Image and Graph Deep Learning Methods for Cellular and Molecular Level Science Discoveries

Shuiwang Ji
Department of Computer Science & Engineering
Deep Learning in Pre-AlexNet Era (2009)

- Action recognition from real-world airport surveillance video
- One of the earliest work on deep learning for video analysis
- Winner of the 2009 TRECVID Video Surveillance Challenge

Ji et al., 3D Convolutional Neural Networks for Human Action Recognition. ICML’2010 and TPAMI’2013. 4,000+ citations
• More complex images in cellular and molecular imaging
• Image to image transformation problems
• Beyond grid-like data, such as graphs
• Use of graph techniques for molecular level analysis
Biological image transformations in general
Biological Image Transformations

- Image Denoising

Biological Image Transformations

- Substructure Prediction & Segmentation

U-Net

- Current state-of-the-art and most popular network for bioimage transformations
- Architecture
  - Down-sampling path
  - Up-sampling path
  - Skip connections

Local Operators in U-Net

- Limited receptive field
- Focus more on local dependencies
- Fixed weights of kernels once trained
Attention Operator

\[ O = \text{softmax}(QK^T) \times V \]
Local v.s. Non-local

- Location relationship based
- Pixels only have access to their neighbour pixels in the previous layer
- Determined weights
- Relevance or similarity based
- Pixels have access to other pixels globally
- Input-dependent weights


Fundamentally different with fully connected layers
Global Voxel Transformation Operators (GVTOs)

Size-preserving GVTO

**Input tensor** \( \mathcal{X} \in \mathbb{R}^{d \times h \times w \times c} \)

**1 x 1 x 1 Convs:** query \((\mathcal{Q})\), key \((\mathcal{K})\), value \((\mathcal{V})\)

**Unfold:** \( Q, K, V \in \mathbb{R}^{c \times dhw} \)

\[
Y = V \cdot \text{Normalize}(K^T Q) \in \mathbb{R}^{c \times dhw}
\]

**Fold back:** \( \mathcal{Y} \in \mathbb{R}^{d \times h \times w \times c} \)

**Output tensor** \( \mathcal{X} + \mathcal{Y} \)
Global Voxel Transformation Operators (GVTOs)

Size-preserving GVTO

Up-sampling GVTO

Down-sampling GVTO

- Size-preserving GVTOs
- Apply Up-sampling / Down-sampling GVTOs optionally

Results: 3D Image Denoising

- CARE datasets
  
  
  - 3D Image Denoising

<table>
<thead>
<tr>
<th></th>
<th>SSIM</th>
<th></th>
<th>PSNR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input</td>
<td>Baseline</td>
<td>Ours</td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Baseline</td>
</tr>
<tr>
<td>Planaria</td>
<td>0.226</td>
<td>0.771</td>
<td>0.795</td>
<td>22.22</td>
</tr>
<tr>
<td></td>
<td>0.183</td>
<td>0.740</td>
<td>0.778</td>
<td>21.73</td>
</tr>
<tr>
<td></td>
<td>0.156</td>
<td>0.644</td>
<td>0.694</td>
<td>21.44</td>
</tr>
<tr>
<td>Tribolium</td>
<td>0.368</td>
<td>0.917</td>
<td>0.921</td>
<td>22.68</td>
</tr>
<tr>
<td></td>
<td>0.236</td>
<td>0.900</td>
<td>0.907</td>
<td>21.37</td>
</tr>
<tr>
<td></td>
<td>0.182</td>
<td>0.876</td>
<td>0.885</td>
<td>20.81</td>
</tr>
</tbody>
</table>

*SSIM*: The Structural Similarity Index, the higher the better.

*PSNR*: Peak Signal-to-Noise Ratio, the higher the better.

Results: 3D Image Denoising

- CARE datasets
  

  ○ 3D Image Denoising (Planaria)
Results: 3D-2D Image Projection

- CARE datasets
  

- 3D Image Projection (Flywing)

<table>
<thead>
<tr>
<th></th>
<th>SSIM</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input</td>
<td>Baseline</td>
</tr>
<tr>
<td>Flywing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C0</td>
<td>0.190</td>
<td>0.607</td>
</tr>
<tr>
<td>C1</td>
<td>0.080</td>
<td>0.597</td>
</tr>
<tr>
<td>C2</td>
<td>0.024</td>
<td>0.559</td>
</tr>
</tbody>
</table>
Results: 3D-2D Image Projection

- Projection (Flywing)

![Graph showing SSIM and NRMSE for different conditions (C1, C2, C3) for U-Net (CARE) and GVTNet.]
### Results: Label-Free Prediction

- **Label-free prediction**
  

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Pearson Correlation</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseline (AVG)</td>
<td>Ours (AVG)</td>
<td>Difference</td>
<td>P-value</td>
</tr>
<tr>
<td>fibrillarin</td>
<td>0.876</td>
<td>0.880</td>
<td>0.005</td>
<td>0.021317</td>
</tr>
<tr>
<td>sec61_beta</td>
<td>0.722</td>
<td>0.738</td>
<td>0.015</td>
<td>0.029856</td>
</tr>
<tr>
<td>myosin_iib</td>
<td>0.481</td>
<td>0.504</td>
<td>0.023</td>
<td>0.002878</td>
</tr>
<tr>
<td>dna</td>
<td>0.626</td>
<td>0.638</td>
<td>0.012</td>
<td>0.004611</td>
</tr>
<tr>
<td>alpha_tubulin</td>
<td>0.800</td>
<td>0.806</td>
<td>0.007</td>
<td>0.051277</td>
</tr>
<tr>
<td>lamin_b1</td>
<td>0.843</td>
<td>0.853</td>
<td>0.009</td>
<td>4.04E-05</td>
</tr>
<tr>
<td>beta_actin</td>
<td>0.758</td>
<td>0.765</td>
<td>0.007</td>
<td>0.05905</td>
</tr>
<tr>
<td>dic_lamin_b1</td>
<td>0.645</td>
<td>0.648</td>
<td>0.003</td>
<td>0.01688</td>
</tr>
<tr>
<td>membrane_caax_63x</td>
<td>0.699</td>
<td>0.711</td>
<td>0.012</td>
<td>3.04E-06</td>
</tr>
<tr>
<td>zo1</td>
<td>0.460</td>
<td>0.486</td>
<td>0.026</td>
<td>0.023023</td>
</tr>
<tr>
<td>tom20</td>
<td>0.703</td>
<td>0.719</td>
<td>0.015</td>
<td>3.40E-07</td>
</tr>
<tr>
<td>st6gal1</td>
<td>0.200</td>
<td>0.214</td>
<td>0.014</td>
<td>0.003901</td>
</tr>
</tbody>
</table>

*Shown are **pearson correlations** averaged on 20 test samples per dataset*
Results: Label-Free Prediction

- Label-free prediction

Transfer Learning

Pre-training on Dataset A

Dataset A
- Input microscopy
- Augmented microscopy

Cropped patches

Training
- Augmentation models

With/without fine-tuning

Inference on new Datasets

Datasets B/C/D...
- Input microscopy

Predicting
- Augmentation models
- Augmented microscopy
Results: Transfer Learning

- Transfer Learning - without fine-tuning

SSIM - cross dataset prediction
Self-Supervised Transformations

- Supervised image transformation requires matched pairs of training images, which might be hard to obtain
- Can we do unsupervised learning?
- We use self-supervised approaches
- Self-supervised learning is a type of unsupervised learning approach that predict one part of input from other parts

- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the masked from the visible.
- Predict the any occluded part from all available parts.
- Pretend there is a part of the input you don’t know and predict that.

Figure: LeCun
We derive a **self-supervised upper bound** of the supervised loss.

**Theorem 1.** Consider a normalized noisy image $\mathbf{x} \in \mathbb{R}^m$ (obtained by subtracting the mean and dividing by the standard deviation) and its ground truth signal $\mathbf{y} \in \mathbb{R}^m$. Assume the noise is zero-mean and i.i.d among all the dimensions, and let $J$ be a subset of $m$ dimensions uniformly sampled from the image $\mathbf{x}$. For any $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$, we have

$$
\mathbb{E}_{\mathbf{x}, \mathbf{y}} \| f(\mathbf{x}) - \mathbf{y} \|^2 + \| \mathbf{x} - \mathbf{y} \|^2 \leq \mathbb{E}_x \| f(\mathbf{x}) - \mathbf{x} \|^2 + 2m \mathbb{E}_J \left[ \frac{\mathbb{E}_x \| f(\mathbf{x})_J - f(\mathbf{x}_{J^c})_J \|^2}{|J|} \right]^{1/2}
$$

(6)

**The self-supervised bound as the new loss**

## Results: Noise2Same

<table>
<thead>
<tr>
<th>Methods</th>
<th>Datasets</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ImageNet</td>
<td>HänZi</td>
<td>Planaria</td>
<td>BSD68</td>
<td></td>
</tr>
<tr>
<td>Traditional</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input</td>
<td>9.69</td>
<td>6.45</td>
<td>21.52 / 21.09 / 20.82</td>
<td>20.19</td>
<td></td>
</tr>
<tr>
<td>Supervised</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise2True</td>
<td>23.39</td>
<td>15.66</td>
<td>31.57 / 30.15 / 28.13</td>
<td>29.06</td>
<td></td>
</tr>
<tr>
<td>Self-Supervised + noise model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laine et al. [12]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>28.84</td>
<td></td>
</tr>
<tr>
<td>Self-Supervised</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laine et al. [12]</td>
<td>20.89</td>
<td>10.70</td>
<td>-</td>
<td>27.15</td>
<td></td>
</tr>
<tr>
<td><strong>Noise2Same</strong></td>
<td><strong>22.26</strong></td>
<td><strong>14.38</strong></td>
<td><strong>29.48 / 26.93 / 22.41</strong></td>
<td><strong>27.95</strong></td>
<td></td>
</tr>
</tbody>
</table>
Results: Noise2Same

<table>
<thead>
<tr>
<th>Input</th>
<th>BM3D</th>
<th>Noise2Self</th>
<th>Ours</th>
<th>Noise2Noise</th>
<th>Noise2True</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Input Image" /></td>
<td><img src="image2" alt="BM3D Image" /></td>
<td><img src="image3" alt="Noise2Self Image" /></td>
<td><img src="image4" alt="Ours Image" /></td>
<td><img src="image5" alt="Noise2Noise Image" /></td>
<td><img src="image6" alt="Noise2True Image" /></td>
<td><img src="image7" alt="Ground Truth Image" /></td>
</tr>
<tr>
<td><img src="image8" alt="Input Image" /></td>
<td><img src="image9" alt="BM3D Image" /></td>
<td><img src="image10" alt="Noise2Self Image" /></td>
<td><img src="image11" alt="Ours Image" /></td>
<td><img src="image12" alt="Noise2Noise Image" /></td>
<td><img src="image13" alt="Noise2True Image" /></td>
<td><img src="image14" alt="Ground Truth Image" /></td>
</tr>
<tr>
<td><img src="image15" alt="Input Image" /></td>
<td><img src="image16" alt="BM3D Image" /></td>
<td><img src="image17" alt="Noise2Self Image" /></td>
<td><img src="image18" alt="Ours Image" /></td>
<td><img src="image19" alt="Noise2Noise Image" /></td>
<td><img src="image20" alt="Noise2True Image" /></td>
<td><img src="image21" alt="Ground Truth Image" /></td>
</tr>
</tbody>
</table>
## Results: Noise2Same

<table>
<thead>
<tr>
<th>Condition</th>
<th>Input</th>
<th>NLM</th>
<th>Noise2Self</th>
<th>Ours</th>
<th>Noise2True (CARE)</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition 1</td>
<td><img src="input1.png" alt="Image" /></td>
<td><img src="nlm1.png" alt="Image" /></td>
<td><img src="noise2self1.png" alt="Image" /></td>
<td><img src="ours1.png" alt="Image" /></td>
<td><img src="noise2true1.png" alt="Image" /></td>
<td><img src="groundtruth1.png" alt="Image" /></td>
</tr>
<tr>
<td>Condition 2</td>
<td><img src="input2.png" alt="Image" /></td>
<td><img src="nlm2.png" alt="Image" /></td>
<td><img src="noise2self2.png" alt="Image" /></td>
<td><img src="ours2.png" alt="Image" /></td>
<td><img src="noise2true2.png" alt="Image" /></td>
<td><img src="groundtruth2.png" alt="Image" /></td>
</tr>
<tr>
<td>Condition 3</td>
<td><img src="input3.png" alt="Image" /></td>
<td><img src="nlm3.png" alt="Image" /></td>
<td><img src="noise2self3.png" alt="Image" /></td>
<td><img src="ours3.png" alt="Image" /></td>
<td><img src="noise2true3.png" alt="Image" /></td>
<td><img src="groundtruth3.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Deep Learning on Grid-Like Data

Image

Text

"Hello world!"

Audio

Grid-like Data!

Pictures from Google Images
From Images to Graphs
Objects + Relationships = Graph

Social networks  
Drug/Material molecules  
Citation networks  
Brain networks
Tasks on Graph Data

- Node classification/regression
- Graph classification/regression
- Link prediction
- Community detection
- …
Tasks on Graph Data

- e.g., Node classification
  - Citation Networks
Tasks on Graph Data

- e.g., Graph classification
  - Molecular property prediction

Toxic

Non-toxic

?
Molecular Property Prediction

Quantum Mechanics
- QM8
- QM9

Physical Chemistry
- ESOL
- Lipophilicity
- FreeSolv

Biophysics
- HIV
- BACE
- PCBA
- MUV

Physiology
- BBBP
- Tox21
- ToxCast
- SIDER
- ClinTox

MoleculeNet datasets, different datasets focus on different molecule properties

Geometric, energetic, electronic, thermodynamic properties
Water solubility, hydration free energy, lipophilicity
Bioactivity, binding affinities
Barrier permeability, toxicology, adverse reactions

3D Graphs

3D graphs
- Graph structure
- Node-level positional info

Important for a lot of real-world applications

Molecular modeling
Protein structural representation
Quantum system simulation
Research Obstacles

- No unified framework for 3D graphs
- Structural representation is not complete

SchNet: distance

DimeNet: distance & angle

Only distance & angle is not enough!

A Unified Framework: 3DGN

- A generic and unifying framework for 3D graphs
- An interface for manipulating 3D graphs at different levels of granularity

A Novel MP Method: SMP

Spherical message passing
✓ An accurate and complete architecture for 3DGN
✓ Existing methods are special cases

SMP considers distance, angle, and torsion

### QM9

<table>
<thead>
<tr>
<th>Property</th>
<th>Unit</th>
<th>PPGN</th>
<th>SchNet</th>
<th>PhysNet</th>
<th>Cormorant</th>
<th>MGCN</th>
<th>DimeNet</th>
<th>DimeNet++</th>
<th>SphereNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mu)</td>
<td>D</td>
<td>0.047</td>
<td>0.033</td>
<td>0.0529</td>
<td>0.13</td>
<td>0.0560</td>
<td>0.0286</td>
<td>0.0297</td>
<td>0.0269</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>(a_0)</td>
<td>0.131</td>
<td>0.235</td>
<td>0.0615</td>
<td>0.092</td>
<td>0.0300</td>
<td>0.0469</td>
<td>0.0035</td>
<td>0.00465</td>
</tr>
<tr>
<td>(\epsilon_{\text{HOMO}})</td>
<td>meV</td>
<td>40.3</td>
<td>41</td>
<td>32.9</td>
<td>36</td>
<td>42.1</td>
<td>27.8</td>
<td>24.6</td>
<td>23.6</td>
</tr>
<tr>
<td>(\epsilon_{\text{LUMO}})</td>
<td>meV</td>
<td>32.7</td>
<td>34</td>
<td>24.7</td>
<td>36</td>
<td>57.4</td>
<td>19.7</td>
<td>19.5</td>
<td>18.9</td>
</tr>
<tr>
<td>(\Delta\epsilon)</td>
<td>meV</td>
<td>60.0</td>
<td>63</td>
<td>42.5</td>
<td>60</td>
<td>64.2</td>
<td>34.8</td>
<td>32.6</td>
<td>32.3</td>
</tr>
<tr>
<td>(&lt;M^2_0&gt;)</td>
<td>(a_0)^2</td>
<td>0.592</td>
<td>0.073</td>
<td>0.765</td>
<td>0.673</td>
<td>0.110</td>
<td>0.331</td>
<td>0.331</td>
<td>0.292</td>
</tr>
<tr>
<td>ZPVE</td>
<td>meV</td>
<td>3.12</td>
<td>1.7</td>
<td>1.39</td>
<td>1.98</td>
<td>1.12</td>
<td>1.29</td>
<td>1.21</td>
<td>1.12</td>
</tr>
<tr>
<td>(U_0)</td>
<td>meV</td>
<td>36.8</td>
<td>14</td>
<td>8.15</td>
<td>28</td>
<td>12.9</td>
<td>8.02</td>
<td>6.32</td>
<td>6.26</td>
</tr>
<tr>
<td>(U)</td>
<td>meV</td>
<td>36.8</td>
<td>19</td>
<td>8.34</td>
<td>-</td>
<td>14.4</td>
<td>7.89</td>
<td>6.28</td>
<td>7.33</td>
</tr>
<tr>
<td>(H)</td>
<td>meV</td>
<td>36.3</td>
<td>14</td>
<td>8.42</td>
<td>-</td>
<td>14.6</td>
<td>8.11</td>
<td>6.53</td>
<td>6.40</td>
</tr>
<tr>
<td>(G)</td>
<td>meV</td>
<td>36.4</td>
<td>14</td>
<td>9.40</td>
<td>-</td>
<td>16.2</td>
<td>8.98</td>
<td>7.56</td>
<td>8.0</td>
</tr>
<tr>
<td>(c_v)</td>
<td>cal/mol K</td>
<td>0.055</td>
<td>0.033</td>
<td>0.0280</td>
<td>0.031</td>
<td>0.0380</td>
<td>0.0249</td>
<td>0.0230</td>
<td>0.0215</td>
</tr>
<tr>
<td>std. MAE</td>
<td>%</td>
<td>1.84</td>
<td>1.76</td>
<td>1.37</td>
<td>2.14</td>
<td>1.86</td>
<td>1.05</td>
<td>0.98</td>
<td>0.94</td>
</tr>
</tbody>
</table>

### OC20

<table>
<thead>
<tr>
<th>Model</th>
<th>ID</th>
<th>OOD Ads</th>
<th>OOD Cat</th>
<th>OOD Both</th>
<th>Energy MAE [eV] ↓</th>
<th>ID</th>
<th>OOD Ads</th>
<th>OOD Cat</th>
<th>OOD Both</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGCNN</td>
<td>1.0479</td>
<td>1.0527</td>
<td>1.0232</td>
<td>0.9608</td>
<td></td>
<td>1.39%</td>
<td>1.38%</td>
<td>1.59%</td>
<td>1.57%</td>
</tr>
<tr>
<td>SchNet</td>
<td>1.0858</td>
<td>1.1044</td>
<td>1.0720</td>
<td>1.0391</td>
<td></td>
<td>1.34%</td>
<td>1.39%</td>
<td>1.42%</td>
<td>1.44%</td>
</tr>
<tr>
<td>DimeNet</td>
<td>1.0117</td>
<td>1.0734</td>
<td>0.9814</td>
<td>0.9767</td>
<td></td>
<td>1.45%</td>
<td>1.41%</td>
<td>1.53%</td>
<td>1.41%</td>
</tr>
<tr>
<td>DimeNet++</td>
<td>0.8819</td>
<td>0.9106</td>
<td>0.8357</td>
<td>0.8408</td>
<td></td>
<td>1.94%</td>
<td>1.69%</td>
<td>2.13%</td>
<td>1.84%</td>
</tr>
<tr>
<td>SphereNet</td>
<td>0.8352</td>
<td>0.8723</td>
<td>0.7959</td>
<td>0.7952</td>
<td></td>
<td>1.96%</td>
<td>2.02%</td>
<td>2.19%</td>
<td>1.90%</td>
</tr>
</tbody>
</table>

SphereNet filters
All the distance, angle, and torsion make contributions
Neural Nets are Black Boxes

- They cannot be fully trusted!
  - It prevents their use in critical applications pertaining to fairness, privacy, and safety

- We need explanation techniques
  - Explain the relationships between input graph and output predictions

Reference: https://towardsdatascience.com/guide-to-interpretable-machine-learning-d40e8a64b6cf
Explaining GNNs

• Existing methods focus on nodes, edges, and features.
  • Less intuitive, not human-intelligible

• Subgraph-level explanations
  • Subgraphs are highly related to the functionalities of graphs
  • More human-intelligible
  • Example: network motifs, the simple building blocks of complex networks

The Proposed SubgraphX

- Explain GNN predictions using subgraphs
  - Find the most important subgraph for a target prediction

- Explore subgraphs with the search algorithm
  - Monte Carlo Tree Search
  - Learn the trade-off between exploration and exploitation

- A Game-Theoretical Scoring Function
  - Shapley values to measure the importance of different subgraphs
  - Efficient approximation schemes to compute Shapley values by considering interactions within the information aggregation range

Hao Yuan…Shuiwang Ji: On Explainability of Graph Neural Networks via Subgraph Explorations. https://arxiv.org/abs/2102.05152
The Proposed SubgraphX

- Shapley values can capture the interactions in GNNs!

![Diagram showing the application of Shapley values in GNNs]

Department of Computer Science & Engineering
Results: SubgraphX

- Data are labeled by the motifs (house-like motif or cycle motif)
- First row: correct prediction; second row: incorrect prediction.
Results: SubgraphX

Fidelity: the change of prediction by removing important substructures (higher is better)
From Graph Prediction to Generation
A fundamental problem in drug discovery is to find novel molecules with desired properties. Search in the chemical space with molecular property prediction. Extremely expensive since chemical space is huge.

Generate molecules with desired properties directly. Circumvent the expensive search. Molecule generation is still in the preliminary stage.
Given a molecule dataset, use it to train a molecule generator which can generate valid molecular graphs.
Given a pretrained molecule generator, optimize it to generate molecules with high property scores.
Decompose the molecular graph into a sequence of discrete tokens $S_G = (a_1, a_2, b_{21}, a_3, ...)$.

Use the composition of multiple invertible modulo shift transforms to convert discrete latent variables $z$ to the next discrete token $x$ based on

$$x = (z + \mu \mod t),$$

where $t$ is the number of categories of $x$.

Conditional information is extracted with Relational Graph Convolutional networks (R-GCN)* and incorporated into $\mu$.

Formulate the sequential generation of molecular graph as a Markov Decision Process,
- state --- currently unfinished sub-graph,
- action --- new node or edge addition,
- reward --- property score of the finished molecular graph.

Start from a model pretrained on a molecule dataset, then fine-tune the model with reinforcement learning by Proximal Policy Optimization (PPO)* algorithm.

Energy-Based Models

- Energy-based models (EBMs)
  - $E_\theta(x): X \rightarrow R$ is the energy function, e.g., a deep neural network.
  - The probability distribution given by an EBM is:
    \[
    p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z_\theta}.
    \]
  - $Z_\theta = \int e^{-E_\theta(x)} \, dx$ for continuous data and $Z_\theta = \sum e^{-E_\theta(x)}$ for discrete data. $Z_\theta$ is usually intractable.
  - EBMs have been used as generative models in various domains, e.g., images and videos.

GraphEBM

- Energy function is parameterized as a permutation invariant graph neural network.
- Maximum likelihood training with MCMC (e.g., Langevin dynamics).
- Goal-directed: push down energies with flexible degrees $f(y)$ based on property values $y$
  $$\text{loss} = f(y)E_\theta(X^{\oplus}, A^{\oplus}) - E_\theta(X^{\ominus}, A^{\ominus}).$$
- Multi-objective: combine multiple trained energy functions associated with different properties
  $$E_{\theta^*}(X, A) = E_{\theta_1}(X, A) + E_{\theta_2}(X, A).$$
- Generating with MCMC sampling.

## Results: GraphDF

<table>
<thead>
<tr>
<th>Method</th>
<th>Validity %</th>
<th>Validity w/o check %</th>
<th>Uniqueness %</th>
<th>Novelty %</th>
<th>Reconstruct %</th>
</tr>
</thead>
<tbody>
<tr>
<td>JT-VAE</td>
<td>100</td>
<td>n/a</td>
<td>100</td>
<td>100</td>
<td>76.7</td>
</tr>
<tr>
<td>GCPN</td>
<td>100</td>
<td>20</td>
<td>99.97</td>
<td>100</td>
<td>n/a</td>
</tr>
<tr>
<td>MoFlow</td>
<td>100</td>
<td>81.76</td>
<td><strong>99.99</strong></td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>GraphCNF</td>
<td>96.35</td>
<td>n/a</td>
<td>99.98</td>
<td>99.98</td>
<td>100</td>
</tr>
<tr>
<td>GraphAF</td>
<td>100</td>
<td>68</td>
<td>99.1</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>GraphDF (ours)</td>
<td><strong>100</strong></td>
<td><strong>89.03</strong></td>
<td>99.16</td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

Table 1 Molecule generation performance on ZINC 250K dataset. GraphDF can model the underlying chemical rule more accurately and achieve much higher validity w/o check rate.

<table>
<thead>
<tr>
<th>Method</th>
<th>Penalized logP</th>
<th>QED</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st</td>
<td>2nd</td>
<td>3rd</td>
</tr>
<tr>
<td>JT-VAE</td>
<td>5.3</td>
<td>4.93</td>
<td>4.49</td>
</tr>
<tr>
<td>GCPN</td>
<td>7.98</td>
<td>7.85</td>
<td>7.8</td>
</tr>
<tr>
<td>MRNN</td>
<td>8.63</td>
<td>6.08</td>
<td>4.73</td>
</tr>
<tr>
<td>MoFlow</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>GraphAF</td>
<td>12.23</td>
<td>11.29</td>
<td>11.05</td>
</tr>
<tr>
<td>GraphDF (ours)</td>
<td><strong>13.70</strong></td>
<td><strong>13.18</strong></td>
<td><strong>13.16</strong></td>
</tr>
</tbody>
</table>

Table 2 Property optimization performance evaluated by top-3 property scores. GraphDF has stronger capacity to search molecules with high property scores.
Results: GraphEBM

- **Random generation**
  - can generate non-trivial molecules

- **Goal-directed generation**
  - can improve the property of generated molecules

- **Multi-objective generation**
  - can improve multiple properties
DIG: Dive into Graphs

- A research-oriented testbed for graph neural networks
- Includes unified implementation of common methods, datasets, evaluations
- A turn-key solution to easily test new ideas and compare with baselines
- Currently, DIG includes four major research directions for graphs
  - Graph generation
  - Self-supervised learning on graphs
  - Interpretability of graph neural networks
  - 3D graphs

https://github.com/divelab/DIG (Coming soon!)
Conclusion and Outlook

• Concerted efforts to solve challenging real-world problems
• Develop new methods, and use them to solve problems in biology, neuroscience, chemistry, physics, etc.
• Scales range from cellular, molecular, to atomic in quantum world
• Interested in fundamental methodology, and fundamental science
Team and Acknowledgements

- Lei Cai (Scientist, Microsoft)
- Hongyang Gao (Assistant Professor, Iowa State)
- Zhengyang Wang (Scientist, Amazon)
- Meng Liu (Active PhD student)
- Yi Liu (Active PhD student)
- Youzhi Luo (Active PhD student)
- Bora Oztekin (Active Undergrad)
- Limei Wang (Active PhD student)
- Yaochen Xie (Active PhD student)
- Zhao Xu (Active PhD student)
- Hao Yuan (Active PhD student, joining Facebook)
Image and Graph Deep Learning Methods for Cellular and Molecular Level Science Discoveries

sji@tamu.edu
http://people.tamu.edu/~sji/
https://github.com/divelab/DIG/ coming in 3 weeks