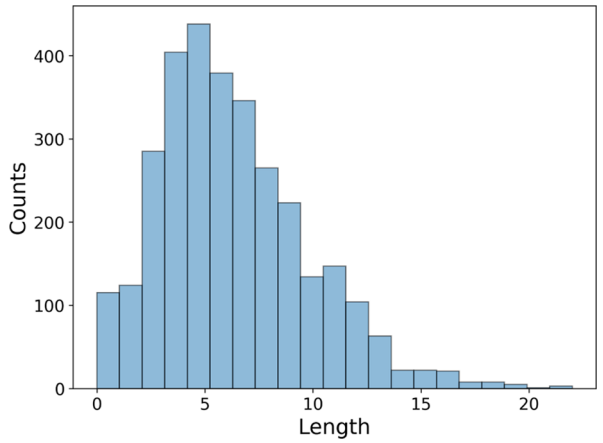


Figure 1. Distribution of length of SPE substrings trained on ChEMBL.

The length was computed by counting the number of atom-level characters in the SMILES strings. As shown in Figure 2, the SPE fragments are human-readable and mostly correspond to chemically meaningful substructures and functional groups. The full list of SPE vocabulary can be downloaded from the project GitHub repository. Several machine learning architectures[43] and techniques[37,44] can interpret the model predictions by computing the importance/contribution scores of the input tokens. In this regard, the SPE tokens are more interpretable than the individual atom characters.

Table 2 shows some examples of tokenized SMILES from SPE. Compared to atom-level tokenization, SPE provides a more compact representation of SMILES for deep learning models. Figure 3 shows the results of SPE and atom-level tokenization for the ChEMBL25 dataset. The SPE tokenization has a mean length of approximately 6 tokens, while the atom-level tokenization has a mean length of approximately 40. Such shorter input sequences can dramatically benefit DLNN models in different aspects. Because of the sequential nature of RNN-based models, they require longer training time and suffer long-term dependencies in case of long input sequences. As a result, for the same deep learning application, SPE can save the computational cost and accelerate the training and inference processes.

3.2. Molecular Generation Case Study. In this study, we focus on the distribution learning problem[8] in which models

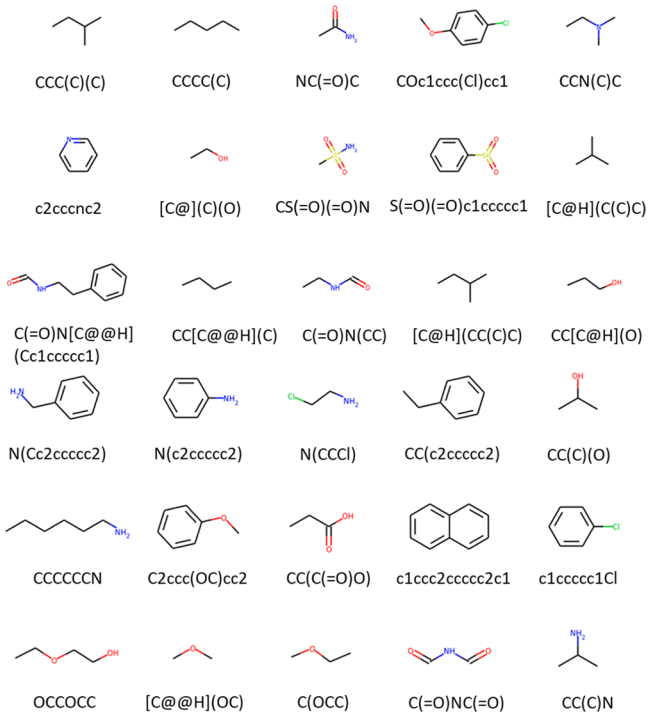


Figure 2. Representative SPE fragments.

Table 2. Example of Tokenized SMILES with SPE

SMILES	tokenized SMILES substrings
CC(CCCCC(=O)Nc1ccc(C(F)(F)F)cc1)NCC(O)c1cccc(Cl)c1	"CC(", "CCCC", "C(=O)Nc1ccc(", "C(F)(F)F)cc1)", "N", "CC(O)", "c1cccc(Cl)c1"
CCC(O)(C(=O)Nc1ccccc1Cl)C(F)(F)F	"CCC(O)(", "C(=O)N", "c1ccccc1Cl)", "C(F)(F)F"
O=C1CS/C(=N/N=C\c2ccco2)N1Cc1ccccc1	"O=C1", "CS", "/C(", " =N/N", " =C\\", "c2ccco2)", "N1", "Cc1ccccc1"

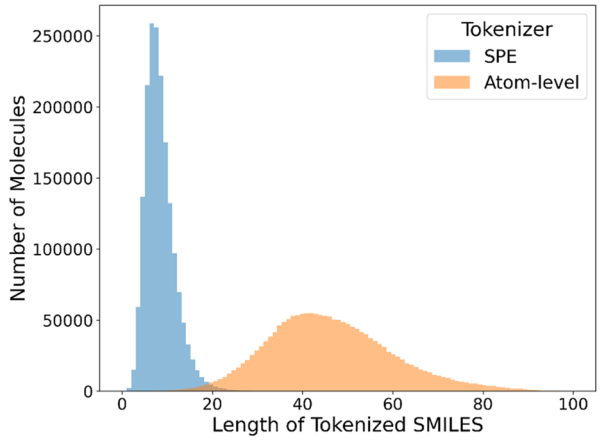
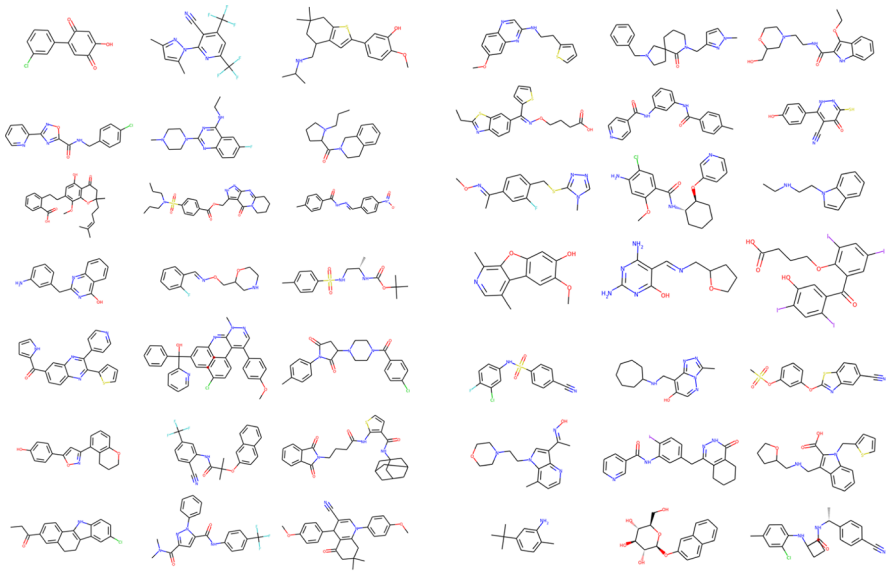


Figure 3. Distribution of length of tokenized SMILES of ChEMBL. Blue: SPE tokenization; orange: atom-level tokenization.

learn to reproduce the distribution of the training set. The trained models are expected to be able to generate molecules similar to the training molecules. We evaluated the performance of SPE versus atom-level tokenization using an RNN-based language model architecture described in the Experimental Section. The models were trained using 9 million SMILES (1 canonical + 5 noncanonical SMILES for each compound) generated from the curated ChEMBL25 dataset. Once the

https://dx.doi.org/10.1021/acs.jcim.0c01127
J. Chem. Inf. Model. 2021, 61, 1560−1569



(a) SMILES Pair Encoding (b) Atom Level

Figure 4. Random sampled examples of generated molecules. (a) Examples from the model trained with SPE tokenization; (b) examples from the model trained with atom-level tokenization

models were trained, 1 million SMILES were sampled from each model. Figure 4 shows some examples of molecules.

Table 3 lists the validity, novelty, uniqueness, internal diversity, and nearest neighbor similarity of the sampled SMILES from each model. The model trained with atom-level tokenization can produce 97.0% valid SMILES, whereas the model trained with SPE tokenization can produce 94.1% valid SMILES. Both models can generate nearly 100% unique molecules. The SPE model achieved higher novelty (98.3% vs 97.8%), internal diversity (0.897 vs 0.886), and nearest neighbor similarity (0.391 vs 0.386) scores. Overall, SPE is able to generate more diverse population of novel molecules compared to the atom-level tokenization.

Table 3. Performance Metrics for Molecular Generation

metric	SPE	atom-level
validity	0.941	**0.970**
uniqueness	**0.994**	0.992
novelty	**0.983**	0.978
internal diversity	**0.897**	0.886
nearest neighbor similarity	**0.391**	0.386

We also accessed ability of the models to resemble the substructures of the training set. Figure 5 summarizes the number of substructures extracted from the 1.7 million molecules from the training set, 1 million SMILES sampled from the SPE model, and 1 million SMILES sampled from the atom-level model. From the training set, we extracted 14,500 functional groups, 35,107 ring systems, 300,315 BRICS fragments, and 474,646 scaffolds (listed in Table S1). Both models can generate more substructures than the data they were trained on. Table 4 lists the percentages of substructures in the training set that have been recovered by the models. The two evaluated models have similar substructure coverage. However, the SPE model consistently outperforms the atom-level model in

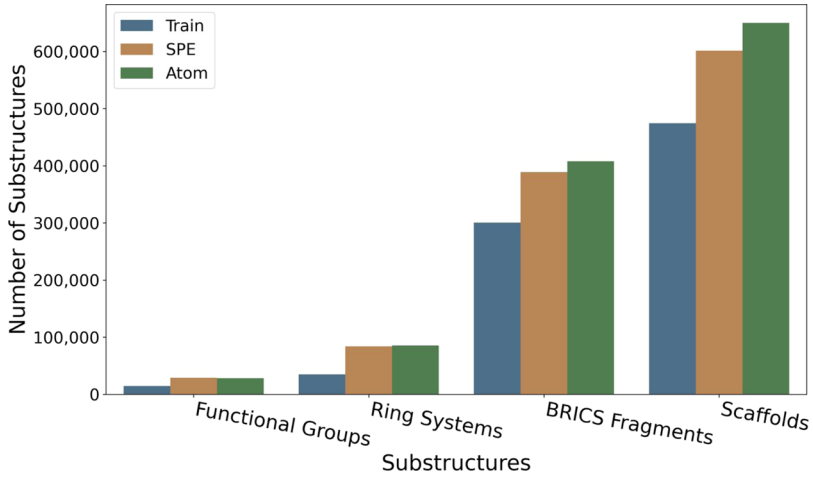


Figure 5. Number of extracted substructures from the training set, molecules sampled from the SPE model, and molecules sampled from the atom-level model.

https://dx.doi.org/10.1021/acs.jcim.0c01127
J. Chem. Inf. Model. 2021, 61, 1560−1569

Table 4. Results of Substructure Coverage for Molecular Generation Benchmark

metrics		SPE	atom-level
fragments	full	0.196	0.194
	top 1000	1.0	1.0
	top 5000	**0.997**	0.987
functional groups	full	0.380	0.380
	top 1000	**0.984**	0.971
	top 5000	**0.688**	0.659
scaffolds	full	0.127	0.126
	top 1000	**0.988**	0.976
	top 5000	**0.872**	0.825
ring systems	full	0.291	**0.297**
	top 1000	**1.0**	0.992
	top 5000	**0.781**	0.761

recovering the top-1000 and top-5000 common substructures in the training set.

3.3. Molecular Property Prediction Case Study. We compared the molecular activity prediction models trained with the SPE, atom-level, and *k*-mer tokenization methods using 24 regression datasets (pIC_{50}). In addition, we also used RF models trained with ECFP6 as a baseline comparison. The models were evaluated on 10 random 80:10:10 splits. RMSE, R^2, and MAE (Tables S2−S5) were used as evaluation metrics. In general, SMILES-based deep learning models can perform *on par* or better than the baseline RF models (Figure 6); especially on the four large datasets, COX-2, acetylcholinesterase, erbB1, and HERG, the SMILES-based models significantly outperform the RF models.

We further computed the Cohen's *d* values to evaluate the magnitude of difference in performance between the SPE and the other two tokenization methods(Figure 7). The thresholds of small, medium, and large effects were set to 0.2, 0.5, and 0.8 as recommended.[32,33] Generally, a Cohen's *d* value < 0.2 means the difference in performance is trivial. As shown in Figure 7a, models trained with SPE tokenization afforded comparable or better performances for 23 out of 24 datasets compared to those trained with atom-level tokenization. Specifically, SPE tokenization resulted in a large effect on Cannabinoid and a medium effect on A2a, LCK, Estrogen, and Aurora-A. Models trained with SPE also showed comparable or better performances for 22 out of 24 datasets compared to those trained with *k*-mer tokenization (Figure 7b). More results of the model comparison on Cohen's *d* can be found in the Supporting Information

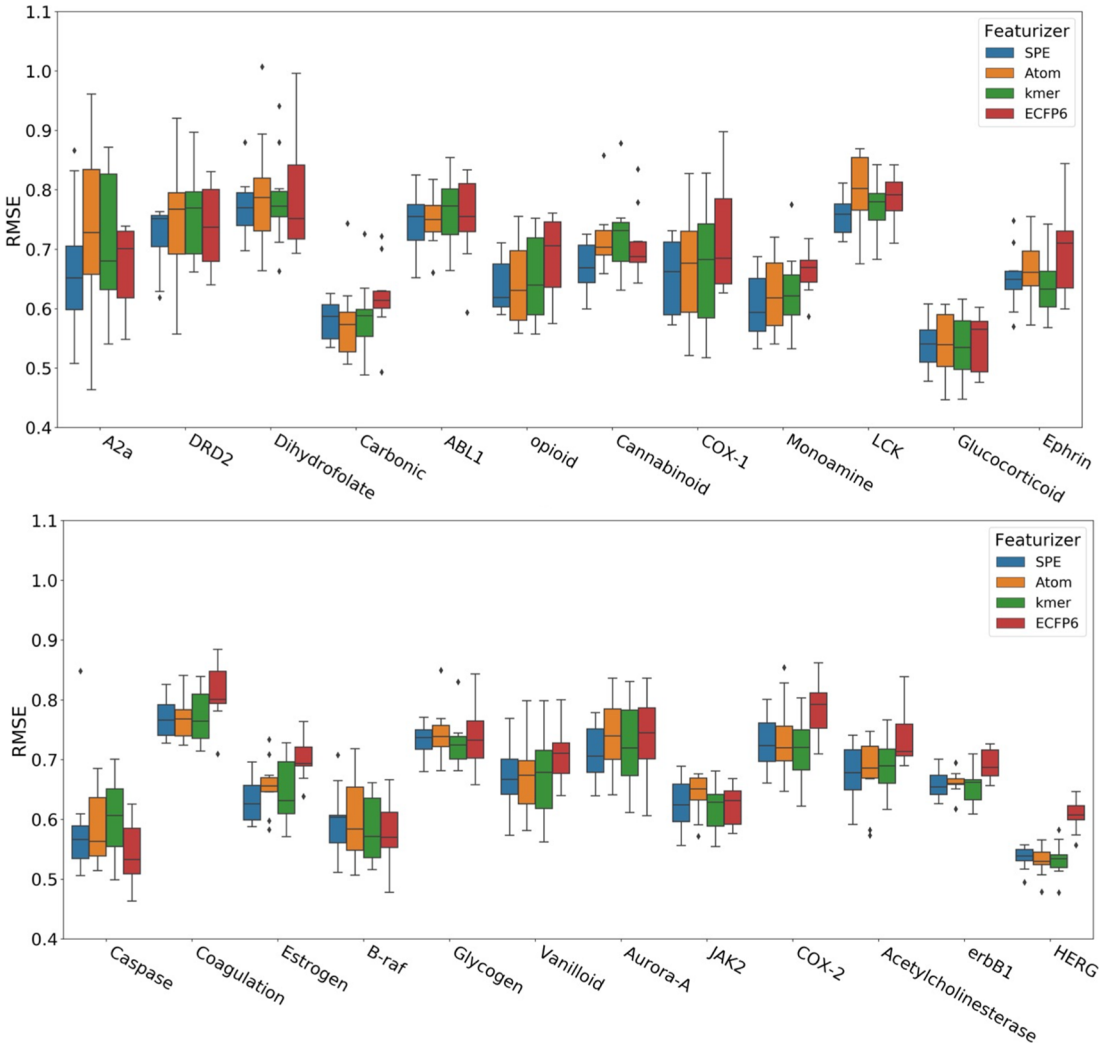
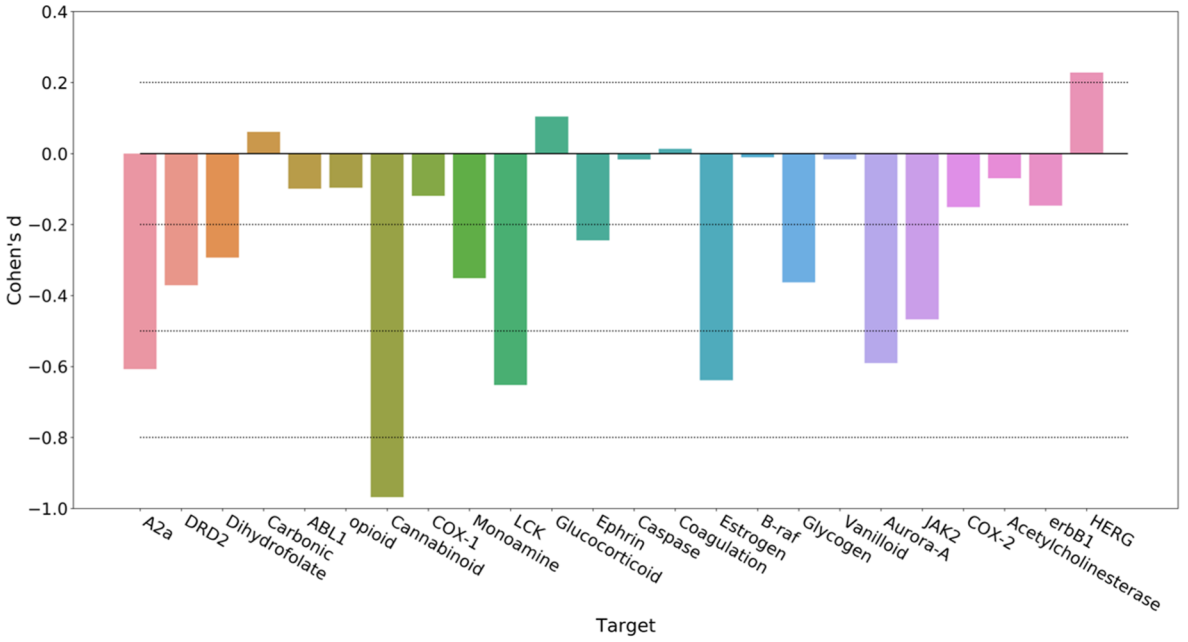
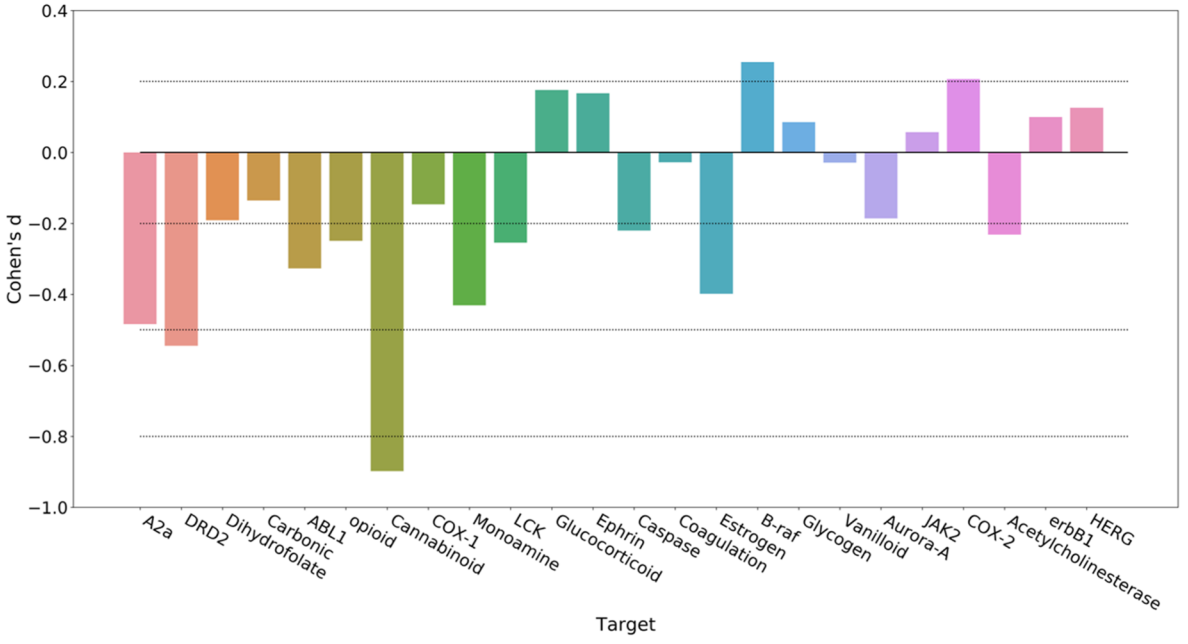


Figure 6. Test set performance on QSAR benchmark datasets. Targets are ordered by the number of molecules from small to large. (a) SPE vs atom-level (negative means SPE is better). (b) SPE vs *k*-mer (negative means SPE is better).

https://dx.doi.org/10.1021/acs.jcim.0c01127
J. Chem. Inf. Model. 2021, 61, 1560−1569



(**a**) SPE vs. Atom-level (Negative means SPE is better)



(**b**) SPE vs. k-mer (Negative means SPE is better)

Figure 7. Effect size (Cohen's *d* value) of difference between models trained with SPE tokenization and a compared tokenization method: (a) atom-level tokenization; (b) *k*-mer tokenization. A positive *d* value means the compared tokenization method performs better than SPE tokenization. A negative *d* value means SPE tokenization performs better than the tokenization method. The size effect with a |*d*| (absolute value of *d*) less than 0.2 means no difference, between 0.2 and 0.5 means minor difference, between 0.5 and 0.8 means medium difference, and greater than 0.8 means large difference.

(Figures S2−S5). In addition to the strong performances, the models with SPE were trained on average 5 times faster because of the shorter input sequence.

4. CONCLUSIONS

In this study, we proposed SPE, a data-driven substructure tokenization algorithm for deep learning. SPE learns a vocabulary of high-frequency SMILES substrings from ChEMBL and then tokenizes new SMILES into a sequence of tokens for deep learning models. SPE splits SMILES into human-readable and chemically explainable substrings and shows reliable and robust performances on both generative and predictive tasks. In the generative task, we showed that SPE is capable of generating a more diverse set of novel molecules that are more similar to the training set compared to the atom-level tokenization. In the predictive tasks, SPE showed better or comparable performances compared to the atom-level and *k*-mer tokenization methods. In addition to the strong performances, SPE has shorter input sequences which save the

https://dx.doi.org/10.1021/acs.jcim.0c01127
J. Chem. Inf. Model. 2021, 61, 1560−1569

computational cost of both model training and inferencing. Overall, SPE could represent a better tokenization method for the development of future deep learning applications in cheminformatics.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c01127.

Distributions of QSAR benchmark datasets; number of extracted substructures; performance of QSAR models trained with SPE tokenization; performance of QSAR models trained with atom-level tokenization; performance of QSAR models trained with *k*-mer tokenization; performance of RF models trained with ECFP6; Cohen's *d*: atom-level vs *k*-mer; Cohen's *d*: Cohen's *d*: SPE vs ECFP6; Cohen's *d*: atom-level vs ECFP6; and Cohen's *d*: *k*-mer vs ECFP6 (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Denis Fourches − *Department of Chemistry, Bioinformatics Research Center, North Carolina State University, Raleigh, North Carolina 27695, United States;* orcid.org/0000-0001-5642-8303; Email: dfourch@ncsu.edu

Author

Xinhao Li − *Department of Chemistry, Bioinformatics Research Center, North Carolina State University, Raleigh, North Carolina 27695, United States;* orcid.org/0000-0002-1821-2680

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c01127

Funding

The authors thank the financial support from DARPA/ARO (grant W911NF1810315) as well as the NC State Chancellor's Faculty Excellence Program.

Notes

The authors declare no competing financial interest.

■ ABBREVIATION

SPE, SMILES pair encoding; DLNN, deep learning neural networks; NLP, natural language processing; RNN, recurrent neural network; CNN, convolutional neural network; BPE, the byte pair encoding; SMILES, simplified molecular input line entry system; QSAR, quantitative structure activity relationship; LSTM, long short-term memory; TTA, test time augmentation; MolPMoFiT, molecular prediction model fine-tuning; RF, Random Forest.

■ REFERENCES

(1) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The Rise of Deep Learning in Drug Discovery. *Drug Discov. Today* **2018**, *23*, 1241−1250.

(2) Lavecchia, A. Deep Learning in Drug Discovery: Opportunities, Challenges and Future Prospects. *Drug Discov. Today* **2019**, *24*, 2017−2032.

(3) Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackermann, Z.; Tran, V. M.; Chiappino-Pepe, A.; Badran, A. H.; Andrews, I. W.; Chory, E. J.; Church, G. M.; Brown, E. D.; Jaakkola, T. S.; Barzilay, R.; Collins, J. J. A Deep Learning Approach to Antibiotic Discovery. *Cell* **2020**, *180*, 688−702.

(4) Maziarka, Ł.; Danel, T.; Mucha, S.; Rataj, K.; Tabor, J.; Jastrz ebski, S. Molecule Attention Transformer. **2020**, arXiv:2002.08264.

(5) Muratov, E. N.; Bajorath, J.; Sheridan, R. P.; Tetko, I. V.; Filimonov, D.; Poroikov, V.; Oprea, T. I.; Baskin, I. I.; Varnek, A.; Roitberg, A.; Isayev, O.; Curtalolo, S.; Fourches, D.; Cohen, Y.; Aspuru-Guzik, A.; Winkler, D. A.; Agrafiotis, D.; Cherkasov, A.; Tropsha, A. QSAR without Borders. *Chem. Soc. Rev.* **2020**, *49*, 3525.

(6) Elton, D. C.; Boukouvalas, Z.; Fuge, M. D.; Chung, P. W. Deep learning for molecular design-a review of the state of the art. *Mol. Syst. Des. Eng.* **2019**, *4*, 828−849.

(7) 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.

(8) 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.

(9) Fu, T.; Xiao, C.; Li, X.; Glass, L. M.; Sun, J. MIMOSA: Multi-Constraint Molecule Sampling for Molecule Optimization. **2020**, arXiv:2010.02318.

(10) Fu, T.; Xiao, C.; Sun, J. CORE: Automatic Molecule Optimization Using Copy & Refine Strategy. Proceedings of the AAAI Conference on Artificial Intelligence, 2019; Vol. *34*.

(11) Öztürk, H.; Özgür, A.; Schwaller, P.; Laino, T.; Ozkirimli, E. Exploring Chemical Space Using Natural Language Processing Methodologies for Drug Discovery. *Drug Discov. Today* **2020**, *25*, 689.

(12) Lipton, Z. C.; Berkowitz, J.; Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. **2015**, arXiv preprint arXiv:1506.00019.

(13) Kim, Y. Convolutional Neural Networks for Sentence Classification. *EMNLP 2014—Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014; pp 1746−1751.

(14) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Advances in Neural Information Processing Systems*; 2017; Vol. *2017*, pp 5999−6009.

(15) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Model.* **1988**, *28*, 31−36.

(16) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Model.* **1989**, *29*, 97−101.

(17) Vidal, D.; Thormann, M.; Pons, M. LINGO, an Efficient Holographic Text Based Method to Calculate Biophysical Properties and Intermolecular Similarities. *J. Chem. Inf. Model.* **2005**, *45*, 386−393.

(18) Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp 1715−1725.

(19) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−D1107.

(20) O'Boyle, N.; Dalke, A. DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. **2018**, ChemRxiv:7097960.v1.

(21) Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-Referencing Embedded Strings (SELFIES): A 100% Robust Molecular String Representation. *Mach. Learn. Sci. Technol.* **2020**, *1*, 045024.

(22) Cortés-Ciriano, I.; Bender, A. Deep Confidence: A Computationally Efficient Framework for Calculating Reliable Prediction Errors for Deep Neural Networks. *J. Chem. Inf. Model.* **2019**, *59*, 1269−1281.

(23) Swain, M. *MolVS: Molecule Validation and Standardization*.
(24) Landrum, G. *RDKit: Open-Source Cheminformatics*.
(25) 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.
(26) Li, X.; Fourches, D. Inductive Transfer Learning for Molecular Activity Prediction: Next-Gen QSAR Models with MolPMoFiT. *J. Cheminf.* **2020**, *12*, 27.
(27) Howard, J.; Ruder, S. Universal Language Model Fine-Tuning for Text Classification. In *ACL 2018—56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018; Vol. *1*, pp 328−339.
(28) Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; Rarey, M. On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces. *ChemMedChem* **2008**, *3*, 1503−1507.
(29) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887−2893.
(30) Ertl, P. An Algorithm to Identify Functional Groups in Organic Molecules. *J. Cheminf.* **2017**, *9*, 36.
(31) Zhang, J.; Mercado, R.; Engkvist, O.; Chen, H. Comparative Study of Deep Generative Models on Chemical Space Coverage. **2020**, No. 2. ChemRxiv:13234289.v1.
(32) Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*; L. Erlbaum Associates: Hillsdale, N.J., 1988.
(33) Nicholls, A. Confidence Limits, Error Bars and Method Comparison in Molecular Modeling. Part 2: Comparing Methods. *J. Comput. Aided Mol. Des.* **2016**, *30*, 103−126.
(34) Bjerrum, E. J. Smiles Enumeration as Data Augmentation for Neural Network Modeling of Molecules. **2017**, arXiv preprint arXiv:1703.07076.
(35) 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*, 1−13.
(36) Arús-Pous, J.; Blaschke, T.; Ulander, S.; Reymond, J.-L.; Chen, H.; Engkvist, O. Exploring the GDB-13 Chemical Space Using Deep Generative Models. *J. Cheminf.* **2019**, *11*, 20.
(37) Karpov, P.; Godin, G.; Tetko, I. V. Transformer-CNN: Swiss Knife for QSAR Modeling and Interpretation. *J. Cheminf.* **2020**, *12*, 17.
(38) Merity, S.; Keskar, N. S.; Socher, R. Regularizing and Optimizing LSTM Language Models. In *6th International Conference on Learning Representations*, ICLR 2018—Conference Track Proceedings, 2018.
(39) Smith, L. N. A Disciplined Approach To Neural Network Hyper-Parameters: Part 1 − Learning Rate, Batch Size, Momentum, and Weight Decay. **2018**, arXiv preprint arXiv:1803.09820 March 26.
(40) Fabian, P.; Michel, V.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Vanderplas, J.; Cournapeau, D.; Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Thirion, B.; Grisel, O.; Dubourg, V.; Passos, A.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(41) Paszke, A.; Gross, S.; et al. *Automatic Differentiation in PyTorch, 31st Conference on Neural Information Processing Systems*; NIPS, 2017.
(42) Howard, J.; Gugger, S. Fastai: A Layered API for Deep Learning. *Information* **2020**, *11*, 108.
(43) Zheng, S.; Yan, X.; Yang, Y.; Xu, J. Identifying Structure-Property Relationships through SMILES Syntax Analysis with Self-Attention Mechanism. *J. Chem. Inf. Model.* **2019**, *59*, 914−923.
(44) Goh, G. B.; Hodas, N. O.; Siegel, C.; Vishnu, A. SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties. **2017**, arXiv preprint arXiv:1712.02034.

https://dx.doi.org/10.1021/acs.jcim.0c01127
J. Chem. Inf. Model. 2021, 61, 1560−1569