Chemical Science

EDGE ARTICLE

View Article Online
View Journal | View Issue

Cite this: *Chem. Sci.*, 2018, **9**, 513

MoleculeNet: a benchmark for molecular machine learning†

Zhenqin Wu, ‡[a] Bharath Ramsundar,‡[b] Evan N. Feinberg,§[c] Joseph Gomes, §[a] Caleb Geniesse,[c] Aneesh S. Pappu,[b] Karl Leswing[d] and Vijay Pande*[a]

Molecular machine learning has been maturing rapidly over the last few years. Improved methods and the presence of larger datasets have enabled machine learning algorithms to make increasingly accurate predictions about molecular properties. However, algorithmic progress has been limited due to the lack of a standard benchmark to compare the efficacy of proposed methods; most new algorithms are benchmarked on different datasets making it challenging to gauge the quality of proposed methods. This work introduces MoleculeNet, a large scale benchmark for molecular machine learning. MoleculeNet curates multiple public datasets, establishes metrics for evaluation, and offers high quality open-source implementations of multiple previously proposed molecular featurization and learning algorithms (released as part of the DeepChem open source library). MoleculeNet benchmarks demonstrate that learnable representations are powerful tools for molecular machine learning and broadly offer the best performance. However, this result comes with caveats. Learnable representations still struggle to deal with complex tasks under data scarcity and highly imbalanced classification. For quantum mechanical and biophysical datasets, the use of physics-aware featurizations can be more important than choice of particular learning algorithm.

Received 15th June 2017
Accepted 30th October 2017

DOI: 10.1039/c7sc02664a

rsc.li/chemical-science

1 Introduction

Overlap between chemistry and statistical learning has had a long history. The field of cheminformatics has been utilizing machine learning methods in chemical modeling (*e.g.* quantitative structure activity relationships, QSAR) for decades.[1–6] In the recent 10 years, with the advent of sophisticated deep learning methods,[7,8] machine learning has gathered increasing amounts of attention from the scientific community. Data-driven analysis has become a routine step in many chemical and biological applications, including virtual screening,[9–12] chemical property prediction,[13–16] and quantum chemistry calculations.[17–20]

In many such applications, machine learning has shown strong potential to compete with or even outperform conventional *ab initio* computations.[16,18] It follows that introduction of novel machine learning methods has the potential to reshape research on properties of molecules. However, this potential has been limited by the lack of a standard evaluation platform for proposed machine learning algorithms. Algorithmic papers often benchmark proposed methods on disjoint dataset collections, making it a challenge to gauge whether a proposed technique does in fact improve performance.

Data for molecule-based machine learning tasks are highly heterogeneous and expensive to gather. Obtaining precise and accurate results for chemical properties typically requires specialized instruments as well as expert supervision (contrast with computer speech and vision, where lightly trained workers can annotate data suitable for machine learning systems). As a result, molecular datasets are usually much smaller than those available for other machine learning tasks. Furthermore, the breadth of chemical research means our interests with respect to a molecule may range from quantum characteristics to measured impacts on the human body. Molecular machine learning methods have to be capable of learning to predict this very broad range of properties. Complicating this challenge, input molecules can have arbitrary size and components, highly variable connectivity and many three dimensional conformers (three dimensional molecular shapes). To transform molecules into a form suitable for conventional machine learning algorithms (that usually accept fixed length input), we have to extract useful and related information from a molecule into a fixed dimensional representation (a process called featurization).[21–23]

To put it simply, building machine learning models on molecules requires overcoming several key issues: limited

[a]*Department of Chemistry, Stanford University, Stanford, CA 94305, USA. E-mail: pande@stanford.edu*
[b]*Department of Computer Science, Stanford University, Stanford, CA 94305, USA*
[c]*Program in Biophysics, Stanford School of Medicine, Stanford, CA 94305, USA*
[d]*Schrodinger Inc., USA*

† Electronic supplementary information (ESI) available. See DOI: 10.1039/c7sc02664a

‡ Joint first authorship.

§ Joint second authorship.

amounts of data, wide ranges of outputs to predict, large heterogeneity in input molecular structures and appropriate learning algorithms. Therefore, this work aims to facilitate the development of molecular machine learning methods by curating a number of dataset collections, creating a suite of software that implements many known featurizations of molecules, and providing high quality implementations of many previously proposed algorithms. Following the footsteps of WordNet[24] and ImageNet,[25] we call our suite MoleculeNet, a benchmark collection for molecular machine learning.

In machine learning, a benchmark serves as more than a simple collection of data and methods. The introduction of the ImageNet benchmark in 2009 has triggered a series of breakthroughs in computer vision, and in particular has facilitated the rapid development of deep convolutional networks. The ILSVRC, an annual contest held by the ImageNet team,[26] draws considerable attention from the community, and greatly stimulates collaborations and competitions across the field. The contest has given rise to a series of prominent machine learning models such as AlexNet,[27] GoogLeNet,[28] ResNet[29] which have had broad impact on the academic and industrial computer science communities. We hope that MoleculeNet will trigger similar breakthroughs by serving as a platform for the wider community to develop and improve models for learning molecular properties.

In particular, MoleculeNet contains data on the properties of over 700 000 compounds. All datasets have been curated and integrated into the open source DeepChem package.[30] Users of DeepChem can easily load all MoleculeNet benchmark data through provided library calls. MoleculeNet also contributes high quality implementations of well known (bio) chemical featurization methods. To facilitate comparison and development of new methods, we also provide high quality implementations of several previously proposed machine learning methods. Our implementations are integrated with DeepChem, and depend on Scikit-Learn[31] and Tensorflow[32] underneath the hood. Finally, evaluation of machine learning algorithms requires defined methods to split datasets into training/validation/test collections. Random splitting, common in machine learning, is often not correct for chemical data.[33] MoleculeNet contributes a library of splitting mechanisms to DeepChem and evaluates all algorithms with multiple choices of data split. MoleculeNet provide a series of benchmark results of implemented machine learning algorithms using various featurizations and splits upon our dataset collections. These results are provided within this paper, and will be maintained online in an ongoing fashion as part of DeepChem.

The related work section will review prior work in the chemistry community on gathering curated datasets and discuss how MoleculeNet differs from these previous efforts. The methods section reviews the dataset collections, metrics, featurization methods, and machine learning models included as part of MoleculeNet. The results section will analyze the benchmarking results to draw conclusions about the algorithms and datasets considered.

2 Related work

MoleculeNet draws upon a broader movement within the chemical community to gather large sources of curated data. PubChem[34] and PubChem BioAssay[35] gather together thousands of bioassay results, along with millions of unique molecules tested within these assays. The ChEMBL database offers a similar service, with millions of bioactivity outcomes across thousands of protein targets. Both PubChem and ChEMBL are human researcher oriented, with web portals that facilitate browsing of the available targets and compounds. ChemSpider is a repository of nearly 60 million chemical structures, with web based search capabilities for users. The Crystallography Open Database[36] and Cambridge Structural Database[37] offer large repositories of organic and inorganic compounds. The protein data bank[38] offers a repository of experimentally resolved three dimensional protein structures. This listing is by no means comprehensive; the methods section will discuss a number of smaller data sources in greater detail.

These past efforts have been critical in enabling the growth of computational chemistry. However, these previous databases are not machine-learning focused. In particular, these collections don't define metrics which measure the effectiveness of algorithmic methods in understanding the data contained. Furthermore, there is no prescribed separation of the data into training/validation/test sets (critical for machine learning development). Without specified metrics or splits, the choice is left to individual researchers, and there are indeed many chemical machine learning papers which use subsets of these data stores for machine learning evaluation. Unfortunately, the choice of metric and subset varies widely between groups, so two methods papers using PubChem data may be entirely incomparable. MoleculeNet aims to bridge this gap by providing benchmark results for a reasonable range of metrics, splits, and subsets of these (and other) data collections.

It's important to note that there have been some efforts to create benchmarking datasets for machine learning in chemistry. The Quantum Machine group[39] and previous work on multitask learning[10] both introduce benchmarking collections which have been used in multiple papers. MoleculeNet incorporates data from both these efforts and significantly expands upon them.

3 Methods

MoleculeNet is based on the open source package DeepChem.[30] Fig. 1 shows an annotated DeepChem benchmark script. Note how different choices for data splitting, featurization, and model are available. DeepChem also directly provides molnet sub-module to support benchmarking. The single line below runs benchmarking on the specified dataset, model and featurizer. User defined models capable of handling DeepChem datasets are also supported.

```
deepchem.molnet.run_benchmark (datasets, model, split,
featurizer)
```


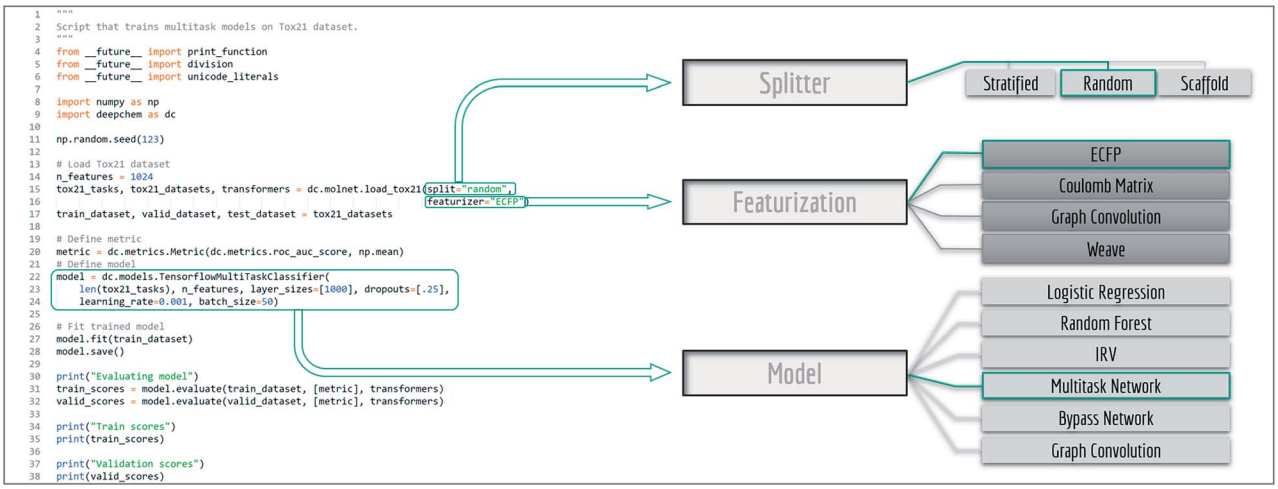

Fig. 1 Example code for benchmark evaluation with DeepChem, multiple methods are provided for data splitting, featurization and learning.

In this section, we will further elaborate the benchmarking system, introducing available datasets as well as implemented splitting, metrics, featurization, and learning methods.

3.1 Datasets

MoleculeNet is built upon multiple public databases. The full collection currently includes over 700 000 compounds tested on a range of different properties. These properties can be subdivided into four categories: quantum mechanics, physical chemistry, biophysics and physiology. As illustrated in Fig. 2, separate datasets in the MoleculeNet collection cover various levels of molecular properties, ranging from molecular-level properties to macroscopic influences on human body. For each dataset, we propose a metric and a splitting pattern (introduced in the following texts) that best fit the properties of the dataset. Performances on the recommended metric and split are reported in the results section.

In most datasets, SMILES strings[40] are used to represent input molecules, 3D coordinates are also included in part of the collection as molecular features, which enables different methods to be applied. Properties, or output labels, are either 0/1 for classification tasks, or floating point numbers for regression tasks. At the time of writing, MoleculeNet contains 17 datasets prepared and benchmarked, but we anticipate adding further datasets in an on-going fashion. We also highly welcome contributions from other public data collections. For more detailed dataset structure requirements and instructions on curating datasets, please refer to the tutorial notebook in the example folder of DeepChem github repository.

Table 1 lists details of datasets in the collection, including tasks, compounds and their features, recommended splits and metrics. Contents of each dataset will be elaborated in this subsection, function calls to access the datasets can be found in the ESI.†

3.1.1 QM7/QM7b. The QM7/QM7b datasets are subsets of the GDB-13 database,[41] a database of nearly 1 billion stable and synthetically accessible organic molecules, containing up to seven "heavy" atoms (C, N, O, S). The 3D Cartesian coordinates of the most stable conformation and electronic properties (atomization energy, HOMO/LUMO eigenvalues, *etc.*) of each molecule were determined using *ab initio* density functional theory (PBE0/tier2 basis set).[17,18] Learning methods

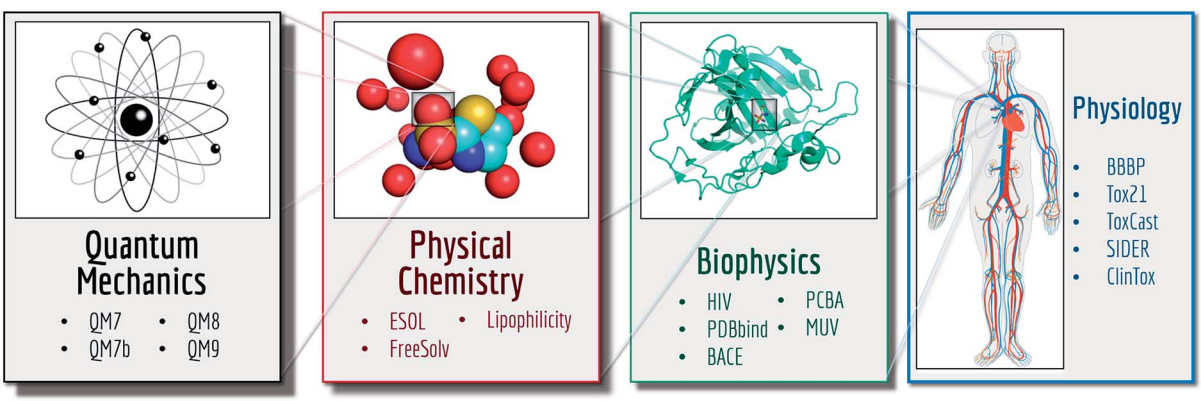

Fig. 2 Tasks in different datasets focus on different levels of properties of molecules.

Table 1 Dataset details: number of compounds and tasks, recommended splits and metrics

Category	Dataset	Data type	Tasks		Compounds	Rec – split	Rec – metric
Quantum mechanics	QM7	SMILES, 3D coordinates	1	Regression	7165	Stratified	MAE
	QM7b	3D coordinates	14	Regression	7211	Random	MAE
	QM8	SMILES, 3D coordinates	12	Regression	21 786	Random	MAE
	QM9	SMILES, 3D coordinates	12	Regression	133 885	Random	MAE
Physical chemistry	ESOL	SMILES	1	Regression	1128	Random	RMSE
	FreeSolv	SMILES	1	Regression	643	Random	RMSE
	Lipophilicity	SMILES	1	Regression	4200	Random	RMSE
Biophysics	PCBA	SMILES	128	Classification	439 863	Random	PRC-AUC
	MUV	SMILES	17	Classification	93 127	Random	PRC-AUC
	HIV	SMILES	1	Classification	41 913	Scaffold	ROC-AUC
	PDBbind	SMILES, 3D coordinates	1	Regression	11 908	Time	RMSE
	BACE	SMILES	1	Classification	1522	Scaffold	ROC-AUC
Physiology	BBBP	SMILES	1	Classification	2053	Scaffold	ROC-AUC
	Tox21	SMILES	12	Classification	8014	Random	ROC-AUC
	ToxCast	SMILES	617	Classification	8615	Random	ROC-AUC
	SIDER	SMILES	27	Classification	1427	Random	ROC-AUC
	ClinTox	SMILES	2	Classification	1491	Random	ROC-AUC

benchmarked on QM7/QM7b are responsible for predicting these electronic properties given stable conformational coordinates. For the purpose of more stable performances as well as better comparison, we recommend stratified splitting (introduced in the next subsection) for QM7.

3.1.2 QM8. The QM8 dataset comes from a recent study on modeling quantum mechanical calculations of electronic spectra and excited state energy of small molecules.[42] Multiple methods, including time-dependent density functional theories (TDDFT) and second-order approximate coupled-cluster (CC2), are applied to a collection of molecules that include up to eight heavy atoms (also a subset of the GDB-17 database[43]). In total, four excited state properties are calculated by three different methods on 22 thousand samples.

3.1.3 QM9. QM9 is a comprehensive dataset that provides geometric, energetic, electronic and thermodynamic properties for a subset of GDB-17 database,[43] comprising 134 thousand stable organic molecules with up to nine heavy atoms.[44] All molecules are modeled using density functional theory (B3LYP/6-31G(2df,p) based DFT). In our benchmark, geometric properties (atomic coordinates) are integrated into features, which are then applied to predict other properties.

The datasets introduced above (QM7, QM7b, QM8, QM9) were curated as part of the Quantum-Machine effort,[39] which has processed a number of datasets to measure the efficacy of machine-learning methods for quantum chemistry.

3.1.4 ESOL. ESOL is a small dataset consisting of water solubility data for 1128 compounds.[13] The dataset has been used to train models that estimate solubility directly from chemical structures (as encoded in SMILES strings).[22] Note that these structures don't include 3D coordinates, since solubility is a property of a molecule and not of its particular conformers.

3.1.5 FreeSolv. The Free Solvation Database (FreeSolv) provides experimental and calculated hydration free energy of small molecules in water.[16] A subset of the compounds in the dataset are also used in the SAMPL blind prediction challenge.[15] The calculated values are derived from alchemical free energy calculations using molecular dynamics simulations. We include the experimental values in the benchmark collection, and use calculated values for comparison.

3.1.6 Lipophilicity. Lipophilicity is an important feature of drug molecules that affects both membrane permeability and solubility. This dataset, curated from ChEMBL database,[45] provides experimental results of octanol/water distribution coefficient (log D at pH 7.4) of 4200 compounds.

3.1.7 PCBA. PubChem BioAssay (PCBA) is a database consisting of biological activities of small molecules generated by high-throughput screening.[35] We use a subset of PCBA, containing 128 bioassays measured over 400 thousand compounds, used by previous work to benchmark machine learning methods.[10]

3.1.8 MUV. The Maximum Unbiased Validation (MUV) group is another benchmark dataset selected from PubChem BioAssay by applying a refined nearest neighbor analysis.[46] The MUV dataset contains 17 challenging tasks for around 90 thousand compounds and is specifically designed for validation of virtual screening techniques.

3.1.9 HIV. The HIV dataset was introduced by the Drug Therapeutics Program (DTP) AIDS Antiviral Screen, which tested the ability to inhibit HIV replication for over 40 000 compounds.[47] Screening results were evaluated and placed into three categories: confirmed inactive (CI), confirmed active (CA) and confirmed moderately active (CM). We further combine the latter two labels, making it a classification task between inactive (CI) and active (CA and CM). As we are more interested in discover new categories of HIV inhibitors, scaffold splitting (introduced in the next subsection) is recommended for this dataset.

3.1.10 PDBbind. PDBbind is a comprehensive database of experimentally measured binding affinities for bio-molecular complexes.[48,49] Unlike other ligand-based biological activity datasets, in which only the structures of ligands are provided, PDBbind provides detailed 3D Cartesian coordinates of both ligands and their target proteins derived from experimental

(*e.g.*, X-ray crystallography) measurements. The availability of coordinates of the protein–ligand complexes permits structure-based featurization that is aware of the protein–ligand binding geometry. We use the "refined" and "core" subsets of the database,[50] more carefully processed for data artifacts, as additional benchmarking targets. Samples in PDBbind dataset are collected over a relatively long period of time (since 1982), hence a time splitting pattern (introduced in the next subsection) is recommended to mimic actual development in the field.

3.1.11 BACE. The BACE dataset provides quantitative (IC_{50}) and qualitative (binary label) binding results for a set of inhibitors of human β-secretase 1 (BACE-1).[51] All data are experimental values reported in scientific literature over the past decade, some with detailed crystal structures available. We merged a collection of 1522 compounds with their 2D structures and binary labels in MoleculeNet, built as a classification task. Similarly, regarding a single protein target, scaffold splitting will be more practically useful.

3.1.12 BBBP. The Blood–brain barrier penetration (BBBP) dataset comes from a recent study[52] on the modeling and prediction of the barrier permeability. As a membrane separating circulating blood and brain extracellular fluid, the blood–brain barrier blocks most drugs, hormones and neurotransmitters. Thus penetration of the barrier forms a long-standing issue in development of drugs targeting central nervous system. This dataset includes binary labels for over 2000 compounds on their permeability properties. Scaffold splitting is also recommended for this well-defined target.

3.1.13 Tox21. The "Toxicology in the 21st Century" (Tox21) initiative created a public database measuring toxicity of compounds, which has been used in the 2014 Tox21 Data Challenge.[53] This dataset contains qualitative toxicity measurements for 8014 compounds on 12 different targets, including nuclear receptors and stress response pathways.

3.1.14 ToxCast. ToxCast is another data collection (from the same initiative as Tox21) providing toxicology data for a large library of compounds based on *in vitro* high-throughput screening.[54] The processed collection in MoleculeNet includes qualitative results of over 600 experiments on 8615 compounds.

3.1.15 SIDER. The Side Effect Resource (SIDER) is a database of marketed drugs and adverse drug reactions (ADR).[55] The version of the SIDER dataset in DeepChem[56] has grouped drug side-effects into 27 system organ classes following MedDRA classifications[57] measured for 1427 approved drugs (following previous usage[56]).

3.1.16 ClinTox. The ClinTox dataset, introduced as part of this work, compares drugs approved by the FDA and drugs that have failed clinical trials for toxicity reasons.[58,59] The dataset includes two classification tasks for 1491 drug compounds with known chemical structures: (1) clinical trial toxicity (or absence of toxicity) and (2) FDA approval status. List of FDA-approved drugs are compiled from the SWEETLEAD database,[60] and list of drugs that failed clinical trials for toxicity reasons are compiled from the Aggregate Analysis of ClinicalTrials.gov (AACT) database.[61]

3.2 Dataset splitting

Typical machine learning methods require datasets to be split into training/validation/test subsets (or alternatively into *K*-folds) for benchmarking. All MoleculeNet datasets are split into training, validation and test, following a 80/10/10 ratio. Training sets were used to train models, while validation sets were used for tuning hyperparameters, and test sets were used for evaluation of models.

As mentioned previously, random splitting of molecular data isn't always best for evaluating machine learning methods. Consequently, MoleculeNet implements multiple different splittings for each dataset (Fig. 3). Random splitting randomly splits samples into the training/validation/test subsets. Scaffold splitting splits the samples based on their two-dimensional structural frameworks,[62] as implemented in RDKit.[63] Since scaffold splitting attempts to separate structurally different molecules into different subsets, it offers a greater challenge for learning algorithms than the random split.

In addition, a stratified random sampling method is implemented on the QM7 dataset to reproduce the results from the original work.[18] This method sorts datapoints in order of increasing label value (note this is only defined for real-valued output). This sorted list is then split into training/validation/test by ensuring that each set contains the full range of provided labels. Time splitting is also adopted for dataset that includes time information (PDBbind). Under this splitting method, model will be trained on older data and tested on newer data, mimicking real world development condition.

MoleculeNet contributes the code for these splitting methods into DeepChem. Users of the library can use these splits on new datasets with short library calls.

3.3 Metrics

MoleculeNet contains both regression datasets (QM7, QM7b, QM8, QM9, ESOL, FreeSolv, lipophilicity and PDBbind) and classification datasets (PCBA, MUV, HIV, BACE, BBBP, Tox21, ToxCast and SIDER). Consequently, different performance metrics need to be measured for each. Following suggestions from the community,[64] regression datasets are evaluated by mean absolute error (MAE) and root-mean-square error (RMSE), classification datasets are evaluated by area under curve (AUC) of the receiver operating characteristic (ROC) curve[65] and the precision recall curve (PRC).[66] For datasets containing more than one task, we report the mean metric values over all tasks.

To allow better comparison, we propose regression metrics according to previous work on either same models or datasets. For classification datasets, we propose recommended metrics from the two commonly used metrics: AUC-PRC and AUC-ROC. Four representative sets of ROC curves and PRCs are depicted in Fig. 4, resulting from the predictions of logistic regression and graph convolutional models on four tasks. Details about these tasks and AUC values of all curves are listed in Table 2. Note that these four tasks have different class imbalances, represented as the number of positive samples and negative samples.

As noted in previous literature,[66] ROC curves and PRCs are highly correlated, but perform significantly differently in case of

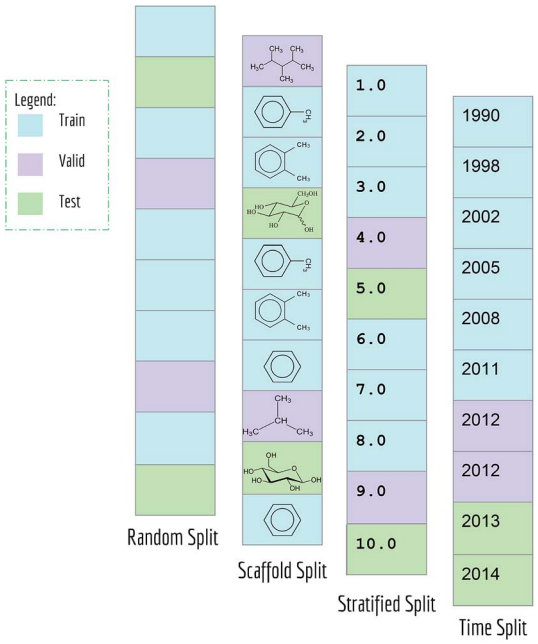

Fig. 3 Representation of data splits in MoleculeNet.

high class imbalance. As shown in Fig. 4, the fraction of positive samples decreases from over 80% (panels A and B) to less than 5% (panels G and H). This change accompanies the difference in how the two metrics treat model performances. In particular, PRCs put more emphasis on the low recall (also known as true positive rate (TPR)) side in case of highly imbalanced data: logistic regression slightly outperforms graph convolutional models in the low TPR side of ROC curves (panels C, E and G, lower left corner), which creates different margins on the low recall side of PRCs.

ROC curves and PRCs share one same axis, while using false positive rate (FPR) and precision for the other axis respectively. Recall that FPR and precision are defined as follows:

$$\mathrm{FPR} = \frac{\text{false positive}}{\text{false positive} + \text{true negative}}$$

$$\text{precision} = \frac{\text{true positive}}{\text{false positive} + \text{true positive}}$$

When positive samples form only a small proportion of all samples, false positive predictions exert a much greater influence on precision than FPR, amplifying the difference between PRC and ROC curves. Virtual screening experiments do have extremely low positive rates, suggesting that the correct metric to analyze may depend on the experiment at hand. In this work, we hence propose recommended metrics based on positive rates, PRC-AUC is used for datasets with positive rates less than 2%, otherwise ROC-AUC is used.

3.4 Featurization

A core challenge for molecular machine learning is effectively encoding molecules into fixed-length strings or vectors. Although SMILES strings are unique representations of molecules, most molecular machine learning methods require further information to learn sophisticated electronic or topological features of molecules from limited amounts of data. (Recent work has demonstrated the ability to learn useful representations from SMILES strings using more sophisticated methods,[67] so it may be feasible to use SMILES strings for further learning tasks in the near future.) Furthermore, the enormity of chemical space often requires representations of molecules specifically suited to the learning task at hand. MoleculeNet contains implementations of six useful molecular featurization methods (Fig. 5).

3.4.1 ECFP. Extended-Connectivity Fingerprints (ECFP) are widely-used molecular characterizations in chemical informatics.[21] During the featurization process, a molecule is decomposed into submodules originated from heavy atoms, each assigned with a unique identifier. These segments and identifiers are extended through bonds to generate larger substructures and corresponding identifiers.

After hashing all these substructures into a fixed length binary fingerprint, the representation contains information

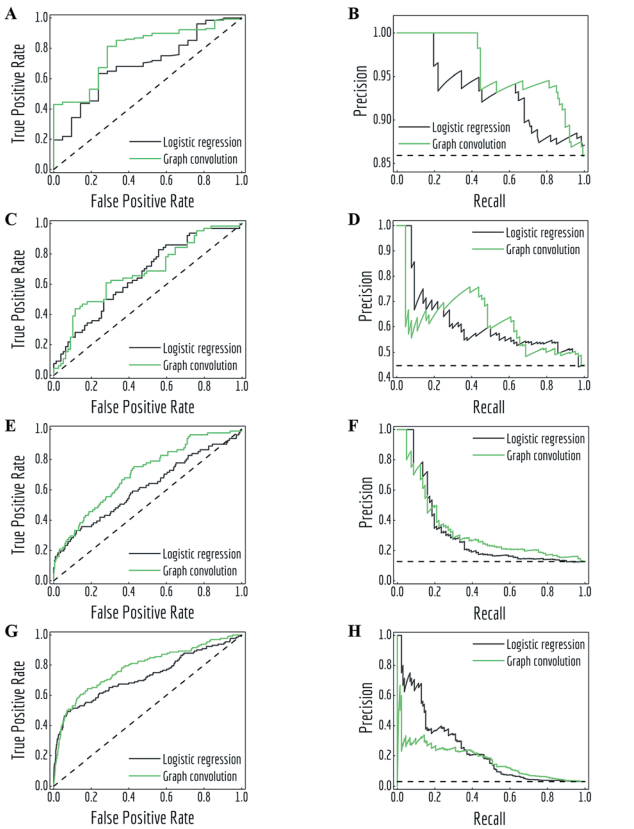

Fig. 4 Receiver operating characteristic (ROC) curves and precision recall curves (PRC) for predictions of logistic regression and graph convolutional models under different class imbalance condition (details listed in Table 2). (A, B) task "FDA_APPROVED" from ClinTox, test subset; (C, D) task "Hepatobiliary disorders" from SIDER, test subset; (E, F) task "NR-ER" from Tox21, validation subset; (G, H): task "HIV_active" from HIV, test subset. Black dashed lines are performances of random classifiers.

Table 2 Task details and area under curve (AUC) values of sample curves

Task	P/N[a]	Model	ROC	PRC
"FDA_APPROVED" ClinTox, test subset	128/21	Logistic regression	0.691	0.932
		Graph convolution	0.791	0.959
"Hepatobiliary disorders" SIDER, test subset	64/79	Logistic regression	0.659	0.612
		Graph convolution	0.675	0.620
"NR-ER" Tox21, valid subset	81/553	Logistic regression	0.612	0.308
		Graph convolution	0.705	0.333
"HIV_active" HIV, test subset	132/4059	Logistic regression	0.724	0.236
		Graph convolution	0.783	0.169

[a] Number of positive samples/number of negative samples.

about topological characteristics of the molecule, which enables it to be applied to tasks such as similarity searching and activity prediction. The MoleculeNet implementation uses ECFP4 fingerprints generated by RDKit.[63]

3.4.2 Coulomb matrix. *Ab initio* electronic structure calculations typically require a set of nuclear charges $\{Z\}$ and the corresponding Cartesian coordinates $\{\mathbf{R}\}$ as input. The Coulomb Matrix (CM) **M**, proposed by Rupp *et al.*[17] and defined below, encodes this information by use of the atomic self-energies and internuclear coulomb repulsion operator.

$$M_{IJ} = \begin{cases} 0.5Z_I^{2.4} & \text{for } I = J \\ \dfrac{Z_I Z_J}{|\boldsymbol{R}_I - \boldsymbol{R}_J|} & \text{for } I \neq J \end{cases}$$

Here, the off-diagonal elements correspond to the Coulomb repulsion between atoms I and J, and the diagonal elements correspond to a polynomial fit of atomic self-energy to nuclear charge. The Coulomb matrix of a molecule is invariant to translation and rotation of that molecule, but not with respect to atom index permutation. In the construction of Coulomb matrix, we first use the nuclear charges and distance matrix generated by RDKit[63] to acquire the original Coulomb matrix, then an optional random atom index sorting and binary expansion transformation can be applied during training in order to achieve atom index invariance, as reported by Montavon *et al.*[18]

3.4.3 Grid featurizer. The grid featurizer is a featurization method (introduced in the current work) initially designed for the PDBbind dataset in which structural information of both the ligand and target protein are considered. Since binding affinity stems largely from the intermolecular forces between ligands and proteins, in addition to intramolecular interactions, we seek to incorporate both the chemical interaction within the binding pocket as well as features of the protein and ligand individually.

The grid featurizer was inspired by the NNscore featurizer[68] and SPLIF[69] but optimized for speed, robustness, and generalizability. The intermolecular interactions enumerated by the featurizer include salt bridges and hydrogen bonding between protein and ligand, intra-ligand circular fingerprints, intra-protein circular fingerprints, and protein–ligand SPLIF fingerprints. A more detailed breakdown can be found in the ESI.†

3.4.4 Symmetry function. Symmetry function, first introduced by Behler and Parrinello,[70] is another common encoding of atomic coordinates information. It focuses on preserving the rotational and permutation symmetry of the system. The local environment of an atom in the molecule is expressed as a series of radial and angular symmetry functions with different distance and angle cutoffs, the former focusing on distance between atom pairs and the latter focusing on angles formed within triplets of atoms.

As symmetry function put most emphasis on spatial positions of atoms, it is intrinsically hard for it to distinguish different atom types (H, C, O). MoleculeNet utilizes a slightly modified version of original symmetry function[71] which further separate radial and angular symmetry terms according to the type of atoms in the pair or triplet. Further details can be found in the article[71] or our implementation.

3.4.5 Graph convolutions. The graph convolutions featurization support most graph-based models. It computes an initial feature vector and a neighbor list for each atom. The feature vector summarizes the atom's local chemical environment, including atom-types, hybridization types, and valence structures. Neighbor lists represent connectivity of the whole molecule, which are further processed in each model to generate graph structures (discussed in further details in following parts).

3.4.6 Weave. Similar to graph convolutions, the weave featurization encodes both local chemical environment and connectivity of atoms in a molecule. Atomic feature vectors are exactly the same, while connectivity uses more detailed pair features instead of neighbor listing. The weave featurization calculates a feature vector for each pair of atoms in the molecule, including bond properties (if directly connected), graph distance and ring info, forming a feature matrix. The method supports graph-based models that utilize properties of both nodes (atoms) and edges (bonds).

3.5 Models – conventional models

MoleculeNet tests the performance of various machine learning models on the datasets discussed previously. These models could be further categorized into conventional method and graph-based method according to their structures and input types. The following sections will give brief introductions to benchmarked algorithms. The results section will discuss

performance numbers in detail. Here we briefly review conventional methods including logistic regression, support vector classification, kernel ridge regression, random forests,[72] gradient boosting,[73] multitask networks,[9,10] bypass networks[74] and influence relevance voting.[75] The next section graph-based models will give introductions to graph convolutional models,[22] weave models,[23] directed acyclic graph models,[14] deep tensor neural networks,[19] ANI-1 (ref. 71) and message passing neural networks.[76] As part of this work, all methods are implemented in the open source DeepChem package.[30]

3.5.1 Logistic regression. Logistic regression models (Logreg) apply the logistic function to weighted linear combinations of their input features to obtain model predictions. It is often common to use regularization to encourage learned weights to be sparse.[77] Note that logistic regression models are only defined for classification tasks.

3.5.2 Support vector classification. Support vector machine (SVM) is one of the most famous and widely-used machine learning method.[78] As in classification task, it defines a decision plane which separates data points of different class with maximized margin. To further increase performance, we incorporates regularization and a radial basis function kernel (KernelSVM).

3.5.3 Kernel ridge regression. Kernel ridge regression (KRR) is a combination of ridge regression and kernel trick. By using a nonlinear kernel function (radial basis function), it learns a non-linear function in the original space that maps features to predicted values.

3.5.4 Random forests. Random forests (RF) are ensemble prediction methods.[72] A random forest consists of many individual decision trees, each of which is trained on a subsampled version of the original dataset. The results for individual trees are averaged to provide output predictions for the full forest. Random forests can be used for both classification and regression tasks. Training a random forest can be computationally intensive, so benchmarks only include random forest results for smaller datasets.

3.5.5 Gradient boosting. Gradient boosting is another ensemble method consisting of individual decision trees.[73] In contrast to random forests, it builds relatively simple trees which are sequentially incorporated to the ensemble. In each step, a new tree is generated in a greedy manner to minimize loss function. A sequence of such "weak" trees are combined together into an additive model. We utilize the XGBoost implementation of gradient boosting in DeepChem.[79]

3.5.6 Multitask/singletask network. In a multitask network,[10] input featurizations are processed by fully connected neural network layers. The processed output is shared among all learning tasks in a dataset, and then fed into separate linear classifiers/regressors for each different task. In the case that a dataset contains only a single task, multitask networks are just fully connected neural networks (Singletask Network). Since multitask networks are trained on the joint data available for various tasks, the parameters of the shared layers are encouraged to produce a joint representation which can share information between learning tasks. This effect does seem to have limitations; merging data from uncorrelated tasks has only moderate effect.[80] As a result, MoleculeNet does not attempt to train extremely large multitask networks combining all data for all datasets.

3.5.7 Bypass multitask networks. Multitask modeling relies on the fact that some features have explanatory power that is shared among multiple tasks. Note that the opposite may well be true; features useful for one task can be detrimental to other tasks. As a result, vanilla multitask networks can lack the power to explain unrelated variations in the samples. Bypass networks attempt to overcome this variation by merging in per-task independent layers that "bypass" shared layers to directly connect inputs with outputs.[74] In other words, bypass multitask networks consist of $n_{\text{tasks}} + 1$ independent components: one "multitask" layers mapping all inputs to shared representations, and n_{tasks} "bypass" layers mapping inputs for each specific task to their labels. As the two groups have separate parameters, bypass networks may have greater explanatory power than vanilla multitask networks.

3.5.8 Influence relevance voting. Influence Relevance Voting (IRV) systems are refined K-nearest neighbour classifiers.[75] Using the hypothesis that compounds with similar substructures have similar functionality, the IRV classifier

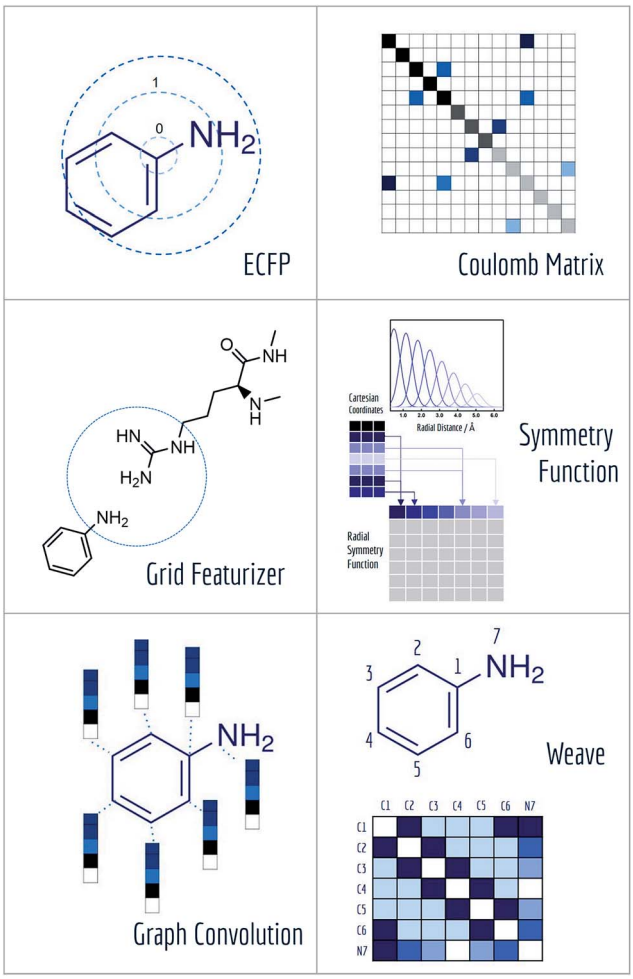

Fig. 5 Diagrams of featurizations in MoleculeNet.

makes its prediction by combining labels from the top-K compounds most similar to a provided test sample.

The Jaccard–Tanimoto similarity between fingerprints of compounds is used as the similarity measurement:

$$S\left(\vec{A},\vec{B}\right) = \frac{A \cap B}{A \cup B}$$

Then IRV model calculates a weighted sum of labels of top K similar compounds to predict the result, in which weights are the outputs of a one-hidden layer neural network with similarities and rankings of top-K compounds as input. Detailed descriptions of the model can be found in the original article.[75]

3.6 Models – graph based models

Early attempts to directly use molecular structures instead of selected features has emerged in 1990s.[81,82] While in recent years, models propelled by the very similar idea start to grow rapidly. These specifically designed methods, namely graph-based models, are naturally suitable for modelling molecules. By defining atoms as nodes, bonds as edges, molecules can be modeled as mathematical graphs. As noted in a recent paper,[76] this natural similarity has inspired a number of models to utilize the graph structure of molecules to gain higher performances. In general, graph-based models apply adaptive functions to nodes and edges, allowing for a learnable featurization process. MoleculeNet provides implementations of multiple graph-based models which use different variants of molecular graphs. Fig. 6 provide simple illustrations of these methods' core structures. We describe these methods in details in the following sections. To further validate the model implementations, we compare the performances of these models with their original sources, results can be found in the ESI.†

3.6.1 Graph convolutional models. Graph convolutional models (GC) extend the decomposition principles of circular fingerprints. Both methods gradually merge information for distant atoms by extending radially through bonds. This information is used to generate identifiers for all substructures. However, instead of applying fixed hash functions, graph convolutional models allow for adaptive learning by using differentiable network layers. This creates a learnable process capable of extracting useful representations of molecules suited to the task at hand (note that this property is shared, to some degree, by all deep architectures considered in MoleculeNet. However, graph convolutional architectures are more explicitly designed to encourage extraction of useful featurizations).

On a higher level, graph convolutional models treat molecules as undirected graphs, and apply the same learnable function to every node (atom) and its neighbors (bonded atoms) in the graph. This structure recapitulates convolution layers in visual recognition deep networks.

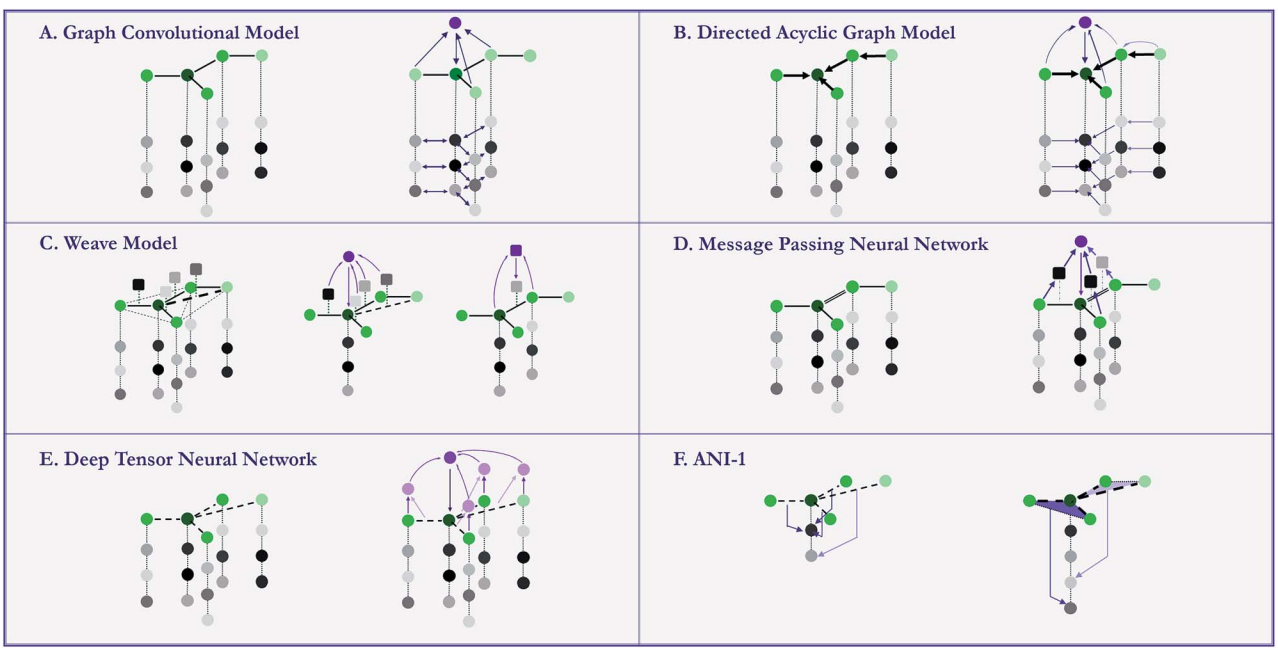

Fig. 6 Core structures of graph-based models implemented in MoleculeNet. To build features for the central dark green atom: (A) graph convolutional model: features are updated by combination with neighbour atoms; (B) directed acyclic graph model: all bonds are directed towards the central atom, features are propagated from the farthest atom to the central atom through directed bonds; (C) Weave model: pairs are formed between each pair of atoms (including not directly bonded pairs), features for the central atom are updated using all other atoms and their corresponding pairs, pair features are also updated by combination of the two pairing atoms; (D) message passing neural network: neighbour atoms' features are input into bondtype-dependent neural networks, forming outputs (messages). Features of the central atom are then updated using the outputs; (E) deep tensor neural network: no explicit bonding information is included, features are updated using all other atoms based on their corresponding physical distances; (F) ANI-1: features are built on distance information between pairs of atoms (radial symmetry functions) and angular information between triplets of atoms (angular symmetry functions).

MoleculeNet uses the graph convolutional implementation in DeepChem from previous work.[56] This implementation converts SMILES strings into molecular graphs using RDKit[63] As mentioned previously, the initial representations assign to each atom a vector of features including its element, connectivity, valence, *etc.* Then several graph convolutional modules, each consisting of a graph convolutional layer, a batch normalization layer and a graph pool layer, are sequentially added, followed by a fully-connected dense layer. Finally, the feature vectors for all nodes (atoms) are summed, generating a graph feature vector, which is fed to a classification or regression layer.

3.6.2 Weave models. The Weave architecture is another graph-based model that regards each molecule as a undirected graph. Similar to graph convolutional models, it utilizes the idea of adaptive learning on extracting meaningful representations.[23] The major difference is the size of the convolutions: to update features of an atom, weave models combine info from all other atoms and their corresponding pairs in the molecule. Weave models are more efficient at transmitting information between distant atoms, at the price of increased complexity for each convolution.

In our implementation, a molecule is first encoded into a list of atomic features and a matrix of pair features by the weave model's featurization method. Then in each weave module, these features are inputted into four sets of fully connected layers (corresponding to four paths from two original features to two updated features) and concatenated to form new atomic and pair features. After stacking several weave modules, a similar gather layer combines atomic features together to form molecular features that are fed into task-specific layers.

3.6.3 Directed acyclic graph models. Directed Acyclic Graph (DAG) models regard molecules as directed graphs. While chemical bonds typically do not have natural directions, one can arbitrarily generate a DAG on a molecule by designating a central atom and then define directions of all bonds in certain orientations towards the atom.[14] In the case of small molecules, taking all possible orientations is computationally feasible. In other words, for a molecule with n_a atoms, the model will generate n_a DAGs, each centered on a different atom.

In the actual calculations of a graph, a vector of graph features is calculated for each atom based on its atomic features (reusing the graph convolutions featurizer) and its parents' graph features. As features gradually propagate through bonds, information converges on the central atom. Then a final sum of all graphs gives the molecular features, which are fed into classification or regression tasks. Note that n_a graphs are evaluated for each molecule, which can cause a significant increase in required calculations.

3.6.4 Deep tensor neural networks. Deep Tensor Neural Networks (DTNN) are adaptable extensions of the Coulomb matrix featurizer.[19] The core idea is to directly use nuclear charge (atom number) and the distance matrix to predict energetic, electronic or thermodynamic properties of small molecules. To build a learnable system, the model first maps atom numbers to trainable embeddings (randomly initialized) as atomic features. Then each atomic feature a_i is updated based on distance info d_{ij} and other atomic features a_j. Comparing with Weave models, DTNNs share the same idea in terms of updating based on both atomic and pair features, while the difference is using physical distance instead of graph distance. Note that the use of 3D coordinates to calculate physical distances limits DTNNs to quantum mechanical (or perhaps biophysical) datasets.

We reimplement the model proposed by Schütt *et al.*[19] in a more generalized fashion. Atom numbers and a distance matrix are calculated by RDKit,[63] using the Coulomb matrix featurizer. After embedding atom numbers into feature vectors a_i, we update a_i in each convolutional layer by adding the outputs from all network layers which use d_{ij} and a_j $(i \neq j)$ as input. After several layers of convolutions, all atomic features are summed together to form molecular features, used for classification and regression tasks.

3.6.5 ANI-1. ANI-1 is designed as a deep neural network capable of learning accurate and transferable potentials for organic molecules. It is based on the symmetry function method,[70] with additional changes enabling it to learn different potentials for different atom types. Feature vector, a series of symmetry functions, is built for each atom in the molecule based on its atom type and interaction with other atoms. Then the feature vectors are fed into different neural network potentials (depending on atom types) to generate predictions of properties.

This model is first introduced by Smith *et al.*[71] In their original article, the model is trained on 58k small molecules with 8 or less heavy atoms, each with multiple poses and potentials. Training set in total has 17.2 million data points, which is far bigger than qm8 or qm9 in our collection. Since we only have molecules in their most stable configuration, we cannot expect similar level of accuracy. Further comparison and benchmarking with similar size of training set is left to future work.

3.6.6 Message passing neural networks. Message passing neural network (MPNN) is a generalized model proposed by Gilmer *et al.*[76] that targets to formulate a single framework for graph based model. The prediction process is separated into two phases: message passing phase and readout phase. Multiple message passing phases are stacked to extract abstract information of the graph, then the readout phase is responsible for mapping the graph to its properties.

Here we reimplemented the best-performing model in the original article: using an Edge network as message passing function and a set2set model[83] as readout function. In message passing phase, an edge-dependent neural network maps all neighbour atoms' feature vectors to updating messages, which are then merged using gated recurrent units. In the final readout phase, feature vectors for all atoms are regarded as a set, then an LSTM using attention mechanism is applied on the set for multiple steps, with its final state used as the output for the molecule.

4 Results and discussion

In this section, we discuss the performance of benchmarked models on MoleculeNet datasets. Different models are applied

depending on the size, features and task types of the dataset. All graph models use their corresponding featurizations. Non-graph models use ECFP featurizations by default, Coulomb Matrix (CM) and Grid featurizer are also applied for certain datasets.

We run a brief Gaussian process hyperparameter optimization on each combination of dataset and model. Then three independent runs with different random seeds are performed. More detailed description of optimization method and performance tables can be found in ESI.† Note that all benchmark results presented here are the average of three runs, with standard deviations listed or illustrated as error bars.

We also run a set of experiments focusing on how variable size of training set affect model performances (Tox21, FreeSolv and QM7). Details will be presented in the following texts.

4.1 Biophysics and physiology tasks

Tables S2, S3† and Fig. 7–9 report AUC-ROC or AUC-PRC results of 4 to 9 different models on biophysics datasets (PCBA, MUV, HIV, BACE) and physiology datasets (BBBP, Tox21, Toxcast, SIDER, ClinTox). Some models were too computationally expensive to be run on the larger datasets. All of these datasets contain only classification tasks.

Most models have train scores (listed in Tables S2 and S3†) higher than validation/test scores, indicating that overfitting is a general issue. Singletask logistic regression exhibits the largest gaps between train scores and validation/test scores, while models incorporating multitask structure generally show less overfit, suggesting that multitask training has a regularizing effect. Most physiological and biophysical datasets in MoleculeNet have only a low volume of data for each task. Multitask algorithms combine different tasks, resulting in a larger pool of data for model training. In particular, multitask training can, to some extent, compensate for the limited data amount available for each individual task.

Graph convolutional models and weave models, each based on an adaptive method of featurization,[22,23] show strong validation/test results on larger datasets, along with less overfit. Similar results are reported in previous graph-based algorithms,[14,19,22,23,76] showing that learnable featurizations can provide a large boost compared with conventional featurizations.

For smaller singletask datasets (less than 3000 samples), differences between models are less clear. Kernel SVM and

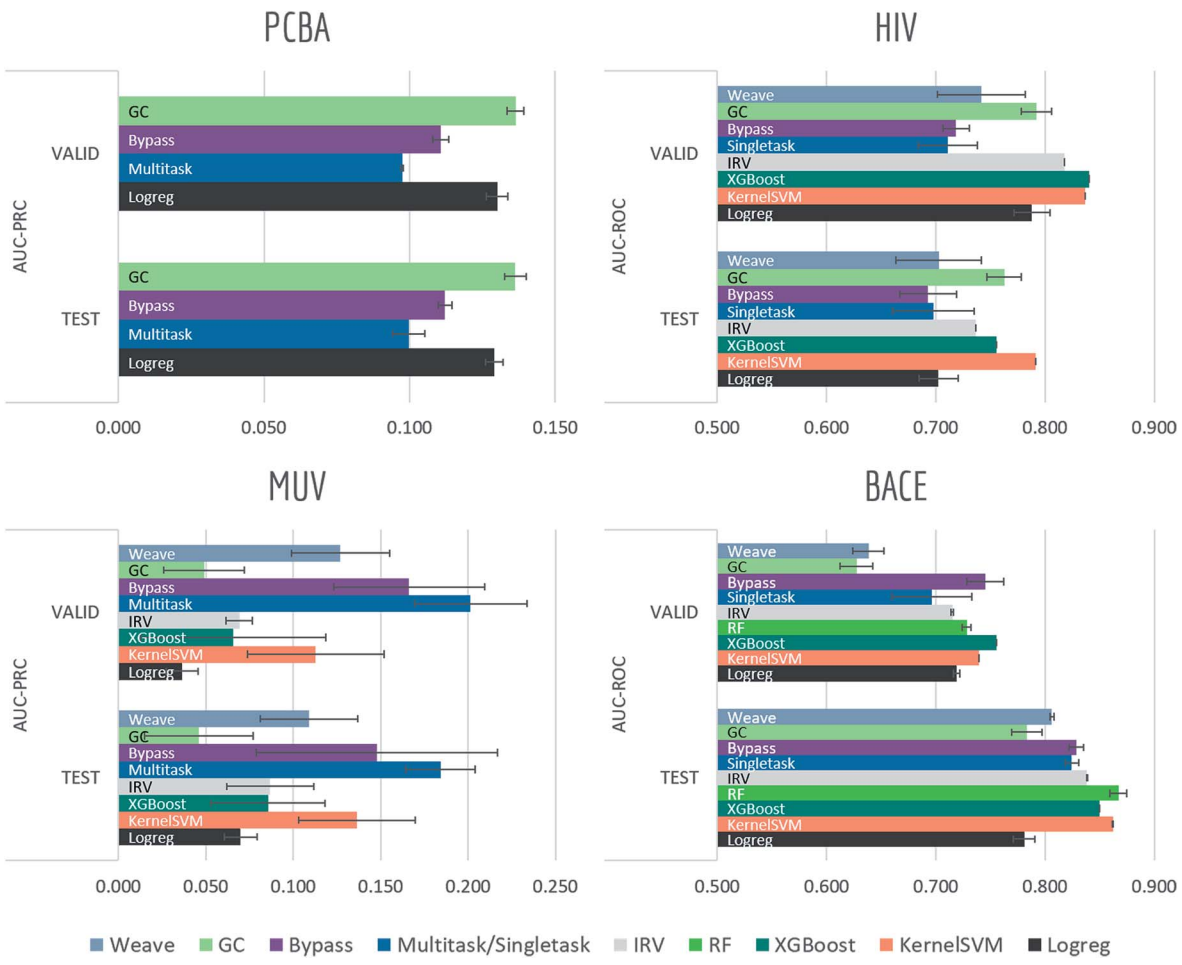

Fig. 7 Benchmark performances for biophysics tasks: PCBA, 4 models are evaluated by AUC-PRC on random split; MUV, 8 models are evaluated by AUC-PRC on random split; HIV, 8 models are evaluated by AUC-ROC on scaffold split; BACE, 9 models are evaluated by AUC-ROC on scaffold split. For AUC-ROC and AUC-PRC, higher value indicates better performance (to the right).

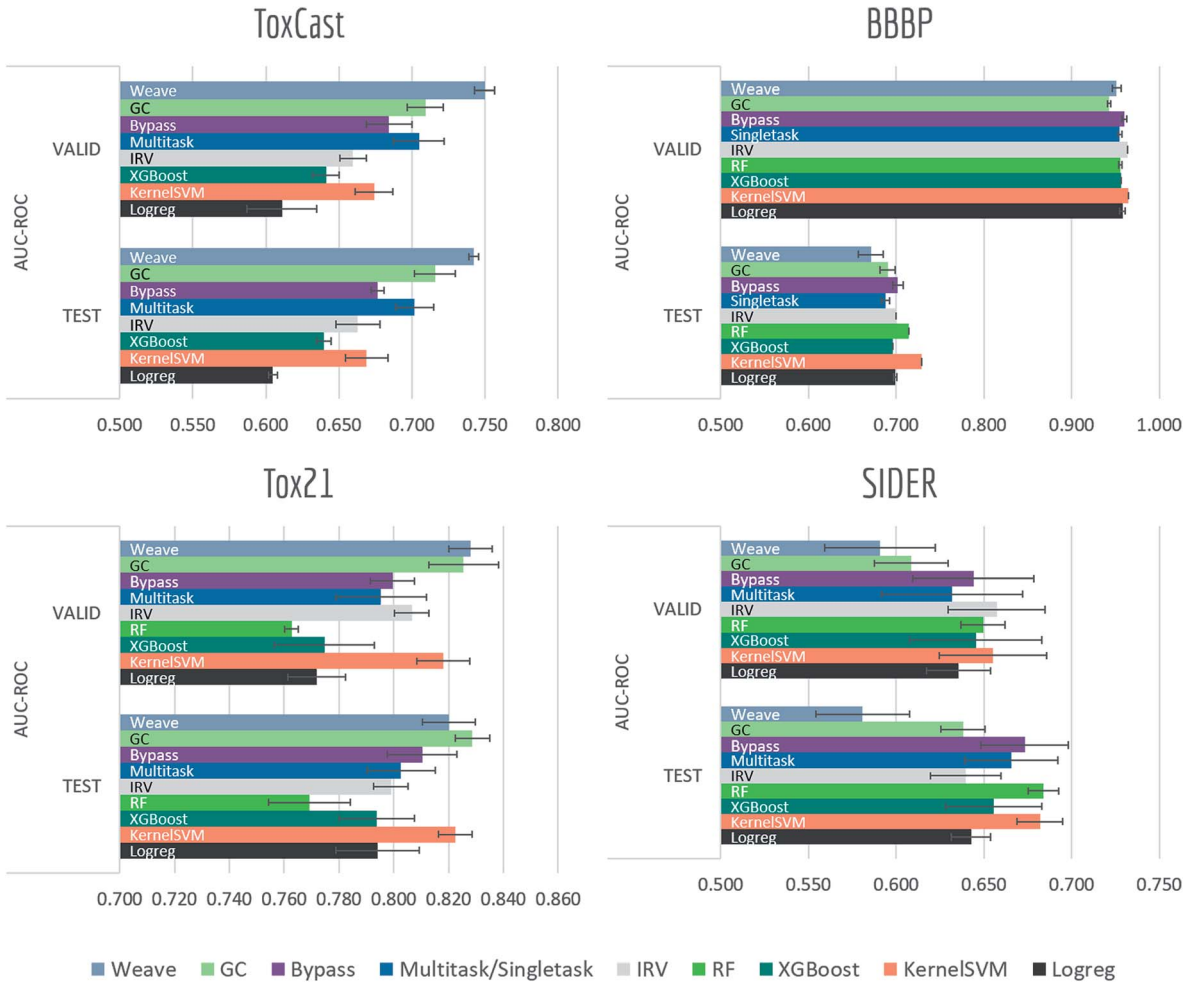

Fig. 8 Benchmark performances for physiology tasks: ToxCast, 8 models are evaluated by AUC-ROC on random split; Tox21, 9 models are evaluated by AUC-ROC on random split; BBBP, 9 models are evaluated by AUC-ROC on scaffold split; SIDER, 9 models are evaluated by AUC-ROC on random split. For AUC-ROC, higher value indicates better performance (to the right).

ensemble tree methods (gradient boosting and random forests) are more robust under data scarcity, while they generally need longer running time (see Table S1†). Worse performances of graph-based models are within expectation as complex models generally require more training data.

Bypass networks show higher train scores and equal or higher validation/test scores compared with vanilla multitask networks, suggesting that the bypass structure does add robustness. IRV models achieve performance broadly comparable with multitask networks. However, the quadratic nearest neighbor search makes the IRV models slower to train than the multitask networks (see Table S1†).

Three datasets (HIV, BACE, BBBP) in these two categories are evaluated under scaffold splitting. As compounds are divided by their molecular scaffolds, increasing differences between train, validation and test performances are observed. Scaffold splits provide a stronger test of a given model's generalizability compared with random splitting. Two datasets (PCBA, MUV) are evaluated by AUC-PRC, which is more practically useful under high class imbalance as discussed above. Graph convolutional model performs the best on PCBA (positive rate 1.40%), while results on MUV (positive rate 0.20%) are much less stable, which is most likely due to its extreme low amount of positive samples. Under such high imbalance, graph-based models are still not robust enough in controlling false positives.

Here we performed a more detailed experiment to illustrate how model performances change with increasing training samples. We trained multiple models on Tox21 with training sets of different size (10% to 90% of the whole dataset) Fig. 10 displayed mean out-of-sample performances (and standard deviations) of five independent runs. A clear increase on performance is observed for each model, and graph-based models (graph convolutional model and weave model) always stay on top of the lines. By drawing a horizontal line at around 0.80, we can see graph-based models achieve the similar level of accuracy with multitask networks by using only one-third of the training samples (30% *versus* 90%).

4.2 Biophysics task – PDBbind

The PDBbind dataset maps distinct ligand–protein structures to their binding affinities. As discussed in the datasets section, we

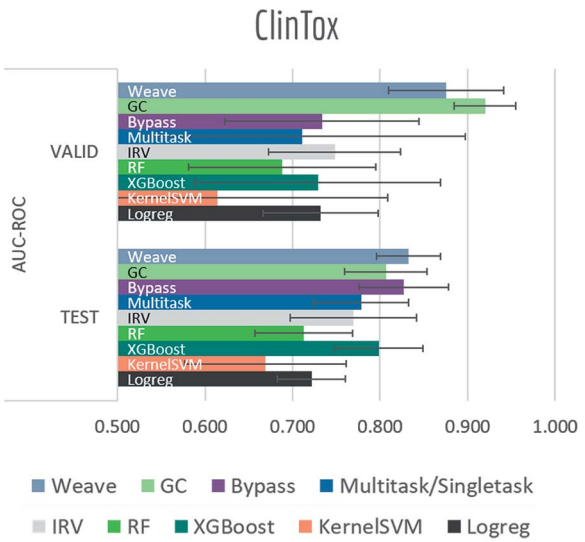

Fig. 9 Benchmark performances for physiology tasks: ClinTox, 9 models are evaluated by AUC-ROC on random split.

created grid featurizer to harness the joint ligand–protein structural information in PDBbind to build a model that predicts the experimental K_i of binding. We applied time splitting to all three subsets: core, refined, and full subsets of PDBbind (Core contains roughly 200 structures, refined 4000, and full 15 000. The smaller datasets are cleaned more thoroughly than larger datasets.), with all results displayed in Table S4† and Fig. 11. Clearly as dataset size increased, we can see a significant boost on validation/test set performances. At the same time, for the two larger subsets: refined and full, switching from pure ligand-based ECFP to grid featurizer do increase the performances by a small margin in both Singletask networks and random forests. While for core subset, all models are showing relatively high errors and two featurizations do not show clear differences, which is within expectation as sample amount in core subset is too small to support a stable model performance. Note that models on the full set aren't

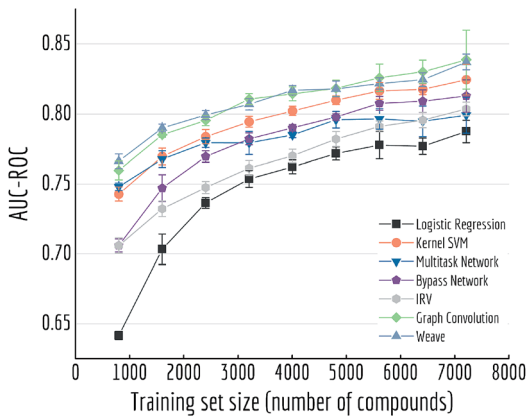

Fig. 10 Out-of-sample performances with different training set sizes on Tox21. Each datapoint is the average of 5 independent runs, with standard deviations shown as error bars.

significantly superior to models with less data; this effect may be due to the additional data being less clean.

Note that all models display heavy overfitting. Additional clean data may be required to create more accurate models for protein–ligand binding.

4.3 Physical chemistry tasks

Solubility, solvation free energy and lipophilicity are basic physical chemistry properties important for understanding how molecules interact with solvents. Fig. 12 and Table S5† presented performances on predicting these properties.

Graph-based methods: graph convolutional model, DAG, MPNN and weave model all exhibit significant boosts over vanilla singletask network, indicating the advantages of learnable featurizations. Differences between graph-based methods are rather minor and task-specific. The best-performing models in this category can already reach the accuracy level of *ab initio* predictions ($\pm$0.5 for ESOL, $\pm$1.5 kcal mol^{-1} for FreeSolv).

We performed a more detailed comparison between data-driven methods and *ab initio* calculations on FreeSolv. Hydration free energy has been widely used as a test of computational chemistry methods. With free energy values ranging from $-$25.5 to 3.4 kcal mol^{-1} in the FreeSolv dataset, RMSE for calculated results reach up to 1.5 kcal mol^{-1}.[15] On the other hand, though machine learning methods typically need large amounts of training data to acquire predictive power, they can achieve higher accuracies given enough data. We investigate how the performance of machine learning methods on FreeSolv changes with the volume of training data. In particular, we want to know the amount of data required for machine learning to achieve accuracy similar to that of physically inspired algorithms.

For Fig. 13, we similarly generated a series of models with different training set volumes and calculated their out-of-sample RMSE. Each data point displayed is the average of 5 independent runs, with standard deviations displayed as error bars. Both graph convolutional model and weave model are capable of achieving better performances with enough training samples (30% and 50% of the data respectively). Given the size of FreeSolv dataset is only around 600 compounds, a weave model can reach state-of-the-art free energy calculation performances by training on merely 200 samples. On the other hand, comparing with singletask network's performance, weave model achieved the same level of accuracy with only one-third of the training samples.

4.4 Quantum mechanics tasks

The QM datasets (QM7, QM7b, QM8, QM9) represent another distinct category of properties that are typically calculated through solving Schrödinger's equation (approximately using techniques such as DFT). As most conventional methods are slower than data-driven methods by orders of magnitude, we hope to learn effective approximators by training on existing datasets.

Table S6† and Fig. 15 display the performances in mean absolute error of multiple methods. Tables S7–S9† show detailed performances for each task (due to difference in range

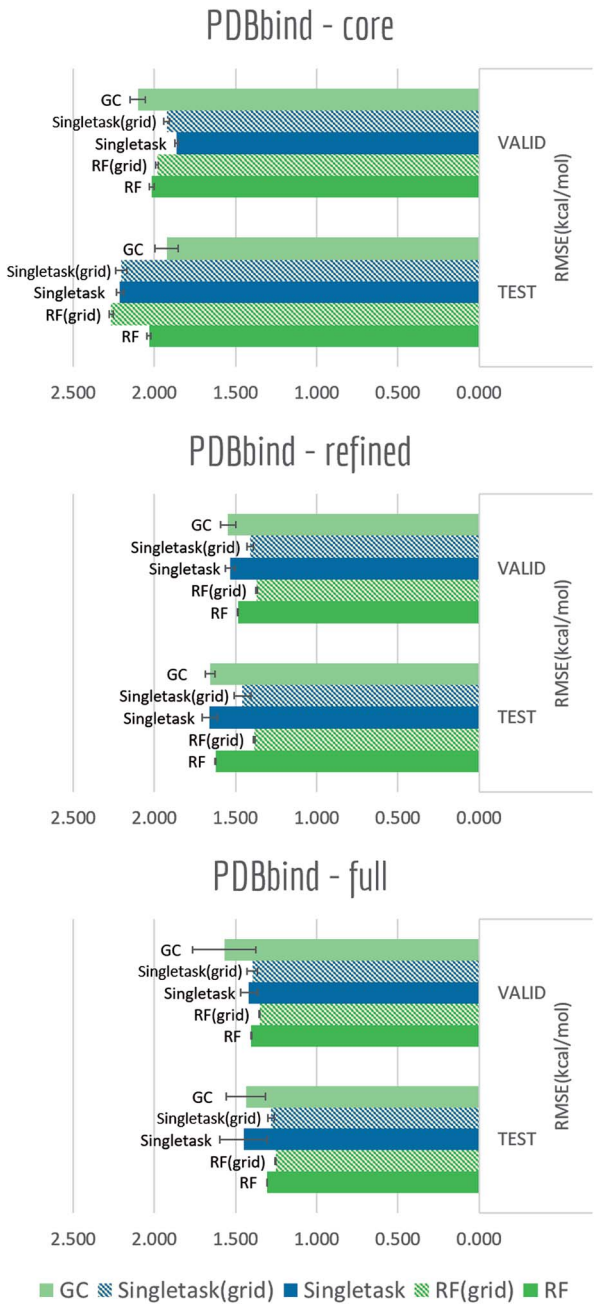

Fig. 11 Benchmark performances of PDBbind: 5 models are evaluated by RMSE on the three subsets: core, refined and full. Time split is applied to all three subsets. Noe that for RMSE, lower value indicates better performance (to the right).

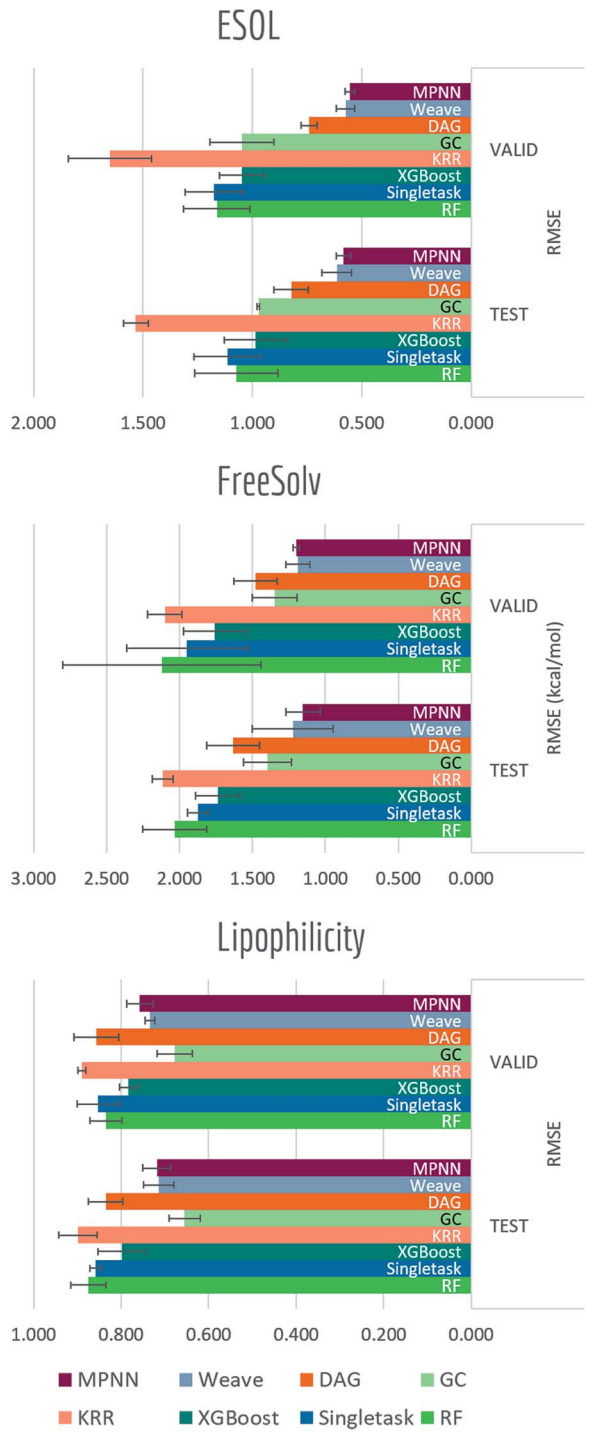

Fig. 12 Benchmark performances for physical chemistry tasks: ESOL, 8 models are evaluated by RMSE on random split; FreeSolv, 8 models are evaluated by RMSE on random split; lipophilicity, 8 models are evaluated by RMSE on random split. Note that for RMSE, lower value indicates better performance (to the right).

of labels, mean performances of QM7b and QM9 are more skewed). Unsurprisingly, significant boosts on performances and less overfitting are observed for models incorporating distance information (multitask networks and KRR with Coulomb matrix featurization, DTNN, MPNN). In particular, KRR and multitask networks (CM) outperform their corresponding baseline models in QM7 and QM9 by a large margin, while DTNN and MPNN display less error comparing with graph convolutional models as well. At the same time, DTNN and MPNN gains better performances than multitask networks and KRR (CM) on most tasks. Table S7† shows that DTNN outperforms KRR (CM) on 12/14 tasks in QM7b (though the mean error shows the opposite result due to averaging errors on different magnitudes). In total, DTNN and MPNN covers the best-performing models on 28/39 of all tasks in this

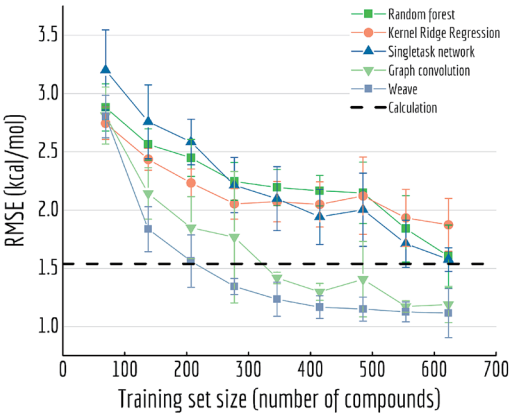

Fig. 13 Out-of-sample performances with different training set sizes on FreeSolv. Each datapoint is the average of 5 independent runs, with standard deviations shown as error bars.

category, again reflecting the superiority of learnable featurization.

Another variable training size experiment is performed on QM7: predicting atomization energy. All mean absolute error performances are displayed in Fig. 14. Clearly incorporation of spatial position creates the huge gap between models, DTNN and multitask networks (CM) reach similar level of accuracy as reported in previous work on this dataset (there is still a gap between the MoleculeNet implementation and best reported numbers from previous work,[18,19] which would likely be closed by training models longer). ANI-1 is also reported to achieve comparable performances on similar task in the previous work[71] with a much larger dataset. Apparently its worse performance is restricted by training set size, as the MAE is keep decreasing with more training samples.

For QM series, proper choice of featurization appears critical. As mentioned previously, ECFP only consider graph substructures, while Coulomb matrix and graph featurizations used by DTNN and MPNN are explicitly calculated on charges

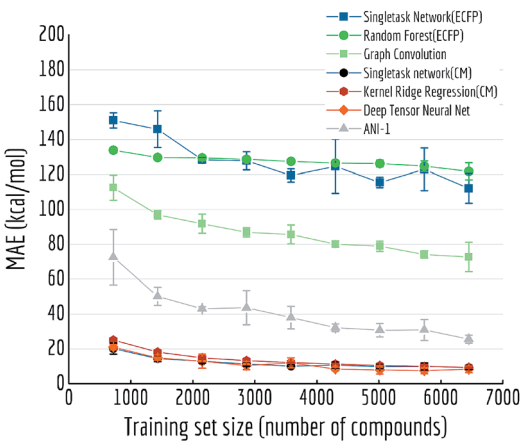

Fig. 14 Out-of-sample performances with different training set sizes on QM7. Each datapoint is the average of 5 independent runs, with standard deviations shown as error bars.

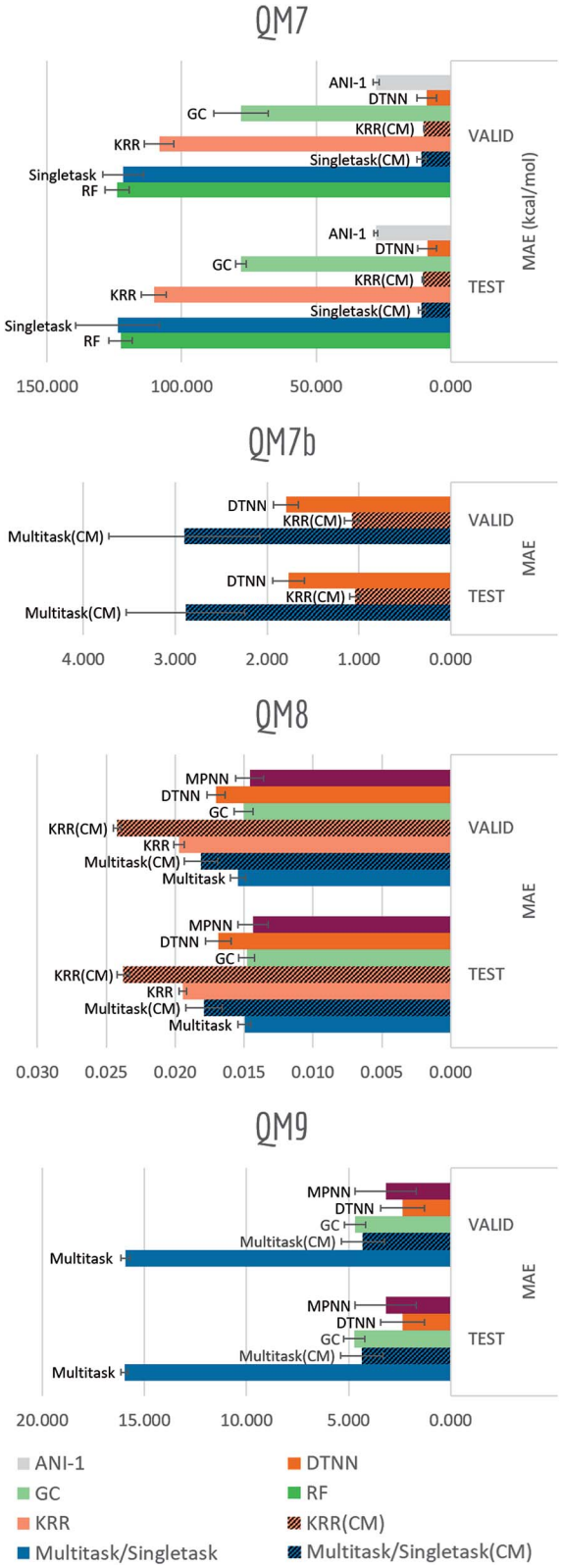

Fig. 15 Benchmark performances for quantum mechanics tasks: QM7, 8 models are evaluated by MAE on stratified split; QM7b, 3 models (QM7b only provides 3D coordinates) are evaluated by MAE on random split; QM8, 7 models are evaluated by MAE on random split; QM9, 5 models are evaluated by MAE on random split. Note that for MAE, lower value indicates better performance (to the right).

Table 3 Summary of performances (test subset): conventional methods *versus* graph-based methods. Graph-based models outperform conventional methods on 11/17 datasets

Category	Dataset	Metric	Best performances – conventional methods	Best performances – graph-based methods
Quantum mechanics	QM7	MAE	KRR (CM): 10.22	**DTNN: 8.75**
	QM7b	MAE	KRR (CM): 1.05	**DTNN: 1.77**[a]
	QM8	MAE	Multitask: 0.0150	**MPNN: 0.0143**
	QM9	MAE	Multitask (CM): 4.35	**DTNN: 2.35**
Physical chemistry	ESOL	RMSE	XGBoost: 0.99	**MPNN: 0.58**
	FreeSolv	RMSE	XGBoost: 1.74	**MPNN: 1.15**
	Lipophilicity	RMSE	XGBoost: 0.799	**GC: 0.655**
Biophysics	PCBA	AUC-PRC	Logreg: 0.129	**GC: 0.136**
	MUV	AUC-PRC	**Multitask: 0.184**	Weave: 0.109
	HIV	AUC-ROC	**KernelSVM: 0.792**	GC: 0.763
	BACE	AUC-ROC	**RF: 0.867**	Weave: 0.806
	PDBbind (full)	RMSE	**RF(grid): 1.25**	GC: 1.44
Physiology	BBBP	AUC-ROC	**KernelSVM: 0.729**	GC: 0.690
	Tox21	AUC-ROC	KernelSVM: 0.822	**GC: 0.829**
	ToxCast	AUC-ROC	Multitask: 0.702	**Weave: 0.742**
	SIDER	AUC-ROC	**RF: 0.684**	GC: 0.638
	ClinTox	AUC-ROC	Bypass: 0.827	**Weave: 0.832**

[a] As discussed in Section 4.4, DTNN outperforms KRR (CM) on 14/16 tasks in QM7b while the mean-MAE is skewed due to different magnitudes of labels.

and physical distances, which are exactly the required input for solving Schrödinger's equation.

5 Conclusions

This work introduces MoleculeNet, a benchmark for molecular machine learning. We gathered data for a wide range of molecular properties: 17 dataset collections including over 800 different tasks on 700 000 compounds. Tasks are categorized into 4 levels as illustrated in Fig. 2: (i) quantum mechanical characters; (ii) physical chemistry properties; (iii) biophysical affinity and activity with bio-macromolecules; (iv) macroscopic physiological effects on human body.

MoleculeNet contributes a data-loading framework, featurization methods, data splitting methods, and learning models to the open source DeepChem package (Fig. 1). By adding interchangeable featurizations, splits and learning models into the DeepChem framework, we can apply these primitives to the wide range of datasets in MoleculeNet.

Broadly, our results show that graph-based models (graph convolutional models, weave models and DTNN) outperform other methods by comfortable margins on most datasets (11/17, best performances comparison in Table 3), revealing a clear advantage of learnable featurizations. However, this effect has some caveats: graph-based methods are not robust enough on complex tasks under data scarcity; on heavily imbalanced classification datasets, conventional methods such as kernel SVM outperform learnable featurizations with respect to recall of positives. Furthermore, for the PDBbind and quantum mechanics datasets, the use of appropriate featurizations which contains pertinent information is very significant. Comparing fully connected neural networks, random forests, and other comparatively simple algorithms, we claim that the PDBbind and QM7 results emphasize the necessity of using specialized features for different tasks. DTNN and MPNN which use distance information perform better on QM datasets than simple graph convolutions. While out of the scope of this paper, we note similarly that customized deep learning algorithms[12] could in principle supplant the need for hand-derived, specialized features in such biophysical settings. On the FreeSolv dataset, comparison between conventional *ab initio* calculations and graph-based models for the prediction of solvation energies shows that data-driven methods can outperform physical algorithms with moderate amounts of data. These results suggest that data-driven physical chemistry will become increasingly important as methods mature. Results for biophysical and physiological datasets are currently weaker than for other datasets, suggesting that better featurizations or more data may be required for data-driven physiology to become broadly useful.

By providing a uniform platform for comparison and evaluation, we hope MoleculeNet will facilitate the development of new methods for both chemistry and machine learning. In future work, we hope to extend MoleculeNet to cover a broader range of molecular properties than considered here. For example, 3D protein structure prediction, or DNA topological modeling would benefit from the presence of strong benchmarks to encourage algorithmic development. We hope that the open-source design of MoleculeNet will encourage researchers to contribute implementations of other novel algorithms to the benchmark suite. In time, we hope to see MoleculeNet grow into a comprehensive resource for the molecular machine learning community.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

We would like to thank the Stanford Computing Resources for providing us with access to the Sherlock and Xstream GPU nodes. Thanks to Steven Kearnes and Patrick Riley for early discussions about the MoleculeNet concept. Thanks to Aarthi Ramsundar for help with diagram construction. Thanks to Zheng Xu for feedback on the MoleculeNet API. Thanks to Patrick Hop for contribution of the lipophilicity dataset to MoleculeNet. Thanks to Anthony Gitter and Johnny Israeli for suggesting the addition of AuPRC for imbalanced datasets. Thanks to Keri McKiernan for composing the tutorial of preparing datasets. The Pande Group is broadly supported by grants from the NIH (R01 GM062868 and U19 AI109662) as well as gift funds and contributions from Folding@home donors. We acknowledge the generous support of Dr Anders G. Frøseth and Mr Christian Sundt for our work on machine learning. B. R. was supported by the Fannie and John Hertz Foundation.

References

1 J. Gasteiger and J. Zupan, *Angew. Chem., Int. Ed.*, 1993, **32**, 503–527.
2 J. Zupan and J. Gasteiger, *Neural networks in chemistry and drug design*, John Wiley & Sons, Inc., 1999.
3 A. Varnek and I. Baskin, *J. Chem. Inf. Model.*, 2012, **52**, 1413–1437.
4 J. B. Mitchell, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2014, **4**, 468–481.
5 J. Devillers, *Neural networks in QSAR and drug design*, Academic Press, 1996.
6 G. Schneider and P. Wrede, *Prog. Biophys. Mol. Biol.*, 1998, **70**, 175–222.
7 Y. LeCun, Y. Bengio and G. Hinton, *Nature*, 2015, **521**, 436–444.
8 J. Schmidhuber, *Neural Network*, 2015, **61**, 85–117.
9 J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl and V. Svetnik, *J. Chem. Inf. Model.*, 2015, **55**, 263–274.
10 B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding and V. Pande, 2015, arXiv preprint arXiv:1502.02072.
11 T. Unterthiner, A. Mayr, G. Klambauer, M. Steijaert, J. Wenger, H. Ceulemans and S. Hochreiter, *Deep Learning and Representation Learning Workshop (NIPS 2014)*, 2014.
12 I. Wallach, M. Dzamba and A. Heifets, 2015, arXiv preprint arXiv:1510.02855.
13 J. S. Delaney, *J. Chem. Inf. Model.*, 2004, **44**, 1000–1005.
14 A. Lusci, G. Pollastri and P. Baldi, *J. Chem. Inf. Model.*, 2013, **53**, 1563–1575.
15 D. L. Mobley, K. L. Wymer, N. M. Lim and J. P. Guthrie, *J. Comput.-Aided Mol. Des.*, 2014, **28**, 135–150.
16 D. L. Mobley and J. P. Guthrie, *J. Comput.-Aided Mol. Des.*, 2014, **28**, 711–720.
17 M. Rupp, A. Tkatchenko, K.-R. Müller and O. A. V. Lilienfeld, *Phys. Rev. Lett.*, 2012, **108**, 058301.
18 G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller and O. A. V. Lilienfeld, *New J. Phys.*, 2013, **15**, 095003.
19 K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller and A. Tkatchenko, 2016, arXiv preprint arXiv:1609.08259.
20 R. T. McGibbon, A. G. Taube, A. G. Donchev, K. Siva, F. Hernández, C. Hargus, K.-H. Law, J. L. Klepeis and D. E. Shaw, *J. Chem. Phys.*, 2017, **147**, 161725.
21 D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.
22 D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik and R. P. Adams, 2015, arXiv preprint arXiv:1509.09292.
23 S. Kearnes, K. McCloskey, M. Berndl, V. Pande and P. Riley, 2016, arXiv preprint arXiv:1603.00856.
24 G. A. Miller, *Commun. ACM*, 1995, **38**, 39–41.
25 J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, *CVPR09*, 2009.
26 O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, *Int. J. Comput. Vis.*, 2015, **115**, 211–252.
27 A. Krizhevsky, I. Sutskever and G. E. Hinton, *NIPS Proceedings*, 2012.
28 C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, 2014, arXiv preprint arXiv:1409.4842.
29 K. He, X. Zhang, S. Ren and J. Sun, 2015, arXiv preprint arXiv:1512.03385.
30 *DeepChem: Deep-learning models for Drug Discovery and Quantum Chemistry*, http://github.com/deepchem/deepchem, accessed 2017-09-27.
31 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
32 M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean and M. Devin, *et al.*, 2016, arXiv preprint arXiv:1603.04467.
33 R. P. Sheridan, *J. Chem. Inf. Model.*, 2013, **53**, 783–790.
34 E. E. Bolton, Y. Wang, P. A. Thiessen and S. H. Bryant, *Annu. Rep. Comput. Chem.*, 2008, **4**, 217–241.
35 T. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, Z. Zhou, L. Han, K. Karapetyan, S. Dracheva, B. A. Shoemaker, E. Bolton, A. Gindulyte and S. H. Bryant, *Nucleic Acids Res.*, 2012, **40**, D400–D412.
36 S. Gražulis, D. Chateigner, R. T. Downs, A. Yokochi, M. Quirós, L. Lutterotti, E. Manakova, J. Butkus, P. Moeck and A. Le Bail, *J. Appl. Crystallogr.*, 2009, **42**, 726–729.
37 C. R. Groom, I. J. Bruno, M. P. Lightfoot and S. C. Ward, *Acta Crystallogr., Sect. B: Struct. Sci., Cryst. Eng. Mater.*, 2016, **72**, 171–179.
38 H. Berman, K. Henrick and H. Nakamura, *Nat. Struct. Mol. Biol.*, 2003, **10**, 980.
39 *Quantum Machine*, http://quantum-machine.org/datasets/, accessed 2017-09-27.
40 D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.
41 L. C. Blum and J.-L. Reymond, *J. Am. Chem. Soc.*, 2009, **131**, 8732–8733.

42 R. Ramakrishnan, M. Hartmann, E. Tapavicza and O. A. V. Lilienfeld, *J. Chem. Phys.*, 2015, **143**, 084111.
43 L. Ruddigkeit, R. V. Deursen, L. C. Blum and J.-L. Reymond, *J. Chem. Inf. Model.*, 2012, **52**, 2864–2875.
44 R. Ramakrishnan, P. O. Dral, M. Rupp and O. A. V. Lilienfeld, *Sci. Data*, 2014, **1**, 140022.
45 Experimental in vitro DMPK and physicochemical data on a set of publicly disclosed compounds.
46 S. G. Rohrer and K. Baumann, *J. Chem. Inf. Model.*, 2009, **49**, 169–184.
47 *AIDS Antiviral Screen Data*, http://wiki.nci.nih.gov/display/NCIDTPdata/AIDS+Antiviral+Screen+Data, accessed 2017-09-27.
48 R. Wang, X. Fang, Y. Lu and S. Wang, *J. Med. Chem.*, 2004, **47**, 2977–2980.
49 R. Wang, X. Fang, Y. Lu, C.-Y. Yang and S. Wang, *J. Med. Chem.*, 2005, **48**, 4111–4119.
50 Z. Liu, Y. Li, L. Han, J. Li, J. Liu, Z. Zhao, W. Nie, Y. Liu and R. Wang, *Bioinformatics*, 2014, **31**, 405–412.
51 G. Subramanian, B. Ramsundar, V. Pande and R. A. Denny, *J. Chem. Inf. Model.*, 2016, **56**, 1936–1949.
52 I. F. Martins, A. L. Teixeira, L. Pinheiro and A. O. Falcao, *J. Chem. Inf. Model.*, 2012, **52**, 1686–1697.
53 *Tox21 Challenge*, http://tripod.nih.gov/tox21/challenge/, accessed 2017-09-27.
54 A. M. Richard, R. S. Judson, K. A. Houck, C. M. Grulke, P. Volarath, I. Thillainadarajah, C. Yang, J. Rathman, M. T. Martin, J. F. Wambaugh, T. B. Knudsen, J. Kancherla, K. Mansouri, G. Patlewicz, A. J. Williams, S. B. Little, K. M. Crofton and R. S. Thomas, *Chem. Res. Toxicol.*, 2016, **29**, 1225–1251.
55 M. Kuhn, I. Letunic, L. J. Jensen and P. Bork, *Nucleic Acids Res.*, 2016, **44**, D1075–D41079.
56 H. Altae-Tran, B. Ramsundar, A. S. Pappu and V. Pande, 2016, arXiv preprint arXiv:1611.03199.
57 *Medical Dictionary for Regulatory Activities*, http://www.meddra.org/, accessed 2017-09-27.
58 K. M. Gayvert, N. S. Madhukar and O. Elemento, *Cell Chem. Biol.*, 2016, **23**, 1294–1301.
59 A. V. Artemov, E. Putin, Q. Vanhaelen, A. Aliper, I. V. Ozerov and A. Zhavoronkov, *bioRxiv, Biochem.*, 2016, 095653.
60 P. A. Novick, O. F. Ortiz, J. Poelman, A. Y. Abdulhay and V. S. Pande, *PLoS One*, 2013, **8**(11), e79568.
61 *Aggregate Analysis of ClincalTrials.gov (AACT) Database*, http://www.ctti-clinicaltrials.org/aact-database, accessed 2017-09-27.
62 G. W. Bemis and M. A. Murcko, *J. Med. Chem.*, 1996, **39**, 2887–2893.
63 G. Landrum, *RDKit: Open-Source Cheminformatics Software*, http://www.rdkit.org/.
64 A. N. Jain and A. Nicholls, *J. Comput.-Aided Mol. Des.*, 2008, **22**, 133–139.
65 T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2009.
66 J. Davis and M. Goadrich, *Proceedings of the 23rd International Conference on Machine Learning*, 2006.
67 R. Gómez-Bombarelli, D. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, 2016, arXiv preprint arXiv:1610.02415.
68 J. D. Durrant and J. A. McCammon, *J. Chem. Inf. Model.*, 2011, **51**, 2897–2903.
69 C. Da and D. Kireev, *J. Chem. Inf. Model.*, 2014, **54**, 2555–2561.
70 J. Behler and M. Parrinello, *Phys. Rev. Lett.*, 2007, **98**, 146101.
71 J. S. Smith, O. Isayev and A. E. Roitberg, 2016, arXiv preprint arXiv:1610.08935.
72 L. Breiman, *Mach. Learn.*, 2001, **45**, 5–32.
73 J. H. Friedman, *Ann. Stat.*, 2001, 1189–1232.
74 B. Ramsundar, B. Liu, Z. Wu, A. Verras, M. Tudor, R. P. Sheridan and V. Pande, *J. Chem. Inf. Model.*, 2017, **57**(8), 2068–2076.
75 S. J. Swamidass, C.-A. Azencott, T.-W. Lin, H. Gramajo, S.-C. Tsai and P. Baldi, *J. Chem. Inf. Model.*, 2009, **49**, 756–766.
76 J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals and G. E. Dahl, 2017, arXiv preprint arXiv:1704.01212.
77 J. Friedman, T. Hastie, R. Tibshirani, *et al.*, *Ann. Stat.*, 2000, **28**, 337–407.
78 C. Cortes and V. Vapnik, *Mach. Learn.*, 1995, **20**, 273–297.
79 T. Chen and C. Guestrin, 2016, arXiv preprint arXiv:1603.02754.
80 S. Kearnes, B. Goldman and V. Pande, 2016, arXiv preprint arXiv:1606.08793.
81 I. I. Baskin, V. A. Palyulin and N. S. Zefirov, *J. Chem. Inf. Comput. Sci.*, 1997, **37**, 715–721.
82 D. B. Kireev, *J. Chem. Inf. Comput. Sci.*, 1995, **35**, 175–180.
83 O. Vinyals, S. Bengio and M. Kudlur, 2015, arXiv preprint arXiv:1511.06391.