Scaffold Embeddings: Learning the Structure Spanned by Chemical Fragments, Scaffolds and Compounds

Austin Clyde
Department of Computer Science
University of Chicago
Chicago, IL 60637
aclyde@uchicago.edu

Bharat Kale, Maoyuan Sun, Michael E. Papka
Department of Computer Science
Northern Illinois University
DeKalb, Illinois 60115
{bkale, smaoyuan, papka}@niu.edu

Arvind Ramanthan, Rick Stevens
Data Science and Learning Divison
Argonne National Laboratory
Lemont, IL 60439
{ramathana, stevens}@anl.gov

Abstract

Chemical space is often discussed with a focus on the chemicals rather than the spatiality. We focus on exploring a natural structure for representing chemical space as a structured domain: embedding drug-like chemical space into an enumerable hypergraph based on scaffold/fragment classes linked through an inclusion operator. Storing the associated structure for over 100 billion molecules is intractable for a standard database and it would lack generative qualities. This paper shows transformer models can be used two-fold to generate novel molecules based on relations to other molecules as well as retrieve molecules like a database based on relational queries. Thus, the transformer model is used to navigate an underlying graph representation of chemical space on-the-fly. We develop a user interface to illustrate how on-the-fly chemical space generation and traversal can be interactive and useful for medicinal chemistry and chemical space visualization.

1 Introduction

Sampling, enumerating, and understanding "chemical *space*" is the grand challenge of computational chemistry [26, 39, 32]. The explosion of computational drug discovery and need for novel chemical therapies has drawn in machine learning (ML) and high-performance computing (HPC) to produce novel chemotypes and explore the diversity of chemical space [9]. Few authors have actually addressed the *spatiality* of "chemical space". A space, in the computational and mathematical sense, is a set with some structure (prior). Nearly all treatments of chemical space assume no spatial formalism at all outside of a basic vector space. The basic vector space follows from the use of computational kernels, molecular descriptors, and then generalized to "the total descriptor space that encompasses all the small carbon-based molecules that could in principle be created" [11]. Deep learning embeddings have generally replaced the kernel-driven descriptor conception of chemical space with a statistical unit ball structure (as resulting from a *particular* language model or variational autoencoder (VAE) embedding) [3]. Without a doubt, the chemical aspect of chemical space has been expanded greatly through these generation programs, but without too much formal attention to the *spatial*-aspect.

Learning Meaningful Representations of Life (NeurIPS 2021).

In this paper, we propose a strong prior for chemical space based on chemical scaffolds. We show that chemical scaffolds turn chemical space into a mathematical ordered lattice (or a graph). We train a transformer model on generating the graph relations from node to node. This can be seen two-fold as both a generative problem, predicting new nodes which may have a particular relationship to another node, and as a document retrieval/compression problem, retrieving the already computed nodes which a particular relationship to another node. As chemical space enumerations continue to grow, maintaining a structured relational dataset with over 100 billion molecules is nearly intractable.

Strong priors provide *context* to computational workflows by aligning humans theoretical retentions with the protentions of a generative model. For example, a non-conditional GAN is useful for generation a corpus of samples, but is not very useful for generating a series of example logos for a new start-up company. Conditioning, in this case, is the application of a prior ontology–what entities in the world one might want to discuss with the model. Strong priors include the model in a discpline's discourse by forcing it to interact based on the particular ontological and theoretical distinctions already there in a field. While many object that stronger priors may limit the overall breadth of a generative project, we remark that drug discovery requires money and time across computational and non-computational disciplines which means either models choose to supplement the existing process or only desire to completely rework it. Given the rapid need for expanding chemical space with the gamut of therapeutics needs existing today, we pursue the former of contextualizing models into current disciplines and workflows to supplement the discovery process rather than attempt to disrupt and explode it. We showcase this novel generative strategy through an interactive web server which combines algorithmic computational chemistry and generative deep learning into a single interface for viewing regions of chemical space.

The enormous design space of chemical compounds, estimated to be about 10^{60} [4], motivates an immediate need for efficient and often automated exploration for synthesis and assay development for various applications, including drug discovery and materials design. Computational enumeration of chemical space is a long-studied problem since the early ages of computing [7]. The current state of the art projects have enumerated around 2 billion drug-like compounds, and GDB has around 166 billion compounds of up to 17 atoms of C, N, O, S, and halogens [37, 40]. Even with these vast libraries, recent work has shown a vast difference between the diversity enumerated in ultra-large libraries and the underlying space [22]. Especially in the context of drug discovery, an emerging need in the cheminformatics community is the ability to *navigate* this enormous design space in the hopes of generating new molecules (or designs) that can optimally bind to a protein/drug-target of interest or *refine* molecules based on specific physio-chemical and safety features that make it attractive as a drug that can be formulated for the market.

Given the vastness of drug-like chemical space, how can we computationally explore it? In 1875, Caley published a short note on his enumeration of alkanes utilizing a tree structure [6]. Though Caley's enumeration ended up having a few errors, it is a very early account of treating chemical space as a structured mathematical object [38]. Over 100 years later, the ideas of enumerating structurally similar compounds and comparing their activity became known as quantitative structure relationship studies (QSAR/SAR). QSAR/SAR is the standard method in medicinal chemistry for taking an interesting chemical compound to an optimized and potent drug lead. In 1984, Klopman developed Computer-Automated Structure Evaluation (CASE), which "perform[s] automatically all operations related to the structure-activity analysis" [28]. A success in its own right, CASE utilized the graph topology of molecules to generate QSAR studies or predict activity based on fragments. This graph structure naturally leads to studying subgraphs and their relations, such as decomposing the graph into a class of similar molecules sharing a framework (scaffold), linkers connecting rings, and sidechains [2]. Utilizing these ideas, various tool-kits and genetic algorithms have been designed to combine or grow molecular fragments into optimized drugs [30, 8]. While these ideas in organization lay the framework for certain practices of medicinal chemistry, the methods do not address the problem of enumerating compounds in an organized way to find diverse chemical scaffolds.

Deep learning (DL) offers a new set of tools and algorithms for generating novel molecular pieces. With the introduction of generative models which can be sampled, such as variational autoencoders [25], or generative adversarial networks [18], de-novo molecular generation took hold as a practice in drug discovery [36]. Molecules were embedded into a continuous representation and then given a decoder, sampled from continuous space—allowing property optimization and molecular generation based on some distance metric in the latent representation. These approaches have had much success.

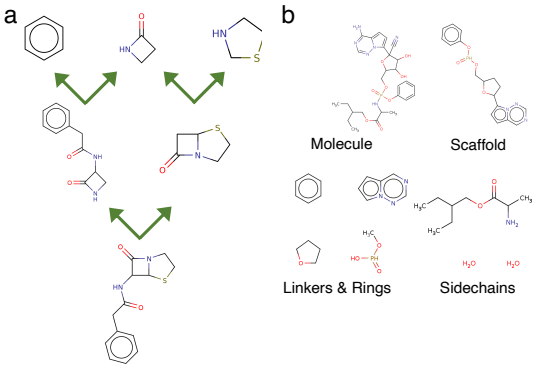

Figure 1: **Decomposition of a chemical scaffold and molecule**. (a) Starting at the bottom, a chemical scaffold with three rings is decomposed into two 2-ring scaffolds, which can be decomposed further into 1-ring scaffolds. (b) Remdesivir is decomposed from the molecule, to its scaffold, rings and linkers, and sidechains.

We extend on this work by focusing on computational organization and enumeration specifically—seeking more structure than $\mathcal{N}(X, \Sigma)$ or $\mathbb{R}^n$.

Our contribution in this paper is twofold: (1) representing large molecular libraries using molecular building blocks (i.e., fragments, scaffold, linkers/decorations), and (2) learning to navigate latent representations of molecular hypergraphs leveraging transformer networks to *operate* on molecular building blocks to generate new molecules. This project is distinct from prior molecular generation problems as we focus on enumerability and organization over property prediction. We demonstrate that our building blocks representation provides a natural mechanism to organize large chemical spaces in a statistically meaningful manner. Further, the transformer networks suggest the design of novel molecules that can 'expand' on a given scaffold design, which can be used for subsequent rounds of virtual screening studies.

2 Chemical Space as a Structured Domain

Consider the set of drug-like molecules $\mathcal{M}$. $\mathcal{M}$ is not directly computable as it is a concept class for *molecules*. For computation, molecules require a computable representation, and this is the start of the difficulty. Representations are models of molecules which can be identified with a molecule. Graphs are a natural model of molecules, where nodes are atoms and edges are vertices [24]. SMILES are another representation of molecules, which are a breadth-first search over the graph in a particular syntax. SMILES, unlike graphs, are not injective over molecules (if two SMILES strings are not equal, it does not imply the underlying molecules are not equivalent) [35]. There are other representations which are less common such as point clouds, junction trees, or voxelization [14]. We define R_X to be a general representation mapping from molecules to some set X from $\mathcal{M}$.

Embeddings are distinct from representations. Embeddings are functions which take a representation X to embedding space Y. For instance, molecular fingerprints are an algorithm which takes graphs of molecules to $\mathbb{R}^n$ by utilizing a hashing function around the nodes or regions of a graph [44]. Node2vec models take graphs to $\mathbb{R}^n$. A simple variational autoencoder's encoder can take SMILES to a Gaussian unit ball $\mathcal{N}(X, \Sigma)$. The junction tree variational autoencoder takes a junction tree to a latent unit ball. In the later two examples, the idea of sampling from a normal unit ball is essential for maintaining the density of the sampling space—an important aspect of creating a generative model (see SI section 2 on sampling). Given a decoder, these embedding spaces can be sampled to produce potentially new molecules or molecules through a constrained optimization problem. The two embedding spaces so far have convenient distance metrics, denoted δ_Y.

A number of papers have focused on generative models for the design of new molecules [13, 24, 16, 17, 29]. These approaches either use a string representation (e.g., SMILES representation mapped onto a molecular graph) or an explicit molecular graph representation (e.g., [23]) to *encode* the molecular data into a continuous representation from which new examples can be drawn.

While these methods are very successful at certain property predictions and general optimization, they do not solve the enumerability problem. Both $\mathbb{R}^n$ and $\mathcal{N}(X, \Sigma)$ are continuous and not countable. In particular, every molecule has an open ball around it in embedding space of equivalent points which is a problem for enumerating discrete sets of molecules. In other words, if φ^{-1} is a decoder from an embedding $\mathbb{R}^n \to X$, and $\equiv$ is an equivalence relation on the representation X, there exists $y_1, y_2 \in \mathbb{R}^n$ and $\epsilon > 0$ such that $0 < \delta_{\mathbb{R}^n}(y_1, y_2) < \epsilon$ so

$$\varphi^{-1}(y_1) \not\equiv \varphi^{-1}(y_2).$$

In order to structure the embedding space to be conducive for enumeration, we must find an embedding space that is countable and discrete, just as Caley sought out by means of a tree.

Molecular scaffolds are well defined through algorithms, decompose well into networks, and offer a general description of global properties (such as orientation in a protein binding region) [2, 43]. Molecular scaffolds represent the core of a molecule, typically defined around the number of rings in the structure. Non-ring sturctures in molecules include linkers and sidechains which get collapsed in this representation to a single scaffold representative. In figure 1, we show a molecular scaffold decomposing into smaller scaffolds. In this way, we can take a graph or SMILES representation of a molecule and map it to this discrete embedding structure. The mapping into the scaffold structure is unique. As other authors rely on decoders to decode the embedding space, we will rely on decoders to sample the scaffold for the variety of molecules a part of it.

3 Methods

The conceptual machinery for treating chemical space is developed. There is an elegant statement of the principle of fragment-based drug design through the operations among scaffolds. Further, the framework developed provides intuitive concepts for understanding the diversity and size of chemical space explored or discussed by a model or computational research program. As a computational learning problem, we use transformer as seq2seq models to implement large graph navigation in practice.

3.1 Scaffold Embeddings

Utilizing the concept of scaffolds developed in section 2, we assume the operation `Scaffold` as a given oracle such that `Scaffold` is injective and defined for every molecule. We define $\mathcal{S}$ as the set of all scaffolds.

A hypergraph is a generalized graph where edges group more than two vertices. A hypergraph is n-regular when every vertex is contained in exactly n edges. Scaffolds as hypergraph edges over molecules form a 1-regular graph, as every molecule belongs to exactly one scaffold class, thus every vertex has degree 1 in the hypergraph. We denote the hypergraph as $\mathcal{H} = (\mathcal{M}, \mathcal{S})$. The

Operations on scaffolds. We denote computational operations in `Monospace` font, and add a subscript Φ to represent parameters which may be required for the operations (i.e. $\texttt{Expand}_\Phi$).

1. $\texttt{Expand}_\Phi$ and `Scaffold`: Molecules and scaffolds represent two distinct types which can be converted back and forth (figure 2B). Scaffold classes can be *expanded*, where we envision zooming in, via the $\texttt{Expand}_\Phi$ model (i.e. $\texttt{Expand}_\Phi$: $\mathcal{S} \to \mathcal{M}$). Similarly, molecules can be taken to their scaffold via the program `Scaffold` (`Scaffold`: $\mathcal{M} \to \mathcal{S}$). We utilize RDKit to compute `Scaffold` via the MurckoScaffold module [31]. We note a model can be trained for this task; however, given the efficiency of the algorithm it did not seem fruitful at this time.
2. $\texttt{Successor}_\Phi$ and `Predecessor`: the successors of a scaffold S_1 are the set of all scaffolds S which contains S_1 as a substructure (figure 2). The predecessors of a scaffold S_1 are all scaffolds S which S_1 is a superstructure. In general, there is no algorithm for successor given only a scaffold, as it requires sampling chemical space. However, predecessor has an efficient algorithm with a structure that can always be fragmented into smaller scaffolds without sampling other data. These operations are the atomic building blocks of navigating between scaffold classes (and induces a strict partial ordering $(\mathcal{S}, \prec)$). These operations are from $\mathcal{S}$ to $\mathcal{S}$. We also consider the standard graph structure induced by the relation $\texttt{Successor}_\Phi$ and `Predecessor`, and denote it $\mathcal{S}_\mathcal{G} = (\mathcal{S}, \texttt{Successor}_\Phi)$ where $\texttt{Successor}_\Phi$ can be used

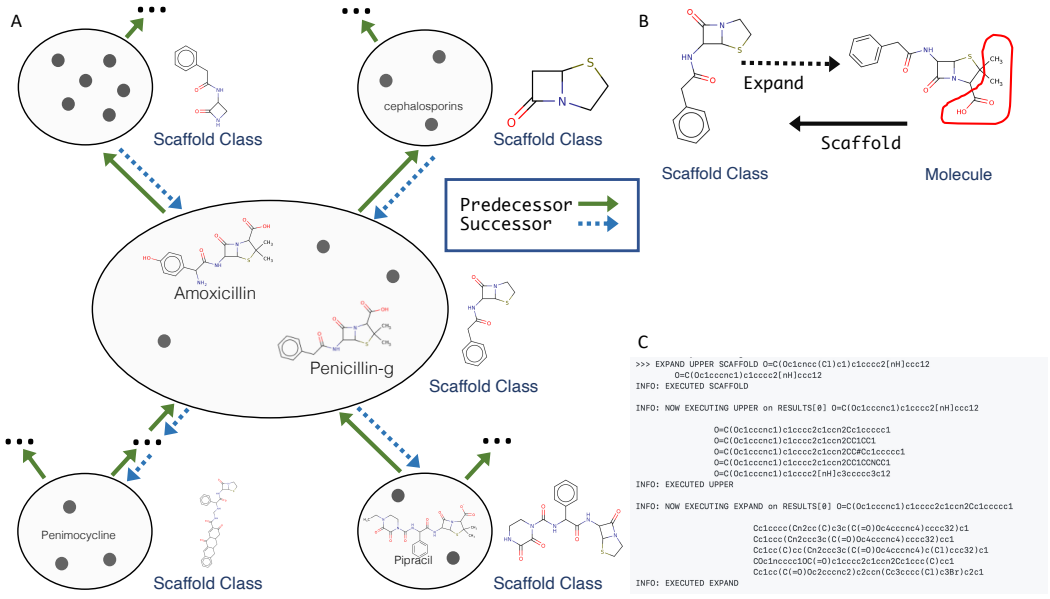

Figure 2: (A) **Scaffold as classes over drug-like chemical space.** Every molecule (represented by dots or depiction inside circles) is inside a single scaffold class. Scaffold classes are related through common substructures, forming a hierarchy of classes. Penimocycline, for example, belongs to a scaffolding class from far Penicillin-g's or Amoxicillin's class, while Pipracil is a direct successor of the Penicillin-g class. The `Predecessor` function is defined via an algorithm, and the $\texttt{Successor}_\Phi$ function requires a generative model when working without data (i.e., given a single scaffold you cannot compute its successor unless you understand chemistry, thus have parameters Φ, but you can compute all of it is predecessors recursively without knowing how to generate new compounds). (B) **Scaffold and molecule relation**. Scaffolds are the core or framework of a molecule, and they represent a class of molecules. Scaffolds, or scaffold classes as we often refer, group molecules together. A class can be extended by adding decorations to the scaffold, such as linkers and sidechains. Through the scaffold function, we obtain the scaffold of a molecule. (C) **Relations from nodes to set of relations**. The programmatic interface for taking a node, a string representation of a molecule, and sampling the transformer model under the hood as a means of sampling chemical space on the fly.

to determine the edge relation. This graph can be directed or undirected, but for our case we consider the undirected graph mostly.

3. $\texttt{Union}_\Phi$ and `Intersection`: two scaffolds S_1 and S_2 can be combined to form a union. More formally, the union of S_1 and S_2 is the set of scaffolds that contain S where S has S_1, and S_2 has immediate predecessors. Similarly, the intersection of S_1 and S_2 is simply the maximum common substructure (MCS) of S_1 and S_2, for which an efficient algorithm exists for small drug-like molecules. [5, 10]. In general, MCS is NP-complete, but there are heuristics for drug-like molecules that provide a rather efficient algorithm [15]. These operations are from $\mathcal{S}$ to $\mathcal{S}$.

These basic operations can be combined into more complex operations such as

$$\texttt{UpperCone}_\Phi(S) = \{A : S \prec A\} \tag{3.1}$$

$$\text{or} \quad \texttt{LowerCone}(S) = \{B : B \prec S\} \tag{3.2}$$

Upper cones of scaffold classes are actually a common object of interest for drug discovery. For instance, Penimocycline is in the upper cone of Penicillin-g's scaffold class (see figure 2). Successful exploration of upper cones is the theoretical cornerstone of fragment based drug design [41, 34]. Recently, fragment X-ray crystalgraphic screens have been performed on important drug targets such as SARS-CoV-2 proteases in search of an inhibitor [12]. Given a set of fragment hits for a protein target in a binding region,$\{m_i\}_{i\in H}$, take the scaffold classes of those hit, $\{S_i^h\}_{i\in H}$. The principle of

fragment based drug design can be expressed as there exists some index set I^* such that $I^* \subseteq H$ and

$$\hat{H} = \bigcap_{i \in I^*} \texttt{UpperCone}_{\Phi}(S_i^h) \tag{3.3}$$

where $\hat{H}$ is a set of scaffold classes, $\hat{H}$ is not empty, and some molecule in a scaffold in $\hat{H}$ is a likely candidate. In other words, a set of fragments can be grown to sets of larger drug-like molecules, and some intersection of those possible larger molecules will be a hit that is likely a drug lead for this protein target. In an embedding space such as $\mathbb{R}^n$, the same principal does not apply, and is dependent on the embedding context (for instance, based on a particular property [21]). Furthermore, there no guarantees about molecules in an interval between two molecules, whereas the intersection of upper cones, for example, does have such guarantees (if it is not empty).

Given there is no algorithm for producing successor scaffolds without relying on sampling chemical space, we treat the problem as a learning problem. We note that we cannot rely on fragments as a vocabulary given this construction as other methods have (for instance, [23] utilized a finite vocabulary containing one member rings, linkers, and sidechains from the dataset). When using such a vocabulary, there are chains of scaffolds that cannot be represented as the $\texttt{Successor}_{\Phi}$ function can only sample scaffold classes S for which every one ring member in the $\texttt{LowerCone}(S)$ is in the finite vocabulary.

3.2 Modeling Hypergraphs with Transformers

While the method outlined has no constraints on compounds' synthetic accessibility, it is a necessary and essential aspect of chemical space exploration for drug discovery. To focus on synthetic accessibility while paying attention to maximizing library size, we utilize a dataset from Synthetically Accessible Virtual Inventory (SAVI) [37]. SAVI contains over 1.7 billion reaction products (along with rich reaction and metadata). We utilize only the SMILES of the products.

We build two datasets from SAVI. The first utilizes RDKit to determine the scaffold for each of the compounds listed [31]. We utilized a 200M sample from the entire dataset and extended the data by a factor of 5 by randomizing the SMILES both for the target (scaffold) and source (molecule) [1]. A set of 20M molecules with a unique scaffold class are held out as validation data. A second dataset is created by taking a subsample of the prior dataset, 20M, and utilizing the ScaffoldGraph package to decompose each scaffold into a network of scaffolds [43]. We sample edges (representing the successor of two scaffold nodes), resulting in a dataset of five million successor pairs. This dataset is extended to 50M utilizing random smiles sampling. Predecessor data is flipping the columns (sources become targets, and targets become sources) for the successor datasets.

On the one hand, $\texttt{Successor}_{\Phi}$ and $\texttt{Expand}_{\Phi}$ are generative models—given a scaffold, those operators are required to sample the space of successor scaffolds or molecules that have that scaffold. On the other hand, they are seq2seq task, taking one sequence to a different sequence. This combination of wanting a dense sampling strategy combined with seq2seq modeling differs from applications we have found in the literature. Common approaches to generative models have been utilizing VAEs or GANs to train some encoder-decoder model on sample reconstruction error with some regularization [14, 20, 19]. Seq2seq approaches in this space have focused on solving problems with a relatively small optimal solution set such as reaction modeling [42]. With the recent success of transformer models performing well on large datasets and seq2seq problems, we decided to follow the modeling as a seq2seq problem as Schwaller et al. have.

We utilize a transformer seq2seq model from the ONMT project [27]. Other works have utilized RNNs, but we utilize a transformer for both the encoder and decoder of the model [45]. Given the goal of not simple generation but rather generalizing a very large hypergraph for which a pure algorithmic solution is intractable, transformer models are a good fit compared to simpler RNN models. Code is compiled into a GitHub repository with scripts for data gathering, data preparation, model training, and sampling. The interface is geared towards developing front-end functions for quick medicinal chemistry questions regarding sampling molecular space.

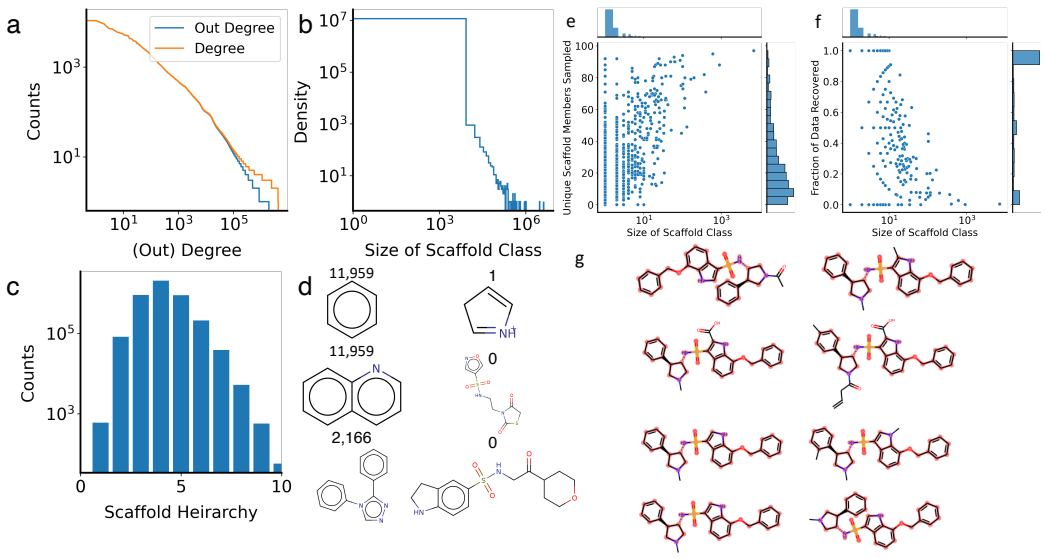

Figure 3: **Structure of scaffold classes** We constructed the scaffold classes (4M) for a random sample from SAVI (20M) molecules for (a)-(d). (a) We consider a random sample of 20M molecules from SAVI, and construct the scaffold classes and graph associated with the classes. Out degree indicates just `Successor` relations. (b) We show the distribution of the cardinality of (a)'s scaffold classes, which follows a power law for part of the distribution, and a uniform distribution for the other. (c) Scaffold classes are ordered into a hierarchy based on the number of rings its framework has. (d) The left column shows the scaffolds with the largest out degrees for hierarchies 1 to 3, and the right column shows random scaffolds of the least degree. **(e-f)** $\texttt{Expand}_\Phi$ **model reconstruction and sampling depth.** 1000 samples scaffold classes are drawn from the validation data, and $\texttt{Expand}_\Phi$ is sampled 100 times. Samples that are not valid smiles or passed verification are removed. (*left*) Samples for each scaffold are intersected with the known molecules in that scaffold class from the validation data, and the fraction found is plotted. Smaller scaffolds are often recovered while larger ones are not. (*right*) Even though the $\texttt{Expand}_\Phi$ model captures most of the dataset for smaller scaffolds, the model generates more valid molecules based on the natural distribution of the scaffold class sizes in the data. **(g) Expansion of a scaffold.** The expansion of a scaffold class, highlighted in red, is expanded by sampling $\texttt{Expand}_\Phi$. Various side chains are added, but no sample is outside of the class.

4 Results

4.1 Computability of Scaffold Classes

We assess the structure of *scaffolding* chemical space, focusing on understanding the size of scaffold classes, how many scaffold groups there are in drug-like chemical space, and how they connect.

We impose a structure on $\mathcal{M}$ by creating scaffold classes $\mathcal{S} = \{S\}_{i \in I_s}$ such that every molecule m belongs to one and only one scaffold class, and all classes in $\{S\}_{i \in I_s}$ are disjoint. We also assign a hierarchy to scaffolds based on the number of rings. $\mathcal{H}_n$ is the set of all scaffold classes with ring size n.

$\mathcal{H}_0$ is the smallest hierarchy, which consists of only one scaffold class S_0, the set of all molecules with no rings (ring-less fragments, linkers, and side-chains). $\mathcal{H}_1$ is the set of all scaffold classes with one ring. The order of $\mathcal{H}_2$ is proportion to $|\mathcal{H}_1|$ choose 2 plus the combination of linkages and sidechain modifications from $\mathcal{H}_0$. We see growth similar to the partition function in theory. However, in practice, the distribution of molecules in real-world datasets typically follows a normal distribution with the mean around three rings (see figure 3).

Given this added structure of scaffolds, do scaffolds reduce the search space over molecules by many magnitude orders? If this is the case, we can search through a computable number of scaffolds, and once a few interesting classes are found, we can enumerate the molecules in that set. This strategy

Model	SMILES Validity	Type Accuracy	Correctness Accuracy
$\texttt{Successor}_\Phi$	98.9%	98.9%	97.9%
`Predecessor`	99.8%	99.8%	94.0%
$\texttt{Expand}_\Phi$	98.6%	-	96.9%

Table 1: **Performance metrics from graph navigation models.** Evaluations were performed with a holdout set from SAVI dataset. SMILES validity is the percent of samples that pass an RDKit parser. Type accuracy determines how many samples have the correct type (`Successor`, `Predecessor`, and `Union` models output type scaffold. In contrast, `Expansion` model outputs molecules (which can include a scaffold representative, and this metric is left out and computed as a part of correctness). Correctness accuracy is the percent of samples which are valid, typed correctly, and are equivalent to the algorithmic solution.

Scaffold	Class Size (Data)	Unique Sampled	Overlap (Recall)
c1ccc(COc2ccccc2)cc1	373,939	168,261	4,146 (1.1%)
O=S(=O)(c1ccccc1)N1CCCCCC1	88,608	145,904	20,097 (22.7%)
O=S(=O)(NCCc1ccccc1)c1ccccc1	911,360	176,539	23,715 (2.6%)
c1ccncc1	818,230	183,838	23,999 (3.0%)
O=S(=O)(NS(=O)(=O)c1cccnc1)c1ccccc1	203,891	173,599	20,331 (10.0%)

Table 2: **Sampling dense classes with** $\texttt{Expand}_\Phi$. Five dense scaffold classes were taken from the validation data and sampled. We sampled 100,000 times for each scaffold, utilizing a temperature of 1.5 and a beam search of length five and capturing the top two best beams from the search. While we do not capture a large set of the data, we believe these classes' sheer size presents a combinatorics problem. The unique samples are all correct and valid.

does not face the curse of 10^{68} drug-like molecules the current unstructured domain $\mathcal{M}$ faces. Given a 200M sample from SAVI, we found only 11.4M (5.7%) scaffold classes were needed to cover the entire dataset, and, in practice, there exists a large subset of molecules (165M) with only 685,000 (0.41%) scaffold classes. This reduction via scaffolds implies for a large subset of molecules, there is a reasonable 5 order of magnitude gain in search over scaffolds than pure molecules (from a database or chemical library perspective).

4.2 Hyerpgraph Navigation

We train three operations ($\texttt{Expand}_\Phi$, $\texttt{Successor}_\Phi$, `Predecessor`) utilizing three separate models. While there is an algorithm for `Predecessor`, we can compare it directly to the algorithm performance. Each model was trained for approximately two days on eight GPUs (NVIDIA Tesla V100). Each model was trained for 500,000 steps with a batch size of 8192.

To sample $\texttt{Expand}_\Phi$ and $\texttt{Successor}_\Phi$ we utilize beam search with a temperature of 1.5, beam size of 5, and randomizing the SMILES input. Samples are then validated utilizing RDKit. In table 4.2, we outline each model's accuracy. A uniform sample of scaffolds from the validation data was taken ($n = 1000$), and 100 samples were drawn for each scaffold class (figure 3(e-f)).

Given the density of some scaffold classes in the data compared to others (figure 3), more advanced sampling methods required for $\texttt{Expand}_\Phi$ on these classes. For scaffold classes with over 10^6 members in the data (mostly 1-ring and 2-ring common scaffolds), resampling validation data from the model is difficult (table 4.2). Given the uniqueness of sampling based on a category like scaffolds, rather than pure sampling points in a distribution or $\mathbb{R}^n$, comparisons to generative models' reconstruction accuracy are not reasonable.

Figure 3g is an example of a series of compounds which belong to a single scaffold class, but are sampled with different sidechains. The variety of sidechains while maintaining the single scaffold core is the basis of a QSAR series.

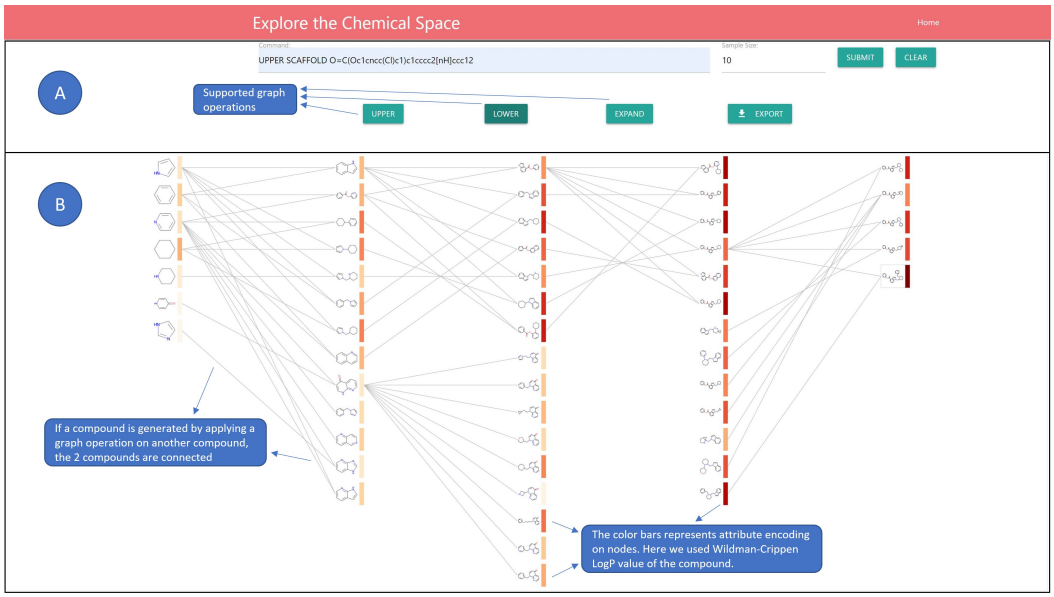

Figure 4: **An interactive visualization to explore the chemical space.** A) Control Panel: Provides controls dedicated to perform various operations on selected compounds. B) Visualization: Presents an interactive visualization of the generated compounds based on the operations.

4.3 Interaction with Transformer

The interface provides an interactive visualization to explore the chemical space. It allows the users to start from a known compound and explore the chemical space by making use of graph operations like upper, lower, and expand. The interface works with the expressions supported by the SGUser command line utility and creates the chemical network from the results. The SGUser command line is available on GitHub.[1] In figure 4(A), the first part of the control panel is similar to the SGUser command line tool. It has options to input an expression which can combine multiple graph operations and expected sample size. We can continue giving expressions, one after the other, and the visualization updates by adding new nodes and edges in the network. Alternatively, we can interactively select a compound from the visualization and use controls representing the various graph operations to explore the space. Once the user reaches a point where enough compounds are visualized, the interface allows the user to download the compounds as strings in a json file for further analysis. The visualization also supports encoding of attributes on nodes. Figure 4(B) shows the encoding of Wildman-Crippen LogP value on each node.

The interface also supports interactive zoom where users can configure the required behaviour during zoom. Currently supported options include - zoom to a) simply enlarge the molecular structures, b) enlarge the molecular structures and highlight a selected sub-structure in all the compounds in the current view, and c) enlarge the molecular structures and add more new samples in between the compounds in the view. Other possible interactions include ordering the compounds based on the visualized attributes on nodes.

5 Conclusion

This paper outlined a set of ordered equivalence classes via molecular scaffolds over the drug-like chemical space. We utilize seq2seq models to move between scaffolds, or classes of compounds, and between the scaffold hierarchy and the underlying molecules themselves. These operations ultimately form a set of algebraic tools for manipulating and navigating the chemical space. This algebra is expressive—enough to represent algorithms in drug design, such as the principle of fragment-based drug design or similar property principle of molecular scaffolds. This construction over $\mathcal{M}$ offers a unique take on the enumerability of the chemical space by collapsing the space into scaffold

[1] anonymous for review

classes, which can zoomed-in or zoomed-out of. We aim to understand better the distribution of synthetically accessible drug space and its relation to scaffolds as we hope scaffold classes reduce the space's overall size. Future work will expand on the interactive platform we developed for navigating the space and introduce more concept classes for finer and coarser granularity. We believe that to accelerate exploring the estimated 10^{60} drug-like molecules, a navigation strategy besides standard databases and compound enumeration is needed.

Acknowledgments and Disclosure of Funding

BK and MEP were supported in part by the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357. MS is supported in part by the NSF Grant IIS-2002082 and the Argonne Leadership Computing Facility as part of the Argonne National Laboratory faculty research program.

References

[1] Josep Arús-Pous et al. "Randomized SMILES strings improve the quality of molecular generative models". In: *Journal of cheminformatics* 11.1 (2019), pp. 1–13.

[2] Guy W Bemis and Mark A Murcko. "The properties of known drugs. 1. Molecular frameworks". In: *Journal of medicinal chemistry* 39.15 (1996), pp. 2887–2893.

[3] Thomas Blaschke et al. "Application of generative autoencoder in de novo molecular design". In: *Molecular informatics* 37.1-2 (2018), p. 1700123.

[4] Regine S Bohacek, Colin McMartin, and Wayne C Guida. "The art and practice of structure-based drug design: a molecular modeling perspective". In: *Medicinal research reviews* 16.1 (1996), pp. 3–50.

[5] Yiqun Cao, Tao Jiang, and Thomas Girke. "A maximum common substructure-based algorithm for searching and predicting drug-like compounds". In: *Bioinformatics* 24.13 (2008), pp. i366–i374.

[6] E Cayley. "Ueber die analytischen Figuren, welche in der Mathematik Bäume genannt werden und ihre Anwendung auf die Theorie chemischer Verbindungen". In: *Berichte der deutschen chemischen Gesellschaft* 8.2 (1875), pp. 1056–1059.

[7] Tim Cernak. "A machine with chemical intuition". In: *Chem* 4.3 (2018), pp. 401–403.

[8] Tim Cernak et al. "The medicinal chemist's toolbox for late stage functionalization of drug-like molecules". In: *Chemical Society Reviews* 45.3 (2016), pp. 546–576.

[9] Yu Cheng et al. "Molecular design in drug discovery: a comprehensive review of deep generative models". In: *Briefings in Bioinformatics* (2021).

[10] Michael M Cone, Rengachari Venkataraghavan, and Fred W McLafferty. "Computer-aided interpretation of mass spectra. 20. Molecular structure comparison program for the identification of maximal common substructures". In: *Journal of the American Chemical Society* 99.23 (1977), pp. 7668–7671.

[11] Christopher M Dobson et al. "Chemical space and biology". In: *Nature* 432.7019 (2004), pp. 824–828.

[12] Alice Douangamath et al. "Crystallographic and electrophilic fragment screening of the SARS-CoV-2 main protease". In: *Nature communications* 11.1 (2020), pp. 1–11.

[13] David Duvenaud et al. "Convolutional networks on graphs for learning molecular fingerprints". In: *arXiv preprint arXiv:1509.09292* (2015).

[14] Daniel C Elton et al. "Deep learning for molecular design—a review of the state of the art". In: *Molecular Systems Design & Engineering* 4.4 (2019), pp. 828–849.

[15] Michael R Garey and David S Johnson. "Computers and intractability". In: *A Guide to the* (1979).

[16] Justin Gilmer et al. "Neural message passing for quantum chemistry". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1263–1272.

[17] Rafael Gómez-Bombarelli et al. "Automatic chemical design using a data-driven continuous representation of molecules". In: *ACS central science* 4.2 (2018), pp. 268–276.

[18] Ian J Goodfellow et al. "Generative adversarial networks". In: *arXiv preprint arXiv:1406.2661* (2014).

[19] Francesca Grisoni et al. "Bidirectional molecule generation with recurrent neural networks". In: *Journal of chemical information and modeling* 60.3 (2020), pp. 1175–1183.

[20] Anvita Gupta et al. "Generative recurrent networks for de novo drug design". In: *Molecular informatics* 37.1-2 (2018), p. 1700111.

[21] Nicolae C Iovanac and Brett M Savoie. "Improved chemical prediction from scarce data sets via latent space enrichment". In: *The Journal of Physical Chemistry A* 123.19 (2019), pp. 4295–4302.

[22] Xiwen Jia et al. "Anthropogenic biases in chemical reaction data hinder exploratory inorganic synthesis". In: *Nature* 573.7773 (2019), pp. 251–255.

[23] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. "Junction Tree Variational Autoencoder for Molecular Graph Generation". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, Oct. 2018, pp. 2323–2332. URL: http://proceedings.mlr.press/v80/jin18a.html.

[24] Steven Kearnes et al. "Molecular graph convolutions: moving beyond fingerprints". In: *Journal of computer-aided molecular design* 30.8 (2016), pp. 595–608.

[25] Diederik P Kingma et al. "Semi-supervised learning with deep generative models". In: *arXiv preprint arXiv:1406.5298* (2014).

[26] Peter Kirkpatrick and Clare Ellis. "Chemical space". In: *Nature* 432.7019 (2004), pp. 823–824.

[27] Guillaume Klein et al. "OpenNMT: Open-Source Toolkit for Neural Machine Translation". In: *Proceedings of ACL 2017, System Demonstrations*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 67–72. URL: https://www.aclweb.org/anthology/P17-4012.

[28] Gilles Klopman. "Artificial intelligence approach to structure-activity studies. Computer automated structure evaluation of biological activity of organic molecules". In: *Journal of the American Chemical Society* 106.24 (1984), pp. 7315–7321.

[29] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. "Grammar variational autoencoder". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1945–1954.

[30] Eric-Wubbo Lameijer et al. "Evolutionary algorithms in drug design". In: *Natural Computing* 4.3 (2005), pp. 177–243.

[31] Greg Landrum et al. "Rdkit: Open-source cheminformatics software". In: *GitHub and SourceForge* 10 (2016), p. 3592822.

[32] Christopher Lipinski and Andrew Hopkins. "Navigating chemical space for biology and medicine". In: *Nature* 432.7019 (2004), pp. 855–861.

[33] Christopher A Lipinski et al. "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings". In: *Advanced drug delivery reviews* 23.1-3 (1997), pp. 3–25.

[34] Christopher W Murray and Tom L Blundell. "Structural biology in fragment-based drug design". In: *Current opinion in structural biology* 20.4 (2010), pp. 497–507.

[35] Noel M O'Boyle. "Towards a Universal SMILES representation-A standard method to generate canonical SMILES based on the InChI". In: *Journal of cheminformatics* 4.1 (2012), pp. 1–14.

[36] Marcus Olivecrona et al. "Molecular de-novo design through deep reinforcement learning". In: *Journal of cheminformatics* 9.1 (2017), pp. 1–14.

[37] Hitesh Patel et al. "SAVI, in silico generation of billions of easily synthesizable compounds through expert-system type rules". In: *Scientific data* 7.1 (2020), pp. 1–14.

[38] Eric M Rains and Neil JA Sloane. "On Cayley's enumeration of alkanes (or 4-valent trees)". In: *Journal of Integer Sequences* 2 (1999), Art–No.

[39] Jean-Louis Reymond. "The chemical space project". In: *Accounts of Chemical Research* 48.3 (2015), pp. 722–730.

[40] Lars Ruddigkeit et al. "Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17". In: *Journal of chemical information and modeling* 52.11 (2012), pp. 2864–2875.

[41] Johannes Schiebel et al. "High-throughput crystallography: reliable and efficient identification of fragment hits". In: *Structure* 24.8 (2016), pp. 1398–1409.

[42] Philippe Schwaller et al. "Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction". In: *ACS central science* 5.9 (2019), pp. 1572–1583.

[43] Oliver B Scott and A W Edith Chan. "ScaffoldGraph: an open-source library for the generation and analysis of molecular scaffold networks and scaffold trees". In: *Bioinformatics* (Mar. 2020). btaa219. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa219. URL: https://doi.org/10.1093/bioinformatics/btaa219.

[44] Tomaž Stepišnik et al. "A comprehensive comparison of molecular feature representations for use in predictive modeling". In: *Computers in Biology and Medicine* 130 (2021), p. 104197.

[45] Ashish Vaswani et al. "Attention is all you need". In: *arXiv preprint arXiv:1706.03762* (2017).

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Github links will be added once review is over
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We are reporting a generative model which is based on expanding our ability to sample chemical space in a structured way. Error bars therefore are not straight forward to consider.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix 1: Connection between Lattices and Graphs

We begin with some basic model theoretic definitions but specific to computational chemistry.

Definition A.1 (Molecular Space)**.** *We define molecular space as $\mathcal{M}$, the set of all compounds.*

Assume now there exists a function Scaffold: $\mathcal{M} \hookrightarrow \mathcal{M}$. This function is explicitly not bijective, but injective.[2]

Definition A.2 (Scaffold Space)**.** *We define scaffold space as $\mathcal{S}$, the set of all scaffolds of $\mathcal{M}$,*

$$\mathcal{S} = \{\texttt{Scaffold}(m) : \forall m \in \mathcal{M}\} = \texttt{Scaffold}(\mathcal{M}).$$

Remark A.3. *$|\mathcal{S}| < |\mathcal{M}|$ and $\mathcal{S} \subset \mathcal{M}$ as $\texttt{Scaffold}$ is injective.*

Definition A.4 (Representation)**.** *$\mathcal{R}$ is a representation of $\mathcal{M}$ if there exists a injective function $r : \mathcal{M} \hookrightarrow \mathcal{R}$.*

Remark A.5. *The distinction between a molecule and a representation of a molecule is important; however, for this study assume $m \in \mathcal{M}$ is readily interpretable through a SMILE string representation. For any $m \in \mathcal{M}$ there exists a smiles string which is not unique but only refers directly to m. There most certainly is a complication with stereochemistry and enantiomers for which we deal with at a later time. We give no fuss to this semantic/syntactical distinction later and ignore representation for the rest of this text.*

Definition A.6 (Type)**.** *The type of a molecule or scaffold $m \in \mathcal{M}$ is molecule if $m \notin \mathcal{S}$, otherwise m is a scaffold.*

In the next section, we will dive into the strange connection between the parameterized duals of the deterministic functions Scaffold and Predecessor.

A.1 Graphs, Lattices, and Global Structure

Structure theorems are essential to visualizing and working with an object. Those familiar with group theory will understand the sense immediately—structure theorems allow the particulars to disappear and the underlying flavor to appear. For molecules, the structure we present give rise a peculiar distinction between the deterministic join-operation (given a molecule, it's trivial to see its scaffold) and the parameterized meet functions. I will plant the idea that the parameterized functions are creative.

Corollary A.7. *($\mathcal{M}$, $\texttt{Scaffold}$, $\texttt{Expand}_\Phi$, $\texttt{Scaffold}$, $\texttt{Predecessor}$, $\texttt{Successor}_\Phi$) is isomorphic to an undirected graph with node set $\mathcal{M}$ and edge set*

$$\{(e_1, \texttt{Scaffold}(e_1)) : \forall e_1 \in (\mathcal{M} \setminus \mathcal{S})\} \cup \{(e_1, \texttt{Predecessors}(e_1)) : \forall e_1 \in \mathcal{S}\}.$$

The following corollary restates this result by using only the parameterized functions.

Corollary A.8. *($\mathcal{M}$, $\texttt{Scaffold}$, $\texttt{Expand}_\Phi$, $\texttt{Scaffold}$, $\texttt{Predecessor}$, $\texttt{Successor}_\Phi$) is isomorphic to an undirected graph with node set $\mathcal{M}$ and edge set*

$$\{(e_1, e_2) : \forall (e_1 \in \mathcal{S}, e_2 \in \texttt{Expand}_\Phi(e_1))\} \cup \{(e_1, e_2) : \forall (e_1 \in \mathcal{S}, e_2 \in \texttt{Successor}_\Phi(e_1))\}.$$

In fact, the duality of these definitions introduce an essential structure over $\mathcal{M}$ called a semi-lattice

Definition A.9 (Semi-lattice)**.** *A semi-lattice is a partially ordered set where greatest lower bounds exist.*

Definition A.10 (Cone-lattice)**.** *A cone-lattice is a lattice, an upper and lower semi-lattice, which is bounded below.*

Corollary A.11. *($\mathcal{S}$, $\texttt{Predecessor}$) is a bounded lower semilattice and ($\mathcal{S}$, $\texttt{Successor}_\Phi$) is a unbounded upper semilattice. ($\mathcal{S}$, $\texttt{Predecessor}$, $\texttt{Successor}_\Phi$) is a cone lattice.*

Corollary A.11 provides an intuition as to why any enumeration effort of $\mathcal{M}$ will need parameters if its unwilling to store the whole dataset. Given ($\mathcal{S}$, Predecessor) is the deterministic dual of ($\mathcal{S}$, $\texttt{Successor}_\Phi$).

Definition A.12 (Φ-definable)**.** *We say a function $f : \mathcal{M} \to \mathcal{M}$ is Φ-definable iff f is a composition of basic Φ-algebraic functions. Since Φ functions return a set, or distribution, we use the notation $\sim$ to mean either sample from or "is in."*

Example A.13. *Let $s \in \mathcal{S}$. Then $c \sim \texttt{Expand}_\Phi(s)$ implies $\texttt{Scaffold}(c) = s$ and $c \in \texttt{Expand}_\Phi(s)$.*

[2] This should be taken as a weakness, as this assumption is not obvious unless we place some restrictions on $\mathcal{M}$.

The following theorem asserts the existence of a Φ-definable function f which will be our navigation around $\mathcal{M}$.

Theorem A.14. *Given two compounds $c_1, c_2 \in \mathcal{M}$, there exists a function f which is $\Phi-$definable such that $c_2 \sim f(c_2)$*

Proof. Let s_{c_1} and s_{c_2} be the scaffolds of c_1 and c_2. Let G be the undirected graph of scaffolds. If s_{c_1} and s_{c_2} were not connected then G would be not be a lattice. Therefore there must exist a path P from s_{c_1}. Any path can be written in terms of $\texttt{Successor}_\Phi$ or $\texttt{Predecessor}$. Thus $c_2 \sim \texttt{Expand}_\Phi(P(c_1))$ and $\texttt{Expand}_\Phi(P(\texttt{Scaffold}c_1))$ is a Φ-definable function as it is the composition of a Φ-definable path and $\texttt{Expand}_\Phi$. □

In fact, this path P is at most the diameter of $\mathcal{S}$. At first, this is not very informative as the diameter of the graph is very large depending on what is inside $\mathcal{M}$ (are polymers included, for instance). Suppose though we consider some restriction $X \subset \mathcal{M}$ and relativize $\mathcal{S}_X = \{\texttt{Scaffold}(m) : \forall m \in X\}$. In practice, the bounds are much better.

Lemma A.15 (Relativization)**.** *Given a set of compounds X, we can relativize scaffold space to $\mathcal{S}_X = \{\texttt{Scaffold}(m) : \forall m \in X\}$ such that $\mathcal{S}_X$ is a bounded lattice.*

Importantly, theorem A.14 holds on $\mathcal{S}_X$ and X. This is just a sublattice in fact; however, it is easier to think of it as a relativization problem as we will rarely be interested in the lattice containing macromolecules which are roughly defined.

One such restriction rule, as a heuristic of course, is commonly known as Lipinski's rule of five [33]. The rule of five states an orally active drug cannot have more than one violation of the following:

- No more than 5 hydrogen bonds donors
- No more than 10 hydrogen bond acceptors
- A molecular mass less than 500
- An octanol-water partition coefficient (log P) that does not exceed 5

Definition A.16 (Lipinski-like Space)**.** *Let $\mathcal{L} \subset \mathcal{M}$ be Lipinski-like (chemical) space, where*

$$\mathcal{L} = \{m : \forall (m \in \mathcal{M}) \quad m \text{ meets Lipinski's rule of five}\}.$$

Lemma A.17. *Let $X \subset \mathcal{M}$ where $m \in X$ has molecular weight less than 500. Then $\mathcal{S}_X$ has diameter less than 16.*

Proof. The smallest ring is a four ring carbon member (cyclobutane), which has molecular weight approx. 56. Therefore the largest scaffold with weight less than 500 has at most 9 rings. This implies the path length is at most 8 from this large molecule to the center of the graph. This implies at most the diameter of the graph is 16. □

Reachability is directly related with the potential complexity of finding a path. Let $c \in X$ as above. Suppose we are searching for an active drug and c shows no activity. In order to move to a different drug, c', at most 8 applications of $\texttt{Predecessor}$ are possible with 8 applications of $\texttt{Successor}_\Phi$. In particular, we can say within 8 applications of $\texttt{Successor}_\Phi$ from the origin $\emptyset$ we can reach any molecule in $\mathcal{X}$.

The difficulty with applying $\texttt{Successor}_\Phi$ lies in its indeterminacy—it is a generative function that so far has zero ordering of the molecules which belong to that scaffold class. For example, suppose each scaffold branches at most 8 times. Then there are 8^8 possible paths to explore. In reality, the branching is much larger than 8, and thus we are dealing with x^8.