Exploring Machine Learning Applications to Enable Next-Generation Chemistry

Exploring Machine Learning Applications to Enable Next-Generation Chemistry

A DISSERTATION PRESENTED

BY

JENNIFER NANSEAN WEI

TO

THE DEPARTMENT OF DEPARTMENT OF CHEMISTRY AND CHEMICAL BIOLOGY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE SUBJECT OF

CHEMICAL PHYSICS

HARVARD UNIVERSITY

CAMBRIDGE, MASSACHUSETTS

NOVEMBER 2018

Dissertation Advisor: Professor Alán Aspuru-Guzik Jennifer Nansean Wei

Exploring Machine Learning Applications to Enable Next-Generation Chemistry

ABSTRACT

As global demand for energy and materials grow while our dependence on petroleum and fossil fuels declines, it is necessary to revolutionize the way we make new materials. Machine learning provides several avenues for accelerating the discovery pipeline. Models employing machine learning optimization have already begun to accelerate materials discovery by identifying new candidates for organic LEDs, and predicting simple synthetic routes for organic molecules. Furthermore, researchers have used machine learning models to perform complicated tasks which were previously thought to be only possible by humans; such models can be leveraged to propose new molecular candidates.

In my PhD work, I have developed machine learning models for three different challenges in chemistry. 1) I developed molecular autoencoders to decode molecular space from an order of 10^{60} to a 200-dimensional vector. In this vector representation, I demonstrate how we can use gradient descent and other optimization techniques to explore this space and find molecules that optimize target properties. 2) I built neural network models for predicting reactions within selected families of molecules, helping us to characterize the reactivity of a molecule. 3) I also developed a model which can predict electron-ionization mass spectra for small molecules in milliseconds, making it possible to expand the coverage of mass spectral libraries and what compounds can be identified with mass spectrometry.

Together, these machine learning models represent a portion of how machine learning can be used to propose new molecules and to accelerate the identification of new molecules. As the field of machine learning develops, there will be many other possible applications to help accelerate the materials discovery platform.

Contents

List of Figures

For my parents.

Acknowledgments

In my PhD I have had the privilege of working at the confluence of machine learning and chemistry. I have several people to thank for introducing me into this domain.

First of all, I thank my advisor, Prof. Alan Aspuru-Guzik. His infectious energy and enthusiasm, as well as his creative drive, has always spurred me on to pursue ambitious and exciting projects. When I first started graduate school, I could not have imageind that I would end up working with machine learning models, or that my thesis would be comprised entirely of this work. It is a testament to his forethought and willingness to let students to explore new fields that my PhD took the course that it did.

I thank my committee members. As my the focus of my PhD switched from applying conceptual DFT methods for predicting reactions to machine learning applications, my committee has also gone through transitions. I'd like to thank Prof. Roy Gordon, Prof. Finale Doshi-Valez, and Prof. Alexander Rush for being on my committee at various times during my PhD. Special thanks goes to Prof. Efthimios Kaxiras, who has been on my committee throughout my PhD. I have really appreciated all of the advice and guidance over the years.

I would like to thank Prof. David Duvenaud, who patiently mentored me on my very first project in applying machine learning to chemistry. He gave me a lot of feedback on how to design the experiments, how to think of good metrics for measuring the results.

I also would like to thank my colleagues in the Aspuru-Guzik lab. There aren't many groups out there where any group member is more than willing to give you an hour of their time to help you brainstorm an idea, to teach you a new concept, or to listen to your latest frustrations with the publishing process or with the computer cluster. There are so many people to name that I will not endeavor to name everyone except for a few: Prof. Rafael Gómez-Bombarelli, thank you for being one of my first mentors in the group, and teaching me how think critically on my earliest projects. Benjamín Sánchez-Lengeling, thank you for being a great sounding board for machine learning ideas and always taking the time to help me figure out that hack in an Jupyter notebook or in RdKit. Joey Knightbrook, Thomas Markovich, and Sam Blau, thanks for taking me under your wing as a fledgling graduate student, and for the career path afterwards. Florian Häse, Löic Roch, Daniel Tabor, thank you for the chocolate, pool, and advice as I wrapped up this PhD.

Thank you to all of my mentors and people I've reached out to for advice throughout graduate school. Amy Gilson, thank you for being one of my first mentors, and teaching me the value of reaching out and emailing people to get the help you need. I am glad that we stayed in touch and became friends even after

we first met through HWIC.

Thank you to the Google Brain Cambridge team for an awesome internship. Thank you to everyone for taking the time to help me learn to write better code, and be a better machine learning researcher. I look forward to joining the team and working with you all on new and exciting applications of machine learning to chemistry.

I thank my friends in my cohort. I enjoyed our campaigns through Pandemic, Gloomhaven, and the many, many games of Terraforming Mars. I thank my friends outside of the Harvard Chemistry Department. There is nothing like a quick chat with someone outside the department to help put my problems into some perspective. Deborah Hanus and Ashley Villar, thank you for bringing me into your reading group, and helping me develop confidence in reading machine learning papers. Deborah, thank you for convincing me to sign up and attend WiML 2016. I really think that that workshop helped me feel like I could fit into the machine learning community. I've attended both NeurIPS and WiML every year since then. Harvard Dragonboat, thanks for being such a great community of paddlers and support network. I always looked forward to the summers where I could paddle along the river with everyone again.

Thank you Jeep. You have brought so me much joy, laughter, compassion, hugs, validation, and more through these trying years of graduate school, and for that, I am deeply grateful.

Finally, and most importantly, I thank my parents. Thank you for fostering my interest in science at a young age. Thank you for teaching me how to be a thoughtful, and critically thinking human being. And thank you for your everlasting support, through high school, through college, to today, to make this PhD possible.

Further Acknowledgements by Chapter

CHAPTER 3 contains content that has been previously published in the following article: Jennifer N. Wei, David Duvenaud, and Alán Aspuru-Guzik. "Neural networks for the prediction of organic chemistry reactions." ACS Cent. Sci. **2** (2016): 725.
DOI: 10.1021/acscentsci.6b00219

CHAPTER 4 contains content that has been published in the following article: Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, Alán Aspuru-Guzik. "Automatic chemical design using a data-driven continuous representation of molecules." ACS Cent. Sci. **4** (2018): 268.
DOI: 10.1021/acscentsci.7b00572

CHAPTER 5 contains contributions from David Belanger, Ryan P. Adams, and D. Sculley.

1

Introduction

Chemistry has given many blessings to mankind, providing drug molecules, plastics and other materials[86]. However, the golden age of chemistry, where new promising molecules can easily be found from natural products, or extracted from petroleum compounds is coming to an end. We have picked all the low hanging fruit of easy-to-find, easy to synthesize compounds, and for a sustainable, carbon neutral future, we must quell our dependence on petroleum.

In order to discover new molecules in the 21st century, we must rapidly increase our throughput for molecular discovery. Machine learning can help us arrive at this destination. Models using machine learning techniques have already been used to optimize tasks such as playing go and teaching robots to walk[109,151]. Machine learning models are even beginning to tackle creative tasks such as producing art and music. It is not only possible, but critical for machine learning models to be developed towards discovering new molecules, and optimizing new reactions.

There are a few issues facing the application of machine learning to chemistry. The first issue is that while there are many of sources of data (from hundreds of years of experiments), there is no central repository for accessing all of the data. Many of these records are not available to the public. Chemistry publications reporting new results and new reactions, typically only report the most successful molecule or reaction. The results do not include negative results. More complete datasets exist at pharmaceutical companies or in other industries, however, these datasets are treated as proprietary. There are a few public datasets of molecules and reactions available, as well as other datasets that are available for sale. These datasets enable us to do some machine learning but come with some limitations. I will discuss some of these datsets in Section 3.1.

The second issue is that the challenges we wish to address in chemistry are not easy to formulate. Often, decisions in chemistry involve trade offs. Making molecules with a high reduction potential for use in flow battery may also cause them to be more susceptible to degradation reactions[162]. Choosing to improve the yield of a synthesis product by using fewer reaction steps will likely necessitate starting with more complicated materials, or more toxic reagents. Conversely, choosing to use cheaper starting materials, you

might have a long synthesis path, requiring multiple purification steps. The choice of how to how to prioritize these factors is subjective. The optimization should depend on the particular application, as well as the experiences of the chemists involved.

The third issue is the challenge of molecular representability. When a human chemist sees a molecule, they infer a lot of things about the molecule, including the electronegativity of regions of the molecule, steric hindrance, etc. The maxim taught in organic chemistry is "structure dictates function." Because the molecule is a graph structure, it is difficult to encapsulate this information into a single vector for a molecular input.

In my doctoral research, I developed machine learning models for three applications in chemistry. I worked with the representations that currently exist to predict reactions and mass spectrometry. I have used machine learning models to compress the molecular space into a new representation for easier optimization and new lead selection.

This thesis is divided into two main parts. In the first part, I provide some background for planning machine learning projects in chemistry. This section covers:

- A heavily abbreviated introduction to machine learning, focusing in particular on architectures employed in Section II
- A discussion of how to design and set up a machine learning project for chemistry applications. In particular, I will discuss dataset selection as well as molecule representations commonly used in machine learning algorithms.

The remainder of this thesis is divided into three different parts, each describing a different application for machine learning models in chemistry:

- Using reversible, data-driven representations to encode molecules and use these representations to drive optimization for particular properties
- Predicting organic chemistry reactions using neural networks and molecular fingerprints
- Predicting mass spectrometry spectra for small molecules using neural networks and fingerprint representations.

I will conclude with an outlook on future directions for machine learning applications to challenges in chemistry, and discuss some avenues for improving their efficiency in the future.

Part I

Part I: Machine Learning and Cheminformatics Background

This page intentionally left blank.

2

A brief introduction to machine learning

ABSTRACT

Machine learning is the practice of identifying useful patterns from a dataset which then can be transformed into a model. This model can then be used to make predictions given similar types of inputs, or to generate outputs that are similar to the inputs that were given to the model. Models that are used to make predictions are known as discriminative models, whereas models that are used to generate output that is similar to the original dataset is known as a generative models. I will discuss some applications of machine learning models in Section II. Here, I will outline some different machine learning model architectures used in the later sections.

2.1 Types of Machine Learning Models

2.1.1 Linear Regression

One of the most basic discriminative models is a linear regression model, a model all scientists become fluent in at a very early stage of their scientific development. Such models take in vector representations of the data input, and apply a linear transformation to this representation. The values in the matrix of the linear transformation are known as the parameters of the model, shown as A in Eq. (2.1):

$$y = Ax + b \tag{2.1}$$

To measure the predictive accuracy of the model, it is necessary to set an objective function. In the case of a regression task, the objective is a function of the difference between the target value and the value predicted by the model. An example of such a function is the mean squared error. Then, one can adjust the parameters in order to improve the accuracy.

For a linear regression model, it is possible to determine analytically the best parameters for fitting the input data. The objective function for a linear regression model can be written explicitly as a function of the parameters of the model and the input data points. As a result, it is possible to find the gradient of the objective function with respect to the parameters, and solve for the values of the parameters when the gradient is 0. Because this objective function is convex in terms of the parameters, there exists only one set parameters which will provide the smallest prediction error. An explicit calculation for the weights is shown in Section 5.1.4 of

3

A brief introduction to cheminformatics and molecular represenations

ABSTRACT

There are several important factors to consider when designing a machine learning project in chemistry. The first is to consider the contents of the dataset to use. The second is to consider which representation to use when representing the molecules or inputs to the machine learning model. The third is to consider the objective of the model. Is the goal of the model to predict certain properties or results? Or is it to generate a new molecule, or perhaps a new synthetic route?

I will discuss the first two points regarding datasets and representation in this introduction. The third question is a rather open ended question and will depend on the end goal of the users. The second half of this dissertation explores three different applications with different objectives for each project.

3.1 Considerations for Dataset Selection

Despite the vast amount of literature published in chemistry, there are surprisingly only a limited number of datasets available to the public. There are many considerations that must be made for selecting a dataset, and in many cases cleaning a dataset for use in machine learning. I will discuss some considerations when selecting and using a new dataset of molecules.

First, it is important to consider what kind of data is contained in the dataset. Are values predicted values, or values from an electronic structure calculation or another kind of calculation. Models trained on datasets containing calculated values alone can only be used to predict other calculated properties. If one desires to use the model to predict a measured property, then it is necessary to either use a transfer learning approach, or otherwise calibrate the measured properties to calculated data.

Second, one must consider the size of the dataset. Many datasets which contain measured properties typically only contain properties for a few hundred molecules at best. On such a small dataset, it becomes easy to overfit the model. While it would be possible to develop a model for predicting this dataset, it may not be very generalizable to other datasets.

A related, important factor to consider is the diversity of molecules in the dataset. Are there only a few families of molecules represented? Are some families of molecules more heavily represented than others? Machine learning models are excellent interpolation models, but cannot extrapolate to examples it has not seen before. In other words, a model that has been trained on only one or two families of molecules may would be excellent at predicting properties for molecules in these families, but likely unable to predict properties for molecules outside of this family.

A few large datasets of molecules exist, some with some predicted properties. A more extensive list is provided in these reviews. Some of the datasets used in these thesis include:

- **QM9** : A dataset for molecules with fewer than 9 heavy (non-hydrogen) atoms, containing calculated properties
- **ZINC** : A dataset of drug molecules. The entire dataset contains millions of molecules, but these can be filtered down to a smaller group.
- **NIST 17 Mass Spectral Library**: This collection of data is not available to the public, but is accessible to researchers whose institutions have access to mass spectral software.

Once a dataset has been settled upon, typically it is necessary to clean the dataset. Cleaning the dataset can take on a whole range of tasks. A typical workflow to prepare a new dataset is as follows: Write a parser to convert the molecule data file into a format readable for machine learning tasks. Using database visualization tools, examine the dataset for any errors. In particular, pay attention to any outliers in the dataset, or any values that are unphysical (e.g. a molecule containing a mass spec intensity peak at an m/z ratio that is much larger than the mass of the molecule itself). Create a linear regression model or a single layer model for predicting the output to ensure that everything runs correctly.

Inevitably, for each dataset which contains new properties, it will be necessary to add a few extra data parsing steps, or add a few more sanity checks into the data processing pipeline. Visualizing the data and the predictions as often as possible is the key to identifying potential errors in the dataset.

3.2 Molecule Descriptors

Machine learning models require vectorized information as inputs to the model. The matrix multiplication operations which underpin the layers of a machine learning model will transform these input vectors into new features or predictions.

For many problems in machine learning, such as vision recognition, vectorizing inputs is straightforward. Images are composed of pixels, with an RGB value or greyscale associated with each pixel, and so these data inputs are already in vectorized form. Converting molecules into a vector representation is not simple.

One can think of a molecule as a graph, with the atoms as nodes and the bonds between the atoms as edges. What is the best way to vectorize information from a graph? This depends on what one wants to learn from the model. For example, if one simply wanted to 'learn' the molecular

weight given a molecule, then a vector representation that includes all the atoms is sufficient.

If one wanted to optimize for something more complicated, such as which molecule might be suitable for flow battery applications, then some of the properties one would want the model to predict include: the reduction potential, the susceptibility of the molecule to reactions with water and other molecules in the environment[162].

To understand what features we'd like to incorporate into our vector representation of a molecule in order to predict these properties, let's consider what properties a human organic chemist would observe in the molecule. Looking at the molecule, a chemist might have an idea of these properties by examining:

The number of rings contained in a molecule, or the overall stability of the molecule The types of subgroups that are bound to the ring. In particular, are these fragments electron withdrawing or electron donating? How many such groups are bound to the molecule? Which positions are electron donating and accepting? How accessible are these positions to other molecules.

All of these aspects of a molecule cannot be captured by simple atom-level descriptors. Instead it is necessary for descriptors to consider neighborhoods around atoms.

I will now describe some existing molecule representation methods.

3.2.1 Descriptors from Cheminformatics

Since the 1960s, chemists have leveraged computable information on molecules towards predicting properties in drug discovery. This discipline is known as cheminformatics; numerous books and reviews have been written about this topic[21,43,44,174].

The key goal of cheminformatics is to identify quantitative structure property relationships (QSAR); these relationships are based on the notion that the functionality of a molecule for any particular application is defined by its structure. As such, by analyzing the structure, either by using quantum mechanics modeling techniques, or some other data-driven methods, it is possible

to determine the molecule's usefulness for the application. In turn, it is possible to use these patterns to identify promising candidates for a given application.

The earliest machine-readable representations of molecules were developed from this field. Most of these descriptors are in the form of some sort of string descriptor. In present research, the most popular string representations include the SMILES (Simplified Molecular Input Line Entry Specification) and the INCHI (IUPAC International Chemical Identifier) representations[175]. Both SMILES strings and INCHI representation encode not only the atoms present in the molecule, but also the connections between the atoms in the molecule. Figure 3.1 has an example of a molecule encoded in SMILES representation, the different colors in the string representing different parts in the molecule. The INCHI representation also contains additional information about stereochemistry. A hashed version of the INCHI, known as the INCHIKEY is a common molecule identifier. The SMILES representation is used heavily in Chapter 4 of this work.

Various descriptors can be derived based on these raw representations; in fact, there are several software packages which provide hundreds of these descriptors for users[98,125]. These properties describe the partial charge of the molecule, the solvent accessible area, etc. Many of these properties can be calculated using heuristic models, e.g. Gasteiger charges. These descriptors can in turn be used to model more complicated properties that are more pertinent to the problem that one wishes to optimize[98].

A more abstract molecule descriptor is the family of fingerprint descriptors. Fingerprints capture local information about the molecule's structure and encode this information in a vector. The type of local information varies for each fingerprint.

Morgan fingerprints/Extended Circular Fingerprints[104,129] capture the local environment around each atom, as shown in Figure3.1. This information records all local neighborhoods of molecule fragments which contain fewer than the maximum bond radius to consider. In order to generate a

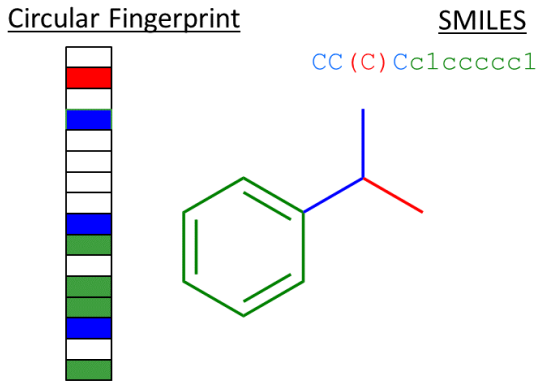


Figure 3.1: SMILES and Circular Fingerprints. Color coding is added as a visual aid, colored portions of the representation correspond to different parts of the representation. In the SMILES representation, a string is used to represent the molecular graph. Branching regions are encoded in parenthesis (red) while cycles are encoded by starting and ending the cyclical region with a number (green). Lower case letters are used to represent atoms which are in an aromatic ring (green). The Circular fingerprint is a binary vector representation of a molecule. Molecular subgraphs centered around different atoms are recorded as bits in the fingerprint.

fixed length vector representation, the information about the local structure is hashed into a vector. Chapters 3 and 5 of this thesis feature applications of this fingerprint.

Other fingerprints vary in their focus on known functional groups within the molecule (e.g. MACCS Fingerprint)[32] or their focus on three dimensional structure (e.g. Topological fingerprints)[107]. The precise fingerprint, or combination of fingerprints to use often depends on the application at hand. It is often useful to test multiple fingerprints, or even combinations of fingerprints, to determine which might be the best for representing molecules for the given application[76].

3.2.2 Chemical Graph theory

The subfield of chemical graph theory considers graph-theory based approaches to molecule representations[122]. These representations typically describe the whole molecule graph, rather than

describing local fragments or representing some estimated physical properties.

Some of the more common descriptors from this field include the adjacency matrix and the Coulomb matrix[136]. Other descriptors include the eigenvalues of the adjacency matrix, path indices, and topological indices. Path indices describe the longest path between two carbon atoms present in a molecule[122]. Topological indices depend on the valences of the constituent atoms in a bond[122].

Descriptors from this field are used less often, they do not enjoy the same level of support in open source packages as the cheminformatic descriptors. Descriptors from this field have been used to predict a wide variety of properties, including the boiling points for alkane, amino alkenes, predicting antihypertensive activity, and NMR shifts[122].

3.2.3 Descriptors developed with machine learning

In the last two years, researchers have now applied machine learning to propose new molecular representations. Some of these representations extend the idea of the fingerprint representation as a two dimensional method for predicting molecules. Known as graph convolutional networks[34,47,75], these representations collect information from smaller subgraphs of the molecular graphs, and aggregate this information into a single vector. The final output from these models is a continuous vector representation; this enables the parameters of the model itself to be tuned to predict a particular property.

New representations have also been developed for three dimensional molecule structures. Starting with the three dimensional geometries allows for better prediction of some properties such as energies. In addition to graph convolutional models, wave transform models have been proposed for modeling the electron densities[83]. Wavelet scattering representations of the electron density have also been used as a rotationally invariant representation of a molecule[36].

3.3 Afterword

All of these descriptors have demonstrated their ability to predict energies/calculated properties for molecules. It is likely that using transfer learning techniques, it would be possible to use these models to model smaller experimental datasets.

However many of these machine learned representations proceeded the work in this thesis, or did not have an easily accessible implementation to use. As such, the work that is presented in the third part of this thesis uses two descriptors from cheminformatics: Morgan fingerprints and SMILES Representation.

Part II

Part II: Machine Learning Applications to Chemistry

This page intentionally left blank.

4

Variational Autoencoders for Optimization in Molecular Space

Apart from minor modifications, this chapter originally appeared as:

"Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules". GÃşmez-Bombarelli, R., Wei, J.N., Duvenaud, D., HernÃąndez-Lobato, J.M., SÃąnchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., Aspuru-Guzi, AlÃąn. ACS Cent. Sci.. 4 (2018): 268.

ABSTRACT

We report a method to convert discrete representations of molecules to and from a multidimensional continuous representation. This model allows us to generate new molecules for efficient exploration and optimization through open-ended spaces of chemical compounds.

A deep neural network was trained on hundreds of thousands of existing chemical structures to construct three coupled functions: an encoder, a decoder and a predictor. The encoder converts the discrete representation of a molecule into a real-valued continuous vector, and the decoder converts these continuous vectors back to discrete molecular representations. The predictor estimates chemical properties from the latent continuous vector representation of the molecule.

Continuous representations of molecules allow us to automatically generate novel chemical structures by performing simple operations in the latent space, such as decoding random vectors, perturbing known chemical structures, or interpolating between molecules.

Continuous representations also allow the use of powerful gradient-based optimization to efficiently guide the search for optimized functional compounds. We demonstrate our method in the domain of drug-like molecules and also in a set of molecules with fewer that nine heavy atoms.

4.1 Introduction

The goal of drug and material design is to identify novel molecules that have certain desirable properties. We view this as an optimization problem, in which we are searching for the molecules that maximize our quantitative desiderata. However, optimization in molecular space is extremely challenging, because the search space is large, discrete, and unstructured. Making and testing new compounds is costly and time consuming, and the number of potential candidates is overwhelming. Only about 10^8 substances have ever been synthesized,[77] whereas the range of potential drug-like molecules is estimated to be between 10^{23} and 10^{60} [113].

Virtual screening can be used to speed up this search.[19,117,144,150] Virtual libraries containing thousands to hundreds of millions of candidates can be assayed with first-principles simulations or statistical predictions based on learned proxy models, and only the most promising leads are selected and tested experimentally.

However, even when accurate simulations are available,[141] computational molecular design is limited by the search strategy used to explore chemical space. Current methods either exhaustively search through a fixed library,[49,57] or use discrete local search methods such as genetic algorithms[72,108,126,127,134,168] or similar discrete interpolation techniques.[6,167,173] Although these techniques have led to useful new molecules, these approaches still face large challenges. Fixed libraries are monolithic, costly to fully explore, and require hand-crafted rules to avoid impractical chemistries. The genetic generation of compounds requires the manual specification of heuristics for mutation and crossover rules. Discrete optimization methods have difficulty effectively

searching large areas of chemical space because it is not possible to guide the search with gradients.

A molecular representation method that is continuous, data-driven, and can easily be converted into a machine-readable molecule has several advantages. First, hand-specified mutation rules are unnecessary, as new compounds can be generated automatically by modifying the vector representation and then decoding. Second, if we develop a differentiable model that maps from molecular representations to desirable properties, we can enable the use of gradient-based optimization to make larger jumps in chemical space. Gradient-based optimization can be combined with Bayesian inference methods to select compounds that are likely to be informative about the global optimum. Third, a data-driven representation can leverage large sets of unlabeled chemical compounds to automatically build an even larger implicit library, and then use the smaller set of labeled examples to build a regression model from the continuous representation to the desired properties. This lets us take advantage of large chemical databases containing millions of molecules, even when many properties are unknown for most compounds.

Recent advances in machine learning have resulted in powerful probabilistic generative models that, after being trained on real examples, are able to produce realistic synthetic samples. Such models usually also produce low-dimensional continuous representations of the data being modeled, allowing interpolation or analogical reasoning for natural images[120], text[12], speech, and music[38,166]. We apply such generative models to chemical design, using a pair of deep networks trained as an autoencoder to convert molecules represented as SMILES strings into a continuous vector representation. In principle, this method of converting from a molecular representation to a continuous vector representation could be applied to any molecular representation, including chemical fingerprints,[130] convolutional neural networks on graphs[34], similar graph-convolutions[75], and Coulomb matrices[135]. We chose to use SMILES representation because

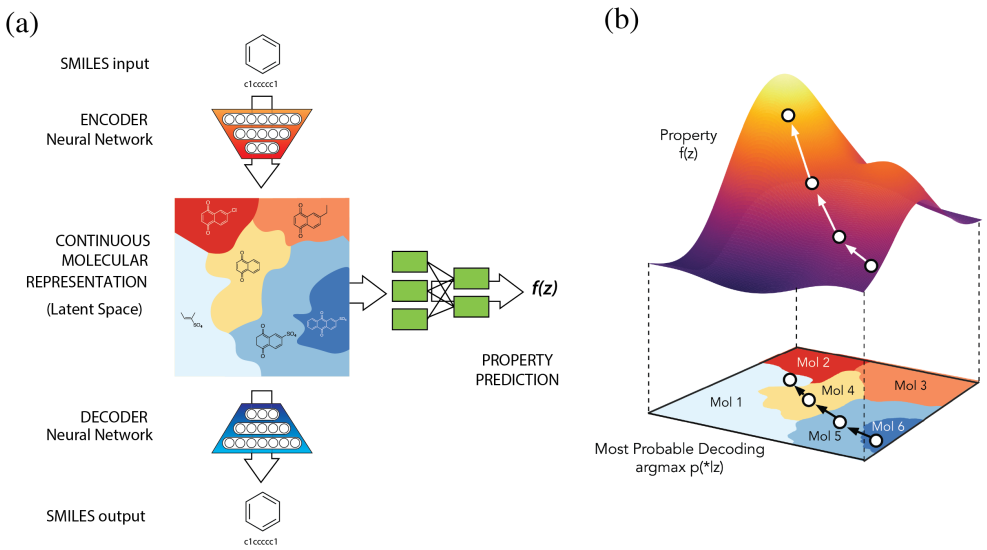


Figure 4.1: (a). A diagram of the autoencoder used for molecular design, including the joint property prediction model. Starting from a discrete molecular representation, such as a SMILES string, the encoder network converts each molecule into a vector in the latent space, which is effectively a continuous molecular representation. Given a point in the latent space, the decoder network produces a corresponding SMILES string. Another network estimates the value of target properties associated with each molecule. **(b)** Gradient-based optimization in continuous latent space. After training a surrogate model $f(z)$ to predict the properties of molecules based on their latent representation z, we can optimize $f(z)$ with respect to z to find new latent representations expected to have high values of desired properties. These new latent representations can then be decoded into SMILES strings, at which point their properties can be tested empirically.

this representation can be readily converted into a molecule.

Using this new continuous vector-valued representation, we experiment with the use of continuous optimization to produce novel compounds. We trained the autoencoder jointly on a property prediction task: we added a multilayer perceptron that predicts property values from the continuous representation generated by the encoder, and included the regression error in our loss function. We examined the effects that joint training had on the latent space, and tested optimization of molecules in this latent space. The code and full training data sets is made available at `https://github.com/aspuru-guzik-group/chemical_vae`

4.1.1 Representation and Autoencoder Framework

The autoencoder in comprised of two deep networks: an encoder network to convert each string into a fixed-dimensional vector, and a decoder network to convert vectors back into strings (Figure 4.1a). The autoencoder is trained to minimize error in reproducing the original string, *i.e.*, it attempts to learn the identity function. Key to the design of the autoencoder is the mapping of strings through an *information bottleneck*. This bottleneck — here the fixed-length continuous vector — induces the network to learn a compressed representation that captures the most statistically salient information in the data. We call the vector-encoded molecule the *latent representation* of the molecule.

For unconstrained optimization in the latent space to work, points in the latent space must decode into valid SMILES strings that capture the chemical nature of the training data. Without this constraint, the latent space learned by the autoencoder may be sparse and may contain large "dead areas", which decode to invalid SMILES strings. To help ensure that points in the latent space correspond to valid realistic molecules, we choose to use a *variational* autoencoder (VAE)[79] framework. VAEs were developed as a principled approximate-inference method for latent-variable models, in which each datum has a corresponding, but unknown, latent representation. VAEs generalize autoencoders, adding stochasticity to the encoder which combined with a penalty term encourages all areas of the latent space to correspond to a valid decoding. The intuition is that adding noise to the encoded molecules forces the decoder to learn how to decode a wider variety of latent points and find more robust representations. Variational autoencoders with recurrent neural network encoding/decoding were proposed by Bowman *et al.* in the context of written English sentences and we followed their approach closely.[12]. To leverage the power of recent advances in sequence-to-sequence autoencoders for modeling text, we used the SMILES[176] representation, a commonly-used text encoding for organic molecules.

The character-by-character nature of the SMILES representation and the fragility of its internal syntax (opening and closing cycles and branches, allowed valences, etc.) can still result in the output of invalid molecules from the decoder, even with the variational constraint. When converting a molecule from a latent representation to a molecule, the decoder model samples a string from the probability distribution over characters in each position generated by its final layer. As such, multiple SMILES strings are possible from a single latent space representation. We employed the open source cheminformatics suite RDKit[125] to validate the chemical structures of output molecules and discard invalid ones. While it would be more efficient to limit the autoencoder to generate only valid strings, this post-processing step is lightweight and allows for greater flexibility in the autoencoder to learn the architecture of the SMILES.

To enable molecular design, the chemical structures encoded in the continuous representation of the autoencoder need to be correlated with the target properties that we are seeking to optimize. Therefore, we added a model to the autoencoder that predicts the properties from the latent space representation. This autoencoder was then trained jointly on the reconstruction task and a property prediction task; an additional multi-layer perceptron (MLP) was used to predict the property from the latent vector of the encoded molecule. To propose promising new candidate molecules, we can start from the latent vector of an encoded molecule and then move in the direction most likely to improve the desired attribute. The resulting new candidate vectors can then be decoded into corresponding molecules. (Figure 4.1b).

Two autoencoder systems were trained; one with 108,000 molecules from the QM9 dataset of molecules with fewer than 9 heavy atoms[121] and another with 250,000 drug-like commercially available molecules extracted at random from the ZINC database.[65]. We performed random optimization over hyperparameters specifying the deep autoencoder architecture and training, such as the choice between a recurrent or convolutional encoder, the number of hidden layers, layer

sizes, regularization and learning rates. The latent space representations for the QM9 and ZINC datasets had 156 dimensions and 196 dimensions respectively.

4.2 Results and discussion

Representation of molecules in latent space Firstly, we analyze the fidelity of the autoencoder and the ability of the latent space to capture structural molecular features. Figure 4.2a) shows a kernel density estimate of each dimension when encoding a set of 5000 randomly selected ZINC molecules from outside the training set. The kernel density estimate shows the distribution of datapoints along each dimension of the latent space. Whereas the distribution of datapoint in each individual dimension shows a slightly different mean and standard deviation, all the distributions are normal as enforced by the variational regularizer.

The variational autoencoder is a doubly-probabilistic model. In addition to the Gaussian noise added to the encoder, which can be turned off by simply sampling the mean of the encoding distribution, the decoding process is also non-deterministic, as the string output is sampled from the final layer of the decoder. This implies that decoding a single point in the latent space back to a string representation is stochastic. Figure 4.2b) shows the probability of decoding the latent representation of a sample FDA-approved drug molecule into several different molecules. For most latent points, a prominent molecule is decoded and many other slight variations appear with lower frequencies. When these resulting SMILES are re-encoded into the latent space, the most frequent decoding also tends to be the one with the lowest Euclidean distance to the original point, indicating the latent space is indeed capturing features relevant to molecules.

Figure 4.2c) shows some molecules in the latent space that are close to ibuprofen. These structures become less similar with increasing distance in the latent space. When the distance approaches the average distance of molecules in the training set, the changes are more pronounced,

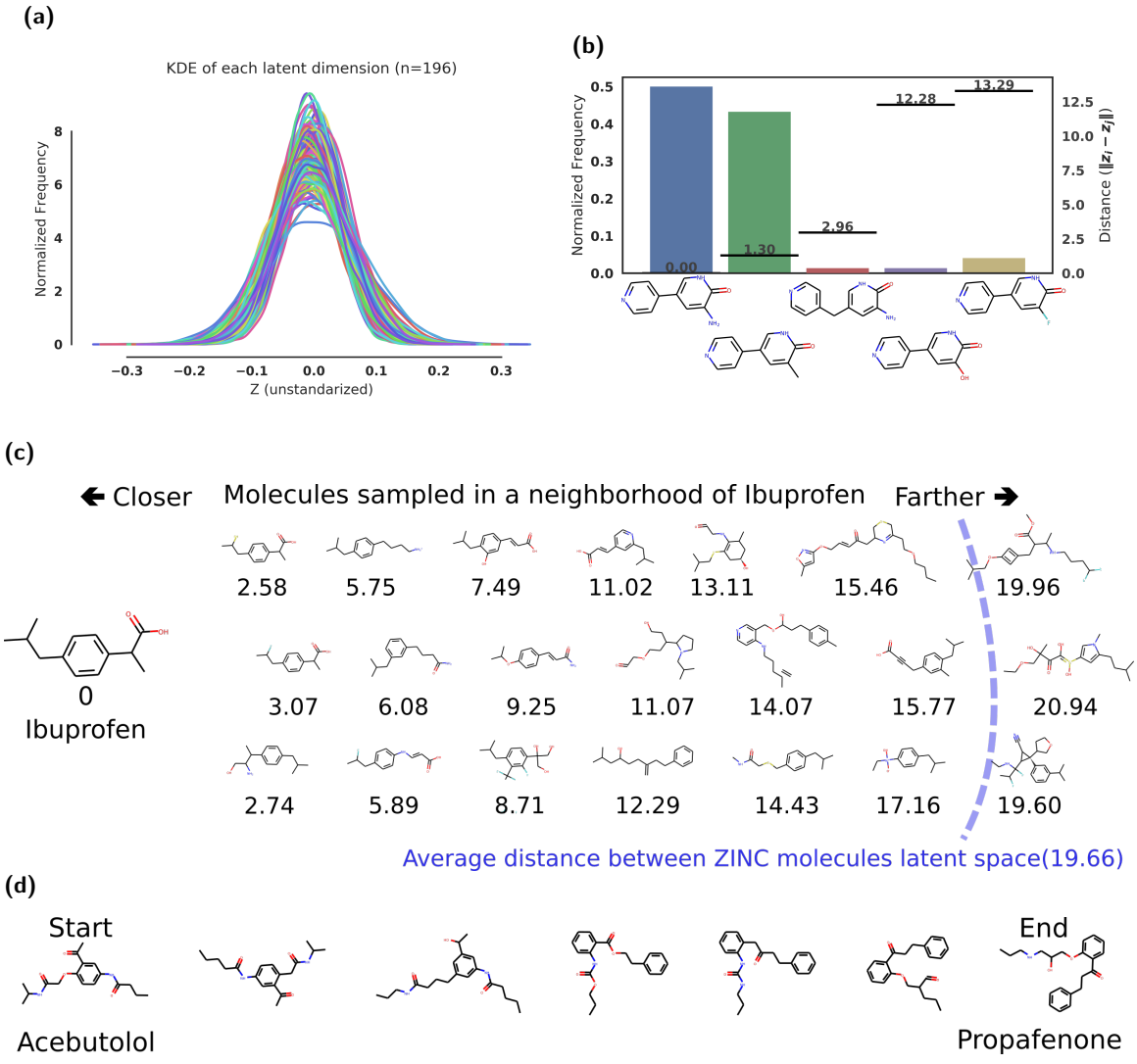


Figure 4.2: Representations of the sampling results from the variational autoencoder. (a) Kernel Density Estimation (KDE) of each latent dimension of the autoencoder, i.e. the distribution of encoded molecules along each dimension of our latent space representation; (b) Histogram of sampled molecules for a single point in the latent space, the distances of the molecules from the original query are shown by the lines corresponding to the right axis; (c) Molecules sampled near the location of ibuprofen in latent space. The values below the molecules are the distance in latent space from the decoded molecule to ibuprofen; (d) *slerp* interpolation between two molecules in latent space using 6 steps of equal distance.

eventually resembling random molecules likely to be sampled from the training set. SI Figure 1d) shows the distribution of distances in latent space between 50,000 random points from our ZINC training set. We estimate that we can find 30 such molecules in the locality of a molecule, i.e. 30 molecules closer to a given seed molecule from our dataset than any other molecule in our dataset. As such, we estimate that our autoencoder that was trained on 250,000 molecules from ZINC encodes 7.5 million molecules. The probability of decoding from a point in latent space is dependent on how close this point is to the latent representations of other molecules; we observed a decoding rate of 73-79% for points that are close to known molecules, and 4% for randomly selected latent points.

A continuous latent space allows interpolation of molecules by following the shortest Euclidean path between their latent representations. When exploring high dimensional spaces, it is important to note that Euclidean distance might not map directly to notions of similarity of molecules[2]. In high dimensional spaces, most of the mass of independent normally-distributed random variables is not near the mean, but in an annulus around the mean[31]. Interpolating linearly between two points might pass by an area of low probability, to keep the sampling on the areas of high probability we utilize spherical interpolation[177] (*slerp*). With *slerp*, the path between two points is a circular arc lying on the on the surface of a N-dimensional sphere. Figure 4.2d) shows the spherical interpolation between two random drug molecules, showing smooth transitions in between. SI Figure 3 shows the difference between linear and spherical interpolation.

Table 4.1 compares the distribution of chemical properties in the training sets against molecules generated with a baseline genetic algorithm, and molecules generated from the variational autoencoder. In the genetic algorithm, molecules were generated with a list of hand-designed rules[72,108,126,127,134,168]. This process was seeded using 1000 random molecules from the ZINC dataset, and generated over 10 iterations. For molecules generated using the variational

Source[a]	Dataset[b]	Samples[c]	logP[d]	SAS[e]	QED[f]	% in ZINC[g]	% in emol[h]
Data	ZINC	249k	2.46 (1.43)	3.05 (0.83)	0.73 (0.14)	100	12.9
GA	ZINC	5303	2.84 (1.86)	3.80 (1.01)	0.57 (0.20)	6.5	4.8
VAE	ZINC	8728	2.67 (1.46)	3.18 (0.86)	0.70 (0.14)	5.8	7.0
Data	QM9	134k	0.30 (1.00)	4.25 (0.94)	0.48 (0.07)	0.0	8.6
GA	QM9	5470	0.96 (1.53)	4.47 (1.01)	0.53 (0.13)	0.018	3.8
VAE	QM9	2839	0.30 (0.97)	4.34 (0.98)	0.47 (0.08)	0.0	8.9

Table 4.1: Comparison of molecule generation results to original datasets. Column a) describes the source of the molecules: data refers to the original dataset, GA refers to the genetic algorithm baseline, and VAE to our variational autoencoder trained without property prediction; b) shows the dataset used, either ZINC or QM9, c) shows the number of samples generated for comparison, for data, this value simply reflects the size of the dataset. Columns d) through f) show the mean and, in parenthesis, the standard deviation of selected properties of the generated molecules and compares that to the mean and standard deviation of properties in the original dataset. d) shows the water-octanal partition coefficient (`logP`)[178]; e) shows the synthetic accessibility score (`SAS`)[39]; and f) shows the Qualitative Estimate of Drug-likeness (`QED`)[10], ranging from 0 to 1; We also examine how many of the molecules generated by each method are found in two major molecule databases: ZINC in column g) and E-molecules[35] in column h) and compare these values against the original dataset.

autoencoder, we collected the set of all molecules generated from 400 decoding attempts from the latent space points encoded from the same 1000 seed molecules. We compare the water-octanol partition coefficient (logP), the synthetic accessibility score (SAS),[39] and Quantitative Estimation of Drug-likeness (QED),[10] which ranges in value between 0 and 1, with higher values indicating that the molecule is more drug-like. SI Figure 2 shows histograms of the properties of the molecules generated by each of these approaches and compares them to the distribution of properties from the original dataset. Despite the fact that the VAE is trained purely on the SMILES strings independently of chemical properties, it is able to generate realistic-looking molecules whose features follow the intrinsic distribution of the training data. The molecules generated using the VAE show chemical properties that are more similar to the original dataset than the set of molecules generated by the genetic algorithm. The two rightmost columns in Table 4.1 report the fraction of molecules that belong to the the 17 million drug-like compounds from which the

Database/Property	Mean[a]	ECFP[b]	CM[b]	GC[b]	1-hot SMILES[c]	Encoder[d]	VAE[e]
ZINC250k/logP	1.14	0.38	-	0.05	0.16	0.13	0.15
ZINC250k/QED	0.112	0.045	-	0.017	0.041	0.037	0.054
QM9/HOMO, eV	0.44	0.20	0.16	0.12	0.12	0.13	0.16
QM9/LUMO, eV	1.05	0.20	0.16	0.15	0.11	0.14	0.16
QM9/Gap, eV	1.07	0.30	0.24	0.18	0.16	0.18	0.21

Table 4.2: MAE prediction error for properties using various methods on the ZINC and QM9 datasets. a) Baseline, mean prediction; b) As implemented in Deepchem benchmark (MoleculeNet)[180], ECFP-circular fingerprints, CM-coulomb matrix, GC-graph convolutions; c) 1-hot-encoding of SMILES used as input to property predictor; d) The network trained without decoder loss; e) full variational autoencoder network trained for individual properties.

training set was selected and how often they can be found in a library of existing organic compounds. In the case of drug-like molecules, the VAE generates molecules that follow the property distribution of the training data, but are new as the combinatorial space is extremely large and the training set is an arbitrary sub-sample. The hand-selected mutations are less able to generate new compounds while at the same time biasing the properties of the set to higher chemical complexity and decreased drug-likeness. In the case of the QM9 dataset, since the combinatorial space is smaller, the training set has more coverage and the VAE generates essentially the same population statistics as the training data.

PROPERTY PREDICTION OF MOLECULES The interest in discovering new molecules and chemicals is most often in relation to maximizing some desirable property. For this reason, we extended the the purely generative model to also predict property values from the latent representation. We trained a multi-layer perceptron jointly with the autoencoder to predict properties from the latent representation of each molecule.

With joint training for property prediction, the distribution of molecules in the latent space is organized by property values. Figure 4.3 shows the mapping of true property values to the latent space representation of molecules, compressed into two dimensions using PCA. The latent space

generated by autoencoders jointly trained with the property prediction task shows in the distribution of molecules a gradient by property values; molecules with high values are located in one region, and molecules with low values in another. Autoencoders that were trained without the property prediction task do not show a discernible pattern with respect to property values in the resulting latent representation distribution.

While the primary purpose of adding property prediction was to organize the latent space, it is interesting to observe how the property predictor model compares with other standard models for property prediction. For a more fair comparison against other methods, we increased the size of our perceptron to two layers of 1,000 neurons. Table 4.2 compares the performance of commonly used molecular embeddings and models to the VAE. Our VAE model shows that property prediction performance for electronic properties (*i.e.*, orbital energies) are similar to graph convolutions for some properties; prediction accuracy could be improved with further hyperparameter optimization.

OPTIMIZATION OF MOLECULES VIA PROPERTIES We next optimized molecules in the latent space from the autoencoder which was jointly trained for property prediction. In order to create a smoother landscape to perform optimizations, we used a Gaussian process model to model the property predictor model. Gaussian processes can be used to predict any smooth continuous function[124] and are extremely lightweight, requiring only a few minutes to train on a dataset of a few thousand molecules. The Gaussian process was trained to predict target properties for molecules given the latent space representation of the molecules as an input.

The 2,000 molecules used for training the Gaussian process were selected to be maximally diverse. Using this model, we optimized in the latent space to find a molecule that maximized our objective. As a baseline, we compared our optimization results against molecules found using a random Gaussian search and molecules optimized via a genetic algorithm.

The objective we chose to optimize was $5\times$ `QED` $-$ `SAS`, where `QED` is the Quantitative

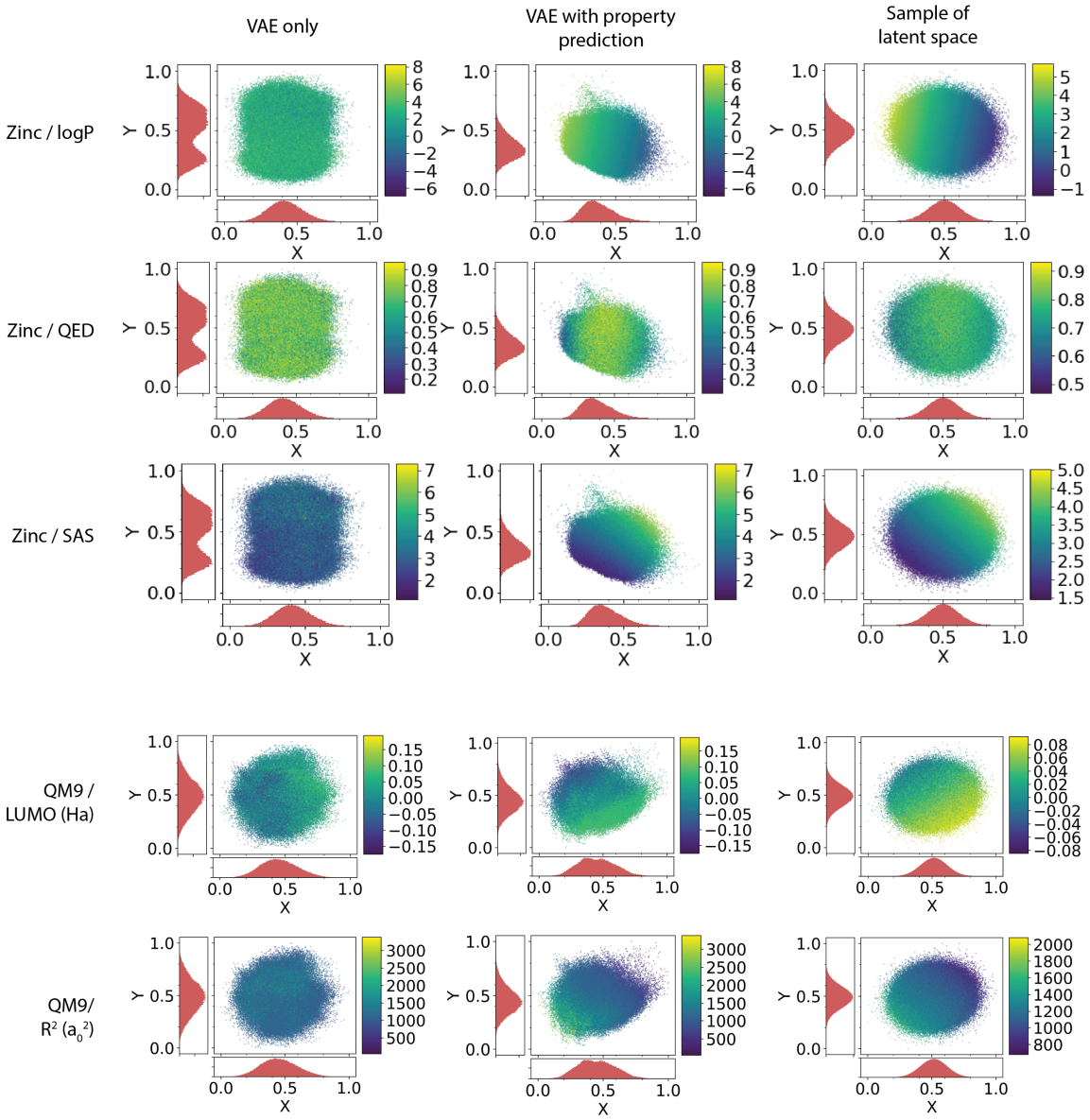


Figure 4.3: Two-dimensional PCA analysis of latent space for variational autoencoder. The two axis are the principle components selected from the PCA analysis, the color bar shows the value of the selected property. The first column shows the representation of all molecules from the listed dataset using autoencoders trained without joint property prediction. The second column shows the representation of molecules using an autoencoder trained with joint property prediction. The third column shows a representation of random points in the latent space of the autoencoder trained with joint property prediction; the property values predicted for these points are predicted using the property predictor network. The first three rows show the results of training on molecules from the ZINC dataset for the `logP`, `QED`, and `SAS` properties; the last two rows show the results of training on the QM9 dataset for the LUMO energy and the electronic spatial extent (R^2).

Estimation of Drug-likeness (QED)[10], and SAS is the Synthetic Accessibility score[39]. This objective represents a rough estimate of finding the most drug-like molecule that is also easy to synthesize. To provide the greatest challenge for our optimizer, we started with molecules from the ZINC dataset that had an objective score in the bottom 10%, i.e. were in the 10th percentile.

From Figure 4.4a) we can see that the optimization with the Gaussian process model on the latent space representation consistently results in molecules with a higher percentile score than the two baseline search methods. Figure 4.4b) shows the path of one optimization from the starting molecule to the final molecule in the two-dimensional PCA representation, the final molecule ending up in the region of high objective value. Figure 4.4c) shows molecules decoded along this optimization path using a Gaussian interpolation.

Performing this optimization on a Gaussian process (GP) model trained with 1,000 molecules leads to a slightly wider range of molecules as shown in Figure 4.4a). Since the training set is smaller, the predictive power of the GP is lower which when optimizing in latent space, and as a result optimizes to several local minima instead of a global optimization. In cases where it is difficult to define an objective that completely describes the desirable traits of the molecule, it may be better to use this localized optimization approach to reach a larger diversity of potential molecules.

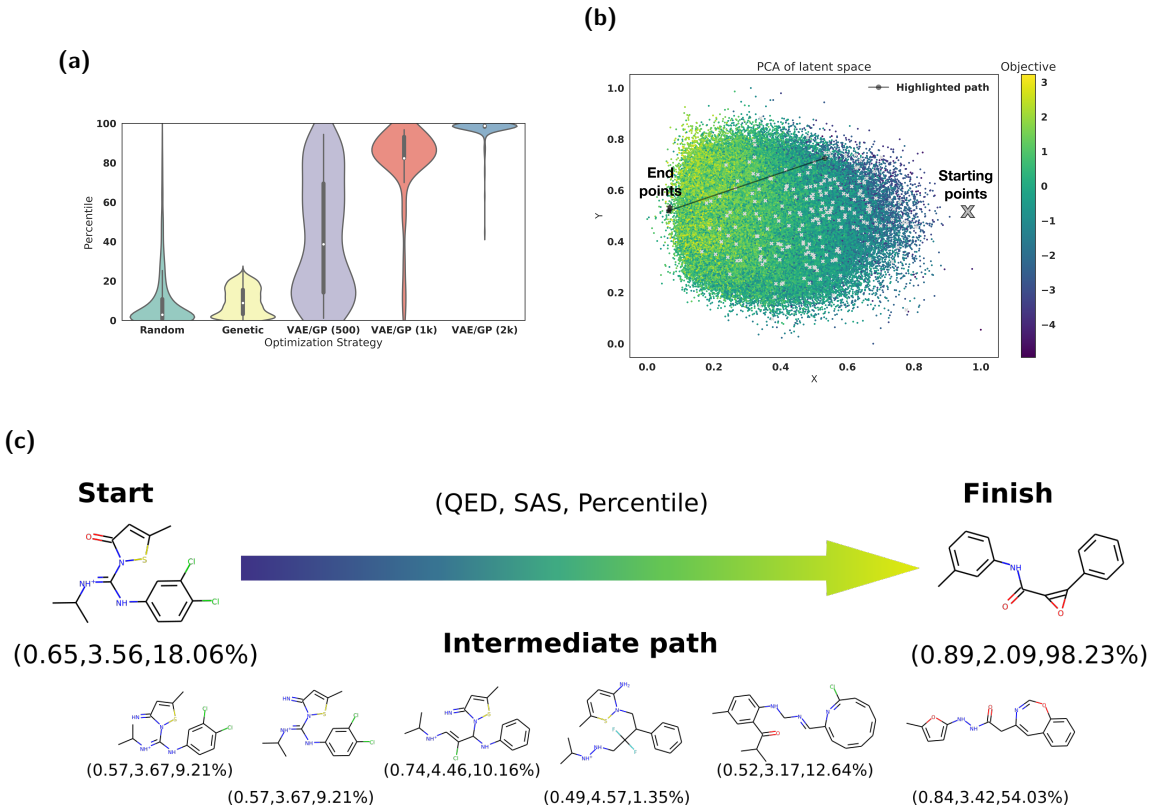


Figure 4.4: Optimization results for the jointly trained autoencoder using $5\times$ `QED` $-$ `SAS` as the objective function. Part (a) shows a box plot which compares the distribution of sampled molecules from normal random sampling, SMILES optimization via a common chemical transformation with a genetic algorithm, and from optimization on the trained gaussian process model with varying amounts of training points. To offset differences in computational cost between the random search and the optimization on the gaussian process model, the results of 400 iterations of random search were compared against the results of 200 iterations of optimization. This graph shows the combined results of four sets of trials. Part (b) shows the starting and ending points of several optimization runs on a PCA plot of latent space colored by the objective functon. Higlighted in black is the path illustrated in c). Part (c) shows a spherical interpolation between the actual start and finish molecules using a constant step size. The QED, SAS, and percentile score are reported for each molecule.

4.3 Conclusion

We propose a new family of methods for exploring chemical space based on continuous encodings of molecules. These methods eliminate the need to hand-build libraries of compounds and allow a new type of directed gradient-based search through chemical space. In our autoencoder model, we observed high fidelity in reconstruction of SMILES strings and the ability to capture characteristic features of a molecular training set. The autoencoder exhibited good predictive power when training jointly with a property prediction task, and the ability to perform gradient-based optimization of molecules in the resulting smoothed latent space.

There are several directions for further improvement of this approach to molecular design. In this work, we used a text-based molecular encoding, but using a graph-based autoencoder would have several advantages. Forcing the decoder to produce valid SMILES strings makes the learning problem unnecessarily hard since the decoder must also implicitly learn which strings are valid SMILES. An autoencoder that directly outputs molecular graphs is appealing since it could explicitly address issues of graph isomorphism and the problem of strings that do not correspond to valid molecular graphs. Building an encoder which takes in molecular graphs is straightforward through the use of off-the-shelf molecular fingerprinting methods, such as ECFP[130] or a continuously-parameterized variant of ECFP such as neural molecular fingerprints.[34] However, building a neural network which can output arbitrary graphs is an open problem.

Further extensions of this work to use a explicitly defined grammar for SMILES instead of forcing the model to learn one[82] or to actively learn valid sequences[66,67] are underway, as also is the application of adversarial networks for this task,[11,56,137] as well as other reinforcement learning algorithms which make transformations by adding bonds or removing them.[185] Several proceeding works have further explored the use of Long Short-Term Memory (LSTM) networks and recurrent networks applied to SMILES strings to generate new molecules[146,182] and predict the outcomes of

organic chemistry reactions.[91]

The autoencoder sometimes produced molecules that are formally valid as graphs but contain moieties that are not desirable because of stability or synthetic constraints. Examples are acid chlorides, anhydrides, cyclopentadienes, aziridines, enamines, hemiaminals, enol ethers, cyclobutadiene, and cycloheptatriene. One option is to train the autoencoder with to predict properties related to steric constraints of other structural constraints. In general, the objective function to be optimized needs to capture as many desirable traits as possible and balance them to ensure that the optimizer focuses on genuinely desirable compounds. This approach has also been tested in a few following works works.[66,67]

The results reported in this work, and its application with carefully composed objective functions, have already and will continue to influence new avenues for molecular design.

4.4 Methods

AUTOENCODER ARCHITECTURE Strings of characters can be encoded into vectors using recurrent neural networks (RNNs). An encoder RNN can be paired with a decoder RNN to perform sequence-to-sequence learning.[160] We also experimented with convolutional networks for string encoding[71] and observed improved performance. This is explained by the presence of repetitive, translationally-invariant substrings that correspond to chemical substructures, e.g., cycles and functional groups.

Our SMILES-based text encoding used a subset of 35 different characters for ZINC and 22 different characters for QM9. For ease of computation, we encoded strings up to a maximum length of 120 characters for ZINC and 34 characters for QM9, although in principle there is no hard limit to string length. Shorter strings were padded with spaces to this same length. We used only canonicalized SMILES for training to avoid dealing with equivalent SMILES representations.

The structure of the VAE deep network was as follows: For the autoencoder used for the ZINC dataset, the encoder used three 1D convolutional layers of filter sizes 9, 9, 10 and 9, 9, 11 convolution kernels, respectively, followed by one fully-connected layer of width 196. The decoder fed into three layers of gated recurrent unit (GRU) networks[22] with hidden dimension of 488. For the model used for the QM9 dataset, the encoder used three 1D convolutional layers of filter sizes 2, 2, 1 and 5, 5, 4 convolution kernels, respectively, followed by one fully-connected layer of width 156. The three recurrent neural network layers each had a hidden dimension of 500 neurons.

The last layer of the RNN decoder defines a probability distribution over all possible characters at each position in the SMILES string. This means that the writeout operation is stochastic, and the same point in latent space may decode into to different SMILES strings, depending on the random seed used to sample characters. The output GRU layer had one additional input, corresponding to the character sampled from the softmax output of the previous time step and was trained using teacher forcing.[179] This increased the accuracy of generated SMILES strings, which resulted in higher fractions of valid SMILES strings for latent points outside the training data, but also made training more difficult, since the decoder showed a tendency to ignore the (variational) encoding and rely solely on the input sequence. The variational loss was annealed according to sigmoid schedule after 29 epochs, running for a total 120 epochs.

For property prediction, two fully connected layers of 1000 neurons were used to predict properties from the latent representation, with a dropout rate of 0.20. To simply shape the latent space, a smaller perceptron of 3 layers of 67 neurons was used for the property predictor, trained with a dropout rate of 0.15. For the algorithm trained on the ZINC dataset, the objective properties include logP, QED, SAS. For the algorithm trained on the QM9 dataset, the objective properties include HOMO energies, LUMO energies, and the electronic spatial extent (R^2). The property prediction loss was annealed in at the same time as the variational loss. We used the Keras[20] and

TensorFlow[1] packages to build and train this model and the rdkit package for cheminformatics[125].

4.5 ACKNOWLEDGEMENT

This work was supported financially by the Samsung Advanced Institute of Technology. The authors acknowledge the use of the Harvard FAS Odyssey Cluster and support from FAS Research Computing. JNW acknowledges support from the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1144152. JMHL acknowledges support from the Rafael del Pino Foundation. RPA acknowledges support from the Alfred P. Sloan Foundation and NSF IIS-1421780. AAG acknowledges support from The Department of Energy, Office of Basic Energy Sciences under award DE-SC0015959. We thank Dr. Anders Frøseth for his generous support of this work.

5

Neural Networks for Predicting Reactions

Apart from minor modifications, this chapter originally appeared as:

"Neural networks for the Prediction of Organic Chemistry Reactions". Wei, J.N., Duvenaud, D., Aspuru-Guzik, Alán. ACS Central Science 2(10), 2016. 725-732.

ABSTRACT

Reaction prediction remains one of the major challenges for organic chemistry, and is a prerequisite for efficient synthetic planning. It is desirable to develop algorithms that, like humans, "learn" from being exposed to examples of the application of the rules of organic chemistry. We explore the use of neural networks for predicting reaction types, using a new reaction fingerprinting method. We combine this predictor with SMARTS transformations to build a system which, given a set of reagents and reactants, predicts the likely products. We test this method on problems from a popular organic chemistry textbook.

5.1 INTRODUCTION

To develop the intuition and understanding for predicting reactions, a human must take many semesters of organic chemistry and gather insight over several years of lab experience. Over the past 40 years, various algorithms have been developed to assist with synthetic design, reaction prediction, and starting material selection[161,164]. LHASA was the first of these algorithms to aid in developing retrosynthetic pathways[29]. This algorithm required over a decade of effort to encode

the necessary subroutines to account for the various subtleties of retrosynthesis such as functional group identification, polycyclic group handling, relative protecting group reactivity, and functional group based transforms[25–28].

In the late 1980s to the early 1990s, new algorithms for synthetic design and reaction prediction were developed. CAMEO[70], a reaction predicting code, used subroutines specialized for each reaction type, expanding to include reaction conditions in its analysis. EROS[45] identified leading structures for retrosynthesis by using bond polarity, electronegativity across the molecule, and the resonance effect to identify the most reactive bond. SOPHIA[138] was developed to predict reaction outcomes with minimal user input; this algorithm would guess the correct reaction type subroutine to use by identifying important groups in the reactants; once the reactant type was identified, product ratios would be estimated for the resulting products. SOPHIA was followed by the KOSP algorithm, and uses the same database to predict retrosynthetic targets[139]. Other methods generated rules based on published reactions, and uses these transformations when designing a retrosynthetic pathway[46,148]. Some methods encoded expert rules in the form of electron flow diagrams[17,18]. Another group attempted to grasp the diversity of reactions by creating an algorithm that automatically searches for reaction mechanisms using atom mapping and substructure matching[85].

While these algorithms have their subtle differences, all require a set of expert rules to predict reaction outcomes. Taking a more general approach, one group has encoded all of the reactions of the Beilstein database, creating a 'Network of Organic Chemistry'[51,161]. By searching this network, synthetic pathways can be developed for any molecule similar enough to a molecule already in its database of 7 million reactions, identifying both one-pot reactions that do not require time-consuming purification of intermediate products[54], or full multistep reactions that account for the cost of the materials, labor, and safety of the reaction[161]. Algorithms that use encoded expert rules or databases of published reactions are able to accurately predict chemistry for queries that

match reactions in its knowledge base. However, such algorithms do not have the ability of a human organic chemist to predict the outcomes of previously unseen reactions. In order to predict the results of new reactions, the algorithm must have a way of connecting information from reactions that it has been trained upon to reactions that it has yet to encounter.

Another strategy of reaction prediction algorithm draws from principles of physical chemistry and first predicts the energy barrier of a reaction in order to predict its likelihood[123,153,171,172,181,187]. Specific examples of reactions include the development of a nanoreactor for early Earth reactions[171,172], Heuristic Aided Quantum Chemistry[123], and ROBIA[153], an algorithm for reaction prediction. While methods that are guided by quantum calculations have the potential to explore a wider range of reactions than the heuristic-based methods, these algorithms would require new calculations for each additional reaction family, and will be prohibitively costly over a large set of new reactions.

A third strategy for reaction prediction algorithms uses statistical machine learning. These methods can sometimes generalize or extrapolate to new examples, as in the recent examples of picture and handwriting identification[59,81], playing video games[101], and most recently, playing Go[151]. This last example is particularly interesting as Go is a complex board game with a search space of 10^{170}, which is on the order of chemical space for medium sized molecules[128]. SYNCHEM was one early effort in the application of machine learning methods to chemical predictions, which relied mostly on clustering similar reactions, and learning when reactions could be applied based on the presence of key functional groups[46].

Today, most machine learning approaches in reaction prediction use molecular descriptors to characterize the reactants in order to guess the outcome of the reaction. Such descriptors range from physical descriptors such as molecular weight, number of rings, or partial charge calculations to molecular fingerprints, a vector of bits or floats that represent the properties of the molecule.

ReactionPredictor[73,74] is an algorithm that first identifies potential electron sources and electron sinks in the reactant molecules based on atom and bond descriptors. Once identified, these sources and sinks are paired to generate possible reaction mechanisms. Finally, neural networks are used to determine the most likely combinations in order to predict the true mechanism. While this approach allows for the prediction of many reactions at the mechanistic level, many of the elementary organic chemistry reactions that are the building blocks of organic synthesis have complicated mechanisms, requiring several steps that would be costly for this algorithm to predict.

Many algorithms that predict properties of organic molecules use various types of fingerprints as the descriptor. Morgan fingerprints and extended circular fingerprints[104,131] have been used to predict molecular properties such as HOMO-LUMO gaps[116], protein-ligand binding affinity[7], drug toxicity levels[184], and even to predict synthetic accessibility[112]. Recently Duvenavud et al. applied graph neural networks[34] to generate continuous molecular fingerprints directly from molecular graphs. This approach generalizes fingerprinting methods such as the ECFP by parameterizing the fingerprint generation method. These parameters can then be optimized for each prediction task, producing fingerprint features that are relevant for the task. Other fingerprinting methods that have been developed use the Coulomb matrix[102], radial distribution functions[169], and atom pair descriptors[16]. For classifying reactions, one group developed a fingerprint to represent a reaction by taking the difference between the sum of the fingerprints of the products and sum of the fingerprints of the reactants[142]. A variety of fingerprinting methods were tested for the constituent fingerprints of the molecules.

In this work, we apply fingerprinting methods, including neural molecular fingerprints, to predict organic chemistry reactions. Our algorithm predicts the most likely reaction type for a given set of reactants and reagents, using what it has learned from training examples. These input molecules are described by concatenating the fingerprints of the reactants and the reagents; this

concatenated fingerprint is then used as the input for a neural network to classify the reaction type. With information about the reaction type, we can make predictions about the product molecules. One simple approach for predicting product molecules from the reactant molecules, which we use in this work, is to apply a SMARTS transformation that describes the predicted reaction. Previously, sets of SMARTS transformations have been applied to produce large libraries of synthetically accessible compounds in the areas of molecular discovery[168], metabolic networks[97], drug discovery[159], and discovering one-pot reactions[52]. In our algorithm, we use SMARTS transformation for targeted prediction of product molecules from reactants. However, this method can be replaced by any method that generates product molecule graphs from reactant molecule graphs. An overview of our method can be found in 5.1, and is explained in further detail in the Prediction Methods section.

We show the results of our prediction method on 16 basic reactions of alkylhalides and alkenes, some of the first reactions taught to organic chemistry students in many textbooks[170]. The training and validation reactions were generated by applying simple SMARTS transformations to alkenes and alkylhalides. While we limit our initial exploration to aliphatic, non-stereospecific molecules, our method can easily be applied a wider span of organic chemical space with enough example reactions. The algorithm can also be expanded to include experimental conditions such as reaction temperature and time. With additional adjustments and a larger library of training data, our algorithm will be able to predict multistep reactions, and eventually, become a module in a larger machine-learning system for suggesting retrosynthetic pathways for complex molecules. The code and full training datasets is made available at

`https://github.com/jnwei/neural_reaction_fingerprint.git`.

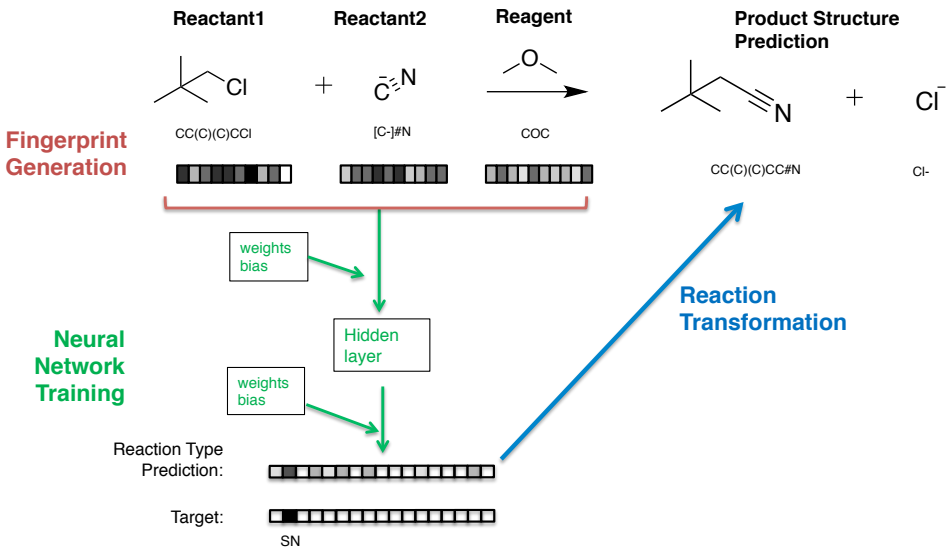


Figure 5.1: An overview of our method for predicting reaction type and products. A reaction fingerprint, made from concatenating the fingerprints of reactant and reagent molecules, is the inputs for a neural network that predicts the probability of 17 different reaction types, represented as a reaction type probability vector. The algorithm then predicts a product by applying a transformation that corresponds with the most probable reaction type to the reactants. In this work, we use a SMARTS transformation for the final step.

5.2 Results and Discussion

5.2.1 Performance on cross-validation set

We created a dataset of reactions of four alkylhalide reactions and twelve alkene reactions; further details on the construction of the dataset can be found in the Methods section. Our training set comprised of 3400 reactions from this dataset, and the test set comprised of 17,000 reactions; both the training set and the test set were balanced across reaction types. During optimization on the training set, k-fold cross-validation was used to help tune the parameters of the neural net. Table 1 reports the cross-entropy score and the accuracy of the baseline and fingerprinting methods on this test set. Here the accuracy is defined by the percentage of matching indices of maximum values in the predicted probability vector and the target probability vector for each reaction.

Figure 5.2 shows the confusion matrices for the baseline, neural, and Morgan fingerprinting

Fingerprint Method	Fingerprint Length	Train NLL	Train Accuracy	Test NLL	Test Accuracy
Baseline	51	0.2727	78.8%	2.5573	24.7%
Morgan	891	0.0971	86.0%	0.1792	84.5%
Neural	181	0.0976	86.0%	0.1340	85.7%

Table 5.1: Accuracy and Negative Log Likelihood (NLL) Error of fingerprint and baseline methods

methods respectively. The confusion matrices for the Morgan and neural fingerprints show that the predicted reaction type and the true reaction type correspond almost perfectly, with few mismatches. The only exceptions are in the predictions for reaction types 3 and 4, corresponding to nucleophilic substitution reaction with a methyl shift and the elimination reaction with a methyl shift. As described in the methods section, these reactions are assumed to occur together, so they are each assigned probabilities of 50% in the training set. As a result, the algorithm cannot distinguish these reaction type and the result on the confusion matrix is a 2x2 square. For the baseline method, the first reaction type, the 'NR' classification, is often over predicted, with some additional overgeneralization of some other reaction type as shown by the horizontal bands.

5.2.2 Performance on predicting reaction type of exam questions

Kayala et al.[74] had previously employed organic textbook questions both as the training set and as the validation set for their algorithm, reporting 95.7% accuracy on their training set. We similarly decided to test our algorithm on a set of textbook questions. To challenge our algorithm, we tested the performance on textbook problems that an organic chemistry student would see. We selected problems 8-47 and 8-48 from the Wade 6th edition organic chemistry textbook, copied below in Figure 5.3[170]. The reagents listed in each problem were assigned as secondary reactants or reagents so that they matched the training set. For all prediction methods, our networks were first trained on the training set of generated reactions, using the same hyperparameters found by the cross-validation search. The similarity of the exam questions to the training set was determined by

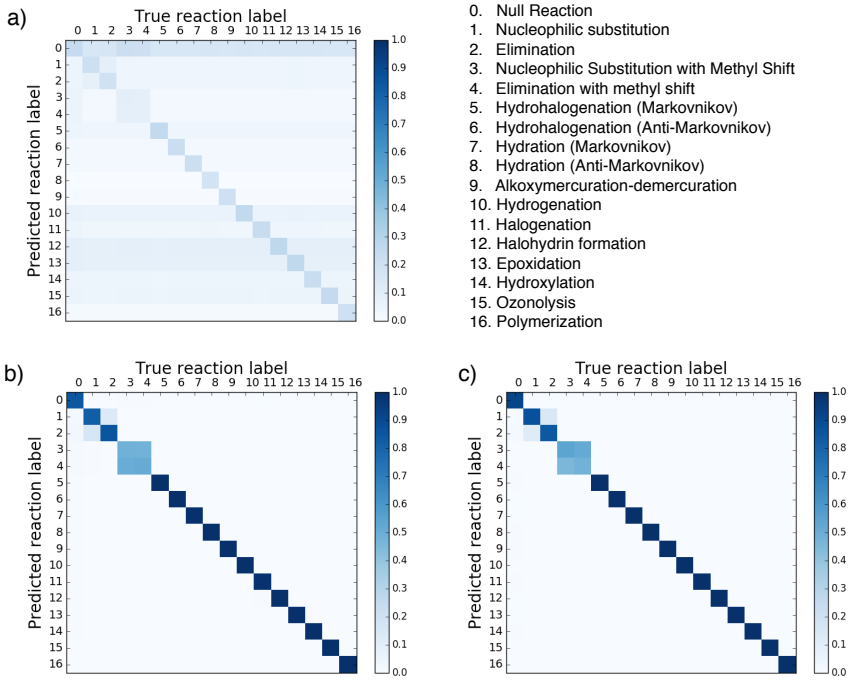


Figure 5.2: Cross validation results for a) Baseline fingerprint, b) Morgan reaction fingerprint, and c) neural reaction fingerprint. A confusion matrix shows the average predicted probability for each reaction type. In these confusion matrices, the predicted reaction type is represented on the vertical axis, and the correct reaction type is represented on the horizantal axis. These figures were generated based on code from Schneider et al.[142].

measuring the Tanimoto[5] distance of the fingerprints of the reactant and reagent molecules in each reactant set. The average Tanimoto score between the training set reactants and reagents and the exam set reactants and reagents is 0.433, and the highest Tanimoto score oberved between exam questions and training questions was 1.00 on 8-48c and 0.941 on 8-47a. This indicates that 8-48c was one of the training set examples. Table **??** show more detailed results for this Tanimoto analysis.

For each problem, the algorithm determined the reaction type in our set that best matched the answer. If the reaction in the answer key did not match any of our reaction types, the algorithm designated the reaction as a null reaction. The higher the probability the algorithm assigned for each reaction type, the more certainty the algorithm has in its prediction. These probabilities are

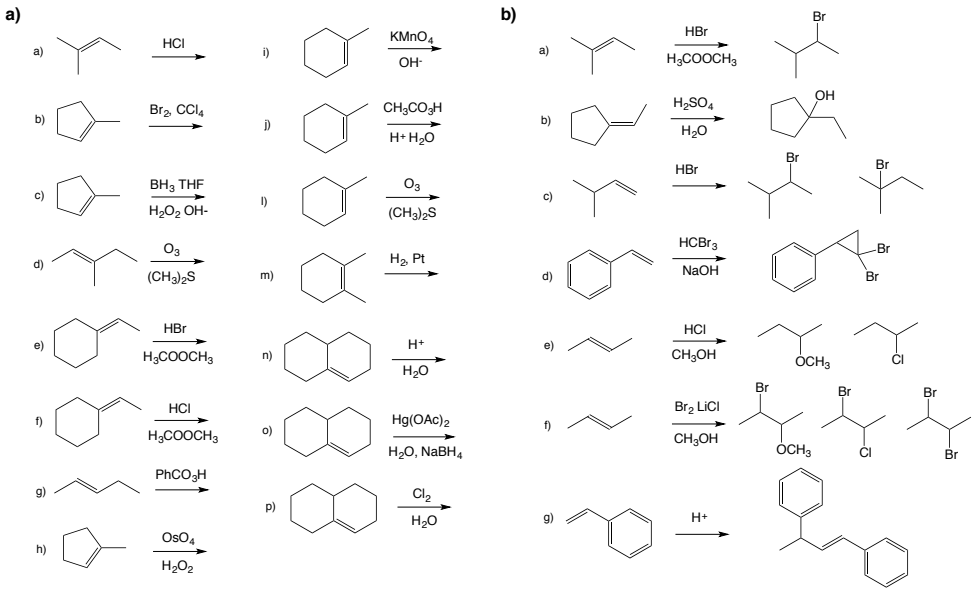


Figure 5.3: Wade problems a) 8-47 and b) 8-48

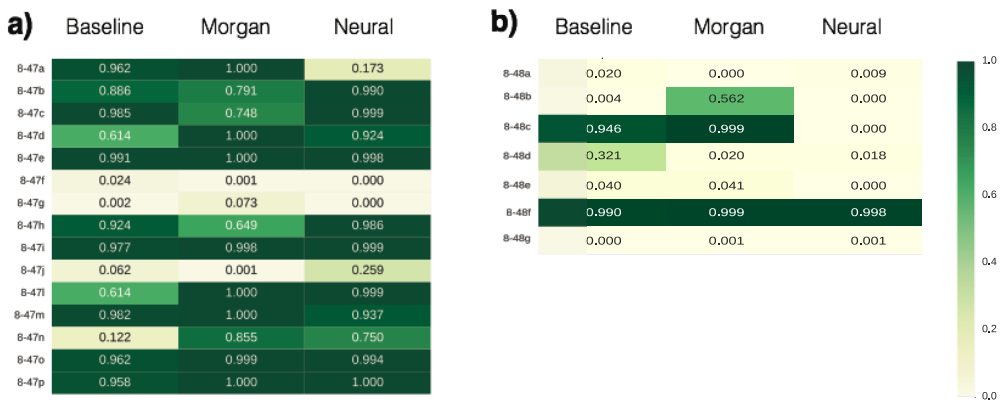


Figure 5.4: Prediction results for a) Wade Problem 8-47 and b) Wade Problem 8-48, as displayed by estimated probability of correct reaction type. Darker (greener) colors represent a higher predicted probability. Note the large amount of correct predictions in 8-47

reported below in Figure 5.4, color-coded with green for probability and yellow/white for low probability.

In problem 8-47, the Morgan fingerprint algorithm had the best performance with 12 of the 15 correct answers, followed by the neural fingerprint algorithm and the baseline method, both of which had 11 out of 15 correct answers. Both the Morgan fingerprint algorithm and the neural fingerprint algorithm predicted the correct answers with higher probability than the baseline method. Several of the problems contained rings, which weren't included in the original training set. Many of these reactions were predicted correctly by the Morgan and neural fingerprint algorithm, but not by the baseline algorithm. This suggests that both Morgan and neural fingerprint algorithms were able to extrapolate the correct reactivity patterns to reactants with rings.

In problem 8-48, students are asked to suggest mechanisms for reactions given the both the reactants and the products. To match the input format of our algorithm, we did not provide the algorithm any information about the products even though it disadvantaged our algorithm. All methods had much greater difficulty with this set of problems possibly because these problems introduced aromatic rings, which the algorithm may have had difficulty distinguishing from double bonds.

5.2.3 Performance on Product Prediction

Once a reaction type has been assigned for a given problem by our algorithm, we can use the information to help us predict our products. In this study, we chose to naively use this information by applying a SMARTS transformation that matched the predicted reaction type to generate products from reactants. Figure 5.5 shows the results of this product prediction method using Morgan reaction fingerprints and neural reaction fingerprints on problem 8-47 of the Wade textbook, analyzed in the previous section. For all suggested reaction types, the SMARTS transformation was applied to the reactants given by the problem. If the SMARTS transformation

for that reaction type was unable to proceed due to a mismatch between the given reactants and the template of the SMARTS transformation, then the reactants were returned as the predicted product instead.

A product prediction score was also assigned for each prediction method. For each reaction, the Tanimoto score[5] was calculated between the Morgan fingerprint of the true product and the Morgan fingerprint of the predicted product for each reaction type, following the same applicability rules described above. The overall product prediction score is defined as average of these Tanimoto scores for each reaction type, weighted by the probability of each reaction type as given by the probability vector. The scores for each question are given in Fig. 5.5.

The Morgan fingerprint algorithm is able to predict 8 of the 15 products correctly, the neural fingerprint algorithm is able to predict 7 of the 15 products correctly. The average Tanimoto score for the products predicted by the Morgan fingerprint algorithm compared to the true products was 0.793 and the average Tanimoto score between the true products and the neural fingerprint algorithm products was 0.776. In general, if the algorithm predicted the reaction type correctly with high certainty, the product was also predicted correctly and the weighted Tanimoto score was high, however, this was not the case for all problems correctly predicted by the algorithm.

The main limitation in the algorithm's ability to predict products despite predicting the reaction type correctly is the capability of the SMARTS transformation to accurately describe the transformation of the reaction type for all input reactants. While some special measures were taken in the code of these reactions to handle some common regiochemistry considerations, such as Markovnikov orientation, it was not enough to account for all of the variations of transformations seen in the sampled textbook questions. Future versions of this algorithm will require an algorithm better than encoded SMARTS transformations to generate the products from the reactant molecules.

	True Product	Major Predicted Product	Morgan Weighted Tanimoto Score	Neural Weighted Tanimoto Score
a			0.9998	0.3438
b			0.8863	0.9945
c			0.8554	0.9996
d			0.9999	0.9450
e			0.9999	0.9987
f			0.3540	0.3537
g			0.4296	0.4261

	True Product	Major Predicted Product	Morgan Weighted Tanimoto Score	Neural Weighted Tanimoto Score
h			0.8030	0.9921
i			0.9986	0.9991
j			0.3924	0.4026
l			0.8274	0.8270
m			0.9999	0.9627
n			0.3492	0.4029
o			0.9993	0.9957
p			0.9999	0.9999

Figure 5.5: Product predictions for Wade 8-47 questions, with Tanimoto score. The true product is the product as defined by the answer key. The major predicted product shows the product of the reaction type with the highest probability according to the Morgan fingerprint algorithm's result. The Morgan weighted score and the neural weighted score are calculated by taking an average of the Tanimoto scores over all the predicted products weighted by the probability of that reaction type which generated that product.

5.3 Conclusion

Using our fingerprint-based neural network algorithm, we were able to identify the correct reaction type for most reactions in our scope of alkene and alkylhalide reactions, given only the reactants and reagents as inputs. We achieved an accuracy of 85% of our test reactions and 80% of selected textbook questions. With this prediction of the reaction type, the algorithm was further able to guess the structure of the product for a little more than half of the problems. The main limitation in the prediction of the product structure was due to the limitations of the SMARTS transformation to describe the mechanism of the reaction type completely.

While previously developed machine learning algorithms are also able to predict the products of these reactions with similar or better accuracy[74], the structure of our algorithm allows for greater flexibillity. Our algorithm is able to learn the probabilities of a range of reaction types. To expand the scope of our algorithm to new reaction types, we would not need to encode new rules, nor

would we need to account for the varying number of steps in the mechanism of the reaction; we would just need to add the additional reactions to the training set. The simplicity of our reaction fingerprinting algorithm allows for rapid expansion of our predictive capabilities given a larger dataset of well-curated reactions[148,161]. Using datasets of experimentally published reactions, we can also expand our algorithm to account for the reaction conditions in its predictions, and later, predict the correct reaction conditions.

This paper represents a step towards the goal of developing a machine learning algorithm for automatic synthesis planning for organic molecules. Once we have an algorithm that can predict the reactions that are possible from its starting materials, we can begin to use the algorithm to string these reactions together to develop a multistep synthetic pathway. This pathway prediction can be further optimized to account for reaction conditions, cost of materials, fewest number of reaction steps and other factors to find the ideal synthetic pathway. Using neural networks helps the algorithm to identify important features from the reactant molecules structure in order to classify new reaction types.

5.4 Methods

5.4.1 Dataset Generation

The data set of reactions was developed as follows: A library of all alkanes containing 10 carbon atoms or fewer was constructed. To each alkane, a single functional group was added, either a double bond or a halide (Br, I, Cl). Duplicates were removed from this set to make the substrate library. Sixteen different reactions were considered, 4 reactions for alkylhalides and 12 reactions for alkenes. Reactions resulting in methyl shifts, or resulting in Markovnikov or anti-Markovnikov product were considered as separate reaction types. Each reaction is associated with a list of secondary reactants and reagents, as well as a SMARTS transformation to generate the product

structures from the reactants.

To generate the reactions, every substrate in the library was combined with every possible set of secondary reactants and reagents. Those combinations that matched the reaction conditions set by our expert rules, were assigned a reaction type. If none of the reaction conditions were met, the reaction was designated a 'Null Reaction' or NR for short. We generated a target probability vector to reflect this reaction type assignment with a one-hot encoding; that is, the index in the probability vector that matches the assigned reaction type had a probability of 1, and all other reaction types had a probability of 0. The notable exception to this rule was for the elimination and substitution reactions involving methyl shifts for bulky alkylhalides; these reactions were assumed to occur together, and so 50% was assigned to each index corresponding to these reactions. Products were generated using the SMARTS transformation associated with the reaction type with the two reactants as inputs. Substrates that did not match the reaction conditions were designated 'null reactions' (NR), indicating that the final result of the reaction is unknown. RDKit[125] was used to handle the requirements and the SMARTS transformation. A total of 1,277,329 alkyhalide and alkene reactions were generated. A target reaction probability vector was generated for each reaction.

5.4.2 Prediction Methods

As outlined in Figure 5.1, to predict the reaction outcomes of a given query, we first predict the probability of each reaction type in our dataset occurring, then we apply SMARTS transformations associated with each reaction. The reaction probability vector, i.e. the vector encoding the probability of all reactions, was predicted using a neural network with reaction fingerprints as the inputs. This reaction fingerprint was formed as a concatenation of the molecular fingerprints of the substrate (Reactant1), the secondary reactant (Reactant2) and the reagent. Both the Morgan fingerprint method, in particular the extended-connectivity circular fingerprint (ECFP), and the

neural fingerprint method were tested for generating the molecular fingerprints. A Morgan circular fingerprint hashes the features of a molecule for each atom at each layer into a bit vector. Each layer considers atoms in the neighborhood of the starting atom that are less than the maximum distance assigned for that layer. Information from previous layers is incorporated into later layers, until highest layer, e.g. maximum bond length radius, is reached[131]. A neural fingerprint also records atomic features at all neighborhood layers, but instead of using a hash function to record features, uses a convolutional neural network, thus creating a fingerprint with differentiable weights. Further discussion about circular fingerprints and neural fingerprints can be found in Duvenaud et al[34]. The circular fingerprints were generated with RDKit, the neural fingerprints were generated with code from Duvenaud et al[34]. The neural network used for prediction had one hidden layer of 100 units. Hyperopt[9] in conjunction with Scikit-learn[110] was used to optimize the learning rate, the initial scale, and the fingerprint length for each of the molecules.

For some reaction types, certain reagents or secondary reactants are required for that reaction. Thus, it is possible that the algorithm may learn to simply associate these components in the reaction with the corresponding reaction type. As a baseline test to measure the impact of the secondary reactant and the reagent on the prediction, we also performed the prediction with a modified fingerprint. For the baseline metric, the fingerprint representing the reaction was a one-hot vector representation for the 20 most common secondary reactants and the 30 most common reagents. That is, if one of the 20 most common secondary reactants or one of the 30 most common reagents was found in the reaction, the corresponding bits in the baseline fingerprint were turned on; if one of the secondary reactants or reagents was not in these lists, then a bit designated for 'other' reactants or reagents was turned on. This combined one-hot representation of the secondary reactants and the reagents formed our baseline fingerprint.

Once a reaction type has been predicted by the algorithm, the SMARTS transformation

associated with the reaction type is applied to the reactants. If the input reactants met the requirements of the SMARTS transformation, the products molecules generated by the transformation is the predicted structure of the products. If the reactants do not match the requirements of the SMARTS transformation, the algorithm instead guesses the structure of the reactants instead, i.e. it is assumed that no reaction occurs.

5.5 Current state of synthesis planning and reaction prediction with Machine Learning

The direction of using machine learning for reaction prediction has exploded over the past two years; many new methods and reviews have come out since the original publication of this work[24,40,147]. Reaction prediction with algorithms has regained significant attention, and I will outline some general approaches for prediction methods following the one we developed. I will discuss both synthesis planing, i.e. proposing reaction pathways for synthesizing a molecule, in addition to reaction prediction, i.e. predicting the resultant molecules given the reactants.

There are a few strategies used for reaction prediction and synthesis planning, they are as follows:

- **Predict reaction steps from reaction templates.** That is, one must first encode a set of reaction templates, and determine which template to use given the starting molecule(s). This is the technique used by the work described in this chapter. Other methods that also use this approach include the AlphaChem model of Segler et al.[145,147] and by the models developed by Coley et al. and Jin et al.[23,69].

- **Predict reaction steps using SMILES based representations of molecules.** This approach treats the task of reaction prediction/synthesis planning as a translation task. The starting 'language' is the set of starting molecules in SMILES form, and the ending 'language' set of

ending molecules, also in SMILES form. Examples of models using this strategy include Nam et al.[106] and also the sequence-to-sequence work of Liu et al.[91]

- **Predict the motions of electrons** This technique models the electron pushing mechanisms that organic chemistry students learn in their classes. This was the approach used by the ReactionPredict algorithms[17,18,73]. More recently, groups have generated electron paths using Bayesian sampling and LSTMs to model the electron steps.[13,41] This method is only been employed so far by reaction prediction methods, but not for synthesis planning methods.

All of these approaches have high accuracy, achieving 70% accuracy or more for their top choice in product prediction.

In order to further improve the predictive accuracy of these models and have it output more data, it is necessary to obtain more experimental data. It is necesary to collect information of not only reactions that are successful, or published in patent datasets, but also collect information about reactions that had low yields, or the reaction conditions which led to poor performance. One group has already started to combine machine learning techniques with data collected from both good and bad reactions to help predict the results of reactions[3,119].

5.6 Acknowledgement

The authors thank Rafael Gomez Bombarelli, Jacob Sanders, Steven Lopez, and Matthew Kayala for useful discussions. J.N.W. acknowledges support from the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1144152. D.D. and A.A.-G. thank the Samsung Advanced Institute of Technology for their research support. The authors thank FAS Research computing for their computing support and computer time in the Harvard Odyssey computer cluster. All opinions, findings and conclusions expressed in this paper are the authors'

and do not necessarily reflect the policies and views of NSF or SAIT.

6

Neural networks for predicting electron ionization-mass spectrometry spectra of small molecules

Abstract

When confronted with a substance of unknown identity, researchers often perform mass spectrometry on the sample and compare the observed spectrum to a library of previously-collected spectra to identify the molecule. While popular, this approach will fail to identify molecules that are not in the existing library. In response, we propose to improve the libraryâĂŹs coverage by augmenting it with synthetic spectra that are predicted using machine learning. We contribute a lightweight neural network model that quickly predicts mass spectra for small molecules. Achieving high accuracy predictions requires a novel neural network architecture that is designed to capture typical fragmentation patterns from electron ionization. We analyze the effects of our modeling innovations on library matching performance and compare our models to prior machine learning-based work on spectrum prediction.

6.1 Introduction

Mass spectrometry (MS) is an important tool used to identify unknown molecular samples in a variety of applications, from characterization of organic synthesis products, to pharmacokinetic studies[63], to forensic studies[186], to analyzing gaseous samples on remote satellites[111].

In electron-ionization mass spectrometry (EI-MS), molecular samples are ionized by an electron beam and broken into fragments. The resultant ions are separated by an electric field until they reach a detector. The mass spectrum is a distribution of the frequency or intensity of each type of ion, ordered by mass-to-charge (*m/z*) ratio.

A popular method for identifying a sample from its mass spectrum is to look up the sample's spectrum in a *reference library*. Here, a similarity function is used to measure the similarity between the query spectrum from the sample and each spectrum in the library. If the measurement noise when obtaining the query spectrum is reasonable, then the library spectrum with the highest similarity will have the identity of the sample.[156,158] A schematic of this process is shown in Figure 6.1a.

This library matching approach is very popular, but it suffers from a *coverage problem*: if the sample consists of a molecule that is not in the library, then correct identification is impossible. This is an issue in practice, since existing mass spectral reference libraries, such as the NIST/NIH/EPA MS database[157], Wiley Registry of Mass Spectral Data[100], and MassBank[61] only contain hundreds of thousands of reference spectra. The coverage problem could be reduced by recording spectra for additional molecules, but this is time consuming and expensive. For example, NIST releases updates to its library every 3 years, containing roughly 20,000 new spectra. Additionally, mass spectra of new molecules are only added to the library if the molecule is of common interest; molecules for newly synthesized compounds are typically not incorporated [155,157].

An alternative solution is to use *de novo* methods that input a spectrum and directly generate a molecule, without using a fixed list of molecules (Section 6.2). However, these approaches currently have low-accuracy and are difficult for practitioners to incorporate into their existing work-flows.

Another method for alleviating the coverage problem is to augment existing libraries with synthetic spectra that are generated by a model. Thus far, this approach has not been practical, as existing spectrum prediction methods are very computationally expensive. These prediction models use quantum mechanics calculations [8,53,55] or machine learning [4] to estimate the probability of each bond breaking under ionization, and thus the frequency of each ion fragment. Since these methods must either compute molecular orbital energies with high accuracy using expensive calculations, or else stochastically simulate the fragmentation of the molecule, the time needed for each model to make a prediction scales with the size of the molecule, taking up to 10 minutes for large molecules [4,8].

In response, we present Neural Electron Ionization Mass Spectrometry (NEIMS), a neural network that predicts the electron-ionization mass spectrum for a given small molecule. Since our model directly predicts spectra, instead of bond breaking probabilities, it is dramatically faster than previously reported methods, making it possible to generate predictions for thousands of possible candidates in seconds. Furthermore, the approach does not rely on specific details of EI, and thus our model could be easily retrained to predict mass spectra for other ionization methods.

We test the performance of our model by predicting mass spectra for small molecules from the NIST 2017 Mass Spectral Library. We find that the predictive capability of our model is similar to previously reported machine learning models, but requires much less time to make predictions. Additionally, we report the similarity of the spectra predicted by NEIMS. The code repository for NEIMS is publicly available at github.com/brain-research/deep-molecular-massspec.

6.2 Related Work

Several algorithms have been developed previously for either predicting spectra or for predicting the molecule's identity given the spectrum. We review some of these techniques here.

DENDRAL One of the earliest efforts in artificial intelligence was a model used to identify molecules from their mass spectrum. Heuristic DENDRAL (Dentritic Algorithm) was a collaboration between chemists and computer scientists at Stanford in the 1960s[15]. This algorithm used expert rules from chemistry to help identify patterns in the spectra and suggest possible identities for the molecule. A few years later, Meta DENDRAL was introduced to learn the expert rules that originally been given to Heuristic DENDRAL[89].

De Novo Identification Methods Several models have been reported to predict identities of samples directly from the spectrum. Many have been developed for tandem mass spectrometry, where the task is to predict the original peptide sequences from digested fragments given the mass spectrum[37]. Some of these methods use machine learning to achieve this task[143,165]. One work even uses machine learning models to identify personal characteristics by analyzing electrospray-ionization mass spectra of samples collected from human fingerprints[186].

While this approach is common for prediction of peptide sequences, it is uncommon for prediction of molecules from spectra. Several previously published models have used neural network models to predict molecule subgroups from spectra, or the class of the molecule[30,118]. One recent work attempts to employ a LSTM sequence-to-sequence model to predict the molecule directly from its mass spectrum, using the Simplified Molecular Input Line Entry Specification (SMILES) to output the molecule[118]. Because of the difficulty of constructing syntactically correct SMILES, this approach was not able to successfully reconstruct the entire SMILES string for any of the input spectra.

In this work, we focus on the prediction of spectra from molecules, such that these predicted

spectra can be used to improve the coverage of library-matching-based identification. The advantage of this approach over de novo approaches is that new libraries of synthetic spectra can be easily incorporated into the existing mass spectrometry software used by practitioners.

Quantum Mechanics Spectral Prediction Methods he first prediction methods for EI-MS spectrum used quantum mechanical simulation techniques to predict fragmentation events. There are three methods of predicting the mass spectrum using first principles[8]. The first is to use quasi-equilibrium theory, also known as Rice-Ramsberger-Kassel-Marcus theory, to estimate the rate constants for ionization reaction[93,94,132]. The second is to estimate the bond order energies within a molecule, and estimate where a molecule may fragment. A related method to this second method is to calculate the cross-section of molecular orbitals upon electron impact to predict the molecule's ionization behavior[55,64]. The third method uses Born-Oppenheimer Molecular Dynamics. Quantum Chemistry Electron-Ionization Mass Spectrometry (QCEIMS) is a particularly recent example of the ab initio molecular dynamics method[8,53,149]. The trajectories resulting from this simulation are then analyzed for the presence of ionic fragments. The distribution of the ion fragments aggregated from all the simulations is then renormalized to generate a calculated EI-MS spectrum. Each of these methods requires at least 1000 seconds per molecule[4], and may even take days or weeks for molecules of 50 atoms. While these methods may be fast for methods involving density functional theory, they do not have the speed needed to rapidly generate a collection of spectra thousands of molecules. Furtermore, some of the basis sets used for the density functional theory might not support the presence of inorganic atoms.

Machine Learning Spectral Prediction Methods Allen et al.[4] introduced Competitive Fragmentation Modelling-Electron Ioinization (CFM-EI) to predict EI-MS spectra. This probabilistic model predicts the probability of breaking molecular bonds under electron ionization, and also predicts the charged fragment that is likely to form. In order to generate the spectra, it is

necessary to run a stochastic simulation to determine the frequency of each molecular fragment. In Section 6.4.2, we directly compare this method with our proposed model.

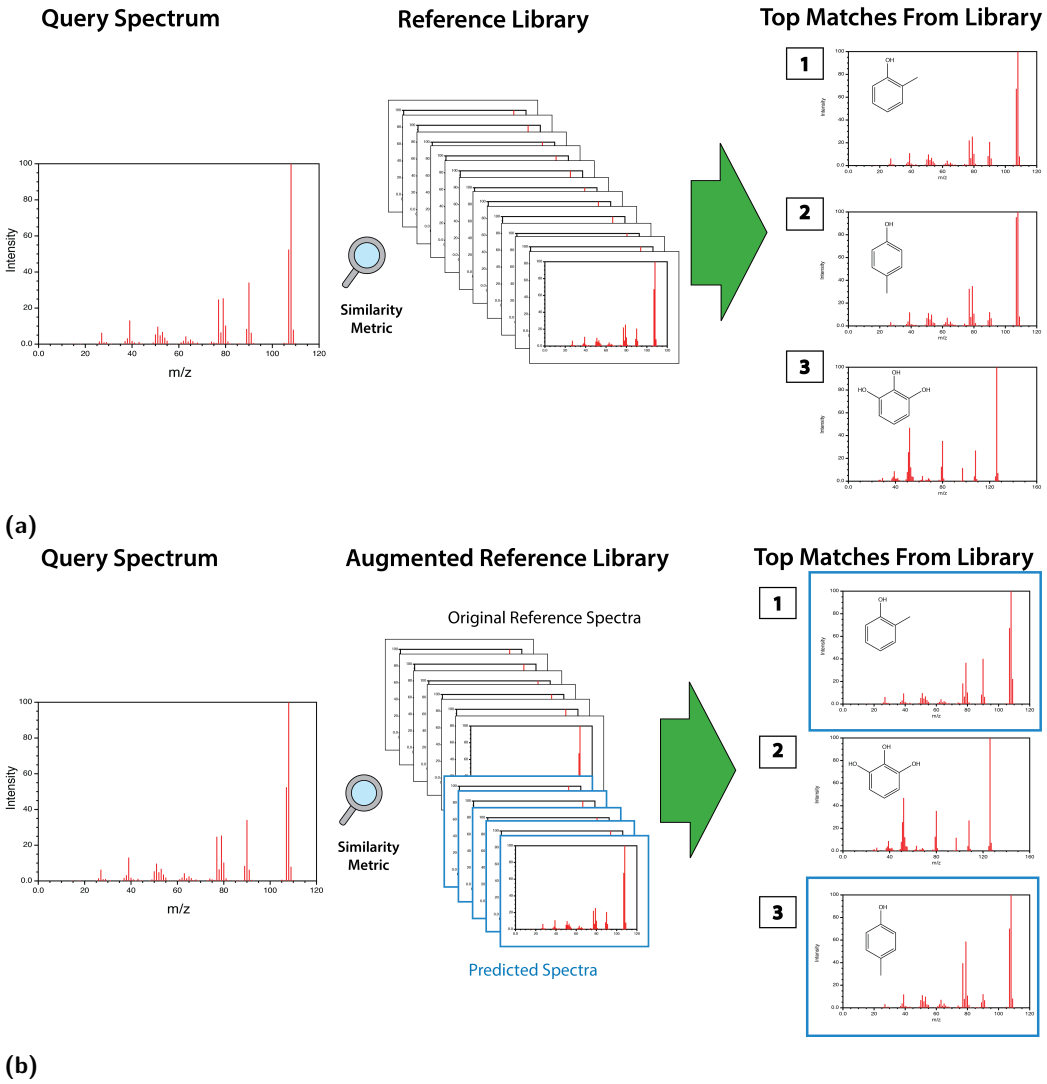


Figure 6.1: Library Matching Task. (a) A depiction of how query spectra are matched to a collection of reference spectra as performed by mass spectrometry software. (b) Query spectra are compared against a library comprised of spectra from the NIST 2017 main library and spectra predicted by our model (outlined in blue). Spectral Images adapted from NIST Webbook[90].

6.3 METHODS

Our goal is to design a model that will accurately predict the EI-MS spectrum for any molecule. This will be used to produce an augmented reference library containing both predicted spectra and experimentally-measured spectra. This task is outlined in Figure 6.1b.

We first discuss how similarity metrics for spectra in Section 6.3.1. Next, we describe our method for spectra prediction in Sections 6.3.2 and 6.3.3. We then explain how we evaluate our model's impact on the library matching task more thoroughly in Section 6.3.4.

6.3.1 SIMILARITY METRICS FOR MASS SPECTRA

The ability to a match a query spectrum from a sample to the correct spectrum in the library depends on the choice of similarity metric between spectra[99,158]. A weighted cosine similarity is commonly used by mass spectrometry software. The exact form of the cosine similarity is given below[158]:

$$\text{Similarity}(\boldsymbol{I}_q, \boldsymbol{I}_l) = \frac{\sum_{k=1}^{M_{max}} m_k I_{qk}^{0.5} \cdot m_k I_{lk}^{0.5}}{\left\| \sum_{k=1}^{M_q} (m_k I_{qk}^{0.5})^2 \right\| \left\| \sum_{k=1}^{M_l} (m_k I_{lk}^{0.5})^2 \right\|}. \tag{6.1}$$

Here, $\boldsymbol{I}_q$ and $\boldsymbol{I}_l$ are vectors of *m/z* intensities representing the query spectrum and the library spectrum respectively, m_k and I_k are the mass-to-charge ratio and intensity found at $m/z = k$, M_l and M_q are the largest indices of $\boldsymbol{I}_q$ and $\boldsymbol{I}_l$ with non-zero values,and M_{max} is the larger of M_l and M_q. The motivation for the weighting by m/z is because the peaks in mass spectra corresponding to larger fragments are more characteristic and useful in practice for identifying the true molecule.

Other similarity metrics besides cosine distance similarities are also employed. For example, one other similarity method involves estimating the relative importance of one peak given the other peaks[99]. Other methods uses a Euclidian difference between peaks, or use a variation of the

Hamming distance[60,158]. Another similarity metric accounts for neutral losses, or the intensity peaks corresponding to the loss of small, neutral fragments from the original molecular ion[103]. It is also possible to use the same form of the similarity function as in (6.1), but with different weighting given to the intensity or the masses[158]. In principle, machine learning could be also used to learn a parameterized similarity metric that yields improved library matching performance. However, this custom metric would be difficult to deploy, since it would require changing the software used by practitioners.

We develop our model with the assumption that Eq. (6.1) will be used for the similarity metric in downstream library matching software that consumes an augmented library.

6.3.2 Spectral Prediction

We treat the prediction of mass spectrometry spectra as a multi-dimensional regression task. The output of our model is a vector that represents the intensity at every integral *m/z* bin. We use this discretization granularity for *m/z* because it is what is provided in the NIST datasets we use for training our model.

In the NEIMS model (Figure 6.3), we first map molecules to additive Extended Circular Fingerprints (ECFPs)[125]. These fingerprints are similar to their binary counterparts[129] in that they record molecular subgraphs made up from local neighborhoods around each atom node in the molecule, but differ in that they count the occurrences for each subgroup. This information is then hashed into a vector representation. The difference is that additive fingerprints record the frequency that each bit is set, rather than just the presence. The RDKit Cheminformatics package[125] was used to generate the fingerprints. These features are then passed into a multi-layer perceptron neural network (MLP). To account for some of the physical phenomena of ionization, we make some application-specific adjustments to the prediction from the MLP, described in Section 6.3.3.

In Section 6.4.1 we compare the performance of NEIMS to that of a simple linear regression

(LR) model. Here, we apply a linear transformation to the ECFP features.

To train the model, we use a modified mean-squared-error loss function. This loss function, shown below, follows the same weighting pattern as in Eq. 6.1:

$$L(\boldsymbol{I}, \hat{\boldsymbol{I}}) = \sum_{k=1}^{M(x)} \left(\frac{m_k I_k^{0.5}}{\left\| \sum_{k=1}^{M} (m_k I_k^{0.5})^2 \right\|} - \frac{m_k \hat{I}_k^{0.5}}{\left\| \sum_{k=1}^{M} (m_k \hat{I}_k^{0.5})^2 \right\|} \right)^2 \tag{6.2}$$

where I is the ground truth spectrum, $\hat{I}$ is the predicted spectrum, and $M(x)$ is the mass of the input molecule. We used stochastic gradient descent to optimize the parameters of the MLP with the Adam optimizer[78]. We use Tensorflow[1] to construct and train the model.

6.3.3 Adjustments for Physical Phenomena

In practice, we have found that the conventional MLP described in the previous section struggles to accurately predict the right-hand side of spectra (Figure 6.2a). Errors in this region, which correspond to large m/z, are particularly damaging for library matching with the weighting in (6.1).

This section introduces a revised neural network architecture (Figure 6.3) designed to better model the underlying fragmentation process that occurs in mass spectrometry. We have found that it improves prediction in the high mass region of the spectrum (Figure 6.2b), which yields improvements in library matching (Section 6.4.1).

As is standard for MLPs used for regression, the predictions of the above MLP model on an input molecule x are an affine transformation of a set of features $f(x)$, which are computed by all but the final layer of the network. For reasons that will become apparent, we refer to the above MLP as performing *forward* prediction. At bin $m/z = i$, we have the following predicted intensity:

$$p_i^f(x) = w_i^{f\top} f(x) + b_i^f, \tag{6.3}$$

where w_i^f and b_i^f are the model's weights and biases for forward prediction at bin i.

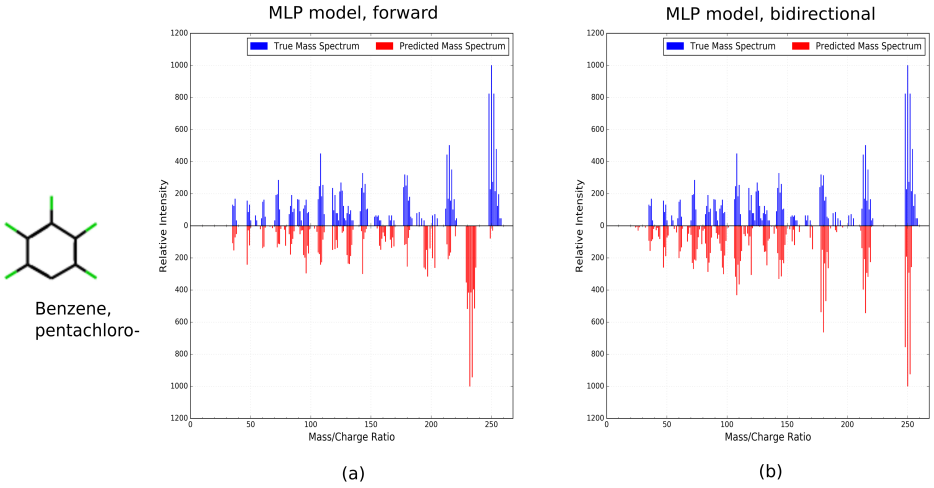


Figure 6.2: Spectral Prediction with MLP forward Model (a) and MLP bidirectional Model (b). For both spectra plots, the true spectrum is shown in blue on top, while the predicted spectrum is shown inverted in red. Note that the spectrum predicted by the bidirectional model shows fewer stray peaks than the forward model, particularly for larger m/z values. These peaks are much easier to predict with the reverse prediction mode.

The input ECFP features, from which $f(x)$ is computed, capture local structures in the molecule, so generally $f(x)$ will be more accurate in capturing the presence of small substructures of molecule x. Often, there is a direct correspondence between the presence of such substructures and spectral peaks with small m/z. For example, in Figure 6.2a a peak occurs at $m/z = 35$, due to the presence of chlorine. Therefore an accurate forward prediction model will have a learned weight w_{35} that will output a high intensity at $i = 35$ if there is evidence in $f(x)$ for the presence of chlorine.

On the other hand, forward prediction often struggles to accurately predict intensities for large fragments that are the result of neutral losses[156]. One reason for this is that the composition of large fragments is not captured well by the ECFP representation. Another reason is that information learned about the cleavage of a small group does not transfer well across molecules of different masses. For pentachlorobenzene, which has a molecular mass of 250 Da, the fragment

that results from the loss of a neutral chlorine atom results in a peak at 215 Da. Meanwhile, for chlorobenzene, which has a mass of 112 Da, the fragment resulting from a loss of a chlorine atom would have a peak at 77 Da. Despite the clear relationship between these intensity peaks, the forward model is not parameterized to capture this pattern.

In response, following the physical phenomenon that created the fragments, we define larger ion peaks as a function of the residual groups that were broken off from the original molecule. Referring to our previous example of pentachlorobenzene ($M(x) = 250$), we can parameterize the m/z ratio of the fragment which lost a chlorine group as $m/z = 250 - 35 = 215$. The corresponding fragment in chlorobenzene would have a mass of $m/z = 215 - 35 = 77$. By defining the peaks in this way, it is possible for these predictions of spectral intensities to be linked by the prediction at index 35. This leads to the indexing scheme of our *reverse prediction* model:

$$p^r_{M(x)+\tau-i}(x) = w_i^{r\top} f(x) + b^r_i, \tag{6.4}$$

Here, $\tau > 0$ is a small shift that allows for peaks to occur at intensities greater than $M(x)$, due to isotopes. In practice, reverse prediction is implemented using a copy of the forward model, with separate sets of parameters for the final affine layer, but shared parameters for $f(x)$. The outputs of this model are post-processed on a per-molecule basis to obey the indexing in (6.4), which depends on each molecule's mass.

Both the forward and reverse predictions are combined to form a *bidirectional* prediction. That is, the final prediction at index i is a combination of both p^f_i and p^r_i. In the case of pentachlorobenzene, the prediction of spectral intensity at $m/z = 215$ is a function of p^f_{215} from the forward mode and $p^r_{35+\tau}$ from the reverse mode. Instead of simply averaging the two prediction modes, we have found that small additional performance improvements can be obtained using a

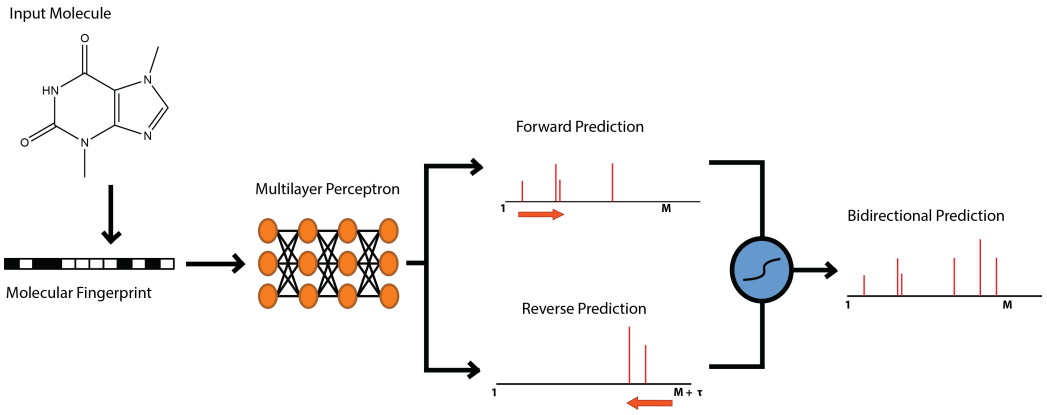


Figure 6.3: Molecular representations are passed into a multilayer perceptron to generate an initial output. This output is used to make a forward prediction starting at $m/z = 0$ and $m/z = M$ and in reverse starting from $m/z = M$ and ending at $m/z = 0$. A sigmoid gating is applied to the inputs as shown in Eq. 6.5

coordinate-wise *gate*. Here, the output $p_i(x)$ at position i is given by:

$$p_i(x) = \sigma(\text{gate}_i)\ p_i^f(x) + (1 - \sigma(\text{gate}_i))\ p_i^r(x), \tag{6.5}$$

where gate_i is an affine transformation of $f(x)$ and $\sigma(\cdot)$ is a sigmoid function. This approach echoes the formulation of the Hybrid Similarity Search designed by Moorthy et al., which accounts for peaks that are created by small fragment ions and those which are created by large fragments which have lost smaller groups [103].

Finally, for all models, we zero out predicted intensities at m/z that are greater than $M(x) + \tau$.

By adding these features, we incorporate some of the physical phenomena that occur in mass spectrometry into our model while maintaining the overall simplicity of the MLP. In this way, we are able to predict the spectrum directly without resorting to sampling bond-breaking events within the molecule, which requires subsequent stochastic sampling to obtain a spectrum.

6.3.4 Library Matching Evaluation

We evaluate NEIMS using an *augmented reference library* consisting of a combination of observed spectra and model-predicted spectra, with library matching performance computed with respect to a *query set* of spectra. These are from the NIST 2017 replicates library, which is a collection of noisier spectra for molecules that are contained in the NIST main library. The inconsistencies in these spectra reflect experimental variation, and make an informative dataset to test our model's performance.

To construct the augmented reference library, we edit the NIST main library, removing spectra corresponding to the query set molecules and replacing them with the predictions from NEIMS. We then perform library matching and calculate the similarity between each query spectrum and every spectrum from the augmented library. We record the rank of the correct spectrum, i.e. the rank of the predicted spectrum corresponding to the molecule which made the query spectrum. The similarity metric is Eq. (6.1).

For the purposes of tuning model hyperparameters, we chose to optimize recall@10, i.e. the percentage of our query set for which the correct spectra had a matching rank of less than or equal to 10 in the library matching task. Half of the replicates library was used for tuning hyperparameters, and the remaining half was used to evaluate test performance. All models were trained on the spectra prediction task for 100,000 training steps with a batch size of 100.

During the library match search, we have a *mass filtering* option. This feature reduces the library size so it only includes spectra from molecular candidates that have a molecular mass that differ by a few Daltons from the mass of the query molecule. If the EI-MS analysis is combined with mass spectrometry techniques using weak ionization methods, it is possible to determine the mass of molecule being analyzed. In the CFM-EI model, the molecular formula is used to filter the search library [4]. Using the molecular mass to filter the library allows more possible candidate spectra to

be considered in the search than using a molecular formula filter.

6.4 Results and Discussion

To analyze the performance of the models, we trained with 240,942 spectra from the NIST 2017 Mass Spectral Main Library. These spectra were selected so that no molecules in the replicates library have spectra in the training set.

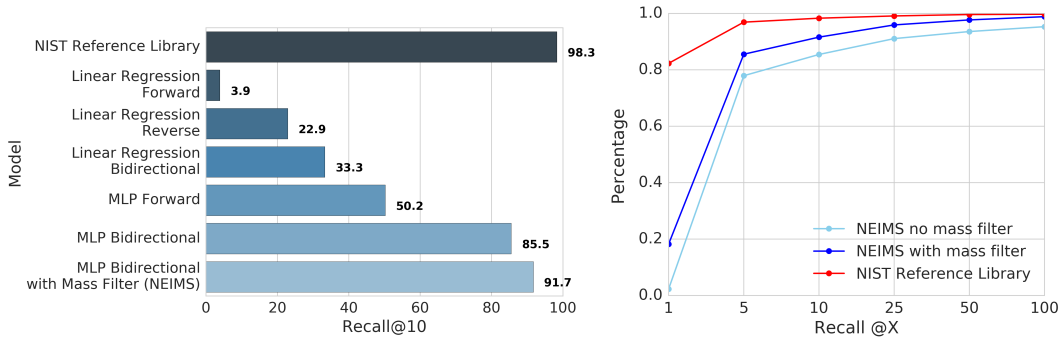
After hyperparameter tuning using Vizier[48], we found that the optimal MLP architecture has seven layers of 2000 nodes, with residual network connections between the layers[58], using ReLU activation and a dropout rate of 0.25.

6.4.1 Library Matching Results

We first examine the effects of our various modeling decisions on performance. Figure 6.4a compares the performance of forward, reverse, and bidirectional versions of the linear regression and MLP models on the library matching task. For bidirectional prediction in the linear regression model, the forward and reverse predictions are simply averaged together, rather than applying the gate described in (6.5).

The top row of Figure 6.4a shows that it is not possible to achieve perfect recall accuracy on the library matching task even when using the full NIST main library as the reference library, without any model-predicted spectra. Observing Figure 6.4b we see that using the NIST main library as the reference library, we have 86% recall@1 accuracy, and 98.3% recall@10 accuracy. This serves as a practical upper bound on achievable library matching accuracy and reflects the experimental inconsistencies between between the main library spectra and replicates spectra[155].

The forward prediction mode for both the linear regression model and the multilayer perceptron (MLP) has poor performance. The linear regression model is improved by 20% when switching to using reverse mode prediction. Using bidirectional prediction mode improves recall@10 accuracy



(a) Library matching performance on different models **(b)** Recall Results at various levels

Figure 6.4: Performance of different model architectures.

by 30% for both the linear regression and the multilayer perceptron model. This finding suggests that the bidirectional prediction mode is more effective at capturing the fragmentation events than the forward-only model.

Figure 6.2 shows the improvement in spectral prediction for pentachlorobenzene using the bidirectional MLP model. Note that the bidirectional model on the right more accurately models intensities at larger *m/z*. The intensity peaks for larger *m/z* are critical for determining the identity of a molecule, and are more heavily weighted in Eq. (6.1).

NEIMS achieves 91.7% recall@10 after applying a mass filter. The mass filter was set to a tolerance of 5 Daltons of the query molecule's mass; this reduces the size of the library to a median of 6,696 spectra for each query molecule. In practice, this tolerance window could be set to a larger window, depending on the uncertainty of the information about the molecular mass of the ion. For the rest of this report, we will refer to the bi-directional multi-layer perceptron model with mass filtering of 5 Daltons as the default settings for NEIMS.

From Figure 6.4b we see that while NEIMS has decent performance for recall levels of 10 and above compared to the NIST spectral library, it has considerably worse performance for recall values of 1 and 5. This result is unsurprising given that the hyperparmaters of the model were trained to maximize performance on recall@10. If recall@1 was instead selected to tune the

Model	Recall@1	Recall@10 (%)	Average run time (ms)
NIST '14 Reference Library	77	99*	–
CFM-EI	42.6	89*	300,000
NEIMS	54.3	92.7	0.47

Table 6.1: Performance on Library matching task for NIST 17. * indicates that values were estimated from Figure 4 of Allen et al.[4]

hyperparameters, the performance accuracy on recall@1 would improve.

6.4.2 Comparison to previously reported models

We next compared our model's performance directly to the performance of the CFM-EI model[4]. The setup of Allen et al. differs from our current setup in a few ways. First, they evaluate their model on the NIST '14 spectral library. Second, for the library matching task, their augmented reference library contains only spectra predicted by their model, and none from the original NIST collection. Third, the cosine similarity metric Eq. (6.1) used for evaluation in library matching in CFM-EI uses a different weighting scheme. In their analysis, the cosine similarity is weighted by $m_k^{0.5}$ instead of m_k in order to de-emphasize the larger peaks in the mass spectrum, as they ran their experiments on other datasets with a higher proportion of larger molecules[4].

To compare the performance of NEIMS to that of CFM-EI, we match their setup identically. We retrain our NEIMS model on the NIST 14 dataset, and evaluate the performance using the NIST 14 replicates as the query set. For library matching, we incorporate only predicted spectra into our augmented library, and using the same modified similarity metric.

The library matching performance for CFM-EI and NEIMS are compared against the NIST14 library for library matching performance are reported in Table 6.1. NEIMS performs slightly better than CFM-EI on the library matching task. More importantly, NEIMS is able to make spectral predictions orders of magnitude faster than CFM-EI. With NEIMS, it would be possible to generate spectra for 1 million molecules in 90 min on a CPU, with potential for considerable

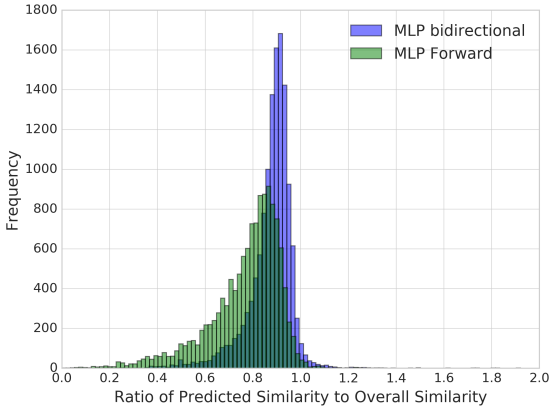


Figure 6.5: Comparing the similiarity between the predicted spectrum and the ground truth spectrum to the overall similarity between spectra for the same molecule.

speedup with using GPU.

6.4.3 Distances between predicted and ground truth spectra

So far, we have evaluated the quality of the NEIMS predictions indirectly, by way of how they affect library matching with an augmented library. Next, we assess the prediction accuracy directly, by measuring the similarity (Eq. 6.1) between spectra in the NIST main library and the model's predictions. We refer to this similarity as the *predicted similarity*.

There is inherent noise in mass spectra due to stochasticity of the underlying physical process and also to experimental inconsistencies [155]. The NIST replicates library provides multiple spectra for each molecule, and we can use these sets of spectra to characterize the scale of this noise for each molecule. Specifically, we define the inherent noise for a given molecule as the average pairwise similarity between all corresponding spectra, both in the NIST main library and the NIST replicates library, and refer to this as the *overall similarity*.

For each molecule, we compute the ratio of the predicted similarity to overall similarity as a normalized metric for the quality of our predictions. A ratio of 1.0 would suggest that there are is limited available headroom for improvements using machine learning, since the model's errors are

comparable to the variability in the data.

Figure 6.5 shows the improvement in this ratio for the MLP bidirectional model over the MLP forward model, confirming that the bidirectional model has better spectral prediction performance. For the MLP bidirectional model, roughly half of the molecules have a predicted similarity to overall similarity ratio that is greater than 0.9, indicating that there is potential for further improvement to the model. Some of these molecules have ratios that are greater than 1, which is possible if there is more variation between the spectra (i.e. a lower overall similarity) than between the predicted spectrum and the main library spectrum (i.e. predicted similarity).

6.5 Conclusion

We demonstrate that NEIMS achieves high library matching performance on an augmented spectral library containing predictions for molecules in the query set. The performance of NEIMS is also slightly better than existing machine learning models for predicting EI-MS spectra, with significant boost in speed of prediction.

The high performance in library matching is attributable to the bidirectional prediction mode. The reverse mode in particular allows the model to more accurately to predict intensities for larger fragments which result from the loss of small neutral subgroups. We observe that the improvement in the library matching task also corresponds with improvement in the similarity of the predicted spectra to the ground truth spectra.

Several adjustments could be made to further improve NEIMS. For example, NEIMS currently does not have a method to model intensity peaks corresponding to isotopes in ion fragments. If we were to train on spectral data with greater precision in the peaks locations, we might be able to learn the exact identities of the atoms based on the decimal values of the *m/z* peak locations.

Mass filtering improved the performance of NEIMS by 6%. This suggests that for experimental

setups where it is possible to know the molecular mass of the sample with some accuracy, it is possible to improve the accuracy of matching on the augmented spectral library. It would also be interesting to explore other settings for mass filtering, such as filtering out spectra which have a molecular mass that is much smaller than the position of the largest *m/z* peak.

Different molecular representations could also be tested. The predictions made from ECFP are limited by the descriptiveness of the fingerprint[84]. In particular, the overlap in representation for different molecular features represents a huge limitation to the representation of the molecule. Additionally, ECFPs are not equipped to represent molecules with multiple stereocenters, which will have different spectra. It would also be interesting to explore whether a bond-based molecular fingerprint representation[75] or other graph-based molecular representations[33,47] may improve performance.

Combining NEIMS with transfer learning methods could allow for spectral prediction specific to individual spectrometry machines. A library of such machine-specific spectra would improve matching[155].

The lightweight framework of NEIMS makes it possible to rapidly generate spectral predictions for large numbers of molecular candidates. This collection of predicted spectra can then be used directly in mass spectrometry software to expand the coverage of molecules which can be identified by mass spectrometry. Because the requirements of NEIMS has limited dependence to EI mass spectrometry, it likely that some of the principles used here could be extended to other types of mass spectrometry.

6.6 Acknowledgments

We thank Stephen Stein for fruitful discussions about mass spectrometry and for providing helpful feedback on this manuscript. We thank Laura Castellanos for her insights about mass spectrometry.

We thank Steven Kearnes for his helpful comments, and Lucy Colwell and Michael Brenner for their helpful conversations.

7

Future Directions

Machine learning has already begun to revolutionize the development of new materials. Recent successes in machine learning have lead to the development of novel blue OLED molecules[49].

The methods that I present in Section II are a collection of early works into the next generation of this direction. New molecules can be proposed from generative model, similar to those proposed in Chapter 4. The reactivity of these molecules can be predicted with the methods presented in Chapter 5. Newly synthesized molecules can be verified by with spectroscopy; machine learning can be used to aid identification by expanding the coverage of existing libraries, as shown in Chapter 6.

I would like to close my dissertations with three main areas I see for improvement in the development of machine learning models for chemistry applications.

Better datasets and benchmarks for testing machine learning models. Standardized, publicly available datasets are needed for the community as the whole to develop new models to push the capacity of machine learning models. Datasets for molecules are available, and contain molecules which reflect molecules that are commonly used for drugs. Datasets for reactions are much more limited. There is one publicly available dataset for reactions, the USPTO dataset[95]. However this dataset contains only successful reactions, with numerous issues with the standardization of the data. Some works have published their datasets splits, which is very helpful for reproducing results. Ideally however, we would have more datasets, which would include more

details of the reaction conditions, as well as examples of non-working reactions.

Additionally, there is a need for additional benchmarks for comparing generative models. It is difficult to measure the relative progress of generative models without some method of comparing the quality of these datasets. At the time of writing, three works have recently been released towards this goal[14,114,115].

Better molecular representations. The current representations for molecules described in Chapter 3.1 have been successfully employed in machine learning models to predict a wide range of properties in chemistry. However, there are some issues with this representations. The SMILES representation has a few issues when combined with the variational autoencoder. The purpose of a variational autoencoder is to group similar objects close together in the latent space. However, SMILES strings that might be similar in terms of edit distances may not actually be close in molecular space: *c1ccccc1* (benzene) will have very different properties from *c1ccccn1* (pyridine). Additionally, even when grammar restraints are incorporated into the generation of the molecule string[82], the model does not have a sense of which molecules are feasible in terms of their valence, and which molecules are not.

It is therefore necessary to have models which can generate graphs from a vector representation. Several papers have been developed towards this direction at the time of writing[68,92,96,183].

However, none of these representations are invariant to graph isomorphism. That is, a molecule which was formed starting from one atom will be considered different from a molecule that was formed starting from another atom. While it is possible to train these representations to be equivalent, it may lead to inefficiencies in the model. While there are several methods such as graph convolutional networks for encoding from a graph, there are limited methods for decoding to a molecule based representation.

Better communication between machine learning experts and chemists. As demonstrated in

the mass spectrometry project of Chapter 6, the best models arise when the design of the model is inspired by the physical characteristics of the problems. By reparametrizing the output from the neural network to account for larger fragments, we were able to improve the performance of spectral reconstructions significantly. Such design decisions can only be reached when there is constant dialogue between the two domains. That way, we can ensure that the models that are developed in the machine learning community are both as accurate and feasible as they can be. This can also help tailor existing models to the needs of individual groups and projects.

With the development of new machine learning models and further developments in chemistry, it will be possible to further accelerate the discovery and development of new molecular materials.

A

Appendix for Part II: Machine Learning Applications to Chemistry

A.1 Supplementary Information for Chpater 4: Variational Autoencoders for Optimization in Molecular Space

This section contains peripheral findings including statistics on the latent space, reconstruction accuracy, training robustness with respect to dataset size, and more sampling interpolation examples.

Table A.1.1: Percentage of successfully decoding of latent representation after 1000 attempts for 1000 molecules from the traning set, 1000 validation molecules randomly chosen from ZINC and a 1000 validation molecules randomly chosen from eMolecules. Both VAEs perform very well for training data, and they are well transferable within molecules of the same class outside the training data, as evidence by the good validation performance of the ZINC VAE and the underperformance of the QM9 VAE against real-life small molecules.

Dataset	ZINC	QM
Training set	92.1	99.6
Test set	90.7	99.4
ZINC	91.0	1.4
eMolecules	83.8	8.8

[h]

Table A.1.2: Percentage of 5000 randomly-selected latent points that decode to valid molecules after 1000 attempts

Dataset	ZINC	QM
Decoding probability	73.9	79.3

Table A.1.3: Variational autoencoder performance over different sizes of datasets. Training and tests were performed using randomly selected molecules from the ZINC dataset, the values reported here are the scores from the validation set. The categorical accuracy reflects the percentage of characters in the output SMILES that were accurately reconstructed. Mean Absolute Errors (MAE) are reported for QED and logP properties. Performance significantly decreases if only 10^5 molecules are used for training.

Training set size	Categorical Accuracy	logP MAE	QED MAE
225,000	99.3%	0.15	0.054
175,000	99.0%	0.18	0.076
125,000	98.5%	0.15	0.076
25,000	91.6%	0.23	0.079

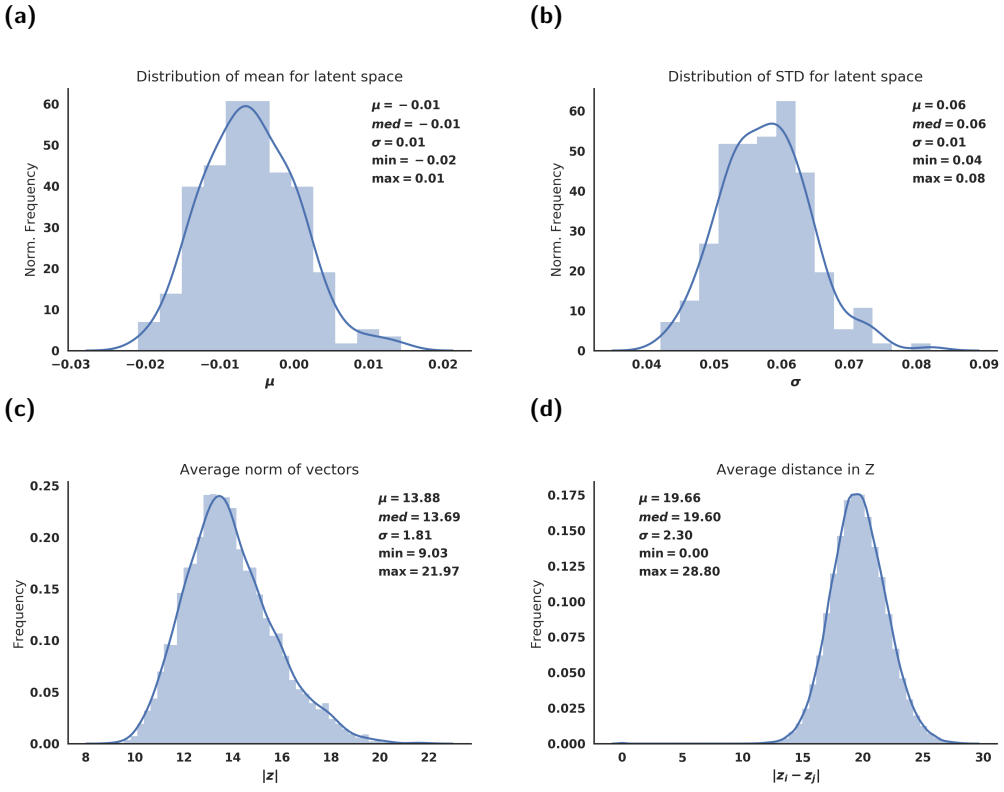


Figure A.1.1: Distribution and statistics of (a) the mean of latent space coordinates (b) standard deviation of latent space coordinates (c) norm of latent space coordinates of the encoded representation of randomly selected molecules from the ZINC validation set. (d) Distribution of Euclidean distances between random pairs of validation molecules in the ZINC VAE

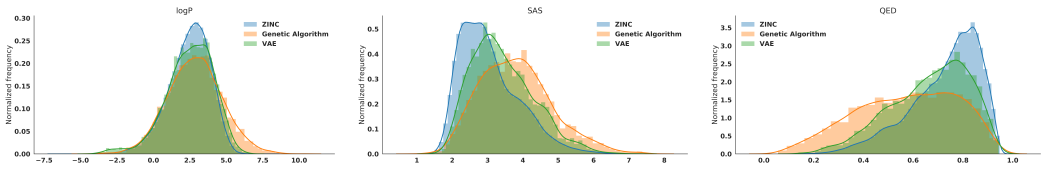


Figure A.1.2: Histograms and KDE plots of the distribution of properties utilized in the jointly trained autoencoder (LogP, SAS, QED). Used to further showcase results from Table 2. For each property we compare the distribution of the source data (ZINC), a generatic algorithm and the VAE.

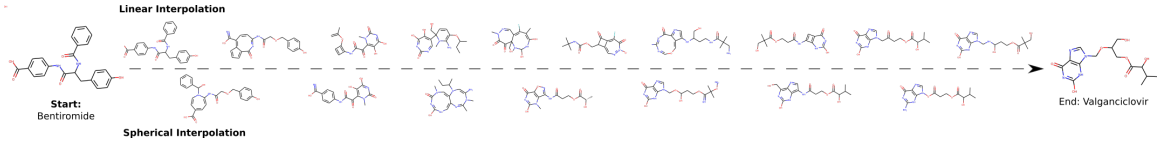


Figure A.1.3: Comparison of between linear and spherical interpolation paths between two randomly selected FDA approved drugs. A constant step size was used.

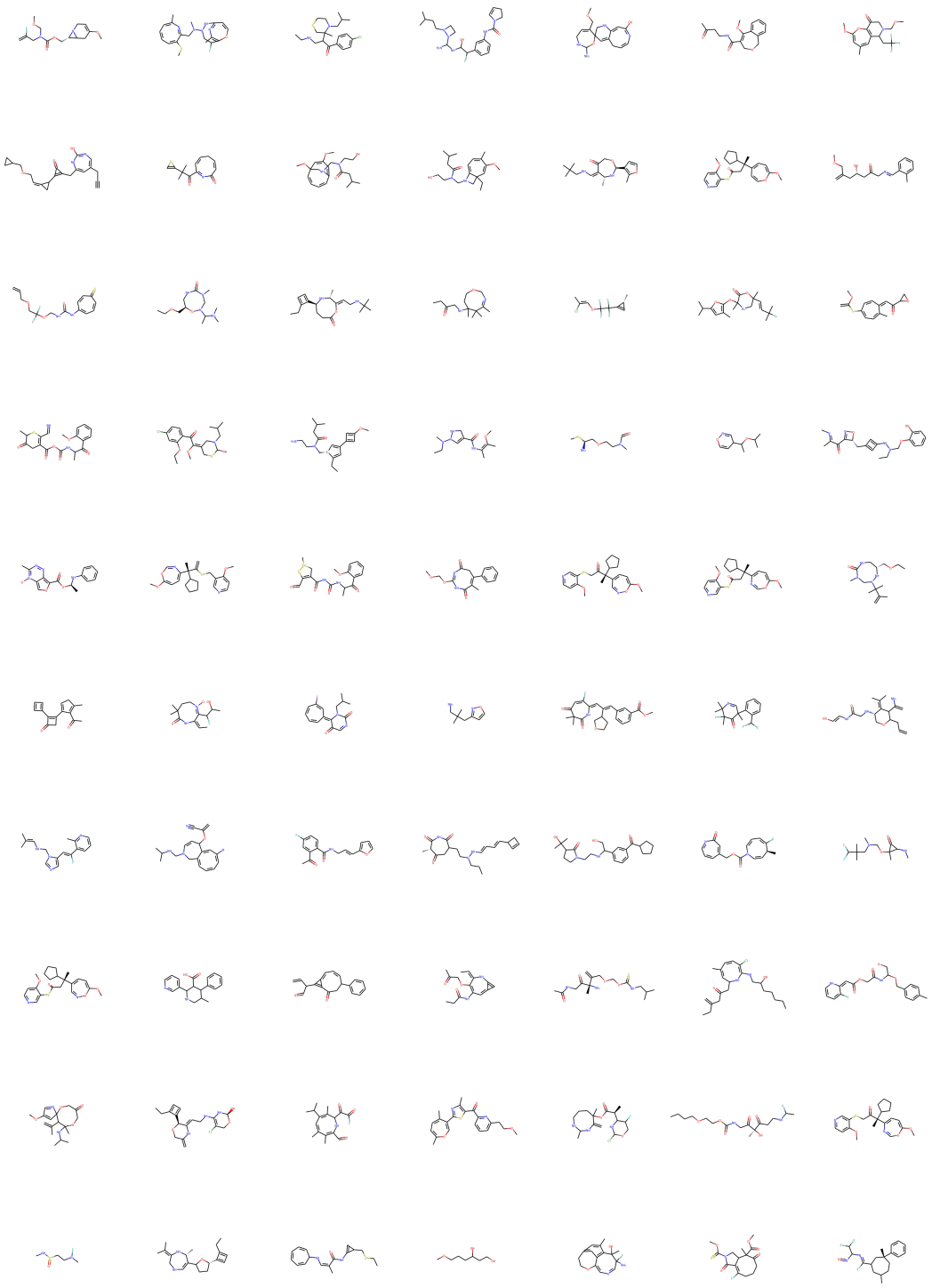


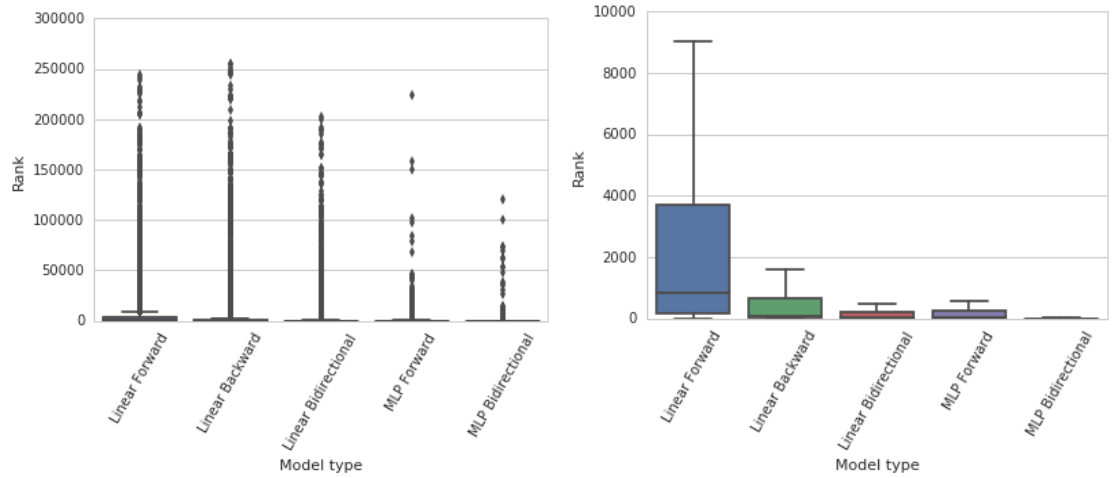
Figure A.1.4: Molecules decoded from randomly-sampled points in the latent space of the ZINC VAE.

A.2 Supplementary Information for Chapter 5: Neural Networks for Predicting Reactions

Table A.2.4: The Tanimoto similarity of the training set to problems Wade 8-47 and 8-48 used in the main article.

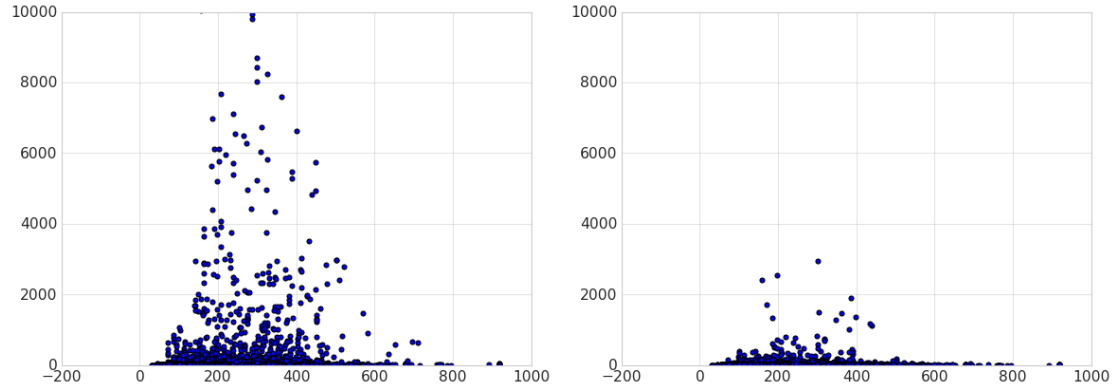
Problem number	Average Training Set Tanimoto Similarity	Highest Training Set Tanimoto Similarity
8-47a	0.30	0.94
8-47b	0.42	0.74
8-47c	0.47	0.86
8-47d	0.41	0.76
8-47e	0.47	0.88
8-47f	0.47	0.88
8-47g	0.35	0.65
8-47h	0.42	0.75
8-47i	0.43	0.80
8-47j	0.48	0.76
8-47l	0.44	0.77
8-47m	0.43	0.82
8-47n	0.45	0.75
8-47o	0.44	0.75
8-47p	0.44	0.76
8-48a	0.42	0.78
8-48b	0.42	0.77
8-48c	0.31	1.00
8-48d	0.34	0.54
8-48e	0.19	0.71
8-48f	0.20	0.66
8-48g	0.33	0.48

A.3 Supplementary Information for Chapter 6: Neural Networks for Predicting Electron-Ionization Mass Spectrometry of Small Molecules



(a) **(b)**

Figure A.3.5: The boxplot for all of the library matching ranks are shown in these figures. (a) shows the box plot distribution excluding outliers, while (b) shows the box plot distribution for all results including outliers.



(a) **(b)**

Figure A.3.6: Library matching ranks (y-axis) for all query spectra based on molecular mass (x-axis). (a) Shows the resulting ranks without the use of the mass filter, while (b) shows the resulting ranks with the mass filter. The matching rank improves significantly by using the mass filter, as indicated by the clustering of data towards the x-axis. This is especially the case for those spectra that are in the center of the mass range.

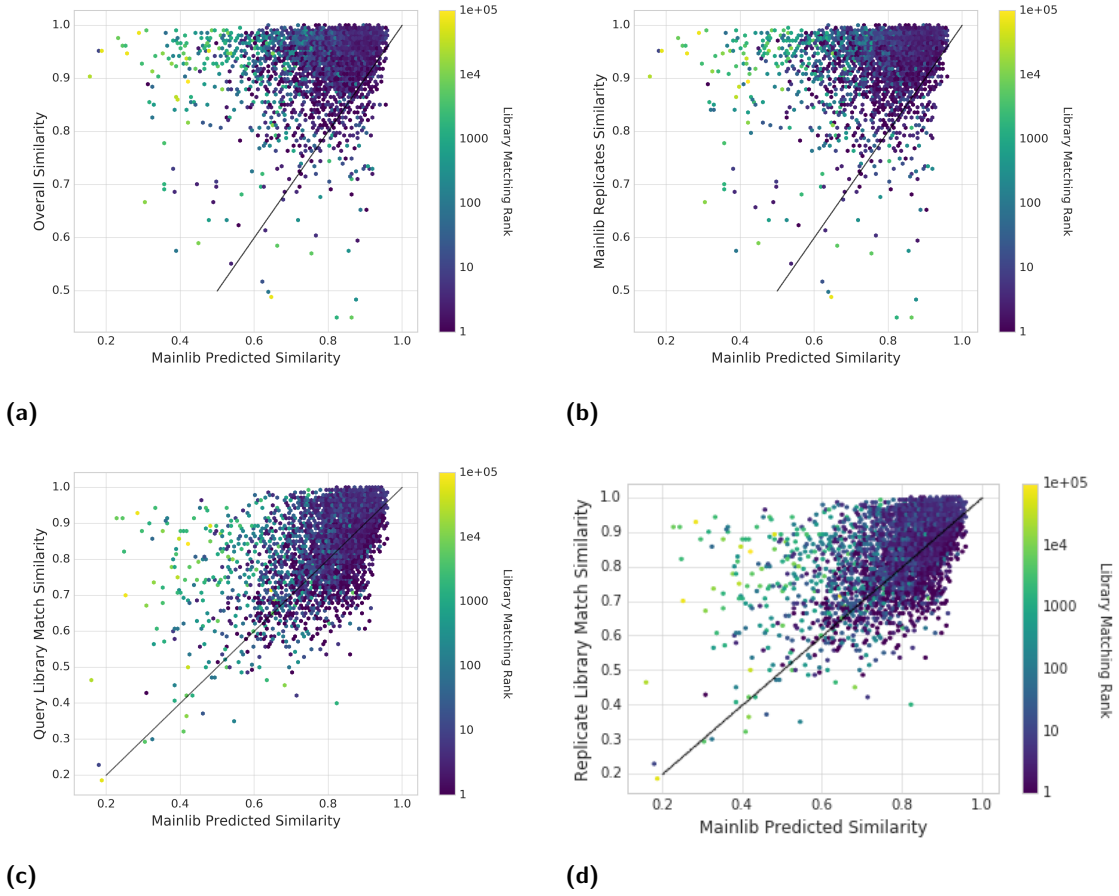


Figure A.3.7: The above shows a collection of similarity plots for each molecule in the query set. The plots show a hex-bin heatmap, with Library matching rank representing the color of the hexbin for all plots. measuring the distances between spectra. (a) shows the distribution between Overall Similarity (i.e. the similarity between all recorded spectra for the same molecule) and predicted similarity. The ratio between these values is shown in Figure 6.5. (b) compares the predicted similarity against the similarity of the main spectrum to the replicates spectra for the same molecule. This gives an indication of similar the predicted spectra is compared to the rest of the replicate spectra. (c) shows the distribution between the Predicted Similarity (x-axis) and the Query Library Match similarity. This gives an indication of how much more similar the query spectrum is to the library matched spectrum compared against the predicted similarity. (d) compares the similarity between the predicted similarity and the similarity between the replicates spectra to the library match spectra. The similarity between the replicates spectra and the library matched spectra gives an indication of the noise in the spectra for the molecule, and how that might have affect on the similarity matching.

Bibliography

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[2] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory — ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings*, pages 420–434. Springer, Berlin, Heidelberg, 2001.

[3] D. T. Ahneman, J. G. Estrada, S. Lin, S. D. Dreher, and A. G. Doyle. Predicting reaction performance in c–n cross-coupling using machine learning. *Science*, **360**, 186 (2018).

[4] F. Allen, A. Pon, R. Greiner, and D. Wishart. Computational prediction of electron ionization mass spectra to assist in gc/ms compound identification. *Analytical chemistry*, **88**, 7689 (2016).

[5] D. Bajusz, A. Rácz, and K. Héberger. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J Cheminform*, **7**, 1 (2015).

[6] D. Balamurugan, W. Yang, and D. N. Beratan. Exploring chemical space with discrete, gradient, and hybrid optimization methods. *J. Chem. Phys.*, **129**, 174105 (2008).

[7] P. J. Ballester and J. B. O. Mitchell. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics*, **26**, 1169 (2010).

[8] C. A. Bauer and S. Grimme. How to compute electron ionization mass spectra from first principles. *The Journal of Physical Chemistry A*, **120**, 3755 (2016).

[9] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox. Hyperopt: a Python library for model selection and hyperparameter optimization. *Comput. Sci. Disc.*, **8**, 014008 (2015).

[10] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins. Quantifying the chemical beauty of drugs. *Nat. Chem.*, **4**, 90 (2012).

[11] T. Blaschke, M. Olivecrona, O. Engkvist, J. Bajorath, and H. Chen. Application of generative autoencoder in de novo molecular design. *Molecular informatics*, **36**, 1700123 (2017).

[12] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating Sentences from a Continuous Space, 2015.

[13] J. Bradshaw, M. J. Kusner, B. Paige, M. H. Segler, and J. M. Hernández-Lobato. Predicting electron paths. arXiv preprint arXiv:1805.10970 (2018).

[14] N. Brown, M. Fiscato, M. H. Segler, and A. C. Vaucher. Guacamol: Benchmarking models for de novo molecular design. arXiv preprint arXiv:1811.09621 (2018).

[15] B. G. Buchanan and E. A. Feigenbaum. Dendral and meta-dendral: Their applications dimension. In *Readings in artificial intelligence*, pages 313–322. Elsevier, 1981.

[16] R. E. Carhart, D. H. Smith, and R. Venkataraghavan. Atom pairs as molecular features in structure-activity studies: definition and applications. *J. Chem. Inf. Model.*, **25**, 64 (1985).

[17] J. H. Chen and P. Baldi. Synthesis explorer: A chemical reaction tutorial system for organic synthesis design and mechanism prediction. *J. Chem. Educ.*, **85**, 1699 (2008).

[18] J. H. Chen and P. Baldi. No electron left behind: A rule-based expert system to predict chemical reactions and reaction mechanisms. *J. Chem. Inf. Model.*, **49**, 2034 (2009). PMID: 19719121.

[19] T. Cheng, Q. Li, Z. Zhou, Y. Wang, and S. H. Bryant. Structure-Based Virtual Screening for Drug Discovery: a Problem-Centric Review. *AAPS J.*, **14**, 133 (2012).

[20] F. Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[21] S. Chonde and S. Kumara. Cheminformatics: An introductory review. In *IIE Annual Conference. Proceedings*, page 2316. Institute of Industrial and Systems Engineers (IISE), 2014.

[22] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[23] C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green, and K. F. Jensen. Prediction of organic reaction outcomes using machine learning. *ACS central science*, **3**, 434 (2017).

[24] C. W. Coley, W. H. Green, and K. F. Jensen. Machine learning in computer-aided synthesis planning. *Accounts of chemical research*, **51**, 1281 (2018).

[25] E. Corey, W. J. Howe, H. Orf, D. A. Pensak, and G. Petersson. General methods of synthetic analysis. Strategic bond disconnections for bridged polycyclic structures. *J. Am. Chem. Soc.*, **97**, 6116 (1975).

[26] E. Corey, A. K. Long, T. W. Greene, and J. W. Miller. Computer-assisted synthetic analysis. Selection of protective groups for multistep organic syntheses. *J. Org. Chem.*, **50**, 1920 (1985).

[27] E. Corey, W. T. Wipke, R. D. Cramer III, and W. J. Howe. Computer-assisted synthetic analysis. Facile man-machine communication of chemical structure by interactive computer graphics. *J. Am. Chem. Soc.*, **94**, 421 (1972).

[28] E. Corey, W. T. Wipke, R. D. Cramer III, and W. J. Howe. Techniques for perception by a computer of synthetically significant structural features in complex molecules. *J. Am. Chem. Soc.*, **94**, 431 (1972).

[29] E. J. Corey. Centenary lecture. computer-assisted analysis of complex synthetic problems. *Q. Rev. Chem. Soc.*, **25**, 455 (1971).

[30] B. Curry and D. E. Rumelhart. Msnet: A neural network which classifies mass spectra. *Tetrahedron Computer Methodology*, **3**, 213 (1990).

[31] P. Domingos and Pedro. A few useful things to know about machine learning. *Commun. ACM*, **55**, 78 (2012).

[32] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, **42**, 1273 (2002).

[33] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. *Adv. Neural. Inf. Process. Syst. 28*, pages 2224–2232 (2015).

[34] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Gómez-Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems*, pages 2215–2223, 2015.

[35] E-molecules. `https://www.emolecules.com/info/plus/download-database`. [Online; accessed 22-July-2017].

[36] M. Eickenberg, G. Exarchakis, M. Hirn, and S. Mallat. Solid harmonic wavelet scattering: Predicting quantum molecular energy from invariant descriptors of 3d electronic densities. In *Advances in Neural Information Processing Systems*, pages 6540–6549, 2017.

[37] J. K. Eng, A. L. McCormack, and J. R. Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, **5**, 976 (1994).

[38] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi. Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders, Apr. 2017.

[39] P. Ertl and A. Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminf.*, **1**, 1 (2009).

[40] F. Feng, L. Lai, and J. Pei. Computational chemical synthesis analysis and pathway design. *Frontiers in chemistry*, 6 (2018).

[41] D. Fooshee, A. Mood, E. Gutman, M. Tavakoli, G. Urban, F. Liu, N. Huynh, D. Van Vranken, and P. Baldi. Deep learning for chemical reaction prediction. *Molecular Systems Design & Engineering* (2018).

[42] K. Fukushima. Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, **36**, 193 (1980).

[43] J. Gasteiger. Solved and unsolved problems of chemoinformatics. *Molecular informatics*, **33**, 454 (2014).

[44] J. Gasteiger. Cheminformatics: Computing target complexity. *Nature chemistry*, **7**, 619 (2015).

[45] J. Gasteiger, M. G. Hutchings, B. Christoph, L. Gann, C. Hiller, P. Löw, M. Marsili, H. Saller, and K. Yuki. A new treatment of chemical reactivity: Development of EROS, an expert system for reaction prediction and synthesis design. In *Organic Synthesis, Reactions and Mechanisms*, pages 19–73. Springer, Berlin, Heidelberg, 1987.

[46] H. Gelernter, J. R. Rose, and C. Chen. Building and refining a knowledge base for synthetic organic chemistry via the methodology of inductive and deductive machine learning. *J. Chem. Inf. Model.*, **30**, 492 (1990).

[47] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural Message Passing for Quantum Chemistry. *ArXiv e-prints* (2017).

[48] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. E. Karro, and D. Sculley, editors. *Google Vizier: A Service for Black-Box Optimization*, 2017.

[49] R. Gómez-Bombarelli, J. Aguilera-Iparraguirre, T. D. Hirzel, D. Duvenaud, D. Maclaurin, M. A. Blood-Forsythe, H. S. Chae, M. Einzinger, D.-G. Ha, T. Wu, G. Markopoulos, S. Jeon, H. Kang, H. Miyazaki, M. Numata, S. Kim, W. Huang, S. I. Hong, M. Baldo, R. P. Adams, and A. Aspuru-Guzik. Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nature Materials*, **15**, 1120 (2016).

[50] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[51] C. M. Gothard, S. Soh, N. A. Gothard, B. Kowalczyk, Y. Wei, B. Baytekin, and B. A. Grzybowski. Rewiring Chemistry: Algorithmic Discovery and Experimental Validation of One-Pot Reactions in the Network of Organic Chemistry. *Agnew. Chem. Int. Ed.*, **124**, 8046 (2012).

[52] C. M. Gothard, S. Soh, N. A. Gothard, B. Kowalczyk, Y. Wei, B. Baytekin, and B. A. Grzybowski. Rewiring Chemistry: Algorithmic Discovery and Experimental Validation of One-Pot Reactions in the Network of Organic Chemistry. *Agnew. Chem. Int. Ed.*, **124**, 8046 (2012).

[53] S. Grimme. Towards first principles calculation of electron impact mass spectra of molecules. *Angewandte Chemie International Edition*, **52**, 6306 (2013).

[54] B. A. Grzybowski, K. J. M. Bishop, B. Kowalczyk, and C. E. Wilmer. The 'wired' universe of organic chemistry. *Nat. Chem.*, **1**, 31 (2009).

[55] M. Guerra, F. Parente, P. Indelicato, and J. P. Santos. Modified binary encounter Bethe model for electron-impact ionization. *ArXiv e-prints* (2013).

[56] G. L. Guimaraes, B. Sanchez-Lengeling, P. L. C. Farias, and A. Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. arXiv:1705.10843 (2017).

[57] J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. S. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A. M. Brockway, and A. Aspuru-Guzik. The Harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid. *J. Phys. Chem. Lett.*, **2**, 2241 (2011).

[58] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *ArXiv e-prints* (2015).

[59] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *ICCV*. Institute of Electrical & Electronics Engineers (IEEE), dec 2015.

[60] H. S. Hertz, R. A. Hites, and K. Biemann. Identification of mass spectra by computer-searching a file of known spectra. *Analytical Chemistry*, **43**, 681 (1971).

[61] H. Horai, M. Arita, S. Kanaya, Y. Nihei, T. Ikeda, K. Suwa, Y. Ojima, K. Tanaka, S. Tanaka, K. Aoshima, et al. Massbank: a public repository for sharing mass spectral data for life sciences. *Journal of mass spectrometry*, **45**, 703 (2010).

[62] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, **2**, 359 (1989).

[63] Y. Hsieh and W. A. Korfmacher. Increasing speed and throughput when using hplc-ms/ms systems fordrug metabolism and pharmacokinetic screening. *Current Drug Metabolism*, **7**, 479 (2006).

[64] K. K. Irikura. Ab initio computation of energy deposition during electron ionization of molecules. *The Journal of Physical Chemistry A*, **121**, 7751 (2017).

[65] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, and R. G. Coleman. Zinc: A free tool to discover chemistry for biology. *J. Chem. Inf. Model.*, **52**, 1757 (2012).

[66] D. Janz, J. van der Westhuizen, and J. M. Hernández-Lobato. Actively Learning what makes a Discrete Sequence Valid, Aug. 2017.

[67] N. Jaques, S. Gu, D. Bahdanau, J. M. Hernández-Lobato, R. E. Turner, and D. Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*, pages 1645–1654, 2017.

[68] W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. arXiv preprint arXiv:1802.04364 (2018).

[69] W. Jin, C. Coley, R. Barzilay, and T. Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2607–2616. Curran Associates, Inc., 2017.

[70] W. L. Jorgensen, E. R. Laird, A. J. Gushurst, J. M. Fleischer, S. A. Gothe, H. E. Helson, G. D. Paderes, and S. Sinclair. CAMEO: a program for the logical prediction of the products of organic reactions. *Pure Appl. Chem.*, **62**, 1921 (1990).

[71] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A Convolutional Neural Network for Modelling Sentences, 2014.

[72] I. Y. Kanal, S. G. Owens, J. S. Bechtel, and G. R. Hutchison. Efficient computational screening of organic polymer photovoltaics. *J. Phys. Chem. Lett.*, **4**, 1613 (2013).

[73] M. A. Kayala, C.-A. Azencott, J. H. Chen, and P. Baldi. Learning to predict chemical reactions. *J. Chem. Inf. Model.*, **51**, 2209 (2011). PMID: 21819139.

[74] M. A. Kayala and P. Baldi. ReactionPredictor: Prediction of Complex Chemical Reactions at the Mechanistic Level Using Machine Learning. *J. Chem. Inf. Model.*, **52**, 2526 (2012).

[75] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, **30**, 595 (2016).

[76] S. Kim, A. Jinich, and A. Aspuru-Guzik. Multidk: A multiple descriptor multiple kernel approach for molecular discovery and its application to organic flow battery electrolytes. *Journal of chemical information and modeling*, **57**, 657 (2017).

[77] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang, and S. H. Bryant. PubChem Substance and Compound databases. *Nucleic Acids Res.*, **44**, D1202 (2016).

[78] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *ArXiv e-prints* (2014).

[79] D. P. Kingma and M. Welling. Auto-encoding Variational Bayes, 2013.

[80] A. N. Kolmogorov. On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. *Doklady Akademii. Nauk USSR,*, **114**, 679 (1965).

[81] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Adv. Neural. Inf. Process. Syst. 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[82] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato. Grammar Variational Autoencoder, Mar. 2017.

[83] D. Kuzminykh, D. Polykovskiy, A. Kadurin, A. Zhebrak, I. Baskov, S. Nikolenko, R. Shayakhmetov, and A. Zhavoronkov. 3d molecular representations based on the wave transform for convolutional neural networks. *Molecular pharmaceutics* (2018).

[84] G. Landrum. Collding bits, 2014.

[85] J. Law, Z. Zsoldos, A. Simon, D. Reid, Y. Liu, S. Y. Khew, A. P. Johnson, S. Major, R. A. Wade, and H. Y. Ando. Route Designer: A Retrosynthetic Analysis Tool Utilizing Automated Retrosynthetic Rule Generation. *J. Chem. Inf. Model.*, **49**, 593 (2009).

[86] P. Le Couteur and J. Burreson. *Napoleon's buttons: 17 molecules that changed history*. Penguin, 2004.

[87] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[88] Y. LeCun, L. Bottou, G. Orr, and K. Muller. Efficient backprop. In G. Orr and M. K., editors, *Neural Networks: Tricks of the trade*. Springer, 1998.

[89] R. K. Lindsay, B. G. Buchanan, E. A. Feigenbaum, and J. Lederberg. Dendral: a case study of the first expert system for scientific hypothesis formation. *Artificial intelligence*, **61**, 209 (1993).

[90] P. J. Linstrom and W. G. Mallard. The nist chemistry webbook: A chemical data resource on the internet. *Journal of Chemical & Engineering Data*, **46**, 1059 (2001).

[91] B. Liu, B. Ramsundar, P. Kawthekar, J. Shi, J. Gomes, Q. Luu Nguyen, S. Ho, J. Sloane, P. Wender, and V. Pande. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Central Science*, **3**, 1103 (2017).

[92] Q. Liu, M. Allamanis, M. Brockschmidt, and A. L. Gaunt. Constrained graph variational autoencoders for molecule design. *CoRR*, abs/1805.09076 (2018).

[93] J. Lorquet. Whither the statistical theory of mass spectra? *Mass Spectrometry Reviews*, **13**, 233 (1994).

[94] J.-C. Lorquet. Landmarks in the theory of mass spectra. *International Journal of Mass Spectrometry*, **200**, 43 (2000).

[95] D. M. Lowe. *Extraction of chemical structures and reactions from the literature*. PhD thesis, University of Cambridge, 2012.

[96] T. Ma, J. Chen, and C. Xiao. Constrained Generation of Semantically Valid Graphs via Regularizing Variational Autoencoders. *ArXiv e-prints*, page arXiv:1809.02630 (2018).

[97] M. Mann, H. Ekker, and C. Flamm. The graph grammar library-a generic framework for chemical graph rewrite systems. In *International Conference on Theory and Practice of Model Transformations*, pages 52–53. Springer, 2013.

[98] A. Mauri, V. Consonni, M. Pavan, and R. Todeschini. Dragon software: An easy approach to molecular descriptor calculations. *Match*, **56**, 237 (2006).

[99] F. McLafferty, R. Hertel, and R. Villwock. Probability based matching of mass spectra. rapid identification of specific compounds in mixtures. *Journal of Mass Spectrometry*, **9**, 690 (1974).

[100] F. W. Mclafferty. *Wiley Registry of Mass Spectral Data*. John Wiley and Sons, 11th edition, 2016.

[101] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wiestra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, **518**, 529 (2015).

[102] G. Montavon, K. Hansen, S. Fazli, M. Rupp, F. Biegler, A. Ziehe, A. Tkatchenko, A. V. Lilienfeld, and K.-R. Müller. Learning invariant representations of molecules for atomization energy prediction. In *Advances in Neural Information Processing Systems*, pages 440–448, 2012.

[103] A. S. Moorthy, W. E. Wallace, A. J. Kearsley, D. V. Tchekhovskoi, and S. E. Stein. Combining fragment-ion and neutral-loss matching during mass spectral library searching: A new general purpose algorithm applicable to illicit drug identification. *Analytical chemistry*, **89**, 13261 (2017).

[104] H. Morgan. The Generation of a Unique Machine Description for Chemical Structure. *J. Chem. Doc.*, **5**, 107 (1965).

[105] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning (ICML)*, 2010.

[106] J. Nam and J. Kim. Linking the Neural Machine Translation and the Prediction of Organic Chemistry Reactions. *ArXiv e-prints* (2016).

[107] R. Nilakantan, N. Bauman, J. S. Dixon, and R. Venkataraghavan. Topological torsion: a new molecular descriptor for sar applications. comparison with other descriptors. *Journal of Chemical Information and Computer Sciences*, **27**, 82 (1987).

[108] N. M. OâĂŹBoyle, C. M. Campbell, and G. R. Hutchison. Computational design and selection of optimal organic photovoltaic materials. *J. Phys. Chem. C*, **115**, 16200 (2011).

[109] M. Pavlov, S. Kolesnikov, and S. M. Plis. Run, skeleton, run: skeletal model in a physics-based simulation. *ArXiv e-prints* (2017).

[110] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825 (2011).

[111] S. Petrie and D. K. Bohme. Ions in space. *Mass Spectrometry Reviews*, **26**, 258 (2006).

[112] Y. Podolyan, M. A. Walters, and G. Karypis. Assessing Synthetic Accessibility of Chemical Compounds Using Machine Learning Methods. *J. Chem. Inf. Model.*, **50**, 979 (2010).

[113] P. G. Polishchuk, T. I. Madzhidov, and A. Varnek. Estimation of the size of drug-like chemical space based on GDB-17 data. *J. Comput.-Aided Mol. Des.*, **27**, 675 (2013).

[114] D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, A. Kadurin, S. Nikolenko, A. Aspuru-Guzik, and A. Zhavoronkov. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *ArXiv e-prints*, page arXiv:1811.12823 (2018).

[115] K. Preuer, P. Renz, T. Unterthiner, S. Hochreiter, and G. Klambauer. Fréchet chemblnet distance: A metric for generative models for molecules. *CoRR*, abs/1803.09518 (2018).

[116] E. O. Pyzer-Knapp, K. Li, and A. Aspuru-Guzik. Learning from the Harvard Clean Energy Project: The Use of Neural Networks to Accelerate Materials Discovery. *Adv. Funct. Mater.*, **25**, 6495 (2015).

[117] E. O. Pyzer-Knapp, C. Suh, R. Gómez-Bombarelli, J. Aguilera-Iparraguirre, and A. Aspuru-Guzik. What is high-throughput virtual screening? a perspective from organic materials discovery. *Annu. Rev. Mater. Res.*, **45**, 195 (2015).

[118] T. Rabinowitz. Mass-spectrometry-prediction. https://github.com/terryrabinowitz/Mass-Spectrometry-Prediction/blob/master/readme.pdf, 2017.

[119] P. Raccuglia, K. C. Elbert, P. D. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, and A. J. Norquist. Machine-learning-assisted materials discovery using failed experiments. *Nature*, **533**, 73 (2016).

[120] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015.

[121] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, **1**, 140022 (2014).

[122] M. Randic, M. Novic, and D. Plavsic. *Solved and unsolved problems of structural chemistry*. CRC Press, 2016.

[123] D. Rappoport, C. J. Galvin, D. Y. Zubarev, and A. Aspuru-Guzik. Complex Chemical Reaction Networks from Heuristics-Aided Quantum Chemistry. *J. Chem. Theory Comput.*, **10**, 897 (2014).

[124] C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[125] RDKit: Open-source cheminformatics. `http://www.rdkit.org`. [Online; accessed 11-April-2017].

[126] J.-L. Reymond. The Chemical Space Project. *Acc. Chem. Res.*, **48**, 722 (2015).

[127] J.-L. Reymond, R. van Deursen, L. C. Blum, and L. Ruddigkeit. Chemical space as a source for new drugs. *Med. Chem. Commun.*, **1**, 30 (2010).

[128] J.-L. Reymond, R. Van Deursen, L. C. Blum, and L. Ruddigkeit. Chemical space as a source for new drugs. *Med. Chem. Comm.*, **1**, 30 (2010).

[129] D. Rogers and M. Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, **50**, 742 (2010). PMID: 20426451.

[130] D. Rogers and M. Hahn. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.*, **50**, 742 (2010).

[131] D. Rogers and M. Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, **50**, 742 (2010). PMID: 20426451.

[132] H. M. Rosenstock, M. Wallenstein, A. Wahrhaftig, and H. Eyring. Absolute rate theory for isolated systems and the mass spectra of polyatomic molecules. *Proceedings of the National Academy of Sciences*, **38**, 667 (1952).

[133] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, **323**, 533 (1986).

[134] C. Rupakheti, A. Virshup, W. Yang, and D. N. Beratan. Strategy To Discover Diverse Optimal Molecules in the Small Molecule Universe. *J. Chem. Inf. Model.*, **55**, 529 (2015).

[135] M. Rupp, A. Tkatchenko, K.-R. MÃijller, and O. A. von Lilienfeld. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.*, **108**, 058301 (2012).

[136] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, **108**, 058301 (2012).

[137] B. Sanchez-Lengeling, C. Outeiral, G. L. Guimaraes, and A. Aspuru-Guzik. Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC), 8 2017.

[138] H. Satoh and K. Funatsu. Further Development of a Reaction Generator in the SOPHIA System for Organic Reaction Prediction. Knowledge-Guided Addition of Suitable Atoms and/or Atomic Groups to Product Skeleton. *J. Chem. Inf. Comput. Sci.*, **36**, 173 (1996).

[139] K. Satoh and K. Funatsu. A Novel Approach to Retrosynthetic Analysis Using Knowledge Bases Derived from Reaction Databases. *J. Chem. Inf. Comput. Sci.*, **39**, 316 (1999).

[140] J. Schmidhuber. Deep Learning in Neural Networks: An Overview. *ArXiv e-prints* (2014).

[141] G. Schneider. Virtual screening: an endless staircase? *Nat. Rev. Drug Discov.*, **9**, 273 (2010).

[142] N. Schneider, D. M. Lowe, R. A. Sayle, and G. A. Landrum. Development of a Novel Fingerprint for Chemical Reactions and Its Application to Large-Scale Reaction Classification and Similarity. *J. Chem. Inf. Model.*, **55**, 39 (2015).

[143] S. S. Schoenholz, S. Hackett, L. Deming, E. Melamud, N. Jaitly, F. McAllister, J. O'Brien, G. Dahl, B. Bennett, A. M. Dai, and D. Koller. Peptide-Spectra Matching from Weak Supervision. *ArXiv e-prints* (2018).

[144] T. Scior, A. Bender, G. Tresadern, J. L. Medina-Franco, K. Martinez-Mayorga, T. Langer, K. Cuanalo-Contreras, and D. K. Agrafiotis. Recognizing Pitfalls in Virtual Screening: A Critical Review. *J. Chem. Inf. Model.*, **52**, 867 (2012).

[145] M. Segler, M. Preuß, and M. P. Waller. Towards" alphachem": Chemical synthesis planning with tree search and deep neural network policies. arXiv preprint arXiv:1702.00020 (2017).

[146] M. H. Segler, T. Kogej, C. Tyrchan, and M. P. Waller. Generating focussed molecule libraries for drug discovery with recurrent neural networks, 2017.

[147] M. H. Segler, M. Preuss, and M. P. Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, **555**, 604 (2018).

[148] C. A. Service. Chemplanner. `'http://www.chemplanner.com/'`.

[149] V. ÃĄsgeirsson, C. A. Bauer, and S. Grimme. Quantum chemical calculation of electron ionization mass spectra for general organic and inorganic molecules. *Chem. Sci.*, **8**, 4879 (2017).

[150] B. K. Shoichet. Virtual screening of chemical libraries. *Nature*, **432**, 862 (2004).

[151] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, **529**, 484 (2016).

[152] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Neural Information Processing Systems 25*, 2012.

[153] I. M. Socorro, K. Taylor, and J. M. Goodman. ROBIA: A Reaction Prediction Program. *Org. Lett.*, **7**, 3541 (2005).

[154] N. Srivastava. Improving neural networks with dropout. Master's thesis, University of Toronto, 2013.

[155] S. Stein. Mass spectral reference libraries: an ever-expanding resource for chemical identification. *Analytical Chemistry*, **84**, 7274 (2012).

[156] S. E. Stein. Chemical substructure identification by mass spectral library searching. *Journal of the American Society for Mass Spectrometry*, **6**, 644 (1995).

[157] S. E. Stein. National Institute of Standards and Technology (NIST) Mass Spectral Database, 2017.

[158] S. E. Stein and D. R. Scott. Optimization and testing of mass spectral library search algorithms for compound identification. *Journal of the American Society for Mass Spectrometry*, **5**, 859 (1994).

[159] Stephan C. Schurer, and Prashant Tyagi, and and Steven M. Muskal. Prospective exploration of synthetically feasible, medicinally relevant chemical space. *J. Chem. Inf. Model.*, **45**, 239 (2005). PMID: 15807484.

[160] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[161] S. SzymkuÄĞ, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk, and B. A. Grzybowski. Computer-Assisted Synthetic Planning: The End of the Beginning. *Agnew. Chem. Int. Ed.*, **55**, 5904 (2016).

[162] D. Tabor, R. GÃşmez-Bombarelli, L. Tong, R. G. Gordon, M. J. Aziz, and A. Aspuru-Guzik. Theoretical and experimental investigation of the stability limits of quinones in aqueous media: Implications for organic aqueous redox flow batteries, Aug 2018.

[163] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[164] M. H. Todd. Computer-aided organic synthesis. *Chem. Soc. Rev.*, **34**, 247 (2005).

[165] N. H. Tran, X. Zhang, L. Xin, B. Shan, and M. Li. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, **114**, 8247 (2017).

[166] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio, 2016.

[167] R. van Deursen and J.-L. Reymond. Chemical Space Travel. *ChemMedChem*, **2**, 636 (2007).

[168] A. M. Virshup, J. Contreras-García, P. Wipf, W. Yang, and D. N. Beratan. Stochastic Voyages into Uncharted Chemical Space Produce a Representative Library of All Possible Drug-Like Compounds. *J. Am. Chem. Soc.*, **135**, 7296 (2013).

[169] O. A. von Lilienfeld, R. Ramakrishnan, M. Rupp, and A. Knoll. Fourier series of atomic radial distribution functions: A molecular fingerprint for machine learning models of quantum chemical properties. *Int. J. Quantum Chem.*, **115**, 1084 (2015).

[170] L. G. Wade. Organic chemistry. In *Organic Chemistry*, page 377. Pearson, Upper Saddle River, NJ, USA, 6th, international edition, 2013.

[171] L.-P. Wang, R. T. McGibbon, V. S. Pande, and T. J. Martinez. Automated Discovery and Refinement of Reactive Molecular Dynamics Pathways. *J. Chem. Theory Comput.*, **12**, 638 (2016).

[172] L.-P. Wang, A. Titov, R. McGibbon, F. Liu, V. S. Pande, and T. J. Martínez. Discovering chemistry with an ab initio nanoreactor. *Nat. Chem.*, **6**, 1044 (2014).

[173] M. Wang, X. Hu, D. N. Beratan, and W. Yang. Designing molecules by optimizing potentials. *J. Am. Chem. Soc.*, **128**, 3228 (2006).

[174] W. A. Warr. Twenty five years of progress in cheminformatics. In *229th American Chemical Society National Meeting*, 2005.

[175] D. Weininger. SMILES, a chemical language and information system. *J. Chem. Inf. Comput. Sci.*, **28**, 31 (1988).

[176] D. Weininger. SMILES a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.*, **28**, 31 (1988).

[177] T. White. Sampling Generative Networks, 2016.

[178] S. A. Wildman and G. M. Crippen. Prediction of physicochemical parameters by atomic contributions. *J. Chem. Inf. Comput. Sci.*, **39**, 868 (1999).

[179] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, **1**, 270 (1989).

[180] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. Moleculenet: A benchmark for molecular machine learning, 2017.

[181] L. Xu, C. E. Doubleday, and K. Houk. Dynamics of 1, 3-Dipolar Cycloaddition Reactions of Diazonium Betaines to Acetylene and Ethylene: Bending Vibrations Facilitate Reaction. *Agnew. Chem. Int. Ed.*, **121**, 2784 (2009).

[182] X. Yang, J. Zhang, K. Yoshizoe, K. Terayama, and K. Tsuda. Chemts: an efficient python library for de novo molecular generation. *Science and Technology of Adv. Mater.*, **18**, 972 (2017).

[183] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6411–6421. Curran Associates, Inc., 2018.

[184] S. Zhang, A. Golbraikh, S. Oloff, H. Kohn, and A. Tropsha. A Novel Automated Lazy Learning QSAR (ALL-QSAR) Approach: Method Development Applications, and Virtual Screening of Chemical Databases Using Validated ALL-QSAR Models. *J. Chem. Inf. Model.*, **46**, 1984 (2006).

[185] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, and P. Riley. Optimization of Molecules via Deep Reinforcement Learning. *ArXiv e-prints* (2018).

[186] Z. Zhou and R. N. Zare. Personal information from latent fingerprints using desorption electrospray ionization mass spectrometry and machine learning. *Analytical Chemistry*, **89**, 1369 (2017). PMID: 28194988.

[187] P. M. Zimmerman. Automated discovery of chemically reasonable elementary reaction steps. *J. Comput. Chem.*, **34**, 1385 (2013).