Convolutional Networks on Graphs for Learning Molecular Fingerprints

Authors: David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre
Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, Ryan P. Adams

Presenter: Ishaan Kumar

Overview

- In this work we look at the problem of generating embeddings (*fingerprints*) for a molecule for various downstream tasks. More info on these tasks later on.
- Specifically, we take an existing method and use differentiable components so that we can learn task specific embeddings.

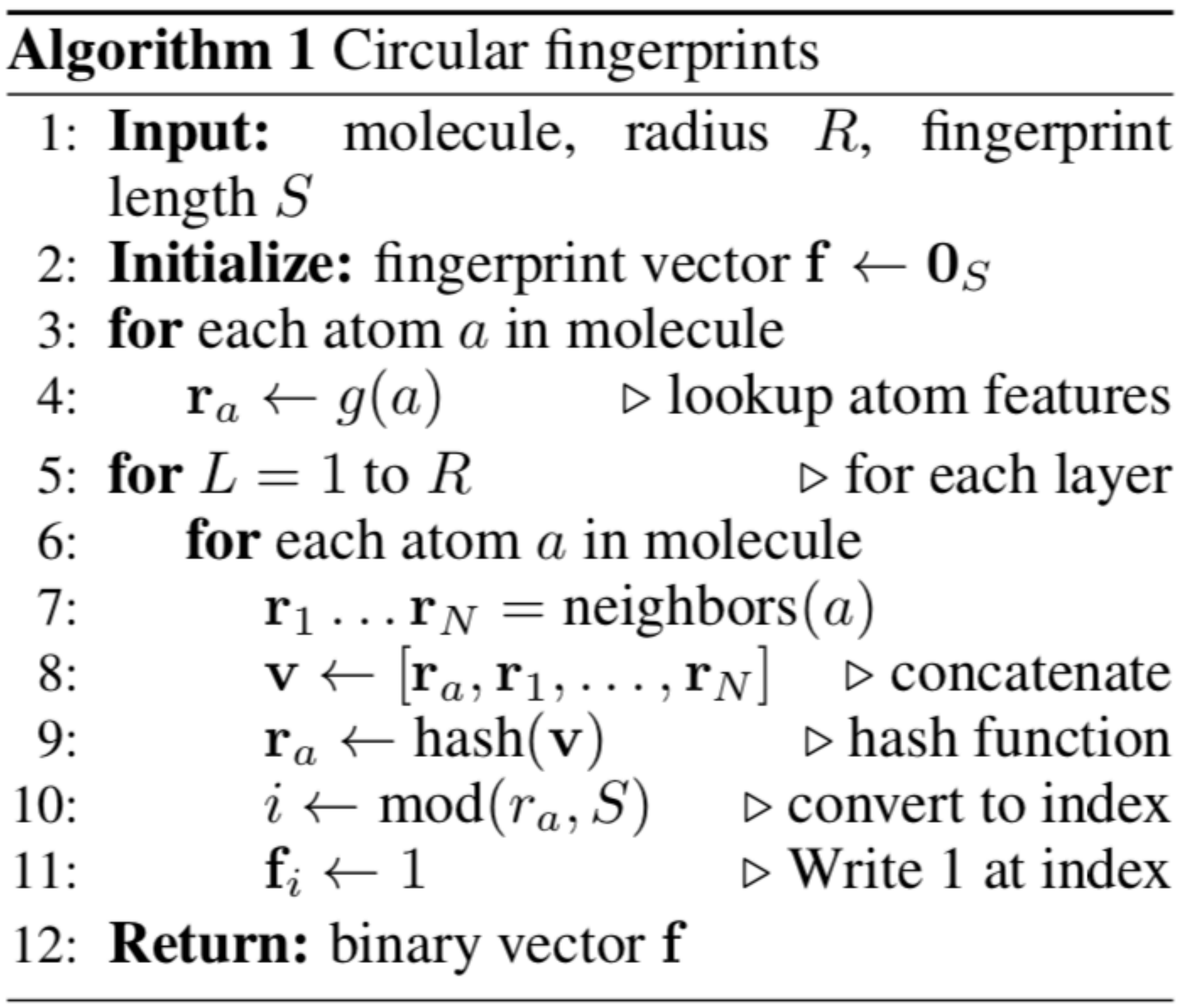
Advantages

- Predictive performance: Machine-optimized fingerprints have better predictive performance than fixed fingerprints.
- Parsimony: Fixed fingerprints must be extremely large to encode all possible substructures without overlap. Neural fingerprints can be optimized to encode only relevant features, reducing downstream computation and regularization requirements.
- Interpretability: No notion of similarity in fixed fingerprints. Neural fingerprint feature can be activated by similar but distinct molecular fragments.

Current approach

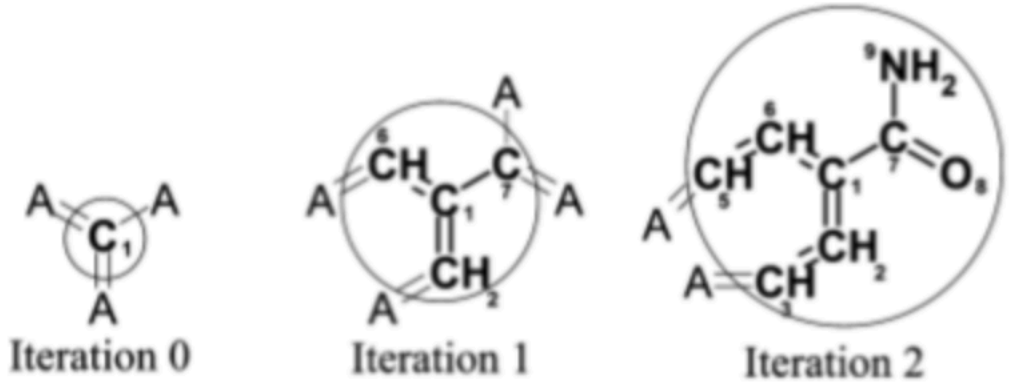
- Use off-the-shelf fingerprint software to generate fixed-len feature vector for an arbitrary sized molecule.
- Feed the generated feature vector to a neural network.
- We focus on SOTA fingerprint generation method - Extended-connectivity circular fingerprints (ECFP)
- Circular fingerprints encode which substructures are present in a molecule such that the encoding process is invariant to atom-relabelling.

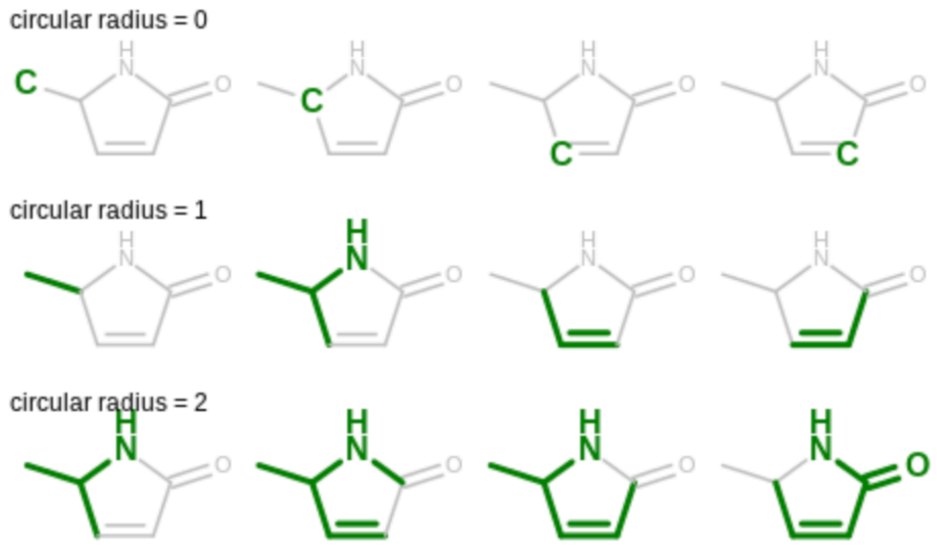
Current approach



Algorithm 1 Circular fingerprints

```
1:  Input: molecule, radius $R$, fingerprint length $S$
2:  Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
3:  for each atom $a$ in molecule
4:      $\mathbf{r}_a \leftarrow g(a)$                ▷ lookup atom features
5:  for $L = 1$ to $R$                              ▷ for each layer
6:      for each atom $a$ in molecule
7:          $\mathbf{r}_1 \ldots \mathbf{r}_N = \text{neighbors}(a)$
8:          $\mathbf{v} \leftarrow [\mathbf{r}_a, \mathbf{r}_1, \ldots, \mathbf{r}_N]$   ▷ concatenate
9:          $\mathbf{r}_a \leftarrow \text{hash}(\mathbf{v})$             ▷ hash function
10:         $i \leftarrow \text{mod}(r_a, S)$             ▷ convert to index
11:         $\mathbf{f}_i \leftarrow 1$                    ▷ Write 1 at index
12: Return: binary vector $\mathbf{f}$
```





Top image: Computing Extended Connectivity Fingerprints | Bottom image: source

Current approach

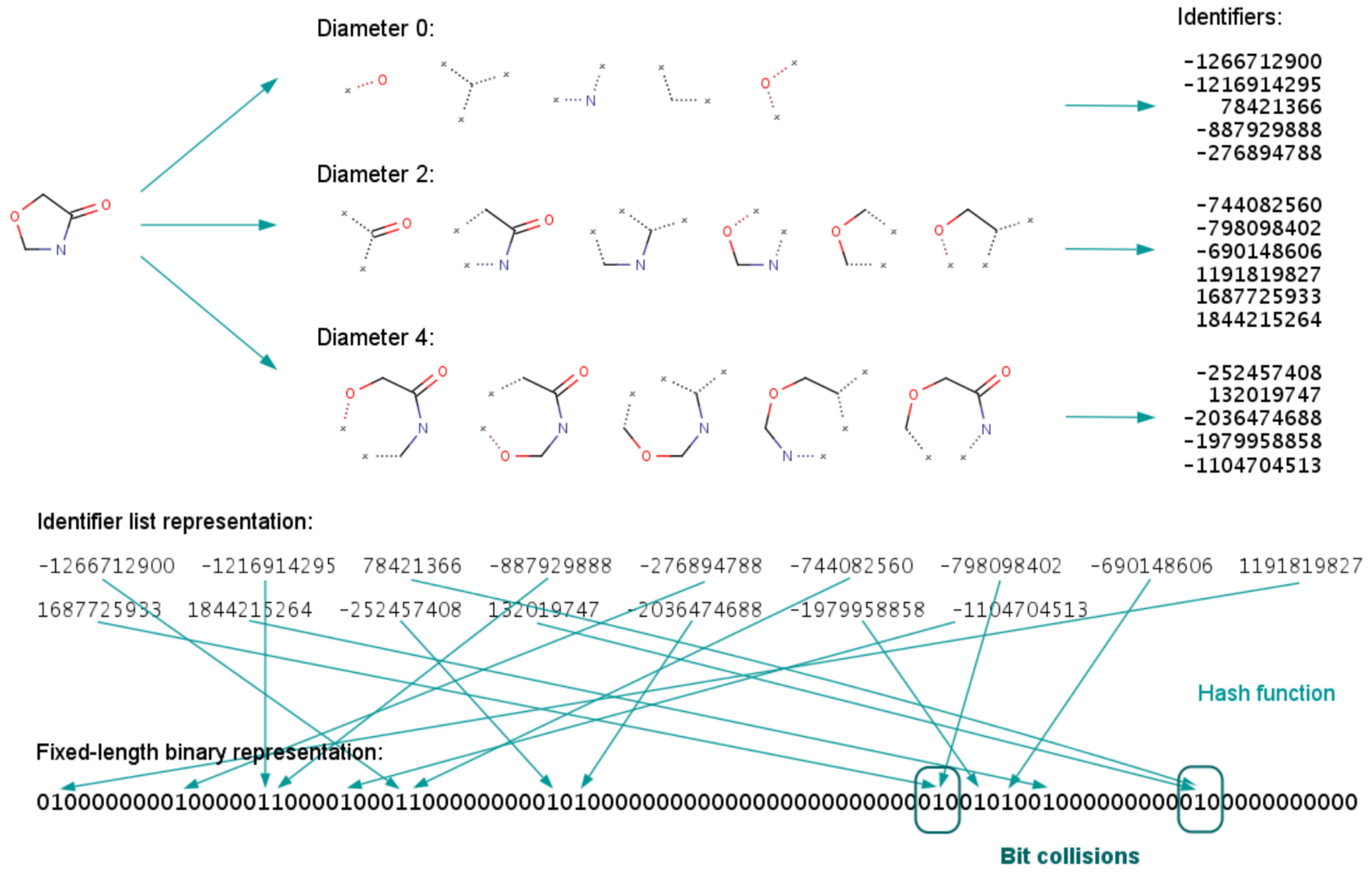
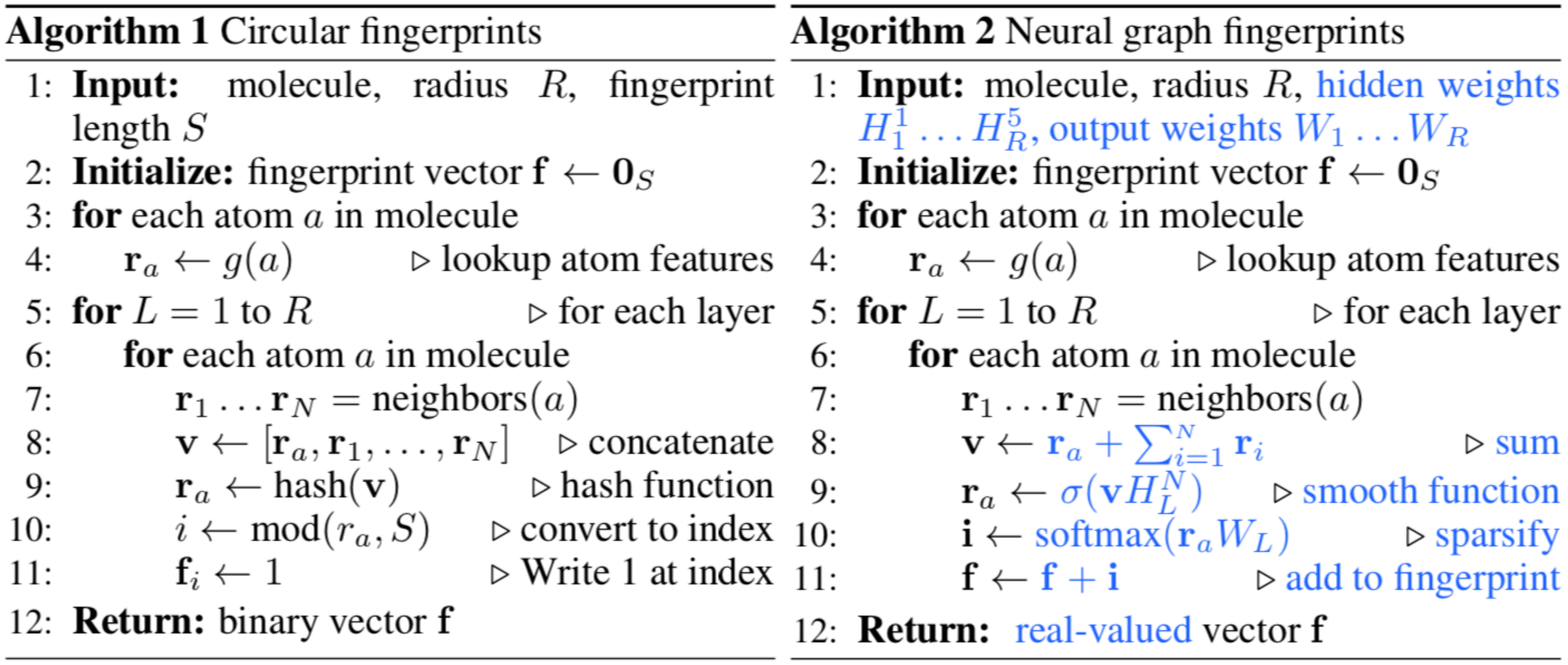


Image source: Extended-Connectivity Fingerprints

Proposed approach

- We start with existing approach and replace its discrete operations with a differentiable analog
- Hashing: Replace hash function with a single-layered neural network.
- Indexing: Replace with a softmax function and the sum of all atoms probabilities is the final fingerprint.
- Canonicalization: Circular fingerprints are atom-order invariant as they sort the neighbour atoms according to their features and bond features. We simply use summation to achieve permutation invariance.

Proposed approach



Algorithm 1 Circular fingerprints

```
1: Input: molecule, radius $R$, fingerprint length $S$
2: Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
3: for each atom $a$ in molecule
4:     $\mathbf{r}_a \leftarrow g(a)$                         ▷ lookup atom features
5: for $L = 1$ to $R$                                         ▷ for each layer
6:     for each atom $a$ in molecule
7:         $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
8:         $\mathbf{v} \leftarrow [\mathbf{r}_a, \mathbf{r}_1, \dots, \mathbf{r}_N]$   ▷ concatenate
9:         $\mathbf{r}_a \leftarrow \text{hash}(\mathbf{v})$               ▷ hash function
10:        $i \leftarrow \text{mod}(r_a, S)$                   ▷ convert to index
11:        $\mathbf{f}_i \leftarrow 1$                          ▷ Write 1 at index
12: Return: binary vector $\mathbf{f}$
```

Algorithm 2 Neural graph fingerprints

```
1: Input: molecule, radius $R$, hidden weights $H_1^1 \dots H_R^5$, output weights $W_1 \dots W_R$
2: Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
3: for each atom $a$ in molecule
4:     $\mathbf{r}_a \leftarrow g(a)$                         ▷ lookup atom features
5: for $L = 1$ to $R$                                         ▷ for each layer
6:     for each atom $a$ in molecule
7:         $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
8:         $\mathbf{v} \leftarrow \mathbf{r}_a + \sum_{i=1}^{N} \mathbf{r}_i$            ▷ sum
9:         $\mathbf{r}_a \leftarrow \sigma(\mathbf{v} H_L^N)$              ▷ smooth function
10:        $\mathbf{i} \leftarrow \text{softmax}(\mathbf{r}_a W_L)$             ▷ sparsify
11:        $\mathbf{f} \leftarrow \mathbf{f} + \mathbf{i}$                  ▷ add to fingerprint
12: Return: real-valued vector $\mathbf{f}$
```

Equivalence

- Circular fingerprints are a special case of Neural fingerprints with large weights.
- In the limit of large weights, tanh approach step functions which when concatenated form a simple hash function.
- In the limit of large input weights, the softmax operator approaches a 1-hot encoded argmax operator which is analogous to an indexing operation.

Equivalence Experiments

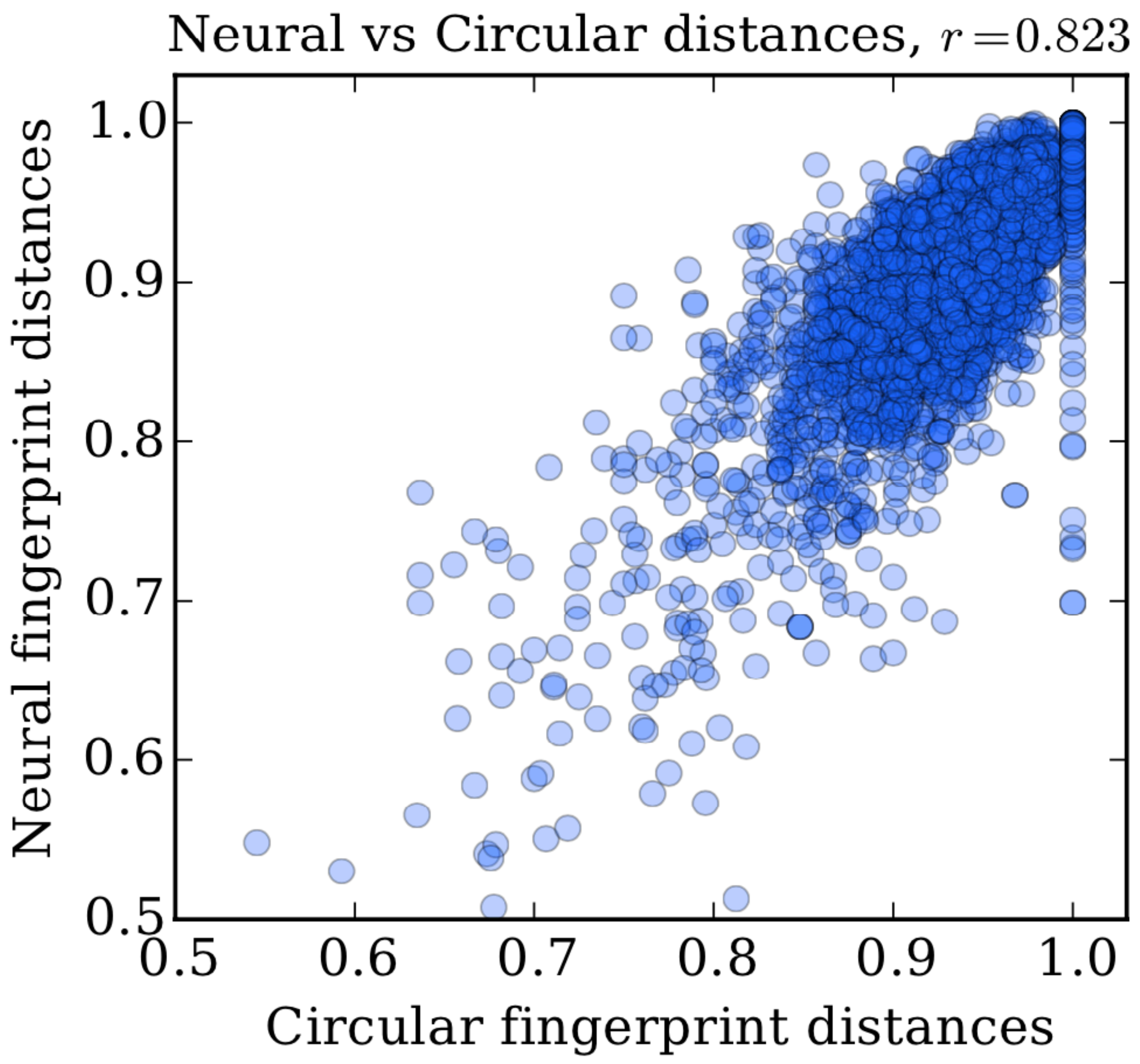
- Compare distances between circular fingerprints to distances between neural fingerprints with large random weights. Use continuous generalization of Jaccard similarity

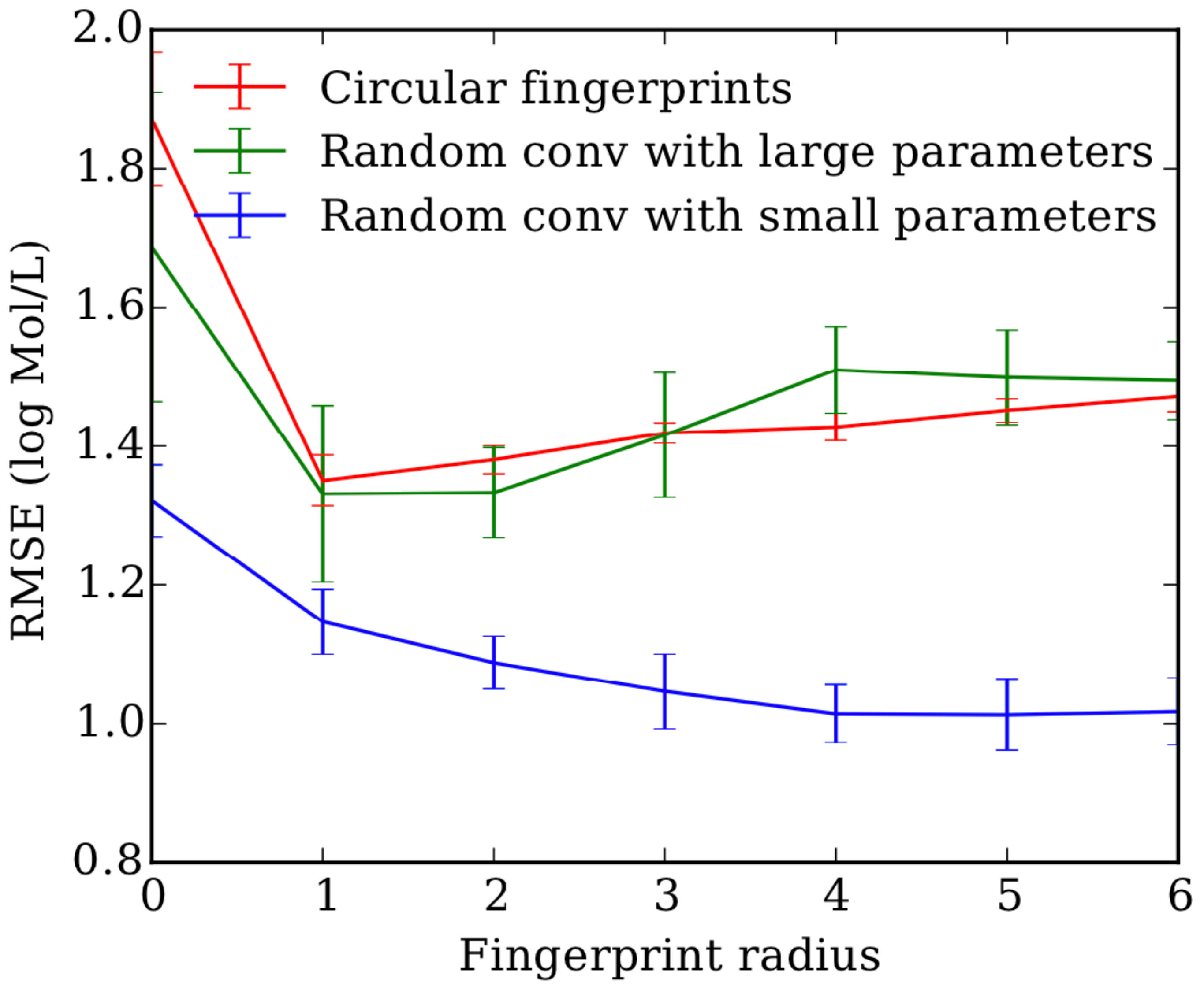
$$\mathrm{distance}(\mathbf{x}, \mathbf{y}) = 1 - \sum \min(x_i, y_i) \Big/ \sum \max(x_i, y_i)$$

- Compare predictive performance of neural fingerprints with large random weights on solubility prediction task.

Equivalence Results

There is a correlation of r = 0.823 between the distances. The line of points on the right of the plot shows that for some pairs of molecules, binary ECFP fingerprints have exactly zero overlap

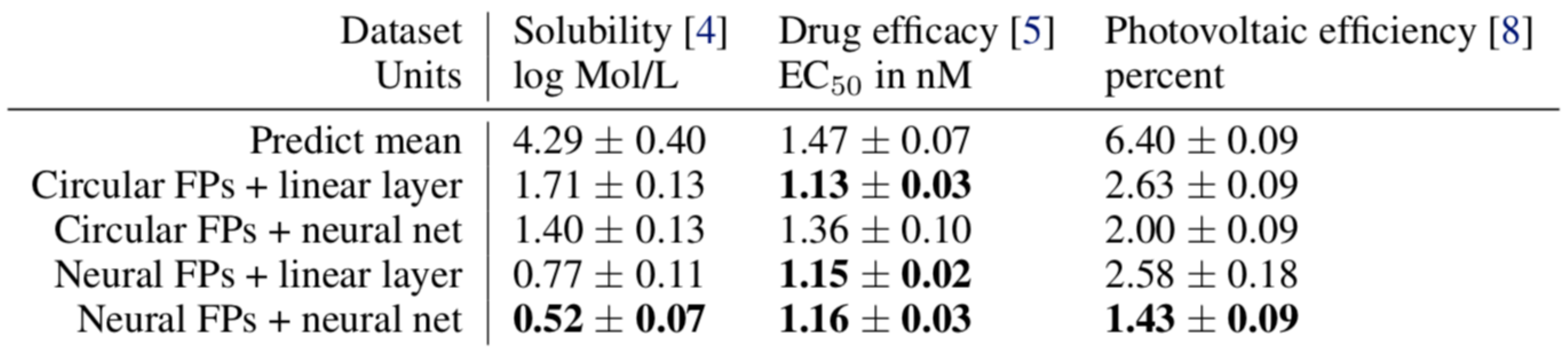




Predictive Performance

- Compare predictive performance of standard circular fingerprints against neural graph fingerprints on the following tasks/domains:
  - Solubility: Aqueous solubility of molecules.
  - Drug efficacy: The half-maximal effective concentration (EC_{50}) *in vitro* of 10,000 molecules against a sulfide-resistant strain of *P. falciparum*, the parasite that causes malaria.
  - Organic photovoltaic efficiency: A subset of 20,000 molecules from Harvard Clean Energy Project that uses expensive DFT simulations to estimate the photovoltaic efficiency of organic molecules.

Predictive Performance

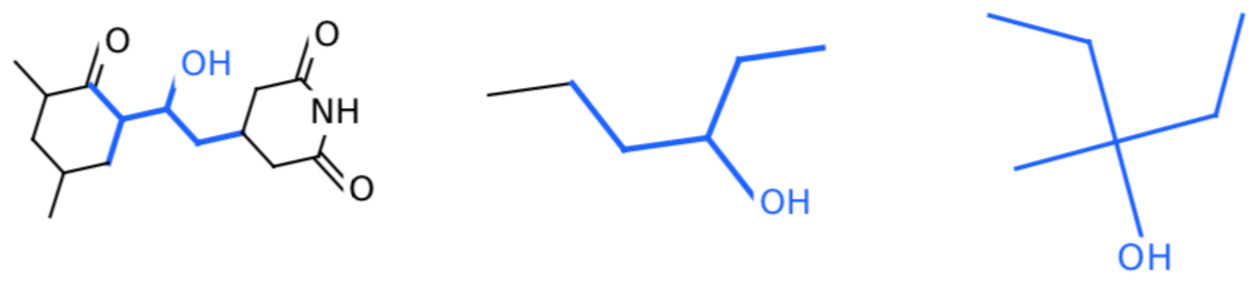


Dataset Units	Solubility [4] log Mol/L	Drug efficacy [5] EC_{50} in nM	Photovoltaic efficiency [8] percent
Predict mean	4.29 ± 0.40	1.47 ± 0.07	6.40 ± 0.09
Circular FPs + linear layer	1.71 ± 0.13	$\mathbf{1.13 \pm 0.03}$	2.63 ± 0.09
Circular FPs + neural net	1.40 ± 0.13	1.36 ± 0.10	2.00 ± 0.09
Neural FPs + linear layer	0.77 ± 0.11	$\mathbf{1.15 \pm 0.02}$	2.58 ± 0.18
Neural FPs + neural net	$\mathbf{0.52 \pm 0.07}$	$\mathbf{1.16 \pm 0.03}$	$\mathbf{1.43 \pm 0.09}$

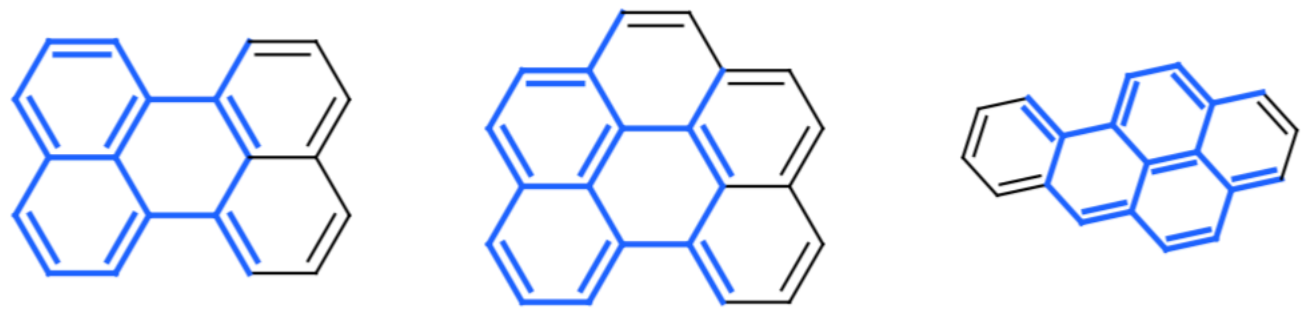
Table 1: Mean predictive accuracy of neural fingerprints compared to standard circular fingerprints.

Interpretability

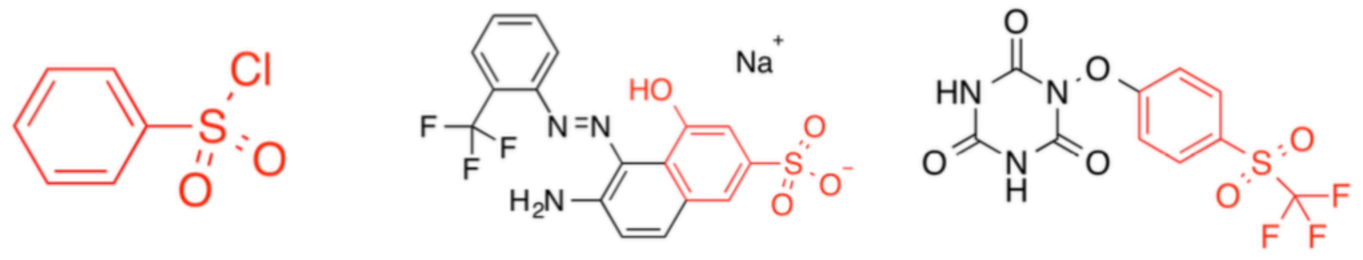
Fragments most activated by pro-solubility feature



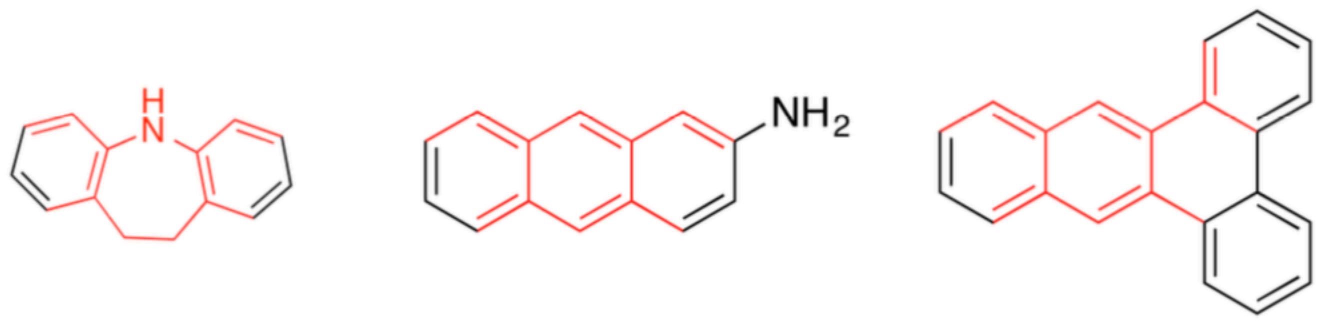
Fragments most activated by anti-solubility feature



Fragments most activated by toxicity feature on SR-MMP dataset



Fragments most activated by toxicity feature on NR-AHR dataset



Limitations

- Computational cost
- Limited computation at each layer
- Limited information propagation across the graph
- Inability to distinguish stereoisomers

Questions