# Advancing Energy Storage through Solubility Prediction: Leveraging the Potential of Deep Learning

Mesfin Diro Chaka [a,c,*], Yedilfana Setarge Mekonnen [b], Qin Wu [d], Chernet Amente Geffe [a]

[a] *Department of Physics, College of Natural and Computational Sciences, Addis Ababa University, P. O. Box 1176, Addis Ababa, Ethiopia,*

[b] *Center for Environmental Science, College of Natural and Computational Sciences, Addis Ababa University, P. O. Box 1176, Addis Ababa, Ethiopia,*

[c] *Computational Data Science Program, College of Natural and Computational Sciences, Addis Ababa University, P. O. Box 1176, Addis Ababa, Ethiopia,*

[d] *Center for Functional Nanomaterials, Brookhaven National Laboratory, Upton, NY 11973, USA,*

[*] Corresponding author
*Email address:* mesfin.diro@aau.edu.et (Mesfin Diro Chaka)

*October 19, 2023*

## 1. AqSolDB Dataset

Table S 1: Randomly sampled smiles strings with their corresponding LogS in AqSolDB dataset

| smiles | Experimental LogS(Mol/L) |
|---|---|
| O=C([O-])CO.[NH4+] | 1.701 |
| OCCCCCCO | 1.626 |
| CC(N)=O | 1.581 |
| CNC | 1.558 |
| CN | 1.369 |
| C1CN1 | 1.366 |
| CC=O | 1.356 |
| NC=O | 1.346 |
| CNN | 1.337 |
| c1cn[nH]c1 | 1.288 |
| C#CCO | 1.251 |
| CC1CN1 | 1.243 |
| C=CCN | 1.243 |
| NC1CC1 | 1.243 |
| O=CC=O | 1.236 |
| C1COC1 | 1.236 |
| CCO | 1.234 |
| CNC=O | 1.229 |
| CC(C)N | 1.228 |
| CCCN | 1.228 |
| NCCN | 1.221 |
| CN(C)N | 1.221 |
| NCCO | 1.214 |
| OCCO | 1.207 |
| OCCF | 1.193 |
| CN(C)C | 1.178 |
| O=CO | 1.176 |
| Clc1cc(Cl)c(Cl)c(-c2c(Cl)c(Cl)c(Cl)c(Cl)c2Cl)c1Cl | -10.412 |
| Clc1cc(Oc2cc(Cl)c(Cl)c(Cl)c2Cl)c(Cl)c(Cl)c1Cl | -10.100 |
| Clc1cc(Oc2c(Cl)c(Cl)cc(Cl)c2Cl)c(Cl)c(Cl)c1Cl | -10.100 |
| Clc1cc(Cl)c(Oc2c(Cl)c(Cl)c(Cl)c(Cl)c2Cl)cc1Cl | -10.100 |
| Clc1cc(-c2c(Cl)c(Cl)c(Cl)c(Cl)c2Cl)cc(Cl)c1Cl | -9.700 |
| Clc1ccc(Oc2c(Cl)c(Cl)c(Cl)c(Cl)c2Cl)c(Cl)c1 | -9.640 |
| Clc1cc(Oc2cc(Cl)c(Cl)c(Cl)c2Cl)cc(Cl)c1Cl | -9.540 |
| Clc1cc(Cl)c(-c2c(Cl)c(Cl)c(Cl)c(Cl)c2Cl)cc1Cl | -9.500 |

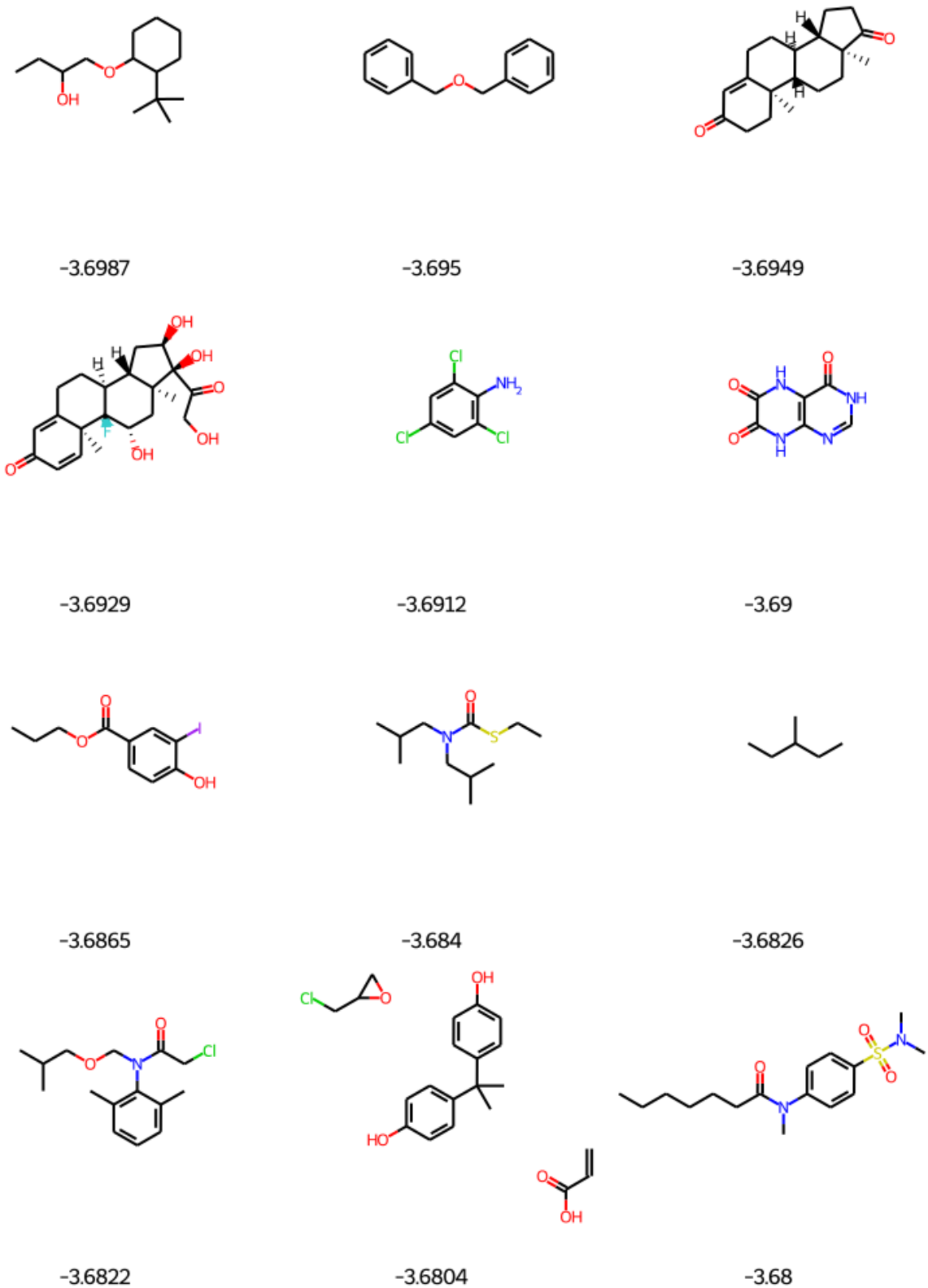



Figure S 1: Some of randomly sampled molecules in AqSolDB dataset

## 2. Explanantory Analysis



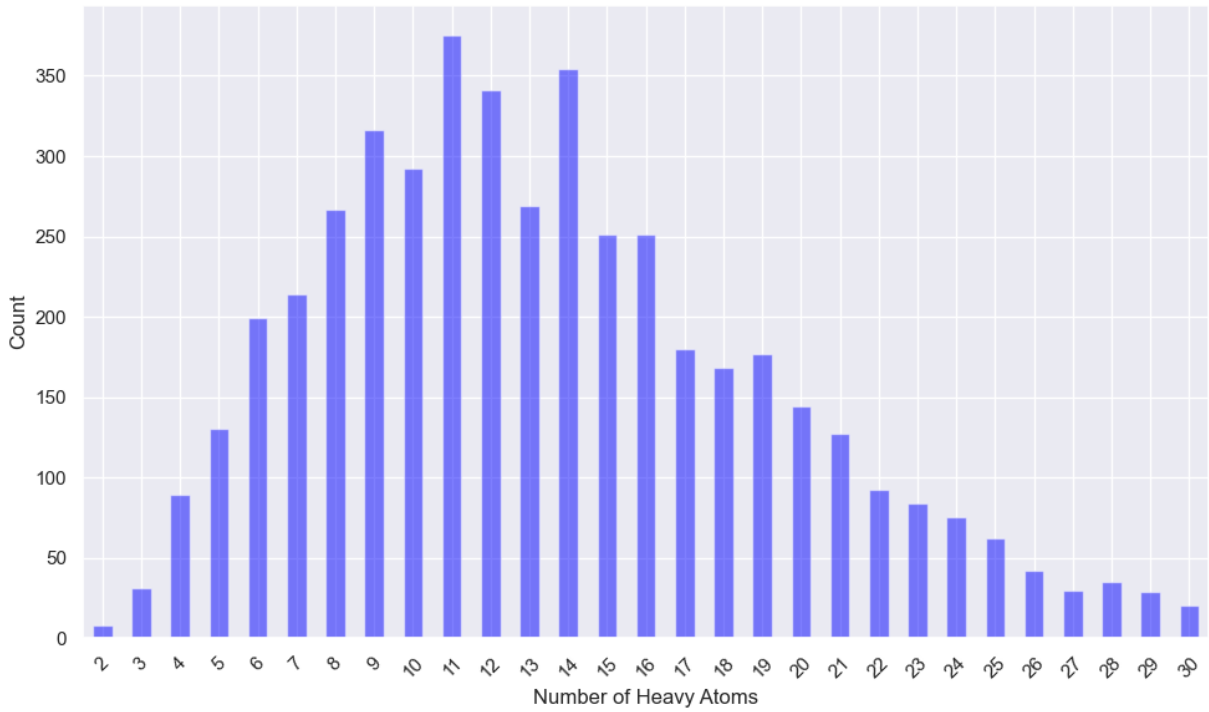
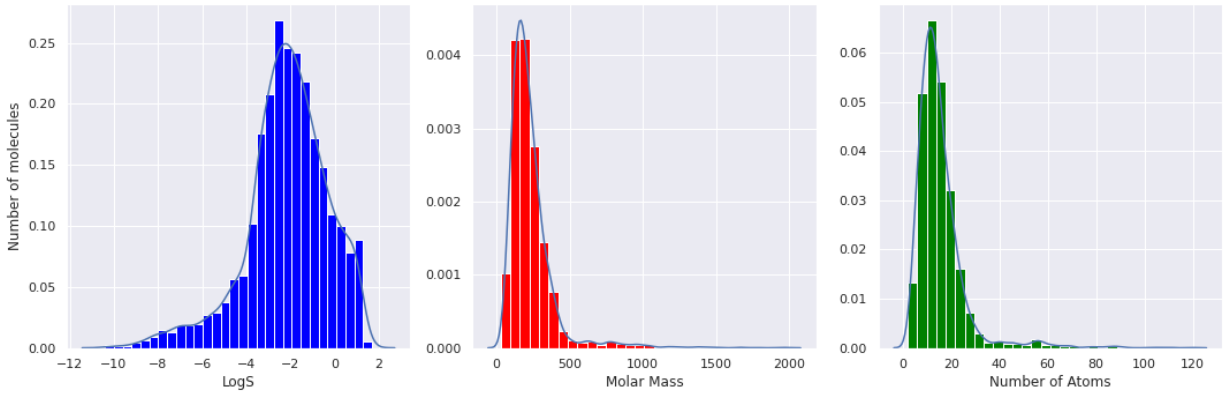



Figure S 2: The total number of heavy atoms in AqSolDB dataset used for the training




(a)

Figure S 3: Histogram of LogS, Molar mass and Numbber of atoms in AqSolDB dataset

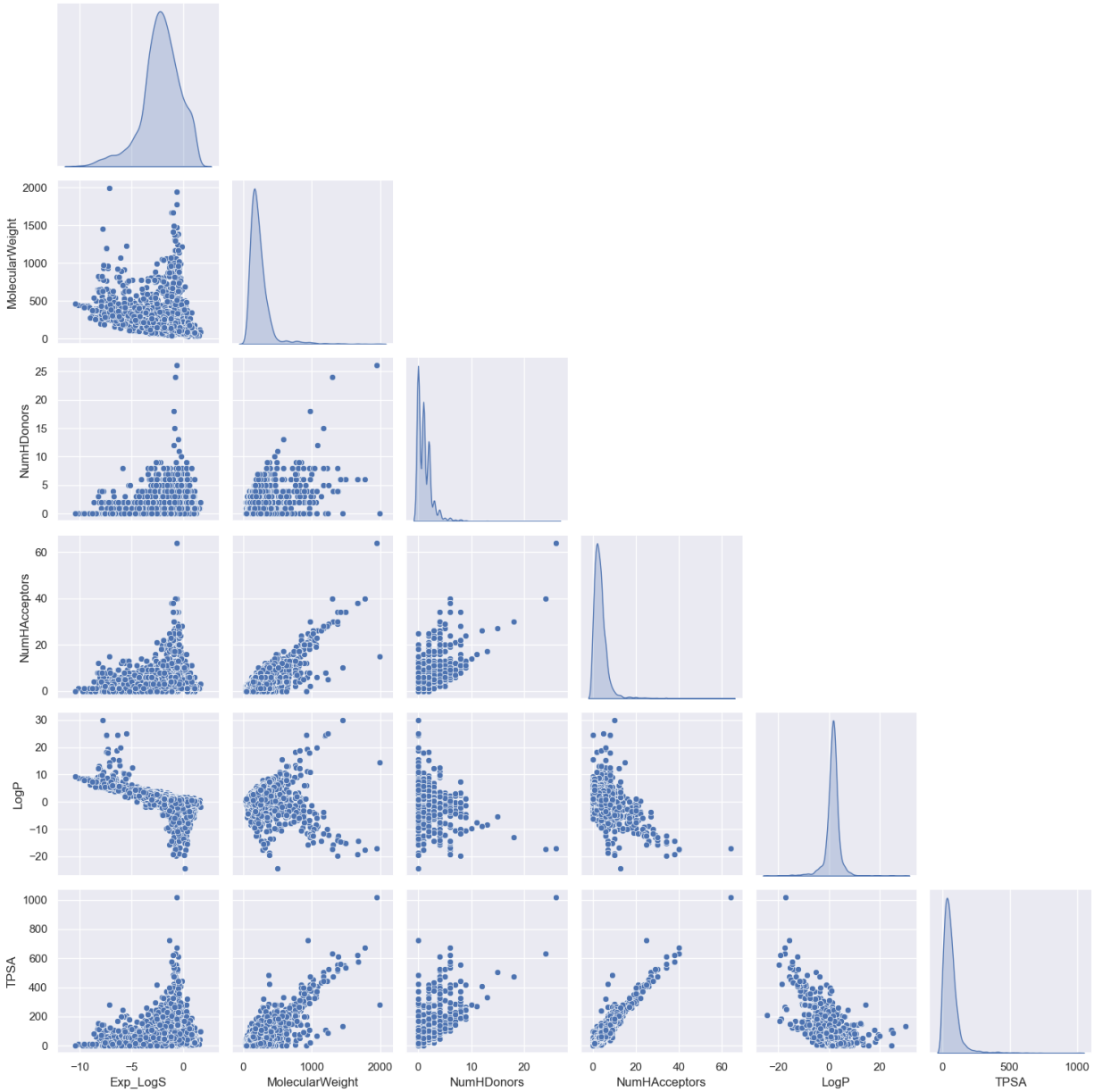
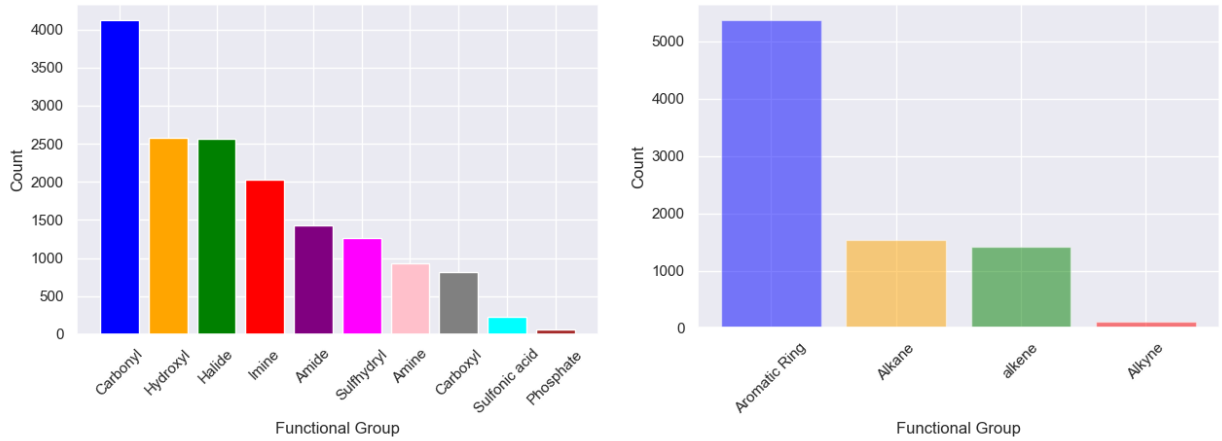



Figure S 4: Pair Plot of Selected Descriptors and Experimental LogS




(a) (b)

Figure S 5: The distribution of dominant functional groups in AqSolDB dataset a) polar functional groups b) non-polar functional groups

### *2.1. Pytorch_geometeric(Pyg) graph representation of AqSolDB from the SMILE Strings*

```
import os.path as osp
from custom_pygdata import  AqSolDB
path = osp.join(osp.dirname(osp.realpath('__file__')),  'data', 'aqsoldb')
mol_sol = AqSolDB(root_dir=path,
                  name='AqSolDB2.csv',
                  smi_idx=-4,
                  target_idx=-3).shuffle()
```

```
Data(x=[290455, 30], edge_index=[2, 590090], edge_attr=[590090, 12], y=[8494, 1], smiles=[8494])
```

### *2.2. Target variable normalized to mean = 0 and std =1*

```
r_mean = mol_sol.data.y.mean()
r_std = mol_sol.data.y.std()
mol_sol.data.y = (mol_sol.data.y - r_mean) / r_std
print("Normalized LogS:\n",mol_sol.data.y)
```

```
Normalized LogS:
 tensor([[ 0.8059],
        [-0.0870],
        [ 0.0549],
        ...,
        [-1.8041],
        [-0.1992],
        [-1.3752]])
```

### 2.3. Sample atomic attributes

```
Sample atomic features in Pyg graph format:
 tensor([[0.0549, 0.0000, 0.0000,  ..., 0.0464, 0.3704, 0.2632],
        [0.0549, 0.0000, 0.0000,  ..., 0.0464, 0.3704, 0.2632],
        [0.0549, 0.0000, 0.0000,  ..., 0.0464, 0.3704, 0.2632],
        ...,
        [0.0000, 0.0000, 1.0000,  ..., 0.0000, 0.0000, 0.0000],
        [0.0000, 0.0000, 1.0000,  ..., 0.0000, 0.0000, 0.0000],
        [0.0000, 0.0000, 1.0000,  ..., 0.0000, 0.0000, 0.0000]])
```

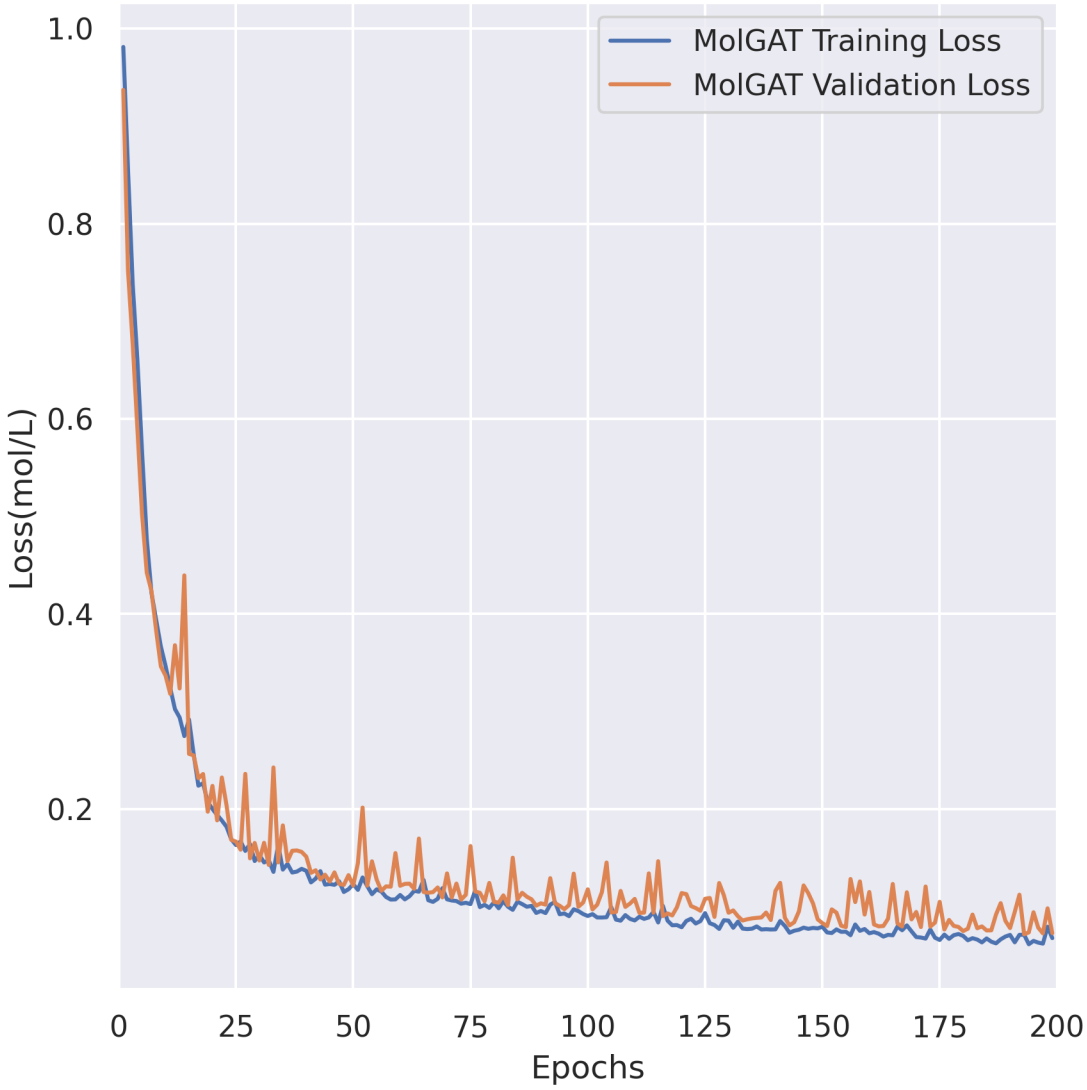
### 2.4. Sample edge attributes

```
Sample edge features in Pyg graph format:
 tensor([[0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [1., 0., 0.,  ..., 0., 0., 0.],
        ...,
        [1., 0., 0.,  ..., 0., 0., 0.],
        [1., 0., 0.,  ..., 0., 0., 0.],
        [1., 0., 0.,  ..., 0., 0., 0.]])
```

## 3. MolGAT Model Implementation

```
MolGAT(
  (conv_list): ModuleList(
    (0): MolGATConv(30, 192, 12, heads=4)
    (1-2): 2 x MolGATConv(192, 192, 12, heads=4)
  )
  (fc_list): ModuleList(
    (0): Linear(in_features=384, out_features=384, bias=True)
    (1): Linear(in_features=384, out_features=192, bias=True)
    (2): Linear(in_features=192, out_features=192, bias=True)
  )
  (fc_out): Linear(in_features=192, out_features=1, bias=True)
)
Number of parameters:  699793
```

## 4. Training MolGAT Model




(a)

Figure S 6: Error loss plot during MolGAT model training on AqSolDB

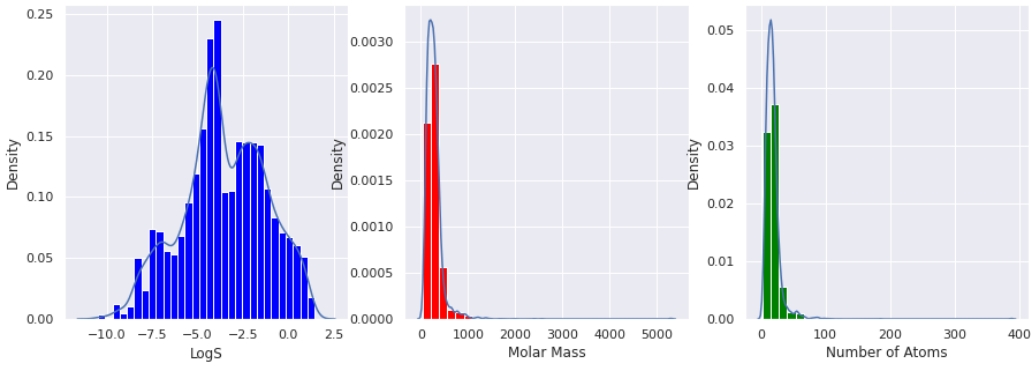



Figure S 7: Distributions of log solubility, molar mass (g/mol), and number of atoms for molecules in screened data set with MolGAT.

### *4.1. Benchmarking*

#### *4.1.1. Random Forest(RF) model training*

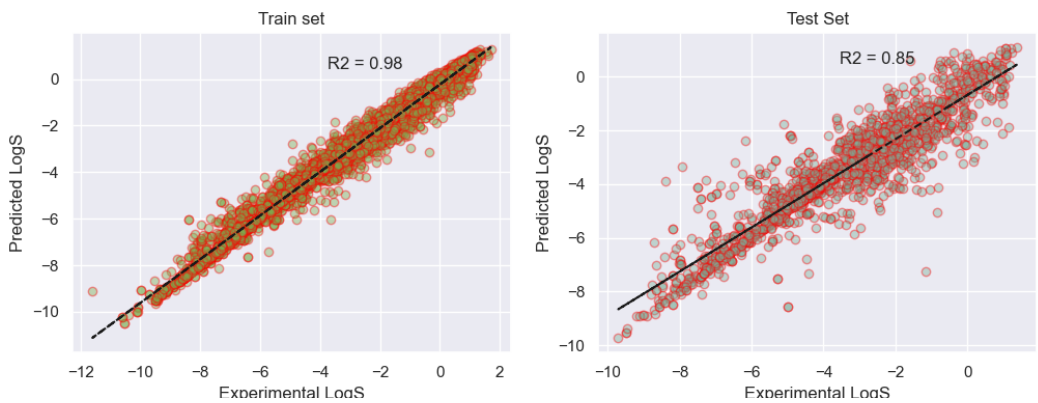



Figure S 8: Parity plot for Random Forest model with AqSolDB.

### 4.1.2. *Message Passing Neural Network(MPNN) Model Training*

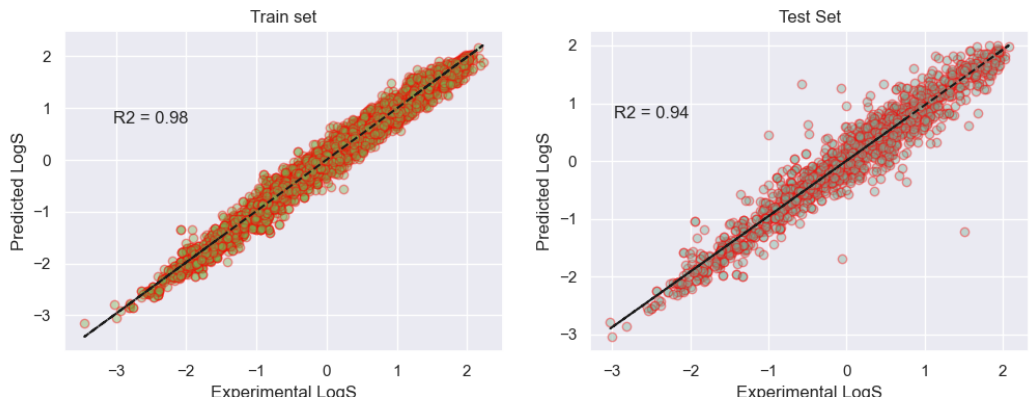



Figure S 9: Parity plot for MPNN model with AqSolDB.

### 4.1.3. *Graph Convolution Model(GCM) Model Training*

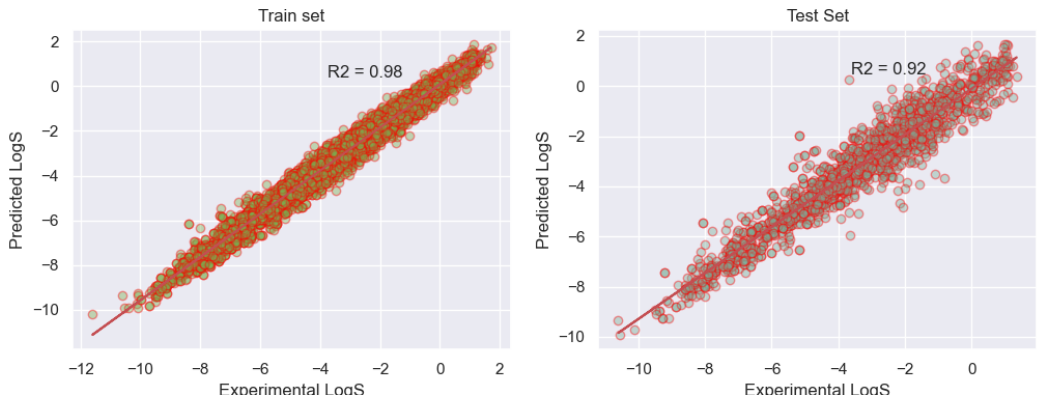



Figure S 10: Parity plot for GCM model with AqSolDB.

### 4.1.4. AttentiveFP Model Training

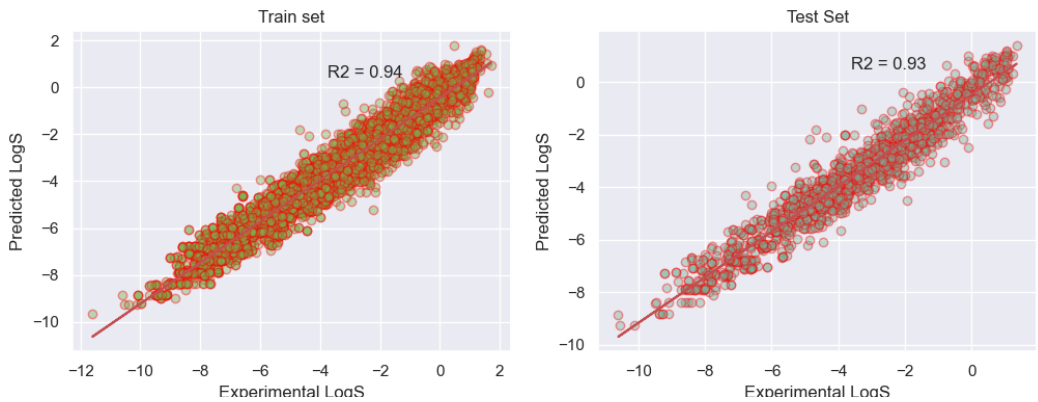



Figure S 11: Parity plot for AttentiveFP model with AqSolDB.

### 4.1.5. Graph attention Network(GAT) Model Training

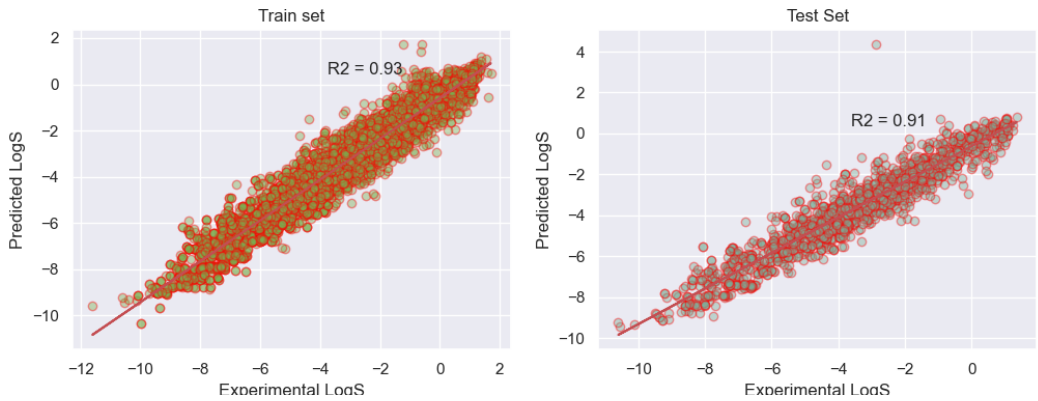



Figure S 12: Parity plot for GAT model with AqSolDB.

## 5. Screened Dataset

### 5.1. Sample molecules from screened dataset

Table S 2: Sample smiles, reaction energy and logS from a total screened dataset using the trained molgat model

| SMILES | Reaction energy(eV) | Predicted LogS(Mol/L) |
|---|---|---|
| $C1=CC(=O)C(=CN=C2C=CC3=NC(=NC3=C2)C4=C(C=C(C=C4)Cl)Cl)C=C1N(O)O$ | -2.35057 | -3.63134 |
| $CONC1=C2C=C(C=CC2=NC1=O)[N+](=O)[O-]$ | -2.31088 | -2.31388 |
| $COC1=CC(=CN=C2C=CC3=NC(=NC3=C2)C4=CC=CC=C4)C(=O)C(=C1)N(O)O$ | -2.30341 | -2.66933 |
| $C1=CC=C(C=C1)C2=NC3=CC(=NC=C4C=C(C=CC4=O)N(O)O)C=CC3=N2$ | -2.28478 | -2.75551 |

| SMILES | Reaction energy(eV) | Predicted LogS(Mol/L) |
|---|---|---|
| C1=CC=C(C=C1)C2=NC(=C3C=CC(=O)C(=C3)N(O)O)N=C2C4=CC=CC=C4 | -2.20292 | -3.32659 |
| C1=CC2=C(C(=C1)[N+](=O)[O−])C(=O)N=C2[O−] | -2.08322 | -2.38020 |
| C1=CC2=NC(=C3C=C(C=CC3=O)N=CC4=C(C(=CC=C4)N(O)O)O)N=C2C=C1 | -2.06541 | -3.33318 |
| CC1=CC(=CN=C2C=CC(=C3N=C4C=CC=CC4=N3)C=C2)C(=O)C(=C1)N(O)O | -2.01272 | -2.42499 |
| C1=CC2=C3C(=CC(=N)C(=O)N3C4=CC=C(C=C4)N(O)O)N=C2C=C1 | -1.94489 | -3.32238 |
| C1=CC2=C3C(=C1)NC(=NC3=CC=C2)C(=O)O | -1.78178 | -3.04786 |
| C1=CC=C2C(=C1)C(=CC3=C(N(C(=S)S3)CC(=O)[O−])O)C=N2 | -1.76830 | -3.55150 |
| CC(=O)N=C1C=CC2=NC(=NC(=S)NC(=O)C3=CC=C(O3)C4=CC=CC=C4N(O)O)SC2=C1 | -1.75821 | -3.09294 |
| CC1=CC=CC2=C(C(=O)N=C12)C3=C(N(C(=S)S3)CCC(=O)O)O | -1.71661 | -3.48864 |
| C1=COC(=C1)C2=NN3C(=N)C(=CC4=CC=C(O4)[N+](=O)[O−])C(=O)N=C3S2 | -1.71067 | -3.51928 |
| C1=CC=C2C(=C1)C(=CC3=C(N(C(=S)S3)C(CCC(=O)[O−])C(=O)[O−])O)C=N2 | -1.70025 | -3.29246 |
| CCCN[C@H](c1cncccc1C)C(C)(C)[NH+](C)C | 2.29304 | -1.58715 |
| CCCC[NH2+]CCOc1cccnc1C | 2.29404 | -1.31283 |
| COc1ccsc1CN[C@H]1CC[C@H]([NH+](C)C)C1 | 2.29995 | -1.29218 |
| Cc1cncccc1CC[NH2+][C@@H](C(C)C)C1CC1 | 2.30251 | -3.13770 |
| CC[NH+](CC)CC[C@H](N)c1cncccc1C | 2.31325 | -0.421087 |
| CC[NH2+]C[C@@H](Oc1cccnc1C)C(C)C | 2.31491 | -1.53976 |
| COc1ccsc1C[NH+](C)[C@H](C)C1(C)CC1 | 2.32320 | -3.54068 |
| Cc1ncccc1O[C@H](C)C[NH2+]CC(C)C | 2.33822 | -1.31810 |
| CCC(C)(C)[NH2+]CCOc1cccnc1C | 2.36882 | -1.49300 |
| CN[C@@H](c1cncccc1C)[C@]1([NH+](C)C)CCC[C@H](C)C1 | 2.79311 | -2.82409 |
| CC(C)(C)[NH2+]Cc1ccc(OCC[NH+]2CCCCC2)cn1 | 2.79488 | -1.70223 |
| CCC(CC)(CCO)CNc1cncc(Br)c1 | 2.81312 | -3.69122 |
| CC[C@H](C)COc1ccc(C[C@H]([NH3+])CC)nc1 | 2.83353 | -1.78672 |
| Cc1ccc(O[C@H](C)[C@@H](C)O)c(C[NH2+]CC(C)C)n1 | 2.85439 | -1.72013 |
| CC[C@@H](C)CN(CC)c1nc2c(s1)[C@H]([NH2+]C)CC(C)(C)C2 | 2.87374 | -2.58608 |