Supplementary Information:

PIL-Net: a physics-informed graph convolutional network for predicting atomic multipoles

Caitlin Whitter,[a] Alex Pothen,[a] and Aurora Clark[b]

[a] Department of Computer Science, Purdue University, 305 N University St., West Lafayette, IN 47907.
Email: cwhitter@purdue.edu; apothen@purdue.edu
[b] Department of Chemistry, The University of Utah, 315 1400 E, Salt Lake City, UT 84112. Email: Aurora.Clark@utah.edu

S1 Neural Network Overview

The Linear layer is the basic building block of all neural networks. It applies a linear function to the data by taking an input vector $\mathbf{x} \in \mathbb{R}^{n_{in}}$ and transforming it into an output vector $\mathbf{y} \in \mathbb{R}^{n_{out}}$ by means of a weight matrix $\mathbf{W}$ and bias vector $\mathbf{b}$,

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}. \tag{S1}$$

Since a Linear layer can only model linear transformations, for modeling non-linear relationships, one can apply a non-linear activation function $\sigma(\cdot)$ to the linear equation,

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}). \tag{S2}$$

Non-linear activation functions aid a model in creating complex mappings between the network's inputs and outputs. There are many non-linear activation functions to choose from, each best for different types of problems, but some popular functions include ReLU, Sigmoid, and Hyperbolic Tangent. With these components, one can build increasingly intricate neural networks by composing a number of transformations.

Graph Convolutional Network A graph convolutional network (GCN)[42] is a type of graph neural network, a neural network architecture for performing machine learning on graph-structured data.

Forward Pass During the forward pass of GCN training, as the data propagates through the layers of the network, the input features are transformed via a convolution step. In this step, the convolutional layers apply trainable weights to the node features and update the feature representation of each node as an aggregate of its neighboring features. The layer-wise propagation rule for a general GCN is described by[42]

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}). \tag{S3}$$

Here $\tilde{A} = A + I_n$ refers to the adjacency matrix of the undirected input graph, including self-loops. I_n is an identity matrix of dimension n, where n is the number of nodes in the graph. The diagonal degree matrix $\tilde{D}$ has elements $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. The matrix $H^{(l)} \in \mathbb{R}^{n\times k}$ is the output from the l^{th} layer of the network, where k is the number of features in the data. We initialize $H^{(0)} = X$, the input matrix of node feature vectors. The matrix $W^{(l)}$ is a layer-specific trainable weight matrix, and $\sigma(\cdot)$ denotes the activation function.

Following convolution, the neural network applies a readout operation, such as a fully connected Linear layer, to transform the intermediate node representations into the output predictions. The loss function quantifies the distance between the predicted and target values, resulting in a loss score. This loss score is the objective function the network seeks to minimize during the backward pass.

Backward Pass To minimize the loss score, following every forward pass through the network, there is a backward pass, where the gradient of the loss score is computed with respect to the network parameters. Subsequently, the optimizer uses the direction of the gradients to update the weight and bias parameters to produce a better loss score in subsequent forward passes. The forward and backward passes repeat iteratively, for all training epochs to follow, until training is complete, as determined by some stopping criterion. Due to memory limitations on the GPUs used for computation, during training, the data are typically partitioned into mini-batches, and forward and backward computations are performed on one mini-batch of data at a time.

S2 Quadrupole and Octupole Multipole Moment Tensor Detracing Procedure

The full representation of each atomic quadrupole moment is a symmetric two-dimensional tensor of size 3×3. Prior to PIL-Net model training, to obtain the representation used by the PIL-Net model, we *detraced* the QM Dataset for Atomic Multipoles[5] (QMDFAM) atomic quadrupole moment tensors using the procedure described by the dataset curators.

$$\widehat{D}\theta_{\alpha\alpha} = \theta_{\alpha\alpha} - \frac{1}{3}\sum_{\beta}\theta_{\beta\beta}, \tag{S4}$$

where $\widehat{D}$ is the detracing operator, θ is a quadrupole tensor, and the Greek indices run over the three Cartesian dimensions x, y and z. This procedure ensures the quadrupole tensor is traceless, i.e. $(\theta_{xx} + \theta_{yy} + \theta_{zz}) = 0$.

Similarly, the full representation of each octupole moment is a symmetric three-dimensional tensor of size 3×3×3. Prior to model training, we *detraced* the octupole moment tensors using an equation that arises from the Cartesian tensor definition of the octupole multipole moment[2]

$$\widehat{D}\Omega_{\alpha\beta\beta} = \Omega_{\alpha\beta\beta} - \frac{1}{3}\sum_{\gamma}\Omega_{\alpha\gamma\gamma}, \tag{S5}$$

where $\widehat{D}$ is the detracing operator, Ω is an octupole tensor, and the Greek indices run over the three Cartesian dimensions x,

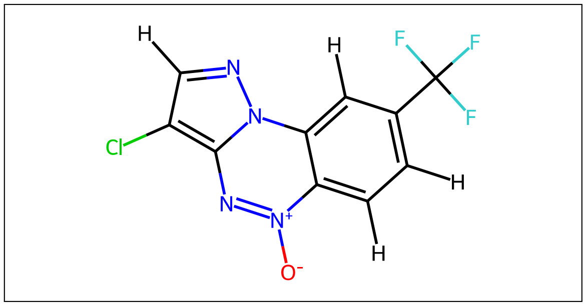

Fig. S1 Rendering of a molecule with chemical formula C10H4ClF3N4O from the QM Dataset for Atomic Multipoles[5], using the RDKit software package.[51]

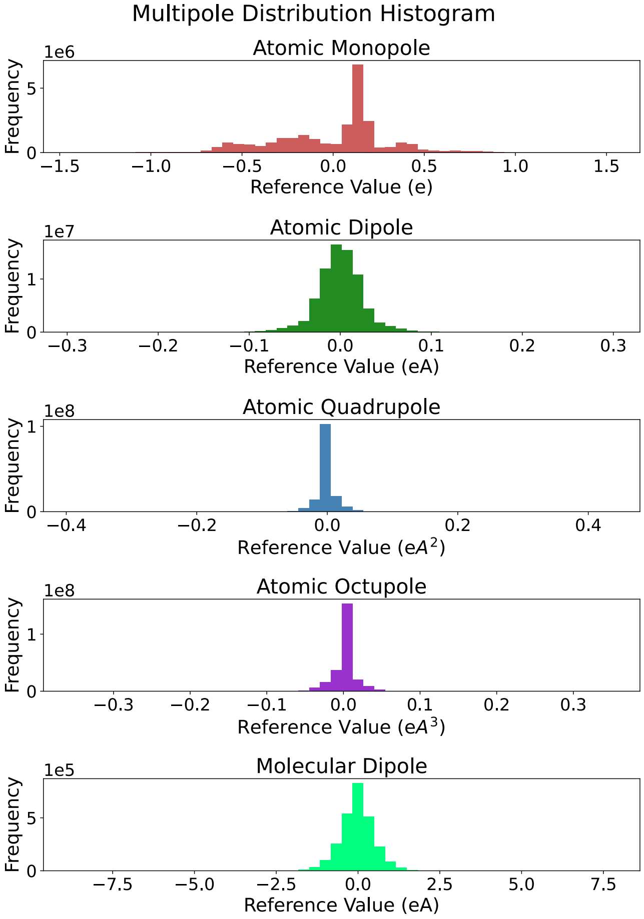

Fig. S2 Distribution of multipole moment values in the QMDFAM[5] training set. Each plot contains all the scalar values from each multipole moment vector, not the magnitude of the vector. Since the length of the vector for each multipole property can differ (e.g., 3 for dipole and 10 for octupole), this accounts for the difference in the frequency counts of the histograms.

y and z. This procedure ensures the octupole tensor is traceless, i.e. $(\Omega_{xxx} + \Omega_{xyy} + \Omega_{xzz}) = (\Omega_{xxy} + \Omega_{yyy} + \Omega_{yzz}) = (\Omega_{xxz} + \Omega_{yyz} + \Omega_{zzz}) = 0$. (The equivalent sums due to symmetry also hold.)

Subsequently, since both the quadrupole and octupole tensors are symmetric, each of their representations can be flattened to form a vector that only includes its unique components. As a result, the atomic quadrupole and octupole moments are transformed into vectors of length 6 and 10, respectively. The quadrupole vector is defined as $Q := [\theta_{xx}, \theta_{xy}, \theta_{xz}, \theta_{yy}, \theta_{yz}, \theta_{zz}]$, and the octupole vector is defined as $O := [\Omega_{xxx}, \Omega_{xxy}, \Omega_{xxz}, \Omega_{xyy}, \Omega_{xyz}, \Omega_{xzz}, \Omega_{yyy}, \Omega_{yyz}, \Omega_{yzz}, \Omega_{zzz}]$. This is the representation the atomic quadrupole and octupole moment tensors have in the QMDFAM, and it is a common way to represent these properties. Therefore, when performing the *detracing* operations, instead of applying Equations S4 and S5 to the θ and Ω tensors, we applied them to the corresponding elements in the Q and O vectors.

S3 Additional QM Dataset for Atomic Multipoles (QMDFAM) Information

Figure S1 displays a rendering of a molecule from the QMDFAM[5]. Table S1 provides additional counts and statistical details about the atoms and molecules in the QMDFAM. Figure S2 and S3 visualize the distribution of dataset values as histograms.

	QM Dataset for Atomic Multipoles
Molecular Conformations	1,013,891
Molecular Conformers	311,781
Atoms	28,609,654 / 15,241,043
Hydrogen	13,368,611
Carbon	10,899,843
Nitrogen	1,967,127
Oxygen	1,737,093
Fluorine	210,429
Sulfur	265,680
Chlorine	160,871
Min Atoms per Molecule	2 / 1
Max Atoms per Molecule	60 / 20
Mean Atoms per Molecule	29.63 / 16.16
Stdev Atoms per Molecule	4.86 / 1.90

Table S1 Further details about the QM Dataset for Atomic Multipoles. [5] This information includes the count of molecular conformations and conformers in the dataset, as well as the total atom count, broken down by element. Additionally, the table lists distribution information related to the size of molecules in the dataset. In each row with two numbers, the first number includes all atoms present in the dataset and the second number only includes the heavy atoms (excludes hydrogen).

S4 Additional PIL-Net Model Training Details

Hyperparameter Tuning To decide the neural network depth (number of stacked modules / convolutional layers) and width (number of hidden neurons per layer), we formed an 80/20 train/test split on the training set. Then, we trained a PIL-Net model for 25 epochs over the new training split and obtained the predictive results on the new test split. We trained a separate model for each tuned hyperparameter. The neural network depth was trained over the values {3, 4, 5} and neural network width over the values {64, 128, 256}. We found the values 5 and 256, respectively, to result in the best performance, when considering both predictive error and training time.

Training Procedure The weight and bias network parameters for the PIL-Net model were initialized with random values from a uniform distribution between -$\sqrt{k}$ and $\sqrt{k}$, where k is the size of

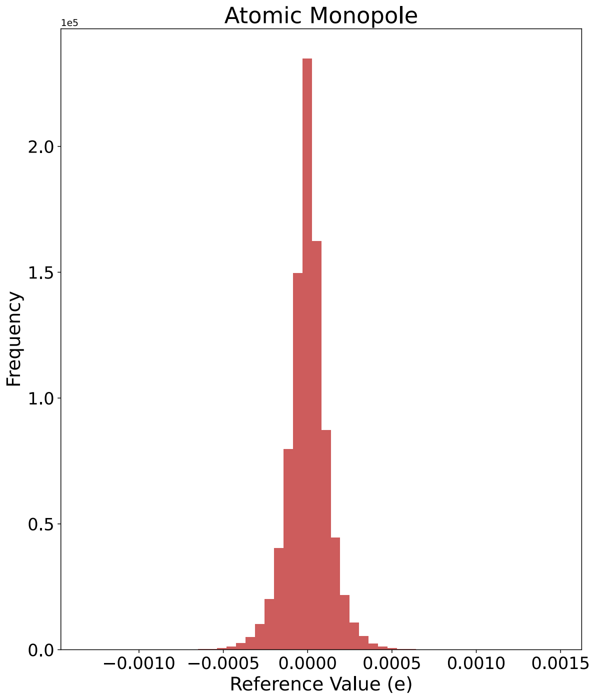

Fig. S3 Distribution of the sum of atomic monopole moment values within each molecule in the QM Dataset for Atomic Multipoles[5] training set.

the input sample, as defined by PyTorch's Linear function.[53] All models were trained using the Adam optimizer with a mini-batch size of 256.

As in the QM Dataset for Atomic Multipoles paper,[5] PIL-Net uses a single model to train over all the multipole target properties, but uses separate model weights for each property, which are updated using separate optimizers. This is due to a slight reduction in predictive error compared with a single shared set of weights and optimizer during initial experiments. The PIL-Net loss function (defined in Equation 7) scales the multipole properties to have a similar range. In the future, if we convert all the atomic multipoles to have the same scale *prior to training*, we might be able to share some of the weights across target properties, which would reduce training time.

At the beginning of training, the learning rate starts at 1E-3. To prevent overfitting, PIL-Net uses the PyTorch[53] *ReduceLROnPlateau* learning rate scheduler. We set the function arguments so that the learning rate is reduced by a factor of 0.5 if the validation loss plateaus (there is less than a 1E-4 reduction in relative loss) for more than five epochs, until the loss reaches 1E-5. For timing considerations, all computations were performed in single precision. The QMDFAM paper[5] identified that using double precision did not improve the predictive results significantly.

S5 Model Evaluation Metrics

Mean Absolute Error (MAE) is defined as the absolute difference between the predicted and target values and provides insight into the magnitude of the error

$$\mathrm{MAE}(\mathbf{X},\mathbf{Y}) = \frac{1}{nd}\sum_{i=1}^{n}\sum_{j=1}^{d}|Y_{i,j}-X_{i,j}|, \tag{S6}$$

where n is the number of rows which comprise the $\mathbf{X}$ and $\mathbf{Y}$ matrices and d is the number of columns (e.g., 16 and 3, respectively, for the matrices corresponding to the atomic dipole moments of a molecule with sixteen atoms).

Coefficient of Determination (R^2) is a measure of the goodness-of-fit of the model. This value indicates how much the variability of the reference values from their mean is explained by the model and is defined as

$$\mathrm{R}^2(\mathbf{X},\mathbf{Y}) = 1 - \frac{\sum_{i=1}^{n}\sum_{j=1}^{d}(Y_{i,j}-X_{i,j})^2}{\sum_{i=1}^{n}\sum_{j=1}^{d}(X_{i,j}-\bar{X}_j)^2}, \qquad \bar{\mathbf{X}} = \frac{1}{n}\sum_{i=1}^{n}X_i, \tag{S7}$$

where n is the number of rows that comprise the $\mathbf{X}$ (reference) and $\mathbf{Y}$ (prediction) matrices, d is the number of columns, and $\bar{\mathbf{X}}$ is the mean of $\mathbf{X}$.

Root Mean Squared Deviation (RMSD) is defined as the square root of the mean squared difference between the predicted and target values and is defined as

$$\mathrm{RMSD}(\mathbf{X},\mathbf{Y}) = \sqrt{\frac{1}{nd}\sum_{i=1}^{n}\sum_{j=1}^{d}(Y_{i,j}-X_{i,j})^2}, \tag{S8}$$

where n is the number of rows which comprise the $\mathbf{X}$ and $\mathbf{Y}$ matrices and d is the number of columns.

Pearson Correlation Coefficient (PCC) measures the linear correlation between the predicted and target values as

$$\mathrm{PCC}(\mathbf{X},\mathbf{Y}) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{d}(X_{i,j}-\bar{X}_j)(Y_{i,j}-\bar{Y}_j)}{\sqrt{[\sum_{i=1}^{n}\sum_{j=1}^{d}(X_{i,j}-\bar{X}_j)^2][\sum_{i=1}^{n}\sum_{j=1}^{d}(Y_{i,j}-\bar{Y}_j)^2]}},$$
$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n}X_i, \quad \bar{Y} = \frac{1}{n}\sum_{i=1}^{n}Y_i, \tag{S9}$$

where n is the number of rows which comprise the $\mathbf{X}$ and $\mathbf{Y}$ matrices and d is the number of columns.

S6 Additional Results Comparison between PIL-Net and AIMNet

Table S2 compares the root mean squared deviation (RMSD) for the atomic dipole and quadrupole properties for the PIL-Net and AIMNet models.

Method	**Train + Valid**	**Test**	**Elements**	**RMSD μ (eÅ)**	**RMSD θ (eÅ^2)**	**Train Time (hours)**
PIL-Net	**1M**	**13K**	**CHONFSCl**	**3.1 E-3**	**2.0 E-3**	**21**
*AIMNet[10]	9M	156K	CHONFSCl	5.4 E-3	1.27 E-2	270

Table S2 Root Mean Squared Deviation (RMSD) results for PIL-Net and AIMNet atomic dipole and quadrupole moment predictions. Section 1 and Methods subsection 2.4 provide further information about AIMNet. The symbols μ and θ correspond to atomic dipole and atomic quadrupole, respectively. For both models, the training time includes training over additional properties as well. Section 3, the Results section, includes further detail. The precision to which the AIMNet error is written reflects the precision reported in the corresponding paper. The current work's results are displayed in boldface. The standard deviation of PIL-Net's prediction error for the atomic dipole and atomic quadrupole properties was 6E-5 and 1E-5, respectively. *AIMNet's atomic dipole and quadrupole dataset values are given as norms of vectors, as opposed to full vector representations.

PIL-Net outperforms AIMNet in RMSD on the atomic dipole and quadrupole properties, particularly on the latter property. Since AIMNet used a dataset that provided the norm of the atomic dipole and quadrupole vectors, as opposed to the vectors themselves, PIL-Net's results are made more impressive, as PIL-Net is solving a more difficult computational problem.

S7 Monopole Total Charge Constraint Implementation Experiment

To examine the impact of employing a "soft" monopole total charge constraint in PIL-Net, where the forced redistribution of charge only occurs if the total charge within a molecule exceeds 10^{-2} e, we trained an additional PIL-Net model that employed a "hard" monopole total charge constraint. The latter model *always* redistributed the charge so that the sum of the atomic monopole predictions equaled exactly zero for all molecules in the training set. In the original PIL-Net models, where we only redistributed the charge if the total charge exceeded 10^{-2} e, the resulting test set mean absolute error (MAE) was 7.4 E-3 (± 5E-5) e for the atomic monopole moment property (shown in Table 2), which resulted in test set MAE 6.59 E-2 (± 6E-4) eÅ for the molecular dipole moment property (shown in Table 6). For the new PIL-Net model that implemented a hard constraint, the resulting test set error was 7.5 E-3 e for the atomic monopole moment property, a slightly larger error than in the original models. Following the application of Equation 9, this error led to test set MAE 6.71 E-2 eÅ for the molecular dipole moment property. As such, the molecular dipole moment error originating from the 10^{-2} e soft constraint was slightly smaller than the error due to the hard constraint. The current PIL-Net implementation of the monopole total charge constraint is justified due to the distribution of the QMDFAM dataset[5] (displayed in Figure S3), where the total charge does not equal zero across all the molecules. For other datasets, depending on their distributions, it may be more appropriate to implement this constraint as a hard constraint, or to use a different numerical bound for the soft constraint.

S8 Computed Uncertainty in Molecular Dipole Moment Approximation Equation

PIL-Net uses Equation 9 to approximate molecular dipole moments as a function of its atomic multipole predictions. Not only do the atomic multipole moment predictions have associated uncertainty, but Equation 9 itself does as well, since it is simply a model. The mean absolute error (MAE) of the molecular dipole moment predictions relates to the uncertainty in the input values and the uncertainty in the model as

$$MAE_{dipole} \approx \Delta_{output} \geq \sqrt{\Delta^2_{input} + \Delta^2_{model}}, \tag{S10}$$

where Δ denotes uncertainty. Equation S10 lowerbounds the value of Δ_{output} because of the potential presence of other uncertainties that contribute to the overall output uncertainty, such as numerical error.

From Table 6, the uncertainty in the Equation 9 output is $\Delta_{output} \approx MAE_{dipole} = 0.0659$ eÅ. From Table 2, the uncertainty in the Equation 9 input variables (per atom) is $\sqrt{0.0020^2 + (|-0.0467| * 0.0074)^2} = 0.0020$ eÅ, given the PIL-Net atomic monopole error 0.0074 e, atomic dipole error 0.0020 eÅ, and the average value of an atomic coordinate in the QMDFAM test set[5] -0.0467 eÅ. The average number of atoms in a QMDFAM test set molecule is 32.52, so the average uncertainty in the Equation 9 input variables (per molecule) is $\Delta_{input} =$ (32.52)(0.0020) = 0.0650 eÅ. Consequently, $\Delta_{model} \leq 0.0109$ eÅ.

These results indicate that Equation 9 is a good model for molecular dipole moment prediction. Most of the error in the output of the equation is accounted for by the error in the input variables. Reducing the uncertainty in the predicted atomic monopoles and dipoles should reduce the overall molecular dipole moment approximation error by a large amount. If we set $\Delta_{input} = 0$, we would expect $\Delta_{output} \approx 0.0109$ eÅ. However, we applied Equation 9 to the reference atomic monopole and dipole moments from the QMDFAM dataset[5] ($\Delta_{input} \approx 0$) empirically, instead of the predicted values, which resulted in much lower MAE than the theoretical value. The MAE was 0.0001 eÅ, demonstrating the utility of the model experimentally. Perhaps removing the uncertainty from the inputs also reduced numerical error, which could account for the difference in the theoretical and empirical Δ_{output} calculations.

S9 Approximation of Higher-Order Molecular Multipole Moments

In this subsection, we describe the process of approximating molecular quadrupole and octupole multipole moments from PIL-Net atomic monopole predictions, as well as using the PSI4 package[63] to compute the corresponding molecular moment reference values. Finally, we compare the approximate and reference values and report the error.

Approximation Equations Following PIL-Net model training, we used the quadrupole and octupole moment operators to approximate the corresponding higher-order molecular multipole moments with minimal additional cost. These operators are defined as a function of their respective atomic monopole moments and the given atomic position information. As these are operators, to obtain the molecular quadrupole and octupole moments in a particular quantum state $|n\rangle$, one would need to take the

	MAE	R^2
Quadrupole Moment Approx.	8.25 E-1 eÅ^2 (± 8E-3)	0.5070 (± 1E-2)
Octupole Moment Approx.	5.90 eÅ^3 (± 6E-2)	-0.8047 (± 4E-2)

Table S3 The resulting Mean Absolute Error (MAE) and Coefficient of Determination (R^2) from approximating the molecular quadrupole and octupole moments using Equations S11 and S12, as well as the corresponding standard deviations.

expectation over the operators[2], which would lead to their respective definitions in Equation 2.

The molecular quadrupole moment operator is defined as[2]

$$\hat{\theta}_{\alpha\beta} = \sum_i q_i\left(\frac{3}{2}\mathbf{r}_{i\alpha}\mathbf{r}_{i\beta} - \frac{1}{2}\mathbf{r}_i^2\delta_{\alpha\beta}\right), \tag{S11}$$

where $\hat{\theta}$ is the quadrupole operator, i iterates over all the atoms belonging to the corresponding molecule, $\mathbf{r}$ is a Cartesian atomic position vector, $\mathbf{r}^2$ denotes a dot product, δ is the Kronecker delta, and the Greek indices run over the three Cartesian dimensions x, y and z. This equation leads to a full two-dimensional symmetric representation of a size 3×3 quadrupole moment tensor. We computed Equation S11 only for the quadrupole tensor entries that form the unique components of the corresponding quadrupole vector. The mapping between quadrupole tensor and quadrupole vector is detailed in Supplementary Information Section S2.

The molecular octupole moment operator is defined as[2]

$$\hat{\Omega}_{\alpha\beta\gamma} = \sum_i q_i\left(\frac{5}{2}\mathbf{r}_{i\alpha}\mathbf{r}_{i\beta}\mathbf{r}_{i\gamma} - \frac{1}{2}\mathbf{r}_i^2(\mathbf{r}_{i\alpha}\delta_{\beta\gamma} + \mathbf{r}_{i\beta}\delta_{\alpha\gamma} + \mathbf{r}_{i\gamma}\delta_{\alpha\beta})\right), \tag{S12}$$

where $\hat{\Omega}$ is the octupole operator, i iterates over all the atoms belonging to the corresponding molecule, $\mathbf{r}$ is a Cartesian atomic position vector, $\mathbf{r}^2$ denotes a dot product, δ is the Kronecker delta, and the Greek indices run over the three Cartesian dimensions x, y and z. This equation leads to a full three-dimensional symmetric representation of a size 3×3×3 octupole moment tensor. We computed Equation S12 only for the octupole tensor entries that form the unique components of the corresponding octupole vector. The mapping between octupole tensor and octupole vector is detailed in Supplementary Information Section S2.

Reference Multipole Computation Since the QM Dataset for Atomic Multipoles (QMDFAM)[5] does not provide reference molecular quadrupole and molecular octupole multipole moments, we chose 50 molecules uniformly at random from our test set for which to generate the moments. Adapting the codes from the QMDFAM paper for computing the reference multipole moments, we computed the MBIS-derived molecular quadrupole and octupole moments for these molecules using the PSI4 package[63] at the PBE0/def2-TZVP level of theory. This PSI4 code took 3.2 hours to run for 50 molecules, or 3.9 minutes per molecule, on an Intel Xeon processor.[59] After detracing these reference values using the procedure described in Supplementary Information Section S2, the molecular quadrupole moment values occurred in range [-4.61 eÅ^2, 5.30 eÅ^2]. For the molecular octupole moments, the range was much wider, at [-40.83 eÅ^3, 42.83 eÅ^3], perhaps due to the effect of large asymmetries in the global charge distribution.

Predictive Results Table S3 displays the mean absolute error (MAE) and coefficient of determination (R^2) results, comparing the molecular quadrupole and octupole multipole moments approximated using Equations S11 and S12 and the corresponding reference values computed using the PSI4 package[63].

Looking at the molecular quadrupole moment error, while the MAE is large, the coefficient of determination, at 51%, indicates a moderate correlation between the predicted and reference values, a promising result. Contrastingly, for the molecular octupole moment, the R^2 value indicates there is no positive correlation between the predicted and reference values. Taken together, these results indicate that using the molecular quadrupole and octupole operators by themselves as approximation equations is not enough to yield excellent results.

As such, the equations will need to be modified to produce better results, while still being simple to compute. However, the two equations are good starting points. With some added complexity, they might be improved. For example, perhaps the molecular quadrupole moment equation can incorporate the atomic quadrupole predictions and the molecular octupole moment equation can incorporate the atomic octupole predictions, following a similar pattern as Equation 9. Another factor to consider is that the reference atomic monopole moments were computed at the PBE0-D3BJ/def2-TZVP level of theory, which includes the D3BJ dispersion correction method. The difference in the DFT level of theory used to compute the reference atomic multipole moments and the reference molecular quadrupole and octupole moments could account for some of the error in these approximations.

It takes a fraction of a second per molecule to compute Equation S11 and Equation S12, as opposed to 3.9 minutes per molecule when computing the reference values using the PSI4 package[63], adding to the appeal of using the former method as a starting point for obtaining the multipole moment values. While the former computations were run on a GPU and the latter a CPU, the performance gap is still evident. Currently, running the PSI4 library[63] with a GPU requires the use of external licensed software.

S10 Progression of PIL-Net Validation Loss During Training

Figure S4 depicts the average PIL-Net unweighted validation loss during training. PIL-Net has reasonable validation loss even from its first epoch of training. Then, from its initial set of predictions, the training procedure reduces the validation loss further by a factor of 4 for the atomic monopoles and by nearly an order of magnitude for the other atomic multipole moments, by the end of training. Moreover, the validation loss slope begins to flatten starting around epoch 100, indicating that training could have stopped earlier with minimal effect on error, which would be particularly beneficial in time-constrained scenarios.

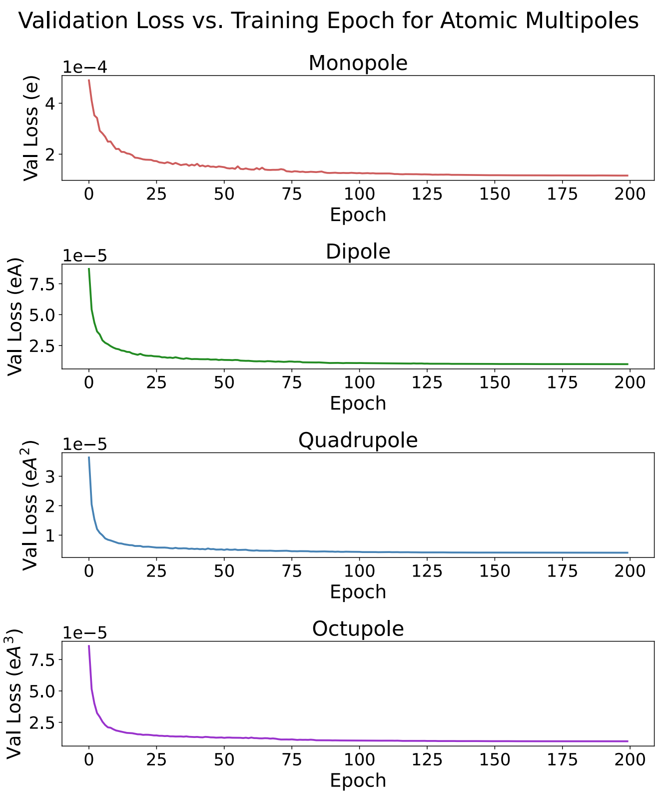

Fig. S4 Validation loss over the course of PIL-Net model training for each of the atomic multipole moment properties.

S11 Scaling Study: Effect of Training Set Size on Predictive Results

Figure S5 (mean absolute error) and S6 (coefficient of determination) present learning curve plots for each atomic multipole moment property. In these experiments, the same training/validation/test set used in the main experiments was maintained, except that the training set was sampled uniformly at random to reduce the training set to the sizes listed along the x-axis of the figures. PIL-Net models were trained using a training set of each size, and the resulting test set MAE and R^2 values are displayed in these figures. Across all these experiments, the PIL-Net architecture and hyperparameters were kept the same as in the original experiments that used the 900K training set size. The training set sizes were chosen to match those of DynamPol[8], Δ-ML-85[11], CMPNN[9], and EGNN[5]. The predictive results from these models (listed in Tables 2 and 5) were used for comparison with PIL-Net. PIL-Net's original training set size (900K) is the largest depicted in Figures S5 and S6. The AIMNet model's[10] training set size (9M) was omitted in these experiments because it is larger than the size of the entire QM Dataset for Atomic Multipoles (QMDFAM)[5]. An additional training set size (400K) was incorporated to split the difference between the two neighboring training set sizes.

As shown in Figure S5, the test set error decreases as training set size increases, as expected. Figure S6 indicates the test set R^2 increases as the training set size increases, also as expected. Interestingly, both the MAE and R^2 values begin to level around the 400K training set size, suggesting that using a smaller QMDFAM training set size could yield similar predictive results as the original 900K training set size, leading to less training time and memory usage.

Furthermore, there is additional overfitting present in the PIL-Net models trained on the smaller training subsets compared to the original training set because the PIL-Net hyperparameters (e.g., number of layers and hidden neurons) were tuned using the original 900K training set size. Tuning these hyperparameters with the smaller subset sizes should reduce overfitting in the model and further improve the MAE and R^2 results. Even so, while using their respective training set sizes, PIL-Net still outperformed the majority of the models it matched or beat in Tables 2 and 5 in terms of MAE and R^2. The only exceptions are CMPNN in MAE for the atomic dipole moment property and DynamPol in R^2 for the atomic dipole and quadrupole moment properties. These results illustrate that PIL-Net's strong performance is not solely attributable to the volume of data on which it was trained.

S12 ANI-1x Out-of-Domain Dataset Experiment

ANI-1x Dataset In this out-of-domain experiment, we demonstrate the transferability of the PIL-Net model. We used the PIL-Net models (PINN version) we trained on the QM Dataset for Atomic Multipoles (QMDFAM)[5] to make atomic multipole moment predictions on molecules from the ANI-1x dataset,[10,20,48,49]

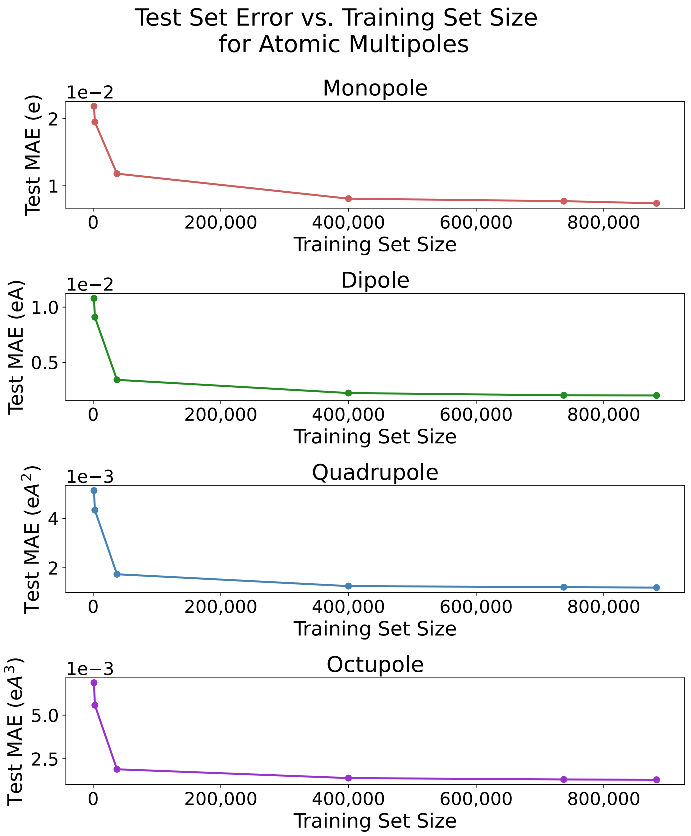

Fig. S5 Mean Absolute Error (MAE) resulting from PIL-Net models trained using different training set sizes.

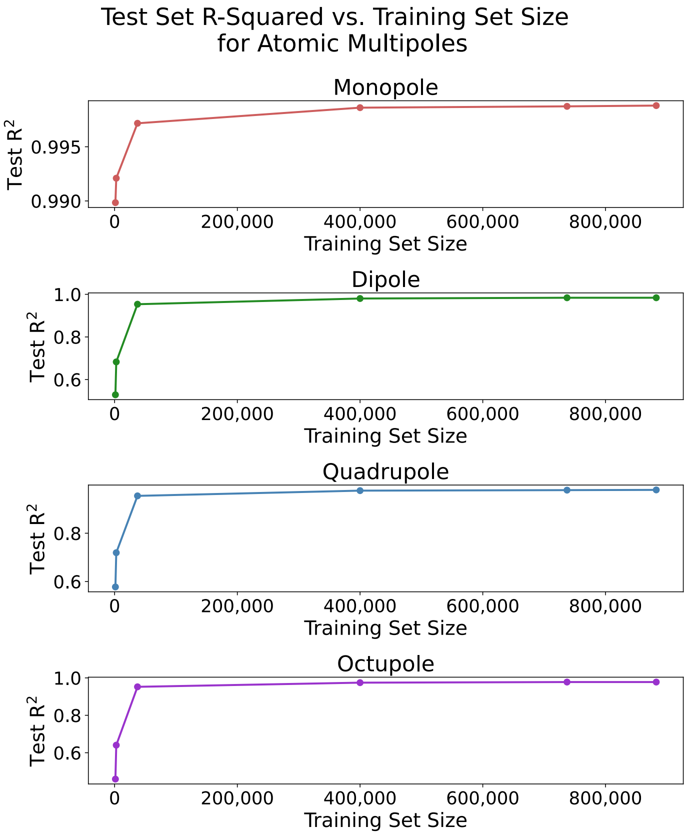

Fig. S6 Coefficient of Determination (R^2) resulting from PIL-Net models trained using different training set sizes.

and present the results. The ANI-1x dataset is a non-equilibrium dataset of 4.6M molecular conformations originating from 2.4K conformers. The ANI-1x molecules are composed of elements H, C, N, and O with mean molecule size 15.16 ± 6.07 atoms. The dataset includes nuclear charge and Cartesian coordinate data.

We applied the feature extraction process described in Methods subsection 2.2.2 to obtain the same set of features for the ANI-1x dataset. Since this dataset does not include SMILES strings for its molecules, we used the RDKit software package[51] to infer the molecules based on the provided dataset feature information. We ensured that even though ANI-1x has fewer elements than QMDFAM, the one-hot encoding on the ANI-1x nuclear charges matched that of QMDFAM in length and positioning. Furthermore, we normalized the interatomic distance feature in accordance with the training set mean and standard deviation we computed for the QM Dataset for Atomic Multipoles. The ANI-1x dataset includes label information related to several atomic multipole moment properties, computed using a variety of methods. As the ANI-1x dataset does not include atomic monopole labels, in this experiment, we compared our predictions to the dataset's MBIS atomic dipole, quadrupole, and octupole moment properties. Unlike the QMDFAM, the ANI-1x dataset reports the *magnitude* of the vector corresponding to each property instead of the vector itself. As such, during inference, once PIL-Net provided its atomic multipole predictions on the ANI-1x dataset, we computed the magnitude of the vectors in order to compare the results with the reference values included in the ANI-1x dataset.

Atomic Multipole	**MAE**	**PCC**
Dipole (μ)	4.29 E-2 (± 6E-4) eÅ	0.6943 (± 9E-3)
Quadrupole (θ)	1.06 E-1 (± 2E-3) eÅ^2	0.2458 (± 2E-1)
Octupole (Ω)	4.78 E-1 (± 9E-4) eÅ^3	0.7772 (± 4E-2)

Table S4 Mean Absolute Error (MAE) and Pearson Correlation Coefficient (PCC) results for the PIL-Net model applied to the ANI-1x dataset,[10,20,48,49] following training on the QM Dataset for Atomic Multipoles.[5]

Predictive Results In Table S4, we report our PIL-Net results from training on the QMDFAM and performing inference on ANI-1x. Comparing these results to those in Table 2, we can observe that even when predicting on a dataset from a different distribution than our training set, all the PIL-Net predictions have less error than that of Δ-ML-85[11], one of the related works. However, one must take into account that the prediction problem is simpler because these PIL-Net results are a function of norms of atomic multipole vectors (as provided by the ANI-1x dataset), not the vectors themselves. To perform further comparison of our results with those of Δ-ML-85, we also report the Pearson correlation coefficient (PCC) of our results. The PIL-Net PCC values indicate that our predictions are well-correlated with the ANI-1x dataset values for the atomic dipole and atomic octupole moment properties. Comparing with the reported Δ-ML-85[11] PCC values in Table 5, we can see that PIL-Net outperformed Δ-ML-85 in correlation for the atomic dipole property. (The Δ-ML-85 paper does not report results for the atomic octupole property.)

While the atomic quadrupole out-of-domain PCC result indicates low correlation between the predicted and reference values, the corresponding standard deviation is large at 2E-1. If we averaged these results over additional trained PIL-Net models, the PCC for the atomic quadrupole property might improve. Moreover, ANI-1x is a non-equilibrium dataset, meaning that it contains some charged molecules, and an assumption behind our monopole total charge constraint is that the molecules are neutral. To that effect, these results also point to the robustness of the PIL-Net model. In future models, making our constraints more general should only serve to improve our results further.

Therefore, we have shown that PIL-Net performs well in an out-of-domain setting. Our model not only excels at predicting multipole moments similar to those on which it has been trained, but it is also transferable to datasets from outside distributions. Training the PIL-Net model for a limited number of additional epochs, specifically on the out-of-domain dataset, may yield further improvements in the error and correlation with the out-of-domain reference values.

S13 Additional Electrostatic Potential Reconstruction Information

In addition to multipole moments and atomic coordinate data, the QM Dataset for Atomic Multipoles (QMDFAM)[5] also provides coordinates along the van der Waals (vdW) surface of the test set molecules (length 3 vector), as well as the corresponding electrostatic potential (ESP), computed at each vdW grid point (scalar value). These ESP values comprise the reference values we use for comparison with our reconstructed ESP values. To obtain the ESP on a molecular surface with a distance of two vdW radii around each molecule, the electron densities were computed from density functional theory calculations at the PBE0-D3BJ/def2-TZVP level of theory.[5] The vdW surfaces are reported in the QMDFAM in Angstrom, and the ESP values are reported in Hartree. Note that two molecules that appeared in the reference ESP dataset but did not have a matching molecule in the multipole moment test set were removed from consideration in our subsequent experiments.

The equations used to reconstruct the ESP of a molecule at each vdW surface grid point are defined as[2,5]

$$\begin{aligned}
V_{\mathrm{mon}}(j) &= \frac{1}{4\pi\varepsilon_0}\sum_i \frac{q_i}{|R_{ij}|} \\
V_{\mathrm{dip}}(j) &= V_{\mathrm{mon}}(j) \;+\; \frac{1}{4\pi\varepsilon_0}\sum_i \frac{\mu_{i,\alpha}R_{ij,\alpha}}{|R_{ij}|^3} \\
V_{\mathrm{quad}}(j) &= V_{\mathrm{dip}}(j) \;+\; \frac{1}{4\pi\varepsilon_0}\sum_i \frac{\theta_{i,\alpha\beta}(3R_{ij,\alpha}R_{ij,\beta} - R_{ij}^2\delta_{\alpha\beta})}{2|R_{ij}|^5} \\
V_{\mathrm{oct}}(j) &= V_{\mathrm{quad}}(j) \;+ \\
&\quad \frac{1}{4\pi\varepsilon_0}\sum_i \frac{\Omega_{i,\alpha\beta\gamma}(15R_{ij,\alpha}R_{ij,\beta}R_{ij,\gamma} - R_{ij}^2(R_{ij,\alpha}\delta_{\beta\gamma} + R_{ij,\beta}\delta_{\alpha\gamma} + R_{ij,\gamma}\delta_{\alpha\beta}))}{6|R_{ij}|^7},
\end{aligned} \tag{S13}$$

where $V_{\mathrm{mon}}(j)$, $V_{\mathrm{dip}}(j)$, $V_{\mathrm{quad}}(j)$, and $V_{\mathrm{oct}}(j)$ represent the reconstructed ESP up to and including the corresponding multipole moment order, and the index j corresponds to the j-th point on the vdW surface of the molecule. For example, $V_{\mathrm{quad}}(j)$ represents the ESP at j-th gridpoint on the molecular vdW surface due to the atomic monopole, atomic dipole, and atomic quadrupole moment contributions. The index i corresponds to the i-th atom in the molecule. The variables q, μ, θ, and Ω correspond to the atomic monopole, dipole, quadrupole, and octupole moments, respectively. The vector R_{ij} is computed as R_j - R_i, where R_j is the Cartesian coordinate vector of the j-th gridpoint along the vdW surface and R_i is the Cartesian coordinate vector of the i-th atom in the molecule. R_{ij}^2 corresponds to the dot product between the R_i and R_j vectors. The Greek indices α, β, and γ run over the three Cartesian dimensions x, y and z, and δ is the Kronecker delta. The prefactor corresponds to Coulomb's constant, where ε_0 is the vacuum permittivity constant. Coulomb's constant has value $\approx$1389.35 $\frac{kJ\cdot Å}{e^2\cdot mol}$. If the prefactor is expressed in these units, the computed ESP will have $\frac{kJ}{mol}$ units, assuming a test charge of +1e.