MOLECULAR FRAGMENT-BASED DIFFUSION MODEL FOR DRUG DISCOVERY

Daniel Levy
School of Computer Science
McGill University
Mila - Quebec Artificial Intelligence Institute
daniel.levy@mila.quebec

Jarrid Rector-Brooks
DIRO
Universite de Montreal
Mila - Quebec Artificial Intelligence Institute
jarrid.rector-brooks@mila.quebec

ABSTRACT

Due to the recent successes of generative models much attention has been paid to *de novo* generation of drug-like molecules using machine learning. A particular class of generative models, diffusion probabilistic models, have recently been shown to work extraordinarily well across a diverse set of generative tasks, and a growing body of literature has applied diffusion probabilistic models directly to the molecule discovery problem. However, existing methods work with atom-based molecule representations, whereas work in the fragment-based drug design community indicates that using a molecular fragment-based approach can provide a much better inductive bias for the generative model. To this end, in our work we attempt to use diffusion probabilistic models to *de novo* generate drug-like molecules with a fragment-based representation, yielding more valid and drug-like molecules than existing approaches.

1 INTRODUCTION

It is extremely expensive and time-consuming to produce new drugs, with some estimates placing the time required at 10 years and cost at over $1 billion (Wouters et al., 2020). By training on vast datasets of existing drugs with known properties, machine learning tools can be used to filter molecules from a database to search for promising candidates for certain applications, and they can also be used to find minor modifications to existing molecules to improve their utility. However, both approaches can be severely limiting. The space of possible chemicals is very large – it has been estimated that there are somewhere between 10^{20} and 10^{60} possible drug-like chemicals, but only 10^8 have ever been created (Bilodeau et al., 2022). Therefore, rather than examining existing molecules, it might be more promising to use machine learning tools in order to generate entirely new candidate molecules *de novo* and target generation towards desirable properties.

Despite promising early results in *de novo* generation, this field is still in its infancy. Even when a machine learning model proports to be able to generate realistic drugs, two major issues persist: the uniqueness of the molecules and their practicality in real-world settings. Many proposed drugs generated from existing machine learning models are strikingly similar to molecules in their training dataset and would not meet the standards of novelty expected in the field of chemistry (Walters & Murcko, 2020). Further, many top machine learning models produce drugs that would be impossible to synthesize by a real chemist – Gao & Coley (2020) found that of the top 100 candidate molecules suggested by certain machine learning models, none of them were practically synthesizable.

Recently, a new class of models known as denoising diffusion probabilistic models have demonstrated success in a broad range of machine learning tasks. This technique works using two processes: a forward noising process, and a reverse denoising process. The noising process sequentially adds noise to the input data until it is transformed into pure noise. A denoising neural network is then trained to take a noisy sample and predict how to undo the noise. If this network is then fed a sample of pure noise, it can denoise it to generate an entirely novel sample. This technique has been explored as a possible avenue for drug discovery with very promising early results in terms of the realism and diversity of the generated molecules (Xu et al., 2022; Igashov et al., 2022; Hoogeboom et al., 2022). However, because these methods generate molecules at the level of individual atoms, they are

limited to small molecules, and they must be extensively trained just to be able to learn to produce realistic-looking molecules, let alone ones that have desirable properties as drugs. Furthermore, there is no indication that generated molecules are synthesizable.

To improve synthesizability while still allowing the model to efficiently generate varied and novel molecules, we propose to create a machine learning model that generates molecules by diffusing over common molecular fragments, rather than individual atoms. The diffusion process is over two sets of variables: the identities of fragments in the generated molecules, and the connections between these fragments. By bypassing the generation of these lower-level fragments, we expect this model to be more efficient in generating large molecules. Furthermore, the generated molecules are expected to be more synthesizable, as they are created from fragments of already-existing drugs. While a fragment-based approach has shown success with methods such as reinforcement learning (Flam-Shepherd et al., 2022; Yang et al., 2021; Gottipati et al., 2020b) and variational autoencoders (Kong et al.; Jin et al., 2018), to our knowledge this is the first application of a diffusion model to molecular fragments.

2 Related Work

To apply deep learning to the task of molecule generation, a good representation of a molecule is needed. There have generally been three ways to represent molecules, which can be thought of as the dimensionality of the representation: 1D, 2D, or 3D.

Existing work in this field has typically represented molecules as SMILES strings, a single string that encodes the connectivity of constituent atoms. While SMILES strings allow researchers to leverage neural networks that have classically been used for language models, they have the downside of requiring that any machine learning model that uses it would have to learn the SMILES grammar, and would find it hard to get an overview of the structure of a molecule. Furthermore, it is not a unique representation: two very different SMILES strings could represent the exact same molecule. Examples include Olivecrona et al. (2017) which uses Deep RL to generate SMILES strings, Gupta et al. (2018), which uses a generative RNN, and Honda et al. (2019) which uses a transformer architecture.

More recent machine learning models have represented molecules as graphs, with atoms represented by nodes and bonds represented by edges. Not only is this far closer to how actual chemists conceive of molecules, but it has also allowed researchers to apply recently developed highly expressive graph neural networks that are able to recognize structures important to the properties of molecules. However, generative models for graphs must deal with the permutation invariance property and the discrete nature of graphs, which can make their implementations more complicated. Examples include MolGAN (De Cao & Kipf, 2018) and Graph VAE (Liu et al., 2018), as well as Jo et al. (2022) and Vignac et al. (2022) which both use diffusion models on graphs to generate molecules.

As molecules are physical objects, it can be more realistic to describe them using the 3-dimensional coordinates of their constituent atoms. Two atoms that may appear far away in a graph representation might actually be physically very close in real space, which can strongly affect the molecule's physical properties. However, there is usually not a single unique 3-dimensional representation of a molecule. Atomic bonds can be rotated, so each molecule will actually exist in a distribution of different *conformations*, depending on these bond angles. Generative models that work with coordinates must be equivariant to rotations and translations: if the whole coordinate frame were to be rotated or translated, then the output of any model would need to be rotated or translated in the same way, as we should be entirely indifferent to the choice of coordinates. These models therefore use equivariant neural networks as their backbone (Han et al., 2022). An example of this approach is Satorras et al. (2021), which uses normalizing flows, Hoogeboom et al. (2022), Xu et al. (2022), and Igashov et al. (2022), which apply denoising diffusion models to atomic coordinates.

Working at the level of molecular fragments rather than individual atoms has proven to be a good way to bypass lower-level generation and to produce more synthesizable molecules. Methods that use this technique include Jin et al. (2018) and Kong et al., which use VAEs to build molecular graphs from subgraphs, and Gottipati et al. (2020a) and Yang et al. (2021), which use reinforcement learning to produce molecules one fragment at a time.

For a survey of the use of generative models for molecule generation, see Bilodeau et al. (2022) and Du et al. (2022).

3 DISCRETE DENOISING DIFFUSION

Denoising diffusion models are a recently developed class of highly effective generative models (Ho et al., 2020; Sohl-Dickstein et al., 2015). For these models, we start with a dataset of samples x^0, and our aim is to learn a distribution $p_\theta(x^0)$ over these samples. The superscript 0 is to denote the "timestep" of the sample: it is initially 0 for unperturbed data, and it is T for pure noise.

To learn $p_\theta(x^0)$, we use two processes. The first is a noising process $q(x^t|x^{t-1})$, that takes in data at some timestep $t-1$, and applies noise from a predefined distribution to perturb x^{t-1} and produce a distribution of noised data x^t. The second is the denoising process $p_\theta(x^{t-1}|x^t)$, which depends on the parameters θ of a neural network. This process learns to "undo" the noise. We can use $q(x^t|x^{t-1})$ to generate noisy data x^t from real samples x^0, and then train the denoiser $p_\theta(x^{t-1}|x^t)$ by minimizing a loss function. With a trained $p_\theta(x^{t-1}|x^t)$, we can generate realistic-looking samples by taking pure noise $\hat{x}^T$ and progressively denoising it using until we have a generated sample $\hat{x}^0$.

Instead of sampling a full trajectory $(x^0, x^1, \dots x^T)$, modern denoising diffusion models are trained by sampling a random t, sampling x^t, and directly predicting x^0 (Ho et al., 2020). This greatly speeds up and stabilizes training, but it requires that the noising process q has certain properties (Vignac et al., 2022):

1. $q(x^t|x^0)$ should be very simple to compute so that training samples x^t can be easily obtained.
2. $q(x^{t-1}|x^t, x^0)$ should be simple so that when generating a sample, we can use our model's prediction of x^0 to predict x^{t-1} given x^t.
3. $q(x^T|x^0)$ should not depend on x^0 as $T \to \infty$, so that it can be used as a prior distribution for sampling x^T when generating samples.

In most diffusion models, these conditions are fulfilled by Gaussian noise. However, when working with categorical features, other types of noise can be used. In Hoogeboom et al. (2021) and Austin et al. (2021), it is shown that transition matrices can fulfill all of these conditions. A transition matrix $\boldsymbol{Q}^t$ is defined by $\boldsymbol{Q}^t_{ij} = q(x^t = j|x^{t-1} = i)$: that is, each entry denotes the probability of transitioning from one state to another. For discrete diffusion operating on one-hot vectors $\boldsymbol{x}$, we define our noising process $q(x^t|x^0) = \mathrm{Cat}(x^t; p = \boldsymbol{x}^{t-1}\boldsymbol{Q}^t)$ where Cat is the categorical distribution. To fulfill the first condition, transition matrices can be multiplied together as $\bar{\boldsymbol{Q}}^t = \boldsymbol{Q}^1\boldsymbol{Q}^2 \dots \boldsymbol{Q}^t$ to skip across multiple timesteps. To fulfill the second condition, the following closed-form expression can be derived:

$$q(x^{t-1}|x^t, x^0) = \frac{q(x^t|x^{t-1}, x^0)q(x^{t-1}|x^0)}{q(x^t|x^0)} = \mathrm{Cat}\left(x^{t-1}; p = \frac{\boldsymbol{x}^t\boldsymbol{Q}^{t\top} \odot \boldsymbol{x}^0\bar{\boldsymbol{Q}}^{t-1}}{\boldsymbol{x}^0\bar{\boldsymbol{Q}}^t\boldsymbol{x}^{t\top}}\right) \quad (1)$$

We can also define the transition matrix so that $\bar{\boldsymbol{Q}}^T$ converges to a given distribution as $T \to \infty$, fulfilling the third condition.

DiGress (Vignac et al., 2022) uses this discrete denoising diffusion approach to generate realistic-looking graphs. They do so by independently applying noise to each node and each possible edge between nodes, and using a graph neural network as their denoising network. We base our work on their implementation, but adapted to work on molecular fragments. Our architecture is described in the next section.

4 FRAGMENT-BASED DISCRETE DIFFUSION

4.1 FRAGMENT REPRESENTATION

In this work, we apply discrete denoising diffusion models to fragment-based representations of molecules. We choose to represent molecules and their fragments as graphs because they are easier to optimize, don't require conformation information in the training data, and because graphs are the chemical structure representations that are needed to actually synthesize a molecule.

To work on the level of fragments rather than individual atoms, we must develop a one-to-one mapping between molecular graphs and their fragment representation, a nontrivial task. First, we define an atom-level molecular graph $\boldsymbol{H} = (\boldsymbol{A}, \boldsymbol{B})$ where $\boldsymbol{A} \in \mathbb{R}^{n \times d_a}$ represents n atoms with a one-hot encoding of d_a possible atom types, and $\boldsymbol{B} \in \mathbb{R}^{n \times n \times d_b}$ represents bonds between atoms with a one-hot encoding of $d_b - 1$ possible bond types, or a one-hot encoding indicating a lack of bond between two atoms.

We can define these same molecules as fragment graphs, $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{E})$. The matrix $\boldsymbol{X} \in \mathbb{R}^{N \times d_x}$ represents N fragments with one-hot encodings for d_x possible fragment types. Each node $x_i \in \boldsymbol{X}$ is equivalent to its own smaller molecular graph with n_i nodes, which we can denote with $\boldsymbol{H}(x_i) = (\boldsymbol{A}(x_i), \boldsymbol{B}(x_i))$. The tensor $\boldsymbol{E} \in \mathbb{R}^{n \times n \times d_e}$ represent $d_e - 1$ possible connection between fragments, or a lack of connection between fragments. These connections are defined by their attachment points in the underlying atomic graphs of the respective fragments. For example, $e_{1,2,3} = 1$ could mean that fragment x_1 and x_2 are connected via bond type 2 (a double bond) at node $A(x_i)_4$ and node $A(x_j)_5$ in their respective graphs. The fact that edge type 3 between fragment types $\boldsymbol{H}(x_1)$ and $\boldsymbol{H}(x_2)$ denotes a double bond between their 4th and 5th atoms is precomputed in a table mapping (frag_id, frag_id, edge_type) to (atom_id, atom_id, bond_type). A diagram showing the connection between the fragment-based representation and the atomic representation is shown in Figure 1.

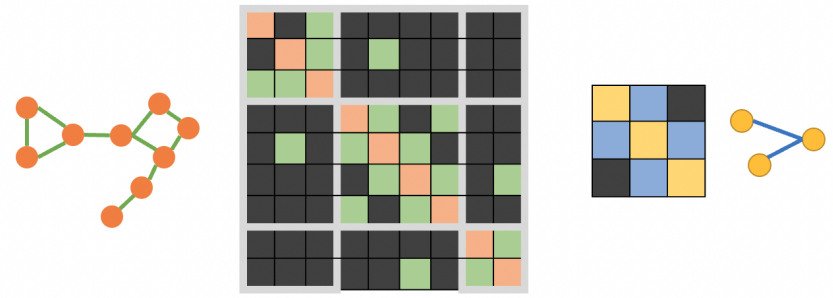

Figure 1: Four representations of the same molecule: its atom-level graph, its atom-level adjacency matrix, its fragment-level adjacency matrix, and its fragment-level graph. Because atom-level adjacency matrices naturally factor into a small set of blocks (fragments) with sparse connections between them, we hypothesize that this is a simpler training objective for the diffusion model to learn.

4.2 Graph Diffusion

The discrete denoising diffusion model we apply to our fragment-based graphs is identical to what is used by DiGress (Vignac et al., 2022), so we describe it here.

To generate noisy samples $\boldsymbol{G}^t$, we sample a timestep t and apply a cumulative transition matrix $\bar{\boldsymbol{Q}}^t$ to each fragment node in $\boldsymbol{X}$ and each possible edge in $\boldsymbol{E}$. The transition matrices are of the form:

$$\bar{\boldsymbol{Q}}^t_X = \bar{\alpha}^t \boldsymbol{I} + \bar{\beta}^t \mathbf{1}_{d_x} \boldsymbol{m}_X \quad \text{and} \quad \bar{\boldsymbol{Q}}^t_E = \bar{\alpha}^t \boldsymbol{I} + \bar{\beta}^t \mathbf{1}_{d_e} \boldsymbol{m}_E \tag{2}$$

Here, $\bar{\alpha}^t$ and $\bar{\beta}^t$ are time-dependent scheduling variables, and $\boldsymbol{m}_X$ and $\boldsymbol{m}_E$ are the marginal probabilities of each node and edge type respectively, taken from the data. A cosine schedule (Nichol & Dhariwal, 2021) is used to adjust $\bar{\alpha}$ and $\bar{\beta}$ over time, such that when $t = 0$, $\bar{\alpha}^0 = 1$ and $\bar{\beta}^0 = 0$, and the graph does not change at all, and when $t = T$, $\bar{\alpha}^T = 0$ and $\bar{\beta}^T = 1$, the probability for a node or edge to transition to a given state is given entirely by the marginal probability of that state.

The denoising process is governed by a graph transformer neural network (Dwivedi & Bresson, 2020) that takes in a noised graph $\boldsymbol{G}^t$ and predicts the unperturbed graph $\hat{\boldsymbol{G}}^0$. The denoising network is trained using a simple cross-entropy loss comparing the predicted distribution of fragment types $\hat{p}^0_X$ and connection types $\hat{p}^0_E$ to the true values $\boldsymbol{X}^0$ and $\boldsymbol{E}^0$, with their contributions weighed by a hyperparameter λ:

$$\mathcal{L}\left((\hat{p}_X{}^0, \hat{p}_E^0), \boldsymbol{X}^0\right) = \sum_{i=1}^{n} \text{cross-entropy}(\hat{p}_{Xi}^0, X_i^0) + \lambda \sum_{i\leq 1, j\leq n} \text{cross-entropy}(\hat{p}_{Ei,j}^0, E_{i,j}^0) \quad (3)$$

At generation time, a random fragment graph is initialized with the number of fragments sampled according to the training distribution of fragment counts, and the fragment types and edges are sampled according to the training distribution marginal probabilities of each fragment and edge type, $\boldsymbol{m}_X$ and $\boldsymbol{m}_E$. This yields a random fragment graph $\hat{\boldsymbol{G}}^{t=T}$. The graph transformer predicts $\tilde{\boldsymbol{G}}^0$, given $\hat{\boldsymbol{G}}^t$. Next, using the posterior formula $p_\theta(\boldsymbol{G}^{t-1}|\boldsymbol{G}^t, \boldsymbol{G}^0)$ given by Equation 1 and marginalizing over all possible values of x and e, a partially-denoised $\hat{\boldsymbol{G}}^{t-1}$ is sampled given $\tilde{\boldsymbol{G}}^0$ and $\hat{\boldsymbol{G}}^t$. This process is repeated until a molecule is generated, $\hat{\boldsymbol{G}}^0$. Once a fragment-based representation of a molecule is generated, using a precomputed library of correspondences between fragment types and atomic graphs and fragment connection types and connecting atoms, we can directly translate our molecules into their atom-based representation.

5 Experiments

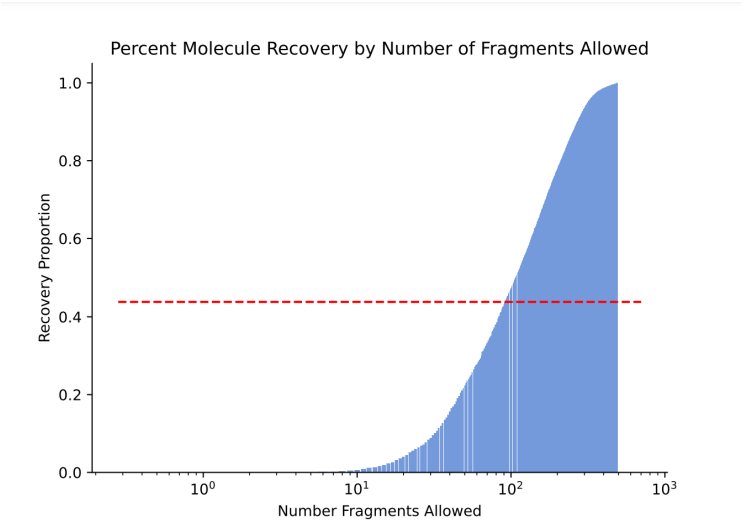

Figure 2: The proportion of the initial 4.7 million molecules which can be constructed using only the top k most frequently seen fragments (k on the x-axis). The dashed red line denotes the recovery proportion we used in practice by selecting the top 100 fragments.

5.1 Dataset

As no publicly available dataset of molecules in a fragment-based representation exists, we instead generated our own molecular fragment dataset from a subset of the ZINC15 (Irwin & Shoichet, 2005; Irwin et al., 2012) dataset. ZINC15 is a large database consisting of over 230 million purchasable compounds. We selected the drug-like molecules currently in-stock to purchase in order to restrict our dataset size, resulting in a set of 4.7 million molecules.

Next, we broke each molecule into its constituent fragments using the Breaking Retrosynthetically Interesting Chemical Substructures (BRICS) algorithm (Degen et al., 2008) (implemented in RDKit (Landrum)). After fragmenting all molecules, we measured how frequently each fragment was seen across the dataset. In order to keep the number of node types the diffusion model must consider to a relatively small number, we selected only the top 100 most frequently seen fragments in the dataset and filtered our dataset to only retain molecules whose fragments were in the top 100 most seen

fragments. All selected fragments can be seen in detail in Appendix B. Notably, even after filtering the dataset based on the top 100 fragments we still obtain a dataset of 2,093,767 molecules. Figure 2 shows the proportion of the original 4.7 million molecules that can be recovered when using only the top k most frequently appearing fragments. The red dotted line represents the number of molecules recovered by keeping the top 100 fragments as we do in practice.

In order to properly construct molecules from fragment graphs we need to know which atoms on a pair of fragments have a bond between them. In light of this, we select a set of edge types for each pair of fragments in the top 100 fragment set as follows. For each fragment pair, we compute a histogram based on the pair of atoms between the fragments which are most commonly seen across the dataset. We then filter then choose the top 3 most seen atom connection points between the fragments and label them as edge ID 1, 2, and 3 respectively, discarding any molecules which have fragment connection points outside the top 3 most seen. If any fragment pair has $l < 3$ attachment points seen in the dataset we only include l edge IDs for that fragment pair. Finally, in order to get a dataset small enough to train our models tractably, we uniformly sampled 100,000 molecules from the molecules left after filtering and fixed them as the dataset used for all experiments.

5.2 Molecule generation

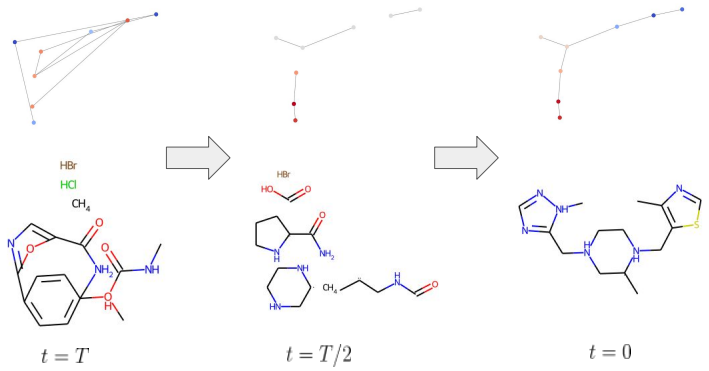

Figure 3: Side-by-side comparison of fragment graphs and their corresponding molecules at different diffusion timesteps. Note that invalid connections between fragments are not shown in the molecular representation.

Our diffusion model was trained with a batch size of 128, AdamW with a learning rate of $2\mathrm{e}{-4}$, and using 1,000 diffusion steps to sample. All models were allowed to train for ten hours. Further hyperparameter details are included in Appendix A

Figure 3 shows an example diffusion sequence generated by the fragment-diffusion model. The top row of the figure shows the generated fragment graphs while the bottom row shows the molecule generated by this fragment graph. Note that if a triple (frag_id, frag_id, edge_type) specified by the diffusion model is not contained within our dataset the fragments in the triple are left disconnected in the resulting overall molecule. We can see that, over time, the fragment-diffusion model samples a simple fragment graph that generates a compelling drug-like molecule at termination. Finally, Figure 4 displays random samples from the fully trained fragment diffusion model.

5.3 Evaluation

Each method was trained (or simply run, in the case of uniform sampling) and evaluated over three random seeds. We measured the methods across four metrics. Molecule Validity measures the percentage of molecules that are not disconnected and that do not violate basic valency constraints.

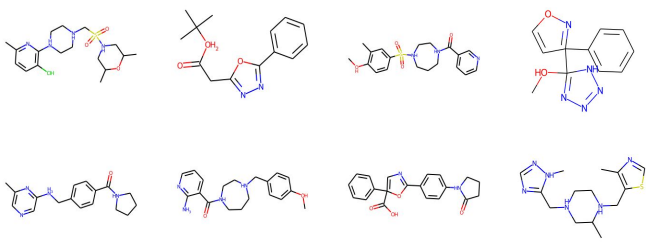

Figure 4: A set of 8 randomly sampled molecular graphs generated by a fully trained model

The Quantitative Estimate of Druglikeness (QED) and Octanol-Water Partition Coefficient (LogP) are both heuristic-based metrics that evaluate how drug-like a molecule appears. The Synthetic Accessibility (SA) Score measures how much a molecule appears to be composed of commonly seen reagants, penalizing rare or complicated substructures. Each of these metrics was computed using RDKit (Landrum).

We compare our model to four baselines. First, we compare with the original DiGress implementation on the atom-level graphs. DiGress uses an augmented set of molecular features (namely, an encoding of the valency of the atoms, their charge, and their weight). However, extending these features to fragments is not straightforward (e.g., how should we measure the valency of a molecular fragment?) and so we did not include them in our fragment-graph diffusion model. As such, we evaluate against both the atom-graph diffusion model without extra molecular features (to more faithfully compare with the fragment-graph) and with extra molecular features. This atom-graph diffusion model was trained with the same hyperparameters as our fragment-based diffusion model. Further, to evaluate how much moving to a fragment-based representation helps with building valid, synthesizable, drug-like molecules we sample fragment graphs in an unconstrained, uniform fashion. To more closely resemble the molecules generated by our fragment-diffusion we also uniformly sample fragment chains, graphs $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{E})$ such that $\forall \boldsymbol{x} \in \boldsymbol{X}$ we have that $|\{\boldsymbol{e} : \boldsymbol{e} \in \boldsymbol{E}, \boldsymbol{x} \in \boldsymbol{e}\}| \in \{1, 2\}$.

A comparison of results is shown in Table 1. We see that our model consistently produces valid molecules, while also yielding the best performance on the QED and LogP metrics by a considerable margin. The atom-based diffusion methods create valid molecules at a significantly lower rate, while the drug-like qualities of their generated molecules are worse than those of fragment-based diffusion. Interestingly, the atom-based diffusion with augmented features performed within the margin of error of the non-augmented version for all metrics besides the SA score, on which the augmented version performs better. This indicates that the lack of domain-specific molecular features is not necessary for fragment-based diffusion to perform well.

The uniform sampling methods performed exceptionally well at building valid molecules, with the chain uniform approach yielding nearly 100% validity across all seeds, indicating that a fragment-based vocabulary for the generative model is indeed a strong inductive bias. Despite the uniform methods' strength in generating valid molecules, the fragment-based diffusion model scores significantly higher on metrics measuring the drug-likeness of generated molecules, indicating that the diffusion model captures the drug-like nature of the training dataset. The fragment-based diffusion model achieves a worse SA score than other methods, perhaps because it generates complex molecules that use rare molecular substructures. It is important to note that the SA score is a rough proxy of synthesizability – the only real way to determine synthesizability is by evaluating with an algorithm that computes a full synthesis pathway, which we leave to future work.

6 Discussion

While our initial results are promising, there are a number of limitations to our approach that could be addressed in future versions of this architecture. Firstly, as explained in Section 5.1, we work with

Table 1: Model performance

Model	% Validity (↑)	QED (↑)	LogP (↑)	SA Score (↓)
Unconstrained Uniform	84.7 ± 1.20	0.41 ± 0.01	1.34 ± 0.08	6.88 ± 0.27
Chain Uniform	99.3 ± 0.40	0.57 ± 0.00	1.21 ± 0.09	$\mathbf{5.41 \pm 0.02}$
Atom-Graph Diffusion	70.0 ± 10.6	0.52 ± 0.07	1.23 ± 0.24	6.11 ± 0.34
Augmented Atom-Graph Diffusion	67.5 ± 6.10	0.49 ± 0.05	1.11 ± 0.07	5.68 ± 0.25
Fragment-Graph Diffusion	$\mathbf{100 \pm 0.00}$	$\mathbf{0.61 \pm 0.03}$	$\mathbf{2.68 \pm 0.15}$	6.88 ± 0.09

only the top 100 fragments, which cover 44% of the molecules in the ZINC database. We also limit ourselves to 3 possible edge types connecting each pair of fragments limiting our coverage further and preventing the model from predicting novel connections between different fragments that may be chemically valid, but not present in the training dataset. These two choices limit the expressivity of our architecture, but including more possible fragments or edge types could decrease performance, as there may not be enough data to learn the properties of rarer fragments. It remains to be seen how to tune these choices to balance the expressivity of the model with its performance. We could also use other methods to define our fragments and their connections: either by consulting a database of chemical reagents and their reactions to determine possible fragments and bonds, or by using the approach of Kong et al., who automatically search atomic graphs for the minimal set of "principal subgraphs" that yield full coverage of their dataset.

Secondly, the current architecture can only sample from the training distribution, which is only the first step in designing a full drug discovery algorithm. Our overall goal is to produce molecules that possess certain properties desirable for a drug. This can be accomplished by adding conditioning to our architecture, which would guide the denoising process toward desirable samples.

Another limiting factor in our implementation is how we diffuse over connections between fragments. Currently, as described in Section 4.1, the connections between fragments are determined by the most common connections between fragments seen in data, with edge labels assigned arbitrarily. An alternative approach would be to additionally learn the connections between fragments. This could be approached by a two-stage approach: first, a diffusion model selects the fragments types and which fragments are connected to which fragments, producing edges $e_{i,j} = (\boldsymbol{x}_i, \boldsymbol{x}_j)$. and once these are established, a second network (either another diffusion model or a standard graph neural network for link prediction) predicts the corresponding bonds $b_{k,l} \in \boldsymbol{A}(\boldsymbol{x}_i) \times \boldsymbol{A}(\boldsymbol{x}_j)$.

Lastly, while we work with molecular graphs due to their simplicity, we would be able to achieve more accurate property prediction if we worked directly with 3D atomic (or fragment) coordinates. Extending our method to 3D conformers would not be simple: while in a graph-based approach, we only need to learn to select fragments and connect them together, if we were to use 3D coordinates instead, we would need to learn three things: which fragments to select, the positions of each fragment relative to each other fragment, and the conformation of the fragments themselves. These three features are all heavily interrelated: if two fragments are close to each other, then they would significantly affect each other's conformations. While this would be a difficult task to approach, we believe it to be a theoretically interesting avenue of research that could be extended to other related problems in 3D space, such as materials discovery, protein binding, or even point cloud generation.

7 Conclusion

In this work, we have shown that combining a denoising diffusion model with a fragment-based representation of molecules can lead to the generation of more realistic, complicated druglike molecules. Further evaluations are needed to test the performance of this model, but it appears to be a promising approach for generating realistic molecules that are larger and more complex than can those that are accessible by current models. In the future, this generation can then be tuned towards specific drug design goals, potentially making it cheaper and faster to develop new drugs.

We hope to extend this work further by modifying the diffusion process over edges between fragments, applying this model to graphs in other fields, and developing an approach for 3-dimensional graphs.

ACKNOWLEDGMENTS

We thank Joey Bose and Prakash Panangaden for their guidance in sculpting this research project.

REFERENCES

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R., and Jensen, K. F. Generative models for molecular discovery: Recent advances and challenges. *WIREs Computational Molecular Science*, 12(5):e1608, 2022. ISSN 1759-0884. doi: 10.1002/wcms.1608. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1608. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1608.

De Cao, N. and Kipf, T. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Degen, J., Wegscheid-Gerlach, C., Zaliani, A., and Rarey, M. On the art of compiling and using'drug-like'chemical fragment spaces. *ChemMedChem: Chemistry Enabling Drug Discovery*, 3(10): 1503–1507, 2008.

Du, Y., Fu, T., Sun, J., and Liu, S. MolGenSurvey: A Systematic Survey in Machine Learning Models for Molecule Design, March 2022. URL http://arxiv.org/abs/2203.14500. arXiv:2203.14500 [cs, q-bio].

Dwivedi, V. P. and Bresson, X. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Flam-Shepherd, D., Zhigalin, A., and Aspuru-Guzik, A. Scalable Fragment-Based 3D Molecular Design with Reinforcement Learning, February 2022. URL http://arxiv.org/abs/2202.00658. arXiv:2202.00658 [cs].

Gao, W. and Coley, C. W. The Synthesizability of Molecules Proposed by Generative Models. *Journal of Chemical Information and Modeling*, 60(12):5714–5723, December 2020. ISSN 1549-9596, 1549-960X. doi: 10.1021/acs.jcim.0c00174. URL https://pubs.acs.org/doi/10.1021/acs.jcim.0c00174.

Gottipati, S. K., Sattarov, B., Niu, S., Pathak, Y., Wei, H., Liu, S., Blackburn, S., Thomas, K., Coley, C., Tang, J., et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International Conference on Machine Learning*, pp. 3668–3679. PMLR, 2020a.

Gottipati, S. K., Sattarov, B., Niu, S., Pathak, Y., Wei, H., Liu, S., Liu, S., Blackburn, S., Thomas, K., Coley, C., Tang, J., Chandar, S., and Bengio, Y. Learning to Navigate The Synthetically Accessible Chemical Space Using Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3668–3679. PMLR, November 2020b. URL https://proceedings.mlr.press/v119/gottipati20a.html. ISSN: 2640-3498.

Gupta, A., Müller, A. T., Huisman, B. J., Fuchs, J. A., Schneider, P., and Schneider, G. Generative recurrent networks for de novo drug design. *Molecular informatics*, 37(1-2):1700111, 2018.

Han, J., Rong, Y., Xu, T., and Huang, W. Geometrically Equivariant Graph Neural Networks: A Survey, February 2022. URL http://arxiv.org/abs/2202.07230. arXiv:2202.07230 [cs].

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Honda, S., Shi, S., and Ueda, H. R. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. *arXiv preprint arXiv:1911.04738*, 2019.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant Diffusion for Molecule Generation in 3D, June 2022. URL http://arxiv.org/abs/2203.17003. arXiv:2203.17003 [cs, q-bio, stat].

Igashov, I., Stärk, H., Vignac, C., Satorras, V. G., Frossard, P., Welling, M., Bronstein, M., and Correia, B. Equivariant 3d-conditional diffusion models for molecular linker design, 2022. URL https://arxiv.org/abs/2210.05274.

Irwin, J. J. and Shoichet, B. K. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768, 2012.

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Jo, J., Lee, S., and Hwang, S. J. Score-based generative modeling of graphs via the system of stochastic differential equations. *arXiv preprint arXiv:2202.02514*, 2022.

Kong, X., Huang, W., Tan, Z., and Liu, Y. Molecule generation by principal subgraph mining and assembling. In *Advances in Neural Information Processing Systems*.

Landrum, G. Rdkit: Open-source cheminformatics. URL http://www.rdkit.org.

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems*, 31, 2018.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):1–14, 2017.

Satorras, V. G., Hoogeboom, E., Fuchs, F. B., Posner, I., and Welling, M. E (n) equivariant normalizing flows. *arXiv preprint arXiv:2105.09016*, 2021.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Vignac, C., Krawczuk, I., Siraudin, A., Wang, B., Cevher, V., and Frossard, P. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Walters, W. P. and Murcko, M. Assessing the impact of generative AI on medicinal chemistry. *Nature Biotechnology*, 38(2):143–145, February 2020. ISSN 1546-1696. doi: 10.1038/s41587-020-0418-2. URL https://www.nature.com/articles/s41587-020-0418-2. Number: 2 Publisher: Nature Publishing Group.

Wouters, O. J., McKee, M., and Luyten, J. Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. *JAMA*, 323(9):844–853, March 2020. ISSN 0098-7484. doi: 10.1001/jama.2020.1166. URL https://doi.org/10.1001/jama.2020.1166.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. GeoDiff: a Geometric Diffusion Model for Molecular Conformation Generation, March 2022. URL http://arxiv.org/abs/2203.02923. arXiv:2203.02923 [cs, q-bio].

Yang, S., Hwang, D., Lee, S., Ryu, S., and Hwang, S. J. Hit and lead discovery with explorative rl and fragment-based molecule generation. *Advances in Neural Information Processing Systems*, 34: 7924–7936, 2021.

A HYPERPARAMETERS

Learning		
Optimizer	Adam	
Batch Size	128	
Learning Rate	2e-4	
Diffusion Model		
Noise Schedule	Cosine	
Diffusion Steps	1000	
Lambda	5	
GNN		
Architecture	Graph Transformer	
Number of Layers	8	
Activation Function	ReLU	
Input MLP Dimensions:		
	Nodes	128
	Edges	64
	Graph	128
	Layers	2
Transformer Dimensions:		
	Nodes	256
	Edges	64
	Graphs	64
	Attention Heads	8

Table 2: Hyperparameters used for both fragment-level molecule generation, and atom-level molecule generation baseline.

B ALL FRAGMENTS

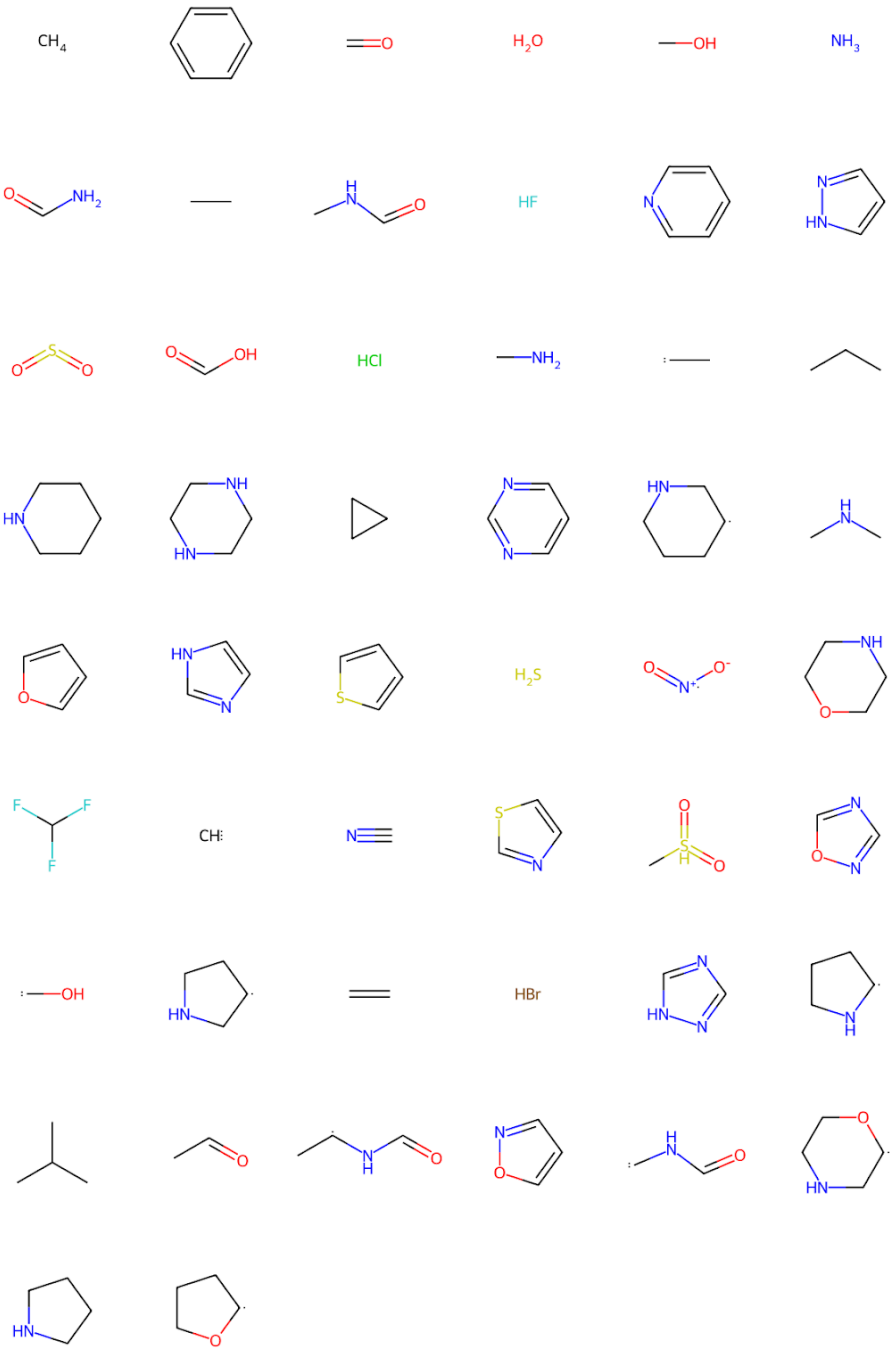

Figure 5: The 100 most common fragments returned by the fragmentation algorithm.