

**GENERATIVE, PREDICTIVE, AND REACTIVE MODELS FOR DATA
SCARCE PROBLEMS IN CHEMICAL ENGINEERING**

by

Nicolae C. Iovanac

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



Davidson School of Chemical Engineering

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Brett M. Savoie, Chair

Davidson School of Chemical Engineering

Dr. Bryan Boudouris

Davidson School of Chemical Engineering

Dr. Vivek Narsimhan

Davidson School of Chemical Engineering

Dr. Gaurav Chopra

School of Chemistry

Approved by:

Dr. John Morgan

*This work is dedicated to my mother, who throughout her illness never lost her kindness, hope,
and love for her family.*

ACKNOWLEDGMENTS

This work would not have been possible without the generous financial support provided by public funding sources. In particular, this material is based upon work supported by the National Science Foundation under Grant No. DGE-1333468. The work performed by B. M. S. was made possible through the Air Force Office of Scientific Research (AFOSR) under support provided by the Organic Materials Chemistry Program (Grant number: FA9550-18-S-0003, Program Manager: Dr Kenneth Caster). This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant no. ACI-1548562. Simulations were performed on the Comet supercomputer at the University of California, San Diego, under the Allocation no. TG-CHE190014.

I would like to thank the members of my preliminary and final examination committees. To Prof. Jeffrey Greeley and Prof. Peilin Liao, many thanks for their thoughtful questions, ideas on the next steps to take my research, and helpful discussion during the exam. To Prof. Bryan Boudouris, many thanks for your inspiring words and career guidance. To Prof. Gaurav Chopra, I sincerely appreciate your willingness to help in my defense on such short notice. I extend the same gratitude towards Prof. Vivek Narsimhan, who I also thank for the opportunity to work on a unique and challenging project with his research group.

My sincere thanks to the many undergraduate researchers that helped make this work possible: Mariana Rodriguez Serrano, Bryan Arciniega, and Mackinzie Farnell. I would also like to extend a special thanks to Robert MacKnight; I wish you the best of luck in your graduate career.

I am grateful to Prof. David Corti for his help in my career search, and for enjoyable discussions on the intricacies of statistical mechanics. I would also like to thank Dr. Stephen Shiring for his assistance with navigating Purdue's computational resources, setting up quantum chemical calculations, and his continued advice and input on graduate studies.

My gratitude and thanks to my advisor, Prof. Brett Savoie. Working with him, I had the opportunity to join a new research group and see it rise to one of the foremost and highly lauded groups in the department. I am thankful for that privilege, and for the chance to be a part of a paradigm shift in machine learning in chemical science. Your mentorship and friendship are greatly appreciated.

To my family, I can express nothing but my sincerest thanks. They supported my decision to pursue an advanced degree, and they believed in and encouraged me when I thought I could not finish. Despite the hardships we faced, we fought through it together, and succeeded. Finally, my thanks and love to my wife, Peem. From the late nights of the first semester, to the final rush of the defense and preparation for the next chapter our life, I truly could not have made it this far without your love and support.

TABLE OF CONTENTS

LIST OF TABLES	10
LIST OF FIGURES	11
ABSTRACT	17
1. INTRODUCTION	19
1.1 Improved Property Prediction	19
1.2 Inverse-Design Approaches	20
1.3 Machine Learning for Chemical Reactions	21
1.4 Dissertation Outline	22
2. LITERATURE REVIEW	24
2.1 Background	24
2.2 Machine Learning Fundamentals	26
2.2.1 Origins and Motivation	26
2.2.2 The Neuron	28
2.2.3 Multi-Layer Perceptrons	29
2.2.4 Gradient Descent and Backpropagation	31
2.2.5 Training a Neural Network in Practice	32
2.2.6 Training, Validation, and Testing Sets	33
2.2.7 Other Network Architectures	34
Convolutional Networks	34
Recurrent Neural Networks	36
2.3 Molecular Representations	37
2.4 Machine Learning for Chemical Science	40
2.5 The Inverse Problem	41
2.6 Deep Generative Chemical Models	42
2.6.1 Autoencoder	42
2.6.2 Variational Autoencoder	44
2.7 The Chemical Variational Autoencoder	46
2.8 Transfer Learning	50

3. IMPROVED CHEMICAL PREDICTION FROM SCARCE DATA SETS VIA LATENT SPACE ENRICHMENT	53
3.1 Introduction.....	53
3.2 Computational Methods.....	56
3.2.1 Databases	56
3.2.2 Quantum Chemistry Methods.....	57
3.2.3 Machine Learning Architecture	59
3.2.4 Model Training	59
3.2.5 Latent Space Enrichment	60
3.3 Results and Discussion	61
3.4 Conclusions	68
4. SIMPLER IS BETTER: HOW LINEAR PREDICTION TASKS IMPROVE TRANSFER LEARNING IN CHEMICAL AUTOENCODERS	69
4.1 Introduction.....	69
4.2 Computational Methods.....	72
4.2.1 Datasets.....	72
4.2.2 Machine Learning Architecture	72
4.2.3 Model Training	74
4.2.4 Training Data	74
4.3 Results and Discussion	75
4.3.1 Multi-Property Prediction.....	79
4.4 Conclusions.....	81
5. IMPROVING THE GENERATIVE PERFORMANCE OF CHEMICAL AUTOENCODERS THROUGH TRANSFER LEARNING	83
5.1 Introduction.....	84
5.2 Computational Methods.....	86
5.2.1 Datasets.....	86
5.2.2 Machine Learning Architecture	86
5.2.3 Sampling Paradigms	87
5.3 Results and Discussion	88
5.3.1 Extrapolative Sampling	88

5.3.2	Interpolative Sampling.....	92
5.4	Conclusions.....	95
6.	ACTIVELY SEARCHING: INVERSE DESIGN OF NOVEL MOLECULES WITH SIMULTANEOUSLY OPTIMIZED PROPERTIES	96
6.1	Introduction.....	97
6.2	Methodology	99
6.2.1	Datasets.....	99
6.2.2	Machine Learning Architecture.....	100
6.2.3	Sampling Paradigms	100
6.2.4	Active Learning Technique	101
6.3	Results and Discussion	102
6.3.1	Single Property Searches	102
6.3.2	Single-Property Active Learning.....	103
6.3.3	Multi-Target Optimization.....	104
6.3.4	External Validation.....	106
6.4	Conclusions.....	108
7.	THERMODYNAMIC PROPERTY PREDICTION IMPROVES STRUCTURAL REALISM/SYNTHESIZABILITY/ACCESSIBILITY OF DEEP GENERATIVE MODELS.	109
7.1	Introduction.....	109
7.2	Computational Methods.....	112
7.2.1	Reaction Autoencoder	112
7.2.2	Reaction Prediction.....	113
7.2.3	Datasets.....	114
7.2.4	Product Characterization	114
7.2.5	Reaction Characterization.....	115
7.2.6	Sampling Techniques.....	115
7.3	Results and Discussion	116
7.3.1	Single Target Search: Bandgap	116
7.3.2	Single-Target Search: Enthalpy of Reaction	118
7.3.3	Multi-Target Search: Enthalpy and Bandgap	119
7.4	Conclusions.....	121

8. CONCLUSIONS AND OUTLOOK	123
8.1 Summary	123
8.2 Future Work and Outlook	124
APPENDIX A. SUPPORTING INFORMATION FOR: IMPROVED CHEMICAL PREDICTION FROM SCARCE DATA SETS VIA LATENT SPACE ENRICHMENT	126
APPENDIX B. SUPPORTING INFORMATION FOR: SIMPLER IS BETTER: HOW LINEAR PREDICTION TASKS IMPROVE TRANSFER LEARNING IN CHEMICAL AUTOENCODERS	137
APPENDIX C. SUPPORTING INFORMATION FOR: IMPROVING THE GENERATIVE PERFORMANCE OF CHEMICAL AUTOENCODERS THROUGH TRANSFER LEARNING	140
APPENDIX D. SUPPORTING INFORMATION FOR: ACTIVELY SEARCHING: INVERSE DESIGN OF NOVEL MOLECULES WITH SIMULTANEOUSLY OPTIMIZED PROPERTIES	147
REFERENCES	150
PUBLICATIONS.....	169

LIST OF TABLES

Table 3.1: pK_a prediction results before and after enrichment for test sets associated with median performance of each enriched model. The three sections of the table, shown in gray, blue, and dark gray, represent 128, 256, and 512 pK_a values, respectively. The expected and predicted pK_a values are shown with MAE in pK_a prediction across each test set at the bottom. The presented compounds were selected by ordering the test set according to increasing absolute prediction error and selecting 10 compounds at equally spaced intervals. Aside from the very best performing compounds, enrichment tends to improve pK_a predictions across all compounds. 63

LIST OF FIGURES

Figure 2.1: Rosenblatt’s original perceptron formulation. Modern perceptron frameworks can be considered a more general approach than Rosenblatt’s, and do not include the “Retina” or “Associative units” intended to mimic biological vision processing. Figure reproduced from [23]	27
Figure 2.2: Schematic of a neuron. More advanced network architectures are possible by various combinations and connections between multiple neurons.	28
Figure 2.3: Schematic of a multi-layer perceptron. Inputs are passed to a hidden layer and nonlinear activation function prior to being operated upon by the output node. "Deep" neural networks are simply those with more than one hidden layer and are capable of capturing highly nonlinear and complex relationships in the input data.	30
Figure 2.4: Operation of a convolutional layer with one kernel (center matrix) operating on an input matrix (left) to produce a feature map (right) as output. The highlighted green square indicates the superposition of the kernel in its first position over the input matrix and the red highlighted square in the feature map denotes the output of this operation. The remaining elements of the feature map corresponding to the full set of convolutions is also shown.	35
Figure 2.5: Example of a recurrent neural network architecture for text processing. The architecture shown here mimics that of Figure 2.3, but with the addition of a feedback loop that allows for network outputs at each timestep to influence subsequent output.	37
Figure 2.6: Several of the most common molecular representations used for computational chemistry and machine learning. Representations are all equivalent to the structure denoted by the ball and stick model in the center. Figure adapted from [10].	39
Figure 2.7: Example of a simple autoencoder. 5-dimensional bit-vectors are compressed to a 3-dimensional representation and then expanded back to full dimensionality, ideally restoring the input vector. The problem presented here is essentially a multi-label classification task, where a 1 in the input vector corresponds to presence of that class, and a 0 indicating that the input does not belong to that class. The output vector thus corresponds to a list of probabilities of belonging to each class. During training, the model attempts to adjust these output distributions to be more confident in its class predictions. Input and output vectors were chosen arbitrarily.	43
Figure 2.8: Overview of the chemical variational autoencoder framework. (a) a schematic of the model architecture. Discrete chemical structures are encoded to a continuous latent representation, and then decoded to return the original input. Vectors may be sampled from the latent space and passed to an ancillary property prediction network, which estimates properties of the corresponding molecule. (b) a schematic of the molecular optimization routine suggested by the authors. Because the chemical variational autoencoder develops a continuous chemical latent space, traditional gradient based optimization techniques can be employed to determine regions of the latent space to sample to attain optimized molecules. Figure reproduced from [67]	47
Figure 2.9: Comparison of latent space for chemical variational autoencoders trained on encoding/decoding as well as prediction of water-octanol partition coefficient and quantitative	

estimate of drug-likeness. The 156-dimensional mean encodings are projected down to the first two principal components for visualization. Without a property prediction task, there is little organization of compounds with respect to properties. The addition of a property prediction task based on latent encodings ensures meaningful organization with respect to properties, although as the case of QED shows this organization is not always linear. Figure reproduced from [67]...... 48

Figure 2.10: A timeline showing the development of generative chemical models, beginning with (e) the chemical variational autoencoder and followed by (a) generative adversarial networks, (b) recurrent neural networks, (f) the grammar variational autoencoder, (g) graph-based VAEs, (c) graph-based RNNs, and (d) graph-based GANs. Figure reproduced from [12]...... 50

Figure 3.1: Autoencoder architecture displaying continuous latent space and joint prediction tasks. The hypothesis investigated in this work is whether joint training of the latent space on a data-rich prediction task can improve the performance of a correlated data-scarce prediction task through the enrichment of the common latent space variable. 55

Figure 3.2: Correlation plots of (A) $\Delta G_{\pm H}$ and (B) E_{sp} with pK_a . For chemically similar compounds, it is possible to predict pK_a as a linear function of $\Delta G_{\pm H}$ to a high degree of accuracy, as demonstrated by the regression of 7 amines (highlighted in red, $R^2 = 0.98$). However, without *a priori* knowledge of chemical structure, such a strategy fails in the general case, as shown by the remainder of uncorrelated points in green. In contrast, E_{sp} does not exhibit correlation with pK_a and is used here as a chemically significant, but uncorrelated, control property. 58

Figure 3.3: Box and whisker plot showing errors statistics for final models. 30 independent testing/training splits were generated, and independent models were trained on each split. The box and whiskers thus correspond to the MAE across the 30 test sets for each of the model paradigms. Median pK_a prediction errors are shown as a blue line, and whiskers are drawn to extend to the range of observed errors. The notches represent the 95% confidence interval about the median. 61

Figure 3.4: Principal component analysis of chemical latent spaces trained on pK_a with $\Delta G_{\pm H}$ enrichment showing latent space reorganization. Each point represents a compound from the full dataset with its encoding vector projected onto the first two principal components of the specified model latent space. Points are colored according to the indicated property. Standardized values are used for $\Delta G_{\pm H}$ coloration. Coefficient of determination and Spearman rank order coefficient are shown in each panel. 65

Figure 3.5: Depiction of the latent space enrichment process. PCA analysis of the latent space of the unenriched (A) and enriched (B) models $MpKa_{128}$ and $MpKa, \Delta G_{128,512}$, demonstrating organization with respect to $\Delta G_{\pm H}$ upon enrichment. (C) Illustration of the proposed enrichment mechanism. (Left) Sparse datasets result in regions of latent space with little organization. (Middle) Joint training on correlated computational data results in organization of experimentally uncharacterized regions of latent space. (Right) A more continuous latent space with well-defined property gradients is formed. 66

Figure 4.1: Overview of chemical autoencoder (CAE) architectures for transfer learning. The autoencoder generates a compressed vectorial representation of chemical space by employing a low dimensional intermediate layer in the model (i.e., the latent space). This compressed representation is obtained by training the CAE on a reconstruction task using abundant chemical structure data. In transfer learning applications, the latent space serves as a common input feature

for two or more property prediction tasks. (A) When utilizing complex predictor networks for joint property training, latent space organization is incomplete due to the predictor’s ability to learn complex surfaces. (B) When a linear predictor network (i.e., a single unit with a linear combination of latent space features) is used for joint training, prediction errors must be minimized by backpropagation through the encoder, resulting in a linear organization restraint on the training objective function..... 71

Figure 4.2: Mean Absolute Error (MAE) in $E_{g,DFT}$ prediction using models trained on varying fractions of DFT data (gray) and models jointly trained on $E_{g,DFT}$ and $E_{g,xTB}$ data (lines). In situations where less than ten percent of the DFT data is available for training, the identity of the compounds within the dataset becomes an important consideration. For all of these models we train 10 models, each on a different random subset of the training data, and select the one exhibiting the best performance on the validation set for final evaluation on the test set. The 95% confidence interval about the mean values are within marker size. 76

Figure 4.3: Principal component analysis (PCA) performed on latent encodings of the entire QM9 dataset. Both models were trained solely on $E_{g,xTB}$ training set data and are distinguished by the complexity of the predictor network. PCA results, MAE, and linear coefficient of determination (R^2) are presented for (left) a predictor network comprised of three fully connected layers of 64 nodes terminating in a single linear unit; and (right) a predictor network comprised of a single linear unit. Both networks achieve comparable performance with respect to MAE on the $E_{g,xTB}$ prediction task (differing by less than 1% of the range of $E_{g,xTB}$ training data) while exhibiting qualitatively different latent space organization. Only the simple linear predictor results in an interpretable latent space dimension (PC_1), corresponding to the HOMO-LUMO gap. 78

Figure 4.4: Mean Absolute Error (MAE) in DFT bandgap prediction using varying fractions of DFT data as well as up to 5 additional properties. While the fraction of available DFT data is allowed to vary, the auxiliary properties are included at the full training fraction. The inclusion of multiple property prediction tasks during training has a negligible impact on prediction accuracy while providing additional organization of the latent space. The 95% confidence interval about the mean values are within marker size. 79

Figure 4.5: Principal component analysis performed on latent encodings for compounds within the training set for a model trained on both DFT data and electronic spatial extent. The projections along the first two principal components are colored according to (A) $E_{g,DFT}$ and (B) R^2 . The chemical landscape shows excellent organization with respect to both properties ($R^2=0.98$ for both). Latent space organization occurs in orthogonal directions for uncorrelated properties. 80

Figure 5.1: Comparison of latent space Sobol sampling for 2D autoencoders trained on (a) chemical reconstruction only and those with (b) an ancillary U_0 prediction. All compounds utilized in training the models have been projected into the latent space and are colored according to U_0 . The points corresponding to the Sobol sampling are colored according to their average validity across 500 decodings. For the model trained to predict U_0 , we observe a clear relationship between increasing validity and decreasing $|U_0|$, whereas for the model trained only on reconstruction, there are no observable trends with respect to either validity or property values..... 88

Figure 5.2: Average sampling validity and uniqueness for the high $|U_0|$ extrapolation trial. Results are averaged across 10 distinct models for each training paradigm, with error bars denoting standard deviation. Baseline models exhibit very limited ability to generate structures with high

$|U_0|$. The addition of ancillary property prediction tasks greatly improves the generative utility of these models..... 90

Figure 5.3: Distribution of U_0 for structures for the training set (a) compared with structures generated from extrapolation along the first principal component of models trained on (b) encoding and decoding alone, (c) an ancillary U_0 prediction task, (d) U_0 and $ZPVE$ prediction tasks, and (e) U_0 , $ZPVE$, and E_g prediction tasks. For each paradigm, 3000 unique structures are generated across the 10 duplicate models for a total of 30,000 structures. The mean of each distribution is denoted with a dashed red line and the largest $|U_0|$ value in the training data is indicated by the dashed green line. The percentage of generated compounds with $|U_0|$ greater than observed in the training data is shown on the right. 91

Figure 5.4: Distribution of E_g for the training data (a) and structures generated from models trained to predict E_g by targeting (b) 1.5-2.0 eV, (c) 5.5-6.0 eV, and (d) 9.5-10.0 eV. While (c) and (d) show good specificity, the model is unable to resolve structures in the 1.5-2.0 eV range (a). For each target, 3000 unique structures are generated across the 10 duplicate models for a total of 30,000 structures. The median of each distribution is indicated by a dashed red line. Targeted regions are highlighted in blue..... 92

Figure 5.5: Average sampling validity and uniqueness for the three targeted E_g paradigms. Results are averaged across 10 distinct models for each training paradigm, with error bars denoting standard deviation. The models show difficulty in generating compounds within the poorly represented 1.5-2.0 eV range and are comparatively much stronger in generating structures with E_g between 5.5-6.0 and 9.5-10.0. The addition of ancillary U_0 and $ZPVE$ prediction tasks greatly increases the proportion of valid and unique structures generated in the 1.5-2.0 eV range. 93

Figure 5.6: Distribution of E_g for structures generated from model trained to predict U_0 , $ZPVE$, and E_g . E_g is targeted within 1.5-2.0 eV while U_0 is extrapolated to bias the discovery of larger compounds. Compared to targeting E_g alone, the distribution of E_g is much wider, which allows for the generation of approximately twice as many low E_g structures compared with models trained on E_g alone. The targeted region is highlighted in blue..... 94

Figure 6.1: Property histograms for molecules generated by models trained on (A) vertical ionization potential, (B) electron affinity, and (C) dipole moment. For each model, 100,000 structures were generated and subsequently characterized at the xTB level. Training distributions are shown in red with generated data in blue. Means of the training distributions and generated distributions are indicated by green and purple dotted lines, respectively. Models are tasked with extrapolating to compounds with property values not observed in the training data, shown in orange..... 102

Figure 6.2: Property histograms for model trained to predict electron affinity. Training distribution is shown in purple (TOP), means are indicated by dashed green line, and the target region of 1-2 eV is highlighted in yellow. Initially, the model has difficulty generating structures with the specified EA (Iteration 1). After 4 and finally 9 iterations, the mean of the generated EAs has shifted to be within the target range, where the distribution also peaks. 104

Figure 6.3: 2D property histogram for model trained to predict VIP, EA, and DM, and tasked with targeted structure generation for these properties. For visualization, only compounds with VIP greater than 10.0 eV are considered. The targeted region, with DM between 4-5 Debye and EA

between 1.5-4.0 eV, is indicated with a box. Ionization potential is extrapolated beyond the training data, while the electron affinity range has little representation, and the dipole moment range is very well represented. Initially, (A) the model is not effective in generating compounds that fulfil all three criteria together. After 8 iterations of the active learning procedure (B), the property distribution of proposed structures has shifted to cover the targeted region and the model is now capable of proposing over 1600 structures fulfilling all three property criteria. 105

Figure 6.4: Final five structures simultaneously achieving desired vertical ionization potential, electron affinity, and dipole moment values validated at ω B97X-D3/def2-TZVP level. The model has learned to make heavy use of oxygen atoms to achieve a high ionization potential while simultaneously maintaining a high electron affinity..... 107

Figure 7.1: The "Jacob's Ladder" of inverse-design. In attempting to reconcile the domains of computation and the physical world, the first step is ensuring that proposed compounds do not violate valency and atom-type constraints ("Does it Exist?") The next step requires consideration of thermodynamic limitations, as without *a priori* constraints it is possible for the discovery workflow to suggest thermodynamically inaccessible molecules ("Can it be Synthesized?"). In this work, we attempt to address how we can reach this rung, as well as suggest methods for moving closer to true inverse design. We anticipate that further development of generative chemical methods will see methods designed to encapsulate bulk-phase morphologies, processing concerns, and other higher order considerations..... 117

Figure 7.2: Property histograms and example molecules generated by models trained to predict molecular bandgap compared to distribution of training data (TOP). For each sampling paradigm, we generated 2000 novel reactions and subsequently characterized them at the ω B97X-D3/def2-TZVP level, with geometries obtained via xTB. We consider generating reactions that lead to a product with (MIDDLE) a bandgap lower than any member of the training set, and (BOTTOM) higher than any compound in the training set. Means of the training distribution and each generated distribution are indicated by dotted green lines, and the validated *E_g* are listed below each example molecule..... 118

Figure 7.3: Property histograms and example reactions generated by models trained to predict enthalpy of reaction compared to distribution of training data (FIRST ROW). For each sampling regime, we generated 2000 novel reactions and subsequently characterized their products using TCIT. We considered generating reactions biased to be (SECOND ROW) endothermic, (THIRD ROW) exothermic, and (FOURTH ROW) strongly endothermic. Means of the training distribution and each generated distribution are indicated by dotted green lines. For each histogram, three example reactions are shown, with reactants separated from products by an arrow listing their associated enthalpy of reaction. 119

Figure 7.4: Kernel density estimates for distribution of bandgap and enthalpy of reaction for reactions generated from (TOP) a model trained only on predicting bandgap and (BOTTOM) a model trained to predict both enthalpy of reaction and product bandgap. Both models were tasked with generating 2000 reactions that produce a product molecule with bandgap between 6 and 8 eV, but the model trained on enthalpy of reaction was also requested to limit proposals to have enthalpy of reaction less than -50 kJ/mol. This more than doubles the number of thermodynamically feasible reactions attaining the bandgap target, from 50 to 120. Multidimensional means are indicated by

green crosshairs. The extent of the bandgap-only histogram is outlined in white for comparison.
..... 121

ABSTRACT

Data scarcity is intrinsic to many problems in chemical engineering due to physical constraints or cost. This challenge is acute in chemical and materials design applications, where a lack of data is the norm when trying to develop something new for an emerging application. Addressing novel chemical design under these scarcity constraints takes one of two routes: the traditional forward approach, where properties are predicted based on chemical structure, and the recent inverse approach, where structures are predicted based on required properties. Statistical methods such as machine learning (ML) could greatly accelerate chemical design under both frameworks; however, in contrast to the modeling of continuous data types, molecular prediction has many unique obstacles (e.g., spatial and causal relationships, featurization difficulties) that require further ML methods development. Despite these challenges, this work demonstrates how transfer learning and active learning strategies can be used to create successful chemical ML models in data scarce situations.

Transfer learning is a domain of machine learning under which information learned in solving one task is *transferred* to help in another, more difficult task. Consider the case of a forward design problem involving the search for a molecule with a particular property target with limited existing data, a situation not typically amenable to ML. In these situations, there are often correlated properties that are computationally accessible. As all chemical properties are fundamentally tied to the underlying chemical topology, and because related properties arise due to related moieties, the information contained in the correlated property can be leveraged during model training to help improve the prediction of the data scarce property. Transfer learning is thus a favorable strategy for facilitating high throughput characterization of low-data design spaces.

Generative chemical models invert the structure-function paradigm, and instead directly suggest new chemical structures that should display the desired application properties. This inversion process is fraught with difficulties but can be improved by training these models with strategically selected chemical information. Structural information contained within this chemical property data is thus transferred to support the generation of new, feasible compounds. Moreover, this transfer learning approach helps ensure that the proposed structures exhibit the specified property targets. Recent extensions also utilize thermodynamic reaction data to help promote the synthesizability of suggested compounds. These transfer learning strategies are well-suited for

explorative scenarios where the property values being sought are well outside the range of available training data.

There are situations where property data is so limited that obtaining additional training data is unavoidable. By improving both the predictive and generative qualities of chemical ML models, a fully closed-loop computational search can be conducted using active learning. New molecules in underrepresented property spaces may be iteratively generated by the network, characterized by the network, and used for retraining the network. This allows the model to gradually learn the unknown chemistries required to explore the target regions of chemical space by *actively* suggesting the new training data it needs. By utilizing active learning, the create-test-refine pathway can be addressed purely *in silico*. This approach is particularly suitable for multi-target chemical design, where the high dimensionality of the desired property targets exacerbates data scarcity concerns.

The techniques presented herein can be used to improve both predictive and generative performance of chemical ML models. Transfer learning is demonstrated as a powerful technique for improving the predictive performance of chemical models in situations where a correlated property can be leveraged alongside scarce experimental or computational properties. Inverse design may also be facilitated through the use of transfer learning, where property values can be connected with stable structural features to generate new compounds with targeted properties beyond those observed in the training data. Thus, when the necessary chemical structures are not known, generative networks can directly propose them based on function-structure relationships learned from domain data, and this domain data can even be generated and characterized by the model itself for closed-loop chemical searches in an active learning framework. With recent extensions, these models are compelling techniques for looking at chemical reactions and other data types beyond the individual molecule. Furthermore, the approaches are not limited by choice of model architecture or chemical representation and are expected to be helpful in a variety of data scarce chemical applications.

1. INTRODUCTION

Whether we consider the development of new medicines, devices for energy storage, or membranes for water purification, our future is largely contingent on our ability to create specialized chemistries. These excursions into the unknown are monumentally difficult (the material and time costs to bring a new drug to market, for example, are enormous) because they require us to grapple with the problem of data scarcity. Our scientific approaches are guided by the work that came before us, the “giants” on whose shoulders we stand and whose data we rely on. What happens when the information that we do have available is not enough to guide us towards the next step, and we have nowhere to stand? While computational techniques have emerged as powerful resources, allowing us to efficiently explore design spaces before committing time and resources to wet-lab trials, high throughput computation alone cannot save us from data limitations. To this end, I have focused my PhD work on devising ways to help overcome this fundamental issue of data scarcity in the chemical sciences. The results of this work may be broadly recapitulated into three categories of research: i) improved prediction of the application properties of small molecules, ii) improved generation of new compounds expressing desired properties, and iii) extensions of these approaches to chemical reaction data.

1.1 Improved Property Prediction

In a perfect scenario, the answer to the problem of limited data is simply to collect more. However, for chemical data it is often infeasible to do so, whether due to cost, safety, or the urgency of the problem at hand. There is also no guarantee that further experimentation will lead to a useful result. Indeed, one of the critical bottlenecks in the search for a new material lies in the search for and screening of potential candidate molecules, the vast majority of which are unsuccessful. Computational methods have served admirably in this regard, providing avenues for high throughput screening and characterization of molecular candidates. However, physics-based modeling provides accuracy commensurate with the level of theory applied. For the precision required in chemical applications, common approaches such as DFT are not suitable for high-throughput exploration of chemical space.

Statistical approaches, including machine learning (ML), provide an alternative route to property prediction, inferring properties from structure based on learned trends. Ordinarily, these methods would be subject to the same data scarcity constraints outlined above; however, these methods provide a means to automatically extract the latent topological features that correlate with certain property expressions. As all chemical features are fundamentally based on molecular topology, it is intuitive that related features are described by similar structural arrangements. Now consider the case where an experimental property, or high-cost computational property, is desired but available in limited numbers, but a correlated property may be calculated quickly and in abundance. Naïve structure-function relationships may be learned on abundant data and fine-tuned on sparse data to properly predict the feature of interest. The bulk of the physics may then be learned on the abundant data, with the deviation at higher levels of theory learned on the high-accuracy data. This approach is utilized in the improved prediction of experimental aqueous pK_a from chemical structure by leveraging data from free energy calculations, and further explored in the improved prediction of bandgap at the DFT level utilizing information from semi-empirical calculations.

1.2 Inverse-Design Approaches

The number of possible candidates may often preclude efforts to fully enumerate a materials design space in the hopes that a match will be found. An attractive alternative is found in the inverse-design approach: directly predicting a *structure* exhibiting a specified property. Although this concept has existed for decades in the literature, it is only within the last decade that the inverse-design problem has been made computationally tractable with the advent of deep-learning approaches. Preliminary results have been promising, but the general problem of inverting function to structure remains unsolved. The problem is further exacerbated by data limitations, making it even more difficult to learn which structural features give rise to particular property expressions. Failure modes are thus observed in the form of outputs that do not correspond with accessible chemistries, such as chemical formulae that violate basic valency rules, or reasonable outputs that do not match the desired properties. These failure modes, an inability to generate valid structures and an inability to predict the properties of the proposed structures *a priori*, are related. Training the model to predict properties that are first-order functions of the chemical topology, such as the dependence of zero-point vibrational energy on the number and type of covalent bonds present in

a structure or the relationship between internal energy and the number of atoms present, provides the generative process with additional information on the structure-function relationship of valid compounds. The transferred information can then help the model to propose valid chemistries, even when the primary property target is in an extrapolative region, by ensuring energetically reasonable connectivity. This transfer learning approach for improved generative performance is examined for the case of generating new molecules with optimized bandgap and internal energy.

In other applications, the desired functionalities may be so distinct from the available training data that it no longer becomes a question of distilling more chemical information, but of acquiring more information in those underrepresented regimes. Active learning is a favorable paradigm for this problem, where the model can indicate the maximally informative data it needs to train on to reach the target objective. A generative chemical model with good predictive capability can generate new data that is at least close to the desired functionalities; the model can iteratively retrain on the new data that it itself is suggesting to gradually improve its understanding of the necessary chemistries to reach the target range. Active learning is particularly well suited for multi-target chemical design, where high dimensionality of the target parameter spaces can lead to limited representation even in substantial databases and is demonstrated in this work in the generation of compounds with simultaneous vertical ionization potential, electron affinity, and dipole moment targets.

1.3 Machine Learning for Chemical Reactions

Assuming that a generative chemical model can propose a candidate molecule representing valid chemistry and displaying the desired application properties, one question still remains: *can it be synthesized?* Answering this question requires consideration of not only distinct molecules, but also the chemical reactions that connect them. Recent approaches to inverse-design have attempted to ensure synthesizability by constraining models to operate with chemical reactions rather than molecules; for a given output, not only is a target molecule obtained, but also the necessary reactants. Unfortunately, just as the outputs of a chemical model are not guaranteed to be valid, the proposals of a reaction-based model often do not correspond with thermodynamically or kinetically favorable reactions.

However, just as the reaction case shares the drawbacks of the individual molecule problem, it also mirrors many of the same solutions. By including relevant thermodynamic reaction data

during training, generative reaction-based models can be biased to propose reactions that tend to be thermodynamically favorable, while also producing products that display desired properties, thus emerging as a necessary condition for further model development. In particular, the final main body chapter of this work outlines how a bandgap optimization problem may be improved by folding in enthalpy of reaction data, leading to optimized products and a feasible reaction pathway to attain them. Avenues for overcoming data scarcity in chemical engineering are thus provided from the level of individual molecules up to one-step chemical reactions.

1.4 Dissertation Outline

Chapter 2 provides an overview of the guiding theory behind this work. The origins of the materials search problem are discussed, starting with forward approaches of predicting function from molecular structure. After describing traditional approaches, ML approaches are then introduced. A brief history and review of ML fundamentals follows, after which the inverse-problem is formally introduced. Traditional approaches are recapitulated before modern deep-learning methodologies are discussed. This chapter concludes with a discussion on the transfer learning methodologies central to much of this work.

Chapter 3, “Improved Chemical Prediction from Scarce Data Sets via Latent Space Enrichment”, discusses a transfer-learning based approach to improving the prediction of aqueous pK_a/pK_b of small molecules through the use of free-energy change calculations performed in excess.

Chapter 4, “Simpler is Better: How Linear Prediction Tasks Improve Transfer Learning in Chemical Autoencoders”, provides an enhancement to the methodology discussed in the previous chapter through use of a novel autoencoder architecture, and also demonstrates its effectiveness on a wider array of properties across varying levels of data scarcity.

Chapter 5, “Improving the Generative Performance of Chemical Autoencoders through Transfer Learning,” provides insight on how generative chemical models may be improved through the inclusion of relevant chemical property data. In particular, an improved ability to generate new molecules with targeted internal energy and bandgap is demonstrated.

Chapter 6, “Actively Searching: Inverse Design of Novel Molecules with Simultaneously Optimized Properties”, introduces an active learning framework to improve the performance of generative chemical models when operating under multiple functional constraints or other

especially data-scarce regimes. The approach is demonstrated for optimization of dipole moment, electron affinity, and vertical ionization potential, both simultaneously and individually, in extrapolative and interpolative sampling paradigms.

Chapter 7, “Thermodynamic Property Prediction Improves Structural Realism/Synthesizability/Accessibility of Deep Generative Models”, discusses some of the most recent reaction-based approaches to generative chemical models, and illustrates how their rapid proliferation may have come at the expense of focusing on experimentally accessible compounds. The dangers of thermodynamically unconstrained molecular design are demonstrated with respect to synthetic feasibility, and a demonstration is provided for the case of bandgap that the inclusion of thermodynamic reaction data is a crucial component of ensuring the experimental feasibility of materials suggested by ML-based routines.

Chapter 8 summarizes the key findings of this work and suggests avenues for continued research. A forecast for future work in the field is also provided.

2. LITERATURE REVIEW

2.1 Background

Clean water and sanitation. Affordable and clean energy. Industry, innovation, and infrastructure. Good health and well-being. These represent a small variety of the United Nations' Sustainable Development Goals for 2030, identified as critical milestones in the attainment of a sustainable future.[1] These goals are all ambitious, multi-faceted, and necessitate broad, multidisciplinary approaches, but what the quoted goals above, in particular, all share is a significant materials development aspect. From the creation of advanced membranes for water filtration and fuel cells, to rational drug design and optimization, the research, development, and deployment of novel materials will prove crucial in working towards these goals. For many applications, materials that exhibit the requisite properties do not yet exist. However, what we do not lack are opportunities to solve these problems; there are estimated to be more synthesizable molecules (on the order of 10^{60} distinct structures) [2] than there are stars within the universe.[3] For a given materials problem, it is not an overgeneralization to suggest that a compound *could be* created to exactly solve it. But, it is within this vast chemical space that a truly daunting challenge emerges: our coverage of this chemical space is woefully limited, with an estimated 10^8 unique molecules in the subset of synthesized compounds.[4] Even accounting for nearly synthesizable structures, our exploration brings us comparatively no closer, with $10^{20} - 10^{24}$ imminently synthesizable compounds estimated.[5]

The design of novel compounds forces us to grapple with the problem of data scarcity, and its reconciliation entails an enormous expenditure of resources. The latest estimates for the cost to bring a new drug to market now exceeds 1 billion USD and requires a thirteen year development cycle on average.[6],[7] These figures are representative of a design problem with standardized approaches and a mature interconnection between government and private sector actors (since much of the early formalization of the molecular search problem has its origin in rational drug design). If we consider the general case, estimates of lab-to-market translation time can exceed *three decades*. [8] The problem is made even more apparent when we consider the urgency of the problems for which these novel materials are intended to solve, from drug design to efficient energy storage media. Although high-throughput assays and other experimental screening

techniques are powerful in their own right,[9] full experimental exploration of chemical space is an intractable problem. There are simply not enough resources, researchers, or time to fully explore even a massively constrained design space. Thus, while many material advances have been made by contemporary material development workflows, some of the most important materials discoveries (such as the discoveries of Teflon and Kevlar, among others) have relied upon luck.[10]

In regard to more efficient screening and procurement of chemical data, computational resources have long been a powerful ally against data scarcity in chemical science. Methods such as density functional theory (DFT) are now common tools for the synthetic chemist or catalyst researcher. DFT and other physics-based simulations leveraging quantum chemistry and/or molecular dynamics allow us to accurately probe chemical systems without devoting additional resources to experimental study. Of course, their utility goes beyond simply serving as surrogates for experimentation. The parallelizability of computational methods on modern hardware allows chemical and parameter spaces to be surveyed at scales far exceeding what is experimentally possible.

One of the earliest frameworks for computer-aided chemical design thus emerged in high-throughput virtual screening (HTVS).[11] In HTVS, computational methods are used to down-select molecular candidates for more accurate simulations or experimental analysis. It is the imposition of this “computational funnel” coupled with automatic techniques to rapidly address problems normally requiring significant time scales that generally defines the high-throughput formula. HTVS has enjoyed exposure as a tool for drug design, and has also found recent success for energy materials, such as the US Government’s Materials Genome Initiative and the Harvard Clean Energy project.[12] Here, distributed computing resources were used to obtain DFT-level geometry and electronic structure calculations for over 3 million photovoltaic candidates, several of which were down-selected for synthesis and successfully tested in photovoltaics.

Despite the force multiplication that computational methods bring, the success of a HTVS approach still depends on intelligent deployment of these resources. Statistical approaches such as Monte Carlo methods allow for efficient sampling of probable candidate molecules to avoid explicit enumeration of entire chemical spaces.[12] However, even enhanced sampling techniques still require validation at each step, thus the search time is still limited by the efficiency of the property calculation method. Rather than relying on physics-based simulation, from an early stage computational chemists have considered the possibility of data-driven approaches to property

prediction, where the structure-function relationships are not rigorously calculated but inferred from the available data.[13] Data science techniques, and later machine learning methodologies, arose to fill this need.

2.2 Machine Learning Fundamentals

2.2.1 Origins and Motivation

Before discussing the applications of machine learning in chemical science, the origins and contemporary definitions of the terms “neural network” and “machine learning” need to be explained. Modeling their work after the approximate function of biological neurons, which accept signals from other cells and “fire” or send a signal of their own if a certain threshold is reached (e.g. pain, cold, instructions to move a body part), McCulloch and Pitts posited the idea of an “artificial neuron” in 1943, comprising a mathematical model where numerical inputs could be summed and produce a binary output; 1 if a defined threshold is reached, 0 otherwise.[14] Despite the simple premise, the authors there and in subsequent work provided mathematical proof that a sufficiently complex network of these neurons (i.e. a “neurological network”) could model any logical function, even approximating the function of the optical nerves for detecting spatial patterns.[15] Imagining the monolithic computers of the time, with connections between computing elements facilitated by manually plugging in wires and inputs provided via punch card, it is easy to see how the term “machine learning” rose to prominence around this time.[16] Although there was much excitement surrounding the concept of the artificial neural network, including machines that could play checkers and simple games, it nonetheless took over a decade for these concepts to be consolidated into a general approach.

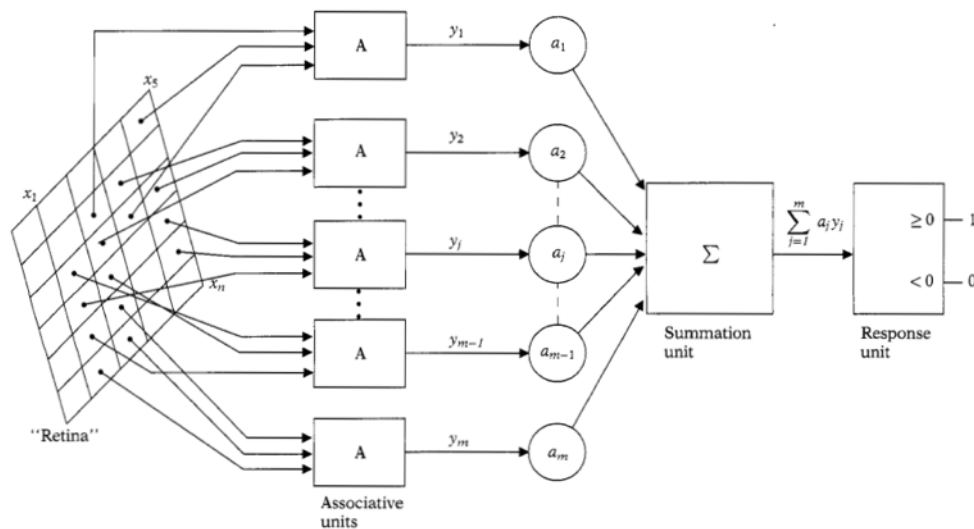


Figure 2.1: Rosenblatt’s original perceptron formulation. Modern perceptron frameworks can be considered a more general approach than Rosenblatt’s, and do not include the “Retina” or “Associative units” intended to mimic biological vision processing. Figure reproduced from [23]

In the mid-1950s, Rosenblatt brought the theories of neural computing to a practical implementation with his development of the “perceptron.” Modeled as a generalized class of networks that attempt to model the processes of the human brain, the perceptron has become a term that remains familiar even today as a fundamental element of machine learning. [17] Figure 2.1 provides insights into one of its earliest implementations and demonstrates how these early approaches mimicked proven-biological processes to solve tough problems such as image recognition. Inputs from “associative units” reading punch cards are subjected to a weighted sum. If this sum exceeds a threshold value, the “response unit” outputs a value of 1, or 0 otherwise. This Heaviside “activation function” was intended to mirror the activation of a biological neuron in response to stimulus. In today’s machine learning parlance, what Rosenblatt referred to by “perceptron” would now be considered a specific type of perceptron, “neuron”, or “node” and the terms are often used interchangeably, although perceptron as a standalone unit has fallen out of favor outside of the context of “multi-layer perceptrons” which will be discussed shortly. Note that there appears to be a case of circular referencing in the literature with respect to the original form of the “perceptron.” While Rosenblatt later constructed a specific device implementing the perceptron idea, the “Mark 1 Perceptron”, Rosenblatt notes in his 1962 book “Principles of

neurodynamics; perceptrons and the theory of brain mechanisms” that he intended for “perceptron” to refer to a class of neural networks rather than a specific device.[18]

2.2.2 The Neuron

A neuron is a computational model that accepts a set of inputs \mathbf{x} and scales them according to \mathbf{w} , its set of weights. The sum of these elementwise multiplications, along with the addition of a bias or intercept term \mathbf{b} , are operated on by an activation function, f in Figure 2.2, which defines the output of the network \mathbf{y} .

Activation functions for the final output of a neural network (the single neuron in this case)

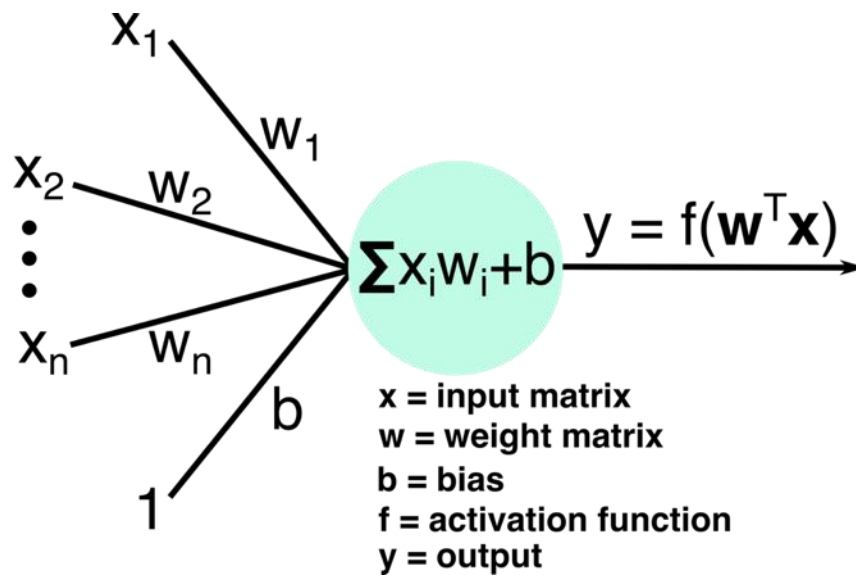


Figure 2.2: Schematic of a neuron. More advanced network architectures are possible by various combinations and connections between multiple neurons.

are largely mediated by the intended problem that is to be modeled. For regression of a real-valued variable, a variety of non-linear activation functions or a simple linear combination can be used. In the latter case, the single neuron would simply reduce to a linear regression, as $y = \mathbf{w}^T \mathbf{x} = \sum_{i=1}^{inputs} w_i x_i + b$. For binary classification tasks, a step function was historically used to distinguish between two classes (0 or 1). This “single layer perceptron” is fairly limited in the problems it is capable of addressing. It cannot, for instance, model the XOR logical operator, nor can it in general handle problems that are not linearly separable.[19]

2.2.3 Multi-Layer Perceptrons

Despite the limitations of the single perceptron, by connecting multiple perceptron units together in a “feed-forward” fashion, such that the outputs of a given “layer” of perceptrons are the inputs to the next layer, a much wider variety of problems may be solved. Consider the network presented in Figure 2.3. It consists of 3 layers: an input layer that simply feeds in the input features, an output layer that computes the objective, and between the two is an additional “hidden” layer, so called because its output values are not outwardly observed. Each input value is separately connected to each hidden node with its own weight. Each hidden node then computes an output value in exactly the same fashion as the single perceptron described earlier, as $h_i = f(\sum_{i=1}^{inputs} w_{1,i}x_i + b_h)$, where f is the activation function, the subscript 1 denotes the first node, and the subscript i indexes the input nodes. The subscript h denotes the constant bias term within the layer. As an example, h_1 in Figure 2.3 may be computed as $h_1 = f(w_{1,1}x_1 + w_{1,2}x_2 + w_{1,3}x_3 + w_{1,4}x_4 + w_{1,5}x_5 + b_h)$, with analogous expressions for the other nodes.

Couching the weight and bias terms as elements in multidimensional tensors, the output of the hidden layer may be represented by the vector $\mathbf{h} = f(\mathbf{w}_h^T \mathbf{x} + \mathbf{b}_h)$, with the activation function operating in elementwise fashion. Similarly, the output of the network can be represented as $\mathbf{y} = g(\mathbf{w}_o^T + \mathbf{b}_o)$, where here the subscript o indicates the output layer, and this matrix representation allows for efficient network calculations. Crucially, the connections between the hidden and output layers must be governed by *nonlinear* activation functions, or a network of any depth will mathematically collapse to the single layer perceptron case. In modern applications, this activation function typically takes the form of the rectified linear unit (ReLU) due to easy and favorable gradient calculations. [20] Now, the hidden layers can be thought of as a nonlinear transformation of the input features into a space where the target function may be approximated. A sufficiently large “multi-layer perceptron” with nonlinear activation in the hidden layers is capable of approximating almost any function.[21],[22]

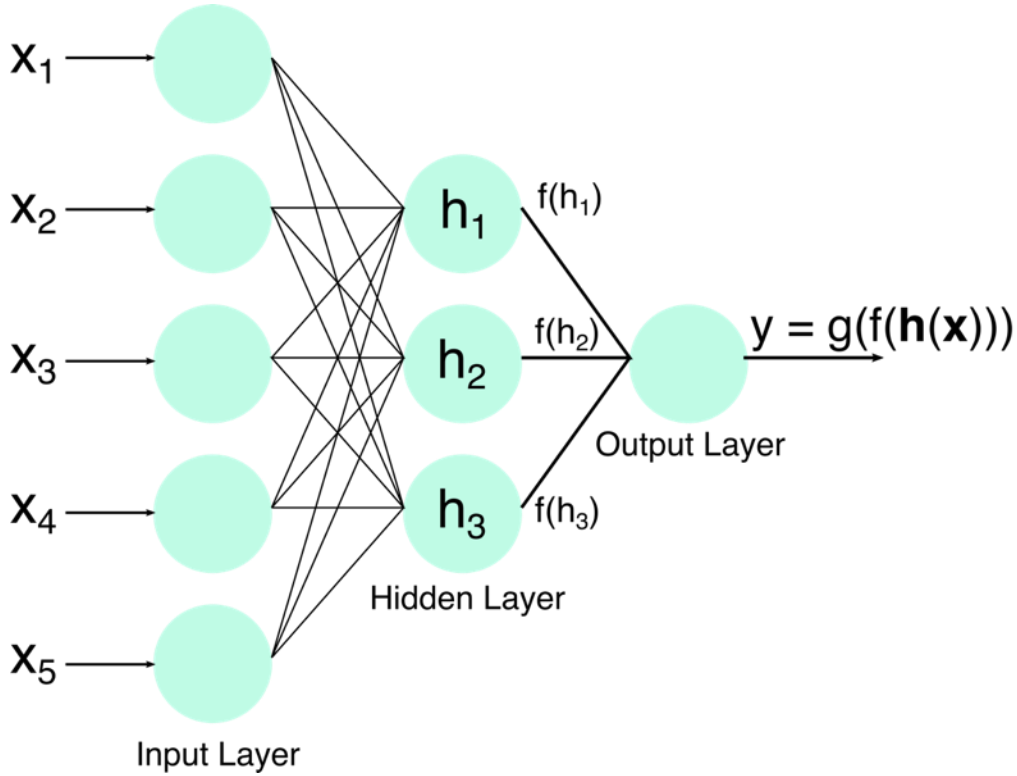


Figure 2.3: Schematic of a multi-layer perceptron. Inputs are passed to a hidden layer and nonlinear activation function prior to being operated upon by the output node. "Deep" neural networks are simply those with more than one hidden layer and are capable of capturing highly nonlinear and complex relationships in the input data.

The question of course arises as to *how* that target function is approximated. This question may be answered by considering a regression task as an example. For a given input, x_i the model will produce an estimate y_i that should ideally match the true target value \hat{y}_i , that is, the difference between the two must be minimized. The “training” of a neural network can thus be approached as an optimization problem, with the goal of selecting the weights and biases which minimize an error or cost function that describes the objective of the network. In the case of regression (chosen for simplicity, not as a limitation), a suitable loss function may be defined in terms of the mean-squared error (MSE) between the predictions and target values, or $J_i(\mathbf{w}, \mathbf{b}, \mathbf{x}_i) = \frac{(\hat{y}_i - y_i)^2}{2}$. MSE tends to be a better choice than mean absolute error (MAE) for regression because it more greatly penalizes outliers and is continuously differentiable, although MAE is commonly used as a readily interpretable metric during model training. MSE can also be used for classification problems, but cross-entropy is typically a better option when dealing with probabilities of classes

2.2.4 Gradient Descent and Backpropagation

For an entire set of input samples, the loss function is computed as the average across all instances, or $J(\mathbf{w}, \mathbf{b}, \mathbf{x}) = \frac{1}{\text{samples}} \sum_{i=1}^{\text{samples}} \frac{(\hat{y}_i - y_i)^2}{2}$. To optimize the neural network (that is, the function it is approximating) for a fixed set of data, it is necessary to find the set of network weights and biases (\mathbf{w} and \mathbf{b}) that minimize the loss function. For a nonlinear and often non-convex problem, gradient descent provides an efficient method for adjusting the parameters of the network. As the name would imply, gradient-descent based approaches involve computing the derivative of the cost function with respect to the parameter of interest. The gradient provides the direction of greatest change of the loss function with respect to the given weight/bias, and by descending along this direction the loss function may be minimized. Thus, for a given iteration of gradient descent, the network weights (and biases with an analogous expression) may be updated as $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla_{\mathbf{w}_t} J(\mathbf{w}_t, \mathbf{b}_t, \mathbf{x})$, where η represents the learning rate, a term which describes how large of an adjustment should be made in the direction of the gradient. The major difficulty lies in computing the gradient of the loss function with respect to each weight. The gradient further away from the output of the network depends on the values of the weights that follow it. Despite the limitations of single layer perceptrons being known even in Rosenblatt's time, as well as how deeper networks could resolve some of the problems, lack of a procedure for adjusting the parameters of such a network meant that deep neural networks were infeasible until the popularization of the backpropagation algorithm.[23] Backpropagation provides a systematic approach to calculate the gradient with respect to the parameters at any location in a network. Applying the chain rule, derivatives computed at the output of the network are used to determine derivatives at earlier positions in the network. This process is repeated until the gradient has been computed with respect to all network weights/biases. As an example, for the MLP under consideration (Figure 2.3), the derivative of the loss function with respect to the weights associated with the hidden layer of the network, $\frac{\partial J}{\partial w_1}$, can be computed to be a function of the output of the hidden layer via chain rule as $\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial z_h} \times \frac{\partial z_h}{\partial w_1}$. While the second term in the right side may be computed directly, the derivative with respect to the raw (i.e., prior to being operated on by the activation function) output from the hidden layer is unknown and must be determined by derivatives further down the network. The full expression for the gradient at the hidden layer may be represented as $\frac{\partial J}{\partial w_1} =$

$\frac{\partial J}{\partial y} \times \frac{\partial y}{\partial z_o} \times \frac{\partial z_o}{\partial h} \times \frac{\partial h}{\partial z_h} \times \frac{\partial z_h}{\partial w_1}$, with analogous expressions for the other weights and biases at the hidden layer. Looking at the terms of the expression from left to right, the backwards propagation of errors becomes clear. First the derivative of the loss with respect to the network output is computed, then the derivative of the network output with respect to the output prior to activation, followed by the derivative of the output prior to activation with respect to the activated output of the hidden layer, and so on, moving backwards from the output until the error reaches the parameter under consideration. Through the use of backpropagation, the gradient of the loss with respect to any parameter in the network may be obtained, thus allowing for the use of gradient descent for optimization.

2.2.5 Training a Neural Network in Practice

There are, however, a few caveats. For a large dataset, computing the loss function across the entire dataset may be computationally prohibitive. Additionally, considering a modern network with millions of tunable parameters (Imagenet, a common model for image detection contains 60 million adjustable parameters[24]) modeling very nonlinear problems leads to a loss surface that is extremely-high dimensional and decidedly non-convex. Moving in the direction of the “true” gradient may lead directly into a saddle point, a point where the gradient is zero, but does not represent an extremum as the loss function increases/decreases in the neighboring regions. If parameter adjustment is based solely on the gradient, the optimization will be unable to escape this point as all step sizes will be zero. Now, rather than computing the loss function based on the entire set of training data, consider the case of performing the calculation and weight update on a subset of the data. Calculation of the gradient on this subset, which can range from size 1 (stochastic gradient descent) to the number of samples less one (minibatch gradient descent) should provide a reasonable, but noisy estimate of the true error due to variations within particular random subsets and outliers. This stochasticity can allow the network to still converge to a minimum on the loss surface, while providing it an opportunity to escape from saddle points or even shallow local minima. In fact, optimization with stochastic gradient descent *is guaranteed* to converge to a local minimum, and in the case of a convex loss surface, optimization *will* converge to the global minimum.[25],[26] Additionally, the requirement to compute the gradient only with respect to a small subset of the data is much more computationally efficient, making it well suited for large

datasets and allowing for more frequent weight updates. Thus, some form of *batch* gradient descent (with the batch size referring to the number of samples used in the gradient calculation) is observed more often than true gradient descent. There are several other commonly used optimization algorithms used in machine learning, such as Adam[27] and RMSprop¹, among others [28], but an understanding of gradient descent lays the groundwork for understanding these newer methods. These approaches build off of standard gradient descent, but typically add an additional parameter such as a regularization term to control network weight magnitudes to prevent overfitting, or momentum and momentum-like terms that mix in information about the gradient at previous timesteps to aid in convergence and avoiding shallow local minima. Regardless of the choice of algorithm, each parameter update is referred to as a “step” and a complete pass through all of the inputs is known as one “epoch”.

2.2.6 Training, Validation, and Testing Sets

As the goal of a predictive statistical model is not necessarily to maximize performance on known data, but rather to provide insights on new data, it is critical that the generalizability of a neural network be demonstrated. This demonstration is often performed by inference on a *test* set. To contrast the set of data used for training the network, the test set is completely isolated from both the network *and* those developing it. Once fully trained, the network’s performance on this unseen data illustrates how well it generalizes. A model which performs poorly on the testing set despite good performance during training is said to be *overtrained*; the model has isolated some nuances of the training set that are not present or that do not describe the data as a whole. As failure on the testing data would necessitate a complete redo of the data selection, training, and evaluation procedure, it is worthwhile to have some gauge of how the model will perform on new data. Further splitting the data into a training, testing, and *validation* set can help in this regard. The validation set is also not utilized in training the model; however, at any time the model/user can perform inference on (or look at) the data to estimate general performance. The distinction between validation and testing data is made because information from the validation set may still leak into model development; while it will not directly affect the weights of the network, it can influence

¹ Despite its prominence as an optimization algorithm for machine learning applications, RMSprop was never formally published, and instead owes its provenance to a lecture by George Hinton, linked here for reference: <https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf>

the choice of network architecture and training parameters (batch size, learning rate, number of epochs, etc.) and thus represents a *biased* estimation of generalizability.

Typical heuristics for training/validation/testing splits typically follow a 70-80/10-20/10-20 split as a percentage of the available data. In very data scarce scenarios, it may not be feasible to utilize an independent validation set as withholding this data will have notable effects on the network performance. In these situations, *cross-validation* is a frequently used technique. Although there are various formulations, a commonly observed example can be found in k-fold cross validation. Here, the data is first split into k folds. One fold is withheld for validation, and the remaining k-1 folds are utilized for training. This process is repeated for all k folds, such that all data has a chance to be in both the training and validation sets, and results are averaged across the folds to determine the set of network parameters that, in general, provide the best results.

2.2.7 Other Network Architectures

Convolutional Networks

The multi-layer perceptron is the simplest and most common form of deep neural network observed today, and an understanding of them predicates an understanding of other network types. However, the large number of parameters of deep MLPs and their corresponding training difficulty has motivated the development of other neural network architectures that more judiciously add parameters to address features common to many learning problems. Two of the most important alternative architectures for chemical data are the convolutional neural network (CNN) and the recurrent neural network (RNN).

The key feature of the CNN archetype is the use of convolutional layers. These layers, demonstrated in Figure 2.4, consist of a matrix of weights, known as a *kernel* or *filter*, in an analogous fashion to the weights of an MLP. The key difference lies in how the weights are connected to input features. For a given input matrix, the kernel will *convolve* about it. The nature of this convolution varies on the dimensionality of the input data, but typical applications see either a one-dimensional convolution (where the kernel moves either left-right or up-down across the input data) or two-dimensional convolution (where the kernel winds around the input matrix in a series of one-dimensional steps both up-down and left-right). For Figure 2.4, the convolution proceeds as follows. The kernel is first centered at the top left of the input. Element-wise

multiplication is performed across all cells where the kernel and input overlap, and these values are summed to produce the corresponding value on the feature map. The kernel then convolves to the next 2x2 submatrix to the right. Once it reaches the edge of the input matrix, it wraps back around to the leftmost side of the input matrix, now shifted one row down. This proceeds until the kernel has convolved across the entire input matrix and produced a feature map as output.

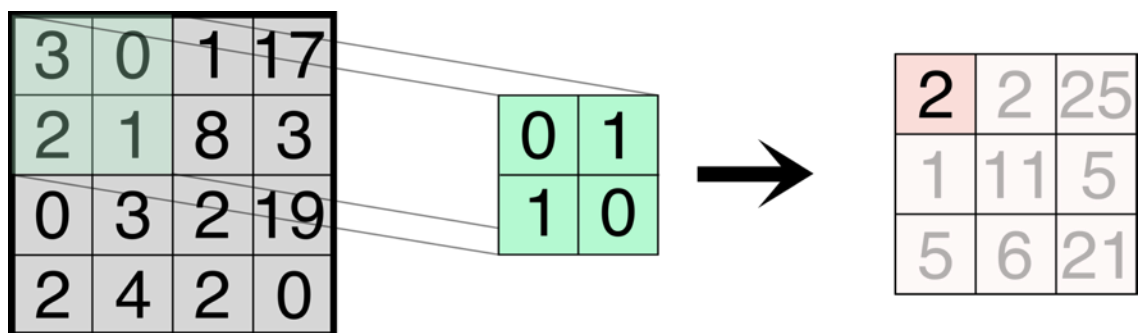


Figure 2.4: Operation of a convolutional layer with one kernel (center matrix) operating on an input matrix (left) to produce a feature map (right) as output. The highlighted green square indicates the superposition of the kernel in its first position over the input matrix and the red highlighted square in the feature map denotes the output of this operation. The remaining elements of the feature map corresponding to the full set of convolutions is also shown.

Note the nature of the output feature map. High values in the input matrix do not necessarily correspond with high values in the output. Rather, there is a spatial dependence between the input and the signal passed through the network. For this given kernel, only high values in the right diagonal will produce a high output value, such as in the third column, first and third rows in the feature map of Figure 2.4. High values in the left diagonal are not picked up, as observed in the third column, second row of the feature map. This ability to detect spatial relationships within data has been a primary driver in the development and deployment of CNNs. They have been widely employed in image detection tasks, where the filters can detect edges and other features specific to classes of objects they are attempting to detect. As the weights of a filter are updated during training, just as the weights of an MLP are updated, not only do these spatial features need not be explicitly hardcoded, they are in fact learned as a part of training so that the network can identify those features most relevant to the objective at hand (usually classification). These learned features are often not human interpretable, yet have been demonstrated to outperform expert selected heuristics.[29] For chemical data, the kernels may learn to detect spatial relationships in the input

tensor which are directly related to spatial relationships within the corresponding molecular graph. A given layer of a convolutional neural network may have multiple kernels of various sizes, allowing each filter to detect specific input that describe particular moieties, such as particular functional groups or a ring of a given size. Crucially, the use of CNNs allow these features to be automatically inferred from the data.

Recurrent Neural Networks

In addition to spatial dependence, another key relationship to expose lies in *temporal* variation, where previous inputs have bearing on future outcomes (and vice versa). Natural language processing is a classic example of a situation where it is vital to capture temporal relationships. If a given input is an adjective, for instance, the next input becomes relatively constrained as either a noun or another adjective. Without capturing this dependence, a network designed to translate languages, for instance, could make the obvious mistake of connecting word types that do not go together, or transposing words in the wrong order. Even if these trivial syntactic errors are restricted via hand selected output rules, a network could easily “forget” its place in a sentence. All of the output words could be used properly and be in the proper positions, but without any knowledge about the *context* of the input the network is operating on, the underlying meaning could be lost. Ideally, the network should be able to leverage information about its prior outputs to inform its future decisions; this is the idea behind recurrent neural networks. As the schematic in Figure 2.5 demonstrates, at each successive timestep y_i , the previous outputs are mixed in to provide a more accurate prediction.

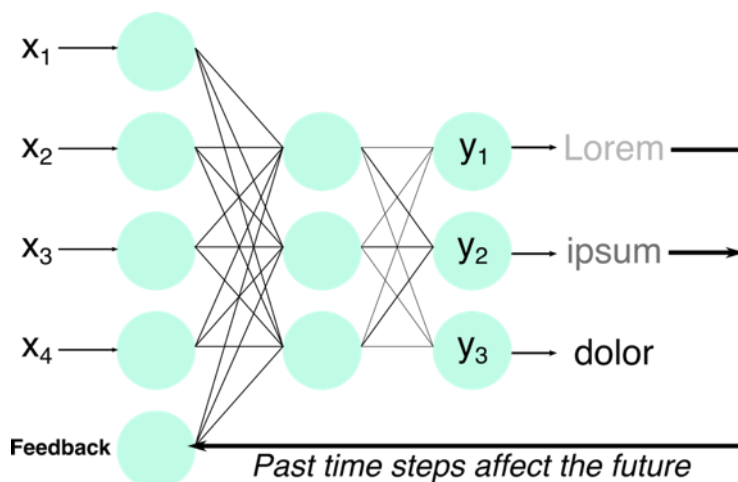


Figure 2.5: Example of a recurrent neural network architecture for text processing. The architecture shown here mimics that of Figure 2.3, but with the addition of a feedback loop that allows for network outputs at each timestep to influence subsequent output.

Although processing molecules with string representations leads to difficulties in producing meaningful output, as the characters within a representation such as a SMILES string (defined in Chapter 2.3 below) must be self-consistent, it also provides the opportunity to borrow techniques perfected for language and text processing. These include advancements in recurrent neural network design, such as long short-term memory units (LSTMs) and gated recurrent units (GRUs) that have become staples of generative chemical models.

2.3 Molecular Representations

Consider the generic neural network defined in the previous section, which accepts some numerical data as input, and through a series of weighted sums and nonlinear transfer functions outputs some numerical value such as a class or property value. A question that immediately arises is: *how can a neural network operate on molecular input?* In comparison to image recognition, where the input domain naturally consists of real valued tensors (e.g., a bitmap image), it is not obvious how to represent molecular data in a machine-readable format. While a Lewis structure is immediately recognizable to a trained person, and encodes all necessary information about bonding arrangements, atom types, aromaticity, stereochemistry, and other salient chemical features in a human interpretable fashion, it lacks uniqueness and has a spatial dependence and a particular context that would be a challenge for a machine to identify, let alone perform useful

operations with.[10] However, the drawing of a Lewis structure represents a graph structure, where the atoms are nodes and the bonds are edges between the nodes. Representations can be utilized that take advantage of this natural graph structure, such as the Simplified Molecular Input Line Entry System, or SMILES.[30] A SMILES string consists of the heavy atoms within a structure, with positional arrangement in the string corresponding to traversal through the graph across connected atoms. The string CCO would represent ethanol, c1ccccc1 would represent benzene (the use of lowercase letters represents aromatic atoms, with the “1” characters representing a ring opening/closure), and C#C would represent acetylene, as a few examples. While a given molecule may have multiple SMILES strings, a “canonicalization” convention exists to guarantee a standardized, unique SMILES for a given molecule regardless of the method used to generate it. While other, potentially more explicit string-based methods such as InChI exist, they have largely fallen out of favor for machine learning applications due to a more complex string “grammar” which requires arithmetic to fully parse and are difficult for a neural network to learn to interpret.

To then convert a 1D machine readable text format into one accessible by a neural network (i.e., a real valued tensor) a common approach is to form a “one-hot encoding”. Here, a matrix is constructed with a number of rows equal to the maximum size of the string (as neural networks require input of a fixed size) and columns equal to the possible characters in the string (e.g., “C”, “O”, “N”, “1”). Reading down a string (and thus down the rows of the one-hot encoding) a “1” is placed corresponding to the column of the observed character, and a “0” is placed elsewhere. This

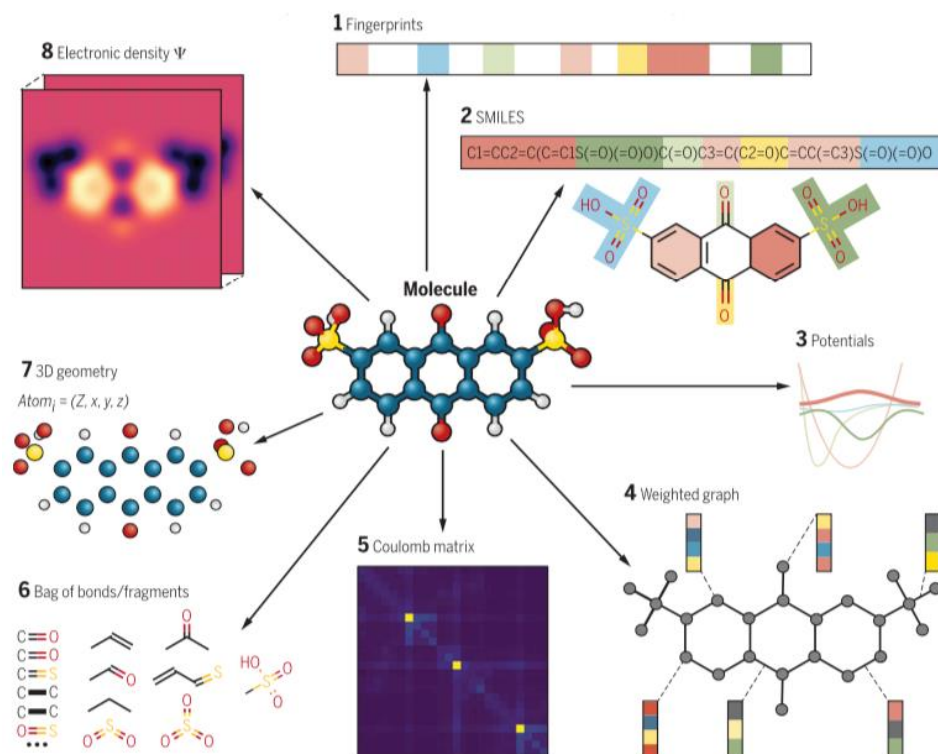


Figure 2.6: Several of the most common molecular representations used for computational chemistry and machine learning. Representations are all equivalent to the structure denoted by the ball and stick model in the center. Figure adapted from [10].

process is repeated until the entire string has been embedded in the one-hot encoding matrix, with strings that fall below the maximum length being “padded” out to the fixed size with a “blank” character that has a corresponding column in the encoding matrix. The one-hot encoding, with binary elements, can be fed directly as input to a neural network.

There are numerous other chemical representations in use, some of which are illustrated in Figure 2.6. A notable example for machine learning applications is the molecular fingerprint. A hashing function can be applied to a molecular graph to identify the presence or lack of certain molecular features, which are stored in a fixed length vector. The hashing function, and thus the relevant features, can either be calculated via a fixed function [31] or the hashing function may be learned as part of a neural network.[32] Determination of optimal chemical representations is still an active area of research.[33],[34],[35]

2.4 Machine Learning for Chemical Science

Statistical approaches can connect structural features with target properties, obviating the need for expensive physics-based simulations. Identification and application of quantitative structure-activity/property relationship (QSAR/QSPR) models have long played a role in computational design of pharmaceuticals, [36] where it was discovered that many important drug-target features could be expressed as linear (and later more complex) functions of some structural parameters, such as the pH at which extraction of antimalarials occurs as a function of molecular weight.[37] More advanced statistical methods, such as support vector machines, [38] provided the opportunity to unearth more complex structure-function relationships, from classifying molecules based on their drug-likeness [39] to the design and subsequent experimental validation of anticonvulsants. [40] Modern deep-learning approaches, however, were not known to the literature until 2012, when a deep-neural network won a competition held by Merck to predict drug targets on a set of withheld testing compounds, beating out teams of chemists and other domain experts.[36] Deep learning approaches for drug screening have since enjoyed success in the prediction of aqueous solubility of lead-compounds, [41] a wide variety of toxicity prediction problems, [42],[43],[44] and drug-drug and drug-food interaction prediction [45] as a few examples, with toolboxes and automated frameworks still an active area of research.[46] Applications beyond drug design have burgeoned in the meantime, including prediction of atomization energies [47], determination of the ability of electrolytes to inhibit dendrite formation, [48] and computational routines that may allow for the prediction of quantum chemical properties to within a greater accuracy than DFT alone.[49]

As a direct example of the potential for deep-learning to assist in the HTVS process, researchers lead by Alán Aspuru-Guzik developed a HTVS approach to design organic LEDs.[50] With a starting set of over 1.6 million candidate molecules obtained via a fragment growth procedure, traditional physics-based simulation at the required time-dependent DFT (TD-DFT) level was infeasible. To efficiently pare down the search space, the authors trained a deep-neural network to accept a molecular structure as input via molecular fingerprint and to output a prediction for the decayed fluorescence rate constant. Only those candidates falling within the specified bounds for the rate constant were considered for the next stage of the computational funnel and subjected to the expensive TD-DFT calculations. As a result of this ML-aided HTVS process, they were able to synthesize and experimentally validate the proposal of new OLED

candidates with high external quantum efficiency. This work and subsequent examples in screening for bimetallic catalysts [51] and clean energy materials, [8] among others, have proven the utility of ML as a complementary technique in the HTVS process across a variety of disciplines.

2.5 The Inverse Problem

Regardless of the usage of enhanced sampling techniques and methodologies to speed up computation, the key drawback of the HTVS process remains: it is, at its core, a trial-and-error approach. A closed set of substances is obtained that ideally contains a sufficiently optimized structure, and hopefully that structure is not missed when explicit enumeration is avoided. Even iterating on this procedure and broadening the search space at each step using the knowledge gained from previous trials still represents limited, incremental advances in chemical space. Rather than attempting to “catch” a target molecule by searching across structures until one with the desired properties is found, it would be more efficient to invert this procedure and directly determine a suitable structure displaying a given property. The idea of “inverse-design” is fairly old, with an appearance in the literature as early as 1983 in the context of protein folding.[52],[53] Here, and in subsequent work [54], the desired conformation of the protein backbone is first specified and a stabilizing amino acid is then predicted, contrasting with the traditional and difficult approach of attempting to determine the conformation of the protein given a particular sequence of amino acids. This idea was given mathematical grounding with Kuhn and Beretan’s *Inverse Strategies for Molecular Design* in 1996.[55] They considered the construction of Hamiltonians, achieved by tuning coefficients of the linear combination of atomic orbitals-molecular orbitals (LCAO-MOs), that directly optimize transition dipole moment. This approach is capable of providing information on *classes* of compounds, but it is difficult to disentangle a chemical structure, or even actionable chemical information, from the optimized Hamiltonian (e.g., inverting the Hamiltonian). Later approaches focused on the optimization of properties with respect to atomic *potentials*. [56] This approach combines the advantage of a mathematical framework, with well-defined gradients for optimization routines, together with a more clearly-defined inversion strategy; structures could be inferred such that they match the optimized potential. While theoretically allowing for the exploration of a broad range of chemical space, these approaches have the notable downside that a full optimization tends to lead to potentials that are non-invertible in that they do not correspond to a valid chemical structure.

Other approaches deemphasize the width of chemical space that may be probed and focus more heavily on producing valid chemical structures. Genetic algorithms have proven to be a popular and successful framework for materials design. These approaches are inspired by biological processes of evolution, and mimic actions such as iteratively performing “selection” of the fittest candidates (the molecules scoring highest on the objective function), followed by operations such as “mutation” (predefined bond breaking/forming operations), and “crossover” (combination of candidates or fragments from candidates) to obtain the next generation of candidate molecules.[57] They have been a part of computational materials design for decades, and have been demonstrated in applications as wide as antireflection coatings, [58] catalysts, [59] determining the arrangements of laminates to maximize buckling load, [60] and even radar absorbing materials.[61] Genetic algorithms find success even today, such as in the design of polymers with optimized bandgap and glass transition temperatures.[62] However, genetic algorithms are still relatively constrained in the amount of chemical space they can search. They rely on modifications to known compounds, and as they require reevaluation of the fitness function at every step, they may be computationally intractable for the data scarce, computationally constrained design spaces they are intended to probe. An ideal approach would combine the ability of continuous representations to operate freely within chemical space along with the more concrete physical significance and assurances of valid chemistry that genetic algorithms provide. The nascent development of such an approach is still observable in current research on deep, generative chemical models.

2.6 Deep Generative Chemical Models

2.6.1 Autoencoder

A generative chemical model is one that is capable of producing unseen molecules as output. To do so, it must learn the underlying distribution of chemical data; just as we can consider a given experimental variable as being drawn from some underlying distribution, we can also consider the same for compounds.

To train such a model, it is clear that we have to provide it with a set of training inputs, $\{x\}$, and that it should also output the same type of data. In the absence of any other labels or information, a natural thought is to simply have the expected output of the model be the input, that

is, we set $x_i = y_i$, or $x_i = \hat{x}_i$, for each input. The model now is simply learning to “generate” data by learning an identity mapping. Now, if we truly allowed the network to learn the identity function, it would not be particularly useful as a generative model (or otherwise). Instead, an *information bottleneck* is imposed on the network that prevents it from learning direct mappings from input to output. Consider the case of input data that consists of 5-dimensional vectors. For a simple autoencoder with an input layer size of 5, and an output layer size of 5 by necessity, we can consider a single hidden layer with 3 units, as presented in Figure 2.7.

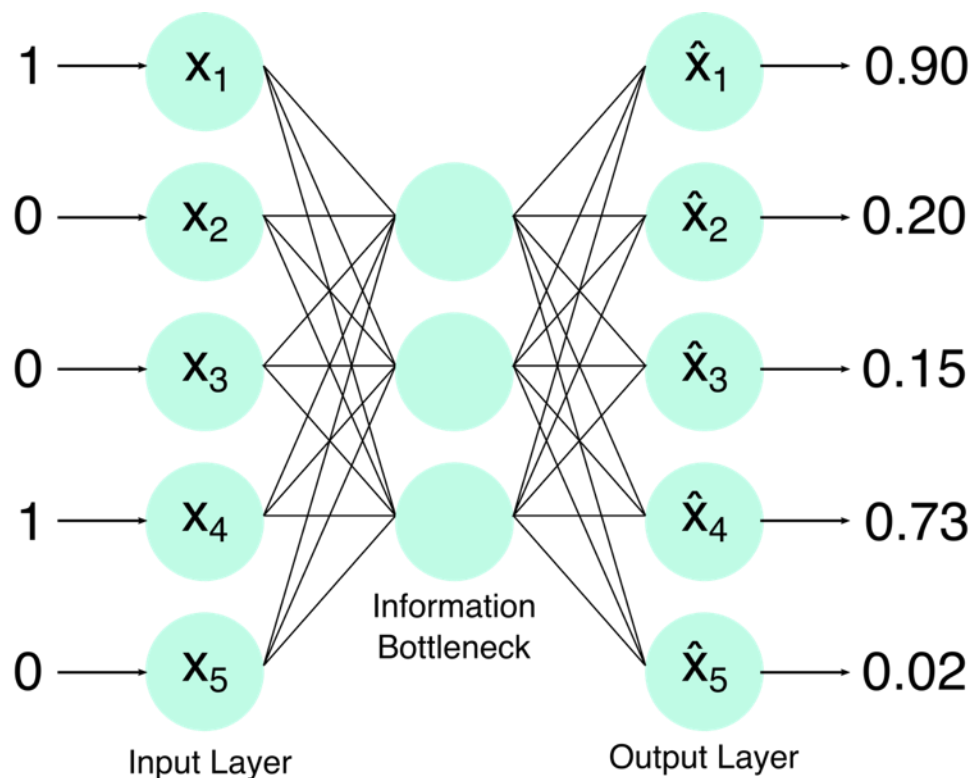


Figure 2.7: Example of a simple autoencoder. 5-dimensional bit-vectors are compressed to a 3-dimensional representation and then expanded back to full dimensionality, ideally restoring the input vector. The problem presented here is essentially a multi-label classification task, where a 1 in the input vector corresponds to presence of that class, and a 0 indicating that the input does not belong to that class. The output vector thus corresponds to a list of probabilities of belonging to each class. During training, the model attempts to adjust these output distributions to be more confident in its class predictions. Input and output vectors were chosen arbitrarily.

In this situation, the autoencoder *cannot* carry all of the information through the encoding and decoding process. Instead, it must learn an efficient way to compress the data such that it can be decompressed to reproduce the input data; hence, *autoencoder*, as the encoder and decoder are

trained in tandem to automatically find an efficient way to represent the data. The ability of autoencoder models to extract the most important latent features for describing the input data has led to their success in image denoising.

Given an input image, traditionally a handwritten digit, that contains significant levels of artifacts or other noise, the autoencoder retains the key, describable features (i.e., the digits) while discarding the features that are not descriptive of the underlying data (i.e., the unpredictable noise). The efficient compression provided by the autoencoder also provides a convenient avenue for visualization of high dimensional data in terms of a few, maximally important dimensions. This mechanism of extracting the most salient descriptors within the data also means that the latent vectors serve as informative inputs to other model types, such as classifiers or regressors based on some property of the input data. These networks may even be trained in tandem, with the effect being, due to backpropagation of error, that the encoding process will also be better suited to predict the features of interest in addition to achieving efficient encoding and decoding. In the past few years, this approach has gained traction within chemical research. Winter *et al.* demonstrated that the use of latent vectors from a chemical autoencoder as molecular descriptors were far superior to traditional descriptors such as chemical fingerprints in various molecular-virtual screening tasks.[63] Chapter 4 of this dissertation builds off of their work and demonstrates how the autoencoder and its associated latent space of chemical representations provide a unique and powerful transfer learning mechanism to facilitate information exchange between related chemical property prediction tasks for improved predictive ability of scarce experimental data.

2.6.2 Variational Autoencoder

As posed, the autoencoder framework *seems* to have the makings of a generative model. The decoder is trained to decode from continuous vectors back to elements of the original input space during training; generating new data should simply be a matter of sampling arbitrary points in the latent space and passing them to the decoder. However, first consider the form of the autoencoder latent space. There is no guarantee of continuity within the latent space; while similar input data should be encoded into points which are less distant than dissimilar input data, arbitrary points within the latent space will tend to be meaningless. The latent space is ill-conditioned in this case, as the autoencoder only tried to ensure that decoding the exact points provided to it by the encoder returns the input structure. For the generation task, that is decoding arbitrary points in

the latent space that do not necessarily correspond with a known input, the decoder is hopelessly overfit, and will not extrapolate well to new data.[64] A regularization term must be applied to improve the generalizability of the decoding procedure and provide an avenue for generative applications; the encodings must be spread out to fill up the latent space such that every position in the latent space is related to the distribution of the training data. Rather than allowing the encoder to map inputs directly to latent vectors, resulting in a sparsely populated latent space, noise can be added to the encodings such that encodings no longer correspond to discrete points, but rather a distribution. The decoder must then learn that not merely single vectors, but a distribution of vectors may all correspond to the same input structure, reducing the amount of “dead space”.

Functionally, this approach is realized by changing the output of the encoder from a single latent vector describing a particular input, to parameters (i.e. mean, variance, etc.) describing a distribution of points corresponding to the input structure.[65],[66] Stochasticity may be thus be added to the training of the autoencoder by sampling vectors from the distribution described by the encoder and passing these through to the decoder. While the form of the encoded distribution is not theoretically restricted, closed form solutions for loss functions and ease of computation are facilitated by the use of a Gaussian distribution, thus the encoder produces as output values of the mean and variance describing the given input data. During training, an auxiliary distribution is sampled to provide a noise vector, which can then be scaled by the latent variance and added to the latent mean value to efficiently “sample” the encoded distribution.

Simply training the network using this probabilistic approach to encoding and decoding will not solve the issues relating to poor decoding inference. Rather, an autoencoder trained in this way has the freedom to reduce its reconstruction loss by simply driving the encoding variances to zero (removing the noise) or adjusting the mean such that encodings are as far apart as needed to effectively isolate them. Both cases recover the standard autoencoder and remove the utility of the distribution-based approach. To prevent this behavior, the distribution of latent vectors can be assumed to follow a Gaussian distribution with zero mean and unitary covariance (again, for convenience). The divergence of the latent encoded distributions with this prior can then be *penalized as part of the training loss*, that is, distributions that are not centered lead to higher loss, and those that are either too spread out or too tight will also lead to higher loss. Mathematically, a Gaussian latent prior is assumed ($p(z) = N(0, I)$) and the KL-divergence between it and the approximate posterior latent distribution is added to the objective function. It is here that the

variational aspect arises, as a family of distributions (the latent encoding of each training element) are chosen to model the intractable posterior latent distribution, and the parameters of those model distributions (the mean and variance of each training element) are updated to minimize the divergence with the true distribution, a process also known as variational inference. The objective function of this model can now be represented as $J(\mathbf{w}, \mathbf{b}, \mathbf{x}_i) = \frac{1}{2} \sum_{j=1}^D (1 + \log(\sigma_{ji}^2) - \mu_{ji}^2 - \sigma_{ji}^2) + \sum_{j=1}^D x_{ji} \log(y_{ji}) + (1 - x_{ji}) \log(1 - y_{ji})$. The loss term of this variational autoencoder (VAE) includes both a reconstruction error term (the second summation), which encourages the autoencoder to make meaningful latent encodings to reconstruct the input data, and a regularization term in the form of the KL-divergence (the first summation), which promotes the generative potential of the model by promoting dense and smoothly varying latent distributions. After the work of Kingma and Welling formally defining the variational autoencoder as a in 2013, the stage was set for a new class of generative-ML models. However, it was not until 2018 that the variational autoencoder, and generative ML in general, was finally applied to chemical data.

2.7 The Chemical Variational Autoencoder

The “modern” age of ML-aided inverse-design is often regarded as having its origins in Gomez-Bombarelli’s seminal work *Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules*.^[67] The authors applied the variational autoencoder framework to encode and decode molecules in the form of SMILES strings, shown schematically in Figure 2.8.

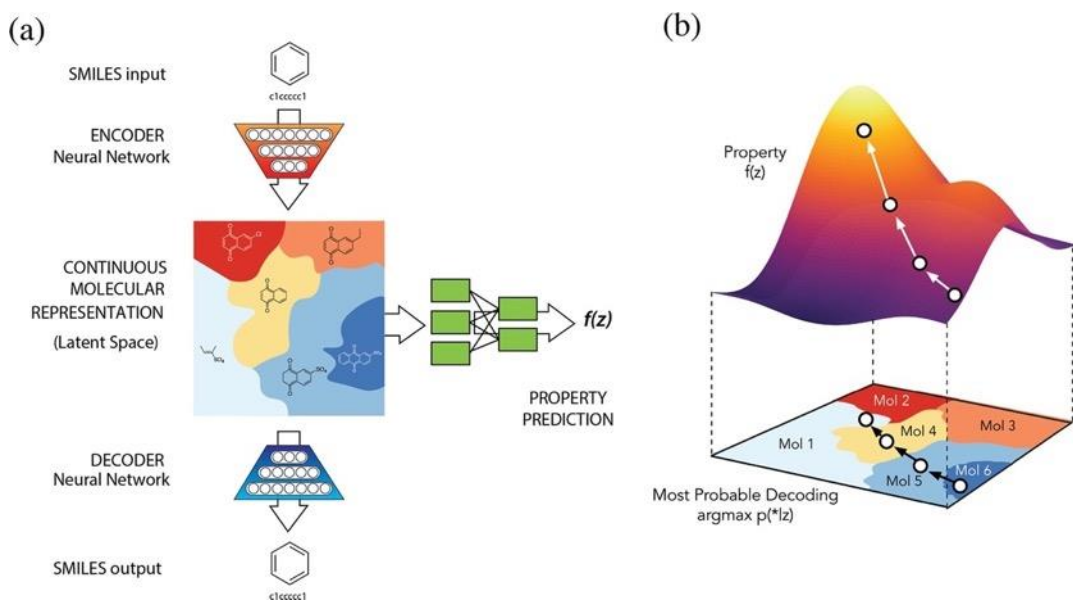


Figure 2.8: Overview of the chemical variational autoencoder framework. (a) a schematic of the model architecture. Discrete chemical structures are encoded to a continuous latent representation, and then decoded to return the original input. Vectors may be sampled from the latent space and passed to an ancillary property prediction network, which estimates properties of the corresponding molecule. (b) a schematic of the molecular optimization routine suggested by the authors. Because the chemical variational autoencoder develops a continuous chemical latent space, traditional gradient based optimization techniques can be employed to determine regions of the latent space to sample to attain optimized molecules. Figure reproduced from [67]

The projection of discrete molecules into latent distributions resulted in the formation of a continuous chemical latent space, from which arbitrary points could be decoded to yield novel compounds. Because the VAE distills key topological features for efficient encoding and decoding, the position within the latent space dictates the type of compound that will be drawn. Sampling around the neighborhood of a known compound will result in similar compounds. Sampling between two compounds will result in compounds representing an interpolation of features between the two, resulting in a gradual structural transformation moving from one known point to another. By sampling outside of the convex hull of the training set, or in sparsely populated regions of the latent space (an extrapolative sampling), the possibility opens of generating compounds structurally dissimilar to those in the training set. The ability to simply generate new molecules representing perturbations off of known compounds is still of limited advantage compared to traditional generative methods. The true advantage lies in the capabilities of having a continuous, mathematical representation of chemistry.

By training only on encoding and decoding, the organization of the chemical latent space is such that compounds with similar structural features are located closely and those with dissimilar features are farther apart. For physical properties, there is no guarantee that position in the latent space will correlate with expression of a particular property as Figure 2.9 demonstrates. The VAE architecture can be expanded to also include a property prediction network. This network accepts input from the latent space and uses it to predict the properties of the associated structure, and is

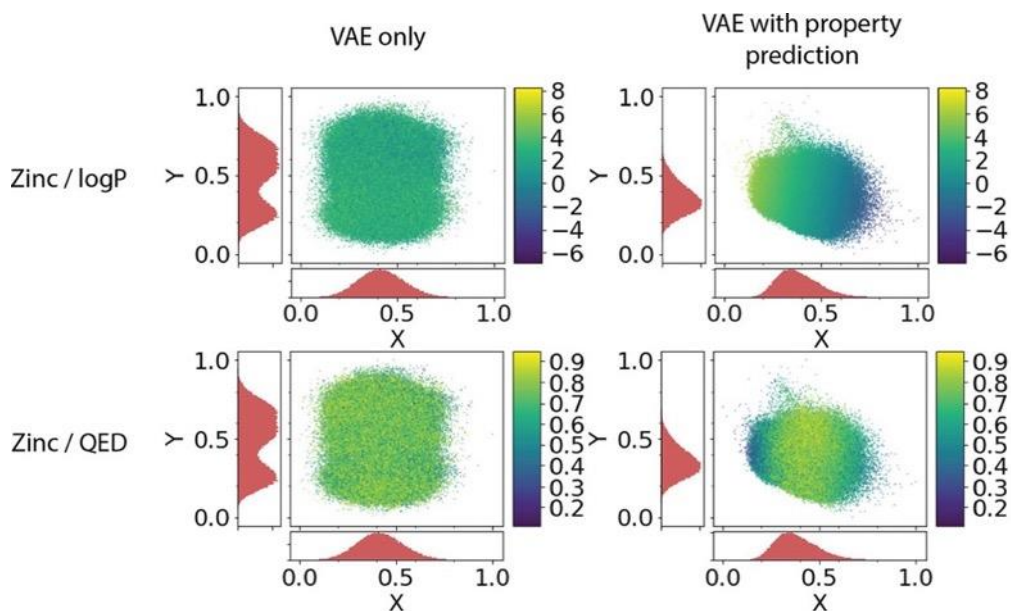


Figure 2.9: Comparison of latent space for chemical variational autoencoders trained on encoding/decoding as well as prediction of water-octanol partition coefficient and quantitative estimate of drug-likeness. The 156-dimensional mean encodings are projected down to the first two principal components for visualization. Without a property prediction task, there is little organization of compounds with respect to properties. The addition of a property prediction task based on latent encodings ensures meaningful organization with respect to properties, although as the case of QED shows this organization is not always linear. Figure reproduced from [67].

trained in tandem with the encoding and decoding process. As errors are propagated back through the network during training, errors stemming from the property prediction result in the encoder weights being updated to provide latent representations that better allow for the prediction tasks; that is, the latent encodings are now forced to explicitly contain information about the target property. Latent chemical representations thus become organized such that compounds with similar *properties* are clustered together. Because a method of characterizing new compounds on the fly is available (the predictor networks), combined with the meaningful organization of latent

encodings with respect to properties in a continuous space, optimization techniques may be applied *directly* to molecules. The authors apply Bayesian optimization techniques to their autoencoder model trained to jointly on encoding/decoding and the prediction of a drug-likeness target allowing them to determine the optimal regions of the latent space to sample to maximize the objective function. They found this approach capable of consistently producing molecules with higher drug-likeness than either a random search or a genetic algorithm, demonstrating the potential of this approach for targeted generative design.

Although the VAE approach is equipped with all of the prerequisites to facilitate true inverse design, its practical utility remains limited. The decoder, which simply outputs strings, is not guaranteed to produce strings that actually correspond with real molecules, particularly when sampling in areas away from known compounds. These “nonsense” strings are produced over 95% of the time with a general molecular search, which precipitated the development of additional networks, such as Kusner’s work on the grammar variational autoencoder, [68] a model discussed in depth in Chapters 5 and 6, as well as further network paradigms presented in Figure 2.10. Regardless of the exact architecture or molecular representation used, these networks still suffer from the same data scarcity constraints inverse-design is intended to solve. Due to the difficulty of learning efficient chemical representations and structure function relationships, these generative networks tend to require large amounts of data to train. Because of the cost associated with compiling large datasets for experimentally relevant properties (pK_a , H_r , E_a , etc.), these implementations are largely limited to simple cheminformatics properties such as the water-octanol partition coefficient, quantitative estimate of drug-likeness, and synthetic accessibility score. As these objectives may not be universally relevant, and are in fact noted as being of questionable utility for inverse-design due to the easy manner in which these simple structure-function relationships may be exploited by a generator, data scarcity significantly impinges on the utility of inverse-chemical models.[69] Improving the performance of these networks, both from a predictive (forward-problem) and generative (inverse problem) standpoint is of paramount importance, and relief of the data scarcity aspect can be accelerated through the use of transfer learning methodologies.

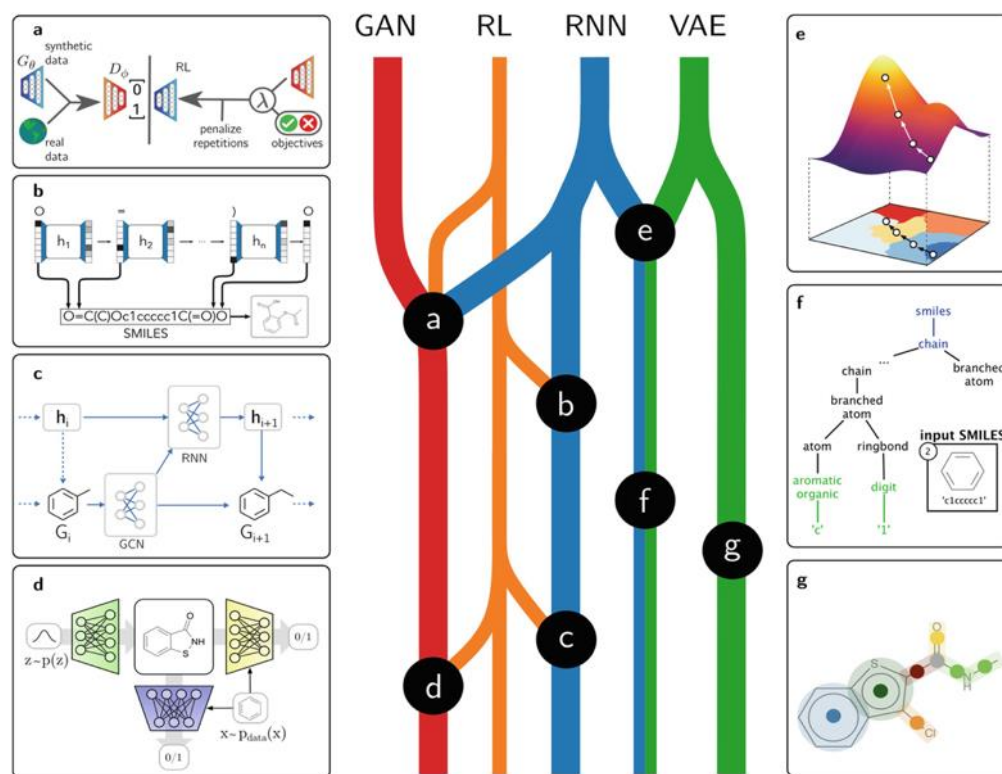


Figure 2.10: A timeline showing the development of generative chemical models, beginning with (e) the chemical variational autoencoder and followed by (a) generative adversarial networks, (b) recurrent neural networks, (f) the grammar variational autoencoder, (g) graph-based VAEs, (c) graph-based RNNs, and (d) graph-based GANs. Figure reproduced from [12].

2.8 Transfer Learning

In the broadest of terms, transfer learning is a technique whereby information learned in one task is *transferred* to help in another related task.[70] Considering the case of human learning, where information learned in past experiences motivates behavior in new situations, it is natural to postulate on whether the same principle may be applied to ML. Historically, this has taken the form of what is alternatively known as fine-tuning. After training a network on a task with abundant data, such as predicting instances of certain objects in standard computer vision databases, the network weights are further updated using the few training samples available for the object class that is desired to be detected in practice. This approach can be utilized for chemical data in the form of multi-fidelity training. While experimental data is laborious to collect and calculations at high levels of theory can require extensive time and computational resources, cost-effective calculations at lower levels of theory can often be calculated in abundance to serve as surrogates.

How can this multi-fidelity data best be leveraged? (Note: there is a growing schism in the literature on whether or not multi-fidelity and more generally, multi-task training should be considered a subclass of transfer learning or a different mechanism altogether. In this text, I have chosen to use the term to refer to any technique design to facilitate information exchange across various tasks) As Ramakrishnan *et al.* note, the majority of the operant physics involved in a single-point energy calculation are already accounted for in classic approaches such as Hartree-Fock or very efficient semi-empirical methods such as GFN2-xTB, producing results with only a 10% deviation from chemical accuracy in some cases.[71] Accounting for the remaining discrepancy is where the bulk of the theoretical difficulty and computational time are found. Attempting to train a statistical model to accurately predict the results of wave-function based methods solely from a chemical structure, particularly in situations where training data is limited by the difficulty in running such calculations, is an unreasonable proposition. The model must learn both the core-physics describing the majority of the property of interest (i.e., the contributions from lower levels of theory) *and* the complexities and nuances that fully explain that property (i.e., the remaining contributions/deviations from higher levels of theory) all from limited training data. The idea behind the difference-model approach to transfer learning is to instead attempt to predict the *difference* between the higher-fidelity and lower-fidelity approaches. The difficulty associated with predicting the bulk contributions to the property of interest are abstracted away, either by means of simply performing the lower-level calculation on the fly (if it may be computed quickly enough) or by training an auxiliary model to predict the low fidelity value which is available in abundance and should be comparatively easier to train. The target high-fidelity value can then be computed directly from chemical structure as the sum of the output of the difference model plus the output of the low-fidelity approach. Difference learning, also known as Δ -ML in chemistry applications, provides a robust approach to leveraging multi-fidelity data, and has been used successfully in the prediction of covalent binding-energies as well as isomerization and atomization enthalpies to within chemical accuracy at the computational cost of DFT or even semi-empirical methods with enough training data. Recently, it has proved integral to the prediction of vibrational spectroscopic maps [72] and NMR-chemical shifts with greater precision than DFT, [73] accurate potential energy surfaces, [74],[75] and an efficient means of obtaining accurate orbital energy data.[76],[77].

Despite these impressive results, there are situations in which difference models are not effective. Difference models are not amenable to classification tasks (e.g., predicting a class of reactions, or binary labels such as toxicity or mutagenicity), nor are they suitable for situations where abundant data is available but for different goals (e.g., different quantum chemical properties) where the difference in values has little physical meaning and no anticipated correlation. In these situations, it was discovered that training on multiple properties at once actually improves the performance of *all* properties under consideration, so long as the data all arise from the same underlying relationships.[78] Intuitively, all chemical properties are fundamentally related to chemical topology, so learning structure-function relationships for one property will help in identifying connections for another. These multi-task approaches are well suited for molecular screening, as there are often multiple properties that must be accounted for simultaneously. Ramsundar *et al.* used this approach to improve prediction accuracy for up to 259 drug-design targets and found that increasing both the number of tasks and the amount of data for each resulted in continuous improvement of all associated prediction tasks.[79] Subsequent studies proved the robustness of the multi-task approach to changes in model architecture, from statistical models like random forests to deep learning approaches, and the identities of the target tasks by demonstrating strong performance on pharmacology datasets which had hitherto never been analyzed in the literature. [80] Chapters 3 and 4 of this work demonstrate how the use of variational autoencoders with a continuous chemical latent space provide new and effective avenues for transfer learning on chemical data, whereby the learned *representations* of chemistry are modified by the inclusion of new data, leading to better predictive accuracy on limited experimental or computational data. These improved representations are further explored in Chapter 5, which demonstrates how multi-target approaches can improve the performance of generative models by transferring structural information contained in chemical property data to help in the generation of new structures optimized for a separate set of application properties.

3. IMPROVED CHEMICAL PREDICTION FROM SCARCE DATA SETS VIA LATENT SPACE ENRICHMENT

Reprinted with permission from J. Phys. Chem. A 2019, 123 (19), 4295-4302. DOI: 10.1021/acs.jpca.9b01398 Copyright 2019 American Chemical Society.

Modern machine learning provides promising methods for accelerating the discovery and characterization of novel chemical species. However, in many areas experimental data remains costly and scarce, and computational models are unavailable for targeted figures of merit. Here we report a promising pathway to address this challenge by using chemical latent space enrichment, whereby disparate data sources are combined in joint prediction tasks to enable improved prediction in data-scarce applications. The approach is demonstrated for pK_a prediction of moderately sized molecular species using a combination of experimentally available pK_a data and DFT-based characterizations of the (de)protonation free energy. A novel autoencoder framework is used to create a continuous chemical latent space that is then used in single and joint training tasks for property prediction. By combining these two datasets in a jointly-trained autoencoder framework, we observe mutual improvement in property prediction tasks in the scarce data limit. We also demonstrate an enrichment mechanism that is unique to latent space training, whereby training on excess computational data can mitigate the prediction losses associated with scarce experimental data and advantageously organize the latent space. These results demonstrate that disparate chemical data sources can be advantageously combined in an autoencoder framework with potential general application to data-scarce chemical learning tasks.

3.1 Introduction

Modern machine learning provides promising methods for accelerating the discovery and characterization of novel chemical species. These include generative networks, [81] QSAR based property prediction tasks, [82] and convolutional networks among others.[32],[10] In a typical supervised learning application, molecules are pre-processed into a machine-readable representation, paired with experimental and/or computational properties, and then one of a variety of machine learning methods are applied to develop a model (e.g., a set of weights, network topology, and activation functions for use in a prediction task). Every aspect of this paradigm is

under active investigation, from understanding the optimal representation of molecules for specific applications, [83],[84],[85],[86],[87] optimal model architectures, [88],[89],[90] optimal training algorithms, [91],[92],[68] and increasing model interpretability.[49],[93],[94] Despite the early stage of much of this research, accomplishments have already been achieved in many areas, including drug activity prediction, [95],[91] de-novo protein design, [96],[97],[98] electronic structure prediction, [99],[47],[71] representation of complex energy surfaces, [100],[101] and materials discovery.[102],[48],[103],[104]

In the current paradigm, data scarcity represents a formidable challenge [105] since enormous amounts of data are typically required for model training.[106] High-throughput physics-based calculations represent a pathway around the problem of experimental data scarcity for molecular properties that can be calculated directly from quantum chemistry and classical simulation.[50] For example, large DFT-based datasets for crystalline materials and small molecules have been used to predict novel catalytic surfaces, [107] provide lead optimization of organometallic compounds, [108] discover new ion conductors, [109],[110] and design new analytes and catholytes for flow batteries, [111] among other applications. There has also been a decades long effort to utilize small molecule docking data from classical molecular dynamics and Monte Carlo for use in drug discovery.[112] Cheminformatics characterizations also provide an inexpensive means of generating large datasets that are amenable to deep learning, albeit with shortcomings associated with low and inconsistent accuracy.[113]

Despite these promising advances, many critical molecular and materials properties are not amenable to direct calculation and have associated experimental data that is intrinsically scarce. For instance, soft materials properties are strongly sensitive to processing leading to poor reproducibility of what is constitutively the same material, and also unavailability of detailed molecular structures that can be used in computation.[114] Likewise, materials properties related to stability and failure involve multiple chemical processes occurring over many timescales that are not accessible via direct simulation.[104] In such applications, it is more common for computational studies to generate correlates of materials figures of merit (e.g., reorganization energies and densities of states for organic semiconductors, or activation energies and minimum energy pathways for degradation reactions) that can supplement scarce experimental data. In this context, transfer learning paradigms that effectively combine disparate data sources to improve prediction accuracy are highly advantageous. Examples of transfer learning include multitask

models (i.e., individual models trained on multiple prediction tasks), [95]-[79] difference models (i.e., models trained on calculating differences between different properties), [115] and latent variable models (i.e., composite models where outputs from learning tasks feed into one another).[76]

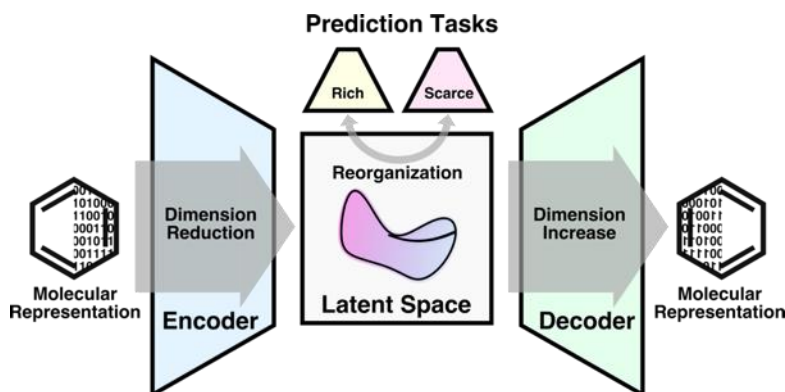


Figure 3.1: Autoencoder architecture displaying continuous latent space and joint prediction tasks. The hypothesis investigated in this work is whether joint training of the latent space on a data-rich prediction task can improve the performance of a correlated data-scarce prediction task through the enrichment of the common latent space variable.

Here we investigate the potential of a novel autoencoder machine learning approach for transfer learning (Fig. 3.1). An autoencoder comprises two jointly-trained models: an encoder, which maps a discrete chemical representation into a continuous vector space, as well as a decoder that converts points in this continuous space back into a discrete representation. The encoder and decoder are trained to minimize the reconstruction error associated with encoding and decoding operations, thus learning efficient representations of molecular structures in a continuous vector space (i.e., the chemical latent space). As was recently demonstrated by Gómez-Bombarelli et al., joint training of the autoencoder with a latent space prediction task results in a reorganization of the latent space such that molecules become organized according to predicted properties.[67] When combined with the ability to decode latent space vectors to new compounds, this approach has exciting potential for molecular discovery. Furthermore, this approach also presents a unique opportunity for transfer learning, since the latent space can be trained on several prediction tasks, thus yielding a framework for combining disparate and sparse chemical data to mutually improve prediction tasks. In this work we have tested the performance of an autoencoder in transfer learning for pKa prediction of small molecules using a combination of experimental and quantum chemistry

data on several hundred species. The autoencoder was trained in combination with multilayer perceptron models, both separately and jointly, on prediction tasks utilizing training sets of varying size. We then examined the prediction accuracy of these models on sets of unseen compounds, as well as corresponding trends within the resulting chemical latent space. In all cases, transfer learning yields improved predictions over single prediction training, with dramatic results in the scarce data regime. The results demonstrate a significant potential for using the autoencoder latent space as a shared variable in transfer learning applications on scarce chemical datasets.

3.2 Computational Methods

3.2.1 Databases

For autoencoder training, we obtained approximately 160 million compounds from the ZINC15 database with molecular weights between 200 Daltons and 500 Daltons and partition coefficient between -1 and 5.[116] These compounds were then screened using RDKit to strip salts and stereochemistry designations, and remove nonorganic compounds and those with less than 3 heavy atoms.[117] From the nearly 100 million remaining compounds, we selected a random 128,000 compound subset for use in autoencoder training. Training sets of similar size have been utilized in previous work,[67] although recently autoencoders trained on substantially larger training sets (72 million compounds) have been demonstrated.[63]

For the property tasks, we obtained experimentally determined pKa values for 819 organic acids and bases from the Handbook of Chemistry and Physics, 98th edition.[118] Compounds with more than 40 atoms were removed, as well as ions and species that exhibited (de)protonation steps requiring structural rearrangement unforeseeable by the automatic protomer generation routine. This process led to the removal of 92 species in total. Quantum chemistry characterizations were performed on the neutral, protonated, and deprotonated species associated with the remaining compounds. At this stage, an additional 56 compounds were removed due to self-consistent field (SCF) convergence failure of their associated anionic deprotonated species; these were species where the deprotomer algorithm (see below) yielded chemically unstable structures. The remaining 673 compounds were successfully subjected to complete quantum chemistry calculations. From this set of 673 compounds, 30 random testing and training splits were

constructed for model training and evaluation, each consisting of 131 and 512 compounds, respectively, for property prediction tasks.

3.2.2 Quantum Chemistry Methods

DFT-based predictions of pKa are sensitive to conformer selection, solvation model, and choice of (de)protomer.[119][120][121] In practice, these choices are often motivated by heuristics and factorial testing on specific chemical species, as no high-throughput generally applicable method currently exists. In the current study, we applied an algorithm to generate the all-trans conformer of each species to use as the initial geometry. These all-trans geometries were further refined with the Universal force-field [122] before being used in quantum chemistry calculations. Based on these geometries, the semi-empirical Geometry, Frequency, Noncovalent, eXtended Tight Binding method (GFN-xTB), was applied to energetically screen (de)protomers.[123][124] The energetically most favorable deprotonated and protonated forms, based on GFN-xTB, were then selected for further quantum chemistry analysis. As noted above, for 56 compounds we were unable to converge the SCF of the resulting deprotomer and these compounds were removed from the data set.

After (de)protomer generation, DFT geometry optimization and frequency calculations were performed on all species to obtain the inputs to the free energy calculations. Geometry optimizations were performed at incrementally increasing levels of theory to robustly converge the geometries. Initial optimization was performed using B97/def2-SVP exchange-correlation functional and basis set, respectively, followed by optimization at the B97/def2-TZVP level of theory, and a final geometry optimization and frequency calculation at the ω B97XD3/def2-TZVP level of theory with the CPCM solvation model. In the case of anions, augmented versions of the listed basis set were used. A dielectric of 80.7 was used for the CPCM model to approximate water. All DFT calculations were performed in ORCA 4.0.1.[125]

The free energy of protonation, ΔG_{+H} , and deprotonation, ΔG_{-H} , were calculated on the basis of the optimized single point energies of the neutral compounds and protomers, with zero-point energy corrections and standard statistical mechanical corrections for the finite temperature enthalpy and entropy.[126] For comparison with the experimental pKa values, we note that the Handbook of Chemistry and Physics does not explicitly state whether reported values correspond to the neutral species (acidic) or the conjugate acid (basic). To resolve this ambiguity during model

training, all species were classified as acidic or basic based on the relative free energy change, $\Delta\Delta G$, associated with protonation and deprotonation:

$$\Delta\Delta G = \Delta G_{-H} - \Delta G_{+H}$$

Compounds which displayed a positive $\Delta\Delta G$ were classified as acids, those with negative $\Delta\Delta G$ were classified as bases, and the corresponding $\Delta G_{\pm H}$ value of (de)protonation was used in the joint training activities. The single point energy, E_{sp} , of each optimized neutral species was also retained for each compound. $\Delta G_{\pm H}$ is a known correlate of pK_a (Fig. 3.2A) for chemically similar compounds, whereas E_{sp} does not correlate with pK_a and is used as a control property (Fig. 3.2B).

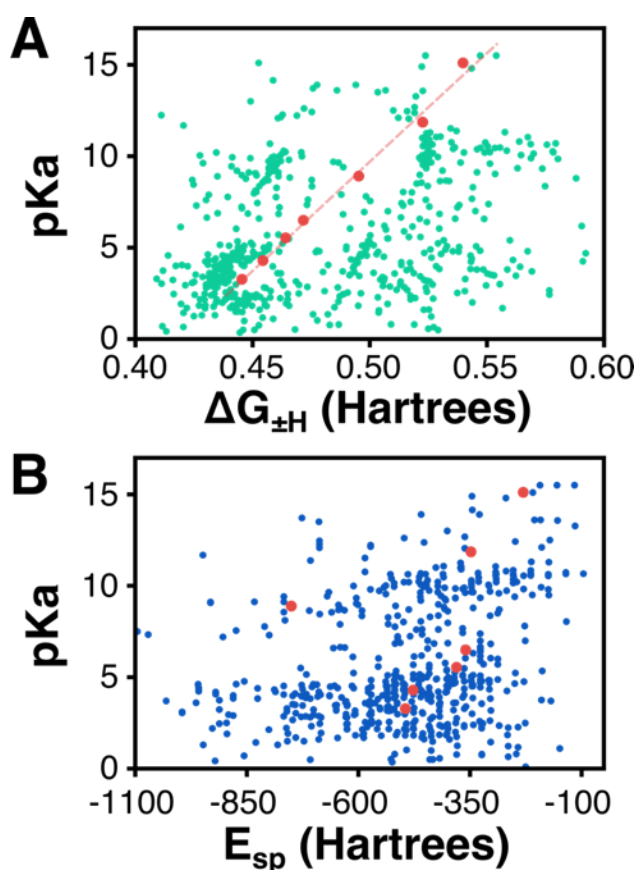


Figure 3.2: Correlation plots of (A) $\Delta G_{\pm H}$ and (B) E_{sp} with pK_a . For chemically similar compounds, it is possible to predict pK_a as a linear function of $\Delta G_{\pm H}$ to a high degree of accuracy, as demonstrated by the regression of 7 amines (highlighted in red, $R^2 = 0.98$). However, without *a priori* knowledge of chemical structure, such a strategy fails in the general case, as shown by the remainder of uncorrelated points in green. In contrast, E_{sp} does not exhibit correlation with pK_a and is used here as a chemically significant, but uncorrelated, control property.

3.2.3 Machine Learning Architecture

Models were constructed using the Keras [127] 2.1.5 API with Tensorflow [128] 1.4.0 backend. Our choice of autoencoder architecture was informed by the recent work of Winter et al., which showed excellent performance of the resulting latent space in prediction tasks.[63] For the encoder portion of the network, SMILES strings were taken as input in batch size of 128 and converted into character index arrays, where each integer value from 1 to 44 indicated a valid SMILES character. These arrays were then padded out to a uniform length of 80 characters (the maximum length of SMILES strings within the pK_a dataset is 75 characters) and converted into one-hot representations, which were flattened and used as input sequences to a gated recurrent unit (GRU) layer with 512 cells. The output of this layer was fed to an additional GRU layer with 1024 cells, followed by a subsequent GRU layer with 2048 cells. The outputs of the three GRUs were concatenated and used as input to a dense layer which produces a 512-dimensional projection of the input compound in the chemical latent space. Points in this high dimensional space were used as inputs to a decoder with an identical architecture to the encoder but reversed in order, to produce an output array containing the probability distribution of the 44 valid SMILES characters. Following the procedure of Winter et al., the autoencoder training was performed for converting from noncanonical SMILES to canonical SMILES; this ‘translation’ method was found to exhibit better performance in property prediction than a simple SMILES-to-SMILES reconstruction.

The 512-dimensional latent space was in turn used as an input to feed-forward property prediction networks for pK_a , $\Delta G_{\pm H}$, and E_{sp} . These networks were comprised of a series of two fully connected layers of 512 and 128 nodes, respectively, and one output node, producing a single scalar output. A separate prediction network of this form was trained and utilized for each property.

3.2.4 Model Training

Before being utilized in property prediction tasks, the autoencoder was first pretrained on a large subset of the ZINC15 database in order to learn efficient chemical encodings and decodings. The network was trained to minimize the categorical cross entropy of the predictions using the Adam optimizer with an initial learning rate of 5×10^{-4} , set to decay by a factor of 0.9 every 3 epochs for a total of 100 epochs. Noise was added to the one-hot encodings from a zero-centered normal

distribution with standard deviation of 0.05, and a dropout fraction of 0.15 was applied to the inputs to the first GRU to reduce overfitting.

The pretrained autoencoder was used as a starting point for joint-training with the prediction networks. At this stage, in addition to minimizing the categorical cross-entropy of the autoencoder, the model was also trained to minimize the mean squared error (MSE) in property prediction. The Adam optimizer was utilized without a learning rate decay for a total of 300 epochs using a batch size of 64, 32, and 16 for the models trained using 512, 256, and 128 pK_a values, respectively (these datasets are further described in the next section). The learning rate was reduced to 2×10^{-4} and dropout at a 0.5 rate was applied between the fully connected layers to reduce overfitting. To assist in the hyperparameter search, we utilized 8-fold cross validation on a separate testing/training split before utilizing these parameters on the remaining splits (See Appendix A). During all joint training tasks, we observed that more predictive models were obtained by first jointly training with respect to pK_a only before introducing the second property into training.

We considered three paradigms for the pK_a prediction task: prediction of pK_a only, prediction of pK_a and $\Delta G_{\pm H}$, and prediction of pK_a and E_{sp} of the neutral species. Training the network on both pK_a and $\Delta G_{\pm H}$ was anticipated to generate a more representative latent space, organized according to correlated properties, and provide better prediction accuracy. E_{sp} is chemically relevant but is not anticipated to be correlated with pK_a , so joint training with the single point energy was taken to be diagnostic of the effect of using correlated properties in transfer learning.

3.2.5 Latent Space Enrichment

To investigate the transfer learning potential of the latent space in scarce data applications, the prediction models were also trained with variable amounts of input data. A scarce data set was generated from the 512 compounds by randomly selecting 128 compounds, and an intermediate sized data set of 256 compounds was generated by adding an additional 128 compounds to the scarce dataset. On these reduced sets, we investigated the potential for latent space “enrichment”, whereby models were trained on limited pK_a data, while having access to the full set of computationally derived data (i.e., for all 512 compounds). This situation is typical of applications where limited experimental data exists, while computational data on unseen compounds can be

calculated. During training of the enriched latent spaces, the loss weights for the pK_a prediction task of the subset of hidden data was set to 0; the net effect was that the network was only able to access the computational data for that particular compound, and the pK_a value was treated as missing.

3.3 Results and Discussion

The pK_a prediction results for models utilizing varying amounts of data, as well as latent space enrichment, are presented in Figure 3.3. The results consist of a summary of mean absolute prediction errors (MAE) on the 30 test sets for each of the 30 individually trained models.

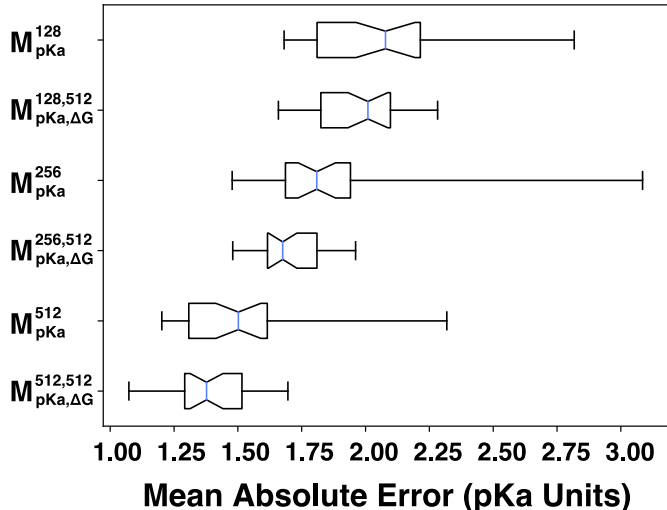


Figure 3.3: Box and whisker plot showing errors statistics for final models. 30 independent testing/training splits were generated, and independent models were trained on each split. The box and whiskers thus correspond to the MAE across the 30 test sets for each of the model paradigms. Median pK_a prediction errors are shown as a blue line, and whiskers are drawn to extend to the range of observed errors. The notches represent the 95% confidence interval about the median.

Comparing the models that were trained on pK_a alone (i.e., M_{pKa}^{128} , M_{pKa}^{256} , and M_{pKa}^{512}) we observe, as expected, consistent improvement in median prediction accuracy as the dataset grows, up to a final value of 1.5 pK_a units for M_{pKa}^{512} . Notably, attempts to train feed-forward multilayer perceptron networks (i.e., latent-space free models) utilizing a similar number of parameters failed to achieve the same accuracy; such networks were unable to find a correlation between input and































pK_a and instead return the mean value of the pK_a for every queried compound. For additional comparison, the pK_a of compounds in the full dataset ranges from -3.8 to 15.5. Interestingly, as the data size associated with each model increases, we do not observe a contraction in MAE range of the pK_a models across different training/test splits; both the M_{pKa}^{128} and M_{pKa}^{512} models display ranges of prediction errors of about 1.1 pK_a units, while the M_{pKa}^{256} model displays a range of 1.6 pK_a units. The large prediction variance suggests that the predictor model complexity is relatively large for all of these datasets (i.e., all samples are in the data-scarce regime) and that there is an increased tendency to memorize data and exhibit poor transferability for the smaller datasets. For the case of M_{pKa}^{256} , the large range may be due to some data splits being particularly difficult to train on despite the median error decreasing.

To investigate the influence of multi-task training on prediction accuracy we also trained models on both pK_a/ΔG_{±H} and pK_a/E_{sp}, while keeping the amount of data for each property constant (i.e., $M_{pKa,ΔG}^{512,512}$ in Fig. 3.3 and $M_{pKa,sp}^{512,512}$ in Fig. A3). In comparison with the pK_a-only model, we observe overall better prediction accuracy for both $M_{pKa,ΔG}^{512,512}$ and $M_{pKa,sp}^{512,512}$ with median MAE values of 1.37 and 1.40, respectively. We note that ΔG_{±H} is a correlate of pK_a, while E_{sp} is chemically relevant but uncorrelated to pK_a. Therefore, the small improvement of $M_{pKa,ΔG}^{512,512}$ compared with $M_{pKa,sp}^{512,512}$ could be explained by the model benefiting from learning on a correlated prediction task, although the effect in this case is small. Likewise, the much lower variance of both multi-task models compared with M_{pKa}^{512} suggests that adding chemically relevant data of any kind provides improvement on data-scarce property prediction.

Although it is expected that the models utilizing more data would also exhibit better predictions, in a many applications additional data generation for all properties is unfeasible. It is more typical that computational data exceeds experimental data and is used to prioritize synthetic targets. To investigate this we also trained models where multi-task training was performed on both pK_a/ΔG_{±H} and pK_a/E_{sp} but with limited amounts of experimental pK_a data (i.e., $M_{pKa,ΔG}^{128,512}$, $M_{pKa,ΔG}^{256,512}$ and $M_{pKa,sp}^{128,512}$, $M_{pKa,sp}^{256,512}$ in Fig. 3.3 and Fig. A3, respectively). Since both training tasks are performed on a common latent space, we refer to these as “enriched” models where the computational data is used to train regions of the latent space that have not been sampled experimentally. Across all enriched models we observe improvement in median MAE predictions.

Likewise, we observe a strong contraction in the variance of the enriched models, (e.g., M_{pKa}^{128} and M_{pKa}^{256} exhibit MAE ranges of 1.1 and 1.6, respectively, while their corresponding enriched models have ranges of 0.6 and 0.5) suggesting that enrichment helps prevent memorization in the scarce data regime. Similar to the jointly trained models described above, we also observe that the enriched models that utilize $\Delta G_{\pm H}$ data exhibit a small but consistent improvement in median MAE compared with the models utilizing E_{sp} data.

Table 3.1: pK_a prediction results before and after enrichment for test sets associated with median performance of each enriched model. The three sections of the table, shown in gray, blue, and dark gray, represent 128, 256, and 512 pK_a values, respectively. The expected and predicted pK_a values are shown with MAE in pK_a prediction across each test set at the bottom. The presented compounds were selected by ordering the test set according to increasing absolute prediction error and selecting 10 compounds at equally spaced intervals. Aside from the very best performing compounds, enrichment tends to improve pK_a predictions across all compounds.

	M_{pKa}^{128}	$M_{pKa\Delta G}^{128,512}$	Expected pK_a		M_{pKa}^{256}	$M_{pKa\Delta G}^{256,512}$	Expected pK_a		M_{pKa}^{512}	$M_{pKa\Delta G}^{512,512}$	Expected pK_a
	4.61	3.05	4.62		2.33	2.09	2.33		9.87	9.21	9.85
	4.68	4.02	4.24		4.79	4.41	4.59		0.78	1.34	0.66
	4.79	3.41	4.08		3.08	2.99	2.69		3.52	3.78	3.81
	1.85	0.61	0.75		4.98	4.41	4.34		4.17	4.87	4.62
	9.94	7.83	11.3		1.38	1.86	2.37		3.07	3.65	3.66
	3.69	2.83	1.77		11.13	13.18	12.5		2.94	3.13	3.74
	8.84	5.73	6.35		3.96	3.36	1.92		9.46	9.55	10.6
	8.39	7.67	4.71		-0.29	0.52	2.52		8.74	8.21	7.15
	6.84	4.47	1.9		6.59	6.93	2.49		2.46	4.46	5.23
	11.27	10.52	-1		13.43	8.63	3.33		12.21	3.71	3.3
MAE	2.29	1.82		MAE	1.81	1.55		MAE	1.23	1.07	

pK_a prediction results for a subset of the test compounds for each model are reported in Table 3.1 (individual prediction results for all associated test compounds are reported in the supporting information). Note that as we have generated 30 distinct testing/training splits, we have 30 distinct models for each of the 9 training paradigms. In an effort to consider the most

representative systems, we selected three splits with prediction errors within the interquartile range (IQR) for the unenriched models (M_{pKa}^{128} , M_{pKa}^{256} , M_{pKa}^{512}) and used those same splits for evaluating the jointly-trained models with the same pKa training data. First, we note that compounds with poor representation in the training set are likewise poorly predicted by all models. For instance, thiourea is the only example of a thiourea in the pKa dataset, and as such predictions for the pKa of thiourea display large errors (>10 pKa units), although enrichment improves the prediction slightly. Note that the autoencoder networks are trained on a much larger set of structures (128,000 compounds) than is used for the property prediction tasks. Hypothetically, the latent space developed from this larger dataset encodes chemical relationships that could improve property predictions in latent space regions that are unrepresented in the property training data but reside along interpolation contours of chemically similar training compounds. Although there is an acidic carbamide in the pKa training data it does not appear that it is chemically similar enough for most of the models to predict the behavior of thiourea. This behavior may not generalize to data rich applications and may also be affected by the size of the auto-encoder training data; however, it points to the deeper issues of the extent to which chemical relationships are encoded in the organization of the latent space and the ability of models to infer chemical relationships that are not directly represented in property training data. Second, we also note that some of the models show poor performance even on well represented molecules. For example, M_{pKa}^{512} exhibits anomalously poor performance on nicotinamide despite the prevalence of amides and nitrogen containing heterocycles in the training data. Significantly, this is rectified through multi-task training with $\Delta G_{\pm H}$ ($M_{pKa, \Delta G}^{512, 512}$) reducing the error from ~ 9 to ~ 0.4 pKa units. This suggests that hyperparameter optimization would likely improve prediction results, although exhaustive model optimization is beyond the scope of the current study.

To gain insight into how enrichment reorganizes the chemical latent space, principal component analysis (PCA) was performed on the latent space encodings for each model. The 673 compounds in the dataset were then encoded into vectors in the 512-dimensional vector latent space, standardized and analyzed by PCA, then projected on the first two principal components and plotted with respect to the training properties. The organization of each latent space is quantified by the coefficient of determination, R^2 , and the Spearman rank order coefficient, ρ (calculation details for each metric are in Appendix A). The R^2 value characterizes the proportion of the property variance that is predicted by the first two principal components. The ρ value is a

metric of monotonicity between variables, with values ranging from -1 to 1 (perfectly monotonically decreasing and increasing, respectively). High values of ρ indicate that the property of interest consistently increases or decreases along a particular direction in the latent space, which is advantageous for generative applications.

The results of this analysis are displayed in Fig. 3.4 for the jointly trained pK_a and $\Delta G_{\pm H}$ models (corresponding plots for other models are shown in the SI). In the enriched models ($M_{\text{pKa},\Delta G}^{128,512}$ and $M_{\text{pKa},\Delta G}^{256,512}$) we observe mutual organization of the latent space according to both pK_a and $\Delta G_{\pm H}$, with the data-rich property exhibiting much higher R^2 and comparable ρ .

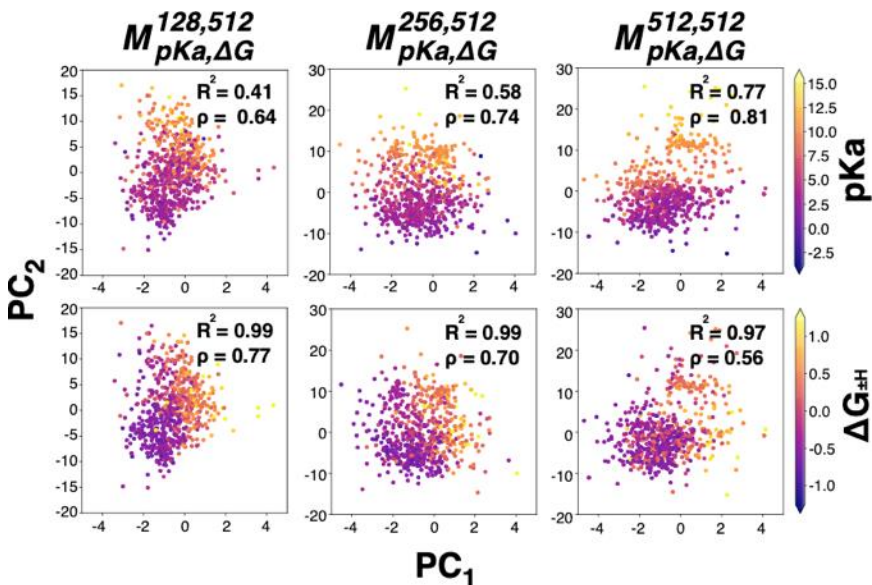


Figure 3.4: Principal component analysis of chemical latent spaces trained on pK_a with $\Delta G_{\pm H}$ enrichment showing latent space reorganization. Each point represents a compound from the full dataset with its encoding vector projected onto the first two principal components of the specified model latent space. Points are colored according to the indicated property. Standardized values are used for $\Delta G_{\pm H}$ coloration. Coefficient of determination and Spearman rank order coefficient are shown in each panel.

Notably, in the $M_{\text{pKa},\Delta G}^{128,512}$ models we observe organization of $\Delta G_{\pm H}$ and pK_a in non-orthogonal directions (dot product of the gradient vectors of 0.9), with $\Delta G_{\pm H}$ organized along both principal components and pK_a organized primarily along the second principal component. This non-orthogonal reorganization supports the hypothesis that the latent space reorganization leads to transfer learning between correlated properties. We also note that this phenomenon decreases as

the size of the dataset increases before nearly disappearing at dataset parity; the dot product of the gradients drops to 0.4 for the $M_{pKa,\Delta G}^{256,512}$ models and we observe nearly orthogonal organization in the $M_{pKa,\Delta G}^{512,512}$ models with a dot product of 0.1. In contrast, we observe that enrichment with E_{sp} data leads to nearly orthogonal organization of E_{sp} and pK_a in the latent space in all cases (Fig. A2), with gradient vector dot products of 0.3, 0.3, and 0.0 for the $M_{pKa,sp}^{128,512}$, $M_{pKa,sp}^{256,512}$, and $M_{pKa,sp}^{512,512}$ models, respectively, leading to little direct transfer learning. The PCA analysis also suggests why the variance of the models decreases for all jointly-trained models. Comparing the jointly-trained latent spaces with the unenriched results (Fig. A1) we note that enrichment generally leads to a small reduction in R^2 and ρ for pK_a , but large improvement in R^2 and ρ for the data rich properties. Joint-training thus results in better overall organization of the high-dimensional latent space, which regularizes the prediction tasks and lowers the prediction variance on unseen compounds.

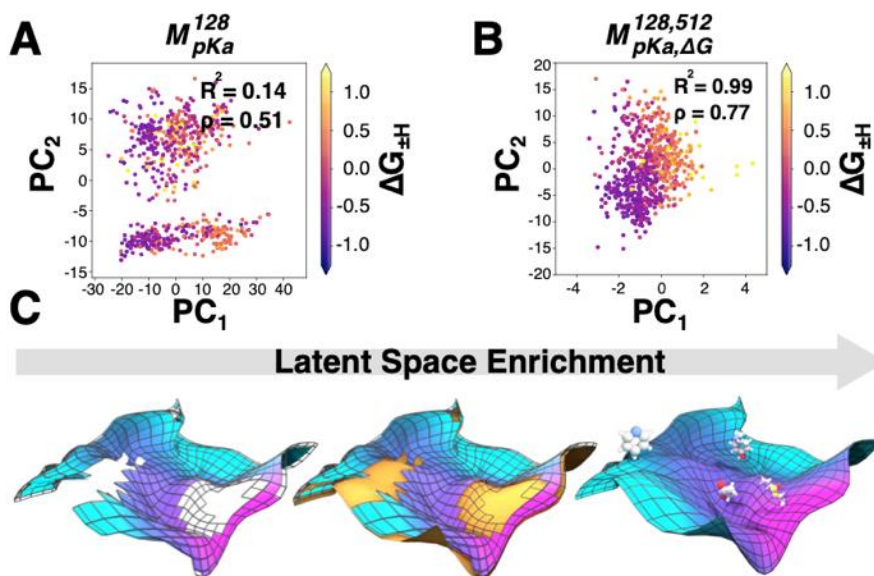


Figure 3.5: Depiction of the latent space enrichment process. PCA analysis of the latent space of the unenriched (A) and enriched (B) models M_{pKa}^{128} and $M_{pKa,\Delta G}^{128,512}$, demonstrating organization with respect to $\Delta G_{\pm H}$ upon enrichment. (C) Illustration of the proposed enrichment mechanism. (Left) Sparse datasets result in regions of latent space with little organization. (Middle) Joint training on correlated computational data results in organization of experimentally uncharacterized regions of latent space. (Right) A more continuous latent space with well-defined property gradients is formed.

A visualization of the effect of enrichment on data-scarce prediction tasks is provided in Figure 3.5. The PCA analysis suggests that there are two transfer learning mechanisms associated with enrichment: first, that correlated properties will partly organize along similar latent space dimensions resulting in improved prediction for data-scarce properties (correlation), and second, that adding chemically relevant data of any kind helps to improve the overall organization of the high-dimensional latent space even for data rich properties (regularization). It is difficult at this point to quantitatively separate the impact of these two mechanisms on the resulting property prediction performance. As observed earlier, the median performance of the models enriched with $\Delta G_{\pm H}$ data shows a marginal improvement over those enriched with E_{sp} data, indicative of the contribution from the correlation effect, but the small magnitude of the effect suggests that regularization is the dominant factor for the current set of enriched models. It is possible that for lower dimensional latent spaces, autoencoders trained on larger datasets, or latent spaces trained on many independent properties, that the relative contributions of the two mechanisms would change.

The enrichment mechanism explored here also suggests further research directions related to how organization of the latent space relates to property prediction and transferability. For instance, it is intuitive that organization of the latent space with respect to chemical properties should lead to better transferability on prediction tasks, but this has not yet been systematically investigated and the presented enrichment results only indirectly address this question. One way this could be studied is to incorporate a combination of R^2 and ρ into the training objective function in order to systematically control the organization of the latent space. Likewise, a useful comparison would be to property prediction networks that are trained on unorganized latent spaces (e.g., by only training on the predictor networks without backpropagation through the encoder). We also note that joint training of the latent space on prediction tasks does not necessarily lead to linear organization with respect to the trained properties. For example, in the case of $M_{pKa,\Delta G}^{512,512}$, although the free energy projections are very well organized ($R^2=0.97$, using a cubic transform of the property data), they are not organized in a manner such that any particular direction is correlated with free energy ($\rho=0.56$). For applications where the latent space is used to identify new compounds, the functional form of the latent space reorganization could be crucial and would need to be addressed through more specific training algorithms.

3.4 Conclusions

A unique aspect of autoencoder models is that joint-training restructures the latent space according to property similarity. This feature has been previously exploited to generate new structures with targeted properties based on their location in a jointly trained latent space. In the multitask training that we have performed here, we have demonstrated that multiple training properties can simultaneously be used to restructure the latent space and provide mutually improved performance in property prediction tasks. A novel aspect of this study is our exploration of the enrichment paradigm where scarce experimental data is supplemented in learning tasks with more abundant computational properties. By organizing the chemical latent space with respect to the abundant dataset, we observe improved performance for scarce property prediction from transfer learning in all cases. These improvements include both higher prediction accuracy and lower variance in models trained to predict the properties of novel species and also increased overall organization of the chemical latent space projections. We anticipate that this approach will be useful in other chemical and material prediction tasks where experimental data is scarce but computational data can be inexpensively obtained. Likewise, there is obvious potential to extend the enrichment procedure reported here to contexts where the latent space is used to iteratively generate additional experimental and computational targets, followed by further refinement of the latent space, and continued target selection. In this manner, latent space enrichment provides a useful framework for basing predictions on all available data until reaching a chemical solution.

4. SIMPLER IS BETTER: HOW LINEAR PREDICTION TASKS IMPROVE TRANSFER LEARNING IN CHEMICAL AUTOENCODERS

Reprinted with permission from J. Phys. Chem. A 2020, 124 (18), 3679-3685. DOI: 10.1021/acs.jpca.0c00042 Copyright 2020 American Chemical Society.

Transfer learning is a subfield of machine learning that leverages proficiency in one or more prediction tasks to improve proficiency in a related task. For chemical property prediction, transfer learning models represent a promising approach for addressing the data scarcity limitations of many properties by utilizing potentially abundant data from one or more adjacent applications. Transfer learning models typically utilize a latent variable that is common to several prediction tasks and provides a mechanism for information exchange between tasks. For chemical applications, it is still largely unknown how correlation between the prediction tasks affects performance, the limitations on the number of tasks that can be simultaneously trained in these models before incurring performance degradation, and if transfer learning positively or negatively affects ancillary model properties. Here we investigate these questions using an autoencoder latent space as a latent variable for transfer learning models for predicting properties from the QM9 dataset that have been supplemented with semi-empirical quantum chemistry calculations. We demonstrate that property prediction can be counter-intuitively improved by utilizing a simpler linear predictor model, which has the effect of forcing the latent space to organize linearly with respect to each property. In data scarce prediction tasks, the transfer learning improvement is dramatic, whereas in data rich prediction tasks, there appears to be little to no adverse impact of transfer learning on prediction performance. The transfer learning approach demonstrated here thus represents a highly advantageous supplement to property prediction models with no downside in implementation.

4.1 Introduction

Machine learning (ML) has made rapid and profound inroads into many areas of chemical science. The prospect of extracting value from dormant data using ML has energized the proliferation of chemical and materials databases.[105][47][129][130][131][132][133] The use of ML in

property prediction is being exploited to cut the cost of physics-based computations and experimental testing. There are also areas like drug and materials discovery, where machine learning is promising to address previously intractable problems like inverse design [10]·[92]·[134] and optimal retrosynthesis.[135]·[136]·[137] Despite the early stage of these efforts, the routine implementation of ML is becoming a reality and there are very few areas where machine learning models are *not* presently being tested against or in conjunction with traditional experimentation, expert systems, and physics-based methodologies. While these promises are exciting, the translation of ML methodologies to applications with intrinsic data scarcity is not straightforward. Datasets for image classification [138], object recognition, [24] and some molecular properties [105]·[139] may contain millions of samples, but more typical chemical applications have access to only a few hundred to a few thousand samples.[140]·[141] For reaction data the situation is more stark. Although dozens of substantial molecular property databases exist, only three significant reaction databases currently exist, and two of them are proprietary.[142]·[143]·[144]

The developing subfield of transfer learning provides some strategies that might alleviate data limitations.[76]·[145]·[99]·[146]·[147] In transfer learning, model proficiency in one learning task is leveraged in one or more related learning tasks.[70]·[148] Transfer learning is potentially very promising in chemical applications since all observable properties are ultimately tied to chemical structures. Thus, the underlying topology of chemical properties is strongly connected, and intuitively should be amenable to transferring knowledge from data rich properties to predict data scarce properties of the same compounds. We recently demonstrated a transfer learning method based on chemical autoencoders for this purpose.[149] In this approach, we utilize a shared latent variable generated from a data-rich pretrained autoencoder as an input for property prediction tasks. We demonstrated that joint training of property prediction models led to organization of compounds in the latent space that, in turn, provided an interpretable mechanism for transfer learning between prediction tasks. Encouraged by this initial demonstration, we have subsequently explored additional strategies for effectively organizing chemical latent spaces with the goal of improving transfer learning efficiency.

In the current study we demonstrate how simple linear prediction models can be used to efficiently organize chemical latent spaces and improve property prediction accuracy in data scarce applications (Fig. 4.1). To date, all chemical autoencoder models that utilize latent space organization have employed complex predictor networks in model training,[63]·[67]·[103]·[68] but

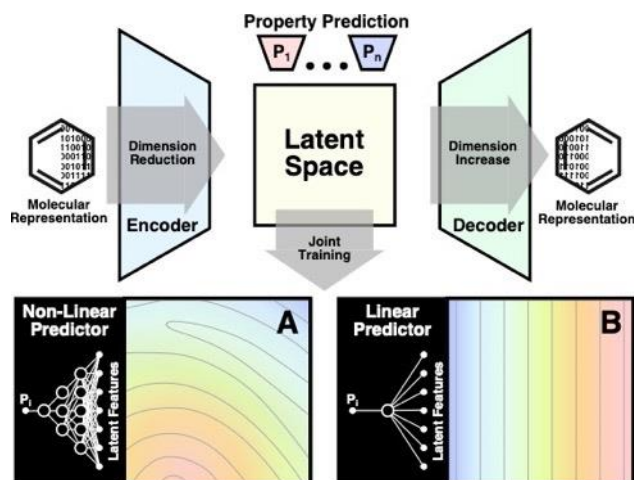


Figure 4.1: Overview of chemical autoencoder (CAE) architectures for transfer learning. The autoencoder generates a compressed vectorial representation of chemical space by employing a low dimensional intermediate layer in the model (i.e., the latent space). This compressed representation is obtained by training the CAE on a reconstruction task using abundant chemical structure data. In transfer learning applications, the latent space serves as a common input feature for two or more property prediction tasks. (A) When utilizing complex predictor networks for joint property training, latent space organization is incomplete due to the predictor’s ability to learn complex surfaces. (B) When a linear predictor network (i.e., a single unit with a linear combination of latent space features) is used for joint training, prediction errors must be minimized by backpropagation through the encoder, resulting in a linear organization restraint on the training objective function.

as demonstrated here, this is actually detrimental to latent space organization. During backpropagation, the prediction error is minimized by jointly adjusting predictor and encoder network weights. In models with complex predictor networks, this only leads to partial organization of the latent space since complex networks can learn complex latent space structures. The incomplete organization of the latent space limits the information exchange between learning tasks and the interpretability of the latent space dimensions, as the relationship between position in the latent space and property values may be unclear. In contrast, utilizing a simple linear model for property prediction tasks forces the encoder to organize the latent space in order to minimize prediction errors. The net result of this is that the objective function has a linear organization restraint with respect to property organization within the latent space.

Using this alternative predictor network, we present a systematic transfer learning study utilizing the QM9 dataset supplemented with semi-empirical calculations.[150] With hybrid DFT level property data for over 100,000 species, this dataset is used to assess the improvement in bandgap prediction from the very data scarce regime up to near saturation, as well as the potential

for multi-property prediction. The DFT level HOMO-LUMO gap ($E_{g,DFT}$) is used as a prediction objective, with varying levels of data scarcity simulated by utilizing varying fractions of the QM9 data for training each model. Comparisons are performed between the prediction accuracy of models trained solely on scarce $E_{g,DFT}$ data and models jointly trained for data rich prediction tasks (i.e., latent space “enrichment”). The use of latent space enrichment via multitask training is found to improve property prediction across all levels of data scarcity, reducing the prediction error for $E_{g,DFT}$ by as much as 0.6 eV and reducing the dependence on dataset size for prediction accuracy by up to two orders of magnitude. The results also demonstrate that information from multiple data rich properties can be incorporated into the models with little to no adverse impact on prediction performance.

4.2 Computational Methods

4.2.1 Datasets

The QM9 dataset, consisting of DFT level (B3LYP/6-31G(2df,p)) calculations for over 134,000 molecules with up to 9 heavy atoms, was utilized for training all models.[150] Based on the dataset specifications, training (80%), validation (10%), and testing (10%) splits were randomly selected. The validation set was utilized for hyperparameter selection, with the testing set withheld until final evaluations. $E_{g,DFT}$, as calculated by DFT within the QM9 database, was used as the prediction target in all models. To investigate the effect of model performance on data scarcity, $E_{g,DFT}$ prediction models were trained on variable amounts of $E_{g,DFT}$ data alone and in combination with one or more data rich properties, including the QM9 reported single point energy (U_0), electronic spatial extent (R^2), heat capacity (C_v), and zero-point vibrational energy (ZPVE). Additionally, the geometries provided in the QM9 dataset were used as initial guesses for a geometry optimization and single point evaluation at the GFN2-xTB (xTB) semi-empirical level.[151] The HOMO-LUMO gap obtained at the semi-empirical level ($E_{g,xTB}$), was used as a correlated property for multi-property prediction (Fig. B1).

4.2.2 Machine Learning Architecture

We have evaluated the performance of 90 models on molecular property prediction tasks. All models share a common autoencoder architecture; individual models are distinguished by their

associated training data and by one or more linear prediction units connected to the autoencoder latent space. The autoencoder architecture was adapted from the grammar variational autoencoder previously developed by Kusner et al., [68] which utilizes a one-hot representation of allowable grammar rules to mitigate latent space voids that result from the fragility of the SMILES representation.[67] The autoencoder accepts one-hot inputs of grammar parse trees to an encoder network comprising three one-dimensional convolutional layers with filter sizes 9, 9, and 10, respectively, and kernel sizes of 9, 9, and 11, respectively. The outputs from the convolutional layers are passed to a fully connected layer of 435 units, that are then separately connected to two fully connected layers of 56 units (i.e., the dimensionality of the latent space) defining the mean and log variance of the encoding distribution, respectively. The decoder accepts samples from the encoding distributions and passes them to a fully connected layer of 56 units that is connected to three gated recurrent units of 501 cells each before terminating in a final fully connected layer outputting probability distributions for the output sequence. ReLU activation functions were used for all units in the autoencoder. A diagram of the autoencoder architecture is provided in Figure B2. Models were created using Keras [127] 2.2.4 with Tensorflow [128] 1.14.0 backend.

After pre-training of the autoencoder on a reconstruction task, the 56-dimensional latent space represents a compressed vectorial representation of chemistry that is utilized as an input feature in subsequent property prediction models. The architecture for each property prediction model thus consists of the pretrained autoencoder connected to one linear unit at the latent space layer for each property prediction task. Preliminary testing of prediction models with other simple forms, including the L2 norm of the latent space vector or a single component of the latent vector, confirmed that a single fully connected unit provided optimal latent space organization with respect to multi-task prediction. A more complex predictor network, consisting of a network of 3 fully connected layers of 64 units trained with dropout rate of 0.15, was also considered to provide a comparison to the linear unit. This network also utilizes the ReLU activation function between layers and terminates in a single linear unit.

Since the autoencoder training data and representation do not distinguish between conformers of the same compound, the latent vectors likewise do not distinguish between conformers. When using this architecture for property prediction, the predictions thus reflect the conformational sampling represented in the training data. In the case of QM9, this implies predictions for locally minimized DFT geometries. To predict the conformational dependence of

these properties requires an autoencoder representation that distinguishes between molecular geometries, which is beyond the scope of the present work.

4.2.3 Model Training

To obtain a useful compressed representation of chemistry, the shared autoencoder was first pre-trained on a reconstruction task. For this purpose, the SMILES strings corresponding to all training compounds were first converted into one-hot grammar parse trees and used as both inputs and labeled outputs for autoencoder pretraining. Pretraining was performed using the Adam 7[27] algorithm with learning rate of 0.005 on the categorical cross entropy loss function for 100 epochs with batch size of 500. Validation loss was monitored every epoch and the learning rate was halved in the event of a plateau.

The initial guess for individual property prediction models consisted of the pretrained autoencoder fully connected to a randomly initialized single unit at the latent space layer for each property prediction task. The model is then retrained with both the reconstruction task and the property prediction tasks, with an additional mean squared error loss term for each included property. ‘Missing’ property data was handled by setting the loss weight of the associated sample to zero. In all cases, the validation set was held fixed at the full ~13k entries.

4.2.4 Training Data

Each model reported here is distinguished by the amount of data and the number of property prediction tasks utilized during training. To investigate the effect of data scarcity on prediction accuracy, varying fractions of the $E_{g,DFT}$ data were utilized for training each model. These fractions were sampled logarithmically with ten points per decade from 0.0001 to 0.8 (corresponding to ~10 to ~100,000 compounds, respectively) and $E_{g,DFT}$ -only models (\mathbf{M}_{DFT}) were trained on each of the thirty-eight fractions as a reference for multi-task transfer learning models. To provide a comparison between predictor networks of varying complexity, two additional single task models were trained using only $E_{g,xTB}$ data at a fraction of 0.8 with either a linear predictor or the multi-layer fully connected predictor described in the model architecture section.

To evaluate the effect of data scarcity with respect to the transfer learning property prediction task, models were trained with variable fractions of $E_{g,xTB}$ and $E_{g,DFT}$ data. In these

models, fractions of 0.0001, 0.001, 0.01, 0.1, and 0.8 of the total $E_{g,xTB}$ and $E_{g,DFT}$ data were included for joint property prediction (i.e., separate models were trained for all combinations of the fractions $E_{g,xTB} = 0.0001, 0.001, 0.01, 0.1, \text{ and } 0.8$ with $E_{g,DFT} = 0.0001, 0.001, 0.01, 0.1, \text{ and } 0.8$).

To evaluate the effect of transfer learning with respect to multiple enrichment properties, multi-task models were trained with variable fractions of the $E_{g,DFT}$ data, and each of the following five sets of enrichment properties: (1) $E_{g,xTB}$, (2) $E_{g,xTB}$ and U_0 , (3) $E_{g,xTB}$, U_0 , and R^2 , (4) $E_{g,xTB}$, U_0 , R^2 , and ZPVE, and (5) $E_{g,xTB}$, U_0 , R^2 , ZPVE, and C_v . Each multi-task model was trained with respect to all QM9 training data for the enrichment properties (i.e., the full 0.8 training split), in combination with variable $E_{g,DFT}$ fractions of 0.0001, 0.001, 0.01, 0.1, and 0.8.

The smaller spacing of $E_{g,DFT}$ fractions in the transfer learning models compared with the reference \mathbf{M}_{DFT} models was necessary to yield a tractable number of total models while still evaluating performance across orders of magnitude differences in available data. The reported results thus consist of 88 models for $E_{g,DFT}$ prediction and 2 models for $E_{g,xTB}$ prediction.

For the data scarce models, randomly selecting training sets can lead to poor representation of the testing data and correspondingly large variations in the prediction errors that would confound the evaluation of transfer learning efficiency. We have minimized this source of error by training ten independent models for each fraction size, with training sets shuffled between each iteration, and the best performing model on the validation data was selected for use in evaluation on the testing set. Thus, the model performance represents transfer learning efficiency in the limit that training sets are well optimized with respect to testing use cases.

4.3 Results and Discussion

A comparison of the $E_{g,DFT}$ prediction results for \mathbf{M}_{DFT} and models jointly trained with varying fractions of $E_{g,xTB}$ data is shown in Figure 4.2. The results demonstrate that by enriching the models with correlated $E_{g,xTB}$ data, the $E_{g,DFT}$ prediction accuracy increases in situations with

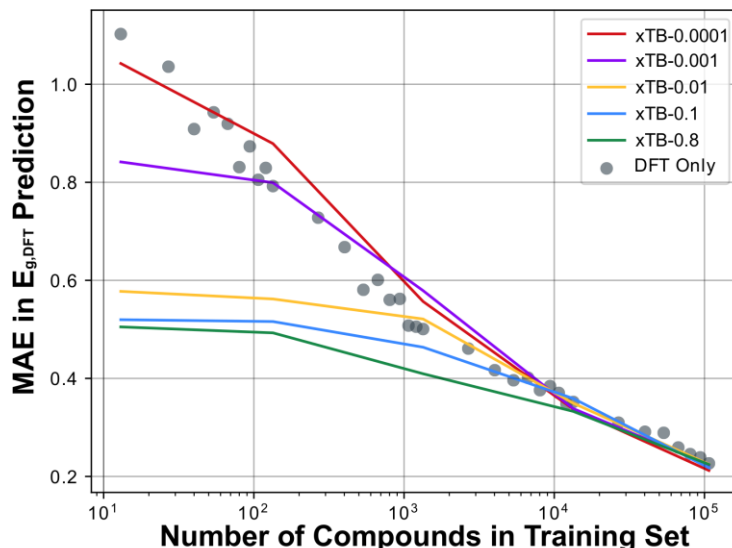


Figure 4.2: Mean Absolute Error (MAE) in $E_{g,DFT}$ prediction using models trained on varying fractions of DFT data (gray) and models jointly trained on $E_{g,DFT}$ and $E_{g,xTB}$ data (lines). In situations where less than ten percent of the DFT data is available for training, the identity of the compounds within the dataset becomes an important consideration. For all of these models we train 10 models, each on a different random subset of the training data, and select the one exhibiting the best performance on the validation set for final evaluation on the test set. The 95% confidence interval about the mean values are within marker size.

limited DFT data, decreasing by as much as half an electron volt in extremely data scarce applications. Several quantitative aspects of this comparison provide insight into the efficiency of transfer learning and the conditions under which it applies.

First, we observe that there is a clear power law relationship ($R^2=0.99$) between the amount of DFT training data and the mean absolute prediction error (MAE) for \mathbf{M}_{DFT} . Above the smallest DFT fraction of 0.0001, models outperform the baseline mean prediction error of 1.07 eV. Models trained with the maximum fraction of $E_{g,DFT}$ data exhibit a MAE of 0.21 eV, equal to the performance observed in other studies utilizing autoencoders in QM9 bandgap prediction.[67]

Second, we note that the enrichment effect saturates after supplying a threshold fraction of 0.01 for the $E_{g,xTB}$ enrichment data (yellow curve in Fig. 4.2). This illustrates that a relatively small fraction of the data is sufficient for these models to learn the difference between the enrichment property, $E_{g,xTB}$, and the data scarce property, $E_{g,DFT}$. At this point, the degree of latent space organization begins to converge (*vide infra*), and the limited DFT data available to train the predictor node serves as a hard limit to the achievable accuracy. Models enriched with fractions of

the $E_{g,xTB}$ enrichment data greater than 0.01 show a transfer learning enhancement further into the data rich $E_{g,DFT}$ regime, although the improvement is smaller.

Third, we note that when the supplied DFT data exceeds the enrichment data, the enriched models perform similarly to \mathbf{M}_{DFT} . This demonstrates that the enrichment prediction task does not inhibit learning of the $E_{g,DFT}$ structure-function relationship, as long as the latent space has sufficient dimensionality to accommodate the organization of multiple properties. In general, the enriched models show MAE improvement compared to \mathbf{M}_{DFT} up to the point of data parity, after which the enriched model error curve collapses down to the DFT only cases.

It is significant that the joint-training leads to improved predictions for the test set of compounds, and not merely the training compounds for which enrichment data was provided. This is evidence that the latent space training organizes regions of chemical space, rather than just specific compounds. To elucidate the relationship between latent space organization and transfer learning efficiency, we also trained a model on $E_{g,xTB}$ data alone using a more complex predictor and performed principal component analysis (PCA) on the latent space encodings of the training compounds.

The resulting principal component plots are presented in Figure 4.3 with a comparison between the organization of $E_{g,xTB}$ and $E_{g,DFT}$. Both models exhibit similar levels of prediction accuracy on the withheld test set of $E_{g,xTB}$, however the latent space organization is inversely related to the predictor model complexity. We have intentionally selected an auxiliary model that produces complex organization to demonstrate that sufficiently complex predictors are capable of learning nonlinear latent structures. While complex predictor networks can result in linear organization within the latent space, only the simplest linear predictor model ensures a continuous gradient and results in an interpretable latent space dimension corresponding to $E_{g,xTB}$. The organization of the latent space with respect to $E_{g,xTB}$ in the linear model also results in the organization of $E_{g,DFT}$, which was not supplied as training data.

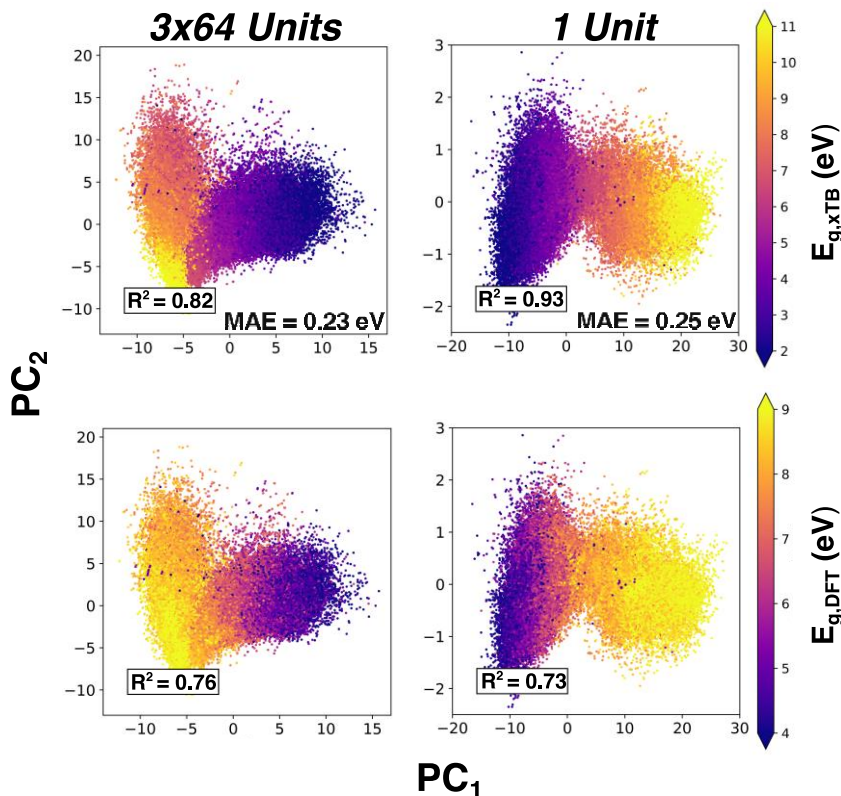


Figure 4.3: Principal component analysis (PCA) performed on latent encodings of the entire QM9 dataset. Both models were trained solely on $E_{g,xTB}$ training set data and are distinguished by the complexity of the predictor network. PCA results, MAE, and linear coefficient of determination (R^2) are presented for (left) a predictor network comprised of three fully connected layers of 64 nodes terminating in a single linear unit; and (right) a predictor network comprised of a single linear unit. Both networks achieve comparable performance with respect to MAE on the $E_{g,xTB}$ prediction task (differing by less than 1% of the range of $E_{g,xTB}$ training data) while exhibiting qualitatively different latent space organization. Only the simple linear predictor results in an interpretable latent space dimension (PC_1), corresponding to the HOMO-LUMO gap.

The original motivation for latent space organization through joint training on a property prediction task was not for transfer learning, but to facilitate molecular discovery.[67] However, the PCA results in Figure 4.3 clarify that property prediction accuracy is not necessarily connected with latent space organization. A sufficiently complex predictor network can achieve accurate predictions despite a highly nonlinear relationship between the position of the compound within the latent space and the property of interest. A complex predictor is thus unsuitable for transfer learning as the latent space organization will not necessarily reflect the intrinsic correlations between data scarce prediction tasks. As non-linear organization of the latent space can make the relationship between position in the latent space and exhibited property nonintuitive, previous

work using autoencoders in generative applications have utilized importance sampling algorithms (e.g., Gaussian processes and genetic algorithms) when searching for promising compounds.[67][92][152] Organization via a simple linear predictor provides an alternative, human interpretable approach and a framework for continuously incorporating new data to refine the latent space organization.

4.3.1 Multi-Property Prediction

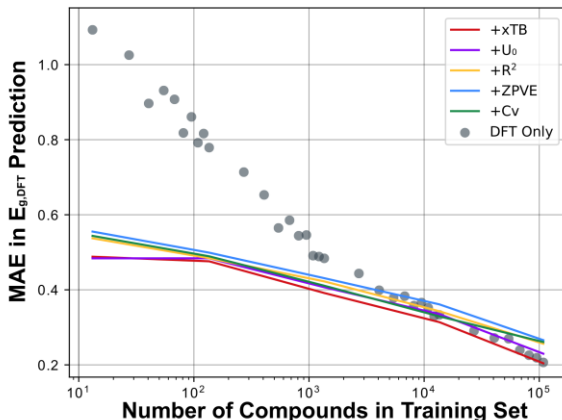


Figure 4.4: Mean Absolute Error (MAE) in DFT bandgap prediction using varying fractions of DFT data as well as up to 5 additional properties. While the fraction of available DFT data is allowed to vary, the auxiliary properties are included at the full training fraction. The inclusion of multiple property prediction tasks during training has a negligible impact on prediction accuracy while providing additional organization of the latent space. The 95% confidence interval about the mean values are within marker size.

Beyond the use of correlated properties for latent space enrichment, we have also explored models trained with multiple enrichment properties. Including data from multiple sources and for multiple properties would be useful in practical data scarce scenarios or in generative applications where multiple orthogonal properties are being optimized. Likewise, including multiple independently correlated properties could induce additional enrichment improvements for data scarce property prediction. To investigate these effects, multi-task models were trained with variable fractions of the $E_{g,DFT}$ data, and each of the following five sets of enrichment properties: (1) $E_{g,xTB}$, (2) $E_{g,xTB}$ and U_0 , (3) $E_{g,xTB}$, U_0 , and R^2 , (4) $E_{g,xTB}$, U_0 , R^2 , and ZPVE, and (5) $E_{g,xTB}$, U_0 , R^2 , ZPVE, and C_v . Figure 4.4 compares the prediction accuracy of the multi-task models with \mathbf{M}_{DFT} .

Up to an $E_{g,DFT}$ data fraction of 0.1, all of the enriched models exhibit improved $E_{g,DFT}$ accuracy compared with \mathbf{M}_{DFT} results. Moreover, the inclusion of additional properties within the data scarce regime results in a difference in MAE of less than 10% across the multi-property models. At higher $E_{g,DFT}$ data fractions there appears to be a consistent increase in error for models trained with two or more enrichment properties due to the compromise in organizing the latent space against multiple properties. For data rich applications on multiple orthogonal properties, there may be a tradeoff between the simplicity of the latent variable model and predictive performance that necessitates the use of more complex predictors. However, in the transfer learning regime this effect is minute in comparison with the overall consistency of the accuracy curves across all of the enriched models. Thus, multi-property training has little to no adverse effect on data scarce property prediction, but it does potentially provide additional latent space organization that could be beneficial in generative applications.

The effect of multi-property enrichment on latent space organization was investigated by performing PCA on the latent space of a model enriched with both $E_{g,DFT}$ and R^2 data (Fig. 4.5).

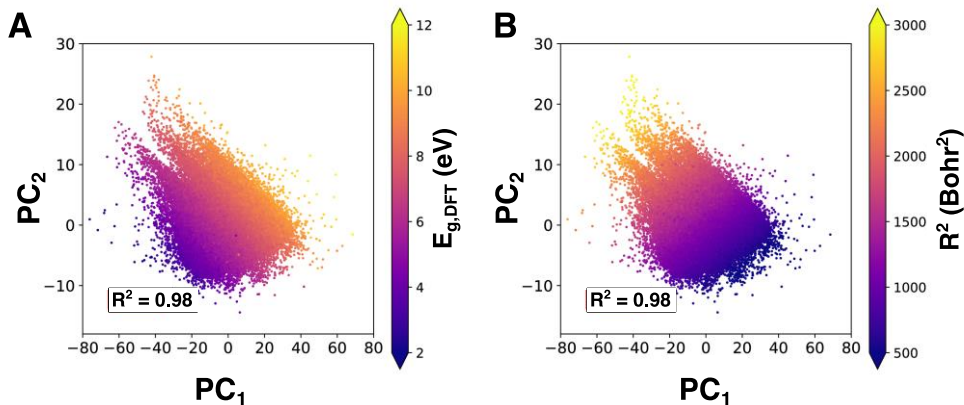


Figure 4.5: Principal component analysis performed on latent encodings for compounds within the training set for a model trained on both DFT data and electronic spatial extent. The projections along the first two principal components are colored according to (A) $E_{g,DFT}$ and (B) R^2 . The chemical landscape shows excellent organization with respect to both properties ($R^2=0.98$ for both). Latent space organization occurs in orthogonal directions for uncorrelated properties.

Figures B3 and B4 provide a similar analysis for $E_{g,xTB}$ and C_v with linear and non-linear predictors, respectively. Since R^2 is simply a measure of the spatial extent of each molecule, it exhibits little correlation with $E_{g,DFT}$. After joint training, we observe a distinct advantage for linear predictors

over complex predictors in multi-property organization. Figure B4 demonstrates that more complex predictor networks may have difficulty operating on disjoint properties, leading to latent spaces with poor, non-linear organization. By instead utilizing a linear predictor, the encoder manages to organize compounds linearly with respect to both molecular properties. However, the lack of correlation between the properties results in R^2 and $E_{g,DFT}$ organization along orthogonal principle components. In contrast, we observed that for correlated properties, compounds are organized along colinear directions within the latent space (Fig. 4.3). Thus, the inclusion of non-correlated properties in the enriched models exhibits little adverse impact on prediction accuracy in the data scarce regime provided the latent space is of sufficient dimensionality to organize with respect to all properties. Although adding additional uncorrelated properties exhibits no advantage with respect to prediction accuracy, it does provide additional interpretability to the latent space as particular directions will now have direct, linear correlation with their respective properties. Likewise, adding additional physical information in the latent space organization may have beneficial effects in generative applications that will be explored in future work.

4.4 Conclusions

Data scarcity remains an impediment to applying ML to many problems in molecular property prediction. We have demonstrated that chemical autoencoders provide a powerful framework for combining data from multiple sources to improve the prediction of data scarce molecular properties. These results provide a systematic demonstration that enrichment data can come from multiple sources, be comprised of multiple properties, and vary in quantity while still enhancing performance on data scarce property prediction tasks. Moreover, the transfer learning models converge to the prediction accuracy of analogous ML models in the data rich regime. Thus, the presented transfer learning framework represents a flexible strategy for circumventing data limitations with little observable drawback.

We have also demonstrated that the use of simpler linear predictor models for latent space organization positively impacts both transfer learning efficiency and the interpretability of the latent space by directly relating position along the various latent axes with chemical properties. Multi-property training results in latent space organization such that correlated properties are collinearly arranged and non-correlated properties occupy orthogonal dimensions. Effective organization is achievable utilizing only a few hundred training samples in the case of strong

correlation between data scarce and data rich prediction properties. Similarly, transfer learning improvements are observable with as little as an order of magnitude excess enrichment data and are not significantly affected by the inclusion of additional uncorrelated properties. Although the inclusion of multiple uncorrelated properties does not assist the targeted molecular prediction tasks presented here, it provides an avenue for multi-objective optimization and chemical discovery that will be explored in future work.

5. IMPROVING THE GENERATIVE PERFORMANCE OF CHEMICAL AUTOENCODERS THROUGH TRANSFER LEARNING

Reprinted with permission from Mach. Learn.: Sci. Technol. 2020, 1 (4), 045010. DOI: 10.1088/2632-2153/abae75 Copyright 2020 IOP Publishing.

Generative models are a sub-class of machine learning models that are capable of generating new samples with a target set of properties. In chemical and materials applications, these new samples might be drug targets, novel semiconductors, or catalysts constrained to exhibit an application-specific set of properties. Given their potential to yield high-value targets from otherwise intractable design spaces, generative models are currently under intense study with respect to how predictions can be improved through changes in model architecture and data representation. Here we explore the potential of multi-task transfer learning as a complementary approach to improving the validity and property specificity of molecules generated by such models. We have compared baseline generative models trained on a single property prediction task against models trained on additional ancillary prediction tasks and observe a generic positive impact on the validity and specificity of the multi-task models. In particular, we observe that the validity of generated structures is strongly affected by whether or not the models have chemical property data, as opposed to only syntactic structural data, supplied during learning. We demonstrate this effect in both interpolative and extrapolative scenarios (i.e., where the generative targets are poorly represented in training data) for models trained to generate high energy structures and models trained to generate structures with targeted bandgaps within certain ranges. In both instances, the inclusion of additional chemical property data improves the ability of models to generate valid, unique structures with increased property specificity. This approach requires only minor alterations to existing generative models, in many cases leveraging prediction frameworks already native to these models. Additionally, the transfer learning strategy is complementary to ongoing efforts to improve model architectures and data representation and can foreseeably be stacked on top of these developments.

5.1 Introduction

The proliferation of machine learning (ML) research in the context of the chemical sciences has yielded powerful modeling paradigms for increasing the accuracy and reducing the cost of predicting the properties of molecules and materials. As this work reaches maturity, the so-called “forward-problem” of predicting function from a given chemical structure is becoming more routine and with new datasets coming online more properties will soon become accessible to prediction using ML models. However, the “inverse-problem” of finding an optimal set of structures under functional constraints is more directly relevant to chemical design and remains unsolved. Deep generative models are one class of ML modeling that is potentially capable of addressing the inverse problem by yielding exemplary structures from the same population as the training distribution. Such models have been effective in generating images that accurately match a desired caption, [153] novel musical scores, [154] and have recently been under intense study in chemical applications to discover new functional molecules and materials.[10] Common generative methods include variational autoencoders, [92][67][155] recurrent neural networks, [156][157][158] generative adversarial networks, [159][160] and adversarial autoencoders, [161] among others, and the optimal model architecture and data representation for chemical generation is still an area of active research. These models have been extensively applied towards the design of drug-like molecules, but other applications also include the generation of novel solar cell materials [103] and crystalline species.[162]

Several of the early papers on generative chemical models have noted low generative validity [67] and diversity [163] when using typical chemical representations (e.g., using SMILES, InChI, and grammar based representations) and architectures. Thus, intense research has been devoted to developing model architectures that guarantee or improve the chemical validity and uniqueness of the molecules produced by deep generative models.[155][164][165][166] In contrast, the role of training data and the possible impact of including additional chemical property information during training remains largely unexplored. In particular, although generative models are typically trained using abundant syntactic data (i.e., molecules with valid Lewis structures from databases like QM9 and ZINC) limited chemical property data is typically incorporated into model training. Likewise, research on which properties are conducive to inverse design has been scarce, and the properties that have been investigated tend to be either narrow in scope, such as simple cheminformatics data like molecular weight and the water-octanol partition coefficient, or

properties that may not be directly verified and are instead based on similarity to known lead molecules.[156] A general approach for targeted generation of compounds based on general quantum chemical properties has not yet been developed.

In the present work we investigate whether transfer learning [148][70][167][146][168][169] (TL) improves the performance of generative models in comparison with baseline models trained to generate molecules with a single property. We focus on multi-task TL, [76][170][79][149] whereby modifications to model architecture are minimal, save only through the inclusion of additional property prediction tasks. Since multi-task TL is largely model independent, it provides a potentially complementary strategy for improving generative models to contemporary efforts to modify model architecture and data representation. The hypothesis driving our exploration of multi-task TL for generative models is that the limited validity and uniqueness exhibited by many generative models is a symptom of insufficient chemical property data being utilized during training. In particular, elementary chemical properties including internal energy, bandgap, and zero-point vibrational energy can be routinely calculated from quantum chemistry and may provide relevant information about validity and molecular stability that cannot be learned from molecular structures alone. This hypothesis is explored using a generative variational autoencoder model with multi-task TL trained on the QM9 dataset, along with semi-empirical calculations for validation of generative predictions. We consider internal energy at zero Kelvins (U_0), zero-point vibrational energy (ZPVE), and HOMO-LUMO gap (E_g) evaluated at the DFT level as multi-task prediction properties for training, and compare how generative performance with respect to validity, uniqueness, and property specificity is affected by inclusion of additional ancillary chemical property data. Since QM9 is a database of small molecules, relatively few high $|U_0|$ samples or low E_g samples exist in the training data. Thus, by characterizing generative results in these property spaces we can characterize how generative performance in an extrapolative task is affected by transfer learning. For targeted structure generation, we observe that multi-task TL increases the percentage of valid and unique high $|U_0|$ structures up to sevenfold with increased property specificity. Searching within areas of physically accessible property values tends to greatly increase the proportion of valid chemical species generated. Even within property regimes with limited representation in the training data, such as structures within the optical bandgap of 1.5-2.0 eV, the inclusion of additional property data improves the ability of the generative models to discover new valid structures. Thus, multi-task

TL can be utilized in both extrapolative and interpolative applications to generate novel compounds with optimized or targeted molecular properties with a higher success rate.

5.2 Computational Methods

5.2.1 Datasets

All models were trained and evaluated using structures from the QM9 dataset. This dataset contains molecular properties computed at the B3LYP/6-31G(2df,p) level for small molecules ranging from 1-9 heavy atoms (C, O, N, F). 80% of the structures and their associated properties were utilized for model training, 10% were withheld as part of a validation set to gauge model performance during training, and the final 10% were kept as an independent testing set for the property prediction tasks. The internal energy at 0K (U_0), zero-point vibrational energy (ZPVE), and HOMO-LUMO gap (E_g) were used individually, or in combination, for training multi-task models.

5.2.2 Machine Learning Architecture

Approximately 100 models were evaluated in total, with models distinguished by the number and kind of prediction tasks utilized during training. Unless otherwise noted, model types were evaluated in an ensemble fashion, with 10 models trained per ensemble. Sampling results for each model type are averaged across the ensemble to provide uncertainty estimates. The autoencoder model utilized in this study was based on the grammar variational autoencoder (GVAE) developed by Kusner et al.[68] The major alterations from the original implementation lies in the use of linear predictor networks during model training and evaluation, which have been previously shown to be advantageous for multi-task training.[170] Models were trained using the RMSprop algorithm with a learning rate of 0.001 for 100 epochs. The loss corresponding to predictor MSE was scaled by 100, KL divergence was scaled by 750, and the categorical cross entropy loss weight from encoding/decoding was scaled by 50 initially, before decaying to 1 according to a sigmoid function. These loss weights were selected to ensure well-balanced performance with respect to both reconstruction on the training data and property prediction accuracy. Exemplary trained models, model schematic, and full training details may be found in Appendix C.

Over 6 million compounds were cumulatively generated in the present study. Since it is infeasible to characterize all of these compounds at the B3LYP/6-31G(2df,p) level, GFN2-xTB (xTB) was utilized to evaluate U_0 and E_g of newly generated species.[151] DFT level predictions for E_g were estimated from a random forest model trained to predict the difference between xTB and B3LYP/6-31G(2df,p) (Fig. C1). The random forest model consists of 120 regressors expanded to max depth and trained to predict the difference in xTB and DFT computed bandgap given a Morgan fingerprint of depth two with 1024 bits. Fingerprints were calculated using the RDKit implementation, and the random forest model followed the Scikit-Learn implementation. xTB predictions for $|U_0|$ were found to be linear correlates for the DFT values (Fig. C2) and were used without modification.

5.2.3 Sampling Paradigms

In the variational chemical autoencoder studied here, new compounds are generated by sampling points from the latent chemical distribution and passing these points to the decoder. By including an ancillary property prediction task, molecular properties vary linearly along particular directions within the latent space (i.e., a principal component). The regions of the latent space to target can thus be determined by linear regression of the target property of training compounds with respect to their position along the principal components. The generative performance of the models was investigated using both extrapolative and interpolative sampling approaches. Extrapolative property sampling was achieved using a one-sided distribution with mean equal to the extreme latent position along the sampled principal component in the training data and variance determined from the variance of the training data along the sampled principal component. All other dimensions were normally sampled according to the training set distribution.

In interpolative sampling, the goal is to recover compounds whose properties lie within a given range (with at least some representation within the training data). The maximum and minimum position along the latent dimension corresponding to the high and low end of the targeted property range were determined via linear regression. This dimension was then sampled from a uniform distribution with specified high and low values. All other dimensions were sampled normally according to the training set distribution.

5.3 Results and Discussion

5.3.1 Extrapolative Sampling

To explore whether adding property prediction tasks can improve generative models we investigated two simplified models: one trained only on encoding and decoding, and another also trained to predict U_0 from the latent encoding. The latent dimensionality was set at two to allow comprehensive sampling to evaluate the validity of generated compounds and for direct visualization of the latent space. 1000 points were sampled from this 2D latent space using the quasi-random Sobol algorithm to maximize the area of the latent space sampled. Due to the stochastic nature of the decoding process, each point was decoded 500 times to determine the average sampling validity. These models are not optimized for either predictive or generative

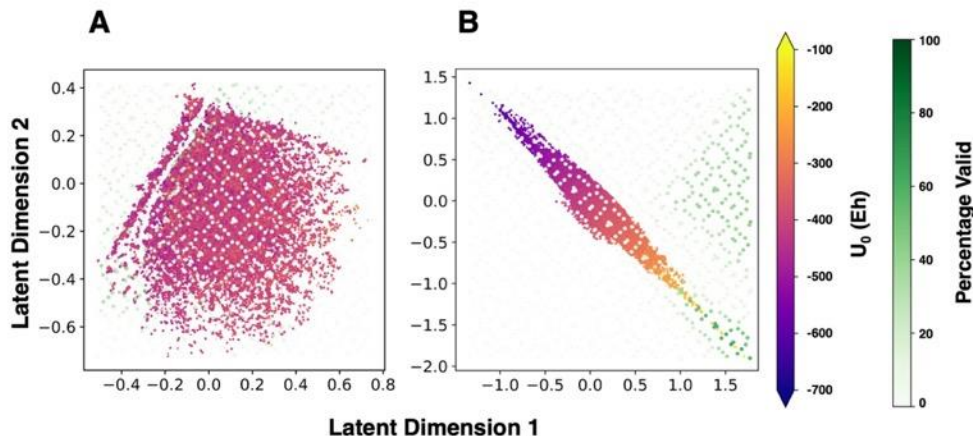


Figure 5.1: Comparison of latent space Sobol sampling for 2D autoencoders trained on (a) chemical reconstruction only and those with (b) an ancillary U_0 prediction. All compounds utilized in training the models have been projected into the latent space and are colored according to U_0 . The points corresponding to the Sobol sampling are colored according to their average validity across 500 decodings. For the model trained to predict U_0 , we observe a clear relationship between increasing validity and decreasing $|U_0|$, whereas for the model trained only on reconstruction, there are no observable trends with respect to either validity or property values.

performance, but have been aggressively simplified to illustrate the basic transfer learning mechanism explored in this work. The results from these experiments are presented in Figure 5.1.

This example illustrates two salient features of adding a property prediction task to the generative model. First, by learning to predict a chemical property the models also learn to

discriminate between valid and invalid structures. For the autoencoder trained without property prediction tasks (Fig. 5.1a), the latent space is organized syntactically, with a region containing almost exclusively aromatic compounds in the top left, separated by a gap from other non-aromatic ring-containing structures (Fig. C3). However, we observe no discernable trend with respect to decoding validity or $|U_0|$ across the latent dimensions. In contrast, the model trained to predict $|U_0|$ (Fig. 5.1b) exhibits sampling validity trends that follow the physical property. Second, adding a property prediction task exposes both training data limitations and features of chemistry that affect generative behavior. The sampling validity of the model trained to predict $|U_0|$ approaches nearly 100% for small molecules (i.e., low $|U_0|$) and lower validity for large structures (i.e., high $|U_0|$). This is consistent with the observation that small molecules are easier to validly decode and are well represented in the training data. In contrast, high $|U_0|$ structures are relatively rare in QM9 given that the dataset is curated for small molecules.

Motivated by this example, we then explored whether generative validity also follows chemical property trends in higher dimensional autoencoders that are more typical of generative applications.[67][161][81][92] In particular, higher dimensional latent spaces are required to effectively compress chemical data when multiple properties are utilized during training (Figs. C4-5). We compared the generative performance of two 56-dimensional autoencoders, one trained only on encoding and decoding and the other also trained to predict U_0 from the latent encoding. Both models were sampled in an extrapolative mode along their first principal component as determined by the latent encodings of the training data, with 30,000 unique structures generated for each model type. For the model trained without property prediction, the extrapolative sampling validity is found to be 25% +/- 7% with uniqueness of 23% +/- 5%, regardless of direction along the first principal component and consistent with no organization of the latent space with respect to validity. Conversely, sampling in the low absolute internal energy regime of the jointly trained model nearly doubles the average validity, raising it to 45% +/- 3%, but drastically reduces the uniqueness to 3% +/- 2%, as there are only a small number of C, N, H, O, and F containing molecules as U_0 approaches zero. Conversely, extrapolating to high $|U_0|$ produces a valid structure only 3% +/- 2% of trials with similar values for uniqueness. These results establish that validity trends in high-dimensional autoencoders follow chemical property values, consistent with the experiments on the low-dimensional models.

It is clear that the inclusion of chemical property data during generative model training affects sampling validity and suggests that supplying additional chemical information may further

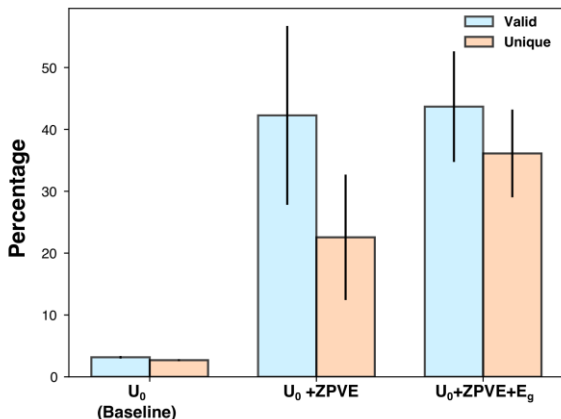


Figure 5.2: Average sampling validity and uniqueness for the high $|U_0|$ extrapolation trial. Results are averaged across 10 distinct models for each training paradigm, with error bars denoting standard deviation. Baseline models exhibit very limited ability to generate structures with high $|U_0|$. The addition of ancillary property prediction tasks greatly improves the generative utility of these models.

improve generative performance. We therefore examined the effect of including additional property prediction tasks based on available QM9 data, including $ZPVE$, which is expected to scale with the number of bonds in the molecule, and E_g , which is not anticipated to correlate with the internal energy of a compound, and repeated the U_0 extrapolation outlined above. The validity and uniqueness statistics for the baseline model trained only on U_0 , and models trained on $U_0/ZPVE$, and $U_0/ZPVE/E_g$ are summarized in Figure 5.2. The addition of ancillary $ZPVE$ and E_g prediction tasks improves the ability of the multitask models to generate high absolute internal energy structures. Inclusion of $ZPVE$ alone leads to an increase in the average sampling validity and fraction of unique structures. Including an additional E_g prediction task results in little further change in sampling validity, but the proportion of unique structures generated increases and the variance of both quantities decrease.

To investigate if the increased validity and uniqueness of the multi-task models also reflected increased property specificity, we have histogrammed the predicted $|U_0|$ of the generated compounds for each model in Figure 5.3. We note a root mean square deviation of ~ 11 Hartrees

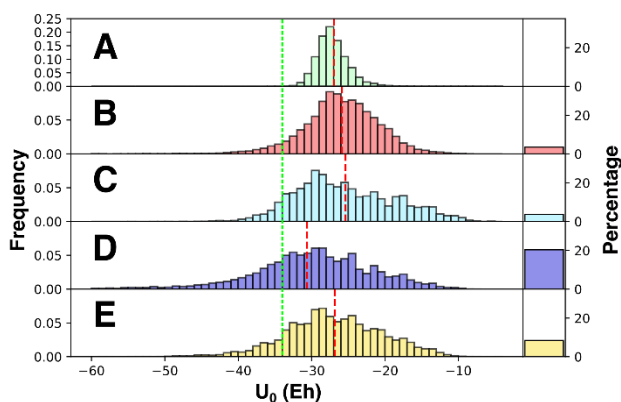


Figure 5.3: Distribution of U_0 for structures for the training set (a) compared with structures generated from extrapolation along the first principal component of models trained on (b) encoding and decoding alone, (c) an ancillary U_0 prediction task, (d) U_0 and $ZPVE$ prediction tasks, and (e) U_0 , $ZPVE$, and E_g prediction tasks. For each paradigm, 3000 unique structures are generated across the 10 duplicate models for a total of 30,000 structures. The mean of each distribution is denoted with a dashed red line and the largest $|U_0|$ value in the training data is indicated by the dashed green line. The percentage of generated compounds with $|U_0|$ greater than observed in the training data is shown on the right.

between $|U_0|$ calculated at the DFT and xTB level (Fig. C2), thus we expect this to be the minimum uncertainty in the resulting estimates and also contributes to width of the resulting distributions. Extrapolative sampling of the base autoencoder (Fig. 5.3b) exhibits a broader distribution of $|U_0|$ values than observed in the training distribution, however the mean $|U_0|$ is consistent with the training data and <5% of the structures exceed the maximum $|U_0|$ of the training data (Fig. 5.3a). Fig. 5.3b shows that including the U_0 prediction task leads to an increase in the number of high $|U_0|$ structures and shifts the mean relative to the training data; however, the mean of the predicted distribution is still within the training distribution, indicating a limited ability to extrapolate beyond the training data. Figure 5.3d shows that inclusion of the $ZPVE$ prediction task shifts the distribution towards higher $|U_0|$ structures well beyond the training data, with ~20% of the generated structures displaying $|U_0|$ greater than the maximum value in the training set. Thus, the increase in observed valid/unique structures comes from the higher density of desired structures within the sampling region. Interestingly, while the further addition of a E_g prediction task in Figure 5.4e increases the validity and uniqueness of the generated structures, the bias towards higher $|U_0|$ structures has been removed. It appears that inclusion of E_g saturates the latent space with valid structures, albeit biased towards the training population. The disparate impact on $|U_0|$

extrapolation of including the *ZPVE* and E_g prediction tasks can be understood in terms of their correlation with $|U_0|$. *ZPVE* scales with molecular size, and so supplying this chemical information improves the extrapolative sampling of high $|U_0|$ structures; however, E_g is weakly correlated with U_0 , but still provides chemical information that improves the overall validity of sampled molecules.

5.3.2 Interpolative Sampling

The extrapolative sampling that was investigated above is representative of chemical discovery applications where champion property values are being sought for new compounds outside of the convex hull of the training data. In contrast, interpolative sampling is relevant to applications where a range of property values are desired with some representation in the training data. For instance, in photovoltaic applications it is relevant to target structures within the optical bandgap of 1.0-2.0 eV rather than compounds with extreme values. To investigate the baseline performance of interpolative sampling we trained autoencoders on an E_g prediction task. Ten separate models were trained to evaluate model variance, and novel compounds were generated for the target E_g ranges of 1.5-2.0 eV, 5.5-6.0 eV, and 9.5-10.0 eV based on the principal

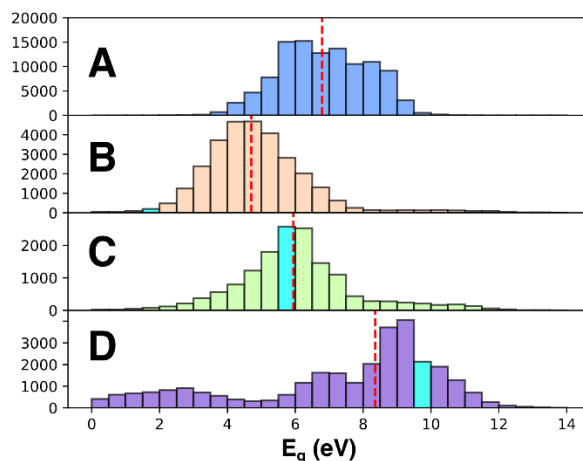


Figure 5.4: Distribution of E_g for the training data (a) and structures generated from models trained to predict E_g by targeting (b) 1.5-2.0 eV, (c) 5.5-6.0 eV, and (d) 9.5-10.0 eV. While (c) and (d) show good specificity, the model is unable to resolve structures in the 1.5-2.0 eV range (a). For each target, 3000 unique structures are generated across the 10 duplicate models for a total of 30,000 structures. The median of each distribution is indicated by a dashed red line. Targeted regions are highlighted in blue.

component corresponding to E_g (see methods). For each range, 3000 unique structures for each of the ten models were decoded and subjected to xTB calculation to determine E_g .

The histograms of E_g values resulting from the interpolative sampling procedure are shown in Figure 5.4. Although all of the targeted ranges have some representation in the training data, the impact of data imbalance is apparent in the results. The generative performance in the 5.5-6.0 eV range, which is well represented in the training set, exhibits good specificity (Fig. 5.4c). A narrow distribution develops with its most frequent value within the targeted region. Structures with bandgap between 9.5-10.0 eV (Fig. 5.4d) are much less well represented in the training data, and consequently the distribution is much broader and is not centered about the target range, although generation is still shifted towards high E_g structures, and a large number of structures are still recovered within the targeted region. The situation is much starker for the lowest bandgap target of 1.5-2.0 eV (Fig. 5.4b), which shows almost no representation in the training data (Fig. 5.4a) due to the rarity of such low bandgap structures in a dataset comprised of small molecules. The models exhibit limited ability to target compounds within this region, as evidenced by the distribution

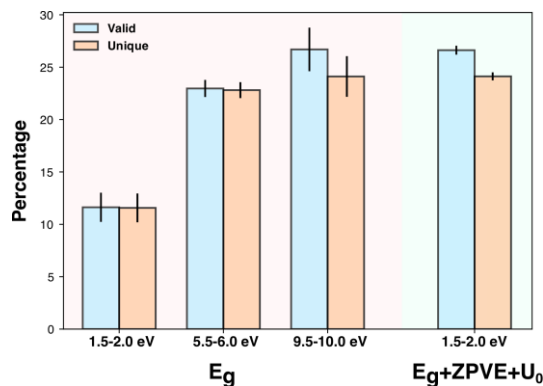


Figure 5.5: Average sampling validity and uniqueness for the three targeted E_g paradigms. Results are averaged across 10 distinct models for each training paradigm, with error bars denoting standard deviation. The models show difficulty in generating compounds within the poorly represented 1.5-2.0 eV range and are comparatively much stronger in generating structures with E_g between 5.5-6.0 and 9.5-10.0. The addition of ancillary U_0 and $ZPVE$ prediction tasks greatly increases the proportion of valid and unique structures generated in the 1.5-2.0 eV range.

centered at 4.5-5.0 eV. Only 200 structures with a bandgap of 1.5-2.0 were generated by these models, out of a total of 30000 generated compounds. Sampling in this region also tends to produce a much lower fraction of valid/unique structures than the higher E_g targets (Fig. 5.5).

Given the scarcity of training compounds exhibiting E_g in the 1.5-2.0 eV range, this generation task is closely analogous to the extrapolative sampling experiments performed for high

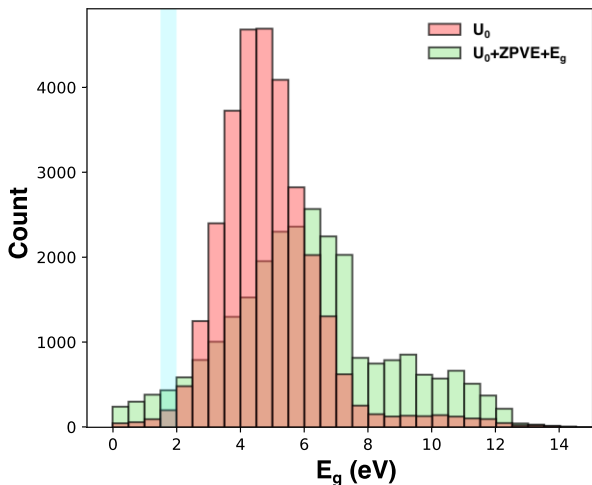


Figure 5.6: Distribution of E_g for structures generated from model trained to predict U_0 , $ZPVE$, and E_g . E_g is targeted within 1.5-2.0 eV while U_0 is extrapolated to bias the discovery of larger compounds. Compared to targeting E_g alone, the distribution of E_g is much wider, which allows for the generation of approximately twice as many low E_g structures compared with models trained on E_g alone. The targeted region is highlighted in blue.

$|U_0|$ structures. The number of structures with bandgap of 1.5-2.0 eV is severely limited within the QM9 database, making this a particularly difficult region to target. Because QM9 exhausts small molecule space, there is also a limited amount of target compounds to actually discover within the space spanned by the training data. In order to effectively target this underrepresented region, it is also necessary to expand the search beyond QM9 towards larger molecules. To investigate if multi-task TL could improve targeted generative performance, we followed the same procedure of high $|U_0|$ extrapolation while also targeting structures with E_g within 1.5-2.0 eV. We trained ten separate models on $E_g/ZPVE/U_0$ prediction tasks and performed targeted sampling for E_g in the 1.5-2.0 eV range while extrapolating along the U_0 principal component to facilitate the generation of larger structures. As shown in Figure 5.5, the inclusion of these ancillary prediction tasks more than triples the number of unique structures generated. However, it is clear from observing the bandgap histogram in Figure 5.6 that the increase in unique structures is not due to greater specificity. In fact, the distribution has been shifted towards higher E_g structures compared to the model trained to predict E_g alone, and the distribution has broadened. It is this broadening of the distribution that

is responsible for the increase in the number of unique structures generated, however it also allows the model to generate compounds with a bandgap of 1.5-2.0 eV. In particular, the number of structures within this range has more than doubled, with 440 structures compared to 200 for models trained on E_g prediction alone. The additional chemical property information provided by the $|U_0|$ and $ZPVE$ prediction tasks has been transferred to the task of generating unique structures with targeted properties by improving the ability of the model to sample within these underrepresented regions of chemical space. For the other bandgap targets, the inclusion of $ZPVE$ and U_0 data also serves to broaden the distribution of generated compounds (Fig. C6); however, as these regions are already targeted effectively, it is not advantageous to add in the additional property data in these cases.

5.4 Conclusions

These results demonstrate that multi-task transfer learning can be extended beyond property prediction towards improving generative performance. We observe that including chemical property data during generative model training provides complementary information to the syntactic, structural data that is typically used during training, with a generic positive impact on generative validity and performance on extrapolation tasks. In particular, by constraining the search to physically accessible property values, the validity of generated species is increased. For extrapolative sampling, it may prove difficult to generate structures that are not well represented in the training data due to significant differences in atom connectivity and topology; however, the information learned in an ancillary property prediction task can be transferred to the generation task, giving the model enough additional information to successfully resolve these new structures. For targeted structure generation within property regimes that may not be well represented in the training data, this mechanism may also be exploited to help the model effectively sample these underrepresented regions of chemical space and resolve compounds with the desired property values. We anticipate that this effect can be employed in other extrapolative applications to generate novel compounds with optimized molecular properties with a higher success rate.

6. ACTIVELY SEARCHING: INVERSE DESIGN OF NOVEL MOLECULES WITH SIMULTANEOUSLY OPTIMIZED PROPERTIES

The contents of this chapter in its entirety, including figures and supporting information, are currently under review for publication in the Journal of Chemistry and Physics A.

Combining quantum chemistry characterizations with generative machine learning models has the potential to accelerate molecular searches in chemical space. In this paradigm, quantum chemistry acts as a relatively cost-effective oracle for evaluating the properties of particular molecules while generative models provide a means of sampling chemical space based on learned structure-function relationships. For practical applications, multiple potentially orthogonal properties must be optimized in tandem during a discovery workflow. This carries additional difficulties associated with specificity of the targets and the ability for the model to reconcile all properties simultaneously. Here we demonstrate an active learning approach to improve the performance of multi-target generative chemical models. We first demonstrate the effectiveness of a set of baseline models trained on single property prediction tasks in generating novel compounds with various property targets, including both interpolative and extrapolative generation scenarios. For property ranges where accurate targeting proves difficult, the novel compounds suggested by the model are characterized using quantum chemistry to obtain the true values, and these new molecules closest to expressing the desired properties are fed back into the generative model for additional training. This gradually improves the generative models' understanding of unknown areas of chemical space and shifts the distribution of generated compounds towards the targeted values. We then demonstrate the effectiveness of this active learning approach in generating compounds with multiple chemical constraints, including vertical ionization potential, electron affinity, and dipole moment targets, and validate the results at the ω B97X-D3/def2-TZVP level. This method requires no modifications to extant generative approaches, but rather utilizes their inherent generative and predictive aspects for self-refinement and can be applied to situations where any number of properties with varying degrees of correlation must be optimized simultaneously.

6.1 Introduction

Machine learning (ML) has emerged as a powerful tool for solving previously intractable problems by extracting latent information from domain data, and has been effectively employed in areas as distinct as manufacturing analytics [172] and cancer detection.[173] In recent years, it has proved particularly successful in the chemical sciences, where ML has been used to predict interatomic potentials [83], quantum chemical properties [49], and structural data of polymers [174]·[175] and crystals.[176] Moving beyond the “forward-problem” of predicting molecular properties from a given chemical structure, generative chemical models have garnered significant interest in solving the “inverse-problem” of predicting a chemical structure from a given descriptor. As a large body of research in chemistry is devoted to creating novel compounds under functional constraints, these generative models have the potential to supplement and automate much of the often-laborious manual chemical optimization methods by providing reasonable chemical suggestions for more expensive experimental synthesis and characterization. Generative adversarial networks [177]·[164]·[160]·[178] (GANs) and various formulations of autoencoder networks [152]·[179]·[180]·[161]·[92]·[181]·[182] have emerged as some of the more popular frameworks for generative machine-learning based chemical design. These methods often provide for a predictive aspect which allows suggestions to be biased towards compounds with particular properties.[67] Much effort has been directed to solving issues related to the ability of these models to generate valid chemistries [183]·[164]·[165]·[166], and they have been successfully demonstrated in generating compounds with specific properties such as bandgap [103] and thermal conductivity.[184]

While models capable of optimizing one molecular property are compelling proof-of-principle demonstrations, multi-property optimization is required in any practical chemical discovery application. Because of the exponential scaling of search spaces with respect to the number of properties, multi-property chemical searches are fundamentally more challenging because they will typically be operating in an extrapolative regime (i.e., searching for properties outside the convex hull of training data ranges) and training data density drops in high dimensions. Several recent studies have highlighted the challenges and potential solutions to pursuing multi-property searches. Janet *et al.* balanced solubility and redox potential in the design of transition metal complexes for redox flow batteries using efficient global optimization to explore an enumerated space of 2.8 million candidate complexes.[185] Domenico *et al.* utilize reinforcement

learning for the design of drug-like molecules where the trade-offs among relevant physiochemical properties like molecular weight and hydrogen bond donors/acceptors, as well as similarity constraints to known drugs, are minimized. [186] Ståhl *et al.* also used a reinforcement learning approach to target and modify fragments in known structures to develop novel structures similar to known lead compounds but with optimized molecular weight, logP, and polar surface area. [187] Nigram *et al.* recently proposed the STONED algorithm, which side-steps the data limitations associated with training deep generative models and instead relies on string permutations of seed structures represented with semantically robust SELFIES.[188] Zhou *et al.* developed a reinforcement learning method based on atom/bond addition/removal to optimize compounds with respect to logP and quantitative estimate of drug likeliness (QED).[189] Interestingly, they also note that common targets for generative models may not be suitable for real world applications. LogP, for instance, may be trivially improved by simply increasing the length of carbon chains in a structure. Of interest is a method that can be applied generally to experimental properties or computational analogues.

Despite the large datasets available (and in many cases necessary) for training, certain combinations of properties are difficult for a generative model to achieve, either because they contradict basic physical relationships, or because they simply have limited representation within the training data. Rather than filtering an enumerated set of compounds or guiding the generation process with methods such as reinforcement learning, we propose leveraging the generative aspect of these models to enrich training data in targeted regions of chemical space. Generative chemical models have the unique feature that syntactically valid outputs are guaranteed to belong within chemical space, meaning that they are suitable samples for further model training. By sampling underrepresented regions of chemical space, new compounds may be discovered that are closer to the desired property space than any elements in the training set. By introducing the model to these new chemistries, the model can better learn which features correlate with the designated figures of merit. This framework falls under the paradigm of active learning. In active learning, a model can ask an expert source (i.e., the oracle) to annotate unlabeled training data that the model believes will be helpful.[190] This is particularly useful in situations where generating labeled data is difficult, as is often the case with chemical property data, because in the optimal case the model will utilize as little data as possible. This method was exploited by Konze *et al.* who used an active learning-based approach to more efficiently screen a large set of ligands without conducting

expensive free energy perturbations on the entire set.[191] This approach was extended in their recent follow-up to train a goal-directed generative model to generate promising ligands for further screening.[192] We propose formulating the entire goal of multi-target chemical optimization as an active learning problem. Rather than attempting to determine optimal regions of chemical space to sample or train on, we query to model to obtain its suggestions for compounds with the desired properties. These compounds are then screened via semi-empirical calculation (i.e., the oracle) and the model is retrained on those new compounds that actually match the target property profiles. In this way, the model develops its own training data to better understand new chemistries and iterating on this procedure provides the opportunity to continuously improve a model’s ability to target compounds with underrepresented properties.

Herein we examine this active learning framework to improve the performance of multi-objective generative chemical models. Utilizing a subset of compounds from the ZINC15 database [116], we develop our own dataset of quantum chemical properties including vertical ionization potential (VIP), electron affinity (EA), and dipole moment (DM) calculated at the semi-empirical level for training. We demonstrate the effectiveness of generative chemical models trained to propose compounds with a single targeted property, as well as multiple properties at once. We also demonstrate the shortcomings of such models, particularly when the combination of desired properties is not found in the training data, and how they may be overcome with active learning. We then validate the properties of the newly suggested compounds at the ω B97X-D3/def2-TZVP level. This active learning scheme can be applied to both single property and multi-property models to extrapolate to new regions of chemical space.

6.2 Methodology

6.2.1 Datasets

All models are initially trained and evaluated using structures from the ZINC15 dataset. This dataset contains 3D structural data for hundreds of millions of small molecules, from which we have chosen a subset of 250,000 compounds with molecular weight between 200 and 500 Daltons and logP between -1 and 5. These compounds were subjected to geometry optimization and electronic structure calculation with GFN2-xTB[151] (xTB) to obtain their DM, VIP, and EA. After removing compounds that failed the initial geometry optimization, we were left with 224,742

structures and their associated properties. 80 percent were utilized for training, with the remaining 20 percent withheld for validation. Additional structures generated during the active learning step were subjected to the same property calculation methods to expand the dataset. In order to validate the properties of the structures generated in the multi-objective active learning study, xTB optimized geometries are used as input to a geometry optimization at the ω B97X-D3/def2-TZVP level to determine the dipole moment of the neutral species.[193] Using the optimized neutral geometries, the DFT calculations are repeated for the cation and anion to calculate the vertical ionization potential and electron affinity, respectively. All DFT calculations were conducted using Orca 4.0.1.[125]

6.2.2 Machine Learning Architecture

Three models were developed for single target predictions and one model was developed for multi-property prediction tasks. We utilize the grammar variational autoencoder [68] (GVAE) to achieve generation of molecules with targeted properties, with the alteration of using a single linear predictor layer so that properties tend to vary linearly along the latent dimensions.[170] Training was conducted using the RMSprop algorithm with a learning rate of 0.001, which was set to decay by a factor of 0.3 in the case of a plateau in the validation loss. KL divergence loss was scaled by 750, and the categorical cross-entropy loss associated with encoding and decoding was decayed from 50 to 1 during training according to a sigmoid function. All properties were normalized to fall within a range of -20 to 20. The normalization and scaling factors were selected to balance property prediction accuracy, encoding and decoding accuracy, and the ability to decode novel structures from arbitrary latent points. Additional training details are provided in Appendix D.

6.2.3 Sampling Paradigms

With a fully trained autoencoder, new molecules may be decoded from arbitrary points in the chemical latent space. Jointly training the autoencoder with a property prediction task based on a linear prediction network ensures that those properties will vary linearly along the principal components of the latent encodings.[170] Compounds with specific properties can then be generated by targeting regions of the latent space based on univariate linear regression between the property of interest and the position along one of the principle components. In the case of multi-

property models, correlation between the properties may lead to latent space organization not being exactly orthogonal. To find the direction to sample, we linearly regress the angles that the points must be rotated by to maximize the R^2 between position along a particular principal component and the property of interest. This allows us to continue to utilize a simple linear regression to target specific properties. All other dimensions are sampled normally with mean and standard deviation determined by the training data.

6.2.4 Active Learning Technique

Depending on the nature of the property and the range for which it is sampled, and particularly for extrapolative property searches, the model may not be able to generate structures with properties that match those suggested by the regression outlined above. To circumvent this, structures are sampled from the regions of latent space that the model predicts to be of interest, which are then used for retraining the model. In each iteration, 100,000 unique structures are sampled from the model and the canonical SMILES of these structures are checked against the training and validation datasets to ensure novelty of the generated structures. Those novel structures are then subjected to xTB calculations for characterization. As the sampling routine is not perfect, these compounds are filtered to ensure they fall within the desired range before usage in retraining. In situations where no compounds fall within the desired range, we instead select molecules with properties that fall above or below (depending on the extrapolated target region) the median value in the training set, thus still providing the model with compounds exhibiting properties that are closer to the targeted range than the original training data. It was found that simply retraining the model on this newly generated data significantly harmed performance; this effect could be due to the significant differences between the new data and the original training data leading to catastrophic forgetting. Instead of only retraining with the new data, which due to the screening process always contains less than 100,000 total structures, compounds from the training set are randomly sampled and added to this new dataset until reaching 100,000 total training structures. This dataset size was found to train effectively using the same hyperparameters initially used in training. This routine was repeated until the desired number of structures with the targeted property ranges was obtained.

6.3 Results and Discussion

6.3.1 Single Property Searches

In previous work, we have demonstrated that the use of chemical autoencoders for targeted structure searching is effective for properties within the GDB19 dataset [150], namely internal energy, zero-point vibrational energy, and HOMO-LUMO gap.[194] To examine the generality of this approach, we investigated three models trained to individually predict VIP, EA, and DM, and sampled 100,000 structures in property ranges poorly represented in the training data. The targeted ranges for VIP, EA, and DM were 10.0 to 11.0 eV, -2.0 to -1.0 eV, and 0.0 to 1.0 Debye, respectively (Fig. 6.1). For VIP and EA, the chosen property ranges are not found within the training data at all, whereas for DM a poorly represented range was instead selected due to the lower bound on DM values and the long tail for compounds with very high DM in the training data. The sampling technique proves very effective for EA, with the mean and the majority of the sampling

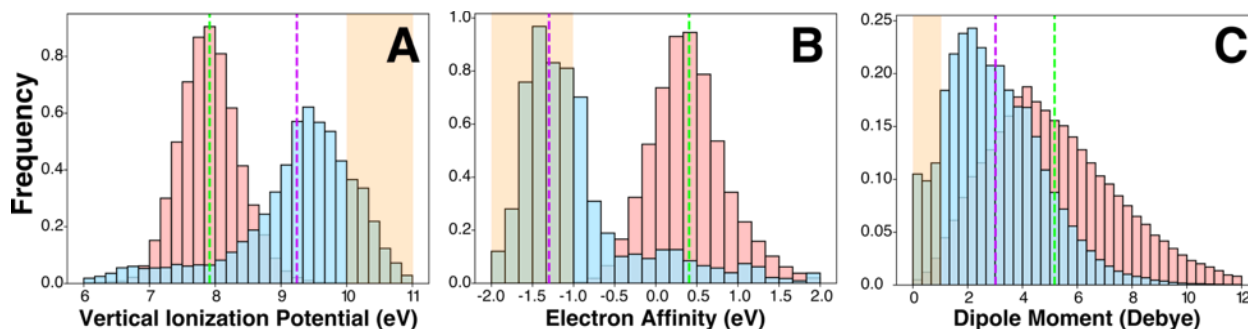


Figure 6.1: Property histograms for molecules generated by models trained on (A) vertical ionization potential, (B) electron affinity, and (C) dipole moment. For each model, 100,000 structures were generated and subsequently characterized at the xTB level. Training distributions are shown in red with generated data in blue. Means of the training distributions and generated distributions are indicated by green and purple dotted lines, respectively. Models are tasked with extrapolating to compounds with property values not observed in the training data, shown in orange.

distribution (57%) falling within the target region. While the results for VIP and DM are not as extreme, both distributions undergo a clear shift towards the targeted region, with a high number of generated structures (17% and 11%, respectively) fulfilling the target criterion in both situations. We also note for the case of DM that the lower bound on possible values may impact the number

of generated structures in this regime. For all three properties, the model has learned enough chemical information from the initial training set alone to determine the relationship between the property of interest and the targeted structures.

6.3.2 Single-Property Active Learning

For property ranges that represent fundamentally different chemistries than those found in the training data, the model may not have learned the necessary structure-function relationships to effectively generate new structures with the targeted properties. As a demonstration, we performed a generative search for structures exhibiting EA values between 1.0 and 2.0 eV, which is approximately one standard deviation higher than the mean EA of the training data, but still in the interpolative regime (Fig. 6.2). While 17% of structures sampled from the model are in the targeted EA range, this is only marginally higher than the training distribution and reflects limited specificity for high EA species. Although the model has not yet learned a strong relationship between chemical structure and the targeted EA range, the sampling still yields a large number of new structures exhibiting EA within or near the targeted range. Using the iterative approach outlined in the methods section, these new structures were incorporated into the training data to allow the network to resolve the functional relationships in the targeted EA region. After 4 iterations of sampling and retraining, the bulk of the sampled distribution shifted, with 30% of structures falling within the desired range. After 9 iterations, the mean of the distribution shifted squarely within the 1-2 eV range and 35% of the sampled structures exhibited EA values within the target. Thus, even for single property optimization, the active learning approach is effective in teaching the model the missing chemistries it needs to understand and generate high EA compounds.

6.3.3 Multi-Target Optimization

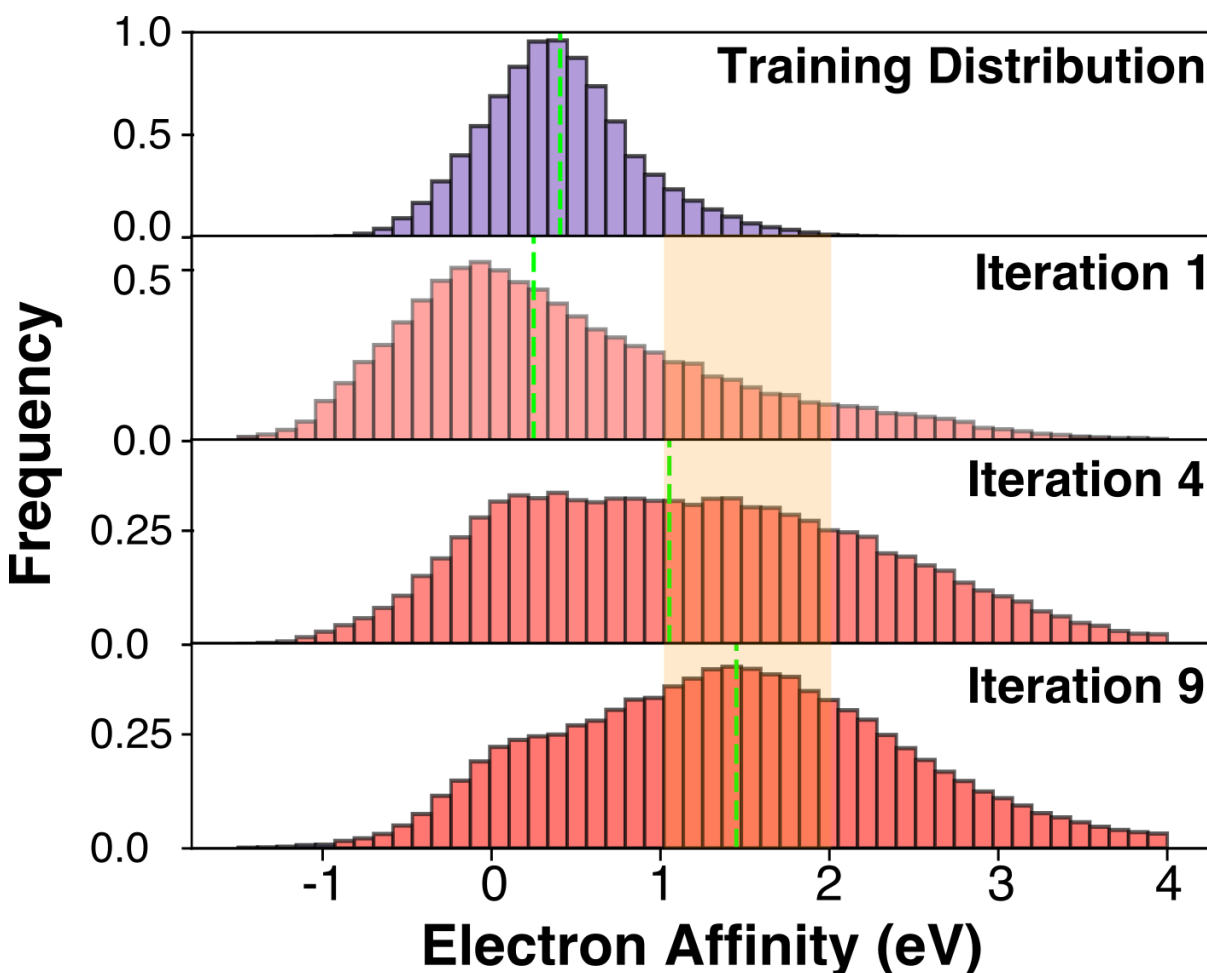


Figure 6.2: Property histograms for model trained to predict electron affinity. Training distribution is shown in purple (TOP), means are indicated by dashed green line, and the target region of 1-2 eV is highlighted in yellow. Initially, the model has difficulty generating structures with the specified EA (Iteration 1). After 4 and finally 9 iterations, the mean of the generated EAs has shifted to be within the target range, where the distribution also peaks.

While one property may be of primary interest in a particular molecular search (i.e., single-target optimization), there are often multiple properties that must be optimized simultaneously. This is often a much more difficult task, as these properties may have varying degrees of correlation and representation in the training data, an issue that is further compounded when considering the exponential growth of property space with respect to the number of optimized properties. For instance, the challenge of multi-property optimization is apparent if we consider searching for

compounds with EA between 1.5-4.0 eV, DM between 4.0-5.0 Debye, and VIP above 10.0 eV. The DM and EA ranges are represented in the training data, and the property range for VIP may be individually sampled effectively, as the experiment in Figure 6.1 demonstrated. However, when considering all three properties in tandem, no training structures simultaneously exhibit this range of values. Moreover, attempting to sample structures from this region of the latent space is

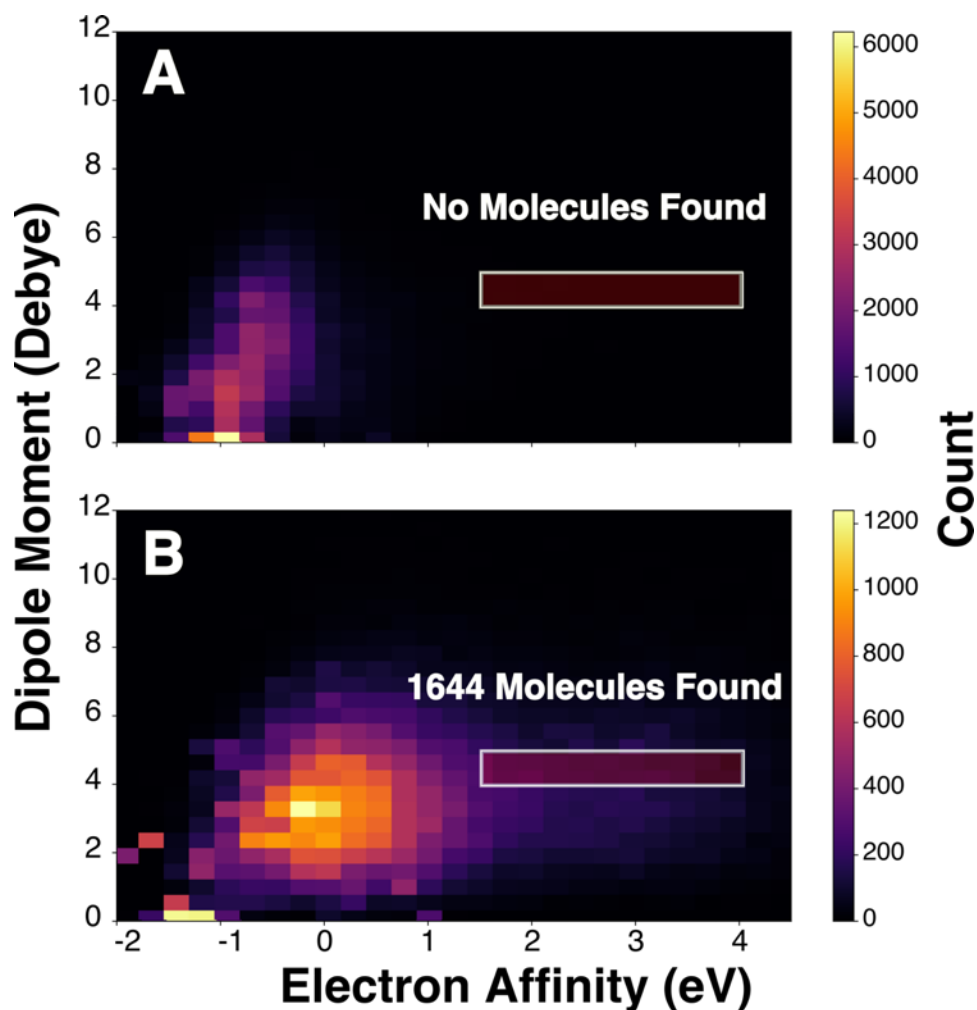


Figure 6.3: 2D property histogram for model trained to predict VIP, EA, and DM, and tasked with targeted structure generation for these properties. For visualization, only compounds with VIP greater than 10.0 eV are considered. The targeted region, with DM between 4-5 Debye and EA between 1.5-4.0 eV, is indicated with a box. Ionization potential is extrapolated beyond the training data, while the electron affinity range has little representation, and the dipole moment range is very well represented. Initially, (A) the model is not effective in generating compounds that fulfil all three criteria together. After 8 iterations of the active learning procedure (B), the property distribution of proposed structures has shifted to cover the targeted region and the model is now capable of proposing over 1600 structures fulfilling all three property criteria.

unsuccessful (Fig. 6.3A), with none of 100,000 sampled compounds falling within the desired property ranges. Figure 6.3A also demonstrates that simply retraining on new molecules optimized for individual properties would be ineffective, as their other properties are highly unlikely to fall within the desired range. However, after 8 iterations of retraining and resampling (Fig. 6.3B), the sampling distribution shifted towards the multi-dimensional property target, with over 1600 target structures being successfully generated. This demonstrates the potential for active learning as a framework to effectively fill in a model's chemical understanding, particularly in the case of multi-target extrapolative searching, where the increased dimensionality of the search space decreases potential coverage of the training data.

6.3.4 External Validation

In a practical scenario, the active learning-based search procedure discussed above would be the first step in a computational funnel to pare down the search space of viable molecules to a promising set for experimental study. However, we can further tighten this computational funnel through additional screening at a higher level of theory. Given the discrepancy between property calculations at the semi-empirical xTB and DFT levels, when selecting molecules for further screening we allowed for a soft-cutoff by adding $\pm 20\%$ of the target range to the property bounds in order to avoid screening out near-misses. To focus only on those compounds with the potential to be easily synthesized, we further reduced the list by screening out radicals, charged species, zwitterions, and structures with experimentally infeasible structures, such as those with linear oxygen chains of more than two atoms. This resulted in 307 candidate structures, which were then characterized at the ω B97X-D3/def2-TZVP level. After DFT characterizations, 22 structures passed the soft-cutoff criteria for all properties (Fig. D1), and 5 passed the exact criteria for all properties (Fig. 6.4). Inspecting the passing structures provides insight into the structure-function relationships that the model has learned to meet the targeted property ranges. We immediately note that all of the proposed structures are oxygen-rich. This is consistent with the high ionization potential target, which is promoted by including highly electronegative atoms and associated

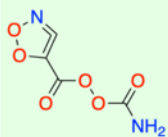
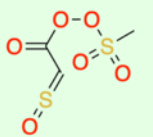
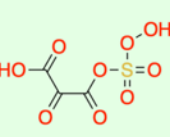

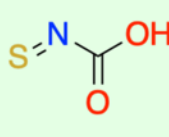
					
Vertical Ionization Potential (eV)	10.69	10.69	11.07	10.56	10.01
Electron Affinity (eV)	1.62	1.64	1.60	1.65	2.71
Dipole Moment (Debye)	4.80	4.60	4.18	4.24	4.11

Figure 6.4: Final five structures simultaneously achieving desired vertical ionization potential, electron affinity, and dipole moment values validated at ω B97X-D3/def2-TZVP level. The model has learned to make heavy use of oxygen atoms to achieve a high ionization potential while simultaneously maintaining a high electron affinity.

functional groups. Only one electronegative fluorine substitution is present in the final structures, although several examples occur in the structures that satisfy the relaxed criteria (Fig. D1). We provisionally interpret the preference for oxygen substitutions over fluorine substitutions as being due to the interplay between the dipole moment and ionization potential targets. In particular, we only observe isolated fluorine substitutions, which is consistent with the relatively high dipole moment target. The high occurrence of oxygen, as well as nitrogen and fluorine, also promotes high electron affinity, the second targeted property. Finally, we note that the model has also learned to avoid symmetric structures, which is necessary for the generation of molecules with a large dipole moment.

As an additional demonstration, we conducted an analogous study on multi-property optimization in an interpolative regime with respect to the training data. The targeted property values ($VIP \in [6.0, 7.0 \text{ eV}]$, $EA \in [0.5, 1.0 \text{ eV}]$, and $DM \in [4, 5 \text{ D}]$), had some representation in the training data (~ 160 samples), but were shifted from the mean of each property value (Fig. D2). Vertical ionization potential was shifted downwards by approximately two standard deviations, while electron affinity was shifted upwards by approximately one standard deviation. Dipole moment was not shifted relative to the mean to ensure some representation of the selected property ranges in the training data. After 6 iterations of the active learning-based retraining, 1599 novel structures were sampled that satisfy the targeted property ranges at the xTB level, and of these 16 matched all property ranges after validation at the ω B97X-D3/def2-TZVP level (Fig. D3).

Intuitively, fewer cycles of retraining were required to find more matching structures, since the property ranges already exhibited some representation with the training dataset.

6.4 Conclusions

Multi-objective chemical optimization presents unique challenges compared with single-objective optimization, such as achieving simultaneous specificity for multiple targets, and data sparsity due to the increased dimensionality of the property search space. We have demonstrated that generative models can be coupled with active learning-based retraining to predict novel structures designed to have specific properties, even when these properties may not be observed in the training data. For difficult targets, particularly multi-objective targets, a generative chemical model can learn the prerequisite chemistries by iteratively retraining on compounds it proposes that are similar to those that are desired. In this way, the model generates its own nascent structure-function relationships that it refines by sampling predicted structures. We have shown the ability of this method to propose compounds with specific vertical ionization potential, electron affinity, and dipole moment individually and simultaneously, and anticipate its utility for other sets of chemical properties. We also note that the quality of training data is a critical factor to ensure the properties of proposed molecules accurately match their true values. The discrepancy noted between property values at the xTB and DFT levels could be relieved by using an auxiliary difference model that predicts the difference between the low and high accuracy computational methods and optimizing with respect to this variable instead. This may allow for more efficient sampling and fewer iterations of the active learning procedure.

7. THERMODYNAMIC PROPERTY PREDICTION IMPROVES STRUCTURAL REALISM/SYNTHESIZABILITY/ACCESSIBILITY OF DEEP GENERATIVE MODELS

The contents of this chapter in its entirety, including figures and supporting information, are currently in preparation for journal submission

Generative chemical models have undergone significant development to improve the quality of their proposals, including improving property specificity and eliminating the prediction of invalid molecules (e.g., molecules that violate elementary bonding rules, or exhibit impossible Lewis structures). Despite this progress, contemporary generative models have limited ability to eliminate spurious structures that are not *prima facie* absurd, but nevertheless are neither stable nor synthesizable. Recent approaches have attempted to address this limitation by using established sets of reactants to bias generated molecules towards regions of chemical space that are already known to be synthetically accessible. Here we expand on this by testing whether including thermodynamic criteria during the generative process can improve the realism of sampled structures. We first demonstrate that a reaction-proposal framework can be extended to propose reactants that yield a product with targeted bandgap, even if the target lies above or below the extent of the training data. We go on to show how the same method can be applied to target individual reactions with specific enthalpy of reaction in both exothermic and endothermic regimes. We then demonstrate that thermodynamically unconstrained optimization of molecular properties leads to proposals that are thermodynamically unfavorable and show that by combining the product property data with thermodynamic descriptors of the related reaction, we can optimize the properties of our target molecule, while ensuring that it can be approached in a thermodynamically favorable reaction. By leveraging thermodynamic data in tandem with property optimization routines, the question of synthesizability may be approached in a data driven fashion.

7.1 Introduction

Machine learning (ML) methodologies are currently under intense investigation for all stages of materials development, including discovery, screening, characterization, and device translation. With respect to discovery, a compelling line of research is focused on whether deep generative

models can meaningfully contribute to the creative process of materials conceptualization. Specifically, while ML-approaches have a long history of assisting the “forward-problem” of predicting the properties of given structures, the “inverse-problem” of extracting and applying structure-function relationships to generate prospective structures has typically been the job of domain experts. Slightly more sophisticated versions of solving the forward-problem are provided by algorithmic searches of chemical space (e.g., genetic algorithms) that bypass the need to actually learning structure-function relationships. In contrast, contemporary deep generative models attempt to directly solve the inverse problem by predicting putative structures that match property values. In the ideal scenario, researchers could reduce costly and time-consuming synthesis-test-refine cycles, and instead select the properties they require and simply have the generative model suggest appropriate structure(s). Nevertheless, there is a large gap between this vision and the capabilities of contemporary models.

In an effort to achieve this vision, the evolution of molecular generative models has been largely driven by the desire to improve the quality of these proposed compounds. Initially this took the form of simply generating valid chemical graphs. While early work on deep generative chemical models brought the idea of inverse-design to the forefront of ML for chemistry, it also shed light on the difficulty of generating valid compounds. General molecular searches resulted in success rates as low as 4% [67], or produced output with a tendency to be too similar to the training data or of questionable design, [157] dampening the utility of these methods.

Follow-up approaches focused on more robust chemical representation, [68] while contemporary deep generative models will often apply a constraint to *ensure* the syntactic validity of generated compounds, such as recursively generating molecular graphs using a predefined vocabulary of compatible moieties [183] and valency rules [195]; by definition, these methods produce valid chemical graphs 100% of the time. However, it is not enough to consider only the syntactic validity of proposed species; it is crucial to also consider *semantic* constraints. Particularly for targeted structure generation, there is a high risk of generating chemical graphs that do not violate any valency rules, but nonetheless refer to clearly unreasonable chemical structures.[69] The structure-function relationships that a generative model learns while optimizing a given property value or fitness score may not necessarily correlate with ease of synthesis. A common failure mode for generative models is to generate unphysical samples in an attempt to "cheat" the scoring function. For instance, while a chain of oxygen atoms may not

violate valency rules, it is neither stable nor synthesizable under ordinary conditions. Despite the clear impracticality of such structures, this type of semantic error is a common failure mode observed in deep generative models.[196]

Recent efforts have explored several strategies for incorporating synthesizability as a learning target for generative models. Numerical proxies like the synthetic accessibility (SA) score [197] have been employed as surrogates for synthesizability with varying levels of success.[69] However, synthesizability is noted to be a much more nuanced function of chemical structure than many other properties, since small perturbations in molecular structure can necessitate dramatically different synthesis routes. Compared with molecular validity, synthesizability is a much more complicated function that requires expert analysis and/or rigorous computation to evaluate. In an attempt to ensure synthesizability, Bradshaw et al. introduced “MoleculeChef,” [198] a framework based on a Wasserstein auto-encoder, [199] along with an ancillary reaction prediction network, with an operating domain constrained to a predefined set of commonly found reagents. In this way, the authors were able to obtain high metrics for the validity, novelty, and even quality [200] of their proposed compounds while also providing for a path towards synthesis by simultaneously suggesting a suitable set of reactants. They go on to demonstrate the potential of this method for computer-aided retrosynthesis by generating a potential set of reactants for a desired product molecule. Analogous to the development of higher levels of density functional theory that more and more closely approach "chemical accuracy", [201] we can cast the development of molecular generative models as attempts to climb the “Jacob's Ladder” of *de novo* design, and move closer to the ideal of true inverse design. Just as earlier approaches represented our first attempts at bridging the experimental gap in the discovery workflow by ensuring syntactically valid structures, we can consider MoleculeChef as an attempt to move to the next rung of the ladder by ensuring synthesizability.

Mirroring constraint and vocabulary-based approaches for improving validity of proposed molecules, constraining the operating space of a generative model to common reagents only partially addresses the issue of synthesizability. For instance, the products predicted from MoleculeChef have no formal constraint on atom balance, and thus can still deviate substantially from the reagent-centered search space to predict unphysical structures. Additionally, the lack of relevant thermodynamic data during training, combined with a reaction predictor that must necessarily be operating outside of the bounds of its training set, may hinder the actual

synthesizability of the proposed compounds. Beyond synthesizability, the *efficacy* of the synthesis pathway is also a relevant design feature. While a given synthetic pathway may be theoretically possible, difficult operating conditions (e.g., extreme temperatures or pressures), troublesome (or even hazardous) byproducts, and kinetic limitations (specificity, yield, rate, etc.) may render it physically impractical. To bridge the gap from *in silico* predictions to the process scale, we cannot only consider a recipe to go from A to B; we must find an optimal path from some set of reactants A (i.e., the common reagents a chemist may have in the lab) to some targeted compound B.

Herein, we demonstrate the importance of including thermodynamic considerations during generative model development. We first demonstrate how generative chemical models can be adapted to not only propose target molecules (in this case, product molecules) exhibiting desirable properties as has been conclusively demonstrated in the literature but can also be extended to also be selective towards the thermodynamic favorability of the proposals. We go on to compare the enthalpies of reactions between proposals that only target the product property and those that also constrain the enthalpy of reaction, and we provide evidence that thermodynamically-unbounded generative design results in a high proportion of physically unobtainable molecules. We conclude by demonstrating the importance of including thermodynamic data in the chemical design framework as a means of not only targeting specific molecular properties, but as a prerequisite for ensuring synthesizability.

7.2 Computational Methods

7.2.1 Reaction Autoencoder

The molecular design framework utilized in this study is based off of the MoleculeChef paradigm, consisting of a reactant autoencoder coupled with an ancillary reaction prediction network. Molecules from a predefined set of reactants are converted into a vectoral representation using a stand-alone gated graph neural network (GGNN).[202] For a given multiset of reactants, graph embeddings are summed together to provide an invariant representation with respect to isomorphisms. The reactant set representations are passed to a feed-forward neural network whose outputs define the mean and standard deviation of the encoding distribution. The decoder samples from this latent distribution to initialize a recurrent neural network (RNN) that outputs probability distributions associated with the possible reactants. Output reactants are passed to the next timestep

of the RNN, and the process is repeated until a “stop” token is returned, ideally returning the original reactant "bag". In learning to encode and decode sets of reactants, the model learns an efficient, compressed chemical representation in its latent space; these reactant latent vectors serve as suitable inputs to property prediction networks. In a departure from the original MoleculeChef implementation, we consider any properties to be simple linear functions of the latent encoding. Our prediction networks thus take the form of a single node with linear activation function. For properties associated with the reaction (ΔH_r^\ominus), the property is computed at training time from the enthalpies of formation (ΔH_f) of the reactants and their known products. For properties associated with the product molecules, we take the maximum value observed in our set of products.

Three reactant autoencoder models were trained in total. All were trained to encode and decode sets of reactants, while one was trained to also predict the bandgap of the product associated with each reactant bag, one was trained to predict the enthalpy of reaction of the associated reaction, and one was trained to predict both.

7.2.2 Reaction Prediction

Because MoleculeChef encodes and decodes sets of reactants, information on the products and associated reaction is only handled implicitly through the property prediction layers. To actually obtain a predicted product from a sampled set of reactions, an auxiliary reaction prediction model must be employed. We utilize the open source implementation of MolecularTransformer (MT). [203] This attention-based seq-2-seq model does not rely on atom-mapped reactions to obtain its state-of-the-art predictions and is thus particularly well suited to the MoleculeChef problem where only the identities of the reactants are known. The use of a multiheaded attention-based architecture in lieu of the earlier RNN based models is particularly novel because it allows the encoder and decoder to simultaneously analyze multiple tokens, preventing the RNN-like inductive bias against tokens farther away in a SMILES sequence. MT is the first documented model to achieve up to 90.4% top-1 accuracy on the USPTO_MIT dataset and up to 78.1% top-1 accuracy on the USPTO_STEREO dataset. However, in a similar vein to the atom discrepancy observed in the USPTO dataset, because MT does not rely on atom mapping, it may make edits that lead to atom types not observed in the reactants. Again, here we assume that these additional atoms enter the reaction in their standard state and do not contribute to the enthalpy of reaction.

7.2.3 Datasets

We utilize the same split of the Lowe's USPTO reaction dataset [204] as the authors of MoleculeChef, [205] and also filter out any spectator molecules [206] and remove any duplicate reactions across the train, validate, test splits. We refine the reaction screening compared to the original implementation as follows: 1) we consider only those reactions containing C, H, N, O, F, Cl, and Br, 2) we remove 21 pathological cases that could not be handled by our ΔH_f characterization method at the time of writing, 3) we remove any charged species, including zwitterions, 4) we remove any molecules with radical groups, and 5) we remove any reactions containing species with fused rings. In order to maintain a similar number of reactions for training purposes, we relax the constraint that all reactants must appear in least 15 reactions to be included in our fixed reactant vocabulary, and instead only ensure that they appear at least 10 times. After this enhanced screening, we are left with 17910, 1012, and 1384 reactions in the training, validation, and testing sets, respectively, with an overall 2789 unique reactants among them. We note that within the USPTO dataset there are many reactions where the product molecule(s) may be missing atoms compared to the reactant molecules (within the training dataset, the median discrepancy is one heavy atom). In such situations, we assume that the missing heavy atoms have been reduced to standard state, and thus do not contribute to the enthalpy of reaction.

7.2.4 Product Characterization

The close connection of the HOMO-LUMO gap E_g to physical properties such as bandgap, and its inclusion in a variety of molecular databases, such as QM9, motivated its use as the target product property in this study. Due to the lack of property data within the USPTO dataset, we have constructed our own property dataset for model development. RDKit [117] is used to embed the original SMILES representation into a 3D structure based on simple connectivity rules, after which the structure is optimized with the Merck molecular force field (MMFF). [207] These geometries are used as input to a geometry optimization at the semi-empirical level using GFN2-xTB [151]. Subsequently, these geometries are used to seed an optimization at the BP86/def2-SVP [208] [209] level of theory before a final optimization and property calculation at the ω B97X-D3/def2-TZVP [210][211] level.

During the course of this work, 15,000 additional novel reactions were generated. ΔH_f data was computed using the same TCIT calculation routine, while E_g was calculated at a simpler level of theory. Product proposals had their geometries optimized at the xTB level, the results of which were used directly for a single-point calculation at the ω B97X-D3/def2-TZVP level to determine E_g .

7.2.5 Reaction Characterization

Enthalpies of reaction are computed based on the standard enthalpies of formation at 298K of the constitutive product and reactant molecules. Accurate and efficient calculation of ΔH_r is achieved via Taffi Component Increment Theory (TCIT). [212] TCIT is the first component theory derived exclusively from quantum chemistry data, which can provide ΔH_f predictions for linear molecules close to chemical accuracy (~ 1 kcal/mol) in an on-the-fly manner. In recent work, [213] TCIT has been extended to cyclic molecules by introducing a transfer-leaning based ring correction model which proved to be transferable and cost-effective. Currently, TCIT covers a large swathe of organic chemical space which makes it feasible to efficiently generate high throughput ΔH_r^\ominus data sources. Because the space of reactants is closed, only the new products generated as a result of this study require additional characterization with TCIT.

7.2.6 Sampling Techniques

Because a continuous latent space of chemical reactions is obtained through training, arbitrary points in the latent space can be decoded to obtain new sets of reactions, and therefore new products. By including a property prediction task on top of this latent space, we fold in additional chemical information to these encodings, which has been demonstrated to generically improve generative [77] and predictive [170] (that is, of other properties) performance. Importantly, it also ensures that properties tend to vary predictably along particular directions in this latent space. The use of a linear predictor from latent encoding to property vector not only pushes the bulk of the training difficulty to the encoder, ensuring that the properties are embedded within the encoding, but it also ensures that properties vary linearly along the directions of greatest variance (principal components). This provides for easy targeted sampling of new reactions with particular enthalpies of reaction and/or product E_g , as the position in the latent space to sample may be determined

through simple linear regression. To avoid duplicate reactions, we store the multiset of reactants in the training and validation sets and screen sampled reactions against them. While MT could suggest multiple products from a given set of reactants, we only consider the most likely product and thus consider a unique set of reactants as having only one set of products.

7.3 Results and Discussion

7.3.1 Single Target Search: Bandgap

As a baseline test of the performance of this network paradigm for chemical design, we can perform the common test of generative chemical models and task the model with proposing molecules optimized for a particular property, but now with the added benefit of also obtaining a guess for a potential set of reactants to synthesize the target molecule. We anticipate that there may be some difficulty in this regard, as compared to other approaches the model must implicitly optimize the product molecule through multiple levels of abstraction: it must select a set of reactants which, when subjected to a reaction as defined by the ancillary Molecular Transformer, produce a molecule with that has the desired property. To gauge the performance of the framework in this task, we examine a model trained to encode and decode sets of reactants and to predict the bandgap of the associated product molecule. We then consider the task of generating molecules whose bandgap lies in extreme ranges of the training set, either less than 8 eV or greater than 12 eV, representing the 2nd and 97th percentiles, respectively. Observing the results in Figure 7.1, we note that despite the similar representation of both data ranges, extrapolating to high bandgap molecules results in a higher success rate than attempting to generate low bandgap molecules. Considering the bandgap as a measure of the kinetic stability vs. reactivity of a molecule [214] it is unsurprising that our training data, obtained from published patent data, tends to be primarily composed of stable molecules, hence a relatively high E_g . We would therefore expect the generation of stable compounds to be easier than an extrapolation to unstable moieties which were not observed during training. Regardless, in both the low and high bandgap generation attempts, we produce a distribution of structures that, on average, approaches the bounds of the target property. Inspecting the subset of structures drawn from each sampling regime, we can speculate on the chemical relationships the model has learned. Compared to structures in the training data, the low E_g structures exhibit a higher prevalence of aromatic rings, corresponding with more

reactive structures. In contrast, we observe no aromatic structures within the high bandgap examples, although we do see a preference towards smaller, aliphatic rings. We also note a comparative lack of halogens in the high E_g compounds, as well as the substitution of sulfide and thioketone groups with more stable sulfone groups. A multitude of stable groups have been learned and are utilized to achieve the target of high E_g by generating unreactive structures.

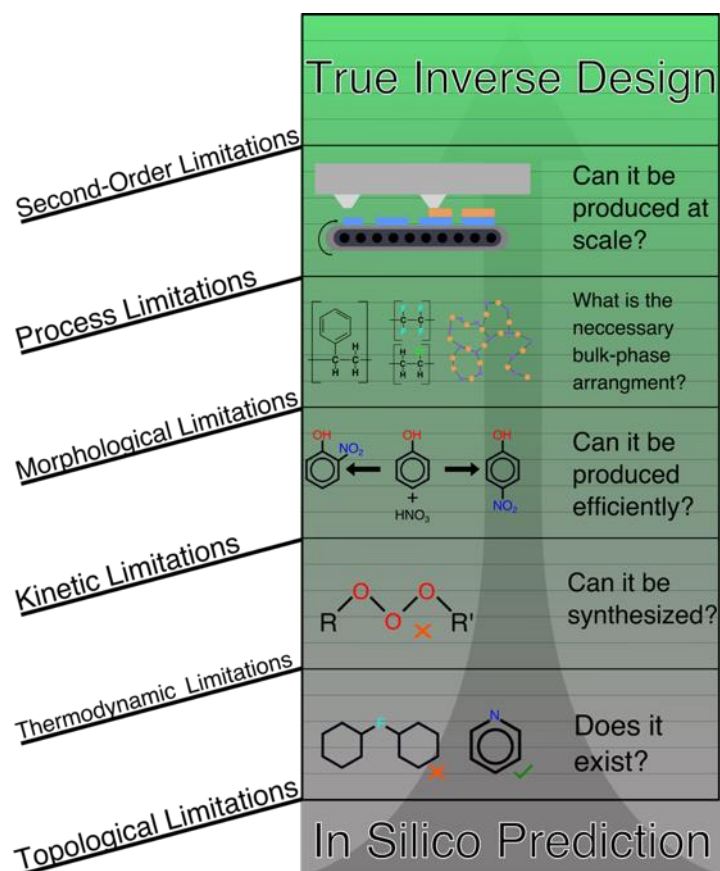


Figure 7.1: The "Jacob's Ladder" of inverse-design. In attempting to reconcile the domains of computation and the physical world, the first step is ensuring that proposed compounds do not violate valency and atom-type constraints ("Does it Exist?") The next step requires consideration of thermodynamic limitations, as without *a priori* constraints it is possible for the discovery workflow to suggest thermodynamically inaccessible molecules ("Can it be Synthesized?"). In this work, we attempt to address how we can reach this rung, as well as suggest methods for moving closer to true inverse design. We anticipate that further development of generative chemical methods will see methods designed to encapsulate bulk-phase morphologies, processing concerns, and other higher order considerations.

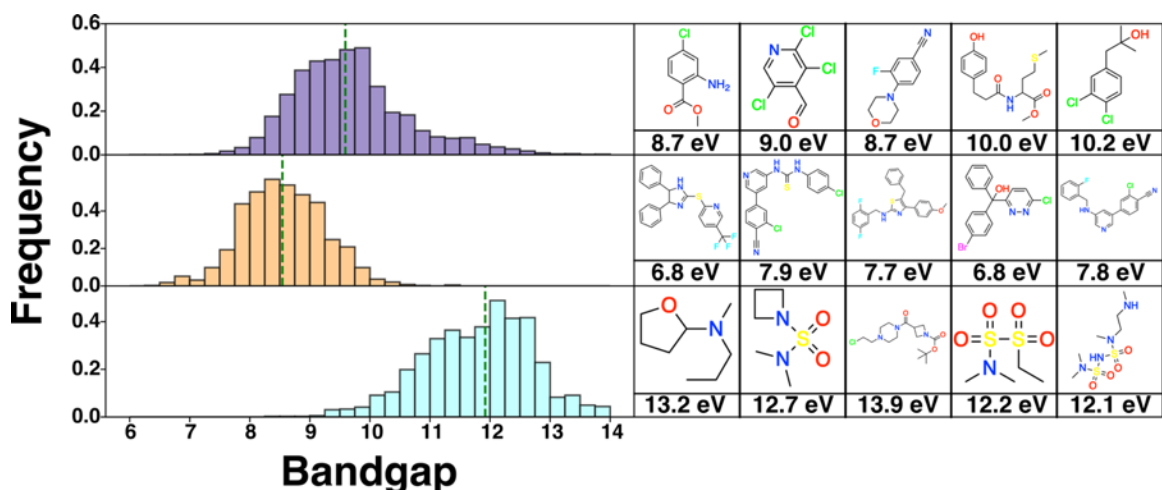


Figure 7.2: Property histograms and example molecules generated by models trained to predict molecular bandgap compared to distribution of training data (TOP). For each sampling paradigm, we generated 2000 novel reactions and subsequently characterized them at the ω B97X-D3/def2-TZVP level, with geometries obtained via xTB. We consider generating reactions that lead to a product with (MIDDLE) a bandgap lower than any member of the training set, and (BOTTOM) higher than any compound in the training set. Means of the training distribution and each generated distribution are indicated by dotted green lines, and the validated E_g are listed below each example molecule.

7.3.2 Single-Target Search: Enthalpy of Reaction

We have demonstrated the ability of this framework to propose molecules optimized for quantum chemical properties, along with providing a set of suitable reagents for synthesis. Now, equipped with the ability to natively handle reaction data, we examine the ability of our model to propose reactions exhibiting certain ΔH_r . In particular, we investigate a model trained to encode and decode sets of reactants along with predicting the ΔH_r of their associated reaction and sample 2000 reactions from various property ranges. We consider the potential of biasing reactions to be: endothermic, exothermic, and strongly endothermic, and summarize the results in Figure 7.3. We observe that the training data tends to consist of slightly endothermic reactions, a detail we attribute to the provenance of the dataset as a compilation of published reactions in the patent literature. This training bias contributes to the comparative ease in generating endothermic reactions, even those approaching thermodynamic infeasibility (ΔH_r above ~ 400 kJ/mol). Additionally, since our set of reactants tend to have negative enthalpy of formation, -209 kJ/mol on average, there are more avenues for moving up the energy landscape than moving down. Nonetheless, we still see the capability of the model to suggest exothermic reactions and can bias its proposals such that

they tend to have negative ΔH_r on average. For the model to do so, it must learn which types of reactions tend to produce stable products, a complex function described only implicitly in the training data. Through the addition of a reaction characterization routine, we can consider not only the optimization of individual molecules, but also the reactions that link them together.

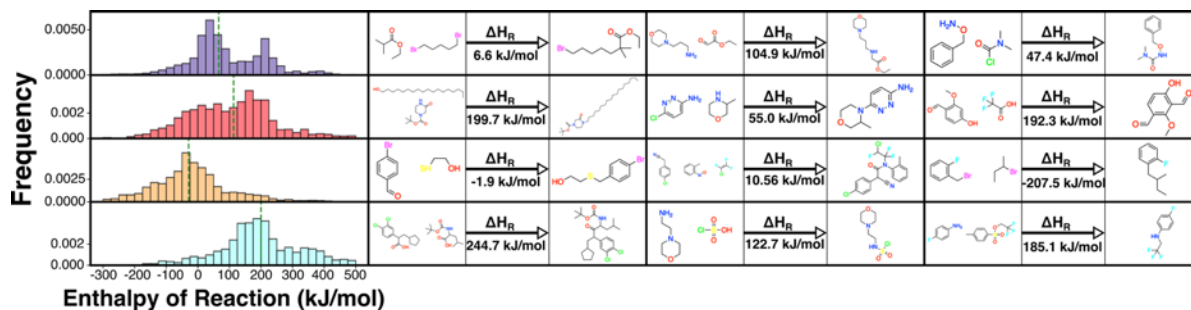


Figure 7.3: Property histograms and example reactions generated by models trained to predict enthalpy of reaction compared to distribution of training data (FIRST ROW). For each sampling regime, we generated 2000 novel reactions and subsequently characterized their products using TCIT. We considered generating reactions biased to be (SECOND ROW) endothermic, (THIRD ROW) exothermic, and (FOURTH ROW) strongly endothermic. Means of the training distribution and each generated distribution are indicated by dotted green lines. For each histogram, three example reactions are shown, with reactants separated from products by an arrow listing their associated enthalpy of reaction.

7.3.3 Multi-Target Search: Enthalpy and Bandgap

To this point, we have considered the generation of reactions with certain ΔH_r and the proposal of product molecules with particular bandgap as distinct tasks. However, as we have alluded to previously, the advancement of molecular searches in chemical space requires the union of the domains of property specificity and thermodynamic quality. We can see this need clearly represented with a closer look at the results of our single-target bandgap search. If we characterize the reactions associated with the proposals and examine their enthalpy of reaction, as presented in Figure 7.4 (TOP), we see that many proposals are extremely unfavorable, with exceedingly high enthalpies of reaction. Here we observe the key failing of non-thermodynamically restricted molecular discovery workflows: after a few reasonable proposals exhibiting the desired property, the model will be pushed towards more and more extravagant chemistries to achieve the target, "cheating" the generative process analogously to the failure modes observed in molecular-generative models.

Earlier, we demonstrated that reaction data can be directly folded into the molecular search framework to propose reactions with specific H_r . We can combine the two tasks, and now consider a model trained to encode and decode sets of reactants, predict the properties of a potential product if they were to react, and predict the enthalpy of reaction of the associated reaction. We sample 2000 reactions where the enthalpy of reaction is constrained to fall below -50 kJ/mol (to bias towards exothermic reactions while accounting for error in the model's H_r prediction) with bandgap targets of 6-8 eV as in the single-target bandgap trial. We compare the results of this search to the bandgap-only case in Figure 7.4. Compared to optimizing for bandgap alone, we are not only able to shift the generation of new reactions such that they tend to exhibit a higher or lower bandgap, critically we are able to ensure that their associated reactions tend to be thermodynamically favorable. Although the reactions do not strictly fall within the target range, the inclusion of the thermodynamic property data ensures that we do not observe the same broad ΔH_r distribution associated with nonphysical reactions, and we are able to more than double the number of successful proposals from 50 reactions to 120. Utilizing this framework, we have subsequently been exploring an enhanced retrosynthesis algorithm. Given a molecule with a particular characteristic, E_g in this case, and a known synthesis pathway, we can suggest a molecule with similar properties, but that has been converted to either an exo- or endothermic reaction. We plan to discuss this application further in future work.

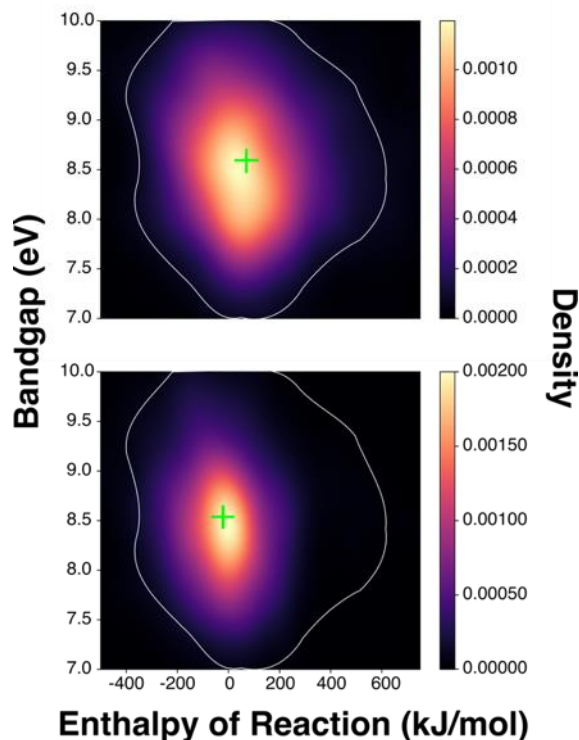


Figure 7.4: Kernel density estimates for distribution of bandgap and enthalpy of reaction for reactions generated from (TOP) a model trained only on predicting bandgap and (BOTTOM) a model trained to predict both enthalpy of reaction and product bandgap. Both models were tasked with generating 2000 reactions that produce a product molecule with bandgap between 6 and 8 eV, but the model trained on enthalpy of reaction was also requested to limit proposals to have enthalpy of reaction less than -50 kJ/mol. This more than doubles the number of thermodynamically feasible reactions attaining the bandgap target, from 50 to 120. Multidimensional means are indicated by green crosshairs. The extent of the bandgap-only histogram is outlined in white for comparison.

7.4 Conclusions

Connecting computationally designed molecules to actionable data has emerged as one of the key barriers to the realization of true inverse design. Much as early generative chemical models strove to address the topological validity of their proposals, recent developments have focused on ensuring the synthesizability of suggested molecules. We have demonstrated the acute need to include thermodynamic data as part of a ML-aided materials design pipeline to address questions of synthetic feasibility. While the ability of generative chemical models to propose novel species with targeted property data has been shown in the literature, we have demonstrated that without *a priori* constraints on their thermodynamic properties, proposals (even those with a topologically reasonable set of proposed reactants) can be experimentally inaccessible. We have gone on to show

that the enthalpy of reaction, a property associated with the reaction as a whole and not the individual molecules, may also be targeted in a data driven fashion and optimized in tandem with molecular property targets to ensure the proposal of a thermodynamically sound reaction leading to a champion molecular proposal. As the method we have utilized and the mechanism we have exposed are both decoupled from the input/output representations, we hope to motivate the general use of thermodynamic data during model training and deployment to help address concerns as to the synthesizability, stability, real-world potential of molecules produced via ML-aided techniques.

Finally, we note that even with aggressive reaction screening the nature of the Lowe patent dataset means that many reactions remain problematic. Often, only the major or target product is listed in the literature, leaving an incomplete atom balance and few clues as to the form the missing atoms have assumed. With a more robust reaction predictor, the reactions in the training set could be fully enumerated by use of a double-ended search, as both the reactants and major products are known, leaving the missing products to be those most likely to form under the given constraints. This would allow for more accurate estimates of the enthalpy of reaction and improve the quality of proposed reactions. In future work, we plan to correct for these limitations through the use of a forthcoming reaction prediction method, [215] allowing us to also account for activation energy of proposed reactions. In this way, we can fully modulate the path through reaction space our proposals follow, and therefore address both thermodynamic and kinetic considerations of the design process. With this work and subsequent studies, we hope to take one step closer to the goal of true inverse design.

8. CONCLUSIONS AND OUTLOOK

8.1 Summary

In the effort to bridge the gap between *in silico* predictions and true inverse design, there is still much work to be done. Current efforts are just now addressing the question of synthesizability, with many higher order considerations (stability, kinetic favorability, processing concerns, bulk-phase effects, etc.) still outstanding. Despite significant progress, with the inverse-design revolution beginning in 2016 and with further improvements already under development, the application of modern ML approaches to chemistry is hindered by data constraints for crucial application properties. The provenance of available data, the complexity and scope of media utilized in engineering, and the dearth of data for specific applications confound traditional analysis techniques. In my research to date, I have focused on methods for circumventing the issue of data scarcity in the development of effective models for chemical design applications, particularly approaches based on transfer and active-learning methodologies.

Transfer learning is an approach that leverages proficiency in one task to help in another, related task. My first study as a graduate student focused on applying transfer learning towards the prediction of the aqueous pK_a of small molecules. While experimental pK_a data is extremely limited in scope compared with other molecular properties, deprotonation free energy data can be calculated relatively easily using quantum chemistry and is a direct correlate of pK_a . By constructing the largest free-energy change database of its kind in the literature, an effective model for the prediction of pK_a could be developed with minimal experimental data requirements. In subsequent work, it was demonstrated that that this transfer learning approach can be generalized to the prediction of other molecular properties in data scarce scenarios, and that its efficiency could be improved with a novel neural network architecture.

Beyond the “forward-problem” of predicting molecular properties from a given chemical structure, the “inverse-problem” of finding an optimal set of molecular structures under functional constraints was also considered. It was postulated that the failure modes observed in contemporary inverse models, namely poor functional selectivity and high occurrence of unphysical chemical structures, were symptoms of insufficient chemical property data being utilized during training. Thus, the effect of tasking a neural network to predict these auxiliary properties along with targeted

properties for structure generation was examined. Several key findings were produced from this study. By simply constraining the search to compounds with physically accessible property values, the synthesizability of generated species is increased. This also increases property specificity of the suggested compounds. When attempting to reach property ranges beyond those in the training data, it may prove difficult to generate corresponding structures due to significant differences in atom connectivity and topology. By transferring the information learned in an ancillary property prediction task to the generation task, the model is given enough additional chemical information to successfully resolve these new structures. This provides for a closed-loop computational method to iteratively refine predictions and generate molecules with specific properties by continuously retraining the model on new compounds with unfamiliar chemistries that it itself proposes. This active-learning approach was recently demonstrated for multi-objective chemical optimization of materials with simultaneous ionization potential, electron affinity, and dipole moment targets. The curse of dimensionality for multi-target chemical design can lead to data scarce situations even among substantial datasets, making the closed-loop approach a particularly favorable method.

To help resolve issues with the synthetic feasibility of the proposals of generative chemical networks, an extension of the aforementioned frameworks and methodologies to reaction data was also explored. By operating with sets of reactants and including key thermodynamic data describing reaction feasibility, generative chemical models can be upgraded to not only target compounds with specific application properties, but also provide a reasonable set of reactants to get there. Anticipated improvements in reaction prediction tools and property characterization routines will help to further the benefits of this approach.

While data scarcity presents a unique problem for chemical science, it may be navigated with the enhanced learning methodologies and approaches addressed in this work. Their utility has been demonstrated in myriad applications, from more accurate property prediction to the discovery of new compounds with targeted features, and they are expected to be useful in any application where chemical data, whether experimental or computational, is difficult or laborious to collect.

8.2 Future Work and Outlook

Despite the significant progress of machine learning in the realm of small molecules and recent incursions into reaction space, the reaction prediction problem remains unsolved. In contrast to molecular property prediction, the reaction prediction problem has many distinct attributes that

require further ML methods development, including the causal relationships between reactants and products, specific featurization challenges (e.g., how to account for solvation conditions), and a uniquely large domain space, dwarfing the already vast space of individual molecules, contrasting with a uniquely limited space of available property data. The inability to handle reactions cascades into an inability to manage bulk phase data in much the same way due to the effect of processing history on most relevant mesoscale properties. For these reasons, further development of reaction-based models requires both fundamental methods development that addresses the unique challenges of reaction prediction and applied ML thrusts focused on discovering new reactions and elaborating complex reaction networks. Addressing the reaction problem will allow research to ascend to the next rung of the Jacob's Ladder of inverse design (Fig. 7.1) and allow morphological and processing limitations to be addressed. Additionally, advancements in processing power and development of more advanced and more affordable graphics processing units (GPUs) will allow for the deployment of new neural network architectures better suited to solving chemical problems, particularly those relying on multimodal data (e.g., spectra, optical imaging, time-series information) that will become prevalent when addressing reactions and bulk phase arrangements. To provide an example, training the chemical autoencoders used in this work on a standard CPU would have taken on the order of *one year*; however, GPU-equipped community clusters did not become available to the general research community at Purdue until 2017 with the Halstead-GPU and Brown-GPU clusters. Proliferation and advancement of chemical-ML research is thus anticipated not only on the basis of greater processing power, but also more widespread access to the resources necessary for development of ML-models.

APPENDIX A. SUPPORTING INFORMATION FOR: IMPROVED CHEMICAL PREDICTION FROM SCARCE DATA SETS VIA LATENT SPACE ENRICHMENT

Principal Component Analysis

Additional principal component projections are shown below, including models trained solely on pK_a and those enriched with E_{sp} data.

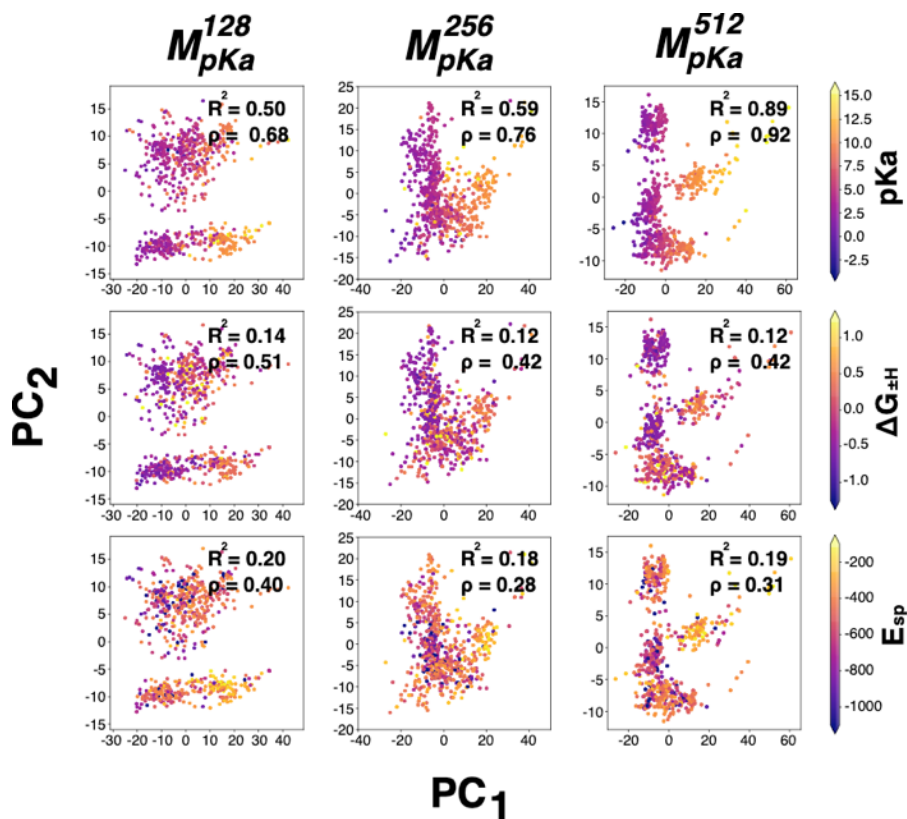


Figure A1. Projections within the principal component space for the unenriched models. We note that the resulting projections only display any significant ordering with respect to pK_a . The pK_a projections are of similar quality to those obtained with enrichment, but the secondary property projections are greatly improved after enrichment.

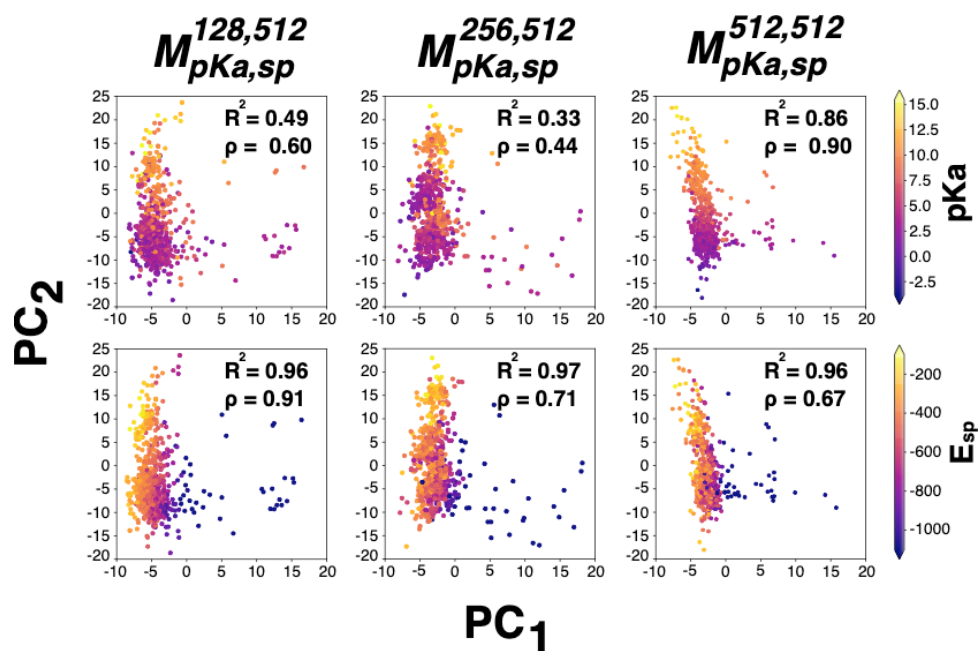


Figure A2. Principal component projections for models enriched with E_{sp} data. In contrast to ΔG_{\pm} enrichment, E_{sp} and pK_a properties are organized along orthogonal principal components dimensions

Latent Space Metric Summary

Statistical summaries of several latent space organization metrics are provided in Tables A1-3 with explanations of the metrics in the following section.

Table A1: Statistical summaries of latent space projections based on pK_a. Data highlighted in green to indicate transform providing best fit to data.

Model Type	R^2_{Linear}	R^2_{Square}	R^2_{Cubic}	R^2_{Root}	R^2_{Log}	$R^2_{Inverse}$	ρ
$M^{128}_{pK_a}$	0.50	0.49	0.44	0.49	0.46	0.03	0.68
$M^{256}_{pK_a}$	0.59	0.55	0.47	0.58	0.57	0.06	0.76
$M^{512}_{pK_a}$	0.86	0.89	0.86	0.84	0.80	0.07	0.92
$M^{128,512}_{pK_{a\Delta G}}$	0.41	0.39	0.33	0.40	0.38	0.03	0.64
$M^{256,512}_{pK_{a\Delta G}}$	0.58	0.54	0.44	0.57	0.55	0.06	0.74
$M^{512,512}_{pK_{a\Delta G}}$	0.75	0.77	0.73	0.73	0.70	0.05	0.81
$M^{128,512}_{pK_{a\Delta sp}}$	0.46	0.49	0.46	0.44	0.41	0.02	0.60
$M^{256,512}_{pK_{a\Delta sp}}$	0.30	0.33	0.31	0.29	0.27	0.02	0.44
$M^{512,512}_{pK_{a\Delta sp}}$	0.85	0.86	0.82	0.83	0.80	0.07	0.90

In general, organization of compounds with respect to pK_a follows a linear or quadratic trend along the principal components, with goodness of fit increasing as the amount of pK_a data used for training increases. The Spearman rank order coefficient, ρ , also increases in a similar manner, and appears to be invariant with respect to enrichment.

Table A2: Statistical summaries of latent space projections based on $\Delta G_{\pm H}$. Data highlighted in green to indicate transform providing best fit to data.

Model Type	R^2_{Linear}	R^2_{Square}	R^2_{Cubic}	R^2_{Root}	R^2_{Log}	$R^2_{Inverse}$	ρ
$M_{pK_a}^{128}$	0.04	0.00	0.00	0.09	0.14	0.00	0.51
$M_{pK_a}^{256}$	0.04	0.00	0.00	0.09	0.12	0.00	0.42
$M_{pK_a}^{512}$	0.05	0.00	0.00	0.09	0.12	0.00	0.42
$M_{pK_{a\Delta G}}^{128,512}$	0.52	0.94	0.99	0.33	0.25	0.00	0.77
$M_{pK_{a\Delta G}}^{256,512}$	0.55	0.95	0.99	0.33	0.22	0.00	0.70
$M_{pK_{a\Delta G}}^{512,512}$	0.52	0.93	0.97	0.30	0.19	0.00	0.56

$\Delta G_{\pm H}$ favors a different organization mechanism compared to pK_a , with either log transformation or cubic transformations providing the best fit. In the case of the jointly trained models, data points are extremely well organized after a cubic transformation according to ΔG_{\pm} .

Table A3: Statistical summaries of latent space projections based on E_{sp} . Data highlighted in green to indicate transform providing best fit to data.

Model Type	R^2_{Linear}	R^2_{Square}	R^2_{Cubic}	R^2_{Root}	R^2_{Log}	$R^2_{Inverse}$	ρ
$M_{pK_a}^{128}$	0.06	0.02	0.01	0.06	0.05	0.20	0.40
$M_{pK_a}^{256}$	0.03	0.01	0.01	0.03	0.03	0.18	0.28
$M_{pK_a}^{512}$	0.05	0.02	0.01	0.04	0.04	0.19	0.31
$M_{pK_{a,sp}}^{128,512}$	0.57	0.85	0.96	0.61	0.66	0.25	0.91
$M_{pK_{a,sp}}^{256,512}$	0.60	0.86	0.97	0.65	0.69	0.33	0.71
$M_{pK_{a,sp}}^{512,512}$	0.51	0.82	0.96	0.56	0.61	0.22	0.67

E_{sp} exhibits similar trends in R^2 found for $\Delta G_{\pm H}$, with dramatic improvement in R^2 upon enrichment. Without enrichment, E_{sp} shows the best organization with respect to the inverse transform but is still only weakly organized.

MAE Box and Whisker Plot

We include the box and whisker plot of the MAE in pK_a prediction for all model paradigms, including those models enriched with single-point energy data.

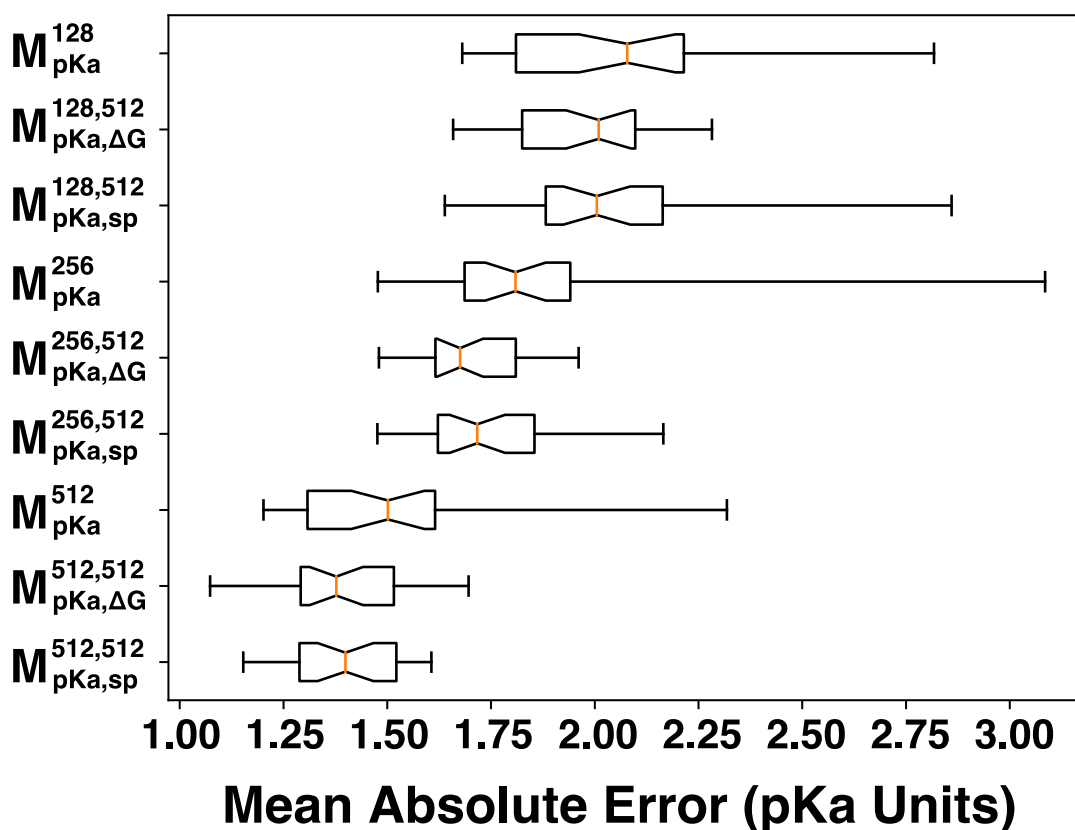


Figure A3: Box and whisker plot showing test error statistics for all model paradigms over thirty independently trained models for thirty independent training/testing splits. Median MAE values are indicated by an orange line, and whiskers extend to the range of observed errors. Notches represent the 95% confidence interval about the median. We note that those models enriched using free energy data exhibit lower median MAE values than those enriched with single point energy, although the overall effects are comparable.

Cross Validation Analysis

To aid in the hyperparameter search, we utilized cross-validation on a single testing/training split to gain insight as to how our models should perform on unseen data. For a given set of data (128,256,512 pKa values), the data is divided into 8 folds, one serving as a “testing” fold. The model is trained on the remaining 7 “training” folds and evaluated on the held-out fold. This is repeated across the 8 folds until each has an opportunity to act as the testing fold. The final model, trained on all 8 folds, should ideally similar or better performance to that observed during cross validation. The results of this cross-validation procedure are shown in Figure A4 below.

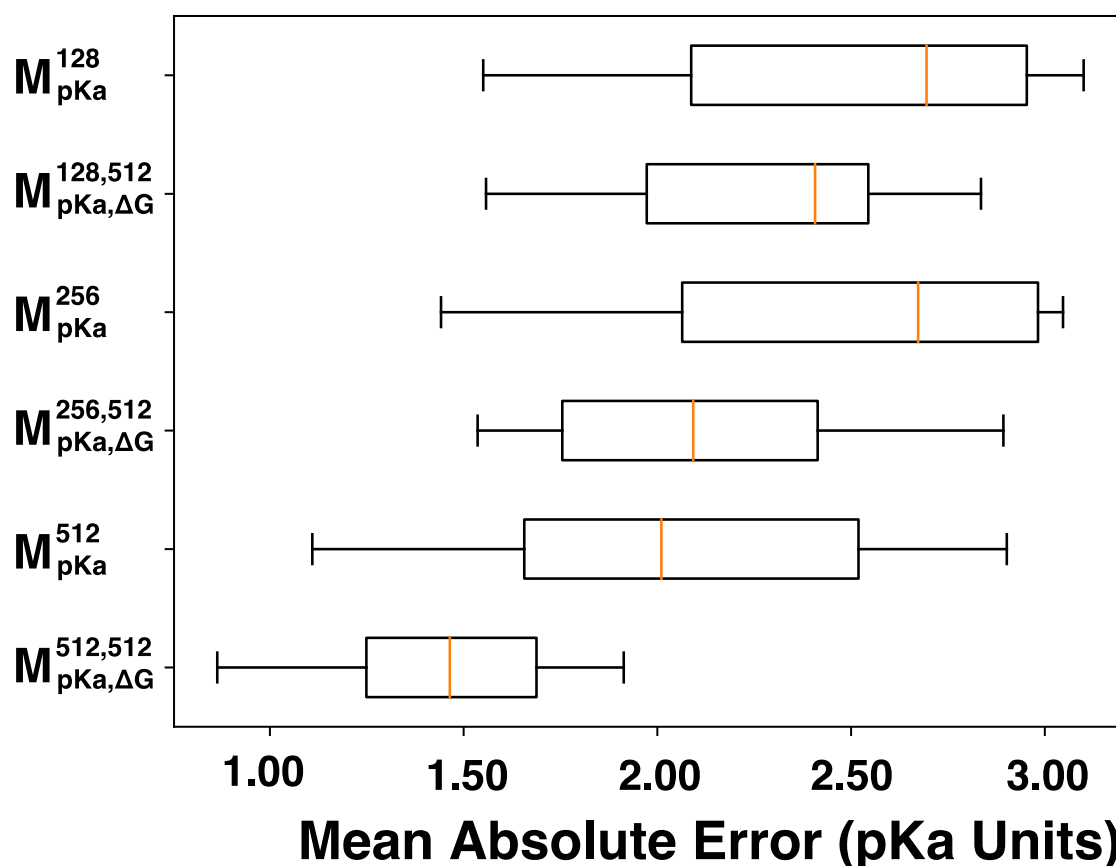


Figure A4: MAE for 8-fold cross validation on a particular testing and training split. Median MAE shown by the orange line. Notches representing the confidence interval about the median are not included as they extend beyond the IQR. We observe that, in all cases, enrichment results in a reduction in median MAE as well as a contraction in both the range of MAE values and the IQR.

Latent Space Characterization Methods

R² Calculation: The projections in the principal component space may be considered as elements of \mathbf{R}^3 having components x (principal component 1), y (principal component 2), and z (property value, e.g., pKa). Assuming the points in this space are well-organized, we can envision the scenario in which these points lie on a two-dimensional plane defined by

$$\mathbf{z} = a\mathbf{x} + b\mathbf{y} + c$$

Where a, b, c are constants and $\mathbf{x}, \mathbf{y}, \mathbf{z}$ represent the set of all projected points. The equation of the plane can then be determined by solving the expression

$$\mathbf{X}\mathbf{a} = \mathbf{z}$$

$$\mathbf{X} = \begin{pmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & N \end{pmatrix} \quad \mathbf{a} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}$$

Because \mathbf{X} is linearly dependent, we compute the pseudoinverse to obtain the least squares solution:

$$(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} \mathbf{a} = \mathbf{X}^+ \mathbf{X} \mathbf{a} = \mathbf{X}^+ \mathbf{z}$$

$$\mathbf{a} = \mathbf{X}^+ \mathbf{z}$$

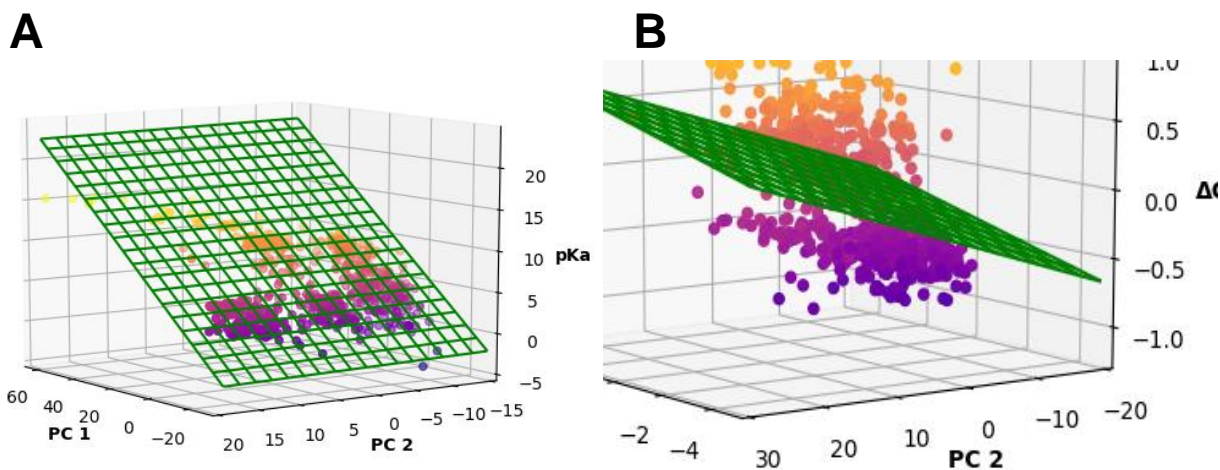


Figure A5. Illustrative examples of regression analysis of latent spaces. (A) the plane indicated by the green wireframe is fit to the principal component projections of the pKa data for the $M_{pK_a}^{512}$ model. We see that the projections within the latent space exhibit a strong linear correlation with position ($R^2=0.89$) and clear gradient in pKa across the regression plane ($\rho = 0.92$). (B) In the case of the $\Delta G_{\pm H}$ projections for the $M_{pK_a, \Delta G}^{512, 512}$ model, the data displays a clear gradient in $\Delta G_{\pm H}$ but the ordering within the latent space is non-linear. It was found that a cubic transform significantly improved R^2 in this example.

Figures A5a and A5b provide examples of this fitting procedure for the $M_{pK_a}^{512}$ and $M_{pK_a, \Delta G}^{512, 512}$ with respect to pKa and $\Delta G_{\pm H}$, respectively. As evidenced by Figure A5b, training does not always linearly organize the latent space, so various transformations are applied to the property data in an attempt to linearize it. The transforms are illustrated below, where f is a linear function of x and y .

$z = f(x, y)$	Linear
$z^2 = f(x, y)$	Square
$z^3 = f(x, y)$	Cubic
$\frac{1}{z^2} = f(x, y)$	Root
$\log(z) = f(x, y)$	Log
$z^{-1} = f(x, y)$	Inverse

In the case of the transformed data, the preceding expression becomes

$$\mathbf{a} = \mathbf{X}^+ \mathbf{z}_{Transformed}$$

R^2 is then calculated according to standard statistical procedure

$$SSE = \sum_{n=1}^{640} (z_i - \bar{z})^2$$

$$SSR = \sum_{n=1}^{640} (\hat{z}_i - \bar{z})^2$$

$$R^2 = \frac{SSR}{SSE}$$

Where z_i represents the true property value, \bar{z} represents the average value, and \hat{z}_i represents the regressed value of z

Spearman Correlation (ρ) Calculation: ρ is typically calculated for rank-ordered bivariate systems to provide a measure of monotonic behavior. In order to reduce the three-dimensional principal component space (composed of the two principal components and the property value) into one suitable for bivariate analysis, we treat the direction of the gradient, represented by the unit vector $\hat{\mathbf{u}}$, as a variable that is optimized with respect to ρ . In the case of perfect ordering, the gradient should vary monotonically along this direction. The angle between this directional vector and the x-axis is calculated, and the data is transformed such that the gradient direction becomes collinear with the x-axis (principal component 1).

$$\theta = \arccos(\hat{\mathbf{u}} \cdot \hat{\mathbf{e}}_1)$$

$$\mathbf{R}_\theta = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix}$$

$$\mathbf{X}\mathbf{R}_\theta = \mathbf{X}'$$

We then rank the points according to position along the x-axis, such that elements of X_R contain the rank of the corresponding entry in X' , and consider the bivariate problem of how the rank of the property value varies with increasing rank along the x-axis. ρ is then calculated as follows

$$x_R = \text{ranked}(X') \quad z_R = \text{ranked}(z)$$

$$\text{Cov}(x_R, z_R) = \overline{(x_R - \widehat{x_R})(z_R - \widehat{z_R})}$$

$$\rho = \frac{\text{Cov}(x_R, z_R)}{\sigma(x_R)\sigma(z_R)}$$

We iteratively solve for the gradient direction using the Nelder-Mead algorithm as implemented in the SciPy package to determine the direction which maximizes the value of ρ .

APPENDIX B. SUPPORTING INFORMATION FOR: SIMPLER IS BETTER: HOW LINEAR PREDICTION TASKS IMPROVE TRANSFER LEARNING IN CHEMICAL AUTOENCODERS

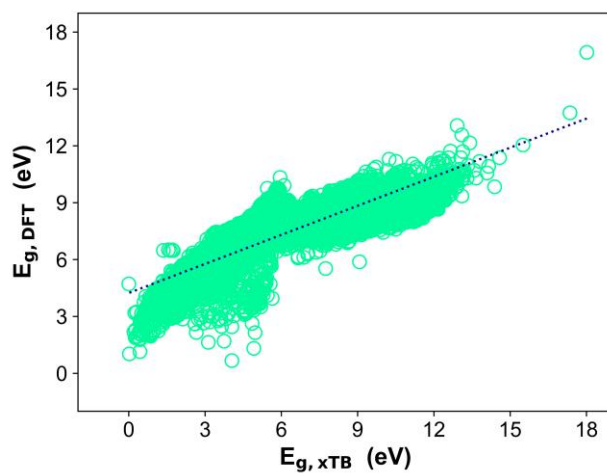


Figure B1: Correlation plot between bandgap calculated at the DFT and xTB levels of theory ($R^2=0.78$). $E_{g,xTB}$ systematically overestimates the bandgap compared with $E_{g,DFT}$. All values are positive. Utilizing a linear regression on all available DFT and xTB training data results in a MAE in predicting the DFT bandgap of the test set of 0.47 eV.

Model Architecture

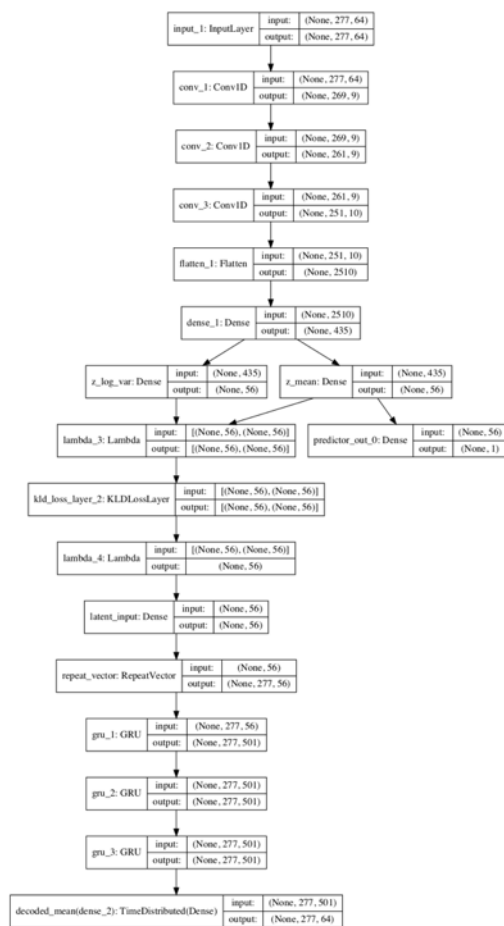


Figure B2. Variational autoencoder architecture utilized for property prediction and latent space analysis as implemented in Keras 2.2.4 with Tensorflow 1.14.0 backend.

Multi-property Prediction PCA Plots

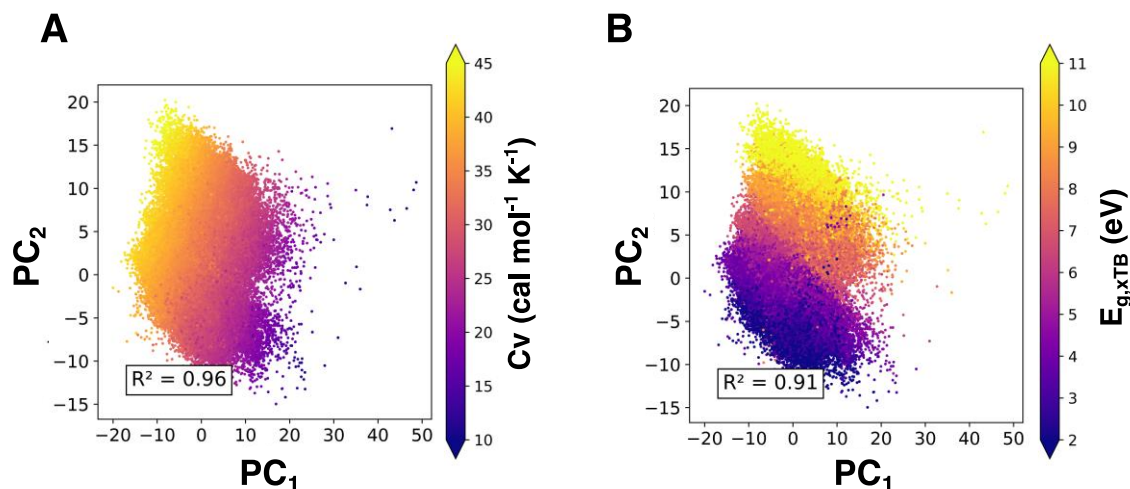


Figure B3: Principal component analysis performed on latent encodings for entire QM9 dataset for a model trained on both xTB data and heat capacity with a predictor network consisting of a single linear node. Projections are colored according to (A) C_v and (B) $E_{g,xTB}$. In contrast to the more complex predictor network utilized in Figure B4, the linear network is capable of organizing multiple properties linearly and along orthogonal directions within the latent space

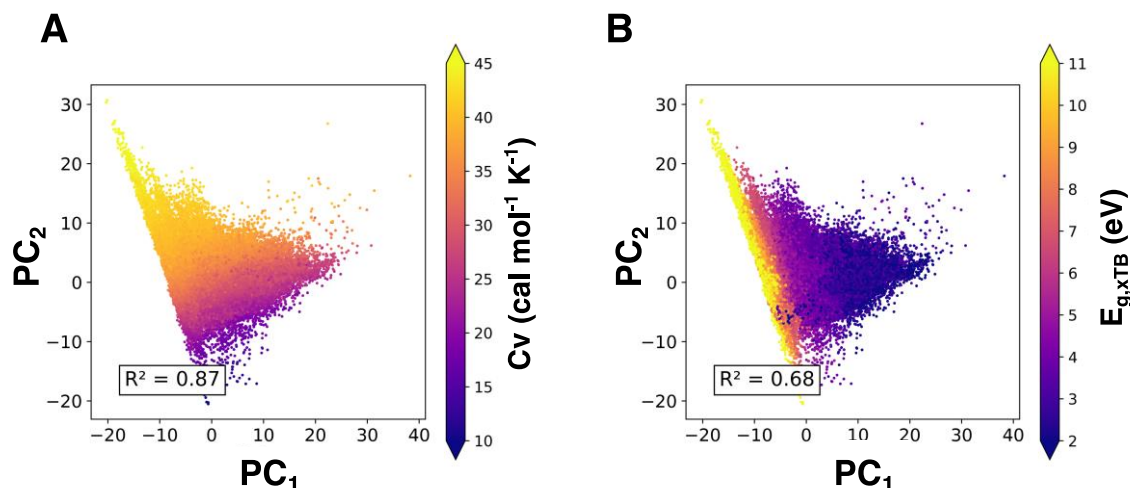


Figure B4: Principal component analysis performed on latent encodings for entire QM9 dataset for a model trained on both xTB data and heat capacity with a predictor network consisting of 3x64 nodes. While the latent space is well organized in a linear fashion according to C_v , the organization with respect to $E_{g,xTB}$ is poorly organized. Rather than varying linearly, the projections suddenly ramp up to very high $E_{g,xTB}$ structures.

APPENDIX C. SUPPORTING INFORMATION FOR: IMPROVING THE GENERATIVE PERFORMANCE OF CHEMICAL AUTOENCODERS THROUGH TRANSFER LEARNING

DFT and GFN2-xTB Comparison

The DFT and GFN2-xTB (xTB) predictions for E_g on the training set of compounds is presented in Figure C1A. xTB significantly underpredicts the bandgap in comparison with DFT with a bias towards predictions of ~ 4 eV. To correct for this discrepancy, a random forest (RF) model was trained to predict the difference between E_g computed at the xTB and DFT level using the Morgan fingerprint of each compound as a feature. RF test set predictions are presented in Figure C1B.

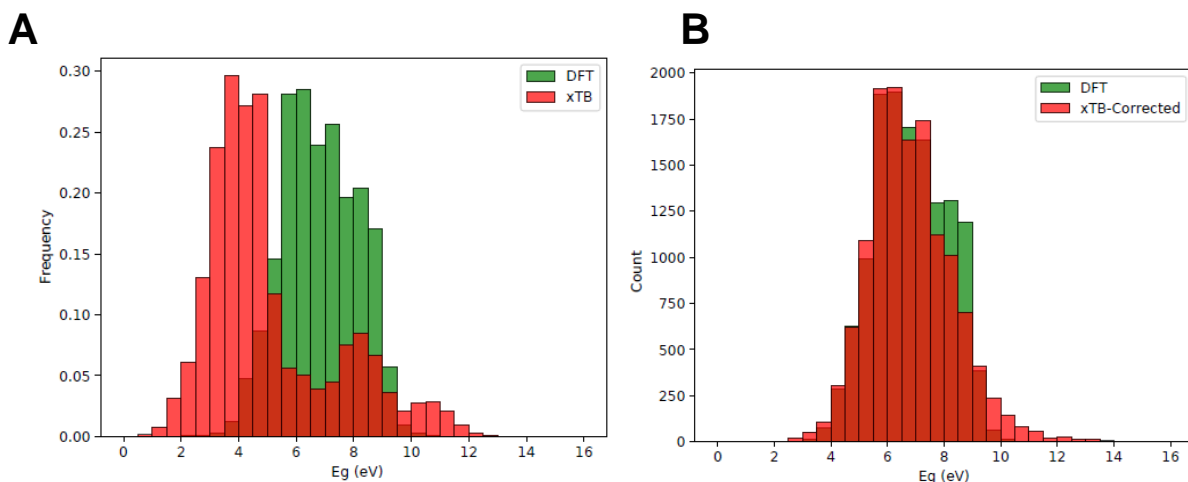


Figure C1: (A) E_g distributions for structures within the training set calculated at the xTB and DFT levels of theory. (B) structures within the test set calculated at the DFT level of theory and xTB level after correction with the ancillary random forest model.

Figure C2 shows a comparison of xTB and DFT values for U_0 on the training set of compounds. Although differing in absolute magnitude, the xTB calculated U_0 is a linear correlate of the DFT value and is used in the main text without modification for characterizing generated compounds.

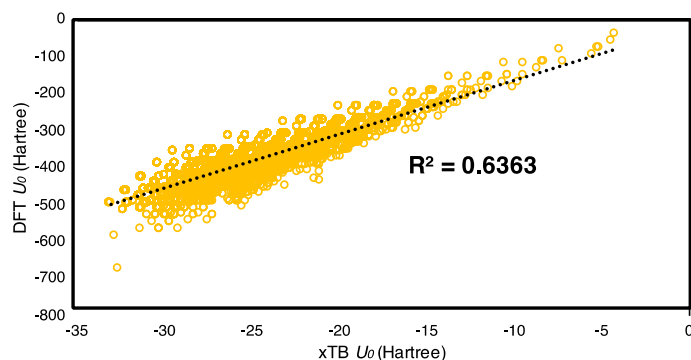


Figure C2: Correlation plot between U_0 calculated at the xTB and DFT levels of theory for compounds in the training set. The xTB value is linearly correlated with the DFT value ($R^2 = 0.64$).

2D Latent Space Organization

Figure C3 shows the 2D latent space for the model trained solely on reconstruction, where the structures have been colored according to aromaticity. The compression leads to isolation of encodings for aromatic compounds in the top left of the plot.

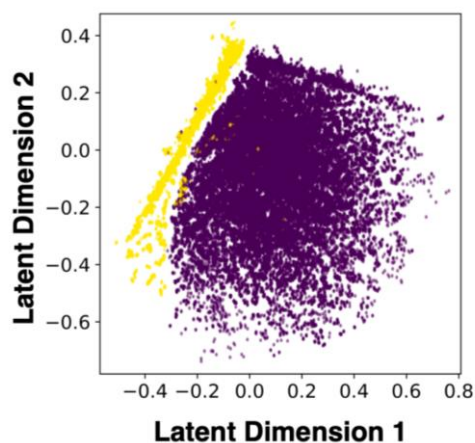


Figure C3: Latent space of 2D autoencoder trained only on chemical reconstruction. Training compounds have been colored according to aromaticity. Purple corresponds to non-aromatic structures, while yellow corresponds to aromatic structures. Even without a property prediction task, compounds have been organized according to similar structure, with nearly all aromatic compounds found in a narrow strip to the leftmost side of the latent space, offset from the rest of the non-aromatic compounds.

Effect of Latent Space Dimensionality on Validity and Multi-Task Training

The dimensionality of the latent space determines the level of data compression performed by the autoencoder and correspondingly the amount of chemical property data that can be accommodated during training before incurring information loss. Models with latent space dimensionality ranging from 2 to 128 were trained on reconstruction and benchmarked for validity (Fig. C4). Increasing the dimensionality of the latent space increases the proportion of valid structures but quickly levels off. A similar effect is noted for reconstruction accuracy but is much more pronounced. Below a certain dimensionality, there is no chance for an accurate reconstruction. Above this threshold, the increase in reconstruction accuracy quickly plateaus. The original choice of a 56D latent space thus appears justified, as further increases do not improve the capability of the model.

The dimensionality of the latent space likewise impacts that number of property prediction tasks

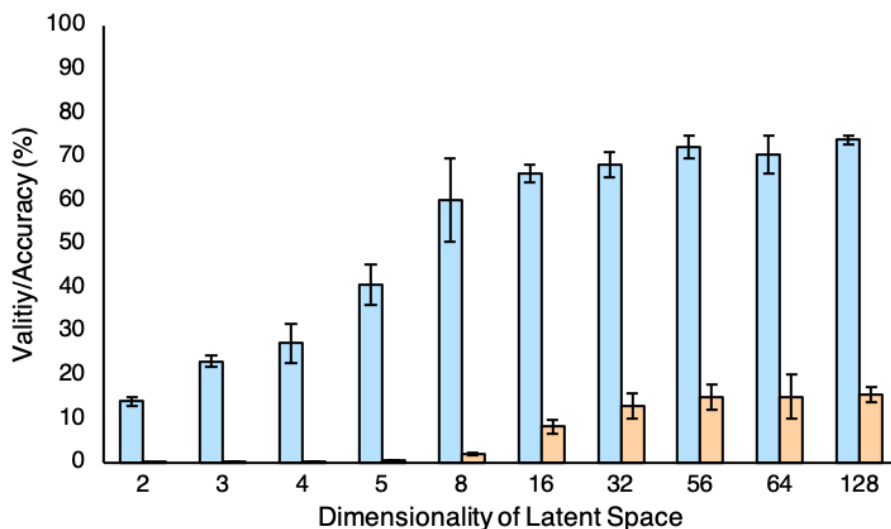


Figure C4: Analysis of validity (blue) and reconstruction accuracy (orange) as a function of latent space dimensionality. 5,000 structures are randomly selected from the test set and encoded into the latent space. These encodings are then decoded 100 times and tested for validity (blue) or equality against the input structure (orange). This process is repeated for 10 models each evaluated on a different random 5,000 structure subset of the test set. The results are averaged across 5,000,000 decodings for each dimensionality

that can simultaneously be trained while maintaining prediction accuracy. Models with latent space dimensionality ranging from 2 to 56 were trained on reconstruction and between 1-5 property prediction tasks, then benchmarked for property prediction accuracy (Fig. C5). We observe a

straightforward trend such that an N-dimensional latent space is capable of effectively training on N-unrelated property prediction tasks, above which model performance degrades significantly.

Effect of Multi-Property Training in Well-Represented Generative Tasks

In the main text we present results showing that multi-task TL increases the number of low E_g structures that can be generated. We have also compared the generative performance of models

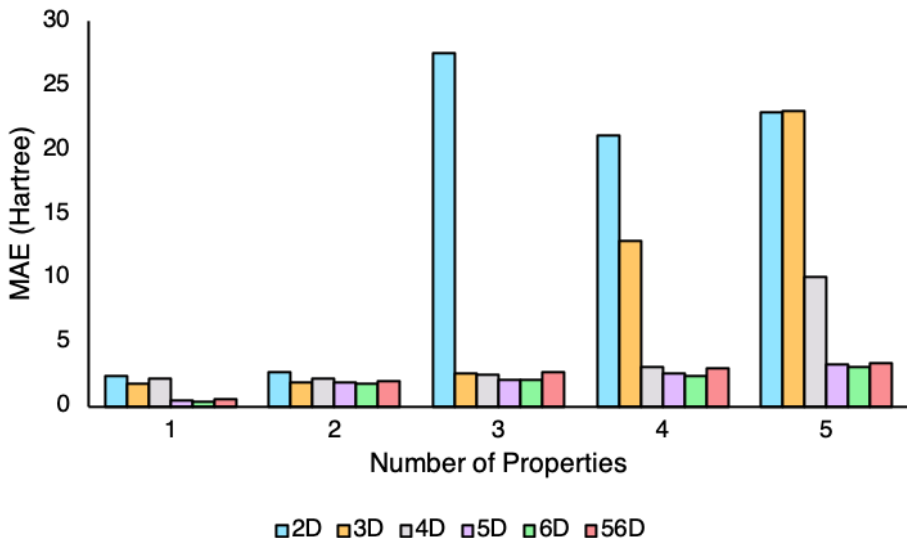


Figure C5: The dimensionality of the latent space provides a hard limit to the number of property prediction tasks that can be effectively trained. Up to N properties may be encoded into an N dimensional latent space without impacting prediction accuracy, although very small dimensionalities ($N < 5$) may perform worse in general on property prediction tasks. Once the number of properties exceeds the dimensionality of the latent space, it results in a massive penalty to predictive accuracy. MAE is presented with respect to U_0 , which all models shared in common as a prediction task.

trained on E_g vs. $U_0/ZPVE/E_g$ for generating structures in the E_g ranges of 5.5-6.0 eV and 9.5-10.0 eV (Figure C6). In these cases, the E_g only model outperforms the multi-task models with U_0 extrapolation, which reflects that these E_g ranges are already well represented in the training data.

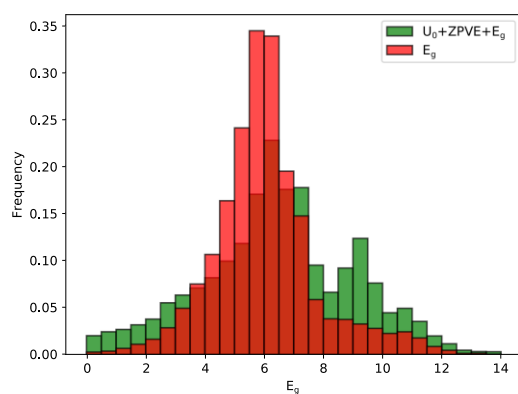
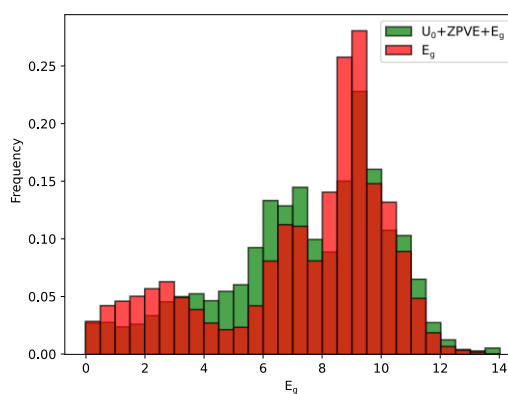
A**B**

Figure C6: Distributions of E_g for structures generated by targeting (A) 5.5-6.0 eV and (B) 9.5-10.0 eV. Models were trained to predict E_g only (red bars) or E_g in combination with $ZPVE$ and U_0 . For these targeting regimes, the model trained on E_g alone is already capable of generating the desired compounds, and the inclusion of additional properties does not provide further benefit.

Complete Model Training Details

The autoencoder accepts one-hot inputs of grammar parse trees to an encoder network comprising three one-dimensional convolutional layers with filter sizes 9, 9, and 10, respectively, and kernel sizes of 9, 9, and 11, respectively. The outputs from the convolutional layers are passed to a fully connected layer of 435 units, that are then separately connected to two fully connected layers of 56 units (i.e., the dimensionality of the latent space) defining the mean and log variance of the encoding distribution, respectively. The decoder accepts samples from the encoding distributions and passes them to a fully connected layer of 56 units that is connected to three gated recurrent units of 501 cells each before terminating in a final fully connected layer outputting probability distributions for the output sequence. ReLU activation functions were used for all units in the autoencoder. Property prediction is achieved by passing latent vectors to a single linear unit producing scalar output. An additional unit is included for each property. A diagram of the autoencoder architecture is provided in Figure C7. Models were created using Keras 2.2.4 with Tensorflow 1.14.0 backend. To obtain a useful compressed representation of chemistry, the shared autoencoder was first pre-trained on a reconstruction task. For this purpose, the SMILES strings corresponding to all training compounds were first converted into one-hot grammar parse trees and used as both inputs and labeled outputs for autoencoder pretraining. Pretraining was performed using the RMSprop algorithm with learning rate of 0.005 on the categorical cross entropy loss function for 100 epochs with batch size of 500. Validation loss was monitored every epoch and the learning rate was halved in the event of a plateau. The pretrained models were utilized as initializations for joint training on property prediction tasks, where in addition to encoding and decoding the model was tasked with predicting up to four chemical properties from the latent encodings. For each property, the latent chemical encoding is fed to a single linear unit producing the desired property as scalar output. Here, the learning rate was reduced to 0.001, and the loss weights assigned to the different tasks were adjusted. Loss corresponding to predictor MSE was scaled by 100, variational loss was scaled by 750, and the categorical cross entropy loss weight from encoding/decoding was set to 50 initially, before decaying to 1 according to a sigmoid function.

\

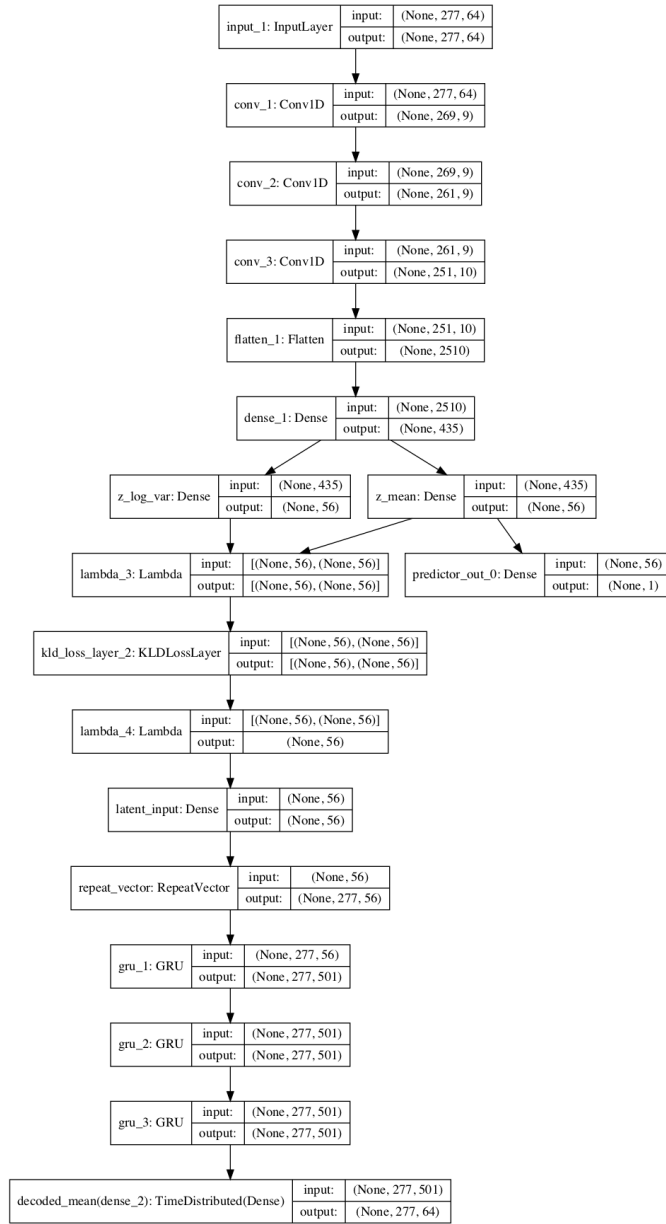


Figure C7: Variational autoencoder architecture utilized in this study as implemented in Keras 2.2.4 with Tensorflow 1.14.0 backend.

APPENDIX D. SUPPORTING INFORMATION FOR: ACTIVELY SEARCHING: INVERSE DESIGN OF NOVEL MOLECULES WITH SIMULTANEOUSLY OPTIMIZED PROPERTIES

Model Training Details

The implemented autoencoder accepts one-hot inputs of grammar parse trees to an encoder network comprising three one-dimensional convolutional layers with filter sizes 9, 9, and 10, and kernel sizes of 9, 9, and 11. The outputs from the convolutional layers are passed to a fully connected layer of 435 units, that are then separately connected to two fully connected layers of 56 units (i.e., the dimensionality of the latent space) defining the mean and log variance of the encoding distribution. The decoder accepts samples from the encoding distributions and passes them to a fully connected layer of 56 units that is connected to three gated recurrent units of 501 cells each before terminating in a final fully connected layer outputting probability distributions for the output sequence. ReLU activation functions were used for all units in the autoencoder. Property prediction is achieved by passing latent vectors to a single linear unit producing scalar output. An additional unit is included for each property. Models were created using Keras[127] 2.2.4 with Tensorflow[128] 1.14.0 backend.

It is difficult to train a network on both encoding/decoding and property prediction from scratch. It often proves much easier to first train the network on the encoding/decoding task first to obtain useful compressed chemical representations before transferring those weights for use in a joint training task. For this purpose, the SMILES strings corresponding to all training compounds in the original GVAE dataset were first converted into one-hot grammar parse trees and used as both inputs and labeled outputs for autoencoder pretraining. Pretraining followed the same routine used in the original GVAE implementation.[68] The pretrained model was then utilized as initialization for joint training on property prediction tasks, where in addition to encoding and decoding the model was tasked with predicting up to three chemical properties from the latent encodings. For each property, the latent chemical encoding is fed to a single linear unit producing the desired property as scalar output. Training was conducted using the RMSprop algorithm with a learning rate of 0.001, which was set to decay by a factor of 0.3 in the case of a plateau in the validation loss. In order to balance the performance across all tasks and to ensure stable training, the loss weights assigned to the different tasks were adjusted. Variational loss was scaled by 750,

and the categorical cross-entropy loss from encoding/decoding was set to 50 initially, before decaying to 1 according to a sigmoid function. The MSE losses for the property predictors were not scaled, but all properties were normalized to fall in the range -20 to 20.

Multi-Target Active Learning: Extrapolation

In the main text, we report the results of an active learning study for generating structures exhibiting properties in the extrapolative regime ($VIP > 10.0\text{eV}$, $EA \in [1.5, 4.0\text{eV}]$, and $DM \in [4, 5\text{D}]$) with respect to the training data. After validation of the sampled structures at the $\omega\text{B97X-D3/def2-TZVP}$ level, 5 structures were found that exhibited properties within all three targeted ranges, and 22 structures were found that exhibited properties within $\pm 20\%$ of all targeted property ranges (**Fig. D1**).

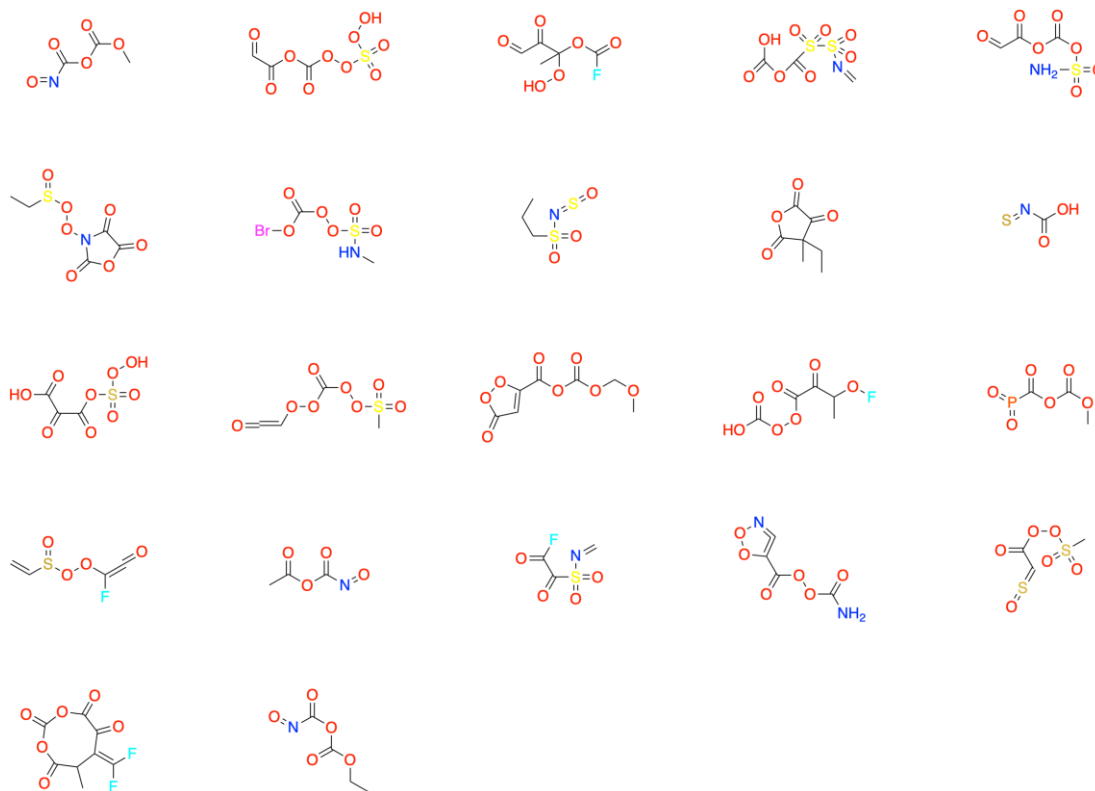


Figure D1. Final 22 structures for the extrapolative study reported in the main text simultaneously achieving property targets within $\pm 20\%$ of the targeted ranges for vertical ionization potential, electron affinity, and dipole moment. All properties were validated by DFT calculations at $\omega\text{B97X-D3/def2-TZVP}$ level. Only the 5 compounds reported in the main text fall strictly within the desired range, but we can still observe many of the same moieties and structural arrangements in this expanded set, including the ubiquity of oxygen, the lack of symmetric structures, and low prevalence of rings.

Multi-Target Active Learning: Interpolation

Figure D2 presents results for a case study where the targeted property ranges are in the interpolative regime with respect to the training data ($VIP \in [6.0, 7.0 \text{ eV}]$, $EA \in [0.5, 1.0 \text{ eV}]$, and $DM \in [4, 5 \text{ D}]$), but still relatively poorly represented. After 6 iterations of the active learning-based retraining, 1599 structures were discovered that satisfy the targeted property ranges at the xTB level, and of these 16 match all property ranges after validation at the ω B97X-D3/def2-TZVP level (**Fig. D3**). Here, the lower targeted VIP and EA ranges result in nitrogen-rich structures, compared with the extrapolative sampling case study.

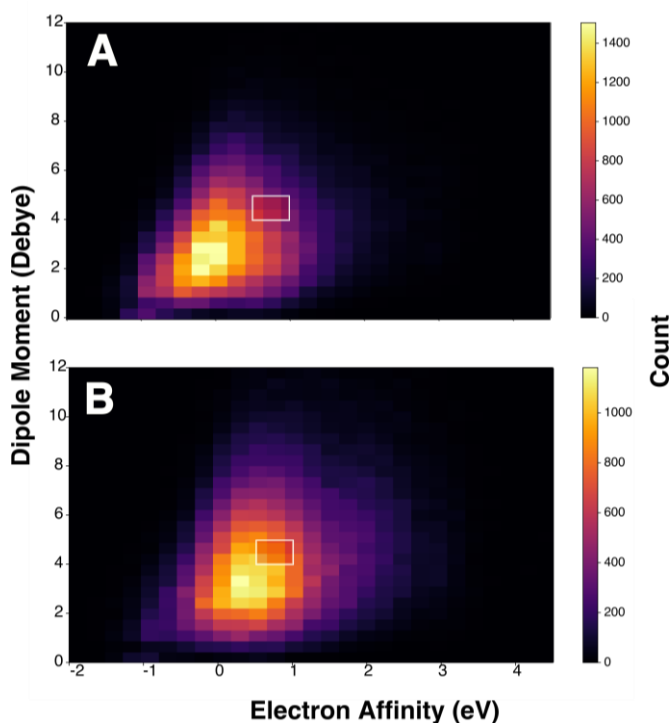


Figure D2. 2D property histogram for a model iteratively trained to predict VIP, EA, and DM, and tasked with targeted structure generation for these properties. VIP, EA, and DM targets are 6.0 to 7.0 eV, 0.5 to 1.0 eV, and 4 to 5 Debye, respectively. For visualization, only compounds with VIP between 6-7 are considered. The targeted region, with DM between 4-5 Debye and EA between 0.5-1.0 eV, is indicated with a box. All property ranges have some representation in the training data. Initially, (A) the model is capable of suggesting 811 compounds that display the three property targets simultaneously. After 6 iterations of the active learning procedure (B), the property distribution of proposed structures has shifted closer to the targeted region and the specificity of the model has doubled, with 1599 of the proposed structures displaying all three property targets.

REFERENCES

- [1] “#Envision2030: 17 goals to transform the world for persons with disabilities | United Nations Enable,” Feb. 09, 2016.
<https://www.un.org/development/desa/disabilities/envision2030.html> (accessed Jun. 07, 2021).
- [2] P. Kirkpatrick and C. Ellis, “Chemical space,” *Nature*, vol. 432, p. 823, Dec. 2004, doi: 10.1038/432823a.
- [3] L. M. Manojlović, “Photometry-based estimation of the total number of stars in the Universe,” *Appl. Opt.*, vol. 54, no. 21, pp. 6589–6591, Jul. 2015, doi: 10.1364/AO.54.006589.
- [4] D. C. Elton, Z. Boukouvalas, M. D. Fuge, and P. W. Chung, “Deep learning for molecular design—a review of the state of the art,” *Mol. Syst. Des. Eng.*, vol. 4, no. 4, pp. 828–849, 2019, doi: 10.1039/C9ME00039A.
- [5] A.-D. Gorse, “Diversity in medicinal chemistry space,” *Curr. Top. Med. Chem.*, vol. 6, no. 1, pp. 3–18, 2006, doi: 10.2174/156802606775193310.
- [6] J. A. DiMasi, H. G. Grabowski, and R. W. Hansen, “Innovation in the pharmaceutical industry: New estimates of R&D costs,” *J. Health Econ.*, vol. 47, pp. 20–33, May 2016, doi: 10.1016/j.jhealeco.2016.01.012.
- [7] S. M. Paul *et al.*, “How to improve R&D productivity: the pharmaceutical industry’s grand challenge,” *Nat. Rev. Drug Discov.*, vol. 9, no. 3, Art. no. 3, Mar. 2010, doi: 10.1038/nrd3078.
- [8] D. P. Tabor *et al.*, “Accelerating the discovery of materials for clean energy in the era of smart automation,” *Nat. Rev. Mater.*, vol. 3, no. 5, pp. 5–20, May 2018, doi: 10.1038/s41578-018-0005-z.
- [9] R. Macarron *et al.*, “Impact of high-throughput screening in biomedical research,” *Nat. Rev. Drug Discov.*, vol. 10, no. 3, Art. no. 3, Mar. 2011, doi: 10.1038/nrd3368.
- [10] B. Sanchez-Lengeling and A. Aspuru-Guzik, “Inverse molecular design using machine learning: Generative models for matter engineering,” *Science*, vol. 361, no. 6400, pp. 360–365, Jul. 2018, doi: 10.1126/science.aat2663.
- [11] 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.*, vol. 45, no. 1, pp. 195–216, 2015, doi: 10.1146/annurev-matsci-070214-020823.

- [12] D. Schwalbe-Koda and R. Gómez-Bombarelli, “Generative Models for Automatic Chemical Design,” in *Machine Learning Meets Quantum Physics*, K. T. Schütt, S. Chmiela, O. A. von Lilienfeld, A. Tkatchenko, K. Tsuda, and K.-R. Müller, Eds. Cham: Springer International Publishing, 2020, pp. 445–467. doi: 10.1007/978-3-030-40245-7_21.
- [13] C. Hansch, P. P. Maloney, T. Fujita, and R. M. Muir, “Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients,” *Nature*, vol. 194, no. 4824, Art. no. 4824, Apr. 1962, doi: 10.1038/194178b0.
- [14] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.
- [15] W. Pitts, “How we know universals the perception of auditory and visual forms,” p. 21.
- [16] A. L. Samuel, “Some Studies in Machine Learning Using the Game of Checkers,” *IBM J. Res. Dev.*, vol. 3, no. 3, pp. 210–229, Jul. 1959, doi: 10.1147/rd.33.0210.
- [17] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.
- [18] F. Rosenblatt, *Principles of neurodynamics; perceptrons and the theory of brain mechanisms*. Washington, 1962. [Online]. Available: <http://hdl.handle.net/2027/mdp.39015039846566>
- [19] M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 2017.
- [20] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU),” *ArXiv180308375 Cs Stat*, Feb. 2019, Accessed: Jun. 07, 2021. [Online]. Available: <http://arxiv.org/abs/1803.08375>
- [21] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989, doi: 10.1007/BF02551274.
- [22] “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks,” *Neural Netw.*, vol. 3, no. 5, pp. 551–560, Jan. 1990, doi: 10.1016/0893-6080(90)90005-6.
- [23] L. N. Kanal, “Perceptron,” in *Encyclopedia of Computer Science*, 2003, pp. 1383–1385.
- [24] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

- [25] R. Ge, F. Huang, C. Jin, and Y. Yuan, “Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition,” in *Conference on Learning Theory*, Jun. 2015, pp. 797–842. Accessed: Jun. 03, 2021. [Online]. Available: <http://proceedings.mlr.press/v40/Ge15.html>
- [26] S. Ruder, “An overview of gradient descent optimization algorithms,” *ArXiv160904747 Cs*, Jun. 2017, Accessed: Jun. 04, 2021. [Online]. Available: <http://arxiv.org/abs/1609.04747>
- [27] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *ArXiv14126980 Cs*, Dec. 2014, Accessed: Jul. 12, 2018. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [28] E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa, “A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks,” in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, Dec. 2018, pp. 92–99. doi: 10.1109/CTEMS.2018.8769211.
- [29] D. Cireşan, U. Meier, and J. Schmidhuber, “Multi-column Deep Neural Networks for Image Classification,” *ArXiv12022745 Cs*, Feb. 2012, Accessed: Nov. 18, 2019. [Online]. Available: <http://arxiv.org/abs/1202.2745>
- [30] D. Weininger, “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules,” *J. Chem. Inf. Comput. Sci.*, vol. 28, no. 1, pp. 31–36, Feb. 1988, doi: 10.1021/ci00057a005.
- [31] D. Rogers and M. Hahn, “Extended-Connectivity Fingerprints,” *J. Chem. Inf. Model.*, vol. 50, no. 5, pp. 742–754, May 2010, doi: 10.1021/ci100050t.
- [32] D. Duvenaud *et al.*, “Convolutional Networks on Graphs for Learning Molecular Fingerprints,” *arXiv:1509.09292*, Sep. 2015, Accessed: Jul. 11, 2018. [Online]. Available: <http://arxiv.org/abs/1509.09292>
- [33] T.-S. Lin *et al.*, “BigSMILES: A Structurally-Based Line Notation for Describing Macromolecules,” *ACS Cent. Sci.*, vol. 5, no. 9, pp. 1523–1531, Sep. 2019, doi: 10.1021/acscentsci.9b00476.
- [34] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, “SELFIES: a robust representation of semantically constrained graphs with an example application in chemistry,” *ArXiv190513741 Phys. Physicsquant-Ph Stat*, May 2019, Accessed: Jun. 10, 2019. [Online]. Available: <http://arxiv.org/abs/1905.13741>
- [35] A. Capecchi, D. Probst, and J.-L. Reymond, “One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome,” *J. Cheminformatics*, vol. 12, no. 1, p. 43, Jun. 2020, doi: 10.1186/s13321-020-00445-4.

- [36] G. B. Goh, N. O. Hodas, and A. Vishnu, “Deep learning for computational chemistry,” *J. Comput. Chem.*, vol. 38, no. 16, pp. 1291–1307, 2017, doi: <https://doi.org/10.1002/jcc.24764>.
- [37] L. F. Fieser, M. G. Ettlinger, and G. Fawaz, “Naphthoquinone antimalarials; distribution between organic solvents and aqueous buffers,” *J. Am. Chem. Soc.*, vol. 70, no. 10, pp. 3228–3232, Oct. 1948, doi: 10.1021/ja01190a008.
- [38] R. Burbidge, M. Trotter, B. Buxton, and S. Holden, “Drug design by machine learning: support vector machines for pharmaceutical data analysis,” *Comput. Chem.*, vol. 26, no. 1, pp. 5–14, Dec. 2001, doi: 10.1016/S0097-8485(01)00094-8.
- [39] Ajay, W. P. Walters, and M. A. Murcko, “Can We Learn To Distinguish between ‘Drug-like’ and ‘Nondrug-like’ Molecules?,” *J. Med. Chem.*, vol. 41, no. 18, pp. 3314–3324, Aug. 1998, doi: 10.1021/jm970666c.
- [40] M. Shen, C. Béguin, A. Golbraikh, J. P. Stables, H. Kohn, and A. Tropsha, “Application of Predictive QSAR Models to Database Mining: Identification and Experimental Validation of Novel Anticonvulsant Compounds,” *J. Med. Chem.*, vol. 47, no. 9, pp. 2356–2364, Apr. 2004, doi: 10.1021/jm030584q.
- [41] A. Lusci, G. Pollastri, and P. Baldi, “Deep Architectures and Deep Learning in Chemoinformatics: The Prediction of Aqueous Solubility for Drug-Like Molecules,” *J. Chem. Inf. Model.*, vol. 53, no. 7, pp. 1563–1575, Jul. 2013, doi: 10.1021/ci400187y.
- [42] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, “DeepTox: Toxicity Prediction using Deep Learning,” *Front. Environ. Sci.*, vol. 3, 2016, doi: 10.3389/fenvs.2015.00080.
- [43] T. B. Hughes, G. P. Miller, and S. J. Swamidass, “Modeling Epoxidation of Drug-like Molecules with a Deep Machine Learning Network,” *ACS Cent. Sci.*, vol. 1, no. 4, pp. 168–180, Jul. 2015, doi: 10.1021/acscentsci.5b00131.
- [44] Y. Xu, Z. Dai, F. Chen, S. Gao, J. Pei, and L. Lai, “Deep Learning for Drug-Induced Liver Injury,” *J. Chem. Inf. Model.*, vol. 55, no. 10, pp. 2085–2093, Oct. 2015, doi: 10.1021/acs.jcim.5b00238.
- [45] J. Y. Ryu, H. U. Kim, and S. Y. Lee, “Deep learning improves prediction of drug–drug and drug–food interactions,” *Proc. Natl. Acad. Sci.*, vol. 115, no. 18, pp. E4304–E4311, May 2018, doi: 10.1073/pnas.1803294115.
- [46] M. Korshunova, B. Ginsburg, A. Tropsha, and O. Isayev, “OpenChem: A Deep Learning Toolkit for Computational Chemistry and Drug Design,” *J. Chem. Inf. Model.*, vol. 61, no. 1, pp. 7–13, Jan. 2021, doi: 10.1021/acs.jcim.0c00971.
- [47] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, “Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning,” *Phys. Rev. Lett.*, vol. 108, no. 5, p. 058301, Jan. 2012, doi: 10.1103/PhysRevLett.108.058301.

- [48] Z. Ahmad, T. Xie, C. Maheshwari, J. C. Grossman, and V. Viswanathan, “Machine Learning Enabled Computational Screening of Inorganic Solid Electrolytes for Suppression of Dendrite Formation in Lithium Metal Anodes,” *ACS Cent. Sci.*, vol. 4, no. 8, pp. 996–1006, Aug. 2018, doi: 10.1021/acscentsci.8b00229.
- [49] F. A. Faber *et al.*, “Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error,” *J. Chem. Theory Comput.*, vol. 13, no. 11, pp. 5255–5264, Nov. 2017, doi: 10.1021/acs.jctc.7b00577.
- [50] R. Gómez-Bombarelli *et al.*, “Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach,” *Nat. Mater.*, vol. 15, no. 10, pp. 1120–1127, Oct. 2016, doi: 10.1038/nmat4717.
- [51] Z. Li, S. Wang, W. S. Chin, L. E. Achenie, and H. Xin, “High-throughput screening of bimetallic catalysts enabled by machine learning,” *J. Mater. Chem. A*, vol. 5, no. 46, pp. 24131–24138, Nov. 2017, doi: 10.1039/C7TA01812F.
- [52] C. Pabo, “Molecular technology: Designing proteins and peptides,” *Nature*, vol. 301, no. 5897, Art. no. 5897, Jan. 1983, doi: 10.1038/301200a0.
- [53] B. Gutte, M. Däumigen, and E. Wittschieber, “Design, synthesis and characterisation of a 34-residue polypeptide that interacts with nucleic acids,” *Nature*, vol. 281, no. 5733, Art. no. 5733, Oct. 1979, doi: 10.1038/281650a0.
- [54] K. Yue and K. A. Dill, “Inverse protein folding problem: designing polymer sequences,” *Proc. Natl. Acad. Sci.*, vol. 89, no. 9, pp. 4163–4167, May 1992, doi: 10.1073/pnas.89.9.4163.
- [55] C. Kuhn and D. N. Beratan, “Inverse Strategies for Molecular Design,” *J. Phys. Chem.*, vol. 100, no. 25, pp. 10595–10599, Jan. 1996, doi: 10.1021/jp960518i.
- [56] M. Wang, X. Hu, D. N. Beratan, and W. Yang, “Designing Molecules by Optimizing Potentials,” *J. Am. Chem. Soc.*, vol. 128, no. 10, pp. 3228–3232, Mar. 2006, doi: 10.1021/ja0572046.
- [57] D. Whitley, “A genetic algorithm tutorial,” *Stat. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994, doi: 10.1007/BF00175354.
- [58] M. F. Schubert, F. W. Mont, S. Chhajed, D. J. Poxson, J. K. Kim, and E. F. Schubert, “Design of multilayer antireflection coatings made from co-sputtered and low-refractive-index materials by genetic algorithm,” *Opt. Express*, vol. 16, no. 8, pp. 5290–5298, Apr. 2008, doi: 10.1364/OE.16.005290.
- [59] U. Rodemerck, M. Baerns, M. Holena, and D. Wolf, “Application of a genetic algorithm and a neural network for the discovery and optimization of new solid catalytic materials,” *Appl. Surf. Sci.*, vol. 223, no. 1, pp. 168–174, Feb. 2004, doi: 10.1016/S0169-4332(03)00919-X.

- [60] R. L. Riche and R. T. Haftka, "Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm," *AIAA J.*, vol. 31, no. 5, pp. 951–956, 1993, doi: 10.2514/3.11710.
- [61] B. Chambers and A. Tennant, "Optimised design of Jaumann radar absorbing materials using a genetic algorithm," *IEE Proc. - Radar Sonar Navig.*, vol. 143, no. 1, pp. 23–30, Feb. 1996, doi: 10.1049/ip-rsn:19960316.
- [62] C. Kim, R. Batra, L. Chen, H. Tran, and R. Ramprasad, "Polymer design using genetic algorithm and machine learning," *Comput. Mater. Sci.*, vol. 186, p. 110067, Jan. 2021, doi: 10.1016/j.commatsci.2020.110067.
- [63] R. Winter, F. Montanari, F. Noé, and D.-A. Clevert, "Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations," *Chem. Sci.*, 2019, doi: 10.1039/C8SC04175J.
- [64] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, doi: 10.1109/TPAMI.2013.50.
- [65] D. P. Kingma and M. Welling, "An Introduction to Variational Autoencoders," *Found. Trends® Mach. Learn.*, vol. 12, no. 4, pp. 307–392, Nov. 2019, doi: 10.1561/22000000056.
- [66] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *ArXiv13126114 Cs Stat*, Dec. 2013, Accessed: Apr. 29, 2019. [Online]. Available: <http://arxiv.org/abs/1312.6114>
- [67] R. Gómez-Bombarelli *et al.*, "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules," *ACS Cent. Sci.*, vol. 4, no. 2, pp. 268–276, Feb. 2018, doi: 10.1021/acscentsci.7b00572.
- [68] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, "Grammar Variational Autoencoder," *arXiv:1703.01925*, Mar. 2017, Accessed: Jul. 11, 2018. [Online]. Available: <http://arxiv.org/abs/1703.01925>
- [69] W. Gao and C. W. Coley, "The Synthesizability of Molecules Proposed by Generative Models," *J. Chem. Inf. Model.*, Apr. 2020, doi: 10.1021/acs.jcim.0c00174.
- [70] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, May 2016, doi: 10.1186/s40537-016-0043-6.
- [71] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Big Data Meets Quantum Chemistry Approximations: The Δ -Machine Learning Approach," *J. Chem. Theory Comput.*, vol. 11, no. 5, pp. 2087–2096, May 2015, doi: 10.1021/acs.jctc.5b00099.
- [72] A. A. Kananenka, K. Yao, S. A. Corcelli, and J. L. Skinner, "Machine Learning for Vibrational Spectroscopic Maps," *J. Chem. Theory Comput.*, vol. 15, no. 12, pp. 6850–6858, Dec. 2019, doi: 10.1021/acs.jctc.9b00698.

- [73] P. A. Unzueta, C. S. Greenwell, and G. J. O. Beran, “Predicting Density Functional Theory-Quality Nuclear Magnetic Resonance Chemical Shifts via Δ -Machine Learning,” *J. Chem. Theory Comput.*, vol. 17, no. 2, pp. 826–840, Feb. 2021, doi: 10.1021/acs.jctc.0c00979.
- [74] P. O. Dral, A. Owens, A. Dral, and G. Csányi, “Hierarchical machine learning of potential energy surfaces,” *J. Chem. Phys.*, vol. 152, no. 20, p. 204110, May 2020, doi: 10.1063/5.0006498.
- [75] A. Nandi, C. Qu, P. L. Houston, R. Conte, and J. M. Bowman, “ Δ -machine learning for potential energy surfaces: A PIP approach to bring a DFT-based PES to CCSD(T) level of theory,” *J. Chem. Phys.*, vol. 154, no. 5, p. 051102, Feb. 2021, doi: 10.1063/5.0038301.
- [76] M. L. Hutchinson, E. Antono, B. M. Gibbons, S. Paradiso, J. Ling, and B. Meredig, “Overcoming data scarcity with transfer learning,” *arXiv:1711.05099*, Nov. 2017, Accessed: Jan. 27, 2019. [Online]. Available: <http://arxiv.org/abs/1711.05099>
- [77] N. C. Iovanac and B. M. Savoie, “Improving the generative performance of chemical autoencoders through transfer learning,” *Mach. Learn. Sci. Technol.*, vol. 1, no. 4, p. 045010, Oct. 2020, doi: 10.1088/2632-2153/abae75.
- [78] T. Unterthiner and A. Mayr, “Deep Learning as an Opportunity in Virtual Screening,” p. 9.
- [79] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, “Massively Multitask Networks for Drug Discovery,” *arXiv:1502.02072*, Feb. 2015, Accessed: Jan. 27, 2019. [Online]. Available: <http://arxiv.org/abs/1502.02072>
- [80] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, “Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships,” *J. Chem. Inf. Model.*, vol. 55, no. 2, pp. 263–274, Feb. 2015, doi: 10.1021/ci500747n.
- [81] P. B. Jørgensen, M. N. Schmidt, and O. Winther, “Deep Generative Models for Molecular Science,” *Mol. Inform.*, vol. 37, no. 1–2, p. 1700133, 2018, doi: 10.1002/minf.201700133.
- [82] A. Cherkasov *et al.*, “QSAR Modeling: Where Have You Been? Where Are You Going To?,” *J. Med. Chem.*, vol. 57, no. 12, pp. 4977–5010, Jun. 2014, doi: 10.1021/jm4004285.
- [83] K. Hansen *et al.*, “Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space,” *J. Phys. Chem. Lett.*, vol. 6, no. 12, pp. 2326–2331, Jun. 2015, doi: 10.1021/acs.jpcllett.5b00831.
- [84] A. P. Bartók, R. Kondor, and G. Csányi, “On representing chemical environments,” *Phys. Rev. B*, vol. 87, no. 18, p. 184115, May 2013, doi: 10.1103/PhysRevB.87.184115.
- [85] J. Behler, “Atom-centered symmetry functions for constructing high-dimensional neural network potentials,” *J. Chem. Phys.*, vol. 134, no. 7, p. 074106, Feb. 2011, doi: 10.1063/1.3553717.

- [86] B. Huang and O. A. von Lilienfeld, "Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity," *J. Chem. Phys.*, vol. 145, no. 16, p. 161102, Oct. 2016, doi: 10.1063/1.4964627.
- [87] 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.*, vol. 115, no. 16, pp. 1084–1093, 2015, doi: 10.1002/qua.24912.
- [88] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, "Machine learning for molecular and materials science," *Nature*, vol. 559, no. 7715, pp. 547–555, Jul. 2018, doi: 10.1038/s41586-018-0337-2.
- [89] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "SchNet – A deep learning architecture for molecules and materials," *J. Chem. Phys.*, vol. 148, no. 24, p. 241722, Mar. 2018, doi: 10.1063/1.5019779.
- [90] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nat. Commun.*, vol. 8, p. 13890, Jan. 2017, doi: 10.1038/ncomms13890.
- [91] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande, "Low Data Drug Discovery with One-Shot Learning," *ACS Cent. Sci.*, vol. 3, no. 4, pp. 283–293, Apr. 2017, doi: 10.1021/acscentsci.6b00367.
- [92] T. Blaschke, M. Olivecrona, O. Engkvist, J. Bajorath, and H. Chen, "Application of Generative Autoencoder in De Novo Molecular Design," *Mol. Inform.*, vol. 37, no. 1–2, p. 1700123, Jan. 2018, doi: 10.1002/minf.201700123.
- [93] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, "How to represent crystal structures for machine learning: Towards fast prediction of electronic properties," *Phys. Rev. B*, vol. 89, no. 20, p. 205118, May 2014, doi: 10.1103/PhysRevB.89.205118.
- [94] L. S. Whitmore, A. George, and C. M. Hudson, "Mapping chemical performance on molecular structures using locally interpretable explanations," *arXiv:1611.07443*, Nov. 2016, Accessed: Jan. 23, 2019. [Online]. Available: <http://arxiv.org/abs/1611.07443>
- [95] G. E. Dahl, N. Jaitly, and R. Salakhutdinov, "Multi-task Neural Networks for QSAR Predictions," *arXiv:1406.1231*, Jun. 2014, Accessed: Jan. 27, 2019. [Online]. Available: <http://arxiv.org/abs/1406.1231>
- [96] V. Yarov-Yarovoy, J. Schonbrun, and D. Baker, "Multipass membrane protein structure prediction using Rosetta," *Proteins Struct. Funct. Bioinforma.*, vol. 62, no. 4, pp. 1010–1025, 2006, doi: 10.1002/prot.20817.

- [97] S. Ovchinnikov, H. Park, D. E. Kim, F. DiMaio, and D. Baker, “Protein structure prediction using Rosetta in CASP12,” *Proteins Struct. Funct. Bioinforma.*, vol. 86, no. S1, pp. 113–121, 2018, doi: 10.1002/prot.25390.
- [98] C. A. Rohl, C. E. M. Strauss, K. M. S. Misura, and D. Baker, “Protein Structure Prediction Using Rosetta,” in *Methods in Enzymology*, vol. 383, Academic Press, 2004, pp. 66–93. doi: 10.1016/S0076-6879(04)83004-0.
- [99] M. Welborn, L. Cheng, and T. F. Miller III, “Transferability in Machine Learning for Electronic Structure via the Molecular Orbital Basis,” *J. Chem. Theory Comput.*, vol. 14, no. 9, pp. 4772–4779, Sep. 2018, doi: 10.1021/acs.jctc.8b00636.
- [100] J. Behler and M. Parrinello, “Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces,” *Phys. Rev. Lett.*, vol. 98, no. 14, p. 146401, Apr. 2007, doi: 10.1103/PhysRevLett.98.146401.
- [101] K. Yao, J. E. Herr, D. W. Toth, R. Mckintyre, and J. Parkhill, “The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics,” *Chem. Sci.*, vol. 9, no. 8, pp. 2261–2269, 2018, doi: 10.1039/C7SC04934J.
- [102] Y. Wang *et al.*, “Design principles for solid-state lithium superionic conductors,” *Nat. Mater.*, vol. 14, no. 10, pp. 1026–1031, Oct. 2015, doi: 10.1038/nmat4369.
- [103] P. B. Jørgensen *et al.*, “Machine learning-based screening of complex molecules for polymer solar cells,” *J. Chem. Phys.*, vol. 148, no. 24, p. 241735, Jun. 2018, doi: 10.1063/1.5023563.
- [104] J. J. de Pablo *et al.*, “New frontiers for the materials genome initiative,” *Npj Comput. Mater.*, vol. 5, no. 1, p. 41, Apr. 2019, doi: 10.1038/s41524-019-0173-4.
- [105] Z. Wu *et al.*, “MoleculeNet: a benchmark for molecular machine learning,” *Chem. Sci.*, vol. 9, no. 2, pp. 513–530, 2018, doi: 10.1039/C7SC02664A.
- [106] Z. Obermeyer and E. J. Emanuel, “Predicting the Future — Big Data, Machine Learning, and Clinical Medicine,” *N. Engl. J. Med.*, vol. 375, no. 13, pp. 1216–1219, Sep. 2016, doi: 10.1056/NEJMp1606181.
- [107] B. R. Goldsmith, J. Esterhuizen, J.-X. Liu, C. J. Bartel, and C. Sutton, “Machine learning for heterogeneous catalyst design and discovery,” *AIChE J.*, vol. 64, no. 7, pp. 2311–2323, 2018, doi: 10.1002/aic.16198.
- [108] T. Z. H. Gani, E. I. Ioannidis, and H. J. Kulik, “Computational Discovery of Hydrogen Bond Design Rules for Electrochemical Ion Separation,” *Chem. Mater.*, vol. 28, no. 17, pp. 6207–6218, Sep. 2016, doi: 10.1021/acs.chemmater.6b02378.
- [109] W. D. Richards, Y. Wang, L. J. Miara, J. Chul Kim, and G. Ceder, “Design of $\text{Li}_{1+2x}\text{Zn}_{1-x}\text{PS}_4$, a new lithium ion conductor,” *Energy Environ. Sci.*, vol. 9, no. 10, pp. 3272–3278, 2016, doi: 10.1039/C6EE02094A.

- [110] W. D. Richards *et al.*, “Design and synthesis of the superionic conductor Na₁₀SnP₂S₁₂,” *Nat. Commun.*, vol. 7, p. 11009, Mar. 2016, doi: 10.1038/ncomms11009.
- [111] S. Er, C. Suh, M. P. Marshak, and A. Aspuru-Guzik, “Computational design of molecules for an all-quinone redox flow battery,” *Chem. Sci.*, vol. 6, no. 2, pp. 885–893, 2015, doi: 10.1039/C4SC03030C.
- [112] N. S. Pagadala, K. Syed, and J. Tuszynski, “Software for molecular docking: a review,” *Biophys. Rev.*, vol. 9, no. 2, pp. 91–102, Apr. 2017, doi: 10.1007/s12551-016-0247-1.
- [113] D. Fourches, E. Muratov, and A. Tropsha, “Trust, But Verify: On the Importance of Chemical Structure Curation in Cheminformatics and QSAR Modeling Research,” *J. Chem. Inf. Model.*, vol. 50, no. 7, pp. 1189–1204, Jul. 2010, doi: 10.1021/ci100176x.
- [114] Andrew L. Ferguson, “Machine learning and data science in soft materials engineering,” *J. Phys. Condens. Matter*, vol. 30, no. 4, pp. 043002–043028, 2017.
- [115] M. E. Taylor, P. Stone, and Y. Liu, “Transfer Learning via Inter-Task Mappings for Temporal Difference Learning,” *J. Mach. Learn. Res.*, vol. 8, no. Sep, pp. 2125–2167, 2007.
- [116] T. Sterling and J. J. Irwin, “ZINC 15 – Ligand Discovery for Everyone,” *J. Chem. Inf. Model.*, vol. 55, no. 11, pp. 2324–2337, Nov. 2015, doi: 10.1021/acs.jcim.5b00559.
- [117] “RDKit: Open-source cheminformatics; <http://www.rdkit.org>, (accessed Feb 8, 2019).”, Accessed: Feb. 08, 2019. [Online]. Available: <http://www.rdkit.org>
- [118] Rumble John R., “Handbook of Chemistry and Physics 98th Edition,” “*Physical Constants of Organic Compounds*,” in *CRC Handbook of Chemistry and Physics, 98th Edition (Internet Version 2018)*, 2018.
http://hbcponline.com/faces/documents/05_25/05_25_0001.xhtml (accessed Apr. 25, 2018).
- [119] T. Matsui, Y. Shigeta, and K. Morihashi, “Assessment of Methodology and Chemical Group Dependences in the Calculation of the pKa for Several Chemical Groups,” *J. Chem. Theory Comput.*, vol. 13, no. 10, pp. 4791–4803, Oct. 2017, doi: 10.1021/acs.jctc.7b00587.
- [120] B. Thapa and H. B. Schlegel, “Theoretical Calculation of pKa’s of Selenols in Aqueous Solution Using an Implicit Solvation Model and Explicit Water Molecules,” *J. Phys. Chem. A*, vol. 120, no. 44, pp. 8916–8922, Nov. 2016, doi: 10.1021/acs.jpca.6b09520.
- [121] H. S. Yu, M. A. Watson, and A. D. Bochevarov, “Weighted Averaging Scheme and Local Atomic Descriptor for pKa Prediction Based on Density Functional Theory,” *J. Chem. Inf. Model.*, vol. 58, no. 2, pp. 271–286, Feb. 2018, doi: 10.1021/acs.jcim.7b00537.
- [122] A. K. Rappe, C. J. Casewit, K. S. Colwell, W. A. Goddard, and W. M. Skiff, “UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations,”

- J. Am. Chem. Soc.*, vol. 114, no. 25, pp. 10024–10035, Dec. 1992, doi: 10.1021/ja00051a040.
- [123] P. Pracht, C. A. Bauer, and S. Grimme, “Automated and efficient quantum chemical determination and energetic ranking of molecular protonation sites,” *J. Comput. Chem.*, vol. 38, no. 30, pp. 2618–2631, Nov. 2017, doi: 10.1002/jcc.24922.
- [124] S. Grimme, C. Bannwarth, and P. Shushkov, “A Robust and Accurate Tight-Binding Quantum Chemical Method for Structures, Vibrational Frequencies, and Noncovalent Interactions of Large Molecular Systems Parametrized for All spd-Block Elements (Z = 1–86),” *J. Chem. Theory Comput.*, vol. 13, no. 5, pp. 1989–2009, May 2017, doi: 10.1021/acs.jctc.7b00118.
- [125] F. Neese, “Software update: the ORCA program system, version 4.0,” *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, vol. 8, no. 1, p. e1327, 2018, doi: 10.1002/wcms.1327.
- [126] F. Jensen, *Introduction to Computational Chemistry*. John Wiley & Sons, 2017.
- [127] Chollet, F.K., “Keras,” <https://github.com/fchollet/keras>, 2015, [Online]. Available: <https://github.com/fchollet/keras>,
- [128] Abadi, M. , et al., “TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16),” pp. 265–283, 2016.
- [129] G. Montavon *et al.*, “Learning Invariant Representations of Molecules for Atomization Energy Prediction,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 440–448. [Online]. Available: <http://papers.nips.cc/paper/4830-learning-invariant-representations-of-molecules-for-atomization-energy-prediction.pdf>
- [130] L. C. Blum and J.-L. Reymond, “970 Million Druglike Small Molecules for Virtual Screening in the Chemical Universe Database GDB-13,” *J. Am. Chem. Soc.*, vol. 131, no. 25, pp. 8732–8733, Jul. 2009, doi: 10.1021/ja902302h.
- [131] R. Ramakrishnan, M. Hartmann, E. Tapavicza, and O. A. von Lilienfeld, “Electronic spectra from TDDFT and machine learning in chemical space,” *J. Chem. Phys.*, vol. 143, no. 8, p. 084111, Aug. 2015, doi: 10.1063/1.4928757.
- [132] G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny, “Computational Modeling of β -Secretase 1 (BACE-1) Inhibitors Using Ligand Based Approaches,” *J. Chem. Inf. Model.*, vol. 56, no. 10, pp. 1936–1949, Oct. 2016, doi: 10.1021/acs.jcim.6b00290.
- [133] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, “Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD),” *JOM*, vol. 65, no. 11, pp. 1501–1509, Nov. 2013, doi: 10.1007/s11837-013-0755-4.

- [134] M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen, “Molecular de-novo design through deep reinforcement learning,” *J. Cheminformatics*, vol. 9, no. 1, p. 48, Sep. 2017, doi: 10.1186/s13321-017-0235-x.
- [135] C. W. Coley, W. H. Green, and K. F. Jensen, “Machine Learning in Computer-Aided Synthesis Planning,” *Acc. Chem. Res.*, vol. 51, no. 5, pp. 1281–1289, May 2018, doi: 10.1021/acs.accounts.8b00087.
- [136] C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green, and K. F. Jensen, “Prediction of Organic Reaction Outcomes Using Machine Learning,” *ACS Cent. Sci.*, vol. 3, no. 5, pp. 434–443, May 2017, doi: 10.1021/acscentsci.7b00064.
- [137] M. H. S. Segler and M. P. Waller, “Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction,” *Chem. – Eur. J.*, vol. 23, no. 25, pp. 5966–5971, 2017, doi: 10.1002/chem.201605499.
- [138] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.
- [139] A. Jain *et al.*, “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation,” *APL Mater.*, vol. 1, no. 1, p. 011002, Jul. 2013, doi: 10.1063/1.4812323.
- [140] Y. Zhuo, A. Mansouri Tehrani, and J. Brgoch, “Predicting the Band Gaps of Inorganic Solids by Machine Learning,” *J. Phys. Chem. Lett.*, vol. 9, no. 7, pp. 1668–1673, Apr. 2018, doi: 10.1021/acs.jpcclett.8b00124.
- [141] “John R. Rumble, ed., CRC Handbook of Chemistry and Physics, 100th Edition (Internet Version 2019), CRC Press/Taylor & Francis, Boca Raton, FL.”
- [142] A. J. Lawson, J. Swienty-Busch, T. Géoui, and D. Evans, “The Making of Reaxys—Towards Unobstructed Access to Relevant Chemistry Information,” in *The Future of the History of Chemical Information*, vol. 1164, 0 vols., American Chemical Society, 2014, pp. 127–148. doi: 10.1021/bk-2014-1164.ch008.
- [143] D. Lowe, “Chemical reactions from US patents (1976-Sep2016).” Jun. 13, 2017. doi: 10.6084/m9.figshare.5104873.v1.
- [144] J. E. Blake and R. C. Dana, “CASREACT: More Than a Million Reactions,” *J Chem Inf Comput Sci*, vol. 30, no. 4, pp. 394–399, Nov. 1990, doi: 10.1021/ci00068a008.
- [145] E. D. Cubuk, A. D. Sendek, and E. J. Reed, “Screening billions of candidates for solid lithium-ion conductors: A transfer learning approach for small data,” *J. Chem. Phys.*, vol. 150, no. 21, p. 214701, Jun. 2019, doi: 10.1063/1.5093220.

- [146] J. S. Smith *et al.*, “Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning,” *Nat. Commun.*, vol. 10, no. 1, pp. 1–8, Jul. 2019, doi: 10.1038/s41467-019-10827-4.
- [147] R. S. Simões, V. G. Maltarollo, P. R. Oliveira, and K. M. Honório, “Transfer and Multi-task Learning in QSAR Modeling: Advances and Challenges,” *Front. Pharmacol.*, vol. 9, 2018, doi: 10.3389/fphar.2018.00074.
- [148] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.
- [149] N. C. Iovanac and B. M. Savoie, “Improved Chemical Prediction from Scarce Data Sets via Latent Space Enrichment,” *J. Phys. Chem. A*, vol. 123, no. 19, pp. 4295–4302, May 2019, doi: 10.1021/acs.jpca.9b01398.
- [150] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, “Quantum chemistry structures and properties of 134 kilo molecules,” *Sci. Data*, vol. 1, p. 140022, Aug. 2014, doi: 10.1038/sdata.2014.22.
- [151] C. Bannwarth, S. Ehlert, and S. Grimme, “GFN2-xTB—An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions,” *J. Chem. Theory Comput.*, vol. 15, no. 3, pp. 1652–1671, Mar. 2019, doi: 10.1021/acs.jctc.8b01176.
- [152] B. Sattarov, I. I. Baskin, D. Horvath, G. Marcou, E. J. Bjerrum, and A. Varnek, “De Novo Molecular Design by Combining Deep Autoencoder Recurrent Neural Networks with Generative Topographic Mapping,” *J. Chem. Inf. Model.*, Feb. 2019, doi: 10.1021/acs.jcim.8b00751.
- [153] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *ArXiv151106434 Cs*, Jan. 2016, Accessed: Mar. 26, 2020. [Online]. Available: <http://arxiv.org/abs/1511.06434>
- [154] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, “Deep Learning Techniques for Music Generation -- A Survey,” *ArXiv170901620 Cs*, Aug. 2019, Accessed: Mar. 27, 2020. [Online]. Available: <http://arxiv.org/abs/1709.01620>
- [155] W. Jin, R. Barzilay, and T. Jaakkola, “Junction Tree Variational Autoencoder for Molecular Graph Generation,” *ArXiv180204364 Cs Stat*, Mar. 2019, Accessed: Mar. 27, 2020. [Online]. Available: <http://arxiv.org/abs/1802.04364>
- [156] A. Gupta, A. T. Müller, B. J. H. Huisman, J. A. Fuchs, P. Schneider, and G. Schneider, “Generative Recurrent Networks for De Novo Drug Design,” *Mol. Inform.*, vol. 37, no. 1–2, p. 1700111, 2018, doi: 10.1002/minf.201700111.
- [157] M. H. S. Segler, T. Kogej, C. Tyrchan, and M. P. Waller, “Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks,” *ACS Cent. Sci.*, vol. 4, no. 1, pp. 120–131, Jan. 2018, doi: 10.1021/acscentsci.7b00512.

- [158] J. Arús-Pous, T. Blaschke, S. Ulander, J.-L. Reymond, H. Chen, and O. Engkvist, “Exploring the GDB-13 chemical space using deep generative models,” *J. Cheminformatics*, vol. 11, no. 1, p. 20, Mar. 2019, doi: 10.1186/s13321-019-0341-z.
- [159] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik, “Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models,” *ArXiv170510843 Cs Stat*, May 2017, Accessed: Sep. 14, 2018. [Online]. Available: <http://arxiv.org/abs/1705.10843>
- [160] A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper, and A. Zhavoronkov, “druGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico,” *Mol. Pharm.*, vol. 14, no. 9, pp. 3098–3104, Sep. 2017, doi: 10.1021/acs.molpharmaceut.7b00346.
- [161] A. Kadurin *et al.*, “The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology,” *Oncotarget*, vol. 8, no. 7, pp. 10883–10890, Dec. 2016, doi: 10.18632/oncotarget.14073.
- [162] J. Noh *et al.*, “Inverse Design of Solid-State Materials via a Continuous Representation,” *Matter*, vol. 1, no. 5, pp. 1370–1384, Nov. 2019, doi: 10.1016/j.matt.2019.08.017.
- [163] M. Benhenda, “ChemGAN challenge for drug discovery: can AI reproduce natural chemical diversity?,” *ArXiv170808227 Cs Stat*, Aug. 2017, Accessed: Oct. 09, 2019. [Online]. Available: <http://arxiv.org/abs/1708.08227>
- [164] N. De Cao and T. Kipf, “MolGAN: An implicit generative model for small molecular graphs,” *ArXiv180511973 Cs Stat*, May 2018, Accessed: Apr. 21, 2020. [Online]. Available: <http://arxiv.org/abs/1805.11973>
- [165] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia, “Learning Deep Generative Models of Graphs,” *ArXiv180303324 Cs Stat*, Mar. 2018, Accessed: Apr. 21, 2020. [Online]. Available: <http://arxiv.org/abs/1803.03324>
- [166] M. Simonovsky and N. Komodakis, “GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders,” *ArXiv180203480 Cs*, Feb. 2018, Accessed: Nov. 13, 2019. [Online]. Available: <http://arxiv.org/abs/1802.03480>
- [167] C. A. Grambow, Y.-P. Li, and W. H. Green, “Accurate Thermochemistry with Small Data Sets: A Bond Additivity Correction and Transfer Learning Approach,” *J. Phys. Chem. A*, vol. 123, no. 27, pp. 5826–5835, Jul. 2019, doi: 10.1021/acs.jpca.9b04195.
- [168] M. M. Sultan and V. S. Pande, “Transfer Learning from Markov Models Leads to Efficient Sampling of Related Systems,” *J. Phys. Chem. B*, vol. 122, no. 21, pp. 5291–5299, May 2018, doi: 10.1021/acs.jpcc.7b06896.
- [169] W.-F. Zeng, X.-X. Zhou, W.-J. Zhou, H. Chi, J. Zhan, and S.-M. He, “MS/MS Spectrum Prediction for Modified Peptides Using pDeep2 Trained by Transfer Learning,” *Anal. Chem.*, vol. 91, no. 15, pp. 9724–9731, Aug. 2019, doi: 10.1021/acs.analchem.9b01262.

- [170] N. C. Iovanac and B. M. Savoie, “Simpler is Better: How Linear Prediction Tasks Improve Transfer Learning in Chemical Autoencoders,” *J. Phys. Chem. A*, vol. 124, no. 18, pp. 3679–3685, May 2020, doi: 10.1021/acs.jpca.0c00042.
- [171] “Iovanac, N. C. and Savoie, B. M. Simpler is Better: How Linear Prediction Tasks Improve Transfer Learning in Chemical Autoencoders. Under Review”.
- [172] R. Rendall *et al.*, “Image-based manufacturing analytics: Improving the accuracy of an industrial pellet classification system using deep neural networks,” *Chemom. Intell. Lab. Syst.*, vol. 180, pp. 26–35, Sep. 2018, doi: 10.1016/j.chemolab.2018.07.001.
- [173] A. Esteva *et al.*, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, no. 7639, pp. 115–118, Feb. 2017, doi: 10.1038/nature21056.
- [174] S. Altarazi, R. Allaf, and F. Alhindawi, “Machine Learning Models for Predicting and Classifying the Tensile Strength of Polymeric Films Fabricated via Different Production Processes,” *Materials*, vol. 12, no. 9, p. 1475, Jan. 2019, doi: 10.3390/ma12091475.
- [175] M. A. S. Matos, S. T. Pinho, and V. L. Tagarielli, “Application of machine learning to predict the multiaxial strain-sensing response of CNT-polymer composites,” *Carbon*, vol. 146, pp. 265–275, May 2019, doi: 10.1016/j.carbon.2019.02.001.
- [176] T. Xie and J. C. Grossman, “Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties,” *Phys. Rev. Lett.*, vol. 120, no. 14, p. 145301, Apr. 2018, doi: 10.1103/PhysRevLett.120.145301.
- [177] 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),” *figshare*, Aug. 2017, doi: 10.26434/chemrxiv.5309668.v3.
- [178] M. Łukasz *et al.*, “Mol-CycleGAN: a generative model for molecular optimization,” *J. Cheminformatics Lond.*, vol. 12, no. 1, Jan. 2020, doi: <http://dx.doi.org/10.1186/s13321-019-0404-1>.
- [179] D. Polykovskiy *et al.*, “Entangled Conditional Adversarial Autoencoder for de Novo Drug Discovery,” *Mol. Pharm.*, vol. 15, no. 10, pp. 4398–4405, Oct. 2018, doi: 10.1021/acs.molpharmaceut.8b00839.
- [180] J. Lim, S. Ryu, J. W. Kim, and W. Y. Kim, “Molecular generative model based on conditional variational autoencoder for de novo molecular design,” *J. Cheminformatics*, vol. 10, no. 1, p. 31, Jul. 2018, doi: 10.1186/s13321-018-0286-7.
- [181] R.-R. Griffiths and J. Miguel Hernández-Lobato, “Constrained Bayesian optimization for automatic chemical design using variational autoencoders,” *Chem. Sci.*, vol. 11, no. 2, pp. 577–586, 2020, doi: 10.1039/C9SC04026A.

- [182] S. H. Hong, S. Ryu, J. Lim, and W. Y. Kim, “Molecular Generative Model Based on an Adversarially Regularized Autoencoder,” *J. Chem. Inf. Model.*, vol. 60, no. 1, pp. 29–36, Jan. 2020, doi: 10.1021/acs.jcim.9b00694.
- [183] W. Jin, R. Barzilay, and T. Jaakkola, “Junction Tree Variational Autoencoder for Molecular Graph Generation,” *ArXiv180204364 Cs Stat*, Feb. 2018, Accessed: Jun. 10, 2019. [Online]. Available: <http://arxiv.org/abs/1802.04364>
- [184] S. Wu *et al.*, “Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm,” *Npj Comput. Mater.*, vol. 5, no. 1, Art. no. 1, Jun. 2019, doi: 10.1038/s41524-019-0203-2.
- [185] J. P. Janet, S. Ramesh, C. Duan, and H. J. Kulik, “Accurate Multiobjective Design in a Space of Millions of Transition Metal Complexes with Neural-Network-Driven Efficient Global Optimization,” *ACS Cent. Sci.*, vol. 6, no. 4, pp. 513–524, Apr. 2020, doi: 10.1021/acscentsci.0c00026.
- [186] A. Domenico, G. Nicola, T. Daniela, C. Fulvio, A. Nicola, and N. Orazio, “De Novo Drug Design of Targeted Chemical Libraries Based on Artificial Intelligence and Pair-Based Multiobjective Optimization,” *J. Chem. Inf. Model.*, Aug. 2020, doi: 10.1021/acs.jcim.0c00517.
- [187] N. Ståhl, G. Falkman, A. Karlsson, G. Mathiason, and J. Boström, “Deep Reinforcement Learning for Multiparameter Optimization in de novo Drug Design,” *J. Chem. Inf. Model.*, vol. 59, no. 7, pp. 3166–3176, Jul. 2019, doi: 10.1021/acs.jcim.9b00325.
- [188] A. Nigam, R. Pollice, M. Krenn, G. dos Passos Gomes, and A. Aspuru-Guzik, “Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES,” *Chem. Sci.*, 2021, doi: 10.1039/D1SC00231G.
- [189] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, and P. Riley, “Optimization of Molecules via Deep Reinforcement Learning,” *Sci. Rep.*, vol. 9, no. 1, Art. no. 1, Jul. 2019, doi: 10.1038/s41598-019-47148-x.
- [190] B. Settles, “Active Learning Literature Survey,” University of Wisconsin-Madison Department of Computer Sciences, Technical Report, 2009. Accessed: Oct. 16, 2020. [Online]. Available: <https://minds.wisconsin.edu/handle/1793/60660>
- [191] K. D. Konze *et al.*, “Reaction-Based Enumeration, Active Learning, and Free Energy Calculations To Rapidly Explore Synthetically Tractable Chemical Space and Optimize Potency of Cyclin-Dependent Kinase 2 Inhibitors,” *J. Chem. Inf. Model.*, vol. 59, no. 9, pp. 3782–3793, Sep. 2019, doi: 10.1021/acs.jcim.9b00367.
- [192] P. Ghanakota *et al.*, “Combining Cloud-Based Free-Energy Calculations, Synthetically Aware Enumerations, and Goal-Directed Generative Machine Learning for Rapid Large-Scale Chemical Exploration and Optimization,” *J. Chem. Inf. Model.*, vol. 60, no. 9, pp. 4311–4325, Sep. 2020, doi: 10.1021/acs.jcim.0c00120.

- [193] C. Adamo and V. Barone, “Toward reliable density functional methods without adjustable parameters: The PBE0 model,” *J. Chem. Phys.*, vol. 110, no. 13, pp. 6158–6170, Mar. 1999, doi: 10.1063/1.478522.
- [194] N. C. Iovanac and B. M. Savoie, “Improving the generative performance of chemical autoencoders through transfer learning,” *Mach. Learn. Sci. Technol.*, vol. 1, no. 4, p. 045010, Oct. 2020, doi: 10.1088/2632-2153/abae75.
- [195] Q. Liu, M. Allamanis, M. Brockschmidt, and A. Gaunt, “Constrained Graph Variational Autoencoders for Molecule Design,” in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7795–7804. Accessed: May 14, 2020. [Online]. Available: <http://papers.nips.cc/paper/8005-constrained-graph-variational-autoencoders-for-molecule-design.pdf>
- [196] P. Renz, D. Van Rompaey, J. K. Wegner, S. Hochreiter, and G. Klambauer, “On failure modes in molecule generation and optimization,” *Drug Discov. Today Technol.*, vol. 32–33, pp. 55–63, Dec. 2019, doi: 10.1016/j.ddtec.2020.09.003.
- [197] P. Ertl and A. Schuffenhauer, “Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions,” *J. Cheminformatics*, vol. 1, no. 1, p. 8, Jun. 2009, doi: 10.1186/1758-2946-1-8.
- [198] J. Bradshaw, B. Paige, M. J. Kusner, M. Segler, and J. M. Hernández-Lobato, “A Model to Search for Synthesizable Molecules,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 7937–7949. Accessed: May 14, 2020. [Online]. Available: <http://papers.nips.cc/paper/9007-a-model-to-search-for-synthesizable-molecules.pdf>
- [199] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, “Wasserstein Auto-Encoders,” *ArXiv171101558 Cs Stat*, Dec. 2019, Accessed: Nov. 09, 2020. [Online]. Available: <http://arxiv.org/abs/1711.01558>
- [200] N. Brown, M. Fiscato, M. H. S. Segler, and A. C. Vaucher, “GuacaMol: Benchmarking Models for de Novo Molecular Design,” *J. Chem. Inf. Model.*, vol. 59, no. 3, pp. 1096–1108, Mar. 2019, doi: 10.1021/acs.jcim.8b00839.
- [201] J. P. Perdew and K. Schmidt, “Jacob’s ladder of density functional approximations for the exchange-correlation energy,” *AIP Conf. Proc.*, vol. 577, no. 1, pp. 1–20, Jul. 2001, doi: 10.1063/1.1390175.
- [202] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated Graph Sequence Neural Networks,” *ArXiv151105493 Cs Stat*, Sep. 2017, Accessed: May 18, 2020. [Online]. Available: <http://arxiv.org/abs/1511.05493>

- [203] P. Schwaller *et al.*, “Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction,” *ACS Cent. Sci.*, vol. 5, no. 9, pp. 1572–1583, Sep. 2019, doi: 10.1021/acscentsci.9b00576.
- [204] D. M. Lowe, “Extraction of chemical structures and reactions from the literature,” Thesis, University of Cambridge, 2012. doi: 10.17863/CAM.16293.
- [205] W. Jin, C. W. Coley, R. Barzilay, and T. Jaakkola, “Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network,” *ArXiv170904555 Cs Stat*, Dec. 2017, Accessed: Jun. 26, 2021. [Online]. Available: <http://arxiv.org/abs/1709.04555>
- [206] P. Schwaller, T. Gaudin, D. Lányi, C. Bekas, and T. Laino, “‘Found in Translation’: predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models,” *Chem. Sci.*, vol. 9, no. 28, pp. 6091–6098, 2018, doi: 10.1039/C8SC02339E.
- [207] T. A. Halgren, “Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94,” *J. Comput. Chem.*, vol. 17, no. 5–6, pp. 490–519, 1996, doi: 10.1002/(SICI)1096-987X(199604)17:5/6<490::AID-JCC1>3.0.CO;2-P.
- [208] A. D. Becke, “Density-functional exchange-energy approximation with correct asymptotic behavior,” *Phys. Rev. A*, vol. 38, no. 6, pp. 3098–3100, Sep. 1988, doi: 10.1103/PhysRevA.38.3098.
- [209] A. Schäfer, H. Horn, and R. Ahlrichs, “Fully optimized contracted Gaussian basis sets for atoms Li to Kr,” *J. Chem. Phys.*, vol. 97, no. 4, pp. 2571–2577, Aug. 1992, doi: 10.1063/1.463096.
- [210] J.-D. Chai and M. Head-Gordon, “Systematic optimization of long-range corrected hybrid density functionals,” *J. Chem. Phys.*, vol. 128, no. 8, p. 084106, Feb. 2008, doi: 10.1063/1.2834918.
- [211] F. Weigend and R. Ahlrichs, “Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy,” *Phys. Chem. Chem. Phys.*, vol. 7, no. 18, pp. 3297–3305, Aug. 2005, doi: 10.1039/B508541A.
- [212] Q. Zhao and B. M. Savoie, “Self-Consistent Component Increment Theory for Predicting Enthalpy of Formation,” *J. Chem. Inf. Model.*, Mar. 2020, doi: 10.1021/acs.jcim.0c00092.
- [213] Q. Zhao, N. C. Iovanac, and B. M. Savoie, “Transferable Ring Corrections for Predicting Enthalpy of Formation of Cyclic Compounds,” *J. Chem. Inf. Model.*, May 2021, doi: 10.1021/acs.jcim.1c00367.
- [214] J. Aihara, “Reduced HOMO–LUMO Gap as an Index of Kinetic Stability for Polycyclic Aromatic Hydrocarbons,” *J. Phys. Chem. A*, vol. 103, no. 37, pp. 7487–7495, Sep. 1999, doi: 10.1021/jp990092i.

- [215] Q. Zhao and B. Savoie, “More and Faster: Simultaneously Improving Reaction Coverage and Computational Cost in Automated Reaction Prediction Tasks,” Oct. 2020, doi: 10.26434/chemrxiv.13076087.v1.

PUBLICATIONS

Journal Publications with First Authorship

Improving the Generative Performance of Chemical Autoencoders through Transfer Learning

(Iovanac, N. C.; Savoie, B. M. Improving the Generative Performance of Chemical Autoencoders through Transfer Learning. *Mach. Learn. Sci. Technol.* 2020, 1 (4), 045010. <https://doi.org/10.1088/2632-2153/abae75>.)

Simpler Is Better: How Linear Prediction Tasks Improve Transfer Learning in Chemical Autoencoders

(Iovanac, N. C.; Savoie, B. M. Simpler Is Better: How Linear Prediction Tasks Improve Transfer Learning in Chemical Autoencoders. *J. Phys. Chem. A* 2020, 124 (18), 3679–3685. <https://doi.org/10.1021/acs.jpca.0c00042>.)

Improved Chemical Prediction from Scarce Data Sets via Latent Space Enrichment

(Iovanac, N. C.; Savoie, B. M. Improved Chemical Prediction from Scarce Data Sets via Latent Space Enrichment. *J. Phys. Chem. A* 2019, 123 (19), 4295–4302. <https://doi.org/10.1021/acs.jpca.9b01398>.)

Actively Searching: Inverse Design of Novel Molecules with Simultaneously Optimized Properties

(Iovanac, N.C.; MacKnight, R.; Savoie, B.M. *In Review*)

Other Publications

Zhao, Q.; **Iovanac, N. C.**; Savoie, B. M. A Self-Consistent Component Increment Theory for Predicting Enthalpy of Formation. Submitted

Samp, M. A.; **Iovanac, N. C.**; Nolte, A. J. Sodium Alginate Toughening of Gelatin Hydrogels. *ACS Biomater. Sci. Eng.* 2017, 3 (12), 3176–3182. <https://doi.org/10.1021/acsbiomaterials.7b00321>.