Supplementary Information for Exploring the deviation from Nernst-Einstein conductivity in ionic liquids using machine learning

Aditi Seshadri, Lyndon T. M. Hess, Shuwen Yue

Contents

S1 Nernst-Einstein Equation, Ionicity, Ion Dissociation, and Charge Transfer

The molar conductivity of ILs often deviates from the Nernst-Einstein conductivity (Eq. 1).[1,2] This deviation can be expressed as the ratio of the measured molar ionic conductivity (σ) and the Nernst-Einstein conductivity (σ_{NE}), and has been referred to in the literature as ionicity (I), the inverse Haven Ratio (H_R^{-1}), and the Nernst-Einstein ratio (Eq. 2).[1–4]

$$\sigma_{NE} = \frac{(N_A e^2)}{(k_B T)}(v_+ z_+^2 D_+ + v_- z_-^2 D_-) \tag{1}$$

Where N_A is the Avogadro number, e is the elementary charge, v_i is the stoichiometric coefficient for the cation (v_+) or anion (v_-), z_i is the charge on the cation (z_+) or anion (z_-), and D_i is the self-diffusivity of the cation (D_+) or anion (D_-).

$$I = H_R^{-1} = \frac{\sigma}{\sigma_{NE}} \tag{2}$$

With regards to ion dissociation, ionicity is often interpreted as the ratio of "free" or dissociated ions, to the total number of ions in the system.[3,4] However, the definition of an ion pair in a pure IL system and the interpretation of ionicity for ILs with ionicity values above one is unclear.[1,3,5,6] The deviation of ILs from Nernst-Einstein behavior could also be attributed to partial charge transfer between ions, as the Nernst-Einstein equation assumes that the charges on the ions are integers.[3,5–8]

S2 Ionic Liquids Dataset

The data preprocessing procedure used in this study can be found in Figure 1.

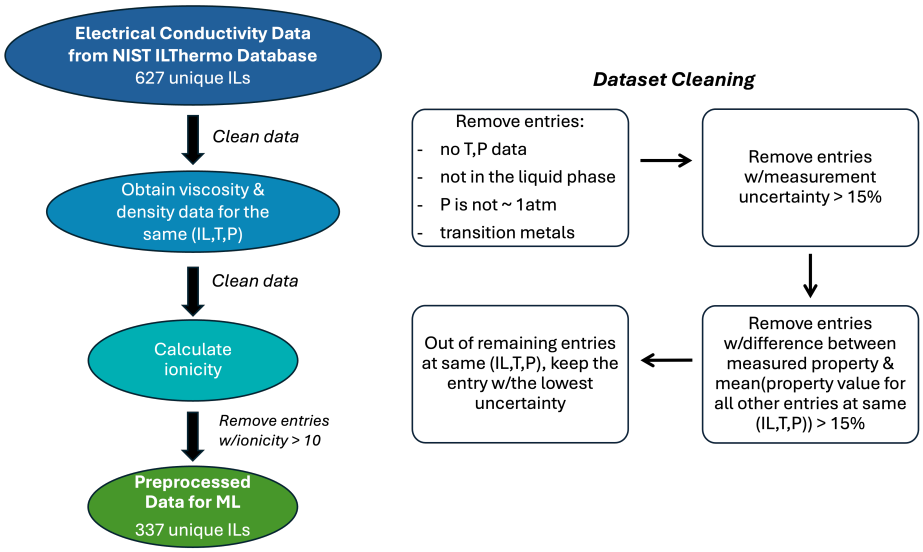

Figure 1: Overview of ILs dataset preprocessing workflow.

S2.1 Cation Family Classification

To evaluate the diversity of ILs present in the dataset, each IL was classified into a given cation family. The cation family for each IL was determined by using the RGroupDecompose function in RDKit and matching the SMILES strings for each IL cation with predefined SMILES strings for each cation family.[9] In the case that some cation families overlap (ex: imidazolium and ammonium), the IL was classified as belonging to the less general cation family. The structures of the cation families used in this classification process are given in Fig. 2. The distribution of unique ILs by cation family is given in Fig. 3.

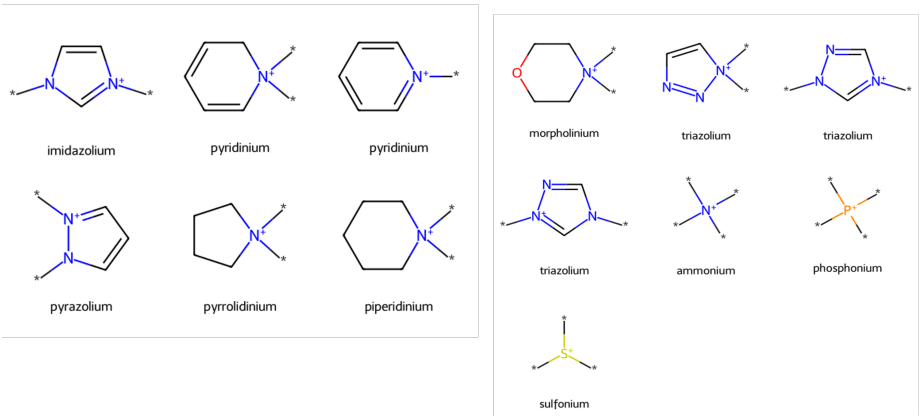

Figure 2: Structures for each cation family used in classifying ILs by cation family. The "*" corresponds to any atom.

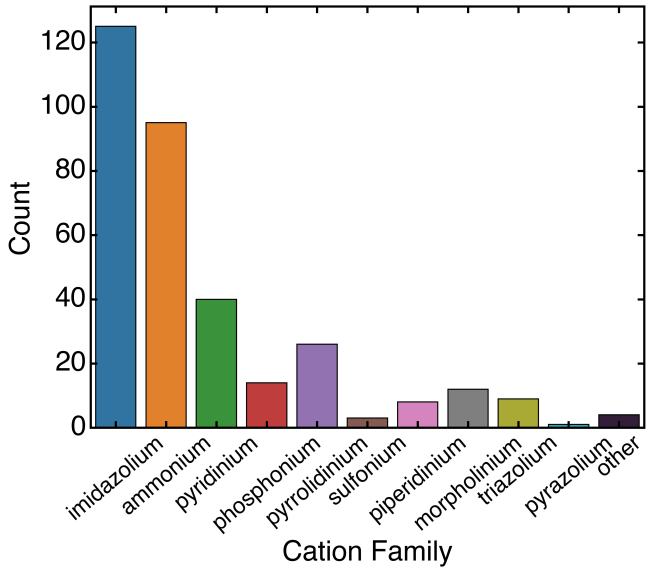

Figure 3: Distribution of IL cation families for the 337 unique ILs studied in this work.

S2.2 Ionicity Estimation

As one of the motivations of this work was to develop a tool for estimating ionicity that could be used in a high-throughput screening of ILs, an approach for accurately estimating the ionic radii of ILs that could be easily incorporated into this workflow was needed. Here, we compared different approaches for accurately estimating the ionic radii of ILs using three different Python packages: PubChemPy, Mordred, and RDKit.[9–11] Past studies that used viscosity data, along with the Stokes-Einstein equation to estimate the Nernst-Einstein conductivity of ILs assumed a spherical geometry and either estimated the ionic radii directly from the volume of the ion (Eq. 3), or from a ratio of the volume to the surface area (Eq. 4).[1–3]

$$r_{ion} = \sqrt[3]{\frac{3V_{ion}}{4\pi}} \tag{3}$$

$$r_{ion} = \frac{3V_{ion}}{SA_{ion}} \tag{4}$$

Where V_{ion} corresponds to the volume of the ion and SA_{ion} corresponds to the surface area of the ion. The different approaches we compared for estimating the ionic radii of ILs can be found in Table 1.

Table 1: Summary of computational approaches used to estimate the ionic radii of ILs

Python package	Function	Values used in estimating ionic radii
RDKit	ComputeMolVolume	Volume (Eq. 3)
RDKit	DCLV GetVDWVolume	Volume (Eq. 3)
RDKit	DCLV GetVDWVolume & DCLV GetSurfaceArea	Volume & Surface Area (Eq. 4)
PubChemPy	Volume3D	Volume (Eq. 3)
Mordred	McGowan volume	Volume (Eq. 3)

To determine which approach was deemed the most accurate, we compared the estimated ionicity values using each ionic radii estimation method to those obtained from experimental NMR diffusivities. Additionally, we compared the ionicity values to those reported by Nordness and Brennecke[3], where the ionic radii was estimated from ion volumes and surface areas obtained from UNIFAC parameters and Umaña *et al.*[2], where ionic radii were calculated from the estimated ion volume.[2,3] For these ionicity calculations using different ionic radii estimation methods, the ionic radii of the cation and anion were the only parameters that varied across the approaches. The results from this analysis are summarized in Fig. 4.

As shown in Fig. 4, the ionic radii estimated by Nordness and Brennecke[3] using UNIFAC parameters and a volume-surface area ratio appears to be the closest to the experimental data. However, as one of the aims of this work was

to build a tool for predicting the ionicity of ILs that could be integrated into a high throughput screening or inverse design workflow, we wanted to ensure that for any IL, the ionic radii could be estimated quickly. Although the approach using ion volumes and surface areas estimated using RDKit still appears to overestimate the ionicity of different ILs, it does so to a lesser extent than methods that only rely on the ion volume (Fig. 4). Thus, to create the ionicity dataset, we used the ratio between the ion volume and surface area. Before estimating the van der Waals volume and surface area of the ion, the geometry of the ion was first minimized using the MMFF94 force field. The RDKit cheminformatics package was used for the geometry optimizations, volume, and surface area calculations.[9]

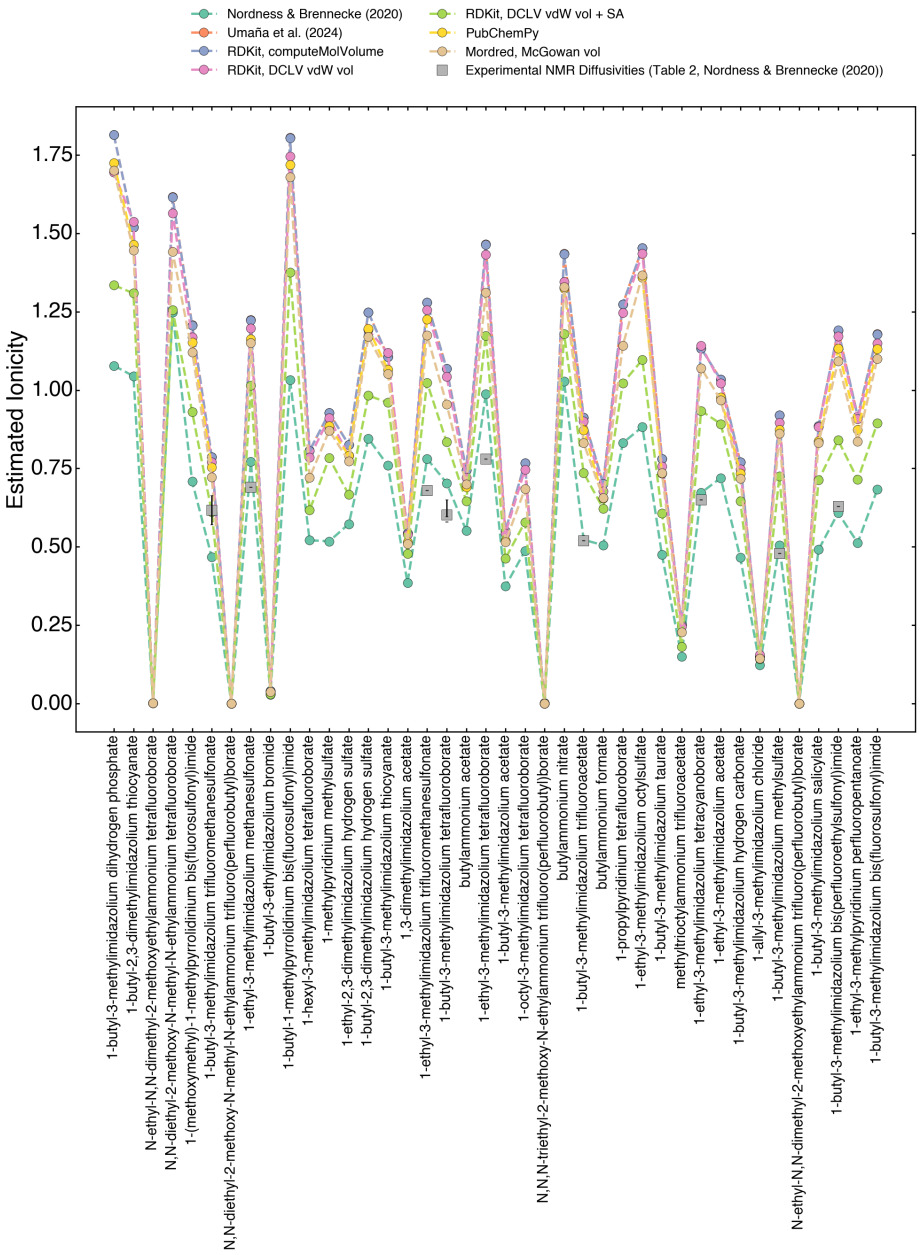

Figure 4: Estimated ionicity values for various ILs calculated using the different ionic radii estimation methods included in Table 1. Additionally, ionic radii estimates for each IL reported by Umaña *et al.*[2] (orange) and Nordness and Brennecke[3] (green) are provided.[2,3] Ionicity values estimated using experimental NMR diffusivities tabulated by Nordness and Brennecke[3] are also provided for some ILs (gray squares).[3]

S3 Sigma Profile Descriptors

For calculating the sigma profiles for each cation and anion, the openCOSMO-RS package was used with ORCA 6.0.0.[12,13] The openCOSMO-RS conformer pipeline was used to calculate the sigma profiles for each ion from an input SMILES string. This pipeline involves generating multiple conformers of each ion using RDKit, and then optimizing the geometry of each conformer with increasing levels of theory, ending with BP86/def2-TZVPD.[12]

To evaluate whether the information relating to the surface charge density of ILs was important in predicting their deviation from Nernst-Einstein behavior, sigma profiles were calculated for each cation and anion. To obtain more interpretable descriptors from the sigma profiles, the zeroth, first, second, and third moments of the sigma profile were calculated (Eq. 5).

$$M_i = \int P(\sigma) A \times \sigma^i d\sigma \tag{5}$$

Where M_i is the i^th moment of the sigma profile, $P(\sigma)$ is the probability of a given surface charge density (σ) and A is the surface area of the ion.

The zeroth moment corresponds to ion surface area. The first moment is related to the ion charge. The second moment relates to the polarity of the ion. The third moment relates to the skewness of the sigma profile.[12,14] Also, the weighted average positive sigma (WAPS) and weighted average negative sigma (WANS), which have been proposed to quantify the anion and cation charge localization, respectively, were calculated (Eqs. 6, 7).[15,16]

$$WAPS = \frac{\int_{\sigma=0}^{\infty} \sigma \cdot P(\sigma) d\sigma}{A \int_{\sigma=0}^{\infty} P(\sigma) d\sigma} \tag{6}$$

$$WANS = \frac{\int_{\sigma=-\infty}^{0} \sigma \cdot P(\sigma) d\sigma}{A \int_{\sigma=-\infty}^{0} P(\sigma) d\sigma} \tag{7}$$

Lastly, the area under the sigma profile curve over eight different intervals (σ_j to σ_{j+1}) was calculated (Eq. 8). Descriptors calculated using Eq. 8 are commonly used in ML studies and are often interpreted on the basis that a sigma profile can be broken into three regions: the positive polarity region ($\sigma < -0.082$), the negative polarity region ($\sigma > 0.082$), and the nonpolarizable region ($\sigma \in (-0.082, 0.082)$) (Fig. 5).[17–21] A table listing the specific σ_j values for each interval used in this work and the interpretation of each S_i descriptor can be found in Table 2. Generally, S_1 corresponds to the positive polarity regions of the sigma profile and an ion with a high hydrogen bond donor strength, while S_8 corresponds to the negative polarity region of the sigma profile and a strong hydrogen bond acceptor strength. Descriptors S_4 and S_5 correspond to the nonpolarizable region of the sigma profile.[18]

$$S_i = \int_{\sigma_j}^{\sigma_{j+1}} P(\sigma) \cdot A d\sigma \tag{8}$$

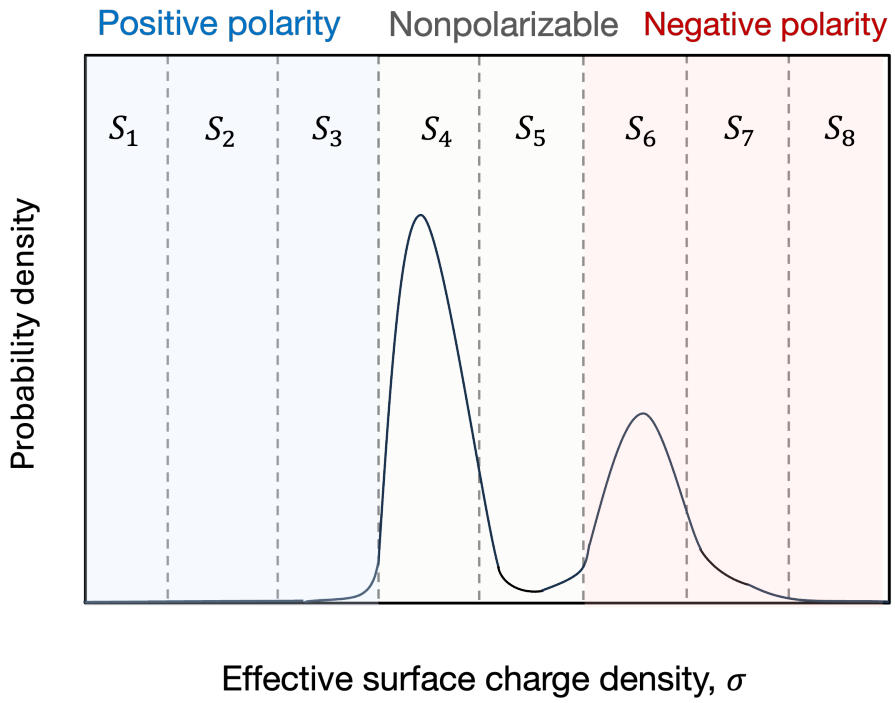

Figure 5: Illustration of a sigma profile highlighting the positive polarity (blue), nonpolarizable (gray), and negative polarity (red) regions. The eight S_i descriptors used in this work were calculated as the area under the sigma profile curve in the indicated dashed intervals shown above.)

Table 2: Corresponding σ_j values used for calculating S_i descriptors (the area under the sigma profile curve over different intervals) and the interpretation of each descriptor. The thresholds for each of the three regions (positive polarity/hydrogen bond donor, nonpolarizable, and negative polarity/hydrogen bond acceptor) in the sigma profile were based on the literature. [18]

$S_i = \int_{\sigma_j}^{\sigma_{j+1}} P(\sigma) \cdot A d\sigma$	$(\sigma_j, \sigma_{j+1})(eV/\mathring{A}^2)$	Interpretation
S_1	(-0.03,-0.0225)	Positive polarity region (hydrogen bond donor)
S_2	(-0.0225,-0.015)	
S_3	(-0.015, -0.0082)*	
S_4	(-0.0082, 0)*	Nonpolarizable region
S_5	(0, 0.0082)*	
S_6	(0.0082,0.015)*	Negative polarity region (hydrogen bond acceptor)
S_7	(0.015, 0.0225)	
S_8	(0.0225,0.03)	

* Bounds for S_3 - S_6 were adjusted to allow for S_4 and S_5 to encompass the nonpolarizable region of the sigma profile (-0.0082, 0.0082)

S4 ML Model Development

Linear models with L1 and L2 regularization, as well as decision tree-based models (random forest and XGBoost models) were trained using the scikit-learn and xgboost Python packages.[22,23] Before training any ML models, any input features that did not have a value for a given IL (i.e. NaN), had a standard deviation equal to zero, or were highly correlated (the absolute value of the correlation coefficient was greater than or equal to 0.9) in the training set were removed. Additionally, all features were scaled using scikit-learn's MinMaxScaler function such that the values for each feature lay between zero and one.[22] The same transformation that was used for scaling the training dataset was used on the test dataset.

The dataset was split such that 90% of the data was used for training the models and 10% was used for the test set. To improve the generalizability of the models, the dataset was split such that no IL was present in both the training and test datasets. Additionally, when predicting ionicity and molar conductivity, the model performance when the model was trained on y (ex: molar conductivity) versus the log transformation/log(y) (ex: log(molar conductivity)) was evaluated. When calculating all model performance metrics (MAE, MSE, etc.), the log transformation was removed.

To identify the optimal model hyperparameters, 5-fold cross-validation was used. Weights and Biases was used to track model performance[24]. The GroupKFold function in scikit-learn was used to ensure that the same IL was not present in both the training and validation set.[22] For the linear model with L1 regularization, the hyperparameter values tested ranged from 1e-6 to 1e50. For the linear model with L2 regularization, the regularization strength ranged from 1e-2 to 1e50. For both the random forest and XGBoost models, the maximum depth was varied from [1,40] and the number of estimators was varied from [1,1000]. The optimal hyperparameters for the linear models with L1 regularization and XGBoost models can be found in Tables 3,4,5,6. Information relating to the number of descriptors with nonzero coefficients for the optimal linear models with L1 regularization and the coefficient values for each descriptor can be found in Tables 7, 8, 9, 10, 11, 12, 13, 14.

The ML model performance using sigma profiles and sigma profile-derived descriptors were compared to a model trained using 2D RDKit descriptors for a baseline comparison.[9] All ML models were compared to a dummy regressor that returned the mean of the training data.

Table 3: Optimal hyperparameters for linear models with L1 regularization trained to predict the molar ionic conductivity of ILs

Input features	alpha
M_i, WAPS & WANS	0.001
RDKit	0.01
RDKit and M_i, WAPS & WANS	0.01
RDKit and S_i	0.001
RDKit and sigma profiles	0.01
S_i	0.001
S_i and M_i, WAPS & WANS	0.001
sigma profiles	0.01

Table 4: Optimal hyperparameters for XGBoost models trained to predict the molar ionic conductivity of ILs

Input features	max_depth	n_estimators
M_i, WAPS & WANS	5	720
RDKit	4	1000
RDKit and M_i, WAPS & WANS	6	720
RDKit and S_i	6	990
RDKit and sigma profiles	3	1000
S_i	13	860
S_i and M_i, WAPS & WANS	7	840
sigma profiles	4	1000

Table 5: Optimal hyperparameters for linear models with L1 regularization trained to predict the ionicity of ILs

Input features	alpha
M_i, WAPS & WANS	0.0001
RDKit	0.001
RDKit and M_i, WAPS & WANS	0.001
RDKit and S_i	0.001
RDKit and sigma profiles	0.001
S_i	0.0001
S_i and M_i, WAPS & WANS	0.001
sigma profiles	0.001

Table 6: Optimal hyperparameters for XGBoost models trained to predict the ionicity of ILs

Input features	max_depth	n_estimators
M_i, WAPS & WANS	8	80
RDKit	14	120
RDKit and M_i, WAPS & WANS	22	160
RDKit and S_i	14	70
RDKit and sigma profiles	10	110
S_i	11	130
S_i and M_i, WAPS & WANS	12	120
sigma profiles	23	90

Table 7: Comparison of the input number of descriptors and number of descriptors with nonzero coefficients after training linear models with L1 regularization to predict ionicity.

Input feature set	Number of input features	Number of features with nonzero coefficients
S_i	17	16
RDKit	155	46
M_i, WAPS & WANS	13	13

Table 8: Comparison of the input number of descriptors and number of descriptors with nonzero coefficients after training linear models with L1 regularization to predict molar ionic conductivity.

Input feature set	Number of input features	Number of features with nonzero coefficients
S_i	17	12
RDKit	155	17
M_i, WAPS & WANS	13	12

Table 9: Nonzero model coefficients corresponding to the linear model with L1 regularization that was trained to predict ionicity using S_i input descriptors.

Descriptors	Coefficients
anion S_3	-0.098
cation S_5	-1.491
anion S_2	0.197
cation S_6	-0.224
anion S_4	-0.357
cation S_2	-0.388
cation S_8	0.521
Temperature (K)	-0.401
anion S_6	-0.160
cation S_7	-0.241
Pressure (kPa)	-0.110
cation S_4	0.254
anion S_5	-0.039
anion S_7	-0.291
anion S_8	-0.318
cation S_1	0.084

Table 10: Nonzero model coefficients corresponding to the linear model with L1 regularization that was trained to predict ionicity using M_i, WAPS, and WANS input descriptors.

Descriptors	Coefficients
anion M_2	-0.784
anion M_1	-0.062
Pressure (kPa)	-0.085
cation M_1	-1.182
cation WANS	-0.791
anion $M_{hbdonor}$	0.411
cation M_0	-0.500
cation M_3	0.523
cation $M_{hbacceptor}$	0.224
anion M_0	-0.033
Temperature (K)	-0.407
anion WAPS	0.354
cation M_2	-0.795

Table 11: Nonzero model coefficients corresponding to the linear model with L1 regularization that was trained to predict ionicity using RDKit input descriptors.

Descriptors	Coefficients
Pressure (kPa)	-0.049
Temperature (K)	-0.367
MaxAbsEStateIndex cation	-0.0003
SPS cation	0.342
BCUT2D_MWLOW cation	0.091
BalabanJ cation	0.256
Kappa3 cation	-0.276
PEOE_VSA13 cation	-0.038
PEOE_VSA3 cation	0.361
PEOE_VSA6 cation	-0.207
PEOE_VSA7 cation	-0.059
PEOE_VSA8 cation	0.007
SMR_VSA3 cation	-0.190
EState_VSA3 cation	0.368
EState_VSA9 cation	-0.358
NHOHCount cation	-0.057
NumAromaticCarbocycles cation	-0.096
fr_Ar_NH cation	-0.020
fr_NH1 cation	-0.035
fr_NH2 cation	-0.082
fr_Ndealkylation1 cation	-0.053
fr_benzene cation	-0.104
fr_ester cation	0.043
fr_piperdine cation	-0.081
MinEStateIndex anion	-0.120
FpDensityMorgan1 anion	-0.162
AvgIpc anion	-0.038
Ipc anion	0.097
Kappa2 anion	-0.107
PEOE_VSA1 anion	-0.103
PEOE_VSA10 anion	-0.113
PEOE_VSA11 anion	-0.039
PEOE_VSA2 anion	0.237
SMR_VSA6 anion	-0.151
SMR_VSA7 anion	0.001
SlogP_VSA12 anion	-0.031
TPSA anion	-0.032
EState_VSA4 anion	-0.154
EState_VSA8 anion	-0.005
VSA_EState2 anion	-0.057
VSA_EState8 anion	-0.040
FractionCSP3 anion	-0.034
fr_Al_OH anion	0.152
fr_COO2 anion	-0.238
fr_aryl_methyl anion	0.969
fr_phos_acid anion	0.038

Table 12: Nonzero model coefficients corresponding to the linear model with L1 regularization that was trained to predict molar ionic conductivity using S_i input descriptors.

Descriptors	Coefficients
anion S_3	-0.261
cation S_5	-2.408
cation S_6	-0.832
anion S_4	-1.141
cation S_2	-0.807
cation S_8	1.067
cation S_3	-0.229
Temperature (K)	1.506
cation S_7	-0.515
anion S_5	0.474
anion S_7	-0.041
anion S_8	-0.954

Table 13: Nonzero model coefficients corresponding to the linear model with L1 regularization that was trained to predict molar ionic conductivity using M_i, WAPS, and WANS input descriptors.

Descriptors	Coefficients
anion M_2	-1.893
anion M_1	-0.464
Pressure (kPa)	0.069
cation M_1	-1.215
cation WANS	-1.938
anion $M_{hbdonor}$	0.707
cation M_0	-0.269
cation M_3	0.998
anion M_0	0.167
Temperature (K)	1.476
anion WAPS	0.687
cation M_2	-1.603

Table 14: Nonzero model coefficients corresponding to the linear model with L1 regularization that was trained to predict molar ionic conductivity using RDKit input descriptors.

Descriptors	Coefficients
Temperature (K)	1.246
MaxAbsEStateIndex cation	-0.654
FpDensityMorgan1 cation	0.022
BCUT2D_MWLOW cation	0.691
PEOE_VSA13 cation	-0.051
TPSA cation	-0.040
NHOHCount cation	-0.151
NumAromaticCarbocycles cation	-0.130
fr_benzene cation	-0.681
PEOE_VSA12 anion	0.048
PEOE_VSA3 anion	0.082
PEOE_VSA7 anion	-0.430
PEOE_VSA9 anion	-0.128
SMR_VSA6 anion	-0.378
RingCount anion	-0.016
fr_Al_COO anion	-0.507
fr_NH0 anion	0.517

S5 Distribution of Ionicity and Molar Conductivity Data

Histograms showing the distribution of the molar conductivity and ionicity values in the training and test dataset splits can be found in Fig. 6.

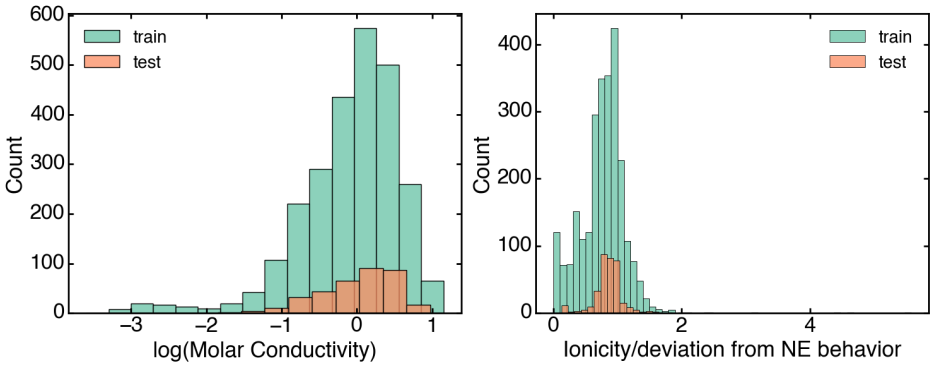

Figure 6: Distribution of molar conductivity and ionicity values in the training and test splits.

S6 Additional Model Performance Results

S6.1 ML Model Performance - 5-fold Cross-Validation and Train/Test Splits

Within each set of linear and decision tree models (linear regressor with L1 regularization, linear regressor with L2 regularization, random forest regressor, and XGBoost regressor), the linear models with L1 regularization and XGBoost models appeared to perform best. Thus, these two model types were chosen for evaluation on the test set and feature importance analysis. The average MAE for both the linear models with L1 regularization and XGBoost models trained on different sets of input features can be found in Figs. 7, 8, 9, 10, 11, 12. The MAE, MSE, and R^2 values on the training and test sets for the linear models with L1 regularization and XGBoost models can also be found in Tables 15 and 16. It should be noted that although when predicting the ionicity the average model MAE is less than that of the dummy regressor that returned the mean of the training data, the relative model improvement over the dummy regressor is smaller when predicting ionicity compared to molar conductivity (Figs. 7, 8, 9, 10). This result can be expected as the distribution of values for molar conductivity is much greater than that for ionicity (Fig. 6).

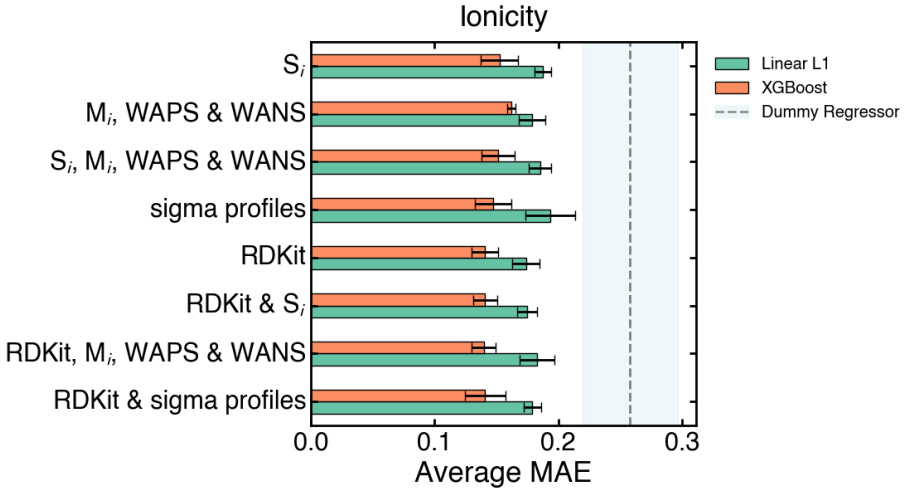

Figure 7: Average MAE on the validation set from 5-fold cross-validation for linear models with L1 regularization and XGBoost models trained to predict ionicity. Also shown is the average MAE of a dummy regressor that always returned the mean of the training set.

Training on different combinations of these input feature sets did not seem to improve the performance, suggesting that there may be an overlap in information encompassed by the two descriptor sets, despite one set (RDKit descriptors)

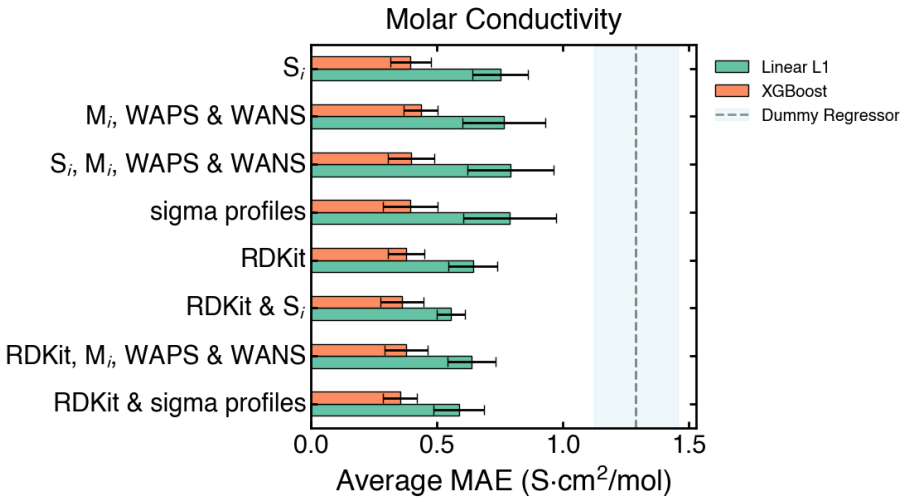

Figure 8: Average MAE on the validation set from 5-fold cross-validation for linear models with L1 regularization and XGBoost models trained to predict molar conductivity. Also shown is the average MAE of a dummy regressor that always returned the mean of the training set (for molar conductivity, models were trained on log(conductivity) and the dummy regressor returned the mean(log(conductivity)).

being based on 2D structure information and the other set (sigma profile descriptors) being based on quantum mechanical calculations. Alternatively, it is possible that additional feature selection methods may need to be used before training the models on the expanded sets due to the fact that the dataset is small.

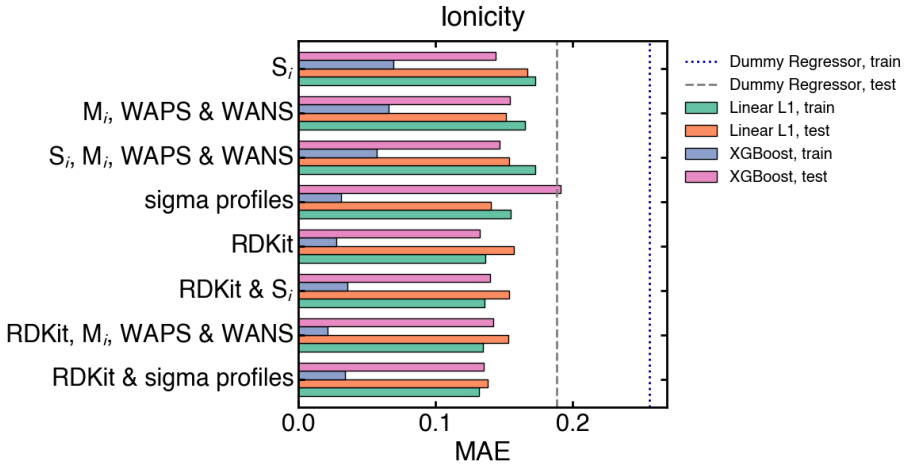

Figure 9: MAE on the training and test sets for linear models with L1 regularization and XGBoost models trained to predict ionicity using various input feature sets. Also shown is the MAE of a dummy regressor that always returned the mean of the training set

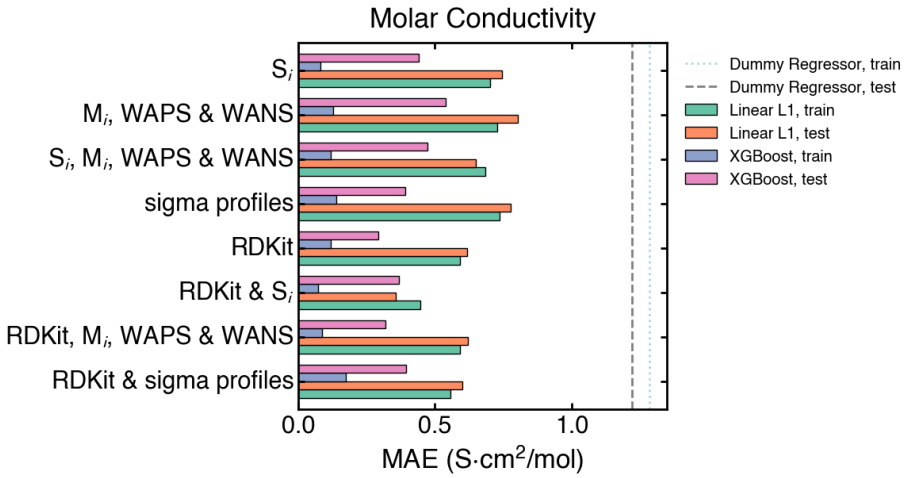

Figure 10: MAE on the training and test sets for linear models with L1 regularization and XGBoost models trained to predict molar conductivity and ionicity using various input feature sets. Also shown is the MAE of a dummy regressor that always returned the mean of the training set (for molar conductivity, models were trained on log(conductivity) and the dummy regressor returned the mean(log(conductivity))

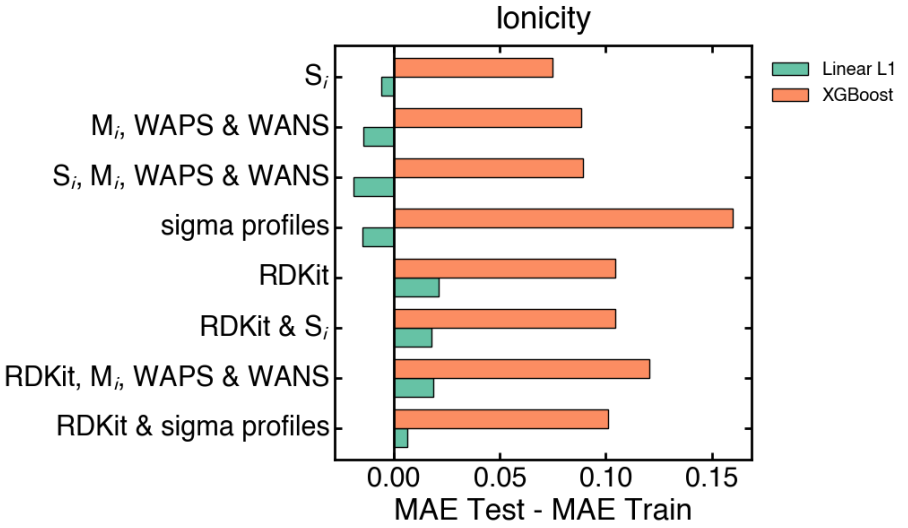

Figure 11: Difference between MAE on the training and test sets for linear models with L1 regularization and XGBoost models trained to predict ionicity using various input feature sets.

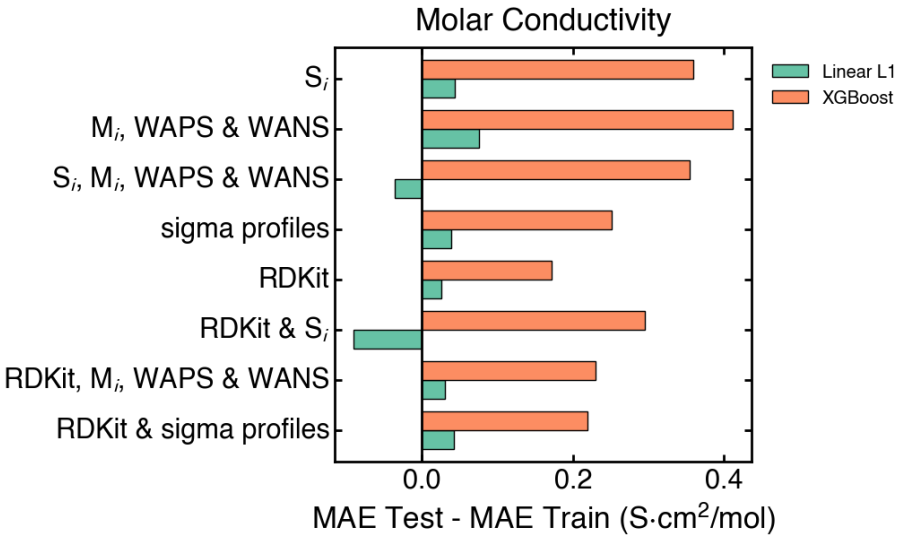

Figure 12: Difference between MAE on the training and test sets for linear models with L1 regularization and XGBoost models trained to predict molar conductivity using various input feature sets.

Table 15: Training and test set model performance (MAE, MSE, R^2) for linear models with L1 regularization and XGBoost models that were trained to predict molar ionic conductivity. MAE values are reported in S·cm^2/mol and MSE values are reported in (S·cm^2/mol)^2. Also provided is the model performance of a dummy regressor that always returned the mean of the training set (for molar conductivity, models were trained on log(conductivity) and the dummy regressor returned the mean(log(conductivity)).

Model Type	Input Features	MAE, Train	MAE, Test	MSE, Train	MSE, Test	R^2, Train	R^2, Test
Dummy Regressor	N/A	1.28	1.22	4.49	3.28	-0.216	-0.340
Linear L1	M_i, WAPS & WANS	0.727	0.803	1.68	1.64	0.544	0.329
Linear L1	RDKit	0.592	0.618	0.960	0.728	0.740	0.702
Linear L1	RDKit, M_i, WAPS & WANS	0.591	0.621	0.958	0.737	0.740	0.698
Linear L1	RDKit & S_i	0.447	0.357	1.06	0.249	0.712	0.898
Linear L1	RDKit & Sigma profiles	0.557	0.600	0.835	0.682	0.774	0.721
Linear L1	S_i	0.702	0.745	1.44	1.20	0.609	0.511
Linear L1	S_i,M_i, WAPS & WANS	0.684	0.649	1.54	1.08	0.581	0.560
Linear L1	Sigma profiles	0.738	0.776	1.75	1.47	0.525	0.398
XGBoost	M_i, WAPS & WANS	0.127	0.538	0.067	0.676	0.982	0.723
XGBoost	RDKit	0.120	0.291	0.057	0.241	0.985	0.901
XGBoost	RDKit, M_i, WAPS & WANS	0.088	0.318	0.030	0.268	0.992	0.890
XGBoost	RDKit & S_i	0.073	0.369	0.023	0.336	0.994	0.863
XGBoost	RDKit & Sigma profiles	0.174	0.393	0.125	0.366	0.966	0.850
XGBoost	S_i	0.081	0.439	0.026	0.368	0.993	0.849
XGBoost	S_i , M_i, WAPS & WANS	0.119	0.473	0.057	0.494	0.985	0.798
XGBoost	Sigma profiles	0.139	0.390	0.076	0.389	0.979	0.841

Table 16: Training and test set model performance (MAE, MSE, R^2) for linear models with L1 regularization and XGBoost models that were trained to predict ionicity. Also provided is the model performance of a dummy regressor that always returned the mean ionicity in the training set.

Model Type	Input Features	MAE, Train	MAE, Test	MSE, Train	MSE, Test	R^2, Train	R^2, Test
Dummy Regressor	N/A	0.256	0.188	0.128	0.077	0.000	-0.162
Linear L1	RDKit, M_i, WAPS & WANS	0.135	0.153	0.051	0.054	0.598	0.190
Linear L1	M_i, WAPS & WANS	0.165	0.151	0.073	0.053	0.429	0.203
Linear L1	RDKit	0.136	0.157	0.052	0.055	0.595	0.171
Linear L1	RDKit & S_i	0.136	0.154	0.052	0.053	0.595	0.198
Linear L1	RDKit & sigma profiles	0.132	0.138	0.050	0.046	0.610	0.312
Linear L1	S_i	0.173	0.167	0.076	0.067	0.407	-0.007
Linear L1	S_i, M_i, WAPS & WANS	0.173	0.154	0.075	0.061	0.411	0.086
Linear L1	Sigma profiles	0.155	0.140	0.061	0.046	0.522	0.303
XGBoost	RDKit, M_i, WAPS & WANS	0.022	0.142	0.002	0.051	0.982	0.232
XGBoost	M_i, WAPS & WANS	0.066	0.154	0.016	0.056	0.873	0.160
XGBoost	RDKit	0.028	0.132	0.004	0.047	0.971	0.294
XGBoost	RDKit & S_i	0.036	0.140	0.006	0.047	0.951	0.292
XGBoost	RDKit & sigma profiles	0.034	0.135	0.005	0.050	0.963	0.246
XGBoost	S_i	0.069	0.144	0.016	0.048	0.877	0.271
XGBoost	S_i, M_i, WAPS & WANS	0.057	0.147	0.012	0.046	0.909	0.307
XGBoost	Sigma profiles	0.031	0.191	0.005	0.099	0.959	-0.490

S6.2 ML Model Performance - Comparison to Literature

The mean squared error (MSE) for the linear model with L1 regularization and XGBoost models using RDKit descriptors to predict molar conductivity are provided in Table 17.

Table 17: MSE on both the training and test sets for linear and XGBoost models trained to predict molar conductivity. For comparison, the model performance for a neural network model trained using RDKit descriptors, experimental property descriptors (viscosity, density, heat capacity, melting point), and the combination of RDKit and experimental property descriptors by Umaña *et al.* [2] is also provided. MSE values are reported in $(S \cdot cm^2/mol)^2$.

Model Type	Input Features	Train MSE	Test MSE
Linear L1	RDKit	0.960	0.728
XGBoost	RDKit	0.057	0.241
Neural Network (Umaña *et al.* [2])	RDKit*	0.109	0.823
Neural Network (Umaña *et al.* [2])	Experimental properties	0.899	0.941
Neural Network (Umaña *et al.* [2])	RDKit* + Experimental properties	0.081	0.662

* Umaña *et al.* [2] selected a specific subset of RDKit descriptors as input to their model, whereas in this work we included all 2D RDKit descriptors for the cation and anion after removing any highly correlated descriptors.

Although the set of input RDKit descriptors used by Umaña *et al.* [2] were not the same as those used in this work, the linear and XGBoost model performance appears comparable to the neural network model trained by Umaña *et al.* [2] (Table 17).[2] The linear model, compared to the XGBoost and neural network models, also appears to be less overfit to the training dataset, and shows a slight improvement in performance on the test set compared to the training set.

Although few papers have directly used ML models to study the deviation of ILs from Nernst-Einstein behavior, Nordness and Brennecke [3] estimated the experimental uncertainty in ionicity values for ILs as being around 10%.[3]

S6.3 Ionicity Model Performance – Correction Factor to the Nernst-Einstein Equation

The accuracy of molar conductivity estimates obtained when using ionicity model predictions to correct the Nernst-Einstein equation and molar conductivity model predictions can be found in Tables 18, 19. The MAE values of the Nernst-Einstein conductivities on the training and test dataset splits were 0.795 $S \cdot cm^2/mol$ and 0.461 $S \cdot cm^2/mol$, respectively. The MSE values of the Nernst-Einstein conductivities on the training and test dataset splits were 3.191 $(S \cdot cm^2/mol)^2$ and 0.722 $(S \cdot cm^2/mol)^2$, respectively. The R^2 values of the Nernst-Einstein conductivities on the training and test dataset splits were 0.135 and 0.705, respectively.

Table 18: Comparison between molar ionic conductivity prediction accuracy (MAE, MSE, R^2) when using (a) linear models trained to predict ionicity with the Nernst-Einstein conductivities and (b) linear models trained directly on molar conductivity data. MAE values are reported in $S \cdot cm^2/mol$ and MSE values are reported in $(S \cdot cm^2/mol)^2$

.

Input Feature Set	Ionicity Models						Molar Conductivity Models					
	Train MAE	Test MAE	Train MSE	Test MSE	Train R^2	Test R^2	Train MAE	Test MAE	Train MSE	Test MSE	Train R^2	Test R^2
S_i	0.372	0.269	0.532	0.245	0.856	0.900	0.702	0.745	1.441	1.196	0.609	0.511
RDKit	0.299	0.257	0.410	0.186	0.889	0.924	0.592	0.618	0.960	0.728	0.740	0.702
M_i, WAPS & WANS	0.367	0.254	0.617	0.204	0.833	0.917	0.727	0.803	1.683	1.641	0.544	0.329

Table 19: Comparison between molar ionic conductivity prediction accuracy (MAE, MSE, R^2) when using (a) XGBoost models trained to predict ionicity with the Nernst-Einstein conductivities and (b) XGBoost models trained directly on molar conductivity data. MAE values are reported in $S \cdot cm^2/mol$ and MSE values are reported in $(S \cdot cm^2/mol)^2$

.

Input Feature Set	Ionicity Models						Molar Conductivity Models					
	Train MAE	Test MAE	Train MSE	Test MSE	Train R^2	Test R^2	Train MAE	Test MAE	Train MSE	Test MSE	Train R^2	Test R^2
S_i	0.157	0.233	0.099	0.175	0.973	0.928	0.081	0.439	0.026	0.368	0.993	0.849
RDKit	0.060	0.210	0.021	0.116	0.994	0.953	0.120	0.291	0.057	0.241	0.985	0.901
M_i, WAPS & WANS	0.147	0.274	0.105	0.216	0.971	0.912	0.127	0.538	0.067	0.676	0.982	0.723

S6.4 ML Model Performance for Imidazolium, Ammonium, and Pyridinium ILs

We report model performance for predicting molar ionic conductivity and ionicity when considering only imidazolium, ammonium, or pyridinium family ILs in each train/test split. All models were trained to predict molar ionic conductivity using the entire training set (i.e. on data that included ILs from imidazolium, ammonium, and pyridinium cation families as well as other cation families). MAE values are reported in $S \cdot cm^2/mol$ and MSE values are reported in $(S \cdot cm^2/mol)^2$. Also provided is the model performance of a dummy regressor that always returned the mean of the entire training set (for molar conductivity, models were trained on log(conductivity) and the dummy regressor returned the mean(log(conductivity)). The complete set of train/test performance metrics across all descriptor sets and models are provided in CSV format (Ionicity_TrainTest_LinearL1_XGBoost_CationFamilyPerformance.csv and MolarConductivity_TrainTest_LinearL1_XGBoost_CationFamilyPerformance.csv) at https://github.com/YueGroup/IL-Ionicity-Paper.

S7 Additional Feature Importance/Dataset Analysis Results

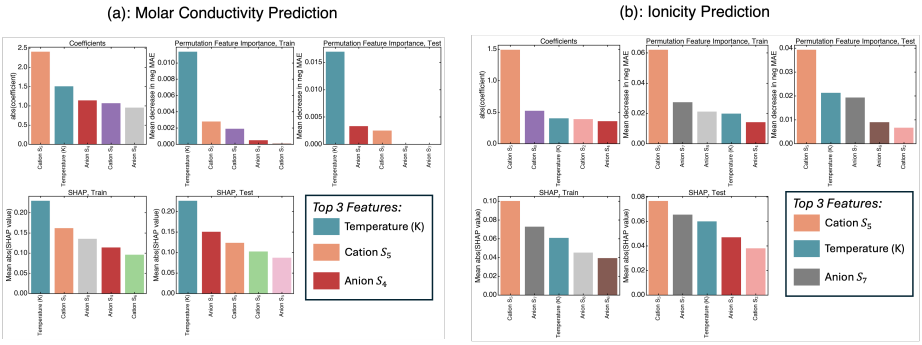

Figure 13: Feature importance rankings for (a) molar conductivity and (b) ionicity prediction using a linear L1 model trained with the following input features: S_i, Temperature, and Pressure. The S_i features correspond to the area under the sigma profile curve over the 8 intervals defined in S3. The top three features correspond to those with the highest ranking that are common across at least four of the five feature importance ranking results (model coefficients, permutation feature importance on train/test sets, SHAP analysis on train/test sets).

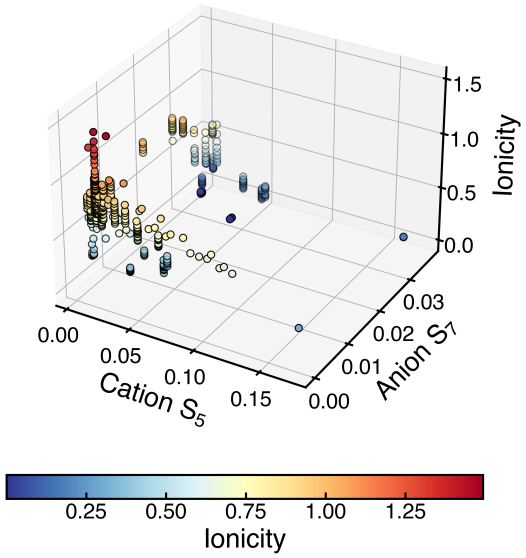

Figure 14: Three-dimensional plot of Cation S_5, Anion S_7, and ionicity for ILs in the ammonium cation family

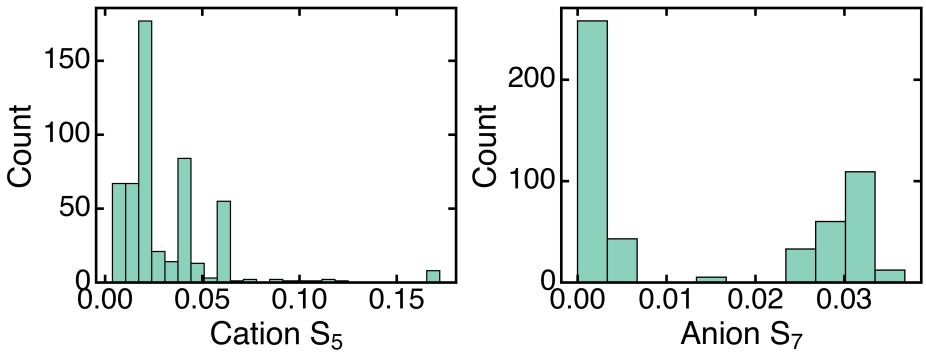

Figure 15: Distribution of values for the Cation S_5 and Anion S_7 descriptors for ILs in the ammonium cation family

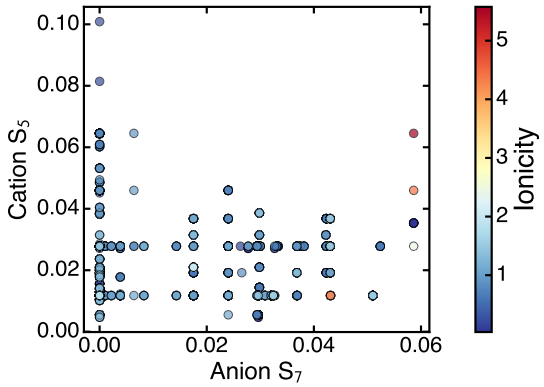

Figure 16: Relationship between the Cation S5 descriptor and the Anion S7 descriptor for ILs belonging to the imidazolium cation family. Each point corresponds to an individual IL at a given temperature and pressure and is colored by ionicity.

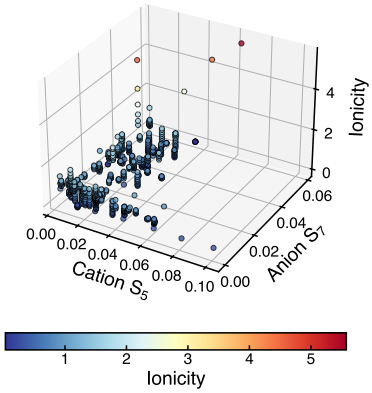

Figure 17: Three-dimensional plot of Cation S5, Anion S7, and ionicity for ILs in the imidazolium cation family

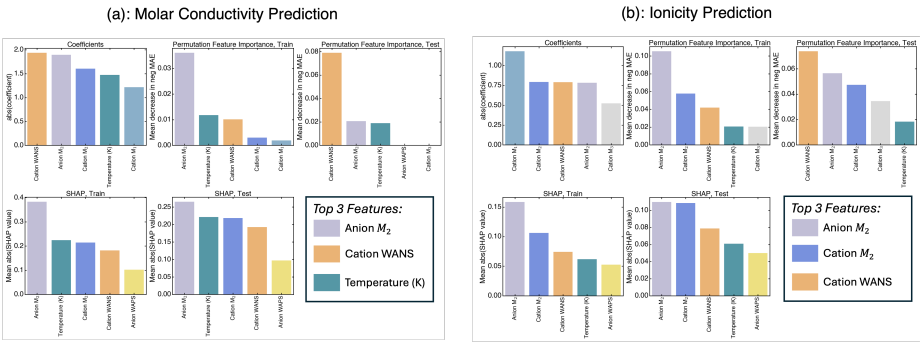

Figure 18: Feature importance rankings for (a) molar conductivity and (b) ionicity prediction using a linear L1 model trained with the following input features: M_i, Cation WANS, Anion WAPS, Temperature and Pressure. The M_i features correspond to the zeroth, first, second, and third moments of the sigma profile ($i = 0 - 3$). The Cation WANS and Anion WAPS features are considered measures of the charge delocalization of the cation and anion.[15,16] The top three features correspond to those with the highest ranking that are common across at least four of the five feature importance ranking results (model coefficients, permutation feature importance on train/test sets, SHAP analysis on train/test sets).

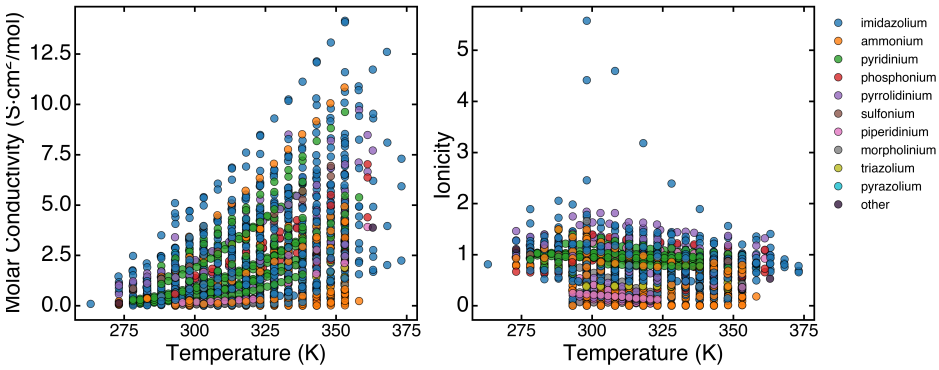

Figure 19: Molar conductivity and ionicity values for every IL at different temperatures colored by IL cation family. The pressure values for each data point were close to atmospheric pressure (101.325 ± 5 kPa)

S8 Common Functional Groups Analysis

We performed a quantitative and qualitative analysis to determine if there were common functional groups present in ILs with high or low ionicities. For this analysis, we focused on unique ILs with ionicity values corresponding to temperatures within one standard deviation of the mean temperature across the entire dataset (i.e. 315 K ± 8 K). Only one data point per IL, specifically the ionicity value corresponding to a temperature closest to 315 K, was retained. After discarding ILs with outlying temperatures (i.e. no ionicity values at temperatures within 315 K ± 8K), there were 244 unique ILs. In this set, 100 ILs had an ionicity below 0.9 (i.e. "low ionicity"), 87 had an ionicity above 1.1 (i.e. "high ionicity"), and 57 had an ionicity between 0.9 and 1.1. These thresholds were chosen based on the estimation by Nordness and Brennecke [3] that the experimental uncertainty in IL ionicity values is around 10%. [3]

We then identified functional groups in each ionic liquid using SMARTS strings for each ion and a dictionary of functional groups. [25,26] Specifically, we identified pairs of functional groups in an anion, cation pair that were predominant in ILs with a high (above 1.1) or low (below 0.9) ionicity. The results were examined qualitatively by visually comparing IL structures as well as quantitatively through a χ^2 test. For the χ^2 test, the null hypothesis is that the expected number of ILs in group i with feature k is: $F_k \cdot G_i/(G_0+G_1)$. Where i is either 0 or 1, G_0 is the number of ILs with an ionicity below 0.9, G_1 is the number of ILs with an ionicity above 1.1, and F_k is the number of ILs with cation functional group A and anion functional group B. Furthermore, to evaluate whether there was a greater similarity within ILs with a high (> 1.1) or low (< 0.9) ionicity compared to the entire set of 244 ILs, we performed a Tanimoto similarity test using vectors encoding the presence of functional groups fingerprints. [27]

For the carboxylate based anion paired with the nitrogen based cation, there were 37 ILs with low ionicity and 2 ILs with high ionicity, with a p-value of 2.2×10^{-7}.

S9 Variance Inflation Factors

We estimated variance inflation factors using the statsmodels Python package for S_i (area under the sigma profile curve) input descriptor set, and the M_i, WAPS, WANS input descriptor set.[28] For each of these sets, we only included the descriptors that had non-zero coefficients in the optimal linear models (with L1 regularization) when predicting ionicity and molar ionic conductivity. The variance inflation factors for each set of descriptors used in the conductivity and ionicity prediction tasks can be found in Tables 20, 21, 22, 23.

Table 20: Estimated variance inflation factors (VIF) for the S_i descriptor set. Only descriptors with a nonzero model coefficient in the optimal linear L1 model for predicting ionicity were considered when calculating the variance inflation factors.

Descriptor	VIF
anion S_3	2.416
cation S_5	3.459
anion S_2	1.947
cation S_6	1.712
anion S_4	1.928
cation S_2	2.045
cation S_8	3.243
Temperature (K)	1.016
anion S_6	12.325
cation S_7	3.755
Pressure (kPa)	1.168
cation S_4	4.614
anion S_5	4.088
anion S_7	18.809
anion S_8	2.458
cation S_1	1.022

Table 21: Estimated variance inflation factors (VIF) for the S_i descriptor set. Only descriptors with a nonzero model coefficient in the optimal linear L1 model for predicting molar ionic conductivity were considered when calculating the variance inflation factors.

Descriptor	VIF
anion S_3	1.428
cation S_5	1.215
cation S_6	1.893
anion S_4	1.644
cation S_2	1.659
cation S_8	3.162
cation S_3	1.558
Temperature (K)	1.007
cation S_7	3.577
anion S_5	2.148
anion S_7	3.048
anion S_8	1.716

Table 22: Estimated variance inflation factors (VIF) for the M_i, WAPS, WANS descriptor set. Only descriptors with a nonzero model coefficient in the optimal linear L1 model for predicting ionicity were considered when calculating the variance inflation factors.

Descriptor	VIF
anion M_2	2.208
anion M_1	1.892
Pressure (kPa)	1.117
cation M_1	2.002
cation WANS	7.845
anion $M_{hbdonor}$	1.284
cation M_0	4.661
cation M_3	26.718
cation $M_{hbacceptor}$	15.578
anion M_0	6.961
Temperature (K)	1.011
anion WAPS	7.828
cation M_2	37.385

Table 23: Estimated variance inflation factors (VIF) for the M_i, WAPS, WANS descriptor set. Only descriptors with a nonzero model coefficient in the optimal linear L1 model for predicting molar ionic conductivity were considered when calculating the variance inflation factors.

Descriptor	VIF
anion M_2	2.177
anion M_1	1.886
Pressure (kPa)	1.060
cation M_1	1.159
cation WANS	7.634
anion $M_{hbdonor}$	1.276
cation M_0	4.636
cation M_3	4.161
anion M_0	6.960
Temperature (K)	1.010
anion WAPS	7.790
cation M_2	2.948

Notes and references

[1] R. K. Cashen, M. M. Donoghue, A. J. Schmeiser and M. A. Gebbie, *The Journal of Physical Chemistry B*, 2022, **126**, 6039–6051.

[2] J. E. Umaña, R. K. Cashen, V. M. Zavala and M. A. Gebbie, *Digital Discovery*, 2025, **4**, 1423–1436.

[3] O. Nordness and J. F. Brennecke, *Chemical Reviews*, 2020, **120**, 12873–12902.

[4] M. Watanabe, *Electrochemistry*, 2016, **84**, 642–653.

[5] F. Philippi, D. Rauber, M. Springborg and R. Hempelmann, *The Journal of Physical Chemistry A*, 2019, **123**, 851–861.

[6] O. Hollóczki, F. Malberg, T. Welton and B. Kirchner, *Phys. Chem. Chem. Phys.*, 2014, **16**, 16880–16890.

[7] F. Philippi, K. Goloviznina, Z. Gong, S. Gehrke, B. Kirchner, A. A. H. Pádua and P. A. Hunt, *Physical Chemistry Chemical Physics*, 2022, **24**, 3144–3162.

[8] M. Kar, N. V. Plechkova, K. R. Seddon, J. M. Pringle and D. R. MacFarlane, *Australian Journal of Chemistry*, 2019, **72**, 3.

[9] G. Landrum, P. Tosco, B. Kelley, R. Rodriguez, D. Cosgrove, R. Vianello, sriniker, P. Gedeck, G. Jones, NadineSchneider, E. Kawashima, D. Nealschneider, A. Dalke, M. Swain, B. Cole, S. Turk, A. Savelev, A. Vaucher, M. Wójcikowski, I. Take, V. F. Scalfani, R. Walker, K. Ujihara, D. Probst, tadhurst cdd, guillaume godin, A. Pahl, J. Lehtivarjo, F. Bérenger and strets123, *rdkit/rdkit: 2024_03_6 (Q1 2024) Release*, 2024, `https://zenodo.org/doi/10.5281/zenodo.13469390`.

[10] H. Moriwaki, Y.-S. Tian, N. Kawashita and T. Takagi, *Journal of Cheminformatics*, 2018, **10**, 4.

[11] M. Swain, *PubChemPy*, 2025, `https://github.com/mcs07/PubChemPy`, Programmers: _:n0.

[12] S. Müller, T. Nevolianis, M. Garcia-Ratés, C. Riplinger, K. Leonhard and I. Smirnova, *Fluid Phase Equilibria*, 2025, **589**, 114250.

[13] F. Neese, *WIREs Computational Molecular Science*, 2022, **12**, e1606.

[14] A. Klamt, *COSMO-RS: from quantum chemistry to fluid phase thermodynamics and drug design*, Elsevier, Amsterdam, 1st edn, 2005.

[15] K. Kaupmees, I. Kaljurand and I. Leito, *The Journal of Physical Chemistry A*, 2010, **114**, 11788–11793.

[16] K. Kaupmees, I. Kaljurand and I. Leito, *Journal of Solution Chemistry*, 2014, **43**, 1270–1281.

[17] J. Palomar, J. S. Torrecilla, V. R. Ferro and F. Rodríguez, *Industrial & Engineering Chemistry Research*, 2008, **47**, 4523–4532.

[18] J. Palomar, J. S. Torrecilla, J. Lemus, V. R. Ferro and F. Rodríguez, *Physical Chemistry Chemical Physics*, 2010, **12**, 1991.

[19] Z. Chen, J. Chen, Y. Qiu, J. Cheng, L. Chen, Z. Qi and Z. Song, *ACS Sustainable Chemistry & Engineering*, 2024, **12**, 6648–6658.

[20] O. Nordness, P. Kelkar, Y. Lyu, M. Baldea, M. A. Stadtherr and J. F. Brennecke, *Journal of Molecular Liquids*, 2021, **334**, 116019.

[21] T. Lemaoui, T. Eid, A. S. Darwish, H. A. Arafat, F. Banat and I. AlNashef, *Materials Science and Engineering: R: Reports*, 2024, **159**, 100798.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and Duchesnay, *Journal of Machine Learning Research*, 2011, **12**, 2825–2830.

[23] T. Chen and C. Guestrin, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco California USA, 2016, pp. 785–794.

[24] L. Biewald, *Experiment Tracking with Weights and Biases*, 2020, `https://www.wandb.com/`.

[25] C. James, D. Weininger and J. Delany, *Daylight Theory Manual*, 2011, `https://www.daylight.com/dayhtml/doc/theory/`.

[26] *Daylight>SMARTS Examples*, 2025, `https://www.daylight.com/dayhtml_tutorials/languages/smarts/smarts_examples.html`.

[27] T. T. Tanimoto, 1958.

[28] S. Seabold and J. Perktold, 9th Python in Science Conference, 2010.