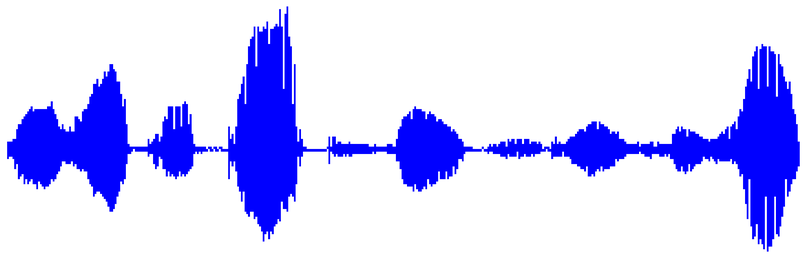
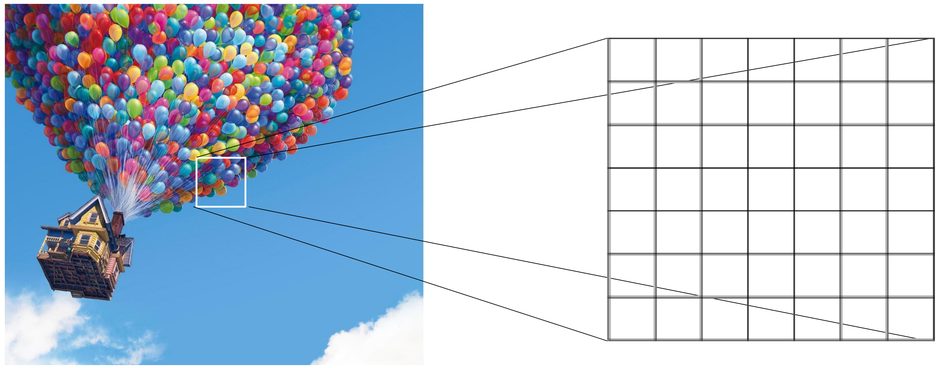
GTC FALL 2022
INTRODUCTION TO GRAPH NEURAL NETWORKS [DLIT41288]

Instructor: Dr. Ekaterina Sirazitdinova | NVIDIA

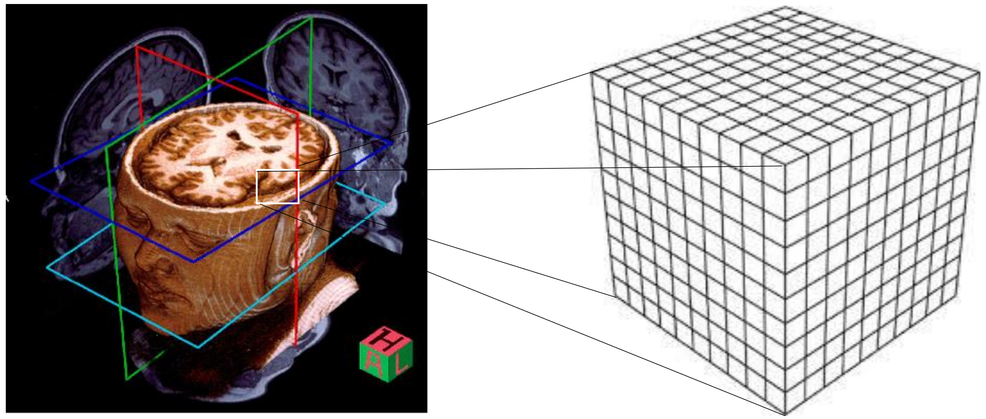
STRUCTURED DATA



1D: SIGNAL, SPEECH, TEXT

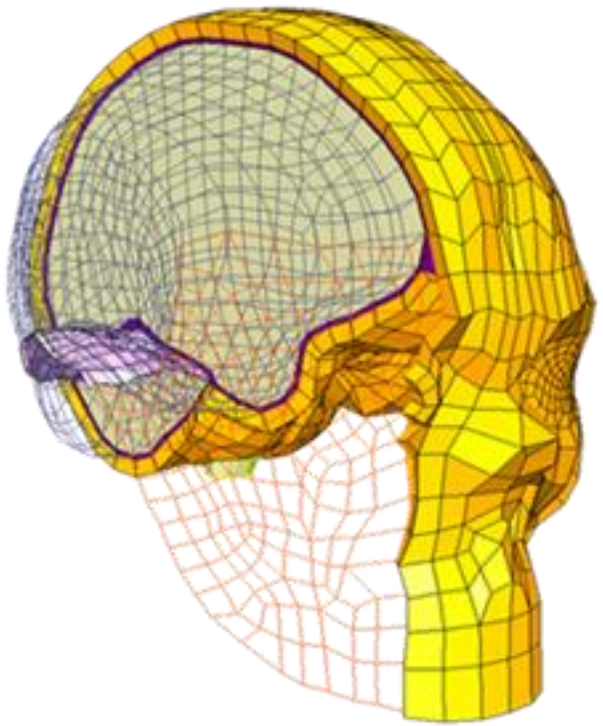


2D: IMAGES



3D: VOLUMES

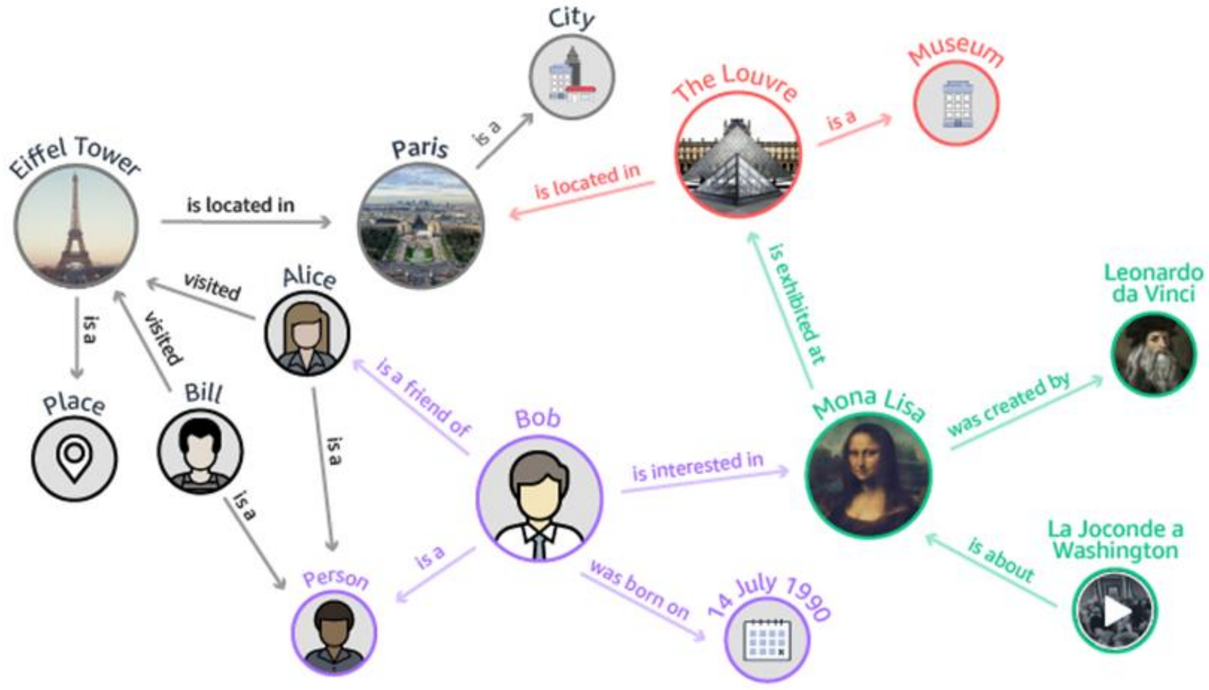
UNSTRUCTURED DATA



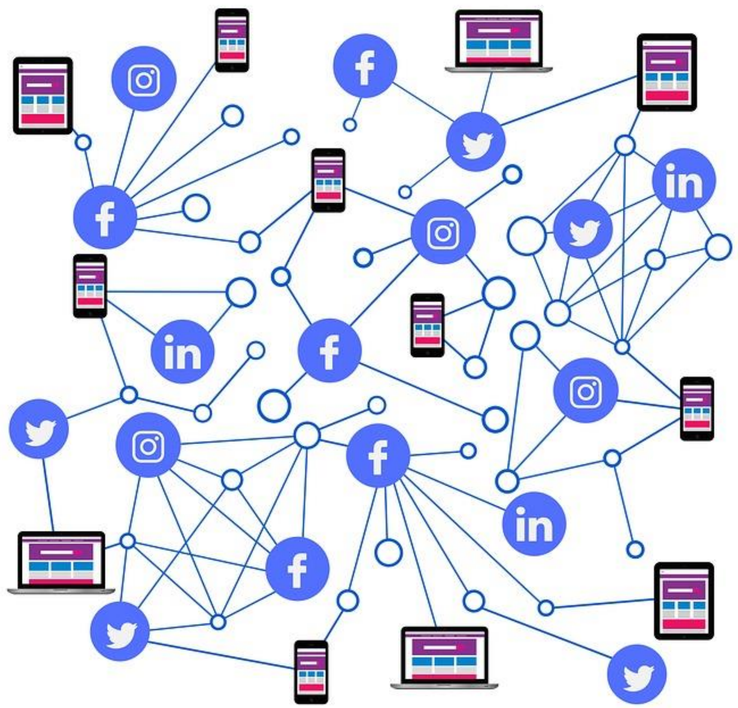
VOLUMETRIC MESHES

SURFACE MESHES

UNSTRUCTURED DATA



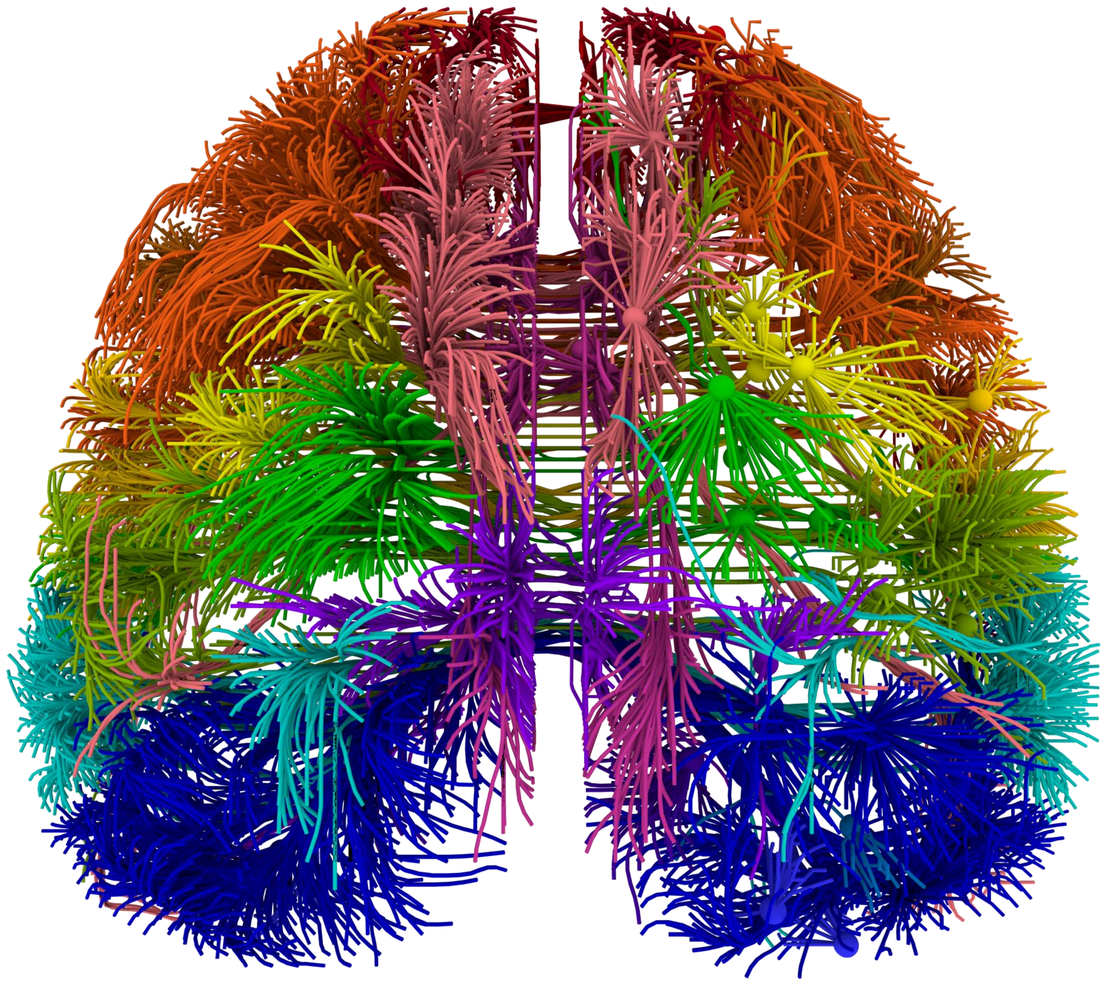
KNOWLEDGE GRAPHS

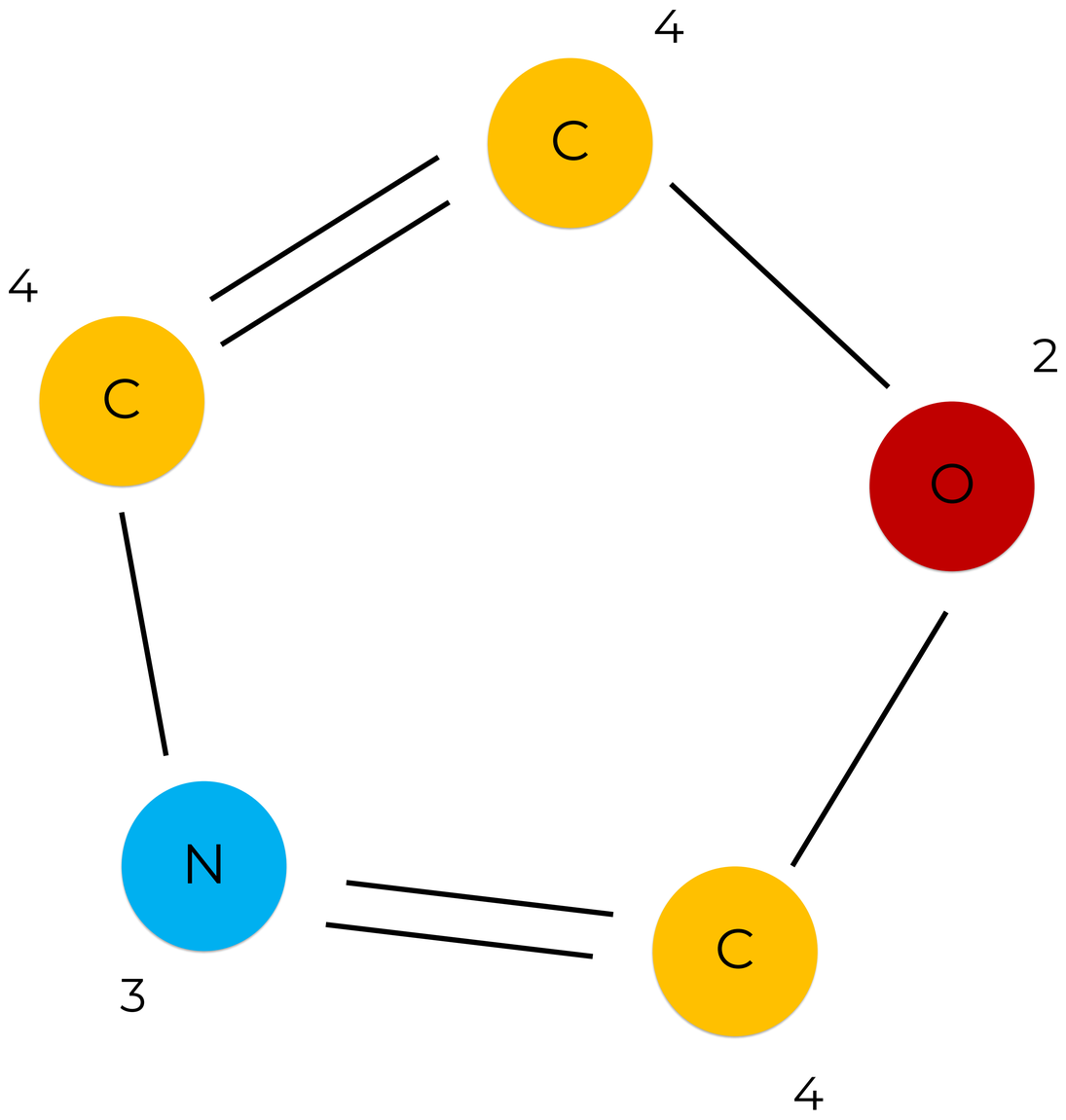


SOCIAL NETWORKS & RECOMMENDER SYSTEMS

HEALTHCARE

Connectomes (e.g. brain fMRI/DTI)





CHEMISTRY & PHARMACY

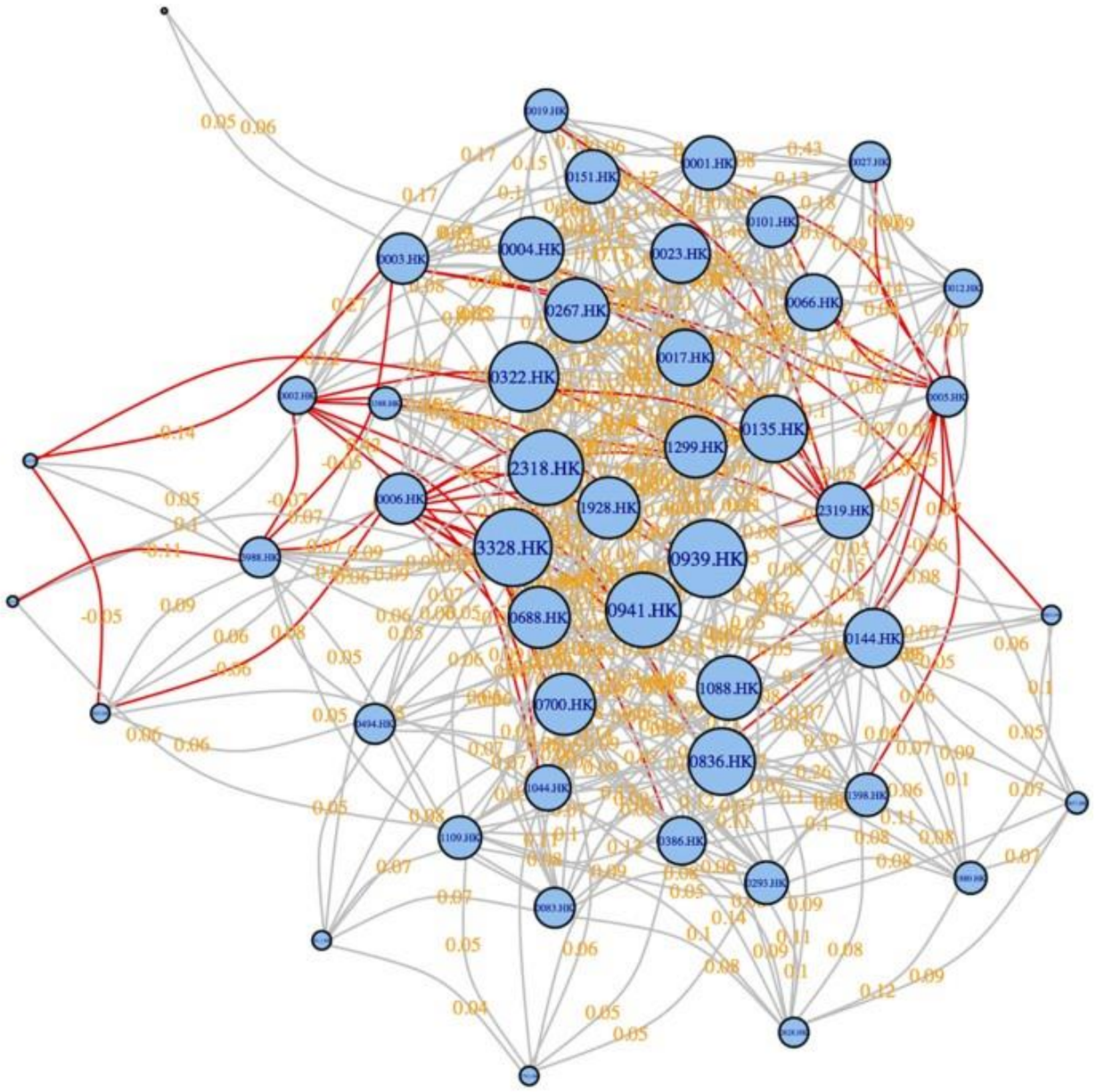
Molecule analysis

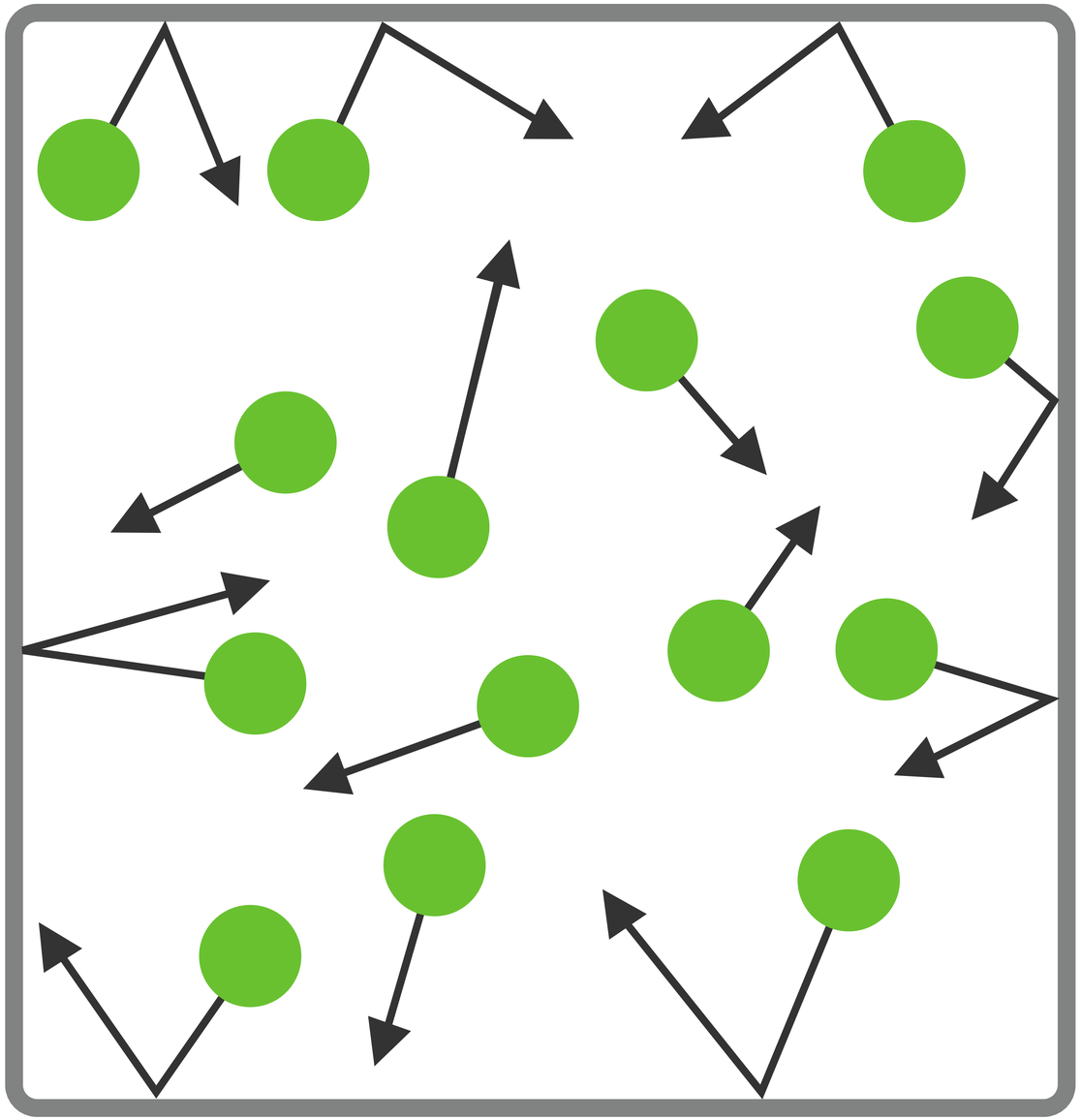
Drug discovery

Drug repurposing

FINANCIAL SOUNDNESS INDICATORS

Stock market prediction



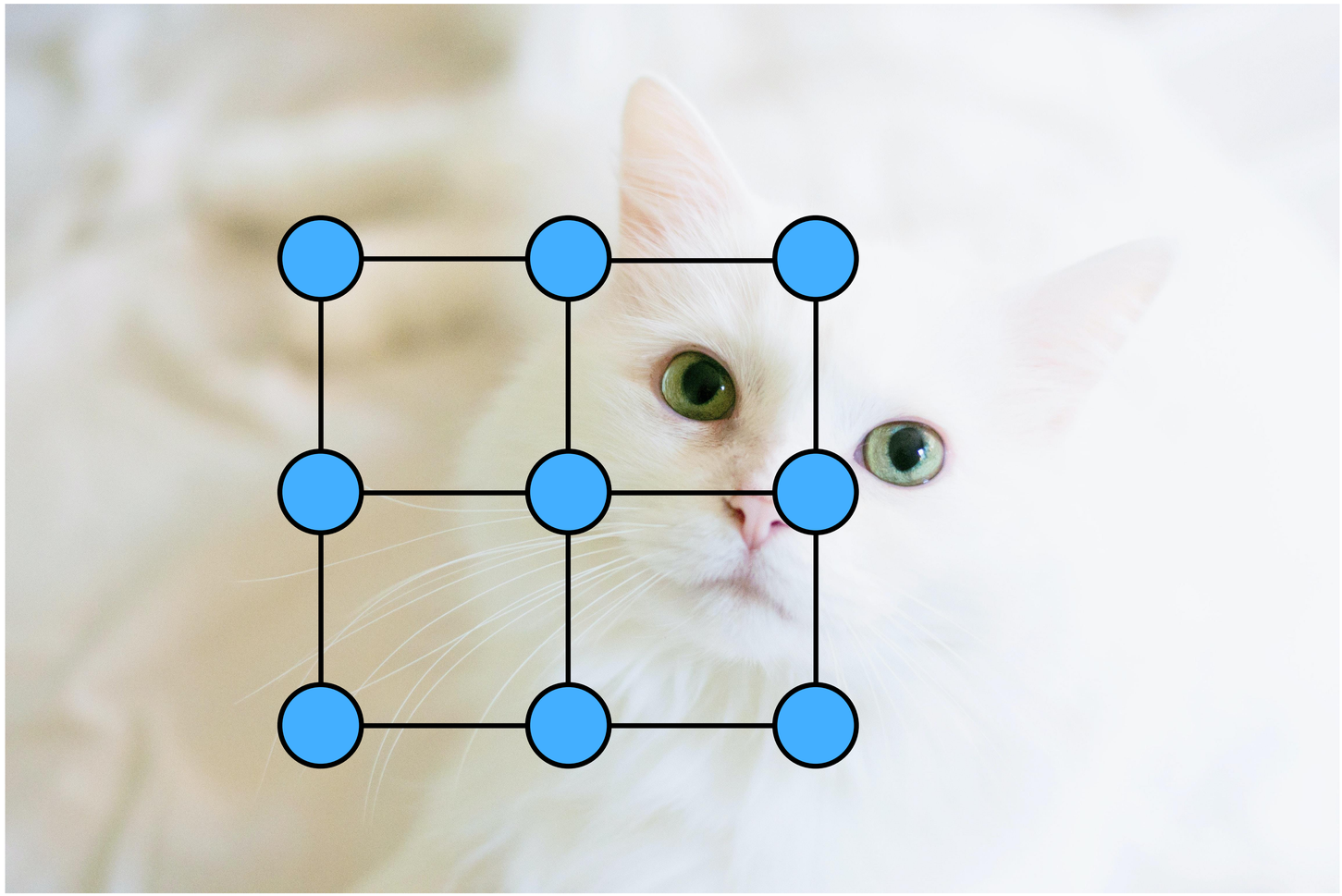


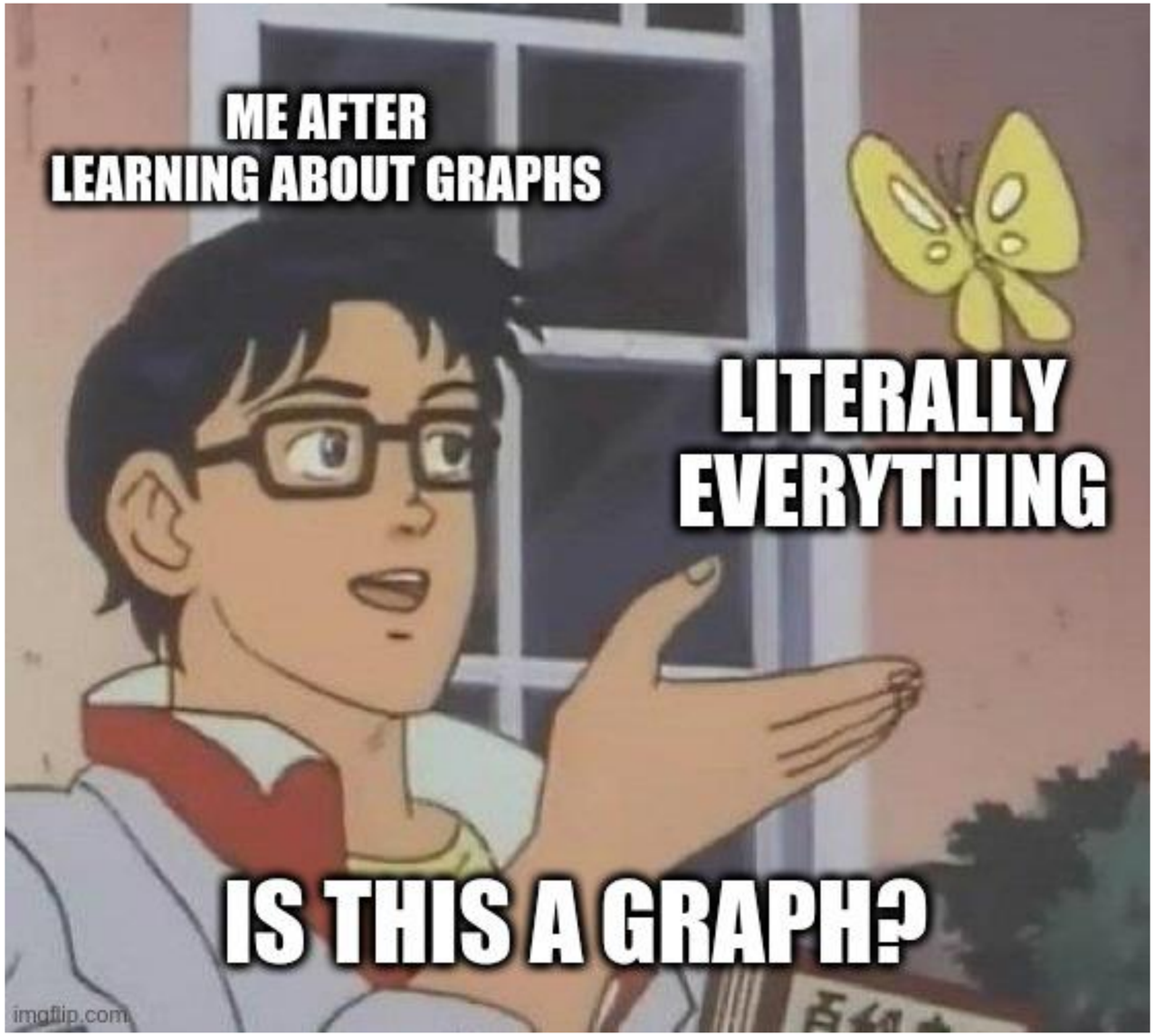
PHYSICS

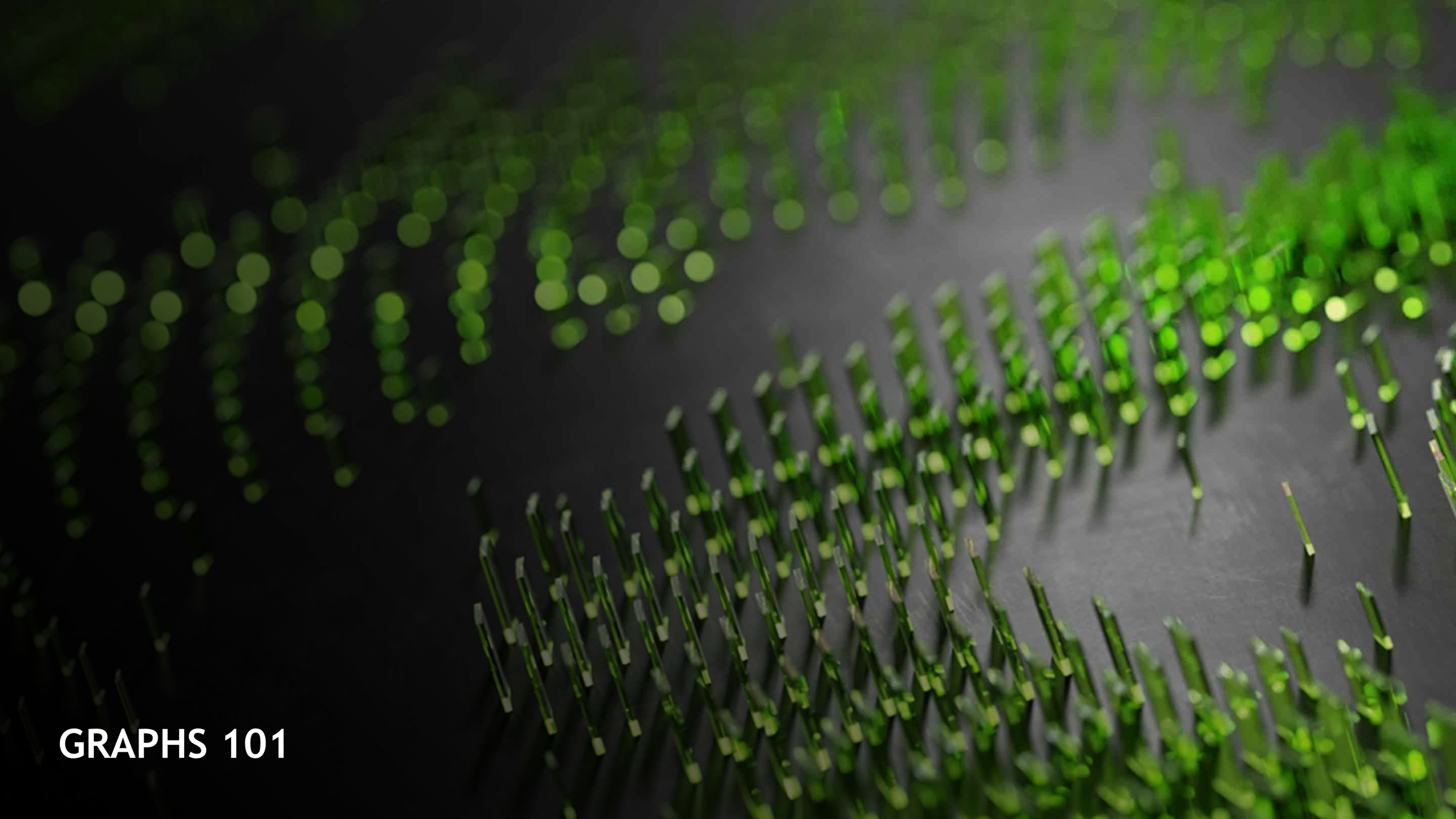
Particle systems

Thermodynamics

IMAGES AS GRAPHS

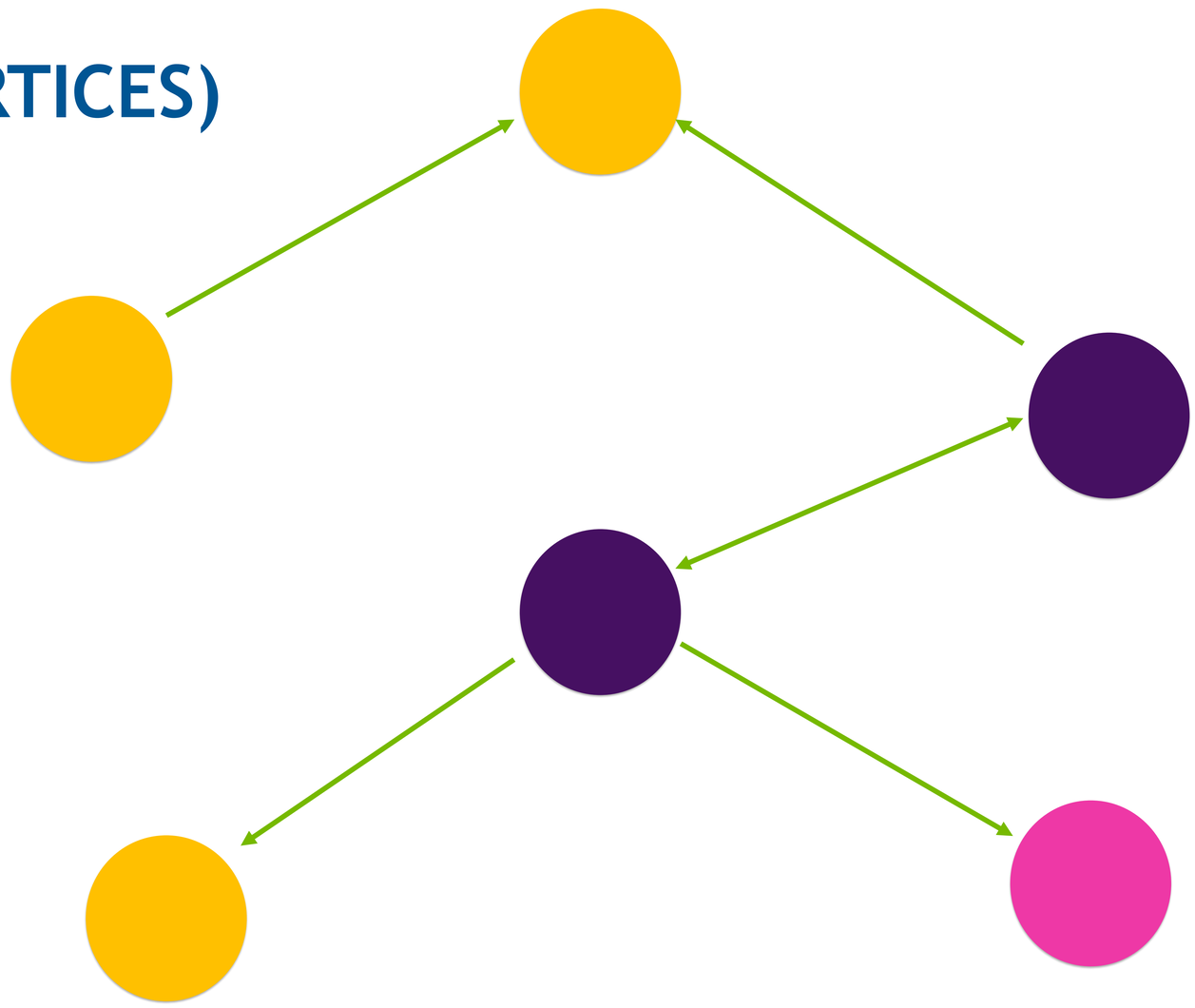



ME AFTER
LEARNING ABOUT GRAPHS
LITERALLY
EVERYTHING
IS THIS A GRAPH?
imgflip.com



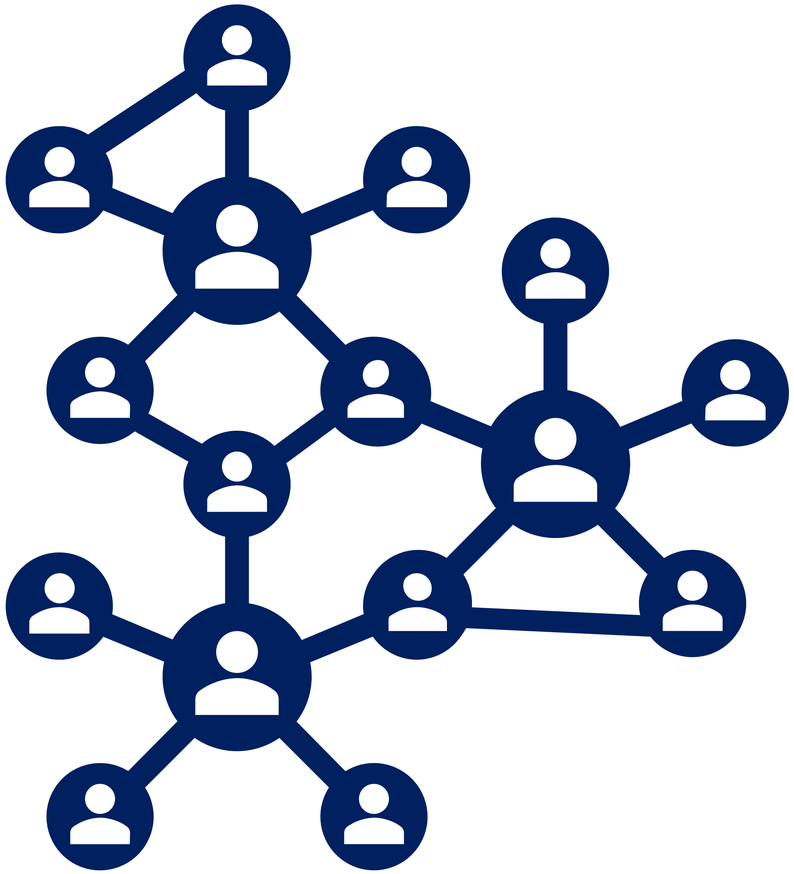
GRAPHS 101

NODES (VERTICES)

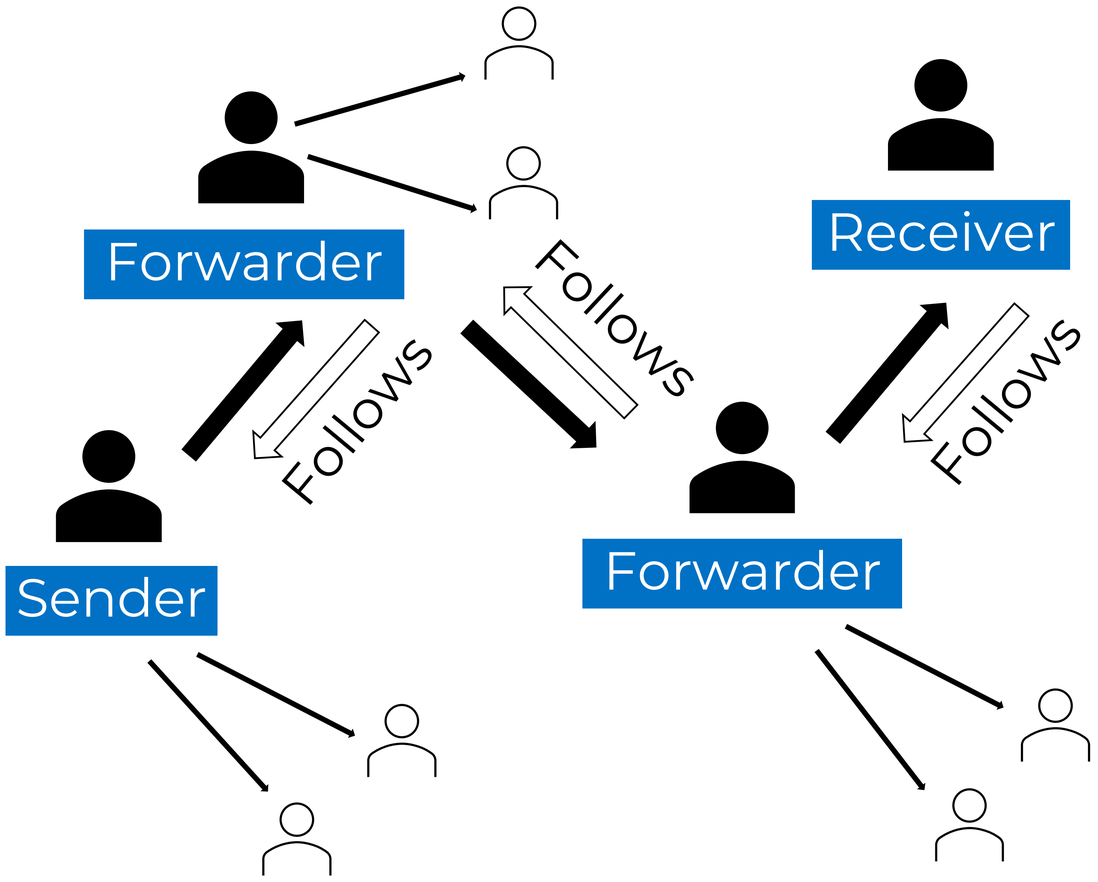


EDGES

DIRECTED VS UNDIRECTED



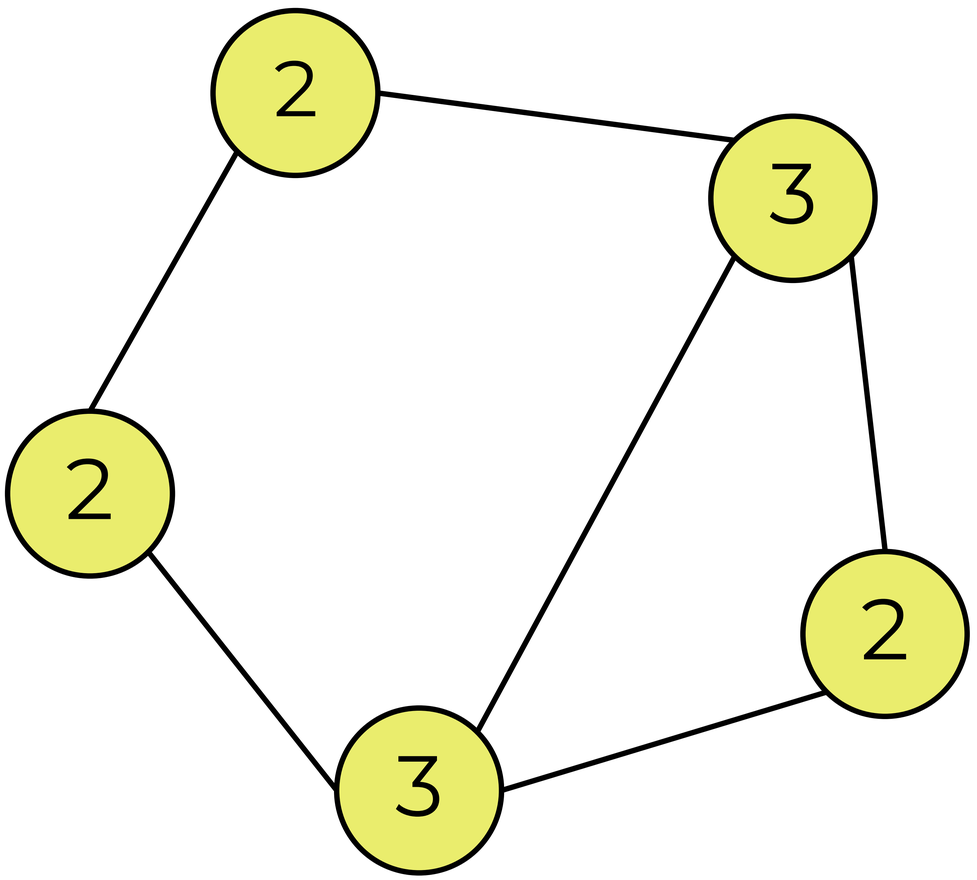
FACEBOOK

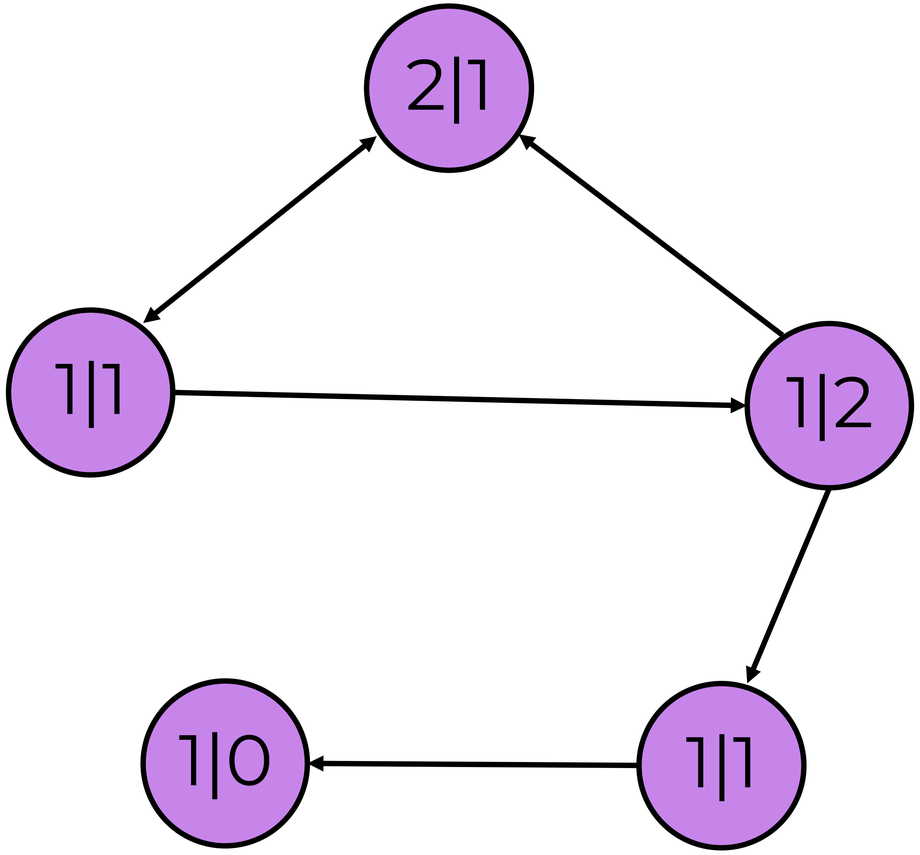


TWITTER

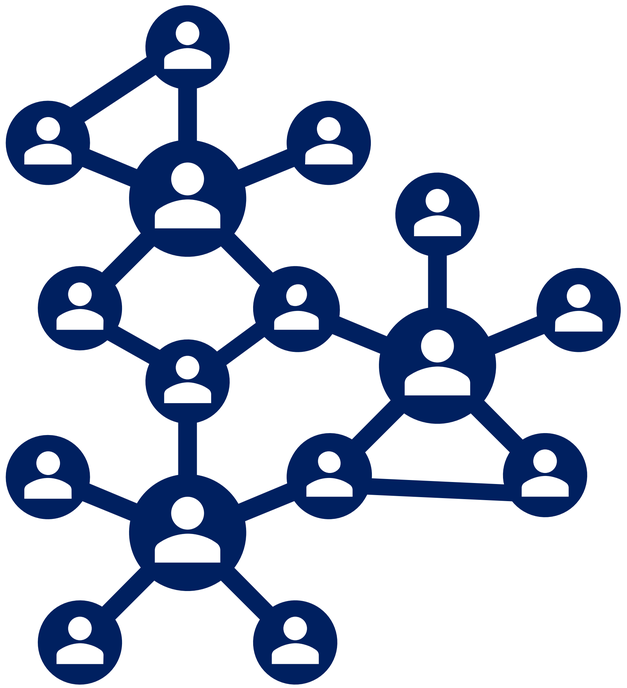
NVIDIA.

DEGREE, IN-DEGREE, OUT-DEGREE

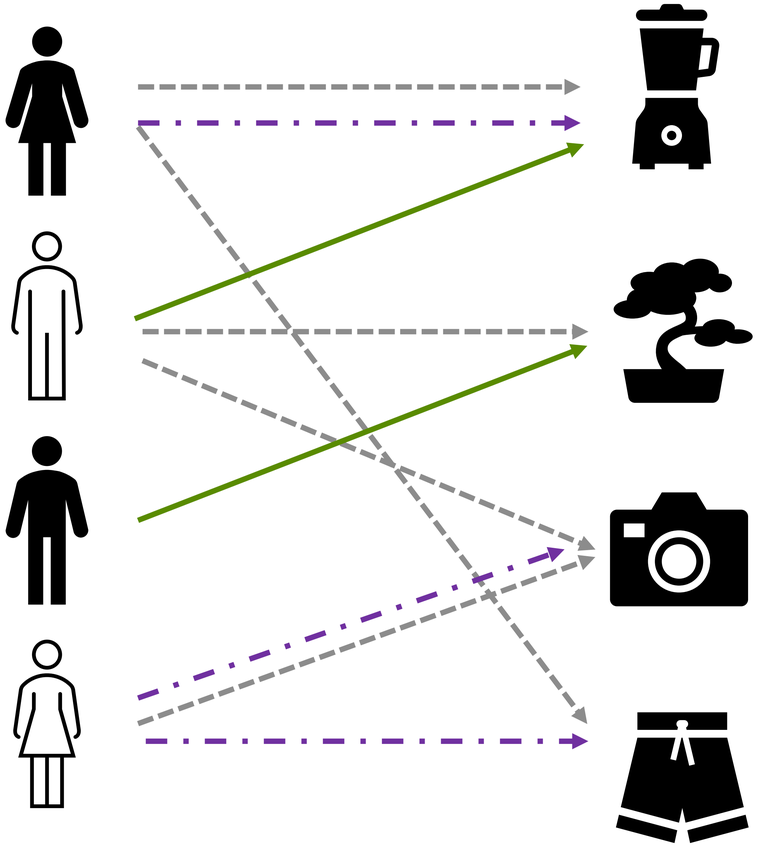




HOMOGENEOUS VS HETEROGENEOUS

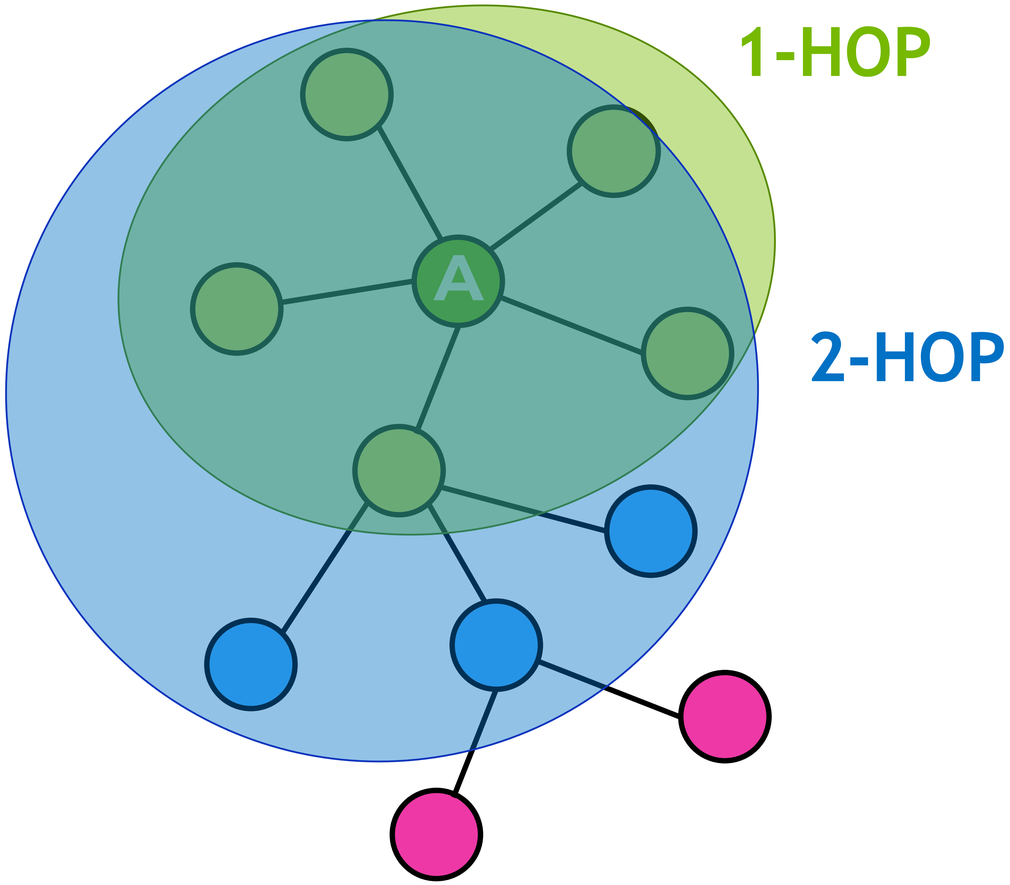


SOCIAL NETWORK



E-COMMERCE

NEIGHBORHOOD

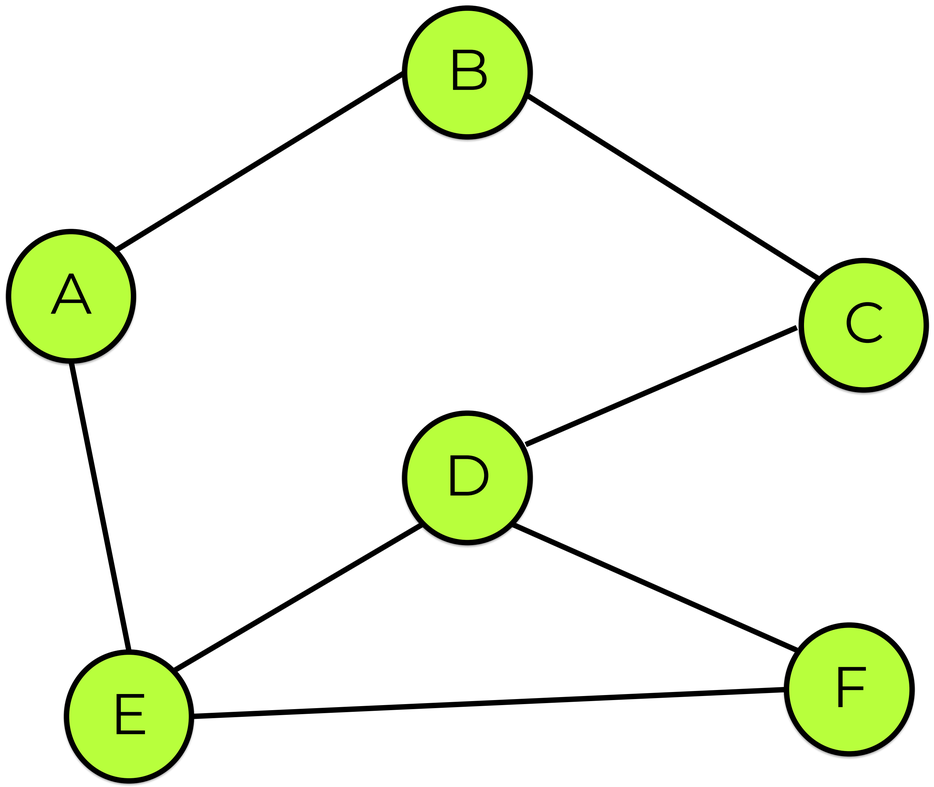


REPRESENTING GRAPHS

REPRESENTING GRAPHS

Adjacency matrix (A)

	A	B	C	D	E	F
A	0	1	0	0	1	0
B	1	0	1	0	0	0
C	0	1	0	1	0	0
D	0	0	1	0	1	1
E	1	0	0	1	0	1
F	0	0	0	1	1	0



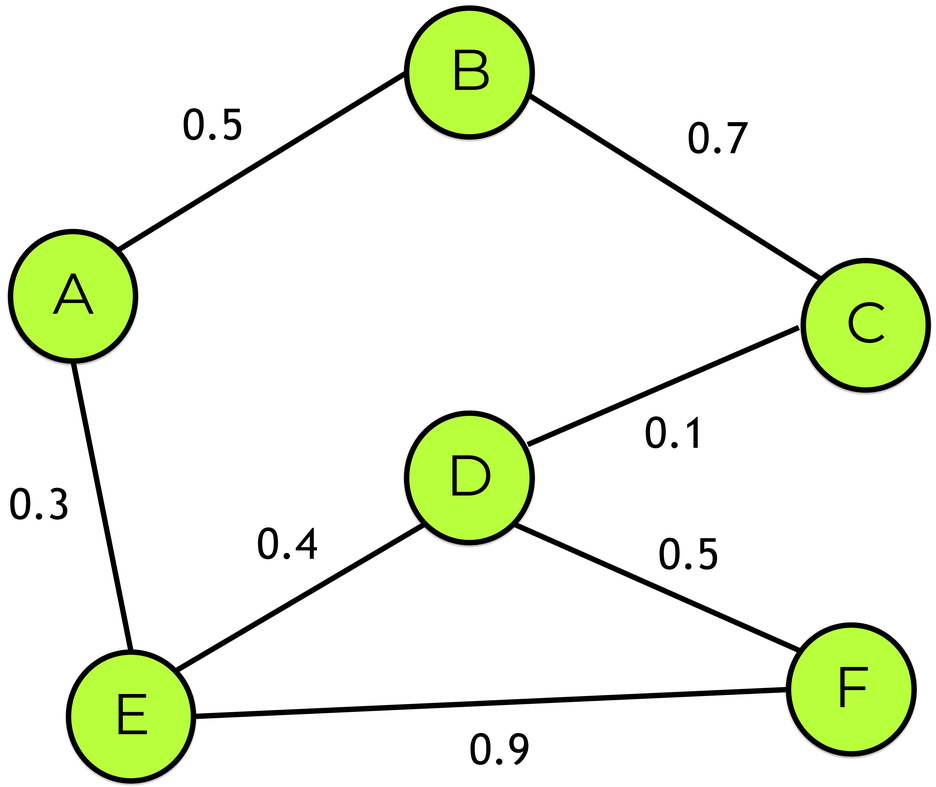
Collecting the neighbourhood of the each node

NVIDIA

REPRESENTING GRAPHS

Adjacency matrix (A) of weighted graph

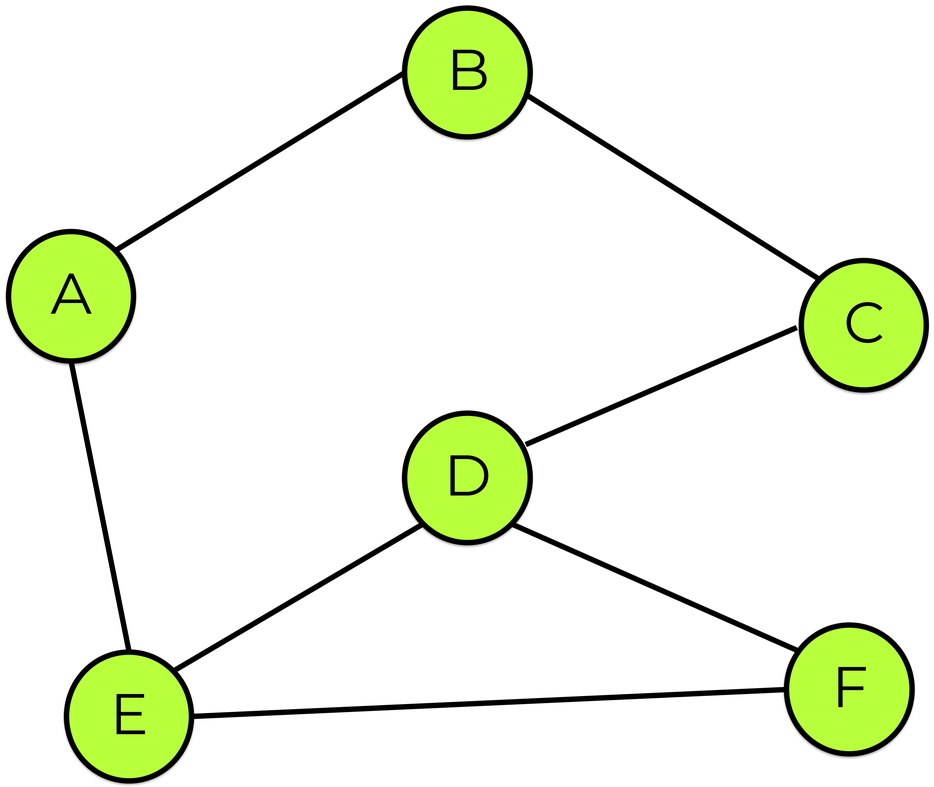
	A	B	C	D	E	F
A	0	0.5	0	0	0.3	0
B	0.5	0	0.7	0	0	0
C	0	0.7	0	0.1	0	0
D	0	0	0.1	0	0.4	0.5
E	0.3	0	0	0.4	0	0.9
F	0	0	0	0.5	0.9	0



REPRESENTING GRAPHS

Degree matrix (D)

	A	B	C	D	E	F
A	2	0	0	0	0	0
B	0	2	0	0	0	0
C	0	0	2	0	0	0
D	0	0	0	3	0	0
E	0	0	0	0	3	0
F	0	0	0	0	0	2

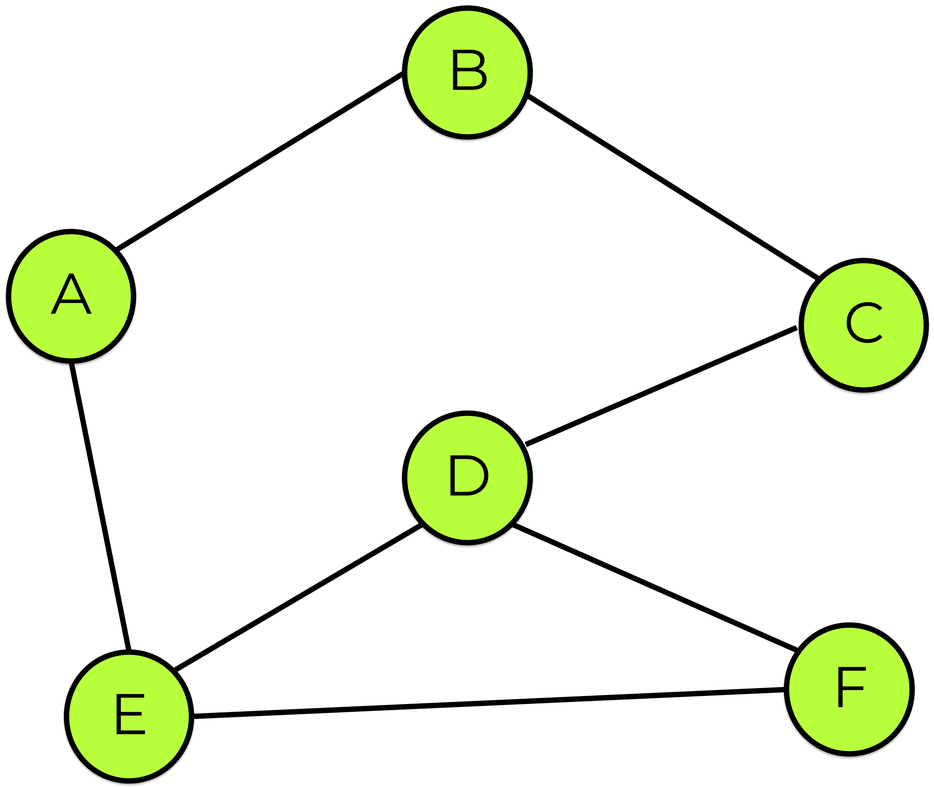


Count of how many edges are connected to each node

REPRESENTING GRAPHS

Laplacian matrix (L = { D – A })

	A	B	C	D	E	F
A	2	-1	0	0	-1	0
B	-1	2	-1	0	0	0
C	0	-1	2	-1	0	0
D	0	0	-1	3	-1	-1
E	-1	0	0	-1	3	-1
F	0	0	0	-1	-1	2



Tells how smooth the graph is

REPRESENTING GRAPHS

Adjacency list

Adjacency

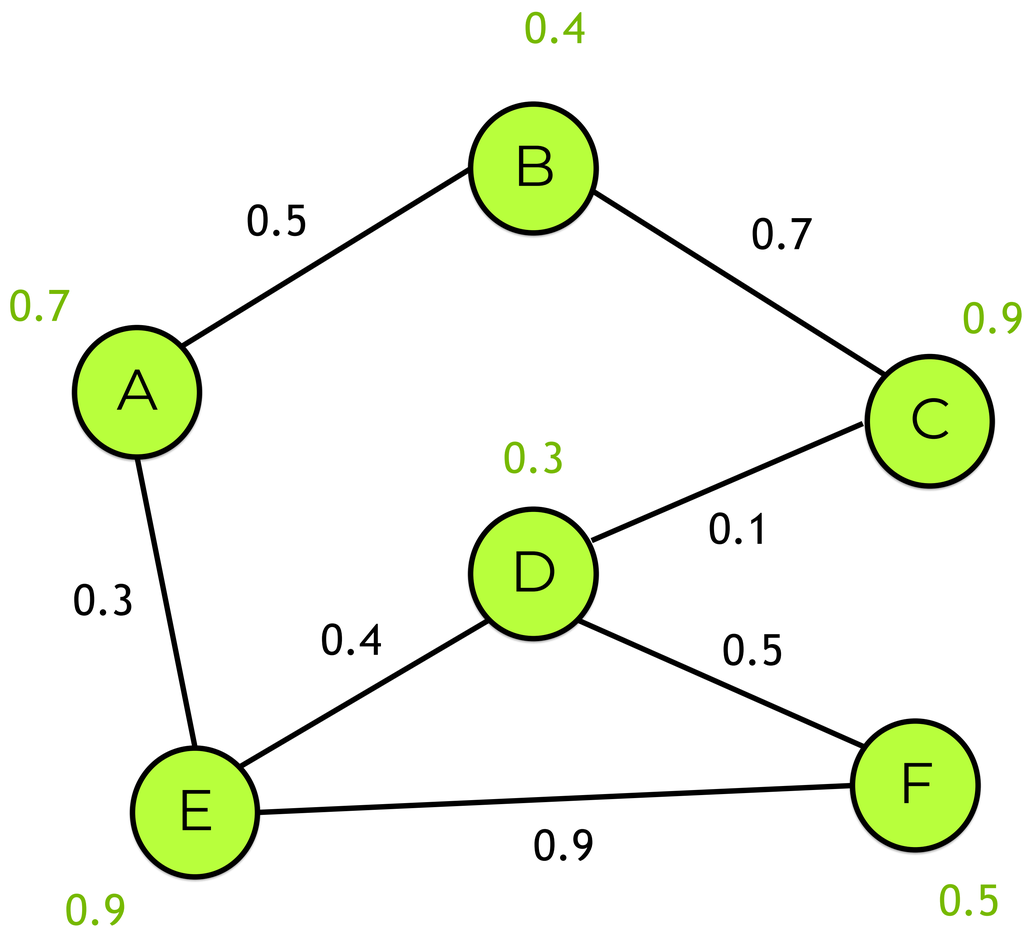
```
[
    [1, 2],
    [1, 5],
    [2, 3],
    [3, 4],
    [4, 5],
    [4, 6],
    [5, 6]
]
```

Nodes

```
[
    0.7,
    0.4,
    0.9,
    0.3,
    0.9,
    0.5
]
```

Edges

```
[
    0.5,
    0.3,
    0.7,
    0.1,
    0.4,
    0.5,
    0.9
]
```



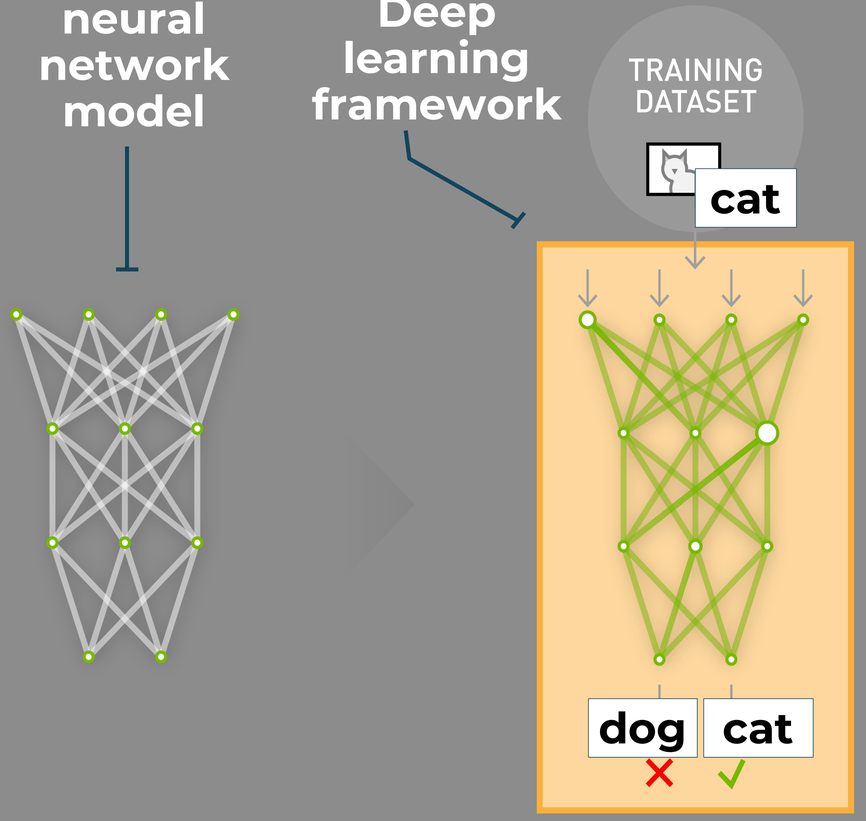
AI: LEARNING FROM DATA

TRAINING

Learning a new capability from existing data

Untrained neural network model

Deep learning framework

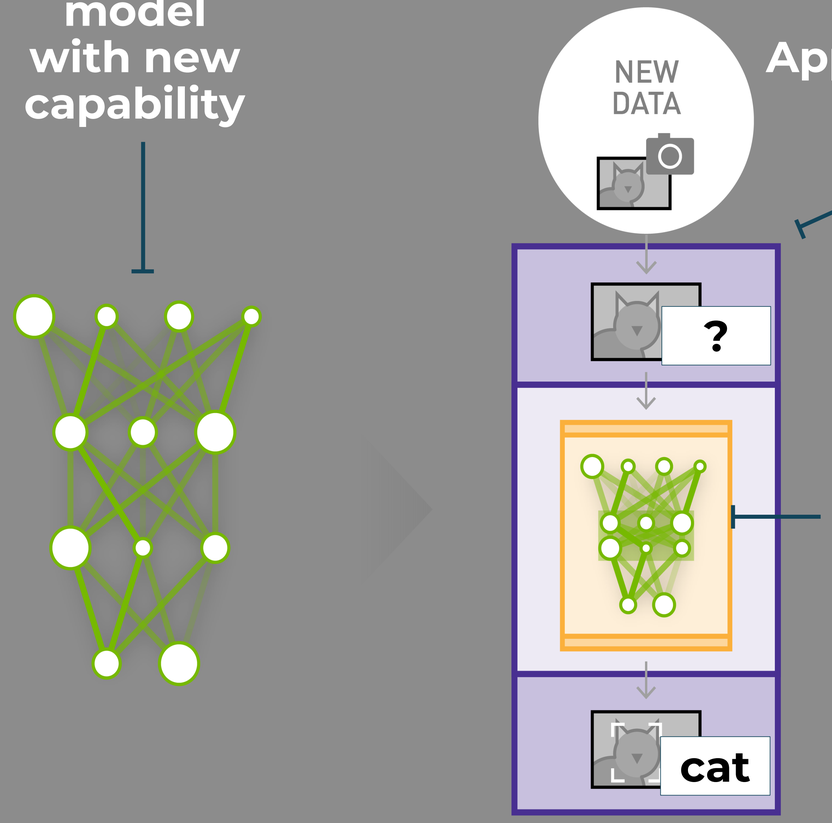


Trained model with new capability

INFERENCE

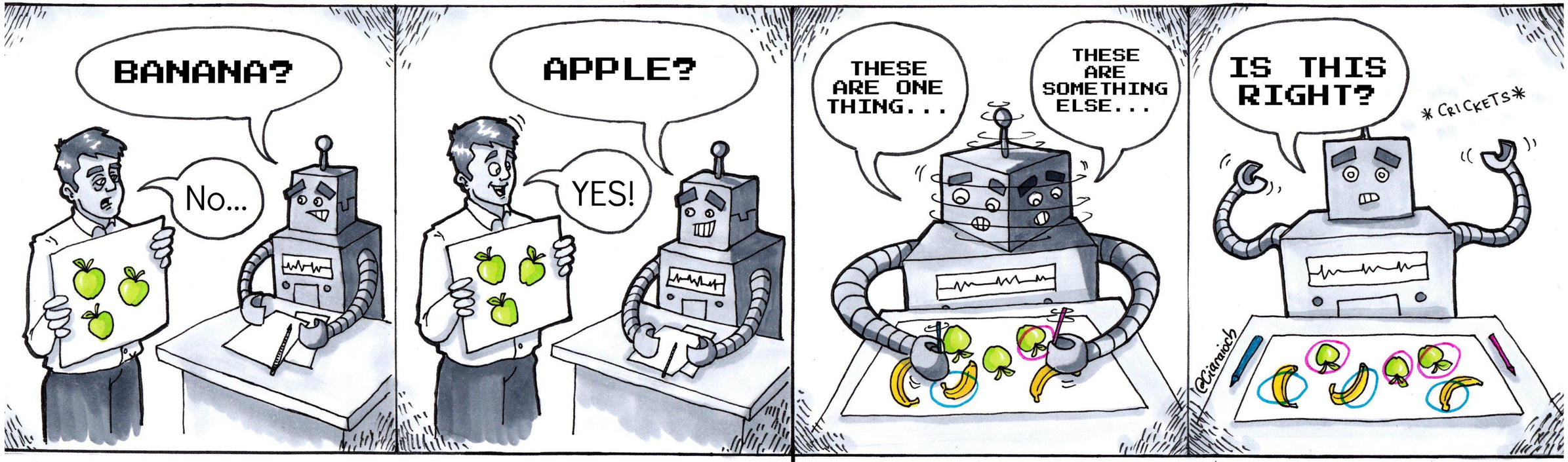
Applying this capability to new data

App or service



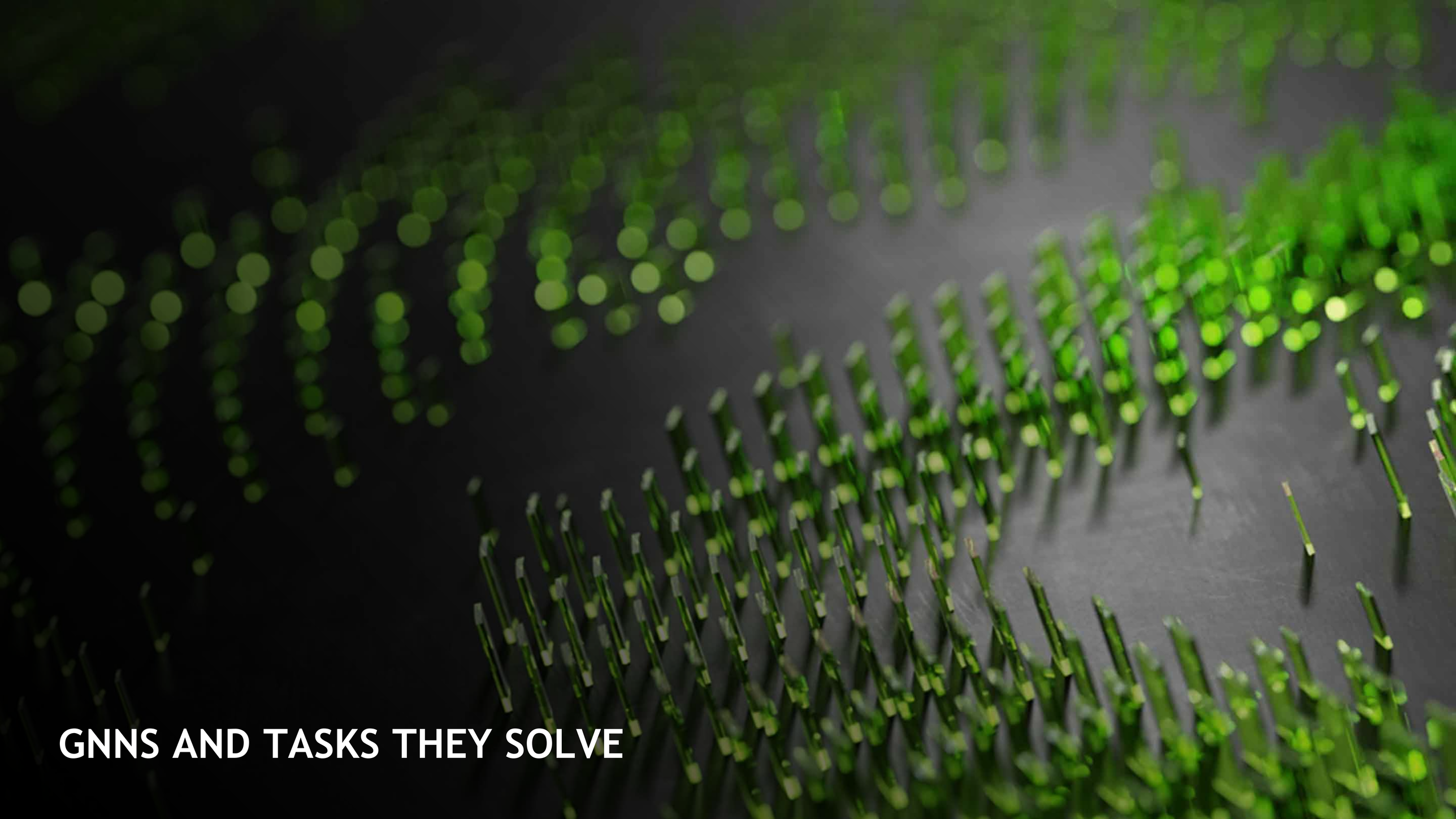
Trained model

NVIDIA

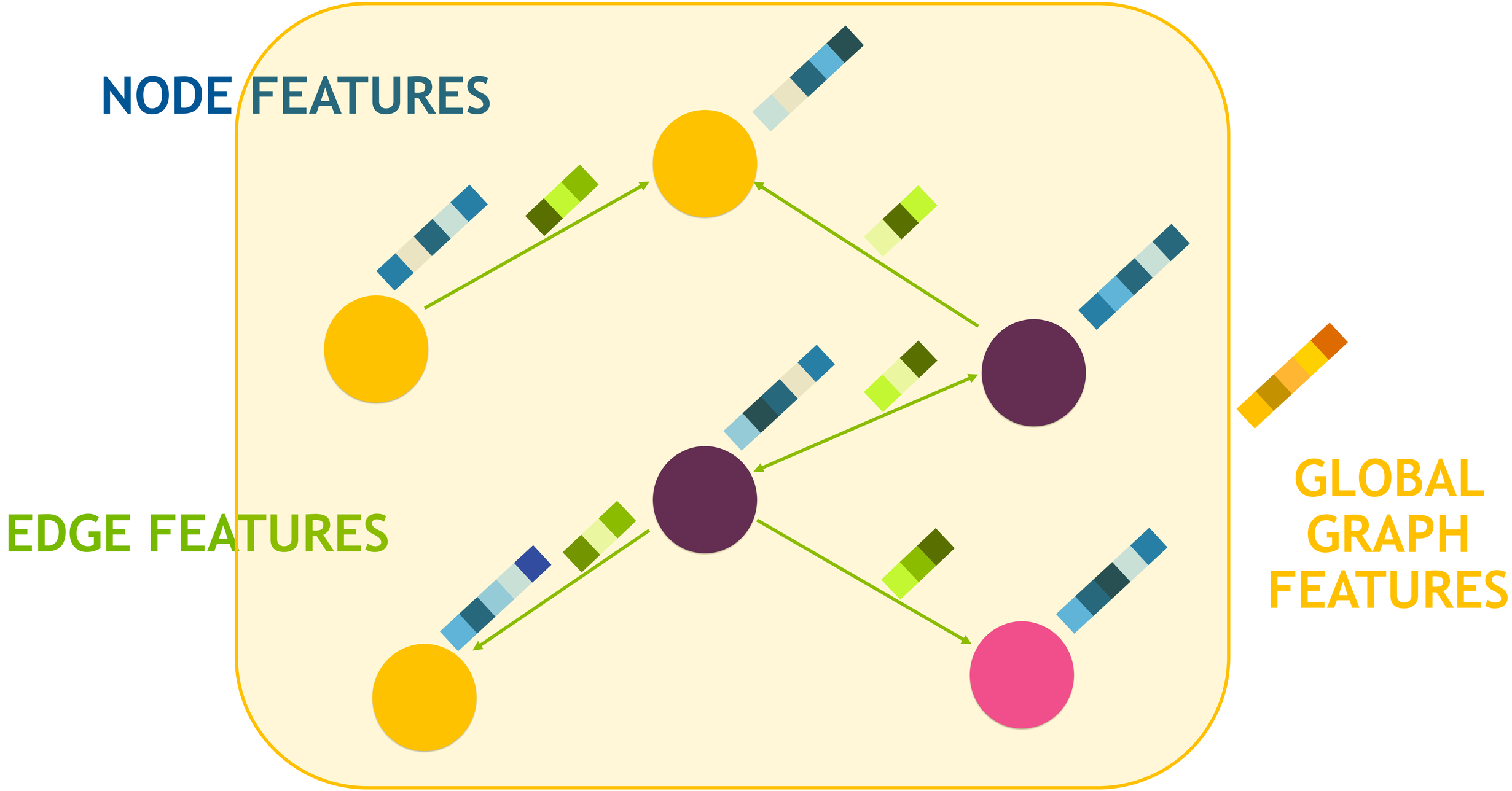


Supervised Learning | **Unsupervised Learning**

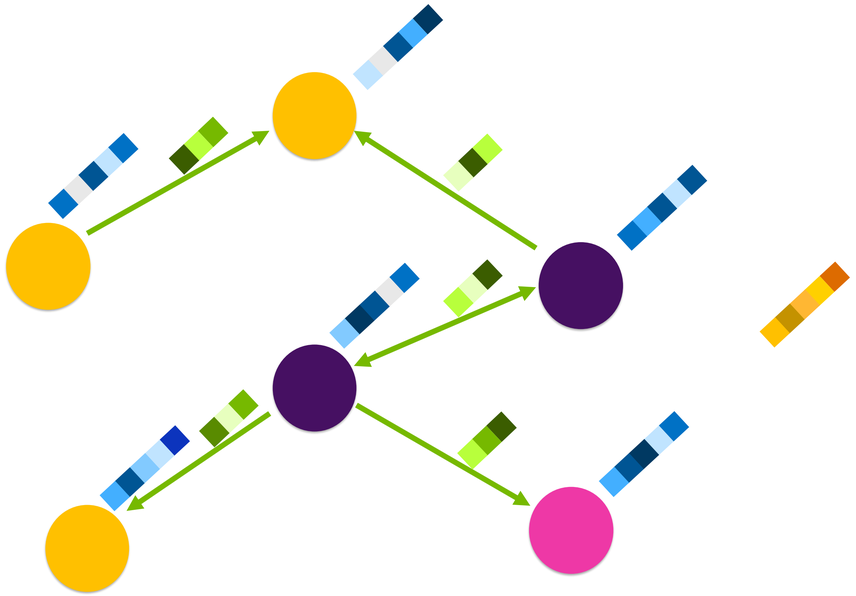
Image credit: @Ciaraioch



GNNS AND TASKS THEY SOLVE


NODE FEATURES
EDGE FEATURES
GLOBAL GRAPH FEATURES

INPUT



Learnable function
(Graph Neural Net)

OUTPUT

New node features

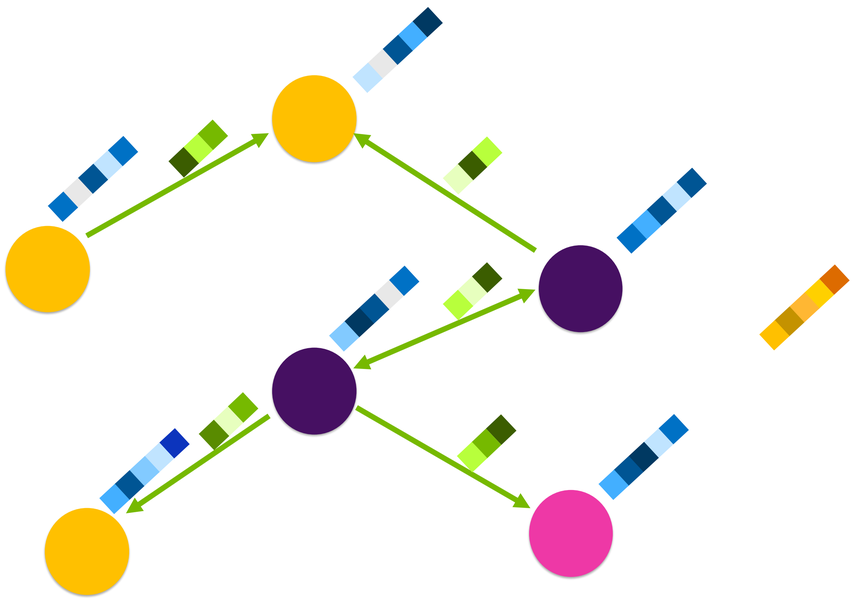
New edge features

New global graph features

NVIDIA

GRAPH-LEVEL TASKS

INPUT



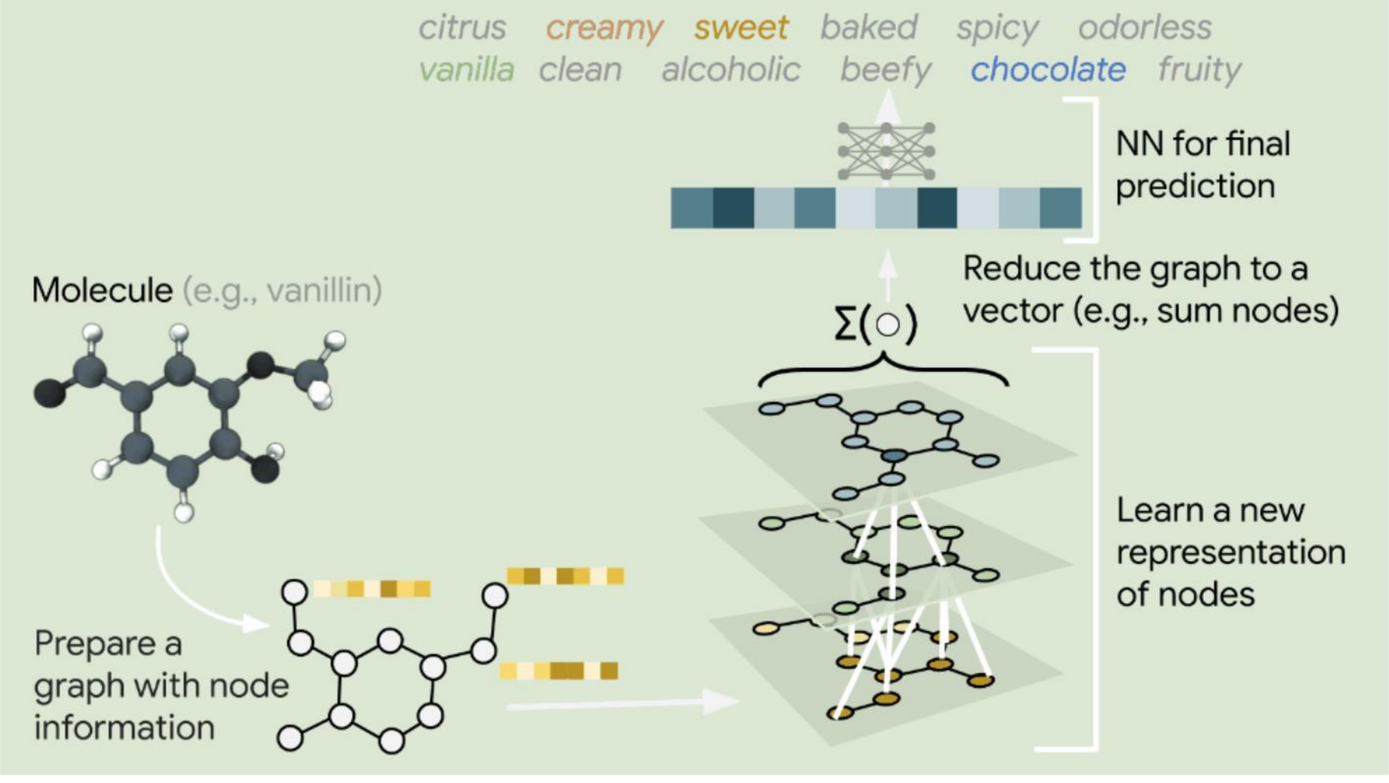
Learnable function
(Graph Neural Net)

OUTPUT

New global
graph features

NVIDIA

GRAPH-LEVEL TASKS
Classification



Experiment by Google: predict the smell of molecules by framing molecules as graphs

Image credit: ai.googleblog.com

GRAPH-LEVEL TASKS
Regression

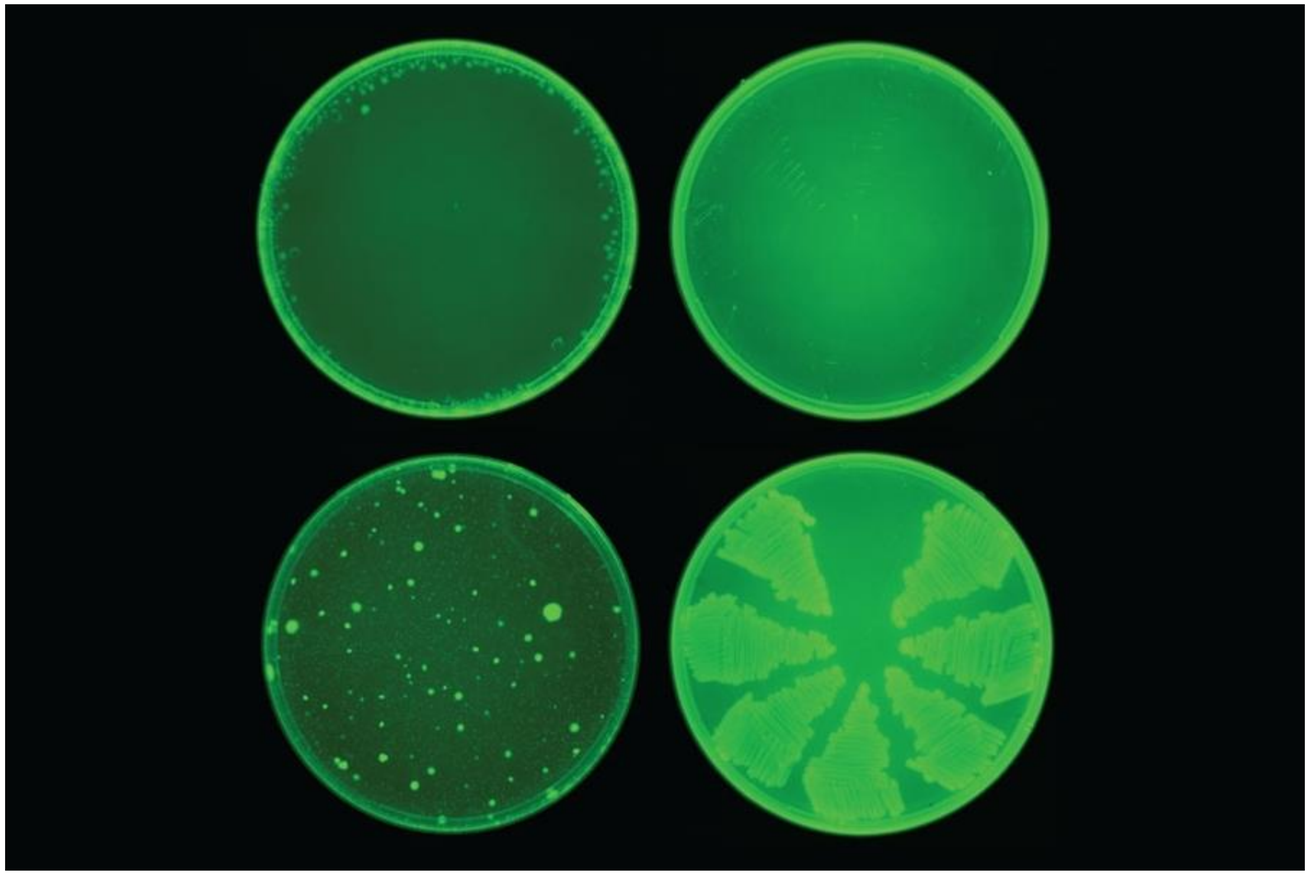


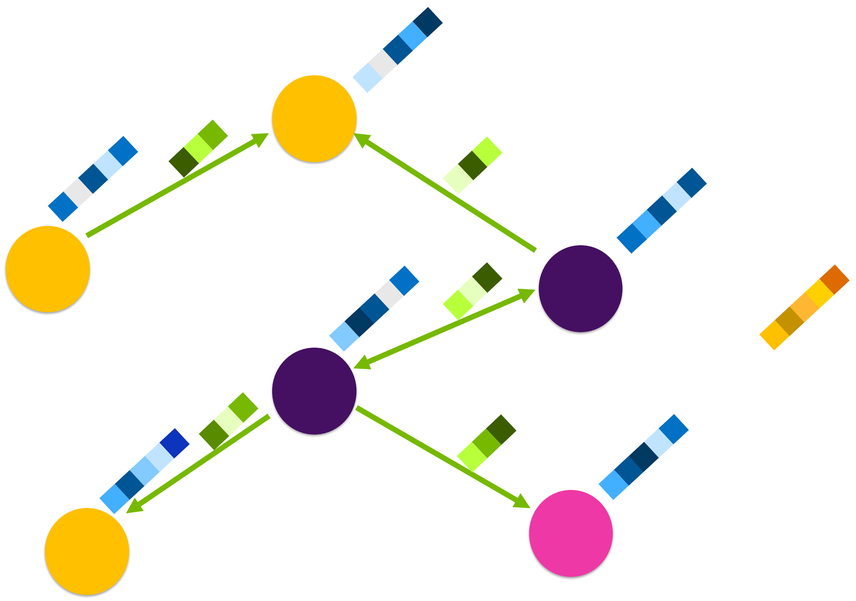
Image credit: news.mit.edu

Predictive rating of molecule toxicity

Discovery by MIT: *Halicin*, a new antibiotic compound effective against multi-resistant bacteria

NODE-LEVEL TASKS

INPUT



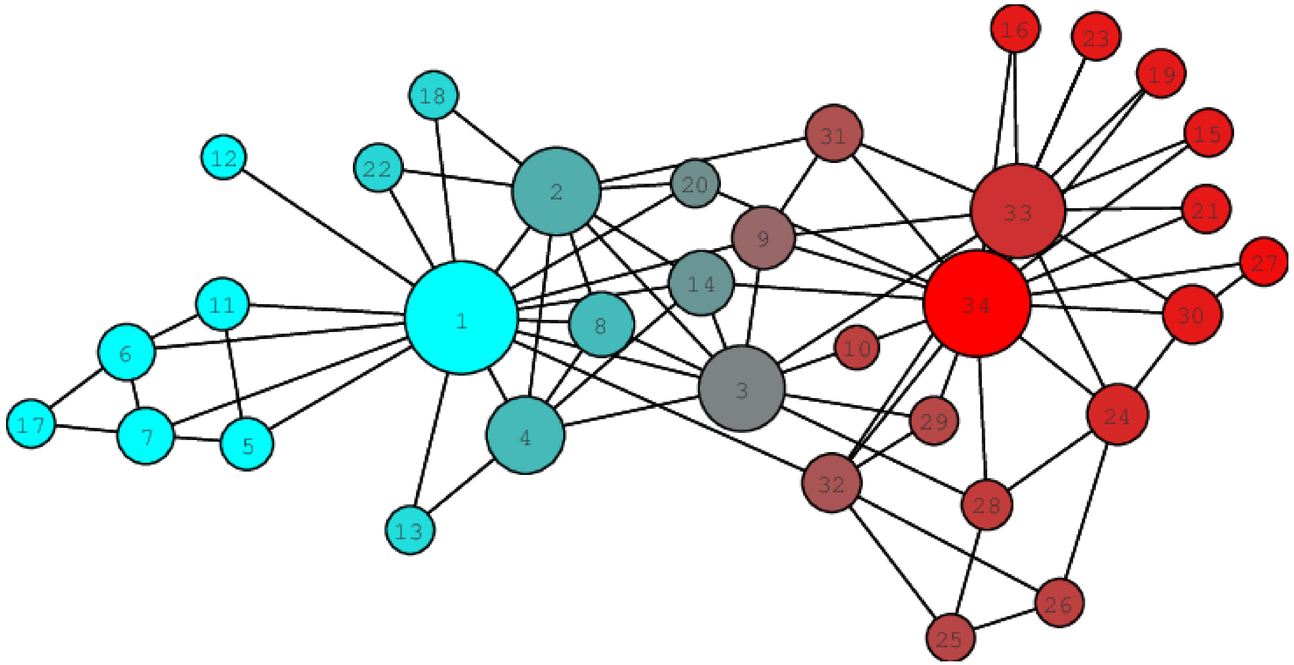
Learnable function
(Graph Neural Net)

OUTPUT

New node
features

NODE-LEVEL TASKS
Classification

Zachary Karate Club – the ‘Hello World’ of GNNs



NODE-LEVEL TASKS
Regression

Predict values in each node: continuous outcome variables

Research at Cornell: utilizing geospatial and temporal information for crop yield prediction

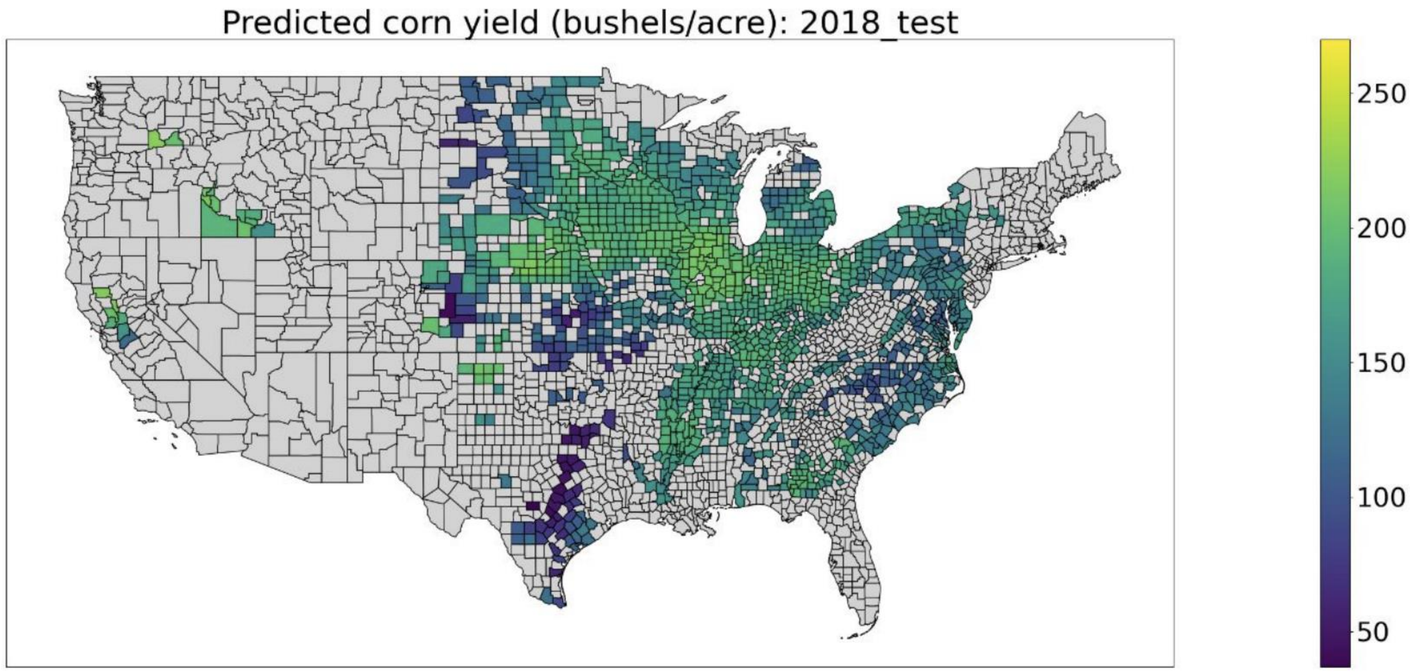
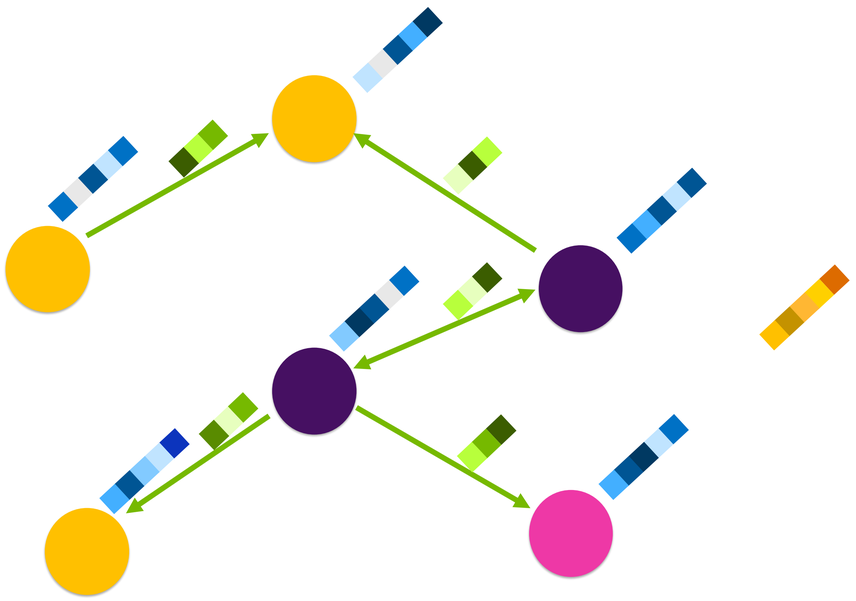


Image credit: proceeding NeurIPS 2021

EDGE-LEVEL TASKS

INPUT



Learnable function
(Graph Neural Net)

OUTPUT

New edge
features

EDGE-LEVEL TASKS
Link prediction

E-commerce / digital media: recommend product / content A to a customer / user B

Social network: recommend friends

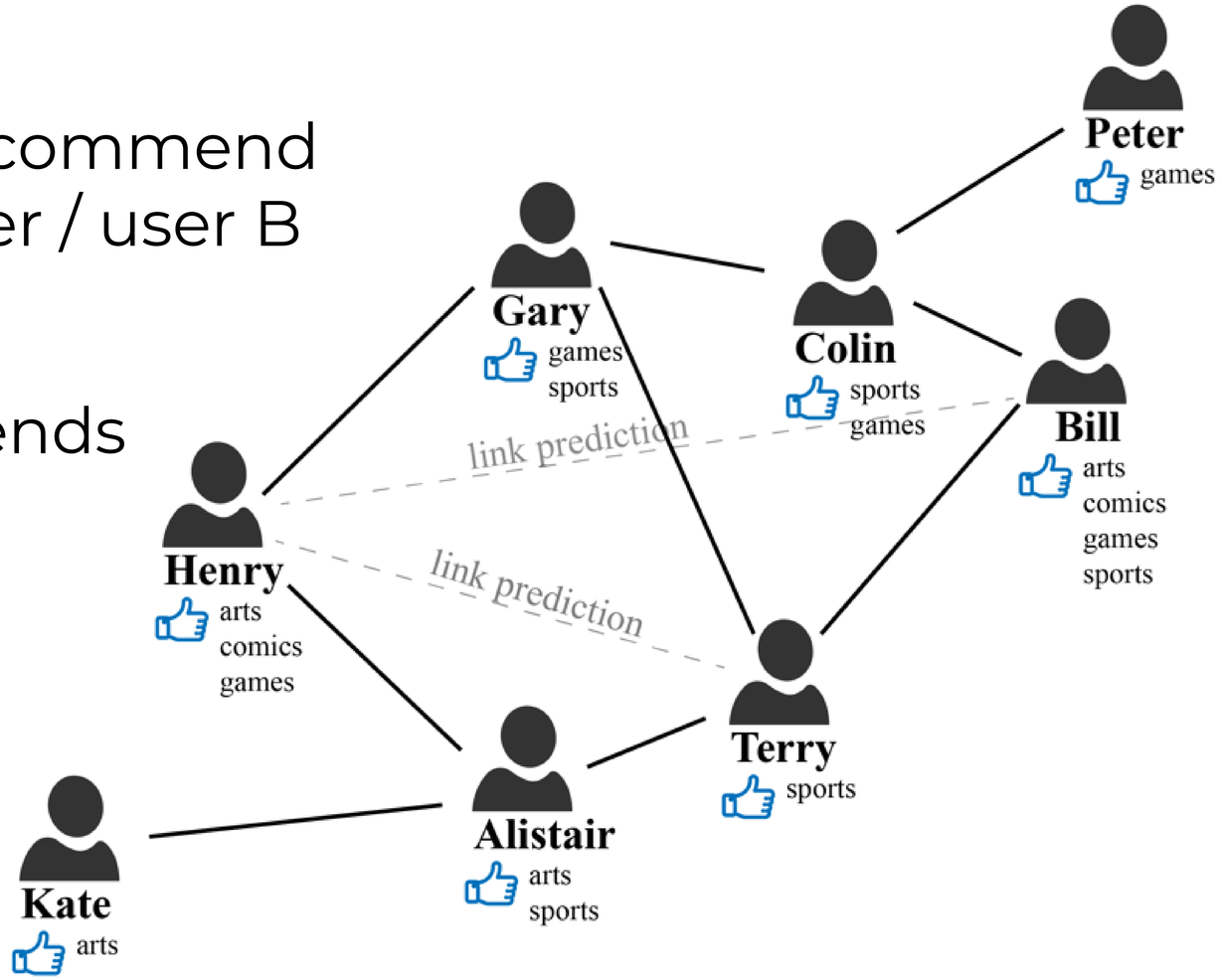


Image credit: aws.amazon.com/blogs

EDGE-LEVEL TASKS
Classification

Predict fradulent transactions:

APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions

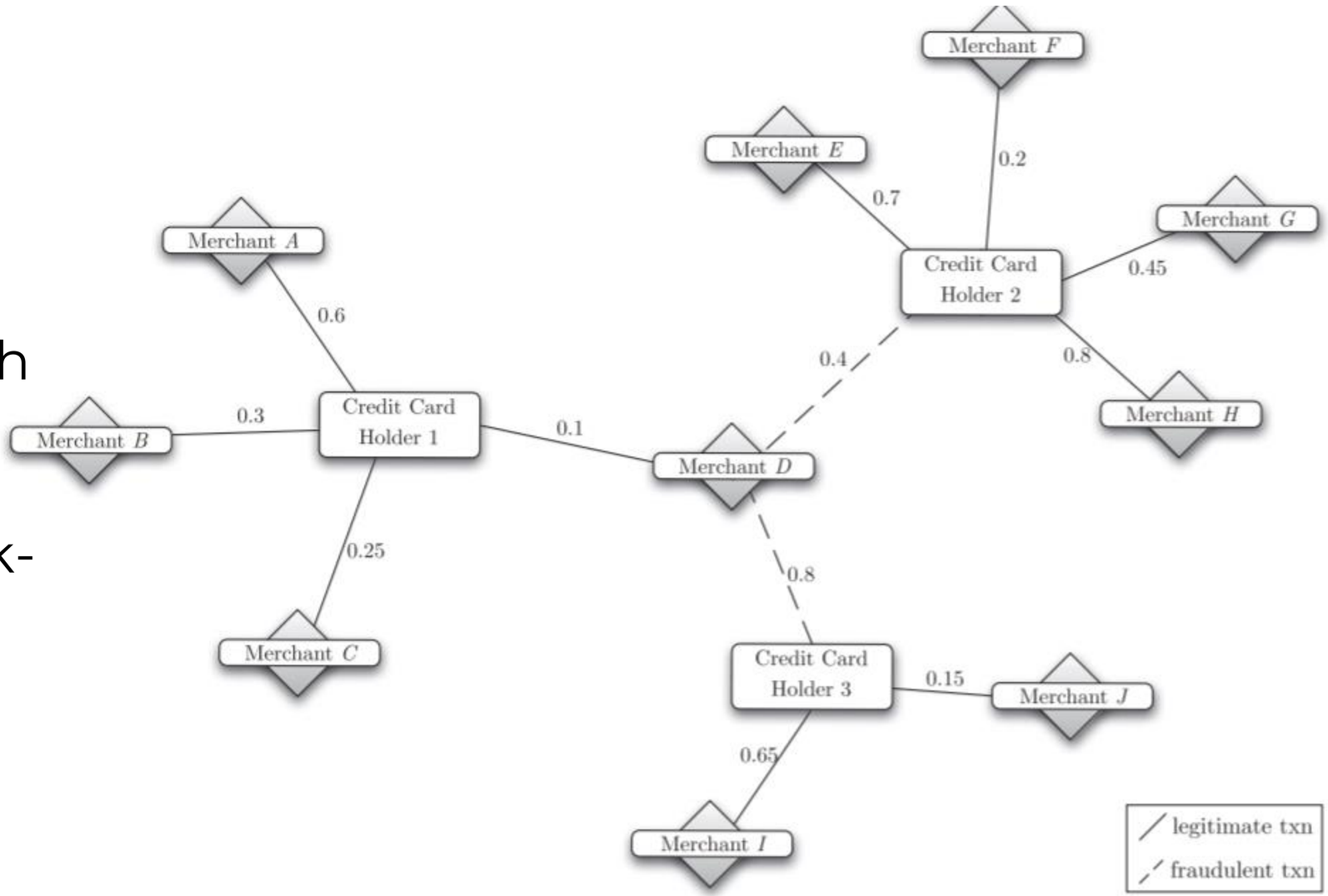
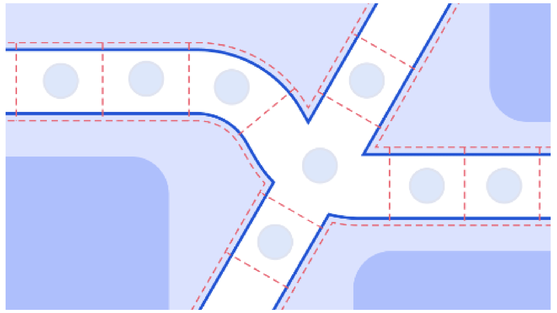


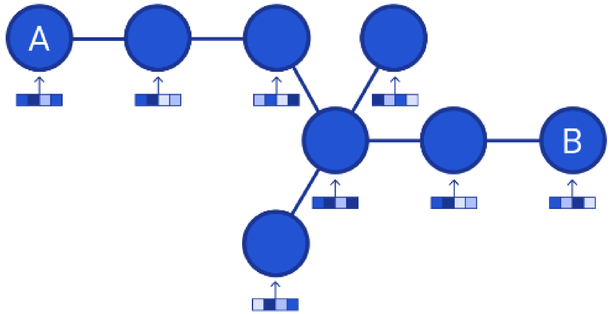
Image credit: Decision Support Systems 2015

EDGE-LEVEL TASKS

Regression

Google Maps: estimate time to travel from A to B





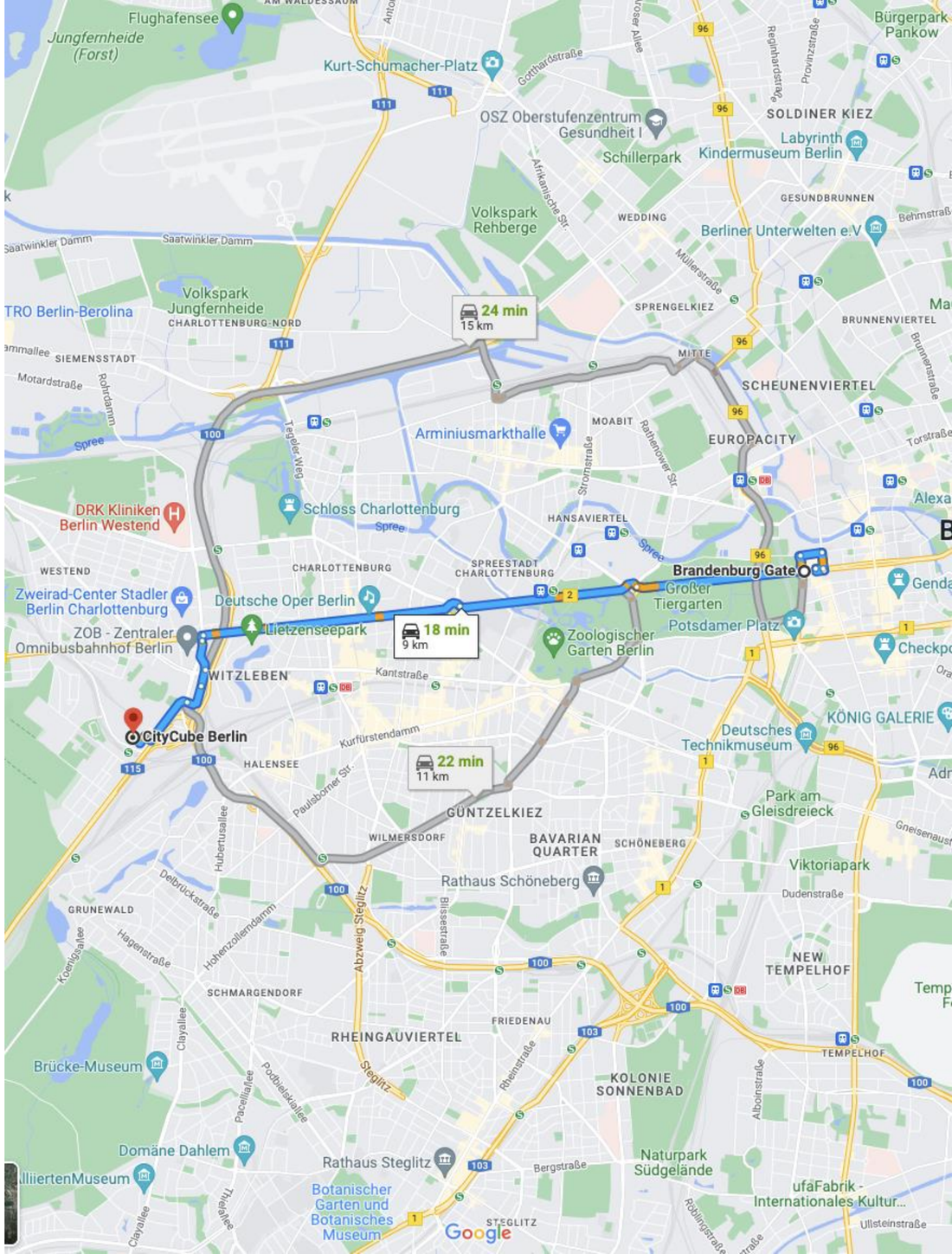
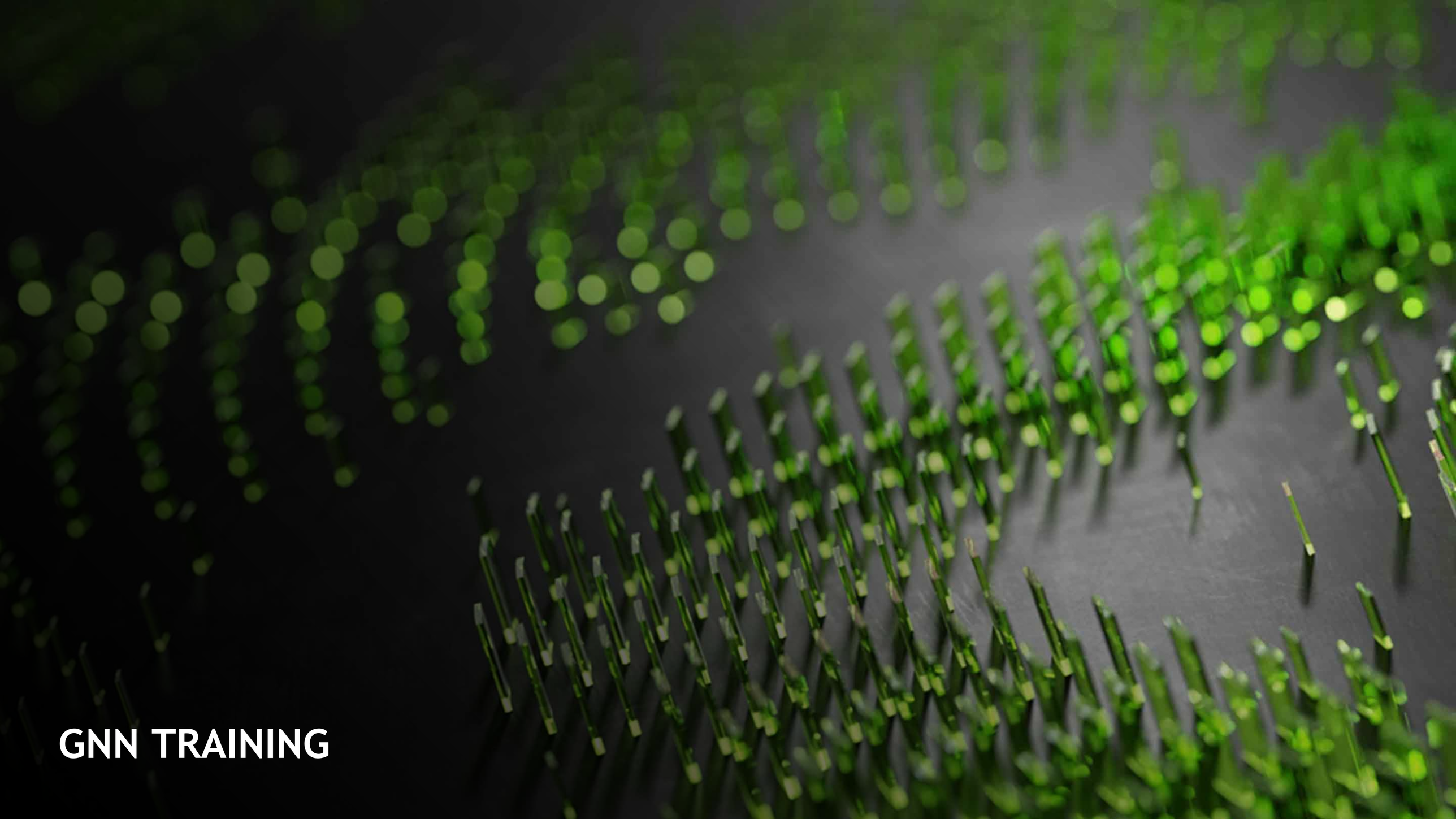


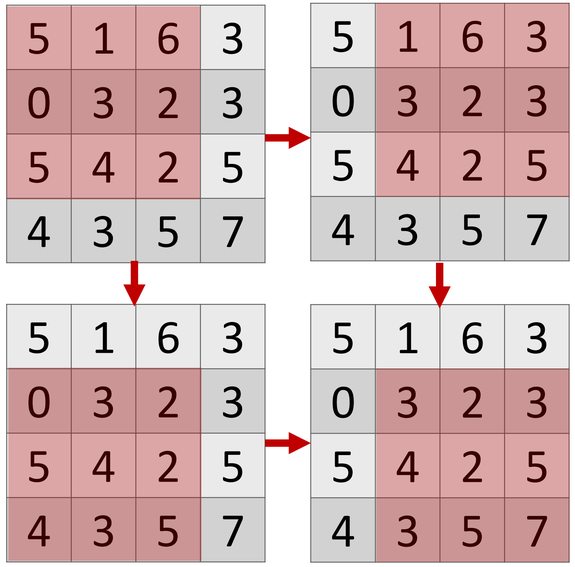
Image credit: DeepMind



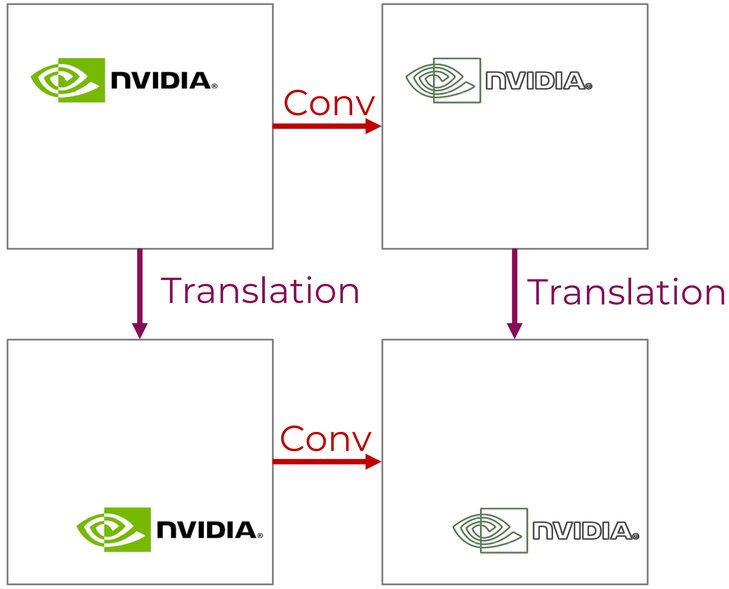
GNN TRAINING

HOW DO WE PERFORM CONVOLUTION?

CONVOLUTION ON REGULAR DOMAIN



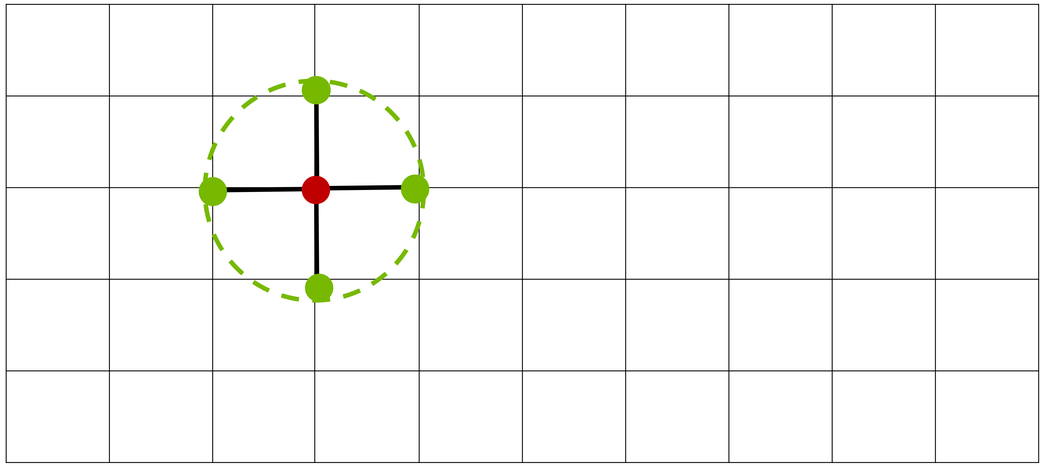
TRANSLATION EQUIVARIANCE



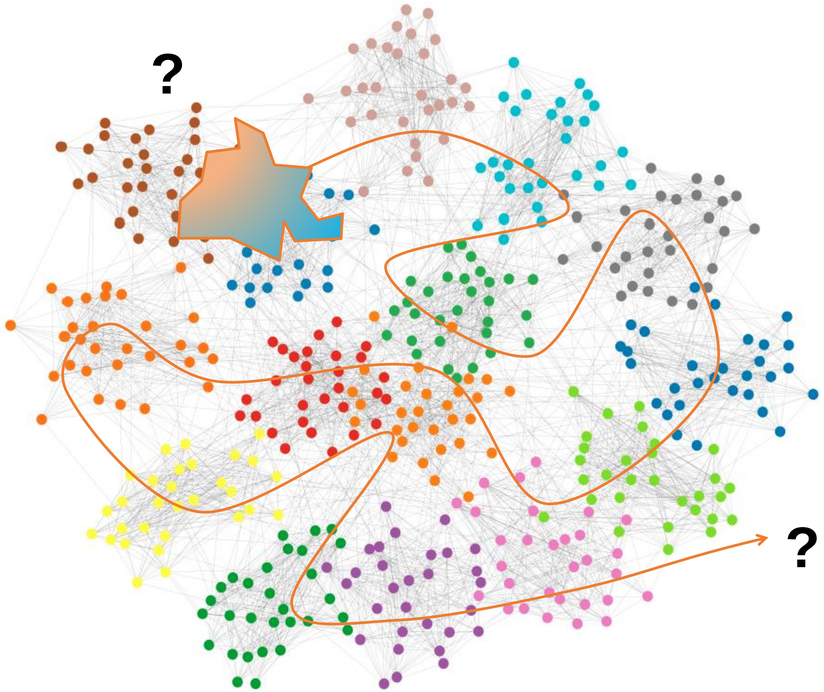
Translation equivariance: *translating the input before filtering is equivalent to translating the filter output of the untranslated input*

NO NOTION OF TRANSLATION ON A GRAPH

REGULAR GRID



CONVOLUTION ON IRREGULAR DOMAIN?



General convolution = aggregating and mapping neighborhood information for each node

AGGREGATING NEIGHBORHOOD INFORMATION: EMBEDDINGS ON GRAPHS

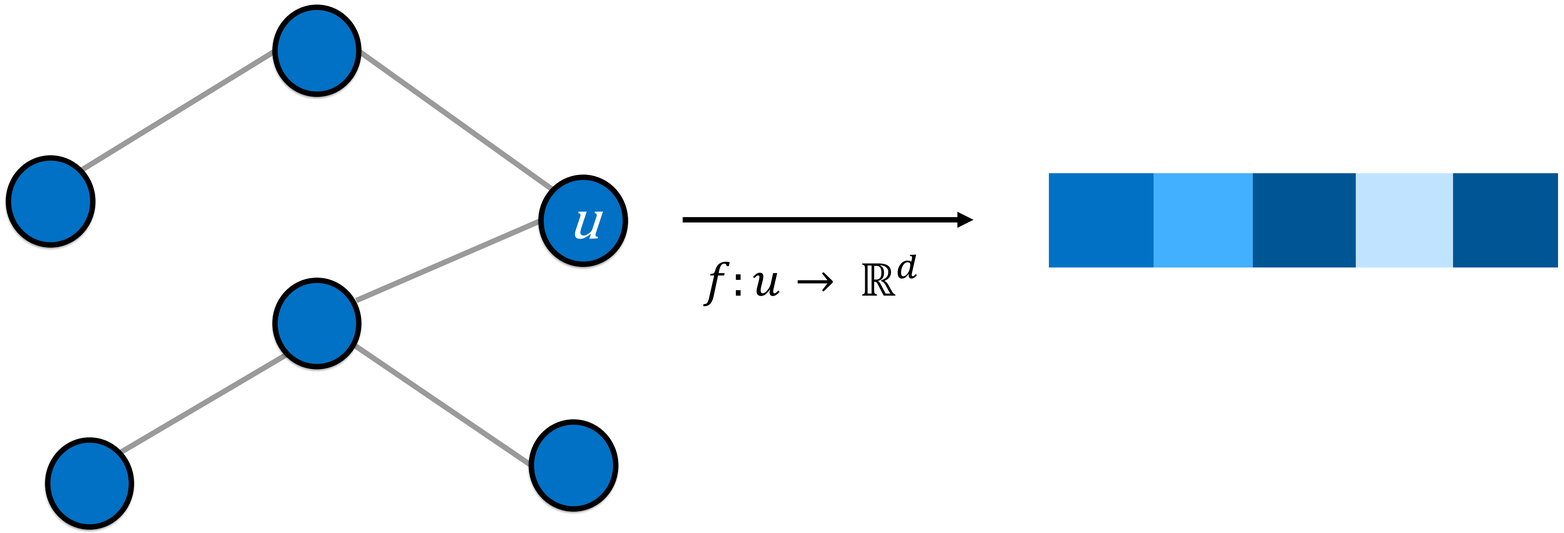
*The core of GCNs is the **neighborhood aggregation** scheme which generates node **embedding** by combining **information** from **neighborhoods***

Encoding nodes to d-dimensional *embeddings*

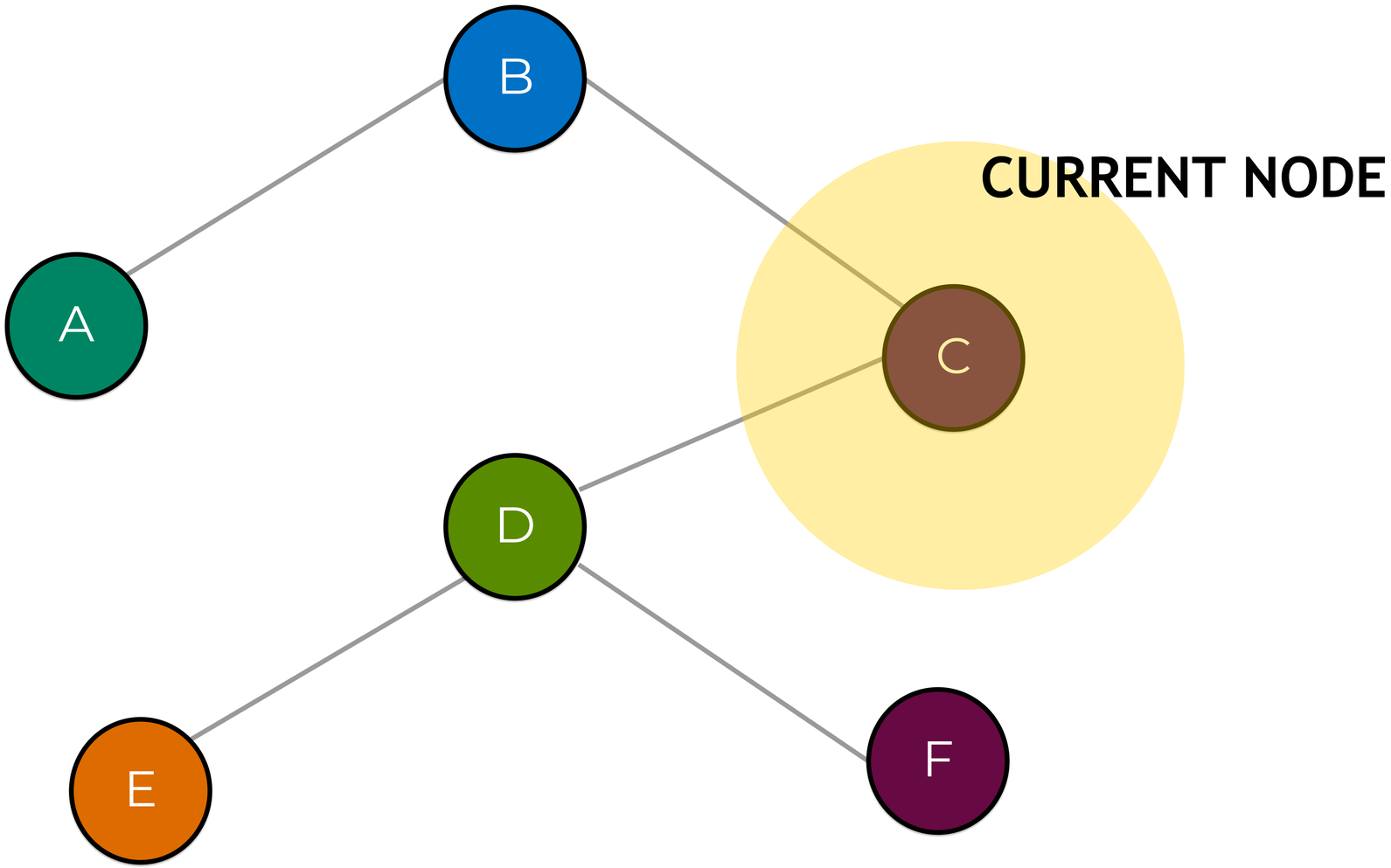
GNN is a deep graph encoder

Input: graph

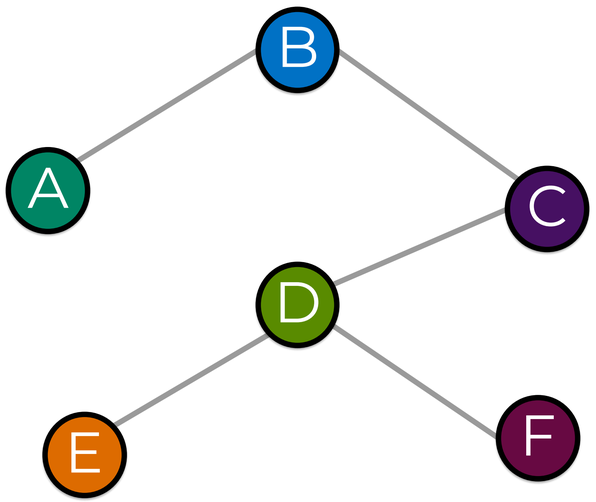
Output: embeddings

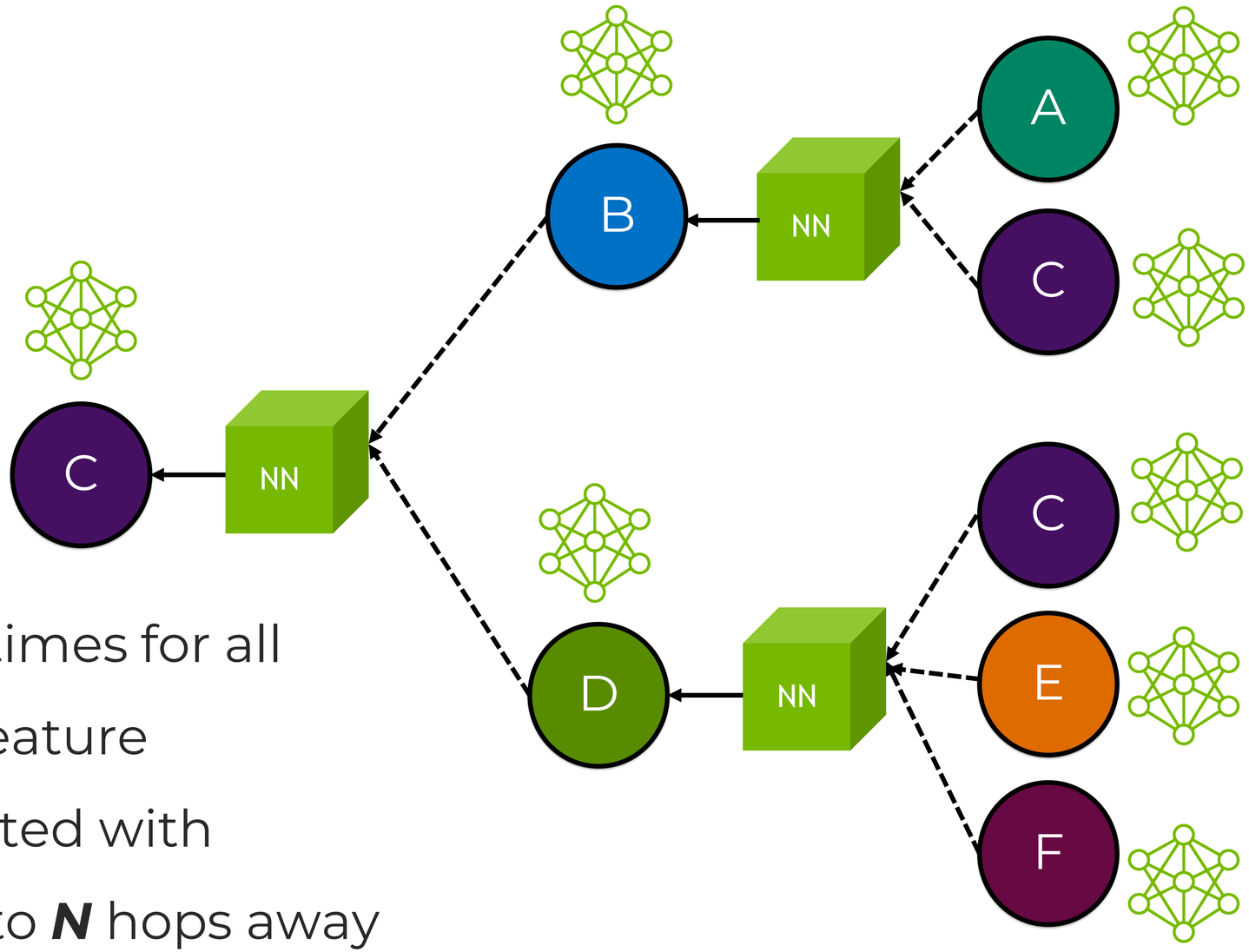


INPUT GRAPH



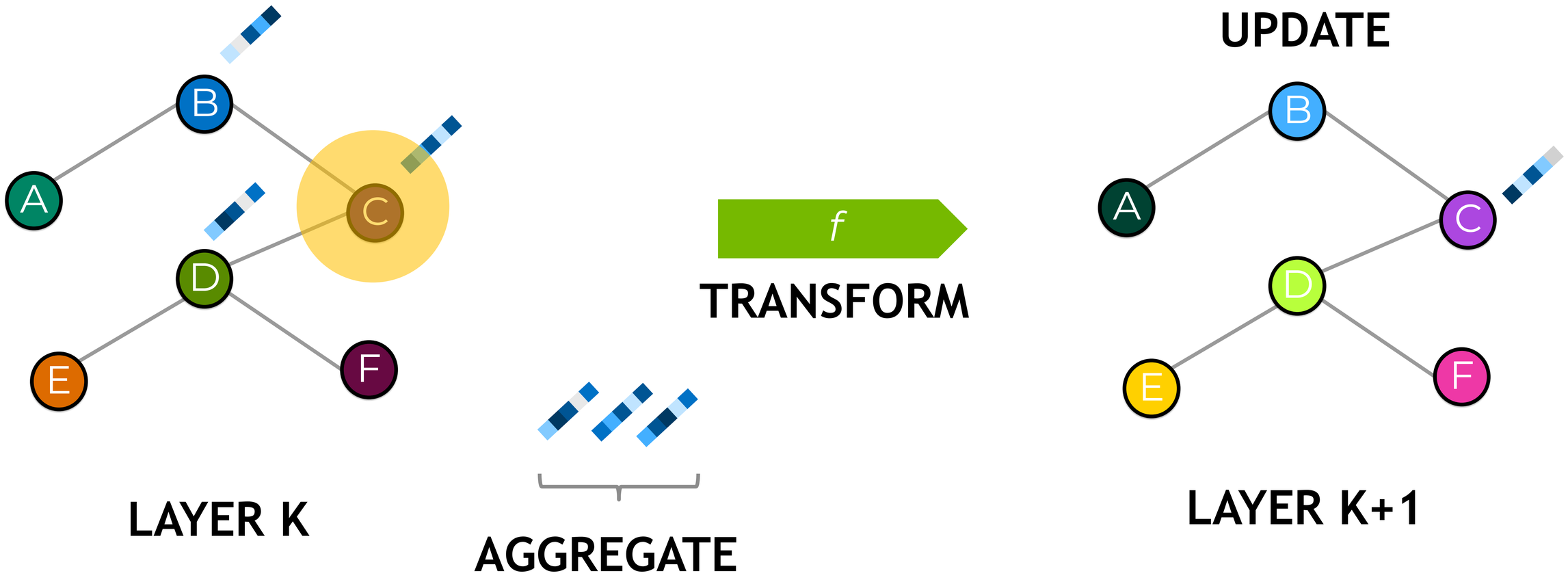
MESSAGE PASSING LAYER





By repeating this step ***N*** times for all nodes in the graph, the feature representations are updated with information of nodes up to ***N*** hops away

STACKING GNN LAYERS



GNN TRAINING

Same concepts as in classical CNN training

Activations

Normalization

Loss function

Gradient descent

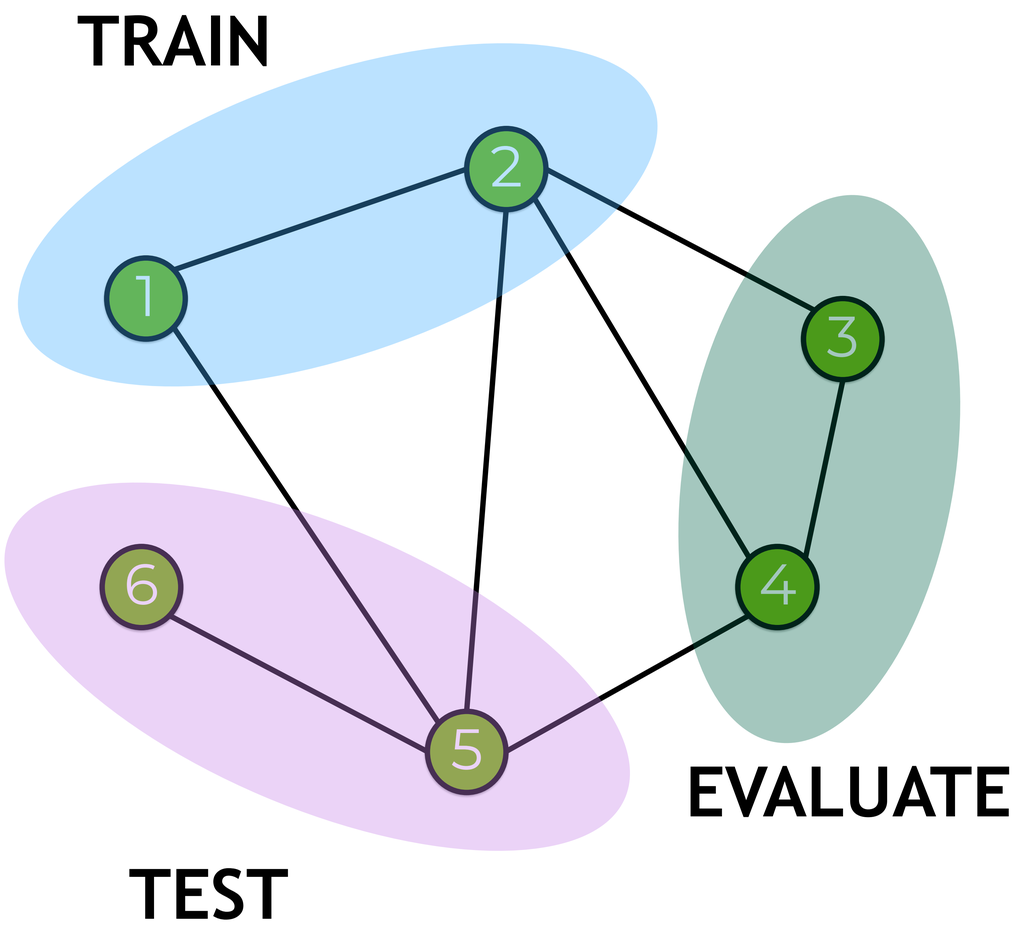
Backpropagation

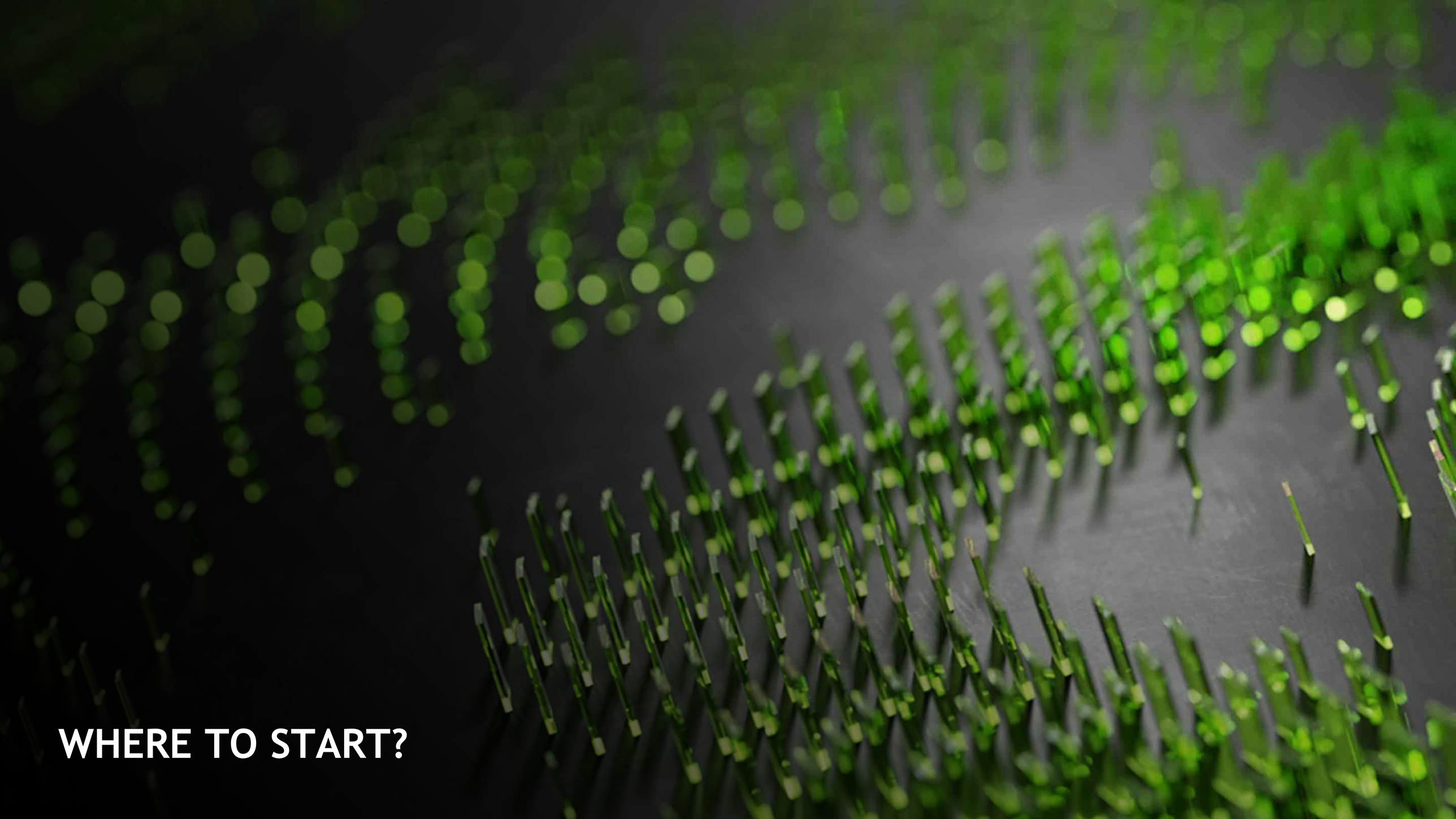
TRAIN-VALIDATION-TEST SPLIT

Transductive split:

Compute embeddings using the entire graph and split the nodes into:

- Training
- Validation
- Testing





WHERE TO START?

DGL DEEP GRAPH LIBRARY

PyTorch geometric

TensorFlow

NVIDIA DGL CONTAINERS ON NGC

NVIDIA has partnered with the DGL team. Our early release consists of two containers:

- **DGL container:** the latest versions of DGL and PyTorch, their dependencies, and the latest performance optimizations to run code with GPU-accelerated performance immediately
- **SE(3)-Transformer for DGL container:** accelerated neural network training environment based on DGL, SE(3)-Transformer, and PyTorch. Suited for recognizing 3-dimensional shapes, for segmenting LIDAR point clouds or in pharmaceutical and drug discovery research

Deep Graph Library

NVIDIA NGC

Deep Graph Library (DGL) is a framework-neutral, easy-to-use, and scalable Python library used for implementing and training Graph Neural Networks (GNN). Being framework-neutral, DGL is easily integrated into an existing PyTorch, TensorFlow, or an Apache MXNet workflow.

To enable developers to quickly take advantage of GNNs, we have partnered with the DGL team to provide a containerized solution that includes DGL, the latest PyTorch, and a set of tested dependencies. Our early release includes two containers, a ready-to-use DGL container, and an SE(3)-Transformer for DGL container, an accelerated neural network training environment based on DGL, SE(3)-Transformer, and PyTorch, suited for recognizing 3-dimensional shapes. SE(3)-Transformer for DGL container, is useful for segmenting LIDAR point clouds or in pharmaceutical and drug discovery research, for example. Apply for early access to our DGL container or the SE(3)-Transformer for DGL container.

Apply for Early Access

developer.nvidia.com/dgl-container-early-access

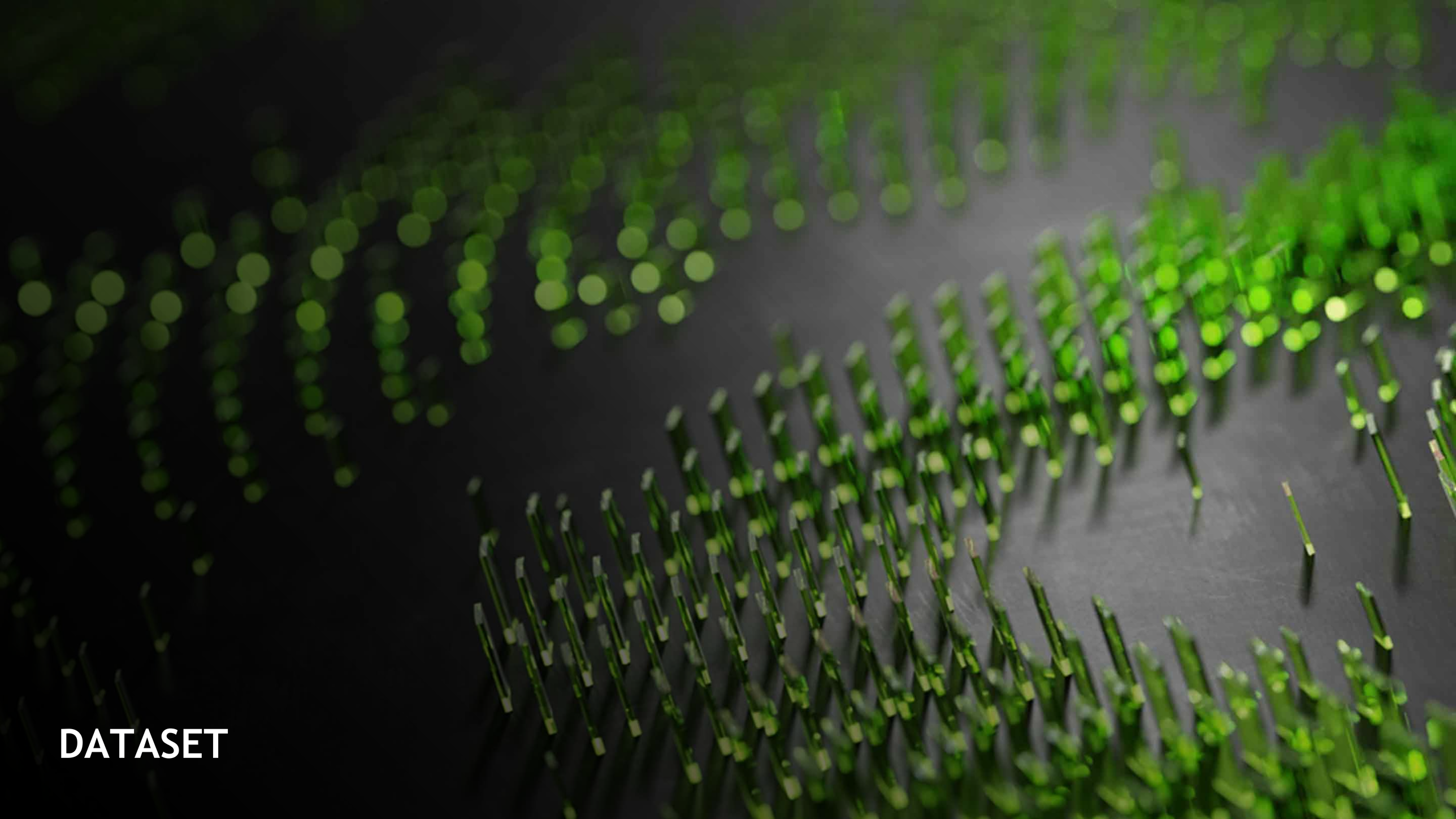
Accelerated with NVIDIA

DGL
Container
This container is built with the latest version of Deep Graph Library(DGL), PyTorch, and their dependencies.
View Labels Pull Tag

Accelerated with NVIDIA

SE(3)-Transformer for DGL
Container
This container is built with the latest version of Deep Graph Library(DGL) PyTorch and their dependencies. This container is built on top of th...
View Labels Pull Tag

developer.nvidia.com/blog/accelerating-se3-transformers-training-using-an-nvidia-open-source-model-implementation/



DATASET

CORA DATASET

2708 scientific publications classified into the following classes:

- Case_Based
- Genetic_Algorithms
- Neural_Networks
- Probabilistic_Methods
- Reinforcement_Learning
- Rule_Learning
- Theory

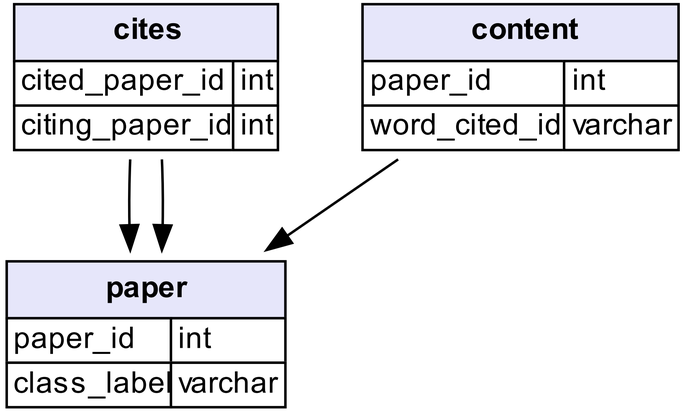
An ***edge*** captures a citation between a pair of papers (undirected graph)

Node features: Each publication in the dataset is represented by a 0/1-valued word vector that maps the presence of the corresponding word in a dictionary of 1433 unique words.

```python
import dgl.data

dataset = dgl.data.CoraGraphDataset()
print('Number of categories:', dataset.num_classes)
```

```
  NumNodes: 2708
  NumEdges: 10556
  NumFeats: 1433
  NumClasses: 7
  NumTrainingSamples: 140
  NumValidationSamples: 500
  NumTestSamples: 1000
Done loading data from cached files.
Number of categories: 7
```



Prithviraj Sen, et al. "Collective classification in network data." AI Magazine, 2008.

KEY TAKEAWAYS

Model anything as a graph

Kinds and properties of graphs

Graph representations: adjacency matrix vs adjacency lists

Flavours of learning: node-level, graph-level, edge-level

Embeddings

Message passing: aggregate, transform, update

Frameworks: DGL, PyTorch Geometric, TF-GNN

FURTHER LITERATURE

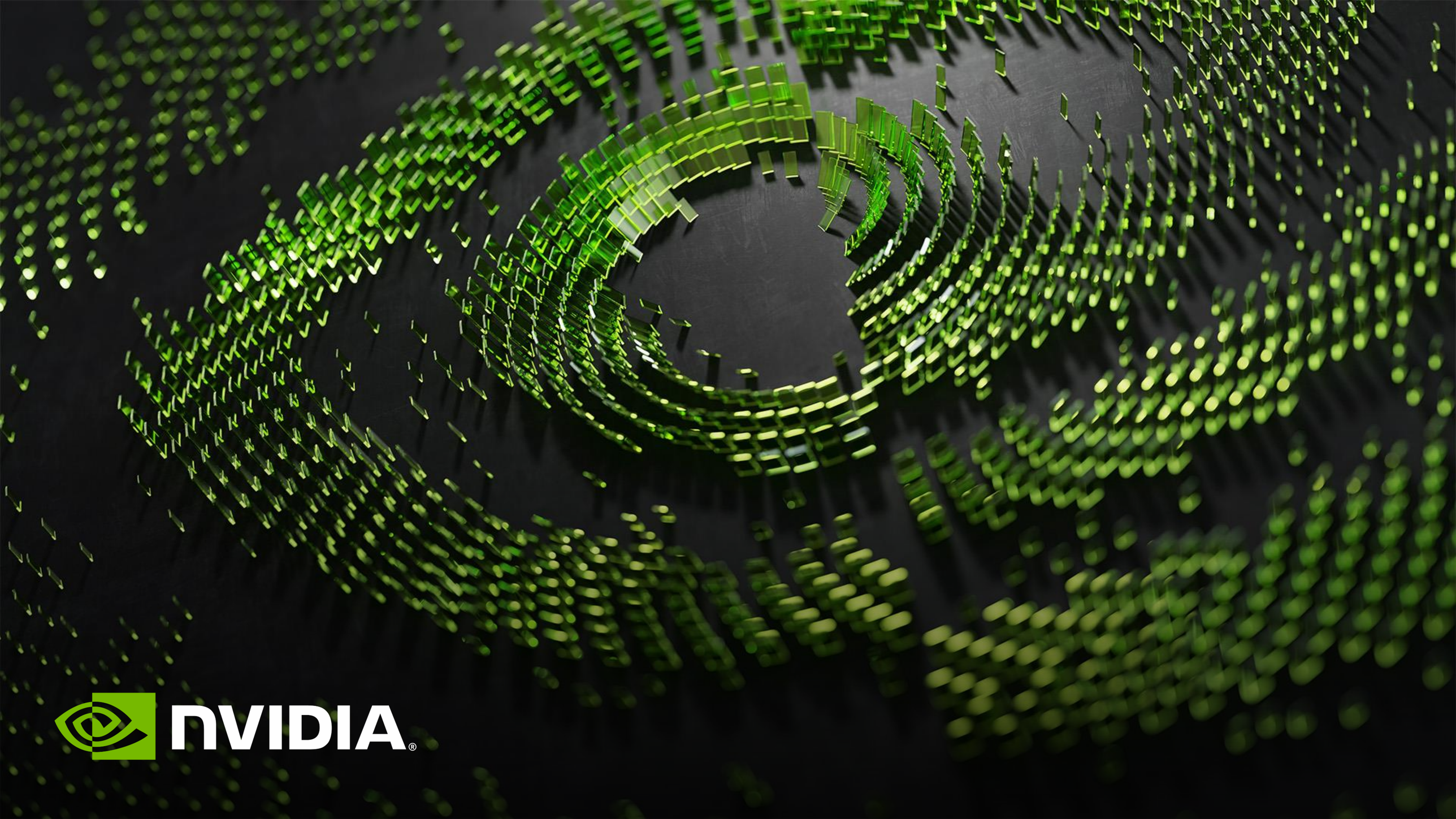
[INTRO] “A Gentle Introduction to Graph Neural Networks” *Sanchez-Lengeling et al.*, Distill, 2021, distill.pub/2021/gnn-intro/

[INTRO] “Understanding Convolutions on Graphs” *Daigavane et al.*, Distill, 2021, distill.pub/2021/understanding-gnns/

[REVIEW] “Graph-Based Deep Learning for Medical Diagnosis and Analysis: Past, Present and Future” *Aristizabal et al.*, arXiv, 2021, arxiv.org/abs/2105.13137

[REVIEW] **“**Graph neural networks: A review of methods and applications” *Zhou et al.*, AI Open, 2021, doi.org/10.1016/j.aiopen.2021.01.001

[REVIEW] “[...] Dynamic Graph Neural Networks: A Survey” *Skarding et al.*, IEEE/arXiv, 2021, arxiv.org/abs/2005.07496


NVIDIA