Electronic Supplementary Material (ESI) for Physical Chemistry Chemical Physics.

Automatic Identification of Chemical Moieties

(SUPPLEMENTARY INFORMATION)

Jonas Lederer, Michael Gastegger, Kristof T. Schütt
Michael Kampffmeyer, Klaus-Robert Müller, Oliver T. Unke

S1 DETERMINING COVALENT BONDS FROM 3D MOLECULE STRUCTURE

Our approach utilizes RDKit [1] to determine the covalent bonds of each molecule from its 3D structure. This is achieved by constructing a RDKit molecule object from the atomic positions and atomic numbers. Assuming a non-charged molecule, this information is sufficient for RDKit to derive the structural formula of the molecule. Subsequently, the covalent bonds are obtained by applying the module rdkit.Chem.rdmolops.GetAdjacencyMatrix.

S2 BREADTH-FIRST SEARCH

In this section, we describe how to obtain the hard similarity matrix $\mathbf{C}_\mathrm{h}$ (introduced in Section 2.3 of the main text) by utilizing a breadth-first search. Latter is performed on the graph represented by $\mathbf{C}_\mathrm{h}^0$ (defined in eq. (5) of the main text). In $\mathbf{C}_\mathrm{h}^0$, each atom is set to be similar to its 1st-order neighbors provided that they belong to the same environment type. The 1st-order neighbors are defined by $\mathbf{A}_\mathrm{cov}$ and represent atom-pairs sharing the same covalent bond. The procedure is described by Algorithm 1.

Algorithm 1

Input: $\mathbf{C}_\mathrm{h}^0$
  $\mathbf{C}_\mathrm{h} \leftarrow \mathbf{0}$
  $\mathbf{C}'_\mathrm{h} \leftarrow \mathbf{C}_\mathrm{h}^0$
  while $\mathbf{C}_\mathrm{h}$!= $\mathbf{C}'_\mathrm{h}$ **do**
    $\mathbf{C}_\mathrm{h} \leftarrow \mathbf{C}'_\mathrm{h}$
    $\mathbf{C}'_\mathrm{h} \leftarrow \mathbf{C}_\mathrm{h}^0\mathbf{C}_\mathrm{h}$ ▷ matrix multiplication between $\mathbf{C}_\mathrm{h}^0$ and $\mathbf{C}_\mathrm{h}$
    $\mathbf{C}'_\mathrm{h} \leftarrow \Theta\left(\mathbf{C}'_\mathrm{h} - \mathbf{1}\right)$ ▷ set all entries c of $\mathbf{C}'_\mathrm{h}$ to 1, in case $c \geq 1$ and 0 otherwise
  end while
  return $\mathbf{C}_\mathrm{h}$

Θ is the Heaviside step function and $\mathbf{1}$ is a matrix of same dimension as $\mathbf{C}_\mathrm{h}$, where all entries are equal to 1. With each iteration the order of included neighbors increases until all atoms of one environment type, which are connected by a set of covalent bonds, are assigned to the same moiety. Hence, after convergence, the resultant matrix $\mathbf{C}_\mathrm{h}$ represents the hard moiety similarity matrix.

S3 SATURATION EXPERIMENT

The maximum number of environment types K and the entropy trade-off factor α both control the number of formed environment types. To facilitate fine-tuning the model, it is desirable to decouple those parameters such that the number of formed environment types only depends on α for any K above a certain threshold. Figure S1a shows that this is the case for sufficiently large α. We see a saturation of environment types with increasing K. For small values of α, however, the number of formed types is directly proportional to K. The reason for this is the decreasing signal to noise ratio with increasing size of the assignment matrix $\mathbf{S}$. For $\alpha = 0.1$ and above we consider the number of used types to be independent of K in good approximation. For the runs at $(\alpha = 0.3, K = 100)$ and $(\alpha = 0.3, K = 300)$ the respective total environment type assignments are depicted. Each bar represents the amount of atoms assigned to a particular environment type. It can be seen that the last four environment types only exhibit very few atom assignments and the effective number of used types is comparable in both settings.

Figure S1b shows the relation between used environment types and molecule size for a set of 1000 molecules at $K = 100$. For $\alpha = 0.01$ almost each atom is assigned to its own environment type. With increasing α, the number of used types for each molecule becomes less dependent on the molecule size. For $\alpha = 0.3$ the number of used types per molecule is almost independent of its size.

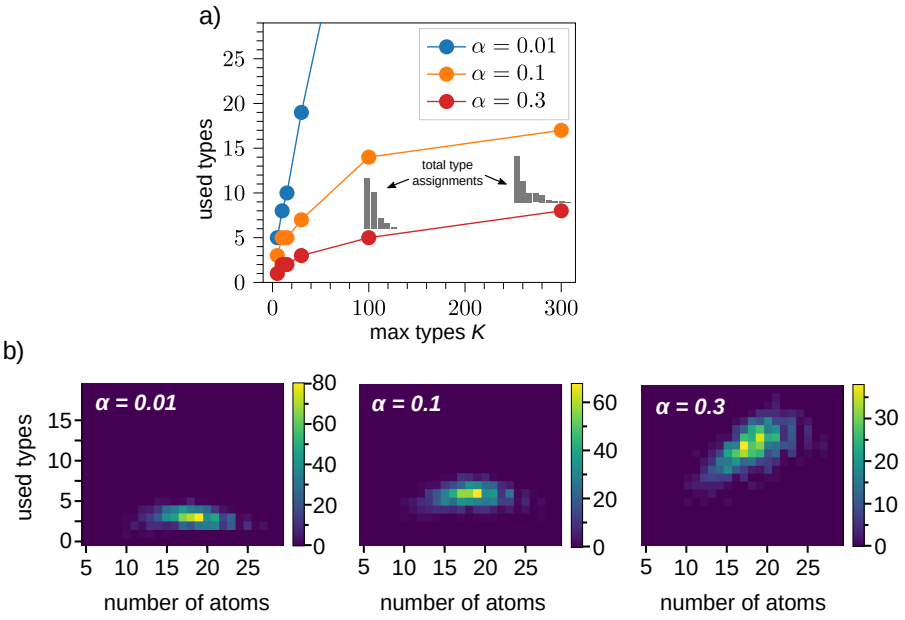

Fig. S1. Saturation analysis of the used environment types w.r.t. to the hyperparameters K and α. a) The used environment types over the maximum number of types K is depicted for three different values of the entropy trade-off factor α. For two runs, additional bar plots indicate the total number of atoms assigned to the respective environment types (each bar represents a single environment type, with its height indicating the number of atoms assigned to that type). b) The number of used environment types depending on the number of atoms in the respective molecules is depicted for three different values of α. For all three runs $K = 100$.

S4 Training of SchNet and MoINN

S4.1 Hyperparameters

The model-specific hyper-parameters of MoINN are the following:

- The cut-off radius r_{min} of the min-cut adjacency matrix $\tilde{\mathbf{A}}$ used in the loss term $\mathcal{L}_{\text{cut}}$
- The cut-off radius r_{bead}, which determines the distance dependency in similarity matrix $\mathbf{C}$
- The entropy trade-off factor α.
- The maximum number of environment types, which is determined by the cluster dimension K of the type assignment matrix.

The cut-off radius r_{bead} is associated with the expected size of moieties. r_{bead} should be chosen sufficiently small to allow for assigning distant groups of atoms to the same environment type. The cut-off radius r_{min} influences the size of the identified moieties. To substantiate those claims, we evaluate the environment types of several MoINN models trained on QM9 with different cutoff radii, while leaving the remaining hyper-parameters unchanged ($\alpha = 0.16$, $K = 100$). We perform two parameter sweeps of the cutoff radii, (i) only changing r_{min} and keeping $r_{\text{bead}} = 3.5\,\text{Å}$ constant, (ii) gradually increasing r_{min} and r_{bead} by an equal amount. For both experiments, Fig. S2 shows the relation between the number of used environment types and the cutoff radius statistically evaluated on 1000 samples from the test set. For both scenarios, the number of used environment types decreases with increasing cutoff radius. This is because with increasing r_{min}, the mincut loss term $\mathcal{L}_{\text{cut}}$ becomes increasingly dominant and at some point most molecules are represented by a single moiety. The abrupt decline in the number of used types is more prominent, when increasing both cutoff radii r_{min} and r_{bead}. The results in Fig. S2 suggest that choosing r_{min} slightly above the covalent bond distance of organic atoms, results in the best representation of the molecular graph for QM9. Figure S3 shows, for an exemplary molecule, how the identified moieties increase with increasing cutoff. The same molecule is depicted for two MoINN models trained with different r_{min} values. The remaining hyper-parameters of the two models are identical ($r_{\text{bead}} = 3.5$, $\alpha = 0.16$, $K = 100$). It can be clearly seen, that the size of the identified moieties increases with increasing mincut cutoff radius, and thus, the number of used environment types decreases.

S4.2 Training Set Up

Both, pretrained MoINN model and end-to-end MoINN model, are trained on the QM9 dataset using the same hyper-parameters $r_{\text{min}} = 1.8$, $r_{\text{bead}} = 3.5$, $\alpha = 0.16$, $K = 100$. Training set and validation set comprise 11,000 and 1,000 datapoints, respectively, while the remaining points are used for testing. In the case of end-to-end MoINN, a warmup phase (of the cut loss $\mathcal{L}_{\text{cut}}$ and entropy loss $\mathcal{L}_{\text{ent}}$ over 50 and 65 epochs, respectively) is added to increase training stability. We utilize the Adam optimizer. The learnable weights, mentioned in Eq. (1), are initialized at ($\mathbf{W}_1$) uniform and ($\mathbf{W}_2$) orthogonal. For the pretrained MoINN model, the feature representation $\mathbf{X}$ is provided by a pretrained SchNet model. Latter has been trained on the internal energy for 110,000 randomly selected molecules in the QM9 dataset. On a test set of 13,885 molecules, the internal energy is predicted well with a mean absolute error (MAE) of 0.014 eV.

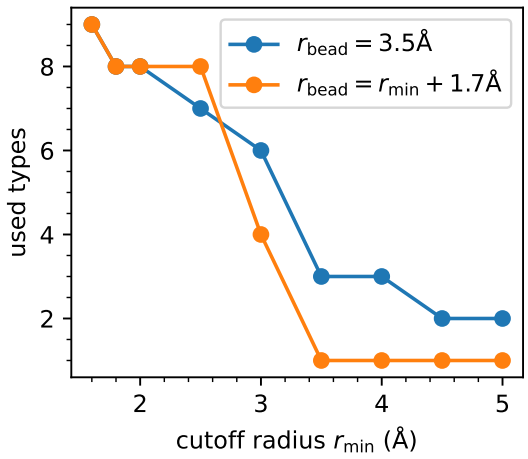

Fig. S2. The number of used environment types plotted over the cutoff radius r_{min} with (blue) constant r_{bead} and (orange) variable r_{bead}. Both scenarios depict a stable region around $r_{\text{min}} = 2\,\text{Å}$ and unreasonably low number of used environment types for $r_{\text{min}} > 3.5\,\text{Å}$. Between those two regions, we observe an abrupt decrease of used environment types, which is more prominent with variable r_{bead}.

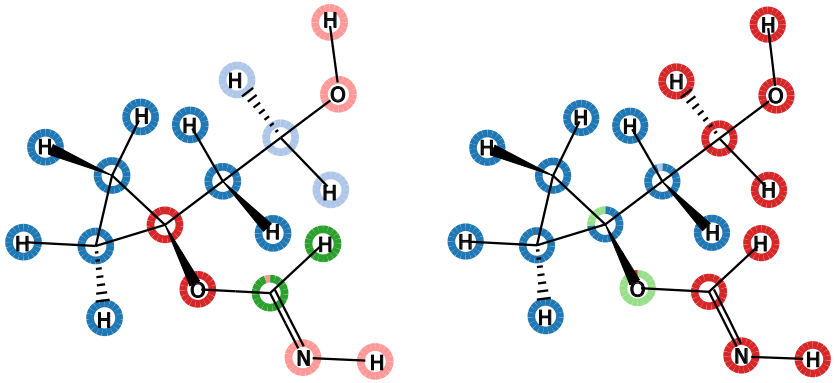

Fig. S3. Environment types of $C_6O_2NH_{11}$ obtained with two different cutoff radii, (**left**) $r_{\text{min}} = 1.6\,\text{Å}$ and (**right**) $r_{\text{min}} = 2.5\,\text{Å}$. Choosing a short mincut cutoff radius results in many small moieties, while with a larger cutoff the moiety size increases.

S4.3 Comparison between Pretrained and End-to-End MolNN

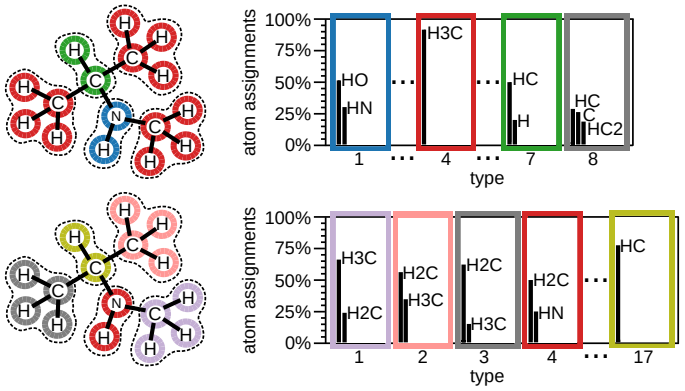

Fig. S4. Comparison between (**top**) pretrained and (**bottom**) end-to-end MolNN model. For each model the type environments and moieties are depicted (left) for an exemplary molecule and (right) in from of a statistical evaluation based on the test set.

Figure S4 depicts the provided results (type environments and moieties) of both, the pretrained model and end-to-end, model. It can be seen that the identified moieties for an exemplary molecule are equivalent for pretrained and end-to-end model. The statistical evaluation shows that for the entire test set, pretrained and end-to-end model identify similar moieties. However, while in the pretrained case, all methyl groups correspond to the same type environment, in the end-to-end case methyl groups in different molecules may be assigned to a different type. This is substantiated by the statistical evaluation, which shows that end-to-end training results in significantly more environment types used, many containing

similar moieties. This suggests that training MoINN in an end-to-end fashion results in a more fine-grained division into environment types: Slight variations in the local environment can be captured during the representation learning, while for the pretrained case $\mathbf{X}$ is fixed.

The advantage of a more fine-grained environment type division is that individual types capture more detailed information about the atomic environments. Hence, the end-to-end model is more suitable for tasks such as data reduction or providing type-based feature representations. However, for applications such as coarse-graining, using a pretrained MoINN allows to represent a greater variety of similar structural motifs with the same environment type. Furthermore, if the aim is extracting chemical insight from the dataset, it may be more natural to use a pretrained MoINN to identify chemical moieties. Intuitively, similar moieties should be associated with similar properties, regardless of their precise (fine-grained) atomic environment, which is better captured with pretrained representations.

S4.4 Training with Varying Training Set Size

The MoINN models have been trained on a relatively small set of 11,000 data points, to allow for exhaustive testing on the remaining samples. For completeness, however, we show that a larger training set of 110,000 samples results in very similar results as depicted in Fig. S5. The environment types of the four exemplary molecules show very strong alignment with those in Fig. 2. Also the statistical distribution of environment types is comparable. Some types as, e.g., the type environment number two (purple) or environment number four (red) are almost identical between the model trained on the large dataset and the model trained on the smaller training set.

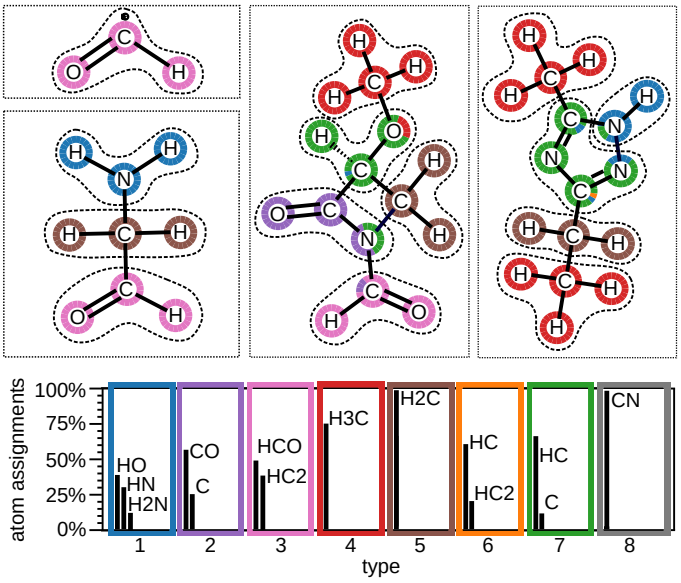

Fig. S5. Common moieties of the QM9 dataset provided by the pretrained MoINN model trained on 110k data points. Equivalent to Fig. 2, the top shows four exemplary molecules along with type assignments (colored circles) and moieties (enclosed by dashed lines). The bottom shows the distribution of environment types and corresponding most common moieties for the test set (1000 molecules), black bars indicate the relative amount of atoms assigned to the respective moieties. For each environment type, over 70% of its atom assignments correspond to at most three different moieties.

S5 Detailed Description of Show Cases and Additional Experiments

This section provides more detailed descriptions of the show cases in Section 3 as well as some additional experiments to corroborate our findings.

S5.1 Identification of Chemical Moieties

Besides identifying the most common moieties in datasets, MoINN also allows to extract information about more complex substructures such as, e.g., different molecular ring systems. Here we compare the environment types in saturated rings and aromatic rings. Figure S6 shows the average ratio of environment types in ring systems containing between five and seven heavy atoms. The ratio has been computed for the entire test set of 121,885 samples. It can be clearly seen that atoms in aromatic rings are mostly assigned to two environment types, while saturated rings exhibit several environment types. Figure S7 depicts the respective number of environment types and beads in each ring. It can be seen that aromatic rings tend to exhibit fewer environment types and individual moieties than saturated rings.

For a qualitative comparison between saturated and aromatic rings we show some exemplary molecules in Fig. S8. Equivalent to the quantitative study, the depicted aromatic and saturated rings contain between five and seven heavy

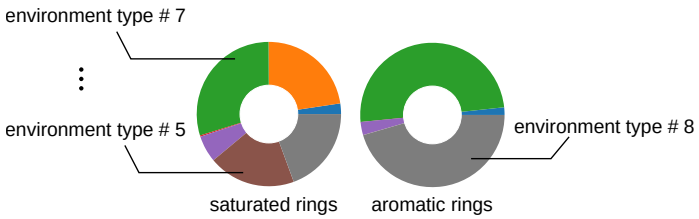

Fig. S6. Average ratio of environment types in saturated and aromatic rings. Each color represents a corresponding environment type.

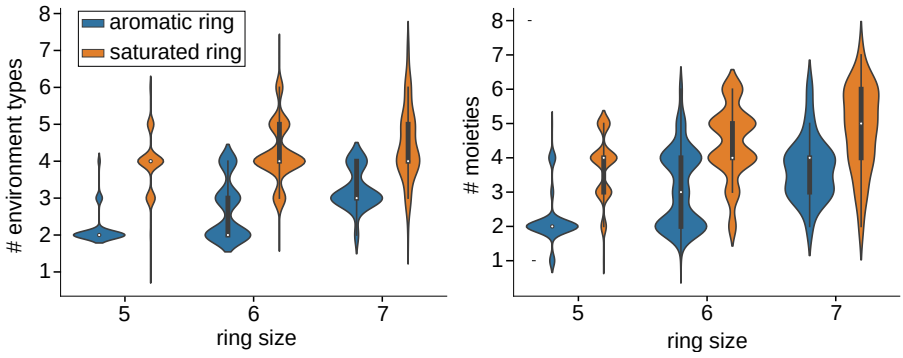

Fig. S7. Comparison between aromatic rings and saturated rings w.r.t. the number of comprised (left) environment types and number of (right) individual moieties. The number of moieties has been determined by the breadth-first search described in Section S2.

atoms. The shown examples corroborate our findings above. Hence we can conclude that MoINN distinguishes between saturated and unsaturated rings. While saturated rings are predominantly divided into several small moieties, aromatic rings are often represented as an individual entity.

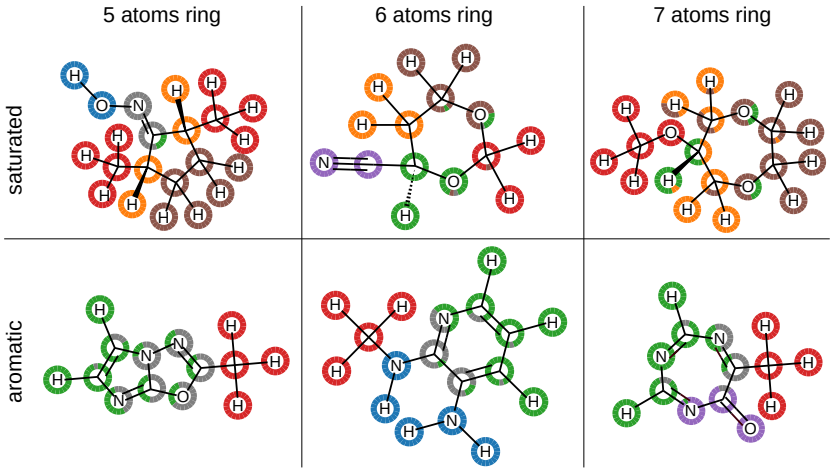

Fig. S8. Six exemplary molecules and their environment type assignments containing saturated and aromatic rings. Rings of three different sizes are compared.

The most common moieties identified by MoINN (compare Fig. 2) mostly represent small molecular substructures. However, MoINN in combination with the breadth first search (described in Section S2) also identifies larger substructures. The four largest substructures identified in the QM9 dataset are depicted in Fig. S9. Since the type assignments of MoINN strongly depend on the atomic environments, the largest structures are composed of atoms with very similar atomic environment. Hence, those moeities often comprise the entire molecule. Note that for the task of identifying common moities in the dataset it may be preferable to split up those large substructures into the most common moieties which in this case would be methylene and methine groups (compare Fig. 2). However, also less trivial large substructures are identified. Two of those are shown in Fig. S10.

S5.2 Sampling of Representative Molecules

Section 3.2. shows how MoINN can be utilized to extract representative samples from a dataset facilitating the selection of structures for expensive reference calculations. This is achieved by extracting fingerprints from the type environments and

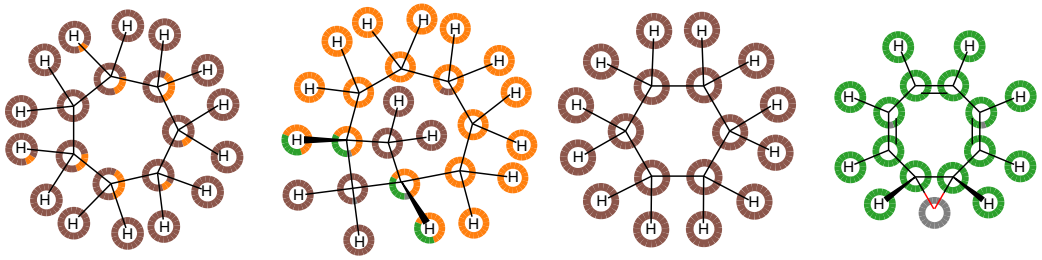

Fig. S9. Largest substructures in QM9 identified by breadth first search based on the environment types of MoINN.

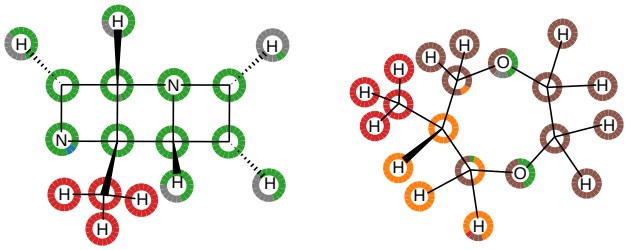

Fig. S10. Large moieties that occur alongside different moieties in the molecule.

minimizing a self reconstruction problem. This way we obtain a small basis set of structures/molecules that represents the dataset. Here we describe the details of this experiment and we extend the experiment to larger data subsets to show that for increasing training set size all approaches converge to similar performance.

The training and validation sets are drawn from a subset of 10,000 samples respectively using the respective methods (random, stratified, MoINN). Table S1 shows the different training and validation set sizes for all runs. For each training set size and validation set size we repeat the procedure five times and train corresponding SchNet models (with the hyperparameters $r_{\text{cutoff}} = 10\,\text{Å}$, $\text{batchsize} = 100$, $\text{featuresize} = 128$, $\#\text{gaussians} = 50$, $\#\text{interactions} = 3$) for 500 epochs. For the stratified sampling, the dataset is divided into bins, each bin containing molecules of equal size (same number of atoms). Subsequently, the subsets are obtained by uniformly drawing samples from the bins. For MoINN, we compare two approaches, namely, first, the one described in Section 3.2., where we solve a self reconstruction problem and, second, an approach based on medoids [3, 4]. The latter finds clusters of MoINN fingerprints and selects k medoids (cluster centers) as representative basis set. The results are shown in Fig. S11. All models are evaluated based on a test set of 10,000 samples which have been set aside before any training set selection.

TABLE S1
Training set and validation set sizes.

	set size					
training	100	200	500	1000	4000	10000
validation	100	200	500	1000	1000	1000
testing	10000	10000	10000	10000	10000	10000

It is evident that up to a training set size of 1000, MoINN sampling provides an advantage for prediction accuracy. As the training set size increases, all methods exhibit similar prediction accuracy. However, utilizing a medoid sampling based on MoINN fingerprints yields worse performance than random sampling, making it an unsuitable candidate for data reduction procedures. The self-reconstruction approach identifies fingerprints that enable the reconstruction of remaining fingerprints through linear combination, whereas the medoids approach identifies fingerprints that are the most dissimilar from each other. This validates our assumption that discovering a basis of MoINN fingerprints is superior to identifying a set of fingerprints that merely represent the variance of the dataset.

S5.3 Coarse-Graining

In Section 3.3 it is shown that MoINN can be utilized to derive CG representations. Here we compare the latter to other CG representations ranging from manually designed representations to automated frameworks. This is depicted in Figure S12 where we compare the CG representation provided by MoINN with a CG representation proposed by Wang et. al. [5], the OPLS UA representation introduced by Jorgensen et. al. [6, 7] and the automated coarse graining for the Martini force field [8] designed by Potter et. al. [9] (here referred to as Potter-Martini).

It can be seen that Potter-Martini provides the most coarse representation, followed by Wang, where the molecule is represented by the five backbone atoms. The MoINN representation is a mixture between the latter and the OPLS

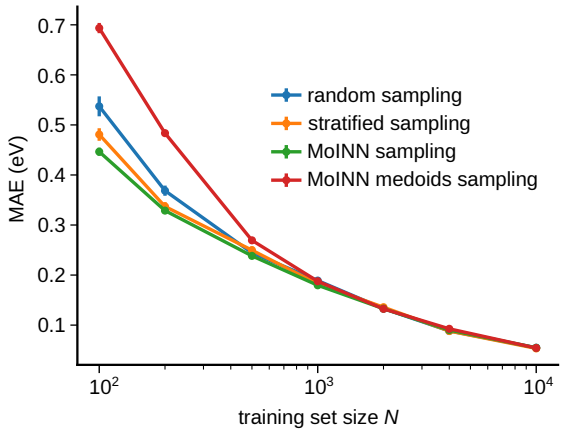

Fig. S11. Mean absolute error (MAE) of energy predictions for SchNet models trained on randomly sampled training sets (blue), training sets obtained by stratified sampling (orange) and training sets selected with MoINN in a self reconstruction manner (green) and using the k-medoids approach (red). Each data point is averaged over five independent training runs and standard errors are indicated by error bars.

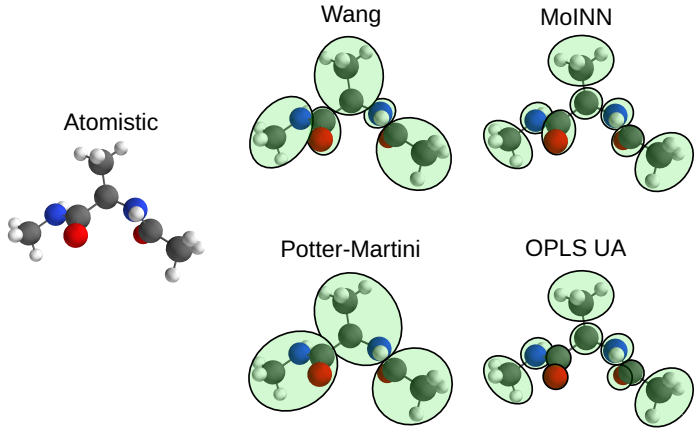

Fig. S12. Comparison between different CG representations provided by Wang et. al. [5], the automated approach for the Martini force field (here referred to as Potter-Martini), our Method MoINN and OPLS UA [6, 7].

UA representation. A clear advantage of the CG representation of MoINN is that it provides bead types. This may be particularly useful for beads that exhibit identical compositions but different local environments. We will show this on the example of decaalanine later in this Section.

To run CG-MD simulations we train several SchNet models. All considered SchNet models are trained for 300 epochs with a batch size of 100 and a learning rate $\alpha = 10^{-5}$. Learning rate scheduler ($\alpha_{\text{decay}} = 0.8$, $\alpha_{\text{patience}} = 25$) and early stopping is used to avoid overfitting. The dataset is split into 900k training samples and 100k validation samples. For the atomistic SchNet models, we choose a cutoff of $10.0\,\text{Å}$, feature size $F = 128$, 6 interaction blocks, and a basis expansion of 50 gaussians. For the coarse-grained model the cutoff is set to $5.0\,\text{Å}$, we choose feature size $F = 128$, 6 interaction blocks, and a distance expansion of 10 gaussians.

The force-matching loss function, utilized for training SchNet on the coarse-grained force field, is given by

$$\mathcal{L} = \rho \left\| \hat{U} - U \right\|^2 + \frac{1-\rho}{n} \sum_{i=0}^{n} \left\| \mathbf{C}_{\text{h}} \hat{\mathbf{F}}_i + \frac{\partial U}{\partial \mathbf{R}_i^{\text{CG}}} \right\|^2 . \tag{S1}$$

The trade-off factor is set to $\rho = 0.1$. Analogously to the training of an atomistic SchNet model, the environment types defined by MoINN are used to obtain atom type embeddings in the CG SchNet model, i.e., the CG beads are treated as pseudo-atoms. Including the energy error in the loss function is necessary for an ML model that predicts an accurate potential of mean force (PMF). Even though the PMF differs from the potential energy of the atomistic system by definition, taking the energy loss into account with a sufficiently small trade-off factor allows for fitting the forces accurately, while ensuring a reasonable energy difference between the PMF minima.

For the subsequent CG-MD simulation, we utilize the MD framework provided by SchNetPack [10]. The latter provides all necessary tools such as integrator, thermostat and logging methods. Our CG-MD simulation comprises 300 trajectories that have been initialized according to the Boltzmann distribution at the six minima of the potential energy surface. The six energy minima are determined based on the density of states in the training dataset. In detail, we select those six conformations that are closest to the maxima of the sample density and perform structure relaxations using the CG SchNet

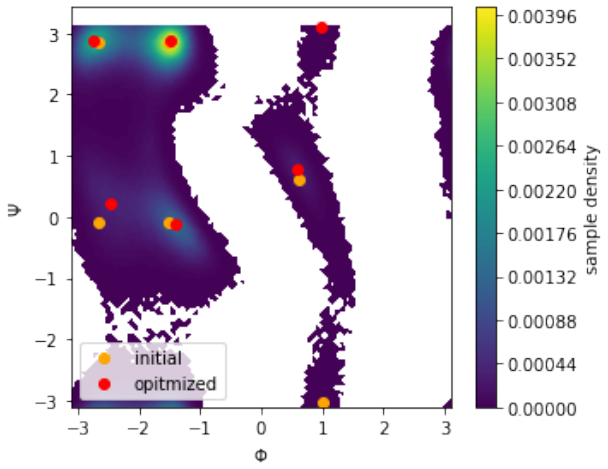

Fig. S13. Density of states of alanine dipeptide projected onto its torsion angles ϕ and ψ with indicated free energy minima. The orange (initial) dots correspond to those sates that are associated with the largest sample densities and the red (optimized) dots indicate the states corresponding to the PMF minima.

model, respectively. Figure S13 depicts the density projected to the torsion angles ψ and ϕ of alanine-dipetide and indicates the sates which represent the minimum energies of the PMF. The Boltzmann distribution

$$p_i \propto e^{-\epsilon_i / k_\mathrm{B} T}$$

describes the probability of the physical system to be in a certain state i with the corresponding energy ϵ_i. Here we sample at room temperature $T = 300\,\mathrm{K}$ and k_B denotes the Boltzmann constant. Starting from 300 initial states, we run MD simulations in the NVT setting for $8\,\mathrm{ns}$ with an integration step of $2\,\mathrm{fs}$. The thermal bath provided by the Langevin thermostat is updated each 100 steps.

In our work, we show on the example of alanine-dipeptide that MoINN can be employed to find CG representations of molecules. However, MoINN can be applied to molecules of any size due to its transferability with respect to the number of atoms. Figure S14 depicts the environment type assignments for decaalanine provided by an end-to-end MoINN model which has solely been trained on small molecules from QM9 (#heavy atoms ≤ 9). For large molecules, the larger number of different environment types associated with end-to-end MoINN models (see Section S4†), results in more meaningful moiety representations.

Similar to the case of alanine-dipeptide, the environment types can be utilized to define a coarse-grained representation of the molecule. However, molecules with #heavy atoms $\gg 9$ are likely to exhibit some atomic environments that strongly deviate from those in the QM9 dataset. This explains some undesired behaviour such as, e.g., assigning $\mathrm{NH_2}$ and $\mathrm{CH_3}$ to the same type or single carbon atoms being assigned to individual beads. Hence, finding CG-beads using only the automatic breadth first search algorithm is not recommended. Nevertheless, the identified environment types resonate with chemical intuition and tremendously facilitate selecting CG-beads. In this case, when defining the CG representation, we mainly rely on the automated breadth first search process with a few exceptions: (i) Beads are only assigned the same type if they comprise the same composition of atom types. (ii) Individual hydrogen atoms are assigned to their nearest heavy atoms. The respective bead then gets the type of the heavy atom. (iii) Only atoms, which are connected by covalent bonds, can be pooled to the same bead. As mentioned above, CG representations based on MoINN have the advantage that bead types are provided. In the case of decaalanine we can see that the terminal methyl groups are assigned to a different type then the methyl groups at the backbone of the molecule. This may facilitate learning an appropriate force field for this molecule representation.

The unintuitive environment type assignments mentioned above ($\mathrm{NH_2}$ & $\mathrm{CH_3}$ or NH & OH) are a consequence of similiar local environments and thus similar feature representations learned by the MPNN. Hence, some groups may exhibit very similar environment types while still containing a different composition of atoms. However, the soft environment type assignments allow to distinguish, different moieties, such as, e.g., NH & OH or $\mathrm{NH_2}$ & $\mathrm{CH_3}$ based on the MoINN fingerprints. While being dominantly assigned to a respective environment type, atoms can still be partly assigned to other types.

S5.4 Dynamic Clustering and Reaction Coordinate Analysis

As described in Section 3.4, we can extract reaction coordinates from the type assignments provided by MoINN. Since the variation of the structure during the reaction is also covered in the pairwise distances between atoms, also dimensionality reduction of the adjacency matrix should provide a reasonable reaction coordinate. The reaction coordinate based on MoINN is derived as described in Section 3.4. Similarly, for the soft adjacency, we define the reaction coordinate by the

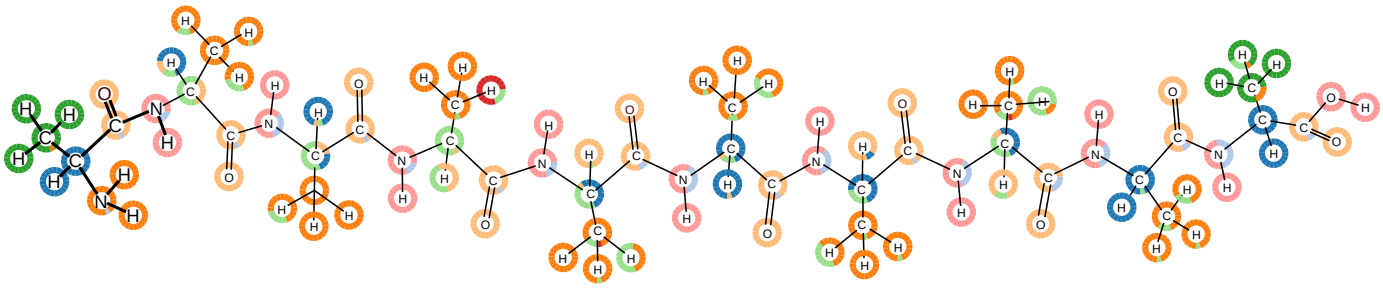

Fig. S14. Environment types for decaalanine provided by MoINN.

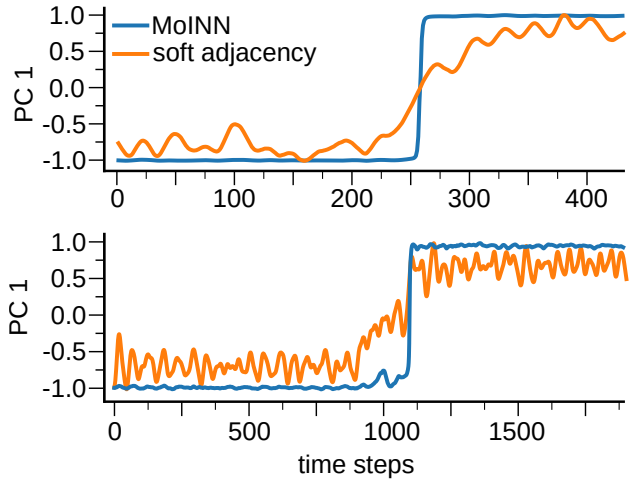

Fig. S15. Reaction coordinates for (**top**) proton transfer in malondialdehyde and (**bottom**) Claisen rearrangement based on MoINN and the soft adjacency matrix, respectively.

first principle component of the flattened adjacency matrix. Equivalent to the min-cut adjacency matrix, used in MoINN, we choose a Cosine cutoff function with the cutoff radius $r_{\text{cut}} = 1.8\,\text{Å}$ to calculate the adjacency of atom pairs.

In Figure S15, we compare reaction coordinates for malondialdehyde and the Claisen rearrangement, on the one hand based on MoINN and on the other hand relying on the adjacency matrix. We observe that, as expected, both reaction coordinates allow to distinguish between reactant and product state. However, MoINN provides a sharper representation of the state transition, while the reaction coordinate based on the soft adjacency matrix appears noisy.

S6 Limitations of Common Graph-Pooling Methods

In this section, we describe the graph-pooling methods MinCUT Pooling and DiffPool, and emphasize their limitations w.r.t. the clustering of molecules. Both approaches utilize a soft assignment matrix for coarsening graphs. They both introduce auxiliary loss terms to ensure a finite number of localized clusters. For the case of molecules, we will show that adapting their proposed loss terms allows for more reasonable clustering. Both approaches allow for hierarchical graph-pooling. For our desired applications, however, considering a single pooling step is sufficient, which is why we do not expand on the hierarchical features of MinCUT and DiffPool.

S6.1 Comprehensive Discussion of MinCUT Pooling

The concept of minCUT pooling was first stated by Bianchi et. al [11]. It describes the acquisition of an assignment matrix $\mathbf{S} \in \mathbb{R}^{N\times K}$ which is used to link N graph nodes to their respective K clusters. $\mathbf{S}$ is also often referred to as affinity matrix and is given by

$$\mathbf{S} = \text{softmax}\left(\text{ReLU}\left(\mathbf{X}\mathbf{W}_1\right)\mathbf{W}_2\right) \ . \tag{S2}$$

$\mathbf{W}_1 \in \mathbb{R}^{F\times H}$ and $\mathbf{W}_2 \in \mathbb{R}^{H\times K}$ are trainable weights matrices, with the hidden dimension H, and $\mathbf{X}$ is the feature representation. Eventually, applying the *softmax* function ensures that all cluster assignments of each row obey $\sum_j^K s_{ij} = 1$ with $s_{ij} > 0$. Thus, $\mathbf{s}_i$ represents the cluster assignment probability distribution of the ith node.

In addition to the task-specific supervised loss, an unsupervised loss is minimized. Latter is given by

$$\mathcal{L} = -\frac{Tr\left(\mathbf{S}^T\tilde{\mathbf{A}}\mathbf{S}\right)}{Tr\left(\mathbf{S}^T\tilde{\mathbf{D}}\mathbf{S}\right)} + \left\| \frac{\mathbf{S}^T\mathbf{S}}{\|\mathbf{S}^T\mathbf{S}\|_F} - \frac{\mathbf{I}_K}{\sqrt{K}} \right\|_F \ . \tag{S3}$$

$\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \in \mathbb{R}^{N\times N}$ is the symmetrically normalized adjacency matrix of the molecular graph, where $\mathbf{D} \in \mathbb{R}^{N\times N}$ denotes the degree matrix, which is a diagonal matrix with elements $d_{ii} = \sum_j^N a_{ij}$. There, $\{a_{ij}\}$ are the entries of the adjacency matrix. Consequently, $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$. $\mathbf{I}_K \in \mathbb{R}^{N\times N}$ is the identity matrix, and $\|\cdot\|_F$ is the Frobenius norm. The first term in (S3) is denoted as the cut loss term $\mathcal{L}_\mathrm{c}$ and favours clusters of adjacent nodes. To avoid converging to the trivial solution of $\mathcal{L}_\mathrm{c}$ which corresponds to assigning all nodes to the same single cluster, a second term is used in (S3). It ensures that the assignment vectors are close to orthogonal and hence it is referred to as the orthogonality loss $\mathcal{L}_\mathrm{o}$. This rewards assignments associated with clusters of balanced size. For more details refer to [11].

S6.2 Comprehensive Discussion of DiffPool

DiffPool was proposed by Ying et. al. [12]. The assignment matrix is given by

$$\mathbf{S} = \mathrm{softmax}\left(\mathrm{GNN}_\mathrm{pool}\left(\mathbf{A}, \mathbf{X}\right)\right) \ , \tag{S4}$$

where $\mathrm{GNN}_\mathrm{pool}$ is a graph neural network. Similar to the MinCUT loss, DiffPool uses an auxiliary unsupervised loss, which reads

$$\mathcal{L} = \left\|\mathbf{A} - \mathbf{S}\mathbf{S}^T\right\|_F - \frac{1}{N}\sum_{nk} S_{nk}\ln\left(S_{nk}\right) \ . \tag{S5}$$

The first term is called *Auxiliary Link Prediction Objective*, and it favours localized clusters of nodes, analogously to MinCUT's cut loss. The second term, the *Entropy Regularization*, is minimized, when the cluster assignments represent one-hot vectors. This avoids the trivial solution of assigning all nodes to a single cluster. Hence this loss term is similar to the orthogonality loss in MinCUT.

S6.3 The Issue of Symmetries in Molecules

The MinCut and DiffPool approaches are designed to avoid assigning distant nodes to the same cluster. However, (S2) and (S4) link the atomic representations to their assignments, such that nodes with a similar environment exhibit similar cluster assignments. Hence, the approaches assume that distant nodes exhibit different feature representations. This is not necessarily the case for molecules, which may be highly symmetric, leading to similar feature representations of distant nodes (atoms).

REFERENCES

[1] J.-P. Ebejer, "Conformer Generation using RDKit," p. 27.

[2] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[3] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert systems with applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[4] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[5] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. De Fabritiis, F. Noé, and C. Clementi, "Machine learning of coarse-grained molecular dynamics force fields," *ACS central science*, vol. 5, no. 5, pp. 755–767, 2019.

[6] W. L. Jorgensen and J. Tirado-Rives, "The opls [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin," *Journal of the American Chemical Society*, vol. 110, no. 6, pp. 1657–1666, 1988. PMID: 27557051.

[7] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, "Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids," *Journal of the American Chemical Society*, vol. 118, no. 45, pp. 11225–11236, 1996.

[8] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. De Vries, "The martini force field: coarse grained model for biomolecular simulations," *The journal of physical chemistry B*, vol. 111, no. 27, pp. 7812–7824, 2007.

[9] T. D. Potter, E. L. Barrett, and M. A. Miller, "Automated coarse-grained mapping algorithm for the martini force field and benchmarks for membrane–water partitioning," *Journal of Chemical Theory and Computation*, vol. 17, no. 9, pp. 5777–5791, 2021.

[10] K. T. Schütt, P. Kessel, M. Gastegger, K. Nicoli, A. Tkatchenko, and K. R. Müller, "SchNetPack: A Deep Learning Toolbox For Atomistic Systems," *Journal of Chemical Theory and Computation*, vol. 15, no. 1, pp. 448–455, 2018.

[11] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *International conference on machine learning*, pp. 874–883, PMLR, 2020.

[12] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical Graph Representation Learning with Differentiable Pooling," in *Neural Information Processing Systems*, vol. 31, pp. 4800–4810, 2018.