Combining Programmable Potentials and Neural Networks for Materials Problems

Ryan Mohr,[1] **Allan Avila,**[1] **Soham Gosh,** [2] **Ananta Bhattarai,** [2] **Muqiao Yang,** [2] **Xintian Feng,** [3]
Martin Head-Gordon, [3, 4] **Ruslan Salakhutdinov,** [2] **Maria Fonoberova,**[1] **Igor Mezić,**[1, 5]

[1] AIMdyn, Inc., Santa Barbara, CA 93101
[2] Carnegie Mellon University, Pittsburgh, PA 15213
[3] Q-Chem Inc., Pleasanton, CA 94588
[4] University of California, Berkeley, CA 94720
[5] University of California, Santa Barbara, CA 93106

mohrr@aimdyn.com, allan@aimdyn.com, sohamg@cs.cmu.edu, abhattar@cs.cmu.edu, muqiaoy@cs.cmu.edu,
xintian@q-chem.com, mhg@cchem.berkeley.edu, rsalakhu@cs.cmu.edu, mfonoberova@aimdyn.com, mezici@aimdyn.com

Abstract

Force field methods can be used in molecular dynamics simulations to compute and infer macro-level properties of chemical or material systems. However, these force fields can often predict erroneous values due to not being accurate at the quantum level. The ideal would be able to simulate the system from first principles (quantum-level) and scale up to get the material properties. However, this is computationally infeasible. In this paper, we develop a Programmable Potentials methodology that utilizes high-fidelity QM/MM data sets to inform a molecular dynamics (MD) potential. This methodology uses encoding functions containing small neural networks that automatically learn the quantum-level interaction logic of the system from the data and uses these encoding functions to modify standard MD potentials, like Lennard-Jones, which capture the correct long range behavior, but do poorly in the near range. We test the method on the adsorption of hydrocarbons in a zeolite framework. We show that the encoding functions, and the resulting model, generalize well outside their training set and that encoding functions trained with data from a simpler hydrocarbon such as $\mathrm{CH_4}$ can build good models for more complex hydrocarbons such as $\mathrm{C_2H_6}$ and $\mathrm{C_3H_8}$.

Introduction

We seek to use data from small quantum mechanical simulations to build generative models at the molecular scale in order to predict bulk material properties. This is not achievable with current methods. Modeling and solving a system on the level of the Schrodinger equation scales exponentially with the number of electrons in the system, making it infeasible for any system of reasonable size. Density functional theory and periodicity assumptions on the system (so-called QM/MM methods) mitigate this problem somewhat, but are still infeasible for many systems (Jensen 2007). For example, using QM/MM methods for direct simulation of molecular dynamics (MD) for zeolite-catalyzed reactions on the nanosecond time scale requires the sequential evaluation of roughly a million energy and force evaluations, each of which requires about 100 seconds using 8 cores. This equates to approximately 0.1 billion seconds for a simulation, or over 3 years.

Our approach constructs a generative model by taking classical molecular dynamics potentials (such as Lennard-Jones potentials) which capture the correct long-range behavior and combines them with small neural networks which encode the quantum level logic and correct the near-range behavior. The encoding functions multiplicatively modify the classical molecular potentials. Our approach is termed Programmable Potentials with neural networks (PP-NN).

The formulation of Programmable Potentials (Thakur, Mohr, and Mezić 2016) prior to this work relied on the researcher to manually determine and specify the interaction logic and encode this in a boolean algebra. This was a time consuming process and possible only for simple reactions. The idea we present here is to learn this logic automatically by leveraging neural networks and tunable proximity functions, thereby replacing the "hand-tuned" encoding functions. An additional advantage of using neural networks as the encoding functions, and implementing them in packages such as TensorFlow or PyTorch, allows one to leverage the auto-differentiation routines of those frameworks, so that moving from training to deployment in a molecular dynamics simulation bypasses an error-prone, manual differentiation step to get the force fields.

We applied our methods to the materials science problem of hydrocarbons binding in a zeolite framework, both undoped (MFI) and doped (HMFI). Doped zeolites are nano-porous doped silica catalysts that are the world's most widely used catalytic framework for cracking, dehydrogenation, and oligomerization of hydrocarbons (Primo and Garcia 2014). The classic example of doping is replacement of a small fraction Si by Al and an accompanying proton to create a solid acid catalyst (Corma 1995). An illustration is shown in Figure 1. Doping a zeolite turns a non-functional material into a functional material, where catalyst function is controlled by short-range electronic effects from doping, as well as shape-selective effects (pore size and

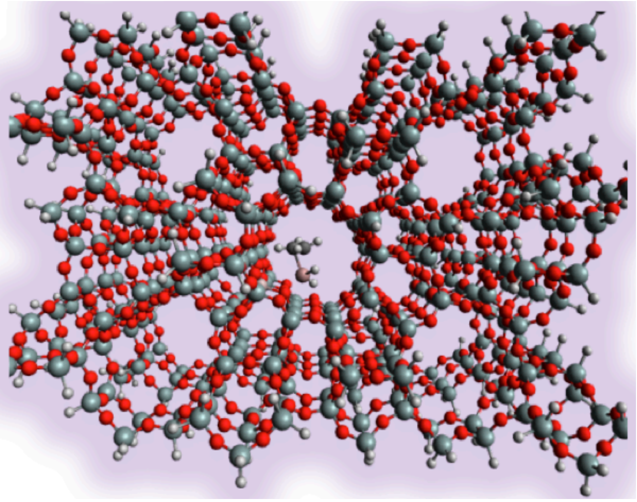


Figure 1: An illustration of the 3-d pore structure of the zeolite, MFI, showing 437 T sites (Si atoms). Doping is the replacement of one or more Si atoms by other elements such as Al, Sn, etc, or, as illustrated here, extra-framework atoms that attach to the side of pores, often at sites that are already doped. The properties of the doped site are fundamentally different than Si, and, in favorable cases, make functional materials with unique properties, controlled by the combination of short-range electronic and steric/shape effects, and long-range electrostatic and dispersion interactions.

geometry) (Smit and Maesen 2008), and long-range effects (remote charges, and polarization) from the extended framework (Mansoor et al. 2018).

Methods

Pure force field methods often do not capture a potential field correctly. While giving the correct form at long range, the short range energies can be incorrect. Ideally, one would like to use a full quantum mechanics simulation, but this is intractable except for small systems. QM/MM hybrid approaches are introduced to relax these limitations. As mentioned above however, these methods still suffer from computation intractability if one would like to simulate them over a reasonable time scale, which would be needed to extract macroscale properties of the material.

Programmable Potentials is a force field methods that takes into account the reaction logic of the system, quantum or otherwise, given a data set. Since it is a force field method, this makes it amenable for molecular dynamics simulations. Using small, high-fidelity QM/MM data sets gives a force field model that can be evaluated outside its training set, setting the stage for long time-scale simulations that are still informed by quantum level information.

The Programmable Potential model takes the form of

$$E_{PP} = \sum_{j} \sum_{k>j} V_{jk} S_{jk} \tag{1}$$

where V_{kj} is a pairwise potential between atoms j and k, such as the Lennard-Jones potential in Eq. (3), and S_{jk} is an encoding function that encodes the interaction logic of the system. In bond-breaking reaction systems, the encoding functions are usually restricted to have values between 0 and 1 and act as a switch that turns individual bonds on or off. In static systems, like binding of guest molecules such as CH_4 in a host zeolite framework, the encoding function values are not squashed in this way.

In original formulation of Programmable Potentials (Thakur, Mohr, and Mezić 2016), the coarse-level logic of the system was encapsulated in the encoding functions. This logic was first constructed using a Boolean algebra generated by indicator functions on intervals $[0, \alpha)$. Once the logic was constructed, the indicator functions were replaced with smooth variants (see Eq.(2)) whose parameters were then optimized over a training data set. This process could be quite complicated as the system moved past very simple reactions and would require a lot of man-hours and ingenuity. The formation in this paper replaces these hand-tuned encoding functions with encoding functions utilizing small neural networks to automatically learn the interaction logic. The neural networks used were densely connected networks with, generally, no more 8 layers with neurons ranging between 8 and 64 neurons for a layer.

For reactions between a small number of atoms, we can apply an encoding function for each interaction. However, for the zeolite problem, using an encoding function for every interaction is infeasible. This would require optimizing thousands of neural networks simultaneously due to the number of atoms in the zeolite framework. Therefore, we learned an encoding function for each unique interaction-type and limited them to zeolite atoms close to the hydrocarbon. For example, every silicon-carbon interaction used the same encoding function. This reduced the number of encoding functions that needed to be trained from thousands to 4 (for the undoped zeolite). This also enabled transfer learning to other systems by learning a "universal" encoding function for each interaction type.

For each unique atom pair type interaction, we use one neural network to learn the corresponding encoding function. For example, in the undoped zeolite problem, we have four separate neural networks in total, encoding S_{SiC}, S_{SiH}, S_{OC}, S_{OH} respectively. The networks take the pairwise distance as input, and outputs the modified Lennard-Jones potential. Recall that the modification for a pairwise potential only happens for atoms in the QM region. For those atoms outside QM region, we keep the original LJ pairwise potential for the energy calculation. The overall process is summarized in Algorithm 1.

An additional advantage of using neural networks as the encoding functions, and implementing them in frameworks such as TensorFlow or PyTorch, allows one leverage the auto-differentiation methods of those framework so that moving from training to deployment in a molecular dynamics simulation bypasses an error-prone, manual differentiation step to get the force fields.

Finally, the choice of the functional form (1), rather than a pure neural network, gives a generative model. Neural networks interpolate well, but often have difficulty extrapolating outside the data sets they were trained on. Our methods

Algorithm 1 Programmable Potentials with Neural Networks

```
/* Add pairwise potentials for atoms in QM region */
for atom $j$ in hydrocarbon do
    for zeolite atom $k$ in QM region do
        $E_{pp} \leftarrow E_{pp} + V_{kj}(r_{kj}) \cdot S_{kj}(r_{kj})$
    end for
end for

/* Modify pairwise potentials for atoms outside QM region */
for atom $j$ in hydrocarbon do
    for zeolite atom $k$ not in QM region do
        $E_{pp} \leftarrow E_{pp} + V_{kj}(r_{kj})$
    end for
end for
return $E_{pp}$
```

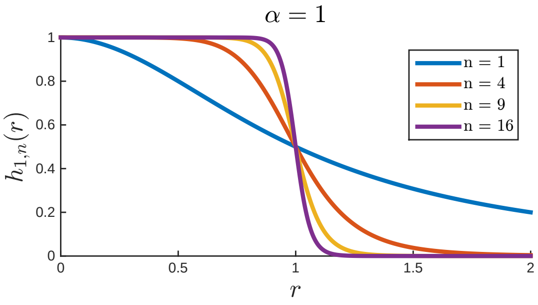


Figure 2: Proximity function with cutoff distance $\alpha = 1$.

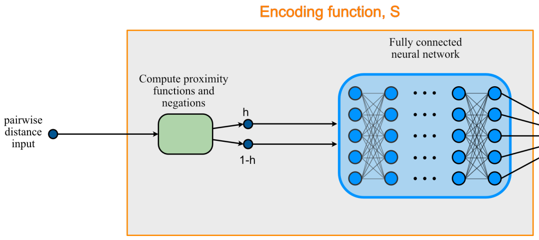
extrapolate well outside the training set.

The encoding functions S_{jk} are structured as follows. Pairwise distances between atoms are the inputs. For each pairwise distance input, a proximity function is formed. The proximity function has the form

$$h(r; \alpha, n) = \frac{1}{1 + (\frac{r}{\alpha})^n}, \tag{2}$$

where the parameter α controls the cutoff distance and the parameter n controls how fast the cutoff is. These are both tunable parameters allowing the model to determine the correct scales during optimization. Figure 2 shows proximity functions at different parameter values n with a fixed cutoff distance $\alpha = 1$. The proximity functions and their negations $(1 - h)$ are then fed into a neural network which combines them to learn the interaction logic from the data set. Figure 3 sketches the basic architecture of the encoding function. We call this architecture Programmable Potentials with Neural Networks (PP-NN). We compare this new architecture with the baseline model consisting of pure Lennard-Jones interactions.

In addition to comparing the new PP-NN architecture with the baseline LJ model, we also compare this with a variety of Programmable Potentials with manually specified encoding functions. As an example, consider developing a programmable potential for CH_4 in the zeolite framework with



Encoding function, S
Fully connected neural network
Compute proximity functions and negations
pairwise distance input
h
1-h
S

Figure 3: Basic encoding function architecture.

manually specified interaction logic. The molecular dynamics (MD) baseline potential that we are utilizing is a 12-6 Lennard-Jones (LJ) potentials, shown below in equation (3)

$$V_{LJ}(r_{ij}) = \sqrt{\epsilon_i \epsilon_j}\left[\left(\frac{R_i + R_j}{r_{ij}}\right)^{12} - 2\left(\frac{R_i + R_j}{r_{ij}}\right)^{6}\right], \tag{3}$$

where r_{ij} is the interatomic distance between atom i and j, $\sqrt{\epsilon_i \epsilon_j}$ is the minimum of the potential energy and $R_i + R_j$ yield the distance at which the potential is minimal.

The baseline LJ potential is modified with encoding functions according to unique atom pair types. Therefore, we only develop encoding functions for Si-C, Si-H, O-C, O-H, Al-C, and Al-H interactions. We have selected an encoding function which only modifies the potentials of atoms whose interatomic distance is within a cut off radius. Furthermore, the encoding function depends only the interatomic distance between the zeolite and methane atom of interest. For example the LJ potential between a carbon and a zeolite atom will be modified by an encoding function that depends only on the interatomic distance between the carbon and zeolite atom. The analytic form of the resulting "hand-tuned" potential for a Si-C and O-H interaction below.

$$\begin{aligned} V(r_{\mathrm{SiC}}) &= V_{LJ}(r_{\mathrm{SiC}}) * S(r_{\mathrm{SiC}}) \\ &= V_{LJ}(r_{\mathrm{SiC}}) * (1 - h(r_{\mathrm{SiC}})) \\ V(r_{\mathrm{OH}_1}) &= V_{LJ}(r_{OH_1}) * S(r_{OH_1}) \\ &= V_{LJ}(r_{OH_1}) * (1 - h(r_{OH_1})), \end{aligned} \tag{4}$$

where in the above equations $h(r; \alpha, n)$ is the standard proximity function that is used within the programmable potentials methodology.

As mentioned above, the programmable potential seeks to modify the LJ potentials of zeolite atoms that fall within a cut off radius of the hydrocarbon. This procedure essentially divides the entire system into two parts, where one is governed by the molecular dynamics interactions and the other is governed by the quantum mechanic interactions. However, the QM/MM data provided by Q-Chem was generated in a slightly different manner. Specifically, the QM region was defined by Q-chem to be composed of the closest silicon atom to the hydrocarbon along with eight additional neighbors of that silicon atom (4 oxygen and 4 silicon atoms). Therefore, the Q-Chem data was generated via a connectivity based QM region whereas our (hand-tuned) programmable potentials were generated via a radial cut off

QM region. The neural network programmable potential that we have developed is also a connectivity based potential. Therefore, we have also focused on developing connectivity based programmable potentials with hand-tuned logic to serve as additional baseline for the neural network programmable potentials.

We list the models tested for the undoped zeolite problem below.

- Case 1: Pure Lennard-Jones (LJ) potential using Q-Chem provided LJ parameters. No optimization. This case serves as the baseline.
- Case 2: Pure Lennard-Jones potential with optimization of LJ parameters for carbon-hydrogen interactions.
- Case 3: Radial distance based programmable potential using Q-chem provided LJ parameters and optimization of carbon-hydrogen encoding functions.
- Case 4: Radial distance based programmable potentials with optimization of the LJ and encoding functions for carbon-hydrogen.
- Case 5: Connectivity based programmable potential using Q-chem provided LJ parameters and optimization of carbon-hydrogen encoding functions.
- Case 6: Connectivity based programmable potentials with optimization of the LJ parameters and encoding functions for C/H interactions.
- Case 7 (NN in plots): Connectivity based programmable potential with neural network using Q-chem provided LJ parameters. This architecture is the focus of this paper.

Results

We found that our models generalize well outside the training set. For example, undoped MFI has 12 chemically distinguishable silicon atoms, called symmetry regions and labeled T1 to T12. For all models, only the T1 symmetry data was used for training. The 11 other symmetry regions were used for testing. Despite being trained on less than 10% of the data, the model beat a pure Lennard-Jones baseline on the T1 symmetry and also generalized well to the other symmetry regions. We compare the results with the true data using the mean relative error

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|T_i - F_i|}{\max_i(T) - \min_i(T)}, \qquad (5)$$

where T_i is the true energy of the system (from high-fidelity QM/MM computations) at sample location i and F_i is the energy of the model at sample location i.

We also tested how well our trained models generalized to other hydrocarbons. For example, we trained encoding functions using the CH_4, undoped zeolite (MFI), T1 symmetry data. These trained encoding functions were then used to build a model for C_2H_6 in MFI. This new model beat a pure Lennard-Jones potential for C_2H_6+MFI on almost all the symmetry regions. With a small amount of retraining of the CH_4-learned encoding functions using C_2H_6+MFI T1 symmetry data, the model beat a pure Lennard-Jones potential on all symmetry regions and was comparable or beat a Programmable Potentials model trained directly on the C_2H_6+MFI T1 symmetry data (see Figure 6). We also tested transferring the CH_4 encoding functions to C_3H_8 (Figure 7). One lesson learned here was that the capacity of the neural network needed to match the complexity of the problem we wanted to transfer the learning to. For example, the first approach used fully-connected neural networks having 4 layers with 8 neurons each. While it adequately learned the CH_4+MFI data (with good generalizations to the other symmetries), it did not do a good job of transferring to the C_2H_6+MFI data sets. Increasing the capacity of the network (5 layers with 16, 32, 64, 32, 16 neurons respectively) allowed good transferability. The conjecture is that the network needs to have enough capacity to match the complexity of the quantum logic of the problem to be transfered to. Currently, this is done in an ad-hoc manner. Having more formalized or precise heuristics or methods to determine such a match requires additional effort and could be a topic for a future research.

Finally, we tested transferring encoding functions from the undoped MFI framework to the doped HMFI framework, a more difficult problem (Figure 8). Doped zeolite is formed by replacing a small fraction of silicon atoms with aluminum atoms. These aluminum atoms present a challenge to transferring the encoding functions. This is due to the undoped MFI data sets not containing any information about how aluminum reacts with the hydrocarbons. The introduction of aluminum can be considered a black swan event from the viewpoint of the encoding functions. Despite this, the encoding functions generalized well from CH_4+MFI (methane in undoped zeolite) to CH_4+HMFI (methane in doped zeolite). Again the encoding functions were only trained using CH_4+MFI T1 symmetry data and generalized well to the other symmetry regions of CH_4 in doped zeolite (of which there are only 10 instead of 12).

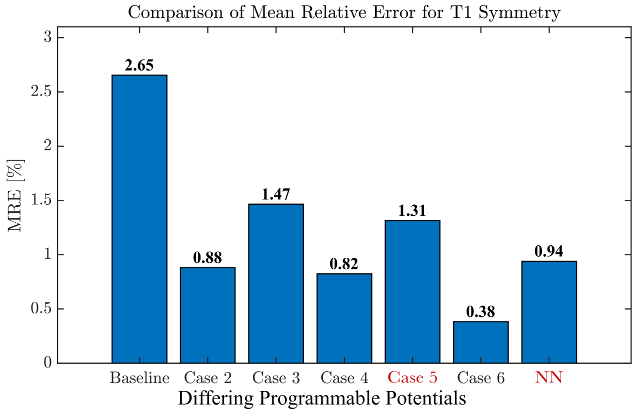


Figure 4: CH_4+MFI train set error. The T1 data was used for training the models. Case 1 is the pure LJ baseline. Cases 5 and NN (case 7) are the most directly comparable. Both use connectivity-based definitions of the QM region and neither optimizes any LJ parameters. The main difference is in whether they use hand-tuned logic (case 5) or use neural networks (NN).

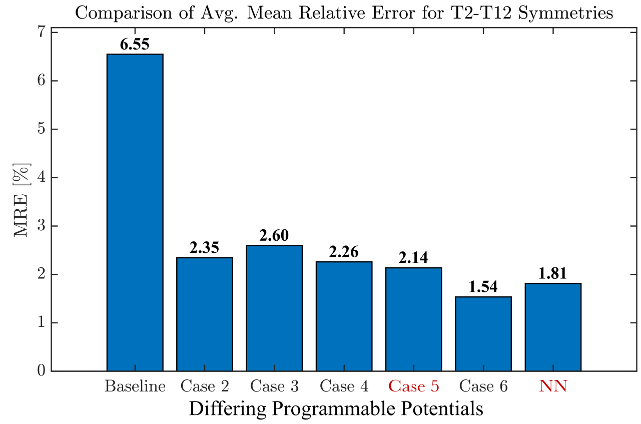


Figure 5: CH_4+MFI average test set error. The models were trained on the T1 data set and tested on the T2–T12 symmetry data. The MRE on the data sets for each test symmetry region was computed and then averaged. Cases 5 and NN (case 7) are the most directly comparable.

Conclusions

We have presented the Programmable Potentials with Neural Networks (PP-NN) methodology in the context adsorbing hydrocarbons in zeolite frameworks. The methodology builds generative models out of classical molecular dynamics potentials—such as Lennard-Jones potentials, which give the correct long-range behavior—and small, targeted neural networks that learn the quantum level logic between atoms if given QM/MM data sets or other quantum mechanical data sets.

The learned encoding functions can be trained on simpler hydrocarbons and transmitted to more complex hydrocarbons and beat the baseline Lennard-Jones potential. Using the previous trained encoding functions as initial conditions for the more complex hydrocarbons, gives better performance than an encoding function that was randomly initialized and trained on the more complex hydrocarbons data set. This ability to transfer seems to require a higher capacity network than is needed to just learn the potential for the simpler hydrocarbon.

Looking forward, the computational catalysis community stands to benefit from the development of efficient, accurate and transferable machine-learning architectures such as we have developed in this program. This community has high standards for acceptable accuracy, because significant errors in the predicted energies enter observable quantities such as relative populations of binding sites, turnover frequencies, etc., in the exponential of Boltzmann factors or Arrenhius rate constants. Small errors in the relative binding energies can lead to erroneous predictions for gas adsorption, and likewise can even lead to erroneous predictions of reaction mechanisms. The goal in these communities is to have ML learned force fields that have the accuracy of quantum methods, but at the cost of force field methods. The Programmable Potentials neural network (PP-NN) methodology is a promising initial step towards this goal. Since, once trained, the PP-NN has an execution time on the order of classical force field methods, molecular dynamics simulations having quantum-level accuracy could be simulated in hours or days rather than years for a comparable QM/MM simulation.

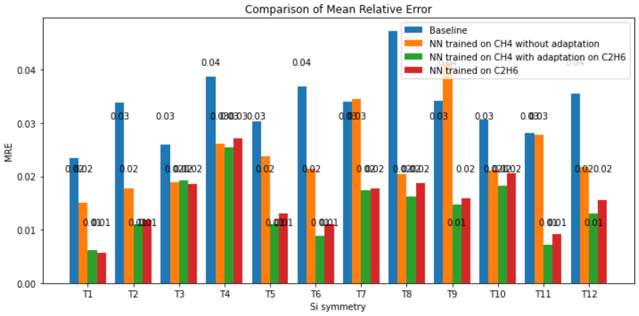


Figure 6: CH_4+MFI generalization to C_2H_6+MFI. Baseline (blue) is a pure LJ potential on the C_2H_6+MFI data set. CH_4 without adaptation (orange) uses the encoding functions trained using the CH_4+MFI T1 data set to build a Programmable Potential for C_2H_6+MFI. CH_4 with adaptation on C_2H_6 (green) is the same of the previous model except now a small amount of training epochs using the T1 symmetry data set of C_2H_6+MFI is used to update the encoding functions. The final model (red) uses encoding functions trained from scratch on the C_2H_6+MFI T1 data set without leveraging on information from CH_4+MFI.

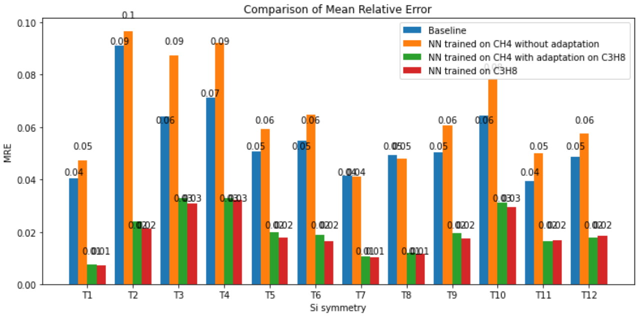


Figure 7: CH_4+MFI generalization to C_3H_8+MFI (undoped to doped zeolite). The same as in Figure 6 except all of the C_2H_6+MFI data sets were replaced with C_3H_8+MFI data sets.

Acknowledgements

This work was supported under DARPA contract HR0011-19-9-0026.

References

Corma, A. 1995. Inorganic Solid Acids and Their Use in Acid-Catalyzed Hydrocarbon Reactions. *Chem. Rev.* 95(3): 559–614.

Jensen, F. 2007. *Introduction to Computational Chemistry*. John Wiley & Sons, second edition.

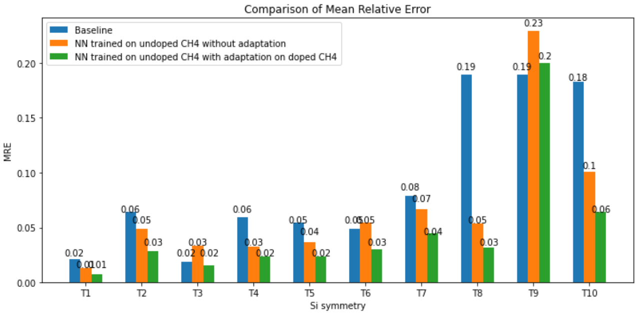


Figure 8: CH_4+MFI generalization to CH_4+HMFI. The encoding functions trained using CH_4+MFI (methane in undoped zeolite) T1 symmetry data is used to build Programmable Potential models for CH_4+HMFI (methane in doped zeolite). Baseline (blue) is a pure LJ potential. CH_4 without adaptation uses the already trained encoding functions and does not do do any retraining. CH_4 with adaptation uses the trained encoding functions and trains for a small number of epochs using the CH_4+HMFI T1 symmetry data.x

Mansoor, E.; Van der Mynsbrugge, J.; Head-Gordon, M.; and Bell, A. T. 2018. Impact of long-range electrostatic and dispersive interactions on theoretical predictions of adsorption and catalysis in zeolites. *Catal. Today* 312(SI): 51–65.

Primo, A.; and Garcia, H. 2014. Zeolites as catalysts in oil refining. *Chem. Soc. Rev.* 43(22): 7548–7561.

Smit, B.; and Maesen, T. L. M. 2008. Towards a molecular understanding of shape selectivity. *Nature* 451(7179): 671–678.

Thakur, G. S.; Mohr, R.; and Mezić, I. 2016. Programmable Potentials: Approximate N-body potentials from coarse-level logic. *Scientific Reports* 6(1): 33415.