

Poisson-Boltzmann based machine learning (PBML) model for electrostatic analysis

Jiahui Chen¹, Yongjia Xu², Xin Yang³, Zixuan Cang⁴, Weihua Geng^{3,*}, and Guo-Wei Wei^{5,6,*}

¹Department of Mathematics, University of Arkansas, Fayetteville, AR 72701, USA

²Google LLC, 1600 Amphitheater Pkwy, Mountain View, CA 94043, USA

³Department of Mathematics, Southern Methodist University, Dallas, TX 75275, USA

⁴Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA

⁵Department of Mathematics, Michigan State University, MI 48824, USA

⁶Department of Biochemistry and Molecular Biology, Michigan State University, MI 48824, USA

*Correspondence: wgeng@smu.edu, wei@math.msu.edu

ABSTRACT Electrostatics is of paramount importance to chemistry, physics, biology, and medicine. The Poisson-Boltzmann (PB) theory is a primary model for electrostatic analysis. However, it is highly challenging to compute accurate PB electrostatic solvation free energies for macromolecules due to the nonlinearity, dielectric jumps, charge singularity, and geometric complexity associated with the PB equation. The present work introduces a PB based machine learning (PBML) model for biomolecular electrostatic analysis. Trained with the second-order accurate MIBPB solver, the proposed PBML model is found to be more accurate and faster than several eminent PB solvers in electrostatic analysis. The proposed PBML model can provide highly accurate PB electrostatic solvation free energy of new biomolecules or new conformations generated by molecular dynamics with much reduced computational cost.

SIGNIFICANCE This manuscript provides a Poisson-Boltzmann based machine learning (PBML) model for biomolecular electrostatic analysis. The features as the input to the ML models are generated with mathematical algorithms using biomolecular structures and force field. The learned model, which is trained using the most accurate PB solver MIBPB on more than 4000 biomolecules shows improved efficiency and accuracy in electrostatic analysis compared with the popular PB solvers.

1 INTRODUCTION

Electrostatics is ubiquitous in the molecular world. The analysis of molecular electrostatics is of crucial importance to the bioscience research community. There are two significant types of electrostatic analyses, namely, *qualitative* analysis for general electrostatic characteristics, such as visualization and electrostatic steering, and *quantitative* analysis for statistical, thermodynamic and/or kinetic observable, such as solvation free energy, solubility, and partition coefficient.

Molecular electrostatics can be analyzed by explicit or implicit models. Explicit solvent models resolve electrostatic effect in atomic detail and thus are more accurate but can be very expensive for large biomolecular systems. Implicit solvent models describe the solvent as a dielectric continuum, while the solute molecule is modeled with an atomistic description (1). A wide variety of two-scale implicit solvent models has been developed for electrostatic analysis, including generalized Born (GB) (2), polarizable continuum (3) and Poisson-Boltzmann (PB) models (4).

PB models have been applied to calculating protein titration states (5), protein-protein and protein-ligand binding energetics (6), RNA nucleotide protonation (7), chromatin packing (8), etc. The PB theory has also been used for the evaluation of biomolecular electrostatic forces for molecular Langevin dynamics or Brownian dynamics (9). GB methods are faster than PB methods, but provide only heuristic estimates for PB electrostatic energies.

Due to its success in describing biomolecular systems, the PB model has attracted a wide attention in both mathematical and biophysical communities. In the past two decades, many efforts have been given to the development of accurate, efficient, reliable and robust PB solvers. A large number of methods have been proposed in the literature, including the finite difference method (FDM) (10), finite element method (FEM) (11), and boundary element method (BEM) (12, 13). Among them, the FDM is prevalently used in the field due to its simplicity in implementation. The emblematic solvers in this category are Amber PBSA (14), Delphi (15) APBS (16),

MIBPB (17), CHARMM PBEQ (10), etc.

The PB equation is a nonlinear elliptic equation with singular source term. For biomolecules, the continuum-discrete interface is nonsmooth, resulting in a nonlinear elliptic interface problem with discontinuous coefficients and singular source terms (18). These difficulties make finding the numerical solution to the PB equation for biomolecules challenging. There are numerous efforts in developing high order methods for elliptic interface problems in the past two decades (19–21). Among them, matched interface and boundary (MIB) method offers arbitrarily high-order accuracy in principle and up to sixth-order accurate MIB schemes have been constructed for three-dimensional complex interfaces (21). The MIB based second-order accurate PB solver, the MIBPB, was constructed to take care of biomolecular interfaces and singular charges (22). The recent development of the Eulerian solvent excluded surface (ESES) (23), which provides analytical biomolecular surface representation in the Cartesian domain, improves the stability and robustness of the MIBPB solver.

Nonetheless, the generation of highly accurate electrostatic potentials for large biomolecules can be extremely expensive. For example, it takes days to solve the PB model on a protein with about 50,000 atoms at the mesh of 0.2 Å on a single CPU. Additionally, the information generated for the electrostatic analysis of a given biomolecule is not transferable to other proteins. Therefore, one has to carry out the separated electrostatic analysis of different proteins or the same protein with different protonation states or conformations. These issues call for innovative approaches, such as machine learning and dynamic programming, to biomolecular electrostatic analysis.

Recently we have witnessed the explosion of machine learning studies in science and engineering. In particular, deep neural networks which discover intricate structures in large datasets, have fueled the rapid growth in application such as computer vision, natural language processing, speech recognition, handwriting recognition, (24–26), etc. Machine learning has become an indispensable tool in the analysis and prediction of large and diverse molecular and biomolecular data sets, including bioactivity of small molecular drugs (27) and genomics (28). Studies in computational biology and biophysics, such as the predictions of solvation free energies, protein-ligand binding affinities, mutation impacts, toxicity, partition coefficients, B-factors etc. adopt machine learning approaches (29–32). These developments open the door for machine learning based electrostatic analysis.

The objective of the present work is to develop a machine learning solution of the PB equation for the electrostatic analysis of biomolecules. To this end, we first construct an accurate and efficient mathematical representation of electrostatic potential to effectively characterize the probability distribution of biomolecular electrostatics. Theoretically, the exact form of this distribution is not available even if one solves the PB equation for all possible biomolecular structures. However, in practice, this probability distribution can be sampled by using a PB solver, which provides machine learning training labels.

Our approach is based on a *representability* hypothesis and a *learning* hypothesis. The *representability* hypothesis states that the electrostatic potential of a biomolecule can be described by a set of partial charges and their geometric relations to the solvent. This hypothesis guilds the construction of the feature vector for the characterization of the probability distribution of biomolecular electrostatics. The *learning* hypothesis states that biomolecular electrostatics can be effectively represented by a feature vector as described by the representability hypothesis. When the probability distribution of biomolecular electrostatics is sufficiently sampled from a training set, a machine learning model can be established based on training labels and associated feature vectors to accurately predict the electrostatic potential of an unseen dataset which shares the same probability distribution with the training set.

The protocol described above calls for an accurate PB solver, which calculates machine learning labels and thus the probability distribution of molecular and biomolecular electrostatics. To this end, we apply the accurate MIBPB solver (22) at a refined mesh size 0.2 Å to generate solvation energy labels to minimize the numerical errors.

The representability hypothesis does not specify how to construct an accurate and efficient representation. An average biomolecule in the human body consists of about 6,000 atoms that lie in an 18,000-dimensional Euclidean space ($\mathbb{R}^{18,000}$). Such a high-dimensionality makes the first principle calculations intractable. Additionally, using macromolecular structures in deep convolutional neural networks (CNNs) is extremely expensive. For example, the 3D coordinates representation of a biomolecule with about 50 Å side length at a low resolution of 0.5 Å requires feature dimension of 100^{3n} , where n is the number of element types. The variable sizes of biomolecules also hinder the application of machine learning algorithms. These challenges motivate the development of scalable and intrinsically low-dimensional representations of biomolecular structures. Our hypothesis is that intrinsic physics lie in low-dimensional manifolds or spaces embedded in a high dimensional data space (30). Recently, a few low-dimensional representations for biomolecules have been developed in terms of algebraic topology (30, 33), differential geometry (34), and graph theory (35). All of these approaches can be used to represent biomolecular electrostatics. In this work, we adopt the graph theory representation due to its simplicity, in conjunction with a collection of features from the fast GB models to predict electrostatics from the PB model.

The rest of this paper is organized as follows. After this Introduction section, the Materials and Methods section describes models and algorithms used in the present work. We give a brief review of the PB model, the GB model, the graph theory, and machine learning algorithms used in developing the proposed Poisson-Boltzmann based machine learning (PBML) model for biomolecular electrostatic analysis. Simulation results and related discussion are presented in the Results section. Various convergence tests have been carried out to search for the most accurate PB solver to calculate ma-

chine learning labels. Our feature vectors are optimized with respect to from a few simple machine learning algorithms, namely, linear regression, random forest, and gradient boosted decision trees, to more complicated deep neural network (DNN). We demonstrate that the proposed PBML model is more accurate and reliable than commonly used PB solvers. This paper ends with a Conclusion section.

MATERIALS AND METHODS

In this section, we briefly review essential concepts and methods underpinning the proposed PBML model.

The Poisson-Boltzmann (PB) model

As shown in Fig. 1(a), the PB model governs electrostatics with the interior solute domain Ω_1 with fixed charges q_k located at atomic centers $\mathbf{r}_k$ for $k = 1, \dots, N_c$, and the exterior solvent domain Ω_2 with dissolved ions approximated by the Boltzmann distribution. These two domains are separated by the dielectric interface Γ . Among a variety of surface models, the most commonly used one is the solvent excluded surface or molecular surface (36). For simplicity, a linearized PB model is considered in the present work as

$$-\nabla \cdot \epsilon(\mathbf{r}) \nabla \phi(\mathbf{r}) + \bar{\kappa}^2(\mathbf{r}) \phi(\mathbf{r}) = \sum_{k=1}^{N_c} q_k \delta(\mathbf{r} - \mathbf{r}_k), \quad (1)$$

where $\phi(\mathbf{r})$ is the electrostatic potential and $\epsilon(\mathbf{r})$ are the dielectric constants given by

$$\epsilon(\mathbf{r}) = \begin{cases} \epsilon_1, & \mathbf{r} \in \Omega_1, \\ \epsilon_2, & \mathbf{r} \in \Omega_2, \end{cases} \quad (2)$$

and $\bar{\kappa}$ is the screening parameter with the relation $\bar{\kappa}^2 = \epsilon_2 \kappa^2$ where κ is the inverse Debye length measuring the ionic effective length. The PB model has the interface conditions on the molecular surface defined as

$$\phi_1(\mathbf{r}) = \phi_2(\mathbf{r}), \quad \epsilon_1 \frac{\partial \phi_1(\mathbf{r})}{\partial \mathbf{n}} = \epsilon_2 \frac{\partial \phi_2(\mathbf{r})}{\partial \mathbf{n}}, \quad \mathbf{r} \in \Gamma \quad (3)$$

where ϕ_1 and ϕ_2 are the limit values when approaching the interface from the inside or outside the solute domain, and $\mathbf{n}$ is the outward unit normal vector on Γ . The lack of appropriate or rigorous treatments of these interface conditions is the major error source for many existing PB solvers. The far-field boundary condition for the PB model is $\lim_{|\mathbf{r}| \rightarrow \infty} \phi(\mathbf{r}) = 0$, which is approximated using the Screened Coulombic potential.

The PB electrostatic solvation free energy is obtained by

$$\Delta G = \frac{1}{2} \sum_{k=1}^{N_c} q_k (\phi(\mathbf{r}_k) - \phi_0(\mathbf{r}_k)) \quad (4)$$

where $\phi_0(\mathbf{r}_k)$ is the solution of the PB equation as if there were no solvent-solute interface. Note that near the interface

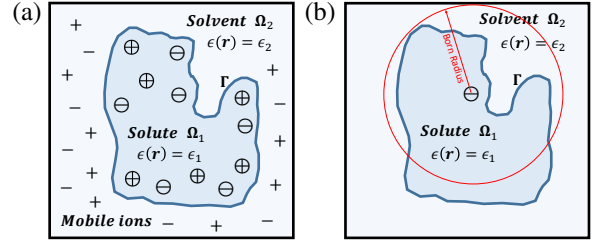


Figure 1: An illustration of the PB model and the GB model. (a): the PB model with solute region Ω_1 and solvent region Ω_2 , separated by molecular surface Γ , (b): the GB model represented by spherical cavities with Born radii and centered charges (one is shown here).

Γ , the interpolation of $\phi(\mathbf{r}_k)$ in Eq. (4) using ϕ at the grid points can be another major error source. An interface based scheme like MIB is required to interpolate $\phi(\mathbf{r}_k)$ (37).

The generalized Born (GB) model

With the same mathematical setting as that for the PB model, the Generalized Born (GB) model is devised to approximate the PB model. Compared with the PB model, the GB model offers a relatively simple and computationally more efficient approach to compute the long-range electrostatic interactions in biomolecules, which is the bottleneck in classical all-atom simulations. As illustrated in Fig. 1(b), the GB approximation of electrostatic solvation free energy can be expressed as the superposition of spherical cavities with effective Born radii and centered charges (only one is shown in the figure) (38) :

$$\begin{aligned} \Delta G^{\text{GB}} &\approx \sum_{ij} \Delta G_{ij}^{\text{GB}} \\ &= \frac{1}{2} \left(\frac{1}{\epsilon_2} - \frac{1}{\epsilon_1} \right) \frac{1}{1 + \alpha \beta} \sum_{ij} q_i q_j \left(\frac{1}{f_{ij}(r_{ij}, R_i, R_j)} + \frac{\alpha \beta}{A} \right) \end{aligned} \quad (5)$$

where r_{ij} the distance between atoms i and j , $\beta = \epsilon_1/\epsilon_2$, $\alpha = 0.571412$, A the electrostatic size of the molecule, using reciprocal of the Born radius $R_i^{-1} = \left(-\frac{1}{4\pi} \oint_{\Gamma} \frac{\mathbf{r} - \mathbf{r}_i}{|\mathbf{r} - \mathbf{r}_i|^6} \cdot d\mathbf{S} \right)^{1/3}$,

$$f_{ij} = \sqrt{r_{ij}^2 + R_i R_j \exp\left(-\frac{r_{ij}^2}{4R_i R_j}\right)}, \quad (6)$$

To carry out the boundary integral in evaluating R_i^{-1} , the MSMS package (39) is used for the triangulation of Γ . Note this integration is the most time-consuming step in the GB calculation. The ESES software (23) can be used to improve the current GB model if a higher level accuracy is desired.

The graph theory representation

Graph theory is a prime subject of discrete mathematics and concerns graphs as mathematical structures for modeling pairwise relations between vertices, nodes, or points.

Such pairwise relations define graph edges. Algebraic graph theory, particularly spectral graph theory, studies algebraic connectivity, characteristic polynomial, and eigenvalues and eigenvectors of matrices associated with the graph, such as adjacency matrix or Laplacian matrix. Graphs have been widely used in chemistry and biomolecular modeling (40). However, the diagonalization of the interaction Laplacian matrix has the computational complexity of $O(N^3)$ with N being the number of matrix elements. Alternatively, geometric graph theory bypasses the time-consuming matrix diagonalization and can be made of $O(N)$ in computational complexity (35).

In conjugation with machine learning algorithms, multiscale weighted colored subgraph (MWCS) was found to outperform many other methods in representing complex biomolecular structures (41, 42). We first consider weighted colored subgraph (WCS) to describe electrostatic interactions in a protein of N atoms. It incorporates kernels to characterize pairwise distance-weighted atomic correlations. All interactions are classified according to element types, leading to colored subgraphs. To use WCS for analyzing protein electrostatic interactions, we formulate all the atoms and their pairwise interactions into a weighted graph $G(V, E)$ with vertices V and edges E . As such, the i th atom is labeled by both its position $\mathbf{r}_i$ and element type α_i . Therefore, we express vertices V as

$$V = \{(\mathbf{r}_i, \alpha_i) | \mathbf{r}_i \in \mathbb{R}^3, \alpha_i \in C, i = 1, 2, \dots, N\}, \quad (7)$$

where $C = \{C, N, O, S, H\}$ contains all the commonly occurring element types in a protein. Obviously, for different biomolecular systems, we need to modify C accordingly.

To describe pairwise interactions between atoms in a protein, we define a colored set $\mathcal{P} = \{\alpha\beta\}$ with $\alpha, \beta \in C$. For each subset of element pairs $\mathcal{P}_k$, $k = 1, 2, \dots, 15$, a set of involved vertices $V_{\mathcal{P}_k}$ is a subset of V containing all atoms that belong to the pair in $\mathcal{P}_k$. For example, a partition $\mathcal{P}_2 = \{CN\}$ contains all pairs of atoms in the protein with one atom being a carbon and another atom being a nitrogen. Based on this setting, all the edges in such WCS describing pairwise atomic interactions are defined by

$$E_{\mathcal{P}_k}^{\sigma, \tau, \zeta} = \{\Phi_{\tau, \zeta}^{\sigma}(\|\mathbf{r}_i - \mathbf{r}_j\|) | \alpha_i \beta_j \in \mathcal{P}_k; i, j = 1, \dots, N\} \quad (8)$$

where $\|\mathbf{r}_i - \mathbf{r}_j\|$ defines a Euclidean distance between i^{th} and j^{th} atoms, σ indicates the type of radial basic functions (e.g., $\sigma = L$ for Lorentz kernel, $\sigma = E$ for exponential kernel), τ is a scale distance factor between two atoms, and ζ is a parameter of power in the kernel (i.e., $\zeta = \kappa$ when $\sigma = E$, $\zeta = \nu$ when $\sigma = L$). The kernel $\Phi_{\tau, \zeta}^{\sigma}$ characterizes a pairwise correlation satisfying the following conditions

$$\Phi_{\tau, \zeta}^{\sigma}(\|\mathbf{r}_i - \mathbf{r}_j\|) = \begin{cases} 0 & \text{as } \|\mathbf{r}_i - \mathbf{r}_j\| \rightarrow 0, \\ 1 & \text{as } \|\mathbf{r}_i - \mathbf{r}_j\| \rightarrow \infty. \end{cases} \quad (9)$$

Commonly used radial basis functions include generalized

exponential functions

$$\Phi_{\tau, \kappa}^E(\|\mathbf{r}_i - \mathbf{r}_j\|) = e^{-(\|\mathbf{r}_i - \mathbf{r}_j\|/\tau(r_i + r_j))^{\kappa}}, \quad \kappa > 0, \quad (10)$$

and generalized Lorentz functions

$$\Phi_{\tau, \nu}^L(\|\mathbf{r}_i - \mathbf{r}_j\|) = \frac{1}{1 + (\|\mathbf{r}_i - \mathbf{r}_j\|/\tau(r_i + r_j))^{\nu}}, \quad \nu > 0, \quad (11)$$

where r_i and r_j are, respectively, the van der Waals radius of the i^{th} and j^{th} atoms.

Centrality is widely used in graph theory or network analysis to describe node importance (43). Specifically, closeness and harmonic centralities are defined as $1/\sum_j \|\mathbf{r}_i - \mathbf{r}_j\|$ and $\sum_j 1/\|\mathbf{r}_i - \mathbf{r}_j\|$, respectively. The degree of centrality simply counts the number of edges upon a node. Our atomic centrality for i^{th} atom can be regarded as an extension of the harmonic formulation

$$\mu_i^{k, \sigma, \tau, \nu, w} = \sum_{j=1}^{|V_{\mathcal{P}_k}|} w_{ij} \Phi_{\tau, \nu}^{\sigma}(\|\mathbf{r}_i - \mathbf{r}_j\|), \quad \alpha_i \beta_j \in \mathcal{P}_k, \quad \forall i = 1, 2, \dots, |V_{\mathcal{P}_k}|, \quad (12)$$

where w_{ij} is a weight function assigned to each atomic pair, with $w_{ij} = 1$ for atomic rigidity or $w_{ij} = q_j$ for atomic charge.

In order to describe a centrality for the whole MWCS $G(V_{\mathcal{P}_k}, E_{\mathcal{P}_k}^{\sigma, \tau, \zeta})$, we take into account a summation of the atomic centralities

$$\mu^{k, \sigma, \tau, \nu, w} = \sum_{i=1}^{|V_{\mathcal{P}_k}|} \mu_i^{k, \sigma, \tau, \nu, w}. \quad (13)$$

It is this subgraph centrality that makes partition $\{CN\}$ equivalent to partition $\{NC\}$.

Since we have 15 choices of the set of weighted colored edges $\mathcal{P}_k$, we can obtain corresponding 15 subgraph centralities $\mu^{k, \sigma, \tau, \nu, w}$. By varying kernel parameters (σ, τ, ν, w), one can achieve multiscale centralities for multiscale weighted colored subgraph (MWCS) (41). For a two-scale WCS, we obtain a total of 60 descriptors for a protein.

Together with vertices V , the collection of all edges $E = \{E_{\mathcal{P}_k}^{\sigma, \tau, \zeta} | k = 1, 2, \dots, 15\}$ defines weighted graph $G(V, E)$. However, here $G(V, E)$ has a limited descriptive power in machine learning prediction. MWCSs $G(V_{\mathcal{P}_k}, E_{\mathcal{P}_k}^{\sigma, \tau, \zeta})$ and their centralities $\mu^{k, \sigma, \tau, \nu, w}$ are used in the present work to describe protein electrostatics.

Machine learning algorithms

General description

In the present work, the prediction of PB electrostatic solvation free energy is formulated as a standard supervised learning. The training data set $\mathcal{D}$ can be expressed as

$$\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)}) | \mathbf{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}, i = 1, \dots, M\},$$

where $\mathbf{x}^{(i)}$ is the feature vector for the i th sample in the training set, $y^{(i)} = \Delta G^{(i)}$ as label is the electrostatic solvation free energy of the i th sample, n and M are the sizes of feature vector and the training set. $\Delta G^{(i)}$ will be given by the accurate MIBPB solver, which is justified by a convergence analysis in the Result section. The feature vector will be generated from the graph theory and the GB model.

A variety of machine learning algorithms, including linear regression (LR), random forest (RF), gradient boosting decision tree (GBDT) and deep neural network (DNN) can be applied to predict the electrostatic free energy of the PB model. LR is a simple approach designed for the linear approximation of the mapping. RF and GBDT are both decision tree based ensemble methods. RF builds a large number of uncorrelated trees and utilizes bootstrap and aggregating (i.e., bagging). GBDT makes use of gradient descent in conjugation with the boosting procedure, which successively introduces weak learners to compensate for the errors of existing learners. DNN methods become powerful when errors are back-propagated to correct neural weights. However, DNN methods typically involve a large number of weights and thus are subject to overfitting. DNN methods might not offer better predictions unless the size of the training data is sufficiently large.

Feature Descriptions

Our ML model currently uses 367 features considering protein structures, force field, graph theory representation, etc.

- GB model related features (240) : For each of the 15 paired elements among {C,N,S,O,H}, there are 16 features as GB terms, absolute GB terms, intermediate GB terms, absolute intermediate GB terms, charges with dielectric, absolute charges with dielectric, GB charge terms, absolute GB charge terms, charges, absolute charges, plus 6 rigidity indices. We use the BornRadius code (2) developed in Onufriev's group wrapped with python scripts to generate these features.
- Protein features (51): area (7) (C, N, O, S, H, CNOS, CNOSH), charge (7), absolute charge (7), van der Waals force (15), Coulomb force (15).
- Environment features (31): volume (6), hydropathy (6), area (6), weight (6), pharma (6), sum of charge (1)
- Features from graph theory representation (45): Exp and Lor kernels (15 $\times$ 3 kernels)

In this work, we use MWCS centrality $\mu^{k,\sigma,\tau,\nu,w}$ to generate a mathematical representation of biomolecular electrostatics. In general, we seek an intrinsically low-dimensional representation of electrostatic solvation free energies for a large set of proteins. Specifically, we consider both the exponential kernel ($\sigma = E$) and the Lorentz kernel ($\sigma = L$). Each kernel is parametrized with both the weight selections of rigidity ($w = 1$) and charge ($w = q$). We select the powers

of the exponential kernel $\nu = 2$ and the Lorentz kernel $\nu = 3$. The atomic features with 15 types of element partitions ($\mathcal{P}_k$, $k = 1, 2, \dots, 15$) are considered. However, the scale distance factor τ is to be optimized during the training and testing processing. Additionally, the GB electrostatic solvation free energies is employed as a special feature that incorporated seamlessly with machine learning algorithms.

Generalized Born (GB) methods based gradient boosting decision tree (GBDT)

The main idea GBDT model is to first use the features and the labels to build a decision tree model, which is able to give predicted labels. Then the residue between the original labels and the predicted labels will be used as new labels, together with the original features to build another decision tree model, which gives another predicted labels. Using the difference between the initial labels and the predicted labels as the new labels, this procedure can be done recursively and final predicted labels will be the summation of all predicted labels. The model is optimized by minimizing the cost function between the initial labels and their predicted values using the gradient descent method.

In our framework of GB-based GBDT model, our first decision tree is the GB model, whose solvation energy is treated as the first predicted labels to initial labels, the PB-model based solvation energy. The rest of trees are built from our MWCS features. The loss function depends on a number of trees, the structure of trees and MWCS features. In this work, the loss function L will also be optimized with respect to MWCS parameters. The details of this model can be found in the supplementary material.

Generalized Born (GB) methods based deep neural network (DNN)

For DNN, we use the 367 features for each protein as the inputs to the network for training, test, and prediction purposes. In our model, the label is defined as $y^{(i)} = \Delta G_i^{\text{PB}} - \Delta G_i^{\text{GB}}$ for $i = 1, \dots, M$, where ΔG_i^{GB} is calculated as the core feature used outside the network. This core feature gives the global estimate while other features fed into the network provides local details. The quantity ΔG_i^{PB} is obtained as training/test data by solving the PB equation with MIBPB at refined mesh (e.g. $h=0.2$). In prediction, we have $\hat{\Delta G}_i^{\text{PB}} = \Delta G_i^{\text{GB}} + \hat{y}^{(i)}$, where $\hat{y}^{(i)}$ is the predicted value from the DNN. The DNN has multiple layers and the weights of the network are obtained by back-propagation. We tune the parameters of the network by sampling the parameter space to receive an optimized combination of the parameters for best prediction accuracy.

RESULTS

This section reports results from the proposed PBML model. Evaluation metrics, data selection, machine learning label calculation, and feature selection are discussed before results

are shown. We first justify the choice of MIBPB solver (17) to generate labels compared with popular PB solvers such as Amber (14) and DelPhi (15). We demonstrate the accuracy and efficiency of the proposed PBML model in electrostatic solvation free energy predictions. The numerical results of MIBPB, Amber and DelPhi are generated with an Intel Xeon E5-2670v2 from HPCC of Michigan State University (MSU), and the machine learning results are produced on a desktop with Intel Core i5 7500 and 16G Memory, using the scikit-learn python package. The electrostatic solvation free energies are generated with the room temperature $T = 298.15\text{K}$ and dielectric constant $\epsilon_1 = 1$, and $\epsilon_2 = 80$.

Evaluation metrics

Through this paper, we use the mean absolute percentage error (MAPE) and absolute relative error (ARE) for the analysis of prediction accuracy, which are defined as

$$E_{\text{MAPE}} = \frac{100\%}{M} \sum_{i=1}^M \left| \frac{y^{(i)} - \hat{y}^{(i)}}{y^{(i)}} \right|, E_{\text{ARE}} = \left| \frac{y^{(i)} - \hat{y}^{(i)}}{y^{(i)}} \right|.$$

where $y^{(i)}$ is the i th label, i.e., the PB electrostatic solvation free energy of the i th molecule and $\hat{y}^{(i)}$ is the predicted value.

Data preparation

In the present work, the selected 4294 protein structures are obtained from the PDBbind v2015 refined set and PDBbind v2018 refined set as the training set (44). The PDBbind v2015 core set of 195 proteins as listed in Supplementary Material is adopted as the test set. The training set has proteins sized from 997 to 27,713 atoms while the test set proteins range from 1,702 to 26,236 atoms.

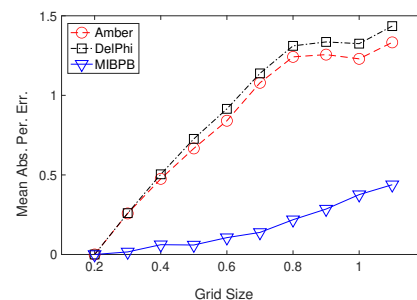
A data pre-processing is required before a PB solver can be called. The protein structures in the original data set are protein-ligand complexes. Missing atoms and side-chains are filled using the protein preparation wizard utility of the Schrodinger 2015-2 Suite with default parameter setting. The Amber ff14SB general force field is applied for the atomic van der Waals radii and partial charges.

Simulation Results

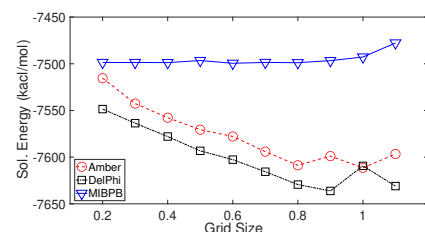
Convergence comparison of the PB solvers

We first carry out the convergence analysis of three PB solvers for the test set of 195 proteins to justify the use of MIBPB solver to produce accurate electrostatic solvation energy as the labels. For each protein, we compute their electrostatic solvation free energies at ten different mesh sizes, ranging from 0.2 Å to 1.1 Å. For each PB solver, its results at finest mesh size 0.2 Å are used as the references to evaluate the relative errors for other meshes. As shown in Fig. 2(a), the MAPEs using Amber and DelPhi are less than 1.5 %, but that from MIBPB is less than 0.5% at all mesh sizes.

(a) 195 proteins



(b) protein 3gnw



(c) protein 3owj

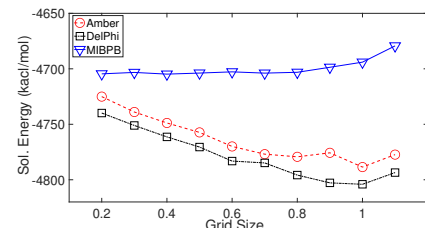


Figure 2: Convergence comparison among Amber, DelPhi, and MIBPB; (a) MAPEs at ten grid sizes for Amber, DelPhi, and MIBPB in computing the electrostatic solvation free energies of 195 test proteins. For a protein in each method, the reference value is computed at the mesh size of 0.2 Å; (b)-(c) Illustration of the electrostatic solvation free energies obtained by Amber, DelPhi, and MIBPB at ten different mesh sizes from 0.2 to 1.1 Å for proteins 3gnw and 3owj.

We next examine the electrostatic solvation free energies computed by three PB solvers on two sampled proteins. As shown in Fig. 2(b-c), using the test proteins 3gnw(b) and 3owj(c), the energies obtained by MIBPB do not change much over the mesh refinement, while those computed by Amber and DelPhi vary more significantly. We also observed that energies obtained by Amber and DelPhi converge toward those of MIBPB as the mesh is refined. These tests justify that MIBPB as the most accurate method among these three PB solvers, to be used to compute labels for the ML models. Some further convergence tests and comparison between PB solvers can be found in supplementary material.

Comparison between different ML models

After justifying the use of MIBPB solver to generate the labels, we next apply LR, RF, GBDT, and DNN to produce corresponding learned models using the training data set. We then use these learned models to predict the solvation energy for the 195 proteins in the test set. The MAPE for each learned model is shown in Table 1. The result shows that

DNN has better performance than other three methods, thus we use DNN as our ML algorithm for a further comprehensive training and test of the PBML model.

Table 1: The MAPEs of LR, RF, GBDT, DNN for the test set of 195 proteins. For LR and RF, we use the default parameters. For GBDT, we set the learning rate 0.05, the number of estimators 1500, maximum depth 5; The DNN is trained with 448 different combination of parameters and the final optimized choice has batch size 400, epoch 500, layers 367, 2048, 2048, 512, 512, 1.

	LR	RF	GBDT	DNN
MAPE	1.5317	0.8238	0.4432	0.3796

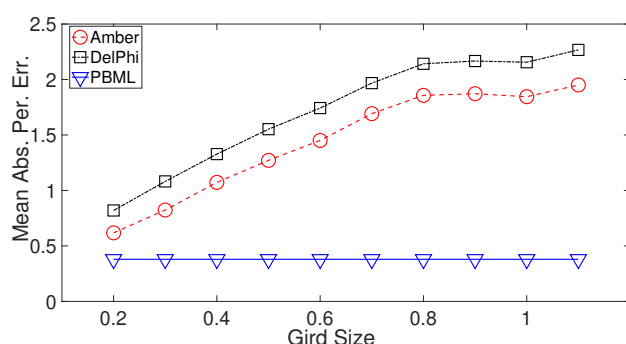


Figure 3: Comparison of the MAPEs of Amber, DelPhi and PBML (use result in Table 1 from DNN model) of the electrostatic solvation free energies of the test set at ten mesh sizes. The reference values are the results of MIBPB at the grid size of 0.2 Å.

Performance of the PBML model

Our final PBML model is essentially the GB-based DNN model, which uses the GB core feature with additional 367 features as described before. To understand the advantages of the model and its prediction, we plotted a comparison of the its MAPE with those of Amber and DelPhi at ten mesh sizes in Fig. 3. Note that although Amber and DelPhi MAPEs reduce significantly as the mesh is refined, even at the finest mesh of 0.2 Å, these two methods have not reached the accuracy of PBML, which does not depend on grid size once the model is trained/learned.

To further check the accuracy and efficiency of our PBML model. We compute the solvation energy of 195 test proteins using both the MIBPB at $h = 0.5$ and the PBML model. Note the PBML model is trained with the 4000+ protein training set as described before labeled by solvation energy computed using MIBPB at $h = 0.2$. For this test, we use results from MIBPB at $h = 0.2$ as benchmark values while use results from MIBPB at $h = 0.5$ for the comparison with results from the PBML model for the 195 proteins in the test set. Fig. 4(a) shows the relative error in solvation energy and

from individual sample or average we see the PBML model is obviously more accurate than the MIBPB model at $h = 0.5$. Fig. 4(b) shows the elapsed time and from individual sample or average we see the PBML model is significantly more efficient than the MIBPB model at $h = 0.5$. All figures are plotted using log scale in error and time since results from different proteins are very variant.

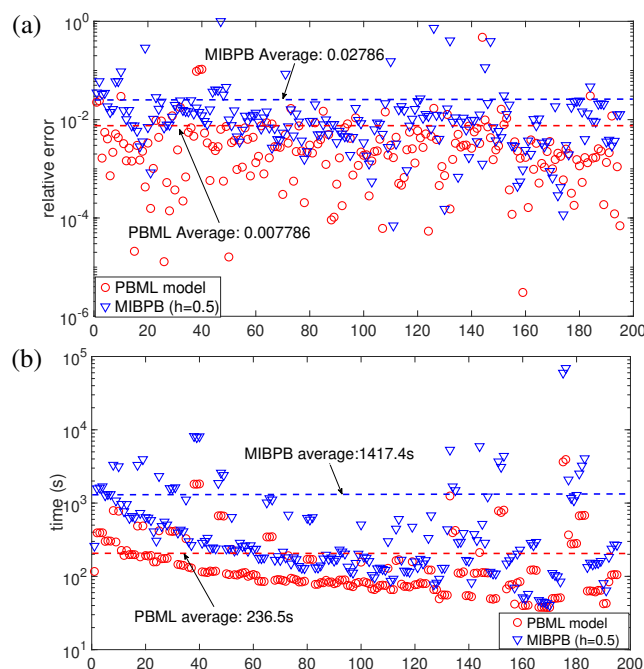


Figure 4: Accuracy and efficiency comparison on computing solvation energy on 195 proteins using MIBPB at $h = 0.5$ and DNN based PBML model:(a): relative error in solvation energy; (b): time. The average relative errors for PBML and MIBPB are 0.00776 and 0.02786. The average time for PBML and MIBPB is 236.5s and 1417.4s respectively.

CONCLUSION

This work introduces the Poisson-Boltzmann based machine learning (PBML) model for the prediction of electrostatic solvation free energies of biomolecules. Our goal is to offer an efficient ML-based electrostatic analysis of new molecules or new conformations of molecular dynamics at a small fraction of time used in solving the Poisson-Boltzmann (PB) equation at a similar level of accuracy, or at a similar level of computational time but with a much higher accuracy than a commonly used PB solver can ever deliver. To this end, we first search the most accurate PB solver for generating ML labels. The second-order accurate MIBPB solver turns out to converge faster than two other eminent PB solvers, namely the DelPhi and the Amber PB solver. Additionally, we adopt multiscale weighted colored subgraph (MWCS) for ML feature generations, which produces excellent low-dimensional intrinsic representations of biomolecules. One

global core feature is computed from the generalized Born (GB) model. To maintain the efficiency, we employ a few ML algorithms, including linear regression (LR), random forest (RF), gradient boosting decision tree (GBDT), and Deep Neural Network (DNN). It is found that the present PBML model using DNN can more efficiently and accurately produce electrostatics over traditional grid-based PB solvers.

AUTHOR CONTRIBUTIONS

JC, CZ, WG, and GWW designed the research. JC, YX, and XY carried out all simulations, analyzed the data. JC, WG, and GWW wrote the article.

ACKNOWLEDGMENTS

The work of WG is supported in part by NSF grant DMS-2110922. The work GWW is supported in part by NSF grants DMS-2052983 and IIS-1900473 and NIH grants R01AI164266, and R35GM148196.

LITERATURE CITED

- Honig, B., and A. Nicholls, 1995. Classical electrostatics in biology and chemistry. *Science* 268:1144–9.
- Onufriev, A., D. Bashford, and D. A. Case, 2000. Modification of the generalized Born model suitable for macromolecules. *Journal of Physical Chemistry B* 104:3712–3720.
- Tomasi, J., B. Mennucci, and R. Cammi, 2005. Quantum mechanical continuum solvation models. *Chem. Rev.* 105:2999–3093.
- Fogolari, F., A. Brigo, and H. Molinari, 2002. The Poisson-Boltzmann equation for biomolecular electrostatics: a tool for structural biology. *Journal of Molecular Recognition* 15:377–92.
- Bashford, D., and M. Karplus, 1990. pK_a 's of ionizable groups in proteins: atomic detail from a continuum electrostatic model. *Biochemistry* 29:10219–25.
- Onufriev, A. V., and E. Alexov, 2013. Protonation and pK changes in protein-ligand binding. *Quarterly reviews of biophysics* 46(02):181–219.
- Tang, C. L., E. Alexov, A. M. Pyle, and B. Honig, 2007. Calculation of pK s in RNA: On the Structural Origins and Functional Roles of Protonated Nucleotides. *Journal of Molecular Biology* 366:1475–96.
- Zhang, Q., D. A. Beard, and T. Schlick, 2003. Constructing irregular surfaces to enclose macromolecular complexes for mesoscale modeling using the discrete surface charge optimization (DISCO) algorithm. *J. Comput. Chem.* 24:2063–2074.
- Madura, J. D., J. M. Briggs, R. C. Wade, M. E. Davis, B. A. Luty, A. Il'in, J. Antosiewicz, M. K. Gilson, B. Bagheri, L. R. Scott, and J. A. McCammon, 1995. Electrostatics and diffusion of molecules in solution - simulations with the University of Houston Brownian Dynamics program. *Computer Physics Communications* 91:57–95.
- Jo, S., M. Vargyas, J. Vasko-Szedlar, B. Roux, and W. Im, 2008. PBEQ-Solver for online visualization of electrostatic potential of biomolecules. *Nucleic Acids Research* 36:W270 –W275.
- Baker, N. A., D. Sept, M. J. Holst, and J. A. Mccammon, 2001. The adaptive multilevel finite element solution of the Poisson-Boltzmann equation on massively parallel computers. *IBM Journal of Research and Development* 45:427–438.
- Geng, W. H., and R. Krasny, 2013. A treecode-accelerated boundary integral Poisson-Boltzmann solver for continuum electrostatics of solvated biomolecules. *J. Comput. Phys.* 247:62–87.
- Lu, B., X. Cheng, J. Huang, and J. A. McCammon, 2013. AFMPB: An Adaptive Fast Multipole Poisson-Boltzmann Solver for Calculating Electrostatics in Biomolecular Systems. *Comput. Phys. Commun.* 184:2618–2619.
- Cai, Q., M. J. Hsieh, J. Wang, and R. Luo, 2009. Performance of Nonlinear Finite-Difference Poisson-Boltzmann Solvers. *Journal of Chemical Theory and Computation* 6(1):203–211.
- Li, L., C. Li, S. Sarkar, J. Zhang, S. Witham, Z. Zhang, L. Wang, N. Smith, M. Petukh, and E. Alexov, 2012. Delphi: a comprehensive suite for Delphi software and associated resources. *BMC Biophysics* 5:9:2046–1682.
- Baker, N. A., D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon, 2001. Electrostatics of nanosystems: Application to microtubules and the ribosome. *PNAS* 98:10037–10041.
- Chen, D., Z. Chen, C. Chen, W. H. Geng, and G. W. Wei, 2011. MIBPB: A software package for electrostatic analysis. *J. Comput. Chem.* 32:657 – 670.
- Geng, W., and S. Zhao, 2017. A two-component Matched Interface and Boundary (MIB) regularization for charge singularity in implicit solvation. *Journal of Computational Physics* 351:25–39.
- Peskin, C. S., 1977. Numerical analysis of blood flow in the heart. *J. Comput. Phys.* 25:220–52.
- LeVeque, R. J., and Z. L. Li, 1994. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.* 31:1019–1044.

21. Yu, S. N., Y. C. Zhou, and G. W. Wei, 2007. Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces. *J. Comput. Phys.* 224:729–756.
22. Geng, W., S. Yu, and G. W. Wei, 2007. Treatment of charge singularities in implicit solvent models. *Journal of Chemical Physics* 127:114106.
23. Liu, B., B. Wang, R. Zhao, Y. Tong, and G. W. Wei, 2017. ESES: software for Eulerian solvent excluded surface. *Journal of Computational Chemistry* 38:446–466.
24. LeCun, Y., Y. Bengio, and G. Hinton, 2015. Deep learning. *Nature* 521:436–444.
25. Korotcov, A., V. Tkachenko, D. P. Russo, and S. Ekins, 2017. Comparison of deep learning with multiple machine learning methods and metrics using diverse drug discovery data sets. *Molecular pharmaceutics* 14:4462–4475.
26. Jiménez, J., M. Skalic, G. Martínez-Rosell, and G. De Fabritiis, 2018. K DEEP: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of chemical information and modeling* 58:287–296.
27. Hughes, T. B., G. P. Miller, and S. J. Swamidass, 2015. Modeling epoxidation of drug-like molecules with a deep machine learning network. *ACS Cent. Sci.* 1:168–180.
28. Lam, P. C.-H., R. Abagyan, and M. Totrov, 2018. Hybrid receptor structure/ligand-based docking and activity prediction in ICM: development and evaluation in D3R Grand Challenge 3. *Journal of computer-aided molecular design* 1–12.
29. Sunseri, J., M. Ragoza, J. Collins, and D. R. Koes, 2016. A D3R prospective evaluation of machine learning for protein-ligand scoring. *Journal of computer-aided molecular design* 30:761–771.
30. Cang, Z. X., L. Mu, and G. W. Wei, 2018. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLOS Computational Biology* 14(1).
31. Wu, K., and G. W. Wei, 2018. Quantitative Toxicity Prediction Using Topology Based Multitask Deep Neural Networks. *Journal of Chemical Information and Modeling* 58:520–531.
32. Wu, K., Z. Zhao, R. Wang, and G. W. Wei, 2018. TopP-S: Persistent Homology-Based Multi-Task Deep Neural Networks for Simultaneous Predictions of Partition Coefficient and Aqueous Solubility. *Journal of Computational Chemistry* 39:1444–1454.
33. Xia, K. L., and G. W. Wei, 2014. Persistent homology analysis of protein structure, flexibility and folding. *International Journal for Numerical Methods in Biomedical Engineering* 30:814–844.
34. Nguyen, D. D., and G. W. Wei, 2017. The impact of surface area, volume, curvature and Lennard-Jones potential to solvation modeling. *Journal of Computational Chemistry* 38:24–36.
35. Xia, K. L., K. Opron, and G. W. Wei, 2013. Multiscale multiphysics and multidomain models — Flexibility and Rigidity. *Journal of Chemical Physics* 139:194109.
36. Cannolly, M. L., 1983. Solvent-Accessible Surfaces of Proteins and Nucleic Acids. *Science* 221:709–713.
37. Nguyen, D. D., B. Wang, and G. W. Wei, 2017. Accurate, robust and reliable calculations of Poisson-Boltzmann binding energies. *Journal of Computational Chemistry* 38:941–948.
38. Forouzesh, N., S. Izadi, and A. V. Onufriev, 2017. Grid-based surface generalized Born model for calculation of electrostatic binding free energies. *Journal of chemical information and modeling* 57:2505–2513.
39. Sanner, M. F., A. J. Olson, and J. C. Spehner, 1996. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers* 38:305–320.
40. Bahar, I., A. R. Atilgan, and B. Erman, 1997. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design* 2:173 – 181.
41. Bramer, D., and G.-W. Wei, 2018. Multiscale weighted colored graphs for protein flexibility and rigidity analysis. *The Journal of chemical physics* 148:054103.
42. Nguyen, D. D., Z. Cang, K. Wu, M. Wang, Y. Cao, and G.-W. Wei, 2019. Mathematical deep learning for pose and binding affinity prediction and ranking in D3R Grand Challenges. *Journal of Computer Aided Molecular Design* 33:71–82.
43. Borgatti, S. P., 2005. Centrality and network flow. *Social networks* 27:55–71.
44. Liu, Z., Y. Li, L. Han, J. Liu, Z. Zhao, W. Nie, Y. Liu, and R. Wang, 2015. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics* 31:405–412.

SUPPLEMENTARY MATERIAL

An online supplement to this article can be found by visiting BJ Online at <http://www.biophysj.org>.

Generalize Born (GB) methods based gradient boosting decision tree (GBDT)

It is noted that GB model provides a good approximation to PB model. As such, its ΔG^{GB} can be incorporated into the GBDT algorithm. To this end, we outline the GB method based GBDT algorithm as follows. Normally, the first step is to build a decision tree T_1 to fit $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^M$, leading to predicted labels $\{p_1(\mathbf{x}^{(i)})\}_{i=1}^M$. The errors (or residues) of the predictions are $r_2^{(i)} = y^{(i)} - p_1(\mathbf{x}^{(i)})$. If $r_2^{(i)} \neq 0$ for some i , one builds another decision tree T_2 to fit $\{(\mathbf{x}^{(i)}, r_2^{(i)})\}_{i=1}^M$, leading to new predicted labels $\{p_2(\mathbf{x}^{(i)})\}_{i=1}^M$. The errors are $r_3^{(i)} = r_2^{(i)} - p_2(\mathbf{x}^{(i)}) = y^{(i)} - p_1(\mathbf{x}^{(i)}) - p_2(\mathbf{x}^{(i)})$. The predicted labels for $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^M$ are $\{p_1(\mathbf{x}^{(i)}) + p_2(\mathbf{x}^{(i)})\}_{i=1}^M$. If $|r_3^{(i)}| > 0$, one can build T_3 to fit $\{(\mathbf{x}^{(i)}, r_3^{(i)})\}_{i=1}^M$, leading to new predicted labels $\{p_3(\mathbf{x}^{(i)})\}_{i=1}^M$. In general, the predicted model for the data $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^M$ based on K consecutive decision trees is

$$\hat{y}_K^{(i)} = \sum_{k=1}^K p_k(\mathbf{x}^{(i)}), i = 1, 2, \dots, M. \quad (14)$$

This is the so called boosting tree procedure. One can setup a loss function

$$L_k = \sum_{i=1}^M l_k(y^{(i)} - \hat{y}_k^{(i)}) = \sum_{i=1}^M \frac{1}{2} (y^{(i)} - \hat{y}_k^{(i)})^2, \quad (15)$$

to minimize the loss via the gradient descent optimization of decision trees. A general minimization procedure is the follows. For $k = 2$ to K ,

- Calculate gradient

$$r_k^{(i)} = -\frac{\partial l_{k-1}}{\partial p_{k-1}(\mathbf{x}^{(i)})} \quad (16)$$

- Construct decision tree T_k to fit $\{(\mathbf{x}^{(i)}, r_k^{(i)})\}_{i=1}^M$, leading to p_k as the learner function of T_k .
- Choose learning rate α_k such that

$$\alpha_k := \operatorname{argmin}_{\alpha} \sum_{i=1}^M l_{k-1}(y^{(i)} - \hat{y}_{k-1}^{(i)}) + \alpha p_k(\mathbf{x}^{(i)}) \quad (17)$$

- Update model

$$\hat{y}_k^{(i)} = \hat{y}_{k-1}^{(i)} + \eta \alpha_k p_k, \quad (18)$$

where η is the shrinkage, a predefined parameter.

In our work, we select “ T_1 ” to be the GB model which leads to $\{p_1(\mathbf{x}^{(i)}) = \Delta G_i^{\text{GB}}\}_{i=1}^M$. The rest of trees are built from our MWCS features. The loss function depends on a number of trees, the structure of trees and MWCS features. In this work, the loss function L will also be optimized with respect to MWCS parameters.

Convergence comparison between Amber, Delphi, and MIBPB on 30 selected proteins

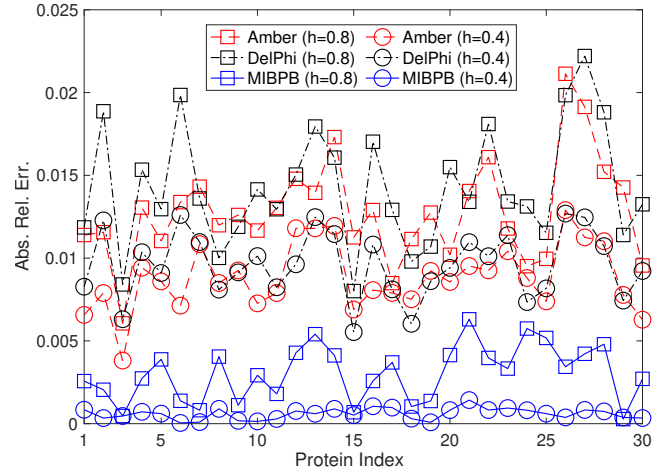


Figure 5: Convergence comparison among Amber, DelPhi, and MIBPB. The graph shows Absolute relative errors of Amber (dashed lines), DelPhi (dash-dot lines), and MIBPB (solid lines) at mesh sizes 0.4 Å (circles) and 0.8 Å (squares) for 30 proteins. For a protein in each method, the reference value is calculated at the mesh size of 0.2 Å.

To further investigate the convergences of the three PB solvers, we randomly picked 30 proteins from the core set and plot their AREs at two mesh sizes 0.4 Å (circle) and 0.8 Å (square) compared with results at 0.2 Å in Fig. 2(b). We see that Amber and DelPhi offer a similar level of convergence, while MIBPB returns significantly smaller errors. These 30 proteins are: 2qmq, 3ebp, 2x8z, 3bkk, 1gpk, 1f8b, 3ge7, 3huc, 3pe2, 3mfv, 2qbr, 3imc, 1saq, 3ivg, 3b3w, 3mss, 2v7a, 2zcq, 3utu, 3u9q, 3kwa, 3gbb, 1uto, 2yge, 2iwx, 3bpc, 2pcp, 2gss, 4dew, 3nox.

The list of 195 proteins from testing set

Below is the list of the 195 proteins used for testing purpose. 1lor, 1ps3, 3d4z, 3ejr, 2qmq, 3l4w, 3l4u, 3l7b, 3g2n, 3ebp, 2w66, 2wca, 2vvn, 2x97, 2xhm, 2x8z, 2x0y, 2cbj, 2j62, 3bkk, 3l3n, 2xy9, 1gpk, 1h23, 1e66, 3cj2, 2d3u, 3gnw, 3f3a, 3f3c, 3f3e, 4gqq, 1u33, 1xd0, 2wbg, 2j78, 2cet, 2zxd, 2zwz, 2zx6, 3udh, 4djv, 4gid, 3fk1, 2qft, 2pq9, 1f8d, 1f8b, 1f8c, 1n2v, 1r5y, 3ge7, 3huc, 3gcs, 3e93, 1q8t, 1q8u, 3ag9, 3owj, 2zjw, 3pe2, 2v00, 3pww, 3uri, 3mfv, 3f80, 3kv2, 2hb1, 2qbr, 2qbp, 3fcq, 1os0, 4tmn, 3pxf, 2xnb, 2fvd, 1qi0, 1w3k, 1w3l, 3imc,

3ivg, 3coy, 3b3s, 3b3w, 3vh9, 3mss, 3k5v, 2v7a, 2brb, 3jvs, 1nvq, 3acw, 2zcr, 2zcq, 1bcu, 1oyt, 3utu, 3u9q, 2yfe, 2p4y, 3uo4, 2wtv, 3myg, 3kgp, 1o5b, 1sqa, 3kwa, 2weg, 3dd0, 2xdl, 1yc1, 2yki, 1p1q, 3bfu, 4g8m, 3g2z, 4de2, 4de1, 1vso, 3gbb, 3fv1, 2y5h, 2xbv, 1mq6, 1loq, 1lol, 1uto, 3gy4, 1o3f, 2yge, 2iwx, 2vw5, 2ymd, 2xys, 2x00, 2r23, 3bpc, 1kel, 3ozt, 3oe5, 3nw9, 1zea, 2pcp, 1igj, 1lbk, 2gss, 10gs, 3su5, 3su2, 3su3, 3n7a, 3n86, 2xb8, 3ao4, 3zsx, 3zso, 3nq3, 3ueu, 3uex, 3lka, 3ehy, 3f17, 3cft, 4des, 4dew, 3dxg, 1w4o, 1u1b, 3ov1, 3s8o, 1jyq, 1a30, 3cyx, 4djr, 3i3b, 3muz, 3vd4, 2vo5, 2vl4, 2vot, 1n1m, 2ole, 3nox, 1hnn, 2g70, 2obf, 1z95, 3b68, 3g0w, 1sln, 2d1o, 1hfs, 2jdy, 2jdm, 2jdu.