Group SELFIES: A Robust Fragment-Based Molecular String Representation

Anonymous Author(s)
Affiliation
Address
email

Abstract

The design of functional molecules relies on the representation used: a flexible and informative representation can improve downstream generation tasks. String representations such as SMILES and SELFIES serve as the basis for chemical language models, and the robustness of SELFIES makes it naturally suited for molecular optimization with genetic algorithms. But while SMILES and SELFIES are atomic representations, several recent approaches take advantage of the inductive bias of molecular fragments. In this work, we present Group SELFIES, introducing group tokens that represent functional groups or entire substructures while maintaining robustness. Group tokens give control over which structures should be preserved during optimization. Experiments indicate that Group SELFIES improves distribution learning and improves the quality of molecules generated by simply taking random Group SELFIES strings. The code is available at `https://anonymous.4open.science/r/group-selfies-4D87/`.

1 Introduction

The discovery of functional molecules for drugs and energy materials is crucial to tackling global challenges in public health and climate change. Generative models can suggest promising molecules to synthesize and test, but their performance relies on the way molecules are represented. Several models generate molecules represented as SMILES strings [1–5], but generated output is sometimes invalid due to syntax errors or incorrect valency. SELFIES [6] is a molecular string representation that overcomes this issue by ensuring that any string of SELFIES characters can be decoded to a molecule with valid valency. This not only makes it a natural representation for language models that output strings, but also for genetic algorithms such as GA+D, STONED, and JANUS [7–9] for molecular optimization.

By encoding prior knowledge of valency constraints before any training or optimization is done, SELFIES improves string-based molecular generation. It improves distribution learning by language models [10]; it improves image2string and string2string translation [11, 12]; it improves molecular generation in data-sparse regimes [13]. Additionally, simple add/replace/delete edits to SELFIES strings can generate new but similar molecules, enabling genetic algorithms that directly manipulate strings to generate molecules [7]. Alternatively, guiding these simple string edits with Tanimoto similarity can interpolate between molecules, as in STONED [8]. Interpolation with STONED can be used to find median molecules, which are used as crossover operations in the genetic algorithm JANUS [9]. Molecular interpolation via STONED was also used to find counterfactual decision boundaries that explain a molecular classifier's decisions [14].

However, while SMILES and SELFIES represent molecules at the atom and bond level, human chemists typically think about molecules in terms of the substructures that they contain. Human chemists can look at an image of a molecule and immediately recognize substructures as well as the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

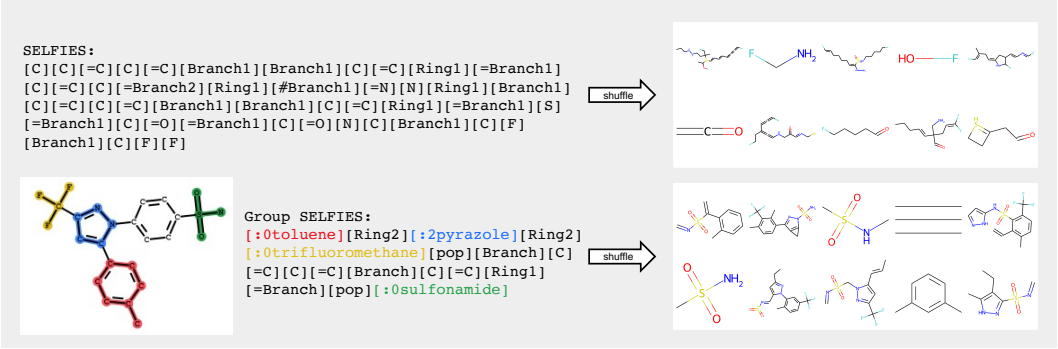

Figure 1: Visual overview of SELFIES and Group SELFIES. SELFIES is robust, so shuffling tokens around will yield new molecules with correct valency. Group SELFIES maintains robustness while adding group tokens, highlighted in color. When Group SELFIES tokens are shuffled, structures like benzene rings are more often preserved, while shuffled SELFIES strings rarely ever preserve structures.

molecular properties they imply. Many fragment-based generative models take advantage of this inductive bias [15–24]. However, these approaches are not string-based, losing desirable properties of string representations: easy manipulation, and direct input into established language models.

Just as SELFIES incorporates prior knowledge of valency constraints, we can also incorporate prior knowledge in the form of functional groups and molecular substructures. In this work, we combine the flexibility of string representations with the interpretability and inductive bias of fragment-based approaches, in addition to the robustness of SELFIES. We present Group SELFIES, a robust string representation that extends SELFIES to include tokens which represent functional groups or entire substructures. We show that arbitrary Group SELFIES strings can be decoded to molecules with valid valency. In this work, a representation is robust if any string in the representation can be decoded to a molecule that satisfies valency constraints. We provide ways for users to specify their own groups, such as extracting fragments from a dataset, leveraging the wide area of cheminformatics research available there. We find that Group SELFIES improves distribution learning versus SELFIES. Additionally, to be agnostic to the type of generative method used, we perform baselines by simply randomly generating strings of (Group) SELFIES and find that Group SELFIES improves the quality of generated molecules.

2 Related work

There are several line notations for molecules, the development of which was mostly guided by a need to store molecules in databases. SMILES [25] is the de-facto standard string-based molecule representation, but arbitrary strings of SMILES tokens may have syntax errors or represent molecules with incorrect valency - SMILES is not robust. SELFIES [6, 26] achieves syntactic robustness by removing the possibility of syntax errors, and achieves valency robustness by tracking the number of available bonds to form and "correcting" tokens that would try to perform invalid actions. All strings of SELFIES tokens can be decoded into SMILES strings that have no syntax errors and satisfy valency constraints. Since this work builds directly on top of SELFIES, Group SELFIES can be directly substituted into molecular optimization algorithms that rely on SELFIES-like representations [7–9].

Other historical line notations include Wiswesser Line Notation (WLN) [27–29], Hayward Notation [30], and Skolnik Notation [31], which all predate SMILES and represent molecules non-atomically: they use tokens that represent functional groups such as carboxyls or phenyls as well as ring systems. WLN strings are usually shorter and sometimes easier for trained humans to understand than SMILES, as it is easier to recognize functional groups encoded as single characters than functional groups encoded atomically. SYBYL Line Notation (SLN) [32] allows for "macro atoms" which specify multiple atoms in a substructure. The Hierarchical Editing Language for Macromolecules (HELM) [33] represents complex biomolecules by declaring monomers and then connecting them in a polymer line notation. Human-Readable SMILES [34] applies common abbreviations for chemical substituents

to process and compress SMILES strings into a more human-readable format. SMILES Pair Encoding [35] breaks down SMILES strings by tokenizing them in a data-driven way that recognizes common substructures.

A string representation of molecules such as SELFIES can be thought of as a *programming language* where programs specify how to construct molecules. *Genetic programming* [36] uses genetic algorithms to design programs that fulfill desired constraints. In particular for linear genetic programming [37], programs are represented as linear sequences of atomic instructions, rather than as a tree of expressions. Linear sequences of atomic instructions are easily mutated by changing any instruction in the sequence, since any sequence of atomic instructions will still be a runnable program. In this way, SELFIES can be thought of as a domain-specific language for linear genetic programming.

Data-Efficient Graph Grammar Learning (DEG) [22] is a recent approach for extracting useful formal graph grammars from small datasets of molecules. The learned production rules of the graph grammar can be thought of as similar to functional groups. In this context, a "useful grammar" means that molecules generated by applying random applicable production rules usually have high scores.

3 Grammar

3.1 SELFIES Framework

Before introducing the modifications made in Group SELFIES, we give a recap of SELFIES and the reasons that it is robust.

SELFIES is similar to SMILES in its encoding and decoding process. To encode a molecule, we traverse its molecular graph and output the processed traversal as a string. To decode a SELFIES string, we read through it token-by-token, building the molecular graph along the way. However, there are two main differences that make SELFIES fully robust:

(1) Each token in SELFIES is overloaded to ensure that it can be interpreted sensibly in all contexts. For instance, all tokens in SELFIES can also be interpreted as numbers, which is useful when expressing branch and ring lengths.

(2) SELFIES keeps track of the available valency at each step in the decoding process; if a bond would be formed that would exceed this valency, it changes the bond order or ignores the bond. For instance, when decoding `[C][O][=C]`, adding `[=C]` would exceed the valency of `[O]`, so SELFIES changes the bond order and adds `[C]` instead.

By preserving these properties throughout our modifications, we ensure that Group SELFIES is also fully robust.

3.2 Basic Notation in Group SELFIES

`[X]` adds an atom with the atomic symbol X.

`[Branch]` creates a new branch off the current atom and saves the current atom as a branchpoint to come back to later, and is analogous to an opening parenthesis `(` in SMILES. `[pop]` exits the current branch, returning to the most recent branchpoint, and is analogous to a closing parenthesis `)` in SMILES. Unlike in SMILES, however, `[Branch]` and `[pop]` tokens need not come in pairs, which helps maintain robustness. This token is also different from the `[BranchX]` tokens in SELFIES. Experiments in Appendix A.4 indicate this change does not substantially affect the performance of Group SELFIES.

`[RingX]` indicates that a ring bond will be formed from the current atom. The next `X` tokens immediately following `[RingX]` will be interpreted as a number N, and we will count backwards N atoms in placement order to determine the target of the ring bond. For example, `[Ring2]` indicates that the next 2 tokens will be interpreted as a 2-digit base-16 number N. Ring bonds are stored until after all tokens have been read by the decoder; only then are ring bonds placed, and only if it would not violate valency. Due to the addition of groups, it is sometimes necessary to form ring bonds to atoms that are added after the current atom (e.g. ring bonds within groups). To indicate this, we use the `[->]` token before the number token to specify that we will count forwards instead of backwards.

All tokens can be modified by adding =, #, \ or / to change the bond order or stereochemistry of their parent bond (e.g. [#Branch] or [/C]). The parent bond is the bond to the previous atom.

3.3 Groups

Our main modification to SELFIES is the addition of group tokens and groups, which can represent groups of atoms. To distinguish group tokens from other tokens, we include a : character at the front of the token (e.g. [:4benzene].

In order to map group tokens to their corresponding groups and to specify the set of groups that we want to use, we use a dictionary that maps group names to groups. We call this dictionary a "grammar", though it is not a formal grammar. Optionally, a priority value can be specified for each group, indicating the priority with which the group should be recognized when encoding into Group SELFIES.

Each group is defined as a normal molecule with the addition of *attachment points*, indicating how the group can participate in bonding. Each attachment point has a specified maximum valency, which allows us to continue tracking available valency while decoding. These attachment points are labeled by *attachment indices*. All group tokens are of the form [:S<group-name>], where S is the starting attachment index of the group, and <group-name> is any alphanumeric string that does not start with a number.

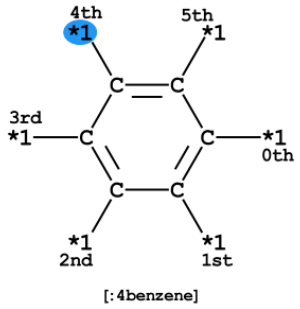

Figure 2: Representation of a possible "benzene" group and its use in a corresponding group token. The *N notation represents an attachment point with valency N. Each attachment point is labeled with its attachment index 0th,1st... The starting attachment point represented in the token is also highlighted.

3.4 Encoding and Decoding

To encode a molecule in Group SELFIES, we first extract substructure matches of groups from the molecule. By default, this is done greedily by iterating through the grammar and extracting the largest groups first, but we can override this by specifying a priority for each group. We then traverse the graph as in the encoding process for SMILES and SELFIES, but additionally keep track of the attachment indices when coming in and out of a group.

When decoding Group SELFIES, the process is essentially the same as regular SELFIES except when reading group tokens. When a group token is read by the decoder, the grammar is used to lookup the corresponding group. Then, all atoms of the group are placed and the main chain is connected to the starting attachment point. We can then select the next attachment point we want to branch off of by using *relative indices*. By adding the current attachment index to a relative index modulo the total number of attachment points in the group, we can select the next attachment point. This relative index is specified by interpreting the next token as a number. From the specified attachment point, we implicitly branch off of the group, and we will need to "pop" using the [pop] token to return to the group and create further branches. If the selected attachment point is occupied, then the next available attachment point is used. If all attachment points have been used up, then the group is immediately "popped" and we return to the most recent branchpoint before the group.

We provide a detailed example of encoding/decoding the molecule celecoxib in Appendix A.1. The robustness of Group SELFIES was verified on 36M molecules from the eMolecules database [38]. We

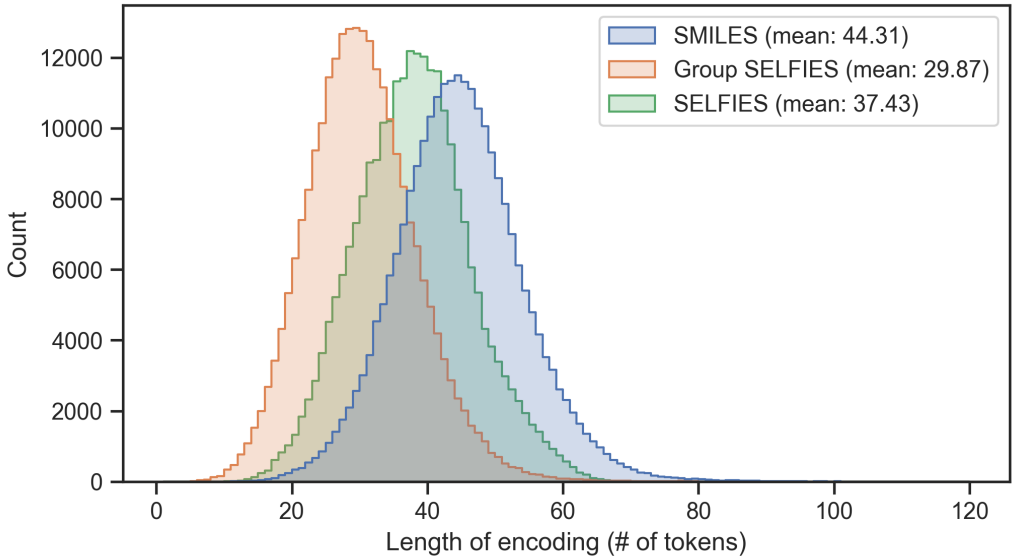

Figure 3: Histogram of lengths of SMILES, SELFIES, and Group SELFIES strings of the ZINC dataset. Here, Group SELFIES uses a grammar containing 53 groups.

provide an "essential set" of 23 groups which encode all relevant forms of chirality in the eMolecules database. Equipped with this essential set, every molecule can be encoded-decoded while maintaining chirality. It is also an option to not use the essential set, or only use a subset of it.

3.5 Getting fragments

In order to obtain a useful grammar from a set of molecules, we used various fragmentation algorithms, including a naïve technique that cleaves side chains from rings and a method based on MMPA (matched molecular pair analysis) [39]. Similar fragments are merged together to reduce redundancy. Several other fragmentation algorithms from cheminformatics can be readily applied [40–44], though we have not implemented them. Fragments can be specified using a SMILES-like syntax, which could be useful for choosing groups that are synthetically available to the user. Fragments can also be obtained from several fragment libraries found in the literature [45, 46].

4 Experiments

4.1 Group SELFIES is shorter

Group SELFIES strings are typically shorter than their SMILES and SELFIES equivalents when using a generic set of groups. In Figure 3, this generic set was generated by taking a random selection of 10,000 molecules from the ZINC dataset [47] and fragmenting them into 30 useful groups using various algorithms (see Getting fragments). We then combined these 30 groups with the 23 groups of the essential set. Figure 3 shows histograms of the lengths of SMILES/SELFIES/Group SELFIES strings of the entire ZINC dataset. Length is the number of characters in SMILES strings, and the number of tokens in (Group) SELFIES strings. Group SELFIES strings are usually shorter than their SELFIES and SMILES counterparts because group tokens can represent multiple atoms in a molecule.

4.2 Group SELFIES preserves useful properties during generation

To specifically compare the SELFIES and Group SELFIES representations, we use a primitive generative model which samples random strings. First, we convert a subset of $N = 100{,}000$ molecules from ZINC into (Group) SELFIES strings. Then, we tokenize all strings and combine

them into a single bag of tokens. To generate a new string, we first pick a random (Group) SELFIES string from our chosen subset and take its length l. We then randomly sample l tokens from the bag, and concatenate into a generated string. We generate N random strings for each representation. For Group SELFIES, we use the same 53 groups used for the length histogram, above.

We show histograms of the SAScore [48] and QED [49] of molecules generated from ZINC in Figure 4. The distributions of generated Group SELFIES more closely overlap with the original ZINC dataset than the generated SELFIES, showing that even with an extremely simplistic generative model, Group SELFIES can preserve important structural information. We perform a similar analysis for a dataset of nonfullerene acceptors (NFA) [50] in Appendix A.2.

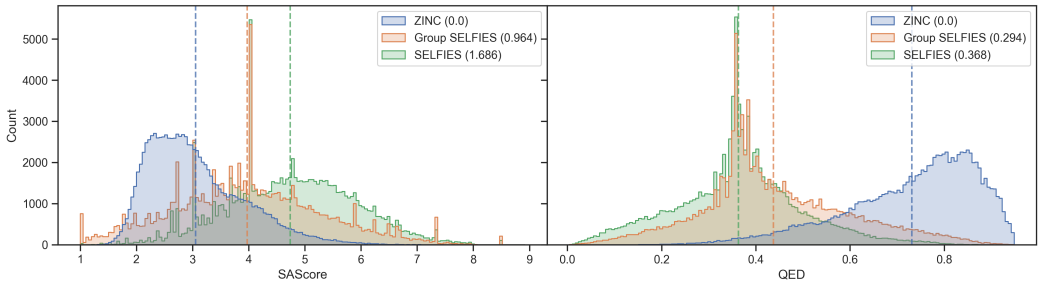

Figure 4: Molecules generated by our primitive generative model are binned by SAScore and QED. For both properties, the generated Group SELFIES have greater overlap with the original ZINC distribution. Bracketed values indicate the Wasserstein distance (a measure of overlap) to the ZINC distribution. Dashed lines indicate the means.

4.3 Group SELFIES improves distribution learning

To further quantify the effectiveness of Group SELFIES in generative models, the MOSES benchmarking framework [51] was used. Variational autoencoders (VAEs) based on a benchmark implementation from MOSES were trained with both Group SELFIES and SELFIES strings for 125 epochs. The grammar for the Group SELFIES implementation was created by fragmenting the training set and selecting the 300 most diverse groups. A set of 100,000 molecules was then generated from each model and evaluated on the metrics provided by MOSES.

Model	Valid (↑)	Unique@1k (↑)	Unique@10k (↑)	FCD (↓)		SNN (↑)	
				Test	TestSF	Test	TestSF
Train	*1.0*	*1.0*	*1.0*	*0.008*	*0.4755*	*0.6419*	*0.5859*
Group-VAE-125	**1.0(0)**	**1.0(0)**	0.9985(4)	**0.1787(29)**	**0.734(109)**	**0.6051(4)**	**0.5599(3)**
SELFIES-VAE-125	**1.0(0)**	0.9996(5)	**0.9986(4)**	0.6351(43)	1.3136(128)	0.6014(3)	0.5566(2)

Model	Frag (↑)		Scaf (↑)		IntDiv (↑)	IntDiv2 (↑)	Filters (↑)	Novelty (↑)
	Test	TestSF	Test	TestSF				
Train	*1.0*	*0.9986*	*0.9907*	*0.0*	*0.8567*	*0.8508*	*1.0*	*1.0*
Group-VAE-125	**0.9995(0)**	**0.9977(1)**	**0.9649(21)**	0.0608(65)	**0.8587(1)**	**0.8528(1)**	**0.9623(7)**	0.7187(11)
SELFIES-VAE-125	0.9989(0)	0.9965(1)	0.9588(15)	**0.0675(37)**	0.8579(1)	0.8519(1)	0.96(4)	**0.7345(16)**

Table 1: Group SELFIES VAE and SELFIES VAE evaluated on MOSES metrics.

For most metrics, Group SELFIES performs approximately the same as SELFIES. Validity is the percentage of generated molecules that are accepted by RDKit's parser. Uniqueness is the percentage of generated molecules that are not identical to any other generated molecule. Similarity to nearest neighbor (SNN) is the average Tanimoto similarity between generated molecules and the nearest neighbor in the reference set. Fragment similarity (Frag) is a cosine similarity based on the distribution of BRICS fragments [44] of generated and reference molecules. Scaffold similarity (Scaf) is a cosine similarity based on the distribution of Bemis-Murcko scaffolds [52] of generated and reference molecules. Internal diversity (IntDiv) measures the chemical diversity of the generated molecules using Tanimoto similarity. Filters is the fraction of generated molecules that pass filters for unwanted fragments. Novelty is the fraction of generated molecules not in the training set.

The Group SELFIES model performs especially well on the Fréchet ChemNet Distance (FCD) metric [53] when compared to SELFIES. FCD measures the difference between the activations of the penultimate layer of ChemNet (a model trained to predict the bioactivity of molecules) in the validation set and in the generated set. Due to how the model was trained, the activations are likely to encode a mixture of biological and chemical properties important to drug likelihood. This makes comparing these activations more informative than comparing standard properties like logP or molecular weight, where the correlation to bioactivity is weaker and less deliberate.

To visualize FCD, some indices of the penultimate activations of ChemNet are graphed in Figure 5. Generated Group SELFIES match these distributions more closely than generated SELFIES.

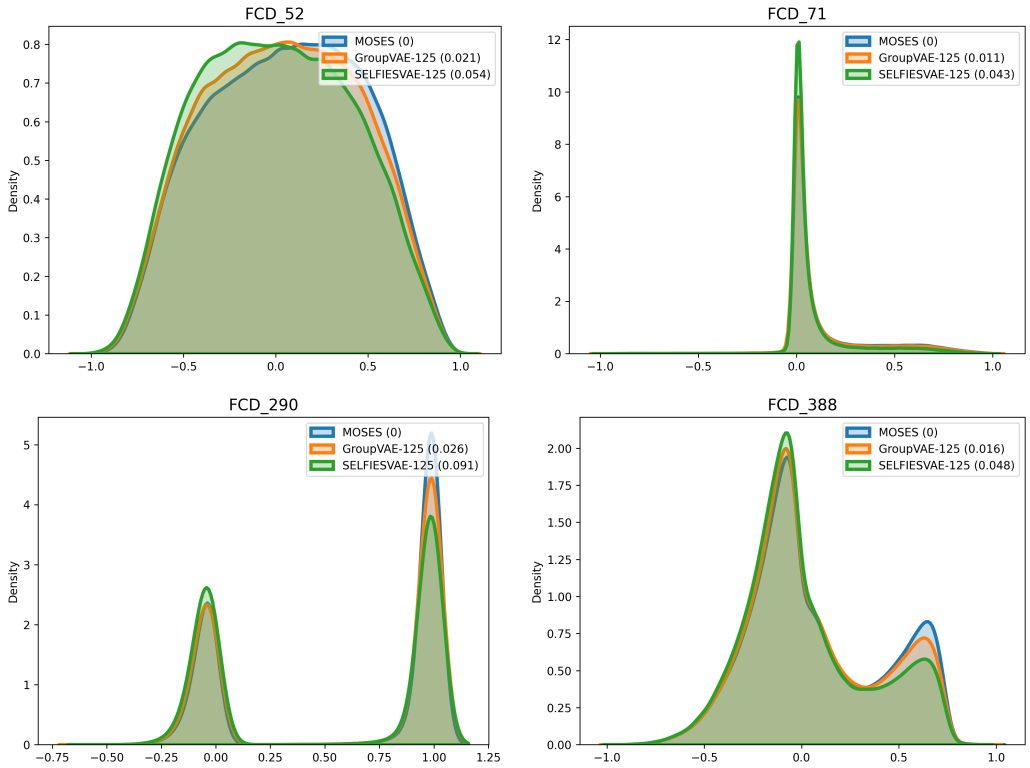

Figure 5: Distribution of some values from the second-to-last layer of ChemNet for molecules generated by Group SELFIES and SELFIES compared to the validation set. The difference in distributions is used to calculate FCD. Bracketed values in the legend represent the Wasserstein distance to the original MOSES distribution.

5 Discussion

Various techniques applied to SELFIES can mitigate the challenges of preserving structure. One example is to simply combine substrings of SELFIES strings together. Indeed, further experiments in Appendix A.3 show that simply replacing all group tokens by their SELFIES substrings shows similar performance to Group SELFIES. But in general, an inserted substring will not necessarily have that exact substructure when decoded. It is also likely that upon further insertions, that substructure will not be preserved. It is also not clear how this can create groups with 3 or more branches. Another way to enrich SELFIES is to generate many equivalent SELFIES strings that correspond to different traversals. This dramatically increases the number of generated molecules, improving diversity in genetic algorithms, as long as sample efficiency is not a concern.

Group SELFIES provides more fine-grained control of substructures. An important scaffold can be preserved during optimization; chiral and charged groups can be preserved during optimization, ensuring that charged tokens do not show up everywhere and generate many radicals; synthetically accessible building blocks can be chosen to improve synthesizability.

As a side note, Group SELFIES is theoretically capable of representing molecules which are traditionally not able to be represented with SMILES, such as ferrocene and molecules with axial chirality. This can be done by taking the entire ferrocene or axial chiral substructure and abstracting it into a group, leaving attachment points on the outside for varying functionalization. This is not implemented, however.

One tradeoff of Group SELFIES is that encoding and decoding is usually slower than with SELFIES, likely due to overhead with RDKit. The encoder is particularly slow as it relies on performing a substructure match for every group in the grammar. The decoder is faster than the encoder, though still about 10x slower than the SELFIES decoder. We believe the added substructure control of Group SELFIES will improve optimization overall, owing to increased sample efficiency.

While Group SELFIES does not use a formal grammar, DEG [22] does use a formal grammar, and complements this work: learned production rules can be added in as groups in Group SELFIES.

In this work we presented results with a primitive generative model. We leave the task of using other generative models with Group SELFIES to future work, such as combining JANUS [9] with Group SELFIES. Future work can also incorporate more flexibility into the grammar. For example, a group token can represent an entire scaffold, except without the atom identities. Other tokens can then identify the atom types on the scaffolds. This would allow optimization of the atom types while maintaining the topological structure of the scaffold. The groups in Group SELFIES also cannot overlap, making it unclear how to generate polycyclic compounds; more work is needed to develop a representation that acknowledges how groups might overlap. Cellular complexes might be a source of ideas here [54].

References

[1] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[2] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[3] Thomas Blaschke, Josep Arús-Pous, Hongming Chen, Christian Margreitter, Christian Tyrchan, Ola Engkvist, Kostas Papadopoulos, and Atanas Patronov. Reinvent 2.0: an ai tool for de novo drug design. *Journal of chemical information and modeling*, 60(12):5918–5922, 2020.

[4] Henry Moss, David Leslie, Daniel Beck, Javier Gonzalez, and Paul Rayson. Boss: Bayesian optimization over string spaces. *Advances in neural information processing systems*, 33: 15476–15486, 2020.

[5] Benjamin Sanchez-Lengeling, Carlos Outeiral, Gabriel L Guimaraes, and Alan Aspuru-Guzik. Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (organic). 2017.

[6] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

[7] AkshatKumar Nigam, Pascal Friederich, Mario Krenn, and Alán Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. *arXiv preprint arXiv:1909.11655*, 2019.

[8] AkshatKumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alan Aspuru-Guzik. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (stoned) algorithm for molecules using selfies. *Chemical science*, 12(20):7079–7090, 2021.

[9] AkshatKumar Nigam, Robert Pollice, and Alán Aspuru-Guzik. Parallel tempered genetic algorithm guided by deep neural networks for inverse molecular design. *Digital Discovery*, 2022.

[10] Daniel Flam-Shepherd, Kevin Zhu, and Alán Aspuru-Guzik. Language models can learn complex molecular distributions. *Nature Communications*, 13(1):1–10, 2022.

[11] Kohulan Rajan, Achim Zielesny, and Christoph Steinbeck. Decimer: towards deep learning for chemical image recognition. *Journal of Cheminformatics*, 12(1):1–9, 2020.

[12] Kohulan Rajan, Achim Zielesny, and Christoph Steinbeck. Stout: Smiles to iupac names using neural machine translation. *Journal of Cheminformatics*, 13(1):1–14, 2021.

[13] Nathan C Frey, Vijay Gadepally, and Bharath Ramsundar. Fastflows: Flow-based models for molecular graph generation. *arXiv preprint arXiv:2201.12419*, 2022.

[14] Geemi P Wellawatte, Aditi Seshadri, and Andrew D White. Model agnostic generation of counterfactual explanations for molecules. *Chemical science*, 13(13):3697–3705, 2022.

[15] Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

[16] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018.

[17] Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. Mars: Markov molecular sampling for multi-objective drug discovery. *arXiv preprint arXiv:2103.10432*, 2021.

[18] Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

[19] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures. In *International conference on machine learning*, pages 4849–4859. PMLR, 2020.

[20] Soojung Yang, Doyeong Hwang, Seul Lee, Seongok Ryu, and Sung Ju Hwang. Hit and lead discovery with explorative rl and fragment-based molecule generation. *Advances in Neural Information Processing Systems*, 34:7924–7936, 2021.

[21] Daniel Flam-Shepherd, Alexander Zhigalin, and Alán Aspuru-Guzik. Scalable fragment-based 3d molecular design with reinforcement learning. *arXiv preprint arXiv:2202.00658*, 2022.

[22] Minghao Guo, Veronika Thost, Beichen Li, Payel Das, Jie Chen, and Wojciech Matusik. Data-efficient graph grammar learning for molecular generation. *arXiv preprint arXiv:2203.08031*, 2022.

[23] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems*, 31, 2018.

[24] Pavel Polishchuk. Crem: chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics*, 12(1):1–18, 2020.

[25] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28 (1):31–36, 1988.

[26] Mario Krenn, Qianxiang Ai, Senja Barthel, Nessa Carson, Angelo Frei, Nathan C Frey, Pascal Friederich, Théophile Gaudin, Alberto Alexander Gayle, Kevin Maik Jablonka, et al. Selfies and the future of molecular string representations. *arXiv preprint arXiv:2204.00056*, 2022.

[27] WJ Wiswesser. Simplified chemical coding for automatic sorting and printing machinery. *Reading, Pa., Willson Products Inc*, 1951.

[28] William J Wiswesser. 107 years of line-formula notations (1861-1968). *Journal of Chemical Documentation*, 8(3):146–150, 1968.

[29] John J Vollmer. Wiswesser line notation: an introduction. *Journal of Chemical Education*, 60 (3):192, 1983.

[30] H Winston Hayward. *A new sequential enumeration and line formula notation system for organic compounds*. Number 21. Office of Research and Development, Patent Office, 1961.

[31] Herman Skolnik and Avis Clow. A notation system for indexing pesticides. *Journal of Chemical Documentation*, 4(4):221–227, 1964.

[32] R Webster Homer, Jon Swanson, Robert J Jilek, Tad Hurst, and Robert D Clark. Sybyl line notation (sln): a single notation to represent chemical structures, queries, reactions, and virtual libraries. *Journal of chemical information and modeling*, 48(12):2294–2307, 2008.

[33] Tianhong Zhang, Hongli Li, Hualin Xi, Robert V Stanton, and Sergio H Rotstein. Helm: a hierarchical notation language for complex biomolecule structure representation, 2012.

[34] Diego Garay-Ruiz, Carles Bo, and Diego Garay Ruiz. Human-readable smiles: Translating cheminformatics to chemistry. 2021.

[35] Xinhao Li and Denis Fourches. Smiles pair encoding: a data-driven substructure tokenization algorithm for deep learning. *Journal of Chemical Information and Modeling*, 61(4):1560–1569, 2021.

[36] John R Koza and Riccardo Poli. Genetic programming. In *Search methodologies*, pages 127–164. Springer, 2005.

[37] Markus Brameier, Wolfgang Banzhaf, and Wolfgang Banzhaf. *Linear genetic programming*, volume 1. Springer, 2007.

[38] Products: Structure data downloads. URL `https://search.emolecules.com/info/products-data-downloads.html`.

[39] Jameed Hussain and Ceara Rea. Computationally efficient algorithm to identify matched molecular pairs (mmps) in large data sets. *Journal of chemical information and modeling*, 50 (3):339–348, 2010.

[40] Chunquan Sheng and Wannian Zhang. Fragment informatics and computational fragment-based drug design: an overview and update. *Medicinal Research Reviews*, 33(3):554–598, 2013.

[41] Tairan Liu, Misagh Naderi, Chris Alvin, Supratik Mukhopadhyay, and Michal Brylinski. Break down in order to build up: decomposing small molecules for fragment-based drug design with e molfrag. *Journal of chemical information and modeling*, 57(4):627–631, 2017.

[42] Peter S Kutchukian, Sung-Sau So, Christian Fischer, and Chris L Waller. Fragment library design: Using cheminformatics and expert chemists to fill gaps in existing fragment libraries. In *Fragment-Based Methods in Drug Discovery*, pages 43–53. Springer, 2015.

[43] Simon Müller. Flexible heuristic algorithm for automatic molecule fragmentation: application to the unifac group contribution model. *Journal of cheminformatics*, 11(1):1–12, 2019.

[44] Jörg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using'drug-like'chemical fragment spaces. *ChemMedChem: Chemistry Enabling Drug Discovery*, 3(10):1503–1507, 2008.

[45] Peter Ertl and Tim Schuhmann. A systematic cheminformatics analysis of functional groups occurring in natural products. *Journal of natural products*, 82(5):1258–1263, 2019.

[46] Suliman Sharif, Ruibin Liu, Asuka A. Orr, Daniel Khavrutskii, Sunhwan Jo, Bettina Lier, Anastasia Croitoru, Chris Burke, Aziza Frank, Jacob Weiner, Nathaniel McClean, Aarion Romany, Mingtian Zhao, Takayuki Serizawa, Jared Deacon, Ian Jones, Shaoqi Zhan, Anmol Kumar, Mike Woster, Rebecca Pinette-Dorin, Elena Y. Chow, Sevien Schulhoff, and Alexander D. MacKerell Jr. Global-Chem: A Chemical Knowledge Graph of common small molecules and their IUPAC/SMILES/SMARTS for selection of compounds relevant to diverse chemical communities, 8 2022.

[47] John J Irwin, Khanh G Tang, Jennifer Young, Chinzorig Dandarchuluun, Benjamin R Wong, Munkhzul Khurelbaatar, Yurii S Moroz, John Mayfield, and Roger A Sayle. Zinc20—a free ultralarge-scale chemical database for ligand discovery. *Journal of chemical information and modeling*, 60(12):6065–6073, 2020.

[48] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):1–11, 2009.

[49] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

[50] Steven A Lopez, Benjamin Sanchez-Lengeling, Julio de Goes Soares, and Alán Aspuru-Guzik. Design principles and top non-fullerene acceptor candidates for organic photovoltaics. *Joule*, 1 (4):857–870, 2017.

[51] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alan Aspuru-Guzik, and Alex Zhavoronkov. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology*, 2020.

[52] Mark A Murcko and Guy W Bemis. The properties of known drugs. 1. *Molecular Frameworks. J. Med. Chem*, 39:2887–2893, 1996.

[53] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. 58 (9):1736–1741. ISSN 1549-9596. doi: 10.1021/acs.jcim.8b00234. URL https://doi.org/10.1021/acs.jcim.8b00234. Publisher: American Chemical Society.

[54] Mustafa Hajij, Kyle Istvan, and Ghada Zamzmi. Cell complex neural networks. *arXiv preprint arXiv:2010.00743*, 2020.