Electron-Passing Neural Networks for Atomic Charge Prediction in Systems with Arbitrary Molecular Charge

Derek P. Metcalf,[†] Andy Jiang,[†] Steven A. Spronk,[‡] Daniel L. Cheney,[‡] and C. David Sherrill[*,†]

[†]*Center for Computational Molecular Science and Technology, School of Chemistry and Biochemistry and School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0400, USA*

[‡]*Molecular Structure and Design, Bristol Myers Squibb Company, P. O. Box 5400, Princeton, NJ 08543*

E-mail: sherrill@gatech.edu.

Abstract

Atomic charges are critical quantities in molecular mechanics and molecular dynamics, but obtaining these quantities requires heuristic choices based on atom-typing or relatively expensive quantum mechanical methods to generate a density to be partitioned. Most machine learning efforts in this domain ignore total molecular charges, relying on overfitting and arbitrary rescaling in order to match the total system charge. Here we introduce the electron-passing neural network (EPNN), a fast, accurate neural network atomic charge partitioning model that conserves total molecular charge by construction. EPNNs predict atomic charges very similar to those obtained by partitioning quantum mechanical densities, but at such a small fraction of the cost that they can be easily computed for large biomolecules. Charges from this method may be used directly for molecular mechanics, as features for cheminformatics, or as input to any neural network potential.

Introduction

Atomic charge partitioning is the process by which portions of the electron density are assigned to atomic nuclei. This procedure is critical for evaluating electrostatic interactions between atoms and molecules with molecular mechanics (MM) and molecular dynamics (MD). A number of approaches exist to partition an electron density computed from a quantum mechanical (QM) method.[1–3] However, the QM computation is much more time-consuming than the MM computation, and hence obtaining the charges from QM is not practical in normal MM/MD applications, except for those involving replicas of only a few distinct small molecules whose charges can be determined once prior to the MM/MD procedure. Similarly, QM is generally unsuitable for determining charges that might be needed in high-throughput computational screening applications. Atomic charges can instead be assigned heuristically, such as with formal charges, but a more sophisticated approach is to tabulate atom-types and choose charges for each atom-type to best reproduce an experimental or high accuracy computable quantity. However, these approaches typically suffer from the inability to encode geometric dependence such as how a system varies during an MD simulation.[4] Recent machine learning approaches provide fast atomic charge estimates and a means to encode the geometric dependence of charges without relying on a costly QM-based procedure.[5–7] At present, however, these methods have no way of encoding total system charges, so are relegated to predictions on systems of the same charge as the training examples or require *ad hoc* scaling of atomic charges. This requirement forces the user to construct individual models for each charge state, which is inefficient and intractable when considering systems with large total charges. Simply scaling charges *ad hoc* has no physical basis and relies entirely on overfitting to geometries from the training set, which may be practical for some applications but is not generally practical. One recent work describes AIMNet-ME, an architecture that encodes total system charge during iterative message-passing updates by allocating individual atomic feature vectors for each possible total charge state.[8] In this way, the model can predict properties while being aware of total system charges, unlike previous methods. While this showed good performance for the small systems studied, it remains unclear if this approach is transferable to large systems or to systems with charges besides -1, 0, or 1.

Here, we introduce an alternative message-passing neural network model for determining atomic charges of neutral or charged molecules with arbitrary charges, that is applicable to large molecular systems. In this initial study, we target the modeling of protein-ligand systems relevant for drug design efforts. Our model is trained on a collection of 3503 neutral and charged molecular fragments relevant for proteins or drug-like molecules, and we demonstrate high accuracy when predicting charges on different neutral and charged systems. We demonstrate the computational efficiency of the approach by applying it to the Galectin-3C protein with 2220 atoms.

Methods

Message-Passing Neural Networks (MPNNs)

The MPNN is a variety of graph neural network described by Gilmer *et al.* as a general framework, unifying many existing schemes to learn from graph structured data.[9] The MPNN has found success in application to chemical problems, where a graph is defined by the tuple $\mathcal{G} = (\mathbf{H}, \mathbf{\Omega})$, with nodes $\mathbf{H} \in \mathbb{R}^{N \times d_h}$ as the set of N atom centers with d_h corresponding features per atom. Edges $\mathbf{\Omega} \in \mathbb{R}^{N \times N \times d_e + 1}$ contain d_e features per atom pair in addition to a binary adjacency variable, which is often chosen according to some distance cutoff or by covalent bonding, but generally can be generated by any process.

The primary goal of the MPNN is to propagate information about neighboring atoms via a *message-passing* step and accumulate that information with an *update*, which can modify the node features and/or edge features. After several such message-passes and updates, a *readout* step is performed to evaluate the graph for some property. In chemistry, the molecular energy is a key quantity, so the readout typically consists of a node-wise evaluation function to acquire "atomic energies," which are then summed to recover an estimate of the molecular energy. In principle, this general approach can be used to regress toward any system-wide, atomic, or atomic-pairwise property, provided appropriate labels. Below are details of a typical MPNN implementation; for connections to other works the authors refer the reader to Reference 9.

Initial states of the graph $\mathcal{G}^0$ can be chosen intuitively and are problem-dependent. For molecular graphs, atomic initial states are often chosen as $\mathbf{h}_v^0 = (OHE(Z_v), [0, ..., 0])$, where $OHE(Z_v)$ is a one-hot encoding of the element of atom v, and $(., .)$ denotes concatenation. The additional zeroes pad the initial state to length d_h and are updated during message-passing. Initial edge states are often defined as the interatomic distance projected on a set of radial basis functions, such as a set of Gaussian functions $\mathbf{e}_{vw}^0 = \{e^{-\eta_j (r_{vw} - \mu_j)^2})\}_j,\ j \in \{1, ..., d_e\}$ for atom pair v, w. η and μ are parameters of the Gaussian distribution which may differ for each j, and r_{vw} is the Euclidian distance between atoms v and w.

Message passes and updates are performed T times, as defined by the user. A message from atom w to v at step t is generated by $M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t) = NN_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t)$ with step-specific dense feed-forward neural network NN_t. All messages to atom v are accumulated by a symmetric function, in this case a sum:

$$m_v^{t+1} = \sum_{w \neq v} M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t) \quad . \tag{1}$$

Node feature vectors $\mathbf{h}_v^t$ update according to

$$\mathbf{h}_v^{t+1} = U_t(\mathbf{h}_v^t, m_v^{t+1}) = NN^U(\mathbf{h}_v^t, m_v^{t+1}) \quad , \tag{2}$$

for example, where NN^U is typically a small, dense feed forward neural network. Edges may update analogously but typically do not. Finally, the dimension of the readout function at final step T depends on the target property. In the case of molecular energy, atomic energies are usually from neural network evaluations of the final node features and accumulated with a sum:

$$R = \sum_{v}^{N} NN^R(\mathbf{h}_v^T) \quad , R \in \mathbb{R}. \tag{3}$$

Here, NN^R is another small dense feed-forward neural network.

Electron-Passing Neural Networks (EPNNs)

The EPNN introduced here is a modified MPNN able to predict atomic charges (node-level property) while conserving the total charge (graph-level property) of a molecule or system. This is accomplished by initializing the system with a set of atomic charges $\{q_v^0\} \ \forall \ v \in \mathcal{G}$ such that the total charge $Q^{\mathcal{G}}$ is correct, then only updating charges with operations that conserve the total charge.

Specifically, the EPNN employs an initial message-passing phase followed by an electron-passing phase. This general recipe is detailed in Figure 1, with precise function definitions below. The message-passing phase, just as in the original MPNN formulation, serves to confer and aggregate geometric information about neighboring atoms into the atomic hidden states. Message passes are performed according to equations 1 and 2, where there is a different neural network for each message-passing iteration:

$$M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t) = NN_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}^t). \tag{4}$$

In our implementation, edge features are not updated and instead remain radial basis functions independent of message passing step.

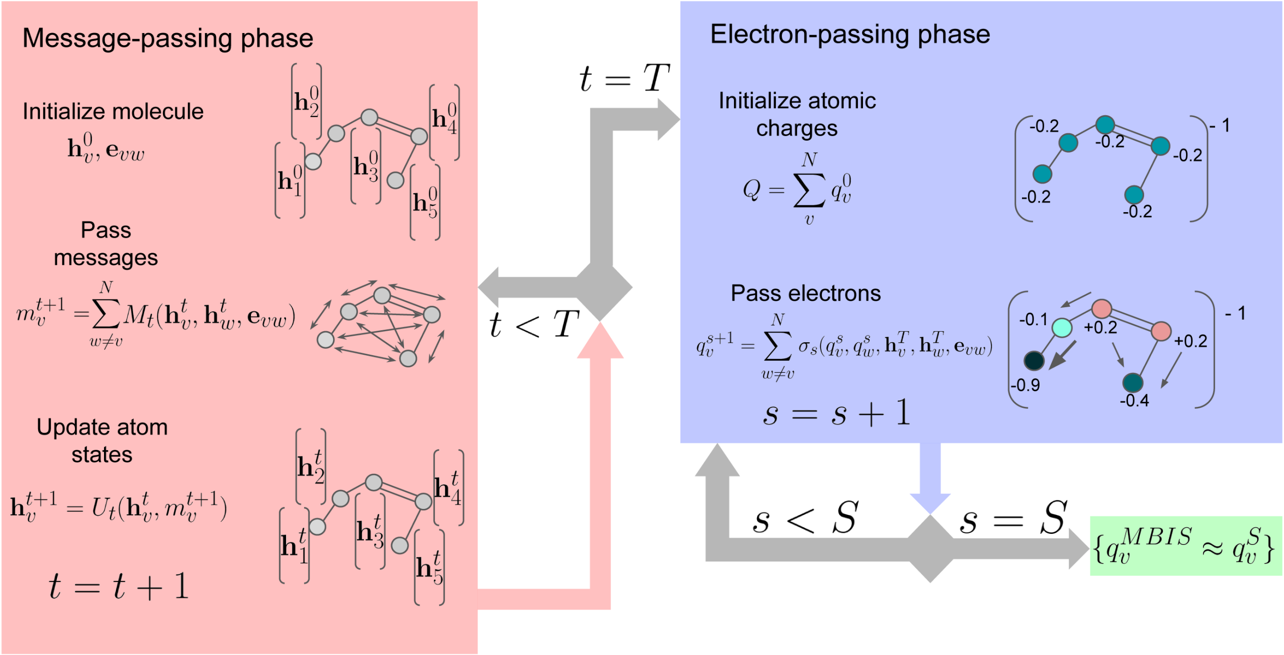


Figure 1: A schematic for electron-passing neural networks.

In lieu of a simple readout function to recover a graph-level property, an electron-passing phase is initiated to recover node-level properties subject to the constant charge constraint. The electron-passing functions σ are also different per electron passing step s as

$$\sigma_s(q_v^s, q_w^s, \mathbf{h}_v^T, \mathbf{h}_w^T, \mathbf{e}_{vw}) = NN_s(q_v^s, q_w^s, \mathbf{h}_v^T, \mathbf{h}_w^T, \mathbf{e}_{vw}) - NN_s(q_w^s, q_v^s, \mathbf{h}_w^T, \mathbf{h}_v^T, \mathbf{e}_{wv}), \tag{5}$$

describing an electron pass from atom w to atom v. Electron passes from all atoms are accumulated and atomic charges are updated with

$$q_v^{s+1} = \sum_{w \neq v}^{N} \sigma_s(q_v^s, q_w^s, \mathbf{h}_v^T, \mathbf{h}_w^T, \mathbf{e}_{vw}). \tag{6}$$

To ensure conservation of electrons, the passing function must be antisymmetric with respect to permuting the two atoms

$$\sigma_s(q_v^s, q_w^s, \mathbf{h}_v^T, \mathbf{h}_w^T, \mathbf{e}_{vw}) = -\sigma_s(q_w^s, q_v^s, \mathbf{h}_w^T, \mathbf{h}_v^T, \mathbf{e}_{wv}), \tag{7}$$

a condition enforced by the functional form of equation 5. After S electron-passing steps have elapsed, the predicted atomic charges are read directly from q_v^S. Unlike during message-passing, where node features $\mathbf{h}_v$ are updated, electron-passing only updates node charges q_v. During training, these charges can be compared with any target charge partitioning such as MBIS, or used in a composite way to reproduce a computed quantity, such as electrostatic interaction energies. Discrepancies from target quantities can be used as an error and backpropagated to optimize the collection of neural networks that define the EPNN. Total message-passing steps T and electron-passing steps S are treated as hyperparameters and are optimized for validation set accuracy. We found $T, S = 3$ were optimal in our studies.

Importantly, the electron-passing phase is functionally independent from the message-passing phase. This means any atomic featurization can be used before electron-passing, not just traditional message-passing as shown in this study. Simpler descriptors such as symmetry functions could act equivalently as inputs, and the iterative nature of the electron-passing phase partially overcomes the locality of the symmetry functions.[10]

Data Collection

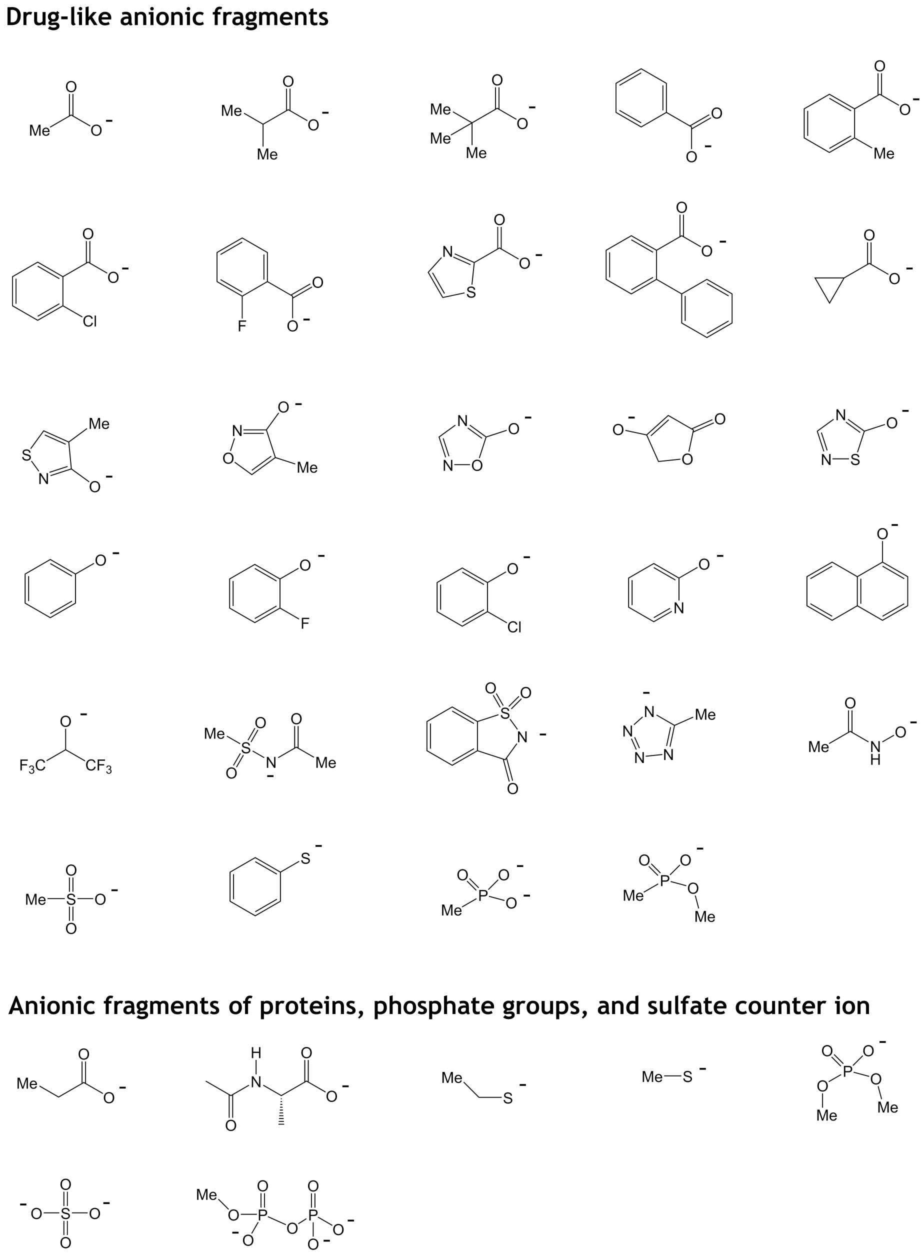
For our model, we choose to infer charges from the popular minimal-basis iterative stockholder (MBIS) method[3] in a supervised manner. Our training and validation data come from three sources: a subset of 1338 small neutral molecules from the QM9 dataset,[11] a set of 62 hand-curated anionic and cationic molecules relevant to drug discovery shown in Figures 2 and 3, respectively, and a 2979 sidechain molecular dimer subset from the sidechain-sidechain interaction (SSI) dataset.[12] SSI contains paired neutral and charged sidechain monomers, including cationic arginine and lysine and anionic aspartate and glutamate. The variety in training and validation data attempts to ensure sound predictions on pharmacologically relevant small molecules and proteins without sacrificing accuracy on neutral systems. The MBIS atomic charges were computed for each system using the HORTON software package[13] to partition densities from density functional theory (DFT) computations with the PBE0 functional[14] and aug-cc-pVDZ basis[15] performed in the Psi4 electronic structure package.[16,17] Additional computational details and all structures and charges are included in the supporting information.

To capture transferability to large systems, we constructed two test cases from the Galectin 3C protein.[18] First, the entire protein was extracted from the PDB entry 6QLP. This geometry was used for model evaluation but we did not compute its charges with the MBIS procedure. Next a three-residue, 80-atom fragment was extracted from the protein, its cleaved bonds capped with hydrogens, and its charges computed with MBIS as above. This particular fragment was chosen for illustrative purposes since it has a negative charge, unlike the positively charged protein.

Results and Discussion

Small Molecule Performance

Of the 4379 structures in the combined QM9 subset, SSI dataset, and the manually curated set representing pharmaceutically relevant molecules, a random 80% were used for training and the remainder for validation. Validation performance relative to the target MBIS values is shown in Figure 4, where each each dataset is shown separately and charge states are indicated by color. Performance on this validation set is competitive with other charge models while utilizing relatively little data.[4,6–8] Noteworthy is the relative challenge in localizing ions, especially anions. The larger errors in these systems are likely a result of larger charge magnitudes (since the atomic charges in molecular ions can more easily have magnitudes near 1), but also suffer from data sparsity. As larger datasets for charged systems become



Drug-like anionic fragments

Anionic fragments of proteins, phosphate groups, and sulfate counter ion

Figure 2: Small anionic molecules relevant to drug discovery. Data preparation and validation information and molecular geometries are included in the supporting information.

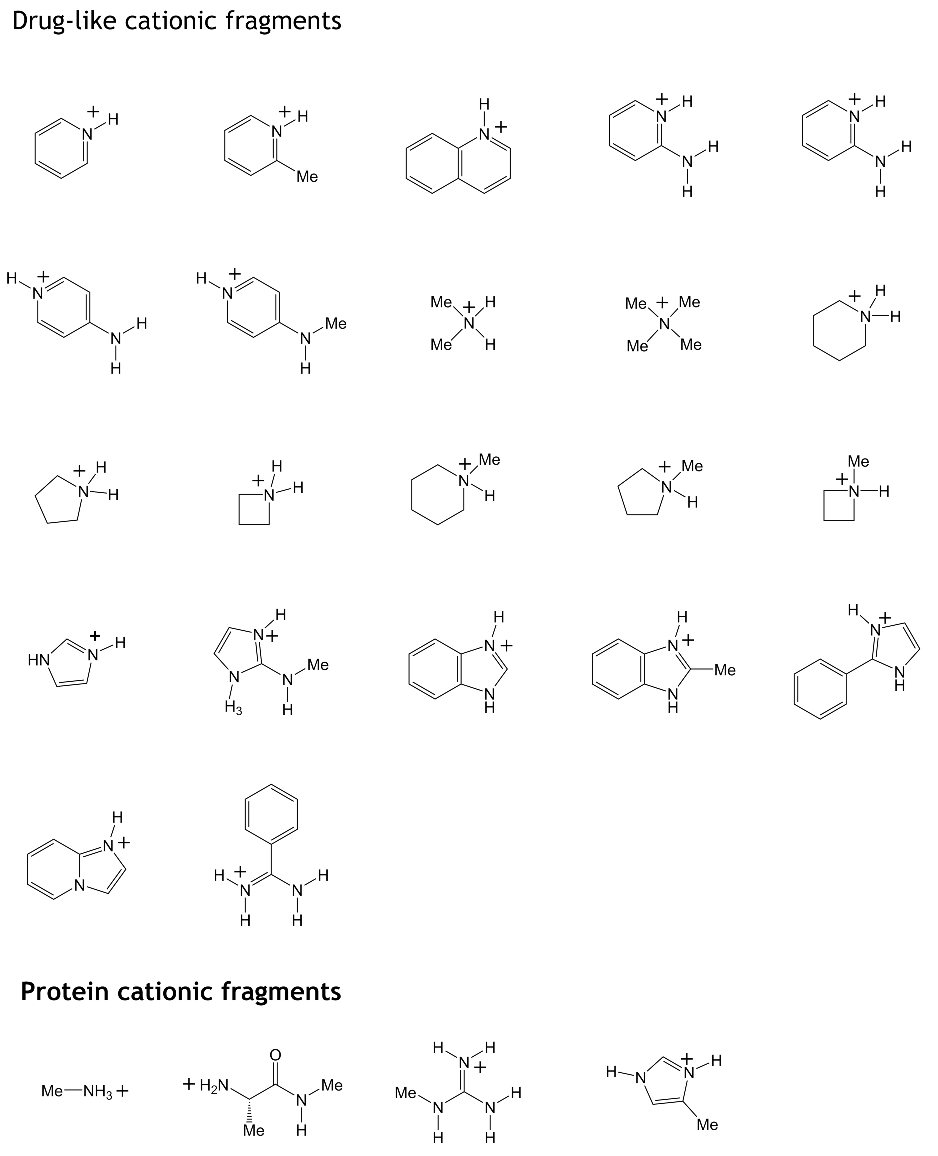


Figure 3: Small cationic molecules relevant to drug discovery. Data preparation and validation information and molecular geometries are included in the supporting information.

available, we believe predictions on anions will approach those of cations and neutral systems. Predictions on the SSI dataset are especially accurate, much of which has to do with shared monomers between the training and validation sets. Nonetheless, prediction errors are smaller than the variance due to charge transfer in a typical charge-neutral or charge-charge interaction in the SSI dataset, so the models present good transferability.

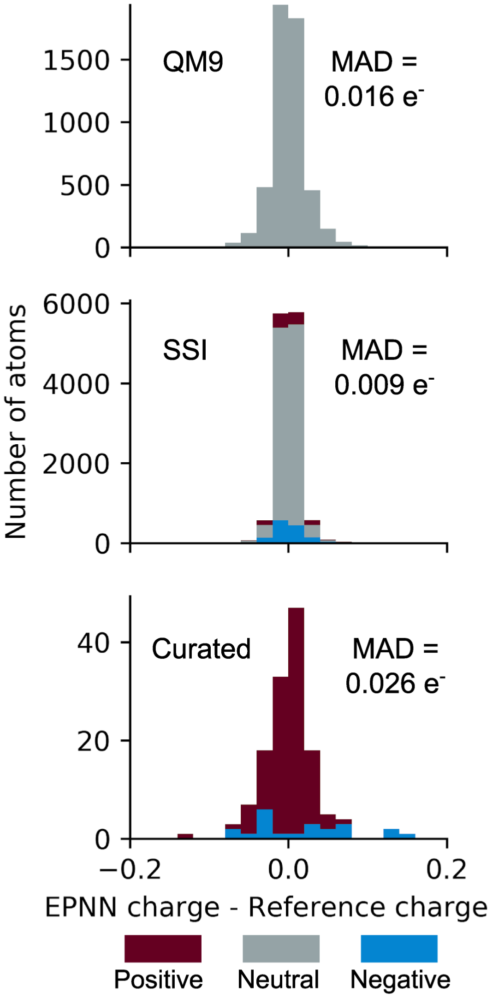


Figure 4: Summary errors from the validation set, differentiated by source data set (QM9, SSI, Curated) and by the charge state of the molecular system.

Charge Transfer and Polarization

Understanding the change in atomic charges in response to nearby molecules is critical to explain many condensed phase phenomena. The SSI dataset, which enumerates interactions between protein sidechains, constitutes a significant fraction of the training and validation data for this charge model. By examining interactions between charged and uncharged sidechains, we can immediately quantify the combined charge transfer and polarization effects. An example of glutamic acid and glutamine from the validation set is illustrated in Figure 5, where the model is able to replicate the charge distribution of each monomer in vacuum, as well as in the presence of another side chain, with reasonable accuracy. The difference in a monomer's charges as one moves from the gas phase to the dimer complex, which we may denote as Δq_i for each atom i, is due to charge transfer to/from the other monomer and polarization of the monomer's electrons, induced by the presence of the opposing monomer. Most of the error in the ML prediction of Δq_i is due to inaccuracy when predicting gas-phase monomeric charges, resulting in a slightly exaggerated view of polarization and charge transfer. Nonetheless, the accuracy of the model is able to qualitatively capture even very subtle effects, like in this charge-neutral interaction.

Protein Validation

To show application to large systems and the extensivity of our charge model with respect to total system charge, we conducted an illustrative study using the Galectin 3C protein.[18] To reduce the protein to a manageable size in order to perform the reference MBIS charge partitioning, we isolated three connected residues and capped the ends to retain sensible bonding. This subsystem has net charge -1, which is fed directly as input along with the Cartesian coordinates of the system. The subsystem is comprised of 80 atoms, larger than any molecule in the training set. Just as with the smaller molecules, the charge is able to self-organize and accurately match the target MBIS partitioning. This comparison is illustrated in Figure 6A. Next, the rest of the protein was added, increasing the net charge to $+2$, a

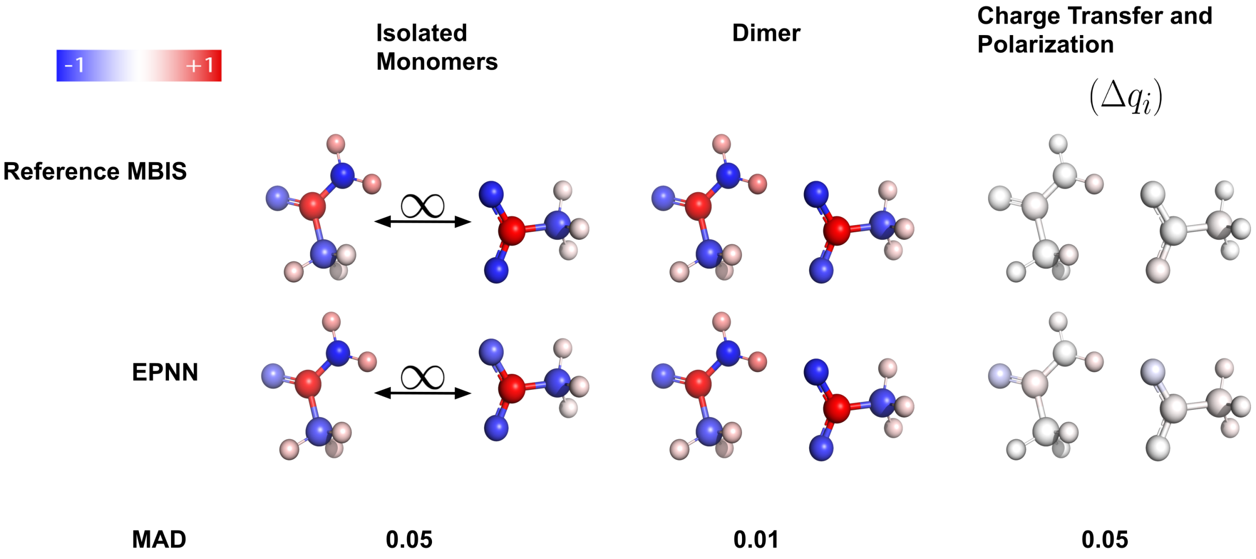


Figure 5: An example polarization and charge transfer interaction between glutamine (left) and glutamic acid (right) sidechains from the SSI dataset.

3-electron change. Of note is the fact 32 atoms in this structure are formally charged, and just tend to cancel, but the net charge is the only input required. The system was again evaluated with the EPNN. The model produces very similar predictions for the subsystem, highlighting its extensivity in system size and total charge magnitude. Indeed, the only large variations in charge from the subsystem computation are in atoms whose bonding was compromised by the capping procedure. A visual comparison is shown in Figure 6B, where charges for the entire protein are approximated but the subset from the fragment study are highlighted.

On a single 8-core Intel Core i7-9800X CPU, the 80-atom subsystem DFT computation took 19 minutes. By comparison, the EPNN evaluation of this system on the same hardware took 0.16 seconds. The 2220-atom supersystem EPNN evaluation took just 62 seconds.

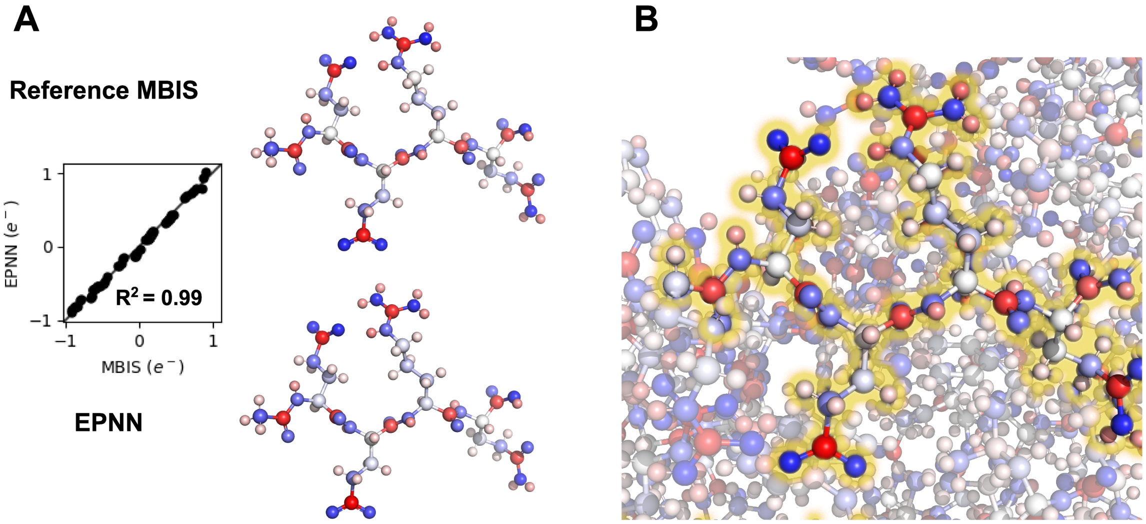


Figure 6: **A.** A comparison between reference MBIS charges and predicted EPNN charges on an anionic fragment of Galectin 3C. Blue indicates negatively charged atoms and red indicates positively charged. **B.** EPNN charge predictions on the entire 2220-atom Galectin 3C protein, net charge +2. The fragment from **A** is backlit to show similarity in charge prediction despite system size and charge changes.

Conclusions

A neural network scheme is introduced which maps directly from Cartesian coordinates and total system charge to an atomic charge partitioning. Unlike other works, this architecture conserves total charge by construction, requiring a single model and no arbitrary scaling. Additionally, the model does not grow or need additional encoding or training data in each specific charge state to express arbitrary total charges. This architecture, dubbed an electron-passing neural network (EPNN), attains high accuracy on charged and neutral drug- and protein-like molecules with a small training set. Subtle charge transfer and polarization effects due to intermolecular interactions are reproduced with good fidelity. The quality of charges estimated with the EPNN is independent of system size – accurate charge predictions on a protein fragment persist when the entire protein of over 2000 atoms and different charge state is evaluated. With appropriate distance cutoff criteria, the computational complexity of the EPNN is linear in number of atoms.

The electron-passing layer is a simple, modular unit that can be used with any atomic featurization as input, making it ideal to use in conjunction with existing machine learning models for energy or other properties. The physically meaningful atomic charge output can be used for molecular mechanics, machine-learned potentials, or as features for cheminformatics. As large datasets including charged systems become publicly available, we hope electron-passing layers can be trained for systems including different elements and a wider range of chemical systems.

Acknowledgement

The authors gratefully acknowledge financial support from Bristol Myers Squibb, and from the U.S. National Science Foundation through grants CHE-1955940 and DGE-1650044.

Supporting Information Available

All code and data used for this study are available at https://github.com/derekmetcalf/epnn

The SI includes computational details in data collection and architectural details of the model.

References

(1) Hirshfeld, F. L. *Theor. Chem. Acc.* **1977**, *44*, 129–138.

(2) Maslen, E.; Spackman, M. *Aust. J. of Phys.* **1985**, *38*, 273–287.

(3) Verstraelen, T.; Vandenbrande, S.; Heidar-Zadeh, F.; Vanduyfhuys, L.; Speybroeck, V. V.; Waroquier, M.; Ayers, P. W. *J. Chem. Theory Comput.* **2016**, *12*, 3894–912.

(4) Bleiziffer, P.; Schaller, K.; Riniker, S. *J. Chem. Inf. Model.* **2018**, *58*, 579–590.

(5) Zubatyuk, R.; Smith, J. S.; Leszczynski, J.; Isayev, O. *Sci. Adv.* **2019**, *5*.

(6) Bleiziffer, P.; Schaller, K.; Riniker, S. *J. Chem. Inf. Model.* **2018**, *58*, 579–590, PMID: 29461814.

(7) Sifain, A. E.; Lubbers, N.; Nebgen, B. T.; Smith, J. S.; Lokhov, A. Y.; Isayev, O.; Roitberg, A. E.; Barros, K.; Tretiak, S. *J. Phys. Chem. Lett.* **2018**, *9*, 4495–4501.

(8) Zubatyuk, R.; Smith, J.; Nebgen, B. T.; Tretiak, S.; Isayev, O. *ChemRxiv* **2020**,

(9) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. *ArXiv* **2017**, *abs/1704.01212*.

(10) Behler, J. *J.Chem. Phys.* **2011**, *134*, 074106.

(11) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. *Sci. Data* **2014**, *1*.

(12) Burns, L. A.; Faver, J. C.; Zheng, Z.; Marshall, M. S.; Smith, D. G. A.; Vanommeslaeghe, K.; MacKerell, A. D.; Merz, K. M.; Sherrill, C. D. *J. Chem. Phys.* **2017**, *147*, 161727.

(13) T. Verstraelen, P. Tecmer, F. Heidar-Zadeh, K. Boguslawski, M. Chan, Y. Zhao, T.D. Kim, S. Vandenbrande, D. Yang, C.E. González-Espinoza, S. Fias, P.A. Limacher, D.

Berrocal, A. Malek, P.W. Ayers HORTON 2.0.1, http://theochem.github.com/horton/, 2015.

(14) Perdew, J. P.; Ernzerhof, M.; Burke, K. *J. Chem. Phys.* **1996**, *105*, 9982–9985.

(15) Papajak, E.; Zheng, J.; Xu, X.; Leverentz, H. R.; Truhlar, D. G. *J. Chem. Theory Comput.* **2011**, *7*, 3027–3034.

(16) Parrish, R. M. et al. *J. Chem. Theory Comput.* **2017**, *13*, 3185–3197.

(17) Smith, D. G. A. et al. *J. Chem. Phys.* **2020**, *152*, 184108.

(18) Berman, H.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.; Weissig, H.; Shindyalov, I.; Bourne, P. *Nucleic Acids Res.* **2000**, *28*, 235–242.