Stanford Graph Learning Workshop

Jure Leskovec
Graphs

Graphs represent objects and their relationships as nodes and edges.

Graphs capture relations, which leads to better models.
Graphs in Many Domains

Often most valuable data are graphs:

- Healthcare
- IT industry
- Commerce & Retail
- Transportation
- Finance & Insurance
- Manufacturing
Today We are Showing

- Research advancements in Graph ML
- PyG ecosystem and partnerships
- Industrial applications

- How these technologies propel advancements of AI and applications of AI to next wave for science and industry

- We are going to announce new tools and partnerships that power these applications
Thank you!

Thank you to all the researchers, developers and partners who join us to make this event amazing.

Welcome to Stanford Graph Learning Workshop!
Each node defines a computation graph

- Each edge is a message function $\text{Msg}(\cdot)$
- $\oplus$ is a message aggregation function

Graph Neural Networks (GNNs) can learn patterns and relations on a giant scale and train predictive models:

1) GNNs learn to combine features from neighboring nodes

2) GNNs learn the graph patterns and relations
Key Benefits of GNNs

- GNNs adapt to local \textit{shape} of data
  - Other methods assume fixed input (matrix, sequence) while GNNs capture local patterns around each node

- GNNs glue together other Neural Network architectures (CNNs, Transformers) and integrate multimodal data
Key Benefits of GNNs

GNNs are extremely general: They subsume CNNs and Transformers as special cases:

- **Example**: CNN layer with 3x3 filter

\[
GNN \text{ formulation: } h_v^{(l+1)} = \sigma(\sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + B_l h_v^{(l)}), \forall l \in \{0, ..., L - 1\}
\]

\[
CNN \text{ formulation: } h_v^{(l+1)} = \sigma(\sum_{u \in N(v)} W_u^{(l)} h_u^{(l)} + B_l h_v^{(l)}), \forall l \in \{0, ..., L - 1\}
\]
Many Applications of GNNs

Drug discovery

Fraud detection

Recommender systems

Customer 360 journey analysis

Fake news detection

Supply chain optimization
The hottest subfield in ML

ICLR 2022 Submission Top 50 Keywords

- reinforcement learning
- deep learning
- representation learning
- graph neural network
- self-supervised learning
- federated learning
- generalization
- robustness
- neural network
- transformer
- transfer learning
- contrastive learning
- computer vision

Learning on Graphs Conference
9th – 12th December 2022
https://logconference.org
Dynamic Financial Networks

Financial Networks: Describe financial entities and their connections

Tasks: Fraud detection, Anti-money laundering, Anomaly detection

Central Bank of a European country forecasted financial transactions:

- Central Bank was able to process dynamic graphs with tens of millions of transactions:
  - 43-73% improvement in accuracy
  - Robust performance with changes in transaction pattern
ROLAND: Tool for Dynamic Graphs

We developed ROLAND:

- Easy creation of dynamic GNN from any static GNN
- Scalable and adaptive training

Key idea: Recurrently update node embeddings at each layer, by injecting a new module to a static GNN:

Input:
- Previous embeddings from the same layer
- Current embeddings from the previous layer

Output: Updated embeddings
https://github.com/snap-stanford/roland
ROLAND: Implementation

ROLAND is built with **PyG GraphGym** to efficiently explore the GNN design space.


Jure Leskovec (@jure), Stanford University
PyG: GNN Library

PyG provides the state-of-the-art in Graph Representation Learning

- Models
  - User-Defined Models
  - Pre-Defined Models and Examples

- Operators
  - Message Passing
  - torch_cluster → Pooling
  - Normalization
  - Readout

- Storage
  - Data Loaders
    - Mini-Batching
    - Neighbor Sampling
    - Subgraph Sampling
  - Data
  - Transforms
  - Datasets

- Engine
  - PyTorch
  - torch_scatter → torch_sparse

Jure Leskovec (@jure), Stanford University
PyG 2.1: What’s New

- **pyg-lib**: A low-level GNN engine to further accelerate PyG, joint effort across different partnerships
- Improved GNN design via *principled* aggregations
- New model milestones: Scalable Link prediction, 1000+ layer deep GNNs, GNNs for heterophily graphs, ...
PyG: GNN Library

PyG is the most used Graph Representation Learning framework

- 80+ GNN architectures
- 200+ benchmark datasets
- Extendable via a message passing interface
- Dedicated sparsity-aware CUDA kernels
- Scalable and easy to use

conda install pyg -c pyg

stars 15k  forks 2.7k
PyG Powers Products

- anomaly and outlier detection
- knowledge graph for drug discovery
- recommender system for audio books and podcasts
- fraud detection
- fluid simulations
Graph ML Case Studies

Deep Learning on 3D Meshes
A learned solution to node-level classification on irregular graphs via graph neural networks.

Predicting Los Angeles Traffic with Graph Neural Networks
By Julie Wang, Amelia Woodward, Tracy Cai as part of the Stanford CS224W course project.

Alexa, Queue That Banger!
Recommending Spotify Playlist Tracks with Neural Collaborative Filtering Using Graph Machine Learning

Graph ML Tutorials: [https://medium.com/stanford-cs224w](https://medium.com/stanford-cs224w)

- Graph Machine Learning tutorials created by Stanford students of CS224W.
Stanford Graph ML Videos

Stanford CS224W: Machine Learning with Graphs on YouTube

https://youtube.com/playlist?list=PLoROMvodv4rPLKxlpqhhPgdQy7imNkDn

- Over 700,000 views
PyG: Huge Community

**PYG.ORG:**

- ~1600 research papers written using PyG
- ~100K monthly downloads
- ~300 external contributors/developers
- ~3k members on Slack (Join us: [https://data.pyg.org/slack.html](https://data.pyg.org/slack.html))

*team@pyg.org*

https://pyg.org

[@pyg-team/pytorch-geometric](https://pyg-team/pytorch-geometric)
Scaling-up Graph Learning

- In real-world applications, graph sizes easily reach billions of nodes and edges.
- Now with PyG, we can learn deep learning models at a giant scale and train a predictive model for a graph.
New PyG Abstractions

- Graphs contain both the wireframe and rich features on its nodes.
- Scaling up requires distributing this information out-of-core.
- New PyG graph store and PyG feature store abstractions enable modular scalability.
Partnership with NVIDIA

NVIDIA joins as an official partner to accelerate GPU needs:

- GPU-accelerated neighbor sampling via cugraph integration
- GPU-accelerated heterogeneous GNN execution via typed matrix multiply
Partnership with Intel

Intel joins as an official partner to accelerate CPU needs:

- Accelerated inference of GNNs on the CPU
- Accelerated neighbor sampling techniques (up to 20x speed-ups)
In 2021 we run OGB-LSC with over 500+ participating teams

- Huge leap forward on model performance

We are announcing OGB-LSC-2:
Knowledge Graphs

Knowledge Graph stores facts as triples \((h, r, t)\)

- (‘Fulvestrant’, ‘causes’, ‘Migrane’)
- (‘Fulvestrant’, ‘treats’, ‘Breast Neoplasm’)

- **Examples:**

  ![Diagram](image)

  **WIKIDATA**

  **Freebase**
Predictive Queries over KGs

\[ C_? \exists P : \text{ASSOC}(d_1, P) \land \text{ASSOC}(d_2, P) \land \text{TARGET}(P, C_?) \]

“Predict drugs \( C_? \) that might target proteins that are associated with the given disease nodes \( d_1 \) and \( d_2 \)”

[Embedding Logical Queries on Knowledge Graphs. Hamilton, et al., NeurIPS 2018]
[Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. Ren, et al., ICLR 2020]
KG Reasoning with SMORE

SMORE: Scalable framework for multi-hop Knowledge Graph reasoning

Scales to full Freebase KG: 86M nodes, 338M edges

https://github.com/google-research/smore
Graphs & Language Models

Knowledge graphs help train more effective large language models.

Step (1)

Text corpus
Knowledge graph
[ENT] If it is not used for hair, a round brush is an example of art supplies.

Step (2)

Cross-modal Encoder
[ENT] If it is not used for hair, a round brush is an example of [MASK] [MASK].

Step (3)

Self-supervised Objective
Masked LM
art supplies
LM Head
KG link prediction
(round brush, at, hair)
LinkPred Head

Pretrain

Effect of pretraining

RoBERTa +Grease LM
DRAGON (Ours)

Effect of KG

https://arxiv.org/abs/2203.15827
Conclusion

- Today we announce new advances in research, new tools and platforms and exciting partnerships and updates to the PyG ecosystem
- We have an exciting program for you with talks from brightest minds in science and industry
<table>
<thead>
<tr>
<th>Time</th>
<th>Speaker(s)</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:30 - 10:00</td>
<td>Matthias Fey, PyG</td>
<td>What’s New in PyG</td>
</tr>
<tr>
<td>10:00 - 10:20</td>
<td>Ivaylo Bahtchevanov, PyG</td>
<td>Building PyG Open Source Community</td>
</tr>
<tr>
<td>10:20 - 10:40</td>
<td>Manan Shah &amp; Dong Wang, Kumo.ai</td>
<td>Scaling-up PyG</td>
</tr>
<tr>
<td>11:00 - 11:20</td>
<td>Rishi Puri, Nvidia</td>
<td>Accelerating PyG with Nvidia GPUs</td>
</tr>
<tr>
<td>11:20 - 11:40</td>
<td>Ke Ding, Intel</td>
<td>Accelerating PyG with Intel CPUs</td>
</tr>
<tr>
<td>11:40 - 12:00</td>
<td>Andreas Damianou, Spotify</td>
<td>Podcast Recommendations Spotify</td>
</tr>
<tr>
<td>12:00 - 12:20</td>
<td>Hema Raghavan, Kumo.ai</td>
<td>Query the Future using PyG</td>
</tr>
<tr>
<td>12:20 - 12:30</td>
<td>Joseph Huang, Stanford</td>
<td>Stanford CS LINXS Summer Program</td>
</tr>
<tr>
<td>13:30 - 13:50</td>
<td>Marinka Zitnik, Harvard</td>
<td>Graph AI to Enable Precision Medicine</td>
</tr>
<tr>
<td>13:50 - 14:10</td>
<td>Bryan Perozzi, Google</td>
<td>Challenges and Solutions for GNNs at Google</td>
</tr>
<tr>
<td>14:10 - 14:30</td>
<td>Srijan Kumar, GaTech</td>
<td>GNNs for Web Safety and Integrity</td>
</tr>
<tr>
<td>14:30 - 14:50</td>
<td>Luna Dong, Meta</td>
<td>Graph Mining for Intelligent Assistants</td>
</tr>
<tr>
<td>14:50 - 15:10</td>
<td>Michi Yasunaga, Stanford</td>
<td>Graph Learning in NLP Applications</td>
</tr>
<tr>
<td>15:30 - 15:50</td>
<td>Weihua Hu, Stanford</td>
<td>Learning Backward Compatible Embeddings</td>
</tr>
<tr>
<td>15:50 - 16:10</td>
<td>Hongyu Ren, Stanford</td>
<td>Multi-hop Reasoning in Knowledge Graphs</td>
</tr>
<tr>
<td>16:10 - 17:00</td>
<td>Panel</td>
<td>Challenges and Opportunities for Graph Learning</td>
</tr>
<tr>
<td></td>
<td>• Naren Chittar, JPMorgan Chase (moderator)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Evan Feinberg, Genesis Therapeutics</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Yunyao Li, Apple</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Neil Shah, Snap</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Karthik Subbian, Amazon</td>
<td></td>
</tr>
</tbody>
</table>
Logistics

- **Q&A:** Subscribe to PyG Slack: [https://data.pyg.org/slack.html](https://data.pyg.org/slack.html) and join [#workshop-2022](#workshop-2022)

- Live stream: [https://youtu.be/GYW286H3SKw](https://youtu.be/GYW286H3SKw)
Thank you!

Thank you for attending!

Thank you Stanford Data Science Initiative for organizing:

Joseph Huang
Malwana Adalat
Rok Sosic
Ivaylo Buhtchevanov (PyG)