Improving the reliability of molecular string representations for generative chemistry

Etienne Reboul,[*,†,‡] Zoe Wefers,[‡] Jérôme Waldispühl,[‡] and Antoine Taly[*,†]

[†]*Laboratoire de biochimie théorique, institut de biologie physico-chimique,Paris*

[‡]*Waldispühl group, Department of computer science,Mcgill,Montreal*

E-mail: reboul@ibpc.fr; taly@ibpc.fr

Abstract

Generative chemistry has seen rapid development recently. However, models based on string representations of molecules still rely largely on SMILES[1] and SELFIES[2] that have not been developed for this context. The goal of this study is to first analyze the difficulty encountered by a small generative model when using SMILES and SELFIES. Our study found that SELFIES and canonical SMILES[3] are not fully reliable representations, i.e. do not ensure both the viability and fidelity of samples. Viable samples represent novel, unique molecules with correct valence, while fidelity ensures the accurate reproduction of chemical properties from the training set. In fact, 20% of the samples generated using Canonical SMILES as input representation do not correspond to valid molecules. At variance, samples generated using SELFIES less faithfully reproduce the chemical properties of the training dataset.

As a mitigation strategy of the previously identified problems we have developed data augmentation procedures for both SELFIES and SMILES. Simplifying the complex syntax of SELFIES yielded only marginal improvements in stability and overall fidelity to the training set. For SMILES, we developed a stochastic data augmentation procedure called ClearSMILES, which reduces the vocabulary size needed to represent a SMILES dataset, explicitly represents aromaticity via Kekulé SMILES,[3] and reduces the effort required by deep learning models to process SMILES. ClearSMILES reduced the error rate in samples by an order of magnitude, from 20% to 2.2%, and improved the fidelity of samples to the training set.

Introduction

Traditional *In silico* techniques for *de novo* drug design are based on virtual screening (VS) of large online libraries. The hits found by the VS are purchased or synthesized in order to experimentally confirm their activity. The validated compounds are then optimized by medicinal chemists to achieve a desired profile of biological properties, that is, pharmacokinetics, toxicity, and activity. The traditional pipeline is as time-consuming as it is expensive. The goal of Generative Chemistry is to apply machine learning (ML) to create new compounds from a random distribution that fit the desired chemical profile.[4] One of the most popular classes of generative models is the Variational AutoEncoder (VAE).[2,5–8]

There are two predominant molecular representations for a VAE: graphs and string.[9] Molecules are most naturally represented by graph data structures, as they can be assimilated to a graph where atoms are vertices and chemical bonds are edges. However, inferring from the graph-based model can be slow and cumbersome.[10]

String representations are linear and easy to manipulate. They can be made model readable by converting the sequence of tokens to a unique one-hot encoding matrix. This one-hot matrix is more memory-friendly but less comprehensive than a graph. The early models using string representation performed poorly compared to the graph-based model counterpart.[11] The performance gap was bridged by piggybacking on advances in natural language processing (NLP), especially by using a transformer-like model.[6,12,13] Currently, the state-of-the-art for a string-based generative model performs roughly as well as the graph-based one.

The last decades have seen the emergence of multiple string molecular representations

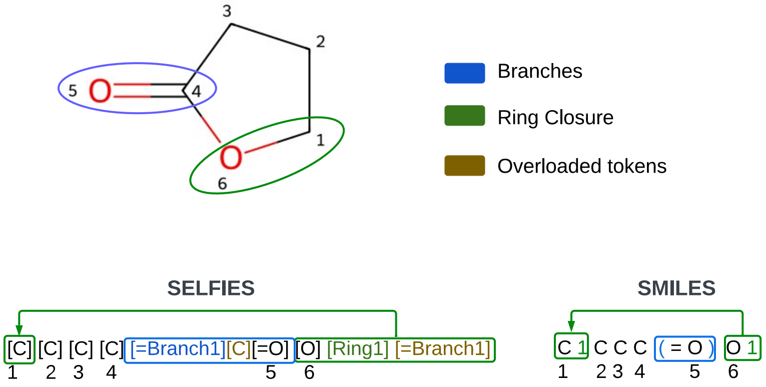

Figure 1: Example of SMILES and SELFIES for Gamma-ButyroLactone (GBL). The branch of butyrolactone is cirled in blue on the GBL's 2D structure and boxed in blue in the SELFIES and SMILES examples. The ring closure is circled in green on GBL's 2D structure and boxed in green also in the SELFIES and SMILES examples. The number displayed in the figure represent the arbitrary indexing of GBL's atoms.

dedicated to chemo-informatics usage. The oldest and most established string representation of molecules is SMILES.[1] It is widely used to store chemical structures in virtual online databases and as input to a multitude of chemo-informatics tools, including the staple of the open-source community such as RDKIT[14] and openbabel.[15] SMILES syntax is very simple and human-readable (Figure 1). Atoms are represented by their corresponding atomic symbols in the periodic table. They form the backbone of the molecule from which branches sprout to represent nonlinear chemical patterns, such as ester groups. They are represented by closed matching parentheses. Matching pairs of digits placed after the atom tokens are used to connect nonadjacent atoms, thereby forming rings.

Although SMILES are the most commonly used string representation of molecules, they are not robust. Every string of characters that complies with the SMILES branch and ring grammar does not necessarily represent a valid molecule. This usually happens when one or more of the bonds implied by the SMILES string break the valence of one of the atoms. SELFIES[2,16] were created on top of SMILES such that every string of SELFIES tokens can be translated into SMILES that represent a valid molecule. This was achieved via a rigorous and fully defined algorithm, which deletes erroneous ring, branches, and atom tokens that would imply a break of the valence of any atom. For this reason, it is stated that every string of SELFIES tokens represents a valid molecule. This is particularly advantageous in the context of machine learning, as it guarantees that the output has a valid valence. However, it does not guarantee that the generated molecules will have properties similar to those of the input molecules, such as druglikeness. The SELFIES grammar diverges significantly from the SMILES grammar. Unlike SMILES, SELFIES do not utilize pairs of digits or parenthesis tokens to represent rings and branches (Figure 1). Instead, they employ branch and ring tokens to indicate the positions of branches and rings. The sizes of branches and rings are specified by a number of overloaded tokens added after the branch or ring token. In this context, overloaded tokens are interpreted as numbers instead of their original meaning, the map to a numerical value is shown in the Supplementary Table 1.

There is a current trend in NLP for models to become bigger and more expensive to train.[17] We have chosen instead a simple RNNAttn-VAE[6] as our generative for the following reason: it is faster and cheaper to train due to its simple architecture, the single-head attention used in the decoder will be simpler to interpret, and its simplicity will make it easier to pinpoint the difficulty the neural network has in grasping the grammar of SELFIES and SMILES. The short-term aim is to have a small generative model that performs roughly similarly to more complex models that are transformer-based. In the long term, any larger model using the progresses proposed below should only benefit from them.

The main avenue we are planning to improve is the reliability of string molecular representation. Reliability for generative chemistry can be understood as the combination of viability and fidelity. A reliable string representation enables a generative model to generate novel and unique compounds with valid valence. Furthermore, a reliable string representation allows a generative model to generate molecules that faithfully capture the complex feature of the training set, such as drug-ability.

For that purpose, this study is centered around two parts : The first part is to assess what goes wrong with the generation of samples using SELFIES and SMILES as the representation for our RNNAttn-VAE. The second part is centered around mitigating strategies by developing data augmentation procedures for both SMILES and SELFIES.

Methods

Datasets

The different models were trained using the MOSES database.[11] The MOSES database is a filtered subset of the ZINC15[18] Clean Leads collection to have a representative data set of drug-like compounds. To that effect, the selected molecules were filtered using the following criteria: a weight between 250 to 350 Da, the number of rotatable bonds equal to or below seven, no charged atoms in the molecule (e.g. no nitrate group), only molecule with C, N, S, O, F, Cl, Br, and H atoms were kept, no molecules with cycles longer than eight atoms. The MOSES database consists of training set (1.7 M), test set (176 k), and scaffold test set (176 k). The scaffold test set is a dataset of molecule with unique scaffolds that never appear in the training set.

RNNAttn-VAE

A VAE is made up of two parts : an encoder and a decoder.[19] The encoder takes an input array x and compresses to the latent space of d_{latent} dimensions (where $d_{latent}<d_{input}$) as a sample array z. z is defined with the equation 1. Where μ represents the mean, σ represents the standard deviation and ϵ is a random value between 0 and 1.

$$z = \mu + \sigma \times \epsilon \tag{1}$$

The sample z represents a compressed information that contains key features of the input x and is used as input for the decoder to reconstruct x', a close as can be copy of x. By making the mean and variance of the latent space follow a known distribution, the trained decoder can be used to generate new samples from random noise generated from the known distribution.

Our code was developed from that of RNNAttn-VAE.[6] The original base code is available on github : https://github.com/oriondollar/TransVAE. Briefly, our encoder and decoder are composed of 126 Gated Recurrent Unit (GRU) layer with Batch Norm repeated 3 times. A single attention head is placed after the last GRU layer of the encoder, followed by a convolutional bottleneck. The latent space of our VAE is composed of 15 or 22 latent dimensions. A Softmax function is used to obtain the output of the model from the output of the decoder.

SMILES and SELFIES were tokenized and then converted to a one-hot encoded matrix of dimensions $n \times m$ where n is the number of tokens in the longest SMILES/SELFIES and

m is the number of possible tokens. This was used as input, and each model was trained during 100 epochs.

Metrics

fidelity metrics

The fidelity assessment of the compound was based on the following metrics : Quantitative Estimate of Drug-likeness (QED),[20] Synthetic Accessibility estimation (SA),[21] Molecular weight (MW), Topological Polar Surface Area (TPSA[22]). All metrics were computed using the RDKIT toolkit (version 2023.09.1).

samples uniqueness and novelty

The uniqueness for valid samples was calculated using the IUPAC International Chemical Identifier (InChI).[23] This ensures that molecules generated from samples are actually unique, as a given molecule can have a myriad of valid SMILES. The novelty rate was computed by computing the size of the intersection between each InChI's of samples and InChI's of training set.

SMILES validity

The validity, specifically the proper valence, of the molecules generated using SMILES was tested using RDKIT. A molecule is considered valid if RDKIT can generate a 2D structure from the generated SMILES. Validity rates are calculated as the ratio of unvalid molecules to total molecules and expressed as a percentage. For unvalid molecules, the cause of unvalidity is determined by parsing the error message and classifying it into the following categories that are similar to those found in the literature:[9,24] Aromaticity error, Ring error, Parenthesis error, Valence error, Syntax error.

SELFIES stability

The decoding algorithm SELFIES will invariably output a valid SMILES, with correct valence. Therefore, we developed a proxy validity metric called stability, as a reflection of the model's ability to learn and reproduce the SELFIES grammar. A SELFIES is considered stable if the regenerated SELFIES obtained by decoding the original SELFIES to SMILES and then re-encoding it back to SELFIES, is identical to the original SELFIES i.e. a lossless encoding-decoding. This is achieved using the selfies Python module (version 2.1).

To further quantify the loss of information related to the SELFIES instability, we introduced a proxy loss based on the amount of token lost between the original and regenerated SELFIES. The token loss is the weighted difference between the number of tokens in the regenerated SELFIES ($n_{\text{regenerated}}$) and the original SELFIES (n_{original}), as defined in equation 2 :

$$\text{token loss} = \frac{n_{\text{regenerated}} - n_{\text{original}}}{n_{\text{original}}} \times 100 \tag{2}$$

To identify the root cause of the SELFIES instability, we developed a pipeline based on the alignment of original and regenerated SELFIES to highlight the transformation made by the SELFIES algorithm.

First, a regenerated SELFIES is obtained for the stability test. Both the original SELFIES and regenerated SELFIES are tokenized to be labeled in 4 different categories: a for the atom tokens, b for the branch tokens, r for the ring tokens, and n for the overloaded/numerical tokens.

The labels and the tokens are then merged to get annotated tokens. Those annotated tokens are then aligned using the Needleman Wunsch algorithm implemented in the string2string (0.0.150) Python module. The mutations detected by alignment were classified as the 3 standard mutations in biology: insertion, deletion, and substitution.

Note that the alignments are not perfect. When decoding, the SELFIES algorithm can remove, delete, or change the length of the ring and branches. The Needleman-Wunsch algorithm will detect the modification but will map the token incorrectly when substitution and addition happen simultaneously.

Shannon Information entropy

The Shannon information entropy defined in equation 3 is used to measure the dispersion of information in the latent space.

$$S_j = -\sum_{i=1}^{N} p_i(\mu_j)\log(p_i(\mu_j)) \tag{3}$$

Data augmentation

ClearSMILES

ClearSMILES is a stochastic data augmentation procedure that aims to simultaneously achieve dimensionality reduction and minimize the global attention effort required for processing a SMILES.

In the initial step, we generated 100,000 randomized Kekulé SMILES[3] using the vectorized randomization function provided by the RDKit Python module. In Kekulé SMILES, the aromaticity is explicitly stated as a conjugated system, as illustrated in **Figure 3 (c)**). The conjugated system is represented by a combination of double bonds and aliphatic atoms, i.e., non-aromatic atoms following Huckel's rule. Double bonds are represented by '=' tokens and aliphatic atoms are represented by uppercase tokens like 'C'. For traditional and canonical SMILES, aromaticity is implied using lowercase tokens, such as 'c'. Aromatic tokens are never used in Kekulé SMILES, reducing the set of tokens needed to represent SMILES. This results in a reduction in the dimensionality of the one-hot encoding of SMILES.

In the second step, we remove duplicates and retain only unique randomized Kekulé SMILES.

Moving on to the third step, we filter the unique SMILES to keep only those with the lowest maximum digit. This serves a dual purpose. It limits the number of digit tokens needed to describe the rings in SMILES, achieving secondary dimensionality reduction of the one hot-encoding. It also guarantees that the selected randomized SMILES exhibit maximum disentanglement between rings, as depicted in **Figure 3 (b)**).

In the fourth step, we calculate the memory map for each remaining SMILES. The memory map is a sequential map that pairs a token with the number of open semantic features, rings and branches, in its position (**Figure 3 (d)**). . The memory score of the SMILES is then calculated as the arithmetic mean of the memory map. The SMILES with the minimum memory score are ClearSMILES. The objective of this step is to identify one or more SMILES that minimize the number of semantic features that a machine learning model will remember while processing the SMILES.

In the final step, the ClearSMILES are sorted alphanumerically. The first ClearSMILES selected becomes the unique sampled ClearSMILES. This step is implemented to prevent oversampling and improve consistency when working with large batches of randomized Kekulé SMILES.

SELFIES

For the SELFIES we propose two data manipulations to simplify the grammar. The first is to replace overloaded tokens with explicit numerical tokens. Overloaded tokens are the n tokens after a branch or ring token, n is the digit contained in the ring or branch tokens (e.g. [Branch**1**], [Ring**2**]). They defined N, the number of tokens in a branch, or the number of atoms that go back to close a ring. To compute N , the overloaded tokens are mapped to their index c_k (SI table 1) and used in a hexadecimal system defined by equation 4:

$$N = 1 + \sum_{k=1}^{n} 16^{n-k} c_k \tag{4}$$

The overloaded tokens are transformed to explicit numerical tokens by replacing their symbols with their corresponding index ([C] to [0]). The second manipulation is to replace the overloaded tokens with a single explicit numerical token indicating the length of the branch or ring, that is, $[N]$. The digit n in the branch or ring token preceding the overloading token (s) is deleted since it no longer has a purpose ([Branch2][Ring1][Ring1] to [Branch][17]).

Results

regular molecular string representations

Small VAE struggle to emulate molecular string representations

We sampled batches of 300,000 strings from our Variational Autoencoders (VAEs) trained over 100 epochs. For canonical SMILES-based models, we found validity rates for each batch of samples of roughly 80% (**Table 1**). We found no significant differences in the validity rates when comparing models with 15 and 22 latent dimensions (Supplementary Table 9).

We observed that the stability of SELFIES generated by non-augmented SELFIES-based VAE is consistently below 50%. This implies a partial alteration or loss of information from the original samples during the translation to SMILES, rendering them potentially less suitable as a molecular string representation for a small generative model.

Table 1: Viability metrics : Validity/stability, novelty,uniqueness for the 300k samples generated by each string based VAE with 22 latent dimensions. All metrics are expressed as percentages.

Representation	augmentation	Validity/Stability	novelty	uniqueness
SMILES	canonical	80.75	99.57	99.92
SMILES	ClearSMILES	**97.80**	99.13	99.92
SELFIES	regular	45.42	99.91	99.96
SELFIES	no overload	46.42	**99.93**	**99.98**
SELFIES	no hexadecimal	48.89	99.91	99.92

Identification of Error Origins

Given SMILES and SELFIES are error-prone, it is important to analyze the root causes of errors as a lever to minimize them.

We first looked at the types of errors with the samples produced by the canonical SMILES based VAE. The main source of error is by far aromaticity, but a significant number of errors are also associated with parenthesis, rings, and valence, but not syntax (Figure 2).

We found that there were generally no significant differences in error types between models with 15 and 22 latent dimensions. However, we did notice one notable exception: the occurrence of ring-related errors was 1.84 times more frequent in the SMILES VAE with 15 latent dimensions compared to 22 latent dimensions, as shown in the Supplementary Figure 10. All the results presented in Figure 2 are consistent with the results described previously in the literature.[24]

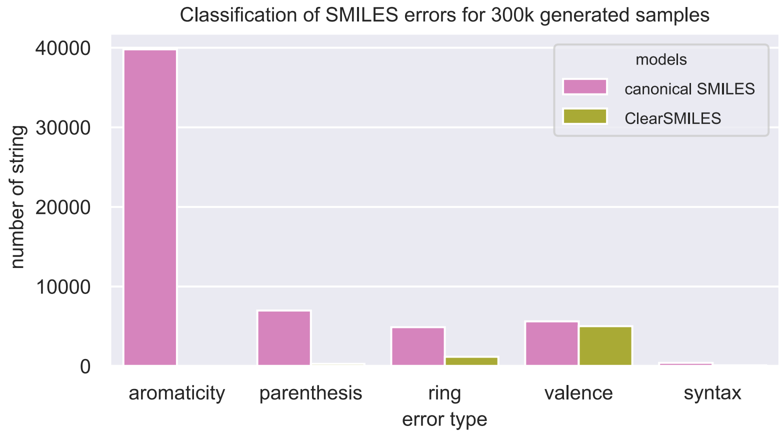

Figure 2: Error of samples generated with canonical SMILES and ClearSMILES based VAEs with 22 latent dimensions

Due to the high rate of error caused by poor aromaticity representation and the sensitivity of ring-related error to compression rate in the latent space, we decided to compare the rings present in our samples to those in our training set. We wanted to assess whether the VAE can generate samples with rings that have similar features to those found in the MOSES training set. We selected the number of atoms per ring and the number of rings per molecule as metrics to make our comparison.

We found that only 584 molecules out of 1,6 million present in the MOSES training do not have rings. For molecules with a ring, the number of rings ranged from 1 to 8 and a ring contained 3 to 7 atoms. Most molecules had 2 to 4 rings, mostly 5 or 6 atoms per ring, as illustrated in the Supplementary Figure 6.

We made a contingency table of the previously mentioned ring metrics for both our baseline, the MOSES database, and the samples generated by our VAEs. The MOSES database contains more than one million molecules, whereas batch samples contain only 300,000 molecules. To make a meaningful comparison between our dataset and our samples, we normalize the contingency tables by computing the frequency of each feature. We subtracted the normalized contingency table of MOSES from all normalized contingency tables from the sample batches. The results are presented in Supplementary Figure 7 (a) for samples generated by VAE based on canonical SMILES.

The results for samples generated by the canonical SMILES VAE with 22 latent dimensions show that samples tend to have fewer rings with 6 atoms in molecules with 3 or 4 rings and more rings with 5 atoms in a molecule with 2 rings. Increasing the compression rate of the latent space, from 22 to 15 latent dimensions, slightly widened the already identified disparities between the samples and the characteristics of the baseline ring, as shown in the Supplementary Figure 8. This might indicate a higher rate of failure for molecules with bigger rings and more rings, leading to a skewed distribution of the ring feature where molecules with smaller rings and fewer rings are more prevalent.

We checked the proportion of outlier rings, meaning those with characteristics not found in the MOSES training set, such as rings with more than 7 atoms or molecules containing more than 8 rings. We found that only 0.77% of the rings were outliers in the samples generated by Canonical SMILES VAE with 22 or 15 latent dimensions, indicating that outlier rings are a minor issue.

Considering the above observations, the SMILES should be augmented to limit long-range dependencies between matching tokens and digits, i.e. reducing the token distance between neighboring atoms. This aims to reduce the attention effort required to process an SMILES for the RNNAttn-VAE. The aromaticity of rings should also be explicitly stated in SMILES with an easily recognizable pattern of tokens. To that end, we propose a new data augmentation for SMILES: ClearSMILES (see below).

For unstable SELFIES samples, we computed the normalized token difference between the samples and their regenerated counterpart. We found that in the overwhelming majority of cases unstable SELFIES exhibit some sort of loss of information during their translation to SMILES. We aligned the unstable SELFIES with their regenerated SELFIES using the Needleman-Wunsh algorithm. We found that loss of information is predominantly related to the deletion of branches and rings (see Supplementary Information). We repeated the same analysis of the ring features we performed on the SMILES samples for the SELFIES samples. We found that SELFIES exhibit the same problem as SMILES, leading to a skewed distribution in favor of molecules with smaller and fewer rings. The notable difference is that selfie samples are even further away from the MOSES baseline than SMILES. Also, the percentage of outlier rings is an order of magnitude greater than that of SMILES samples, becoming a non-negligible issue. Our main hypothesis to explain the difference between SMILES and SELFIES is that VAE struggle to emulate the complex hexadecimal encoding using overloaded tokens to define branch and ring length. Thus, SELFIES augmentation should focus on simplifying the encoding of branch and ring length for SELFIES.

Augmented molecular string representation

ClearSMILES is based on a 5-step pipeline with the following objectives: The first aim is to reduce the number of tokens in the vocabulary that are required to represent a full set of SMILES. Thus, a reduction in dimensionality is achieved for the one hot matrix used as input of the VAEs. The second aim is to minimize the number of semantic features open across all tokens in a SMILES. In other words, it means trying to find a solution with the minimum amount of token between matching digits and matching tokens, while also minimizing the number of ring or smiles concurrently open. In a perfect theoretical case, the number of tokens between matching parentheses or digits is very small, and all branches and rings are closed before another one begins. This will limit the attention effort a deep learning model or a machine learning model has to make to process a SMILES. The different steps of the ClearSMILES pipeline are detailed in the Methods section and in Figure 3

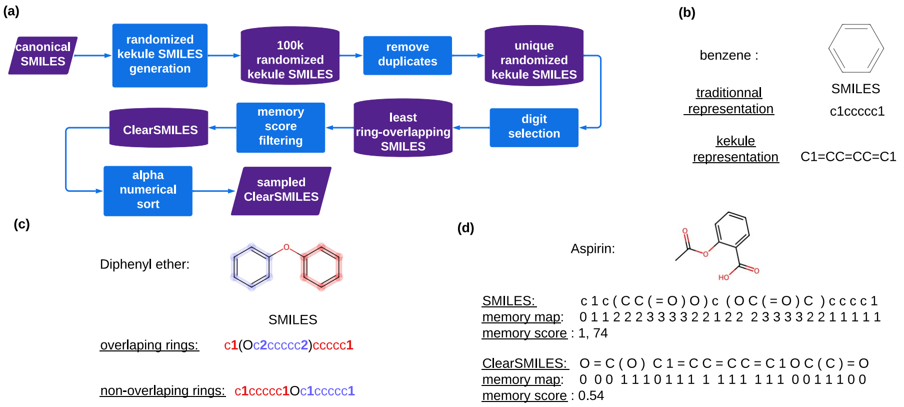
(a)
canonical SMILES
randomized kekule SMILES generation
100k randomized kekule SMILES
remove duplicates
unique randomized kekule SMILES
digit selection
least ring-overlapping SMILES
memory score filtering
ClearSMILES
alpha numerical sort
sampled ClearSMILES
(b)
benzene :
SMILES
traditionnal representation
c1ccccc1
kekule representation
C1=CC=CC=C1
(c)
Diphenyl ether:
O
SMILES
overlaping rings: c1(Oc2ccccc2)ccccc1
non-overlaping rings: c1ccccc1Oc1ccccc1
(d)
Aspirin:
O
O
HO
O
SMILES: c 1 c (C C (= O) O) c (O C (= O) C) c c c c 1
memory map: 0 1 1 2 2 2 3 3 3 3 2 2 1 2 2 2 3 3 3 3 2 2 1 1 1 1 1
memory score : 1, 74
ClearSMILES: O = C (O) C 1 = C C = C C = C 1 O C (C) = O
memory map: 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0
memory score : 0.54

Figure 3: (a) Flowchart outlining the ClearSMILES pipeline (b) benzene smiles with traditional representation of aromaticity compared to Kekulé representation (c) Contrasting overlapping and non-overlapping rings in SMILES using diphenyl ether. The red and blue highlights represent bonds and atoms in the first and second rings, respectively, following the SMILES token color-coding. (d) Memory maps and score for Aspirin using SMILES and ClearSMILES

For SELFIES we introduced two data augmentation, the first is to remove the overloading for the hexadecimal encoding of branch and ring length, the second data augmentation is to remove the hexadecimal entirely, and replace by a single non-overloaded token. For more details, please refer to the Methods section.

Analysis of augmented string properties

We applied the ClearSMILES data augmentation procedure to the 1.9 million SMILES originating from the combined MOSES training, test, and scaffold sets. This process yielded a total of 6.9 million ClearSMILES, each having a varying number of equivalent solutions, ranging from 1 to 128. Half of the ClearSMILES generated had two or less solutions, with

80% of them producing four or fewer equivalent solutions, as shown in Supplementary Table 4. The highest decile encompassed the widest range, ranging from 8 to 128 equivalent solutions.

To prevent oversampling of certain molecules and ensure some level of consistency, ClearSMILES are sorted alphanumerically using the default python sorting algorithm, and only the first ClearSMILES is kept. The retained ClearSMILES will later be referred to as the sampled ClearSMILES.

Our initial comparison focused on the number of tokens per string for canonical SMILES and sampled ClearSMILES, showing a change in the distribution of token lengths between the two, as illustrated in the Supplementary Figure 5. Specifically, sampled ClearSMILES tend to have more tokens per string compared to their canonical counterparts, as summarized in the Supplementary Table 6.

However, it is worth noting that the difference is relatively smaller when considering the longest ClearSMILES and canonical SMILES, i.e. the SMILES with the highest number of tokens for both categories. We found that the longest ClearSMILES is only 9.25% longer than the longest canonical SMILES. This is important, as the size of longest SMILES is equal to the number of row in the one-hot encoded matrix used as input to the VAE. This means that the increase in the number of rows for ClearSMILES's one hot encoded matrices is fairly limited compared to the canonical SMILES'es one-hot encoded matrices. We then performed a comparative analysis between the sampled ClearSMILES and the canonical SMILES, with a specific focus on their memory scores, branches, and rings.

The memory scores obtained from sampled ClearSMILES and the canonical SMILES generated by RDKIT exhibit significantly different distributions, as depicted in **Figure 4** (b). The memory scores for sampled ClearSMILES form a Gaussian-like distribution centered around 0.86 with a small standard-deviation of 0.13. In contrast, canonical SMILES memory scores have a distribution that strays further from a Gaussian distribution with a standard deviation of 0.66. Also, the average memory score for ClearSMILES is 0.86, approximately half of the average memory score of 1.72 for canonical SMILES.

Quantile analysis of memory scores reveals that around 90% of sampled ClearSMILES have a memory score below or equal to 1, while fewer than 20% of canonical SMILES exhibit a memory score below or equal to 1. This indicates that for an overwhelming majority of sampled ClearSMILES can be processed with a low attention effort by a neural network, whereas only a few canonical SMILES can.

We computed for each molecule the difference between the number of branches in the sampled ClearSMILES and the number of branches in the corresponding canonical SMILES, which we refer to as $\Delta branches$ (SI Table 5). In roughly 75% of the cases, there was no variation in the number of branches between a ClearSMILES sample and its corresponding canonical SMILES. There are slightly more sampled ClearSMILES with more branches than sampled ClearSMILES with less branches compared to their canonical SMILES counterpart. Overall, our findings indicate that sampled ClearSMILES exhibit very little alteration in the distribution of the number of branches compared to that of canonical SMILES.

However, a substantial difference emerges in the distribution of branch sizes between canonical SMILES and sampled ClearSMILES, as depicted in **Figure 4** (a). Sampled ClearSMILES tend to exhibit a greater prevalence of branches with sizes falling within the range of 2 to 6 tokens. Notably, there is a remarkable 52-fold reduction in the occurrence of branches strictly longer than 10 tokens in sampled ClearSMILES compared to canonical SMILES.

After analyzing the distribution of branches in Sampled ClearSMILES and canonical SMILES, we looked at the differences in terms of ring representation in strings between ClearSMILES and canonical SMILES. Because the number of rings is an inherent property of a given molecule, it remains invariant between canonical SMILES and sampled ClearSMILES and therefore was not studied.

Note that the distances between pair digits that represent a ring closure can vary depending on the path taken to traverse the molecule to generate a SMILES. If a ring closure

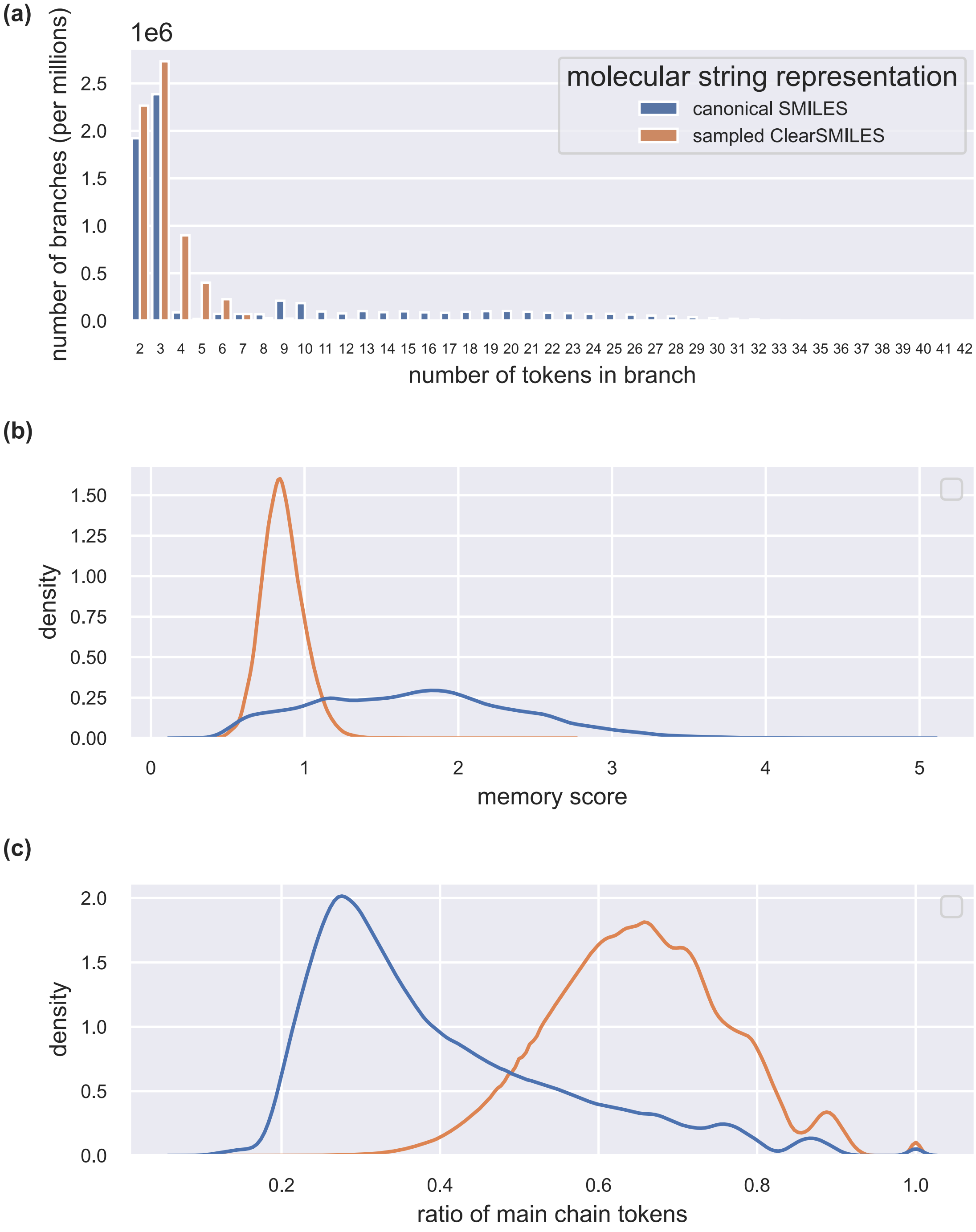

Figure 4: Histogram of the branch token distribution (a), kernel density estimation (kde) of the memory score (b) and the ratio of main chains tokens for the MOSES database : canonical SMILES (blue) and ClearSMILES (orange)

is open while another is still open, the digit representing the new ring closure is obtained by increasing the digit of the ring still open by one. When a ring is closed, the digit used to represent it can be reused for a new ring opening. Therefore, the maximum number of digits required to describe all rings can also fluctuate on the basis of the separation degree between the rings.

We computed the contingency table for the number of branches using the following two criteria: digit used in each pair representing a ring closure and the number of tokens between the paired digits. Due to the substantial difference in scale between each category, we used a logarithmic base 10 scale to enhance readability, as illustrated in Figure 5.

Sampled ClearSMILES require significantly fewer digits than canonical SMILES to represent the same molecules, as depicted in Figure 5. In fact, just two digits are sufficient to describe 99.72% of the sampled ClearSMILES, a stark contrast to canonical SMILES, where only 74.27% can be represented with two digits.

Moreover, sampled ClearSMILES exhibit a notable reduction in the occurrence of paired digits with substantial gaps between them compared to canonical SMILES, as illustrated in Figure 5. There is a remarkable 4 order of magnitude decrease in the number of paired digits with gaps exceeding 25 tokens between them with sampled ClearSMILES compared to canonical SMILES. However, paired digits with gaps of less than 16 tokens are more frequent in canonical SMILES. For smaller gap values between pair digits, Sampled ClearSMILES tends to have slightly more gap because the kekule representation in ClearSMILES uses a mixture of uppercase atom symbol (aliphatic) coupled with double bonds to represent aromaticity. In canonical SMILES aromaticity is simply represented by lowercase atom symbols. Thus, representing aromatic rings in ClearSMILES will always use more tokens than in canonical SMILES.

We then studied the graph complexity associated with our molecular string representation. Evaluating the complexity of a string representation presents a non-trivial challenge. The most effective proxy measure we identified was quantifying the number of tokens within

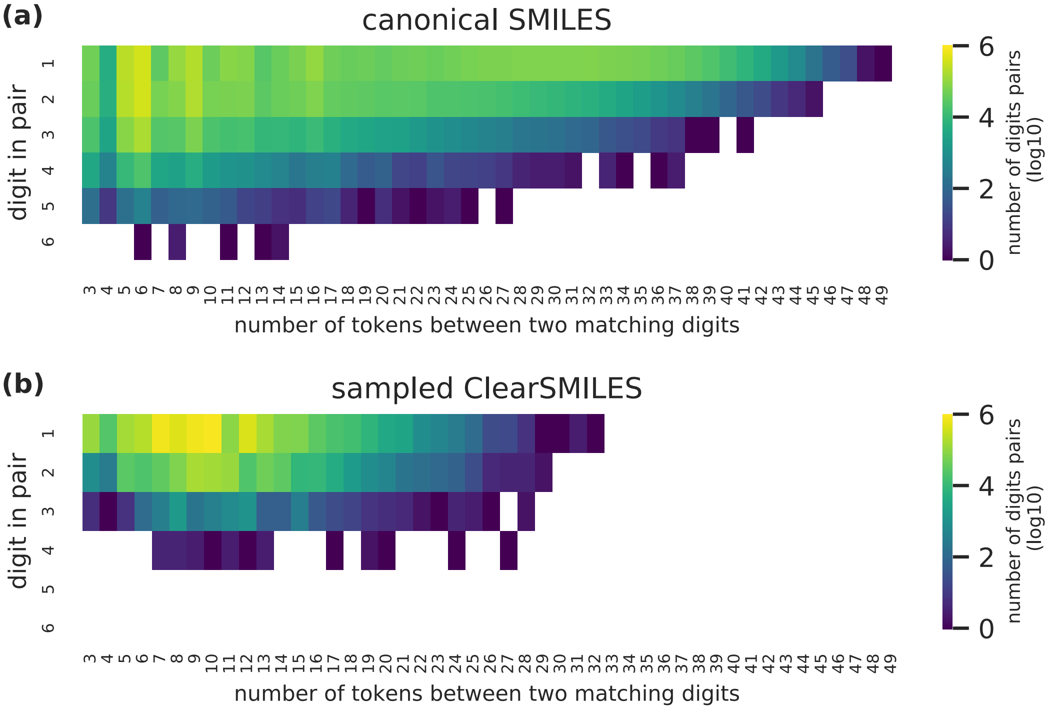

Figure 5: Heatmap illustrating the contingency table of paired digits in SMILES-based molecular string representation. A base-10 logarithmic scale is applied to enhance visibility, given the large differences in scale between categories. White spaces represent categories with zero values

the primary chain of the string. A higher token count in the main chain indicates a greater linearity in the string graph. The increased linearity in the graph, in turn, corresponds to greater simplicity in the graph structure.

Sampled ClearSMILES and canonical SMILES exhibit distinct distributions in terms of token length. To make a meaningful comparison, we calculated the ratio between the number of tokens in the main chain and the total number of tokens. This approach reveals that sampled ClearSMILES display a more uniform distribution, generally yielding higher ratios compared to canonical SMILES, as depicted in **Figure 4** (c).

For training purposes, we kept only the sampled ClearSMILES where the highest digit value used was 2 or lower, indicating up to two rings. This led to the exclusion of approximately 0.27% of the initial dataset, which predominantly contained molecules with multiple rings that share two bonds, such as the Adamantyl group, found in half of the excluded molecules. Additionally, we removed the longest strings based on token count, setting a maximum token limit of 58. The original MOSES data set train and test split was otherwise maintained.

Influence on Vocabulary and One-Hot Encoding

In natural language processing, vocabulary refers to the collection of tokens or words required to describe the entirety of the corpus, which, in this case, is the MOSES dataset. It is important to note that vocabulary is representation-specific, meaning that it will differ between SMILES, SELFIES, and their augmented counterparts. ClearSMILES have a vocabulary length of 16 tokens, with the following tokens excluded: Aromatic tokens: '[nH]','c','n', 'o', 's'; Bond token: '-' and Digit token: '3','4','5','6'. This results in a 30% reduction in vocabulary size with ClearSMILES compared to canonical SMILES vocabulary (cf. SI).

The removal of overloading from the SELFIES grammar resulted in the addition of 16 tokens and the removal of 1 token, increasing the total from 25 tokens to 40 tokens. This represents a 60% increase in the number of tokens. The newly added tokens are all numerical

(0-15).

The removal of the hexadecimal system in SELFIES led to the inclusion of 45 new tokens and the removal of 11 tokens, resulting in a total token count increase from 25 to 59. This means a 2.36-fold increase in the number of tokens.

In the vocabulary of all molecular string representations, we've added a few special tokens: '<start>' to mark the beginning of a string, '<end >' to signal its end, and '_' to pad strings to a uniform maximum length.

We conducted a comparison of the dimensions of the one-hot matrices. These matrices are characterized by dimensions of $n \times m$, n represents the maximum number of tokens per string, and m corresponds to the vocabulary size. A comprehensive summary of the one-hot matrix dimensions can be found in **Table 2**.

Table 2: Summary of one-hot encoded matrix dimensions for each molecular string representation. The resizing factor represents the ratio of the number of dimensions in the current string representation to the number of dimensions in the original string representation.

String Molecular Representation	**n**	**m**	**Resizing Factor**
SMILES	56	26	1
ClearSMILES	58	19	0.78
SELFIES	57	28	1
SELFIES no overload	57	43	1.53
SELFIES no hexadecimal	52	62	2.02

Our analysis revealed that ClearSMILES achieved a 22% reduction in the size of the one-hot matrices when compared to SMILES. Augmented SELFIES exhibited larger one-hot matrices, showing a 53% increase for SELFIES without overload and a 2.02-fold increase for SELFIES without the hexadecimal system. For simplicity sake, the n of all matrices were set to the highest value of n, which is 58.

Analysis of ClearSMILES performance

We sampled batches of 300,000 strings from our ClearSMILES-based VAEs which showed a notable change compared to those based on canonical SMILES. This shift led to a substantial increase of around 18 points in validity rates, i.e. the number of invalid molecules is decreased by roughly an order of magnitude (Table 1). There is a slight decrease of one percent point in the validity rates when going from 22 to 15 latent dimensions for samples generated by the ClearSMILES VAE as shown in the Supplementary Table 9.

The samples generated by ClearSMILES-based VAEs demonstrate a substantial reduction in the number of errors in all categories, except for valence errors. In this specific category, samples generated by ClearSMILES VAE with 15 latent dimensions exhibit a slight increase in the number of errors. Samples generated by the ClearSMILES VAE with 22 latent dimensions show a slightly lower number of errors compared to samples generated by the canonical SMILES VAE. (Figure 2).

It is important to note that ClearSMILES lacks the lowercase aromaticity representation, which means that RDKit cannot classify any error as aromaticity errors in ClearSMILES samples. Any erroneous mixture of aliphatic atoms with double bonds, which explicitly represents the conjugated system of aromatic rings in ClearSMILES, is categorized as a valence error instead of an aromaticity error. Even taking this fact into account, it is evident that ClearSMILES enables VAE to drastically reduce the number of valence and aromatic errors as their sum is much smaller than the sum of those errors with samples generated from the canonical SMILES VAE. A possible explanation to why there is a higher amount of valence error for samples generated with a ClearSMILES VAE with 15 latent dimensions can be that there is an increase in the aromaticity error that is mislabelled as a valence error by RDKit.

To further our analysis, we proceeded to compute the contingency table of ring features for the samples generated by the ClearSMILES VAE with 22 and 15 latent dimensions are shown in the Supplementary Figure 7 (b) and the Supplementary Figure 8 (b). We found that samples generated by a ClearSMILES VAE with 22 latent dimensions have ring feature closer to the MOSES baseline than the samples generated by a canonical SMILES VAE with also 22 latent dimensions. However, this comparison does not hold for samples generated by

VAE with 15 latent dimensions, where canonical SMILES outperforms ClearSMILES on this metric. Canonical SMILES samples generated by a VAE with 15 or 22 latent dimensions do not exhibit a significant difference in ring feature. However, for ClearSMILES samples there is a moderate yet noticeable stray from the MOSES ring feature distribution when the compression rate increases from 22 to 15 latent dimensions. This difference could be explained by the fact that the lowercase representation of aromaticity in canonical SMILES is more compact than the kekulé representation in ClearSMILES which could be more resilient to compression in the latent space.

For rings with outlier length, i.e. rings with more than 7 atoms, ClearSMILES-based model does not generate rings with more than 13 atoms per ring. The canonical SMILES-based model generated samples with up to 17 atoms per ring. Thus, the use of ClearSMILES limits the degree of ring 'aberrancy' without completely eliminating it. The percentages of outlier rings for samples generated by 15 and 22 latent dimensions ClearSMILES VAE are 0.38% and 0.29%, which is approximately half the rate of outlier rings for samples generated by either canonical SMILES VAE(s).

Analysis of augmented SELFIES performance

The data augmentation proposed for SELFIES did not result in a substantial enhancement of stability rates, with the most significant improvement being a maximum of a 5-point increase observed between regular SELFIES and no-overload SELFIES when using a VAE with 15 latent dimensions. Interestingly, the transition from VAEs with 22 latent dimensions to those with 15 latent dimensions showed a marginal but consistent improvement in stability rates across all data augmentation procedures and regular SELFIES.

We analyzed the token loss of each batch of 300K samples generated by the augmented SELFIES VAE. We computed the details of the distribution using the kernel density estimation (see Supplementary Information). We found that there is no distinct change in the distribution of the token loss for augmented SELFIES. We categorize the token loss as loss, gain or unchanged, as illustrated in the Supplementary Table 2. Around 91% of the unstable augmented SELFIES have a net loss of tokens. The rest of the unstable augmented SELFIES experience either a gain in token or no change in the token length.

The rings feature of samples were analyzed with contingency as before, showing that for samples generated by augmented SELFIES VAE with 22 latent dimensions, the ring features stray further from the MOSES baseline than the samples generated by the SELFIES VAE. However, it is the other way around with samples generated by VAE with 15 dimensions, although the improvement is marginal. In general, the percentages of aberrant rings for augmented samples remain in the same order of magnitude as the percentages of the original SELFIES samples with a range of 6% to 8%.

Fidelity Assessment of all models

We conducted a fidelity assessment of compounds using various metrics, including Quantitative Estimate of Drug Likeness (QED), Synthetic Accessibility estimation (SA), Molecular Weight (MW), and Topological Polar Surface Area (TPSA). Kernel Density Estimation was computed for each metric, as depicted in Figure 6. Furthermore, we calculated the Wasserstein distance for each metric between all models and the MOSES train set baseline to ensure a fair comparison. The results for models with 22 latent dimensions are presented in Table 3. The results for all models are available in the Supplementary Table 8.

Our analysis revealed that for QED, as illustrated in Figure 6 (a), SMILES-based models generate samples that consistently fit better to the training set compared to the SELFIES-based models generated. SELFIES-generated samples tend to exhibit lower QED values, indicating molecules that are less drug-like. This discrepancy is particularly evident for QED values below 0.6, with models based on SELFIES producing a higher number of samples in this category than models based on SMILES. Table 3 highlights that the Wasserstein distance for QED between MOSES and samples is approximately three times lower for SMILES samples compared to SELFIES samples, confirming that SMILES samples align more closely

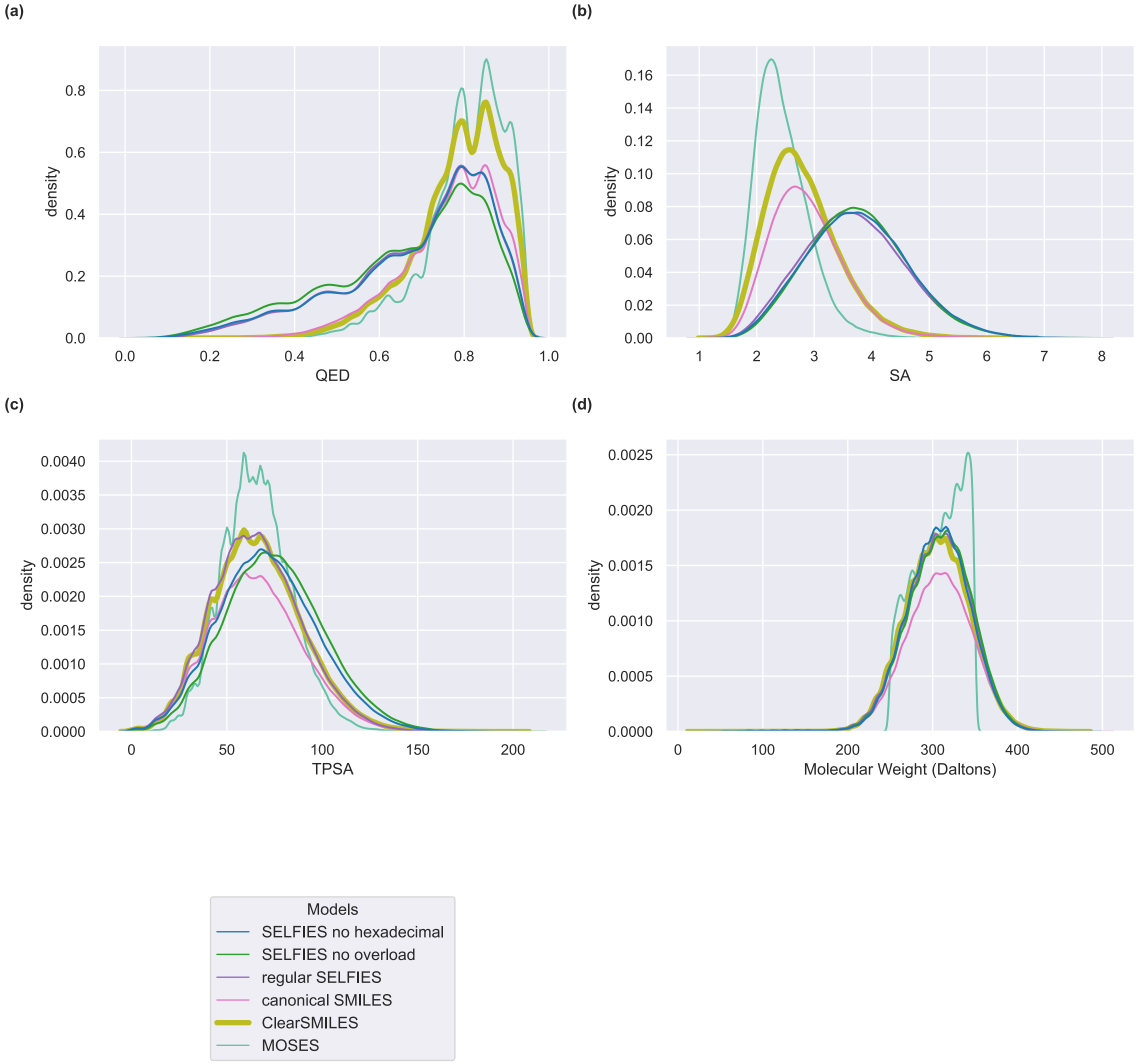

Figure 6: Assessment of various metrics for all valid samples generated by all models: (a) QED, (b) SA, (c) TPSA, (d) Molecular weight

with the MOSES baseline. Notably, the samples generated by the 22 latent dimension VAE using ClearSMILES as its string molecular representation achieve the lowest Wasserstein distance with a very satisfying fit.

The density kernel estimation for the synthetic accessibility estimation score (SA), shown in Figure 6 (b), indicates a clear difference between the distribution of samples and the MOSES baseline. SELFIES samples consistently underperform in terms of synthetic accessibility, with SMILES-based models exhibiting a distribution closer to the MOSES baseline than SELFIES-based models. The difference in Wasserstein distance further confirms this trend, with SMILES samples showing approximately 2 to 5 times lower values than SELFIES samples.

The TPSA distributions of the samples, in all representations, generally align well with the MOSES baseline, as shown in Figure 6 (c). Wasserstein metrics suggest comparable performance between SMILES-based and regular SELFIES samples, with a slightly lower distance for regular SELFIES. However, augmented SELFIES samples deviate further from the baseline in terms of TPSA, showing a 2-fold increase in the Wasserstein distance compared to regular SELFIES.

The molecular weight distributions of the samples, considering all representations, also align well with the MOSES baseline, as illustrated in Figure 6 (d). Wasserstein distance analysis indicates that the augmented SELFIES slightly outperforms other representations in terms of proximity to the baseline. Interestingly, there seems to be a slight positive effect on the Wasserstein distances by increasing the compression rate of the VAE, transitioning from 22 latent dimensions to 15 dimensions, except for samples generated by regular SELFIES VAEs as shown in the Supplementary Table 8.

Table 3: Wasserstein distance between the quantitative metrics (Molecular Weights, QED, SA, TPSA) and ring-related descriptors of the 300k random samples from the MOSES training set and each batch of 300K samples generated by VAE(s) with 22 latent dimension.

Model	TPSA	MolWeight	QED	SA
SELFIES no hexadecimal	7.798	**6.805**	0.115	1.316
SELFIES no overload	10.094	6.693	0.141	1.315
regular SELFIES	4.363	7.017	0.112	1.258
canonical SMILES	4.716	7.149	0.041	0.430
ClearSMILES	**4.304**	7.344	**0.022**	**0.345**

Discussion

In this study, we confirmed that both canonical SMILES and SELFIES fall short as robust string molecular representations and are prone to errors. The error analysis of samples generated by a Canonical SMILES based VAE revealed multiple difficulties encountered by a small generative model. The first difficulty is in accurately capturing abstract chemical properties, such as aromaticity and ring features. The lowercase representation of aromaticity in RDKit canonical SMILES does not provide an explicit representation of the conjugated system of aromatic rings. Consequently, VAEs using canonical SMILES must infer a complex set of rules from little to no information. In Kekulé SMILES the lowercase aromatic representation is replaced by an explicit representation of conjugated systems with a mixture of aliphatic tokens (uppercase) and double bonds. This combination of tokens likely forms a repetitive pattern that the VAE can recognize, aiding in the differentiation from aliphatic rings. Kekulé SMILES also benefit from reusing existing aliphatic tokens for aromatic atoms, removing the need for extra tokens.

The results suggest that Recurrent Neural Networks (RNNs) in the VAE are able to grasp the general syntax of SMILES but struggle to consistently keep track of parentheses (branches), digits (rings), and valence across a long SMILES. This is related to a known problem with RNN called the vanishing gradient problem, wherein training on a long sequence of tokens weakens the gradient used for back-propagation, making it increasingly difficult to properly capture long-term dependencies.[25,26] One of the most common ways to deal with this problem is self-attention,[27,28] a mechanism that aims to mimic human attention. However, considering the previously mentioned results, the single head of attention in the RNNAttn-VAE does not seem to be enough to fully compensate for the vanishing gradient problem. Furthermore, it seems that RNN in our VAE struggle even more with abstract chemical properties such as aromaticity. The Canonization algorithm for SMILES aims to provide a consistent way to find a single-atomic ranking to generate a unique SMILES per molecule. It does not seek to find a solution that minimizes or limits the number of long-term dependencies. ClearSMILES explicitly takes into account the long-term dependencies in a SMILES with the memory score, a heuristic serving as a proxy for the number and arrangement of open semantic features (ring closure or branch) across a SMILES. The lower the memory score, the fewer concurrently open semantic features. ClearSMILES identifies unique randomized Kekulé SMILES with minimal digits for ring closures and the lowest memory score. Fewer digits decrease the One-Hot Encoded Matrix dimensionality for VAE input, further reduced by replacing the lowercase aromatic tokens by a mixture of already used tokens. SMILES with low memory scores reduce long-range dependencies by minimizing tokens between paired parentheses and digits, leading to lower graph complexity. Removing aromatic tokens and minimizing digits for SMILES reduces vocabulary by 30%, resulting in a one-hot encoded matrix that is 22% smaller than that for canonical SMILES.

ClearSMILES reduces the generation of invalid SMILES from our VAE by an order of magnitude, increasing the validity rate from 80% to almost 98%. Samples generated using ClearSMILES had a druglikeness (QED), synthetic accessibility (SA) and Topological Polar Surface Area (TPSA) closer to the MOSES training baseline than the one generated using canonical SMILES. Therefore, although samples generated by Canonical SMILES slightly outperform ClearSMILES in terms of molecular weight, ClearSMILES largely outperforms SMILES in terms of both validity and fidelity.

We found that most samples generated using SELFIES as a molecular string representation experienced instability. This means that they could not be translated into SMILES without some loss or alteration of information. They also struggled more than SMILES-based representations in faithfully reproducing ring features. SELFIES produced about 7% outlier rings, compared to less than 0.8% for SMILES. This is likely due to the significantly more complex syntax of SELFIES, which makes it more difficult for a small generative model to properly emulate the ring features.

VAEs using SELFIES struggle to faithfully reproduce critical dataset properties such as druglikeness and synthetic accessibility. The emphasis of the SELFIES algorithm on achieving 100% validity results in information loss, particularly in the deletion of ring closures and branches, leading to "malformed" molecules with excessively large rings or improbably long chains. The comparison of Wassertein distances between stable and unstable SELFIES indicates that unstable SELFIES contribute significantly to the poor performance of sampled SELFIES. This is consistent with the fact that unstable SELFIES lose part of their information when translated to SMILES due to an erroneous token arrangement. This is consistent with the findings of the Texas SELFIES,[29] where by authorizing the decoding of SMILES with valences rule break , the author was able to remove low quality samples.The data augmentation proposed for SELFIES in this study provided only a limited increase in stability and did not address issues such as outlier rings or the chemical quality of the samples. The primary limitation appears to be the curse of dimensionality, linked to the substantial increase in vocabulary size associated with replacing overloaded tokens. Future work on SELFIES could explore ways to limit the number of tokens used to describe branch lengths and positions without resorting to overloading.

The number of latent dimensions in the original paper of RNNAttn-VAE[6] is 128 or 256 latent. The author's choice is interesting because it diverges significantly from the VAE principle of compressing data in the latent space to retain only essential input features. Taking into account the length (n) of the one-hot input matrices of the PubChem subset ($n = 128$) or their ZINC subset ($n \approx 54$) we can see that there is little to no compression but rather a decompression. Even considering the size of the vocabulary which is somewhat small, roughly 30 unique tokens, mapping sparse binary data to a continuous representation following a standard normal distribution appears trivial for a RNNAttn-VAE. Therefore, we assumed that the RNNAttn-VAE could in fact learn relevant features of data at a much higher rate of compression. In that spirit, we chose to start with 22 latent dimensions to have a little bit more than a twofold compression compared to our input size ($n = 56 - 59$).

To confirm this hypothesis, we reused the Shannon Information Entropy measurement for each latent dimension of the latent space introduced by Dollar et al.[6] Shannon entropy allows us to detect the repartition of information in the latent space. We show that the data repartition is well distributed for all SMILES-based models as shown in SI Figure 9.

Conclusion

In summary, our study successfully improved the reliability of SMILES by implementing a new data augmentation technique called ClearSMILES. This approach increased the validity rate of the samples by approximately 18 percentage points and improved the ability of a small Variational AutoEncoders (VAE) to generate samples that faithfully capture the chemical properties of compounds present in training sets. ClearSMILES achieves a compact data representation, limiting the size of the one-hot encoding matrix compared to that of the canonical SMILES for identical molecules.

Although the SELFIES representation ensures that 100% of SELFIES can be translated to SMILES with a correct valence, it leads to SELFIES instability, where information loss or alteration occurs when a sequence of SELFIES tokens with incorrect valence is altered or removed when translated to SMILES. Unfortunately, more than half of the generated SELFIES-based samples exhibit this instability, primarily characterized by the deletion of ring information during translation, resulting in a notable outlier ring rate, with rings containing up to 22 atoms. Furthermore, SELFIES samples less faithfully reproduce critical chemical properties found in the training set compared to SMILES. This is particularly true for unstable SELFIES, in terms of their estimated drug likeness and synthetic accessibility. As a result, we conclude that SELFIES, in their current form, may not be the most suitable representation for small generative models. They likely require more complex transformer-based models to better understand their complex syntax and reduce instability issues.

Acknowledgement

This research was enabled in part by the support provided by Calcul Québec (https://www.calculquebec.ca/) and the Digital Research Alliance of Canada (https://alliancecan.ca/) that provided us with the computational resources in Cedar and Graham clusters to run the experiments. The authors thank the Agence Nationale de la Recherche (ANR-21-CE45-0014) and the labex DYNAMO (11-LABX-0011) for the funding.

Data and Software Availability

The code for the ClearSMILES pipeline can be found in github :
https://github.com/EtienneReboul/ClearSMILES
The training data, models checkpoints and samples can be found on zenodo:
https://doi.org/10.5281/zenodo.14420504

References

(1) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **1988**, *28*, 31–36, Publisher: American Chemical Society.

(2) Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology* **2020**, *1*, 045024, arXiv:1905.13741 [physics, physics:quant-ph, stat].

(3) O'Boyle, N. M. Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *Journal of Cheminformatics* **2012**, *4*, 22.

(4) Vanhaelen, Q.; Lin, Y.-C.; Zhavoronkov, A. The Advent of Generative Chemistry. *ACS medicinal chemistry letters* **2020**, *11*, 1496–1505.

(5) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science* **2018**, *4*, 268–276, Publisher: American Chemical Society.

(6) Dollar, O.; Joshi, N.; C. Beck, D. A.; Pfaendtner, J. Attention-based generative models for de novo molecular design. *Chemical Science* **2021**, *12*, 8362–8372, Publisher: Royal Society of Chemistry.

(7) Jin, W.; Barzilay, R.; Jaakkola, T. Junction Tree Variational Autoencoder for Molecular Graph Generation. 2018; `https://arxiv.org/abs/1802.04364v4`.

(8) Ochiai, T.; Inukai, T.; Akiyama, M.; Furui, K.; Ohue, M.; Matsumori, N.; Inuki, S.; Uesugi, M.; Sunazuka, T.; Kikuchi, K.; Kakeya, H.; Sakakibara, Y. Variational autoencoder-based chemical latent space for large molecular structures with 3D complexity. *Communications Chemistry* **2023**, *6*, 1–14, Publisher: Nature Publishing Group.

(9) Nemoto, S.; Mizuno, T.; Kusuhara, H. Investigation of chemical structure recognition by encoder–decoder models in learning progress. *Journal of Cheminformatics* **2023**, *15*, 45.

(10) Salha, G.; Hennequin, R.; Remy, J.-B.; Moussallam, M.; Vazirgiannis, M. FastGAE: Scalable Graph Autoencoders with Stochastic Subgraph Decoding. 2021; `http://arxiv.org/abs/2002.01910`, arXiv:2002.01910 [cs, stat].

(11) Polykovskiy, D. et al. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology* **2020**, *11*, 565644.

(12) Liao, Z.; Xie, L.; Mamitsuka, H.; Zhu, S. Sc2Mol: a scaffold-based two-step molecule generator with variational autoencoder and transformer. *Bioinformatics* **2023**, *39*, btac814.

(13) Kim, H.; Na, J.; Lee, W. B. Generative Chemical Transformer: Neural Machine Learning of Molecular Geometric Structures from Chemical Language via Attention. *Journal of Chemical Information and Modeling* **2021**, *61*, 5804–5814, Publisher: American Chemical Society.

(14) Wang, S.; Witek, J.; Landrum, G. A.; Riniker, S. Improving Conformer Generation for Small Rings and Macrocycles Based on Distance Geometry and Experimental Torsional-Angle Preferences. *Journal of Chemical Information and Modeling* **2020**, Publisher: American Chemical Society.

(15) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *Journal of Cheminformatics* **2011**, *3*, 33.

(16) Krenn, M. et al. SELFIES and the future of molecular string representations. *Patterns* **2022**, *3*, 100588.

(17) Sharir, O.; Peleg, B.; Shoham, Y. The Cost of Training NLP Models: A Concise Overview. 2020; `http://arxiv.org/abs/2004.08900`, arXiv:2004.08900 [cs].

(18) Sterling, T.; Irwin, J. J. ZINC 15 – Ligand Discovery for Everyone. *Journal of Chemical Information and Modeling* **2015**, *55*, 2324–2337, Publisher: American Chemical Society.

(19) Subramaniam, S.; Mehrotra, M.; Gupta, D. Virtual high throughput screening (vHTS) - A perspective. *Bioinformation* **2008**, *3*, 14–17.

(20) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature Chemistry* **2012**, *4*, 90–98.

(21) Ertl, P.; Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics* **2009**, *1*, 8.

(22) Prasanna, S.; Doerksen, R. J. Topological Polar Surface Area: A Useful Descriptor in 2D-QSAR. *Current medicinal chemistry* **2009**, *16*, 21–41.

(23) Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics* **2015**, *7*, 23.

(24) Schoenmaker, L.; Béquignon, O. J. M.; Jespers, W.; van Westen, G. J. P. UnCorrupt SMILES: a novel approach to de novo design. *Journal of Cheminformatics* **2023**, *15*, 22.

(25) Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* **1994**, *5*, 157–166, Conference Name: IEEE Transactions on Neural Networks.

(26) Hochreiter, S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **1998**, *06*, 107–116.

(27) Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. 2016; `http://arxiv.org/abs/1409.0473`, arXiv:1409.0473 [cs, stat].

(28) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. 2023; `http://arxiv.org/abs/1706.03762`, arXiv:1706.03762 [cs].

(29) Skinnider, M. A. Invalid SMILES are beneficial rather than detrimental to chemical language models. *Nature Machine Intelligence* **2024**, *6*, 437–448, Publisher: Nature Publishing Group.