Ionic Liquid Design Using Molecular Simulation and Statistical Methods

Wesley A. Beckner

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Jim Pfaendtner, Chair

David Beck

Dan Schwartz

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

Abstract

Ionic Liquid Design Using Molecular Simulation and Statistical Methods

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

Solicitous use of time is crucial for designing novel materials. For ionic liquids (ILs), the material domain of this work, there are theoretically $10^{14\text{-}18}$ possible pairwise molecular structures—too many to create and observe experimentally.[1,2] For this particular design problem, we turn to computational methods. Here too, an exhaustive approach is intractable. We need an efficient algorithm for determining the most relevant systems to create *in silico*. Luckily, there are many algorithms available for such search spaces. In particular, the Darwinian processes of evolutionary algorithms (EAs), which work by mutating a candidate solution until it attains a desired fitness, are an approach well suited to the task. In the case of material design, the fitness is determined by a quantitative structure property relationship (QSPR) usually in the form of a computationally inexpensive machine learning (ML) model. Because the ML model is based on examples, it pairs well with the search strategy of an EA—starting molecular configurations for the EA are based on the same examples that have informed the ML model. Because of this, when a particular solution deviates far from the structural motifs of the training data, the uncertainty estimate in its property prediction is high. When this occurs, the molecular structure from the EA can be simulated *en masse* using molecular simulation to either: a) *explore* the structure landscape and inform/update the model or b) *exploit* the structure landscape because the prediction is close to our target. This approach is highly flexible since it is agnostic toward the underlying structural landscape; the underlying surface need not be continuous or smooth. By the same token, however, the algorithm's inability to calculate explicit gradients relating features or structures to the target property slow its convergence. In the final section of this work, a method of leveraging property-structure surfaces is explored through the generative capabilities of a class of stochastic neural networks—variational autoencoders—for the explicit rationalization of desired IL thermodynamic properties.

Table of Contents

List of Figures

List of Tables

Introduction

There is a long, rich history of material discovery. Well before chemical theory would play a roll, early materials were discovered by trial and error—and accident. Steel, gunpowder, linens, and paper products kindled early civilization and enabled other technologies: war, transport, culture, and language. The early 20th century saw the first phases of the incorporation of theory into the industrial material design process. Franz Haber and Carl Fritz, working for BASF before the second world war, discovered and scaled a method to synthesize ammonia from the air, and ushered in the modern reality of agricultural surplus. Before their discovery, the industry of Chemical Engineering was limited to the making of colorful dyes.

In the 21st century, the first rational design enabling aspect for materials is the articulation of *need*. Scientists in all fields have identified precise descriptions of what they would *like to have* in materials for a given application. Science has diversified and specialized—from medicine to energy—and can describe such things as what proteins they would like their material to interact with, how quickly it should degrade, what chemical processes it should catalyze, and what phase change behavior it should exhibit. The second rational design enabling aspect for materials is the abundance of data. Materials databases allow data-driven approaches to rational design.

The data-driven approach is not entirely unlike the early trial and error methodology. Albeit, now, the trial and error cycles are done computationally. Cheap, synthetic models attempt to incorporate the important chemical features of the material, stochastic search methods permutate those features, and rigorous calculation or experimentation is done to investigate whether the fast, cheap model has accurately predicted the resulting properties. This paradigm is called adaptive learning. Its' approach is not without analogs in other sciences—it is Daniel Kahneman's *Thinking, Fast and Slow*, wherein a slow, deliberate system two brain is constantly updating a fast and presumptuous system one brain; it is game theory's multi armed bandit problem, wherein in a series of sequential decisions, a tradeoff must be made between exploring the unknown environment and exploiting the known environment. *Adaptive Learning* is a framework of thinking for deliberate application of our models of quantum-chemical reality.

Ionic liquids (ILs) are a class of materials whose melting temperature are at 100 degrees Celsius or lower. They are comprised of a positively charged cation and negatively charged anion. Because of this, they have a high affinity for the liquid phase—a high boiling point—and so are described as "well behaved". Unlike their organic solvent counterparts, they're easy to contain; are non-volatile and non-corrosive. They are involved in a range of energy and medical applications: they've been used to synthesize nanoparticles, stabilize bioactive molecules, and are a potential solvent for redox flow batteries (RFBs).[3,4]

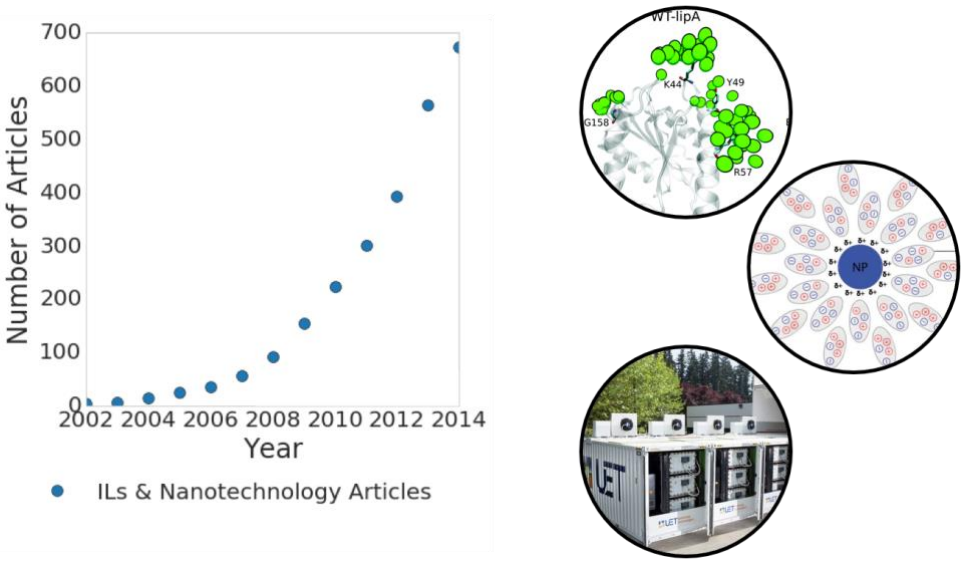


Figure 1. Left: ILs appearing as the subject of nanotechnology articles. Right, top to bottom: ILs in biomolecule studies, ILs as solvents in nano-particle synthesis, and an RFB installation from UniEnergy Technology.[3–6]

The number of articles announcing the application of ILs in nanotechnology has grown exponentially since the early 2000s. Their thermal stability, high thermal conductivity, low vapor pressure, and wide electrochemical windows make them a potential material for many alternative energy technologies. Lastly, the functional groups of the cation are highly-tunable, earning them the classification of "designer" solvents. For these reasons, they are an excellent class of materials to appropriate an adaptive learning strategy.

The first Chapter of this thesis presents statistical methods of model creation, specifically to predict IL viscosity at a broad range of state conditions, temperatures and pressures, and categorical types. This is contrasted with prior methods of viscosity estimation that were limited to narrow classes of compounds and/or dependent on experimental data for the ILs in question. Regularized methods of regression are explored to glean insight on the relevant features that play into the created model. Chapter Two moves beyond feature engineering for these types of models and focuses specifically on model architecture and how chemical intuition can guide the design of these architectures. It also introduces a stochastic search method, a type of evolutionary algorithm, for exploiting the property models created in the first two Chapters. In the third Chapter, the principles of Chapters One and Two, along with molecular dynamics (MD) calculations, are combined to create an adaptive learning and design framework to create ILs with specific densities and heat capacities. In the final Chapter, a method of embedding discrete chemical objects into continuous latent spaces for rational design is investigated for these systems.

Chapter 1 The Statistical Method: Predicting Ionic Liquid Viscosity Across a Wide Range of Chemical Functionalities and Experimental Conditions Using Engineered Features[1]

Introduction

Recent years have seen a huge rise in the successful application of machine or statistical learning type approaches to the discovery and design of new materials. Efforts such as Materials Genome Initiative (MGI) have led to the creation of public data repositories like the Harvard Clean Energy Project[7], Materials Project[8], Open Quantum Materials Database (OQMD)[9], and Automatic FLOW for Materials Discovery (AFLOW)[10]. In the area of solid crystalline materials, research pipelines based on high throughput calculations have enabled rapid population of massive databases. However, many important materials for a wide range of applications are liquids, including emerging solvent classes such as ionic liquids (ILs) or deep eutectic solvents.[11] In contrast to crystalline materials, liquids present a host of challenges that prevent direct mimic of the successful MGI type approaches. For example, calculation of relevant properties with molecular simulations requires statistical sampling (e.g., molecular dynamics (MD) or Monte Carlo) compared with the energy minimization and structural calculations used in solid systems. Intrinsic properties of liquids show much stronger dependence on thermodynamic state variables. Finally, in a potential advantage compared to crystalline materials, public datasets of experimental measurements such as those available from NIST Webbook offer huge opportunities for training statistical learning models.

Many of the current applications of linear and non-linear statistical learning methods in physical sciences are inspired by studies using quantitative structure—property relationships, QSPR; or quantitative structure—activity relationships, QSAR, which were widely used starting in the 1960s and beyond.[12,13] The thousands of QSPR/QSAR models that have been developed in the previous half century have incited both praise and criticism of their reliability and limitations.[14,15] In response to this and in the interest of promoting high-quality models, the Organization for Economic Co-operation and Development (OECD) developed five guiding principles; that a QSAR/QSPR model should have: 1) a defined endpoint, 2) an unambiguous algorithm, 3) a defined domain of applicability, 4) appropriate measures of goodness of fit, robustness and predictive-power, and 5) a mechanistic interpretation when possible.[16] In the interest of objectives 4) and 5) some have stressed creating a small and focused set of descriptors for model development.[17] Others have stressed resourcing the highest levels of domain area knowledge and statistical knowledge, via collaborations between experimental and computational scientists when necessary, to aid with these objectives.[14] However, the continued exponential growth in available data, computing power, and open source software, further complicates the challenge by affecting the weight one would appropriate to any one of these principles. For instance, the

[1] Reproduced in part with permission from Wesley Beckner, Coco M. Mao and Jim Pfaendtner. Statistical models are able to predict ionic liquid viscosity across a wide range of chemical functionalities and experimental conditions. Mol. Syst. Des. Eng., 2018 3, 253.

modern ability to access extremely large experimental datasets and chemical search spaces introduces quite a different problem than that in the pioneering work of Hansch, Leo, and others,[13] where relatively small datasets confined them to a limited feature space.

Taking into account these guidelines and the realities of dealing with ever-growing data sets,[18] this paper describes our application of statistical machine learning models to the prediction of the viscosity of ionic liquid (IL) solvents. ILs have great potential for application in nanomaterials synthesis, bioremediation, and bio-catalysis/enzyme stabilization.[11] They have also been identified as a potential supporting electrolyte material for redox flow batteries (RFBs).[19–21] Especially in the case of RFBs, the solvent viscosity is critical to understand and control as it directly relates to the device's efficiency (via the total energy density). Within RFBs, there are two primary methods of increasing the energy density, and therefore efficiency, of a flow battery: 1) increasing the solubility of the active material or 2) reducing the viscosity of the electrolyte.[22–24] Both of these methods are a consequence of having to actively pump the electrolyte across a membrane to facilitate charge transfer—one would desire that, that volume of fluid have either a high chemical potential energy or a low viscosity. Unfortunately these two characteristics, energy density and viscosity, have tended to be inversely correlated. It is for this reason that an accurate algorithm to determine viscosity based on the molecular constituents of an IL is extremely valuable. With these challenges in mind, we set out to use the public data available in the NIST ILThermo[25] database for training and testing of different predictive models.

Apart from RFBs, in many applications the viscosity of the IL plays a huge economical factor; essentially whenever active transport of the IL is needed. Because of this, many predictive models of IL viscosity have been attempted. They have, however, been largely unsuccessful due to either not reproducing experimental values across categorically different ILs or requiring the use of IL-specific experimental data in predictions.[26,27,36,28–35] Briefly, Matsuda et al. employed group contribution (GC) type descriptors with some accuracy. Their model, however, did not perform very well on a test dataset, reporting an R-squared value of 0.6226.[36] Another GC approach was introduced by Gardas and Coutinho, where they fit the GC-type inputs to the Vogel-Tammann-Fulcher (VTF) equation. This is considered to be one of the most accurate, temperature dependent viscosity models to date but is limited to a narrowly defined set of ILs.[31,34] Zhao et al., used GC-VTF methods to parameterize a UNIFAC-VISCO model. While they reported a low error rate for the regression on their training data, their model is meant to predict binary mixtures of ILs and so is not very useful in terms of exploring a structural search space.[37] As a last look at the GC-type models, Padusyzński & Domańska did an extensive data scraping of the literature to produce a feed forward artificial neural network (FF-ANN) using GC-type inputs and Fatehi et al. applied an FF-ANN to GC-type inputs supplemented with electronegativity descriptors.[31,38] Their models did very well across many IL types. They did not however, examine how their models might perform given an IL type from outside their training data, something we determine for our model in this work.

Other attempts at IL viscosity models have been made without the use of GC-type inputs, notably, hole theory models by Bandrés et al. and volumetric VTF models by Slattery et al., but have required the use of experimental data in some form or another.[33,35] In this work, we introduce a method to accurately predict viscosity for categorically different ILs and broad ranges of temperature (T), pressure (P), and viscosity.

Additionally, we explore the sensitivity of the approach to underlying molecular structure and include these results in the supporting information.

The remainder of this manuscript is organized as follows. The next section combines methodological details with the model development. Following this we apply linear and nonlinear statistical learning methods to understand key structural predictors of viscosity and provide robust statistical analysis on a large data set of experimentally measured IL viscosities. Finally**,** we discuss the applicability of the model across different IL types as well as the underlying features that explain the variance in the viscosity across our training data.

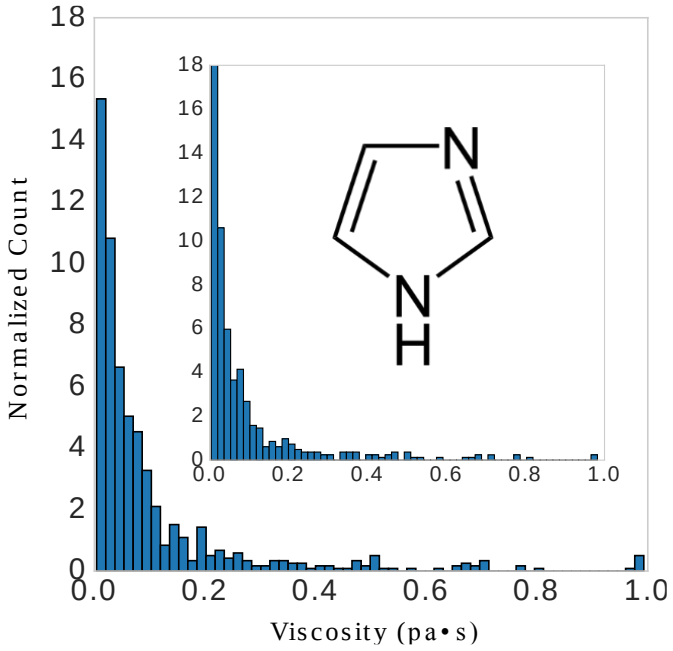


Figure 1-1. 723 data points with temperature, pressure, and viscosity ranges of 273.15-373.15 K, 60-160 kPa, and 0.0035-0.993 Pa·s, respectively. Inset: 453 data points with temperature, pressure, and viscosity ranges of 273.15-373.15 K, 100-160 kPa, and 0.004229-0.982 Pa·s, respectively. The imidazolium base structure is illustrated in the whitespace.

Methods and Model Development

Data Collection and Structure Dependence

Many prior attempts to model IL viscosity[27,28,37,39–41] required narrow definition of cation or anion classes. Therefore, we filtered our starting dataset to emphasize variance in structure in an attempt to understand the limits of a single statistical model. We began with 1405 experimental data points from the ILThermo database and screened for a T range of 273.15-373.15 K, P range of 60-160 kPa, and viscosity range of 0.0035-0.993 Pa·s. The original dataset (including experimental references) before screening is available in the ESI as viscosity_data.csv. Classes of structure included imidazolium, phosphonium, pyridinium, and pyrrolidinium based salts. After this initial screening, the final dataset contained 723 data points consisting of 33 unique salts; 22 anions and 16 cations. We then created a subset of 28 unique imidazolium salts containing 403 data points

including a temperature range of 273.15-373.15 K, pressure range of 100-160 kPa, viscosity range of 0.004229-0.982 Pa·s. We then applied the following protocol to this subset to evaluate how our general model might perform on salt-types not included in its training data, see Fig. 1-1.

Feature Generation

The open source python packages PyChem[42,43] and RDKit[44] were used to generate 633 physiochemical descriptors for each cation and anion in the starting dataset (1266 per IL). T and P, 1/T, T^2, and ln(T) were included in the feature set to give a final data frame of dimensions 723 by 1271. All features were centered about zero and scaled to unit variance. After removing columns with zero variance the final data frame consisted of 723 data points with 771 descriptors.

LASSO: Model Parameterization

The least absolute shrinkage and selection operator (LASSO)[45] algorithm from the SciKit-learn[46] machine learning toolkit was used to shrink the feature space. The primary hyper-parameter of the LASSO model (λ) was optimized in three separate schemes: 5-fold cross-validation (CV), shuffle-split, and bootstrap confidence test algorithms (see Fig. 1-2).

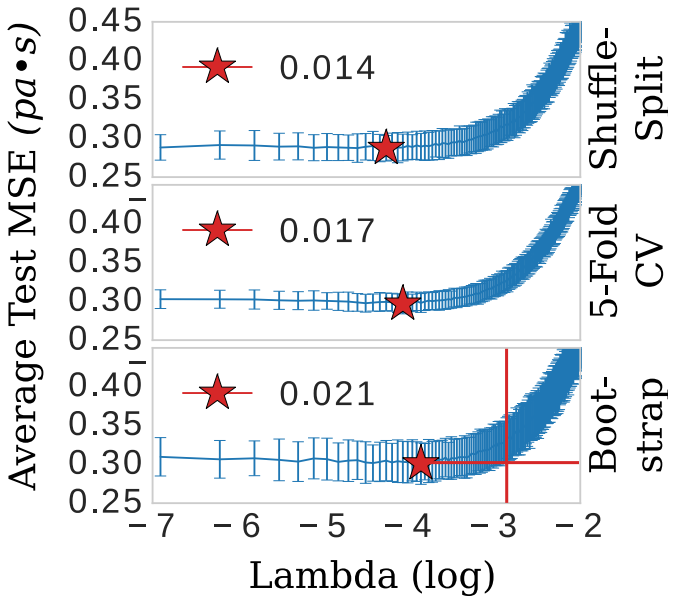


Figure 1-2. Shuffle-split, cross validation, and bootstrap algorithms were used to systematically search for the optimum λ value, the tuning parameter that determines the shrinkage penalty for LASSO. The red crosshairs in the bootstrap panel show that the most conservative (highest) value of λ, 0.021, is still within a standard deviation of the test MSE of a λ value as high as 0.054, i.e. a good selection for λ is highly contingent on the population of training data.

Explained briefly, these algorithms break the data into 80/20 training/testing sets for 300 iterations. In the bootstrap scheme, the final training set is sampled from the training fraction with replacement, offering the possibility of the same data point being sampled multiple times. The shuffle-split schematic is identical to bootstrap apart from that the data is sampled *without* replacement; the original dataset is randomly shuffled and split between train and test. 5-fold CV was performed with slight modification. Keeping in line with the theme of the other methods, a random 80/20 split was made of the original dataset. 5-fold CV was then implemented in the standard way on the 80% fraction until the next iteration, in which another random 80/20 split was made. Each of these algorithms was implemented 300 times; trained and the mean squared error (MSE) evaluated on either 1) the testing portion of the dataset (bootstrap and shuffle-split) or 2) the aggregate from the five folds (5-fold CV).

LASSO: Feature Selection

The three confidence tests provided a starting point for feature shrinkage, however, the statistical variance in the test MSE indicated these optimum λ values were highly dependent on the randomly selected training data. Noting the bootstrap scheme in the bottom panel of Fig. 1-2, the average test MSE for a λ value of 0.021 was still within a standard deviation of the same test MSE for a λ value as high as 0.054 (or log λ -2.9 in Fig. 1-2)—meaning, these two λ values share the same test MSE 68% of the time. While not certain, it is reasonable to posit that a λ value of 0.021 may be leaking noise into the model based on training data selections with incidental patterns in their feature vectors not related in any physical way to viscosity. To test for this, we selected 0.021 as the value for λ and trained the LASSO models on bootstrapped datasets for 1000 iterations to obtain confidence intervals for individual feature coefficients, see Fig. 1-3.

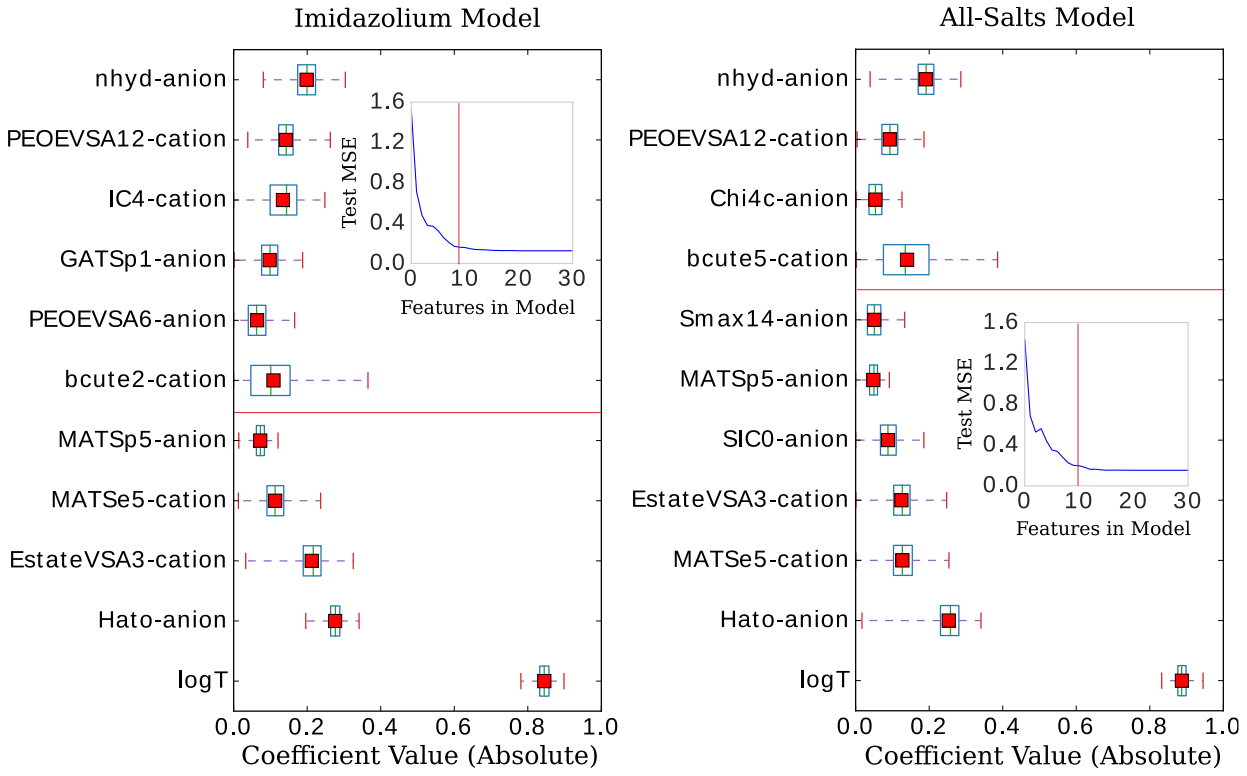


Figure 1-3. Confidence intervals for the most influential features in the respective LASSO models. Insets show that the mean squared error does not improve past the 11 (red, vertical bar) most influential features. Models were trained 1000 times on bootstrapped datasets. X-axis displays the absolute values of the coefficients (values below the red, horizontal line are negative). Y-axis is sorted in ascending order (top to bottom) by the mean value of the coefficient. Green line indicates the median value, red box indicates the mean value, blue box indicates the 2nd and 3rd quantiles, and small, red bars indicate the range. The p-values for all coefficients are very close to zero, with the highest being 1e-66.

A final, bootstrapped model was taken as the mean value of every non-zero return of the coefficients. This model was then used to predict viscosities for a validation set. Features were sorted by the absolute value of their mean and progressively removed from the model to determine at what point the test MSE no longer improved, see the insets of Fig 1-3. Test MSE no longer improved after the top 11 most influential features were included in the models—regardless of whether all four categories of salt or only imidazolium-type salts constituted the training data. These top 11 features had p-values close to 0. After converging on the top 11 features, the LASSO models were trained at λ values ranging from 0 to 1 on their respective feature vectors. The expected approach to zero of the coefficients for each model are shown in Fig. 1-4.

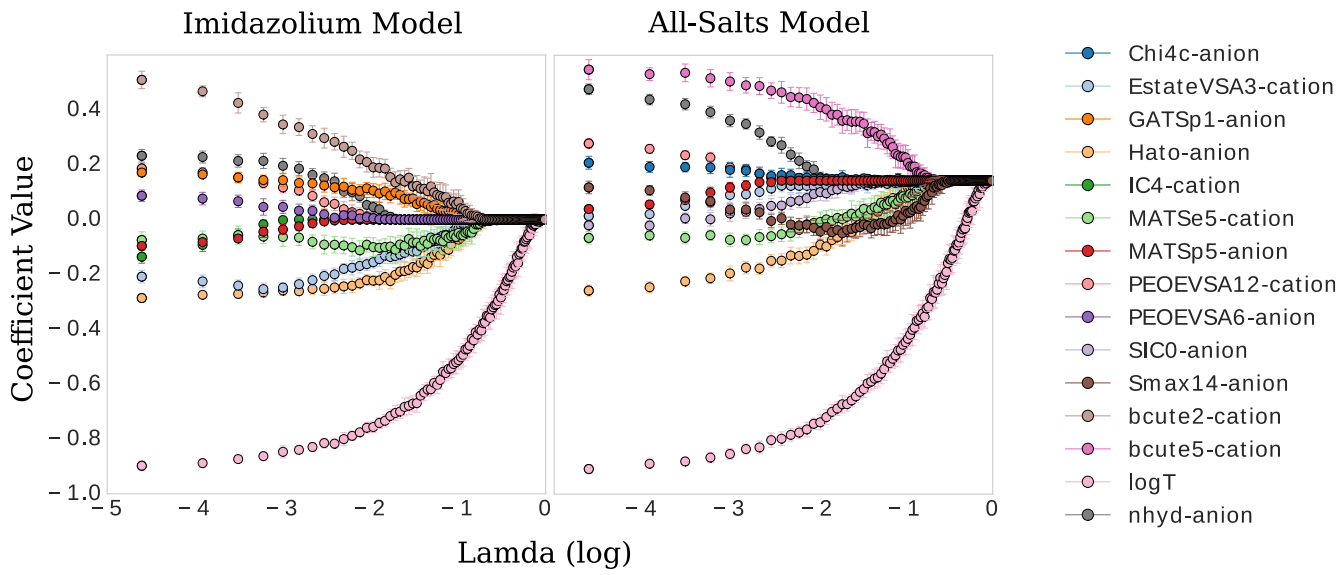


Figure 1-4. Coefficient values versus log λ for the most influential features in the imidazolium and general model, respectively.

Neural Network

The respective selected feature sets were used to train neural networks. The random search algorithm, RandomizedSearchCV was used to parameterize the multilayer perceptron (MLP) Regressor algorithm (a type of FF-ANN), both from SciKit-Learn. In the random search algorithm, ten settings were tested among the following distributions for the specified parameter. For activation: identity, logistic, tanh, and relu; for solver: Limited memory Broyden–Fletcher–Goldfarb–Shanno algorithm (lbfgs), stochastic gradient descent (sgd), and adam; for learning rate: constant, invscaling, and adaptive; and a uniform distribution from zero to one was sampled for the regularization parameter, α. The listed parameter values were sampled without replacement while the uniform distribution for α was sampled with replacement. The final, selected settings for the specified parameter were the following. For activation: tanh, specifying a hyperbolic tan function for the activation function of the hidden layer; for solver: lbfgs, specifying a quasi-Newton method of solving for the weight optimization (a fast and accurate solver for smaller datasets, note that this stochastic solver recalculates its learning rate, α, at every step and nullifies any user-specified starting α); and max_iter: 1e8, the maximum number of iterations.

The remaining parameters were left at their default values: batch_size: auto; early_stopping: False; hidden_layer_sizes: 100; random_state: None; validation_fraction: 0.1; warm_start: False. A full description of these parameters are available in the SciKit-Learn documentation.[46]

Final Evaluation

For both the LASSO and the final ANN, bootstrapping was performed to estimate the variance in the predictions of a validation set. In the typical case, bootstrapping is a method of "internal validation" where

subsets of the data are sampled with replacement and the resulting model is evaluated on the data excluded from the sample. The process is repeated to obtain error estimates for the entire dataset.[47,48] We adapted this typical use case with "external validation" i.e. a portion of our dataset was reserved for validation, was never included in the model training, and the resulting models were used to obtain error estimates for this validation set. The process was as follows. For the ANN model, an 80%-10%-10% split of the entire dataset was used for train-test-validation. Bootstrapping was performed on the 80%-10% sets and validated on the same 10% validation set for 300 iterations (i.e. the validation set data never entered the training data). The same procedure was performed on the imidazolium salt-types only—this time using the other salt-types as the validation set—to investigate how the general model might perform when exposed to salt-types not included in its training data. The procedure was identical for the LASSO model with the exception that a 90%-10% split was made between training and validation since a test set is not required to determine the LASSO coefficients.

Results and discussion

LASSO: Feature Selection

There have been prior approaches to develop models that predict viscosities for ILs. Few of them, however, have been successful across categorically different ILs.[26–30] We investigated the dependency of our approach on structure by applying our procedure on only imidazolium-type salts, investigating the difference in selected features from the general model, and evaluating the ANN performance on the remaining three categories of salts. These selections of the data had a similar distribution of viscosity, T, and P to isolate the dependency of the model on core IL structure, see Fig. 1-1.

The LASSO was used to shrink the physiochemical feature space. The selected features were similar between the imidazolium and general model, see Fig. 1-3. Both models consistently under-predicted viscosity values whose true values were above 0.2 Pa·s and over-predicted the values of those below. The general LASSO model on average had a validation set error of 0.0108 ± 0.0008 Pa·s while the imidazolium LASSO model had a higher validation set error of 0.025 ± 0.003 Pa·s when evaluated on non-imidazolium salts and—as expected—a lower validation set error when evaluated on imidazolium salts, 0.0079 ± 0.0006 Pa·s.

Description of the selected features

In the following we briefly explain the features that were selected by the imidazolium or general model, see also Table 1-1. The features are described extensively in the electronic supporting information (ESI) of the original publication and additional information can be found in the corresponding references.

Table 1-1. Brief summary of selected descriptors by LASSO.

Descriptor	Type	Description
bcute5	Autocorrelation	Burden autocorrelation of Sanderson electronegativity with topological interval 5
bcute2	Autocorrelation	Burden autocorrelation of Sanderson electronegativity with topological interval 2
MATSp5	Autocorrelation	Moran autocorrelation of polarizability with topological interval 5
MATSe5	Autocorrelation	Moran autocorrelation of Sanderson electronegativity with topological interval 5

GATSp1	Autocorrelation	Geary autocorrelation of polarizability with topological interval 1
PEOEVSA12	MOE	sum of atomic van der Waals surface area contributions to partial charges within 0.25-0.3
PEOEVSA6	MOE	sum of atomic van der Waals surface area contributions to partial charges within -0.05-0
EstateVSA3	MOE	sum of atomic van der Waals surface area contributions to electropological states within 0.717-1.165
SIC0	Basak	complementary information content with 0th order neighborhood of vertices in a hydrogen-filled graph
IC4	Basak	structural information content with 4th order neighborhood of vertices in a hydrogen-filled graph
Smax14	Electropological	Maximum electrotopological state of sp hybridized carbon
Chi4c	Connectivity	Fourth order cluster index
Hato	Topological	Topological index of molecular branching

Spatial autocorrelation descriptors. Autocorrelation is a general statistical measure of, broadly defined, how a property of pairwise variables spaced at temporal or spatial intervals are more (positive autocorrelation) or less (negative autocorrelation) similar than they would be for a set of stochastic observations. Several autocorrelation calculations have been introduced in the past century. In ecological processes these have been appropriated primarily due to the importance of stochastic independence to apply the assumptions of classical statistics.[49] Perhaps more fundamentally, time correlation functions have lent themselves to the exact mathematical expression for transport coefficients such as those found in the Green-Kubo relations. Many spatial autocorrelation functions are included in RDKit.

In the following acronyms the small letters signify the type of weighting used in the autocorrelation: atomic masses (m), van der Waals volumes (v), Sanderson electronegativities (e), atomic electronegativities (ae), and polarizabilities (p). The number represents the topological distance between atoms, i.e., the lag associated with the atomic property evaluated at those atomic points. Our models selected two Moran autocorrelations:[50] MATSe5-cation and MATSp5-anion; two Burden autocorrelations:[51] BCUTae5-cation and BCUTae2-cation; and one Geary autocorrelation:[52] GATSp1-anion.

Inspecting the coefficients in Fig. 1-3, a negative Moran autocorrelation of the Sanderson electronegativities and polaraizabilities of the 5th topographical interval coincides with a decrease in viscosity in both models. In the imidazolium model, a positive Geary autocorrelation of polarizability on the 1st topographical interval coincides with an increase in viscosity. Burden autocorrelations of atomic electronegativies of the 2nd (Imidazolium) and 5th (all salts) topographical interval coincides with an increase in viscosity in both models. Also of note, even with the large variance (the greatest in all the selected features by the model) in the Burden coefficients, the associated p-value is extremely low (1e-66), indicating a high probability that this is a descriptive feature for viscosity.

Electrotopological state (E-state) descriptors. The E-state formalism was introduced as a way to economically navigate molecular structure space. In this formalism three intrinsic states of a molecular substructure within a molecule are quantified: its elemental content, its valance state (electronic organization), and its topological state in regard to its atomic neighbors.[53–56] The idea for this approach is that the information density per descriptor can be far greater than an atomic substructure count, where relational/environmental

information is lost. However, the "leanness" of the descriptor comes at a cost: ambiguity is introduced when multiple fragments of the same substructure are contained in the molecule. This has been the subject of some studies where either averages, max/mins, or sums are returned for atomic fragments or the molecule as a whole.[54,55] One E-state descriptor was selected by the general model: Smax14-anion, which is the maximum E-state of any carbon with a triple bonded neighbor. A high maximum E-state for this molecular substructure decreases the viscosity of the IL.

Molecular Operating Environment (MOE)-type descriptors. The MOE-type descriptors use connectivity information and van der Waals radii to calculate the atomic van der Waals surface area (VSA) contribution of an atom-type to a given property.[17] Our models selected three MOE-type descriptors. Gasteiger[57] partial charges (a rapid, iterative approach to calculation of partial charges using only topological data): PEOE-VSA12-cation (both models, increases viscosity), PEOE-VSA6-anion (imidazolium model, increases viscosity); and E-state indices: E-state-VSA3-cation (both models, decreases viscosity).

Basak descriptors. Basak descriptors contain weighted structural and chemical information content for describing physicochemical properties.[58–60] Our models selected two of these types of descriptors. SIC0-anion, complementary information content with 0^{th} order neighborhood of vertices in a hydrogen filled topological graph (general model, decreases viscosity). IC4-cation, structural information content with 4^{th} order neighborhood of vertices in a hydrogen-filled topological graph (imidazolium model, increases viscosity).

Connectivity descriptors. The connectivity descriptors are distinguished by path, cluster, and chain calculations of bond orders (fragments of one bond, two bonds, etc.).[51] They are similar to the Basak and E-state families of descriptors in that a count is made of a specified fragment type. With the connectivity descriptors however, the final score for a given fragment is not influenced by the occurrence of other fragments (as is the case with structural information i.e. entropic calculation in Basak) and all valance/electronic state of atoms/fragments are lost (which are encapsulated in the E-state formalism). Our models selected one of these types of descriptors. Chi4c, a Simple fourth order cluster index (general model, increases viscosity).

Topological descriptors. Hato-anion, a harmonic topological index, is a metric of molecular branching proposed by Narumi.[61] One advantage of this descriptor is that the connectivity state of every atom is used in the calculation of the index, leading to a highly unique index for a given molecule. In the Hato calculation, a lower value indicates a higher degree of molecular branching (e.g. neopentane will have a lower index than pentane). Both models selected this descriptor and It is the most influential molecular-structure based feature leading to a decrease in viscosity (i.e. highly branched anions lead to a more viscous salt).

Constitutional descriptors. Nhyd-anion is a count of hydrogen atoms contained in the molecule. Both models selected this feature as the most influential structural component leading to an increase in viscosity.

Comparison of models

Of the five cation features selected by the imidazolium model, only MATSe5, E-state-VSA3, and PEOE-VSA12 were included in the general model. It is worth nothing, however, that the similar variance and mean value of the Burden features (BCUTEae5 and BCUTae2) for the cation imply a covariant relationship between these two variations of autocorrelating atomic electronegativity. A similar number of the anion-specific features are shared by both models: nhyd, MATSp5, and Hato. Indeed, the anionic features nhyd and Hato are extremely influential in both models (the absolute value of their coefficients are large), second only to log(T). For the cation-specific features: the imidazolium model included BCUTae2, and IC4 while the general model included BCUTae5. For the anion-specific features: the imidazolium model included GATSp1 and PEOE-VSA6 while the general model included Chi4c, SIC0, and Smax14. Interestingly, despite the structural difference between the two models being that of the cationic moiety, the largest difference in the feature vectors pertains to the anion (four shared features with a fifth that is likely to be covariant for the cation compared to three shared features for the anion). At first this would appear unlikely, even more so when considering the cation/anion differences between the models—22 anions and 16 cations for the general model and 21 anions and 10 cations for the imidazolium model. That is to say, even though the imidazolium model is only missing a single anion compared to the general model, it appears to select quite a different set of anion-specific features. However, considering the large coefficient values for the Hato and nhyd descriptors, the overall effect of the anionic moiety on viscosity is very similar for both models, as these two features have an overwhelming influence compared to the other anionic features that are present.

In addition to performing 1000 bootstrap iterations of training LASSO at the optimum λ value, we also evaluated the coefficient values of those top selected features at λ values ranging from 0.01 to 1 to track their approach to 0. One might expect the feature coefficients to fall to zero in the order of their absolute value ranking at λ 0.021. However, since two features working in tandem might better approximate the descriptive quality of one, this may very well not be the case.

There are clear parallels in both models. The features approach zero in three separate clusters. log(T) formed the first cluster in solitude, holding its coefficient value ahead of the other features at higher values of λ. The models begin to differ in the next two clusters. The inverse-ranked approach to zero of the second cluster is as follows; beginning with the imidazolium model: BCUTe2-cation, Hato-anion, MATSe5-cation, Estate-VSA3-cation, GATSp1-anion; for the general model: Smax14-anion, BCUTe5-cation, MATSe5-cation, and Hato-anion. The inverse-ranked approach to zero of the third cluster is as follows; beginning with the imidazolium model: nhyd-anion, PEOE-VSA6-anion, PEOE-VSA12-cation, MATSp5-anion, and IC4-cation; for the general model: SIC0-anion, Chi4c-anion, Estate-VSA3-cation, nhyd-anion, MATSp5-anion, and PEOE-VSA12-cation.

There are a few interesting observations here. First, although the Burden descriptors had the highest variance, p-values, and very low coefficients at λ 0.021, they both were included in the second cluster. This indicates that although descriptive, the response of this variable to selections of the underlying training data varies greatly. Second, the Moran autocorrelations MATSe5 and MATSp5 appear covariant: they both approach

zero but as MATSp5 becomes zero, MATSe5 begins to increase and doesn't hit zero until the second cluster. Lastly, a qualitative comparison can be made between log(T) and the molecular coefficients. Log(T) approaches zero on a very smooth curve, regardless of the behavior of the other features. In comparison, the physiochemical features in the model approach zero tortuously, acting in response to one another to some degree in all cases.

Neural Network

The general ANN model was highly accurate on its validation set, with validation set MSE of 4.7e-4 ± 2.4e-5 Pa·s, see Fig. 5. Recently, Zhao, et al.[29] published a UNIFAC-VISO model of IL viscosity for the same class of structures but a narrower range of temperature (293.15-363.15 K) and a single pressure (0.1 mPa). Their approach resulted in a relative absolute average deviations (RAAD) of 3.92% on a test set with the caveat that their test set contained structurally identical ILs as that of the training data, only differing by the mole fractions of the binary IL mixtures. Converting our validation MSE to RAAD[62], we have a comparative performance in our final general ANN model of 7.1 ± 1.3%. Table A1.1 in Appendix I breaks this RAAD down by structure and T. Perhaps more relevant, Padusyzński & Domańska reported a testing set RAAD (reported as AARD in their publication) of 14.7% for their GC FF-ANN model for nine classes of pure ILs with a broad range of temperature and pressure (253-573 K and 0.1-350 mPa, respectively). A comparison of these results and others are presented in Table 1-2.

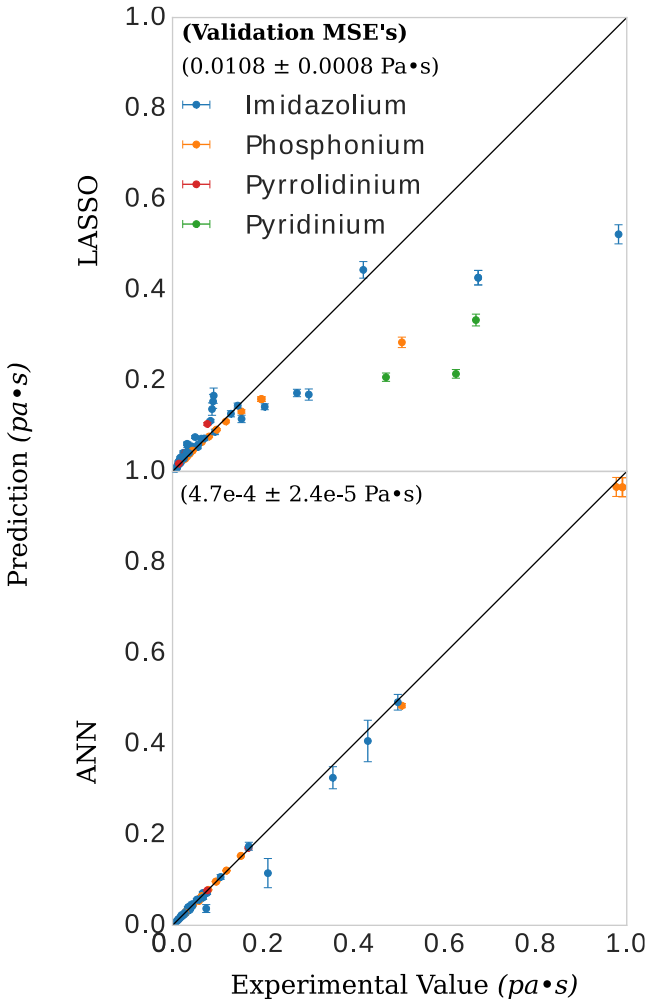


Figure 1-5. Viscosity prediction versus experimental value for the models. The bootstrap was performed on 90% of the available data. The remaining 10% of the data is shown in the figure, along with the prediction and error estimates from the bootstrap models. These error estimates are produced from the variance in the aggregate predictions of all the bootstrapped models. Some of the predictions have a relatively high variance compared to others. There are two possible explanations for this. For one, higher viscosities will inherently have a larger variance simply due to scaling (the same percent variance will appear larger for higher raw values of viscosity, note the two phosphonium predictions in the top right corner, bottom panel). Second, some of the anion-types occur in salt pairs less often than others. Subsequently, whether or not that anion appeared in the subsampled training set will influence the prediction on the validation datum. The three imidazolium-type salts with the highest variance contained either tetrafluoroborate or dimethylphosphate as their anions. The inset numbers indicate the value and standard deviation of the error for both models. The ANN models had, on average, mean squared errors of 4.7e-4 Pa·s for all data points in the validation set and a standard deviation of 2.4e-5 Pa·s for viscosity values ranging from 0.006 to 0.99 Pa·s—an error that translates into a relative absolute average deviation (RAAD) of 7.1 ± 1.3%. Top: LASSO model; bottom: ANN model.

As discussed throughout this chapter, we are interested in how the general model would perform when predicting viscosity for salt-types not included in its training data (i.e. imidazolium, phosphonium, pyridinium, and pyrrolidinium). As a proxy for this, we applied the same protocol to the imidazolium salt-types only and—after observing the changes in selected descriptors in the previous section—evaluated the model on the other three salt-types. This model had a validation set MSE of 0.006 ± 0.001 Pa·s when evaluated on imidazolium ILs and a validation set MSE of 0.08 ± 0.01 Pa·s when evaluated on the non-imidazolium ILs: phosphonium, pyridinium, and pyrrolidnium. This leads us to emphasize caution when applying the general model to salts that are very structurally different than those used in the training data. The comparison of prediction vs experimental viscosity is presented in Fig. A1-1. To our knowledge, other statistical models in the literature have not performed a similar such evaluation. We stress, however, that considering salt-types not in the training data is paramount in the construction of a predictive model for the purposes of designing as-of-yet undiscovered ILs.

Table 1-2. Summary of exemplary IL viscosity models

Structurally Predictive	Pure or Mixed IL	Model	Parameters	Data points	Test Set Error	Disadvantage	Reference
Yes	Pure	physiochemical FF-ANN	11	723	7.1%	Higher test set error than comparable FF-ANN	Our model
Yes	Pure	Physiochemical FF-ANN	13	736	1.3%	Not tested on categorically different ILs from training data	Fatehi et al., 2017[38]
Yes	Pure	GC FF-ANN	242	13,470	14.7%	Not tested on categorically different ILs from training data	Padusyzński & Domańska, 2014[31]
No	Pure/Mixed	UNIFAC-VISCO GC VTF	16/32[a]	52	3.92%	Requires pure IL experimental viscosity data	Zhao et al., 2016[29]
No	Pure/Mixed	QSPR	N/A	5,046	N/A	QSPRs proposed without model	Yu et al., 2012[32]
No	Pure	Hole theory	7	8[b]	N/A	Requires experimental surface tension data	Bandrés et al., 2011[33]
Yes	Pure	GC VTF	3/24[c]	482	13-21%[d]	Applicable to limited set of ILs	Gardas & Coutinho, 2009[34]
Semi	Pure	Volumetric VTF	3	23	9%	Some coefficients are anion-specific, others require QM calculations	Slattery et al., 2007[35]
Yes	Pure	GC	8	300	N/A[c]	Poor prediction for test dataset	Matsuda et al, 2007[36]

[a] 16 parameters per IL pair (32 parameters for binary mixtures).

[b] At least 200 data points collected per IL, eight ILs were included in final regression.

[c] Three parameters for VTF model, two of which were determined from 24 GC parameters.

[d] The test datasets were provided by Padusyzński & Domańska, not the original authors.

[e] The authors reported an R^2 of 0.6226 on a test data set.

Conclusions

We have demonstrated a method of generating accurate models for viscosity using publicly available data from ILThermo and open source software PyChem, RDKit and SciKit-Learn. We present a model that is highly predictive of viscosity across categorically different ILs: imidazolium, phosphonium, pyridinium, and pyrrolidinium based salts. We also evaluated the methodology by which we produced those models; applying the same steps but to a structural separate subset of our data—the imidazolium salts—and tested the model on salt-types it had not seen in its training set, the phosphonium, pyridinium, and pyrrolidinium salts. We found that with structurally different training data, the imidazolium model was able to encapsulate viscosity trends for the other salt-types.

The methodology of using LASSO to pre-select features to then use in a neural network allowed us to benefit from the high interpretability of the LASSO method but also the high flexibility of the neural net. That is, we could evaluate the physical/chemical significance of the features that were selected while also arriving at a highly accurate model with the final neural network. It also allowed more rapid parameterization of the neural net and to avoid overfitting to our training data; i.e. keeping the feature size to training data ratio as low as possible.

In future work, the full value of the models should be actualized by combining them with structural search algorithms to high-throughput screen for low viscosity ILs. One of the most promising search algorithms recently introduced have been genetic algorithms, which allow for flexible fitness tests and a tree-like search structure. The fitness tests can prioritize certain model features, such as those ranked highest by the LASSO coefficient versus log λ evaluations, searching a semi-infinite structural space.

Chapter 2 Building Blocks for an Adaptive Learning Approach: Neural Networks and Genetic Algorithms

Introduction

So-called "data driven" approaches to the modeling of natural phenomenon have infiltrated areas that were once the strict domain of theory and empiricism; areas where models were rationalized using physical principles. In these situations, the domain of applicability for a model is usually obvious—the same physical principles used to rationalize the model guide its appropriation. In contrast, a "data driven" model has the dangerous potential of unwittingly being applied to cases outside the domain of data used in its training.[63] This danger is also an opportunity. The same ambivalence with which the model treats its inputs and outputs that gets it into trouble, also lends itself naturally to adaptive learning strategies. For a statistical model that is employed in an adaptive learning strategy, predictive quality can systematically improve through the acquisition and inclusion of new data without changing the fundamental architecture of the model.

What are the other benefits of an adaptive learning strategy? In the realm of materials science, the materials we use to train models will never be the wonderful, new materials that we are seeking to discover. Consequently, the physical differences between the training data and the candidate material being screened will always need to be reconciled. "Adaptive learning" is simply the official name given to the natural desire to balance the gumption of procuring a good candidate with the need to inform the model.[64] That is, at any point we can either validate (via calculation or experiment) the candidate with the closest desired property profile or validate the candidates with the highest uncertainty and adapt the model. Like the classic multi-armed bandit problem in game theory, "exploration" is done on the *real* material landscape by our validation method and "exploitation" is done on the *conceptualized* landscape by our model. Optimized interactions between both aspects allow us to reap our desired material. In other words, upon the successful creation of a model, adaptive learning is the next step to realize its utility. In our adaptive learning playbook, we focused on the following:

- how can chemical intuition be incorporated into the features and architectures of our models?
- what method should be used for searching the solution space and what method can be used to determine uncertainty in its predictions?

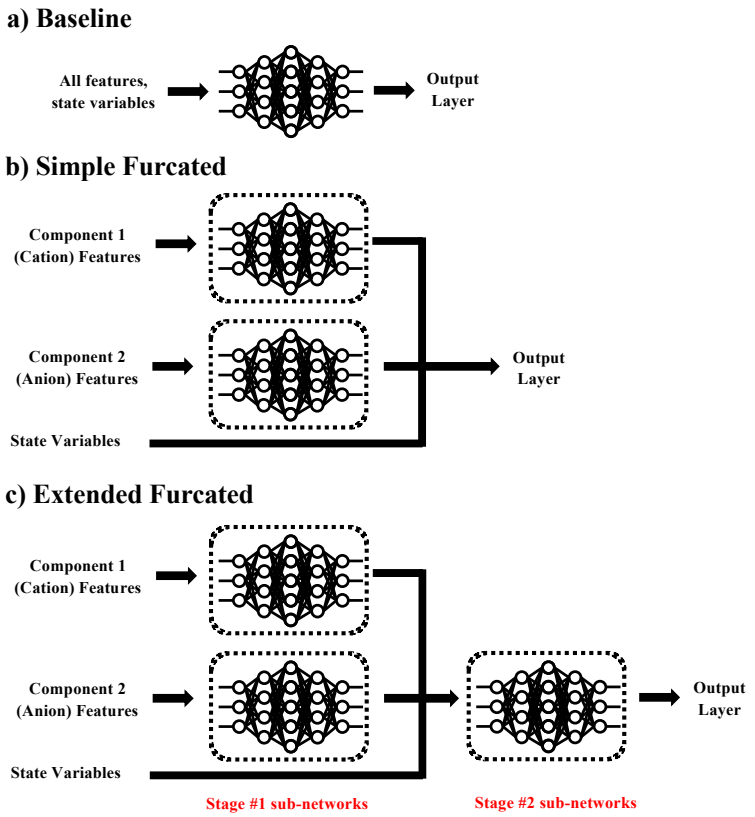
Chapter One discussed how statistical methods like regularization can identify the important chemical features for a given property prediction. The following pages explore how expert domain knowledge can be used to guide the design of the network architectures themselves. As for navigating the solution space, the most direct method would be an exhaustive enumeration. This, however, is computationally infeasible.[1] Other methods are broken down into deterministic (e.g. branch and reduce) and stochastic optimizations (e.g. Tabu search, GAs).[30,36,65,66] Of these approaches, GAs have been used frequently for molecular structure optimizations.[67] The GA is attractive because unlike traditional search methods, GAs perform a guided stochastic search.[68] This guided search can take advantage of other statistical methods such as hill climbing and simulated annealing to achieve faster solutions and overcome local optimums.[69] These "guides" to the algorithm can operate independently of the fitness function—making the framework highly flexible. Other hybrid

deterministic-stochastic methods like decomposition methods, require a high degree of user interaction: it must be preemptively known how the design problem can be broken down into manageable subparts. It is for these reasons that a GA was an attractive search method for adaptive learning and design for ILs.

Designing Furcated Neural Networks[2]

There is limited availability of experimental data for ILs—at least at the level typically prescribed to deep learning methods. At the same time, ILs have a rich development history of empirically-driven equation of state (EOS) or rule-based models to describe their properties[70] and have been highly parameterized within physics-based simulations like Molecular Dynamics and Monte Carlo (MD/MC).[71] A NN that leverages the chemical intuition inherit in these other methods would therefore have an added advantage.

To explore whether furcating the NN in relation to its cationic and anionic moieties and state properties performs better than its fully interconnected counter-part we designed three NN architectures, Fig 2-1.



[2] Reproduced in part with permission from Khushmeen Sakloth, Wesley Beckner, Jim Pfaendtner, and Garrett B Goh. IL-Net: Using Expert Knowledge to Guide the Design of Furcated Neural Networks. IEEE International Conference on Big Data. 2018.

Figure 2-1. Three NN architectures for prediction of IL properties. (a) baseline model consists of fully interconnected layers, (b) simple furcated model linearly combines outputs of subdomains, and (c) extended furcated network recombines subdomain with additional hidden layers before final output.

The NN architectures were evaluated against a 23,982 entry dataset composed of three IL properties: viscosity, density, and heat capacity at temperature and pressure ranges of 278.15-373.15 K and 100 – 20,000 kPa, respectively. For the three overarching architectures, sub-hyperparameter random searches were invoked: subdomains searched through 2-5 fully-connected hidden layers and 16, 32, 64, 238, 256, and 512 neurons per layer with ReLU activation functions. Every layer passed through a 0.5 dropout filter before receiving the next layer. Final selected hyperparameters were according to the validation loss. We developed a python package on GitHub and pypi called Salty, to interface the data with the final models.[72] Further details of the training and model parameterizations can be found in the original publication.[73]

In every case, the extended furcated model outperformed its baseline counterpart, Fig 2-2. The simple furcated model, which consisted of a linear combination of the subnetwork outputs, was unable to learn the intra-molecular interaction of the salts' anion and cation moieties and thus suffered in its' prediction ability.

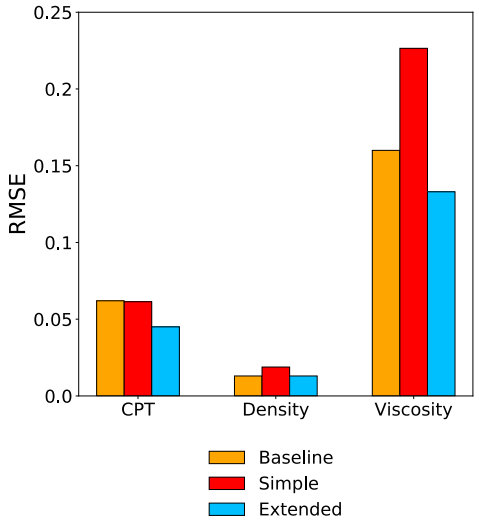


Figure 2-2. RMSEs of baseline, simple furcated, and extended furcated models for heat capacity, density, and viscosity.

The curated dataset was purposefully drawn so that a multi-objective optimization strategy could be performed. In a final demonstration, we showed that by assigning the NNs the predictive task of *all three* properties at once, and overweighting the *actual property of interest*, this final model outperformed its' respective extended furcated, single task model, Fig. 2-3. These three over weighted models had RMSEs of 0.042 JK/mol for heat capacity, 0.0128 kg/m^3 for density and 0.127 Pa/s for viscosity predictions.

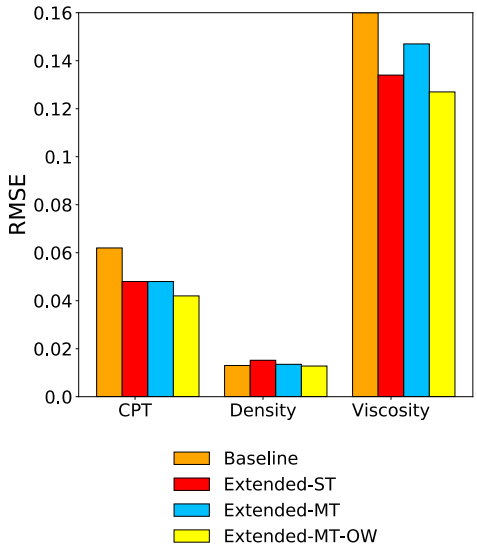


Figure 2-3. RMSEs of baseline and extended single task models (for reference, duplicated from Fig. 2-2.) and extended multi-task and multi-task over-weighted models for heat capacity, density, and viscosity.

In summary, this work demonstrated that expert domain knowledge can, and should, be used to guide the design of NNs when possible and that multi task outputs with the proper reweighting can outcompete single task models. In Chapter Four of this document, I describe how we used this same approach with a relatively recent type of NN model, variational autoencoders.

The Genetic Algorithm

To search through molecular candidate space, we developed an engine as part of the python package Genetic Algorithms for Identifying Novel Solvents (GAINS). The engine works by selecting a starting candidate from a pool of molecular (Chromosome) objects. The candidate is mutated, and the new child assessed with the fitness function. If the fitness of the child is higher than that of its parent, it is selected as the new candidate. This process is repeated until a desired fitness is obtained, Fig. 2-4.

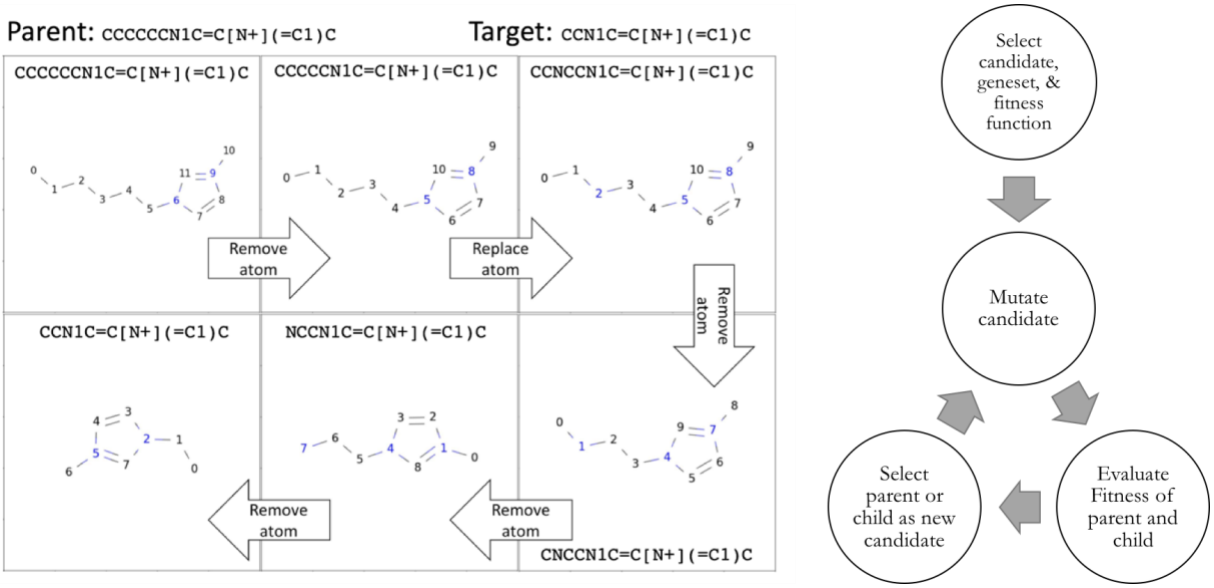


Figure 2-4. Right: schematic of the GAINS engine algorithm beginning with selection of a molecular candidate and subsequent rounds of mutation and selection. Left: an example of the engine mutating a molecular candidate.

The fitness function can be anything so long as a numerical score can be returned from the Chromosome object. Since the Chromosome object inherits the methods of RDKit's Mol and RWMol classes,[44] scores can be based on any number of SMILES or SMARTS representations, or descriptor or fingerprint types (at the time of writing, molecular graphs have become a promising chemical representation due to their 1:1 representation:chemical mapping—these representations can be modularly inserted into the GAINs framework as well). Users have the option to interface with the engine using their own fitness and display functions and genetic pool (starting candidates) and gene sets (atoms and fragments)—the package is highly configurable. Full documentation for the code is available in Appendix II.

The Uncertainty Estimator

A successful adaptive learning and design strategy requires an uncertainty estimator. An uncertainty estimator determines how different a candidate solution is from the data used to guide the search algorithm and associates this with error in its property prediction. Having a reliable uncertainty estimator, the design strategy can more effectively weigh the balance between exploitation and exploration in the structural landscape. In the ML world, this uncertainty is an analog of outlier detection since the models are based on historical examples of data.

There are two primary methods of outlier detection, proximity methods and projection methods. Projection methods involve reducing the dimensionality of the dataset, i.e. "projecting" $n-$ $-$dimensional data onto $m-$dimensional data where $m < n$ This could be done via principal component analysis (PCA), for instance. This approach is especially attractive because it would involve the inputs to the model directly, broadcasting input features onto a lower-dimensional space and then checking whether the salt candidate

produced by the GA was far from this principal axis. Even more attractive, however, are the so-called proximity methods, also known as similarity mappings, due to their ubiquity in drug design.

Most of these similarity mappings or proximity methods are permutations of the Jaccard (or Tanimoto) Index—defined as the intersection of two sets over their union—and molecular fingerprints—bitwise (or count wise) representations of molecular structures.[74] Both similarity mappings and molecular fingerprints have been studied extensively. We included these similarity mappings as part of the scoring capability of the GAINS engine. These similarities and how they ramified themselves in search results are explored in Chapter Three.

Quality of the Genetic Algorithm

Before diving into the primordial adaptive learning work, there are some mentionable details involving the quality of the GA. One interesting result had to do with the sensitivity of the GA to the underlying ML model. The following data was used to train a feed forward NN:

- Unique salts: 44
- Data points: 9053
- Temperature range (K): 219.01-473.15
- Pressure range (kPa): 86.5-206900.0
- Density range (kg/m^3): 852.3-1741.5

and led to about 20% error for the first few candidates generated by the GA and evaluated with MD. The original model had ~9k data points but only 44 unique salt types. The model was short on structural information contributing to IL density. Retraining the model on the following data drastically improved the GA:

- Unique salts: 471
- Data points: 6270
- Temperature range (K): 290.13-314.9
- Pressure range (kPa): 100.7-191.9
- Density range (kg/m3): 871.3-1715.67

Since the GA is searching for salt densities at a specific T and P (300K and 101kPa); it doesn't need to model densities far away from those target values. **The overall dataset is smaller, but it contains more relevant data.** The subsequent model has produced candidates with highly accurate densities. Briefly, for 127 salts, average error rate between the model and MD calculation was 5.1% and was performed at target densities of 1100, 1300, and 1500 kg/m^3. This result informed us that the most accurate models will be trained at properties of the target state property values (T and P).

As the model attempts to procure structures for which it has fewer underlying data, it produces answers with higher error rates. In the context of adaptive learning and design, the property distribution of the underlying training data seems to be a good representative of uncertainty. But there are still are other options for uncertainty metrics that may prove just as valuable; fingerprinting and similarity metrics were discussed as means of evaluating uncertainty. Essentially this is determining whether the model has been trained on the right data to accurately capture the structures procured by the engine: if the similarity metric between the engine's solution is wildly different than that of any of the cations in the training data then there is a very high uncertainty about the prediction.

Iteration Method

We outlined two approaches to applying adaptive learning and design: to improve the model performance within the bounds of its training data and to investigate how *extreme* property values might be obtained that exist outside the bounds of the *original* training data. Towards the second initiative I verified that the model failed outside the bounds of its training data (800-1800 kg/m^3) with error rates as high as 35%. I also confirmed that MD generated data can improve the predictive power of the model when utilized as training data, Fig. 2-5.

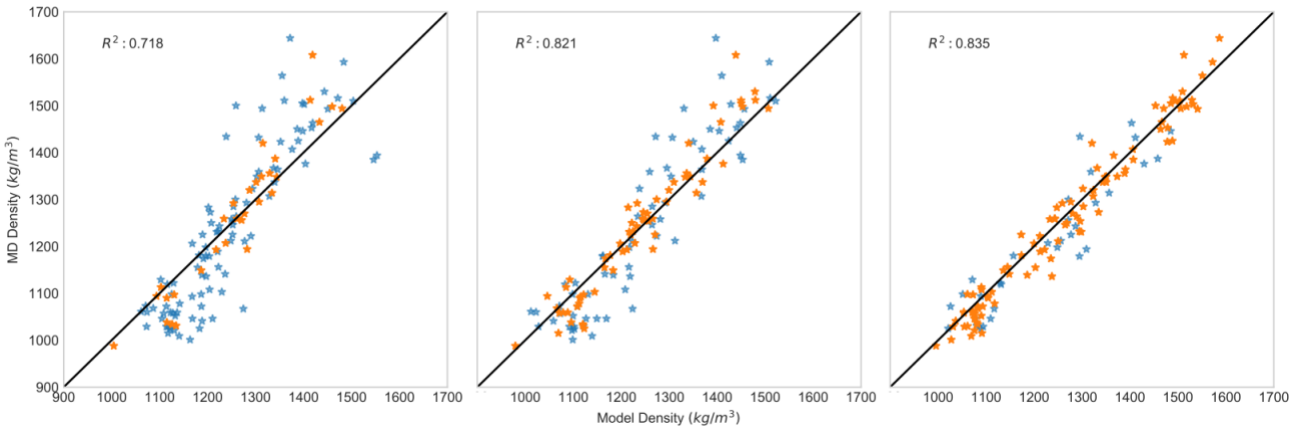


Figure 2-5. Small multiple plot of Calculated density from MD versus the predicted density from the updated ML model using MD data (orange). R^2 inset is for the test data (blue).

In Fig. 2-5: from left to right MD density data (the 127 data points from structures generated by the GA) has been supplied to the NN in 25% increments as training data (orange) and the resultant model was tested on the remaining MD data (blue). The inset R^2 value is for the test (blue) data. As illustrated by the R^2, the model improves in predictive power as this additional data is supplied to the model.

Chapter 3 Adaptive Learning and Design: Combining Fast, Statistical Methods with Accurate Physics-Based Methods to Optimize within Discrete Chemical Space[3]

Introduction

There are growing examples of materials projects that supplement conventional discovery methods with machine learning and data science. The Harvard Clean Energy project is one well known early computational materials database to combine empirical/heuristic-based quantitative structure property relationship (QSPR) modeling with first principles.[7] Other projects with similar ambitions have emerged, fueled by clear policy impetus like the Materials Genome Initiative and new machine learning techniques: generative adversarial networks (GANs), variational autoencoders (VAEs), dropout regularization methods, and new evolutionary strategies.[75–79] Among others, these efforts have manifested in the Materials Project,[8] Aflowlib,[10] and OQMD[9] repositories as well as new strategies for computational molecular design. [63,80,81]

Many recent advances in computational design are confined to solid materials (polymers and crystalline solids) and small molecules (drug-like molecules/pharmaceuticals). Polymer science has benefited enormously from molecular fragment level descriptors while crystal solid studies have made strides applying data-driven models to atomic potentials.[63] Both polymer and crystalline materials have a high-degree of ordering in their molecular configurations, making them amenable to functional relationships between molecular/atomic level inputs and macroscopic properties. Drug-like molecules have a clear benefit as well in terms of computational design: they comprise large structural databases and, inherently, are composed of relatively few atoms. The ZINC database contains over 35 million compounds and the GDB-13 database contains almost one billion compounds.[82,83] Limiting heavy atom counts in molecular candidates both compresses the molecular search space and curbs the "curse of dimensionality" problem in determining data driven QSPRs—significant benefits in a computational design strategy. These huge databases have stimulated innovative methods that map discrete molecular-structure space to a continuous latent space for gradient based and semi-gradient based (i.e. evolutionary strategies) optimizations.[80]

On the other hand, application of machine learning tools to design liquid materials has been challenging due to the computational cost of physics-based simulations (molecular dynamics/Monte Carlo simulations or MD/MC), databases with few molecular entries upon which to base QSPRs, and the lack of microscopic ordering compared to solid/crystalline materials to procure dependable surrogate, machine learning models in place of computationally expensive physics-based simulations. Even when high performance computing is available, MD/MC can still suffer from poor atomic potentials, especially when long-range interactions of charged molecules dictate the transport or thermodynamic properties of interest.[84]

---

[3] Reproduced in part with permission from Wesley A. Beckner and Jim Pfaendtner. Fantastic Liquids and Where to Find Them: Optimizations of Discrete Chemical Space. J. Chem. Inf. Model. 2019, 59, 6, 2617-2625. © American Chemical Society 2019.

We therefore view a computational molecular design strategy for solvents as an optimization problem that requires some unique elements. Primarily, both the QSPR and search algorithm must not rely on enormous amounts of experimental data and, secondarily, since the QSPR model may be data-starved, a physics-based simulation should be used to validate the predicted properties and phase behavior of the procured solvent. These two steps, a QSPR rudimentary predictor and a physics-based validator, should work together, the calculated data of the second reforming the predictive architecture of the first, in an adaptive learning and design (AL&D) framework, Fig. 3-1.

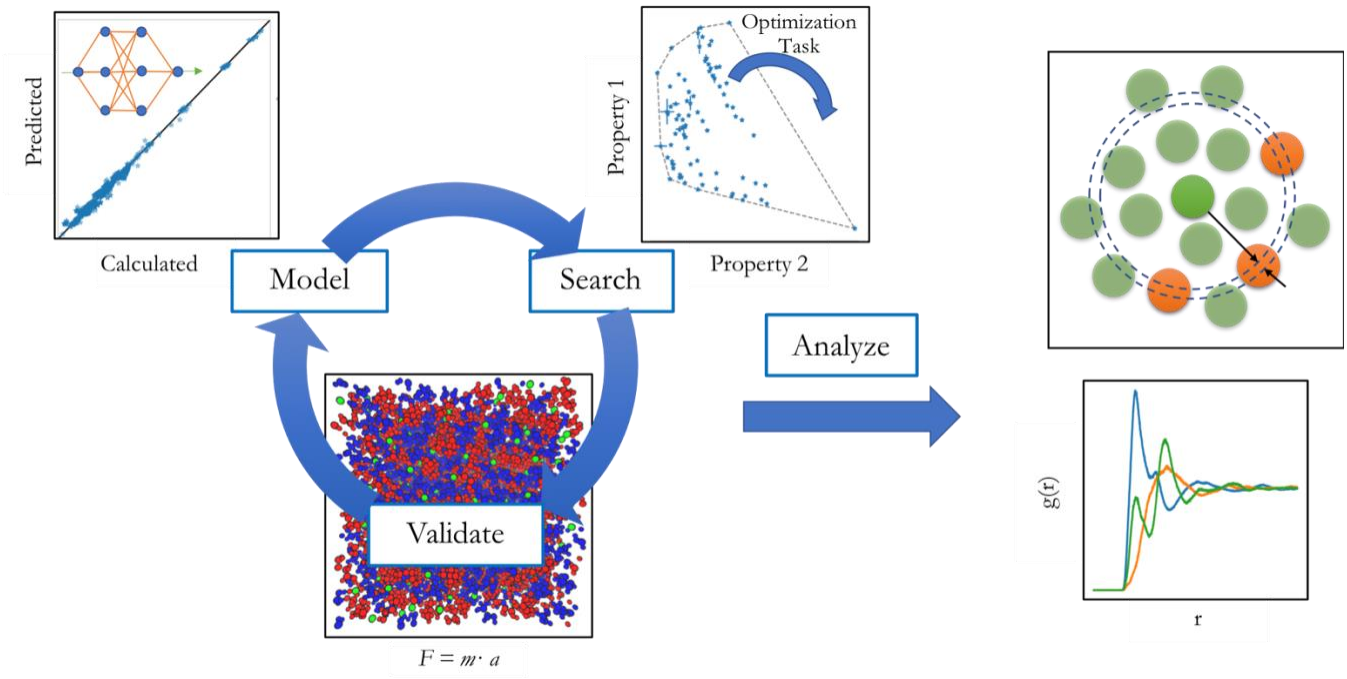


Figure 3-1. AL&D overview. Model: QSPR/NN from experimental data provides a model for design criteria; Search: GA discovers molecular constructs with desired properties; Validate: MD calculates properties and updates the QSPR/NN; Analyze: phase behavior and molecular stability of products are calculated from MD and QM.

The particular solvents we design in this work are ionic liquids (ILs), which have an estimated $10^{14\text{-}18}$ possible molecular configurations.[1,2] To explore this molecular structure space, we employ a genetic algorithm (GA), and use a linear combination of separate QSPR/NN models of heat capacity (C_p) and density (ϱ) as its' fitness criteria. We use quantum mechanics (QM) calculations at Hartree-Fock level of theory followed by molecular dynamics (MD) with a classical force field as our physics-based validation step and verify that we can calculate the properties accurately by comparing our results to experimental data. Most engineering design problems encompass conflicting and/or cooperating objectives. These are broadly classified as Multi-objective Optimization Problems (MOP).[85,86] In MOPs it is common to, rather than find some global optimum, find a set of solutions, referred to as the Pareto set, or Pareto optimal frontier (PF). In this work, as an illustrative example, we investigate our AL&D strategy on this design goal, the PF, using the two properties, C_p and ϱ.

C_p is a critical property for heat transfer fluids and ILs have been identified as strong candidates in this application area, particularly in concentrated solar power.[87,88] ϱ as well, in just about any industrial application of fluids, plays a crucial role in overall application design and is related to other thermophysical properties. For instance, Barycki et al. used molecular mechanics/first principles to calculate ϱ which in turn served as an input to their IL viscosity model.[89] Several other C_p and ϱ models for ILs exist in the literature. These models will primarily consist of either first principles calculations or regressions on experimental data where ILs have been codified into contributing group structures (GCs) or similar physio-chemical descriptors. Rybinska et al., investigated molecular mechanics at three levels of theory to then incorporate as geometric parameters for their QSPR ϱ model and found that Hartree-Fock yielded the highest R^2.[90] Keshavarz et al. introduced a method for calculating ϱ with only elemental content information of the cation-anion pair for some specific classes of ILs.[91] Recently, Paduszyński introduced a GC approach to modeling ϱ based on an extensive database of experimental data and achieved a best relative absolute average deviation (RAAD) of 1.43% on test set data.[92] Several C_p QSPR models have been introduced as well. Paterno et al., used ILThermo data and other sources to develop partial least squares QSPR models of C_p for 65 ILs.[93] Zhao et al., developed a quantum chemical descriptor based QSPR for predicting C_p. Comparing a multiple linear regression model with a type of neural network trained—in lieu of traditional backpropagation algorithm—by an augmented parameterization method (so called extreme learning machine) they were able to achieve RAADs of 2.72% and 0.60%, respectively.[94] Our methodology produces accuracies comparable to these previous studies. In addition, we explore how our error rates improve in our AL&D framework, where we are *computationally synthesizing* new molecules, something that to our knowledge has not been done for these systems and properties.

There are several ways to calculate prediction intervals for a single output from a machine learning model.[95] But these methods can be computationally intensive for complex, nonlinear models requiring either Jacobian or Hessian matrix computation (delta/Bayesian methods) or additional neural network training (mean-variance/bootstrap methods). Computing an indicator on the fly during high throughput chemical screening is valuable when validation steps are expensive/nonexistent or when evaluating convergence criteria in an iterative AL&D approach. We explore a domain-specific strategy for estimating error from our models using chemical similarity scoring and kernel density estimates.

The rest of the paper is organized as follows: We give an overview of our design protocol; the QSPR/NN model, the GA, the QM/MD framework, and how they fit together in the context of our AL&D strategy. Each of these design protocols are discussed in greater detail at the end of the document in the methods section with extensive figures and tables in the supplementary information (SI). After these overviews we discuss the search improvements with iterations in the AL&D cycle, evaluate our on-the-fly method of estimating variance in the QSPR/NN outputs, evaluate our approach on breaking the PF formed by the experimental C_p and ϱ data, and give a more detailed analysis of 'high value' liquid materials discovered by our method.

Results and Discussion

QSPR/NN development

The QSPR/NN model is trained from experimental data obtained from the ILThermo database maintained by the National Institute of Standards and Technology (NIST).[25] The model architecture and train/test protocol follow a similar approach we introduced previously.[73,96] The acquired datasets were filtered for state properties of 99-102 kPa and 297-316 K to focus weight assignments during backpropagation on response to molecular structure variance. This resulted in 1734 data points and 177 unique salt structures in the C_p dataset and 5631 data points and 461 unique salt structures in the ϱ dataset. Between the two datasets, this made for a total of 494 unique salt structures. After obtaining ϱ and C_p data from ILThermo, RDKit[44] was used to generate 94, 2-dimensional, physio-chemical descriptors based on the SMILES representation of the chemical data. This descriptor generation followed our prior work modeling IL viscosity with NNs.[96]

Of the IL systems at the specified state property ranges, 123 systems had three or more repeated measurements in the ϱ dataset and 427 in the C_p dataset. The mean of the relative standard deviations (RSD) of these measurements were 3.1% (N=123) and 0.5% (N=427) for the C_p and ϱ datasets, respectively. The train/test set root mean squared errors (RMSE), R^2 and RAAD of the QSPR/NN models following an 80/20 train/test protocol are indicated in Table 3-1. For both properties, training models came within a percent of the irreducible error (the mean RSD) in the experimental data.

Table 3-1. QSPR/NN performance details where N is train or test set size.

QSPR/NN Metrics	Property			
	C_p (irreducible error = 3.1%)		ϱ (irreducible error = 0.5%)	
	Train (N=1387)	Test (N=347)	Train (N=4504)	Test (N=1127)
RMSE	71 J/mol/K	40 J/mol/K	17.15 kg/m^3	19.64 kg/m^3
R^2	0.90	0.95	0.99	0.98
RAAD	3.83%	4.24%	1.02%	1.08%

GA Framework

The genepool for the GA consisted of the cations and anions forming the outer join of the experimental ϱ and C_p datasets. Cations were sampled from the genepool and mutated by the GA while anions were only sampled/used during fitness evaluation. After selecting the cation and anion, the GA had the following mutation options for the cation: adding/removing bonds or adding/removing fragments and atoms. The available atoms were carbon, nitrogen, oxygen, fluorine, phosphorus, sulfur, and chlorine. A totally of 39 common functional groups and alkanes up to C4 were available as molecular fragments and are included in the supplementary information. After performing a mutation, the cation was aromatized/kekulized to ensure that atomic valencies were followed. If the mutation passed this step, the cation-anion pair were evaluated by

the fitness function, a linear combination of the QSPR/NN ϱ and C_p models. If the cation-anion pair came closer to the property targets for ϱ and C_p, it was identified as a successful mutation.

QM and MD Calculations

The validation step of the AL&D protocol consisted of QM calculations followed by MD simulations of the procured cation-anion structures. The validation step aims to: 1) ensure that the formulated cation is molecularly stable via QM calculations and 2) verify that the predicted properties from the QSPR/NN models are correct. The workflow for the QM calculations and force field generation closely follows our prior work investigating the applicability of the general AMBER force field (GAFF) to IL property prediction.[71] The method was evaluated on cation-anion pairs for which experimental ϱ and C_p values were known, and resulted in a 2.9% and 2.5% RAADs, respectively, Table 3-2 (Table 3-1 provides related details on the total train/test mean squared error for the QSPR/NN). Further details of the QM calculations and MD method to calculate the ion-ion (cation-cation, anion-anion, cation-anion) center-of-mass based radial distribution functions (RDFs), ϱ, and C_p for each IL are discussed in methods and SI including the full details of the comparison of MD and experiment (Fig A3-2). In the last section we demonstrate an analysis of charge and steric center based RDFs for a handful of top performing ILs and compare with experimentally verified IL analogs.

Table 3-2. comparison of RAADs for MD and QSPR/NN calculations and standard deviation (σ) reported in the experimental data for N=11 (Cp) and N=14 (ϱ) common IL systems from the imidazolium, pyridinium, and ammonium families.

	Property	
Method	C_p (N=11)	ϱ (N=14)
Experiment (σ, %)	5.2	0.6
MD (RAAD, %)	2.5	2.9
QSPR/NN (RAAD, %)	3.8	1.3

AL&D Overview

After a completed MD simulation, the RDF was checked for liquid-like behavior. Structures with strong deviations from expected liquid like behavior (e.g., phase separation or solid/glassy like behavior) were removed from the pool of candidate molecules. Details of such examples are included in the SI. This step was performed by manual screening of the MD-output of the candidate molecules but could be automated in the future based on common heuristic analysis of, for example, the RDF. For multiple AL&D cycles, the calculated ϱ, C_p, and SMILES string of the IL were appended to the original dataset. Once all MD simulations for a given round had ended, the QSPR/NN was retrained using the same protocol as described previously. This new QSPR/NN was then used to guide the next GA search.

Prediction Improvement with AL&D Cycles

Predictive quality should systematically improve in an AL&D framework. To evaluate this aspect of our protocol, we set an arbitrary design target of 1000 J/K/mol heat capacity at constant pressure (C_p) and 1000 kg/m^3 ϱ and instantiated four cycles of search, validate, and retrain, Fig. 2. C_p had the most significant improvement in prediction error (n.b., this error is based on the comparison of the NN prediction with MD simulation results) of the two properties, starting at 55% and decreasing to 52%, 28%, and finally 18% by the fourth round. Density had a more modest improvement, but overall lower error rate: 10%, 5%, 6%, to 2%. At which point, 2% falls within the irreducible error set by our validation method, i.e. the MD error for ϱ (2.9%). The error rate trends taken together, indicate that acquiring additional data from MD is an effective AL&D strategy for these systems. At each cycle, the GA was allowed the same amount of time and processing power to achieve its search criteria (in this case 24 hours on a single core). The increased precision of the retrained QSPR/NN model is the most likely reason for the search hit trend: 47, 30, 34, to 7.

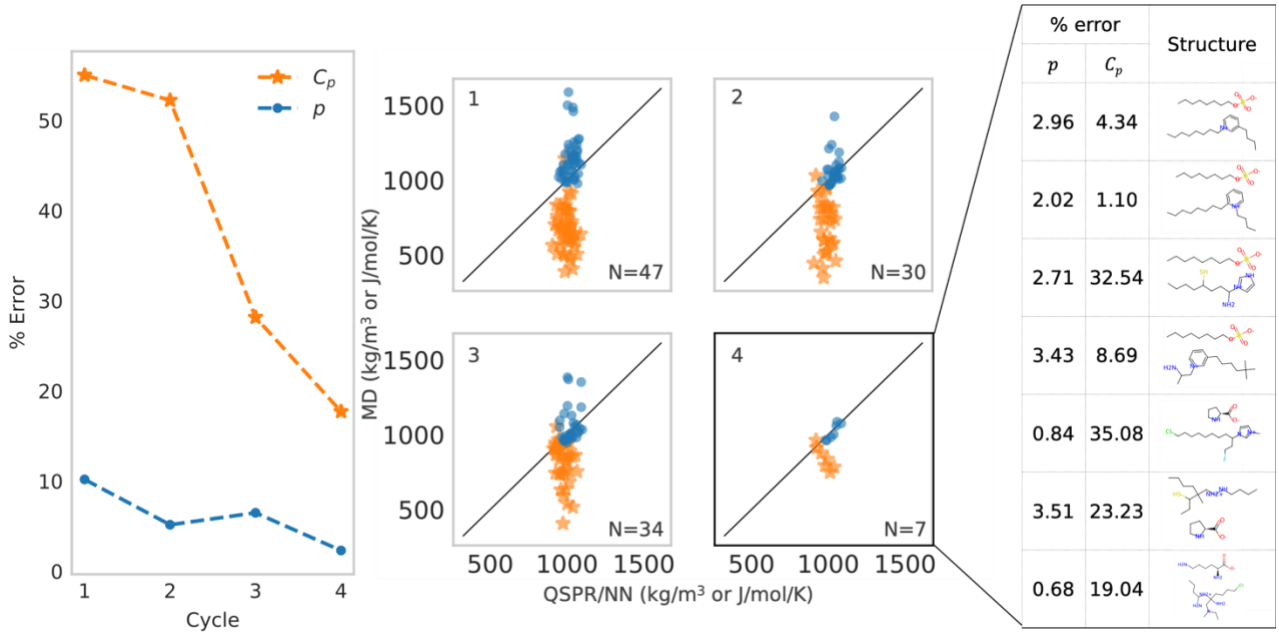


Figure 3-2. Left: % error versus cycle for four rounds of AL&D at target value of 1000 J/K/mol Cp and 1000 kg/m3 ϱ. Center: calculated ϱ (blue circle) and Cp (orange star) on y-axis and predicted ϱ and Cp on x-axis for four rounds of AL&D (round indicated indicated by inset numbers, starting at one in top left and progressing clockwise). Right panel: the seven, fourth round structures and their associated ϱ and Cp error rates from the QSPR/NN prediction (compared to MD).

Observing the right panel of Fig. 3-2., the QSPR/NN tended to fall into one of two categories for C_p prediction: very accurate at around ~5% or greater than 20% error. It performed well on three of the octyl-sulfate paired cations accept when the cation contained a thiol group, otherwise the octyl-sulfate paired cations were structurally very similar. The last three structures returned two L-prolinate anions and one L-lysinate

anion. Detailed analyses on anion behavior related to QM/MD derived properties and relation to the GA are included in the SI.

Error Estimation

The consistency with which the AL&D strategy reduced error in just four cycles, and the high error rate in the first cycle compared to the QSPR/NN train/test set errors and MD errors, prompted us to explore factors that explain the error in designed cations. Two metrics correlated with error in QSPR/NN predictions: univariate kernel density estimates (KDEs) of the experimental data at the design target and the chemical similarity between the designed cations and the cations comprising the genepool. KDEs were calculated for the ϱ and C_p datasets using gaussian shaped deposits with bandwidths of 0.003 and 0.01, respectively. Around the scatter plot on the left portion of Fig. 3-3, the KDE for the entire C_p dataset appears on top, and that for the entire ϱ dataset appears on the right.

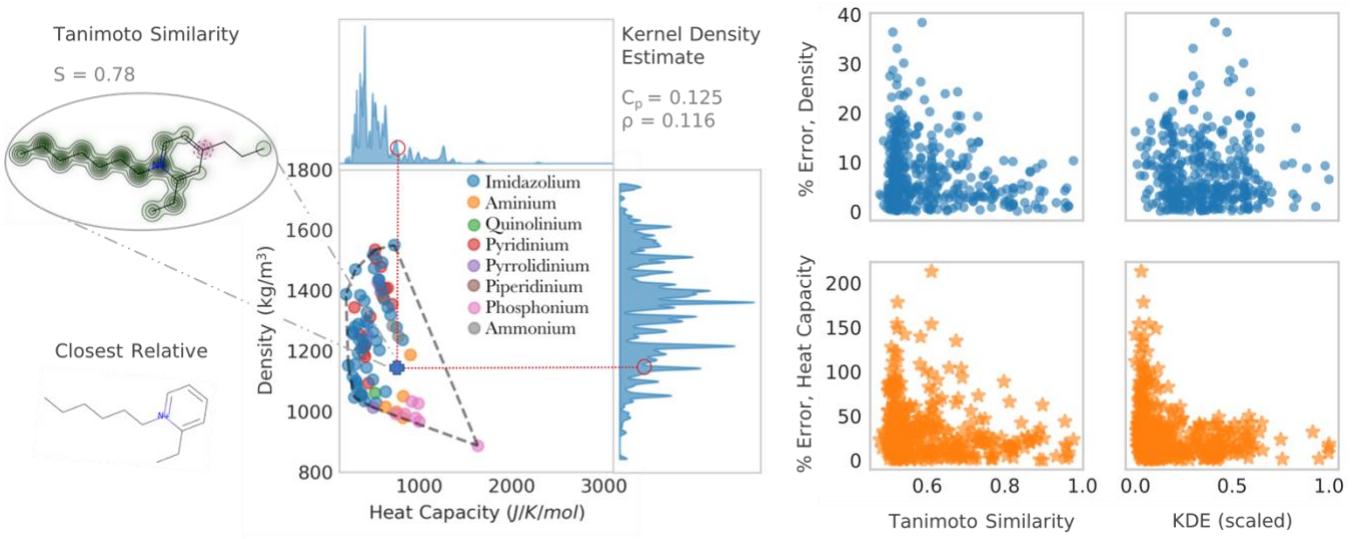


Figure 3-3. Left Panels: demonstration of calculation of the chemical (Tanimoto) similarity and univariate kernel density estimations for a molecular solution. Right panels: % error vs the Tanimoto similarity score of the molecular solution with its' closest chemical relative in the experimental training data and % error vs the univariate kernel density estimate.

Chemical similarities rely on a fingerprinting method, an abstract representation of the structures within a molecule, and a similarity metric. The cations returned in the cation-anion pairs were fingerprinted using an algorithm similar to that used in the Daylight fingerprinter[97] and compared using the Tanimoto similarity metric, defined as the intersection over the union of the two fingerprints. Fig. 3-3 gives an overview of how for a given cation-anion solution procured by the GA, a KDE for C_p and ϱ is calculated along with the highest Tanimoto score of the cation with its closest molecular relative in the genepool. In this example, the Tanimoto score of the designed cation is 0.78 and its closest relative is 1,2-ethyl-hexylpyridinium. The KDEs are 0.125 for C_p and 0.116 for ϱ, their method of evaluation indicated by the red dotted lines. The inner join of the property data forms the convex hull (CH), defined as the minimum area that comprises all data without forming a concave outer angle, and, in Fig. 3-3, appears as the grey dashed line perimeter. The property data

used to construct the CH are color-coded according to the IL class they represent. Incidentally, eight simplices, or linear edges, form the CH for this data.

The results of comparing KDEs and similarity metrics to QSPR/NN prediction error are shown in the right portion of Fig. 3-3. In the bottom panels, the upward bound of the C_p errors have a very clear trend with both the similarity scores and KDE. The ϱ errors have just as strong a correlation with the similarity scores and a weaker correlation with KDE. However, the ϱ errors always fall under 40% while the C_p errors rise as high as 200%. Observing the bottom panels in Fig. 3-3, the C_p error does not have a clear correlation with KDE or Tanimoto Score when only considering errors bellow 40%. In addition, the ϱ KDE in the <0.05 range is poorly sampled, with 5 points (vs 117 for the C_p data, 81% of which fall above 40% error). We therefore conclude that in future work, it would be worthwhile to investigate if further sampling in this low KDE range for ϱ would procure error rates as high as those observed in the C_p range. In any case, deciding whether to accept molecular structures with low KDEs and Tanimoto similarity scores can be evaluated based on an allowable level of uncertainty cutoff.

Breaking the Pareto Front

To evaluate our AL&D approach on a complex design problem, we tasked the GA with finding molecular structures that break the PF formed by the available experimental data. To do this, GA targets were assigned according to the simplices comprising the CH, Fig. 4. Nine separate GA instances, on separate compute nodes were launched, eight assigned to search within 10% of its' assigned simplex and a single GA instance assigned to the center of the CH, which incidentally is anywhere from 10-32% distance from any edge (this is after scaling the CH dimensions down to unity, i.e. the center of a square in this schematic is 50% away from any edge). A few of the solutions (purple) are beyond the targets (red) since the GA was allowed to come within 5% of a target to count as a successful search.

The GA found very few solutions outside the CH of the experimental data, especially where C_p is lowest. This is an interesting result for the following reasons: 1) the GA is operating on two, independent QSPRs/NNs (one for C_p and another for ϱ) that contain data well past the PF (Fig. 3, KDE estimates), and 2) the GA is sampling its molecular genepool from the outer join (union) of those two experimental datasets, i.e. it can sample structures that exist beyond the PF in terms of either one or the other property. This indicates that a real, lower boundary for C_p is incorporated by the model. In other regions of the PF as well, although to a lesser extent, inner solutions are returned far more often than outer ones. This indicates, as well, that a real PF boundary is incorporated into the model. In other words, the 1-dimensional models give natural rise to the 2-dimensional relationship between the two properties insofar as suggested by the experimental data. The green star icons in Fig. 3-4 are the calculated QM/MD properties for every predicted value returned from the QSPR/NN. Of the 271 solutions found by the GA, 18 were found beyond the PF. If we include the results from the 1000 J/mol/K – 1000 kg/m^3 search, a total of 36 ILs are found beyond the PF.

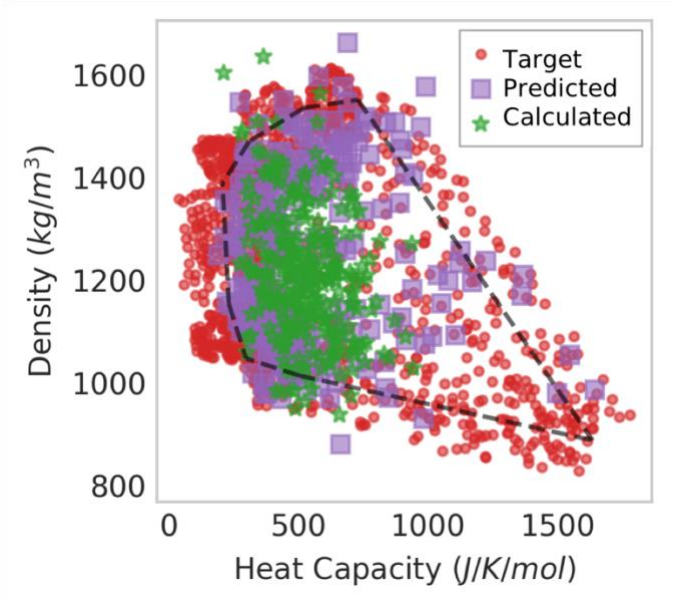


Figure 3-4. Convex hull assigned search task. Final property values from QM/MD calculations are in green (stars), predicted property values from the NN are in purple (squares), and targets set by the GA are in red (circles). The convex hull formed by the experimental data is indicated by the black dotted perimeter.

High-Performance Liquids

The top five highest calculated C_p values were 1143, 1054, 1031, 965, and 962 J/mol/K. Four of the Five were paired with octyl sulfate. At the experimental conditions used to train the QSPR/NN (99-102 kPa and 297-316 K) the highest C_p IL found by our method is greater than 96% of the training data. When the experimental data is filtered to the precise temperature and pressure of the MD simulations, only one of 168 experimental structures has a higher C_p than this IL. The top five highest calculated ϱ values were 1634, 1603, 1587, 1563, and 1542 kg/m^3. These ten ILs are presented in Fig. 3-5, along with charge and steric center based RDFs of these and comparative experimental ILs consisting of the same anion. In Fig. 3-5, blue lines indicate the positive-negative charge center RDFs, orange lines comprise the RDFs of anion-anion negative charge centers while green lines are for cation-cation steric centers—the ring bound nitrogens. The blue cation-anion RDFs are distinguishably higher than the other RDFs, indicating strong cation-anion charge center coordination. Anion-anion RDFs (orange) agree amazingly well between the GA and experimental systems, with the high C_p systems having slightly more intense initial peaks compared to the high ϱ systems. The cation-cation steric center peaks are similar across all systems—experimental and GA produced—with modest initial peaks and little to no second or third peak.

In terms of structural differences, the high ϱ ILs are highly branched, and have high nitrogen and oxygen content, whereas the high C_p ILs contain long carbon tails. High C_p ILs 2 and 4 are very similar, as well as 3 and 5, each group differing by 1 or 2 carbon differences in their sidechains. The cationic charge centers are much more dispersed in the high C_p group, with point charges around +0.13 vs the high ϱ ILs with point charges often greater than +0.5. Molecular configurations for all GA produced ILs are included in the SI

including an extensive center of mass (COM) RDF based evaluation on liquidity. In addition, relational databases with RDFs and IL metadata are available online as well as explanatory notebooks written in python.

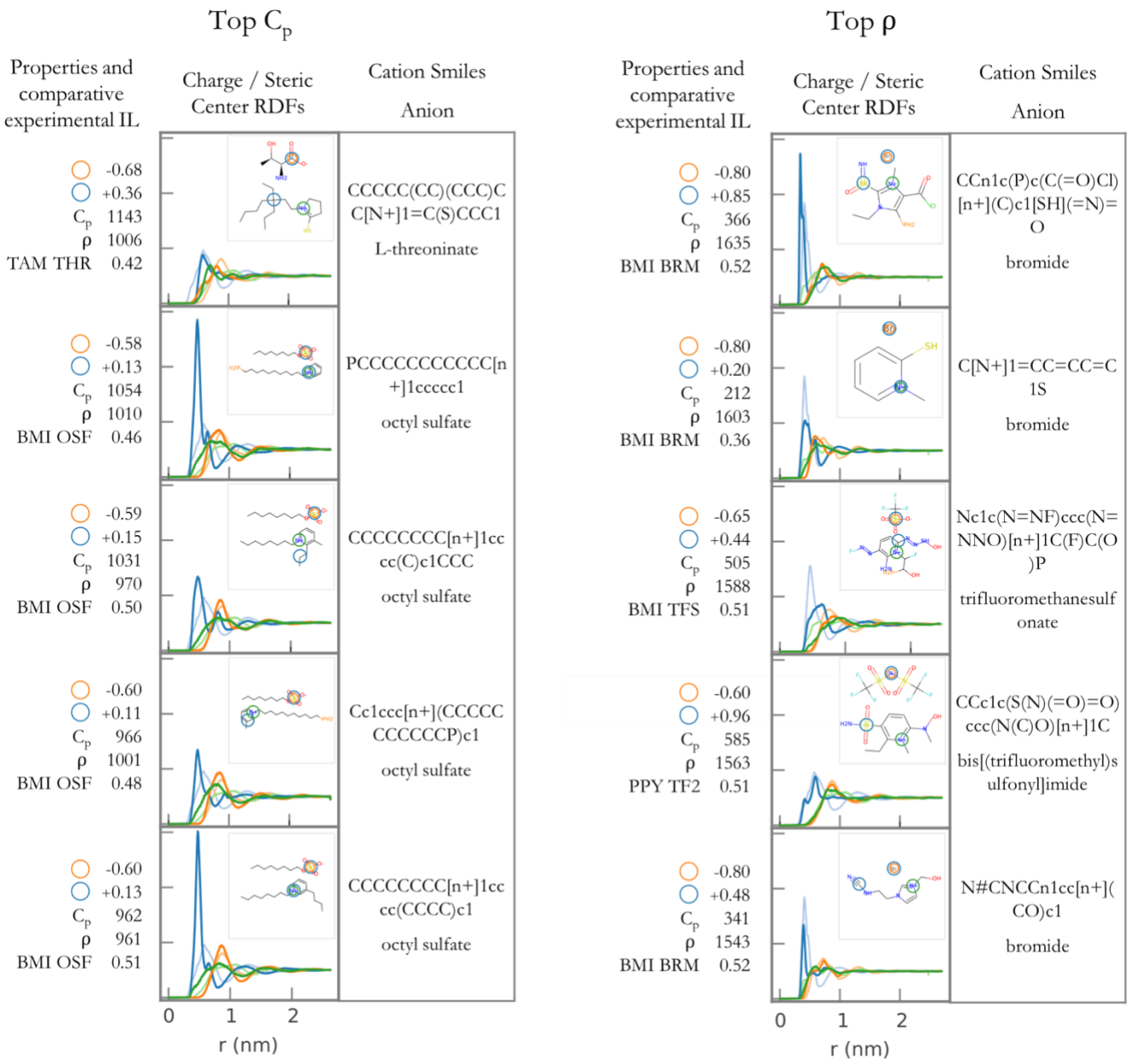


Figure 3-5. Left of center describes charge/steric centers, RDFs, structures, and comparative, experimentally synthesized ILs for the five highest C_p systems and right of center describes the same for the five highest ρ systems. Color coordination is according to the following. Blue lines: positive-negative charge center RDFs; orange lines: anion-anion negative charge centers RDFs; green lines: cation-cation steric centers—the ring bound nitrogen—RDFs; blue circles: location and value of positive charge centers; orange circles: location and value of negative charge centers; green circles: location of ring-bound nitrogens. All point charges are from QM calculations. Additional metadata is indicated left of each RDF panel: MD ρ and C_p values and six letter codes for comparison systems with similarity scores. TAM: tributylmethylammonium; THR: L-threoninate; BMI: 1-butyl-3-methylimidazolium; OSF: octyl sulfate; BRM: bromide; TFS: trifluoromethanesulfonate; PPY: 1-ethyl-2-pentylpyridinium; TF2: bis[(trifluoromethyl)sulfonyl]imide. Experimental IL RDFs appear in lighter hue with the same color coordination as described for the GA systems.

Conclusion

This work presented three major outcomes for AL&D ILs for specific densities and heat capacities: 1) multiple cycles of search, verify, and model, lead to overall lower prediction error rates for these systems, 2) chemical similarity and KDE are a proxy for QSPR/NN variance, and 3) a GA+QSPR/NN approach procures structures outside the experimental PF. In the first case, we showed that C_p started at an average 55% error rate and decreased to 18% by the fourth round while ρ started at 10% error rate and decreased to the irreducible error level of our validation method, ~2%, by the fourth round. In the second case we demonstrated how the upward bounds of error are correlated with a) the chemical similarity of the procured structures to that of the chemical structures within the training data and b) the univariate KDEs—evaluated at the point of the predicted property values—of the training data. This was especially true for C_p and we noted that in the case of ρ, the meaningful, lower ranges of KDE were not as effectively sampled and that this would be an interesting area of future work. In the third case, we demonstrated that by sampling along the CH formed by the experimental data, the GA was able to find 18 molecular structures beyond the PF. Only two out of 271 structures found during the CH sampling and none of 118 structures found during the multi-cycle AL&D rounds were found to be non-liquid as evaluated by their COM based RDFs.

In future work, and with the acquisition of additional data, an AL&D approach like this may be supplemented with continuous chemical representations of these materials, i.e. using GANs, VAEs, or reinforcement learning.[80,98,99] This is an exciting area of materials discovery as a continuous representation of the molecular structures allows for both gradient and stochastic based optimizations, i.e. we can solve harder design problems much faster. The major hurdle to overcome for liquids to adapt this approach is the acquisition of data. Rigorous physics-based simulations like MD can assist here as well. In this work alone, we generated 390 new IL systems, essentially doubling the amount of IL structural data we had at the beginning of the study. Additionally, there are methods of augmenting existing datasets that can be borrowed from natural language processing (NLP) and image analysis tasks. A recent study showed that SMILES enumeration can lead to amplified datasets and better model predictions.[100] Borrowing from NLP tasks, other studies have shown that a model trained on a large cohort of molecular data can be fine-tuned with molecules containing desired properties to achieve predictions oriented around the smaller dataset.[99,101] This approach could be applied here by training a general model using all IL data and then tuning it with IL data containing the property of interest. Nevertheless, GAs remain an important tool for finding fantastic materials: their ability to operate on discrete or continuous representations of structures, as well as their agnosticism toward their fitness function, make them—like their evolutionary analog—extremely flexible.

Methods

QSPR/NN and underlying data

The features for the cationic and anionic moieties of each IL data point were concatenated, along with the experimental temperature and pressure to form a 190-length feature vector. The feature vectors were scaled and centered to zero mean and unit variance. The target properties were taken as the natural log values. Both NN models consisted of two fully connected 100 node layers with rectified linear unit activations and a single

output node with a linear activation. The Adam optimizer was used during back propagation with mean squared error as its accuracy metric. 50% dropout was used to avoid overfitting and early stopping was invoked when accuracy ceased to improve. 20% of the data was reserved for testing. Models were developed in Keras[102] with a Tensor Flow backend. The QSPR/NN models were used to direct the GA in subsequent searches.

QM/MD

In line with the typical level of theory used for estimation of partial charges within the RESP framework,[103] and our prior IL property prediction work,[71] a structural stability test was performed using geometry optimization in the Gaussian 09 software package[104] using HF/6-31G(d) level of theory. At minimum, the geometry had to converge to a stationary point with zero imaginary frequencies for the structure to be passed along to MD for property valuation. Throughout the course of this study approximately 1,300 such calculations were performed on potential target molecules. The generalized amber force field[105] (GAFF) was used to determine all force constants and equilibrium and Lennard-Jones parameters. Electrostatic point charges were assigned by the RESP method using the software program Antechamber.[103,105] Point charges were scaled by a factor of 0.8, a common procedure for IL systems in MD.[71,106–108] ACPYPE and tleap were used to generate input files for each ion.[109,110]

Using Packmol, 125 nm^3 cubic boxes were packed with ILs at 80% of the predicted QSPR/NN ϱ.[111] This decreased ϱ was used to facilitate rapid packing of an initial structure and minimize errors from structure generation. This also permits for further structural relaxation during the subsequent NPT equilibration step where ϱ rapidly equilibrates. Above 1.0 nm particle-mesh Ewald (PME) summations were used to calculate long-range electrostatic interactions and below 1.0 nm electrostatics were calculated explicitly. At 1.2 nm, van der Waals interactions were shifted to 0. The GROMACS[112] engine was used to simulate the IL boxes with periodic boundary conditions. Constant temperature of 298.15 K was maintained with a global stochastic thermostat.[113] After energy minimization, each simulation underwent 5 ns of pressure equilibration in the NPT ensemble using the Berendsen barostat.[114] After pressure equilibration the IL boxes underwent 5 ns of NPT production runs using the Parrinello-Rahman barostat.[115] The ion-ion radial distribution function (RDF) based on COM, ϱ, and C_p for each IL were calculated from these production runs using standard tools within GROMACS. Due to the intensive nature of indexing charge/steric centers from the QM data, RDFs based on these atomic sites (similarly computed with standard tools in GROMACS once indexed) were performed only for the top systems presented in Fig. 5 of this work, but this analysis could be automated in the future.

GA

A simple GA is comprised of a fitness criterion (in this case the QSPR/NN models) and a genepool from which to sample starting structures and a set of mutation options. During GA searches, the genetic line was terminated if: a) the cation-anion pair came within 5% of the target property values or b) 1,000 unsuccessful mutations had occurred since the last accepted mutation. Upon a terminated genetic line, the GA would either a) resample a cation-anion pair from the original genepool or b) reassign the property targets within a predefined range and then proceed to a). The choice between a) or b) was determined by the design parameters initialized before the search and are discussed in greater detail in the SI. Successful cation-anion pairs

underwent a simple geometry optimization to initialize atomic coordinates followed by a UFF[116] energy minimization using the RDKit package.

Chapter 4 Continuous Molecular Representations of Ionic Liquids for Enhanced Design[4]

Introduction

Deep learning is accelerating the chemical discovery pipeline: from machine learning based generative modeling to machine learning accelerated simulations to highly accurate quantitative structure property relationships (QSPRs).[96,117–120] Since 2012, advances in GPU-accelerated training and regularization tricks like dropout, have put deep learning at the forefront of the molecular design toolkit.[121] This accelerated training comes on the heels of back propagation, the algorithm by which connectionist-type learners can appropriate blame in their network weights, and therefore achieve gradient based solutions to mastering their training data. Indeed, the similarity of weighting interconnected neural layers to traditional graphical processing tasks has led to the creation of specialized hardware for those purposes.[122] Further, modular, pythonic libraries like Keras and TensorFlow, have made appropriation of deep learning in the domain sciences very convenient for respective field experts. Lastly, the ability of deep learners to embed discretized objects into a continuous space, further opens up the design paradigm into gradient-based-solution Utopias. We explore one such method of embedding and stochastic generation in this work, within the material ecosystem of ionic liquids (ILs).

Despite the promise of deep learning tools, a huge challenge remains. Deep learning models—typically categorized as neural networks (NNs) with three or more layers—often contain hundreds of thousands, if not millions, of parameters, i.e. they are extremely data hungry.[123] Recent works have investigated how transfer learning or analogical learning might be better utilized in these types of networks to overcome this challenge.[101,124–128] Goh et. al developed ChemNet, a deep neural network (DNN) first pre-trained on molecular descriptor labels (weak supervised learning) and then trained, using transfer learning, on smaller datasets to predict molecular properties. Also relevant is the work of Bombarelli et. al, in which they developed a model comprised of a variational autoencoder (VAE) trained simultaneously with a neural network to both generate and predict the property value of drug-like molecules. In this work, we show that a small molecule database, GDB-17, can be leveraged in a VAE[129–131], to embed cations and anions that can then be used to generate ionic liquid structures. Additionally, we show that this latent space can be exploited with mathematical relationships between embedded structures to procure molecules with desired features.

IL design is a challenging task and there is dedicated material to discussing the details of these challenges elsewhere.[132] Briefly, in the design of ILs, limiting factors include the computational cost of physics-based simulations such as Molecular Dynamics and Monte Carlo (MD/MC). Especially for properties that are electrostatically driven, accurate property calculation takes computationally prohibitive time scales and proper tuning of force field parameters, often derived incrementally and painstakingly.[84] Lastly, IL data is scarce compared to what is typically prescribed to deep learners. Intuitively, this corresponds to a more scarcely

[4] Reproduced in part with permission from Wesley A. Beckner, James Lee and Jim Pfaendtner. Continuous Molecular Representations of Ionic Liquids for Enhanced Design. Work in progress.

populated latent space from which structures are generated and their properties predicted. The lack of training examples means that the network may not be able to scale accurately to unknown data. Although transfer learning has been investigated as a way to overcome data scarcity,[124] they can suffer from amnesia, forgetting generalizations learned in the initial phases of training.[99] In this work, we investigate a novel training schedule to retain as much general chemical knowledge as possible within the network, while also fine-tuning the model toward our specific design task.

Generative modeling requires a decision on molecular representation. Recently, great progress has been made in representing molecules as molecular graphs, where atoms and bonds make up nodes and edges in a connectivity matrix.[133,134] The main advantages of this approach are that representations are invertible (every graph is associated with a single molecule) and unique (every molecule is associate with a single graph). While this certainly reduces the amount of function calls to generate a unique molecular candidate, it is not necessary for the generator to learn 'true' chemistry. Indeed, some strategies for molecular embedding have even oriented around tasking the generator with learning the relationships between various types of molecular representations.[135] In this work, we have the added challenge that each IL 'material' is constituted as a pair of individual ionic species—the cation and the anion. Herein, we explore how these two distinct moieties can be represented in a latent space and use SMILES annotation to represent their individual structures.

SMILES, developed by Weininger[136] and Daylight Chemical Information Systems, is an ASCII character string representation of a depth-first search of a molecule's graph. Uniqueness refers to when a molecule is associated with single representation. Invertedness refers to when a representation is associated with a single molecule, a one-to-one mapping. A molecule may be invertible but non-unique if it is physically asymmetrical by translation or rotation, or if it varies with permutation of atomic indexes. When represented in 2D, such a molecule may have different representations (non-unique). All these representations, however, still refer to the same molecule (invertible). A non-unique string dataset can be made unique by the process of canonicalization, which determines which of all string representations of a molecule will be used as the reference for its molecular graph. Certain canonical string representations such as IUPAC's InChI[137] exist, and while canonicalization algorithms can be applied to SMILES,[138] no standard method exists. This being said, SMILES is the native encoding for many large databases such as ZINC,[139] ChEMBL,[140] and GDB-17[141] which we use in this paper, making it a convenient choice.[120] Since it is a string sequence, one can take advantage of sequence-based deep learning models such as VAEs, which have found success in natural language processing (NLP) and recently in molecular generative models for drug-like molecules.[80] By that same token, the SMILES string is fragile, meaning a small change in syntax can result in a chemically invalid molecule. For this reason, a method of guiding the search through chemical space is required. To validate our VAE outputs, we appropriate an RDKit[44] sanitization step that checks atomic valency.

A goal of a generative model is to explore the chemical space, achieve optimization of a single molecular property, or multi-objective optimization of a set of molecular properties. This is oftentimes difficult, and instead of finding a global optimum of the objectives, a set of solutions known as the Pareto set, or Pareto Frontier, is found.[85,142] The question then is whether the generator can effectively create molecules within the Pareto set. In envisioning a search through chemical space, a discrete local search method is constrained to the molecules allowed by the heuristics defined in the mutation. This may be a very small portion of the entire

chemical space. Furthermore, in the absence of sophisticated fitness schedulers such as simulated annealing, discretized search methods can become locked in local optima, unable to find the global Pareto set solution. In order to traverse the entirety of chemical space and optimize over it, we seek a continuous, gradient-based generative model.

To the end of a gradient-based search method, the VAE has seen success in natural language processing for string sequences and has also been recently used for the generation of drug-like molecules. It was therefore an attractive model for the chemical design of IL materials. A VAE is comprised of two neural networks: an encoder and decoder. The encoder converts one-hot encoded SMILES strings into a vector representation, and the decoder converts the encoded string back into a SMILES string. The output of the encoder, and input of the decoder, is of low-dimension compared to the dimensionality of the hidden layers of the encoder and decoder. For this reason, it is known as a bottleneck layer. In training, the model must learn to represent data as best as possible in this bottleneck layer, learning some representation of the key features of a molecule. This is not unlike the process of dimensionality reduction in principal component analysis. The vector representation of the encoded string is then known as its latent representation, a single point in the latent space. By using the latent representation of a string as the input to a QSPR neural network and training the entire model to include the loss from the QSPR predictor, as well as the reconstruction loss and KL divergence from the VAE, one can organize the latent space in relation to both structure and property. We explore the utility of the latent organization in the second and third sections of this work.

Results and Discussion

Transfer Learning Approach

To evaluate the effectiveness of our transfer learning protocol, we created five models using identical architectures and training data yet different training protocols. After training, the models were tasked with generating structures using 1-butyl-2-methylpyridinium as a cation latent seed (one of the 276 cations in the cation dataset). If after 10,000 sample attempts the model was unable to procure a new and unique, RDKit-sanitizable structure, the search was terminated. The number of returned structures with a positive charge are indicated in Fig. 1.

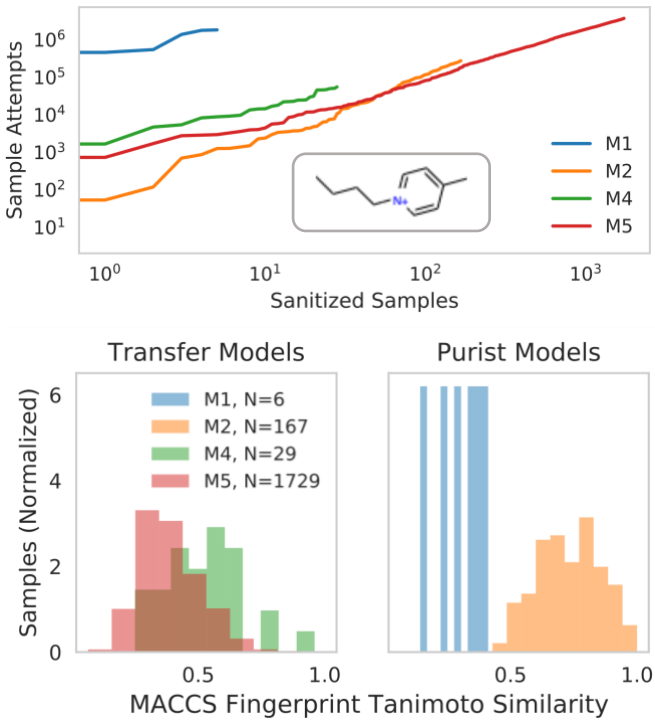


Figure 1. Top: log-log scale sample attempts vs number of RDKit-sanitized structures with a positive charge (also indicated by N in bottom, right panel). Sampled from a single latent space cation seed (inset). Bottom: Tanimoto similarities of MACCS Fingerprints of procured structures compared to cationic seed. Lower values are more dissimilar from the seed and broader histogram distributions contain more structural variety.

In addition to procuring the second highest number of RDKit-sanitized structures, M5 produced structures with greater MACCS variety than its companion transfer learned model, M4. Relaxing charge criteria increases the total number of successful hits for each model, Table 1. Otherwise M5 clearly achieves the best performance for generating cationic candidates for ionic liquids. The transfer learned training protocol for M5 was therefore identified as the training protocol to create dual cation-anion generating VAEs.

Table 1. Total samples generated from each model with single cation seed, 1-butyl-2-methylpyridinium.

Model	Samples	Samples with (+) charge
M1	7122	6
M2	170	167
M3	6	0
M4	43	29
M5	3700	1729

Sampling from Dual Latent Spaces for Target Properties

Three dual cation-anion VAEs were trained following the training protocol of M5, their training histories are presented in Fig. 2. The Gen3 model, with completely separate networks for the cation and anion, achieved high reconstruction accuracies in almost every epoch of training. Gen1 and Gen2 failed to improve in the first million samples of GDB17 data (Fig. 2, up to first dotted line), and only marginally improved during the mixed and pure salts training rounds (after first and second dotted lines in Fig. 2). Of note, Gen1 and Gen2 were able to achieve modestly high accuracies for anions in their final rounds of training. This is due to the anion dataset being much smaller than the cation dataset (98 vs. 276) i.e. the VAE was able to memorize the anion structures, but unable to generalize across molecular entities.

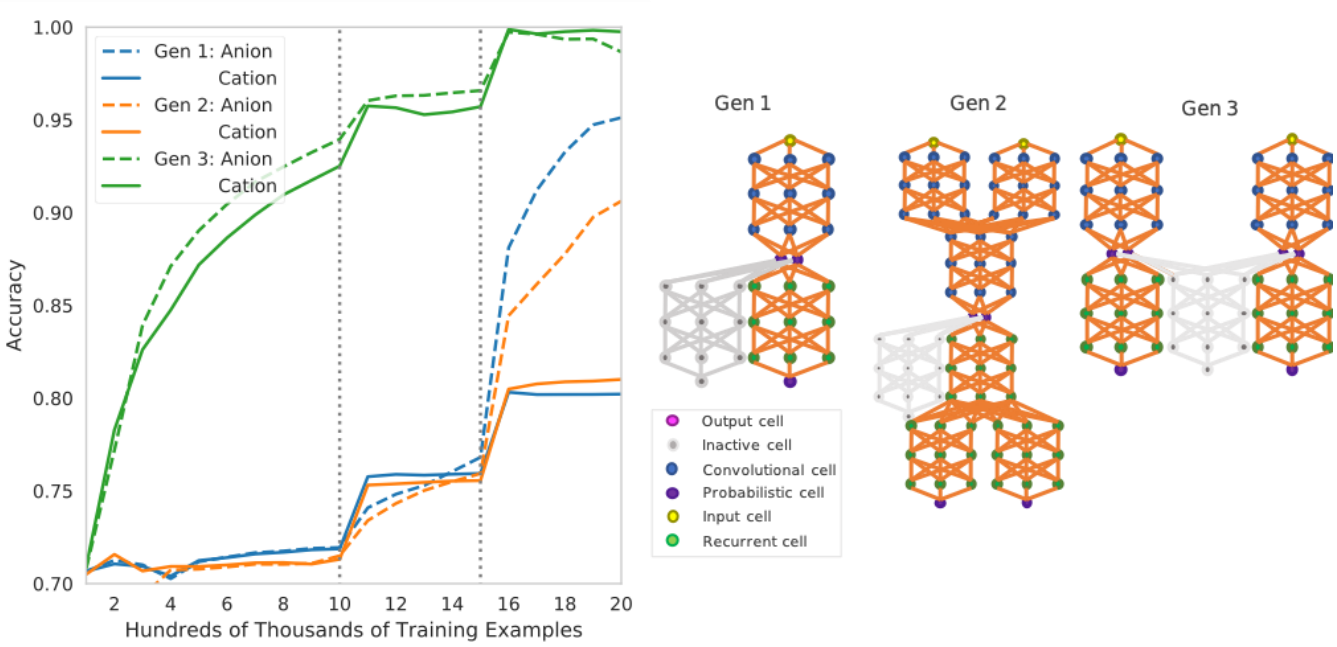


Figure 2. Training histories for models Gen1, Gen2, and Gen3. Training protocols are the same as M5. Training accuracies for Gen1 and Gen2 did not appreciably improve with 1 million GDB-17 examples.

The purpose of exposing our models to GDB-17 data before IL data, is to embed within the network rudimentary chemical understanding, insofar as to be able to recreate SMILES annotation whilst dealing with noise (stochastic embedding) and information loss (bottle necking in the latent space). Interestingly, during the third phase of training, the Gen3 model assigns cation types to specific neighborhoods within the latent space, Fig. 3.

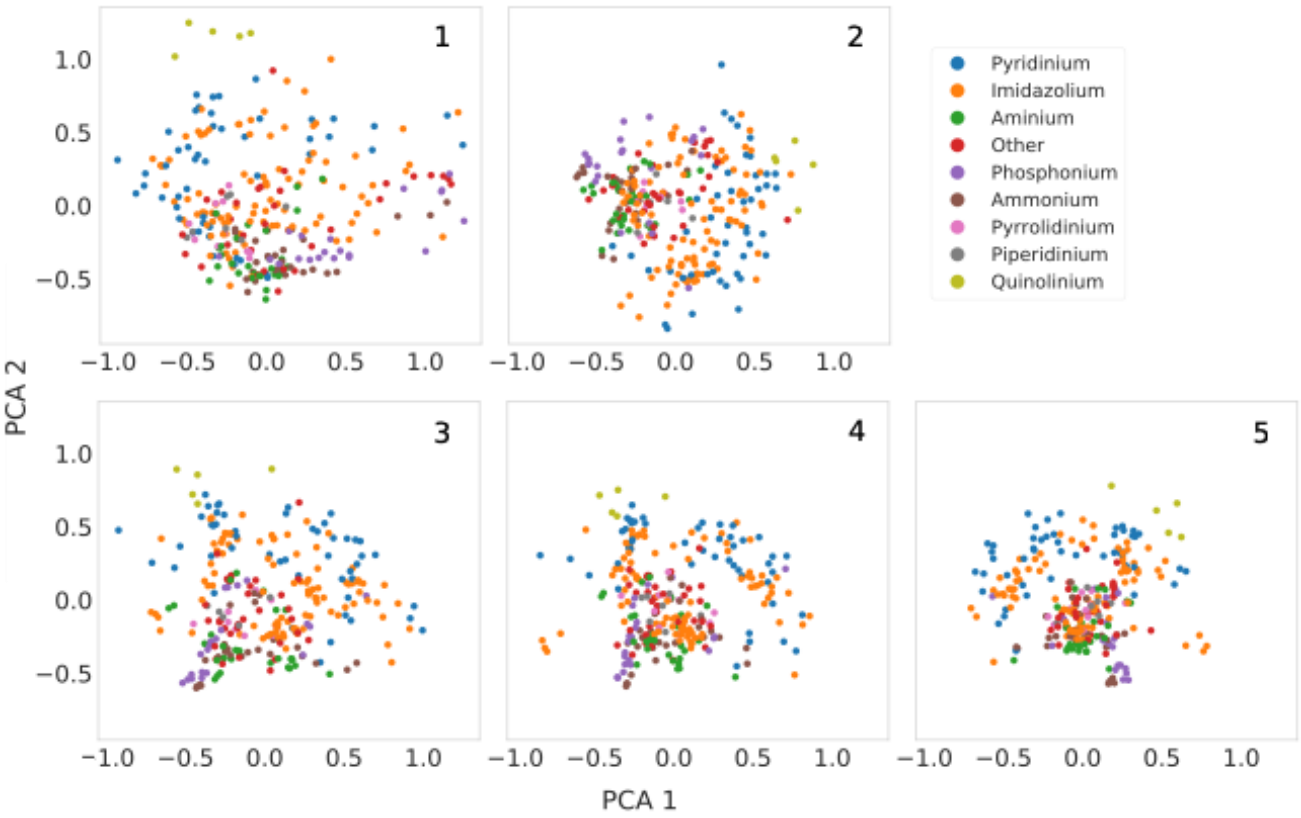


Figure 3. First two principal components during phase III salt embeddings at every 100,000 training examples. Inset number indicates 100,000th training step.

There are clear regions designated for the pyridinium and imidazolium type cations, for which there are ample training data. The quinolinium type cations, have a distinct region as well and the remaining cation types cluster together around the origin of the principal components. Also of interest, from the first to the fifth 100,000 training examples, the quinolinium type cations appear to migrate together in their latent embeddings. This is without exposing the VAE to any kind of 'type labeling'—the VAE is learning for itself these structural categories that have been ascribed by researchers.

To confirm whether these models—Gen1, Gen2, and Gen3—would be useful for generative purposes, we tasked them with generating unique structures as in the case of the cation generator, albeit this time allocating the entire salt database as seeds for the respective latent spaces and allowing the models however many function calls needed to procure 100 unique structures. Gen1 produced 100 structures in 4,073 function calls, Gen2 in 13,968, and Gen3 in 2,597. As forecasted by their poor recreation accuracies however, the Gen1 and Gen2 models achieved low structural variety in their procured candidates. Their molecular returns were typically long, branched and unbranched alkyl chains. The Gen2 model, however, did achieve some greater structural variety than the Gen1 model: while it was slower at creating RDKit-sanitizable structures, when it did procure a structure, it was more chemically meaningful, containing functional groups learned from its training data that were not present in the Gen1 model. Due to its ability to create both heterogeneous

structures and at a low function call level, Gen3 was selected in subsequent QSPR trainings for each of four properties: density, heat capacity, viscosity, and thermal conductivity, the training and validation set histories are presented in Fig. 4.

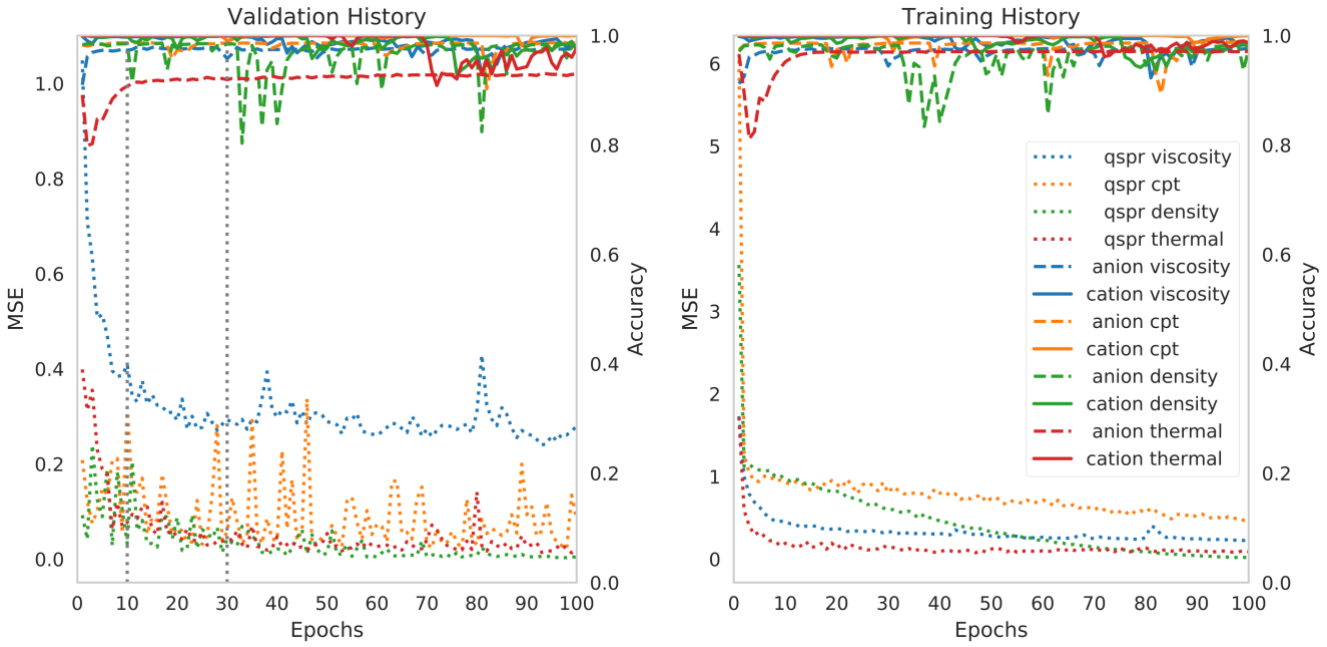


Figure 4. Validation and training set histories for QSPR training on Gen3.

Models were saved at 10, 30, and 100 epochs. Gen3 QSPR validation sets did not appreciably improve after 10 epochs and did not improve at all after 30. Traditionally, a model would be selected somewhere between epoch 10 and 30 to avoid overfitting. In this case, however, a model with the best generative capacity was desired, which didn't necessarily correlate with its predictive ability. Indeed, the QSPR training served to redistribute the placement of chemicals embedded in the latent space, to better navigate it according to the premise that like structures lead to like properties. To validate this, all three QSPR models for each property, were compared against the original Gen3 VAE in their ability to procure salts with property values that bordered on their respective distributions. The target properties were:

1. High heat capacity (> 918 J/mol/K)
2. Low density (< 962.7 kg/m3)
3. Low viscosity (< 0.0106 Pa s)
4. High thermal conductivity (> 0.1667 W/m/K)

Each of these property targets represent the property value of the 10th best ranking salt in the respective distributions. A separate QSPR was used as the property predictor and is described further in Methods. In effect, using a separate QSPR predictor, isolates whether the QSPR training served to better organize the latent space for mining of desired properties. The 10 best salts in each property category were used as seeds

in the function calls. The total number of function calls for each QSPR-VAE and the baseline Gen3 VAE are presented in Fig. 5.

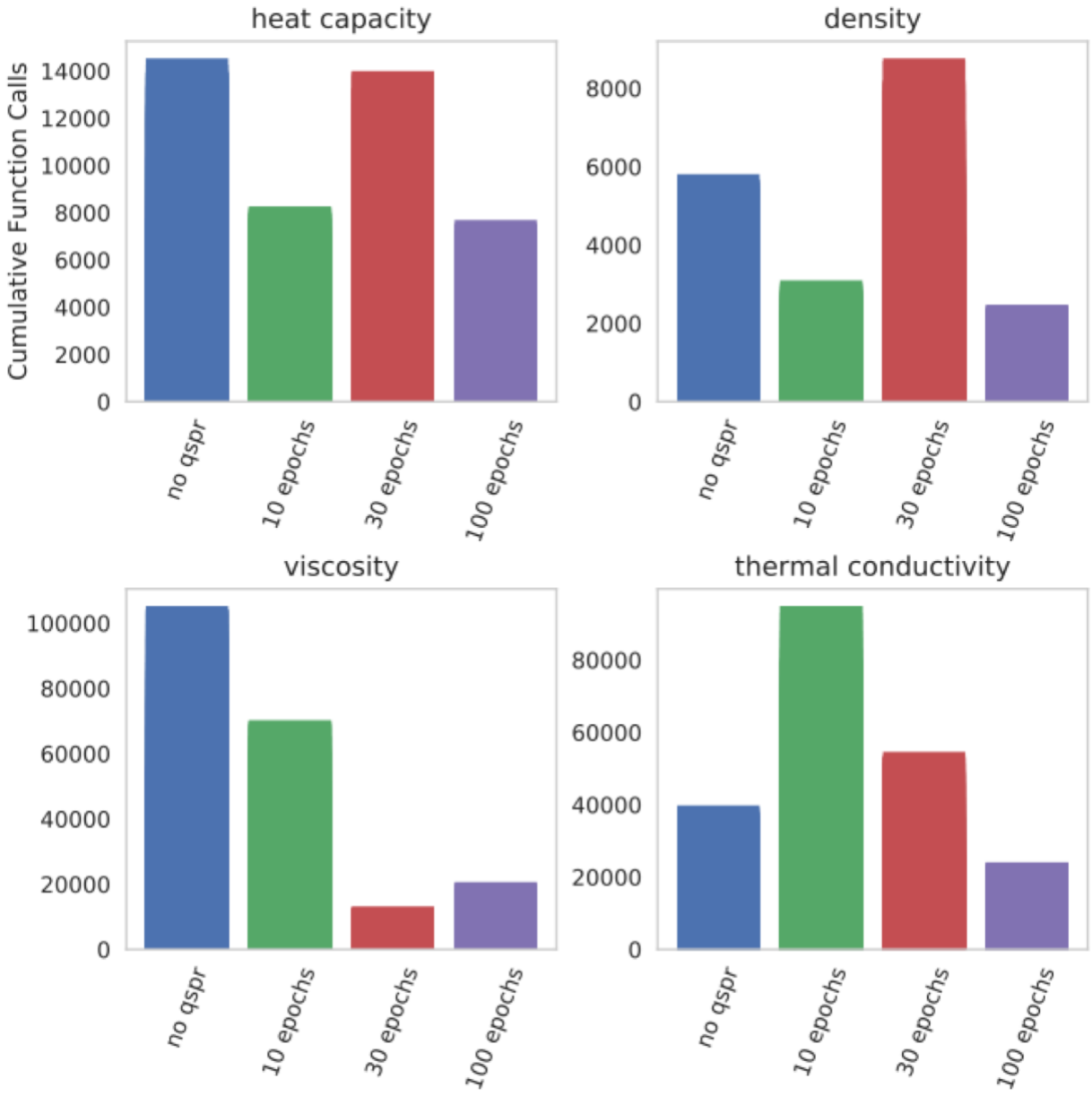


Figure 5. Cumulative function calls for each property and VAE/VAE-QSPR model to create 100 unique structures not within the training datasets with target property values.

We observe that the VAE with 100 epochs, on average, performs the best out of the VAE models. This suggests that by grafting a QSPR predictor onto an existing generative model, the latent space – previously organized by a purely structural relationship – is reorganized by the goal of minimizing QSPR loss. By using the top 10 ILs as seeds to generate from the latent space, we leverage the QSPR-related organization, but

without explicitly calculating a gradient. In the next portion of this work, we investigate how calculating the spherical linear interpolation (SLERP) between ILs embedded in the latent space can lead to new ILs with combinative macroscopic properties.

Interpolating in the Latent Space for Combinative Properties

A distinct advantage of designing in a continuous structural space, is the arbitrary appropriation of mathematical relationships between vector-represented molecules to procure new ones. To demonstrate, we take the Gen3 VAE and interpolate between two embedded ILs with desirable properties, Fig. 6. When designing a heat transfer fluid, we might desire an IL with high heat capacity and thermal conductivity. In this search, a high heat capacity IL was selected with a high thermal conductivity IL, which appear at the top and bottom of the right panel in Fig. 6. After interpolating between them for 10 distinct structures, a candidate was found that had heat capacity and thermal conductivity estimates within our dictated cut-off (higher than 918 J/mol/K and 0.1667 W/m/K).

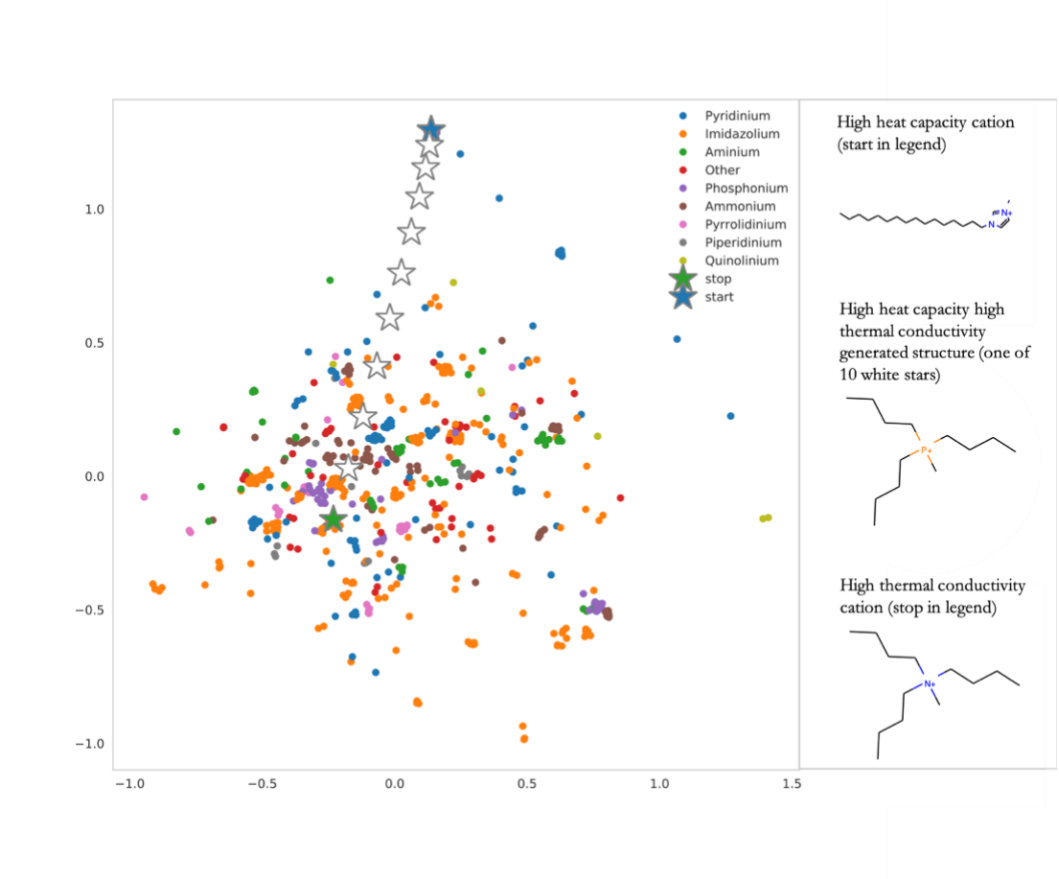


Figure 6. Latent space interpolation in the Gen3 heat capacity-thermal conductivity 100 epoch QSPR model. In the interpolation (represented by stars) the blue star indicates a high heat capacity salt in the training data, green star indicates a high thermal conductivity salt in the training data.

To quantify whether interpolation is a convenient search mechanism, we tallied the total number of function calls to procure 10 structures in each of the pairwise property profiles, Table 2. In each search iteration, the engine randomly selected one of the top 20 highest heat capacity ILs and top 20 thermally conductive ILs and interpolated between these structures.

Table 2. Total function calls to procure 10 candidate IL materials within the specified property targets.

Properties	Interpolative Function Calls		Noisy Seed Function Calls	
	Without anion pairing	With anion pairing	Without anion pairing	With anion pairing
Heat capacity, thermal conductivity	13198	234	1438	340
Heat capacity, viscosity	86288	5817	38084	1006
Heat capacity, density	11727	467	604	27
Density, thermal conductivity	2037	712	1161	208
Density, viscosity	161118	15900	66597	2263
Viscosity, thermal conductivity	14920	1599	19334	2260

Sampling from top 20 performers for each property category with a noise factor outperformed the interpolation strategy. However, when procured candidates were evaluated against all anions in the training datasets to evaluate valid anionic partners for the given property distribution target, the interpolation method performed better than the noisy seed strategy for generating high heat capacity/high thermal conductivity targets and low viscosity/high thermal conductivity targets. The reason evaluating against all anion partners results in lower function calls is that the cations are often promiscuous—they can be attached to a number of anions and still fit within the target property profile.

Conclusion

In this work, we have demonstrated the following objectives: (1) transfer learning is an effective approach to create a generative neural network model of molecule types for which there is scarce training data, (2) training the preconditioned-structural model on subsequent property data can lead to effective reorganization of the latent space for generating molecules with desired properties, and (3) interpolating between molecules of property extremes can result in hybrid-generated structures with structural and property similarities to the two endpoints. For the first objective, we evaluated our training protocol against *null hypotheses*: (a) training a model on only the larger, but molecularly dissimilar dataset and (b) training the model on only the smaller, target dataset. Both null hypothesis protocols resulted in poor structural variety when sampling from the generated model and/or minimal RDKit-sanitizable returns. In the second objective, we tasked our generative models with producing 100 RDKit-sanitizable structures at fringe property distributions (i.e. a non-trivial design task) and found that QSPR-trained models procured structural candidates with fewer function calls than the non-QSPR trained model. Finally, for the third objective, we outlined a multi objective design problem, where we sought fringe property distributions using 6-pairwise property combinations. In these cases, interpolating between top performers from each distribution did produce RDKit-sanitizable structures, but at about the same performance as sampling from both distributions with noisy seeds.

Methods

The Variational Autoencoder

The variational autoencoder (VAE) has seen success in the generation of images[143] and has also been recently used for the generation of drug-like molecules.[80] It was therefore an attractive model for the chemical design of ionic liquid materials. A variational autoencoder is an autoencoder with added stochasticity.

An autoencoder is comprised of two neural networks: an encoder and decoder. The encoder inputs data input x and outputs a latent (hidden) representation z, which is of much lower dimensionality than x. The decoder takes the latent representation (a vector) z and outputs a prediction $\hat{x}$. In training, both the data and target label are the input x, so that the model learns to predict its input. The significance lies in the low dimensionality of the latent representation z, which learns the most important features of the input in order to make accurate predictions. The compression of x to z is like the process of dimensionality reduction in principal component analysis, wherein similar data inputs will be located close together in latent space.

In a variational autoencoder, the input x is no longer mapped to a single vector in latent space. Instead, it is mapped to a distribution over latent space. This distribution is the values of z from which x could have been generated. This takes the form of a vector means and vector of standard deviations, wherein the ith elements refer to the mean and standard deviation of the distribution corresponding to the ith data point. The decoder samples from this distribution a value z and outputs a distribution of values of x that z corresponds to.[131,144]

In this work, we use the variational autoencoder to generate molecular structures. We are also interested in the properties of these generated structures. By using the latent representation of a string as the input to a QSPR neural network and training the entire model to include the loss from the QSPR predictor, as well as the reconstruction loss and KL divergence from the VAE, one can organize the latent space in relation to both structure and property. The importance of latent space organization lies in sampling new structures. In sampling from the latent space, samples closely related in terms of structure and chemical property will be close together in latent space. If one desires to generate a structure with high density, one could then use the latent representation of a known molecule with high density, specify a temperature (meaning a distance from that point) and begin sampling. If one desires to generate molecules with structure and property between two known molecules, one can sample from the interpolation between these two points in latent space. In this work, we use SLERP interpolation.[145]

Transfer Learning Protocol

Our salt database consisted of 688 unique entries comprised of 276 unique cations and 98 unique anions. Salts were taken from ILThermo[25,146] experimental entries for properties: density, heat capacity, viscosity, and thermal conductivity. Three transfer learning and two non-transfer learning (purist) protocols were instantiated using a VAE structure developed previously whereupon small, drug-like molecules were generated.[80] For our transfer learned approach, molecules of the cationic moiety were targeted. The training protocols are described in Table 3.

Table 3. Cation VAE training protocols for models M1-M5.

Model	GDB17 Training Samples	1:1 GDB17:Bootstrapped Cation Training Samples (N=276)	Bootstrapped Cation Training Samples (N=276)
M1	-	-	250,000
M2	-	-	1,000,000
M3	1,000,000	-	-
M4	1,000,000	500,000	-
M5	1,000,000	500,000	500,000

Sampling from Dual Latent Spaces for Target Properties

Three salt models with dual cation-anion input-output were generated in order to procure latent space(s) with which to feed a QSPR model, Fig. 2. As a null hypothesis, Gen1 consisted of the Aspuru-Guzik[80] architecture, albeit with the first and final layers receiving two, and outputting two, SMILES strings for the cation and anion, respectively. Gen2 contained the same structure as Gen1 with the exception that the cation-anion inputs fed into three independent convolutional (CONV) layers before feeding into three combined CONVs and the third gated recurrent unit (GRU) in the decoder fed into two separate branches of three GRUs for each of the cation and anion. Gen3 VAE consisted of two separate Gen1 architectures. Training protocols for the three architectures followed the same protocol as M5. All three models consisted of the same QSPR structure described in previous work,[73] with the alteration that the input to the QSPR consisted of the respective latent spaces.

To test the usefulness of the VAE-latent space search approach, the Gen3 model was saved at 10, 30, and 100 epochs for each of the four QSPR training sessions. To simulate a real world design criterion, each of the four properties was designated as a value to maximize or minimize. For heat capacity and thermal conductivity, the top 10 cations corresponding to the *highest* values for the respective properties were taken as seeds for the VAE-QSPRs. The models were then tasked with returning 100 salts with property values that were equal to or *higher* than the experimental salt values. The same was done for density and viscosity with the alteration that the *lowest* values for the respective properties were taken as seeds and models were tasked with finding salts with equal to or *lower* than these values. These four VAE-QSPR tasks were repeated with the non-QSPR-trained Gen3 model to highlight whether the reorganization of the latent space improved the ability of the generator to find desired values. In order to compare the VAE-QSPRs with the original Gen3 model, a separate RDKit-based QSPR model similar to that described in previous work[96] was used as the property evaluator. We compare the performance of each VAE with increasing training of the QSPR predictor. In this comparison, the cations and anions seeded for the VAE were the top 10 salts for each property

The generators tested are as follows:

1. VAE without QSPR.

2. VAE with 10 epochs of QSPR training.
3. VAE with 30 epochs of QSPR training.
4. VAE with 100 epochs of QSPR training.

The 13 generators (3 QSPR-VAEs specific for each property (4) + the original Gen3 model) are tasked with generating 100 salt structures with the target property, and the number of function calls required was recorded.

Structures generated were passed as input into the same, separate QSPR-RDKit predictor neural network trained on experimental data, and properties were predicted. In this way, we test only the generative performance of each model.

Interpolation in the Latent Space

Calculating distances in high dimensional feature spaces is non-trivial. Often, we will have to actively avoid pockets of data-scarce regions in the VAE. Specifically, sampling from a spherical linear interpolation (SLERP) between embeddings rather than a linear one, helps prevent divergence from a model's prior distribution (the distribution in the latent space). Indeed, SLERP is just one of a handful of sampling techniques used in high dimensional latent space models.[147] In other areas of generative modeling research, SLERP has been used to demonstrate that the model has not simply memorized training data, but can extrapolate outside known examples.

SLERP is a method to interpolate between two vectors along the shortest arc.[145] It can be thought of as the shortest path along a spherical geodesic. More specifically, in the context of unit vectors (which we can extend to any vectors by normalizing), it is the interpolation between two unit vectors along a unit-radius great circle arc centered at the origin, with constant-speed (angular velocity) motion.

Originally developed for the purposes of quaternion interpolation for 3D animation, SLERP can be defined and used independently of quaternions, and beyond their dimensionality (4D vectors along a 4D hypersphere). In the context of this paper, we interpolate between n-dimensional vectors along a n-dimensional sphere, where n is the dimensionality of the latent space.

The formula for SLERP interpolation between two vectors p_0 and p_1(normalized) is independent of the dimensionality of the space in which the arc is subtended, and depends on an interpolation parameter t between 0 and 1, as well as Ω, the angle subtended between the points such that $\cos(\Omega) = p_0 \cdot p_1$, the n-dimensional dot product:

$$SLERP(p_0, p_1; t) = \frac{\sin[(1-t)\Omega]}{\sin(\Omega)} p_0 + \frac{\sin[t\Omega]}{\sin(\Omega)} p_1$$

The interpolation t dictates the point on the arc that the interpolation is set to. For example, $t = 0.10$ refers to a point 10% of the way from p_0 to p_1, such that the angle between the interpolated point and p_0 is 0.1Ω, and the angle between the interpolated point and p_1 is 0.9Ω. Changing t at a constant rate results in a constant

angular velocity along the arc. One will also notice that in the limit of $\Omega \to 0$, the formula or SLERP reduces to that of Linear Interpolation (LERP):

$$LERP(p_0, p_1; t) = (1 - t)p_0 + tp_1$$

In our schema, we would at any given iteration, select two cationic moieties from the top 20 pool of each property and interpolate 10 structures between them using SLERP coordinates. The model was allowed 100 RDKit-sanitization attempts before moving on past the current interpolation (i.e. in an iteration call 0-10 structures were returned). After procuring the interpolated structures, they were evaluated alongside a sampled anion (without anion pairing in Table 2) or alongside all experimental anions (with anion pairing in Table 2). In this way, total VAE function calls were minimized. If the candidate-anion pair was estimated to be within the property target bounds, it was selected as a solution to the search criteria.

Conclusions and Impact

Our work has oriented around materials for energy and medicine and harnessed the investigative power of statistical mechanics and methods. This work can be differentiated into *deep* and *wide* contributions—wide in that the methods can be applied across a spectrum of solvent types (ILs, deep eutectic solvents, mixtures, etc.) and deep in that more properties can be evaluated and the design tools refined. Both cases could lead to impactful discoveries for energy, specifically, and provide a platform for computer aided solvent design, generally. Consider the pioneering work at PNNL, where researchers discovered that adding a sulfate-chloride mixture to the all-vanadium electrolyte of an RFB led to a 70% increase in energy density.[23] This eventually led to the backbone technology of the battery spin-off UniEnergy Technologies, which is now manufacturing the largest compartmentalized RFB system in the world.[6] Computer aided design could further enable these chance discoveries by prescreening the incredible amount of conceivable structures. This is a scenario where the net is cast *wide*.

As for drilling *deep*, GAINS and Salty adopt the best practice for open-source development for this reason. New methods of calculating properties in MD or by other computational means are developed all the time. GAINS operates on the ubiquitous pdb molecular file type and SMILES alpha-numeric representation of molecules—which in turn can be translated into numerous fingerprint representations, descriptor sets, etc.—so that these new methods can plug into an existing discovery framework. Fragment counts within the structures can lend to group contribution (GC) based models of phase behavior, like the well-known UNIFAC (UNIversal quasi-chemical Functional-group Activity Coefficients) model.[1] Run-of-the-mill MD calculations of vaporization energies, melting temperature, etc. can be adapted into more refined models of non-ideal phase behaviors.[148–150] These investigations are exciting because the phase behavior (and how it typically leads to negligible vapor pressure) of ILs (along with some of the physiochemical properties mentioned earlier) is precisely what makes them so valuable as "green" solvents.[151] The most immediately exciting prospect, however, is that this framework could potentially discover very low viscosity ILs—an essential property of solvents in flow batteries.[19,152]

Appendix I

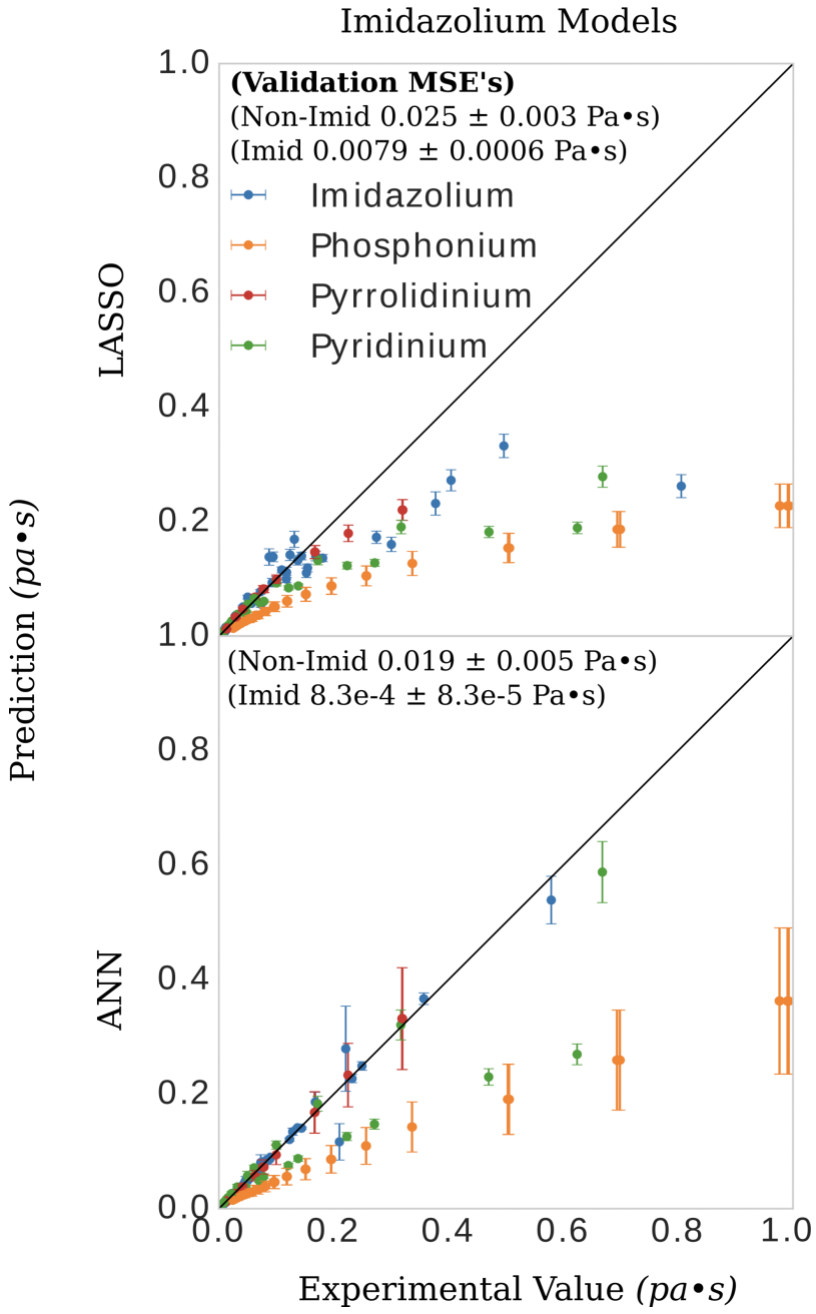


Figure A1-1. The predictions and standard deviations are obtained in the same way as discussed in Fig. 1-5 in Chapter One. Notably, the standard deviations for a given salt-type increase with increasing viscosity—an artifact of the absolute value of those viscosities i.e. the percent variance is about the same. We also see that the ANN model is fairly accurate for some salts not included in its training data: the pyrrolidinium salts and some of the pyridinium salts. Both models produce poor predictions for phosphonium salt-types, especially in the high viscosity regions where data is more scarce.

Table A1-1. Error and standard deviation of validation set predictions from bootstrap for the all-salts ANN model (from bottom right panel of Fig. 5 in part I)

RAAD	Temperature (K)	IL type
7.1 +/- 1.3	All	All
9.4 +/- 1.9	All	Imidazolium
1.8 +/- 0.1	All	Phosphonium
2.2 +/- 0.7	All	Pyrrolidinium
4.9 +/- 2.3	All	Pyridinium
14.9 +/- 4.8	273-298	All
5.8 +/- 1.8	299-323	All
3.7 +/- 0.7	324-348	All
4.6 +/- 2.2	349-373	All
21.5 +/- 7.6	273-298	Imidazolium
7.3 +/- 2.6	299-323	Imidazolium
4.5 +/- 1.0	324-348	Imidazolium
6.3 +/- 3.4	349-373	Imidazolium
6.3 +/- 3.4	273-298	Phosphonium
1.6 +/- 0.2	299-323	Phosphonium
2.2 +/- 0.1	324-348	Phosphonium
1.7 +/- 1.2	349-373	Phosphonium
2.5 +/- 0.8	273-298	Pyrrolidinium
1.8 +/- 0.7	299-323	Pyrrolidinium
N/A	324-348	Pyrrolidinium
N/A	349-373	Pyrrolidinium
11.9 +/- 6.9	273-298	Pyridinium
3.5 +/- 2.0	299-323	Pyridinium
3.6 +/- 3.4	324-348	Pyridinium
N/A	349-373	Pyridinium

Appendix II
QSPR/NN Development

Many duplicate measurements were available from the ILThermo database due to separate experimentalists performing analysis on the same systems at identical temperature and pressure. The database as a whole contained 1734 data points and 177 unique salt structures in the C_p dataset and 5631 data points and 461 unique salt structures in the ϱ dataset at 99-102 kPa and 297-316 K. Of these, 123 systems had three or more repeated measurements in the ϱ dataset and 427 in the C_p dataset. The mean of the relative standard deviations (RSD) of these measurements are presented in Table S1 (the relative form of the standard deviation is taken as a means to compare between C_p and ϱ, otherwise units would be in the form of the respective properties). The train/test set log mean squared errors (MSE), used in the loss calculation during backpropagation, of the QSPR/NN models following an 80/20 train/test protocol are also indicated in Table S1.

Table A2-1. QSPR/NN performance details (where N is train or test set size) and mean RSD of ILThermo data (where N indicates the IL systems with three or more data points from different experimentalists at the same T and P such that standard deviations could be computed).

	Property	
Method	Cp	ϱ
Experiment (mean RSD, %)	3.1 (N=123)	0.5 (N=427)
QSPR/NN Train (log MSE)	0.56 (N=1387)	0.01 (N=4504)
QSPR/NN Test (log MSE)	1.15 (N=347)	0.01 (N=1127)

The following formulae were used as metric calculations in Table 1 in the manuscript:

$$RAAD = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \quad (1)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \quad (2)$$

$$SS_{res} = \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \quad (3)$$

$$SS_{tot} = \sum_{i=1}^{N} (\bar{y}_i - y_i)^2 \quad (4)$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (5)$$

where $\hat{y}$ and $\bar{y}$ refer to predicted and mean values, respectively. As a companion to the RAAD calculation, the mean RSD was computed for experimental data for which there were more than three reported values for the same IL system at the same temperature and pressure:

$$mean\ RSD = \frac{100}{N} \sum_{i=1}^{N} \frac{\sigma_i}{\mu_i} \quad (6)$$

$$\sigma_i = \sqrt{\frac{1}{N-1} \sum_{j=1}^{N} \left(x_j - \mu_i\right)^2} \quad (7)$$

where σ_I and μ_i are the standard deviation and mean property values for a given IL system at identical temperature and pressure.

GA Framework

The GA had 39 common molecular fragments to add to a given molecular candidate, Fig. S1. In addition, the GA could add entire alkane chains up to length C4. Making use of chemical fragments was particularly important in this study since no form of temperature annealing/quenching was used by the GA; step-wise alterations to candidates had to always result in a fitness improvement if they were to be accepted.

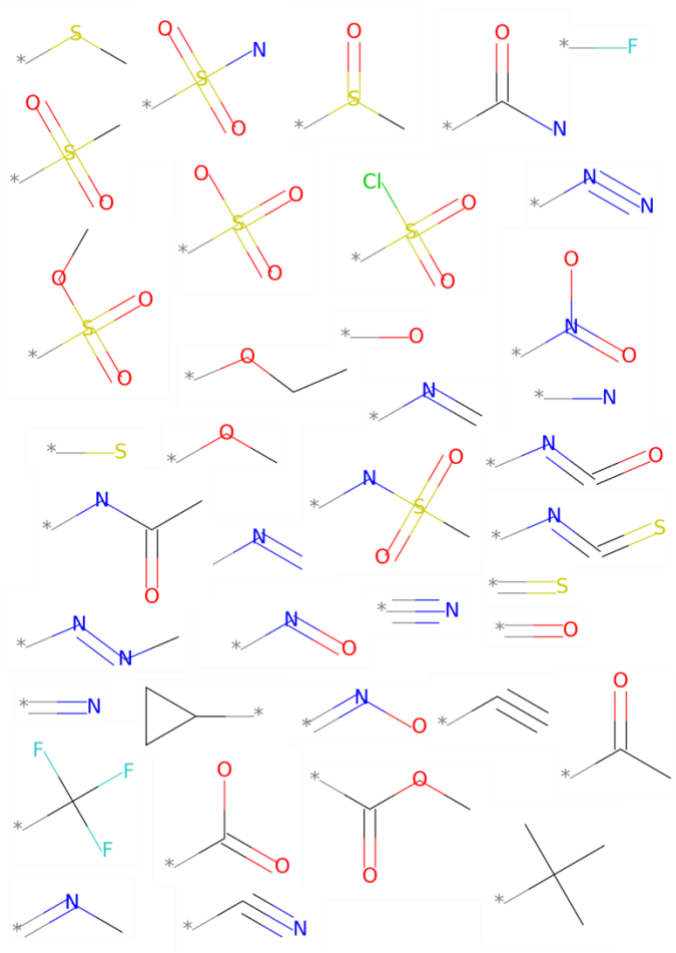


Figure A2-1. Molecular fragments available to the GA.

QM and MD Calculations

The results for verifying the QM/MD protocol using experimental data are shown in Fig. S2. The MD simulations tended to under predict the experimental densities. Several of the data, however, still fall within experimental error. The C_p calculations are distributed evenly about the parity line and fall quite well within experimental error. The MD error bars in these runs are produced from performing three separate MD trials and the experimental error bars are recreated directly from standard deviations reported in the original publications producing the data (this is in contrast to the methodology of producing the experimental error bars in Table A2-1).

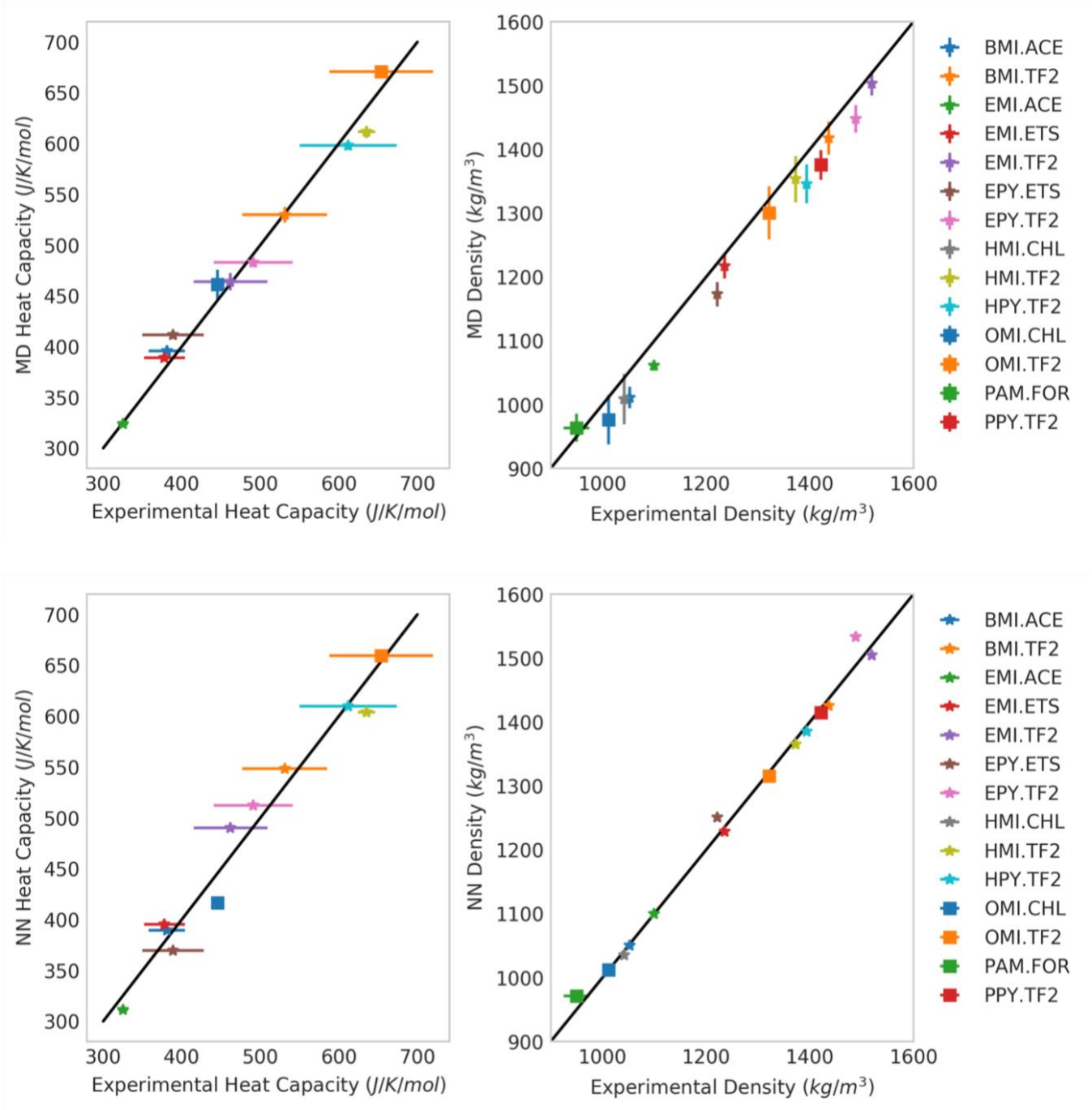


Figure A2-2. Calculated MD vs measured experimental properties. Left: C_p at constant pressure. Right: ϱ. P and C_p relative absolute average deviations (RAAD) are 2.85% and 2.49%, respectively.

AL&D

Fingerprinting

Most small molecule similarity mappings are permutations of the Jaccard (or Tanimoto) Index, defined as the inner join over the outer join; and molecular fingerprints, defined as bitwise (or count wise) representations of molecular structures. We applied three different fingerprint methods in this study, before proceeding with a common topological fingerprinting similar to the Daylight method. A Tanimoto similarity mapping and their similarity score using three different fingerprint types is presented in Fig. A2-3.

Briefly, the Daylight fingerprinting algorithm[97] operates by tabulating fragments within a molecule according to bond paths: First, only single atoms are tabulated; second, fragments separated by a single bond; third, fragments comprising two bonds, etc. until an upper limit of bond distance is achieved; seven, in the case of this work. Because the "allowable" observed bond patterns are not predefined (they are produced by the molecule itself), it is not possible to assign a bit to every observed pattern. Rather, the patterns are used as seeds in a pseudo-random number generator, i.e. they are hashed. The hashed patterns are bits of variable length that are added to a 2048-bit fingerprint with the logical OR. In this schematic, it is not 100% certain from the fingerprint whether a particular molecular fragment is present, but because it allows for many patterns to be considered it is a convenient tool for calculating molecular similarities. The similarity metric used in this work is the Tanimoto similarity,[97] defined as the inner over the outer joins of the fingerprint vectors for two molecules.

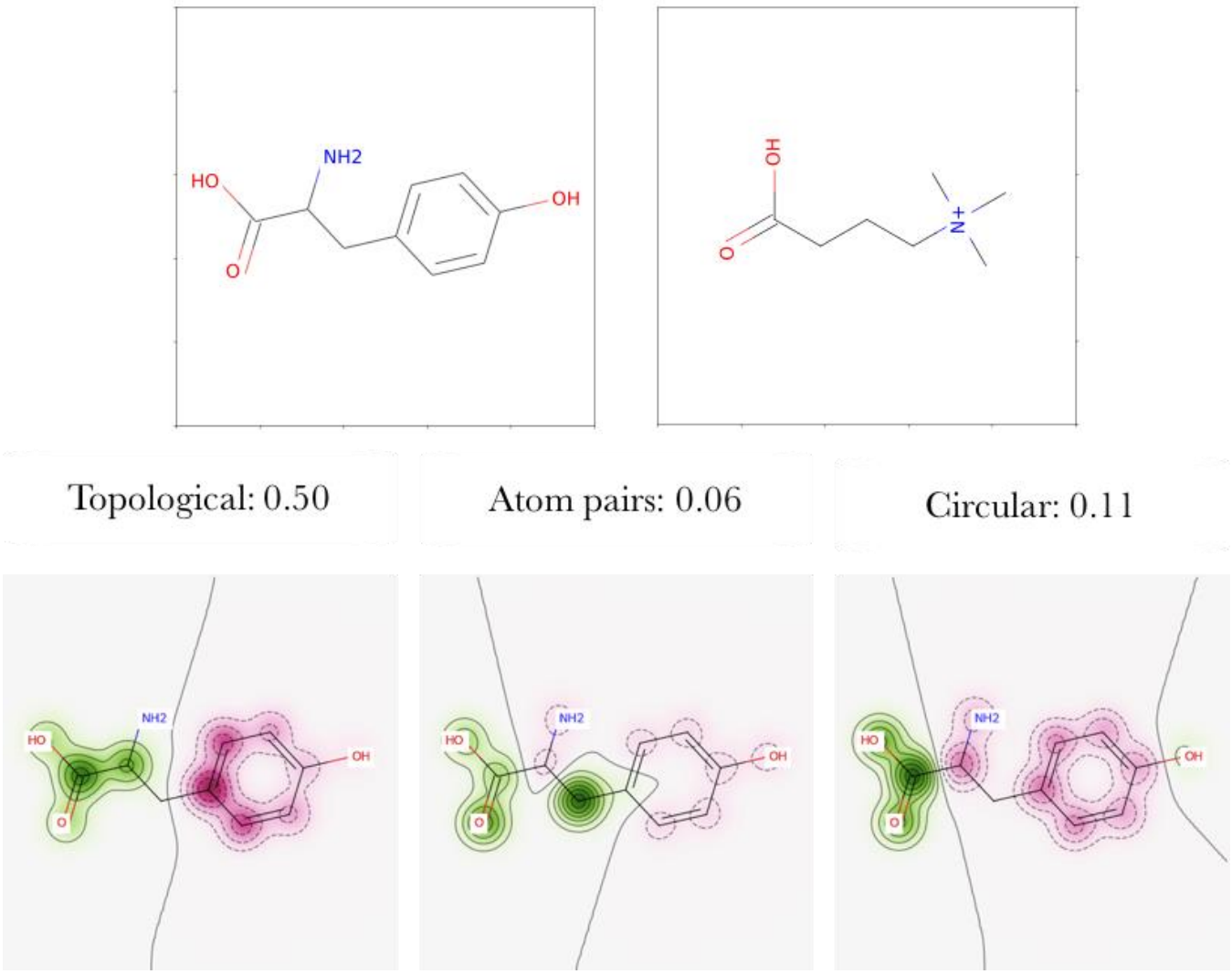


Figure A2-3. Example of Tanimoto similarity mapping using three different fingerprint methods: topological, atom pairs, and circular. Top two panels show the compared molecules.

In Fig. A2-3, green indicates areas of likeness between the two molecules and magenta indicates areas of dissimilarity. Each of the heat maps are independently scaled (this is why each image appears to average the same in the heat scales but procure scores ranging from 0.06 to 0.50). As evidenced by the figure, the similarity mapping between two molecules is heavily dependent on the type of fingerprint method. As explained previously, the topological mapping (Daylight) is performed by iteratively selecting an atom in the molecule and tabulating the molecular fragments that can be built from that atom up to a certain bond length, typically two to seven. Atom pair fingerprints are created by tabulating bonded atom pairs. Circular fingerprints are identical to topological fingerprints accept that instead of tabulating by bond distances, molecular fragments are tabulated by radius about the selected atom. In the above figure, a bond length of two was performed for the topological fingerprinting and a radius of two for the circular fingerprinting. Other types of fingerprint methods are based on functional groups (like the well-known MACCS keys) and hybrid methods. All of the

fingerprint methods result in vectors with varying degrees of sparsity and are subject to additional choices of: bits assigned per hashed feature and total bits per fingerprint.

Kernel Density Measurements

In addition to calculating the univariate KDEs, we calculated the multivariate KDEs using gaussian deposits with bandwidths of 0.4. It is interesting to note the similar distributions between the experimental data (left panel, Fig. A2-4.) and calculated MD properties for all returned GA ILs (right panel, Fig. A2-4).

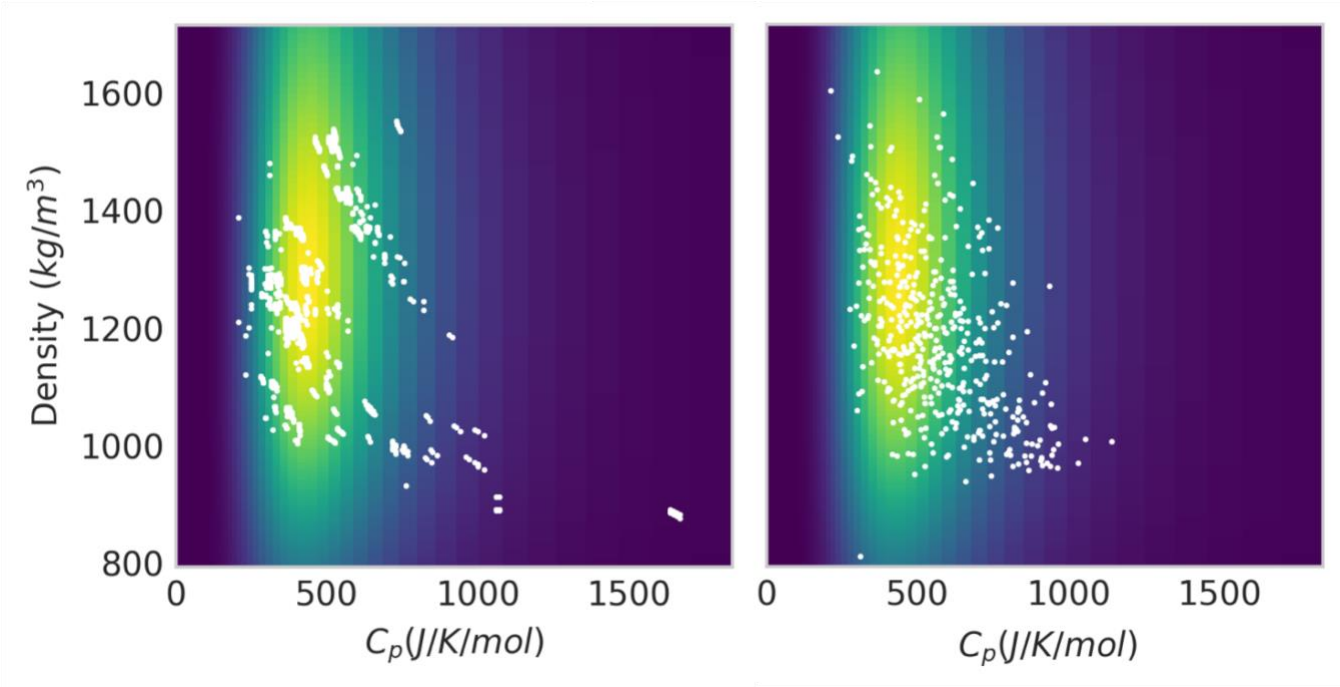


Figure A2-4. Multivariate KDEs for ϱ and C_p. Left: scatter overlay of inner join of experimental data for the two properties. Right: scatter overlay of MD calculations for all ILs generated by the GA.

Discussion and Analysis

Effect of AL&D on Model Prediction of Experimental Data

We considered how training the QSPR/NN models on MD data would affect their performance on evaluating the structure-property relationships of our known, experimental dataset. On the one hand, since chemical substructures lead to different macroscopic properties, we'd expect that adding data from multiple rounds of AL&D would increase performance on predicting values on an experimental validation set. On the other hand, allowing weight adjustments during backpropagation to improve error in one IL system may come at the expense of another IL system (unless they both benefit from learning some shared feature in the hidden layers). This may be especially true if certain substructures dominate in various property ranges—indeed this

is the basis of our error analysis via kernel density estimates (property ranges) and Tanimoto similarity (substructures). In the following we evaluate two extremes: include 100% of the experimental data + subsequent AL&D round data for training vs include 20% of the experimental data + subsequent AL&D round data. When we analyze performance on the experimental data in the first case, we observe that the performances remain relatively stagnant, Table S2. When we evaluate the second case, testing on the 80% of experimental data not used for training, we see that overall C_p predictions have suffered, but improved modestly with added AL&D round data.

Table A2-2. Summary of AL&D round data effect on experimental data predictions. Flush cases: 100% experimental data used in training in addition to indicated round data. RAAD is for experimental data portion of training data. Starved cases: 20% experimental data used in training in addition to indicated round data. RAAD is for 80% experimental test set data.

Round	RAAD, Flush Cases		RAAD, Starved Cases	
	ϱ ($N_{train/test}$ = 5631)	C_p ($N_{train/test}$ = 1734)	ϱ (N_{test} = 4504)	C_p (N_{test} = 1387)
1 (N_{train} = 47)	0.70%	4.93%	0.52%	13.95%
2 (N_{train} = 30)	0.68%	5.32%	1.69%	18.10%
3 (N_{train} = 34)	1.11%	4.31%	1.65%	8.49%
4 (N_{train} = 7)	0.84%	6.06%	1.44%	8.23%

Taking these two extremes together (flush with experimental training data vs starved of experimental training data) adding AL&D data has minimal impact on the experimental error rates. It's worth noting, however, that in the ϱ cases and the flush C_p case, these error rates are already within the realm of both reported experimental error and the mean RSD. These calculations are included on GitHub.

Breaking the Pareto Front

Including the results from the 1000 kg/m^3 – 1000 J/mol/K search, a total of 36 ILs were discovered beyond the PF. Their cations are shown in Fig. S5.

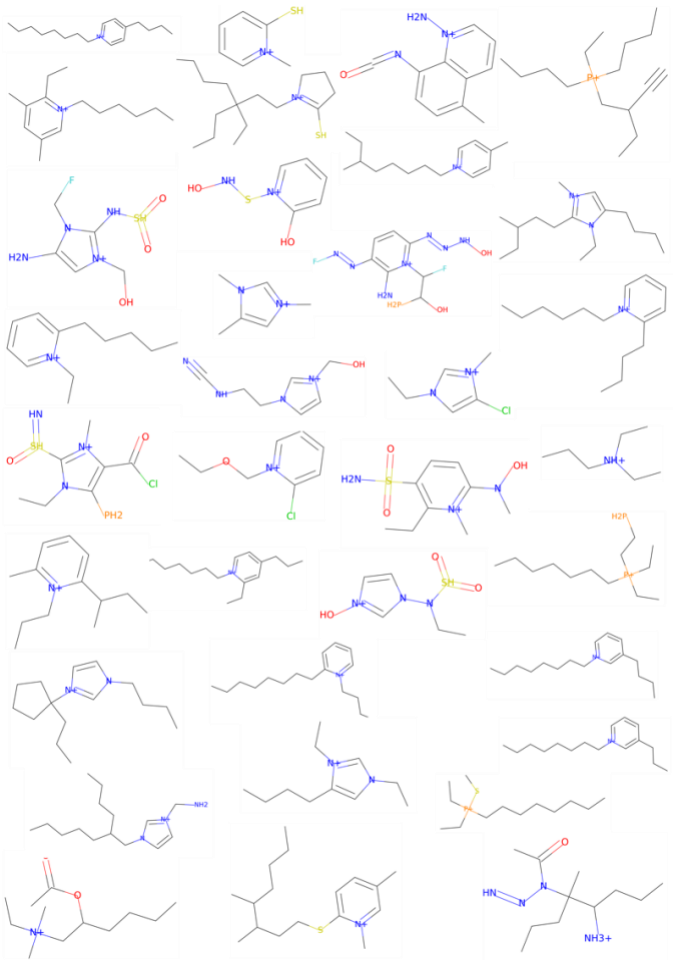


Figure A2-5. 36 cations found outside the PF by the GA.

For the ILs presented in Fig A2-5. in the main manuscript, an extensive RDF analysis based on steric and charge centers was performed and is discussed in greater detail in Fig. A2-6. Starting with the top half of Fig A2-6., these high C_p systems are compared to two ILs from the literature, tributylmethylammonium L-threoninate (TAM-THR) and 1-butyl-3-methylimidazolium octyl sulfate (BMI-OSF). The anion-anion charge center RDFs are in near perfect agreement with each other (top panels) TAM-THR has a modest first peak, same as its THR counterpart produced by the GA. The octyl sulfate anion-anion RDFs are, as well, in great agreement with each other with high initial peaks and tapering second and third peaks. The cation-cation steric centers (ring-bound nitrogens) are indicated in the middle panels. Here again, the experimental ILs mirror patterns in the GA produced ILs: the THR RDFs have a tight first peak and an early second peak compared to the OSF counterparts in both plots, as well as a much more modest dip before oscillating around bulk normalized value (g(r)=1). The cation-anion (+)/(-) charge center RDFs are shown in the bottom panels. These RDFs are distinguished by very high first peaks compared to the anion-anion and cation-cation RDFs, indicating tight coupling between opposite charge groups. All the high C_p systems are distinguished by cations with long alkyl tails and dispersed positive charge centers mostly around +0.13. The literature ILs TAM-THR and BMI-OSF had similar densities to the GA systems but much lower C_p.

The bottom half of Fig A2-6. repeats the same analysis for the high ϱ GA systems. In contrast to the high C_p systems, these GA produced cations had high magnitude charge centers, around +0.4 but up to +0.85 in two of five cases. We compared these GA systems to three systems from the literature: 1-butyl-3-methylimidazolium bromide (BMI-BRM), 1-butyl-3-methylimidazolium trifluoromethanesulfonate (BMI-TFS), and 1-ethyl-2-pentylpyridinium bis[(trifluoromethyl)sulfonyl]imide (PPY-TF2). These experimental systems also had high densities, though not as high as the GA produced counterparts. There are general agreements in the RDF plots. First and second peaks in the cation-cation and anion-anion plots share similar magnitudes. With exception of the highest density GA produced system, the cation-anion initial peaks are much more subdued compared to the high C_p systems.

Table A2-2 provides additional metadata of the top five C_p and ϱ systems. The Tanimoto similarity score in this table is of the procured cation and closest matching cation from all the available experimental data (in contrast to Figs. 5 & S6 which provide similarities of the GA cation and closest cation with matching anion). A dataframe of these and the entire 390 GA produced IL are available on github.

Table A2-3. Top C_p and ρ ILs found by the GA.

	Cation Smiles	Cation Atoms	Tanimoto Score (all cation comparison)	Molecular Relative	Anion	Calculated C_p	Calculated ρ
Top C_p	CCCCC(CC)(CCC)CC[N+]1=C(S)CCC1	18	0.51	N,N,N-triethyldodecan-1-aminium	L-threoninate	1143	1006
	PCCCCCCCCCCC[n+]1ccccc1	18	0.82	1-butylpyridinium	octyl sulfate	1054	1010
	CCCCCCCC[n+]1cccc(C)c1CCC	18	0.60	1,2-diethylpyridinium	octyl sulfate	1031	970
	Cc1ccc[n+](CCCCCCCCCCP)c1	19	0.96	N-octyl-3-methylpyridinium	octyl sulfate	966	1001
	CCCCCCCC[n+]1cccc(CCCC)c1	18	0.72	N-octyl-3-methylpyridinium	octyl sulfate	962	961
Top ρ	CCn1c(P)c(C(=O)Cl)[n+](C)c1[SH](=N)=O	15	0.52	1-butyl-3-methylimidazolium	bromide	366	1635
	C[N+]1=CC=CC=C1S	8	0.60	1-ethylpyridinium	bromide	212	1603

Nc1c(N=NF)ccc(N=NNO)[n+]1C(F)C(O)P	19	0.51	1-butyl-3-methylimidazolium	trifluoromethanesulfonate	505	1588
CCc1c(S(N)(=O)=O)ccc(N(C)O)[n+]1C	16	0.52	1-ethyl-2-methylpyridinium	bis[(trifluoromethyl)sulfonyl]imide	585	1563
N#CNCCn1cc[n+](CO)c1	12	0.62	1-butyl-3-methylimidazolium	bromide	342	1543

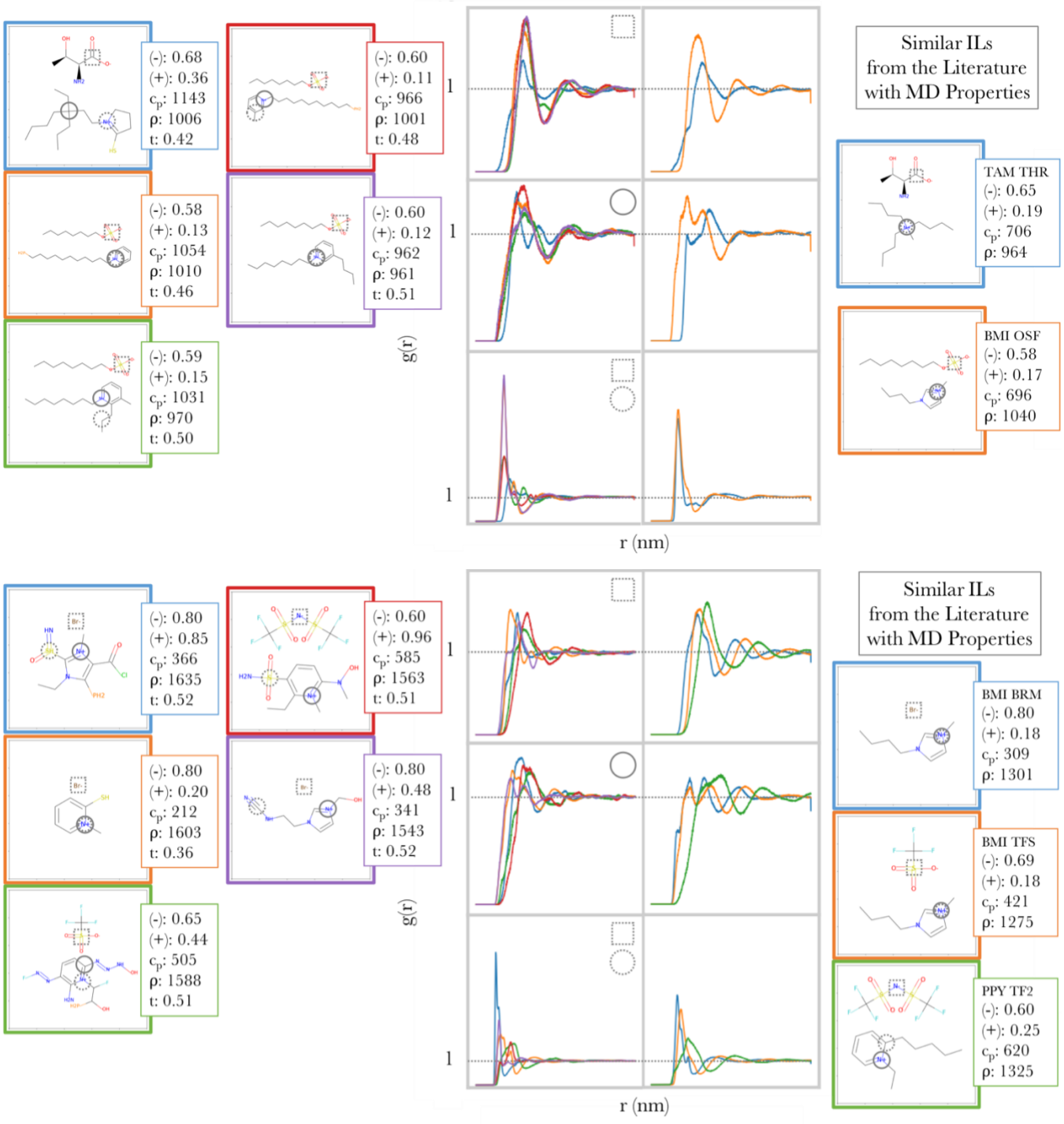


Figure A2-6. Re-portrayal of the information in Fig. 5 of the main manuscript. Charge and steric center based RDFs for five highest C_p (top) and five highest ϱ (bottom) systems designed in this study. Textbox insets indicate: (-), negative charge center electrostatic point charge indicated in the

primary structure diagrams as a dotted square; (+), positive charge center electrostatic point charge indicated in the primary structure diagrams as a dotted circle; C_p, MD calculated C_p at constant temperature; ϱ, MD calculated density; t, Tanimoto similarity score with the experimental cation in the right panel matched with the same anion. Top panel RDFs are of anion-anion negative charge centers (dotted squares). Middle panel RDFs are of cation-cation steric centers (N+ rings or solid circles). Bottom panel RDFs are of cation-anion charge centers (dotted circles and dotted squares). When steric centers (i.e. ring-bound nitrogens) were also the positive charge centers these are indicated by dotted circles circumferenced with solid circles in the primary structure diagrams. Three letter ion codes in the right textbox insets are as follows: TAM: tributylmethylammonium, THR: L-threoninate, BMI: 1-butyl-3-methylimidazolium, OSF: octyl sulfate, BRM: bromide, TFS: trifluoromethanesulfonate, PPY: 1-ethyl-2-pentylpyridinium, TF2: bis[(trifluoromethyl)sulfonyl]imide.

Center of Mass RDFs

We investigated the center-of-mass (COM) RDFs of every IL produced by the GA to determine if the procured solvent was in the liquid phase. Two ILs were rejected during the analysis, shown in Fig. S5. The IL in the left panel is a normal RDF of an IL in this study. In the right panel, the solvents' cations and anions separated from each other during the NPT equilibration in MD; they appear to be immiscible. The IL shown in the middle panel we believe was indicating solid-like behavior. The ϱ for this IL was very low, around 800 kg/m^3. A relational dataframe of all COM based RDFs is available on github along with an explanatory notebook written in python showing cases of strong cation-cation, cation-anion, and anion-anion interactions.

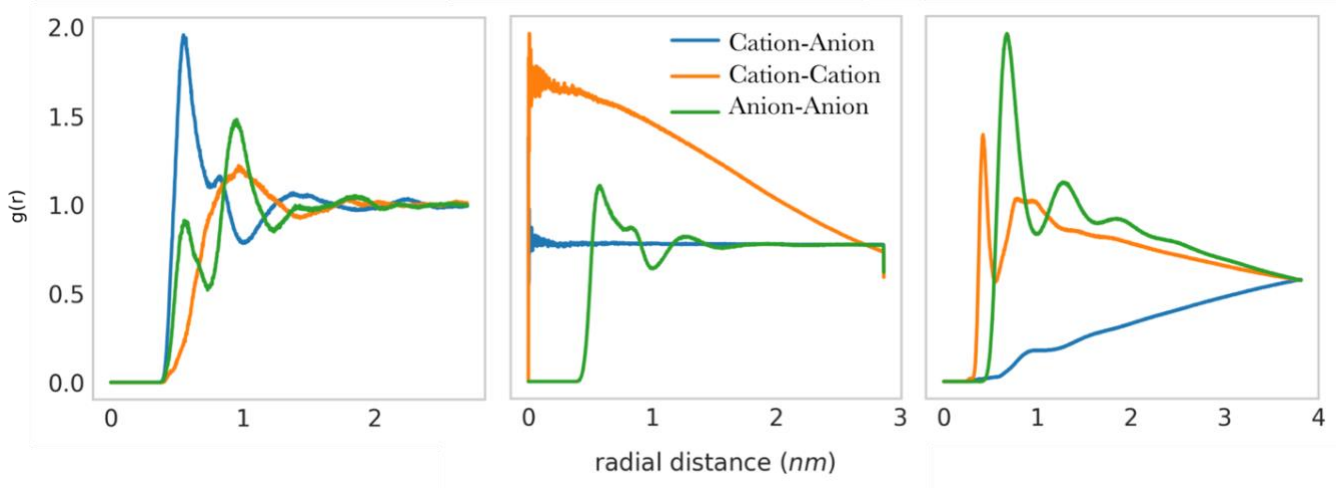


Figure A2-7. Left: Typical RDF of an IL in this work. Middle and right: two rejected solvents during the CH search.

The GA selected a total of 22 anions during its searches, Table 2. Each anion was used in a successful search anywhere from 10 (2-aminoethanesulfonate) to 38 times (L-lysinate). The number of unique

molecular relatives and the number of assigned categories gives an indication of the promiscuity of the anion in selecting a cationic moiety partner. The most promiscuous anion was trifluoroacetate with its cationic partners identifying with 14 closest relatives and spreading across five IL categories while the least promiscuous anion was trifluoromethanesulfonate with five closest relatives and three unique categories. The number of heavy atoms in, and the Tanimoto similarity score of, the cation partners head similar distributions for all of the anions. Although, octyl-sulfate did have the heaviest cationic partners at an average of 16.86 heavy atoms (incidentally, octyl-sulfate has the longest tail of all the anions and may explain its proclivity for heavy cationic partners).

Table A2-4. Selected anions by the GA.

	Anion	Molecular Relative	Category	Cation Heavy Atoms		Tanimoto Similarity Score	
Anion	count	unique	unique	mean	std	mean	std
2-aminoethanesulfonate	10	8	3	14.30	3.68	0.67	0.15
bromide	11	6	2	13.91	3.53	0.57	0.05
L-cysteinate	14	9	3	14.86	3.18	0.63	0.14
dimethylphosphate	14	7	3	14.29	3.60	0.66	0.14
thiocyanate	16	6	3	14.88	3.10	0.60	0.10
ethylsulfate	16	8	2	14.00	3.37	0.62	0.08
trifluoromethanesulfonate	17	5	3	14.12	3.72	0.63	0.13
acetate	18	10	4	15.89	2.97	0.61	0.14
chloride	20	9	2	13.65	2.54	0.64	0.12
methylsulfate	20	7	3	14.45	3.59	0.64	0.13
dicyanamide	20	7	2	14.45	3.15	0.59	0.11
dicyanoazanide	21	7	3	14.76	2.79	0.58	0.09
tricyanomethane	21	10	4	13.10	3.42	0.61	0.12
L-serinate	23	12	4	14.70	3.47	0.55	0.05
methanesulfonate	23	8	4	13.04	3.74	0.58	0.07
bis(trifluoromethylsulfonyl)imide	25	7	3	15.76	2.17	0.58	0.08
trifluoroacetate	25	14	5	13.16	3.69	0.65	0.13
nitrate	25	10	4	13.88	2.70	0.62	0.11
L-prolinate	26	7	3	15.04	3.50	0.60	0.12
L-threoninate	27	13	5	14.56	3.66	0.64	0.12
octyl sulfate	28	10	4	16.86	2.24	0.66	0.13
L-lysinate	38	12	6	15.29	3.44	0.61	0.13

Interestingly, when grouping the calculated properties by anion, there were some notable trends in both C_p and ϱ for a handful of the ions, Fig. S8. Octyl sulfate had a distinguishably higher C_p than the other ions, an interesting feature in that it had also been paired with the heaviest cationic moieties. Overall molecular weight would seem to be a contributor to high C_p. Observing the lower panel, bis(trifluoromethylsulfonyl)imide had a notably higher average ϱ, it's also the most highly branched anion from the selection, with six branching fluorenes and two carbonyl groups. Bromide, as well, had a higher average ϱ.

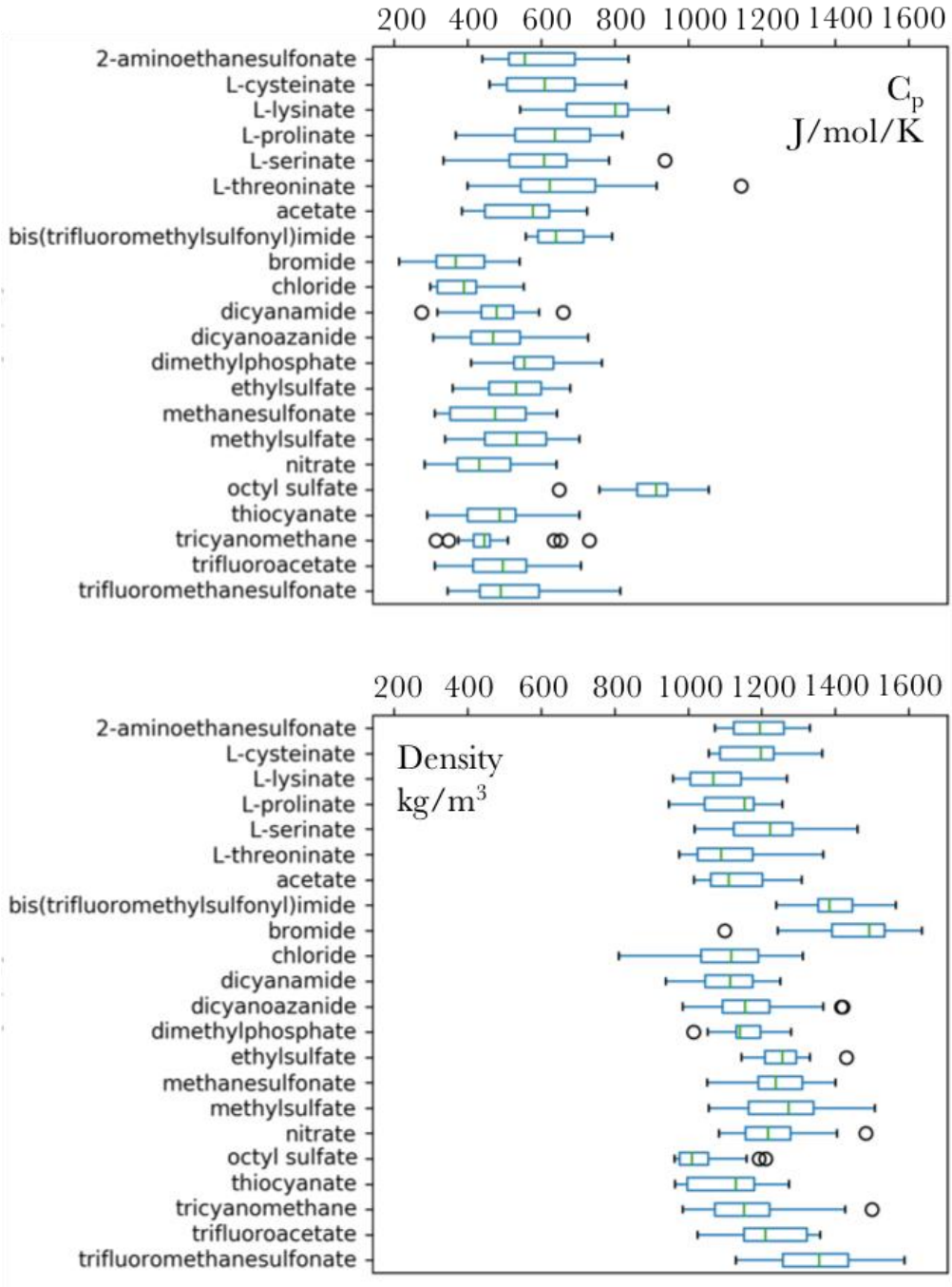


Figure A2-8. MD calculated properties grouped by anion. Outliers are indicated by circle ions, the inner quartile by the blue boxes, median by green lines, and upper and lower outer quartile limits by the blue bars.

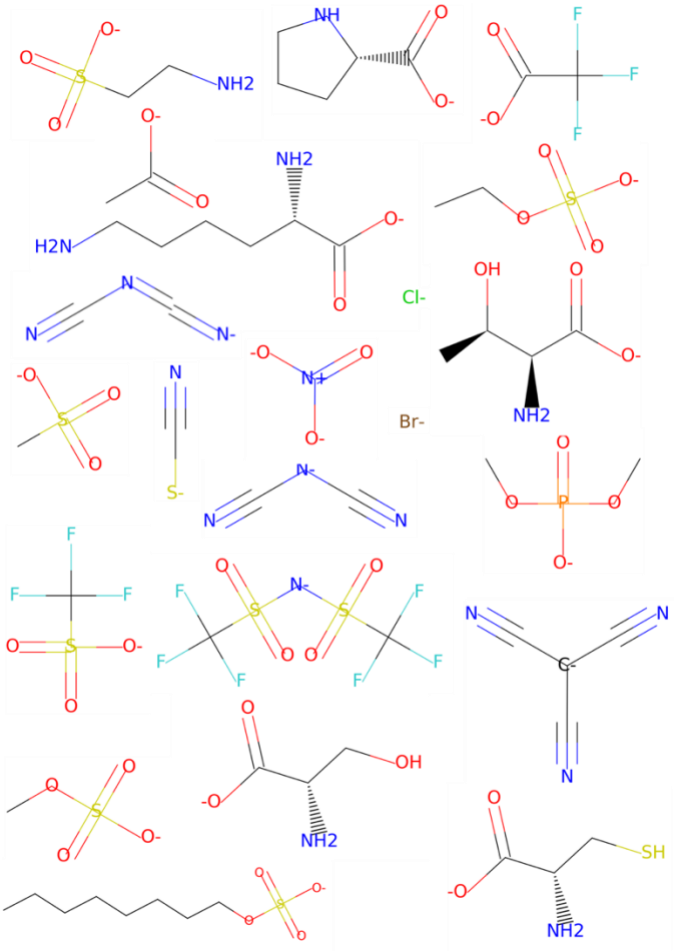


Figure A2-9. Selected anions by the GA.

Appendix III
Dual molecular output architectures

To create the Gen2 architecture, Gen1 is modified to be a furcated neural network.[73] A bifurcated network is utilized in this work, with sub-networks for the cation and anion in both the encoder and decoder. The motivation behind this architecture is that a chemical structure is hierarchical, beginning with the smallest unit – the atom. A collection of atoms forms either a cation or anion, which together form an ionic compound. We aim to learn representations of a specific level of the chemical hierarchy – here the cation and anion individually, by dedicating sub-networks to each. In this work, we do so in the encoder by passing the one-hot encoded cation and anion through three convolutional layers each prior to concatenation. In the decoder, the output of the third gated recurrent unit is used as input into two layers as in Gen1, but here these layers are passed through three more convolutional layers each before cation and anion are reconstructed as SMILES strings.

In Gen3, Gen2 is modified such that the cation and anion are never concatenated. Instead, we train separate latent spaces for each. The idea of bifurcation is extended here; since cation and anion are inherently different chemical units, a latent space for just the cation/anion may achieve better structure-property separation than a concatenated string might. By creating separate latent spaces, we extend the sub-networks for cation and anion such that they combine only when input to the QSPR predictor. In Gen2, the sub-networks combine in the concatenation layer.

Bibliography

1. Karunanithi AT, Mehrkesh A. Computer-aided design of tailor-made ionic liquids. AIChE J [Internet]. 2013 Dec;59(12):4627–40. Available from: http://doi.wiley.com/10.1002/aic.14228

2. Tian YH, Goff GS, Runde WH, Batista ER. Exploring electrochemical windows of room-temperature ionic liquids: A computational study. J Phys Chem B. 2012;116(39):11943–52.

3. Abo-Hamad A, Hayyan M, AlSaadi MA, Hashim MA. Potential applications of deep eutectic solvents in nanotechnology. Chem Eng J. 2015;273:551–67.

4. Sprenger KG, Plaks JG, Kaar JL, Pfaendtner J. Elucidating sequence and solvent specific design targets to protect and stabilize enzymes for biocatalysis in ionic liquids. Phys Chem Chem Phys [Internet]. 2017;19(26):17426–33. Available from: http://xlink.rsc.org/?DOI=C7CP03013D

5. He Z, Alexandridis P. Nanoparticles in ionic liquids: interactions and organization. Phys Chem Chem Phys [Internet]. 2015;17(28):18238–61. Available from: http://xlink.rsc.org/?DOI=C5CP01620G

6. uetechnologies.com [Internet]. Uni Energy Technology. 2017 [cited 2018 Mar 19]. Available from: uetechnologies.com

7. Hachmann J, Olivares-Amaya R, Atahan-Evrenk S, Amador-Bedolla C, Sánchez-Carrera RS, Gold-Parker A, et al. The Harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid. J Phys Chem Lett [Internet]. 2011 Sep 22;2(17):2241–51. Available from: http://pubs.acs.org/doi/10.1021/jz200866s

8. Jain A, Ong SP, Hautier G, Chen W, Richards WD, Dacek S, et al. Commentary: The Materials Project: a materials genome approach to accelerating materials innovation. APL Mater [Internet]. 2013 Jul;1(1):011002. Available from: http://aip.scitation.org/doi/10.1063/1.4812323

9. Saal JE, Kirklin S, Aykol M, Meredig B, Wolverton C. Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD). JOM [Internet]. 2013 Nov 28;65(11):1501–9. Available from: http://link.springer.com/10.1007/s11837-013-0755-4

10. Curtarolo S, Setyawan W, Wang S, Xue J, Yang K, Taylor RH, et al. AFLOWLIB.ORG: a distributed materials properties repository from high-throughput ab initio calculations. Comput Mater Sci [Internet]. 2012 Jun;58:227–35. Available from: http://dx.doi.org/10.1016/j.commatsci.2012.02.002

11. Abo-Hamad A, Hayyan M, AlSaadi MA, Hashim MA. Potential applications of deep eutectic solvents in nanotechnology. Chem Eng J [Internet]. 2015;273:551–67. Available from: http://dx.doi.org/10.1016/j.cej.2015.03.091

12. Nantasenamat C, Isarankura-Na-Ayudhya C, Naenna T, Prachayasittikul V. A practical

overview of quantitative structure-activity relationship. EXCLI J. 2009;8:74–88.

13. Leo A, Hansch C, Church C. Comparison of parameters currently used in the study of structure-activity relationships. J Med Chem. 1969;12(5):766–71.

14. Cherkasov A, Muratov EN, Fourches D, Varnek A, Baskin II, Cronin M, et al. QSAR modeling: where have you been? Where are you going to? J Med Chem [Internet]. 2014;57(12):4977–5010. Available from: http://pubs.acs.org/doi/abs/10.1021/jm4004285

15. Tropsha A. Best practices for QSAR model development, validation, and exploitation. Mol Inform [Internet]. 2010;29(6–7):476–88. Available from: http://doi.wiley.com/10.1002/minf.201000061

16. Dearden JC, Cronin MTD, Kaiser KLE. How not to develop a quantitative structure–activity or structure–property relationship (QSAR/QSPR). SAR QSAR Environ Res [Internet]. 2009;20(3–4):241–66. Available from: http://www.tandfonline.com/doi/abs/10.1080/10629360902949567

17. Labute P. A widely applicable set of descriptors. J Mol Graph Model [Internet]. 2000;18(4–5):464–77. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1093326300000681

18. Beck DAC, Carothers JM, Subramanian VR, Pfaendtner J. Data science: Accelerating innovation and discovery in chemical engineering. AIChE J [Internet]. 2016;62(5):1402–16. Available from: http://doi.wiley.com/10.1002/aic.15192

19. Miller MA, Wainright JS, Savinell RF. Communication—Iron ionic liquid electrolytes for redox flow battery applications. J Electrochem Soc [Internet]. 2016;163(3):A578–9. Available from: http://jes.ecsdl.org/lookup/doi/10.1149/2.0061605jes

20. Chakrabarti MH, Mjalli FS, AlNashef IM, Hashim MA, Hussain MA, Bahadori L, et al. Prospects of applying ionic liquids and deep eutectic solvents for renewable energy storage by means of redox flow batteries. Renew Sustain Energy Rev. 2014;30:254–70.

21. Lloyd D. Redox reactions in deep eutectic solvents: characterisation and application. School of Chemical Technology; 2013.

22. Xu Q, Zhao TS. Fundamental models for flow batteries. Prog Energy Combust Sci [Internet]. 2015;49:40–58. Available from: http://linkinghub.elsevier.com/retrieve/pii/S036012851500012X

23. Wang W, Luo Q, Li B, Wei X, Li L, Yang Z. Recent progress in redox flow battery research and development. Adv Funct Mater. 2013;23(8):970–86.

24. Weber AZ, Mench MM, Meyers JP, Ross PN, Gostick JT, Liu Q. Redox flow batteries: A review. J Appl Electrochem. 2011;41(10):1137–64.

25. Dong Q, Muzny CD, Kazakov A, Diky V, Magee JW, Widegren JA, et al. ILThermo: A free-

access web database for thermodynamic properties of ionic liquids. J Chem Eng Data [Internet]. 2007;52(4):1151–9. Available from: http://pubs.acs.org/doi/abs/10.1021/je700171f

26. Ghatee MH, Zare M, Zolghadr AR, Moosavi F. Temperature dependence of viscosity and relation with the surface tension of ionic liquids. Fluid Phase Equilib [Internet]. 2010;291(2):188–94. Available from: http://dx.doi.org/10.1016/j.fluid.2010.01.010

27. Gardas RL, Coutinho JAP. A group contribution method for viscosity estimation of ionic liquids. Fluid Phase Equilib. 2008;266(1–2):195–201.

28. Fernández A, García J, Torrecilla JS, Oliet M, Rodríguez F. Volumetric, transport and surface properties of [bmim][MeSO4] and [emim][EtSO4] ionic liquids as a function of temperature. J Chem Eng Data [Internet]. 2008;53(7):1518–22. Available from: http://pubs.acs.org/doi/abs/10.1021/je8000766

29. Zhao N, Oozeerally R, Degirmenci V, Wagner Z, Bendová M, Jacquemin J. New method based on the UNIFAC–VISCO model for the estimation of ionic liquids viscosity using the experimental data recommended by mathematical gnostics. J Chem Eng Data [Internet]. 2016;61(11):3908–21. Available from: http://pubs.acs.org/doi/10.1021/acs.jced.6b00689

30. Billard I, Marcou G, Ouadi A, Varnek A. In silico design of new ionic liquids based on quantitative structure-property relationship models of ionic liquid viscosity. J Phys Chem B. 2011;115(1):93–8.

31. Paduszyński K, Domańska U. Viscosity of ionic liquids: An extensive database and a new group contribution model based on a feed-forward artificial neural network. J Chem Inf Model. 2014;54(5):1311–24.

32. Yu G, Zhao D, Wen L, Yang S, Chen X. Viscosity of ionic liquids: Database, observation, and quantitative structure-property relationship analysis. AIChE J [Internet]. 2012;58(9):2885–99. Available from: http://doi.wiley.com/10.1002/aic.12786

33. Bandrés I, Alcalde R, Lafuente C, Atilhan M, Aparicio S. On the viscosity of pyridinium based ionic liquids: An experimental and computational study. J Phys Chem B [Internet]. 2011;115(43):12499–513. Available from: http://pubs.acs.org/doi/abs/10.1021/jp203433u

34. Gardas RL, Coutinho JAP. Group contribution methods for the prediction of thermophysical and transport properties of ionic liquids. AIChE J [Internet]. 2009;55(5):1274–90. Available from: http://doi.wiley.com/10.1002/aic.11737

35. Slattery JM, Daguenet C, Dyson PJ, Schubert TJS, Krossing I. How to predict the physical properties of ionic liquids: A volume-based approach. Angew Chemie Int Ed [Internet]. 2007;46(28):5384–8. Available from: http://doi.wiley.com/10.1002/anie.200700941

36. Matsuda H, Yamamoto H, Kurihara K, Tochigi K. Computer-aided reverse design for ionic liquids by QSPR using descriptors of group contribution type for ionic conductivities and

viscosities. Fluid Phase Equilib. 2007;261(1–2):434–43.

37. Zhao N, Jacquemin J. New method based on the UNIFAC-VISCO model for the estimation of dynamic viscosity of (ionic liquid + molecular solvent) binary mixtures. Fluid Phase Equilib [Internet]. 2017;449:41–51. Available from: http://dx.doi.org/10.1016/j.fluid.2017.06.006

38. Fatehi M-R, Raeissi S, Mowla D. Estimation of viscosities of pure ionic liquids using an artificial neural network based on only structural characteristics. J Mol Liq [Internet]. 2017;227:309–17. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0167732216326988

39. Crosthwaite JM, Muldoon MJ, Dixon JK, Anderson JL, Brennecke JF. Phase transition and decomposition temperatures, heat capacities and viscosities of pyridinium ionic liquids. J Chem Thermodyn. 2005;37(6):559–68.

40. Bini R, Malvaldi M, Pitner WR, Chiappe C. QSPR correlation for conductivities and viscosities of low-temperature melting ionic liquids. J Phys Org Chem. 2008;21(7–8):622–9.

41. Barycki M, Sosnowska A, Gajewicz A, Bobrowski M, Wileńska D, Skurski P, et al. Temperature-dependent structure-property modeling of viscosity for ionic liquids. Fluid Phase Equilib. 2016;427:9–17.

42. PyChem.

43. Cao DS, Xu QS, Hu QN, Liang YZ. ChemoPy: Freely available python package for computational biology and chemoinformatics. Bioinformatics. 2013;29(8):1092–4.

44. Landrum G. RDKit: Open-source cheminformatics [Internet]. [cited 2019 Apr 1]. Available from: http://www.rdkit.org

45. Tibshirani R. Regression shrinkage and selection via the LASSO. J R Stat Soc B. 1996;58(1):267–88.

46. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in python. J Mach Learn Res [Internet]. 2012;12:2825–30. Available from: http://dl.acm.org/citation.cfm?id=2078195%5Cnhttp://arxiv.org/abs/1201.0490

47. Veerasamy R, Rajak H, Jain A, Sivadasan S, Varghese CP, Agrawal RK. Validation of QSAR Models - Strategies and Importance. Int J Drug Des Disocovery. 2011;2(3):511–9.

48. Alexander DLJ, Tropsha A, Winkler DA. Beware of R^2: Simple, Unambiguous Assessment of the Prediction Accuracy of QSAR and QSPR Models. J Chem Inf Model. 2015;55(7):1316–22.

49. Legendre P. Spatial autocorrelation: Trouble or new paradigm? Ecology [Internet]. 1993;74(6):1659–73. Available from: http://doi.wiley.com/10.2307/1939924

50. Ord JK, Getis A. Local spatial autocorrelation statistics: Distributional issues and an application. Geogr Anal. 1995;27(4):286–306.

51. Todeschini R, Consonni V. Handbook of Molecular Descriptors [Internet]. Weinheim, Germany: Wiley-VCH Verlag GmbH; 2000. 3–527 p. (Methods and Principles in Medicinal Chemistry; vol. 4). Available from: http://doi.wiley.com/10.1002/9783527613106

52. Geary RC. The contiguity ratio and statistical mapping. Inc Stat [Internet]. 1954;5(3):115. Available from: http://www.jstor.org/stable/2986645?origin=crossref

53. Hall L, Kier L. The E-state as the basis for molecular structure space definition and structure similarity. J Chem Inf Comput Sci [Internet]. 2000;40(3):784–91. Available from: http://www.ncbi.nlm.nih.gov/pubmed/10850783

54. Butina D. Performance of Kier-Hall E-state descriptors in quantitative structure activity relationship (QSAR) studies of multifunctional molecules. Molecules. 2004;9(12):1004–9.

55. Hall LH, Kier LB. Electrotopological state indices for atom types: A novel combination of electronic, topological, and valence state information. J Chem Inf Model [Internet]. 1995;35(6):1039–45. Available from: http://pubs.acs.org/cgi-bin/doilookup/?10.1021/ci00028a014

56. Mohney B, Kier L, Hall LH. The electrotopological state: An atom index for QSAR. Quant Struct relationships. 1991;10:43–51.

57. Gasteiger J, Marsili M. Iterative partial equalization of orbital electronegativity—a rapid access to atomic charges. Tetrahedron [Internet]. 1980;36(22):3219–28. Available from: http://linkinghub.elsevier.com/retrieve/pii/0040402080801682

58. Basak SC, Gute BD, Grunwald GD. Use of Topostructural, topochemical, and geometric parameters in the prediction of vapor pressure: A hierarchical QSAR approach. 1997;2338(96):651–5.

59. Randic M, Basak S. Optimal molecular descriptors based on weighted path numbers. J Chem Inf Comput Sci [Internet]. 1999;39:261–6. Available from: http://pubs.acs.org/doi/abs/10.1021/ci9800763

60. Basak SC, Magnuson VR, Niemi GJ, Regal RR. Determining structural similarity of chemicals using graph-theoretic indices. Discret Appl Math. 1988;19(1–3):17–44.

61. Narumi H. New topological indices for finite and infinite systems. Commun Math Comput Chem. 1987;22:195–207.

62. Zhao N, Jacquemin J, Oozeerally R, Degirmenci V. New Method for the Estimation of Viscosity of Pure and Mixtures of Ionic Liquids Based on the UNIFAC–VISCO Model. J Chem Eng Data [Internet]. 2016;61(6):2160–9. Available from: http://pubs.acs.org/doi/10.1021/acs.jced.6b00161

63. Ramprasad R, Batra R, Pilania G, Mannodi-Kanakkithodi A, Kim C. Machine learning in materials informatics: recent applications and prospects. npj Comput Mater [Internet]. 2017

Dec 13;3(1):54. Available from: http://arxiv.org/abs/1707.07294

64. Lookman T, Balachandran P V., Xue D, Hogden J, Theiler J. Statistical inference and adaptive design for materials discovery. Curr Opin Solid State Mater Sci [Internet]. 2017;21(3):121–8. Available from: http://dx.doi.org/10.1016/j.cossms.2016.10.002

65. Ng LY, Chong FK, Chemmangattuvalappil NG. Challenges and opportunities in computer-aided molecular design. Comput Chem Eng [Internet]. 2015;81:115–29. Available from: http://dx.doi.org/10.1016/j.compchemeng.2015.03.009

66. Venkatasubramanian V, Chan K, Caruthers JM. Computer-aided molecular design using genetic algorithms. Comput Chem Eng. 1994;18(9):833–44.

67. Supady A, Blum V, Baldauf C. First-Principles Molecular Structure Search with a Genetic Algorithm. J Chem Inf Model. 2015;55(11):2338–48.

68. Venkatasubramaniam V, Patkar PR. Genetic Algorithms Based CAMD. In: Achenie LEK, Ganie R, Venkatasubramanian V, editors. Computer Aided Molecular Design: Theory and Practice. Austin, TX; 2003. p. 95.

69. Sheppard C. Genetic Algorithms with Python. Austin, TC; 2016.

70. Maia FM, Tsivintzelis I, Rodriguez O, Macedo EA, Kontogeorgis GM. Equation of state modelling of systems with ionic liquids: Literature review and application with the Cubic Plus Association (CPA) model. Fluid Phase Equilib [Internet]. 2012;332:128–43. Available from: http://dx.doi.org/10.1016/j.fluid.2012.06.026

71. Sprenger KG, Jaeger VW, Pfaendtner J. The general AMBER force field (GAFF) can accurately predict thermodynamic and transport properties of many ionic liquids. J Phys Chem B. 2015;119(18):5882–95.

72. Beckner W. Salty.

73. Sakloth K, Beckner W, Pfaendtner J, Goh GB. IL-Net: Using Expert Knowledge to Guide the Design of Furcated Neural Networks. In: 2018 IEEE International Conference on Big Data [Internet]. IEEE; 2018. p. 1465–73. Available from: http://arxiv.org/abs/1809.05127

74. Cereto-Massagué A, Ojeda MJ, Valls C, Mulero M, Garcia-Vallvé S, Pujadas G. Molecular fingerprint similarity search in virtual screening. Methods. 2015;71(C):58–63.

75. Rezende DJ, Mohamed S, Wierstra D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. arXiv Prepr arXiv14014082 [Internet]. 2014 Jan 16; Available from: https://arxiv.org/abs/1401.4082

76. Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol Comput [Internet]. 2003 Mar;11(1):1–18. Available from:

http://www.mitpressjournals.org/doi/10.1162/106365603321828970

77. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. NIPS; 2012. p. 1097–105.

78. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Advances in neural information processing systems. NIPS; 2014. p. 2672–80.

79. Higgins I, Matthey L, Glorot X, Pal A, Uria B, Blundell C, et al. Early Visual Concept Learning with Unsupervised Deep Learning. arXiv Prepr arXiv160605579 [Internet]. 2016 Jun 17; Available from: http://arxiv.org/abs/1606.05579

80. Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, et al. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. ACS Cent Sci [Internet]. 2018 Feb 28;4(2):268–76. Available from: http://arxiv.org/abs/1610.02415

81. Kuzminykh D, Polykovskiy D, Kadurin A, Zhebrak A, Baskov I, Nikolenko S, et al. 3D molecular representations based on the wave transform for convolutional neural networks. Mol Pharm [Internet]. 2018 Oct 23;15(10):4378–85. Available from: http://pubs.acs.org/doi/10.1021/acs.molpharmaceut.7b01134

82. Blum LC, Reymond J. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. J Am Chem Soc [Internet]. 2009 Jul;131(25):8732–3. Available from: http://pubs.acs.org/doi/abs/10.1021/ja902302h

83. Sterling T, Irwin JJ. ZINC 15 – Ligand Discovery for Everyone. J Chem Inf Model [Internet]. 2015 Nov 23;55(11):2324–37. Available from: http://pubs.acs.org/doi/10.1021/acs.jcim.5b00559

84. Caleman C, van Maaren PJ, Hong M, Hub JS, Costa LT, van der Spoel D. Force Field Benchmark of Organic Liquids: Density, Enthalpy of Vaporization, Heat Capacities, Surface Tension, Isothermal Compressibility, Volumetric Expansion Coefficient, and Dielectric Constant. J Chem Theory Comput [Internet]. 2012 Jan 10;8(1):61–74. Available from: http://pubs.acs.org/doi/10.1021/ct200731v

85. Tusar T, Filipic B. Visualization of Pareto front approximations in evolutionary multiobjective optimization: a critical review and the prosection method. IEEE Trans Evol Comput [Internet]. 2015 Apr;19(2):225–45. Available from: http://ieeexplore.ieee.org/document/6777535/

86. Agrawal G, Bloebaum C, Lewis K, Chugh K, Huang C-H, Parashar S. Intuitive visualization of Pareto frontier for multiobjective optimization in n-dimensional performance space. In: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference [Internet]. AIAA; 2004. p. 4434. Available from: http://arc.aiaa.org/doi/10.2514/6.2004-4434

87. Pelay U, Luo L, Fan Y, Stitou D, Rood M. Thermal energy storage systems for concentrated

solar power plants. Renew Sustain Energy Rev [Internet]. 2017;79(May):82–100. Available from: http://dx.doi.org/10.1016/j.rser.2017.03.139

88. Paul TC, Morshed AKMM, Fox EB, Khan JA. Enhanced thermophysical properties of NEILs as heat transfer fluids for solar thermal applications. Appl Therm Eng [Internet]. 2017;110:1–9. Available from: http://dx.doi.org/10.1016/j.applthermaleng.2016.08.004

89. Barycki M, Sosnowska A, Gajewicz A, Bobrowski M, Wileńska D, Skurski P, et al. Temperature-dependent structure-property modeling of viscosity for ionic liquids. Fluid Phase Equilib [Internet]. 2016 Nov;427:9–17. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0378381216303193

90. Rybinska A, Sosnowska A, Barycki M, Puzyn T. Geometry optimization method versus predictive ability in QSPR modeling for ionic liquids. J Comput Aided Mol Des. 2016;30(2):165–76.

91. Keshavarz MH, Pouretedal HR, Saberi E. A simple method for prediction of density of ionic liquids through their molecular structure. J Mol Liq. 2016;216:732–7.

92. Paduszyński K. Extensive Databases and Group Contribution QSPRs of Ionic Liquids Properties. 1. Density. Ind Eng Chem Res [Internet]. 2019;acs.iecr.9b00130. Available from: http://pubs.acs.org/doi/10.1021/acs.iecr.9b00130

93. Paternò A, Fiorenza R, Marullo S, Musumarra G, Scirè S. Prediction of ionic liquid's heat capacity by means of their in silico principal properties. RSC Adv [Internet]. 2016;6(42):36085–9. Available from: http://xlink.rsc.org/?DOI=C6RA05106E

94. Zhao Y, Zeng S, Huang Y, Afzal RM, Zhang X. Estimation of heat capacity of ionic liquids using S σ-profile molecular descriptors. Ind Eng Chem Res. 2015;54(51):12987–92.

95. Khosravi A, Nahavandi S, Creighton D, Atiya AF. Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances. IEEE Trans Neural Networks [Internet]. 2011 Sep;22(9):1341–56. Available from: http://ieeexplore.ieee.org/document/5966350/

96. Beckner W, Mao CM, Pfaendtner J. Statistical models are able to predict ionic liquid viscosity across a wide range of chemical functionalities and experimental conditions. Mol Syst Des Eng [Internet]. 2018;3(1):253–63. Available from: http://pubs.rsc.org/en/Content/ArticleLanding/2018/ME/C7ME00094D

97. Butina D. Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets. J Chem Inf Comput Sci [Internet]. 1999 Jul;39(4):747–50. Available from: http://pubs.acs.org/doi/abs/10.1021/ci9803381

98. Olivecrona M, Blaschke T, Engkvist O, Chen H. Molecular de-novo design through deep reinforcement learning. J Cheminform. 2017;9(1):48–62.

99. Segler MHS, Kogej T, Tyrchan C, Waller MP. Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS Cent Sci [Internet]. 2018 Jan 24;4(1):120–31. Available from: http://pubs.acs.org/doi/10.1021/acscentsci.7b00512

100. Bjerrum EJ. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. arXiv Prepr arXiv170307076 [Internet]. 2017 Mar; Available from: http://arxiv.org/abs/1703.07076

101. Goh GB, Siegel C, Vishnu A, Hodas NO. Using rule-based labels for weak supervised learning: a ChemNet for transferable chemical property prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining [Internet]. ACM; 2018. Available from: http://arxiv.org/abs/1712.02734

102. Chollet F. Keras [Internet]. 2015 [cited 2019 Apr 1]. Available from: https://keras.io

103. Cornell WD, Cieplak P, Bayly CI, Kollman P a., Kollmann PA. Application of RESP charges to calculate conformational energies, hydrogen bond energies, and free energies of solvation. J Am Chem Soc. 1993;115(7):9620–31.

104. Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, et al. Gaussian 09, Revision A.02. Wallingford CT: Gaussian, Inc.; 2009.

105. Wang JM, Wolf RM, Caldwell JW, Kollman PA, Case DA. Development and testing of a general amber force field. J Comput Chem. 2004 Jul 15;25(9):1157–74.

106. Liu H, Maginn E. A molecular dynamics investigation of the structural and dynamic properties of the ionic liquid 1-n-butyl-3-methylimidazolium bis(trifluoromethanesulfonyl)imide. J Chem Phys [Internet]. 2011 Sep 28;135(12):124507. Available from: http://www.ncbi.nlm.nih.gov/pubmed/21974535

107. Zhang Y, Otani A, Maginn EJ. Reliable viscosity calculation from equilibrium molecular dynamics simulations: a time decomposition method. J Chem Theory Comput [Internet]. 2015;11:3537–46. Available from: http://dx.doi.org/10.1021/acs.jctc.5b00351

108. Zhang Y, Maginn EJ. A simple AIMD approach to derive atomic charges for condensed phase simulation of ionic liquids. J Phys Chem B. 2012;116(33):10036–48.

109. Case DA, Cheatham TE, Darden T, Gohlke H, Luo R, Merz KM, et al. The Amber biomolecular simulation programs. J Comput Chem [Internet]. 2005 Dec;26(16):1668–88. Available from: http://www.ncbi.nlm.nih.gov/pubmed/16200636%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1989667

110. Sousa da Silva AW, Vranken WF. ACPYPE - AnteChamber PYthon Parser interfacE. BMC Res Notes [Internet]. 2012;5(1):367. Available from: http://bmcresnotes.biomedcentral.com/articles/10.1186/1756-0500-5-367

111. Martínez L, Andrade R, Birgin EG, Martínez JM. PACKMOL: A package for building initial configurations for molecular dynamics simulations. J Comput Chem [Internet]. 2009 Oct;30(13):2157–64. Available from: http://doi.wiley.com/10.1002/jcc.21224

112. Pronk S, Páll S, Schulz R, Larsson P, Bjelkmar P, Apostolov R, et al. GROMACS 4.5: A high-throughput and highly parallel open source molecular simulation toolkit. Bioinformatics. 2013;29(7):845–54.

113. Bussi G, Donadio D, Parrinello M. Canonical sampling through velocity rescaling. J Chem Phys [Internet]. 2007;126(1):014101. Available from: http://scitation.aip.org/content/aip/journal/jcp/126/1/10.1063/1.2408420

114. Berendsen HJC, Postma JPM, van Gunsteren WF, DiNola A, Haak JR. Molecular dynamics with coupling to an external bath. J Chem Phys. 1984;81(8):3684–90.

115. Parrinello M, Rahman A. Polymorphic transitions in single crystals: A new molecular dynamics method. J Appl Phys [Internet]. 1981 Dec;52(12):7182–90. Available from: http://aip.scitation.org/doi/10.1063/1.328693

116. Rappé AK, Casewit CJ, Colwell KS, Goddard WA, Skiff WM. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. J Am Chem Soc. 1992;114(25):10024–35.

117. Kruglov I, Sergeev O, Yanilkin A, Oganov AR. Energy-free machine learning force field for aluminum. Sci Rep [Internet]. 2017;7(1):1–7. Available from: http://dx.doi.org/10.1038/s41598-017-08455-3

118. Botu V, Batra R, Chapman J, Ramprasad R. Machine learning force fields: Construction, validation, and outlook. J Phys Chem C. 2017;121(1):511–22.

119. Huan TD, Batra R, Chapman J, Krishnan S, Chen L, Ramprasad R. A universal strategy for the creation of machine learning-based atomistic force fields. npj Comput Mater [Internet]. 2017;3(1):37. Available from: http://www.nature.com/articles/s41524-017-0042-y

120. Elton DC, Boukouvalas Z, Fuge D, Chung PW. Deep learning for molecular design—a review of the state of the art. 2019;

121. Dahl GE, Jaitly N, Salakhutdinov R. Multi-task Neural Networks for QSAR Predictions. 2014;1–21. Available from: http://arxiv.org/abs/1406.1231

122. NVIDIA DGX-1 With Tesla V100 System Architecture [Internet]. Available from: https://www.nvidia.com/en-us/data-center/resources/dgx-1-system-architecture-whitepaper/

123. Lake B, Salakhutdinov R, Tenebaum J. Human-level concept learning through probabilistic program induction. 2015;350(6266):1332–88.

124. Belhaj M, Protopapas P, Pan W. Deep Variational Transfer: Transfer Learning through Semi-supervised Deep Generative Models. arXiv Prepr arXiv181203123 [Internet]. 2018; Available from: http://arxiv.org/abs/1812.03123

125. Zhang R, Isola P, Efros AA. Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition [Internet]. 2016. p. 1058–67. Available from: http://arxiv.org/abs/1611.09842

126. Goh GB, Hodas NO, Siegel C, Vishnu A. SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties. arXiv Prepr arXiv171202034 [Internet]. 2017; Available from: http://arxiv.org/abs/1712.02034

127. Fare C, Turcani L, Pyzer-Knapp EO. Powerful, transferable representations for molecules through intelligent task selection in deep multitask networks. arXiv Prepr arXiv180906334 [Internet]. 2018; Available from: http://arxiv.org/abs/1809.06334

128. Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, et al. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. IEEE Trans Pattern Anal Mach Intell. 2017;39(4):677–91.

129. Zhao S, Song J, Ermon S. Towards Deeper Understanding of Variational Autoencoding Models. arXiv Prepr arXiv170208658 [Internet]. 2017; Available from: http://arxiv.org/abs/1702.08658

130. Larsen ABL, Sønderby SK, Larochelle H, Winther O. Autoencoding beyond pixels using a learned similarity metric. arXiv Prepr arXiv151209300 [Internet]. 2015; Available from: http://arxiv.org/abs/1512.09300

131. Kingma DP, Welling M. Auto-Encoding Variational Bayes. arXiv Prepr arXiv13126114. 2014;

132. Beckner W, Pfaendtner J. Fantastic Liquids and Where To Find Them: Optimizations of Discrete Chemical Space. J Chem Inf Model. 2019;59:2617–25.

133. Popova M, Shvets M, Oliva J, Isayev O. MolecularRNN : Generating realistic molecular graphs with optimized properties. arXiv Prepr arXiv190513372. 2019;

134. Jin W, Barzilay R, Jaakkola T. Multi-resolution Autoregressive Graph-to-Graph Translation for Molecules. arXiv Prepr arXiv190711223. 2019;

135. Winter R, Montanari F, Noé F, Clevert D-A. Chemical Science Learning continuous and data-driven molecular representations. Chem Sci. 2019;10(6):1692–701.

136. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Model [Internet]. 1988 Feb 1;28(1):31–6. Available from: http://pubs.acs.org/cgi-bin/doilookup/?10.1021/ci00057a005

137. O'Boyle NM. Towards a Universal SMILES representation - A standard method to generate

canonical SMILES based on the InChI. J Cheminform [Internet]. 2012;4(1):22. Available from: Journal of Cheminformatics

138. Koichi S, Iwata S, Uno T, Koshino H, Satoh H. Algorithm for Advanced Canonical Coding of Planar Chemical Structures That Considers Stereochemical and Symmetric Information. J Chem Inf Model. 2007;47(5):1734–46.

139. Virshup AM, Wipf P, Yang W, Beratan DN. Stochastic Voyages into Uncharted Chemical Space Produce a Representative Library of All Possible Drug-Like Compounds. J Am Chem Soc. 2013;135(19):7296–303.

140. Gaulton A, Hersey A, Patr A, Chambers J, Mendez D, Mutowo P, et al. The ChEMBL database in 2017. Nucleic Acids Res. 2016;45(D1):945–54.

141. Ruddigkeit L, Deursen R Van, Blum LC, Reymond J. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. J Chem Inf Model. 2012;52(11):2864–75.

142. Agrawal G, Bloebaum C, Lewis K, Chugh K, Huang C-H, Parashar S. Intuitive Visualization of Pareto Frontier for Multiobjective Optimization in n-Dimensional Performance Space. In: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference [Internet]. American Institute of Aeronautics and Astronautics; 2004. (Multidisciplinary Analysis Optimization Conferences). Available from: https://doi.org/10.2514/6.2004-4434

143. Gregor K, Graves A, Com WG. DRAW : A Recurrent Neural Network For Image Generation. arXiv Prepr arXiv150204623. 2015;

144. Doersch C. Tutorial on Variational Autoencoders. arXiv Prepr arXiv160605908 [Internet]. 2016 Jun 19; Available from: http://arxiv.org/abs/1606.05908

145. Shoemaker K. Animating Rotation with Quaternion Curves. In: ACM SIGGRAPH computer graphics. ACM; 1985. p. 245–54.

146. Kazakov A, Magee JW, Chirico RD, Paulechka E, Diky V, Muzny CD, et al. NIST Standard Reference Database 147: NIST Ionic Liquids Database - (ILThermo), Version 2.0 [Internet]. [cited 2017 Nov 9]. Available from: http://ilthermo.boulder.nist.gov

147. White T. Sampling Generative Networks. arXiv Prepr arXiv160904468. 2016;

148. Sankaranarayanan SKRS, Bhethanabotla VR, Joseph B. Molecular dynamics simulation study of the melting of Pd-Pt nanoclusters. Phys Rev B [Internet]. 2005 May 24;71(19):195415. Available from: https://link.aps.org/doi/10.1103/PhysRevB.71.195415

149. Berger O, Edholm O, Jähnig F. Molecular dynamics simulations of a fluid bilayer of dipalmitoylphosphatidylcholine at full hydration, constant pressure, and constant temperature. Biophys J [Internet]. 1997 May;72(5):2002–13. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0006349597788453

150. Caldwell JW, Kollman PA. Structure and Properties of Neat Liquids Using Nonadditive Molecular Dynamics: Water, Methanol, and N-Methylacetamide. J Phys Chem [Internet]. 1995 Apr;99(16):6208–19. Available from: http://pubs.acs.org/doi/abs/10.1021/j100016a067

151. Greaves TL, Weerawardena A, Fong C, Krodkiewska I, Drummond CJ. Protic Ionic Liquids: Solvents with Tunable Phase Behavior and Physicochemical Properties. J Phys Chem B [Internet]. 2006 Nov;110(45):22479–87. Available from: http://pubs.acs.org/doi/abs/10.1021/jp0634048

152. Pratt HD, Leonard JC, Steele LAM, Staiger CL, Anderson TM. Copper ionic liquids: Examining the role of the anion in determining physical and electrochemical properties. Inorganica Chim Acta [Internet]. 2013;396:78–83. Available from: http://dx.doi.org/10.1016/j.ica.2012.10.005