Deep Inverse Design, Discovery, and Optimization of Molecular Structure through 3D Invariant and Multimodal Machine Learning

Orion Walker Dollar

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Jim Pfaendtner, Chair

David Beck

Lilo Pozzo

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

Abstract

Deep Inverse Design, Discovery, and Optimization of Molecular Structure through 3D Invariant and Multimodal Machine Learning

Orion Walker Dollar

Chair of the Supervisory Committee: Associate Professor Jim Pfaendtner Department of Chemical Engineering

Accelerating the discovery of novel materials and molecules with desired functionalities is crucial for continuing our progression towards technical solutions to some of the world's most pressing issues including disease and climate change. The unique structural and atomic makeup of a molecule dictates its functional behavior, yet we still lack a fundamental understanding of the relationship between structure and function that would allow us to quickly predict the effect of structural modifications on a downstream chemical behavior. Learning this mapping would vastly expedite molecular discovery and is at the heart of the field of de novo molecular design. In this work, we explore a variety of generative statistical machine learning methods for approximating the joint probability manifold of molecular structure and function (JPM-SF). There are several key choices that impact a models' ability to accurately learn this manifold, navigate it, and exploit the regions which it predicts to have optimal properties. The choice of molecular representation both fed to and generated by the model determines the amount of structural information embedded within the model. Many properties are dependent on the distances and angles between atoms and thus 3D representations are often necessary to accurately approximate the JPM-SF. However, the complexity of modeling and generating 3D structures elicits an increased computational burden and empirical results suggest that models which approximate the distribution of 1D sequence-based molecular representations are better at replicating the physicochemical property distributions of the training set from which they are drawn. The choice of model architecture determines the inductive priors given to the model. These can include both structural priors such as the rotational and translational invariance of molecular structures with respect to their intrinsic properties, as well as statistical priors such as the dimensionality and type of distribution from which latent variables are sampled. The choice of sampling and optimization methods, in part, determine the efficiency of the model at achieving a particular goal during inference. This can include adopting design criteria beyond those that the model was explicitly trained to predict and leveraging the mathematical structure of latent variables to aid in exploration and decision making. Each of these choices affects the others and thus we must take a holistic approach when designing a de novo design algorithm to tackle a new problem. Herein, we present a detailed look at the characteristics and performance of three generative de novo molecular design methods with respect to these design decisions. These will serve as case studies to compare the effect that the choices enumerated above have on the ultimate utility of such methods in real-world molecular design scenarios.

Table of Contents

List of Figures

List of Tables

Acknowledgements

To all my friends, family, colleagues, mentors, homies, haters, and rivals – thank you for the instrumental role you've played in helping me achieve this goal.

1. Introduction

The deep learning paradigm shift of the 21st century has brought with it an accompanying shift in the field of chemistry.[1] Generative machine learning (ML) algorithms such as variational autoencoders (VAEs),[2] generative adversarial networks (GANs),[3] and denoising diffusion models[4] have unlocked "inverse" design methods in which hundreds of thousands of novel structures can be generated based on their likelihood of exhibiting a desired behavior.[5] These methods allow us to probe chemical space globally rather than locally and provide experimental researchers with a targeted set of candidate molecules that can be further validated in lab-scale or industrial settings. They also allow us to model the complex relationship between a molecules structure and its function and infer how structural modifications will affect its downstream behavior. Thus, inverse design promises to significantly speed up the life cycle of molecular design, discovery, and optimization.[6]

The design of a novel discoidin domain receptor 1 (DDR1) inhibitor drug can serve as an illustrative example. DDR1 is a tyrosine kinase receptor that has been identified as a key target for treating fibrosis. However, designing a small-molecule drug that has both a high binding affinity and high selectivity for DDR1 has proven challenging with traditional methods. This is largely due to the limited structural diversity of known kinase inhibitors and the high structural homology between the druggable ATP binding site of DDR1 and those of other protein kinases.[7] Attempting to exploit this region of known chemical phase space to target DDR1 has primarily lead to highly promiscuous ligands which also inhibit healthy kinase proteins uninvolved in disease.[8]

Alternatively, if we expand our search space to include all drug-like small molecules we allow for the possibility of discovering novel pharmacophores[9] that are more selective towards the DDR1 binding pocket. However, the total number of potential drug-like molecules is innumerably large and searching this space with brute-force methods is completely infeasible.[10] This problem is thus perfectly suited for generative machine learning methods, with which we can perform a global search of chemical phase space that is biased towards regions that are rich in our desired property. The discovery of chemical structures which are dissimilar to the set of known kinases has been observed to play a critical role in this process,[7] demonstrating the potential utility of inverse design methods which facilitate an increased exploration of chemical phase space.

However, the selectivity of a small-molecule inhibitor towards DDR1 is a necessary but insufficient condition for its efficacy and approval as a drug by the FDA. To reach its target, a drug must travel through a set of complex pathways within the human body, avoiding degradation and competitive inhibition prior to arriving at its final destination.[11] It must also be uninvolved in or otherwise avoid the many potential off-target pathways that result in toxic endpoints and exhibit no toxic physicochemical properties.[12,13] If the target is intracellular, the drug must be able to permeate the cell membrane.[14] Finally, the body must be able to metabolize and excrete the drug to avoid toxic buildup.[15] Satisfying each of these conditions depends on the simultaneous co-optimization of a large number of transport and toxicological properties in addition to optimizing for the drugs specific mechanism of action. Unsuccessful optimization of these ADME/T properties represents a significant proportion of preclinical and clinical failures[16] and can cost billions of dollars in wasted research and development.[17]

It can be difficult for chemists to intuit the effect that small modifications to a molecular structure will have on a single physicochemical property, let alone predict the cascading effects that will occur once that molecule is introduced to the immense complexity of the biological systems within the human body. Yet without the ability to accurately predict this behavior, much of the modern drug discovery process amounts to what is essentially high-stakes trial and error. Generative ML again provides a potential solution to this problem.

Data-driven approximation of the joint probability manifold of structure and function (JPM-SF) allows us not only to explore regions of chemical phase space that are far from the region of known drugs, but also predict and optimize the structures of novel molecules with respect to the most important biochemical properties. Methods for approximating the JPM-SF given only partial observation of properties allow us to learn from sparse experimental data[18,19] and reinforcement learning can be used to bias exploration within the JPM-SF based on properties that are not explicitly modeled.[20] In fact, despite their relative novelty, generative de novo design methods have already been used to discover novel DDR1 inhibitors with optimal ADME/T properties that have been experimentally verified in both *in vitro* cell assays and *in vivo* rodent models.[19]

Clearly, generative machine learning models are already powerful tools capable of performing the inverse design, discovery, and optimization of molecular structures. However, there is still much to learn and improve upon. The incorporation of 3D structural information into generative models has only recently been achieved,[21–23] and there are many open questions pertaining to the expressive power[24] and generalizability[25,26] of these networks. Equivariant generative models have become extremely proficient at predicting and designing the structures of large biomolecules,[27–29] yet similar models are still incapable of outperforming traditional methods at low energy conformer generation[30,31] or molecular docking.[32,33] Some methods have been proposed for increasing the transparency and interpretability of molecular candidates proposed by generative models,[34,35] yet the absence of clear causal mechanisms for predictions made by these models may limit their acceptance given how intricately the accuracy of their predictions are tied to human health.

While the research presented herein provides but a small glimpse into the solutions to these problems, I hope that any reader who happens to stumble upon this document may perhaps find some spark of inspiration in the questions that are posed and the results that follow. We first begin with a brief overview of the most important computational chemistry and machine learning methods used for inverse design in Chapter 2. Chapter 3 introduces the transformer VAE and provides a justification for using Shannon's information entropy as a model ensemble property that determines downstream sampling performance. Chapter 4 explores the use of multimodal conditioning in large-scale language models (LLMs) to provide an accessible text-based interface for de novo molecular design to chemists with little-to-no background in machine learning. Chapter 5 builds upon the ideas introduced in Chapter 3 by incorporating an E(3) invariant encoder used to design and optimize 3D molecular structures with respect to their quantum chemical properties, and by demonstrating conditional sampling along disentangled latent variables. Finally, the major insights from Chapters 3-5 are summarized and potential directions of future work suggested in Chapter 6.

2. Framework & Methods

The umbrella of inverse design covers many different types of models, algorithms, and computational tools. However, there are three components to inverse design that are consistent across almost all use cases.

I. Machine-readable representations of molecular structure
II. Generative machine learning architectures
III. Optimization and sampling methods

Developing a proper understanding of each of these components is key to choosing the methods that will be best suited for a particular drug design or materials discovery task. These choices, while varied, are not independent from one another. For instance, the use of three-dimensional molecular structures necessitates a generative architecture that can ingest and interpret 3D information.

In addition to the choice of methodology, the success or failure of an inverse design model is highly dependent on the availability of high-quality data and the selection of evaluation metrics that adequately describe the functional behavior the model is intended to approximate. This last point is particularly relevant in relation to Goodhart's law – "when a measure becomes a target, it ceases to be a good measure."[36] In molecular design, the direct optimization of a model towards the metric by which the model is evaluated can often result in many false positives due to errors or noise in the predictions that lead to over-optimization towards spurious correlations in the training data.[37] Thus, the practitioner must be careful to avoid optimizing on the same metric that they use to evaluate.

We will now explore each of these components of the generative inverse design pipeline – data collection, data creation, molecular representations, generative machine learning architectures, and optimization methods – in more detail.

2.1 Data Collection

There are a number of existing datasets that are often used to train generative de novo design models without the need to create any data from scratch. PubChem,[38] ZINC,[39] and GDB-17[40] each contain on the order of hundreds of millions to hundreds of billions unique small molecule structures. Due to their size, they are well-suited for pretraining data hungry large-scale generative models that can then be fine-tuned for a particular task. However, the quality of the structures in these datasets can be quite low[41] and its often necessary to apply copious amounts of preprocessing to filter these molecules prior to training a model.

Additionally, none of the aforementioned datasets contain any property labels, thus they are only suitable for unsupervised pretraining. ChEMBL, on the other hand, includes the results of millions of experimental biological assays that measure the affinity of small molecules to many different disease-related protein targets.[42] It has become a staple of models used for drug discovery, however the amount of measured data per small molecule and per target is highly variable so it tends to reinforce study of protein targets which have already been widely explored.

There are a growing number of datasets being published that precompute the low energy 3D conformers for a large number of drug-like small molecules. QM9 served as the gold standard for

many years, containing quantum chemical property calculations for ~134K small molecules with no more than nine heavy atoms.[43] As available compute has grown and interest in scaling 3D generative models to larger molecules has increased, larger 3D conformer datasets have been released. QMugs repeats the procedure from QM9 on the ChEMBL dataset.[44] GEOM-Drugs also generates many conformers of structures from ChEMBL, however they compute fewer properties for fewer molecules than QMugs while generating many more unique conformers per structure.[45] Other examples of small molecule conformer datasets include PubChemQC[46] and ANI-1.[47] Each of these datasets contain subtle differences in construction that determine their suitability for a given task, and we refer the reader to the literature for further details on these distinctions.

The geometric relationship between small molecule binders and a protein binding pocket plays a vital role in determining the protein-ligand affinity. DUD-E[48] and PDBBind[49] are both datasets that contain the static bound structures of crystal ligands within a protein pocket. These datasets help link the experimentally measured bioassays to specific pharmacophore-based hypotheses of binding and allow for models to be trained to conditionally generate 3D ligand structures based on the shape of a known binding pocket.[50,51] Similar to the limitations of ChEMBL, datasets derived from experimentally determined crystal structures of protein-ligand complexes underrepresent certain ligands or pockets with sparse amounts of experimental measurements. The CrossDocked dataset attempts to alleviate this by combinatorically expanding on PDBBind by performing automated docking simulations of every ligand in every pocket that passes a structural homology threshold compared to its known pocket.[52]

2.2 Data Creation

Oftentimes the publicly available data will not be sufficient to train a model to address the specific problem that a researcher wishes to tackle. In this case, it may be necessary to look towards tools for efficiently and accurately measuring molecular properties in a high-throughput manner. This data can be used to both supplement the model during training and validate the predictions made by the model during inference.

Computational chemistry tools in particular are extremely useful for this purpose, as they provide a low-cost method for generating reliable data very quickly. In general, the quality of the measured data is a function of the cost and time invested to acquire that data, so it is important to evaluate the quality of data needed for a specific application and plan accordingly. It's common to pretrain a general model on large amounts of lower quality or noisy data to first learn an approximation of the JPM-SF prior to fine-tuning on lower amounts of higher quality data. A few methods for data creation at varying levels of quality and cost are detailed below.

Molecular Descriptors

Molecular descriptors have played a large role in the development of generative molecular design models and other computational or information-based approaches to understanding chemistry. They are derived from a wide variety of fields including quantum chemistry, information theory, organic chemistry, and graph theory,[53] and can be used as input features or targets for machine learning algorithms. They range from very simple descriptors like molecular weight to complex topological indices such as the E-state index which gives a numerical description of the electrotopological state, S_i, of each atom within a molecule

$$S_i = I_i + \sum_{j=1}^{A} \frac{I_i - I_j}{(d_{ij} + 1)^k} \quad (2.1)$$

where I_i is the intrinsic state of the *i*th atom and d_{ij} is the topological distance between the *i*th and *j*th atom.[54] Because of their abstract and varied nature, you must often sacrifice interpretability when using them however algorithms such as LASSO can trim the number of descriptors considered to only those most important for the prediction task thereby improving the interpretability.[55] Cheminformatics packages such as RDKit[56] or Mordred[57] allow for the quick calculation of hundreds to thousands of descriptors for millions of individual molecular structures making them an extremely useful tool for inverse design.

Molecular Dynamics Simulations and Quantum Chemistry

Molecular dynamics (MD) simulations provide a higher level of detail and accuracy than molecular descriptor calculations at the cost of increased computational overhead. Highly accurate forcefields have been developed for both atomistic-[58] and quantum-level[59] simulations that allow us to retrieve bulk thermodynamic and electronic data for a variety of organic and inorganic molecular structures. An automated high-throughput framework for building, running, and analyzing MD systems can be a viable alternative to molecular descriptor calculations when the set of available descriptors are not sufficient in capturing the desired behavior. Such a framework can also be used to generate an ensemble of 3D conformers if a labeled dataset does already exist but lacks the molecular structural information needed to accurately model the JPM-SF.

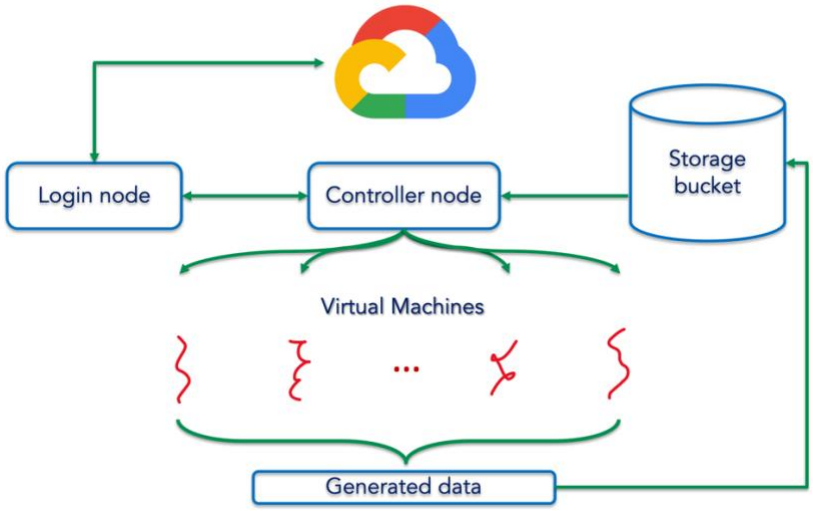


Figure 2-1. A framework for integrating high-throughput molecular dynamics simulations with Google Cloud (adapted from figure created by Melanie Huynh). Login and controller nodes allow for many virtual compute engines to be spun up in parallel to generate massive amounts of data that can be stored in the cloud.

Previous studies have demonstrated the utility of building a large database from high-throughput simulation data, including the previously mentioned 3D conformer datasets as well as the Harvard Clean Energy Project[60] which provided first-principles calculations for over 2.3 million organic photovoltaics. Figure 2-1 shows a high-throughput framework for generating MD data on Google Cloud Platform (GCP). Due to the large number of virtual machines (VMs) that can be created in

parallel, cloud computing has the potential to greatly reduce the total wall time required for a campaign of simulations compared to traditional supercomputing clusters which have a hard cap on the number of nodes that can be used simultaneously. We are already witnessing an increase in the availability of this type of data as cloud computing hardware has become further integrated with scientific computing software.

Literature Mining

A potentially massive source of untapped data exists in the form of unstructured scientific text available in journal articles, conference proceedings, and other scientific ventures that have been published online but not parsed into any structured format. The rate of publication is growing at an exponential rate,[61] yet a significant amount of this information will never see the light of day (Fig. 2-2). Even considering only those studies which have reached a wider audience, it takes an extraordinary amount of manual effort to extract and compile the observations contained within unstructured text, graphs, figures, and tables into a self-consistent database.

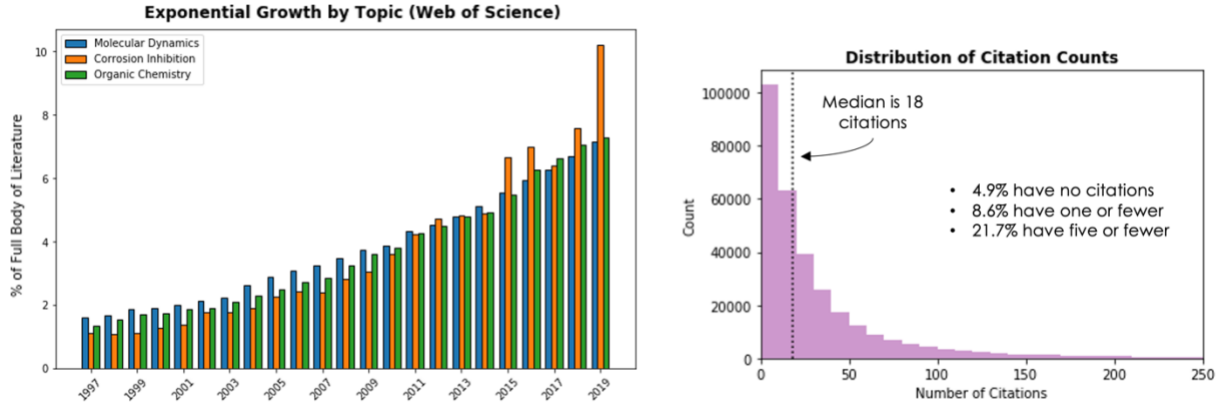


Figure 2-2. Papers are being published at an increasing frequency yet over a fifth of all publications have less than or equal to five total citations. Even those that are highly cited must be parsed and structured to be able to make use of them with external ML models.

Natural language processing (NLP) methods are a natural choice for learning structure from text in an automated manner. Named entity recognition models are able to identify and predict the entity types within a body of literature.[62] Other models have been improved to not only predict the location and identity of entities, but also the relationship between those entities.[63] The entity and relationship categories are inductive biases built into these models and must be chosen ahead of time. There are many examples of models that have been explicitly trained to identify entities and relationships in scientific text,[64,65] including ChemDataExtractor which uses a linear chain conditional random field (CRF) model to extract named chemical entities from unstructured text.[66]

A major drawback of these methods is that large amounts of labor-intensive labels must be generated to train entity recognition models in a supervised manner. For chemistry in particular, labels often require a high degree of domain expertise making them difficult to collect in bulk across a variety of different fields. The incredible maturation of LLMs in the last year also threaten to make supervised entity recognition models obsolete in the near future. While LLMs still tend to hallucinate information at an unacceptable rate when they are queried directly,[67] strategies for mitigating this such as chain-of-

thought reasoning[68] and embedding-based information retrieval[69,70] have already begun to improve the reliability of these models.

Experimentation

In most design cases, while the model may be improved by incorporating computational results, it will ultimately be necessary to fine-tune and/or validate on real experimental data. A similar tradeoff between cost and quality exists for this type of data. For instance, *in vitro* cell assays are typically used to screen and eliminate drug candidates prior to *in vivo* testing in animal models. Only after the drug passes both of these stages can its expected behavior be validated in human clinical trials. The cost of clinical trials far exceeds that of bench-scale experimentation so drug candidates must pass a series of tests before making it to this stage.

There already exists a number of high-throughput experimental workflows[71] that can be used to supplement computational data creation, and methods that use machine learning to guide experimentation are being developed to minimize the number of required experiments necessary to reach a design goal. For instance, robotics and microfluidic platforms have already been tested for their suitability in AI-assisted nanoparticle synthesis.[72] These devices can be integrated with reinforcement learning or Bayesian optimization for *online* guidance on the selection of additional experimental conditions to test.[73,74] If high-throughput data collection or automated experimentation are simply not possible, there are methods that have demonstrated experimental success when fine-tuning on very small amounts of experimental data (less than 100 samples).[75,76]

2.3 Molecular Representations

The choice of molecular representation largely depends on the structural priors that are most important for a given task. 1D sequence representations can capture some characteristics of the molecular graph like ring openings/closures, branches, and atomic valency,[77,78] but due to their sequential nature this information is often stored far in sequence space and requires long-range syntactic parsing to generative valid molecules (see Fig. 2-3b).

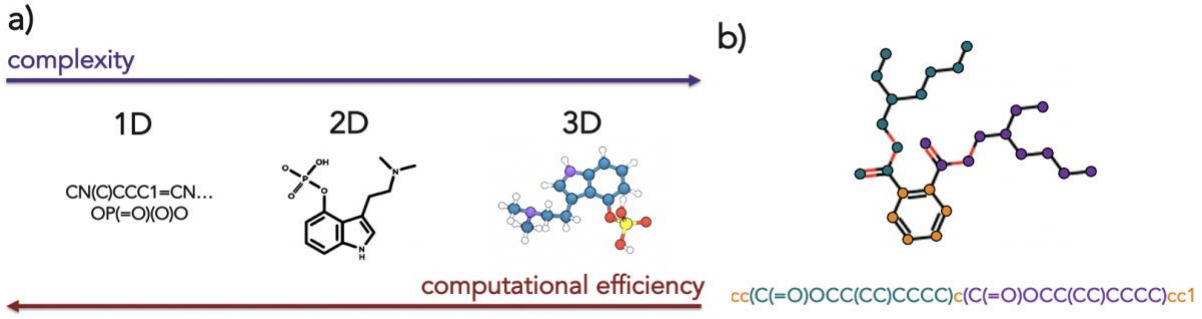


Figure 2-3. Machine-readable molecular representations. a) As the dimensionality of the molecular representation is increased, the complexity and computational cost of working with these representations increases as well. b) The 2D graph structure and 1D SMILES string of DEHP. The sequential nature of SMILES strings causes the atoms that comprise the phenyl group (orange) to be separated far from one another in sequence space by the long, branching ester chains (green, purple).

2D graph representations explicitly represent the atomic connectivity with a set of nodes, edges, and their adjacency matrix.[79] Information is typically passed between connected nodes with message passing neural networks that compute messages between nodes based on their atomic and bonded features and update the node embeddings based on an aggregate of all the incoming messages to a particular node.[80] 3D geometric information can be integrated into these graphs as scalar distances between nodes,[81] vectors which encode the angles between nodes,[82] or tensors that capture higher order geometric features.[23]

There is a compromise that must be made between expressivity and complexity when choosing between 1D, 2D, and 3D molecular representations (Fig. 2-3a). Three-dimensional graph representations are more memory and compute intensive[83] and are more difficult to generate with high fidelity.[84] There is also growing evidence that models which generate higher order representations are less adept at recreating the physicochemical property distributions of the training set[85] (see Section 5.6). However, the importance of the role that molecular geometry plays in its function cannot be understated and it remains a continued challenge to mitigate the problems of working in 3D while still leveraging the increased structural detail provided within these representations. Additional literature on the most relevant molecular representations used today is available in Section 5.3.

2.4 Generative *de novo* Molecular Design Architectures

As the ultimate goal of inverse design is to model the conditional likelihood that a molecular structure exhibits some property measure, the model must be able to generate novel molecular candidates that have not been included in its training data. While there is a rich body of literature on metaheuristic optimization methods like the genetic algorithm[86] or Markov decision processes,[87] we focus on deep generative models which learn to numerically approximate the JPM-SF and sample from it based on some design criteria. These methods include GANs, diffusion models, VAEs, and transformers, although we focus exclusively on the VAE and transformer in the work that follows.

Variational Autoencoder

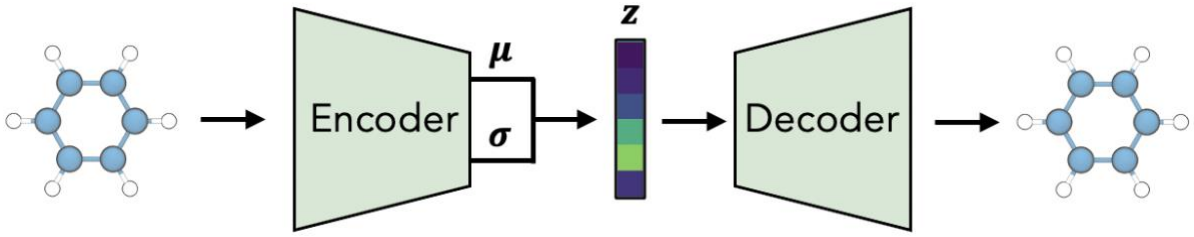


Figure 2-4. Architecture of the variational autoencoder. The joint probability manifold of structure and function (JPM-SF) is modeled as a set of latent variables, **z**. Each latent variable is stochastically sampled from a Gaussian distribution with mean, , and standard deviation, , that are learned during training. During inference, latent variables are sampled as independent Gaussians which allows for the generation of novel structures.

The variational autoencoder is a compression algorithm that returns a numerical representation of the input data that is both maximally expressive and contains minimal noise.[2] The input, $\boldsymbol{x}$, is first sent through an encoder where it is compressed to an intermediate latent representation, $\boldsymbol{z}$, such that

$d_{latent} \ll d_{input}$ where d is the dimensionality of a given layer. The latent representation is then sent to a decoder that attempts to reconstruct the original input as accurately as possible (Fig. 2-4).

A key feature of the VAE is the use of a set of Gaussians to represent the prior distribution of the latent variable, $p(\boldsymbol{z})$, so that knowledge of the marginal likelihood of $\boldsymbol{x}$ allows us to sample directly from the latent space to generate new samples. In practice, we enforce this by minimizing the Kullbach-Leibler divergence (KLD)[88] between $\boldsymbol{z}$ and the standard normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ along with the cross-entropy loss between the output and the ground truth molecular structure. As the latent space is a well-defined continuous numerical space, we can use numerical optimization methods like particle swarm optimization (PSO),[89] gradient descent,[90] or Gaussian process regression (GPR)[91] to search for latent variables that correspond to molecular structures with optimal properties.

The VAE is agnostic to the types of neural network layers used to parameterize the encoder and decoder networks. Thus, it can learn from 1D, 2D, or 3D input molecular representations and even translate from one representation to another.[92] It's also common to append a feed forward property prediction layer that takes the latent variable as an input such that the property is directly embedded within the latent space. This has the effect of smoothing the latent space with respect to the property making it easier for numerical optimization methods to find global minima.[93]

Transformer

The transformer is a natural choice for working with 1D sequence representations of molecules, as the attention mechanisms within a transformer layer can capture the long-range sequence dependencies described in Section 2.3. Transformers are trained such that they learn to predict the next token in a sequence given the prior sequence up to that point. To generate novel sequences during inference, the transformer autoregressively predicts one token at a time until a stop token or the maximum sequence length is reached. If a transformer layer is being used as the decoder network of a VAE, the autoregressive prediction process can be conditioned on the learned latent representations of molecular structures predicted by the encoder.

Scaled dot-product attention computes the value of each hidden layer as a matrix multiplication between a set of queries (Q), keys (K), and values (V). The query and key matrices are used to calculate attention weights that weigh the importance of the sequence embedding information contained within V

$$h_{out} = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (2.2)$$

where d_k is a scaling factor that controls the size of the gradients within each layer.[94] Q, K, and V contain information pertaining to the entire sequence and thus multiplying these matrices allows tokens very far from each other in sequence space to pass information between one another. The learned embedding, h_{out}, can either be used as the input to the next transformer layer or as the learned molecular embedding for property prediction tasks or generative conditioning.

Message Passing Graph Neural Networks

Message passing neural networks (MPNNs) are used to learn how chemical features in molecular graphs are distributed along covalent edges. Graphs can be used to encode both 2D and 3D representations of molecular structure (Section 2.3) thus MPNNs are a natural choice for working with higher order molecular representations. MPNNs typically follow the same format where messages between nodes connected by covalent bonds are first computed according to a message function

$$\boldsymbol{m}_{ij} = Message(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{\varepsilon}_{ij}) \quad (2.3)$$

where $\boldsymbol{h}$ is the embedding of nodes *i* and *j*, and $\boldsymbol{\varepsilon}_{ij}$ is the edge feature between those nodes. Node embeddings are then updated based on an aggregate of all messages into that node according to

$$\boldsymbol{h}_i' = Update(\boldsymbol{h}_i, \sum_{j \in \mathcal{N}} \boldsymbol{m}_{ij}) \quad (2.4)$$

where $\mathcal{N}$ is the list of all neighbors of node *i*. The parameterization of the message and update functions is dependent on the presence or absence of geometric information and the structure of that information. In the simplest example, node/edge embeddings are concatenated and passed through feed-forward layers to learn messages and update nodes.[79] Alternative layers that maintain the rotational, translational, and permutational equivariance of messages and node embeddings are becoming increasingly more common and are an active area of research.[21,95–98]

2.5 Optimization Methods

One of the primary benefits of the VAE architecture is it projects discrete molecular structures into a continuous numerical latent space. While optimization of this space is challenging due to the high dimensionality and rugged nature of the JMP-SF, there is a large existing body of literature and active research community working on improving methods for numerical optimization within a high-dimensional latent space.[99–103] Many of these methods are designed as black-box optimizers and can easily be integrated into a generative VAE model for molecular design without the need to modify the existing architecture or retrain a model from scratch.

In addition to numerical optimizers, there are numerous ways to incorporate optimization directly into the network architecture. Conditional generation is a common method in Bayesian inference models, where the desired molecular properties can be defined by the user and used to condition the generative process based on the likelihood that a molecule exhibits those properties.[104,105] This method depends on the model being given plentiful training data across all possible conditions the user might be interested in exploring. Models that learn disentangled molecular representations can also be used to control generation in an unsupervised manner.[106,107] Disentangled latent variables are independent from one another and are often correlated with independent features of the training set, such as molecular weight or radius of gyration. Finding these disentangled variables and their corresponding features allows for controlling these features by interpolating along a single variable.

3. Attention-based Generative Models for *de novo* Molecular Design[1]

3.1 Abstract

Attention mechanisms have led to many breakthroughs in sequential data modeling but have yet to be incorporated into any generative algorithms for molecular design. Here we explore the impact of adding self-attention layers to generative β-VAE models and show that those with attention are able to learn a complex "molecular grammar" while improving performance on downstream tasks such as accurately sampling from the latent space ("model memory") or exploring novel chemistries not present in the training data. There is a notable relationship between a model's architecture, the structure of its latent memory and its performance during inference. We demonstrate that there is an unavoidable tradeoff between model exploration and validity that is a function of the complexity of the latent memory. However, novel sampling schemes may be used that optimize this tradeoff. We anticipate that attention will play an important role in future molecular design algorithms that can make efficient use of the detailed molecular substructures learned by the transformer.

3.2 Introduction

The design and optimization of molecular structures for a desired functional property has the potential to be greatly accelerated by the integration of deep learning paradigms within existing scientific frameworks for molecular discovery. Traditional "direct" design approaches, in which a set of molecules are selected based on expert intuition and tested for a given property, are often time-consuming and require extensive resources to explore a small, local region of chemical phase space.[6] By contrast, "inverse" approaches, in which structures are derived based on their likelihood to exhibit a given property value, are desirable as they are far less limited in scope and allow for high-throughput screening of thousands to hundreds of thousands of structures.[5] Given the size and complexity of chemical phase space,[10] successful implementation of an inverse design algorithm would allow researchers to reach global structural optima more rapidly thereby increasing the speed of discovery.

A variety of deep generative model architectures have been explored for this purpose,[109] with a particular focus given to the variational autoencoder (VAE).[41,92,110–113] A VAE is capable of broadcasting a machine-interpretable representation of molecular structure (e.g. a SMILES string,[77] SELFIES string[78] or molecular graph[114]) to a dense, continuous latent space or "model memory". This memory has several unique features that make VAEs promising for inverse design: (i) It can be embedded with a property and thus serve as an approximation of the joint probability distribution of molecular structure and chemical property. (ii) During training, it will organize itself meaningfully so that similar molecules are near each other in phase space. (iii) Due to its mapping from discrete to continuous data, it can be navigated with gradient-based optimization methods.[93]

Despite these benefits, generative VAE models suffer from a set of complicating issues that have been the focus of much recent work. Although more robust than their adversarial counterparts, VAEs are still subject to experiencing posterior collapse in which the decoder learns to ignore the latent memory altogether and reconstruct a fuzzy approximation of the input distribution.[115] On the other hand, even

[1] Reproduced in part with permission from O. Dollar, N. Joshi, D. Beck, and J. Pfaendtner. Attention-based generative models for de novo molecular design. *Chemical Science*, *12*(24), 8362-8372 (2021)[108] © The Royal Society of Chemistry 2021

with a meaningful posterior there are often pockets of phase space within the latent memory that do not map to any valid chemical structures. Many recent innovations in architecture, featurization and hyperparameter selection have centered around these problems and have proven quite successful at improving reconstruction accuracy and sampling validity.[114,116,117]

However, we lack a holistic view of the effect of these improvements on the practical utility of a model's latent memory. For instance, metrics to examine the diversity and novelty of sampled molecules are not well-defined.[118] These traits are arguably as important as validity, if not more so. Generating samples is orders of magnitude faster than training and a model that can generalize to regions of chemical phase space far outside the training set is valuable for exploration. Although fewer studies have evaluated generative VAE models in this way, the results reported in the Moses Benchmarking Platform indicate that there is still significant room for improvement.[119]

The rapid technological progression within the field of natural language processing (NLP) may offer some hints towards a future where AI-designed molecules are the norm rather than the exception. Despite the overwhelming number of similarities between model architectures used for molecular generation and those used for NLP, the state-of-the-art in the former lags notably behind that of the latter. While attention mechanisms have been used in the field of chemistry for tasks like graph-based analyses of chemical structure,[120] atom-mapping,[121] and organic reaction predictions,[122] they have not yet been incorporated into any context-independent generative algorithms. Yet the long-range syntactical dependencies learned by attention models have been shown to be greatly beneficial for generative tasks in other domains including the generation of natural language[123] and composition of original pieces of music.[124] Such models have also shown a surprising aptitude for style with their ability to combine wit, poetic prose, and the tenets of philosophy into cogent metaphysical self-reflections on the meaning of virtual existence[125,126]. Although perhaps not as amusing, we anticipate they may exhibit a similar sense of coherence when tasked with exploring novel chemistries.

An examination of the performance of standard recurrent neural networks (RNN), RNN + attention and transformer VAE architectures for the purpose of molecular generation follows. We show the effect of attention on reconstruction accuracy for both the ZINC and PubChem datasets. Novel metrics are proposed that define the models' ability to explore new regions of chemical phase space and compare the relative information density of the latent memory. We show that for all model types there exists a relationship between sample validity and exploration that mimics closely the tradeoff between complexity and generalization within an information bottleneck. Finally, we suggest a simple sampling scheme that offers a compromise between the two and look towards a future where we may optimize this directly during training with more precise control during the nascent development of the latent memory.

3.3 Results and Discussion

Variational Autoencoder and the Information Bottleneck

A VAE consists of an encoder that takes a sequence as input, i.e., a SMILES string, and a decoder that attempts to reconstruct the input as accurately as possible.[2] Prior to decoding, the encoder transforms the input, **x**, into an intermediate latent representation, **z**, that serves as the "model memory." Information is bottlenecked between the encoder and decoder such that $d_{latent} \ll d_{input}$ where d is the dimensionality of a given layer. In this sense a VAE can be thought of as a compression algorithm that produces compact, information dense representations of molecular structures. The

encoder learns how to compress the input data and the decoder learns how to reconstruct the full sequence from the compressed representation (Fig. 3-1).

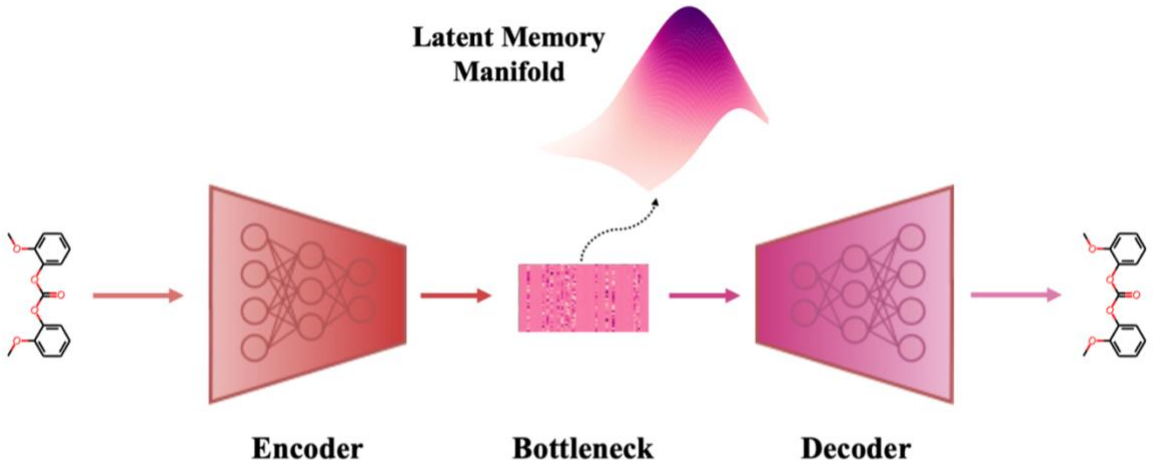


Figure 3-1. Major structural components of the VAE architecture. A machine-interpretable representation of a molecular structure is sent to an encoder where it is compressed to a dense latent representation within the bottleneck. Each of the compressed molecular embeddings represent one point within a larger probability manifold aka "model memory". During training, the model learns to fit this manifold to the true probability distribution of the input data. To ensure the compressed embeddings contain structurally meaningful information, they are sent to a decoder which learns to reconstruct the original molecular structure.

The training objective seeks to minimize the reconstruction loss between the input and output while simultaneously learning the ground truth probability distribution of the training data. The latter half of this objective is especially important to the generative capacity of the model. Knowledge of the marginal likelihood, $p(\boldsymbol{x}|\boldsymbol{z})$, allows us to directly sample new data points by first querying from the model's memory, $\mathbf{z}$, and then decoding. To achieve this, we assume the true posterior can be adequately approximated by a set of Gaussians. The Kullbach-Leibler divergence (KLD)[88] between $\mathbf{z}$ and the standard normal distribution $\mathcal{N}(0,1)$ is minimized alongside the reconstruction loss and thus the full objective function can be formalized according to the variational lower bound as

$$log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \geq \mathcal{L}(\theta,\phi;\boldsymbol{x},\boldsymbol{z}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \beta D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \quad (3.1)$$

where the term on the left is the reconstruction loss of the decoder, $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, and the term on the right is the KLD loss between the encoder output, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, and the standard normal distribution, $p(\boldsymbol{z})$. The KLD loss is scaled by a Lagrange multiplier, β, that controls the relative magnitude of the two terms. This architecture is known as a β-VAE and is a more general form of VAE (β=1).[127]

Intuitively, the addition of gaussian noise can be thought of as a way to increase the "spread" of samples within the latent memory. Rather than encoding individual molecular structures as a single point in phase space, it encodes them as a probability distribution. This allows the model to smoothly interpolate between the continuous representations of known molecular structures and make informed inferences outside of the set of training samples.

The latent memory can also be analyzed within the framework of information bottleneck (IB) theory.[128] During compression, there is an unavoidable tradeoff between the amount of useful

information stored in the model's memory and the amount of low information complexity stored in the model's memory (here and throughout we allude to Tishby et al.'s definition of complexity that is analogous to the information density of the bottleneck; see Appendix A for more details).[129] The IB objective can be written as

$$\max_{\theta,\phi}\left[I\left(q_\phi(\boldsymbol{z}|\boldsymbol{x});\ p_\theta(\boldsymbol{x}|\boldsymbol{z})\right) - \beta I\left(\boldsymbol{x}; q_\phi(\boldsymbol{z}|\boldsymbol{x})\right)\right]^{32} \quad (3.2)$$

where I is the mutual information between two variables. We seek a solution that is both maximally expressive and compressed. Since there is rarely a unique solution to the reconstruction objective, the β parameter discourages the model from finding a needlessly complex (but still valid) local minimum. Thus, in addition to controlling the "spread" of information, the KLD term can be interpreted as a filter of irrelevant information with pore size 1/β. It will be useful to keep this framework in mind as we observe the development of the latent memory during training.

Adding Attention to the VAE

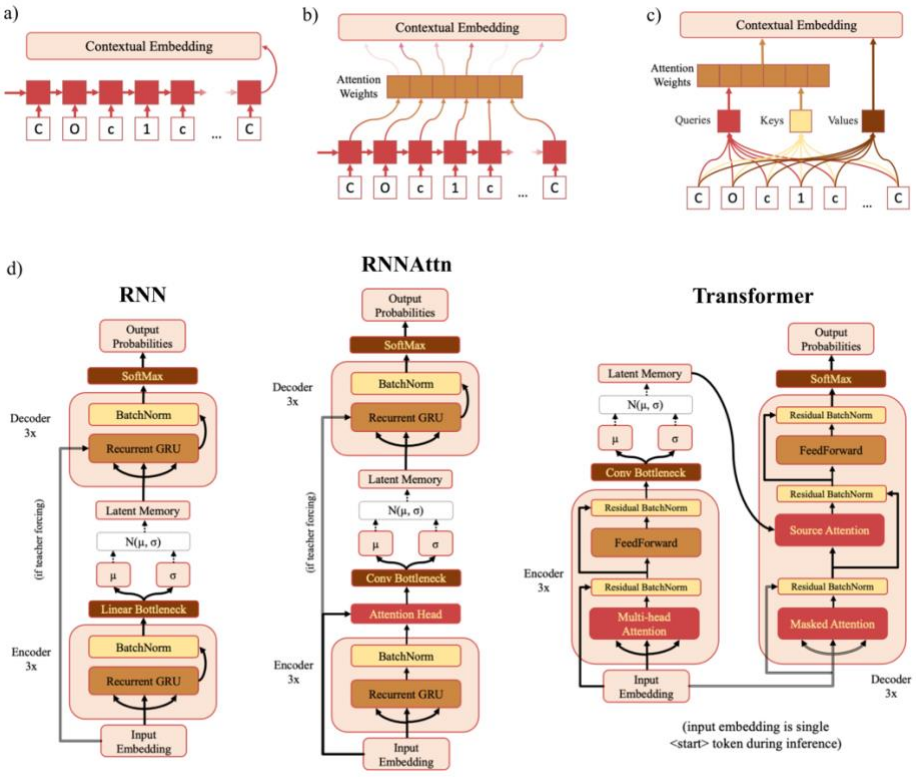


Figure 3-2. Model diagrams. a-c) Schematic illustrations of the sequential layers for each model type – RNN (a), RNNAttn (b) and Transformer (c). Each model consists of six sequential layers – three in the encoder and three in the decoder. The output contextual embeddings of each layer are used as the inputs for subsequent layers within the model. d) Full schematics for each model type. The RNN model consists of three recurrent GRU layers in both the encoder and decoder. The RNNAttn model has the same architecture as the RNN with the addition of a single attention head after the final recurrent GRU layer in the encoder. The transformer is modeled after the original implementation as reported by Vaswani et al.[33] However, rather than passing the output of the encoder directly into the source attention layer, the encoder output is first stochastically compressed and then fed into the decoder.

In standard RNNs, the first recurrent cell takes the first element of the sequence and outputs a hidden state. That hidden state is then propagated down the sequence with each subsequent recurrent cell taking the previous cell's hidden output and the next sequence element as inputs until the entire sequence has been traversed. The final hidden state is the "contextual embedding" of the sequence (Fig. 3-2a). In some architectures the contextual embedding and the latent memory may be the same size. However, oftentimes there will be an additional set of linear bottleneck layers that further compress the output of the encoder GRU layers ($d_{encoder} \rightarrow d_{latent}$).

In attention-based recurrent models (RNNAttn), the flow of information proceeds similarly to a standard RNN. However rather than only using the final hidden output state, a weighted combination of all the hidden states along the sequence is used as the contextual embedding (Fig. 3-2b). The attention weights are learned during training by letting the input sequence "attend" to its own hidden state matrix. This allows the model to eschew the linearity imposed by the RNN architecture and learn long-range dependencies between sequence elements.

Transformer (Trans) models remove recurrence altogether and exclusively use attention head layers.[94] The inputs are a set of keys, values and queries transformed from the initial input sequence that are sent through a series of matrix multiplications to calculate the attention weights and the contextual embedding (Fig. 3-2c). The set of values are analogous to the hidden state matrix output of an RNN and the attention weights are determined by matrix multiplication of the keys and queries. Transformers have the advantage of reducing the path length of information traveling through the model and are highly parallelizable.

The concepts of attention and the variational bottleneck have rarely been used in tandem. Of those studies that have surveyed this type of model, all have used natural language tasks as the basis of their evaluations. A variational attention-mechanism was used for sequence-to-sequence models[130] and a few novel variational transformer architectures have recently been proposed.[131–133] We opt for simplicity, adapting the architecture from Vaswani et al.[94] with as few modifications as possible. This allows us to easily compare the bottlenecks of different model types and is sufficient for the task given the much smaller vocabulary size of SMILES strings compared to NLP vocabularies.[134] Full schematics for each model type are shown in Fig. 3-2d and model dimensions listed in Table 3-1. In addition to the model types listed above, we also trained the Moses implementation of a SMILES-based β-VAE with the hyperparameters suggested by Polykovskiy et al.[119] Trained model checkpoint files and code for training models and generating samples is available at https://github.com/oriondollar/TransVAE.

Table 3-1. Model Architectures. The dimensionality of the model (d_{model}) is defined as the size of the sequential layers. Recurrent model names are written as ModelType-{d_{model}}. Transformer model names are written as Trans{$d_{feedforward}$ / d_{model}}x-{d_{model}}. All models used in this study have a latent dimensionality of size 128.

Model Type	d_{model}	d_{latent}	$d_{feedforward}$
RNN-128	128	128	n/a
RNN-256	256	128	n/a
RNNAttn-128	128	128	n/a
RNNAttn-256	256	128	n/a
Trans1x-128	128	128	128
Trans4x-128	128	128	512
Trans1x-256	256	128	256

Trans4x-256	256	128	1024

Impact of Attention

We first analyze the models' ability to reconstruct molecules from the ZINC and PubChem datasets to determine the role attention plays in learning molecular structure. One of the original motivations for the use of attention was to increase the length of sentences that could be accurately translated by machine translation models.[135] Thus, we expect a similar increase in accuracy when encoding and decoding longer SMILES strings.

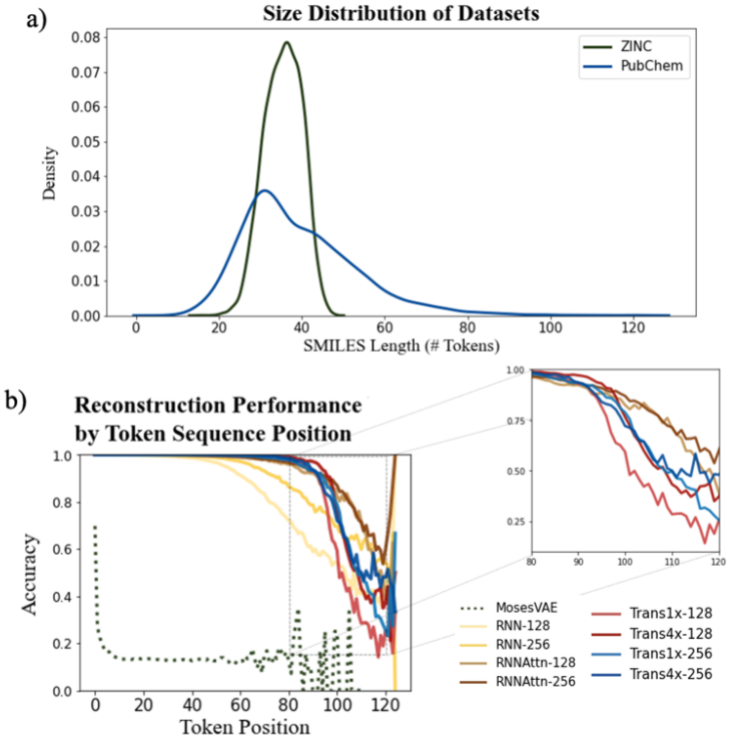


Figure 3-3. Assessing model reconstruction performance on the PubChem dataset (trained for 60 epochs). Input data molecular size distributions (a) and reconstruction accuracies for all model types as a function of the token position (b). Zoomed comparison of attention-based models (inset)

Fig. 3-3a shows the distribution of SMILES string lengths for both datasets where length is determined by the number of tokens (excluding padding, start and stop tokens). The length of a SMILES string is highly correlated with its molecular weight (Fig. A-5) and can be used as a proxy for molecular size. It is clear that by this metric the PubChem dataset has a broader distribution of sizes than ZINC. Both have approximately equal mean lengths (35.4 tokens for ZINC vs. 39.8 tokens for PubChem) however the PubChem data is significantly right skewed with a maximum token length over 50 tokens longer than the maximum within the ZINC dataset.

We can see the downstream effect that widening the molecular size distribution has on reconstruction accuracy in Fig. 3-3b where we show the average reconstruction accuracy for all tokens at a given position within the sequence. With the exception of the Moses architecture, all of the models exhibit high fidelity reconstruction on the ZINC dataset, regardless of model type or model size (Fig. A-6/Table A-2). However, accuracy decreases when larger molecules are embedded into the latent

memory. The model types with attention mechanisms maintain high reconstruction accuracy at longer sequence lengths than the simple recurrent models with the Trans4x-128 architecture maintaining > 99% accuracy on SMILES up to 82 tokens long (~700 Da). This validates our hypothesis that attention will expand the number of potential applications for which these models can be used by increasing the maximum molecule size that can be reliably embedded within the latent memory.

A comparison of the two attention-based architectures (Fig. 3-3b inset) shows that transformers and recurrent attention models perform approximately the same until they approach the data-sparse regime of SMILES longer than ~90 tokens. At this point there is an abrupt drop in performance for the transformer models vs. a gradual decline for the recurrent attention models. The transformer appears to be more sensitive to the choice of model size as increasing the dimensionality of either its attention layers or feedforward layers improves accuracy whereas there is little performance boost when increasing the dimensionality of the recurrent attention model. Even with these improvements, the best performing transformer still exhibits a steeper decline than the worst performing recurrent attention model suggesting that a simpler attention scheme is beneficial to the model's ability to generalize on data that is outside the distribution of the training set.

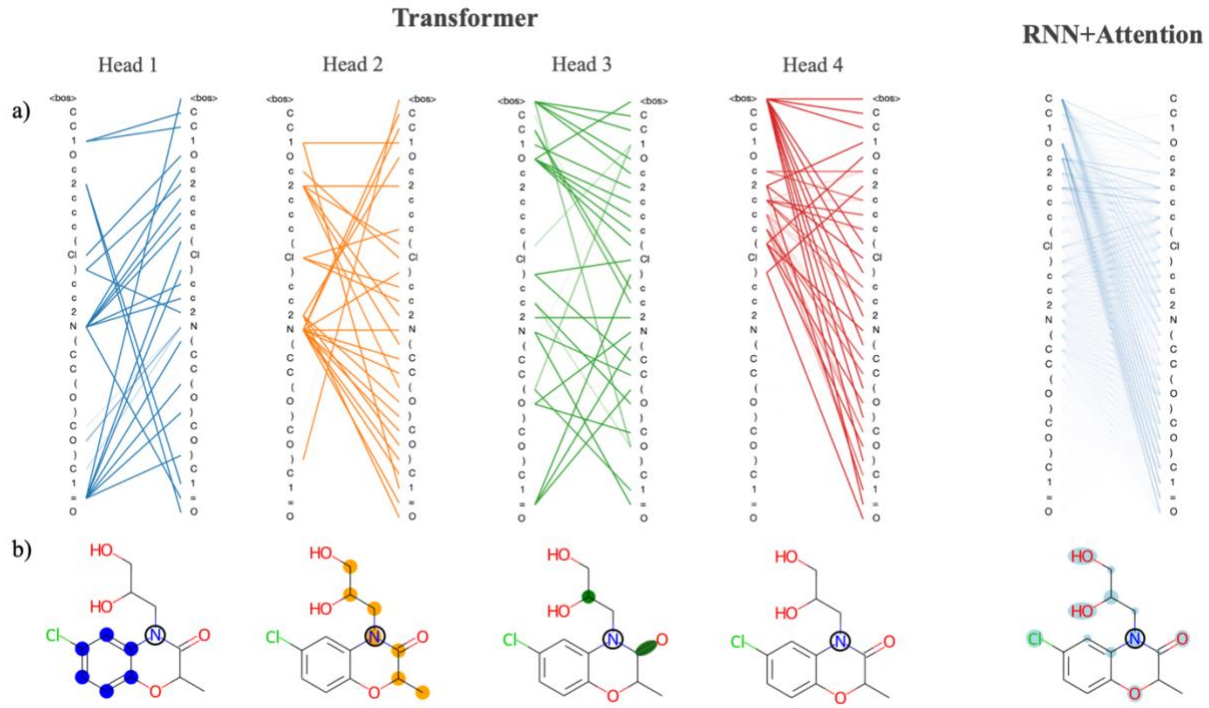


Figure 3-4. Analysis of the attention weights of the Trans4x-256 and RNNAttn-256 models when attending to the molecular structure of diproxadol. The full nxn set of weights are plotted for each attention head within the first layer of the encoder (a). The lines show how each atom/structural feature within the SMILES string is attending to all other features within the same SMILES string (self-attention). The different patterns that emerge from each head represent a unique set of grammatical rules that the model has learned. We also show the attention of a single N atom within diproxadol (b). This molecule was chosen because it is a representative example of the emergent aggregate grammatical trends. From the perspective of the nitrogen, the transformer model has identified the importance of a nearby aromatic ring (head 1), an aliphatic carbon chain of which the nitrogen is a part of (head 2) and a set of structural features including a carbon branch point and nearby double bond (head 3). The attention of the nitrogen in the RNNAttn-256 model is less focused.

There are benefits to the added complexity of the transformer, however. Analysis of the transformer attention weights reveals the model has learned a distinct set of human interpretable structural features that are much more detailed than those learned by the recurrent model with only a single attention head. We use a drug-like molecule from the ZINC dataset, diproxadol, as an illustrative example of the differences between the two (Fig. 3-4). The four transformer attention heads exhibit unique syntactical patterns that demonstrate the model's ability to develop its own "molecular grammar," i.e., rules that define the relationships between atoms and other structural features within a molecule including branches, double bonds, etc. Conversely, the grammar of the recurrent attention model appears to be less well-defined.

The lone nitrogen atom in diproxadol shows us how the heads of the transformer have learned to attend to the immediate molecular environment of a single, centralized atom (Fig. 3-4b). With no supervision, the model extracts its own set of substructures that it has identified as important in relation to the nitrogen atom. Not only does it recognize defining features like the aromatic ring, it can also find non-contiguous features that depend on the structural context around a given atom (see Transformer Head 3 in Fig. 3-4). In this way, the machine-learned substructures are more powerful than graph-based methods that rely on a set of pre-defined substructures because they can extract contextual patterns that are difficult to pre-define but still relevant and interpretable. Others have shown that the transformer is not just restricted to learning intra-molecular features but may also extract an inter-molecular set of grammar rules as well, for instance between products and reactants of organic synthesis reactions.[121]

When analyzing the attention weights across a set of 5000 randomly selected molecules, we find that each attention head corresponds to a different set of higher-level relationships between atomic or structural groups such as aromatic carbons, heteroatoms, branches, and rings. We assess this quantitatively by averaging the attention weights between these groups for each head (Fig. A-8). As an example, the average attention weights between heteroatoms and aromatic carbons are 0.15 and 0.07 for heads 1 and 2. Conversely, the average attention weights between heteroatoms and non-aromatic carbons are ~0.00 and 0.14 for heads 1 and 2, thus the model has partitioned information on the higher-level relationship between heteroatoms and carbon substructures based on their aromaticity. We see this directly reflected in the substructures that were extracted from the diproxadol example and show the learned weights for a variety of structures in Fig. A-9. Attention plays a significant role in the machine-learned "understanding" of molecular structure and as complexity is scaled up, the extracted features become more refined and meaningful. The question then becomes how we can balance the richness of the structural features learned by the transformer with the increased complexity that is required to obtain them.

Information Entropy of Latent Space

The concept of model complexity has been alluded to, previously, as it relates to the model architecture, but we must also define it quantitatively. The most intuitive way to do so is to return to the framework of the information bottleneck. The latent memory provides us a uniform comparison between model types as every molecular embedding within a model's memory is the same size. By evaluating the loss function as written in equation 3.2, we have instructed the model to store as much structurally relevant information within the memory as possible while also minimizing the amount of low information complexity. Therefore, we can use the total information content of the latent memory as a proxy for the complexity of the learned representation as defined by Tishby et al.[129] We calculate

the average Shannon information entropy[136] across all molecular embeddings to compare the information density of latent memories between model types

$$S_j = -\sum_{i=1}^{N} p_i(\mu_j)\log\left(p_i(\mu_j)\right) \quad (3.3)$$

where S is the information density of latent dimension j, and p_i is the probability of finding a given value of μ based on the distribution of latent vectors calculated across all training samples. Note that we use the latent mean vector rather than the reparameterized $\mathbf{z}$ vector because $\mathbf{z}$ is always broadcast to the standard normal distribution even if there is no information stored in a given dimension. We define the total entropy of a model as the sum of S_j across all latent dimensions. This gives us a quantitative metric where a higher entropy indicates a less compressed (and thus more complex) latent representation. Others have drawn similar analogies between Shannon's entropy and system complexity,[137] but to our knowledge this is the first time this metric has been introduced in the context of de novo molecular design.

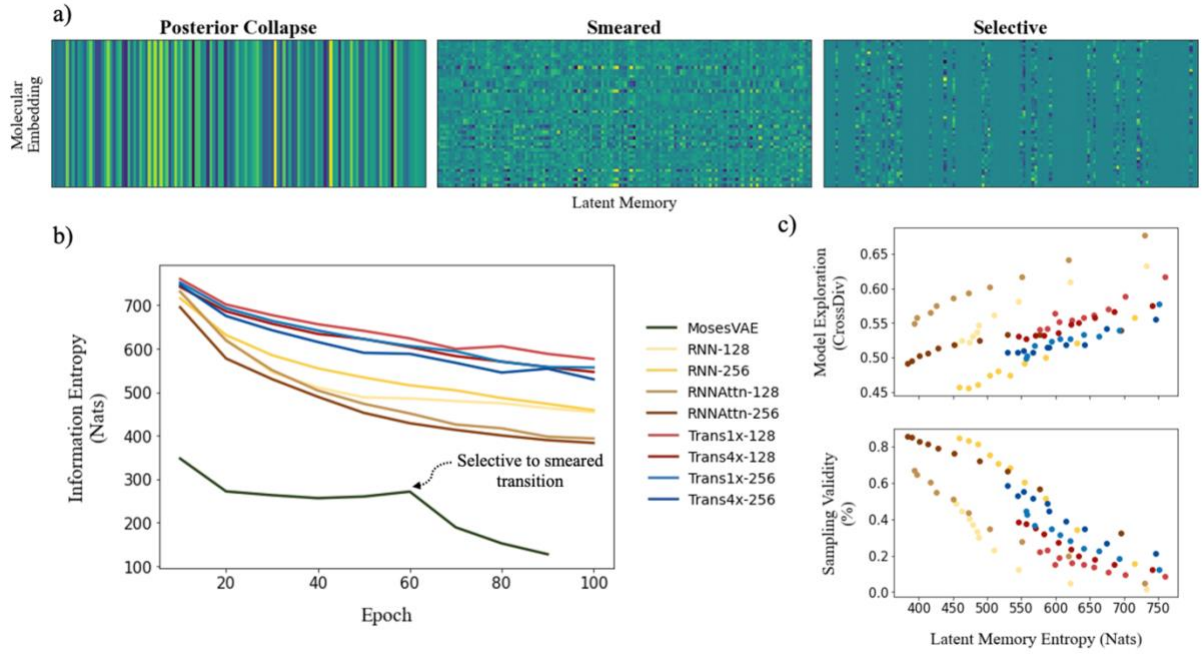


Figure 3-5. Evaluating the effects of model complexity on downstream performance metrics. a) Visualizing a sample of 50 randomly selected molecular embeddings for three commonly observed memory structures (rows are a single molecular embedding and columns are the 128 latent dimensions). The information density (entropy) of each structure increases from left to right. b) Entropy of model memories during training (ZINC). Most models maintain the selective structure throughout training however the MosesVAE model undergoes a transition from selective to smeared at epoch 60. c) Exploration-validity tradeoff as a function of entropy when samples are drawn randomly from all latent dimensions. Cross diversity is evaluated only on valid molecules. The diversity of real molecular structures is shown to increase alongside model complexity as sampling validity decreases.

To illustrate model entropy visually, we show three archetypal memory structures that we have observed in Fig. 3-5a. From left to right the average entropy of these memories increases from 0 nats to 127.4 nats to 393.4 nats respectively. The entropy of posterior collapse is zero because it has learned

the same embedding regardless of the input molecule thus the decoder does not receive new information from the memory. The selective structure is the most commonly observed (Fig. A-10) and occurs when the dimensionality of the true probability manifold is smaller than the number of latent dimensions given to the model.[107] In this case the model learns to ignore superfluous dimensions, assigning them a mean of zero and standard deviation of one to satisfy the KLD loss requirement. We consider the other dimensions meaningful because they contribute to the total information entropy of the memory. The smeared structure is an interesting case in which the burden of information is shared across all dimensions but with each contributing less entropy than the meaningful dimensions from the selective structure. The smeared structure appears as a sudden phase change during training when the number of meaningful dimensions approaches zero (Fig. 3-5b). This effect was only observed for the MosesVAE model.

The progression of entropy during training is shown for each model type. We observe increases in the order MosesVAE < RNNAttn < RNN < Transformer. The high entropy of the transformer models is expected and confirms that the molecular grammar they have learned is both complex and structurally meaningful. It is somewhat unexpected that the RNNAttn models have learned a less complex representation than even the simple recurrent models. Rather than learning grammatical rules, they have learned the most efficient way to distribute information through the bottleneck. The MosesVAE model has the most compressed representation, however it also has the worst reconstruction accuracy which can be attributed to the low information density and the selective to smeared transition at epoch 60. We can now explore the relationship between complexity and the generative capabilities of the models, namely the validity of molecules sampled from the memory and their novelty when compared against the training set.

Strategies for Exploring Chemical Phase Space

A generative model is only as useful as its ability to generate interesting samples. Early molecular design VAEs struggled with generating valid molecules and research has placed a premium on improving the percent validity when a random sampling scheme is employed. However, we believe that exploration is undervalued in the current narrative and that a slightly more error-prone model that prioritizes exploration may actually be more successful at discovering novel functional compounds. Novelty has previously been defined as the percentage of generated samples that are not present in the training set.[119] We introduce another metric, cross diversity, which is defined as follows:

$$CrossDiv(Gen, Train) = 1 - \frac{1}{|Gen|} \sum_{m_{gen} \in Gen} \max_{m_{train} \in Train} J(m_{gen}, m_{train}) \quad (3.4)$$

where Gen and $Train$ are the sample set and training set respectively, m is a molecular fingerprint and $J(m_1, m_2)$ is the Jaccard similarity[138] between two molecules. This metric will be close to zero when all of the generated samples are very similar to molecules from the training set and close to one when they are all far from the training set. Therefore, it can be considered a measure of a model's tendency to explore new regions of phase space.

The structure of a model's memory heavily influences its performance on these metrics. Random sampling favors the lowest entropy memories when the goal is to generate the highest proportion of valid molecules. However, there exists an entropy threshold under which models perform much worse on exploratory metrics (Table 3-2). In fact, although there is some variation between model

architectures, the tradeoff between validity and exploration is generally a function of model entropy that is unavoidable (Fig. 3-5c).

Table 3-2. Comparison of generative metrics for all models with a random sampling scheme. Reconstruction accuracy is calculated based on the models' ability to predict every token within a single SMILES string with 100% accuracy.

Model Type	Entropy (nats)	% Reconstruction Accuracy (ZINC)	% Validity	% Novelty	Cross Diversity
MosesVAE	127.4	0.000	**0.976**	0.696	0.213
RNN-128	453.9	0.996	0.475	0.996	0.516
RNN-256	458.7	0.996	0.846	0.988	0.459
RNNAttn-128	393.4	0.996	0.672	**0.999**	**0.548**
RNNAttn-256	383.2	0.995	0.851	0.995	0.492
Trans1x-128	**576.3**	**0.998**	0.227	0.998	0.538
Trans4x-128	546.4	**0.998**	0.365	0.998	0.530
Trans1x-256	556.6	**0.998**	0.424	0.995	0.502
Trans4x-256	529.5	**0.998**	0.567	0.996	0.503

The difficulty in sampling from high entropy models is a result of the curse of dimensionality[139] that appears within selective memory structures. High entropy dimensions contain all of the meaningful structural information within a model's memory (Fig. 3-6). When the memory is selectively structured, a high entropy means there are a greater number of meaningful dimensions, and it becomes more difficult to avoid leaving "holes" where there is no mapping to a valid structure. This is not a problem for low entropy models as most of the dimensions are either meaningless or contain just a small amount of structural information. While we can easily sample from low entropy models, we miss out on the benefits of an information dense memory which is better at exploring chemical phase space.

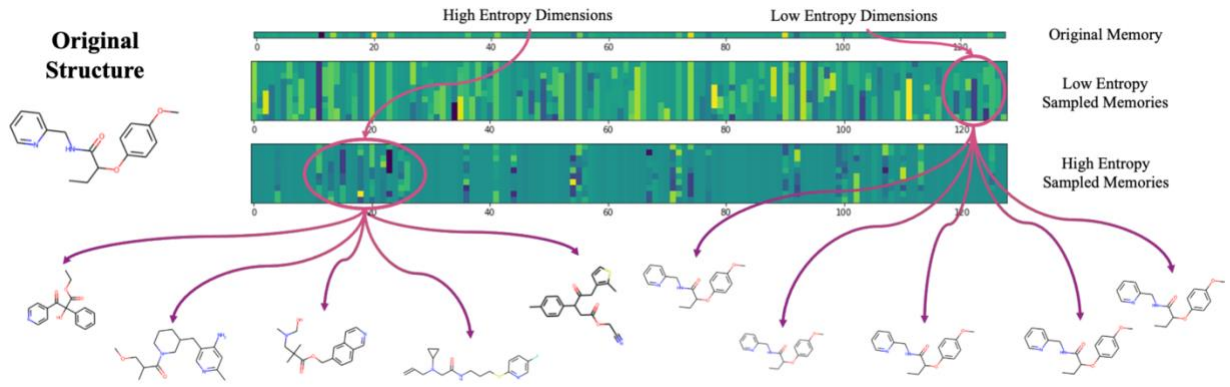


Figure 3-6. The result of exclusively sampling from low entropy dimensions (avg. entropy < 5 nats) vs. high entropy dimensions. Sampling the low entropy dimensions has no effect on the decoded structure confirming that these dimensions are not used by the model. Sampling high entropy dimensions results in a diverse array of structures.

Fig. A-11 shows validity and exploration for five different sampling schemes. By restricting the number of high entropy dimensions that are queried, we avoid the problems inherent to high-dimensional sampling and are able to increase the validity of generated molecules for all model types. This demonstrates the potential of exploiting novel sampling schemes that allow us to maintain the

benefits of a complex, rich latent memory. For instance, we were able to achieve a 32.6% increase in the number of valid molecules generated by the Trans4x-256 model, from 56.7 to 75.2% validity, while only reducing the cross diversity by 15.9%, from 0.503 to 0.423. Moreover, this range is still about two-times higher than the cross diversity of the MosesVAE. We also maintain the allure of the analytical and developmental possibilities that the highly interpretable transformer attention heads afford us by increasing the practical viability of these models in the short-term.

The choice of model type ultimately depends on the individual needs of the researcher, however we can submit a few broad recommendations. Smaller models tend to perform better on exploratory metrics whereas bigger models stick closer to the training set and generate a higher proportion of valid molecules. The addition of attention improves performance in both regards. Therefore, the RNNAttn-128 and RNNAttn-256 models are the most immediately practical. Transformers are the most interpretable and, in our view, have the highest potential for optimization and should be the focus of further development. Additionally, novel input representations such as SELFIES that guarantee 100% sampling validity are a promising alternative to SMILES that may allow us to bypass the complexity vs. validity tradeoff entirely and thus optimize the exploratory capacity of the models directly with sampling schemes that make use of all information-rich latent dimensions.

3.4 Conclusions

We have introduced the concept of attention to the field of molecular design, compared two novel architectures, RNNAttn and TransVAE, to the current state of the art and explored the downstream effect that the structure of the model memory has on a variety of sampling metrics. We find that transformers live up to their reputation based on their ability to learn complex substructural representations of molecular features, and we expect that there is an opportunity to expand our own chemical intuition as we continue to explore the relationships they have learned in more detail. The recurrent attention models, on the other hand, stand out for their superb practical performance exhibiting the best balance between reconstruction accuracy, sampling validity and cross diversity. Despite their promise, there is still much work to be done to improve these models. While the structural features learned by transformers are interesting to analyze, it is not immediately obvious how they might be directly incorporated into future generative algorithms. We also must acknowledge that deep learning-based inverse design remains mainly theoretical, and we will likely need to see many more examples of successful lab-scale design stories before these algorithms see general widespread adoption.

We anticipate there will be two primary directions in which further research may proceed. The first is the direct application of attention based β-VAEs to real-world inverse design problems. There is a growing demand for biodegradable organic alternatives to toxic, high-value commodity chemicals in a number of different industries.[140–142] Many of these involve molecules that are much larger than the average drug-like molecule and we are excited at the prospect of applying attention β-VAEs to these untapped areas. Generative algorithms have the potential to pair nicely with computational reaction networks such as NetGen[143] and we can envision, as an example, a framework in which generated samples are used as the library for a high-throughput search of retrosynthetic pathways for the discovery of bioprivileged molecules.[144]

The second direction is the continued exploration and optimization of attention β-VAE architectures and their hyperparameters, particularly with regards to the formation of the latent memory during

training. There is a definite potential for the implementation of more complex sampling schemes, for instance the two-stage VAE[107] introduces a second model that takes the latent memory as an input and is better able to learn the true probability manifold of the input data. There is evidence that the use of a Gaussian prior restricts the model's ability to directly learn the true probability manifold and so it may be worth exploring alternatives like VampPrior[145] which has already been shown to be able to adequately describe the metastable state dynamics in other physics-based AI models.[146]

Perhaps the most worthwhile pursuit is to continue to develop our knowledge of how the model intuits and compresses structural information, as this could give us insight into novel objective functions that help us encourage the model to better shape its memory and relate it to other pieces of chemical information outside of the current scope. Although the field is advancing rapidly, we are still just at the threshold of the AI-dominated era that Marvin Minsky announced over a half century ago.[147] There may be no aim more practical than furthering our own understanding of the nature of synthetic intelligence to push us further past that threshold. The latent conception of molecular structure is just one component within the broader field of organic chemistry and if coupled with a natural language model-based interpretation of scientific literature, high-throughput classical and quantum calculations, robotics driven lab-scale experimentation and an interactive environment in which our models can communicate and act upon their learning, we may finally begin to approach an intelligence that can solve problems at the pace we introduce them.

3.5 Experimental

Neural Network Hyperparameters

We tested three different model types – RNN, RNNAttn and Trans – for their ability to generate novel molecules. For each model type we also tested multiple architectures as summarized in Table 1-1. The Trans models also include a set of linear layers used to predict the SMILES length directly from the latent memory. This allows us to decode directly from the latent vectors while also masking our source embedding into the decoder and is explained further in Appendix A. The Adam[148] optimizer was used with an initial learning rate of $3e^{-4}$ and an annealer was used to linearly increase β during training. We employed a scaling function that weighed the loss for each token based on its frequency of occurrence. All models were trained for 100 epochs unless stated otherwise.

Neural Network Architecture

As the size of the contextual embedding is significantly larger for the two attention-based architectures vs. the simple recurrent architecture ($n_{seq}xd_{encoder}\ vs.\ d_{encoder}$), we employ a convolutional bottleneck similar to those used in generative image nets[107] rather than a linear bottleneck. More details concerning the convolutional bottleneck can be found in Appendix A.

There are a couple of key differences between the MosesVAE and our own RNN implementation including the size and number of encoder/decoder layers, the use of bidirectionality for the encoder and the absence of batch normalization. For more details on the implementation of the MosesVAE please refer to Fig. A-6/A-7, Table 1-2 and the original paper by Polykovskiy et al.[119] Further details about model construction and training can be found in Appendix A.

Dataset Construction

Two datasets were used to examine how the models perform on different training set distributions. The first is a modified version of the ZINC Clean Leads database[39] with charged atoms removed and a molecular weight range of 250-350 Da. It contains a total of 1,936,963 molecules with an 80/10/10 train/test/dev split. The ZINC data was used to evaluate the models on a traditional AI-driven molecular design task – pharmaceutical discovery. The other is a filtered subset of the PubChem compounds database.[38] It contains molecules with a mean molecular weight of 348 Da, a max of 2693.6 Da and includes some charged compounds with N^+ or O^- containing moieties. Due to the size of the dataset after filtering, a subset of 5,000,000 molecules were randomly selected and used for training with an 80/10/10 train/test/dev split. The PubChem data was used to evaluate the models' performance on reconstructing molecules larger than those typically found in drug-like compound databases. The RDKit[56] Python package was used for downstream analyses of generated molecules including SMILES validity, fingerprints, and physical property calculations.

High Entropy Sampling

When sampling only from high entropy dimensions, we first calculated the entropy of each dimension using equation 3.3. An entropic threshold was selected that determines which dimensions were considered high entropy. This threshold could be calculated analytically, for example using some percentile-based cutoff. We found that in practice a constant threshold of 5 nats / dimension worked well for all model types. Once the meaningful dimensions were selected, we generated molecules by sampling from i) all high entropy dimensions, ii) 5 random high entropy dimensions, iii) 10 random high entropy dimensions and iv) 15 random high entropy dimensions. For k-random high entropy sampling, we randomly picked k dimensions from the N total high entropy dimensions for each new sample. After dimensions were chosen to sample from, new molecules were generated by randomly sampling from the k standard normal distributions corresponding to those dimensions and setting all other dimensions equal to zero.

4. Multimodal Joint Embedding Transformer for Conditional *de novo* Molecular Design and Multi-Property Optimization[2]

4.1 Abstract

Multi-property constrained optimization of molecules using generative de novo design models is vital for the successful application of Artificial Intelligence (AI) towards materials and drug discovery. Yet there remains a gap between the reported performance of such models in the literature and their practical utility in real world design scenarios. Furthermore, existing models are largely inaccessible to chemists without an extensive background in computer science. To address these challenges, we propose a generative foundation model, the **M**ultim**o**da**l** **J**oint **E**mbedding **T**ransformer (MolJET), which performs conditional generation of desired molecular distributions based on human-interpretable chemistry prompts in a zero-shot manner. We assess MolJET on the standard benchmarks available in the GuacaMol and MIMOSA evaluation frameworks. These include structure-based sampling tasks as well as a range of multi-property optimization tasks that probe a models' ability to design drug-like molecules given realistic property constraints. We demonstrate that with self-supervised pretraining, MolJET outperforms 80% of task-optimized models while using zero-shot inferences and beats all baselines after minimal supervision. Moreover, the performance of MolJET on text-only conditioning tasks improves with the inclusion of property modalities during training, highlighting the importance of a multimodal approach to molecular design. MolJET is the first example of text-based de novo molecular design using large-scale multimodal foundation models and should serve as a building block towards further improvements to accessible AI for chemists.

4.2 Introduction

Emerging crises in climate, disease and human health threaten to permanently disrupt global stability and must be actively met with creative solutions. Many such solutions are dependent on the rapid discovery of innovative functional materials or novel drug-like molecules with optimal properties. For instance, the viability of using redox-flow batteries (RFBs) for long-term and large-scale energy storage is contingent on finding stable redox species with fast electrochemical kinetics, a feasible redox potential and high solubility.[149] Due to the immense size and complexity of chemical phase space,[10] the search for suitable materials is far from trivial and traditional "direct" design approaches based on iterative modifications to existing chemical structures are often far too slow.[6]

To address this issue, researchers have increasingly begun to look towards generative de novo design models to efficiently navigate the vast molecular phase space.[150] These models are evaluated on their ability to generate a diverse array of novel molecular structures while simultaneously biasing them towards a desired property distribution.[119] Due to the ubiquity of string-based molecular representations,[77,78] recent innovations in natural language modeling have been successfully applied to de novo molecular design. For instance, transformer architectures have achieved state-of-the-art results on property prediction tasks that require quantum-level accuracy[151] and have also been shown to increase the diversity of candidates sampled from machine-learned molecular distributions.[108]

[2] Reproduced in part with permission from O. Dollar, S. Horawalavithana, S. Vasquez, S. Volkova and J. Pfaendtner. MolJET: Multimodal joint embedding transformer for conditional de novo molecular design and multi-property optimization, *in preparation*, 2023

Aside from string-based representations of molecular structures, there are other textual modalities which could provide additional context to generative models and thus improve their performance. Such modalities include IUPAC names, molecular formulas, descriptions of important chemical moieties or functional groups and natural language descriptions of chemical behavior. Yet despite the large overlap between architectures used for natural language modeling and molecular sequence modeling, there have only been a few attempts to incorporate more than a single modality within a model[152–154] and none have included the capacity for property-driven molecular design. Massive scaling has also been primarily limited to property prediction tasks[155,156] despite growing evidence of the performance benefits derived from increasing model sizes, dataset sizes and compute across all downstream tasks.[157,158]

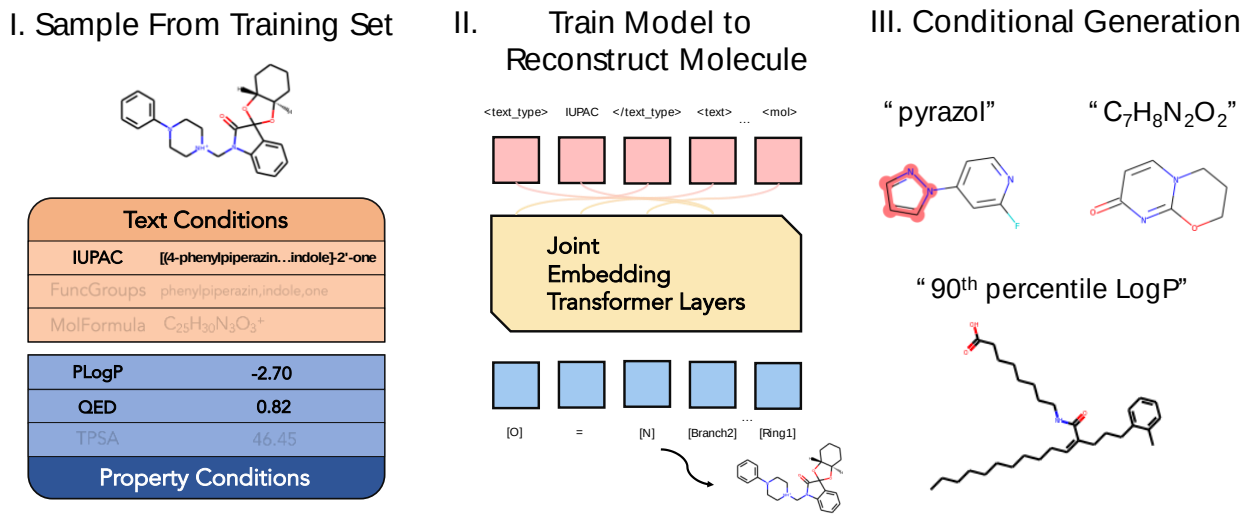


Figure 4-1. MolJET Framework. Prompts are (i) stochastically sampled from the available modalities in the dataset and (ii) used to condition autoregressive reconstruction of SELFIES strings. Conditions are then chosen during inference to (iii) shift the generated molecular distribution towards the desired structural or physicochemical properties.

In this work we introduce MolJET, a large-scale multimodal joint embedding transformer for conditional molecular generation and multi-property optimization. Within this framework, molecular generation is conditioned by text-based prompts that control the structural and physicochemical characteristics of the desired molecular distributions as depicted in Fig. 4-1. We demonstrate conditional generation on three modalities - textual descriptions of molecular structural features, physicochemical properties and 1D atomistic molecular graphs - and provide a general framework for the inclusion of additional modalities during pretraining.

To prove the efficacy of our models in realistic design scenarios, we evaluate MolJET on a diverse set of tasks including molecular rediscovery, similarity and substructure-based sampling, isomer generation, and multi-property optimization.[87,159] With only self-supervised pretraining, MolJET outperforms all task-optimized baseline models on five out of the eight task categories and outperforms the baselines on all eight task categories after minimal task-specific supervised optimization. Furthermore, the prompts are designed to be easily interpretable by chemists without any prior knowledge of deep learning and thus accessible to a wider audience.

4.3 Related Work

Multi-Property Optimization

Several strategies for multi-property optimization of molecular structures have been explored to date. Some works propose to condition the generation of molecular structures with a learnable embedding corresponding to the values of one or more desired properties.[104,111,160] These models jointly learn the conditional distributions during training and then allow for the selection of specific conditions during inference. Others treat optimization as a translation task, in which an improved version of the input molecule is reconstructed during training.[114,161] These models learn the desired molecular distribution directly, however they also require the construction of translation pairs which can be time-consuming and without careful control can introduce biases into the model or result in posterior collapse.[162] Another popular strategy for optimization is by making stepwise modifications to an existing molecular structure through an efficient sampling method like Markov Chain Monte Carlo or a reinforcement-learning driven policy network.[87,163,164] A reward function determines the success of the model and guides further modifications. These models are flexible as they can modify their actions based on any reward, however they often shift the generated distribution too far from the original and can struggle to generate realistic samples.[159,165]

Foundation Models for Chemistry

Given that the vast majority of de novo molecular design models operate on a single molecular representation, there are only a few examples of multimodal learning in the field of chemistry. KV-PLM and CHEMET both combine structural representations of molecules with natural language, the former by embedding SMILES strings directly into a biomedical corpus and the latter by performing cross-modal attention between embeddings of a molecular graph and a description of the molecule.[152,154] However, these models are better suited for classification tasks than generation tasks as it is challenging to build a corpus annotated with molecular structures that is large enough to train a generative model. Other examples of multimodal chemistry models include GeomGCL[166] which performs contrastive learning on 2D and 3D molecular graphs for property prediction and VJTNN[162] which combines junction tree and atomic graph representations during the encoding and decoding of the latent vector in a VAE.

4.4 Model Framework and Prompt Designing

Herein, we describe the **M**ultim**o**da**l** **J**oint **E**mbedding **T**ransformer (MolJET), a large-scale generative foundation model for conditional molecular design and multi-property optimization. The aim of MolJET is to efficiently navigate the molecular phase space while simultaneously reaching a desired property distribution. This task is non-trivial as the molecular landscape is high dimensional and rugged making optimization within this space difficult.[167] We hypothesize that jointly learning across text, molecular structure and properties will enhance the model's ability to learn structure-property relationships and thus improve its performance at designing optimized molecules. We first introduce the multimodal fusion with our prompt design framework, and then present the model architecture and conditional sampling scheme.

Multimodal Fusion with Prompt Designing

Our goal is to learn inter-modal and cross-modal information with an expressive prompt design that can facilitate both the self-supervised pretraining and zero-shot evaluation. We propose an early-fusion strategy to jointly reason over the text, molecular structure, and property modalities with a shared multifaceted representation. We represent the textual description and associated physicochemical properties of a molecule in the prompt sequence $x = (s_1, s_2, \dots, s_n)$ of the form $(s_{text}, s_{prop}, s_{mol})$,

<text_type>…</text_type> <text>..</text> <property>..</property> <val>..</val> <mol>..</mol>

We include <text_type> and <property> tags to differentiate across molecule descriptions (s_{text}) and properties (s_{prop}). The <text> and <val> tags designate the search space on the respective data modalities. The <mol> tag designates the SELFIES string describing the molecular structure (s_{mol}). The proposed prompt design is flexible so that other textual representations of molecules or associated properties may be easily substituted. We also allow each modality to contain multiple sub-prompts. For example, we can represent multiple physicochemical properties separately as sub-prompts in s_{prop}. We introduce a strict ordering of the prompt sequence with the corresponding text, property and molecular structure representations to enable the model to conditionally generate molecular distributions given the other modalities.

Model Architecture

Our objective is to pretrain a large-scale foundation model with the ability to generalize to unseen tasks without requiring any labeled data. This is especially relevant in molecular design scenarios where we need to generate new molecules that have not been previously seen (out-of-distribution generalization). However, it is intractable to enumerate across all possibilities due to the unbounded molecular search space. We present the unsupervised distribution estimation $p(x)$ from a set of prompts $(x_1, x_2, \dots, x_n)$ as the product of conditional multimodal token probabilities,

$$p(x) = \prod_{i=1}^{n} p(s_n | s_1, \dots, s_{n-1}) \quad (4.1)$$

Our model design is inspired by the recent success of applying the transformer encoder architecture on shared multimodal multifaceted representations (e.g., UTF-8 bytes in Perceiver-IO[168], vision-language decoding[169]). In this work, we investigate whether transformer architectures are capable of learning over multimodal molecular information and translating it into a rich knowledge of the relationship between a molecule's structure and its properties. We seek to analyze whether transformer architectures are suitable to distill and accumulate both inter- and cross-modal information from the molecular descriptions, and test whether the pretrained models generalize to novel contexts during de novo molecular design.

To this end, we adopt the autoregressive transformer decoder model architecture similar to GPT-3[123] and apply it on conditional multimodal prompt based molecule generation tasks. We translate the

general left-to-right language modeling objective to a joint modeling objective that predicts the next modality token. We minimize the joint loss defined as

$$\mathcal{L}(\theta) = \frac{1}{|D^{train}|} \sum_{x \in D^{train}} -log p_\theta(s_i | s_{\leq i}) \quad (4.2)$$

The model learns the conditional multimodal token distribution jointly given the in-context references to other modality tokens. We do not use modality-specific encoders in this setup since we translate all modalities into the discrete language space. It remains as a future work to explore how other modalities such as vision (continuous), graph (2D) or atomic coordinates (3D) could be used in our framework to further enrich the learned multimodal molecular representations.

Conditional Molecule Generation

Given the molecular structure represented as a sequence of tokens describing the atoms, their connectivity and their valence states ($m_1,..m_n)$, the conditional multimodal prompt-based molecule generation is as follows:

$$\hat{m} \approx \underset{m}{\text{argmax}}\, log p_\theta(m_t | s_{text}, s_{prop}, m_{<t}) \quad (4.3)$$

We use q temperature sampling to autoregressively sample the SELFIES tokens m_t conditioned on the multimodal prompt. The sampling takes the molecule textual description s_{text}, physicochemical properties s_{prop} and <mol> $\in m_{<t} \subset s_{mol}$ as the initial inputs in the joint multimodal embedding space. In addition, the molecule generation is conditional to the property values in s_{prop}.

$$m_t = q(\cdot | s_{text}, s_{prop}, m_{<t}) \quad (4.4)$$

$$s_{mol(t)} = \cup_{m_{<t} \in s_{mol(t-1)}} \{(m_{\leq t} \circ m_t^n | m_t^n)\}_{n=1}^{N} \quad (4.5)$$

We sample N molecule tokens until we reach a </mol> tag. The sampled tokens are concatenated $\circ$ with other top scoring molecule tokens to generate the molecule structure $s_{mol(t)}$.

4.5 Experimental Setup

Implementation and Training Details

Dataset Creation. We gathered over 100M unique molecular structures from the PubChem compound records database[38] to use for pretraining. Each structure includes a valid SMILES representation, an IUPAC[3] name, and a molecular formula. Functional groups are extracted from the full IUPAC name and SMILES are encoded as SELFIES strings. In accordance with the method outlined in GuacaMol,[159] we calculate the ECFP4 fingerprints[170] for every molecule in our dataset and a holdout set of drug-like molecules used in the benchmarks. Any molecule in the training set with a tanimoto fingerprint similarity of ≥ 0.343 to any molecule in the holdout set is removed. This ensures

[3] IUPAC (International Union of Pure and Applied Chemistry) nomenclature provides an international standard of naming compounds which can be used to create unambiguous structural formula.

the model has not simply memorized solutions to the benchmark tasks during pretraining. Similarly, all isomers corresponding to the two isomer generation tasks were also removed from the training set.

Conditional prompts for each molecule are generated stochastically so the model may only see a portion of the available modalities for any given sample. This allows the user to ignore some modalities during inference while still allowing the model to jointly learn over all possible modalities. The rules for prompt sampling are outlined in Appendix B.

Available Modalities. We provide three modalities on which the models are conditioned - textual molecule descriptions, properties and 1D atomistic molecular graphs. Table 4-1 shows the sub-modalities available for the text and property modality types. Each text type provides a different level of detail regarding the molecular structure and are all commonly used by chemists when describing molecules. The properties are selected to cover a wide range of chemical behavior important to drug design. Each property is calculated using the cheminformatics package RDKit[56] aside from DRD_2 which is predicted by the model published in Olivecrona et al., (2017).[171] We use SELFIES as our 1D atomistic molecular graph to guarantee the validity of all molecules generated during inference.[78]

Table 4-1. Details of the multimodal inputs used in the pretraining and zero-shot evaluation.

Textual Molecule Descriptions	**IUPAC**, text that fully specifies the atomic connectivity of the entire molecule	**FuncGroups**, text that specifies only the atomic connectivity of local environments within the molecule	**MolFormula**, text that does not specify any connectivity information but does specify the overall atomic makeup of the molecule
Physicochemical properties	**Topological polar surface area (TPSA)**, a measure of the overall surface polarity of the molecule[172]	**LogP/Penalized LogP (PLogP)**, a method for estimating the solubility of a molecule.[173] PLogP includes penalties for molecules with low synthesizability	**BertzCT**, a topological index meant to quantify the "complexity" of a molecule[174]
	QED, a quantitative measure of the "drug-likeness" of a molecule[175]	**Number of fluorine atoms**, **Number of aromatic rings**, **Total number of rings**	**DRD_2**, the biological activity of a molecule towards the dopamine receptor D_2

Tokenization. We develop a custom vocabulary that consists of the tokens representing the molecule textual description s_{text}, physicochemical properties s_{prop}, and molecular structure s_{mol}. IUPAC and FuncGroups share a vocabulary learned from a byte-pair encoding of the IUPAC names in the training set. The MolFormulas and SELFIES are tokenized on a per-atom basis. Property values are represented as either scalars or decile ranges labeled 1-10 with each digit tokenized separately. Finally, all tags (<...>,<.../>) and property names are encoded as special tokens.

Task Descriptions

We evaluate MolJET on 22 tasks split across 8 different categories: molecular rediscovery, similarity sampling, substructure sampling, isomer generation, median molecules, multi-property optimization, drug-likeness and biological activity. Each task is taken from either the GuacaMol evaluation framework[159] or the MIMOSA multi-property optimization framework.[87] Table 4-2 provides examples

of tasks from a few of the optimization categories and their corresponding prompts. Detailed descriptions of each task category are provided below.

Table 4-2. Example of the downstream tasks and prompt designs used in the zero-shot evaluation. We color each prompt with the modality(s) that they are associated with. For the prompts for all 22 tasks, please refer to Tables B-6 and B-7.

Task/Example	Prompt
Molecular Rediscovery	<text_type>IUPAC</text_type>
Celecoxib	<text>4-[5-(4-methylphenyl)..benzenesulfonamide</text><mol>
Similarity Sampling	<text_type>FuncGroups</text_type>
Albuterol	<text>butylamino,hydroxyethyl,phenol</text><mol>
Isomer Generation	<text_type>MolFormula</text_type>
$C_{11}H_{24}$	<text>C11H24</text><mol>
Multi-Property Optimization	<text_type>IUPAC</text_type>
	<text>N-[2-[2-(dimethylamino)..prop-2-enamide</text>
Osimertinib	<property>tpsa</property><val>146.0</val>
	<property>logp</property><val>-0.5</val><mol>

Molecular Rediscovery. The model must generate an exact match to the target. This task tests the model's ability to explore regions of molecular phase space which it has not encountered during training.

Similarity Sampling. The model must generate many samples that are structurally similar to the target but not an exact match. This task tests the model's ability to make small structural modifications to a target without diverting too far from the original molecule. This is analogous to how a chemist might approach the design of a new drug by modifying small chemical motifs of a starting structure to improve a specific desired behavior while maintaining other drug-like qualities from the original molecule.

Substructure Sampling. The model must generate many samples that contain a specific structural motif or set of motifs. In some tasks, the model may also be penalized for generating molecules with non-desired motifs or for diverging too far from the pharmacological properties of the molecule from which the desired motif is drawn. This task tests the model's ability to generate functional moieties off a scaffold or "fill in" the scaffold given a set of functional moieties.

Isomer Generation. The model must generate as many structural isomers as it can from a given molecular formula. This task tests the model's ability to map coarse-grained chemical information to a fully connected atomic graph. It also tests if the model can enumerate all possible structures from a local region of chemical phase space.

Median Molecules. The model must generate samples that are maximally similar to two different target molecules. This task tests the model's ability to interpolate between two valid chemical

structures, a common goal when trying to discover a molecule that maximizes the desired properties of two separate existing molecules.

Multi-Property Optimization (MPO). The model must simultaneously match both structural and property requirements as dictated by the task. For instance, the model might be tasked with finding a structural analogue to the antihistamine fexofenadine that is "less greasy" by reducing the LogP and increasing the TPSA while maintaining a high structural similarity to the target. These tasks put the model in realistic drug design scenarios and demonstrate its ability to perform structural sampling while also constraining the generated molecules to the desired property ranges.

To demonstrate the versatility of the MolJET framework, we also evaluate the model on the multi-property optimization tasks outlined in Fu et al., (2021). These require the model to maintain high structural similarity to an input drug-like molecule while simultaneously maximizing PLogP and either QED (**Drug-Likeness**) or DRD_2 (**Biological Activity**). We report performance on these two tasks as success rate which is defined as the proportion of input molecules that the model is able to improve beyond a pre-defined threshold for each property while maintaining high similarity. Further details on the definition of success rate are provided in Jin et al., (2019). Each GuacaMol task is evaluated based on a weighted average of the top 100 scoring molecules for that task. Further details on the definitions of each GuacaMol metric are provided in Brown et al., (2019) and Appendix B.

Conditional Language Model Pretraining. We train two independent versions of MolJET, MolJET-Guac and MolJET-Bio. MolJET-Guac is trained and evaluated with the three text types and TPSA, LogP, BertzCT, number of fluorine atoms and ring counts (total and aromatic). MolJET-Bio is trained and evaluated with the three text types and PLogP, QED and DRD_2. We train two additional model variants - one to study the difference between scalar and decile property value representations ($\text{MolJET-Guac}_{\text{Scalar/Decile}}$) and one without property conditioning to study the cumulative effect that additional modalities have on text-only inference tasks ($\text{MolJET-Guac}_{\text{Text-Only/Text+Prop}}$). The models are pretrained from scratch on the filtered PubChem training set. Further details on the training procedure, hyperparameters, baseline models and sampling scheme can be found in Appendix B.

4.6 Experimental Results

The performances of MolJET-Bio and MolJET-Guac on the MIMOSA and GuacaMol evaluation frameworks are displayed in Tables 4-3 and 4-4. Both models are very competitive during zero-shot inference with MolJET-Guac outperforming ~78% of all baselines on the GuacaMol benchmarks and MolJET-Bio improving the success rate on the Drug-Likeness and Biological Activity tasks by 18.75% and 13.5% respectively. It should be noted that the baselines are fine-tuned on each task in a supervised manner, whereas MolJET has only undergone self-supervised pretraining and is seeing the task-specific optimization prompts for the first time during inference. Thus, the performance on these benchmarks demonstrates the efficacy of our multimodal framework in generalizing to previously unseen molecular distributions.

Multi-Property Optimization

We first show that MolJET is able to leverage information from multiple modalities to simultaneously control the structure and properties of generated molecules during zero-shot inference. By conditioning the model on the modalities that are optimal for a given task, it can generate molecular

distributions that outperform previously state-of-the-art baselines on a variety of multi-property optimization benchmarks. It accomplishes this by inferring how the desired structural features must be modified to satisfy the additional property constraints. We use the conditional generation sampling method to efficiently explore the local region of molecular phase space dictated by the multimodal prompt.

Table 4-3. Benchmark results on the MIMOSA MPO evaluation framework. PLogP, QED and DRD_2 columns refer to the absolute improvement in property values from successful samples.

	Drug-Likeness				**Biological Activity**			
Method	Similarity	PLogP	QED	Success	Similarity	PLogP	DRD_2	Success
VJTNN	0.17	0.46	0.02	1.0%	0.18	0.55	0.27	3.4%
DeepGA	0.35	0.93	0.09	24.9%	0.38	0.68	0.20	29.3%
MIMOSA	**0.42**	0.93	0.10	32.0%	**0.54**	0.75	0.35	43.7%
MolJET-Bio (Zero-shot)	0.37	**1.19**	**0.14**	**38.0%**	0.35	**3.38**	**0.48**	**49.6%**

For example, MolJET-Bio outperforms the previous state-of-the-art, MIMOSA, in both absolute property improvement and success rate on the Drug-Likeness and Biological Activity MPO tasks. It does so by exploring the local region of molecular phase space surrounding the target molecule more efficiently by directly sampling from the conditional distribution. Because MIMOSA makes iterative modifications to the target molecule, it does not venture as far from the original structure during optimization. While this leads to a higher similarity score on both tasks, it fails to find as many molecules that satisfy the property optimization constraints and thus has a lower success rate.

Table 4-4. Benchmark results on GuacaMol which contains both MPO and molecular structure generation tasks. Bold values indicate the best performing model and underlined values indicate the second best performing model measured against the baselines.

Benchmark Category	Best of Data Set	SMILES LSTM	SMILES GA	Graph GA	MolJET-Guac (Zero-shot)	MolJET-Guac + Graph GA
MPOs	0.698	0.778	0.717	0.868	<u>0.838</u>	**0.878**
Rediscovery	0.613	**1.000**	0.523	0.945	**1.000**	**1.000**
Similarity	0.546	**1.000**	0.771	0.977	**1.000**	**1.000**
Substructure	0.643	<u>0.973</u>	0.769	**0.985**	0.817	**0.985**
Isomers	0.716	0.912	0.745	<u>0.954</u>	**1.000**	**1.000**
Median	0.371	0.403	0.362	0.417	<u>0.409</u>	**0.447**
Total	0.623	0.850	0.671	0.877	<u>0.857</u>	**0.900**

We observe a similar trend from zero-shot MolJET-Guac on the GuacaMol MPOs. When breaking the tasks down individually, it outperforms all three baselines on the ranolazine, perindopril, and amlodipine MPOs and is within 1% and 2.5% of the best performing model on the fexofenadine and osimertinib MPOs, respectively (Appendix B). These tasks also require the model to meet one or more property specifications while maintaining high similarity to a target molecule (see Fexofenadine and Perindopril MPOs, Fig. 4-2). In total, MolJET outperforms or is competitive with the leading baseline on seven out of nine MPOs across both evaluation frameworks demonstrating the versatility and efficacy of our multimodal framework.

Conditional Molecular Structure Generation

MolJET-Guac also performs well at the zero-shot molecular structure generation tasks, achieving a perfect score on rediscovery, similarity sampling and isomer generation (Table 4-4). This indicates that the model is able to accurately estimate the molecular structural probability manifold of the training set and navigate it based on the conditional multimodal prompts. Each of the three text modalities provide a different degree of structural specificity with which the model can be conditioned. For instance, tasks with stringent similarity requirements are better suited for IUPAC conditioning, whereas FuncGroup conditioning yields a more diverse set of generated molecules (see Drug-Likeness vs. Fexofenadine MPO in Fig. 4-2). FuncGroup conditioning is also the most flexible as it can be used to combine the structural characteristics of multiple input molecules (see Median Molecules, Fig. 4-2).

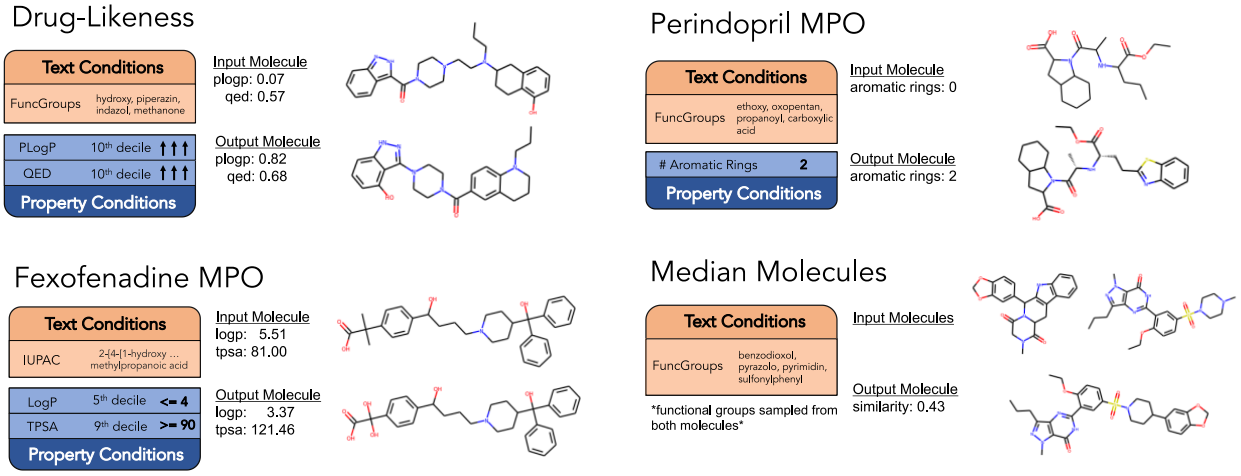


Figure 4-2. Prompts, inputs, and high-scoring samples for four of the de novo design tasks.

We confirm these observations quantitatively by measuring the performance of each text modality individually on the similarity sampling tasks. We choose similarity as it is the most common structural objective for the MPOs and thus highlights important differences in sampling performance for realistic drug design scenarios. The results of this experiment are shown in Fig. 4-3. As expected, we explore the largest subset of relevant phase space when conditioning on FuncGroups. However, there are some circumstances where IUPAC conditioning is just as effective, namely when the molecule is complex such as the stereoisomer mestranol.

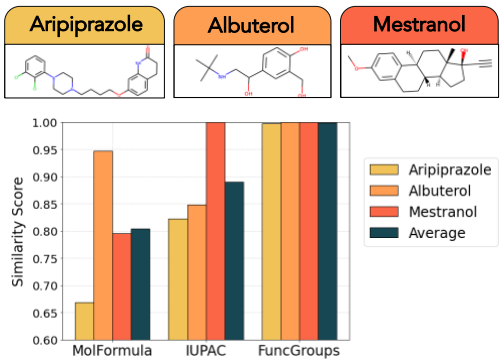


Figure 4-3. Similarity sampling from each text modality.

To estimate how amenable MolJET is to further optimization, we re-run the Graph GA method but replace the starting population with the top 100 molecules generated by MolJET. On average, the Graph GA seeded with molecules generated by MolJET improves upon the zero-shot MolJET by ~5% and the baseline Graph GA by ~2.6% (Table 4-4). This demonstrates the capacity of MolJET to be further improved by task-specific fine-tuning strategies and we leave further work in this direction as future research.

Evaluating Prompt Design

We also run ablations to study a) the effect of the choice of numerical property representation on the GuacaMol tasks with property conditioning and b) the impact of the inclusion of property modalities during training on GuacaMol tasks with text-only conditioning. On the GuacaMol tasks with property conditioning, MolJET-Guac$_{\text{Scalar}}$ performs slightly better than MolJET-Guac$_{\text{Decile}}$ (0.881 vs 0.872). This suggests that the property prediction capacity of the scalar model is only slightly greater than the average distance between decile bins. For most properties, this distance is fairly large so this result indicates a potential area in which MolJET could be improved.

Finally, we evaluate MolJET-Guac$_{\text{Text-Only}}$ and MolJET-Guac$_{\text{Text+Prop}}$ on the text-only inference tasks from GuacaMol (Table 4-5). These tasks do not require any property conditioning during inference and thus the performance of the two models should be expected to be comparable if cross-modal learning does not occur during training. However, we find that MolJET-Guac$_{\text{Text+Prop}}$ performs better on the text-only inference tasks, supporting our hypothesis that our multimodal prompt design framework supports both inter- and cross-modal learning. The property information that is jointly embedded during training enhances the models understanding of molecular structure even when that information is not provided during inference.

Table 4-5. Multimodal model ablations.

Modality	GuacaMol	Reconstruction	
		IUPAC	FuncGroup
Text	0.827	62.1%	60.2%
Text + Property	**0.843**	**68.7%**	**63.4%**

To confirm this behavior, we construct two additional text-only inference tasks, **IUPAC Reconstruction** and **FuncGroup Reconstruction**. IUPAC Reconstruction tests the models' ability to accurately reconstruct a SELFIES string given its IUPAC from a holdout set of IUPAC-SELFIES pairs that were not seen during training. FuncGroup Reconstruction tests the models' ability to generate molecules that contain the requested functional group from a list of 102 functional groups developed by the authors to include a wide range of atom types and complexities. Additional implementation details for each task are outlined in Appendix B. Again, we find that MolJET-Guac$_{\text{Text+Prop}}$ outperforms MolJET-Guac$_{\text{Text-Only}}$, providing additional evidence that both inter- and cross-modal learning occur during training and that multimodal joint embeddings are capable of enhancing the performance of de novo molecular design models.

4.7 Conclusions

We introduce MolJET, a multimodal foundational chemistry model for conditional de novo design of organic molecules. MolJET demonstrates state-of-the-art performance on realistic drug design tasks in a zero-shot manner. Our framework is adaptable and easy to interpret, making it well-suited for the inclusion of other modalities such as scientific text. We make our code, models, and data publicly available and provide API access to our pretrained models to allow chemistry researchers of all backgrounds to participate in the future development of AI-driven de novo molecular design.

5. Efficient 3D Molecular Design with an E(3) Invariant Transformer VAE[4]

5.1 Abstract

This work introduces a 3D equivariant graph-to-string transformer VAE (Vagrant) for generating molecules with accurate DFT-level properties. Vagrant learns to model the joint probability distribution of 3D molecular structure and property by encoding molecular structures into a 3D-aware latent space. Directed navigation through this latent space implicitly optimizes the 3D structure of a molecule, and the latent embedding can be used to condition a generative transformer to predict the candidate structure as a 1D sequence. Additionally, we introduce two novel sampling methods that exploit the latent characteristics of a VAE to improve performance. We show that our method outperforms comparable 3D autoregressive and diffusion methods for predicting quantum chemical property values of novel molecules in terms of both sample quality and computational efficiency.

5.2 Introduction

Generative models for de novo molecular design provide an efficient solution to the inverse design problem[6] by facilitating the rapid exploration of chemical phase space.[176] They have been successfully applied to many real-world design tasks including the discovery of novel drug inhibitors,[177] protein therapeutics,[28] and metal-organic frameworks.[178] Despite these success stories, continued advancements in generative architectures and molecular representations suggest there is still room for further improvement.

For instance, the development of equivariant neural networks[179] allows for the direct optimization of molecular atomic coordinates in 3D. The 3D structure of a molecule dictates much of its functional behavior including properties governed by its electronic structure[180] and geometric constraints such as the affect of steric hindrance on its binding affinity to a protein target.[181] Thus a model with the capacity to embed 3D information will be better able to learn the relationship between a molecule's structure and its properties than models which are restricted to learning on 1D/2D representations only. This has been demonstrated on a variety of tasks including DFT-level property prediction[23] and shape-based ligand design.[105]

Although many studies have proposed de novo generation in 3D,[81,160,182] the choice to generate 3D coordinates adds considerable complexity to the decoding process and model architectures designed for this purpose have some notable drawbacks. Flow models and autoregressive models are difficult to scale due to their long training times and slow sampling.[83] Diffusion models are also slow during inference and have low log-likelihoods compared to other comparable generative methods.[183–185] These problems are compounded by the general inefficiency and instability of 3D generation.[84]

Conversely, one-dimensional sequence generation has been shown to be both efficient[94] and capable of producing high-quality novel samples during inference.[123] Many de novo molecular design models generate sequence-based representations of molecules and it has been shown that such models are

[4] Reproduced in part with permission from O. Dollar, N. Joshi, D. Beck, and J. Pfaendtner. Efficient 3D Molecular Design with an E(3) Invariant Transformer VAE, *in preparation*, 2023

better at capturing molecular property distributions than models trained on 2D graphs.[85] Given these observations, we hypothesize that a model which:

- learns embeddings of 3D molecular structure, and
- optimizes those embeddings to condition the generation of 1D sequences with desired properties

will be more efficient and propose higher quality candidates than models which generate 3D coordinates directly.

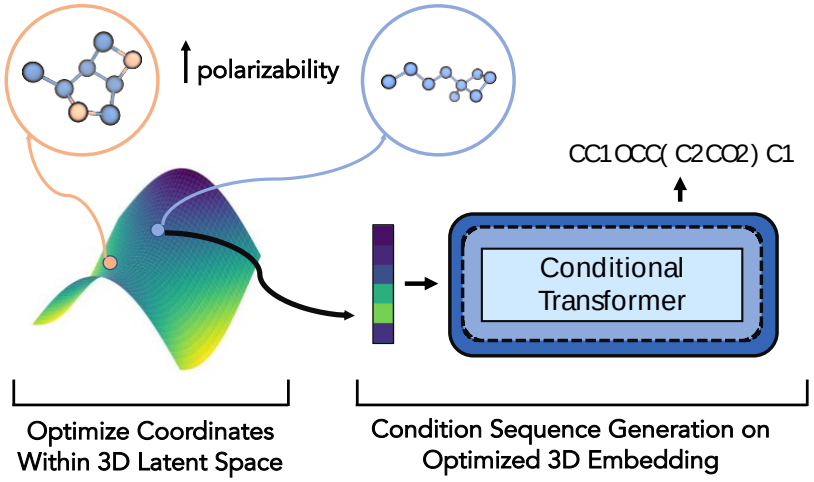


Figure 5-1. Vagrant learns a 3D-aware latent space that contains information about the relationship between a molecule's 3D coordinates and its properties. Navigation within this space implicitly adjusts the molecular coordinates which can then be easily decoded into a 1D sequence.

To explore this idea, we propose Vagrant, a **V**ariational **A**utoencoding **Gra**ph-to-stri**n**g **T**ransformer model. Our method maps a set of 3D molecular structures to an optimizable 3D-aware latent space that is used to condition the generation of SELFIES strings.[78] By separating the networks which act on 3D and 1D representations across the bottleneck of a VAE, we decouple the generative process from the model's ability to optimize 3D geometries (Fig. 5-1). We show that the predicted DFT-level property values of the novel candidates designed by our method are more accurate than candidates designed by comparable 3D autoregressive[160] and diffusion models.[182] We also introduce a novel sampling scheme and confidence metric that allow us to further improve performance without any retraining and at little additional computational overhead.

5.3 Related Works

1D/3D Molecular Representations

One-dimensional sequence representations of molecular structure are lightweight and capture many of the salient structural features of molecules such as branches, rings, and bond types. The SMILES string[77] has previously been the most ubiquitous 1D molecular representation. However, the syntax of SMILES requires models to recognize long-range dependencies and failure to do so often results in invalid molecules.[186] To mitigate this problem, the SELFIES string[78] was introduced which guarantees the syntactic validity of any combination of tokens within the SELFIES vocabulary.

Three-dimensional molecular representations, while more memory-intensive than 1D sequences, can capture the spatial relationship between atoms as well as higher-order node- and edge-level features.[22] Recent work in this direction has focused on guaranteeing that they maintain permutational, translational, and rotational equivariance. Many studies have used the spherical harmonics to accomplish this, by computing a basis that transforms molecular coordinates to a higher-order space that preserves SE(3) equivariance.[21,95,96] However, the computation of the basis set is expensive and cannot be generalized beyond 3D. Representations that preserve E(n) equivariance without relying on spherical harmonics have been shown to be more computationally efficient while maintaining comparable accuracy.[81]

3D Molecule Generation

Many architectures have recently been proposed for generating 3D molecular structures. 3D autoregressive models that sequentially place atoms and bonds in 3D space have been a popular method for shape-based ligand design[51,187,188] and have also been used to simultaneously optimize multiple structural and quantum chemical properties.[160] Diffusion models, which sample entire molecular structures by iteratively removing 3D gaussian noise, have been applied to both these tasks as well.[182,189] Due to the difficulty of reconstructing 3D coordinates from a compressed latent space, VAEs have been mostly restricted to 1D/2D generation.[93,190] However, Huang et al., (2022)[191] showed they could be enhanced with E(3) equivariant layers and used to autoregressively generate the linker between two molecular fragments for PROTAC drug design.

Molecular Translation Models

The translation of one molecular representation to another has been studied for learning robust molecular embeddings,[92] and for quickly translating between disparate representations such as IUPAC strings and SMILES.[192] Particularly relevant to this work is the distillation of representations from high dimension to low. Several studies have used this method of distillation, for instance to caption 3D binding pockets with SMILES strings[193,194] or to autoregressively generate 2D graphs to connect fragments based on their 3D distances and relative orientations.[195]

5.4 Background

Variational Autoencoder

Vagrant uses the VAE framework described by Kingma et al., (2013)[2] to learn a 3D-aware molecular embedding that is both optimizable and easy to sample. A VAE encodes the latent probability manifold, $\mathbf{z}$, of the training set, $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$, by approximating the true posterior of the data, $p(\boldsymbol{z}|\boldsymbol{x})$, and the marginal likelihood, $p(\boldsymbol{x}|\boldsymbol{z})$, given a point along the manifold. As the true posterior is often intractable to compute, we use parameterized neural networks $q_{\emptyset}(\boldsymbol{z}|\boldsymbol{x})$ and $p_{\theta}(\boldsymbol{x}|\boldsymbol{z})$ to encode and decode the training data. The prior is represented by the standard normal distribution, $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and during training $\mathbf{z}$ is sampled using the reparameterization trick. The objective function is then formalized according to the evidence lower bound (ELBO) as

$$log p_{\theta}(\boldsymbol{x}|\boldsymbol{z}) \geq \mathcal{L}(\theta, \emptyset; \boldsymbol{x}, \boldsymbol{z}) = \mathbb{E}_{q_{\emptyset}(\boldsymbol{z}|\boldsymbol{x})}[log p_{\theta}(\boldsymbol{x}|\boldsymbol{z})] - \beta D_{KL}(q_{\emptyset}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \quad (5.1)$$

where the term on the left maximizes the log-likelihood of reconstructing the input given the sampled latent variable, $\mathbf{z}$, and the term on the right minimizes the Kullbach-Liebler divergence[88] between the approximate posterior and the standard normal distribution. We scale the KLD linearly during training with a Lagrange multipler, β, to control the relative magnitude of each loss term and prevent posterior collapse.[127]

E(n) Invariance and Equivariance

The properties of a molecule depend on the relative positions and orientations of its atoms with respect to each other, but do not change with arbitrary rotations or translations of the full molecular structure. Thus, a model which learns from 3D molecular data must construct representations that are invariant to these actions.

Take a molecule with coordinates $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_k) \in \mathbb{R}^{kx3}$ and node features $\boldsymbol{h} = (\boldsymbol{h}_1, \dots, \boldsymbol{h}_k) \in \mathbb{R}^{kxnf}$. A function acting on that molecule is said to maintain equivariance if

$$f(\mathcal{R}[\boldsymbol{x}], \boldsymbol{h}) = (\mathcal{R}[\boldsymbol{x}'], \boldsymbol{h}'), \forall \mathcal{R} \quad (5.2)$$

where $\mathcal{R}$ is any given operator in 3D space, $\boldsymbol{x}'$ is the updated set of molecular coordinates (i.e. $\boldsymbol{x}^{t+\Delta t}$ in a molecular simulation) and $\boldsymbol{h}'$ is the updated set of node labels. Similarly, if the function returns updated features but does not update the molecular coordinates, it is said to maintain invariance if

$$f(\mathcal{R}[\boldsymbol{x}], \boldsymbol{h}) = \boldsymbol{h}', \forall \mathcal{R} \quad (5.3)$$

The group of all rotations, translations and reflections is known as E(n) and when applied to 3D Euclidean space is known as E(3). E(n) invariance/equivariance is maintained when these equations hold true for the E(n) group. Note that in either case, the node features remain invariant and that a 3D molecular embedding constructed from these features will be independent of the coordinate system in which the molecule is embedded.

Deep Probabilistic Language Models

The design of Vagrant is, in part, inspired by work published in the field of machine translation and image captioning.[196–198] These models autoregressively reconstruct labeled sequences, $\boldsymbol{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_m\}$, by conditioning on deep machine-learned embeddings of the input data, $\mathbf{X}$. The architecture of the encoder depends on the type of input data being fed to the model while the decoder is often a left-to-right deep probabilistic transformer. Conditioning is applied using cross-attention as defined by Vaswani et al. (2017)

$$Q = W_q \cdot Enc(\boldsymbol{X}), V = W_v \cdot Enc(\boldsymbol{X}), K = W_k \boldsymbol{Y} \quad (5.4)$$

$$attn(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (5.5)$$

where the keys, K, are a linear transformation of the labeled sequence, the queries, Q, and values, V, are linear transformations of deep learned embeddings of the encoded input data, and d_k is a scaling

value. In the case of Vagrant, we use cross-attention to condition the generation of SELFIES on 3D information learned by the equivariant encoder network.

5.5 Vagrant: Graph-to-String Translation

We now describe the model framework of Vagrant, a graph-to-string translation VAE for generating candidate molecules with optimal 3D properties. Vagrant takes a 3D molecular graph, X, as input and constructs a latent embedding, $\mathbf{z}$, using a set of E(3) equivariant graph neural network (EGNN) layers.[81] This embedding is then used to condition the reconstruction of a SELFIES string using a set of transformer layers with cross-attention (Fig. 5-2a). We additionally embed a quantum chemical property in the latent space by using $\mathbf{z}$ as the input to a property prediction network.[93]

During inference, a latent embedding is sampled, and its corresponding property and SELFIES are predicted by the property prediction and decoder networks (Fig. 5-2b). Because the latent space is conditioned on 3D structural information during training, the sampled embedding contains knowledge of the relationship between the 3D structure of the predicted candidate and the predicted property. We can then navigate the latent space with previously described numerical techniques (see Appendix C for examples) to implicitly adjust the 3D coordinates of candidate structures without the need to generate the coordinates explicitly.

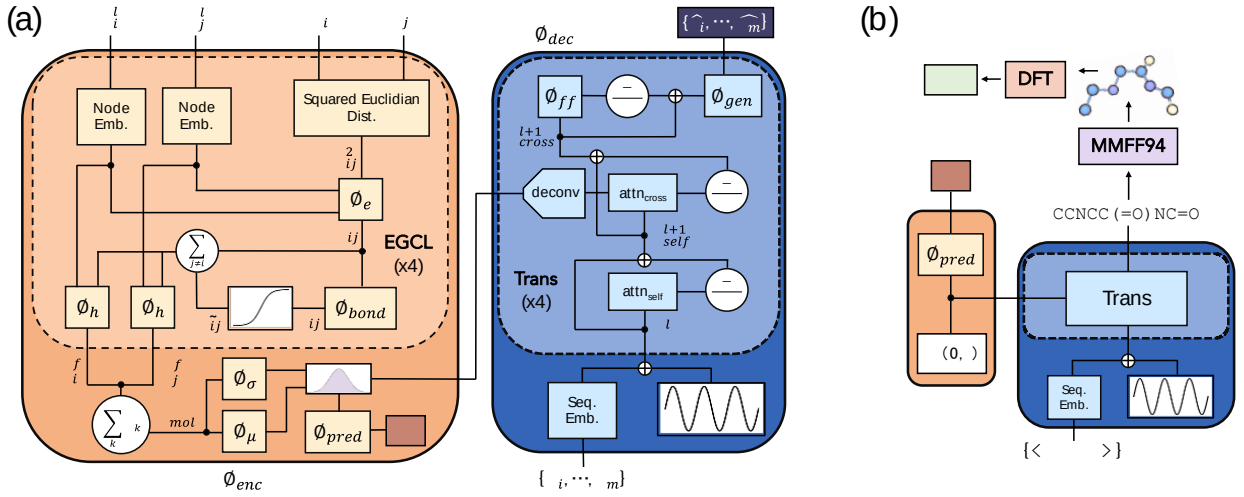


Figure 5-2. Overview of the Vagrant graph-to-string translation framework during a) training and b) inference. a) The invariant encoder (orange) takes the node level features and atomic coordinates as inputs and constructs a 3D-aware latent embedding, $\mathbf{z}$, that is used to condition the transformer decoder (blue). b) A 3D embedding is sampled and used to condition generation of a novel sequence and predict the DFT-level property. External validation tools are used to evaluate the accuracy of the model predictions.

Model Objective. We represent each input molecule, $\mathcal{M} = (X, \boldsymbol{Y})$ as a tuple with 3D molecular graph, X, and SELFIES string, $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$. The 3D graph is used to condition the latent embeddings through the encoder, $\emptyset_{enc}(\boldsymbol{z}|X)$, and the SELFIES string is reconstructed by conditioning the decoder on the resampled latent embedding, $\emptyset_{dec}(\boldsymbol{Y}|\boldsymbol{z})$. We build our graph with nodes $v_i \in \mathcal{V}$ containing the coordinates, $\boldsymbol{x}_i \in \mathbb{R}^3$, and features, $\boldsymbol{h}_i \in \mathbb{R}^{nf}$, of a single atom within the molecular graph. The SELFIES string is a sequence of m tokens where each token, $\boldsymbol{y}_i$, is chosen from the SELFIES vocabulary.

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{KLD} + \mathcal{L}_{pred} \quad (5.6)$$

The training objective seeks to maximize the log likelihood of the reconstructed SELFIES string while minimizing the KLD between $p(\boldsymbol{z})$ and the standard normal distribution, and the mean squared difference between the predicted property and the true property value. Additional details on the training objective can be found in Appendix C.

E(n) Invariant Graph Encoder

We model the encoder , $\emptyset_{enc}(\boldsymbol{z}|X)$, using the equivariant graph convolutional layers (EGCL) defined by Satorras et al. (2021). Each EGCL layer takes the atomic coordinates and node features as input and returns the updated node features for each atom according to

$$\boldsymbol{h}_i^{l+1} = EGCL[\boldsymbol{x}^0, \boldsymbol{h}^l] \quad (5.7)$$

where l is the index of the current layer. Our atomic coordinates are static throughout the encoder thus the EGCL layer is E(n) invariant to the input structure.

Each encoder layer consists of an edge network, $\emptyset_e$, that calculates the message passed between nodes i and j, and a node network, $\emptyset_h$, that updates the features on each node based on the aggregated messages. The messages and updated node features are calculated as

$$\boldsymbol{m}_{ij} = \emptyset_e(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, d_{ij}^2) \quad (5.8)$$

$$\boldsymbol{h}_i^{l+1} = \emptyset_h(\boldsymbol{h}_i^l, \sum_{j \neq i} \tilde{e}_{ij} \boldsymbol{m}_{ij}) \quad (5.9)$$

where d_{ij} is the euclidian distance between atoms i and j, and $\tilde{e}_{ij}$ is a learned attention mask that weights the importance of each message. Both $\emptyset_e$ and $\emptyset_h$ are modeled by fully connected neural networks.

Readout. The output of the final EGCL layer is $\boldsymbol{h}^f$, a matrix containing k vectors representing each atomic node. We condense this matrix into a single molecular representation of size d_{model} with the readout function

$$\boldsymbol{h}_{mol} = \frac{1}{k} \sum_k \boldsymbol{h}_k^f \quad (5.10)$$

The mean of a set of vectors is invariant to any permutation of those vectors thus our condensed molecular representation is still E(n) invariant. We then project $\boldsymbol{h}_{mol}$ through two fully connected layers to obtain the mean and variance vectors and sample from them using the reparameterization trick to obtain the final 3D latent representation, $\mathbf{z}$.

$$\boldsymbol{\mu} = \emptyset_\mu(\boldsymbol{h}_{mol}), \boldsymbol{\sigma} = \emptyset_\sigma(\boldsymbol{h}_{mol}), \varepsilon = \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \varepsilon \quad (5.11)$$

Property Predictor. The property prediction network, $\emptyset_{pred}$, consists of a stack of fully connected networks separated by an activation function. To increase property prediction performance during

sampling, we use $\mathbf{z}$ as the input to $\emptyset_{pred}$ so that the prediction network is robust to small variations within local regions of molecular phase space.

Probabilistic Transformer Decoder

We model the decoder, $\emptyset_{dec}(\boldsymbol{Y}|\boldsymbol{z})$, using a deep transformer with cross-attention. Each transformer layer contains a self-attention head, a cross-attention head, and a fully connected layer with residual connections and batch normalization. The SELFIES string is encoded as a one-hot vector and then expanded through an embedding layer to create the initial sequence representation, $\boldsymbol{y}^0$. This representation is transformed through each transformer module according to

$$\boldsymbol{y}^l_{self} = attn_{self}[Q(\boldsymbol{y}^{l-1}), K(\boldsymbol{y}^{l-1}), V(\boldsymbol{y}^{l-1})] \quad (5.12)$$

$$\boldsymbol{y}^l_{cross} = attn_{cross}[Q(\boldsymbol{y}^l_{self}), K(\boldsymbol{z}), V(\boldsymbol{z})] \quad (5.13)$$

$$\boldsymbol{y}^l = \emptyset_{ff}(\boldsymbol{y}^l_{cross}) \quad (5.14)$$

where Q, K and V are linear transformations of the input and $\emptyset_{ff}$ is a fully connected layer. The attention modules use h heads with unique learnable weight matrices, $W_i^{QKV}, \ldots, W_h^{QKV}$. The self-attention module acts on the sequence representation and learns the long-range dependencies between tokens in the SELFIES string. The cross-attention module then embeds the sequence representation with 3D information from the upsampled latent feature map (Fig. 5-3). The feature map is used to bias the generative process towards molecules with specific quantum chemical property values (see Appendix C for examples).

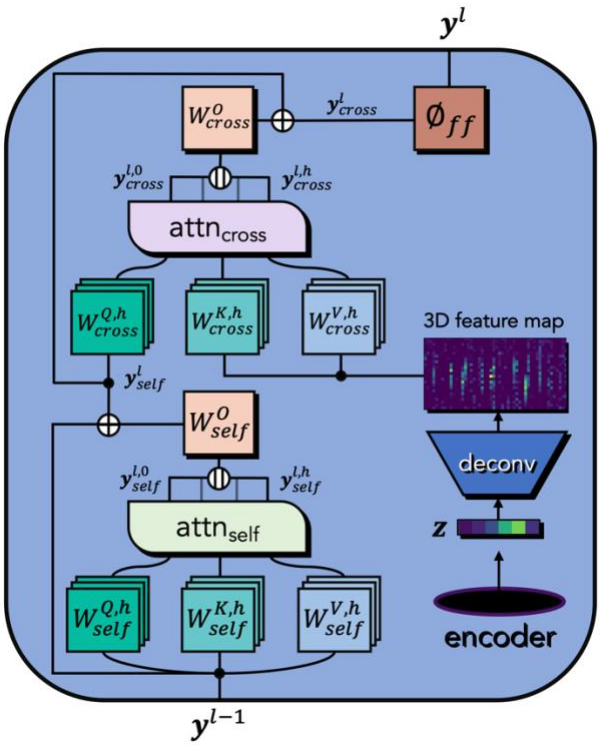


Figure 5-3. Flow of information through a single transformer layer conditioned on the 3D embedding, **z**, from the encoder.

Upsampling Z. The 3D embedding from the encoder is compressed so that only the most important features for connecting the predicted property to the reconstructed SELFIES are kept. However, due to the small size of the compressed embedding, we must expand it before applying cross-attention.

To accomplish this, we upsample $\mathbf{z}$ with a set of deconvolutional layers. Deconvolution is used to extract mid- to high-level image features from a low-dimensional latent representation.[199] In Vagrant, this operation can be thought of as reconstructing the mid- to high-level features of our 3D molecular structure. Empirically, we find deconvolution to be more effective than decompression with a simple fully connected layer (Appendix C).

Each deconvolution layer takes the latent embedding, $\mathbf{z}$, as input and applies a deconvolutional filter, f, to perform a transposed convolution

$$\boldsymbol{z}^{l+1} = deconv[\boldsymbol{z}^{l}, f^{l}_{k,c,s,p}] \quad (5.15)$$

where k is the filter size, c is the number of output channels, s is the stride, and p is the padding. The channels are used to expand $\mathbf{z}$ along the model dimension and the filter size and stride are used to expand $\mathbf{z}$ along the sequence dimension.

Generator. The final transformed sequence representation, $\boldsymbol{y}^{f}$, is passed through a generation network, $\emptyset_{gen}$, that compresses the size of the last dimension from d_{model} to d_{vocab}. The output of $\emptyset_{gen}$ is used to calculate the probabilities of each token in the sequence.

Sampling

Novel molecules are generated from Vagrant by sampling a latent embedding, $\boldsymbol{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})$, and using it to condition autoregressive SELFIES generation. We test two sampling methods – direct and robust. In direct sampling, we only generate one SELFIES and predicted property for each sampled latent variable. In robust sampling, we generate n SELFIES and predicted properties per latent variable by perturbing the latent variable n times within a radius, Δ (Algorithm 1). Robust sampling reduces the variance in local regions of the latent space by finding the most frequently occurring SELFIES within that region and averaging the predicted properties across each of those instances.

Algorithm 1 Robust Sampling

Input: radius Δ, perturbations n
Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = 1$ **to** n **do**
  Perturb $\mathbf{z}_i = \mathbf{z} + \Delta \cdot \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Predict property $\alpha_i = \phi_{\text{pred}}(\mathbf{z}_i)$
  Generate sequence $\mathbf{Y}_i = \phi_{\text{dec}}(\mathbf{z}_i)$
end for
Keep sequence $\mathbf{Y} = \text{argmax}_{x \in n} \sum_i \delta(\mathbf{Y}_i, \mathbf{Y}_x)$
Keep property $\alpha = \frac{1}{\sum_i \delta(\mathbf{Y}_i, \mathbf{Y})} \sum_i \delta(\mathbf{Y}_i, \mathbf{Y}) \alpha_i$

Coherence. Additionally, we introduce a new metric to evaluate the quality of novel samples generated from the latent space. We first define a translation function, f_{trans}, that transforms one molecular representation to another. The translation function does not need to be differentiable, for instance we use the MMFF94 force field[200] as the translation function that maps an output SELFIES to its 3D molecular graph, $X = f_{trans}(\boldsymbol{Y})$.

We use f_{trans} to compare the similarity between a newly generated molecule and the reconstructed version of that molecule when it is translated back into its input representation and fed through the model again. The coherence of that sample is defined as

$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \boldsymbol{Y} = \emptyset_{dec}(\boldsymbol{z}) \quad (5.16)$$

$$\boldsymbol{z'} = \emptyset_{enc}(f_{trans}(\boldsymbol{Y})), \boldsymbol{Y'} = \emptyset_{dec}(\boldsymbol{z'}) \quad (5.17)$$

$$coherence = sim(\boldsymbol{Y}, \boldsymbol{Y'}) \quad (5.18)$$

where sim is the Tanimoto similarity[201] between the originally sampled molecule and the reconstructed molecule.

Coherence measures the internal consistency of a VAE and can be thought of as a reconstruction accuracy for novel samples. In local regions of the latent space where coherence is low, the model is unsure of its structural predictions and unable to reproduce them. Alternatively, when coherence is high, the model consistently maps the same local region of the latent space to the same molecular structure.

5.6 Experiments

Experimental Setup

Datasets. We test the property prediction accuracy of our models on newly generated molecules by pretraining on the QM9 dataset.[43] QM9 contains ~134K 3D structures of small organic molecules and their associated quantum chemical properties calculated at the B3LYP/6-31G(2df,p) level of theory. We also use GEOM-Drugs[45] to test how our model scales to larger molecules. GEOM-Drugs contains the 3D structures of over 317K unique drug-like molecules with multiple structural conformers per species.

Baselines

- Vagrant-1D[108] – a Vagrant variant that operates only on 1D molecular sequences by replacing the equivariant encoder with a transformer
- cG-SchNet[160] – a conditional autoregressive 3D generative model for molecules
- EDM[182] – a conditional equivariant diffusion model for molecules

Evaluation. Models are evaluated on their ability to match the molecular distribution of the training set, successfully generate candidates with valid 3D structures and predict the quantum chemical properties of novel candidates. We use the Guacamol benchmarking platform[159] to report the percent of valid, unique, and novel (**VUN**) molecules generated by the models during sampling as well as the average **KLD** of several physicochemical properties between the generated set and a held-out test set.

For the property prediction task, we report the prediction accuracy for novel candidates as the mean absolute error (**MAE**) between the true and predicted value. We chose to predict the isotropic polarizability as it's been shown to be a key property in determining molecular and materials functionality[202,203] and is commonly used as a target in other studies.

DFT simulations have several failure modes (Appendix C) so we report the simulation success rate, **SSR**, which measures the ability of models to generate starting structures that converge to a stable energy minimum. We also calculate the joint probability of generating a valid, unique, and novel molecule that is successfully simulated as the true success rate, **TSR** = SSR · VUN. The definition of SSR can be expanded to include a prediction accuracy threshold and we report this as **TSR**(MAE ≤ ζ) where ζ is the desired accuracy per sample. This metric approximates the hit rate in a real-world materials design scenario.

DFT Validation. Novel molecules and their predicted properties are evaluated with DFT simulations at the same level of theory used to calculate the properties in the QM9 dataset (see Appendix C for details). For models which generate 3D coordinates (cG-SchNet/EDM), we use the predicted coordinates as the starting structure for simulations. For models which generate 1D sequences (Vagrant/Vagrant-1D), we predict the coordinates using the MMFF94 force field and use the lowest energy conformer as the starting structure. More details on the procedure for generating and evaluating candidate molecules can be found in Appendix C.

Results

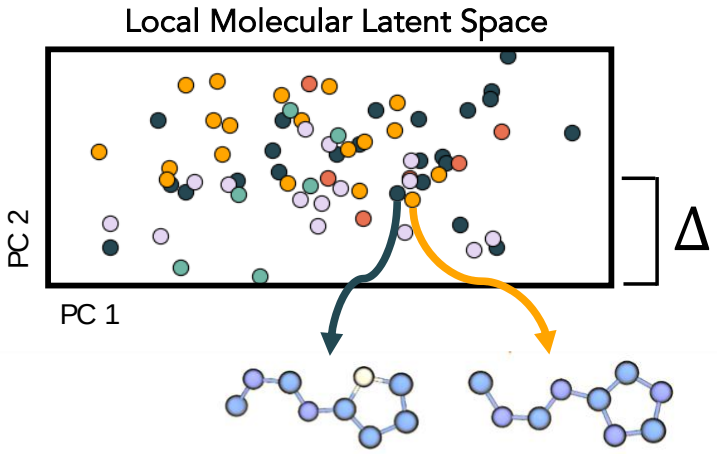
The generative performance of Vagrant on QM9 compared to the baselines is shown in Table 5-1. Vagrant achieves the lowest MAE between the predicted and true values of polarizability for novel candidate molecules. This supports our claim that Vagrant can learn and optimize the relationship between 3D geometries and properties without generating 3D coordinates directly. The poor predictive performance of Vagrant-1D illustrates the necessity of learning 3D embeddings for this task. However, Vagrant also outperforms both the 3D autoregressive and 3D diffusion models suggesting that generating molecular structures in 1D increases the model's capacity to learn the relationship between structure and function by reducing the complexity of generation.

Table 5-1. Performance of Vagrant and model baselines trained on the QM9 dataset. EGNN is a property prediction model that uses the same EGCL layers as Vagrant but is not generative. We report MAE results for this model on a held-out test set. The property prediction accuracy of EGNN can be seen as a lower bound for Vagrant as it uses the same encoder layers.

Model	**Sampling Method**	**α MAE**	**KLD**	**VUN**	**SSR**	**TSR**		
						ζ=1.0	ζ=5.0	ζ=10.0
Vagrant-1D	Direct	24.648	0.936	0.434	0.896	0.010	0.057	0.111
cG-SchNet	Conditional	5.304	0.869	**0.520**	0.854	0.054	0.251	**0.379**
EDM	Conditional	2.135	0.896	0.373	0.862	0.098	0.298	0.320
Vagrant	Direct	**1.904**	**0.940**	0.361	**0.933**	**0.137**	**0.319**	0.331
EGNN (lower bound)	---	0.140	---	---	---	---	---	---

EDM and cG-SchNet generate a higher proportion of valid, unique, and novel molecules than Vagrant. However, this comes at the cost of not representing the property distributions of the generated set as accurately. In addition to the improved property prediction performance, molecules generated by Vagrant have a higher KLD and SSR. A high KLD indicates that the physicochemical property distributions of the generated and test sets are closely matched. This corroborates the observation from Flam-Shepherd et al., (2022) that 1D generative models are better at matching property distributions than their 2D counterparts. The relatively lower SSR of the 3D generative models is primarily due to their inability to generate valid molecular coordinates for many samples (Appendix C). Together, these results demonstrate that Vagrant has learned to model and navigate the joint probability manifold of molecular structure and property better than the other baselines.

To evaluate this claim in a real-world design scenario, we calculate the TSR of each model at accuracy thresholds of 1, 5, and 10. TSR combines VUN, SSR, and MAE into a single metric that estimates the percentage of generated samples that match the specified design criteria. The accuracy thresholds are chosen to evaluate the models in design scenarios with high, mid, and low levels of desired accuracy. Despite its lower VUN, Vagrant excels at the design scenarios that require high and mid levels of accuracy, demonstrating 40% and 7% higher success rates than the next best models when evaluated on the thresholds of 1 and 5, respectively. As the accuracy threshold is increased, VUN is weighted more heavily in the TSR and cG-SchNet becomes the best performing model, however Vagrant is still competitive in these design scenarios as well.



Scatter Label	Blue	Orange
# Occurrences	25	20
Predicted α	87.60	87.00
True α	88.80	84.98
MAE	1.20	2.02

Figure 5-4. Robust sampling. Each color represents a unique SELFIES. Small changes in the latent embedding lead to structural differences upon decoding. Choosing the most frequently occurring SELFIES improves the property prediction performance of the model.

Sampling Methods. Next we evaluate the impact of sampling method and coherence filters on the MAE of molecules generated by Vagrant. We implement a strict coherence filter that requires the reconstructed sample to exactly match the original sample by removing any molecules with a coherence less than 1. We found that the coherence threshold positively correlates with property prediction accuracy and negatively correlates with VUN (we define an incoherent molecule as invalid, see Appendix C) and chose the threshold that maximizes the prediction accuracy. For robust sampling we use 100 perturbations per sample at a radius of 0.1 to ensure adequate exploration of local latent space.

We find that robust sampling increases the property prediction accuracy of Vagrant (Table 5-2). This is due to the high amount of structural variance in small local regions of molecular latent space. When sampling directly, the local variance will cause the model to occasionally choose less probable structures which have higher prediction errors (see Fig. 5-4, Orange). Robust sampling allows the model to reduce this variance by sampling the same local region many times and choosing the most probable structure (see Fig. 5-4, Blue). In aggregate, the structures with the highest local frequencies have lower prediction errors.

Table 5-2. Effect of sampling method on property prediction performance (α MAE)

Sampling Method	Vagrant MAE
Direct	1.904
Direct + Coherent	1.464
Robust	1.478
Robust + Coherent	**1.243**

We also find that our strict coherence filter improves the property prediction performance of Vagrant for both direct and robust sampling (Table 5-2). Direct + coherent sampling improves the property prediction accuracy of Vagrant by a similar amount as robust sampling alone. However, we observe a cumulative effect when pairing robust + coherent sampling indicating that the structural variance and coherence at a particular point in the latent space are at least somewhat independent of one another. These results support our assumption that the coherence of a sampled molecule can be used as an unsupervised indicator of the confidence the model has in that prediction. To our knowledge, this is the first mention of coherence in the literature, but we anticipate this concept to be universally applicable to autoencoders regardless of the data type.

We find that incoherent regions are distributed evenly throughout the latent space (Appendix C) driving our decision to implement coherence as a post-hoc filter. However, if f_{trans} were to be implemented as a differentiable function, a model could be directly biased to generate high coherence samples. For now, we leave this idea for future work.

Computational Efficiency. We compare the training and inference times of Vagrant to each of the baselines in Fig. 5-5. The efficiency benchmarks are conducted on an RTX 2080ti GPU and batch size is tuned to optimize GPU memory usage. Vagrant, Vagrant-1D, and EDM all exhibit similar training times. cG-SchNet is significantly less efficient due to the slow nature of autoregressively learning on partially decomposed graphs. As expected, sampling directly from either VAE model is orders of magnitude faster than sampling from the 3D generative models. The fast inference time of Vagrant is a major benefit of the architecture as it allows for the much quicker exploration of 3D molecular phase

space. The efficiency gap between Vagrant and the 3D generative baselines is thus widened when taking both TSR and inference time into account. It also makes more computationally intensive methods such as robust sampling or coherence filtering viable. For instance, we generate 2 orders of magnitude more molecules during robust sampling than either EDM or cG-SchNet but achieve an inference time on the same order of magnitude.

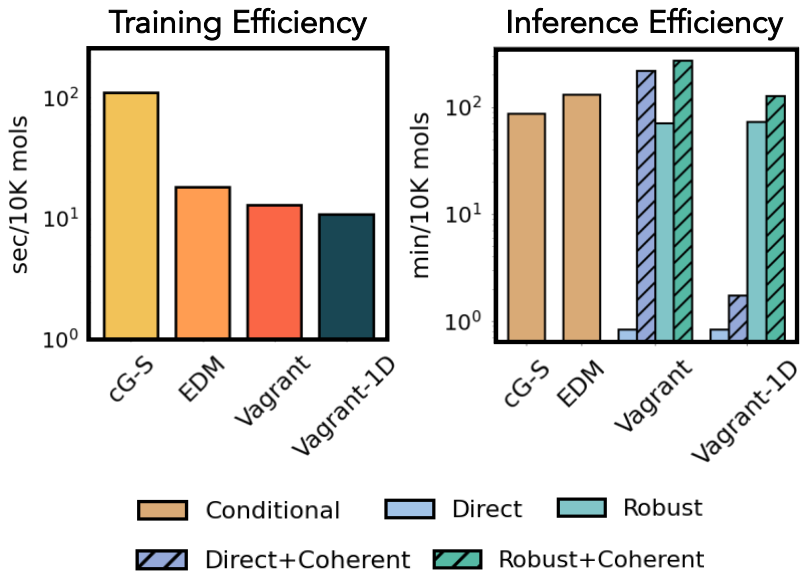


Figure 5-5. Computational efficiency of Vagrant compared to baselines.

GEOM-Drugs. We expect that the increased efficiency of Vagrant's decoding process will allow it to scale easily to larger molecules. We briefly explore this idea by comparing the generative performance of Vagrant and EDM on GEOM-Drugs. We use the training and evaluation protocol outlined in Appendix C. The larger molecules and increased atom diversity in GEOM-Drugs make it much easier for Vagrant to generate novel molecules compared to QM9, and the transformer decoder can easily scale to longer sequence lengths without a drop in validity. EDM, however, struggles to generate valid molecules of that size. Vagrant is also still much better at reproducing the property distributions of the test set (Table 5-3). The 3D structures of a few drug-like molecules generated by Vagrant are shown in Appendix C.

Table 5-3. Generative performance of models pretrained on GEOM-Drugs.

Model	VUN	KLD
EDM	0.359	0.332
Vagrant	**0.997**	**0.688**

This result and the recent success of DiffSBDD, an equivariant diffusion model applied to drug design,[189] suggests our model framework could be successfully scaled up to similar tasks. For instance, the Vagrant decoder could be conditioned on a joint 3D feature map of ligand and binding pocket for shape-based ligand design. Vagrant could also be applied to electrochemical design tasks that require quantum chemical property optimizations, like the discovery of new stable organic redox species for use in flow batteries. We save research on these potential applications of Vagrant for the future.

5.7 Conclusions

We introduce Vagrant, an E(3) invariant graph-to-string transformer model that learns to optimize molecular coordinates implicitly within a 3D-aware latent space. We show that Vagrant can predict the DFT-level properties of newly generated molecules at a higher accuracy and more efficiently than other comparable 3D generative models. Finally, we introduce two novel sampling methods that exploit characteristics of the VAE latent space to further improve generative performance.

6. Conclusions and Prospective Future Work

Herein, we have presented a partial overview of the field of de novo molecular design through close examination of the most salient architectural features of three generative models. In chapters 3 and 5, we analyzed the statistical and generative features of the transformer VAE architecture applied to the problems of drug discovery and quantum chemical property optimization. We found that a statistical understanding of the latent molecular embeddings learned by each model was critical to both evaluating and controlling their generative and optimization processes. The disentanglement between latent embedding variables allowed us to predict the generative performance based on information theoretic ensemble properties and independently control various structural and functional features with simple numerical interpolations.

In chapter 4, we took a different approach that uses prompt-based conditioning to bias the generation of molecular structures towards a user defined joint structural-functional distribution based on multimodal text. We hope that this text-based conditioning approach can expand the reach of de novo molecular design algorithms to chemistry researchers who are not well-versed in data science and machine learning. We also found evidence that positive knowledge transfer occurs when incorporating additional modalities, which points to the potential of foundational multimodal chemistry models to further improve upon the molecular design performance of the current state-of-the-art.

There are myriad ways in which this research could be iterated and expanded upon. While we examined the performance of these models in relation to a variety of organic molecule structural distributions, the properties we modeled are closer to toy systems than real-world design scenarios. While this allowed us to study and compare the features of model architectures, we have not yet applied these architectures to a specific multi-objective molecular design scenario. Published results from similar models suggest that we would find success in this realm, and this is an exciting avenue of future work.

While the statistical observations we made of the latent space allowed us to understand and improve the sampling performance of each model in a post-hoc manner, building these features directly into the model as mathematical priors could also help further improve model performance without relying on complex analytical postprocessing and sampling algorithms. For instance, controlling the mutual information between latent variables, predicted properties, and decoded molecular structures could allow us to balance the JPM-SF such that it is maximally approximative of the features that are most important to the design task. Defining a robust mathematical definition to measure these phenomena could also provide us with an analytical tool for comparing models that perform the same task with disparate architectures, based on the statistical relationship between embedding features and model predictions.

The field of inverse design is at a point where it's simultaneously ripe for further exploration and rapid maturation. The technologies that exist today are already being employed for drug discovery at industry scale, yet we have not even come close to reaching the future potential of AI-driven materials discovery. Continued improvements in model architectures and efficiency, development of large biological and chemistry datasets for massive scaling, and reductions in the cost of compute signal the imminent arrival of a paradigm shift in the way we research and discover novel chemistries. I hope that the reader who has made it this far has found some value in the research and ideas presented herein, and that we may continue to cross paths in these uncharted waters.

Appendix A

Additional Details on Model Construction

All models were constructed using the PyTorch python package and trained on NVIDIA GeForce RTX 2080 Ti GPUs. A batch size of 500 was used for all model types besides Moses. SMILES strings are tokenized during training and embeddings the same size as the model dimension are used as inputs. The maximum length of SMILES token for all model types is 127 including <start>, <stop> and padding tokens. All recurrent layers are unidirectional and masking for the transformer is done sequentially such that for a sequence of length t, the model attempts to predict the token at position $t + 1$. The same model architectures are used for both the ZINC and PubChem datasets. For the recurrent models, teacher forcing is partially used during training by concatenating the input embeddings with the unbottlenecked model memory prior to being sent to the decoder GRU layers.

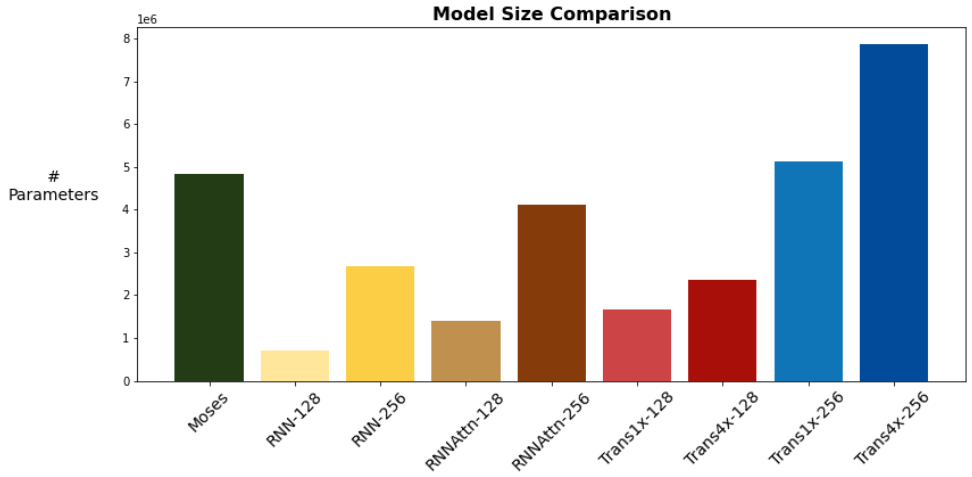


Figure A-1. Size comparison of all model types.

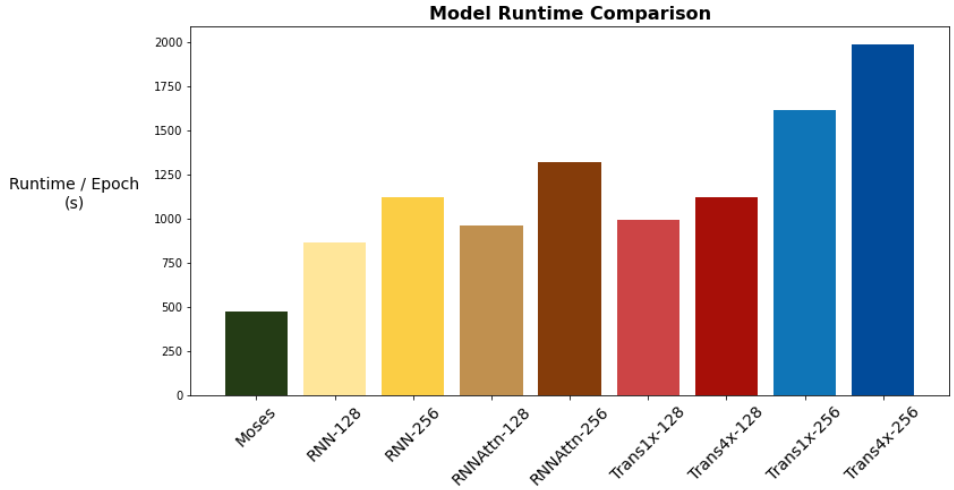


Figure A-2. Runtime of all model types on ZINC training set. The increased efficiency of the Moses model is due to the number of padding tokens. This is variable for the Moses construction based on the input data but fixed for the other models based on the longest SMILES string within the PubChem dataset. Fixing the

maximum sequence length simplified the construction of the convolutional bottleneck for different model dimensions.

Token Weights and KL Annealing

SMILES strings have an unbalanced token distribution so it is necessary to weigh loss by the frequency with which the token appears so that the model does not just learn to repeat the most frequent token. Characters are weighed by their proportional log frequency and then scaled to values between 0.5 and 1.0 (so the most frequent characters can represent no less than 50% of their calculated loss). The padding character is manually set to a weight of 0.1. A KL Annealer is used to linearly increase β to avoid posterior collapse. For all trials except the MosesVAE, β is increased from 0 to 0.083 over 100 epochs. For MosesVAE, β is linearly increased from 0 to 0.05 over 100 epochs.

Convolutional Bottleneck Layers

A convolutional bottleneck was used for the two attention-based architectures based on the size of the contextual embedding exiting the encoder and entering the decoder. Conceptually, the embedding matrix of these models is closer in shape to the learned image representation in a CNN than the contextual embedding vector of the RNN so our intuition was to try bottlenecking with convolutional layers. Empirically, we found our intuition to be correct as the reconstruction performance of the models is much better when using the convolutional bottleneck (Fig. A-3).

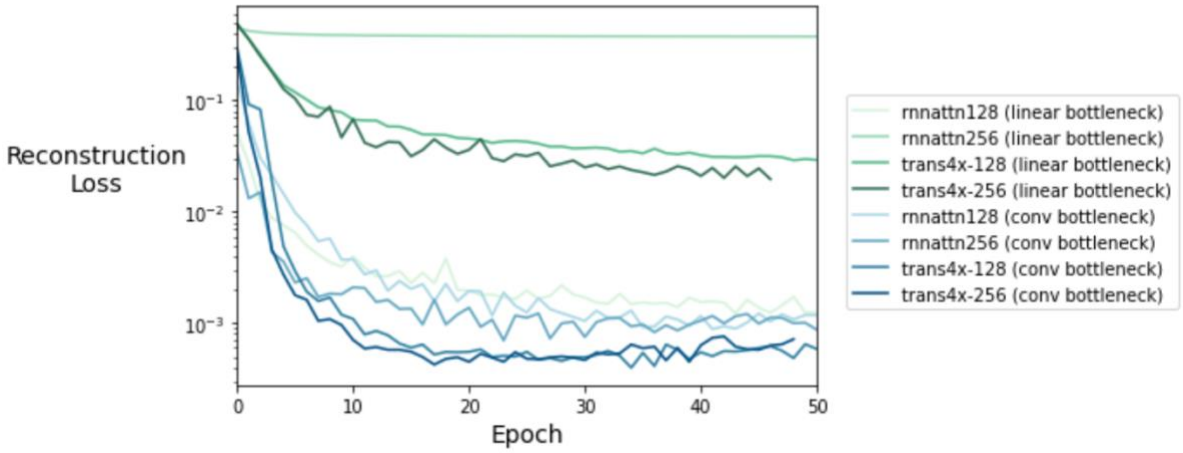


Figure A-3. Comparison of reconstruction loss for the attention-based architectures with a linear bottleneck and convolutional bottleneck. With the exception of the RNNAttn-128 model, the convolutional bottleneck outperforms the linear bottleneck.

We illustrate the compression of data through the convolutional bottleneck in Fig. A-4. The architecture consists of three 1D convolutional layers that compress the contextual embedding to a size 576 vector for model dimensions of 128, 256 or 512. The final 576 size vector is then compressed with a linear layer to the mean and logvar vectors before reparameterization. Each 1D conv layer is attached with a 1D MaxPool layer of size 2. After compression, three 1D deconvolutional layers are used to upsample the bottleneck back to the original size of the contextual embedding. The parameters for the convolutional layers depend on the size of the model and are listed in full in Table A-1.

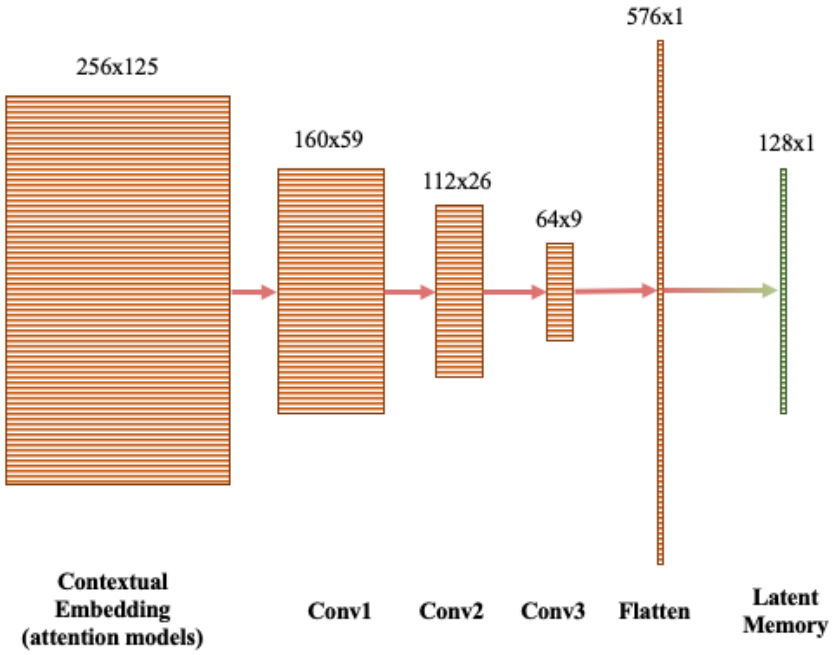


Figure A-4. The shape of the contextual embedding within the model as it travels through the convolutional bottleneck. A similar set of deconvolutional layers are used to upsample back to the original shape from the latent memory before being sent into the decoder.

Table A-1. Convolutional bottleneck parameters.

	Conv 1		Conv 2		Conv 3	
Model Dim	**Channel Size**	**Kernel Size**	**Channel Size**	**Kernel Size**	**Channel Size**	**Kernel Size**
128	96	9	80	9	64	8
256	160	9	112	9	64	8
512	288	9	176	9	64	8

	Deconv 1			Deconv 2			Deconv 3		
Model Dim	**Channel Size**	**Kernel Size**	**Stride**	**Channel Size**	**Kernel Size**	**Stride**	**Channel Size**	**Kernel Size**	**Stride**
128	80	11	4	92	11	3	128	11	1
256	112	11	4	148	11	3	256	11	1
512	176	11	4	260	11	3	512	11	1

Predicting SMILES Length During Inference

There are two inputs to the transformer decoder, the model memory and the input mask which consists of 0s for all padding tokens and 1s for every other token. The mask explicitly tells the model the length of the SMILE string so during inference it must also have this information or else it will

decode molecules from memory that are much longer than they should be. To account for this, we attached two linear layers of size d_{model}*2 to the latent memory and instructed the model to predict the correct length of the molecule during training and included this in our transformer loss function. Then during inference, we first predict the length of the SMILE string using our randomly sampled latent vector as an input and use this to create the correct length mask to send to the decoder (so we do not need to know the length of the SMILE string before we decode it).

Model Complexity

The definition of complexity we use throughout this work stems from the definition introduced by Tishby et al.[129] in which the bottleneck of a model can be analyzed in two ways – predictive ability and compressibility. There is a "generalization gap" which bounds the amount of salient information the model could have captured but didn't and a "complexity gap" which bounds the amount of "unnecessary complexity" that exists within the bottleneck. The concept of complexity in this case is tied to the compressibility of the bottleneck (i.e. it is not an algorithmic complexity but merely a synonym for low information content noise). The use of the phrase "unnecessary complexity" implies a corresponding "necessary complexity" and so we have drawn a link between the total information content contained within the bottleneck and the "complexity" of the model. Thus, mentions of model complexity are referring to the models' ability to efficiently compress all of the salient information needed for reconstruction within the bottleneck. Others have made similar observations about the relationship between the loss function of the β-VAE and the compression of the latent memory,[127] although explicit use of the term complexity to describe this phenomenon has been limited to Tishby et al. and the descriptions herein as far as we are aware.

Additional Figures

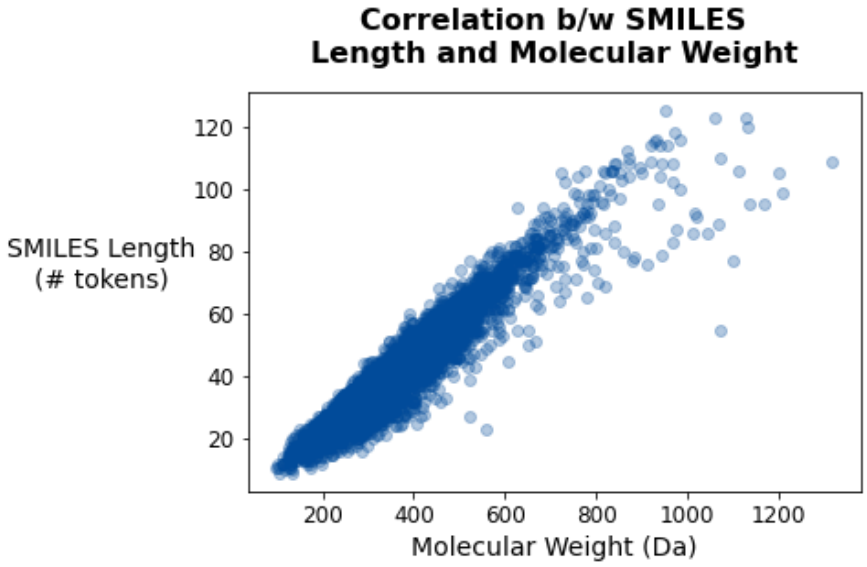


Figure A-5. The relationship between SMILES length and molecular weight (PubChem dataset).

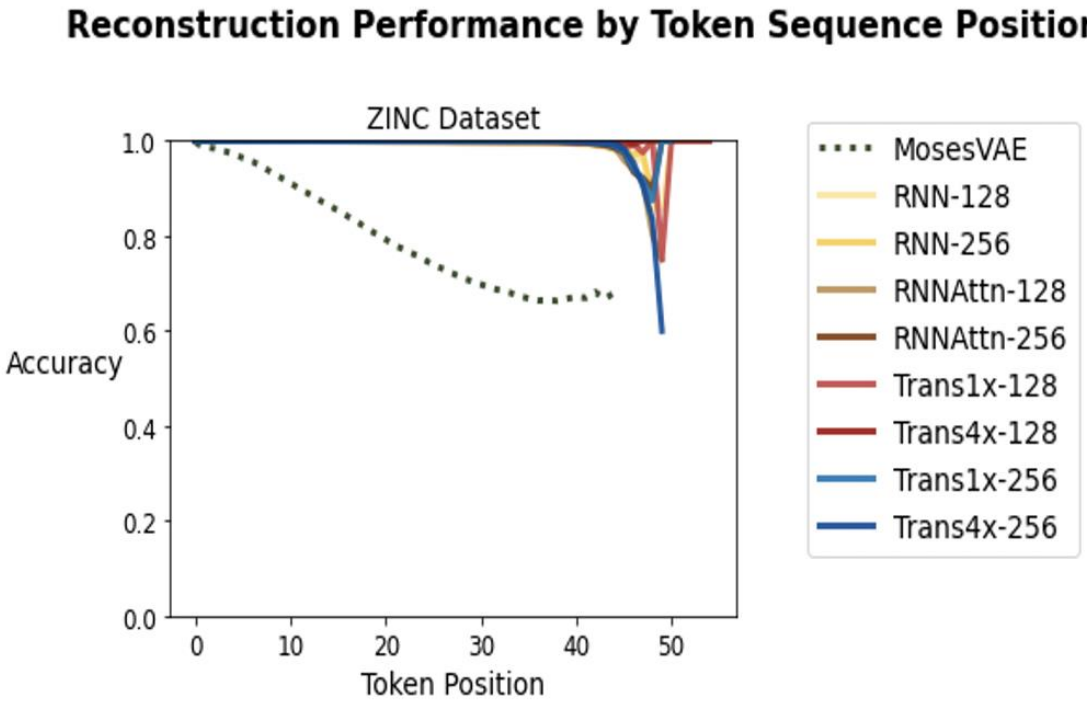


Figure A-6. The reduced reconstruction performance of the Moses model may be the result of a number of architectural and hyperparameter decisions. In addition to the differences mentioned in the procedure, we also used a more concise tokenization scheme (for instance `Br` was treated as a single token rather than tokenizing as `B` and `r` separately), we updated model weights more aggressively for tokens that appeared less frequently, and we used larger token embeddings. The exact degree to which these factors played a role in the model's performance is still unknown. Because we were able to replicate all of the reported metrics from the original Moses paper (Fig. A-7) we believe this is an accurate portrayal of the Moses model and include it to highlight an example of 'smeared' latent memory formation.

Table A-2. Reconstruction performance of all model types on ZINC dataset (MosesVAE was not saved at epoch 100 so accuracy at epoch 90 is reported instead).

Model Type	Token Accuracy	SMILES Accuracy
MosesVAE (Epoch 90)	0.1416	0.000
RNN-128	0.9988	0.9955
RNN-256	0.9986	0.9957
RNNAttn-128	0.9990	0.9963
RNNAttn-256	0.9986	0.9948
Trans1x-128	0.9996	0.9978
Trans4x-128	0.9996	0.9979
Trans1x-256	0.9997	0.9983
Trans4x-256	0.9996	0.9980

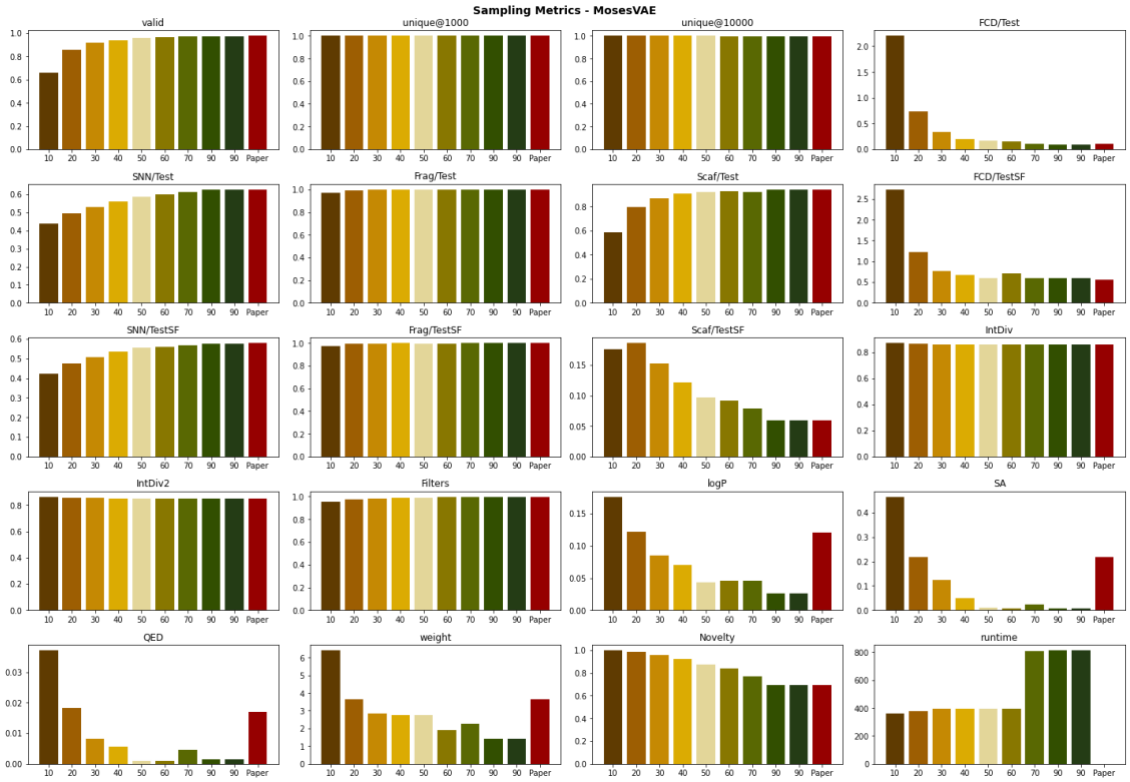


Figure A-7. Evaluating the MosesVAE on the suite of metrics presented in the MOSES paper. After 100 epochs, the model converges to all reported values from the paper validating the use of our trained Moses model as an example of the state-of-the-art as presented by Polykovskiy et al.[119]

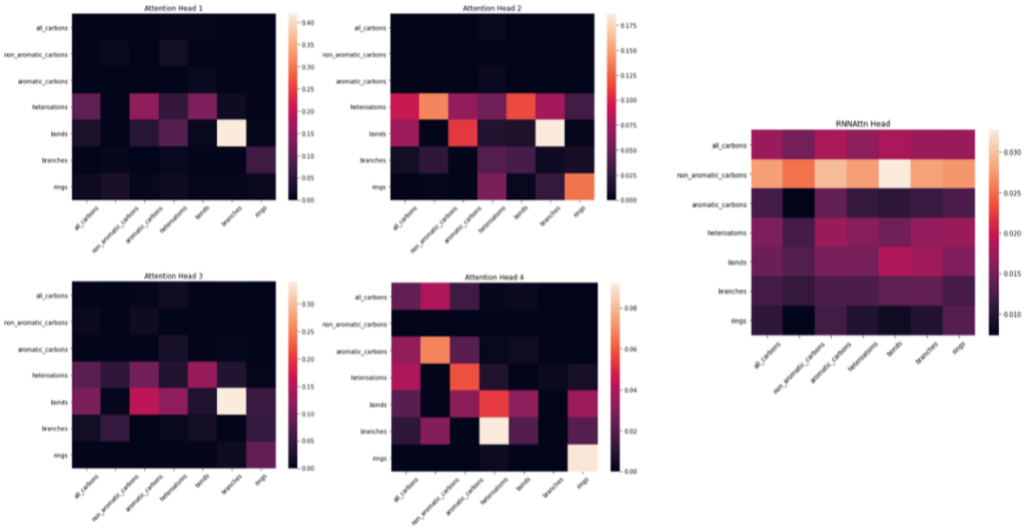


Figure A-8. Analysis of attention weights between structural and atomic groups. The four attention heads of the transformer learn unique molecular grammar rules, even for higher-level relationships such as the relationship between all heteroatoms and all explicitly enumerated bonds present within the structure. The RNNAttn head has given the most weight to the relationship between non-aromatic carbons and all other atomic/structural groups which is more useful for compressing long-range information efficiently than learning specific relationships that are important to molecular structure.

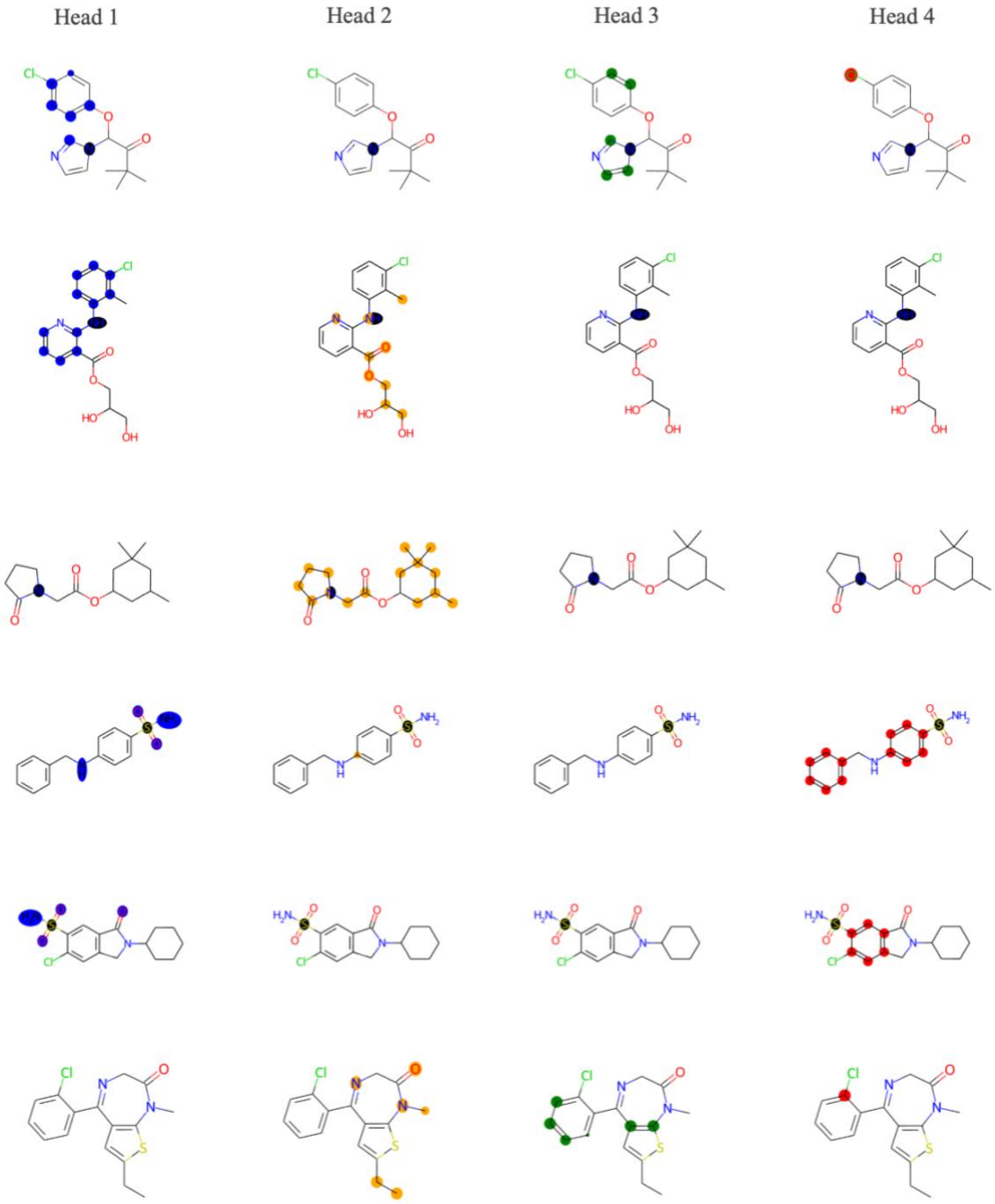


Figure A-9. Visualization of attention weights within the Trans4x-256 model of S and N heteroatoms for a variety of molecular structures. The learned patterns depend on the type of heteroatom. For instance, attention head 1 shows the relationship between N and aromatic carbons however a similar relationship between S and aromatic carbons is stored within head 4. The patterns are usually consistent for the same atom type across different molecular structures, however different patterns may also emerge depending on the molecular context around the atom (i.e. the aromatic S atom vs. the sulfonyl group). These relationships are heavily influenced by the input representation and may potentially be tuned by altering the type of information the model has access to.

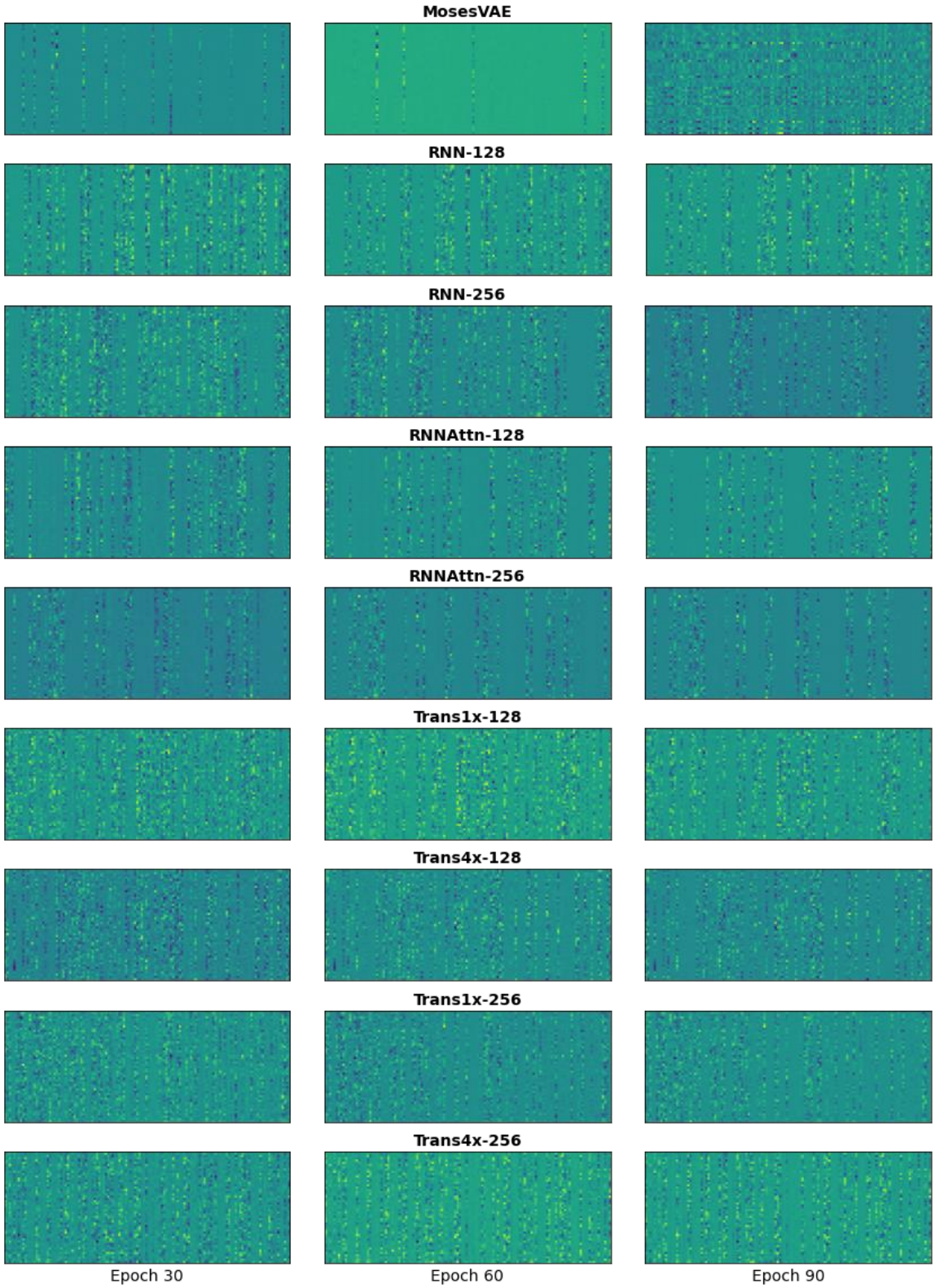


Figure A-10. Memory structures for all model types at epochs 30, 60, and 90

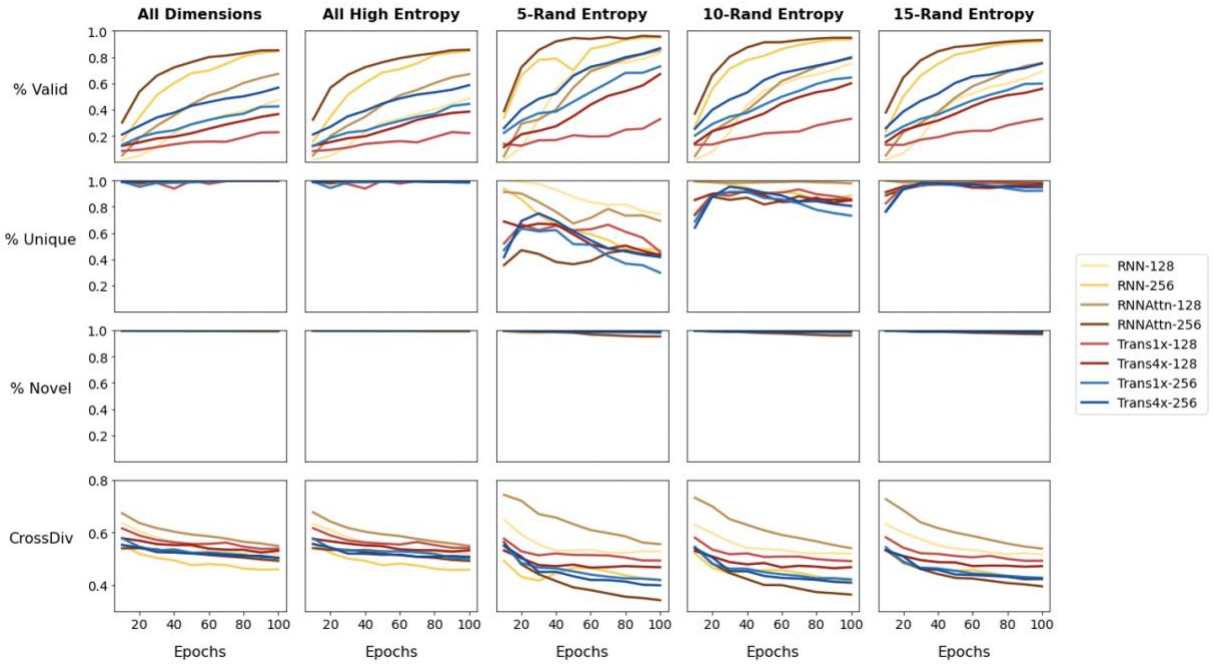


Figure A-11. Five different sampling schemes are tested for their effect on generative performance metrics – sampling all 128 latent dimensions, sampling only those dimensions with a high entropy (> 5 nats) and sampling k-random high entropy dimensions (k=5,10,15). 30,000 molecules were generated for each scheme. There is essentially no difference between sampling all dimensions and randomly sampling just the high entropy dimensions, however there is an improvement in validity when sampling from a small number of randomly selected high entropy dimensions. Sampling 15 random high entropy dims significantly increases % validity for all model types while maintaining high uniqueness, novelty, and exploration.

Appendix B

Prompt Design

Table B-1. Example of the multi-property optimization tasks and prompt designs used in the zero-shot evaluation. We color each prompt with the modality(s) that they are associated with.

Example	Prompt
Osimertinib	<text_type>IUPAC</text_type> <text>N-[2-[2-(dimethylamino)..prop-2-enamide</text> <property>tpsa</property><val>146.0</val> <property>logp</property><val>-0.5</val><mol>
Fexofenadine	<text_type>IUPAC</text_type> <text>2-[4-(1-hydroxy...methylpropanoic acid]</text> <property>tpsa</property><val>9</val> <property>logp</property><val>5</val><mol>
Ranolazine	<text_type>IUPAC</text_type> <text>N-(2,6-dimethylphenyl...piperazin-1-yl]acetamide</text> <property>logp</property><val>8.5</val> <property>aromatic_rings</property><val>0</val><mol> <property>f_count</property><val>1</val><mol>
Perindopril	<text_type>FuncGroups</text_type> <text>ethoxy,oxopentan,octahydroindole,carboxylic acid</text> <property>aromtic_rings</property><val>2</val>
Amlodipine	<text_type>FuncGroups</text_type> <text>aminoethoxymethyl,chlorophenyl,dihydropyridine,dicarboxylate</text> <property>ring_count</property><val>3</val>
Sitagliptin	<text_type>FuncGroups</text_type> <text>amino,trifluoromethyl,triazolo,pyrazin</text> <text_type>MolFormula</text_type> <text>C16H15F6N5O</text> <property>logp</property><val>3</val> <property>tpsa</property><val>6</val><mol>
Zaleplon	<text_type>IUPAC</text_type> <text>N-[3-(3-cyanopyrazolo...N-ethylacetamide]</text> <text_type>MolFormula</text_type> <text>C19H17N3O2</text>
PLogP/QED (*Drug-Likeness*)	<text_type>FuncGroups</text_type> <text>oxo,phenyl,triazaspiro,indole,carboxamide</text> <property>plogp</property><val>10</val> <property>qed</property><val>10</val><mol>
PLogP/DRD2 (*Biological Activity*)	<text_type>FuncGroups</text_type> <text>oxo,triazolo,methoxyethyl,benzimidazol,dimethylacetamide</text> <property>plogp</property><val>10</val> <property>drd2</property><val>10</val><mol>

Table B-2. Example of the conditional molecular structure generation tasks and prompt designs used in the zero-shot evaluation. We color each prompt with the modality(s) that they are associated with.

Task		Example	Prompt
Molecular	Rediscovery	Celecoxib	<text_type>IUPAC</text_type> <text>4-[5-(4-methylphenyl)..benzenesulfonamide</text><mol>
		Troglitazone	<text_type>IUPAC</text_type> <text>5-[[4-[(6-hydroxy...thiazolidine-2,4-dione]]]</text><mol>
		Thiothixene	<text_type>IUPAC</text_type> <text>(9Z)-N,N-dimethyl...thioxanthene-2-sulfonamide</text><mol>
Similarity	Sampling	Albuterol	<text_type>FuncGroups</text_type> <text>butylamino,hydroxyethyl,phenol</text><mol>
		Aripiprazole	<text_type>FuncGroups</text_type> <text>dichlorophenyl,piperazin,quinolin</text><mol>
		Mestranol	<text_type>FuncGroups</text_type> <text>ethynyl,methoxy,methyl,octahydro,phenanthren</text><mol>
Isomer	Generation	$C_{11}H_{24}$	<text_type>MolFormula</text_type> <text>C11H24</text><mol>
		$C_9H_{10}N_2O_2PF_2Cl$	<text_type>MolFormula</text_type> <text>C9H10N2O2PF2Cl</text><mol>
Median	Molecules	Camphor/Menthol	<text_type>FuncGroups</text_type> <text>heptan,methyl,trimethylbicyclo,ylcyclohexan</text><mol>
		Tadalafil/Sildenafil	<text_type>FuncGroups</text_type> <text>pyrazolo,triazatetracyclo,pyrimidin,methylpiperazin</text><mol>
Substructure	Sampling	Valsartan	<text_type>IUPAC</text_type> <text>methanoyl-methyl...phenyl]methyl]amine</text><mol> <property>logp</property><val>2.0</val>< <property>tpsa</property><val>77.0</val>< <property>bertzct</property><val>896.4</val><
		Deco Hop	<text_type>FuncGroups</text_type> <text>amino,hydroxy,quinazoline</text><mol>
		Scaffold Hop	<text_type>FuncGroups</text_type> <text>propanol,benzothiazol</text><mol>

Prompt Sampling Strategy

Prompts are stochastically generated from the available modalities by the following set of rules:

- The text modality is sampled uniformly from the list (IUPAC, FuncGroups, MolFormula, None). If None is selected then no text conditioning is included for that sample. This allows the user to perform property-only conditioning by leaving out the text conditioning during inference.
- If FuncGroups is chosen, then the number of functional groups, N, used for conditioning is sampled uniformly from 1-M where M is the total number of functional groups for the given molecule. Then N functional groups are selected from the list and concatenated with commas.
- Next, the number of property conditions, K, is sampled uniformly from 0-L where L is the total number of property modalities available for training. Then K properties are chosen from the list and their property names and values are added to the prompt after the text type and text. The ordering of property sub-modalities is also stochastic.

Training & Sampling Implementation Details

We use the GPT-NeoX Python library[204] developed with Megatron[205] and DeepSpeed.[206] We optimize the autoregressive log-likelihood (i.e. cross-entropy loss) averaged over a 256-token context. We set the global batch size as 2048, and the learning rate to 2×10^{-4}, and rely on the cosine decay. We use an Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\sigma = 10^{-8}$ and clip the gradient norm at 1.0. We use the Rotary positional embeddings,[207] parallel attention and feed-forward (FF),[204] and all dense layers in comparison to the original transformer decoder model architecture.[208]

We use a q temperature value of 1.0 for sampling for evaluating all 22 tasks. We found that this value gives us the best tradeoff between the validity and diversity of the generated molecules. For each GuacaMol task, we generate 128K samples to use for evaluation. This is on the order of the number of samples that are generated and evaluated during fine-tuning of the GuacaMol baselines. For the Drug-Likeness and Biological Activity tasks, we evaluate on 250 molecules randomly sampled from a subset of the ZINC dataset provided in Jin et al., (2019) in accordance with the methods outlined in Fu et al., (2021). For each molecule, we generate 1K samples which is on the order of the number of samples that are generated and evaluated during fine-tuning of the MIMOSA baselines.

Baseline Models

We compare MolJET to two sets of baselines – one for the GuacaMol tasks and another for the Drug-Likeness/Biological Activity tasks. The GuacaMol baselines include:

- **Best of Data Set**, the metrics evaluated on the top molecules from the ChEMBL dataset[42]
- **SMILES LSTM**, an LSTM model which is fine-tuned with the hill-climbing method[159]
- **SMILES GA**, a genetic algorithm that makes mutations to a SMILES string[209]
- **Graph GA**, a genetic algorithm that makes mutations directly to a molecular graph[210]

The Drug-Likeness/Biological Activity baselines include:

- **VJTNN**, a graph-to-graph translation VAE that utilizes adversarial regularization[162]

- **DeepGA**, a genetic algorithm enhanced with a discriminator neural network to improve molecular diversity[163]
- **MIMOSA**, a Markov chain Monte Carlo sampling strategy augmented by pretrained graph neural networks[87]

Model Performance on Individual GuacaMol Tasks

Table B-3 shows the detailed performance view on the GuacaMol benchmark. Aside from the rediscovery tasks, the final score for each metric is evaluated as a weighted average of the top 100 scoring molecules that were generated during sampling. The scores for individual molecules are based on their ECFP4[170] fingerprint similarities to the targets, calculated property values and structural features. These values are passed through a set of modifiers and thresholds to scale them between 0 and 1. The score is then calculated as the geometric mean of each scaled task-specific value. For further details on the metric definition of each benchmark, please refer to Brown et al., (2019).

Table B-3. Benchmark results on GuacaMol which contains both MPO and molecular structure generation tasks. Bold values indicate the best performing model and underlined values indicate the second best performing model.

Benchmark Category	Benchmark	Best of Data Set	SMILES LSTM	SMILES GA	Graph GA	MOLJET-GUAC (Zero-shot)	MOLJET-GUAC + Graph GA
MPOs	Osimertinib	0.781	0.894	0.880	0.937	0.914	**0.992**
	Fexofenadine	0.817	0.926	0.904	**1.000**	0.997	**1.000**
	Ranolazine	0.836	0.833	0.832	0.913	**0.920**	**0.920**
	Perindopril	0.701	0.764	0.644	0.803	**0.804**	**0.823**
	Amlodipine	0.696	0.885	0.678	0.888	**0.895**	**0.903**
	Sitagliptin	0.509	0.536	0.526	0.809	0.758	**0.823**
	Zaleplon	0.547	0.610	0.552	**0.728**	0.625	0.688
Rediscovery	Celecoxib	0.674	**1.000**	0.570	0.836	**1.000**	**1.000**
	Troglitazone	0.558	**1.000**	0.523	**1.000**	**1.000**	**1.000**
	Thiothixene	0.608	**1.000**	0.476	**1.000**	**1.000**	**1.000**
Similarity	Albuterol	0.522	**1.000**	0.871	**1.000**	**1.000**	**1.000**
	Aripiprazole	0.595	**1.000**	0.747	0.985	0.999	**1.000**
	Mestranol	0.520	**1.000**	0.695	0.945	**1.000**	**1.000**
Substructures	Valsartan	0.259	0.931	0.628	0.958	0.930	**0.977**
	Deco Hop	0.933	**0.996**	0.876	0.995	0.893	**0.996**
	Scaffold Hop	0.738	0.993	0.803	**1.000**	0.632	0.984
Isomers	$C_{11}H_{24}$	0.684	0.963	0.734	0.952	**1.000**	**1.000**
	$C_9H_{10}N_2O_2PF_2Cl$	0.747	0.860	0.757	0.955	**1.000**	**1.000**
Median	Camphor/Menthol	0.334	0.398	0.348	0.405	0.386	**0.416**
	Tadalafil/Sildenafil	0.407	0.408	0.377	0.429	**0.434**	**0.478**
Total	—	0.623	0.850	0.671	0.877	0.857	**0.900**

Reconstruction Tasks

To validate the ablation on the Text + Property vs. the Text-Only models, we construct two additional tasks that evaluate the model's performance on text-only conditioning - IUPAC Reconstruction and FuncGroup Reconstruction. An IUPAC reconstruction is counted as successful if the generated SELFIES string exactly matches the canonical SMILES from the holdout set after being decoded back into a SMILES and canonicalized. IUPAC Reconstruction is evaluated on 10000 randomly sampled IUPAC/SMILES pairs from the holdout validation set. A FuncGroup reconstruction is counted as successful when the SMILES string decoded from the generated SELFIES string matches the substructure pattern matching the requested functional group (we use SMARTS substructures for matching). We hand select 102 functional groups to test the model on its ability to recognize simple functional groups, basic nitrogen heterocycles, basic oxygen heterocycles, basic mixed heterocycles, double ring nitrogen heterocycles, double ring oxygen heterocycles, polycyclic aromatic hydrocarbons, fused rings, and phenyls among others. The full dataset will be made available upon request.

Appendix C

Additional Architectural Details

Node and Edge Features. We featurize nodes with one-hot vectors encoding atom type and scale the features by their relative atomic charges. The scaled one-hot vectors are then passed through an embedding layer to increase the node feature dimensionality from $\mathbb{R}^{kxnf}$ to $\mathbb{R}^{kxd_{model}}$ where nf is the number of scaled one-hot features and d_{model} is the dimensionality of the model. We find that including explicit hydrogens as an atom type improves the generative performance of our models (Table C-1). However, this doesn't scale well to larger molecules so we only include explicit hydrogens when working with the QM9 dataset.

Table C-1. Node and edge feature ablations.

Include H	Include Bonds	Reconstruction Accuracy	VUN	KLD
		0.7442	0.5517	0.9238
√		0.7706	**0.5823**	**0.9289**
	√	0.6966	0.5658	0.9244
√	√	**0.8119**	0.5798	0.9256

We also test two edge feature variants, one where no edge features are provided and one where we explicitly pass the bond types (single, double, triple, aromatic) as a four-dimensional one-hot vector. These features are expanded through an embedding layer to increase the dimensionality to d_{model} and used to update messages according to modified versions of Equations 5.7 and 5.8

$$\boldsymbol{h}_i^{l+1} = EGCL[\boldsymbol{x}^0, \boldsymbol{h}^l, \mathcal{E}] \quad (5.19)$$

$$\boldsymbol{m}_{ij} = \emptyset_e(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, d_{ij}^2, a_{ij}) \quad (5.20)$$

where $\mathcal{E}$ is the set of labeled bonds for molecular graph, X, and a_{ij} is the edge feature between nodes i and j. Including bonds increases the reconstruction accuracy on the held-out test set but decreases VUN and KLD (Table C-1). This suggests that passing explicit bond information to Vagrant causes it to overfit to the distribution of graph structures included in the QM9 dataset. When no bond information is included, the model is forced to infer the relationship between atoms with the learned attention map that weights the importance of messages. Removing the hard-coded relationship between atoms not only increases exploration of novel structures but improves property prediction accuracy as well (Table C-2).

Table C-2. Effect of bond type on property prediction accuracy.

Bond Type	∝ MAE
Explicit	1.652
Inferred	**1.478**

Sequence Features. SELFIES are featurized as a matrix of one hot vectors where each row is a specific token in the sequence. The vocabulary is determined by tokenizing every SELFIES string in the dataset using regex, keeping every unique token, and appending start, stop and pad tokens to control generation and allow for variable length sequences. Individual tokens may represent atoms, structural features such as branches and rings, or chemical constraints such as the maximal valency of a given atom.[78] There are 33 and 87 unique tokens in the vocabularies of the QM9 dataset and GEOM-Drugs dataset, respectively, and we use a maximum sequence length of 125 for both Vagrant and Vagrant-1D.

Left-to-Right Masking. We make predictions of the next token for every position in a sequence by using a $d_{seq}xd_{seq}$ triangular mask filled with ones below the diagonal and zeros above. The mask obscures tokens to the right of the target such that the probability of each token, t, can be written as a Markov chain, $P(\boldsymbol{y}_t|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})$. This is in contrast to bi-directional sequence models which mask portions of a sequence and have the model attend to the masked positions from both directions.[211] During training, we use teacher forcing to condition predictions of each token, $\boldsymbol{y}_t$, on the ground truth prior sequence, $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}\}$. During inference, next token prediction is conditioned on the predicted sequence up to that point, $\{\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_{t-1}\}$. The loss of each token prediction is inversely scaled by its frequency in the training set.

Latent Space Decompression. We describe expanding the latent embeddings from size d_{latent} to size $d_{seq}xd_{model}$ using a set of deconvolutional layers in the main text. Intuitively, the deconvolutional process corresponds to recovering mid- to high-level 3D features of the molecular structures from the compressed latent embeddings. We also present an empirical justification for decompressing with deconvolutional layers in Fig. C-1. The deconvolutional upsampling method achieves a much lower log-likelihood when reconstructing SELFIES from the test set than the linear upsampling method.

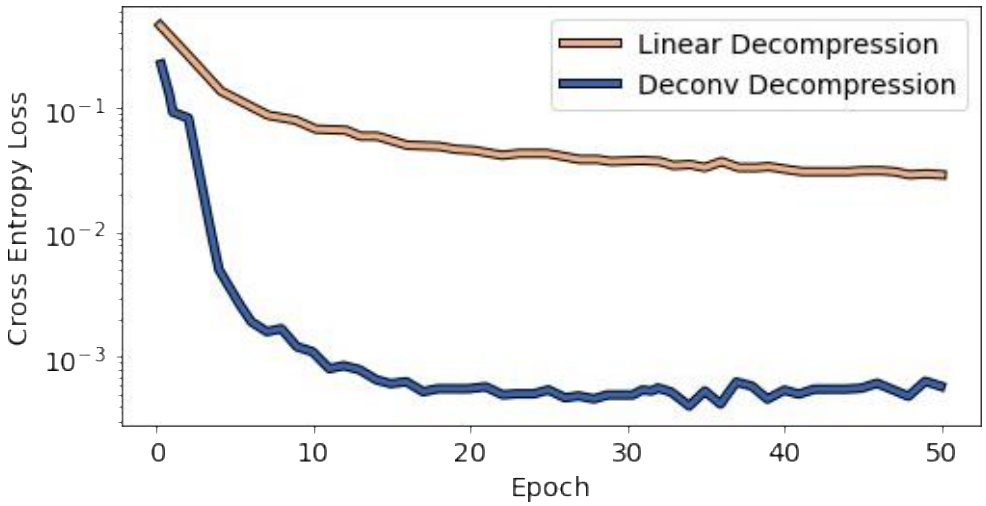


Figure C-1. Loss curve for linear vs. deconvolutional decompression of the latent space.

We also observe marked differences in the upsampled latent feature space when averaging across distinct groups of molecules (Fig. C-2) giving us some insight into the 3D conditioning that is applied to the decoder. Some properties, such as molecular weight, are a direct function of the SELFIES sequence (the molecular weight has a high correlation with the length of a SELFIES string). Differences in the feature maps for the low and high molecular weight groups could thus be learned

from either SELFIES or 3D molecular structures and are not evidence of a 3D-aware latent feature space. However, we also observe differences in the feature maps of molecules grouped by their measured polarizability. The difference between the feature maps of molecules with high polarizability and high molecular weight suggest that the model is not just learning a trivial correlation between an extensive property of the SELFIES string and the 3D property of interest, but is embedding 3D information learned by the encoder directly into the feature map and conditioning the generation of novel SELFIES with these 3D-aware features.

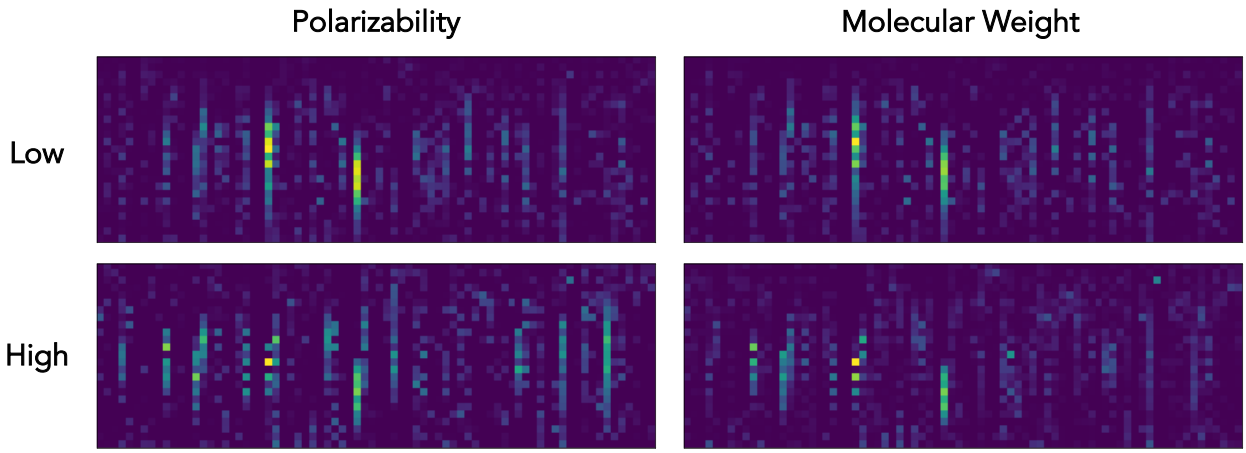


Figure C-2. Upsampled latent feature maps for four distinct groups of molecules – those with low/high polarizability and those with low/high molecular weight.

The differences between feature maps could provide insight into specific features that give independent control over each property (for instance, the 3 bright bands which are only observed on the right side of the high polarizability map). They not only provide intuitive insight into the mechanism by which the model transfers 3D information into it's predicted 1D sequence, but could also potentially be exploited to find a hyperplane within the latent space along which a property of interest could be effectively tuned. We leave additional work on such numerical optimization strategies for the future.

Additional Experimental Details

QM9 Training and Evaluation Procedure. We train Vagrant and all baselines except cG-SchNet for 3000 epochs on the QM9 dataset with an 80/10/10 train/test/validation split. Due to its slow training time, we only train cG-SchNet for 1000 epochs. To assess convergence, each model is evaluated at multiple epochs and the epoch with the lowest value of MAE is chosen as converged. A convergence plot for EDM, cG-SchNet, Vagrant-1D, and Vagrant with robust + coherent sampling is shown in Fig. C-3.

The evaluation procedure for each model is as follows. First, 10K molecules are sampled. Samples from EDM and cG-SchNet are drawn by conditioning on values of isotropic polarizability chosen from the distribution of property values in the training set. Samples from Vagrant and Vagrant-1D are drawn from the standard normal distribution using one of the four sampling combinations defined in the main text (direct, direct+coherent, robust, robust+coherent). Next, we measure the VUN and KLD of the generated sample set. 500 molecules are then drawn at random from the subset of valid,

unique, and novel molecules and simulated with DFT. The SSR, α MAE, and TSR are calculated from the results of the DFT simulations.

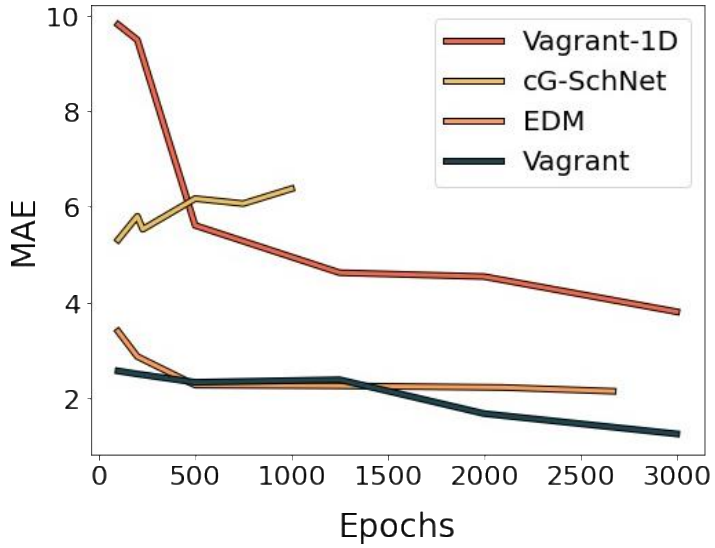


Figure C-3. Convergence of each model. cG-SchNet converges and begins to overfit within the first 100 epochs. EDM also converges faster than Vagrant, however Vagrant demonstrates a a better or equal property prediction performance than EDM throughout the entire training regime. Both Vagrant and Vagrant-1D have yet to converge after 3000 epochs suggesting they may achieve even better prediction performance with further training.

GEOM-Drugs Training and Evaluation Procedure. We train Vagrant for 100 epochs on the GEOM-Drugs dataset with an 80/10/10 train/test/validation split. We evaluate our pretrained model at 10 epochs to provide a fair comparison to the publicly available version of EDM, which was trained on GEOM-Drugs for 13 epochs.[182] We found that additional training past 10 epochs further improved performance on KLD and we present a few molecular structures generated by the 100 epoch model in the Drug-Like Molecule Gallery. The training set contains 233,397 unique SMILES and, after filtering to include only the 30 lowest energy conformers for each SMILES, over 25M unique structural conformers. During training, we predict the electronic energy of each conformer in units of Hartee to embed energetic information into the latent space, however we do not condition on this information during generation. Both EDM and Vagrant are evaluated by generating 10K unconditional samples, followed by measuring the VUN and KLD as described previously.

Inferring the Molecular Graph from 3D Coordinates. Molecules generated by cG-SchNet and EDM are represented as a set of atomic coordinates labeled with an atom type. To evaluate them with VUN and KLD, we must first infer each molecule's graph structure including its atomic connectivity matrix and its bond types. The inferred graph structure determines both the uniqueness and novelty of each molecule compared to the training set and allows us to calculate the physicochemical properties used to evaluate KLD using RDKit.[56] We use the atomic bond distances provided in Hoogeboom et al., (2022) to infer the graph structure of a molecule from its 3D coordinates and atom labels.

Training Hyperparameters. Both Vagrant models trained on QM9 and GEOM-Drugs are built with four EGCL encoder layers and 4 Trans decoder layers. Each have a hidden dimension of 256,

latent dimension of 128, 4 attention heads per attention module, a property prediction depth of 3 and a property prediction width of 256. They are optimized with the Adam optimizer[148] with an initial learning rate of $1e^{-4}$ and a weight decay of $1e^{-16}$. The Lagrange multiplier, β, is annealed linearly during training from an initial value of $1e^{-8}$ at epoch zero to a final value of 0.5. When possible, hyperparameters were kept consistent between Vagrant and all baselines to provide the fairest possible comparison.

Additional Simulation Details

Generating 3D Coordinates from SMILES. To generate reasonable starting coordinates for the candidate molecules proposed by Vagrant and Vagrant-1D, the MMFF94 force field[200] is used to propose a set of structural conformers and the lowest energy conformer is selected as the starting coordinates for DFT validation. The MMFF94 force field calculates the total molecular energy as the sum of bond stretching, angle bending, stretch-bend, out-of-plane bending, torsional, van der Waals and electrostatic terms. Atom types are determined automatically based on their local atomic environments along with their bonded and non-bonded interaction parameters. The partial charges of each atom are calculated as a function of the formal charge on each atom and the atom type.

Initial coordinates are generated by randomly sampling atomic pairwise distances within the possible upper and lower bounds of each atom-atom pair[212] and the coordinates are optimized using gradient descent until the total energy is converged or a maximum number of iterations has been reached. We use a maximum iteration number of 200 and generate n conformers based on the number of rotatable bonds in the candidate molecule according to Table C-3.

Table C-3. Number of conformers to generate determined by the number of rotatable bonds in the sampled molecule.

# Rotatable Bonds	# Conformers
Less than 8	50
8 to 12	200
Greater than 12	300

DFT Simulations. Density functional theory simulations are run using the Guassian software package.[213] Starting coordinates are either generated directly from a model or from the MMFF94 force field. Simulations are run at the B3LYP/6-31G(2df,p) level of theory with maxcycles of 200 for both initial geometry optimization and SCF convergence. Geometry optimization on the initial coordinates is followed by a standard frequency calculation to verify the final structure is at a stationary point on the potential energy surface.

Simulation Failure Modes. There are several failure modes that can prevent a generated candidate molecule from being successfully validated with DFT. The most common failure modes are described below. We present the rates of each mode of failure for all the models in Table C-4.

- **Failure to Generate Coordinates -** The MMFF94 force field may be unable to successfully converge the predicted starting conformer structures before the maximum iteration is reached. This failure mode is exclusive to models that do not explicitly generate 3D coordinates (Vagrant/Vagrant-1D).

- **Imaginary Frequencies -** A system with one or more imaginary vibrational frequencies is not at a stationary point on the potential energy surface.
- **Oscillating Convergence -** A system may not converge to an energy minimum but rather infinitely oscillate around a final energy value. We check for oscillating SCF convergence by evaluating the sign changes of the difference between the last 10 energies calculated during the simulation.
- **Invalid Starting Structure -** Some starting coordinates may not be valid molecules based on atomic valency constraints and pairwise atomic distances.

Table C-4. Simulation failure rates by mode.

Model	Failed Coordinates	Imaginary Frequencies	Oscillating Convergence	Invalid Molecules	Total Failure Rate
Vagrant-1D	6.39%	2.49%	**0.00%**	1.25%	10.44%
cG-SchNet	**0.00%**	**0.20%**	0.40%	14.00%	14.80%
EDM	**0.00%**	1.80%	**0.00%**	12.00%	13.80%
Vagrant	3.76%	2.50%	0.14%	**0.14%**	**6.68%**

Analysis on Coherence

Filter Thresholds. Coherence filters are applied by treating any generated molecule with a coherence less than the filter threshold as invalid. Since coherence is a similarity-based metric, the filter threshold can take any value between 0 and 1. As the threshold is increased, more molecules are filtered and treated as invalid. This decreases the VUN of the model, however because coherence is related to the sample quality, the property prediction accuracy has a corresponding increase (Fig. C-4). The filter threshold can be tuned based on the desired tradeoff between the exploration of novel phase space and the property prediction accuracy of novel candidates.

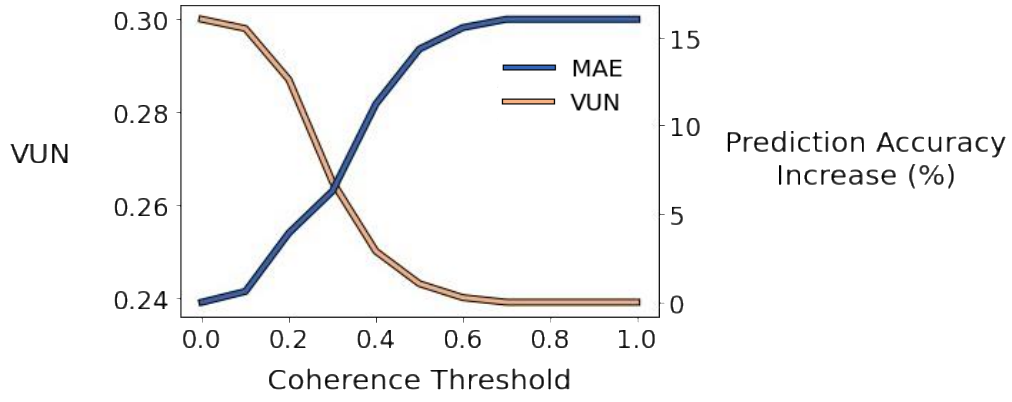


Figure C-4. VUN and MAE as a function of the coherence filter threshold. The values of both metrics converge after a threshold value of 0.6. The value at which these metrics converge will depend on the similarity metric being used and the molecules in the training set.

Latent Coherence. We visualize the coherence of the latent space by sampling 10K latent embeddings, reducing their dimensionality with PCA and coloring each point by its measured

coherence. Regions of high coherence are colored light blue and regions of high incoherence are colored light orange (Fig. C-5). We don't observe any pattern in the distribution of incoherent regions within the latent space, however we do see coherence increase during training. This suggests that coherence is a fundamental property of the latent space that can only be reduced by learning a better approximation of the true probability manifold of the training set. The relationship between molecular structure and property is better approximated in coherent regions of the latent space.

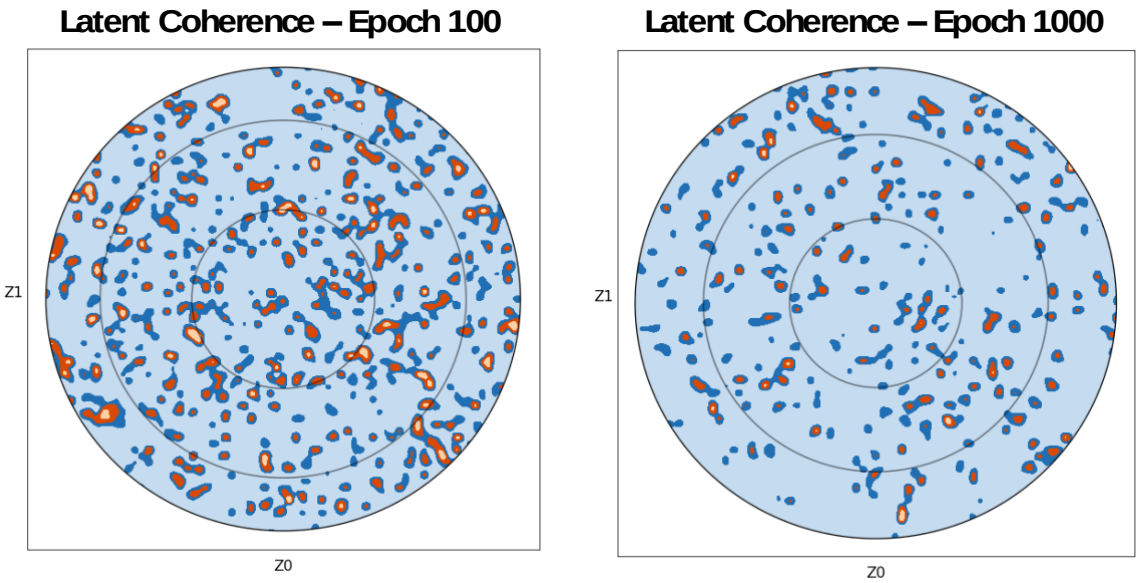


Figure C-5. Heat maps of the latent coherence of Vagrant models measured at epoch 100 and 1000.

Navigating the Latent Space Numerically

Interpolation. We test two numerical methods for navigating the latent space of Vagrant. First, we try interpolating between the latent embeddings of two distinct molecular structures - one with low polarizability and one with high polarizability (Fig. C-6). As many others have observed with similar methods, we see a smooth interpolation between the start and end molecules that corresponds to a gradient in the isotropic polarizability from low to high.

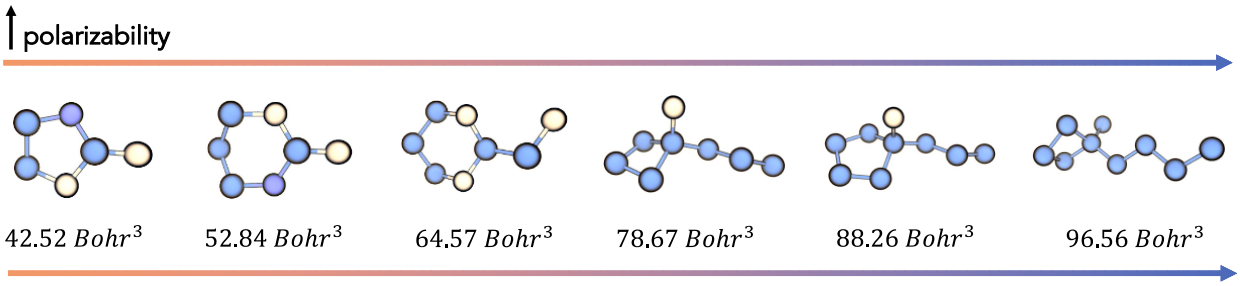


Figure C-6. Interpolating from low to high polarizability. Vagrant's latent space is smooth with respect to both molecular structure and polarizability.

Exploitation. We also try exploiting local regions of phase space that are near molecules in the training set with high values of polarizability. We take the molecule with the highest value of polarizability in the training set, 143.53 Bohr^3, and calculate its latent embedding by passing it through the encoder. We then sample in the vicinity of this embedding to find other molecules with high polarizabilities (Fig. C-7). We can sample from this region of phase space within a medium-length radius (between 1.0 and 2.5) to simultaneously exploit the high value region of polarizability while also exploring novel structures.

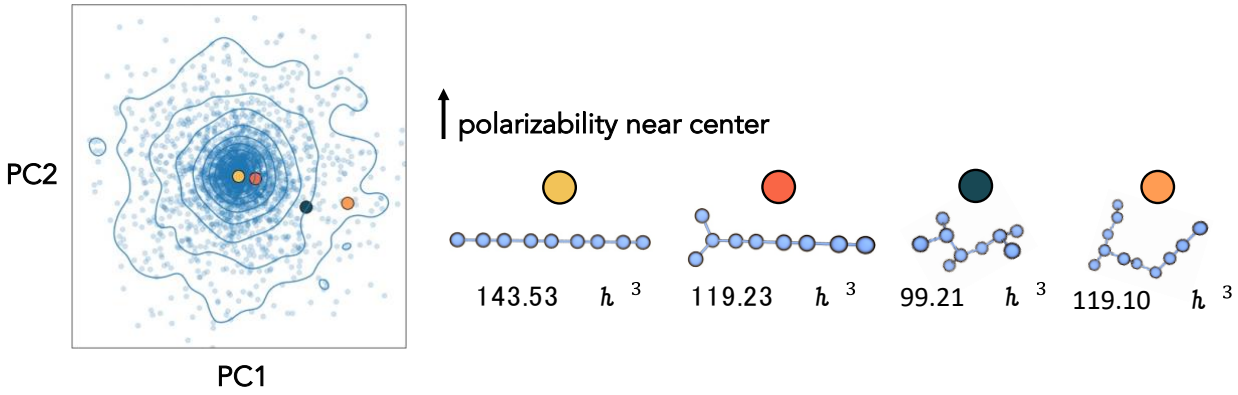


Figure C-7. Sampling at increasing distances from a high polarizability seed sample from the training set. Sampling farther from the center provides good exploration of novel structures while still remaining in a high value region of phase space. The first three structures sampled are shown in red, blue, and orange. Each has an isotropic polarizability in the 99.8th percentile of the training data or higher.

Drug-Like Molecule Gallery

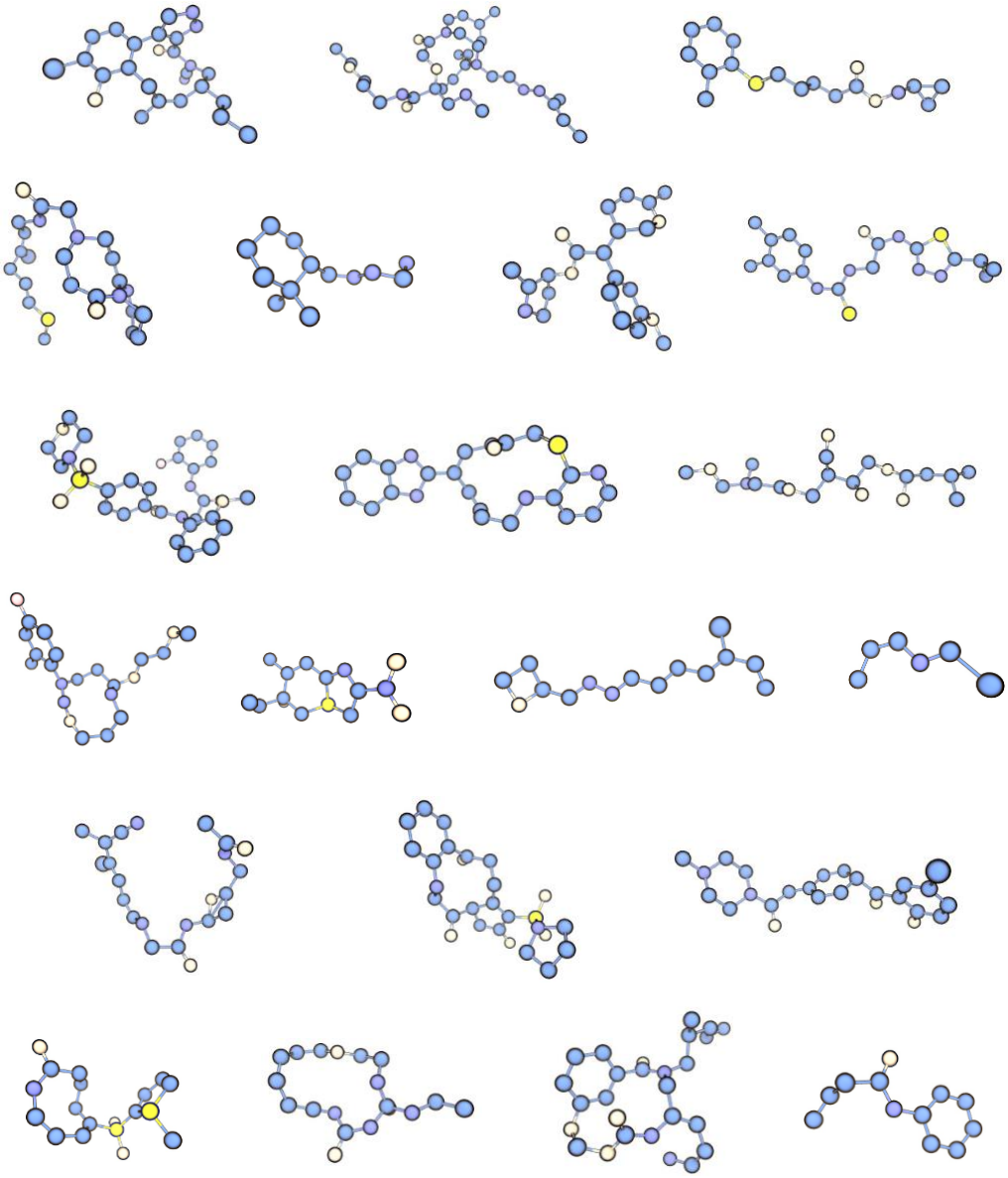


Figure C-8. Examples of drug-like molecules generated by Vagrant after being trained on GEOM-Drugs for 100 epochs.

References

(1) Schneider, P.; Schneider, G. De Novo Design at the Edge of Chaos. *J. Med. Chem.* **2016**, *59* (9), 4077–4086. https://doi.org/10.1021/acs.jmedchem.5b01849.

(2) Kingma, D. P.; Welling, M. Auto-Encoding Variational Bayes. arXiv 2013. https://doi.org/10.48550/arXiv.1312.6114.

(3) Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, *63* (11), 139–144. https://doi.org/10.1145/3422622.

(4) Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S. Deep Unsupervised Learning Using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*; PMLR, 2015; pp 2256–2265.

(5) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse Molecular Design Using Machine Learning: Generative Models for Matter Engineering. *Science* **2018**, *361* (6400), 360–365. https://doi.org/10.1126/science.aat2663.

(6) Kuhn, C.; Beratan, D. N. Inverse Strategies for Molecular Design. *J. Phys. Chem.* **1996**, *100* (25), 10595–10599. https://doi.org/10.1021/jp960518i.

(7) Moll, S.; Desmoulière, A.; Moeller, M. J.; Pache, J.-C.; Badi, L.; Arcadu, F.; Richter, H.; Satz, A.; Uhles, S.; Cavalli, A.; Drawnel, F.; Scapozza, L.; Prunotto, M. DDR1 Role in Fibrosis and Its Pharmacological Targeting. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* **2019**, *1866* (11), 118474. https://doi.org/10.1016/j.bbamcr.2019.04.004.

(8) Han, K.-C.; Yeon Kim, S.; Gyeong Yang, E. Recent Advances in Designing Substrate-Competitive Protein Kinase Inhibitors. *Current Pharmaceutical Design* **2012**, *18* (20), 2875–2882. https://doi.org/10.2174/138161212800672697.

(9) Seidel, T.; Wieder, O.; Garon, A.; Langer, T. Applications of the Pharmacophore Concept in Natural Product Inspired Drug Design. *Mol Inform* **2020**, *39* (11), 2000059. https://doi.org/10.1002/minf.202000059.

(10) Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A. Estimation of the Size of Drug-like Chemical Space Based on GDB-17 Data. *J Comput Aided Mol Des* **2013**, *27* (8), 675–679. https://doi.org/10.1007/s10822-013-9672-4.

(11) Cheng, F.; Li, W.; Zhou, Y.; Shen, J.; Wu, Z.; Liu, G.; Lee, P. W.; Tang, Y. *admetSAR: A Comprehensive Source and Free Tool for Assessment of Chemical ADMET Properties*. ACS Publications. https://doi.org/10.1021/ci300367a.

(12) Basile, A. O.; Yahi, A.; Tatonetti, N. P. Artificial Intelligence for Drug Toxicity and Safety. *Trends in Pharmacological Sciences* **2019**, *40* (9), 624–635. https://doi.org/10.1016/j.tips.2019.07.005.

(13) Vo, A. H.; Van Vleet, T. R.; Gupta, R. R.; Liguori, M. J.; Rao, M. S. An Overview of Machine Learning and Big Data for Drug Toxicity Evaluation. *Chem. Res. Toxicol.* **2020**, *33* (1), 20–37. https://doi.org/10.1021/acs.chemrestox.9b00227.

(14) Bhardwaj, G.; O'Connor, J.; Rettie, S.; Huang, Y.-H.; Ramelot, T. A.; Mulligan, V. K.; Alpkilic, G. G.; Palmer, J.; Bera, A. K.; Bick, M. J.; Di Piazza, M.; Li, X.; Hosseinzadeh, P.; Craven, T. W.; Tejero, R.; Lauko, A.; Choi, R.; Glynn, C.; Dong, L.; Griffin, R.; van Voorhis, W. C.; Rodriguez, J.; Stewart, L.; Montelione, G. T.; Craik, D.; Baker, D. Accurate de Novo Design of Membrane-Traversing Macrocycles. *Cell* **2022**, *185* (19), 3520-3532.e26. https://doi.org/10.1016/j.cell.2022.07.019.

(15) Garza, A. Z.; Park, S. B.; Kocz, R. Drug Elimination. In *StatPearls*; StatPearls Publishing: Treasure Island (FL), 2023.

(16) Arrowsmith, J.; Miller, P. Phase II and Phase III Attrition Rates 2011–2012. *Nature Reviews Drug Discovery* **2013**, *12* (8), 569–569. https://doi.org/10.1038/nrd4090.
(17) Wouters, O. J.; McKee, M.; Luyten, J. Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. *JAMA* **2020**, *323* (9), 844–853. https://doi.org/10.1001/jama.2020.1166.
(18) Oseledets, I. V. Tensor-Train Decomposition. *SIAM J. Sci. Comput.* **2011**, *33* (5), 2295–2317. https://doi.org/10.1137/090752286.
(19) Zhavoronkov, A.; Ivanenkov, Y. A.; Aliper, A.; Veselov, M. S.; Aladinskiy, V. A.; Aladinskaya, A. V.; Terentiev, V. A.; Polykovskiy, D. A.; Kuznetsov, M. D.; Asadulaev, A.; Volkov, Y.; Zholus, A.; Shayakhmetov, R. R.; Zhebrak, A.; Minaeva, L. I.; Zagribelnyy, B. A.; Lee, L. H.; Soll, R.; Madge, D.; Xing, L.; Guo, T.; Aspuru-Guzik, A. Deep Learning Enables Rapid Identification of Potent DDR1 Kinase Inhibitors. *Nat Biotechnol* **2019**, *37* (9), 1038–1040. https://doi.org/10.1038/s41587-019-0224-x.
(20) Ivanenkov, Y. A.; Polykovskiy, D.; Bezrukov, D.; Zagribelnyy, B.; Aladinskiy, V.; Kamya, P.; Aliper, A.; Ren, F.; Zhavoronkov, A. Chemistry42: An AI-Driven Platform for Molecular Design and Optimization. *J. Chem. Inf. Model.* **2023**, *63* (3), 695–701. https://doi.org/10.1021/acs.jcim.2c01191.
(21) Fuchs, F.; Worrall, D.; Fischer, V.; Welling, M. SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2020; Vol. 33, pp 1970–1981.
(22) Batzner, S.; Musaelian, A.; Sun, L.; Geiger, M.; Mailoa, J. P.; Kornbluth, M.; Molinari, N.; Smidt, T. E.; Kozinsky, B. E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials. *Nat Commun* **2022**, *13* (1), 2453. https://doi.org/10.1038/s41467-022-29939-5.
(23) Brandstetter, J.; Hesselink, R.; van der Pol, E.; Bekkers, E. J.; Welling, M. Geometric and Physical Quantities Improve E(3) Equivariant Message Passing. arXiv March 26, 2022. https://doi.org/10.48550/arXiv.2110.02905.
(24) Joshi, C. K.; Bodnar, C.; Mathis, S. V.; Cohen, T.; Liò, P. On the Expressive Power of Geometric Graph Neural Networks. arXiv January 23, 2023. https://doi.org/10.48550/arXiv.2301.09308.
(25) Garg, V.; Jegelka, S.; Jaakkola, T. Generalization and Representational Limits of Graph Neural Networks. In *Proceedings of the 37th International Conference on Machine Learning*; PMLR, 2020; pp 3419–3430.
(26) Li, H.; Wang, X.; Zhang, Z.; Zhu, W. OOD-GNN: Out-of-Distribution Generalized Graph Neural Network. *IEEE Transactions on Knowledge and Data Engineering* **2022**, 1–14. https://doi.org/10.1109/TKDE.2022.3193725.
(27) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596* (7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2.
(28) Watson, J. L.; Juergens, D.; Bennett, N. R.; Trippe, B. L.; Yim, J.; Eisenach, H. E.; Ahern, W.; Borst, A. J.; Ragotte, R. J.; Milles, L. F.; Wicky, B. I. M.; Hanikel, N.; Pellock, S. J.; Courbet, A.; Sheffler, W.; Wang, J.; Venkatesh, P.; Sappington, I.; Torres, S. V.; Lauko, A.; Bortoli, V. D.; Mathieu, E.; Barzilay, R.; Jaakkola, T. S.; DiMaio, F.; Baek, M.; Baker, D. Broadly Applicable and Accurate Protein Design by Integrating Structure Prediction Networks and Diffusion

Generative Models. bioRxiv December 14, 2022, p 2022.12.09.519842. https://doi.org/10.1101/2022.12.09.519842.
(29) Ingraham, J.; Baranov, M.; Costello, Z.; Frappier, V.; Ismail, A.; Tie, S.; Wang, W.; Xue, V.; Obermeyer, F.; Beam, A.; Grigoryan, G. Illuminating Protein Space with a Programmable Generative Model. bioRxiv December 2, 2022, p 2022.12.01.518682. https://doi.org/10.1101/2022.12.01.518682.
(30) Jing, B.; Corso, G.; Chang, J.; Barzilay, R.; Jaakkola, T. Torsional Diffusion for Molecular Conformer Generation. arXiv February 28, 2023. https://doi.org/10.48550/arXiv.2206.01729.
(31) Zhou, G.; Gao, Z.; Wei, Z.; Zheng, H.; Ke, G. Do Deep Learning Methods Really Perform Better in Molecular Conformation Generation? arXiv February 14, 2023. https://doi.org/10.48550/arXiv.2302.07061.
(32) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. arXiv February 11, 2023. https://doi.org/10.48550/arXiv.2210.01776.
(33) Yu, Y.; Lu, S.; Gao, Z.; Zheng, H.; Ke, G. Do Deep Learning Models Really Outperform Traditional Approaches in Molecular Docking? arXiv February 23, 2023. https://doi.org/10.48550/arXiv.2302.07134.
(34) Numeroso, D.; Bacciu, D. Explaining Deep Graph Networks with Molecular Counterfactuals. arXiv November 9, 2020. https://doi.org/10.48550/arXiv.2011.05134.
(35) Preuer, K.; Klambauer, G.; Rippmann, F.; Hochreiter, S.; Unterthiner, T. Interpretable Deep Learning in Drug Discovery. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., Müller, K.-R., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2019; pp 331–345. https://doi.org/10.1007/978-3-030-28954-6_18.
(36) *Central Banking, Monetary Theory and Practice: Essays in Honour of Charles Goodhart, Volume One*; Edward Elgar Publishing, 2003.
(37) Turk, J.-A.; Gendreau, P.; Drizard, N.; Gaston-Mathé, Y. A Molecular Assays Simulator to Unravel Predictors Hacking in Goal-Directed Molecular Generations. ChemRxiv June 13, 2022. https://doi.org/10.26434/chemrxiv-2022-dl347.
(38) Kim, S.; Thiessen, P. A.; Bolton, E. E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B. A.; Wang, J.; Yu, B.; Zhang, J.; Bryant, S. H. PubChem Substance and Compound Databases. *Nucleic Acids Research* **2016**, *44* (D1), D1202–D1213. https://doi.org/10.1093/nar/gkv951.
(39) Irwin, J. J.; Shoichet, B. K. ZINC − A Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model.* **2005**, *45* (1), 177–182. https://doi.org/10.1021/ci049714+.
(40) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.* **2012**, *52* (11), 2864–2875. https://doi.org/10.1021/ci300415d.
(41) Gao, W.; Coley, C. W. The Synthesizability of Molecules Proposed by Generative Models. *J. Chem. Inf. Model.* **2020**, *60* (12), 5714–5723. https://doi.org/10.1021/acs.jcim.0c00174.
(42) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Research* **2012**, *40* (D1), D1100–D1107. https://doi.org/10.1093/nar/gkr777.
(43) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Sci Data* **2014**, *1* (1), 140022. https://doi.org/10.1038/sdata.2014.22.

(44) Isert, C.; Atz, K.; Jiménez-Luna, J.; Schneider, G. QMugs, Quantum Mechanical Properties of Drug-like Molecules. *Sci Data* **2022**, *9* (1), 273. https://doi.org/10.1038/s41597-022-01390-7.
(45) Axelrod, S.; Gómez-Bombarelli, R. GEOM, Energy-Annotated Molecular Conformations for Property Prediction and Molecular Generation. *Sci Data* **2022**, *9* (1), 185. https://doi.org/10.1038/s41597-022-01288-4.
(46) Nakata, M.; Shimazaki, T.; Hashimoto, M.; Maeda, T. PubChemQC PM6: Data Sets of 221 Million Molecules with Optimized Molecular Geometries and Electronic Properties. *J. Chem. Inf. Model.* **2020**, *60* (12), 5891–5899. https://doi.org/10.1021/acs.jcim.0c00740.
(47) Smith, J. S.; Isayev, O.; Roitberg, A. E. ANI-1, A Data Set of 20 Million Calculated off-Equilibrium Conformations for Organic Molecules. *Sci Data* **2017**, *4* (1), 170193. https://doi.org/10.1038/sdata.2017.193.
(48) Mysinger, M. M.; Carchia, M.; Irwin, John. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55* (14), 6582–6594. https://doi.org/10.1021/jm300687e.
(49) Wang, R.; Fang, X.; Lu, Y.; Yang, C.-Y.; Wang, S. The PDBbind Database: Methodologies and Updates. *J. Med. Chem.* **2005**, *48* (12), 4111–4119. https://doi.org/10.1021/jm048957q.
(50) Wang, M.; Hsieh, C.-Y.; Wang, J.; Wang, D.; Weng, G.; Shen, C.; Yao, X.; Bing, Z.; Li, H.; Cao, D.; Hou, T. RELATION: A Deep Generative Model for Structure-Based De Novo Drug Design. *J. Med. Chem.* **2022**. https://doi.org/10.1021/acs.jmedchem.2c00732.
(51) Peng, X.; Luo, S.; Guan, J.; Xie, Q.; Peng, J.; Ma, J. Pocket2Mol: Efficient Molecular Sampling Based on 3D Protein Pockets. In *Proceedings of the 39th International Conference on Machine Learning*; PMLR, 2022; pp 17644–17655.
(52) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *J. Chem. Inf. Model.* **2020**, *60* (9), 4200–4215. https://doi.org/10.1021/acs.jcim.0c00411.
(53) Consonni, V.; Todeschini, R. Molecular Descriptors. In *Recent Advances in QSAR Studies: Methods and Applications*; Puzyn, T., Leszczynski, J., Cronin, M. T., Eds.; Challenges and Advances in Computational Chemistry and Physics; Springer Netherlands: Dordrecht, 2010; pp 29–102. https://doi.org/10.1007/978-1-4020-9783-6_3.
(54) Ivanciuc, O. Electrotopological State Indices. In *Molecular Drug Properties*; John Wiley & Sons, Ltd, 2007; pp 85–109. https://doi.org/10.1002/9783527621286.ch4.
(55) Beckner, W.; M. Mao, C.; Pfaendtner, J. Statistical Models Are Able to Predict Ionic Liquid Viscosity across a Wide Range of Chemical Functionalities and Experimental Conditions. *Molecular Systems Design & Engineering* **2018**, *3* (1), 253–263. https://doi.org/10.1039/C7ME00094D.
(56) Landrum, G. *RDKit: Open-source cheminformatics*. http://www.rdkit.org.
(57) Moriwaki, H.; Tian, Y.-S.; Kawashita, N.; Takagi, T. Mordred: A Molecular Descriptor Calculator. *Journal of Cheminformatics* **2018**, *10* (1), 4. https://doi.org/10.1186/s13321-018-0258-y.
(58) Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I.; Mackerell Jr., A. D. CHARMM General Force Field: A Force Field for Drug-like Molecules Compatible with the CHARMM All-Atom Additive Biological Force Fields. *Journal of Computational Chemistry* **2010**, *31* (4), 671–690. https://doi.org/10.1002/jcc.21367.
(59) Parr, R. G. Density Functional Theory of Atoms and Molecules. In *Horizons of Quantum Chemistry*; Fukui, K., Pullman, B., Eds.; Académie Internationale Des Sciences Moléculaires

Quantiques / International Academy of Quantum Molecular Science; Springer Netherlands: Dordrecht, 1980; pp 5–15. https://doi.org/10.1007/978-94-009-9027-2_2.
(60) Hachmann, J.; Olivares-Amaya, R.; Atahan-Evrenk, S.; Amador-Bedolla, C.; Sánchez-Carrera, R. S.; Gold-Parker, A.; Vogt, L.; Brockway, A. M.; Aspuru-Guzik, A. The Harvard Clean Energy Project: Large-Scale Computational Screening and Design of Organic Photovoltaics on the World Community Grid. *J. Phys. Chem. Lett.* **2011**, *2* (17), 2241–2251. https://doi.org/10.1021/jz200866s.
(61) Larsen, P.; Ins, M. von. The Rate of Growth in Scientific Publication and the Decline in Coverage Provided by Science Citation Index. *Scientometrics* **2010**, *84* (3), 575–603. https://doi.org/10.1007/s11192-010-0202-z.
(62) Hong, Z.; Tchoua, R.; Chard, K.; Foster, I. SciNER: Extracting Named Entities from Scientific Literature. In *Computational Science – ICCS 2020*; Krzhizhanovskaya, V. V., Závodszky, G., Lees, M. H., Dongarra, J. J., Sloot, P. M. A., Brissos, S., Teixeira, J., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2020; pp 308–321. https://doi.org/10.1007/978-3-030-50417-5_23.
(63) Eberts, M.; Ulges, A. Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training. June 28, 2021. https://doi.org/10.3233/FAIA200321.
(64) Jain, S.; van Zuylen, M.; Hajishirzi, H.; Beltagy, I. SciREX: A Challenge Dataset for Document-Level Information Extraction. arXiv May 1, 2020. https://doi.org/10.48550/arXiv.2005.00512.
(65) Luan, Y.; He, L.; Ostendorf, M.; Hajishirzi, H. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. arXiv August 28, 2018. https://doi.org/10.48550/arXiv.1808.09602.
(66) Mavračić, J.; Court, C. J.; Isazawa, T.; Elliott, S. R.; Cole, J. M. ChemDataExtractor 2.0: Autopopulated Ontologies for Materials Science. *J. Chem. Inf. Model.* **2021**, *61* (9), 4280–4289. https://doi.org/10.1021/acs.jcim.1c00446.
(67) Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; Nori, H.; Palangi, H.; Ribeiro, M. T.; Zhang, Y. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. arXiv March 27, 2023. https://doi.org/10.48550/arXiv.2303.12712.
(68) Zhang, Z.; Zhang, A.; Li, M.; Smola, A. Automatic Chain of Thought Prompting in Large Language Models. arXiv October 7, 2022. https://doi.org/10.48550/arXiv.2210.03493.
(69) Trivedi, H.; Balasubramanian, N.; Khot, T.; Sabharwal, A. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. arXiv December 20, 2022. https://doi.org/10.48550/arXiv.2212.10509.
(70) Johnson, J.; Douze, M.; Jégou, H. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data* **2021**, *7* (3), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572.
(71) Fowler, D. M.; Fields, S. Deep Mutational Scanning: A New Style of Protein Science. *Nat Methods* **2014**, *11* (8), 801–807. https://doi.org/10.1038/nmeth.3027.
(72) Tao, H.; Wu, T.; Aldeghi, M.; Wu, T. C.; Aspuru-Guzik, A.; Kumacheva, E. Nanoparticle Synthesis Assisted by Machine Learning. *Nat Rev Mater* **2021**, *6* (8), 701–716. https://doi.org/10.1038/s41578-021-00337-5.
(73) Epps, R. W.; Bowen, M. S.; Volk, A. A.; Abdel-Latif, K.; Han, S.; Reyes, K. G.; Amassian, A.; Abolhasani, M. Artificial Chemist: An Autonomous Quantum Dot Synthesis Bot. *Advanced Materials* **2020**, *32* (30), 2001626. https://doi.org/10.1002/adma.202001626.
(74) Häse, F.; Aldeghi, M.; Hickman, R. J.; Roch, L. M.; Aspuru-Guzik, A. Gryffin: An Algorithm for Bayesian Optimization of Categorical Variables Informed by Expert Knowledge. *Applied Physics Reviews* **2021**, *8* (3), 031406. https://doi.org/10.1063/5.0048164.

(75) Merk, D.; Friedrich, L.; Grisoni, F.; Schneider, G. De Novo Design of Bioactive Small Molecules by Artificial Intelligence. *Molecular Informatics* **2018**, *37* (1–2), 1700153. https://doi.org/10.1002/minf.201700153.
(76) Biswas, S.; Khimulya, G.; Alley, E. C.; Esvelt, K. M.; Church, G. M. Low-N Protein Engineering with Data-Efficient Deep Learning. *Nat Methods* **2021**, *18* (4), 389–396. https://doi.org/10.1038/s41592-021-01100-y.
(77) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Model.* **1988**, *28* (1), 31–36. https://doi.org/10.1021/ci00057a005.
(78) Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-Referencing Embedded Strings (SELFIES): A 100% Robust Molecular String Representation. *Mach. Learn.: Sci. Technol.* **2020**, *1* (4), 045024. https://doi.org/10.1088/2632-2153/aba947.
(79) Mercado, R.; Rastemo, T.; Lindelöf, E.; Klambauer, G.; Engkvist, O.; Chen, H.; Bjerrum, E. J. Graph Networks for Molecular Design. *Mach. Learn.: Sci. Technol.* **2021**, *2* (2), 025023. https://doi.org/10.1088/2632-2153/abcf91.
(80) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*; PMLR, 2017; pp 1263–1272.
(81) Satorras, V. G.; Hoogeboom, E.; Welling, M. E(n) Equivariant Graph Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning*; PMLR, 2021; pp 9323–9332.
(82) Gasteiger, J.; Groß, J.; Günnemann, S. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations (ICLR)*; 2020.
(83) Xu, Y.; Song, Y.; Garg, S.; Gong, L.; Shu, R.; Grover, A.; Ermon, S. Anytime Sampling for Autoregressive Models via Ordered Autoencoding. arXiv February 23, 2021. https://doi.org/10.48550/arXiv.2102.11495.
(84) Shi, Z.; Peng, S.; Xu, Y.; Liao, Y.; Shen, Y. Deep Generative Models on 3D Representations: A Survey. arXiv December 21, 2022. https://doi.org/10.48550/arXiv.2210.15663.
(85) Flam-Shepherd, D.; Zhu, K.; Aspuru-Guzik, A. Language Models Can Learn Complex Molecular Distributions. *Nat Commun* **2022**, *13* (1), 3293. https://doi.org/10.1038/s41467-022-30839-x.
(86) Nigam, A.; Pollice, R.; Aspuru-Guzik, A. JANUS: Parallel Tempered Genetic Algorithm Guided by Deep Neural Networks for Inverse Molecular Design. arXiv August 14, 2021. https://doi.org/10.48550/arXiv.2106.04011.
(87) Fu, T.; Xiao, C.; Li, X.; Glass, L. M.; Sun, J. MIMOSA: Multi-Constraint Molecule Sampling for Molecule Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence* **2021**, *35* (1), 125–133. https://doi.org/10.1609/aaai.v35i1.16085.
(88) Kullback, S.; Leibler, R. A. On Information and Sufficiency. *The Annals of Mathematical Statistics* **1951**, *22* (1), 79–86.
(89) Parsopoulos, K. E.; Vrahatis, M. N. Particle Swarm Optimization Method in Multiobjective Problems. In *Proceedings of the 2002 ACM symposium on Applied computing*; SAC '02; Association for Computing Machinery: New York, NY, USA, 2002; pp 603–607. https://doi.org/10.1145/508791.508907.
(90) Maryak, J. L.; Chin, D. C. Global Random Optimization by Simultaneous Perturbation Stochastic Approximation. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*; 2001; Vol. 2, pp 756–762 vol.2. https://doi.org/10.1109/ACC.2001.945806.
(91) Deringer, V. L.; Bartók, A. P.; Bernstein, N.; Wilkins, D. M.; Ceriotti, M.; Csányi, G. Gaussian Process Regression for Materials and Molecules. *Chem. Rev.* **2021**, *121* (16), 10073–10141. https://doi.org/10.1021/acs.chemrev.1c00022.

(92) Winter, R.; Montanari, F.; Noé, F.; Clevert, D.-A. Learning Continuous and Data-Driven Molecular Descriptors by Translating Equivalent Chemical Representations. *Chemical Science* **2019**, *10* (6), 1692–1701. https://doi.org/10.1039/C8SC04175J.
(93) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4* (2), 268–276. https://doi.org/10.1021/acscentsci.7b00572.
(94) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2017; Vol. 30.
(95) Thomas, N.; Smidt, T.; Kearnes, S.; Yang, L.; Li, L.; Kohlhoff, K.; Riley, P. Tensor Field Networks: Rotation- and Translation-Equivariant Neural Networks for 3D Point Clouds. arXiv May 18, 2018. https://doi.org/10.48550/arXiv.1802.08219.
(96) Weiler, M.; Geiger, M.; Welling, M.; Boomsma, W.; Cohen, T. S. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2018; Vol. 31.
(97) Liao, Y.-L.; Smidt, T. Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs. arXiv February 27, 2023. https://doi.org/10.48550/arXiv.2206.11990.
(98) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. SchNet – A Deep Learning Architecture for Molecules and Materials. *J. Chem. Phys.* **2018**, *148* (24), 241722. https://doi.org/10.1063/1.5019779.
(99) Antonova, R.; Rai, A.; Li, T.; Kragic, D. Bayesian Optimization in Variational Latent Spaces with Dynamic Compression. In *Proceedings of the Conference on Robot Learning*; PMLR, 2020; pp 456–465.
(100) Maus, N.; Jones, H.; Moore, J.; Kusner, M. J.; Bradshaw, J.; Gardner, J. Local Latent Space Bayesian Optimization over Structured Inputs. *Advances in Neural Information Processing Systems* **2022**, *35*, 34505–34518.
(101) Notin, P.; Hernández-Lobato, J. M.; Gal, Y. Improving Black-Box Optimization in VAE Latent Space Using Decoder Uncertainty. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2021; Vol. 34, pp 802–814.
(102) Tripp, A.; Daxberger, E.; Hernández-Lobato, J. M. Sample-Efficient Optimization in the Latent Space of Deep Generative Models via Weighted Retraining. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2020; Vol. 33, pp 11259–11272.
(103) Chiu, C.-H.; Koyama, Y.; Lai, Y.-C.; Igarashi, T.; Yue, Y. Human-in-the-Loop Differential Subspace Search in High-Dimensional Latent Space. *ACM Trans. Graph.* **2020**, *39* (4), 85:85:1-85:85:15. https://doi.org/10.1145/3386569.3392409.
(104) Li, Y.; Zhang, L.; Liu, Z. Multi-Objective de Novo Drug Design with Conditional Graph Generative Model. *J Cheminform* **2018**, *10* (1), 33. https://doi.org/10.1186/s13321-018-0287-6.
(105) Adams, K.; Coley, C. W. Equivariant Shape-Conditioned Generation of 3D Molecules for Ligand-Based Drug Design. arXiv October 6, 2022. https://doi.org/10.48550/arXiv.2210.04893.
(106) Burgess, C. P.; Higgins, I.; Pal, A.; Matthey, L.; Watters, N.; Desjardins, G.; Lerchner, A. Understanding Disentangling in β-VAE. arXiv April 10, 2018. https://doi.org/10.48550/arXiv.1804.03599.
(107) Dai, B.; Wipf, D. Diagnosing and Enhancing VAE Models. arXiv October 30, 2019. https://doi.org/10.48550/arXiv.1903.05789.

(108) Dollar, O.; Joshi, N.; C. Beck, D. A.; Pfaendtner, J. Attention-Based Generative Models for de Novo Molecular Design. *Chemical Science* **2021**, *12* (24), 8362–8372. https://doi.org/10.1039/D1SC01050F.
(109) Elton, D. C.; Boukouvalas, Z.; Fuge, M. D.; Chung, P. W. Deep Learning for Molecular Design—a Review of the State of the Art. *Mol. Syst. Des. Eng.* **2019**, *4* (4), 828–849. https://doi.org/10.1039/C9ME00039A.
(110) Beckner, W.; Ashraf, C.; Lee, J.; Beck, D. A. C.; Pfaendtner, J. Continuous Molecular Representations of Ionic Liquids. *J. Phys. Chem. B* **2020**, *124* (38), 8347–8357. https://doi.org/10.1021/acs.jpcb.0c05938.
(111) Lim, J.; Ryu, S.; Kim, J. W.; Kim, W. Y. Molecular Generative Model Based on Conditional Variational Autoencoder for de Novo Molecular Design. *Journal of Cheminformatics* **2018**, *10* (1), 31. https://doi.org/10.1186/s13321-018-0286-7.
(112) Kusner, M. J.; Paige, B.; Hernández-Lobato, J. M. Grammar Variational Autoencoder. In *Proceedings of the 34th International Conference on Machine Learning*; PMLR, 2017; pp 1945–1954.
(113) Liu, Q.; Allamanis, M.; Brockschmidt, M.; Gaunt, A. L. Constrained Graph Variational Autoencoders for Molecule Design. In *Advances in Neural Information Processing Systems*; 2018.
(114) Jin, W.; Barzilay, R.; Jaakkola, T. Junction Tree Variational Autoencoder for Molecular Graph Generation. In *Proceedings of the 35th International Conference on Machine Learning*; PMLR, 2018; pp 2323–2332.
(115) Goyal, A.; Sordoni, A.; Côté, M.-A.; Ke, N. R.; Bengio, Y. Z-Forcing: Training Stochastic Recurrent Networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2017; Vol. 30.
(116) Mohammadi, S.; O'Dowd, B.; Paulitz-Erdmann, C.; Goerlitz, L. Penalized Variational Autoencoder for Molecular Design. ChemRxiv May 7, 2021. https://doi.org/10.26434/chemrxiv.7977131.v1.
(117) Yan, C.; Wang, S.; Yang, J.; Xu, T.; Huang, J. Re-Balancing Variational Autoencoder Loss for Molecule Sequence Generation. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*; BCB '20; Association for Computing Machinery: New York, NY, USA, 2020; pp 1–7. https://doi.org/10.1145/3388440.3412458.
(118) Coley, C. W. Defining and Exploring Chemical Spaces. *Trends in Chemistry* **2021**, *3* (2), 133–145. https://doi.org/10.1016/j.trechm.2020.11.004.
(119) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; Kadurin, A.; Johansson, S.; Chen, H.; Nikolenko, S.; Aspuru-Guzik, A.; Zhavoronkov, A. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology* **2020**, *11*.
(120) Payne, J.; Srouji, M.; Yap, D. A.; Kosaraju, V. BERT Learns (and Teaches) Chemistry. arXiv July 10, 2020. https://doi.org/10.48550/arXiv.2007.16012.
(121) Schwaller, P.; Hoover, B.; Reymond, J.-L.; Strobelt, H.; Laino, T. Extraction of Organic Chemistry Grammar from Unsupervised Learning of Chemical Reactions. *Science Advances* **2021**, *7* (15), eabe4166. https://doi.org/10.1126/sciadv.abe4166.
(122) Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C. A.; Bekas, C.; Lee, A. A. Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Cent. Sci.* **2019**, *5* (9), 1572–1583. https://doi.org/10.1021/acscentsci.9b00576.
(123) Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; Amodei, D.

Language Models Are Few-Shot Learners. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2020; Vol. 33, pp 1877–1901.
(124) Huang, C.-Z. A.; Vaswani, A.; Uszkoreit, J.; Shazeer, N.; Simon, I.; Hawthorne, C.; Dai, A. M.; Hoffman, M. D.; Dinculescu, M.; Eck, D. Music Transformer. arXiv December 12, 2018. https://doi.org/10.48550/arXiv.1809.04281.
(125) Elkins, K.; Chun, J. Can GPT-3 Pass a Writer's Turing Test? *Journal of Cultural Analytics* **2020**, 17212.
(126) Floridi, L.; Chiriatti, M. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds & Machines* **2020**, *30* (4), 681–694. https://doi.org/10.1007/s11023-020-09548-1.
(127) Alemi, A. A.; Fischer, I.; Dillon, J. V.; Murphy, K. Deep Variational Information Bottleneck. arXiv October 23, 2019. https://doi.org/10.48550/arXiv.1612.00410.
(128) Tishby, N.; Pereira, F. C.; Bialek, W. The Information Bottleneck Method. arXiv April 24, 2000. https://doi.org/10.48550/arXiv.physics/0004057.
(129) Tishby, N.; Zaslavsky, N. Deep Learning and the Information Bottleneck Principle. In *2015 IEEE Information Theory Workshop (ITW)*; 2015; pp 1–5. https://doi.org/10.1109/ITW.2015.7133169.
(130) Bahuleyan, H.; Mou, L.; Vechtomova, O.; Poupart, P. Variational Attention for Sequence-to-Sequence Models. arXiv June 21, 2018. https://doi.org/10.48550/arXiv.1712.08207.
(131) Liu, D.; Liu, G. A Transformer-Based Variational Autoencoder for Sentence Generation. In *2019 International Joint Conference on Neural Networks (IJCNN)*; 2019; pp 1–7. https://doi.org/10.1109/IJCNN.2019.8852155.
(132) Lin, Z.; Winata, G. I.; Xu, P.; Liu, Z.; Fung, P. Variational Transformers for Diverse Response Generation. arXiv March 28, 2020. https://doi.org/10.48550/arXiv.2003.12738.
(133) Wang, T.; Wan, X. T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*; International Joint Conferences on Artificial Intelligence Organization: Macao, China, 2019; pp 5233–5239. https://doi.org/10.24963/ijcai.2019/727.
(134) Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. arXiv June 10, 2016. https://doi.org/10.48550/arXiv.1508.07909.
(135) Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv May 19, 2016. https://doi.org/10.48550/arXiv.1409.0473.
(136) Shannon, C. E. A Mathematical Theory of Communications. *Bell Syst. Tech. J.* **1948**, *27*, 379–423.
(137) Batty, M.; Morphet, R.; Masucci, P.; Stanilov, K. Entropy, Complexity, and Spatial Information. *J Geogr Syst* **2014**, *16* (4), 363–385. https://doi.org/10.1007/s10109-014-0202-2.
(138) Jaccard, P. Nouvelles Recherches Sur La Distribution Florale. *Bull. Soc. Vaud. Sci. Nat.* **1908**, *44*, 223–270.
(139) Bellman, R. Dynamic Programming. *Science* **1966**, *153* (3731), 34–37. https://doi.org/10.1126/science.153.3731.34.
(140) Sheldon, R. A. The Road to Biorenewables: Carbohydrates to Commodity Chemicals. *ACS Sustainable Chem. Eng.* **2018**, *6* (4), 4464–4480. https://doi.org/10.1021/acssuschemeng.8b00376.
(141) Marzorati, S.; Verotta, L.; Trasatti, S. P. Green Corrosion Inhibitors from Natural Sources and Biomass Wastes. *Molecules* **2019**, *24* (1), 48. https://doi.org/10.3390/molecules24010048.
(142) He, W.; Zhu, G.; Gao, Y.; Wu, H.; Fang, Z.; Guo, K. Green Plasticizers Derived from Epoxidized Soybean Oil for Poly (Vinyl Chloride): Continuous Synthesis and Evaluation in PVC Films. *Chemical Engineering Journal* **2020**, *380*, 122532. https://doi.org/10.1016/j.cej.2019.122532.

(143) Broadbelt, L. J.; Stark, S. M.; Klein, M. T. Computer Generated Pyrolysis Modeling: On-the-Fly Generation of Species, Reactions, and Rates. *Ind. Eng. Chem. Res.* **1994**, *33* (4), 790–799. https://doi.org/10.1021/ie00028a003.
(144) Shanks, B. H.; Keeling, P. L. Bioprivileged Molecules: Creating Value from Biomass. *Green Chem.* **2017**, *19* (14), 3177–3185. https://doi.org/10.1039/C7GC00296C.
(145) Tomczak, J.; Welling, M. VAE with a VampPrior. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*; PMLR, 2018; pp 1214–1223.
(146) Wang, D.; Tiwary, P. State Predictive Information Bottleneck. *J. Chem. Phys.* **2021**, *154* (13), 134111. https://doi.org/10.1063/5.0038198.
(147) Minsky, M. Steps toward Artificial Intelligence. *Proceedings of the IRE* **1961**, *49* (1), 8–30. https://doi.org/10.1109/JRPROC.1961.287775.
(148) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. arXiv January 29, 2017. https://doi.org/10.48550/arXiv.1412.6980.
(149) Zhang, C.; Zhang, L.; Ding, Y.; Peng, S.; Guo, X.; Zhao, Y.; He, G.; Yu, G. Progress and Prospects of Next-Generation Redox Flow Batteries. *Energy Storage Materials* **2018**, *15*, 324–350. https://doi.org/10.1016/j.ensm.2018.06.008.
(150) Meyers, J.; Fabian, B.; Brown, N. De Novo Molecular Design and Generative Models. *Drug Discovery Today* **2021**, *26* (11), 2707–2715. https://doi.org/10.1016/j.drudis.2021.05.019.
(151) Ross, J.; Belgodere, B.; Chenthamarakshan, V.; Padhi, I.; Mroueh, Y.; Das, P. Do Large Scale Molecular Language Representations Capture Important Structural Information? arXiv October 21, 2021. https://doi.org/10.48550/arXiv.2106.09553.
(152) Sun, C.; Li, W.; Xiao, J.; Parulian, N. N.; Zhai, C.; Ji, H. Fine-Grained Chemical Entity Typing with Multimodal Knowledge Representation. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*; 2021; pp 1984–1991. https://doi.org/10.1109/BIBM52615.2021.9669360.
(153) Rothchild, D.; Tamkin, A.; Yu, J.; Misra, U.; Gonzalez, J. C5T5: Controllable Generation of Organic Molecules with Transformers. arXiv August 23, 2021. https://doi.org/10.48550/arXiv.2108.10307.
(154) Zeng, Z.; Yao, Y.; Liu, Z.; Sun, M. A Deep-Learning System Bridging Molecule Structure and Biomedical Text with Comprehension Comparable to Human Professionals. *Nat Commun* **2022**, *13* (1), 862. https://doi.org/10.1038/s41467-022-28494-3.
(155) Honda, S.; Shi, S.; Ueda, H. R. SMILES Transformer: Pre-Trained Molecular Fingerprint for Low Data Drug Discovery. arXiv November 12, 2019. https://doi.org/10.48550/arXiv.1911.04738.
(156) Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. arXiv October 23, 2020. https://doi.org/10.48550/arXiv.2010.09885.
(157) Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling Laws for Neural Language Models. arXiv January 22, 2020. https://doi.org/10.48550/arXiv.2001.08361.
(158) Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D. de L.; Hendricks, L. A.; Welbl, J.; Clark, A.; Hennigan, T.; Noland, E.; Millican, K.; Driessche, G. van den; Damoc, B.; Guy, A.; Osindero, S.; Simonyan, K.; Elsen, E.; Rae, J. W.; Vinyals, O.; Sifre, L. Training Compute-Optimal Large Language Models. arXiv March 29, 2022. https://doi.org/10.48550/arXiv.2203.15556.
(159) Brown, N.; Fiscato, M.; Segler, M. H. S.; Vaucher, A. C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.* **2019**, *59* (3), 1096–1108. https://doi.org/10.1021/acs.jcim.8b00839.

(160) Gebauer, N. W. A.; Gastegger, M.; Hessmann, S. S. P.; Müller, K.-R.; Schütt, K. T. Inverse Design of 3d Molecular Structures with Conditional Generative Neural Networks. *Nat Commun* **2022**, *13* (1), 973. https://doi.org/10.1038/s41467-022-28526-y.
(161) Jin, W.; Barzilay, D. R.; Jaakkola, T. Hierarchical Generation of Molecular Graphs Using Structural Motifs. In *Proceedings of the 37th International Conference on Machine Learning*; PMLR, 2020; pp 4839–4848.
(162) Jin, W.; Yang, K.; Barzilay, R.; Jaakkola, T. Learning Multimodal Graph-to-Graph Translation for Molecular Optimization. arXiv January 28, 2019. https://doi.org/10.48550/arXiv.1812.01070.
(163) Nigam, A.; Friederich, P.; Krenn, M.; Aspuru-Guzik, A. Augmenting Genetic Algorithms with Deep Neural Networks for Exploring the Chemical Space. arXiv January 15, 2020. https://doi.org/10.48550/arXiv.1909.11655.
(164) Khemchandani, Y.; O'Hagan, S.; Samanta, S.; Swainston, N.; Roberts, T. J.; Bollegala, D.; Kell, D. B. DeepGraphMolGen, a Multi-Objective, Computational Strategy for Generating Molecules with Desirable Properties: A Graph Convolution and Reinforcement Learning Approach. *Journal of Cheminformatics* **2020**, *12* (1), 53. https://doi.org/10.1186/s13321-020-00454-3.
(165) Popova, M.; Isayev, O.; Tropsha, A. Deep Reinforcement Learning for de Novo Drug Design. *Science Advances* **2018**, *4* (7), eaap7885. https://doi.org/10.1126/sciadv.aap7885.
(166) Li, S.; Zhou, J.; Xu, T.; Dou, D.; Xiong, H. GeomGCL: Geometric Graph Contrastive Learning for Molecular Property Prediction. arXiv September 23, 2021. https://doi.org/10.48550/arXiv.2109.11730.
(167) Stumpfe, D.; Hu, H.; Bajorath, J. Advances in Exploring Activity Cliffs. *J Comput Aided Mol Des* **2020**, *34* (9), 929–942. https://doi.org/10.1007/s10822-020-00315-z.
(168) Jaegle, A.; Gimeno, F.; Brock, A.; Vinyals, O.; Zisserman, A.; Carreira, J. Perceiver: General Perception with Iterative Attention. In *Proceedings of the 38th International Conference on Machine Learning*; PMLR, 2021; pp 4651–4664.
(169) Aghajanyan, A.; Huang, B.; Ross, C.; Karpukhin, V.; Xu, H.; Goyal, N.; Okhonko, D.; Joshi, M.; Ghosh, G.; Lewis, M.; Zettlemoyer, L. CM3: A Causal Masked Multimodal Model of the Internet. arXiv January 19, 2022. https://doi.org/10.48550/arXiv.2201.07520.
(170) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742–754. https://doi.org/10.1021/ci100050t.
(171) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular De-Novo Design through Deep Reinforcement Learning. *Journal of Cheminformatics* **2017**, *9* (1), 48. https://doi.org/10.1186/s13321-017-0235-x.
(172) Prasanna, S.; Doerksen, R. J. Topological Polar Surface Area: A Useful Descriptor in 2D-QSAR. *Current Medicinal Chemistry* **2009**, *16* (1), 21–41. https://doi.org/10.2174/092986709787002817.
(173) Wildman, S. A.; Crippen, G. M. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (5), 868–873. https://doi.org/10.1021/ci990307l.
(174) Bertz, S. H. The First General Index of Molecular Complexity. *J. Am. Chem. Soc.* **1981**, *103* (12), 3599–3601. https://doi.org/10.1021/ja00402a071.
(175) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the Chemical Beauty of Drugs. *Nature Chem* **2012**, *4* (2), 90–98. https://doi.org/10.1038/nchem.1243.

(176) Ashraf, C.; Joshi, N.; Beck, D. A. C.; Pfaendtner, J. Data Science in Chemical Engineering: Applications to Molecular Science. *Annual Review of Chemical and Biomolecular Engineering* **2021**, *12* (1), 15–37. https://doi.org/10.1146/annurev-chembioeng-101220-102232.
(177) Tan, X.; Jiang, X.; He, Y.; Zhong, F.; Li, X.; Xiong, Z.; Li, Z.; Liu, X.; Cui, C.; Zhao, Q.; Xie, Y.; Yang, F.; Wu, C.; Shen, J.; Zheng, M.; Wang, Z.; Jiang, H. Automated Design and Optimization of Multitarget Schizophrenia Drug Candidates by Deep Learning. *European Journal of Medicinal Chemistry* **2020**, *204*, 112572. https://doi.org/10.1016/j.ejmech.2020.112572.
(178) Yao, Z.; Sánchez-Lengeling, B.; Bobbitt, N. S.; Bucior, B. J.; Kumar, S. G. H.; Collins, S. P.; Burns, T.; Woo, T. K.; Farha, O. K.; Snurr, R. Q.; Aspuru-Guzik, A. Inverse Design of Nanoporous Crystalline Reticular Materials with Deep Generative Models. *Nat Mach Intell* **2021**, *3* (1), 76–86. https://doi.org/10.1038/s42256-020-00271-1.
(179) Keriven, N.; Peyré, G. Universal Invariant and Equivariant Graph Neural Networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2019; Vol. 32.
(180) Wilkins, D. M.; Grisafi, A.; Yang, Y.; Lao, K. U.; DiStasio, R. A.; Ceriotti, M. Accurate Molecular Polarizabilities with Coupled Cluster Theory and Machine Learning. *Proceedings of the National Academy of Sciences* **2019**, *116* (9), 3401–3406. https://doi.org/10.1073/pnas.1816132116.
(181) Mann, A. Chapter 17 - Conformational Restriction and/or Steric Hindrance in Medicinal Chemistry. In *The Practice of Medicinal Chemistry (Third Edition)*; Wermuth, C. G., Ed.; Academic Press: New York, 2008; pp 363–379. https://doi.org/10.1016/B978-0-12-374194-3.00017-2.
(182) Hoogeboom, E.; Satorras, V. G.; Vignac, C.; Welling, M. Equivariant Diffusion for Molecule Generation in 3D. In *Proceedings of the 39th International Conference on Machine Learning*; PMLR, 2022; pp 8867–8887.
(183) Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2020; Vol. 33, pp 6840–6851.
(184) Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Shao, Y.; Zhang, W.; Cui, B.; Yang, M.-H. Diffusion Models: A Comprehensive Survey of Methods and Applications. arXiv October 23, 2022. https://doi.org/10.48550/arXiv.2209.00796.
(185) Cao, H.; Tan, C.; Gao, Z.; Chen, G.; Heng, P.-A.; Li, S. Z. A Survey on Generative Diffusion Model. arXiv December 13, 2022. https://doi.org/10.48550/arXiv.2209.02646.
(186) O'Boyle, N.; Dalke, A. DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. ChemRxiv September 27, 2018. https://doi.org/10.26434/chemrxiv.7097960.v1.
(187) Li, Y.; Pei, J.; Lai, L. Structure-Based de Novo Drug Design Using 3D Deep Generative Models. *Chemical Science* **2021**, *12* (41), 13664–13675. https://doi.org/10.1039/D1SC04444C.
(188) Luo, S.; Guan, J.; Ma, J.; Peng, J. A 3D Generative Model for Structure-Based Drug Design. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2021; Vol. 34, pp 6229–6239.
(189) Schneuing, A.; Du, Y.; Harris, C.; Jamasb, A.; Igashov, I.; Du, W.; Blundell, T.; Lió, P.; Gomes, C.; Welling, M.; Bronstein, M.; Correia, B. Structure-Based Drug Design with Equivariant Diffusion Models. arXiv October 24, 2022. https://doi.org/10.48550/arXiv.2210.13695.
(190) Mitton, J.; Senn, H. M.; Wynne, K.; Murray-Smith, R. A Graph VAE and Graph Transformer Approach to Generating Molecular Graphs. arXiv April 9, 2021. https://doi.org/10.48550/arXiv.2104.04345.
(191) Huang, Y.; Peng, X.; Ma, J.; Zhang, M. 3DLinker: An E(3) Equivariant Variational Autoencoder for Molecular Linker Design. arXiv May 15, 2022. https://doi.org/10.48550/arXiv.2205.07309.

(192) Rajan, K.; Zielesny, A.; Steinbeck, C. STOUT: SMILES to IUPAC Names Using Neural Machine Translation. *Journal of Cheminformatics* **2021**, *13* (1), 34. https://doi.org/10.1186/s13321-021-00512-4.
(193) Skalic, M.; Jiménez, J.; Sabbadin, D.; De Fabritiis, G. Shape-Based Generative Modeling for de Novo Drug Design. *J. Chem. Inf. Model.* **2019**, *59* (3), 1205–1214. https://doi.org/10.1021/acs.jcim.8b00706.
(194) Xu, M.; Ran, T.; Chen, H. De Novo Molecule Design Through the Molecular Generative Model Conditioned by 3D Information of Protein Binding Sites. *J. Chem. Inf. Model.* **2021**, *61* (7), 3240–3254. https://doi.org/10.1021/acs.jcim.0c01494.
(195) Imrie, F.; Bradley, A. R.; van der Schaar, M.; Deane, C. M. Deep Generative Models for 3D Linker Design. *J. Chem. Inf. Model.* **2020**, *60* (4), 1983–1995. https://doi.org/10.1021/acs.jcim.9b01120.
(196) Cornia, M.; Stefanini, M.; Baraldi, L.; Cucchiara, R. Meshed-Memory Transformer for Image Captioning; 2020; pp 10578–10587.
(197) Popel, M.; Tomkova, M.; Tomek, J.; Kaiser, Ł.; Uszkoreit, J.; Bojar, O.; Žabokrtský, Z. Transforming Machine Translation: A Deep Learning System Reaches News Translation Quality Comparable to Human Professionals. *Nat Commun* **2020**, *11* (1), 4381. https://doi.org/10.1038/s41467-020-18073-9.
(198) Liu, W.; Chen, S.; Guo, L.; Zhu, X.; Liu, J. CPTR: Full Transformer Network for Image Captioning. arXiv January 27, 2021. https://doi.org/10.48550/arXiv.2101.10804.
(199) Zeiler, M. D.; Krishnan, D.; Taylor, G. W.; Fergus, R. Deconvolutional Networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; 2010; pp 2528–2535. https://doi.org/10.1109/CVPR.2010.5539957.
(200) Tosco, P.; Stiefl, N.; Landrum, G. Bringing the MMFF Force Field to the RDKit: Implementation and Validation. *J Cheminform* **2014**, *6* (1), 37. https://doi.org/10.1186/s13321-014-0037-3.
(201) Bajusz, D.; Rácz, A.; Héberger, K. Why Is Tanimoto Index an Appropriate Choice for Fingerprint-Based Similarity Calculations? *Journal of Cheminformatics* **2015**, *7* (1), 20. https://doi.org/10.1186/s13321-015-0069-3.
(202) Martin, D.; Sild, S.; Maran, U.; Karelson, M. QSPR Modeling of the Polarizability of Polyaromatic Hydrocarbons and Fullerenes. *J. Phys. Chem. C* **2008**, *112* (13), 4785–4790. https://doi.org/10.1021/jp7100368.
(203) Kawczak, P.; Bober, L.; Bączek, T. QSPR Analysis of Some Agonists and Antagonists of α-Adrenergic Receptors. *Med Chem Res* **2015**, *24* (1), 372–382. https://doi.org/10.1007/s00044-014-1130-x.
(204) Black, S.; Biderman, S.; Hallahan, E.; Anthony, Q.; Gao, L.; Golding, L.; He, H.; Leahy, C.; McDonell, K.; Phang, J.; Pieler, M.; Prashanth, U. S.; Purohit, S.; Reynolds, L.; Tow, J.; Wang, B.; Weinbach, S. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. arXiv April 14, 2022. https://doi.org/10.48550/arXiv.2204.06745.
(205) Shoeybi, M.; Patwary, M.; Puri, R.; LeGresley, P.; Casper, J.; Catanzaro, B. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. arXiv March 13, 2020. https://doi.org/10.48550/arXiv.1909.08053.
(206) Rasley, J.; Rajbhandari, S.; Ruwase, O.; He, Y. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; KDD '20; Association for Computing Machinery: New York, NY, USA, 2020; pp 3505–3506. https://doi.org/10.1145/3394486.3406703.

(207) Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv August 8, 2022. https://doi.org/10.48550/arXiv.2104.09864.
(208) Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models Are Unsupervised Multitask Learners. *OpenAI blog* **2019**, No. 1(8):9.
(209) Yoshikawa, N.; Terayama, K.; Sumita, M.; Homma, T.; Oono, K.; Tsuda, K. Population-Based De Novo Molecule Generation, Using Grammatical Evolution. *Chem. Lett.* **2018**, *47* (11), 1431–1434. https://doi.org/10.1246/cl.180665.
(210) H. Jensen, J. A Graph-Based Genetic Algorithm and Generative Model/Monte Carlo Tree Search for the Exploration of Chemical Space. *Chemical Science* **2019**, *10* (12), 3567–3572. https://doi.org/10.1039/C8SC05372C.
(211) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv May 24, 2019. https://doi.org/10.48550/arXiv.1810.04805.
(212) Ebejer, J.-P.; Morris, G. M.; Deane, C. M. Freely Available Conformer Generation Methods: How Good Are They? *J. Chem. Inf. Model.* **2012**, *52* (5), 1146–1158. https://doi.org/10.1021/ci2004658.
(213) Frisch, M.; Trucks, G.; Schlegel, H.; Scuseria, G.; Robb, M.; Cheeseman, J.; Scalmani, G.; Barone, V.; Petersson, G.; Nakatsuji, H. Gaussian 16 Revision c. 01, 2016.