Chemical Science

Automatic Identification of Chemical Moieties

Journal:	*Chemical Science*
Manuscript ID	SC-EDG-04-2023-002180
Manuscript Type:	Edge Article

Chemical Science

ROYAL SOCIETY OF CHEMISTRY

Guidelines for Referees

Thank you very much for agreeing to review this manuscript for *Chemical Science*.

Chemical Science is the flagship journal from the Royal Society of Chemistry publishing findings from across the breadth of the chemical sciences.

Articles should be of exceptional significance to their field, which would also be of wider interest to readers working in other areas across the chemical sciences.

For more information about *Chemical Science* please visit our website here.

The following manuscript has been submitted for consideration as an

EDGE ARTICLE

For acceptance, an Edge article must report primary research which provides a significant new concept, advance or insight into the development of its field, and be of interest to scientists across the broad multi-disciplinary readership of the journal. There are no page limits for Edge articles, so authors can choose how best to present their research. However Electronic Supplementary Information (ESI) is recommended for additional data and experimental details.

We ask referees to **recommend only the most significant work** for publication in *Chemical Science*. When making your recommendation please:

- **Comment on** the novelty and significance to the field, the broader appeal and scientific reliability.
- **Note that routine or incremental** work should not be recommended for publication.
- **Contact the Associate Editor** if there is any conflict of interest, if the work has been previously published or if there is a significant part of the work which you are not able to referee with confidence.

Best regards,

Andrew Cooper

Professor Andrew Cooper
Editor-in-Chief, *Chemical Science*

May Copsey

Dr May Copsey
Executive Editor, *Chemical Science*

Contact us

Please visit our reviewer hub for further details of our processes, policies and reviewer responsibilities as well as guidance on how to review, or click the links below.

What to do when you review

Reviewer responsibilities

Process & policies

Faculty IV
Electrical Engineering and
Computer Science
Institute for Software Engineering
and Theoretical Computer Science
Chair Machine Learning

April 27, 2023

Dear Editor,

we hereby submit a revised version of our manuscript **“Automatic Identification of Chemical Moieties”** (ID: SC-EDG-03-2022-001853) authored by Jonas Lederer, Michael Gastegger, Kristof T. Schütt, Michael Kampffmeyer, Klaus-Robert Müller, and myself which we would like you to re-consider for publication in Chemical Science.

Our work proposes a novel machine learning method for the automatic identification of chemical moieties in molecular data, which uses concepts from representation learning to find a set of common molecular building blocks (moieties) into which all molecules in a dataset can be decomposed. Our technique is an important step towards more efficient and interpretable machine learning procedures in quantum chemistry: The partitioning of molecules into chemical moieties allows chemists to gain a better understanding of the patterns discovered by machine learning in molecular data. This view is also shared by the reviewers, who consider our approach ”novel and of high interest to the computational chemistry/cheminformatics community”. Given the positive sentiment of the reviewers and the high relevance for the field, we consider Chemical Science an ideal platform for this manuscript and we hope that you share our enthusiasm to bring this work to the attention of a wider audience.

We thank both reviewers for their constructive feedback and thorough reading of our manuscript. In the following, we address all technical concerns and issues raised by the reviewers (point-to-point reply):

Reviewer 1

The authors introduce a framework for automated identification of chemical moieties in the context of MPNNs. The proposed method consists of an additional classification head that assigns types to atomic feature vectors coming out of a MPNN, an approach to assign atoms to specific instances of each type based on geometric constraints, and a loss function for learning these divisions in an unsupervised manner. The utility of the identified types is demonstrated in a number of tasks – regression, active learning, coarse graining atomistic simulations and identified reaction coordinates.

The manuscript is very well written and easy to follow, and the described approach is novel and of high interest to the computational chemistry/cheminformatics community.

My biggest concern is that it is challenging to demonstrate that the moieties identified offer an advantage compared to standard approaches, and so it is also difficult to assess how practically useful this method is. While the authors do provide a number of examples, I found a number of places where I wanted more information or comparisons to other methods to make a fair comparison. Perhaps the paper would have been better served by more detail across fewer applications areas. If the authors can address these concerns I would be happy to recommend it for publication.

I have two orthogonal concerns:

First, as the authors state, one of the most useful applications of moiety identification is to help humans make sense of chemical data and it is difficult to get a sense of how well MolINN-derived moieties work for this purpose. Of course being helpful to understanding is a vague notion, but I recommend the authors briefly describe some qualitative characteristics of the types and moieties learned from the QM9 dataset. Are there any 'unintuitive' combinations of moieties that share a type? How does the number of types/moieties grow with the number of training points? How does it treat ring systems? Does it differentially-type saturated and unsaturated rings? What is the largest moiety that is treated as a single unit? How does the pretraining (vs end-to-end) affect these answers? These questions would help the reader better understand how this approach behaves and what applications it would be suitable for.

Author reply: *Concern 1: "help humans make sense of chemical data"*

- *1.1 "Are there any 'unintuitive' combinations of moieties that share a type?": In Fig. 6 and Fig. S12 we now show at the example of deca-alanine that MoINN also finds unintuitive moieties. The NH_2 group is assigned to the same environment type as the CH_3 groups on the backbone of the molecule. The same can be observed for the CO group which belongs to the same environment type as NH. Due to the message passing architecture for representation learning, the atomic representations incorporate features beyond first order neighbors. Hence, the reason why those substructures are assigned to the same type can be explained by their similar local environment.*
- *1.2. "How does the number of types/moieties grow with the number of training points?": The number of moieties does not depend on the size of the training data (see the new Fig. S3).*
- *1.3. "How does it treat ring systems?": In section S5.1 we now show how MoINN treats ring systems. Besides some exemplary molecules we quantitatively describe the difference between aromatic rings and saturated rings in terms of environment type assignments. We observe that while saturated rings are predominantly divided into several small moieties, MoINN tends to identify aromatic rings as individual entities.*
- *1.4. "What is the largest moiety that is treated as a single unit?": The four largest structures identified by MoINN in the QM9 dataset are now shown in section S5.1.*
- *1.5 "How does the pretraining (vs end-to-end) affect these answers?": A comparison between pretrained and end-to-end MoINN is shown in Fig. S2. The end-to-end framework allows to adapt the atomic representations to the optimization, and thus, for more flexible environment types. Hence, the end-to-end MoINN finds more environment types then the pretrained version.*

Secondly, I have some questions about the various examples and in particular around how they compare to standard approaches.

1) In the regression example, a fingerprint is created based on counting the number of occurrences of each moiety. The linear model results seem striking – over a 100-fold improvement in prediction of internal energy over some other approaches. However, more information is needed to make this comparison meaningful:

    a) What is the origin of this stark difference in regression performance – are some of MolINN moieties essential for regression performance? I was not familiar with the Nannoolal et al. work, but it appears to be quite a comprehensive list of groups that look quite similar to exemplified moieties in Figure 2.

Author reply: *The reason why type-based fingerprints provided by MoINN perform better at this task of predicting extensive properties is now discussed in the manuscript on p.5.*

b) What is the actual number of types and moieties identified in the different MolINN methods? How do these compare to the comparison fingerprints? Are there differences in sparseness of the fingerprint representations?

Author reply: *The feature sizes of all fingerprints are now stated in the manuscript (p.5). In addition, we discuss the sparseness of type-based fingerprints (p.5).*

c) The experimental design of the linear regression task is not explained (data split,regularization etc.) and Table 1 does not explain if we are looking at training or held-out data. If it is training data, lower MAEs are not necessarily indicative of superior performance.

Author reply: *All the requested information can now be found in Section S5.1.*

d) I assume "RDkit fragment fingerprints" are generated by FragmentCatalog.FragFPGenerator and the basic catalogue of fragments defined by RDkit? This should be stated.

Author reply: *This is now stated in the manuscript (p.5).*

e) Both the Nannoolal et al. and fragment fingerprints are somewhat nonstandard choices. Can you add a comparison with a Morgan or ECFP fingerprints as well?

Author reply: *We have added the comparison to Morgan fingerprints, results shown in Table 1 and discussed in manuscript (p.5).*

f) What is the performance of the pretrained MolINN model in this regression task? Does it benefit from the preorganization of the input atomic fingerprints in the SchNet model?

Author reply: *We have added the results for the pretrained MoINN in Section S5.1.*

g) Minor point: in Section S4 – training of the end-to-end MoINN. Why did you us 11k (vs 110k) molecules for the unsupervised model? Why the smaller dataset?

Author reply: *The small training set was chosen to obtain a larger test set. In Section 4.4, we now show that a model trained on 110k samples barely deviates from the model trained on 10k data points.*

2) For the active learning example:

a) It is difficult to judge how well the selected fingerprints perform relative to any other diversity-seeking method because they are only compared to random sampling. The message would be strengthened by comparison to another approach – a simple baseline that I imagine would be a strong benchmark for internal energy could be stratified sampling based on atom number.

Author reply: *We included the benchmark stratified sampling w.r.t. atom number, see p.5. While better than random sampling, we find that MoiNN-based sampling is superior to stratified sampling.*

b) Additionally, the problem setup with the minimization seems like a complicated way of selecting a given a number of representative rows, which must be a deterministic problem for a given target number, for example column/row subset problems. I believe that the proposed row sampling problem in a Frobenius sense is solved by dropping rows with smallest norms (e.g. https://mathoverflow.net/questions/270743/sparse-matrix-approximation-using-only-a-few-dense-columns-or-rows) since you make zero error on those rows you retain. Is the limitation the size of the matrix? I assume you tune λ in order to converge to a target sparsity in W? Can you motivate why you choose this setup instead of, for example, K-means or medoids?

Author reply: *In Section S5.2, we now also show how sampling based on k-medoids impacts the prediction performance. As described in the manuscript, we expect better performance from a basis set of fingerprints that allows for reconstructing the remaining fingerprints using linear combinations, than a set of fingerprints that merely represent the variance in the dataset. For our proposed approach, we indeed tune the λ value to obtain a finite number of representative samples (λ is a hyperparameter that can be chosen to achieve the desired sparseness).*

3) The coarse-grained MD example looks impressive, and I suspect a large amount of work underlies this single figure. I am not very experienced with coarse-graining methods but I think the mapping proposed by MolINN seems like a reasonable one – folding hydrogens into their respectively heavy atoms, and different parameters for terminal CH3 and tertiary carbon atoms (it reminds me of OPLS UA).

   a) How does this compare to the mapping schemes used in the cited works (not necessarily the quality of the approximation, but rather how the beads are defined)?

   Author reply: *In Fig. S10, we now compare the CG representation provided by MoINN to other methods ranging from expertise-based manual assignments to automated CG representations. The CG representation found by MoINN indeed resembles the OPLS UA representation, with the advantage that MoINN can actually assign types to the beads. This way, beads of identical composition (e.g. CH_3 groups) can exhibit different types (if they appear in different chemical contexts) which may facilitate learning accurate force fields (compare Fig. S12).*

4) The reaction coordinate example makes intuitive sense, as the simple reaction is the most significant change during the trajectory. While I appreciate it is useful to have automated workflows, I wonder how necessary the unsupervised learning component is?

   a) In order to motivate that MolINN is actually adding something, could you demonstrate why this superior to say, the first PC of the soft adjacency matrix (eq 4)?

   Author reply: *We have added a comparison to PCA on the soft adjacency matrix in the supplement (Fig. S13). We find that the reaction coordinate identified by MoINN allows for a sharp distinction between reactant and product states. In comparison, the adjacency matrix-based reaction coordinate is much more noisy and the separation between reactant and product, while present, is less distinct.*

Reviewer 2

(1) A limitation of the current work is the methodology is only demonstrated in the scope of QM9. While it is not necessary to provide vast demonstrations for all types of standard datasets, certainly it is appropriate for the author's to show utility of their methodology to systems other than the small organic molecular systems of QM9.

Author reply: *In Fig. 6, we now show that MoINN also allows for coarse-graining of large molecules such as decaalanine.*

(2) In the bottom panel of Figure 1, the middle molecule is "cut off" by its neighboring molecules. This appears to be an image formatting issue.

Author reply: *This formatting issue has been fixed.*

(3) In the current form, the results presented in Figure 3 appear premature. A maximum training size of 500 molecules is small for this type of computational task. The authors should extend out the range of the x-axis with additional training runs. It appears the random sampling is decreasing more rapidly along the 200-500 interval; therefore, it should be established whether the two sampling methods are converging to the same value (the likely answer) or if random sampling is better for large training sets compared to MoINN sampling. Moreover, the value of figure 3 can be enhanced by including additional line(s) for other representative sampling methods. The claim of "data reduction" may need to be rethought depending on the results.

Author reply: *We now include an additional baseline, namely stratified sampling based on atom numbers. Further, we extend the experiment to larger training set sizes to show that all methods converge to similar results for increasing data size, as expected. MoiNN sampling is most useful when very little data can be afforded, e.g. when performing reference calculations at a very high level of theory. We agree with the reviewer that this is unnecessary when large training set sizes can be afforded, in which case random sampling is sufficient.*

(4) Significant details of all the case studies are missing and should be included into the SI for interpretability/reproducibility. As an example, for CG molecular dynamics simulations details should be added for the software used, duration of simulation, timestep, cutoff distances, etc. I understand the case studies are meant as a demonstration of MoINN, but I believe these details remain relevant.

Author reply: *All details of the case studies have been added to the supplement in Section S5.*

(5) Sticking with the CG-MD utility, I believe the authors should provide a demonstration of applying the MoINN workflow to a significantly larger molecular structure than anything resembling the QM9 dataset. The dipeptide system the authors present is good, and is correctly identified as a standard model system, however, the power of automating the selection of CG-beads based on functional groups is that it simplifies the modeling of larger more complex structures. I do not have a specific recommendation for a system(s), but they should exceed the number of heavy atoms represented in QM9 dataset by a minimum factor of 2-3. Perhaps longer polypeptide chains and contrast them against beads assigned by standard CG models, such as Martini.

Author reply: *In Fig. 6 and Fig S12, we now show that MoINN also provides meaningful environment types and beads for large molecules such as decaalanine.*

(6) Similar to point (3), numerous details regarding model training and construction are missing. Learning rate, batching, training method, etc.

Author reply: *All the necessary details have been added to the supplement (Section S5).*

(7) Figure S1B is not readable. With such a small size and lack of color scale bar, I am not able to

understand the composition of the 1,000 molecules chosen from QM9.

Author reply: *We increased the size of Figure S1b and added color bars and apologize for the poor quality in the previous version.*

(8) Results presented in Table 3 have no sense with errors of 300-400 eV, as both cited ref 59 and 60 are not valid baselines for these properties

Author reply: *We added Morgan Fingerprints as baseline fingerprints. The reason for the large difference in the performance on extensive properties is now also discussed in the manuscript (p.5).*

(9) This work is not reproducible as there is no code available.

Author reply: *Our code will be published upon acceptance of the manuscript.*

We are confident that the changes we made to the manuscript fully address the concerns raised by the reviewers and our manuscript now meets the standards required for publication in Chemical Science.

Sincerely yours,

O. Unke

Oliver T. Unke (on behalf of all authors)

Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Automatic Identification of Chemical Moieties

Jonas Lederer,*[ab] Michael Gastegger,[ab] Kristof T. Schütt,[ab] Michael Kampffmeyer,[c] Klaus-Robert Müller,[abdef] and Oliver T. Unke,*[abd]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

In recent years, the prediction of quantum mechanical observables with machine learning methods has become increasingly popular. Message-passing neural networks (MPNNs) solve this task by constructing atomic representations, from which the properties of interest are predicted. Here, we introduce a method to automatically identify chemical moieties (molecular building blocks) from such representations, enabling a variety of applications beyond property prediction, which otherwise rely on expert knowledge. The required representation can either be provided by a pretrained MPNN, or be learned from scratch using only structural information. Beyond the data-driven design of molecular fingerprints, the versatility of our approach is demonstrated by enabling the selection of representative entries in chemical databases, the automatic construction of coarse-grained force fields, as well as the identification of reaction coordinates.

1 Introduction

The computational study of structural and electronic properties of molecules is key to many discoveries in physics, chemistry, biology, and materials science. In this context, machine learning (ML) methods have become increasingly popular as a means to circumvent costly quantum mechanical calculations[1–37]. One class of such ML methods are message passing neural networks (MPNNs)[38], which provide molecular property predictions based on end-to-end learned representations of atomic environments.

In contrast to such fine-grained representations, chemists typically characterize molecules by larger substructures (e.g. functional groups) to reason about their properties [39–41]. This gives rise to the idea of using MPNNs for the automatic identification of “chemical moieties”, or characteristic parts of the molecule, to which its properties can be traced back. Since manually searching for moieties that explain (or are characteristic of) certain properties of molecules is a complex and tedious task, the capability of ML to find patterns and correlations in data could ease the identification of meaningful substructures drastically.

Previous work has introduced a variety of different approaches to identify substructures in molecules, with objectives ranging from substructure mining[42–47] over molecule generation[48–52] and interpretability of machine learning architectures[53–62] to coarse-graining[63–65]. However, to ensure the identification of meaningful moieties that can be utilized for a wide range of applications, a procedure is required (i) to be transferable w.r.t. molecule size, (ii) to provide a substructure decomposition of each molecule which preserves its respective global structure (required for, e.g., coarse-graining), and (iii) to allow for identifying several moieties of the same type in individual molecules (due to a common substructure often appearing multiple times). None of the methods mentioned above meets all of these criteria.

In this work, we propose MoINN (Moiety Identification Neural Network) – a method for the automatic identification of chemical moieties from the representations learned by MPNNs. This is achieved by constructing a soft assignment (or affinity) matrix from the atomic features, which maps individual atoms to different types of multi-atom substructures (Fig. 1, top). By employing representations from MPNNs pretrained on molecular properties, the identified moieties are automatically adapted to the chemical characteristics of interest. Alternatively, it is possible to find chemically meaningful substructures by training MPNNs coupled with MoINN in an end-to-end manner. Here, only structural information is required and *ab initio* calculations can be avoided. Crucially, MoINN is transferable between molecules of different sizes and automatically determines the appropriate number of moiety types. Multiple occurrences of the same structural motif within a molecule are recognized as the same type of moiety.

We demonstrate the versatility of MoINN by utilizing the identified chemical moieties to solve a range of tasks, which would otherwise require expert knowledge (Fig. 1, bottom). For exam-

[a] *Berlin Institute of Technology (TU Berlin), 10587 Berlin, Germany. E-mail: jonas.lederer@tu-berlin.de, oliver.unke@googlemail.com*
[b] *BIFOLD – Berlin Institute for the Foundations of Learning and Data, Germany.*
[c] *Department of Physics and Technology, UiT The Arctic University of Norway, 9019 Tromsø, Norway.*
[d] *Google Research, Brain team, Berlin.*
[e] *Department of Artificial Intelligence, Korea University, Seoul 136-713, Korea.*
[f] *Max Planck Institut für Informatik, 66123 Saarbrücken, Germany.*

† Electronic Supplementary Information (ESI) available: Additional details regarding the concept of MoiNN, training procedure, applications, and comparison to other graph-pooling approaches. See DOI: 00.0000/00000000.

ple, the learned moiety types can serve as molecular fingerprints, which allow to estimate the properties of compounds from their composition, or used to extract the most representative entries from quantum chemical databases. Beyond that, moieties can be employed as coarse-grained representations of chemical structures, allowing to automatically determine *beads* for the construction of coarse-grained force fields. Finally, we use MoINN to identify reaction coordinates in molecular trajectories based on the transformation of detected moieties.

2 Method

The automated identification of moieties with MoINN corresponds to a clustering of the molecule into different types of chemical environments. Hence, atoms in comparable environments, i.e. with similar feature representations (see Section 2.1), are likely to be assigned to the same cluster. In the following, the term "environment types" or short "types" will be used, since each cluster is associated with a specific substructure that exhibits particular chemical characteristics. Note that atoms belonging to the same environment type are not necessarily spatially close, because similar substructures may appear multiple times at distant locations in a molecule. This is why, after atoms have been assigned to environment types (see Section 2.2), individual (spatially disconnected) chemical moieties can be found by introducing an additional distance criterion (see Section 2.3). Both steps are combined to arrive at an unsupervised learning objective for decomposing molecules into chemical moieties (see Section 2.4).

2.1 Representation Learning in Message Passing Neural Networks

Message passing neural networks (MPNNs)[38] are able to learn atomic feature representations from data in an end-to-end manner (without relying on handcrafted features). They achieve state-of-the-art performance for molecular property prediction, solely taking atomic numbers and atom positions as inputs[7,11,12,14–17]. The representation learning scheme of an MPNN can be described as follows. First, atomic features are initialized to embeddings based on their respective atomic numbers (all atoms of the same element start with the same representation). Subsequently, the features of each atom are iteratively updated by exchanging "messages" with neighboring atoms, which depend on their current feature representations and distances. After several iterations, the features encode the relevant information about the chemical environment of each atom. In this work, we use SchNet[7,9,28] to construct atomic feature representations. In general, however, MoINN is applicable to any other representation learning scheme.

2.2 Assigning Atoms to Environment Types

Starting from F-dimensional atomic feature representations $\mathbf{x}_1, \ldots, \mathbf{x}_N$ of N atoms (e.g. obtained from an MPNN), a type assignment matrix $\mathbf{S}$, which maps individual atoms to different environment types, is constructed. Following a similar scheme as Bianchi et. at.[66], the type assignment matrix is given by

$$\mathbf{S} = \mathrm{softmax}\left(\mathrm{SiLU}\left(\mathbf{X}\mathbf{W}_1\right)\mathbf{W}_2\right) \ , \tag{1}$$

where $\mathbf{W}_1 \in \mathbb{R}^{F \times K}$ and $\mathbf{W}_2 \in \mathbb{R}^{K \times K}$ are trainable weight matrices, the n-th row of the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the representation $\mathbf{x}_n \in \mathbb{R}^F$ of atom n, and SiLU is the Sigmoid Linear Unit activation function[67]. Here, K is a hyperparameter that denotes the maximum number of possible types. As will be shown later, a meaningful number of types is automatically determined from data and largely independent of the choice of K (see Section 2.4). The softmax function ensures that entries S_{nk} of the $N \times K$ matrix $\mathbf{S}$ obey $\sum_k \mathrm{S}_{nk} = 1 \ \forall n$ with $\mathrm{S}_{nk} > 0$. Thus, each row of $\mathbf{S}$ represents a probability distribution over the K environment types, with each entry S_{nk} expressing how likely atom n should be assigned to type k. Even though assignments are "soft", i.e. every atom is partially assigned to multiple environment types, the softmax function makes it unlikely that more than one entry in each row is dominant (closest to 1). The advantage of a soft type assignment matrix is that its computation is well suited for gradient-based optimization. In other contexts, however, it might be more natural to assign atoms unambiguously to only one environment type. For this reason, we also define a "hard" type assignment matrix $\mathbf{S}_\mathrm{h} \in \mathbb{R}^{N \times K}$ with entries

$$\mathrm{S}_{\mathrm{h},nk} = \begin{cases} 1 & \mathrm{S}_{nk} > \mathrm{S}_{nj} \ \forall j \in [0, K) \backslash \{k\} \\ 0 & \text{otherwise} \end{cases} \ , \tag{2}$$

such that each row contains exactly one non-zero entry equal to 1.

The atomic feature representations making up the matrix $\mathbf{X}$ can either be provided by a pretrained model, or learned in an end-to-end fashion. Depending on the use case, both approaches offer their respective advantages: Since the type assignment matrix $\mathbf{S}$ is directly connected to $\mathbf{X}$ via Eq. 1, pretrained features allow to find types adapted to a specific property of interest. End-to-end learned representations have the advantage that they do not rely on any reference data obtained from computationally demanding quantum mechanical calculations. Instead, they are found from structural information by optimizing an unsupervised learning problem (see Section 2.4).

2.3 Assigning Atoms to Individual Moieties

Molecules may consist of multiple similar or even identical substructures. Consequently, distant atoms with comparable local environments can be assigned to the same type, even though they do not necessarily belong to the same moiety (see Fig. 1). To find the actual chemical moieties, i.e. groups of nearby atoms making up a structural motif, we introduce the $N \times N$ moiety similarity matrix given by

$$\mathbf{C} = \mathbf{S}\mathbf{S}^T \circ \mathbf{A} \ , \tag{3}$$

where "$\circ$" denotes the Hadamard (element-wise) product. Here, the $N \times N$ matrix $\mathbf{S}\mathbf{S}^T$ measures the similarity of the type assignments between atoms, i.e. its entries are close to 1 when a pair of atoms is assigned to the same environment type and close to 0 otherwise. The adjacency matrix $\mathbf{A} \in [0, 1]^{N \times N}$ on the other hand

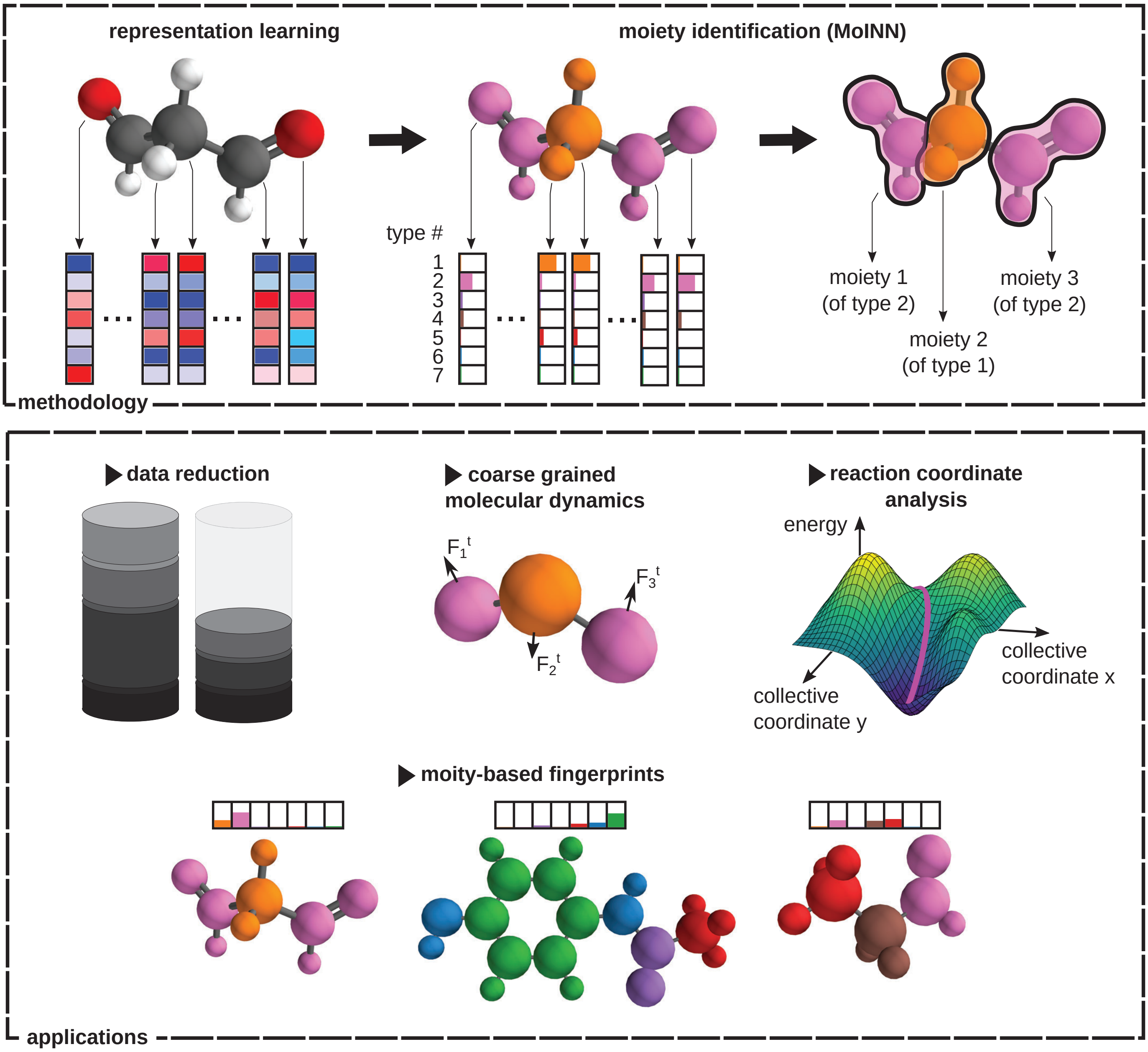

Fig. 1 MoINN methodology and applications. The top shows the moiety identification process for malondialdehyde. First, atomic feature representations (red and blue bars) are learned. Next, MoINN constructs type assignment vectors (pink and orange bars) based on these features. Each entry represents the probability of an atom to be assigned to a specific type of moiety (atoms are colored according to the highest atom-to-type affinity). Based on these assignments and the proximity of atoms, MoINN divides molecules into individual moieties. In this example, three chemical moieties of two distinct types associated with methylene (type 1, orange) and aldehyde (type 2, pink) groups are identified. The moiety representation allows for a variety of applications, which are shown on the bottom. They range from moiety-based fingerprint design, reaction coordinate analysis, and data reduction to coarse grained molecular dynamics.

captures the proximity of atoms. Its entries are defined as

$$\mathrm{A}_{ij}(r_{ij}) = \begin{cases} 0.5\left(1+\cos\left(\frac{\pi r_{ij}}{r_{\mathrm{cut}}}\right)\right) & r_{ij} < r_{\mathrm{cut}} \\ 0 & r_{ij} \geqslant r_{\mathrm{cut}} \end{cases}, \qquad (4)$$

where r_{ij} is the pairwise distance between atoms i and j and r_{cut} is a cutoff distance. For simplicity, we employ a cosine cutoff to assign proximity scores, but more sophisticated schemes are possible (e.g. based on the covalent radii of atoms). The combination of $\mathbf{S}\mathbf{S}^T$ and $\mathbf{A}$ ensures that the entries of the similarity matrix $\mathbf{C}$ are close to 1 only if two atoms are both assigned to the same type *and* spatially close, in which case they belong to the same chemical moiety.

Analogous to the hard assignment matrix $\mathbf{S}_{\mathrm{h}}$ (see Eq. 2), a hard moiety similarity matrix $\mathbf{C}_{\mathrm{h}}$, which unambiguously assigns atoms to a specific moiety, might be preferable over Eq. 3 in some contexts. To this end, we define the matrix

$$\mathbf{C}_\mathrm{h}^0 = \mathbf{S}_\mathrm{h}\mathbf{S}_\mathrm{h}^T \circ \mathbf{A}_\mathrm{cov}\ . \tag{5}$$

where $\mathbf{A}_\mathrm{cov}$ has entries of 1 for each atom-pair connected by a covalent bond (see Section S1†) and 0 otherwise. $\mathbf{C}_\mathrm{h}^0$ describes a graph on which breadth-first search[68] is performed to find its connected components (moieties). This yields a hard similarity matrix $\mathbf{C}_\mathrm{h}$, which maps atoms unambiguously to their individual moieties (for further details, please refer to Section S2†).

2.4 Optimization of Environment Type Assignments and Moiety Assignments

Chemical moieties are identified by minimizing the unsupervised loss function

$$\mathscr{L} = \mathscr{L}_\mathrm{cut} + \mathscr{L}_\mathrm{ortho} + \alpha\mathscr{L}_\mathrm{ent}, \tag{6}$$

where $\mathscr{L}_\mathrm{cut}$, $\mathscr{L}_\mathrm{ortho}$, and $\mathscr{L}_\mathrm{ent}$ are cut loss, orthogonality loss, and entropy loss, and α is a trade-off hyperparameter. The cut loss $\mathscr{L}_\mathrm{cut}$[66] penalizes "cutting" the molecule, i.e. assigning spatially close atoms to different moieties. It is defined as

$$\mathscr{L}_\mathrm{cut} = -\frac{Tr\left(\mathbf{C}^T\tilde{\mathbf{A}}\mathbf{C}\right)}{Tr\left(\mathbf{C}^T\tilde{\mathbf{D}}\mathbf{C}\right)},$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \in \mathbb{R}^{N\times N}$ is a symmetrically normalized adjacency matrix (see Eq. 4). The degree matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ is diagonal with elements $\mathrm{D}_{ii} = \sum_j^N \mathrm{A}_{ij}$, where A_{ij} are the entries of $\mathbf{A}$. Consequently, $\tilde{\mathbf{D}}$ is the degree matrix obtained from the entries of $\tilde{\mathbf{A}}$.

To avoid converging to the trivial minimum of $\mathscr{L}_\mathrm{cut}$ where all atoms are assigned to the same moiety and type, the orthogonality loss[66]

$$\mathscr{L}_\mathrm{ortho} = \left\| \frac{\mathbf{S}\mathbf{S}^T}{\|\mathbf{S}\mathbf{S}^T\|_F} - \frac{\mathbf{I}_N}{\sqrt{N}} \right\|_F$$

drives the type assignment vectors of different atoms (i.e., the rows of $\mathbf{S}$) to be (close to) orthogonal. Here, $\mathbf{I}_N$ is the $N \times N$ identity matrix and $\|\cdot\|_F$ is the Frobenius norm.

Finally, the entropy term[69]

$$\mathscr{L}_\mathrm{ent} = -\frac{1}{N}\sum_{nk} \mathrm{S}_{nk} \ln\left(\mathrm{S}_{nk}\right)$$

favors "hard" assignments and indirectly limits the number of used types (here, S_{nk} are the entries of $\mathbf{S}$, see Eq. 1). Without this term, there is no incentive to use fewer than K types, i.e., the model would eventually converge to use as many different types as possible. Hence, by introducing the entropy term, we avoid relying on expert knowledge for choosing K and instead facilitate learning a meaningful number of types from data.

In principle, the number of used types still depends on K and the entropy trade-off factor α. However, as is shown in Section S3†, there is a regime of α where the number of used types is largely independent of K (as long as K is sufficiently large). Hence, we arbitrarily choose $K = 100$ in our experiments if not specified otherwise.

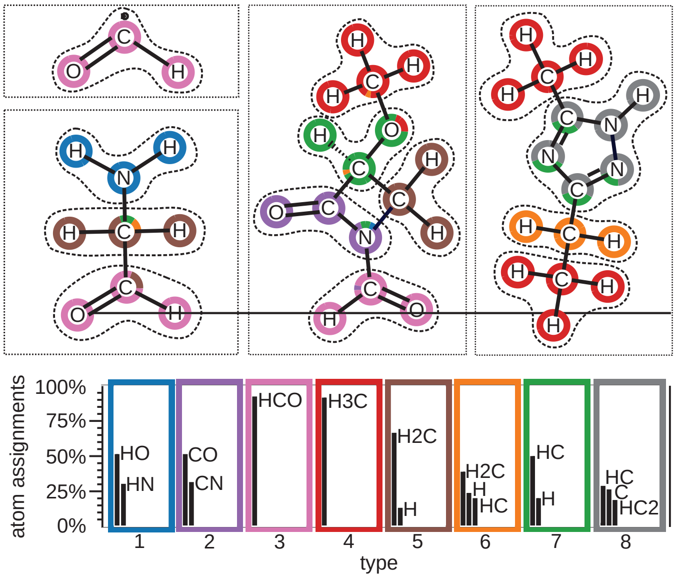

Fig. 2 Common moieties of the QM9 dataset. The top shows four exemplary molecules along with type assignments (colored circles) and moieties (enclosed by dashed lines). The bottom shows the distribution of environment types and corresponding most common moieties for the test set (1000 molecules), black 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.

3 Applications

This section describes several applications of MoINN. First, we use MoINN to identify common moieties in molecular data (Section 3.1). Leveraging these insights, we select representative examples from a database of structures to efficiently reduce the number of reference calculations required for property prediction tasks (Section 3.2). Next, an automated pipeline for coarse-grained molecular dynamics simulations built on top of MoINN is described (Section 3.3). Finally, we demonstrate how to utilize MoINN for automatically detecting reaction coordinates in molecular dynamics trajectories (Section 3.4).

3.1 Identification of Chemical Moieties

To demonstrate the automatic identification of chemical moieties, we apply MoINN to the QM9 dataset[70]. Here, two different models are considered: One utilizes fixed feature representations provided by a SchNet model pretrained to predict energies, while the other model is trained in an end-to-end fashion on purely structural information. In the following, these will be referred to as the pretrained model and the end-to-end model, respectively. Details on the training of SchNet and MoINN, as well as, a comparison between pretrained model and end-to-end model can be found in Section S4†. Also the impact of varying training data size on MoINN outputs is shown there.

Figure 2 depicts the results for the pretrained MoINN model evaluated on a test set of 1,000 molecules that were excluded from the training procedure. The top shows four exemplary molecules with corresponding type assignments and moieties. As expected, we observe that moieties of the same type may occur across different molecules, as well as multiple times in a single molecule. The evaluation of environment types and corresponding moieties for all 1,000 molecules (see bottom of Fig. 2) shows that each type is associated with a small set of similar moieties, i.e., the environment types form a “basis” of common substructures that can be combined to form all molecules contained in the dataset. In Section S5.1†, we evaluate MoINN w. r. t. various ring systems and the largest identified moieties. We observe that while saturated rings are predominantly divided into several small moieties, MoINN tends to identify aromatic rings as individual entities.

To verify that the type assignments are chemically meaningful, we use them to construct molecular fingerprints, from which different chemical properties are predicted via a linear regression model (for more details refer to Section S5.1†). Although it is unrealistic to expect state-of-the-art performance with such a simple model, meaningful molecular fingerprints should at least perform on-par with handcrafted variants. The type-based fingerprints are constructed from the assignments learned by the end-to-end MoINN model as

$$\mathbf{h}_{\text{MoINN}} = \sum_{n} \mathrm{S}_{nk}(\mathbf{X}) \; , \qquad (7)$$

where $\mathbf{X}$ denotes the feature matrix and S_{nk} is the assignment matrix entry for the n-th atom and the k-th type. The feature size of the type-based fingerprints is given by the number of environment types $K = 100$. However, due to the sparsity of the environment types, the effective number of features is 17 (see also Section S4†). For comparison, we use handcrafted fragmentation of molecules stated by Nannolal et. al.[71], fingerprints provided by the fragment catalogue and fragment generator of RDKit[72], and Morgan fingerprints[73] with feature sizes 86, 15387 and 100, respectively. Table 1 compares the test errors of the corresponding linear models.

The type-based fingerprints significantly outperform the other considered fingerprints, which suggests that the environment types provided by MoINN are a chemically meaningful representation of the molecules in the dataset. The performance is particularly good for the considered extensive properties U_0, H, and F. The reason for this is that, in contrast to the other fingerprints, information about the molecule size is implicitly contained in the fingerprints provided by MoINN.

Table 1 Mean absolute error of the predicted dipole moment μ, internal energy U_0, the enthalpy H, and the free energy F based on linear regression on different molecular fingerprints.

property	MoINN	Nannoolal et. al.[71]	RDKit[72]	Morgan[73]
μ (Debye)	**0.07**	0.89	0.62	0.19
U_0 (eV)	**1.46**	454.66	303.44	513.78
H (eV)	**1.64**	455.04	318.25	512.65
F (eV)	**0.65**	464.82	320.85	518.61

3.2 Sampling of Representative Molecules

The quality of the reference dataset used to train ML models greatly impacts their generalization performance[74]. Since the calculation of molecular properties at high levels of theory is computationally demanding, it is desirable to find ways to reduce the amount of reference data needed for training accurate machine learning models. One way to achieve this is by sampling a representative subset of datapoints from chemical space (instead of choosing points randomly). Here, we employ the type-based fingerprints (Eq. 7) described in the previous section to find a subset of molecules as small as possible, which still represents the QM9 dataset sufficiently well. To this end, we minimize the loss function

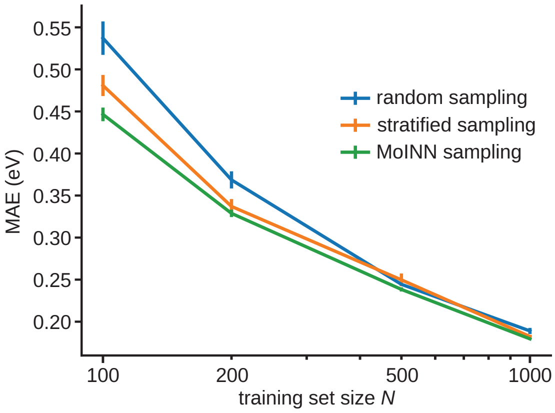

Fig. 3 Mean absolute error (MAE) of energy predictions for SchNet models trained on randomly sampled training sets (blue), training sets obtained by stratified sampling (orange) and training sets selected with MoINN (green). Each data point is averaged over five independent training runs and standard errors are indicated by error bars.

$$\mathscr{L}_{\text{data}} = \|\mathbf{W}\mathbf{H}_{\text{MoINN}} - \mathbf{H}_{\text{MoINN}}\|_F + \lambda \sum_j \sqrt{\sum_i w_{ij}^2} \; . \qquad (8)$$

$\mathbf{H}_{\text{MoINN}} \in \mathbb{R}^{D\times K}$ denotes the fingerprint matrix of D molecules, where each row is given by the fingerprint vector $\mathbf{h}_{\text{MoINN}}$ of a specific molecule. $\mathbf{W} \in \mathbb{R}^{D\times D}$ is a trainable weight matrix with entries $\{w_{ij}\}$. The first term in Eq. 8 describes the reconstruction error. To avoid converging to the trivial solution, where the trainable matrix $\mathbf{W}$ is simply the identity matrix, we introduce a regularization term that enforces sparse rows in the weight matrix $\mathbf{W}$. The trade-off between both terms can be tuned by the factor λ, i.e. larger values of λ will select a smaller subset of representative molecules. Intuitively, minimizing Eq. 8 corresponds to selecting a small number of molecules as “basis vectors”, from which all other molecules can be (approximately) reconstructed by linear combination.

Based on this procedure, we select several QM9 subsets of different size as training sets and compare them to randomly sampled subsets, and subsets obtained by stratified sampling w. r. t. the number of atoms in each molecule. For each of these subsets, we train five SchNet models and evaluate their average performance (Fig. 3). Models trained on subsets chosen by MoINN perform significantly better than those trained on randomly sampled subsets and stratified sampled subsets. This effect is most pronounced for small training set sizes. Thus, selecting data with MoINN is most useful in a setting where only few data points can be afforded, e.g. when using a high level of theory to per-

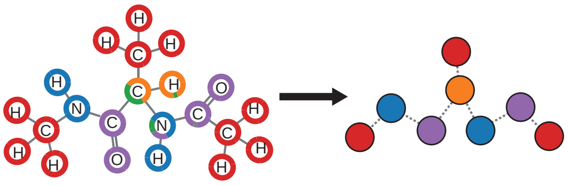

Fig. 4 Automated coarse-graining with MoINN. On the left, the alanine-dipeptide molecule is depicted at atomic resolution, assigned environment types are indicated by colored circles. On the right, the corresponding coarse-grained representation, derived from environment types and moiety assignments, is shown.

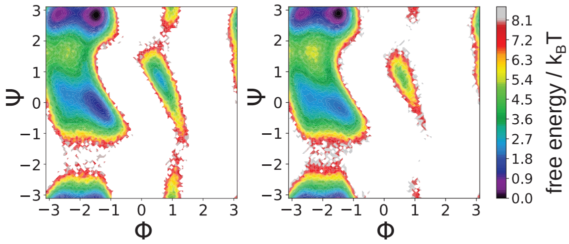

Fig. 5 Ramachandran plots of the free energy surface of alanine-dipeptide with respect to the torsion angles ϕ and ψ for the atomistic MD (left) and the coarse-grained MD (right) (ϕ and ψ are computed with respect to the coarse-grained geometry).

form reference calculations. For more details on the experiment and a comprehensive discussion of the results please refer to Section S5.2†.

3.3 Coarse-Grained Molecular Dynamics

While ML force fields accelerate *ab initio* MD simulations by multiple orders of magnitude[13], the study of very large molecular structures is still computationally demanding. Coarse-graining (CG) reduces the dimensionality of the problem by representing groups of atoms as single interaction sites. Most approaches rely on systematically parametrized CG force fields[75,76], but also data driven approaches have been proposed[77–81]. In both cases, however, the coarse-grained "beads" are usually determined manually by human experts[82].

Here, we propose an automated pipeline for coarse-grained molecular dynamics simulations (CG-MD), which comprises atomistic SchNet models for noise reduction, MoINN for reducing the molecule's degrees of freedom, and a SchNet model trained on the CG representation for simulating the CG dynamics. We apply this approach to the trajectory of alanine-dipeptide in water[83,84], which is a commonly used model system for comparing different CG methods[77–79].

The CG representation, shown in Fig. 4, is inferred from the environment types and moieties provided by the pretrained MoINN model described in Section 3.1, which has been trained on the QM9 dataset. For a comparison to conventional CG representations such as, e.g., OPLS-UA[85,86], or an automated CG approach[87] for the Martini force field[75], please refer to section S5.3†. The original atomistic trajectory of alanine-dipeptide does not include reference energies. This is because the dynamics have been simulated in solvent, which introduces noise to the energy of the system if the solvent is not modeled explicitly. The data contains forces for all atoms in the alanine-dipeptide molecule, which implicitly include interactions with solvent molecules. However, sparsely sampled transition regions between conformers are challenging to learn with force targets only. Coarse-graining introduces additional noise on the energies and forces[79] since some information about the atom positions is discarded.

To reduce the noise, we train an ensemble of five SchNet models to provide a force field for alanine-dipeptide at atomic resolution. Subsequently, we use the corresponding forces $\hat{\mathbf{F}}$ and energies $\hat{U}$ as targets for training the CG SchNet model in a force-matching scheme adapted from Refs. 77–79,88,89 (see Section S5.3† for details). Based on the CG SchNet model, we run MD simulations in the NVT ensemble at room temperature (300 K). The trajectories are initialized according to the Boltzmann distribution at the six minima of the potential energy surface. For keeping the temperature constant, we use a Langevin thermostat. Fig. 5 shows that the free energy surfaces derived from the all-atom and CG trajectories are in good agreement.

MoINN also allows for coarse-graining structures outside the scope of QM9. This is shown in Fig. 6 for decaalanine. The provided CG representation resembles the commonly used OPLS-UA representation[85,86]. However it is striking that the type of terminal methyl groups differs from that of the backbone methyl groups, while in the OPLS-UA representation, by construction, those groups are considered to be interchangeable. For more details how the CG representation is derived from the provided environment types, see Section S5.3†.

3.4 Dynamic Clustering and Reaction Coordinate Analysis

MoINN is also capable of learning environment types for molecular trajectories. In this case, the types describe "dynamic clusters", which can be useful, e.g., for determining collective variables that describe chemical reactions. As a demonstration of this concept, we consider two chemical reactions, namely the proton transfer reaction in malondialdehyde and the Claisen rearrangement of allyl *p*-tolyl ether (see Refs. 90 and 91 for details on how the trajectories were obtained). We train individual end-to-end MoINN models on each reaction trajectory.

For each time step in the trajectory, we construct a high-dimensional coordinate vector

$$\mathbf{h}_{\text{dyn}} = (S_{11}, S_{12}, \ldots, S_{1K}, S_{21}, \ldots, S_{2K}, \ldots, S_{NK}) ,$$

which consists of the type assignment matrix entries $\{S_{nk}\}$. By applying standard dimensionality reduction methods like principal component analysis (PCA)[92,93], it is possible to extract a low-dimensional representation of the largest structural changes in the trajectory. For the two considered cases, we find that a one-dimensional reaction coordinate given by the first principal component provides a good description of the dynamic process and shows a prominent "jump" when the reaction happens (see Fig. 7). In Section S5.4†, we show that the reaction coordinate derived from MoINN allows for a more clear distinction between

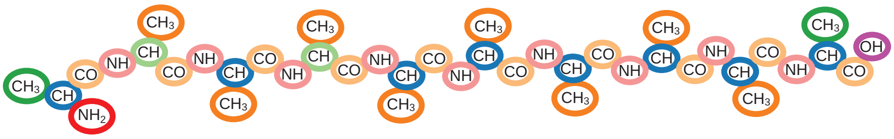

Fig. 6 CG-representation of deca-alanine inferred from environment types provided by MoINN.

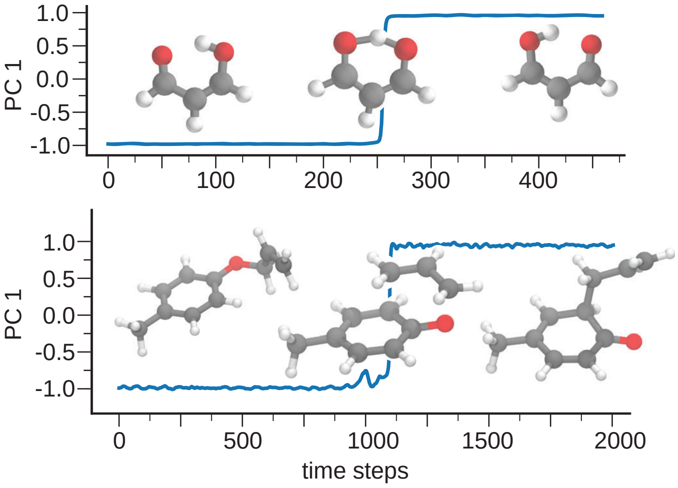

Fig. 7 Automatic detection of reaction coordiantes for the proton transfer in malondialdehyde (top) an the Claisen rearrangement of allyl *p*-tolyl ether (bottom). The identified reaction coordinate is plotted w.r.t the time step of the respective trajectory.

reactants and product than simply using the adjacency matrix based on the pairwise distances of atoms.

4 Discussion

Owing to its computational efficiency, interpretability, and transferability, MoINN is applicable to a wide range of different tasks which otherwise rely on expert knowledge. MoINN involves a representation-based pooling approach which shares common ideas with the graph-pooling approaches DiffPool[69] and MinCut[66]. Latter describe the acquisition of a soft assignment matrix, which allows to pool graph nodes (atoms) to representative super-nodes (atom groups). In DiffPool the assignment matrix is learned utilizing GNNs, while MinCut employs a multilayer perceptron (MLP) architecture. Both methods introduce unsupervised loss functions to ensure a reasonable number of super-nodes while preferably grouping nearby nodes. Similar to MinCut, MoINN learns a mapping from atomic feature representations to type assignments by two dense layers (an MLP). The shallow network architecture results in computationally cheap training and inference. As the most prominent difference, MoINN distinguishes between individual moieties and environment types, while DiffPool and MinCut handle those entities interchangeably. As a result, MoINN stands out with regards to interpretability and transferability.

The distinction between moieties and environment types allows for a more detailed analysis of the identified substructures. While the environment types explain the composition and conformation of the molecular substructures, the moieties represent individual molecular building blocks. Besides the benefits with regards to interpretability, the distinction between moieties and environment types allows to identify multiple identical moieties in the same molecule. This feature is particularly useful for molecular systems since those often possess atom groups (moieties) of the same type multiple times. In contrast, pooling distant nodes is penalized when utilizing MinCut or DiffPool. Hence, even though some distant nodes might exhibit similar feature representations, they are unlikely to be grouped together. This makes it difficult to find common moieties and might hamper transferability w.r.t. molecules of different size, since the mapping between atoms and atom groups becomes more sensitive to small changes in the feature representations. For more details on this problem, please refer to Section S6†.

5 Conclusion and Outlook

We have introduced MoINN, a versatile approach capable of automatically identifying chemical moieties in molecular data from the representations learned by MPNNs. Depending on the task at hand, MoINN can be trained based on pretrained representations or in an end-to-end fashion. While pretrained representations may lead to moieties that are associated with a certain molecular property, training MoINN in an end-to-end fashion circumvents costly first principle calculations. By construction, MoINN allows to identify multiple moieties of the same type (e.g. corresponding to the same functional group) in individual molecules. This design choice also makes MoINN transferable w.r.t. molecule size and allows to automatically determine a reasonable number of moieties and environment types without relying on expert knowledge.

Representing molecules as a composition of chemical moieties paves the way for various applications, some of which have been demonstrated in this work: Human-readable and interpretable fingerprints can be directly extracted from the environment types identified by MoINN and used to estimate molecular properties. In addition, they can be employed for selecting representative molecules from quantum mechanical databases to reduce the number of *ab initio* reference data necessary for training accurate models. Further, we have proposed a CG-MD simulation pipeline based on MoINN, which includes all necessary steps from the identification of CG representations, the machine learning of a CG force field, up to the MD simulation of the CG molecule. The pipeline is fully automatic and does not require expert knowledge. As another example, we have presented the dynamic clustering of chemical reactions, demonstrating that the environment types identified by MoINN capture conformational information that can be used to define reaction coordinates.

A promising avenue for future work is the application of MoINN in the domain of generative models[25,26]. Jin et. al. have shown that generating molecules in a hierarchical fashion can be advantageous[48,50]. MoINN could help to identify promising motifs for molecule generation and hence facilitate the discovery of large molecules. Furthermore, MoINN could be utilized to analyze other interesting reactions. In summary, MoINN extends the applicability of MPNNs to a wide range of chemical problems otherwise relying on expert knowledge. In addition, we expect applications of MoINN to allow new insights into large-scale chemical phenomena, where MPNNs acting on individual atoms are prohibitively computationally expensive to evaluate.

Author Contributions

Jonas Lederer, Oliver T. Unke, Michael Gastegger, and Kristof T. Schütt conceived the work. Together with Michael Kampffmeyer they developed the method. Jonas Lederer, Oliver T. Unke, Michael Gastegger, Kristof T. Schütt and Klaus-Robert Müller collected promising application ideas. Jonas Lederer implemented the method and performed the experiments. All the authors contributed to the discussion, writing, editing, and revision of the manuscript.

Conflicts of interest

The authors have no conflicts of interest to declare.

Acknowledgments

MG has been at BASLEARN – TU Berlin/BASF Joint Lab for Machine Learning, co-financed by TU Berlin and BASF SE. KTS, KRM and OTU acknowledge support by the Federal Ministry of Education and Research (BMBF) for the Berlin Institute for the Foundations of Learning and Data (BIFOLD) (01IS18037A,01IS14013A-E, 01GQ1115 and 01GQ0850). KRM acknowledges partial support by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program, Korea University and No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation). OTU acknowledges funding from the Swiss National Science Foundation (Grant No. P2BSP2_188147).

Notes and references

1 J. Behler and M. Parrinello, *Physical Review Letters*, 2007, **98**, 146401.

2 A. P. Bartók, M. C. Payne, R. Kondor and G. Csányi, *Physical Review Letters*, 2010, **104**, 136403.

3 M. Rupp, A. Tkatchenko, K.-R. Müller and O. A. von Lilienfeld, *Physical Review Letters*, 2012, **108**, 058301.

4 K. Schütt, P.-J. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko and K.-R. Müller, Neural Information Processing Systems, 2017, pp. 991–1001.

5 S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt and K.-R. Müller, *Science Advances*, 2017, **3**, e1603015.

6 L. Zhang, J. Han, H. Wang, R. Car and W. E, *Phys. Rev. Lett.*, 2018, **120**, 143001.

7 K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko and K.-R. Müller, *The Journal of Chemical Physics*, 2018, **148**, 241722.

8 L. Zhang, J. Han, H. Wang, R. Car and W. E, *Physical Review Letters*, 2018, **120**, 143001.

9 K. T. Schütt, P. Kessel, M. Gastegger, K. Nicoli, A. Tkatchenko and K. R. Müller, *Journal of Chemical Theory and Computation*, 2018, **15**, 448–455.

10 H. E. Sauceda, S. Chmiela, I. Poltavsky, K.-R. Müller and A. Tkatchenko, *Machine Learning Meets Quantum Physics*, Springer, 2020, pp. 277–307.

11 O. T. Unke and M. Meuwly, *Journal of chemical theory and computation*, 2019, **15**, 3678–3693.

12 K. Schütt, O. Unke and M. Gastegger, Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 9377–9388.

13 O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko and K.-R. Müller, *Chemical Reviews*, 2021, **121**, 10142–10186.

14 O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda and K.-R. Müller, *Nature Communications*, 2021, **12**, 7273.

15 J. Klicpera, J. Groß and S. Günnemann, International Conference on Learning Representations (ICLR), 2020.

16 S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt and B. Kozinsky, *Nature communications*, 2022, **13**, 2453.

17 K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller and A. Tkatchenko, *Nature Communications*, 2017, **8**, 13890.

18 F. Noé, A. Tkatchenko, K.-R. Müller and C. Clementi, *Annual review of physical chemistry*, 2020, **71**, 361–390.

19 O. A. von Lilienfeld, K.-R. Müller and A. Tkatchenko, *Nat. Rev. Chem.*, 2020, **4**, 347–358.

20 J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller and A. Tkatchenko, *Chem. Rev.*, 2021, **121**, 9816–9872.

21 K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev and A. Walsh, *Nature*, 2018, **559**, 547–555.

22 S. Chmiela, H. E. Sauceda, K.-R. Müller and A. Tkatchenko, *Nature communications*, 2018, **9**, 3887.

23 J. S. Smith, O. Isayev and A. E. Roitberg, *Chemical science*, 2017, **8**, 3192–3203.

24 M. Popova, O. Isayev and A. Tropsha, *Science advances*, 2018, **4**, eaap7885.

25 N. W. Gebauer, M. Gastegger and K. T. Schütt, Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 7566–7578.

26 N. W. Gebauer, M. Gastegger, S. S. Hessmann, K.-R. Müller and K. T. Schütt, *Nature Communications*, 2022, **13**, 973.

27 S. Chmiela, H. E. Sauceda, I. Poltavsky, K.-R. Müller and A. Tkatchenko, *Computer Physics Communications*, 2019, **240**, 38–45.

28 K. T. Schütt, S. S. P. Hessmann, N. W. A. Gebauer, J. Lederer and M. Gastegger, *The Journal of Chemical Physics*, 2023, **158**, 144801.

29 S. Chmiela, V. Vassilev-Galindo, O. T. Unke, A. Kabylda, H. E. Sauceda, A. Tkatchenko and K.-R. Müller, *Science Advances*, 2023, **9**, eadf0873.

30 J. Lederer, W. Kaiser, A. Mattoni and A. Gagliardi, *Advanced Theory and Simulations*, 2019, **2**, 1800136.

31 A. Musaelian, S. Batzner, A. Johansson, L. Sun, C. J. Owen, M. Kornbluth and B. Kozinsky, *Nature Communications*, 2023, **14**, 579.

32 J. Gasteiger, J. Groß and S. Günnemann, International Conference on Learning Representations (ICLR), 2020.

33 J. Gasteiger, S. Giri, J. T. Margraf and S. Günnemann, Machine Learning for Molecules Workshop, 2020.

34 S. Doerr, M. Majewski, A. Pérez, A. Kramer, C. Clementi, F. Noe, T. Giorgino and G. De Fabritiis, *Journal of chemical theory and computation*, 2021, **17**, 2355–2363.

35 B. Huang and O. A. von Lilienfeld, *Nature Chemistry*, 2020, **12**, 945–951.

36 B. Huang and O. A. Von Lilienfeld, *Chemical reviews*, 2021, **121**, 10001–10036.

37 K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. Von Lilienfeld, K.-R. Muller and A. Tkatchenko, *The journal of physical chemistry letters*, 2015, **6**, 2326–2331.

38 J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals and G. E. Dahl, Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 1263–1272.

39 B. E. Evans, K. E. Rittle, M. G. Bock, R. M. DiPardo, R. M. Freidinger, W. L. Whitter, G. F. Lundell, D. F. Veber, P. S. Anderson, R. S. L. Chang, V. J. Lotti, D. J. Cerino, T. B. Chen, P. J. Kling, K. A. Kunkel, J. P. Springer and J. Hirshfield, *Journal of Medicinal Chemistry*, 1988, **31**, 2235–2246.

40 C. D. Duarte, E. J. Barreiro and C. A. Fraga, *Mini reviews in medicinal chemistry*, 2007, **7**, 1108–1119.

41 T. L. Lemke, *Review of organic functional groups: introduction to medicinal organic chemistry*, Lippincott Williams & Wilkins, 2003.

42 P. Ertl, *Journal of cheminformatics*, 2017, **9**, 1–7.

43 J. Klekota and F. P. Roth, *Bioinformatics*, 2008, **24**, 2518–2525.

44 Y. Yamanishi, E. Pauwels, H. Saigo and V. Stoven, *Journal of chemical information and modeling*, 2011, **51**, 1183–1194.

45 C. Borgelt and M. R. Berthold, 2002 IEEE International Conference on Data Mining, 2002. Proceedings., 2002, pp. 51–58.

46 M. Coatney and S. Parthasarathy, Third IEEE Symposium on Bioinformatics and Bioengineering, 2003. Proceedings., 2003, pp. 336–340.

47 A. T. Brint and P. Willett, *Journal of Chemical Information and Computer Sciences*, 1987, **27**, 152–158.

48 W. Jin, R. Barzilay and T. Jaakkola, International Conference on Machine Learning, 2020, pp. 4839–4848.

49 T. S. Hy and R. Kondor, *Multiresolution Graph Variational Autoencoder*, 2021.

50 W. Jin, R. Barzilay and T. Jaakkola, ICML, 2018.

51 W. Jin, R. Barzilay and T. Jaakkola, International Conference on Machine Learning, 2020, pp. 4849–4859.

52 M. Guarino, A. Shah and P. Rivas, 2017.

53 G. Montavon, W. Samek and K.-R. Müller, *Digital Signal Processing*, 2018, **73**, 1–15.

54 W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders and K.-R. Müller, *Proceedings of the IEEE*, 2021, **109**, 247–278.

55 T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schütt, K.-R. Müller and G. Montavon, *IEEE transactions on pattern analysis and machine intelligence*, 2022, **44**, 7581–7596.

56 E. Noutahi, D. Beani, J. Horwood and P. Tossou, *arXiv:1905.11577 [cs, q-bio, stat]*, 2020.

57 K. McCloskey, A. Taly, F. Monti, M. P. Brenner and L. J. Colwell, *Proceedings of the National Academy of Sciences*, 2019, **116**, 11624–11629.

58 B. Chen, T. Wang, C. Li, H. Dai and L. Song, International Conference on Learning Representations, 2020.

59 A. Mukherjee, A. Su and K. Rajan, *Journal of Chemical Information and Modeling*, 2021, **61**, 2187–2197.

60 H. E. Webel, T. B. Kimber, S. Radetzki, M. Neuenschwander, M. Nazaré and A. Volkamer, *Journal of computer-aided molecular design*, 2020, **34**, 731–746.

61 A. H. Khasahmadi, K. Hassani, P. Moradi, L. Lee and Q. Morris, International Conference on Learning Representations, 2019.

62 S. Letzgus, P. Wagner, J. Lederer, W. Samek, K.-R. Müller and G. Montavon, *IEEE Signal Processing Magazine*, 2022, **39**, 40–58.

63 W. Wang and R. Gómez-Bombarelli, *npj Computational Materials*, 2019, **5**, 1–9.

64 M. A. Webb, J.-Y. Delannoy and J. J. Pablo, *Journal of Chemical Theory and Computation*, 2019, **15**, 1199–1208.

65 M. Chakraborty, C. Xu and A. D. White, *The Journal of Chemical Physics*, 2018, **149**, 134106.

66 F. M. Bianchi, D. Grattarola and C. Alippi, International conference on machine learning, 2020, pp. 874–883.

67 D. Hendrycks and K. Gimpel, *arXiv preprint arXiv:1606.08415*, 2016.

68 S. S. Skiena, in *The Algorithm Design Manual*, Springer Publishing Company, Incorporated, 2nd edn, 2008, pp. 162–166.

69 Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton and J. Leskovec, Neural Information Processing Systems, 2018, pp. 4800–4810.

70 R. Ramakrishnan, P. O. Dral, M. Rupp and O. A. Von Lilienfeld, *Scientific data*, 2014, **1**, 1–7.

71 Y. Nannoolal, J. Rarey and D. Ramjugernath, *Fluid Phase Equilibria*, 2008, **269**, 117–133.

72 *RDKit: Open-source cheminformatics*, `http://www.rdkit.org`.

73 D. Rogers and M. Hahn, *Journal of chemical information and modeling*, 2010, **50**, 742–754.

74 L. I. Vazquez-Salazar, E. Boittier, O. T. Unke and M. Meuwly, *Journal of Chemical Theory and Computation*, 2021, **17**, 4769–4785.

75 S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman and A. H. De Vries, *The journal of physical chemistry B*, 2007, **111**,

7812–7824.

76 E. Brini, E. A. Algaer, P. Ganguly, C. Li, F. Rodríguez-Ropero and N. F. A. van der Vegt, *Soft Matter*, 2013, **9**, 2108–2119.

77 B. E. Husic, N. E. Charron, D. Lemm, J. Wang, A. Pérez, M. Majewski, A. Krämer, Y. Chen, S. Olsson, G. de Fabritiis *et al.*, *The Journal of Chemical Physics*, 2020, **153**, 194101.

78 J. Wang, S. Chmiela, K.-R. Müller, F. Noé and C. Clementi, *The Journal of Chemical Physics*, 2020, **152**, 194106.

79 J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. De Fabritiis, F. Noé and C. Clementi, *ACS central science*, 2019, **5**, 755–767.

80 L. Zhang, J. Han, H. Wang, R. Car and W. E, *The Journal of Chemical Physics*, 2018, **149**, 034101.

81 Y. Chen, A. Krämer, N. E. Charron, B. E. Husic, C. Clementi and F. Noé, *The Journal of Chemical Physics*, 2021, **155**, 084101.

82 S. Riniker, J. R. Allison and W. F. van Gunsteren, *Physical Chemistry Chemical Physics*, 2012, **14**, 12423–12430.

83 F. Nüske, H. Wu, J.-H. Prinz, C. Wehmeyer, C. Clementi and F. Noé, *The Journal of Chemical Physics*, 2017, **146**, 094104.

84 C. Wehmeyer and F. Noé, *The Journal of chemical physics*, 2018, **148**, 241703.

85 W. L. Jorgensen and J. Tirado-Rives, *Journal of the American Chemical Society*, 1988, **110**, 1657–1666.

86 W. L. Jorgensen, D. S. Maxwell and J. Tirado-Rives, *Journal of the American Chemical Society*, 1996, **118**, 11225–11236.

87 T. D. Potter, E. L. Barrett and M. A. Miller, *Journal of Chemical Theory and Computation*, 2021, **17**, 5777–5791.

88 W. G. Noid, P. Liu, Y. Wang, J.-W. Chu, G. S. Ayton, S. Izvekov, H. C. Andersen and G. A. Voth, *The Journal of Chemical Physics*, 2008, **128**, 244115.

89 W. G. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das and H. C. Andersen, *The Journal of Chemical Physics*, 2008, **128**, 244114.

90 K. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller and R. J. Maurer, *Nature communications*, 2019, **10**, 5024.

91 M. Gastegger, K. T. Schütt and K.-R. Müller, *Chemical science*, 2021, **12**, 11473–11483.

92 H. Hotelling, *J. Educ. Psy.*, 1933, **24**, 498–520.

93 K. Pearson, *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 1901, **2**, 559–572.

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.

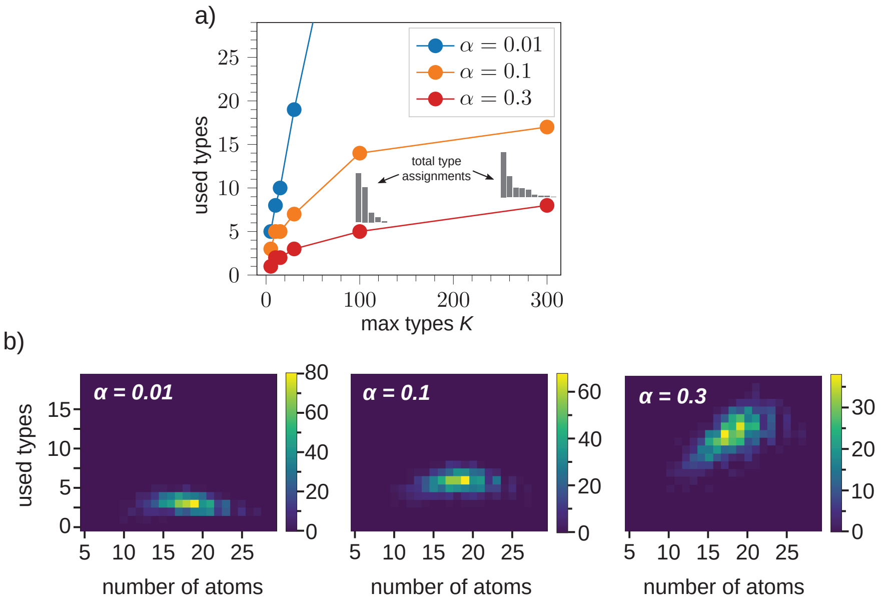

Fig. S1. Saturation analysis of the used environment types w.r.t. to the hyperparameters K and α. a) The used environment types over the maximum number of types K is depicted for three different values of the entropy trade-off factor α. For two runs, additional bar plots indicate the total number of 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.

Choosing r_{min} slightly above the covalent bond distance of organic atoms, results in the best representation of the molecular graph. Hence, r_{min} does not require any tuning. The cut-off radius r_{bead} is associated with the expected size of moieties.

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 $r_{\text{bead}} = 1.8$, $r_{\text{min}} = 3.5$, $\alpha = 0.16$, $K = 100$. 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.

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

Figure S2 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.

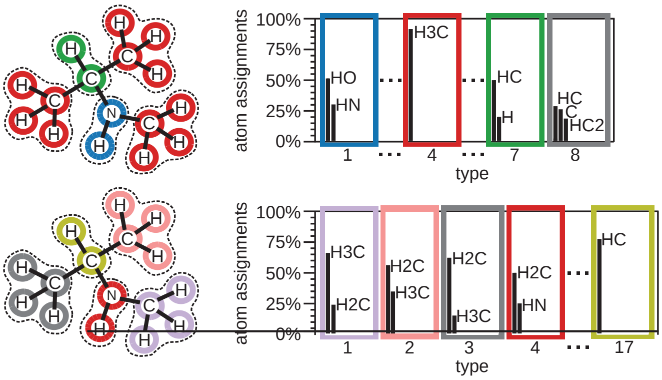

Fig. S2. Comparison between (**top**) pretrained and (**bottom**) end-to-end MoINN model. For each model the type environments and moieties are depicted (left) for an exemplary molecule and (right) in from of a statistical evaluation based on the test set.

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. S3. 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.

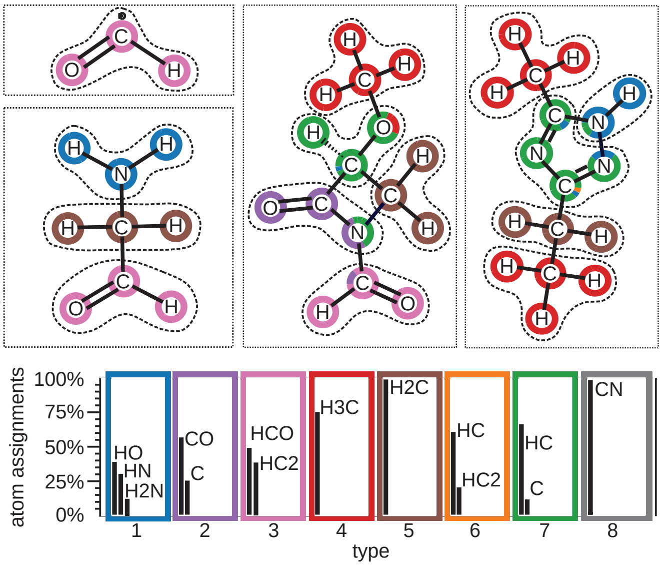

Fig. S3. Common moieties of the QM9 dataset provided by the pretrained MoINN model trained on 110k data points. Equivalent to Fig. 2, the top shows four exemplary molecules along with type assignments (colored circles) and moieties (enclosed by dashed lines). The bottom shows the distribution of environment 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 S4 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 S5 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.

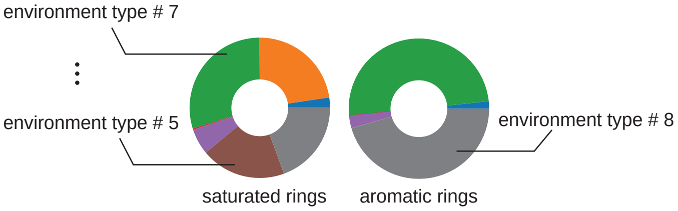

Fig. S4. Average ratio of environment types in saturated and aromatic rings. Each color represents a corresponding environment type.

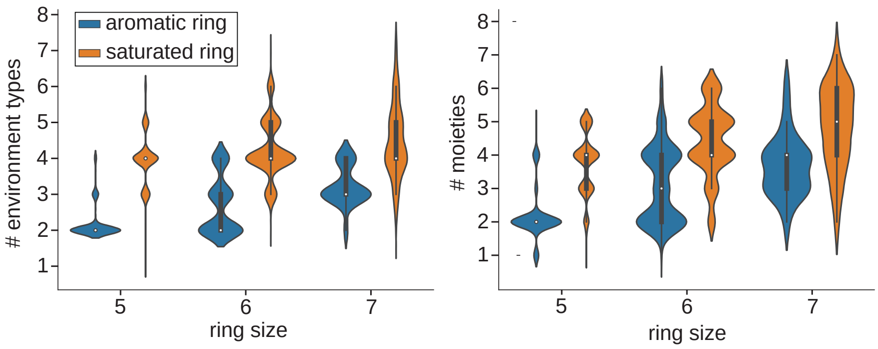

Fig. S5. Comparison between aromatic rings and saturated rings w.r.t. the number of comprised (left) environment types and number of (right) individual moieties. The number of moieties has been determined by the breadth-first search described in Section S2.

For a qualitative comparison between saturated and aromatic rings we show some exemplary molecules in Fig. S6. Equivalent to the quantitative study, the depicted aromatic and saturated rings contain between five and seven heavy 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.

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. S7. 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. S8.

Another approach to verify that the type assignments are chemically meaningful is described in Section 3.1. To this end, we construct molecular fingerprints based on the environment types and compare them to conventional molecular fingerprints by training several linear models on the respective fingerprints. The architecture of those linear models is defined as

$$\mathbf{y} = \mathbf{X}\mathbf{W}$$

with the trainable weight matrix $\mathbf{W} \in \mathbb{R}^{F\times 1}$, the feature matrix $\mathbf{X} \in \mathbb{R}^{N\times F}$, and the output $\mathbf{y} \in \mathbb{R}^{N}$. N denotes the number of samples and F denotes the feature size. All models are trained in an identical fashion, with a learning rate of $\alpha = 10^{-4}$, batch size 100 and the Adam [2] optimizer to minimize the loss term which is simply represented by the mean squared deviation between model prediction and the property of the respective sample. The dataset is divided into

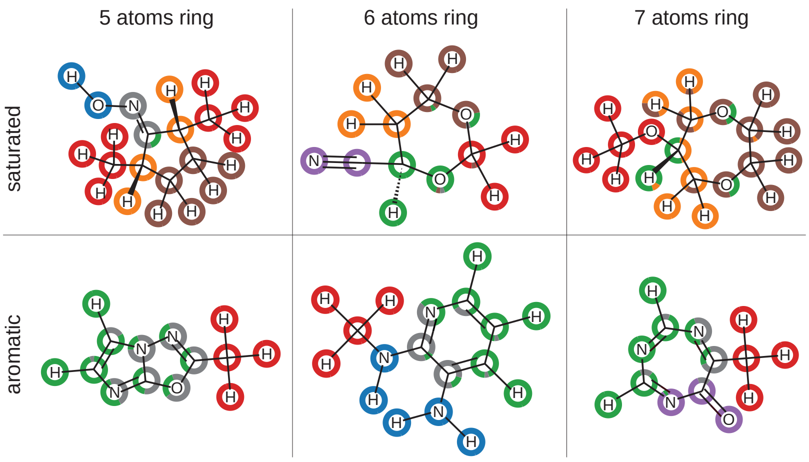

Fig. S6. Six exemplary molecules and their environment type assignments containing saturated and aromatic rings. Rings of three different sizes are compared.

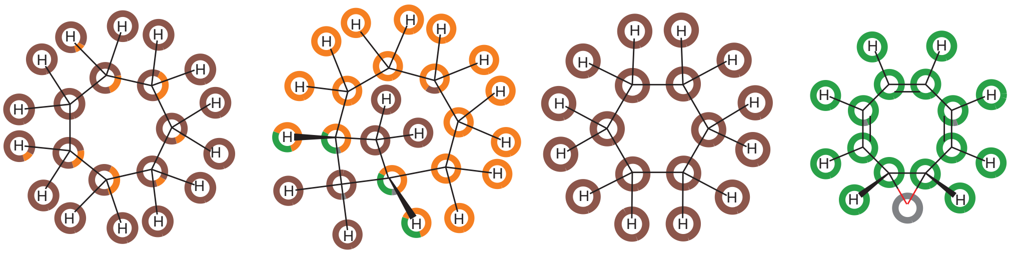

Fig. S7. Largest substructures in QM9 identified by breadth first search based on the environment types of MoINN.

a training set comprising 110k samples, a validation set of 10k samples and a test set of roughly 22k samples. During the training procedure we use a learning rate scheduler ($\alpha_{\text{decay}} = 0.8$, $\alpha_{\text{patience}} = 25$) and early stopping to ensure picking the best performing models for the experiment. The training of all models has converged after at most 500 epochs.

The use of a supervised MoINN model for this task does not yield the same level of performance ($\text{MAE}_\mu = 0.11\,\text{Debye}$, $\text{MAE}_H = 54.65\,\text{eV}$, $\text{MAE}_F = 70.79\,\text{eV}$, $\text{MAE}_{U0} = 67.89\,\text{eV}$). In the unsupervised case, incorporating adaptable representations during the training of MoINN, along with the resulting flexibility of the fingerprints, yields better performance.

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 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 to 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. S9.

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.

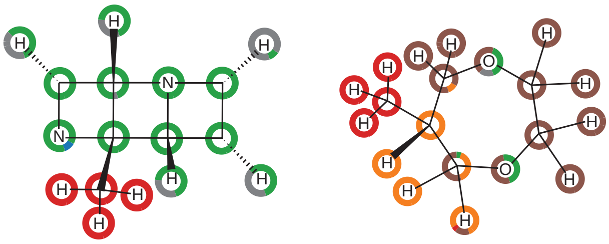

Fig. S8. Large moieties that occur alongside different moieties in the molecule.

TABLE S1
Training set and validation set sizes.

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

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 S10 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 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 ($\alpha_{\mathrm{decay}} = 0.8$, $\alpha_{\mathrm{patience}} = 25$) 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 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 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

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

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 model, respectively. Figure S11 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

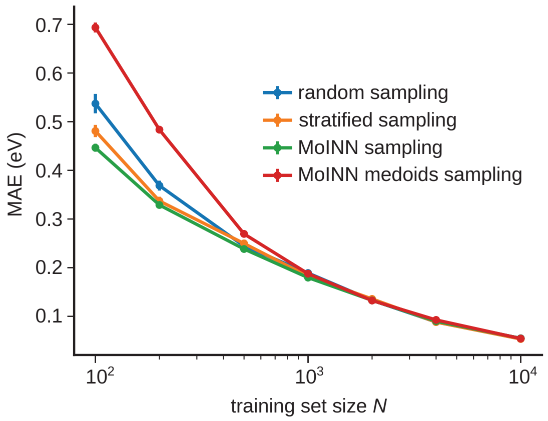

Fig. S9. Mean absolute error (MAE) of energy predictions for SchNet models trained on randomly sampled training sets (blue), training sets obtained by stratified sampling (orange) and training sets selected with MoINN in a self reconstruction manner (green) and using the k-medoids approach (red). Each data point is averaged over five independent training runs and standard errors are indicated by error bars.

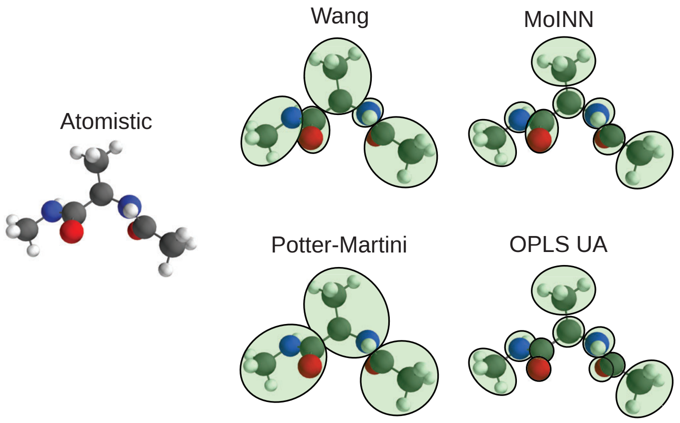

Fig. S10. Comparison between different CG representations provided by Wang et. al. [5], the automated approach for the Martini force field (here referred to as Potter-Martini), our Method MoINN and OPLS UA [6, 7].

of atoms. Figure S12 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 NH_2 and 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.

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

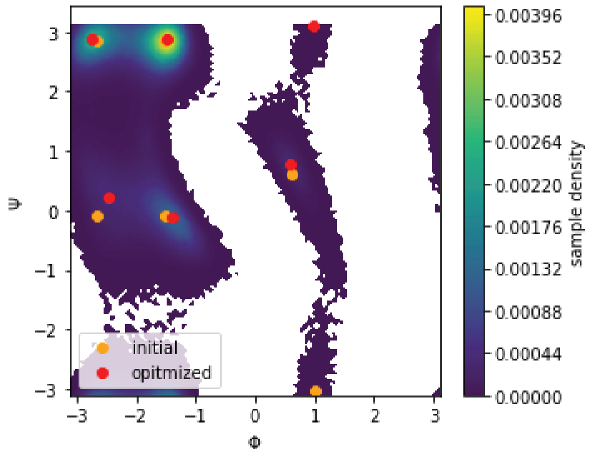

Fig. S11. Density of states of alanine dipeptide projected onto its torsion angles ϕ and ψ with indicated free energy minima. The orange (initial) dots correspond to those sates that are associated with the largest sample densities and the red (optimized) dots indicate the states corresponding to the PMF minima.

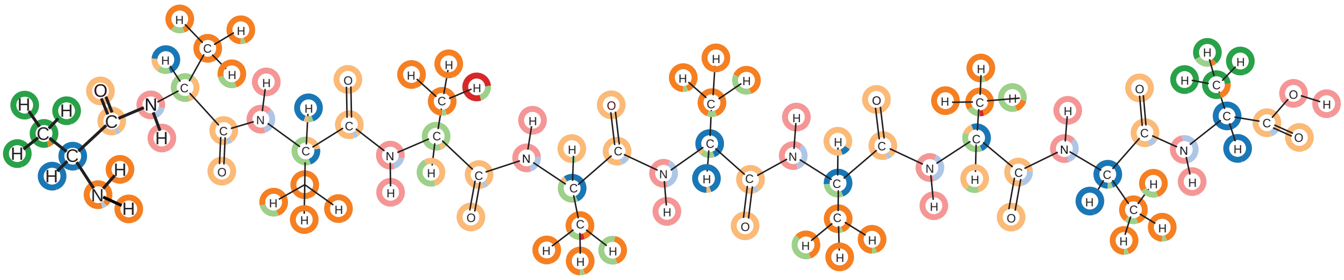

Fig. S12. Environment types for decaalanine provided by MoINN.

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 $r_{\mathrm{cut}} = 1.8\,\text{Å}$ to calculate the adjacency of atom pairs.

In Figure S13, 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.

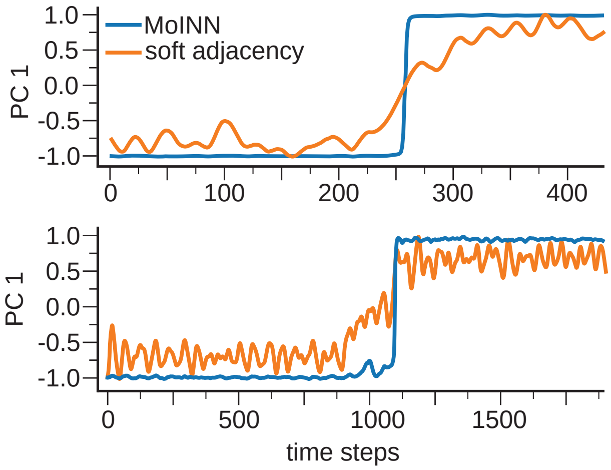

Fig. S13. Reaction coordinates for (**top**) proton transfer in malondialdehyde and (**bottom**) Claisen rearrangement based on MoINN and the soft adjacency matrix, respectively.

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

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

$\mathbf{W}_1 \in \mathbb{R}^{F \times H}$ 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 *i*th 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.