Molecular Graph Convolutions: Moving Beyond Fingerprints

S. Kearnes, V. Pande, *et al.*

Reese Pathak **Anvita Gupta**

11/14/16, CS273B

Outline

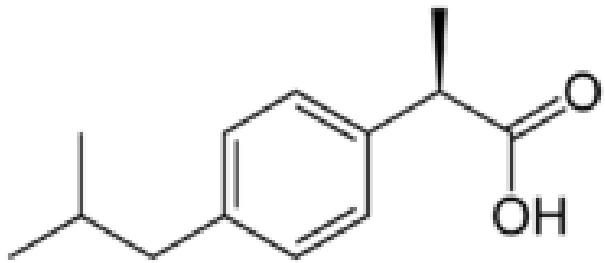
Why computational drug design?

- experimental techniques (alone) are labor intensive, time consuming, expensive
- computational techniques have predicted better drug candidates (HIV, flu, *etc.*)
- places focus on biological understanding of disease
- turns drug design into a data problem
- hastens drug discovery timeline

Background (1)

terminology for the molecular graph

- recall your (high school) chemistry



Ibuprofen

- electronic features, **bond degree**, **donator/acceptor**, *etc.*
- complex structural features, **aromaticity**, **resonance**, *etc.*
- three dimensional features, **chirality**, **bond angle**, *etc.*

Background (2)

example featurization of cheminformatic data

- Morgan fingerprint (Rogers & Hahn 2010)

Input: a chemical structure with k atoms

1. each atom receives a unique integer identifier in $\{1, 2, \ldots, k\}$
2. for each atom, store current neighbor identifiers in array
3. for each atom hash array values to a single new integer identifier
4. go back to step 1 (repeat 2-3 times)

- modifications exist for considering bond degree, isotopes, charge, electronic activity, *etc.*
- yes, this (crude) algorithm really is the state of the art!

Outline

Problem formulation

- **Given:** chemical compounds and associated structural information
- **Predict:** whether or not a drug candidate will successfully bind a target protein
- **computational view** is multi-task binary classification

Dataset

- Same dataset from Ramsundar *et al.* , 2016
- 259 assays against different target proteins from 4 different sources:
  1. *PCBA* - Pubchem Bioassay
  2. *MUV* - Maximum Unbiased Validation
  3. *DuDE* - Database of Useful Decoy Drugs
  4. *Tox21* - Training data from Tox21 Challenge
- 60-20-20 split between training, validation, and test sets
- Some things to consider:
  - Class imbalance?
  - Duplicates (or very similar compounds) in training and test?

Outline

Model Setup

- **Goal:** learn higher order features from Molecular Graph
- Successive convolution of atom and pair features, followed by ReLU activations.
- Results in a new molecular graph with (hopefully) better learned features

Molecular graph model

Model takes in multiple inputs for a single molecule

- a $N \times n_a$ matrix of atom-specific features, A
- a $N \times N \times n_p$ tensor of pairwise features, P
- N, the number of atoms in molecule
- n_a, the number of features computed per atom
- n_p, the number of features aggregated per pair of atoms

Input Featurization (1)

- Featurization of Atoms by:

Feature	Description	Size
Atom type[a]	H, C, N, O, F, P, S, Cl, Br, I, or metal (one-hot or null)	11
Chirality	R or S (one-hot or null)	2
Formal charge	Integer electronic charge	1
Partial charge	Calculated partial charge	1
Ring sizes	For each ring size (3–8), the number of rings that include this atom	6
Hybridization	sp, sp^2, or sp^3 (one-hot or null)	3
Hydrogen bonding	Whether this atom is a hydrogen bond donor and/or acceptor (binary values)	2
Aromaticity	Whether this atom is part of an aromatic system	1
		27

Input Featurization (2)

- Featurization of Atoms by:

Feature	Description	Size
Atom type[a]	H, C, N, O, F, P, S, Cl, Br, I, or metal (one-hot or null)	11
Chirality	R or S (one-hot or null)	2
Formal charge	Integer electronic charge	1
Partial charge	Calculated partial charge	1
Ring sizes	For each ring size (3–8), the number of rings that include this atom	6
Hybridization	sp, sp^2, or sp^3 (one-hot or null)	3
Hydrogen bonding	Whether this atom is a hydrogen bond donor and/or acceptor (binary values)	2
Aromaticity	Whether this atom is part of an aromatic system	1
		27

Input Featurization (3)

- Featurization of Pair Information (Bonds) by:

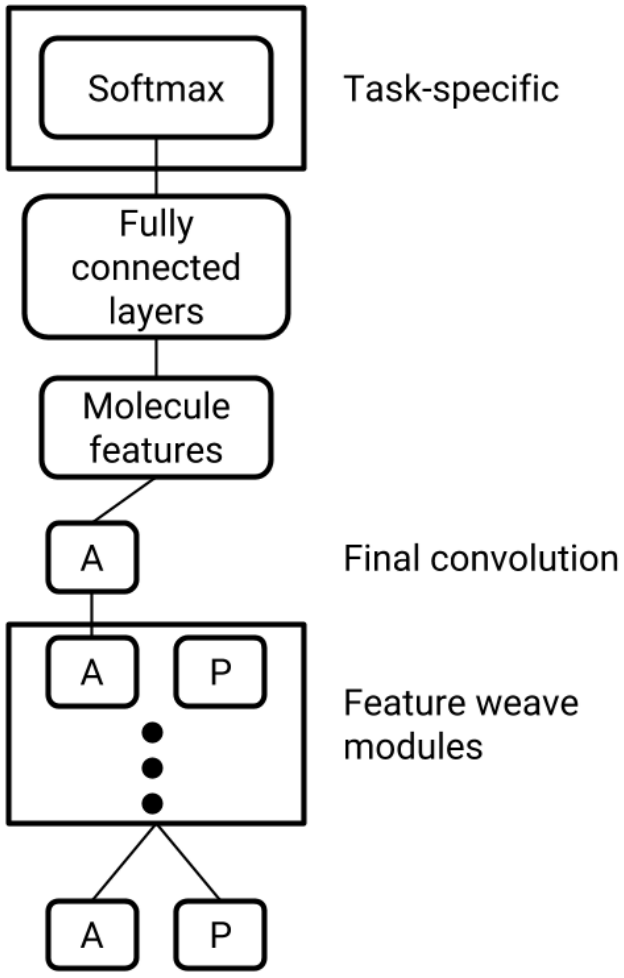
Feature	Description	Size
Bond type[a]	Single, double, triple, or aromatic (one-hot or null)	4
Graph distance[a]	For each distance (1–7), whether the shortest path between the atoms in the pair is less than or equal to that number of bonds (binary values)	7
Same ring	Whether the atoms in the pair are in the same ring	1
		12

Input Featurization (4)

- Featurization of Pair Information (Bonds) by:

Feature	Description	Size
Bond type[a]	Single, double, triple, or aromatic (one-hot or null)	4
Graph distance[a]	For each distance (1–7), whether the shortest path between the atoms in the pair is less than or equal to that number of bonds (binary values)	7
Same ring	Whether the atoms in the pair are in the same ring	1
		12

Model architecture



- A and P are manually featurized inputs
- Weave modules are modified convolutional layers
- only final atom features are used for the final FC layers
  - the rationale for this is unclear, technical issue?
- technical details:
  - trained with Adagrad, batch size 96, learning rate of 0.003
  - TensorFlow

Model Constraints

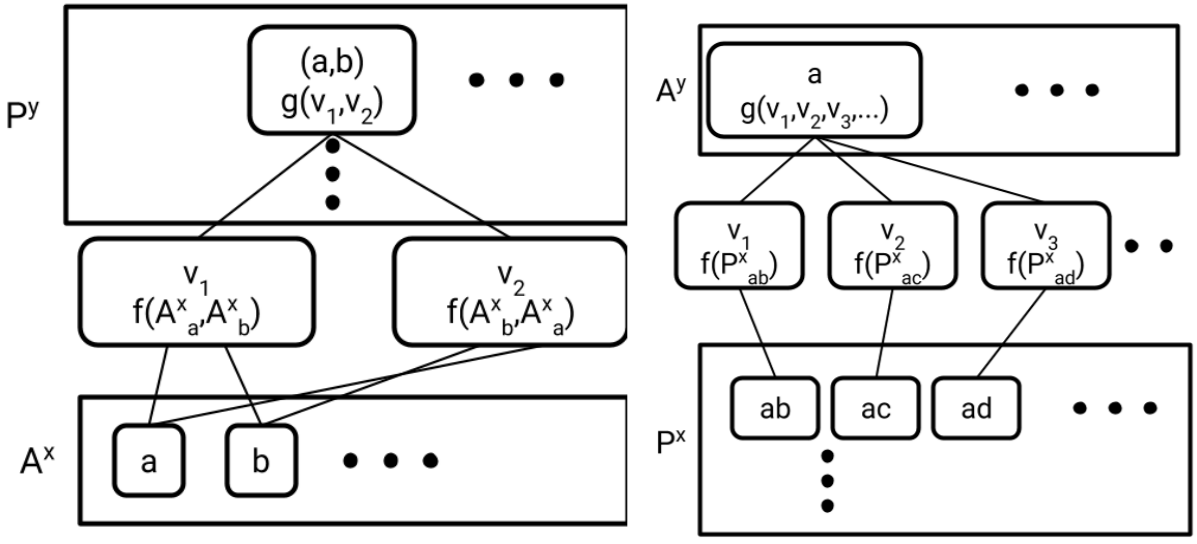
- Atom Table A and Pair Table P should always satisfy a few constraints:
  1. **Property 1** Output of model should be *invariant* to the order that the atom and bond information is encoded in the input.
  2. **Property 2** If the inputs are permuted with a permutation operator Q, then at every layer x, y, A^x and P^y are permuted with Q as well
     - Invariance to order of atom/bond information continues at every level
  3. **Property 3** Symmetry of Pair Values– for all layers y, $P^y_{(a,b)} = P^y_{(b,a)}$

Invariant Preserving Operations (1)

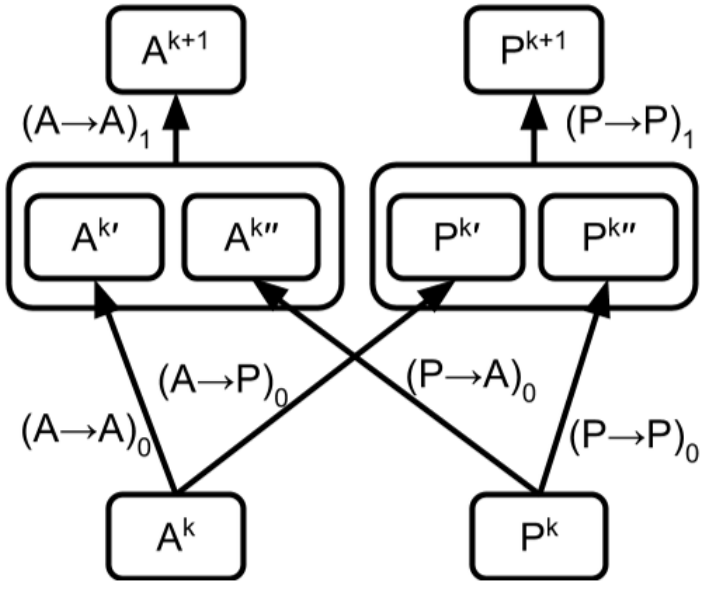
- Goal is to compute A and P at new level y based on previous levels $x_1, x_2, ...x_n$
- $A \rightarrow A$
  - For a particular atom a, calculate row in new A matrix:
  - ${A_a}^y = f({A_a}^{x_1}, {A_a}^{x_2}, ..., {A_a}^{x_n})$
- $P \rightarrow P$
  - ${P_{a,b}}^y = f({P_{a,b}}^{x_1}, {P_{a,b}}^{x_2}, ..., {P_{a,b}}^{x_n})$
- Function f and weights are same for every atom– referred to as a convolution!

Invariant Preserving Operations (2)

- $P \rightarrow A$
  - ${A_a}^y = g(f({P_{a,b}}^x), f({P_{a,c}}^x), ..., f({P_{a,N}}^x)) = \sum_i f({P_{a,i}}^x)$
  - Marginalize over all atoms that a bonds to
- $A \rightarrow P$
  - ${P_{a,b}}^y = g(f({A_a}^x), f({A_b}^x)))$
    - Better than simply adding the features from ${A_a}^x$ to ${A_b}^x$ because gives learnable parameters

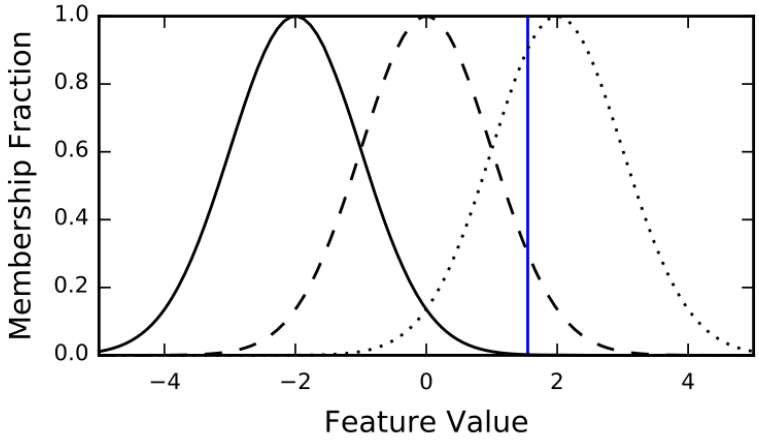


Constructing Weave Module



- Weave module is composed of at least one of each of these operations
- Construction ensures that atom and pair features from one level both contribute to atom and pair features of next level

Molecular Level Features



- Take 11 Histograms (one for each feature, each data point is an atom)
- Concatenate this histograms into set of features
- Fuzzy Set
  - membership in each bin given by how much of distribution lies to its right

Outline

Model evaluation metrics

Comparison against four baselines:

- maximum similarity (Jaccard similarity),
- logistic regression,
- random forest model,
- pyramid multitask neural network (PMTNN)

Classification accuracy metrics:

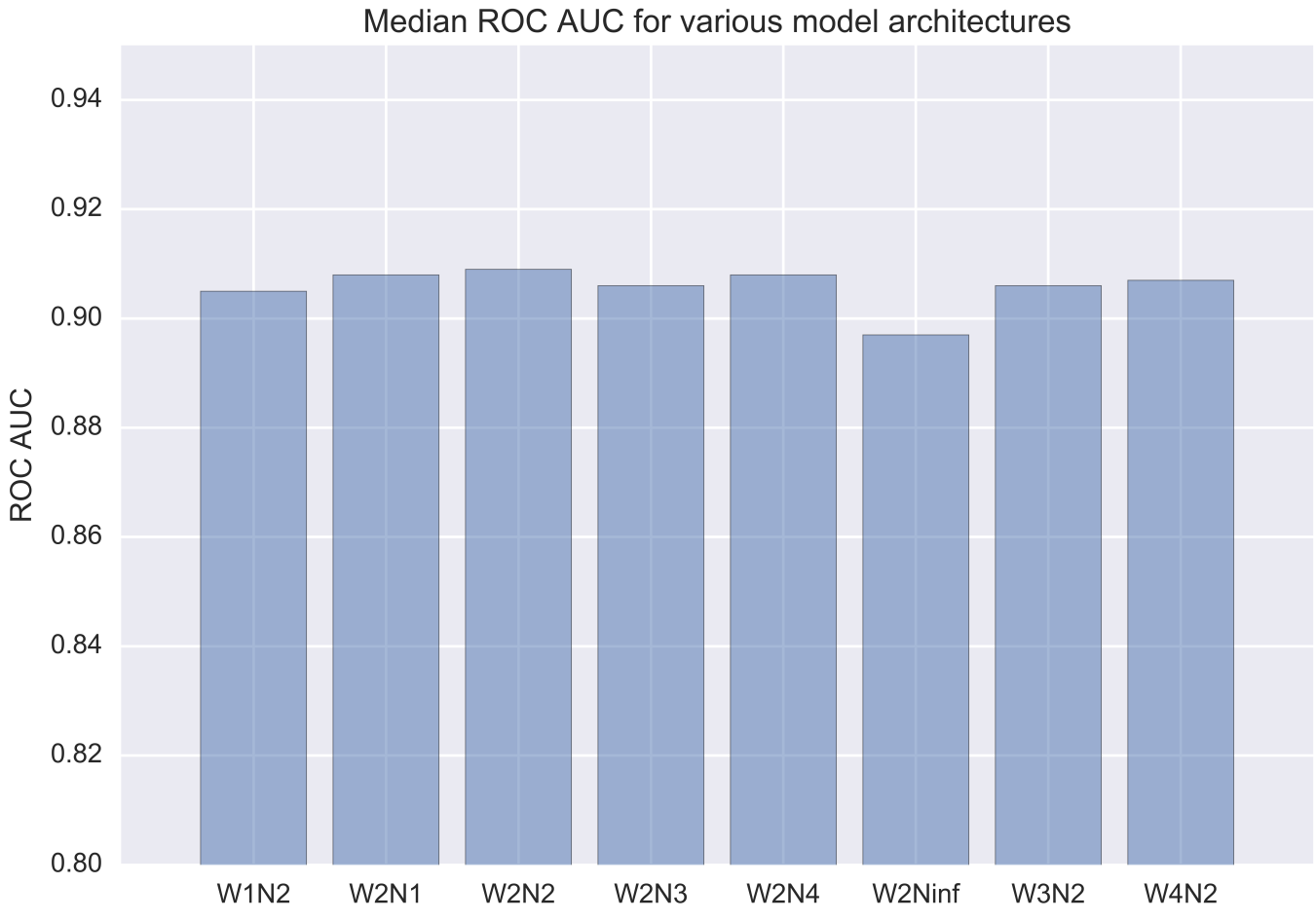
- median ROC AUC
- Δ median ROC AUC (relative to PMTNN)
- 95% Wilson score interval

Model performance

Authors: "our model is *statistically indistinguishable* from the PMTNN for the PCBA, MUV, and Tox21 dataset groups" (emphasis added)

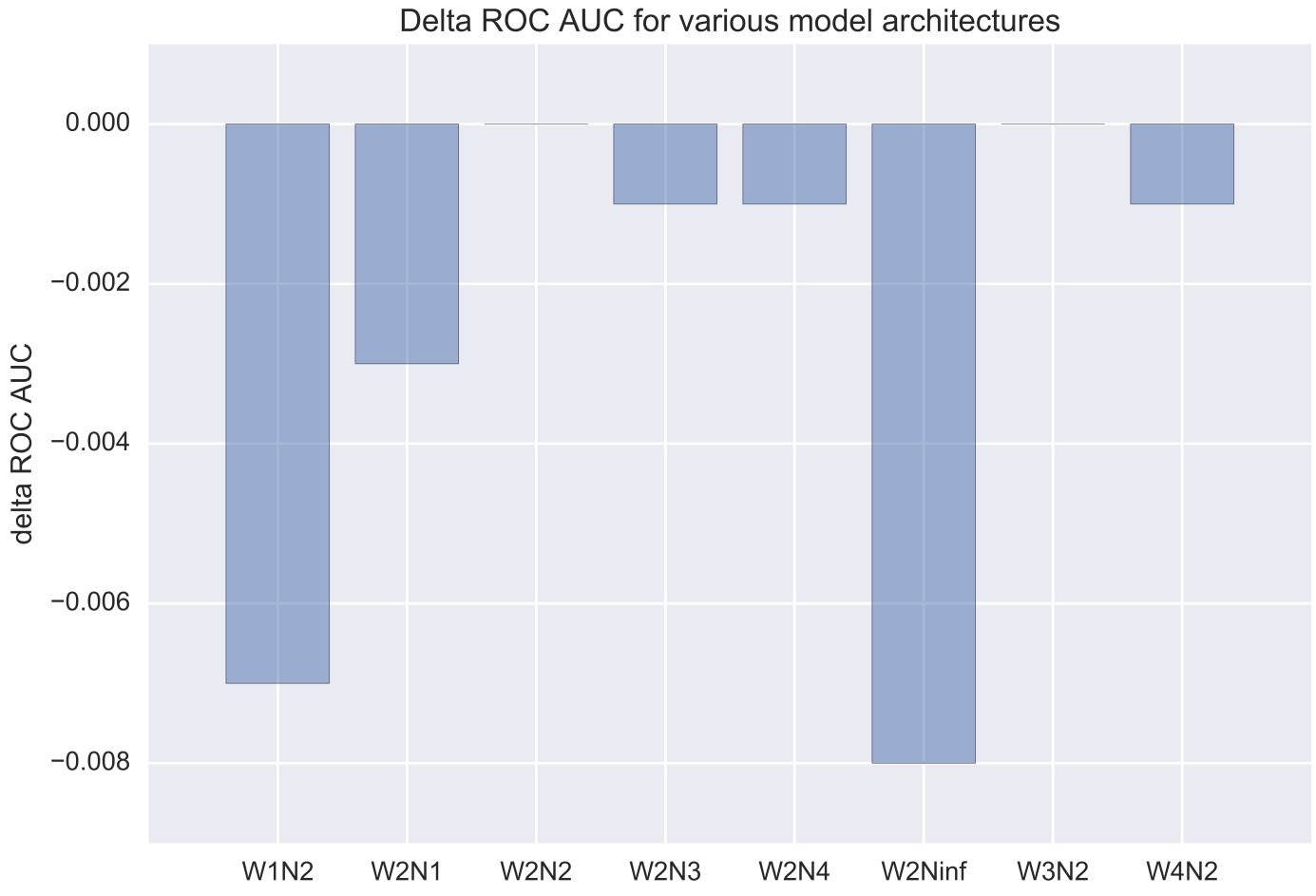
Model	PCBA ($n = 128$)			MUV ($n = 17$)			Tox21 ($n = 12$)		
	Median AUC	Median ΔAUC	Sign Test 95% CI	Median AUC	Median ΔAUC	Sign Test 95% CI	Median AUC	Median ΔAUC	Sign Test 95% CI
MaxSim	0.754	−0.137	**(0.00, 0.04)**	0.638	−0.136	**(0.01, 0.27)**	0.728	−0.131	**(0.00, 0.24)**
LR	0.838	−0.059	**(0.04, 0.13)**	0.736	−0.070	**(0.10, 0.47)**	0.789	−0.073	**(0.01, 0.35)**
RF	0.804	−0.092	**(0.02, 0.10)**	0.655	−0.135	**(0.01, 0.27)**	0.802	−0.047	**(0.01, 0.35)**
PMTNN	0.905			0.869			0.854		
W_2N_2-simple	0.905	−0.003	**(0.27, 0.44)**	0.849	0.012	(0.36, 0.78)	0.866	0.003	(0.39, 0.86)
W_2N_2-sum	0.898	−0.011	**(0.16, 0.31)**	0.818	−0.014	(0.17, 0.59)	0.848	−0.010	(0.09, 0.53)
W_2N_2-RMS	0.902	−0.007	**(0.20, 0.35)**	0.851	−0.026	(0.13, 0.53)	0.854	−0.007	**(0.05, 0.45)**
W_1N_2	0.905	−0.007	**(0.20, 0.35)**	0.840	−0.002	(0.26, 0.69)	0.849	−0.009	(0.09, 0.53)
W_2N_1	0.908	−0.003	**(0.30, 0.46)**	0.858	−0.016	(0.17, 0.59)	0.867	−0.002	(0.19, 0.68)
W_2N_2	0.909	0.000	(0.42, 0.59)	0.847	−0.004	(0.22, 0.64)	0.862	0.004	(0.32, 0.81)
W_2N_3	0.906	−0.001	(0.38, 0.55)	0.838	−0.013	(0.26, 0.69)	0.861	0.000	(0.25, 0.75)
W_2N_4	0.908	−0.001	(0.37, 0.54)	0.836	−0.008	(0.17, 0.59)	0.858	0.001	(0.39, 0.86)
W_2N_∞	0.897	−0.008	**(0.12, 0.25)**	0.841	−0.025	**(0.10, 0.47)**	0.846	−0.006	(0.14, 0.61)
W_3N_2	0.906	0.000	(0.44, 0.61)	0.875	0.010	(0.31, 0.74)	0.859	0.004	(0.47, 0.91)
W_4N_2	0.907	−0.001	(0.33, 0.50)	0.856	−0.007	(0.22, 0.64)	0.862	0.004	(0.32, 0.81)

Deeper networks share similar performance statistics

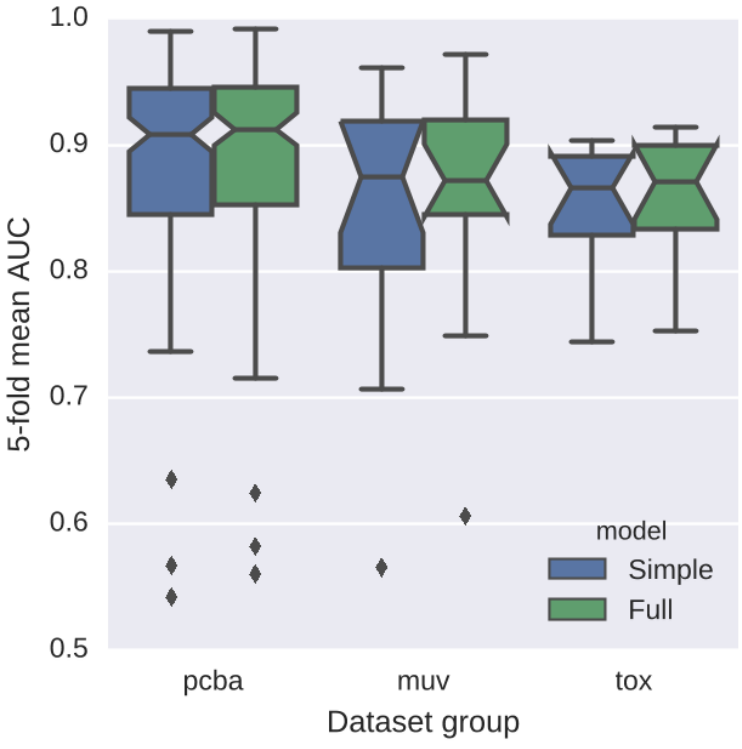


No architecture improves the SOTA

no significant improvement with deeper architectures

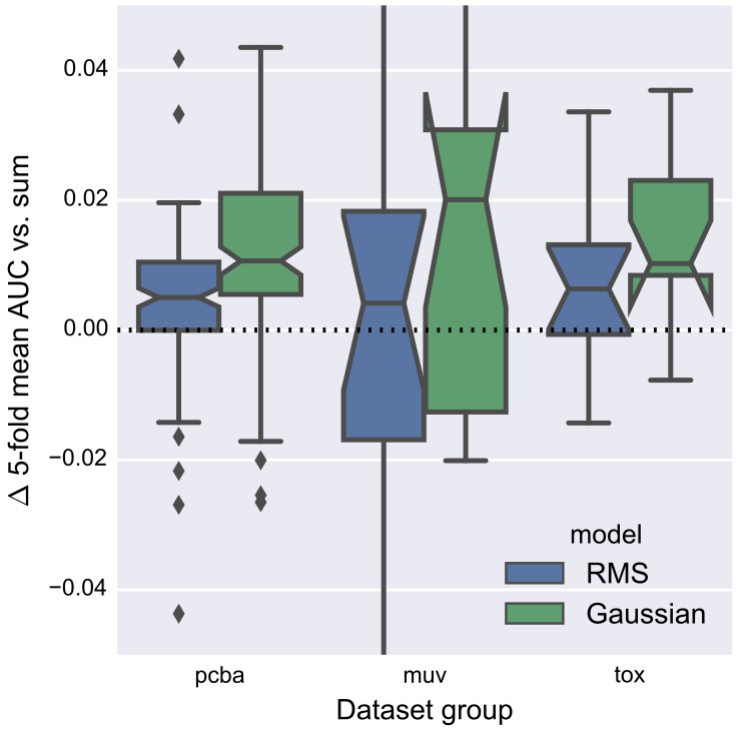


Simple vs. Full Featurization



- Adding the full feature set gives little improvement
- Atom type and bond type seem to give majority of predictive capability
- Why better than Morgan Fingerprint?

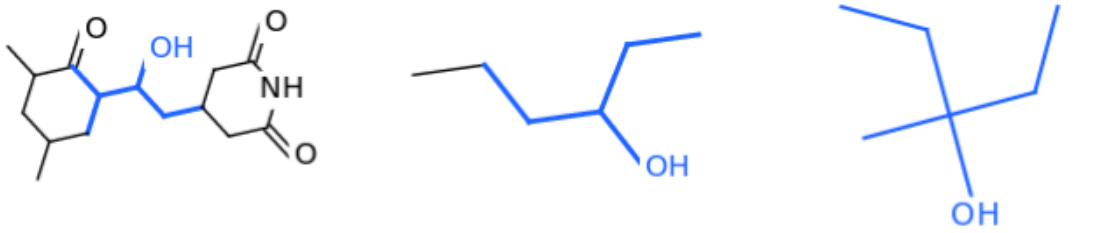
Loss of Information in Molecular Level Features



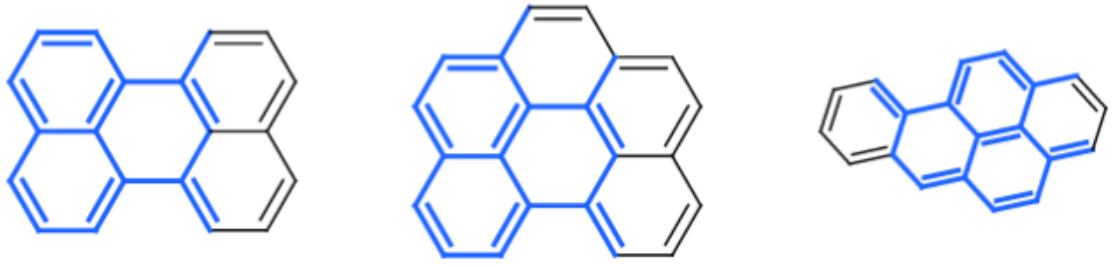
- Performs slightly better with respect to Δ median auROC
- Not as much better as we would expect with such a complicated binning technique
- Why is this step needed? Artificially imposed on CNN, which should be able to learn and predict from its own generated features

Interpretability of Features

Fragments most activated by pro-solubility feature



Fragments most activated by anti-solubility feature



Neural Fingerprints Duvenaud et. al *2015*

- Comparable visualization for the Weave modules would help test if model is actually learning

Class imbalance

some of the data offered by the authors indicate potential class imbalance in the data set

- the authors get > 0.98 ROC AUC on *all* models on the DUD-E data set
- logistic regression gets at least 0.84 ROC AUC on data sets
- authors do not report PRC AUC
- authors do *not* describe removing examples from train set based on molecular similarity

Suggestions for Extensions

- Might avoid artificial combining of features with RNN
  - Accepts variable length input!
- Would like to see more detail on which specific proteins model did better/worse on
- Extension to 3D model of drugs? Basic principles would apply
- Feature importance testing (we suspect features are not very predictive)