An Interpretability-augmented Genetic Expert for Deep Molecular Optimization

Pierre Wüthrich
Elix, Inc.
Tokyo, Japan
pierre.wuthrich@elix-inc.com

Jun Jin Choong
Elix, Inc.
Tokyo, Japan
junjin.choong@elix-inc.com

Shinya Yuki
Elix, Inc.
Tokyo, Japan
shinya.yuki@elix-inc.com

Abstract

The recently proposed Genetic expert guided learning (GEGL) framework has demonstrated impressive performances on several *de novo* molecular design tasks. Despite the displayed state-of-the art results, the proposed system relies on an expert-designed Genetic expert. Although hand-crafted experts allow to navigate the chemical space efficiently, designing such experts requires a significant amount of effort and might contain inherent biases which can potentially slow down convergence or even lead to sub-optimal solutions. In this research, we propose a novel genetic expert named *InFrag* which is free of design rules and can generate new molecules by combining promising molecular fragments. Fragments are obtained by using an additional graph convolutional neural network which computes attributions for each atom for a given molecule. Molecular substructures which contribute positively to the task score are kept and combined to propose novel molecules. We experimentally demonstrate that, within the GEGL framework, our proposed attribution-based genetic expert is either competitive or outperforms the original expert-designed genetic expert on goal-directed optimization tasks. When limiting the number of optimization rounds to one and three rounds, a performance increase of approximately 43% and 20% respectively is observed compared to the baseline genetic expert.

1 Introduction

The ability to discover and design *de novo* molecules with desired properties is of great interest in a multiple applications areas ranging from drug discovery [1] to materials engineering [2, 3]. This high-dimensional optimization task can be addressed via the inverse molecular design paradigm [2] which tries to find suitable candidate compounds given some target properties. This task is non-trivial considering the size of the molecular space; the drug-like chemical space alone is estimated to be about 10^{60} [4]. To tackle this challenge, one has to design an optimization system that is computationally tractable and feasible. At a higher level of abstraction, this optimization problem can be cast as a search/discovery problem. Discoveries in the chemical sciences can be broadly divided into three classes according to Coley et al. [5]. The first class of discoveries is related to physical matter, which encompasses most drug discovery or material engineering efforts where the final objective is to identify molecules with desired target properties. In the case of drug discovery, such compounds are potentially used as part of a therapeutic. The second class represents the discovery

NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications

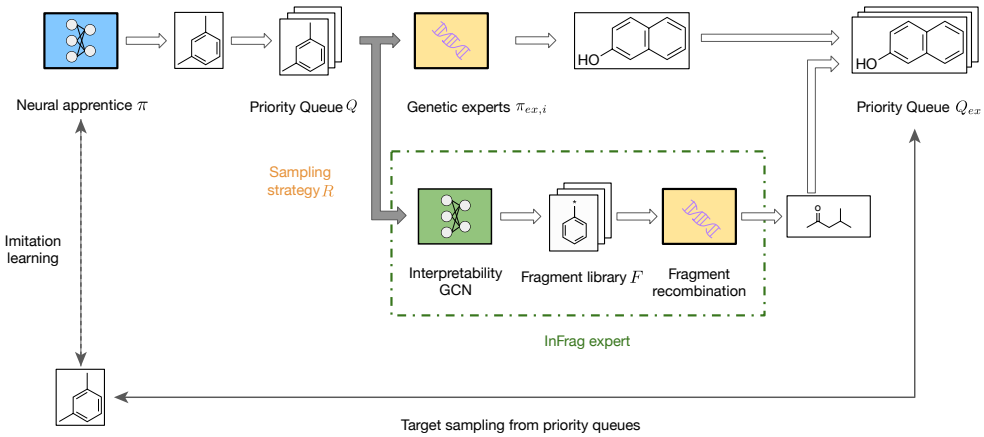

Figure 1: Complete depiction of the *eGEGL* framework. The original framework is enhanced with a fragment-based genetic expert denoted as *InFrag* (depicted in green dotted box). *InFrag* consists of a GCN model which creates attribution for queried molecules from which fragments are extracted. These fragments are then randomly recombined to obtain novel candidate molecules. In addition, we include the possibility to leverage several experts via a chosen sampling strategy *R* which determines how the allocate the sampling budget amongst all experts.

of better processes such as chemical reactions or conditions. Improvements in this area are seen as an important step to transition from an *in silico* to a real-world experimental phase. The final class contains the generative or predictive models which are the core engines driving the design choices behind the generation of *de novo* molecules.

We focus our efforts towards modeling *de novo* molecules. The process of seeking a *de novo* design can be broken into three distinct parts: (1) generating molecules, (2) scoring these molecules with a black-box fitness function, and (3) optimizing the generative model with respect to the computed scores [6]. Each one of these parts has its own set of challenges and pitfalls. For instance, the process of generating *de novo* molecules requires models that are capable of ensuring structural fidelity (i.e., generated molecules should be sensible) and structural validity (i.e., molecules are stable). The scoring function plays an important role in guiding the model during the optimization process. A trivial scoring function might explore undesired or obvious parts of the chemical space. In contrast, an over-complicated scoring function, in terms of imposed constraints, could cause instability during optimization as the model might not be able find suitable solutions. Finally, the optimization process should inherently reduce the training complexity from an enormous search space to circumvent the difficulty of an intractable computation problem.

2 Enhanced Genetic Expert Guided Learning

Ahn et al. [7] recently introduced a novel framework called *Genetic Expert Guided Learning* (GEGL) which combines meta-heuristic optimization with reinforcement learning-like optimization. GEGL has demonstrated strong capabilities and is at the time of writing the state-of-the-art in deep molecular optimization. The framework is composed of 4 distinct components: a neural apprentice, two reward priority queues and a genetic expert. We refer readers to the original work by Ahn et al. [7] for more details.

Building upon the existing GEGL framework, we propose an expert-free genetic expert called *InFrag* which is able to replace the original hand-crafted graph-based genetic expert[8]. *InFrag* first extracts fragments from arbitrary molecules based on the computed attribution for each involved atom. Fragments from high-scoring molecules contributing positively to the corresponding molecular scores are kept in memory and randomly recombined to propose novel candidate molecules. We note that the expert can be integrated into the GEGL framework by either replacing or, alternatively,

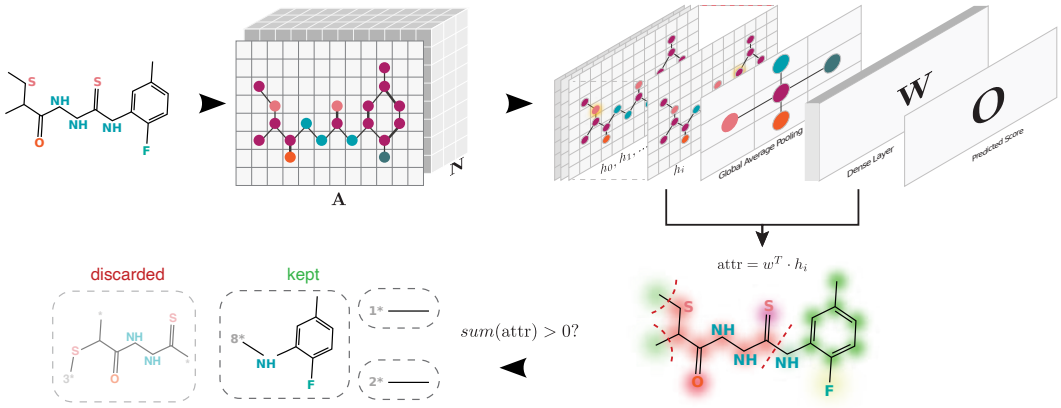

Figure 2: Generation of fragments using interpretability methods. A molecule is first processed to extract the adjacency matrix and node features. A graph convolutional network is used to calculate the last latent node embedding and final dense layer weights to compute the atom-wise class activation maps. The molecule is then fragmented along bonds where the sign of the attributions are opposite to each other. Only fragments contributing positively to the score, as determined by taking the sum over all involved atoms, are kept (depicted by the green arrow) and considered in further steps.

combined with other genetic experts. We denote the complete framework as *eGEGL* for *enhanced GEGL*. The extended framework is depicted in Figure 1.

2.1 Molecular Representation

Molecules are represented as undirected graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a total number of N nodes $v_i \in \mathcal{V}$ representing the atoms and edges between nodes denoted as $e_{i,j} \in \mathcal{E}$ representing the bonds. Furthermore, we extract the binary Adjacency matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$ from the edges. Each node is represented as a 74-dimensional vector $\boldsymbol{n_i} \in \mathbb{R}^{74}$ containing multiple distinct features.

2.2 Attribution-based Fragment Generation

Understanding which substructures of a molecule contributes positively towards its property score is crucial to guide design choices. Motivated by this insight, we propose to leverage an attribution-based model to reason between model input features and predicted scores to extract high-rewarding molecular substructures. However, it is generally impossible to compute the attributions from the scoring functions used to evaluate the fitness of molecules due to their black-box nature. To overcome this issue, we train a pseudo-scoring function model that is encouraged to imitate the original black-box scoring function of the optimization task, allowing us to obtain the desired attributions through this surrogate model. We implement a Graph Convolutional Network (GCN) [9] that takes the above-mentioned node features and adjacency matrix as inputs and is trained to predict task scores. The model is made of 5 blocks each with a graph convolutional layer, layer-normalization layer and a rectified linear unit (ReLU) activation layer. The loss is calculated as the mean squared error (MSE) between the 'true' task score as computed by the scoring function and the predicted score by our model. We pretrain the model on the same dataset as the neural apprentice in the original GEGL work.

To generate attributions, we use the Class Activation Maps (CAM) [10] method due to its simplicity and performance. More precisely, we compute the atom-wise attribution via

$$attribution(atom_i) = w_{out}^T \cdot h_i, \tag{1}$$

where w_{out} corresponds to the weights of the final dense layer of the GCN after the average pooling layer and h_i represents the latent node features of the node corresponding to atom i just before the pooling layer. Sanchez-Lengeling et al. [11] demonstrated that CAM is able to perform relatively well, especially when combined with GCN, compared to other interpretability methods in the context of graph neural networks [11]. We note that although we did not experiment with other

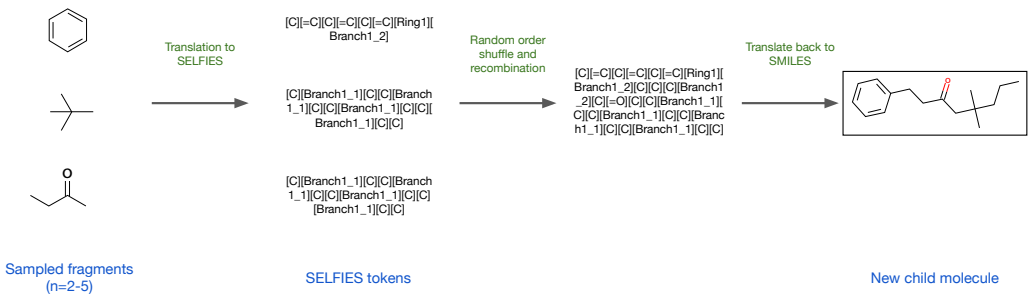

Figure 3: Recombination method for high-rewarding fragments. Fragments are sampled from the fragment library which contains high-rewarding fragments only. The fragments are first translated into their respective SELFIES representation, randomly shuffled and recombined. The final SELFIES token string is translated back to a SMILES representation. The resulting string is the new candidate child molecule resulting from the crossover operation. Please note that the proposed crossover rule is highly general and unbiased as it makes no assumptions about the used molecular representation.

attribution methods and graph neural networks, other methods and models can be used to generate the attributions.

Fragments of interest are identified by comparing the sign of the attribution for each atom involved in a given bond. In the case where the signs are found to be opposite, we fragment the molecule along that bond. To preserve complex substructures, we only consider and fragment single bonds and bonds which are not aromatic (as determined by RDKIT [12]). We then proceed to sum up the individual attributions of all the atoms in the fragment. If the sum is greater than 0.0, we consider the fragment to contribute positively to the predicted score and keep that fragment. Otherwise, the fragment is discarded.

Our objective is to utilize and recombine fragments from high-rewarding molecules only. After processing all of the queried molecules as described above, we end up with fragments contributing positively to the predicted score regardless of its value. In order to sort and only consider high-rewarding fragments, we associate each fragment with the true task score from the black-box scoring function of the molecule it originated from. All fragments are added to a data-structure called the fragment library. The fragment library allows to memorize the best scoring fragments and we impose that fragments in the library to be unique. When a new fragment is being added, we first check whether the fragment is already present in memory. We set the score associated with the fragment to the maximum score between the original in-memory fragment and the new duplicate fragment should it already be present. We apply selection pressure on the fragment library by limiting it's memory size to $1,000$ fragments. This operation allows to remove low-rewarding fragments and therefore ensures that only high-rewarding fragments are used during future optimization rounds. The complete process of fragment generation is graphically depicted in Figure 2.

Like the neural apprentice, the GCN model is trained every epoch such that it can reason over newly discovered and potentially better molecules.

2.3 Fragment recombination

To generate novel candidate molecules with *InFrag*, we first collect all fragments of the fragment library which are stored in SMILES format. New molecules are generated by randomly sampling 2 to 5 fragments and translating the sampled fragments into a SELFIES representation [13]. SELFIES representations posses the convenient property to always be valid, making their use very convenient for the fragment-crossover operation. The sampled fragments are randomly shuffled and combined to obtain a novel candidate SELFIES string from this fragment-crossover procedure. With some predefined probability, we furthermore apply a mutation operation on the obtained string in order to increase exploration and diversity of the generated molecules. Mutations include deletion, insertion or exchange of one of the SELFIES token on the currently operated string. Each type of mutation has the same probability of occurrence. This generative process is repeated until the number of desired molecules is reached. We note that the SELFIES representation makes use of an internal state when

translating from the conventional SMILES representation. This implies that one might require a more complex recombination strategy compared to the one described above to obtain better crossover molecules from the sampled fragments. We empirically found that the simple recombination strategy as described above performs competitively, regardless of the fact that the number of atoms might not necessarily be preserved during the crossover operation as illustrated in Figure 3. This representation-agnostic recombination strategy could be used in other problem settings since no assumptions are made about the task or the underlying representation. In addition, it does not require us to define any recombination rules and include expert knowledge on how fragments should relate to each other. This minimizes the potential for biases introduced by such expert rules.

3 Results and Discussion

3.1 Baselines and benchmarks

In this section, we report performances of the enhanced framework against the original framework. We also include comparisons to a baseline consisting of a simplified version of the STONED genetic operators proposed by Nigam et al. [14] which we will denote as *simplified-STONED* henceforth. More precisely, we limited the number of sampled chemical paths between any two parent molecules to a single one for this genetic expert. For other tasks, we include and compare against results obtained for selected baselines as reported in the original GEGL paper unless noted otherwise.

As for benchmarks, we follow the original GEGL work and compare the trained models on the penalized logp task and a subset goal-directed Guacamol benchmarks which we will describe in more details below. Furthermore, we used the same pretrained neural apprentice LSTM model as described in the original work.

Penalized LogP is a standard benchmark to evaluate *de novo* generative methods. The objective is to maximize the penalized octanol-water partition coefficient score defined as:

$$PenalizedLogP(x) = LogP(x) - SyntheticAccessibility(x) - RingPenalty(x) \quad (2)$$

where *LogP* is the unpenalized octanol-water partition coefficient[15], *SyntheticAccessibility* is a penalty term accounting for synthesizability [16] and *RingPenalty* is a penalty for rings with a size larger than 6. We further impose a constraint onto the generative model by limiting the number of SMILES characters to 81 following previous work.

Goal-directed Guacamol benchmarks are a set of 20 benchmarks proposed by Brown et al. [17] which were specifically designed for comparing generative models. The benchmarks evaluate a set of molecules to account for molecular diversity. More specifically and following the notation in Ahn et al.[7], the final benchmark score for a given molecule set X is computed as

$$Guacamol(X) \coloneqq \sum_{S \in Q} \sum_{s=1}^{S} \frac{r(x_{\Pi(s)})}{S|Q|} \text{ for } s = 1, ..., |X| - 1 \quad (3)$$

where Q is a list of integers and Π is a permutation function which ensures that the molecules $x \in X$ are sorted in descending order with respect to the evaluated property scores. The goal-directed benchmarks contain a variety of tasks such as *rediscovery*, *similarity* or *multi property optimization*. We refer readers to either the original Guacamol or GEGL papers for a more detailed description of each task type [17, 7]. For the tested tasks, we limit the number of characters to 100 for all generated SMILES strings.

3.2 Penalized logP task

Table 1 summarizes the mean and standard deviation comparison between several reported baselines and the enhanced GEGL framework we propose. We ran 5 independent experimental runs for each of our models where each round consists of 200 optimization rounds following previous work. The results demonstrate that there is no gain or decrease in performance by swapping out the genetic expert. This is remarkable because it indicates that expert-designed genetic experts can potentially be replaced with simpler and bias-free genetic-experts without loss in performance.

Table 1: PenalizedLogP results. Results are displayed as mean and standard deviation.

Algorithm	Objective
JT-VAE[18]	4.90 ± 0.33
ChemTS[19]	5.60 ± 0.50
GCPN[20]	7.86 ± 0.07
GB-GA[8]	15.76 ± 5.76
DA-GA[21]	20.72 ± 3.14
GEGL (GB-GA expert)[7]	$\mathbf{31.40 \pm 0.00}$
GEGL (simplified-STONED expert)	$\mathbf{31.40 \pm 0.00}$
GEGL (InFrag expert)	$\mathbf{31.40 \pm 0.00}$

We furthermore note that the *InFrag* genetic expert is able to extract useful information from the pretraining phase of the GCN which can be leveraged from the start of the optimization process. To demonstrate this, we additionally compared the different genetic experts in the setting where evaluating candidate molecules is assumed to be undesirable due to the required additional efforts. Concretely, we limit the optimization process to a single and three optimization rounds respectively and evaluate the generated molecules. As can be seen in Table 2, our *InFrag* genetic expert is able to generate higher-rewarding molecules in this setting, even surpassing the *JT-VAE* and *ChemTS* baselines as can be seen from Table 1 and achieving similar results to the reinforcement learning based *GCPN* method. These results indicate that our genetic expert is appropriate in settings where obtaining *additional* labels for novel candidate molecules is expensive or difficult, as is usually the case in real-world applications. The obtained results are to be expected as the GCN is able to leverage the learned representations from the pretraining phase and therefore can reason and propose high-rewarding novel molecular candidates from the beginning of the optimization process.

Table 2: One- and Three-round PenalizedLogP results. Entries are displayed as mean and standard deviation.

Algorithm	1-round optimization	3-round optimization
GEGL (GB-GA expert)[7]	5.44 ± 0.17	7.92 ± 0.42
GEGL (simplified-STONED expert)	4.29 ± 0.14	6.82 ± 0.25
GEGL (InFrag expert)	$\mathbf{7.81 \pm 0.61}$	$\mathbf{9.50 \pm 0.24}$
Improvement over baseline	$\mathbf{43.6\%}$	$\mathbf{19.9\%}$

3.3 Goal-directed guacamol benchmarks

Finally, we evaluated the proposed enhancements and genetic expert on a subset of the goal-directed benchmark tasks proposed in the Guacamol benchmark. More precisely, we excluded all *Rediscovery* and all *Similarity* tasks for which multiple baselines have been shown to achieve a perfect score of 1.0. Table 3 summarizes the obtained experimental results.

We observe that the genetic experts can be exchanged with each other without significant loss or increase in performance. Furthermore, our empirical results show that no expert dominates across all of the tested tasks. This indicates that combining and sampling from several experts in parallel, with the objective to leverage the best expert amongst all experts, could potentially lead to better overall results.

4 Conclusions

We have shown in this work that it is possible to pretrain and utilize an attribution-based genetic expert to propose novel molecules. The expert we called *InFrag* is able to leverage a pretraining phase to reason over the potential score of candidate molecules and produces atom-wise attributions to generate high-rewarding fragments. Our genetic expert which is capable of expert- and bias-free

Table 3: Results for the goal-directed Guacamol benchmarks for different genetic experts leveraging the GEGL framework. We show selected baselines for comparison as reported by the original GEGL work[7].

	Baselines			GEGL-based models		
Task	GB-GA[8]	MSO[22]	CReM[23]	GB-GA[7]	simplified-STONED	InFrag
C11H24	0.971	0.997	0.966	**1.000**	**1.000**	**1.000**
C9H10N2O2PF2Cl	0.982	**1.000**	0.940	**1.000**	**1.000**	**1.000**
Median molecules 1	0.406	0.437	0.371	**0.455**	**0.455**	**0.455**
Median molecules 2	0.432	0.395	0.434	**0.437**	0.419	0.427
Osimertinib MPO	0.953	0.966	0.995	**1.000**	**1.000**	**1.000**
Fexofenadine MPO	0.998	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Ranolazine MPO	0.920	0.931	0.969	0.958	**0.981**	0.959
Perindopril MPO	0.792	0.834	0.815	0.882	**0.886**	0.882
Amlodipine MPO	0.894	0.900	0.902	**0.924**	0.905	0.905
Sitagliptin MPO	0.891	0.868	0.763	0.922	**0.958**	0.952
Zaleplon MPO	0.754	0.764	0.770	0.834	**0.840**	**0.840**
Valsartan SMARTS	0.990	0.994	0.994	**1.000**	**1.000**	**1.000**
Deco Hop	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Scaffold Hop	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

fragment-level crossover operations is able to produce high-rewarding molecules and performs comparable to other genetic experts when embedded into the GEGL framework. Furthermore, we demonstrated that *InFrag* significantly outperforms other genetic experts when the number of optimization rounds is limited. This implies that *InFrag* is a good choice for real-world optimization cases were evaluation of novel generated molecules might be difficult or expensive. We have not yet experimented with different genetic experts, attribution methods, graph neural network architectures or sampling strategies, leaving a potentially large room for improvement. We leave this area of research for future work.

References

[1] Gisbert Schneider. Automating drug discovery. *Nature Reviews Drug Discovery*, 17(2):97–113, February 2018. ISSN 1474-1784. doi: 10.1038/nrd.2017.232.

[2] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, July 2018. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aat2663.

[3] Robert Pollice, Gabriel dos Passos Gomes, Matteo Aldeghi, Riley J. Hickman, Mario Krenn, Cyrille Lavigne, Michael Lindner-D'Addario, AkshatKumar Nigam, Cher Tian Ser, Zhenpeng Yao, and Alán Aspuru-Guzik. Data-Driven Strategies for Accelerated Materials Design. *Accounts of Chemical Research*, 54(4):849–860, February 2021. ISSN 0001-4842. doi: 10.1021/acs.accounts.0c00785.

[4] Regine S. Bohacek, Colin McMartin, and Wayne C. Guida. The art and practice of structure-based drug design: A molecular modeling perspective. *Medicinal Research Reviews*, 16(1): 3–50, 1996. ISSN 1098-1128. doi: 10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6.

[5] Connor W. Coley, Natalie S. Eyke, and Klavs F. Jensen. Autonomous Discovery in the Chemical Sciences Part I: Progress. *Angewandte Chemie International Edition*, 59(51):22858–22893, 2020. ISSN 1521-3773. doi: 10.1002/anie.201909987.

[6] Markus Hartenfeller and Gisbert Schneider. Enabling future drug discovery by de novo design. *WIREs Computational Molecular Science*, 1(5):742–759, 2011. ISSN 1759-0884. doi: 10.1002/wcms.49.

[7] Sung-Soo Ahn, Junsu Kim, Hankook Lee, and Jinwoo Shin. Guiding Deep Molecular Optimization with Genetic Exploration. *Advances in Neural Information Processing Systems*, 33: 12008–12021, 2020.

[8] 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, March 2019. ISSN 2041-6539. doi: 10.1039/C8SC05372C.

[9] Thomas N. Kipf and Max Welling. Semi-supervised Classification with Graph Convolutional Networks. In *Proceedings on the 5th International Conference on Learning Representations*, ICLR '17, 2017.

[10] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[11] Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Wang, Wesley Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. Evaluating Attribution for Graph Neural Networks. *Advances in Neural Information Processing Systems*, 33:5898–5910, 2020.

[12] Greg Landrum, Paolo Tosco, Brian Kelley, Sriniker, Ric, Gedeck, Riccardo Vianello, NadineSchneider, Andrew Dalke, Dan N, Eisuke Kawashima, Brian Cole, Samo Turk, Matt Swain, AlexanderSavelyev, David Cosgrove, Alain Vaucher, Maciej Wójcikowski, Daniel Probst, Guillaume Godin, Gareth Jones, Vincent F. Scalfani, Axel Pahl, Francois Berenger, JLVarjo, Strets123, JP, DoliathGavid, Gianluca Sforna, and Jan Holst Jensen. rdkit/rdkit: 2020_09_5 (q3 2020) release, 2020.

[13] 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, November 2020. ISSN 2632-2153. doi: 10.1088/2632-2153/aba947.

[14] AkshatKumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alán Aspuru-Guzik. Beyond generative models: Superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. *Chemical Science*, 12(20):7079–7090, May 2021. ISSN 2041-6539. doi: 10.1039/D1SC00231G.

[15] 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.

[16] 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):8, June 2009. ISSN 1758-2946. doi: 10.1186/1758-2946-1-8.

[17] Nathan Brown, Marco Fiscato, Marwin H.S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking Models for de Novo Molecular Design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, March 2019. ISSN 1549-9596. doi: 10.1021/acs.jcim.8b00839.

[18] 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, July 2018.

[19] Xiufeng Yang, Jinzhe Zhang, Kazuki Yoshizoe, Kei Terayama, and Koji Tsuda. ChemTS: An efficient python library for de novo molecular generation. *Science and Technology of Advanced Materials*, 18(1):972–976, December 2017. ISSN 1468-6996. doi: 10.1080/14686996.2017.1401424.

[20] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[21] AkshatKumar Nigam, Pascal Friederich, Mario Krenn, and Alán Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space, 2020.

[22] Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noé, and Djork-Arné Clevert. Efficient multi-objective molecular optimization in a continuous latent space. *Chem. Sci.*, 10:8016–8024, 2019. doi: 10.1039/C9SC01928F.

[23] Pavel Polishchuk. CReM: Chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics*, 12(1):28, April 2020. ISSN 1758-2946. doi: 10.1186/s13321-020-00431-w.