MACIEJ BESTA, PATRICK IFF, FLORIAN SCHEIDL, KAZUKI OSAWA, NIKOLI DRYDEN, MICHAL PODSTAWSKI, TIANCHENG CHEN, TORSTEN HOEFLER

Neural Graph Databases

MACIEJ BESTA, PATRICK IFF, FLORIAN SCHEIDL, KAZUKI OSAWA, NIKOLI DRYDEN, MICHAL PODSTAWSKI, TIANCHENG CHEN, TORSTEN HOEFLER

Neural Graph Databases

https://arxiv.org/abs/2209.09732

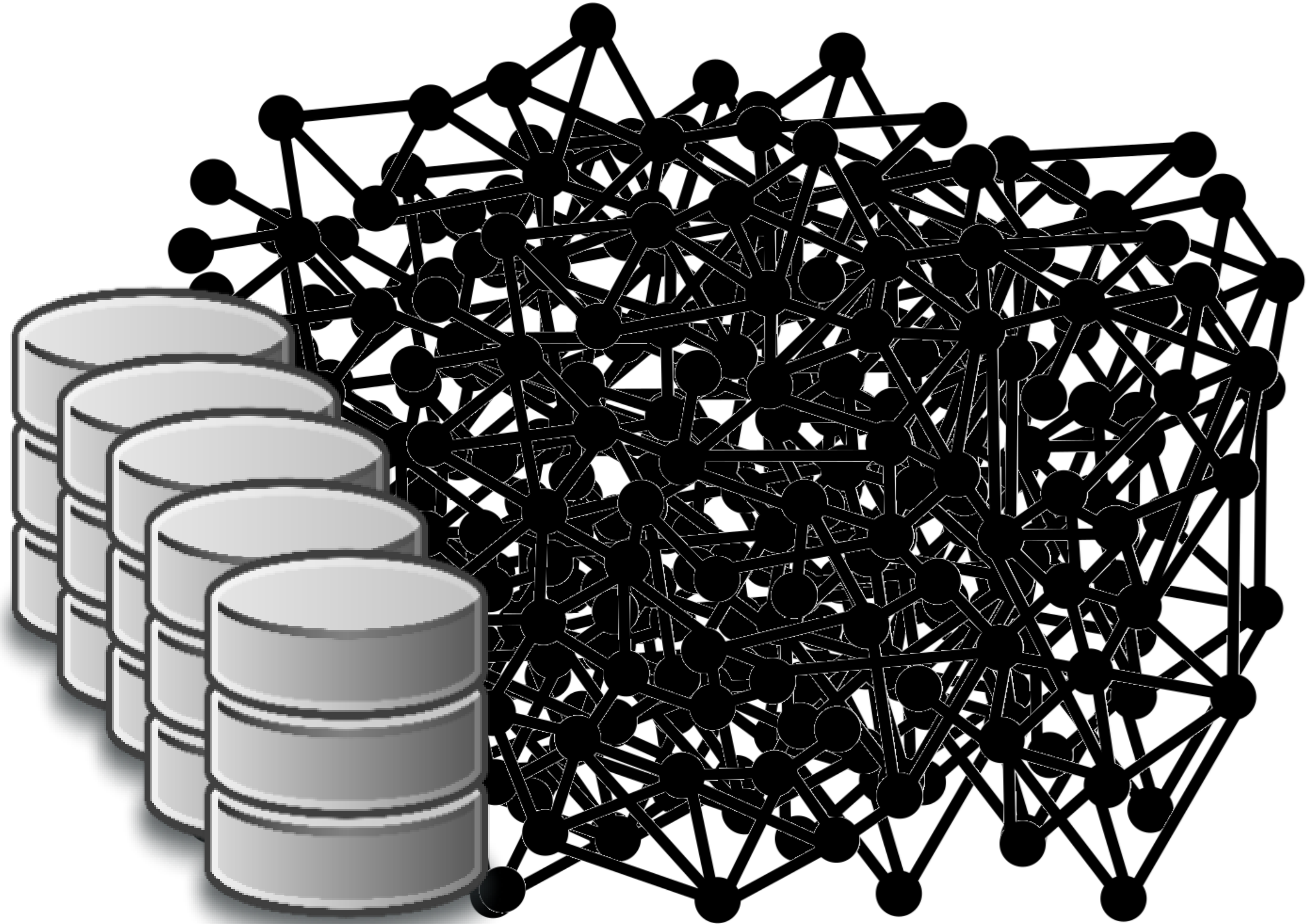
@ LoG'22 (Learning on Graphs'22)

Neural Graph Databases

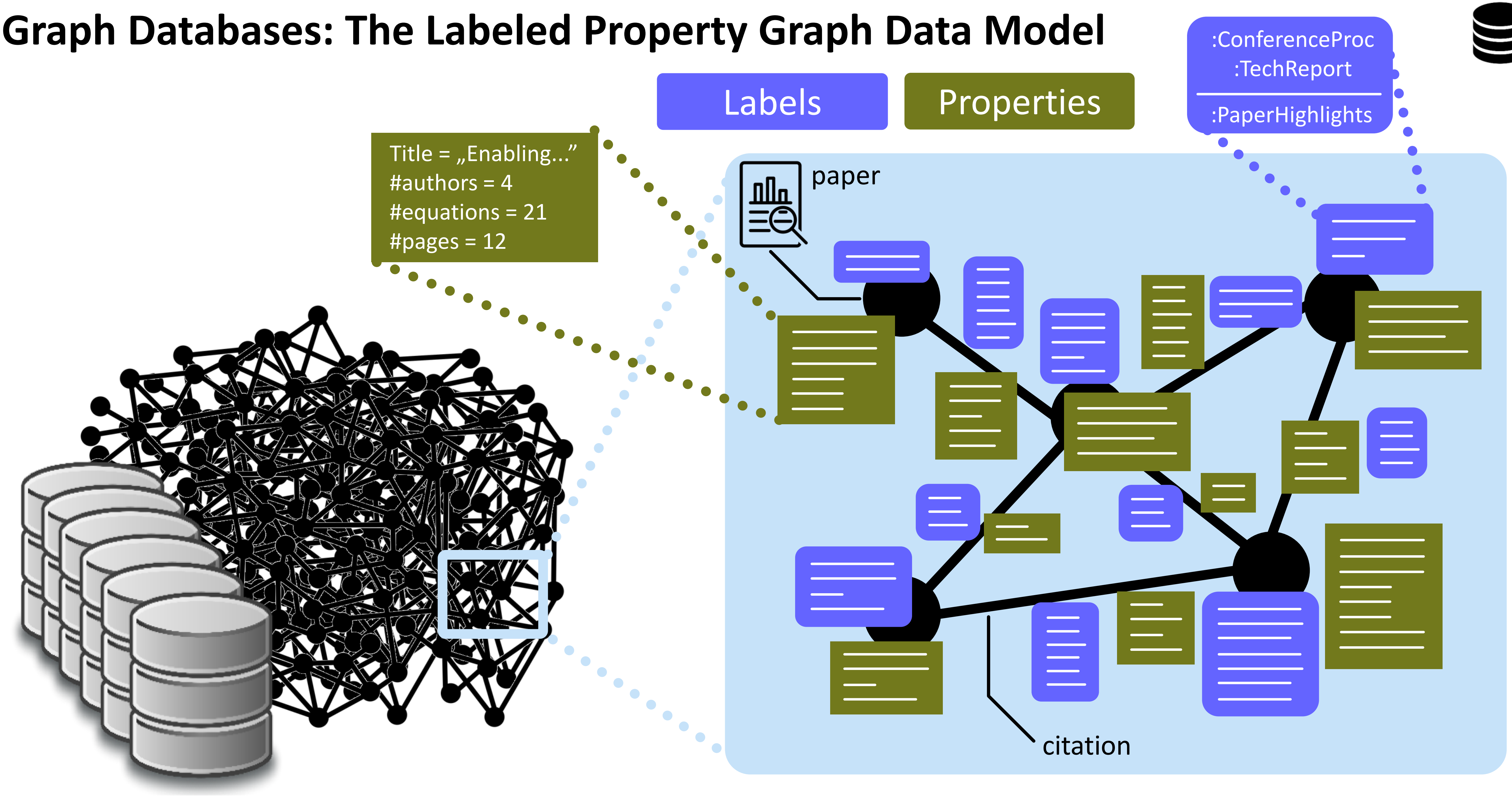
Maciej Besta[1,†] Patrick Iff[1] Florian Scheidl[1] Kazuki Osawa[1] Nikoli Dryden[1]
Michal Podstawski[2,3] Tiancheng Chen[1] Torsten Hoefler[1,†]

[1]Department of Computer Science, ETH Zurich
[2]Warsaw University of Technology, Warsaw, Poland

Graph Databases (GDBs): A Very Brief Introduction

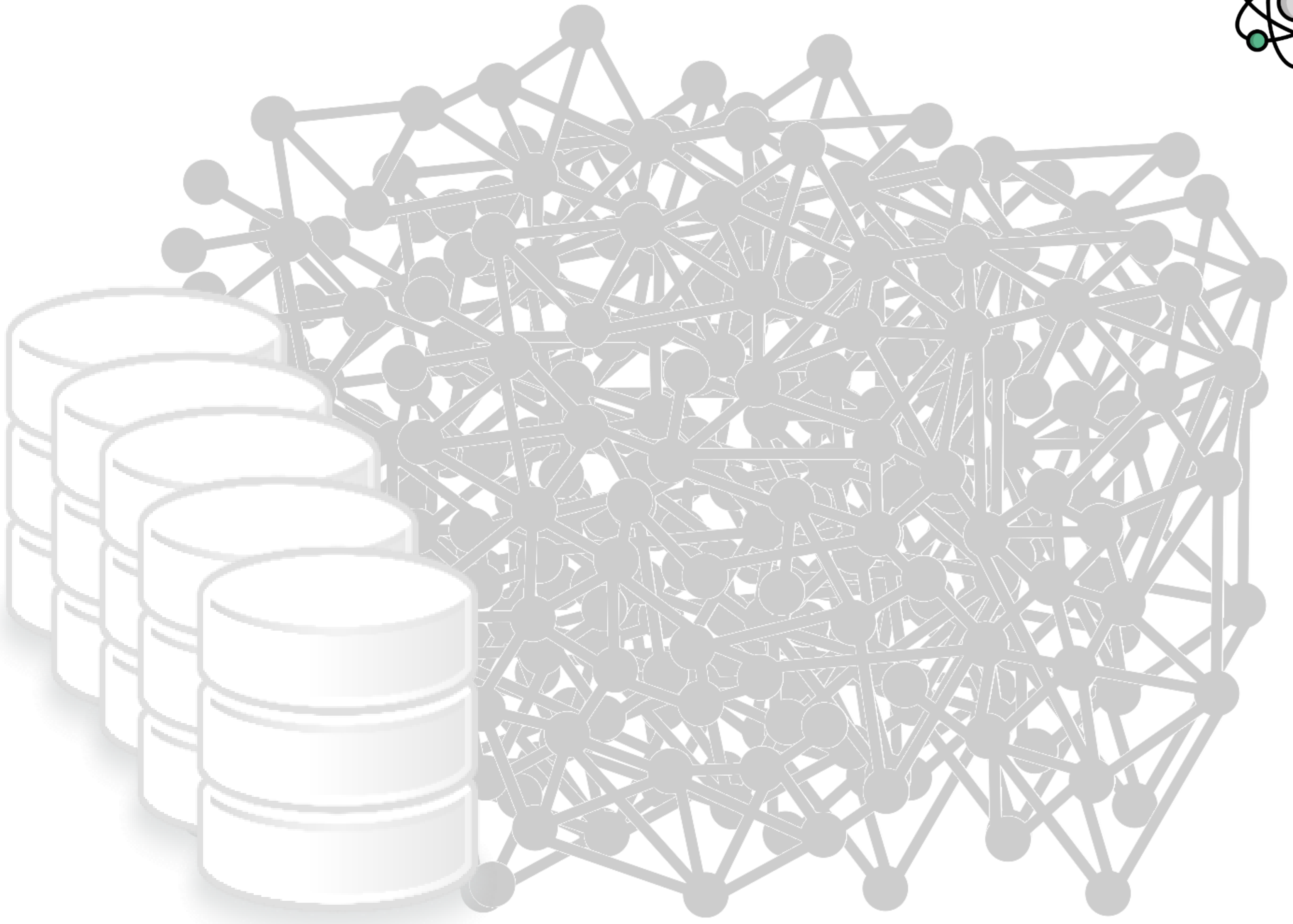


Graph Databases: The Labeled Property Graph Data Model



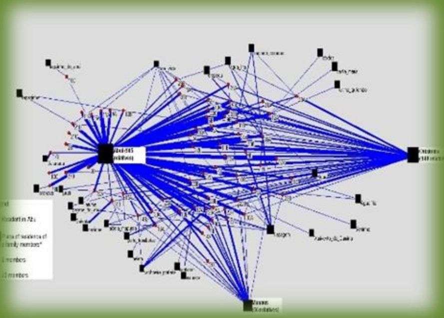
Graph Databases: Where Do We Use Them?

Graph Databases: Where Do We Use Them?

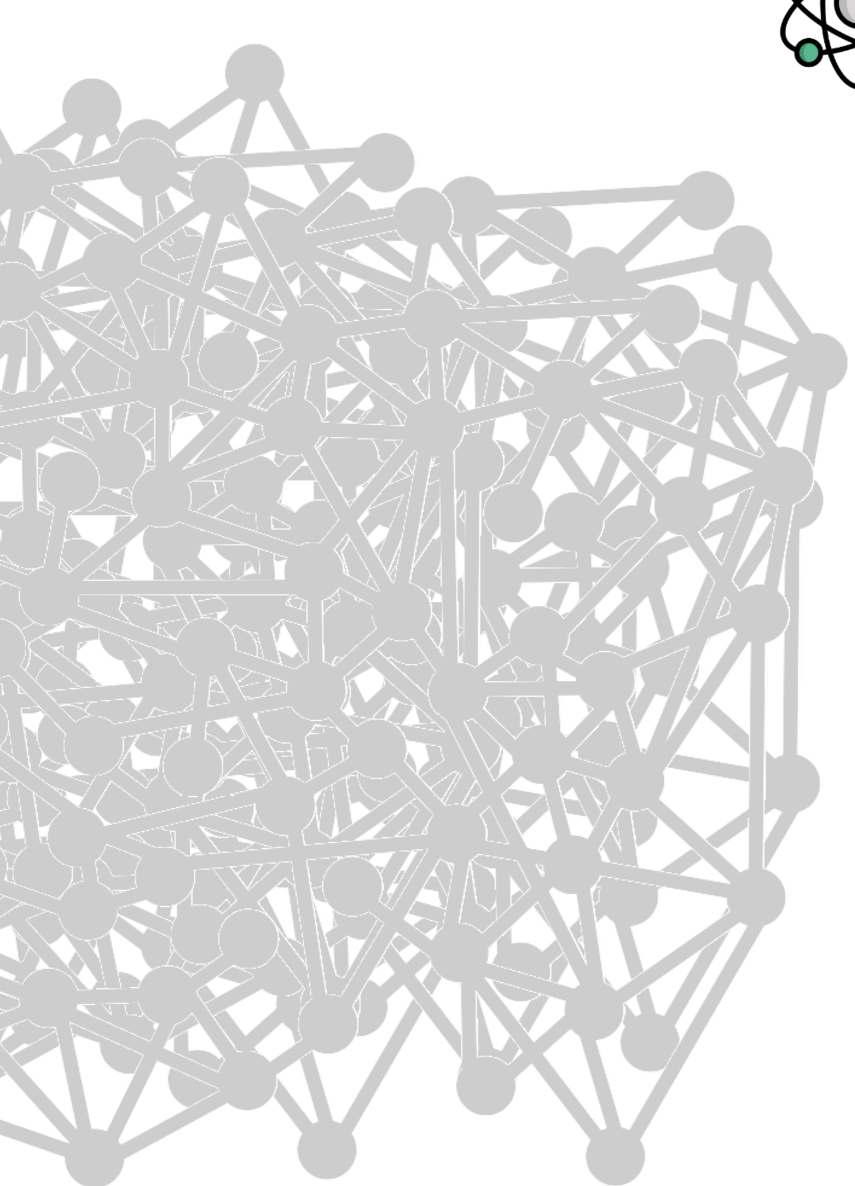


Graph Databases: Where Do We Use Them?

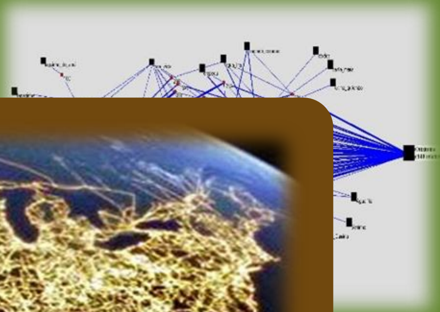
Social sciences



Graph Databases: Where Do We Use Them?

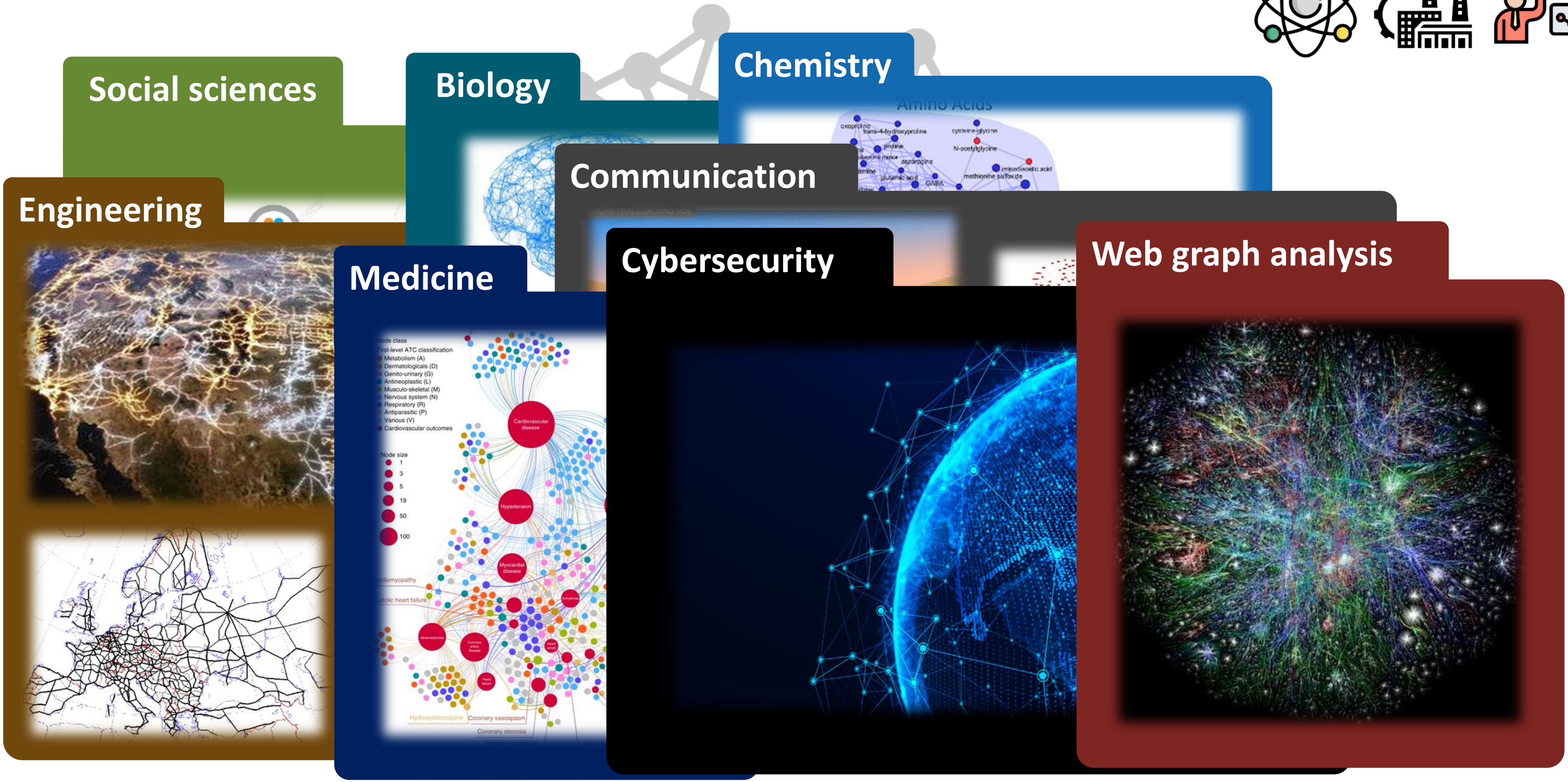


Social sciences

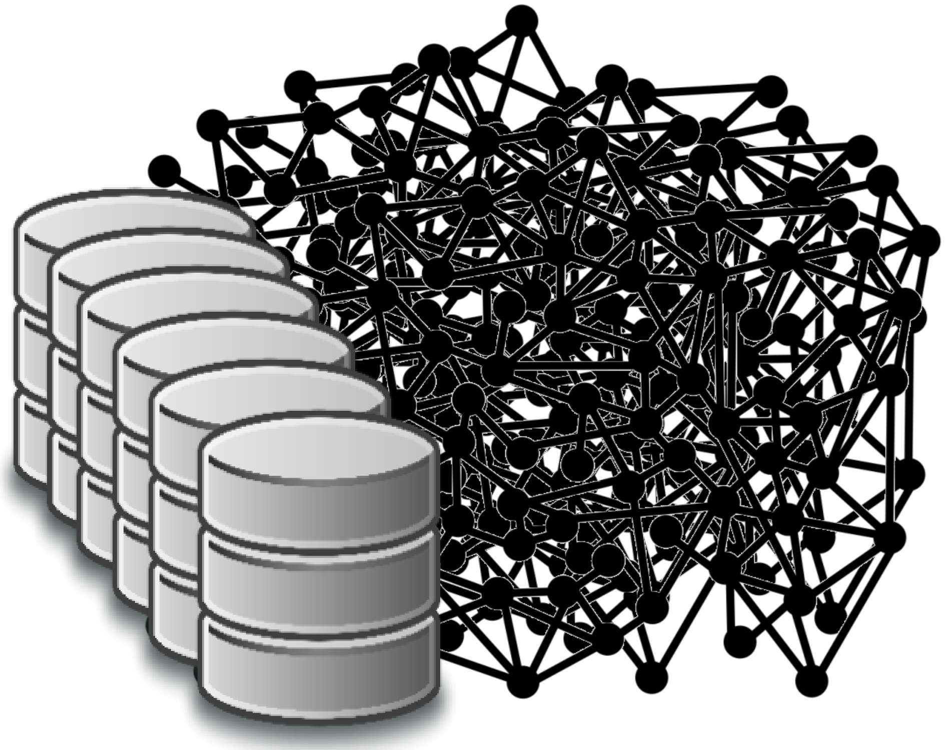


Engineering

Graph Databases: Where Do We Use Them?



That's why there are lots of them!



That's why there are lots of them!

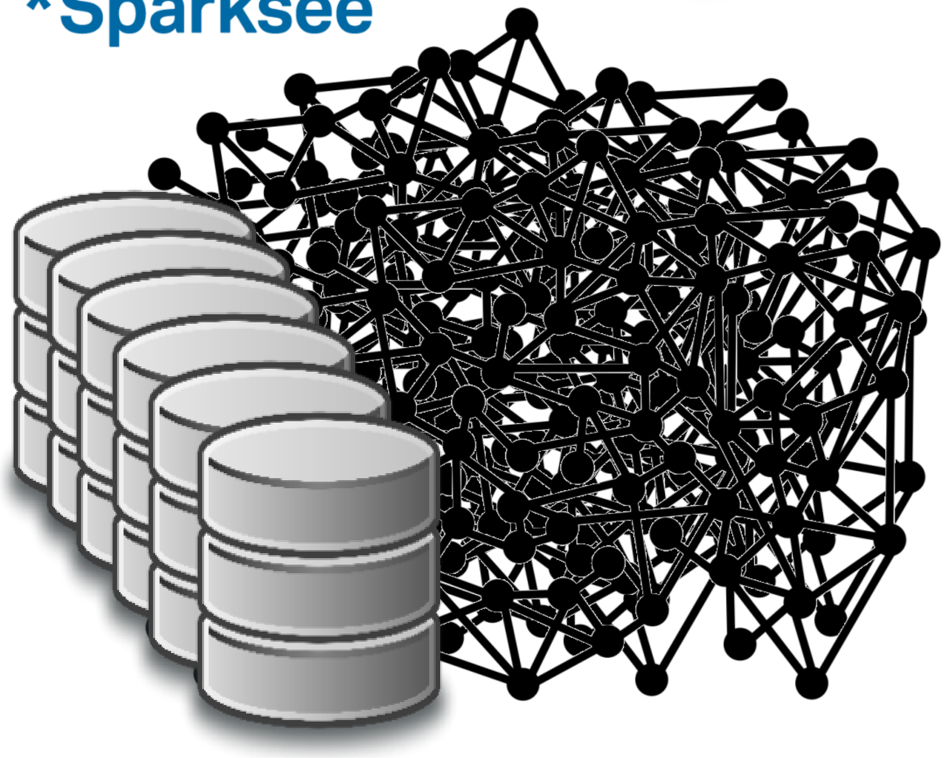
Dgraph STARDOG NebulaGraph ArangoDB

VelocityDB InfiniteGraph

GALAXYBASE AGENS Graph Database HugeGraph ANZO Amazon Neptune fauna

graphbase.ai BangDB neo4j ULTIPA Vaticle TypeDB blazegraph™ GraphDB

*Sparksee ArcadeDB



OrientDB® TITAN

KATANA GRAPH

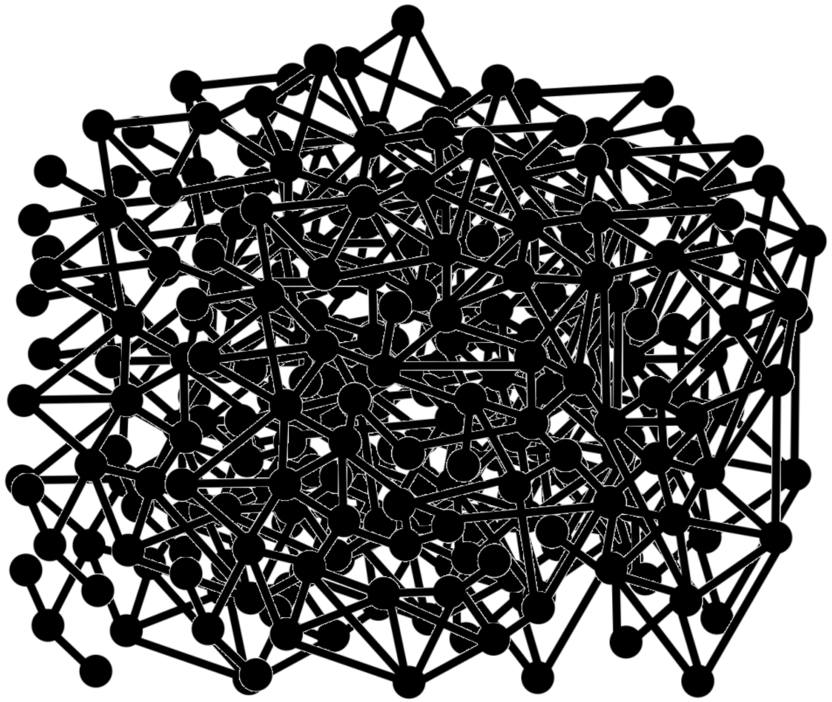
MEM GRAPH

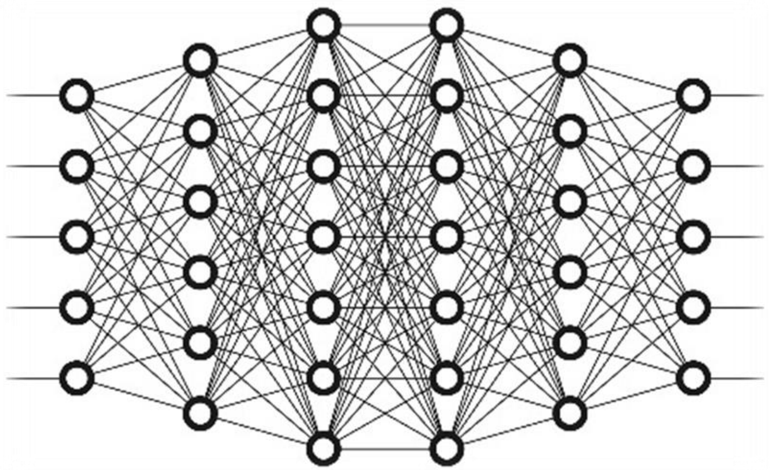
FlockDB Franz Inc. AllegroGraph TigerGraph

JanusGraph

TerminusDB KÙZU

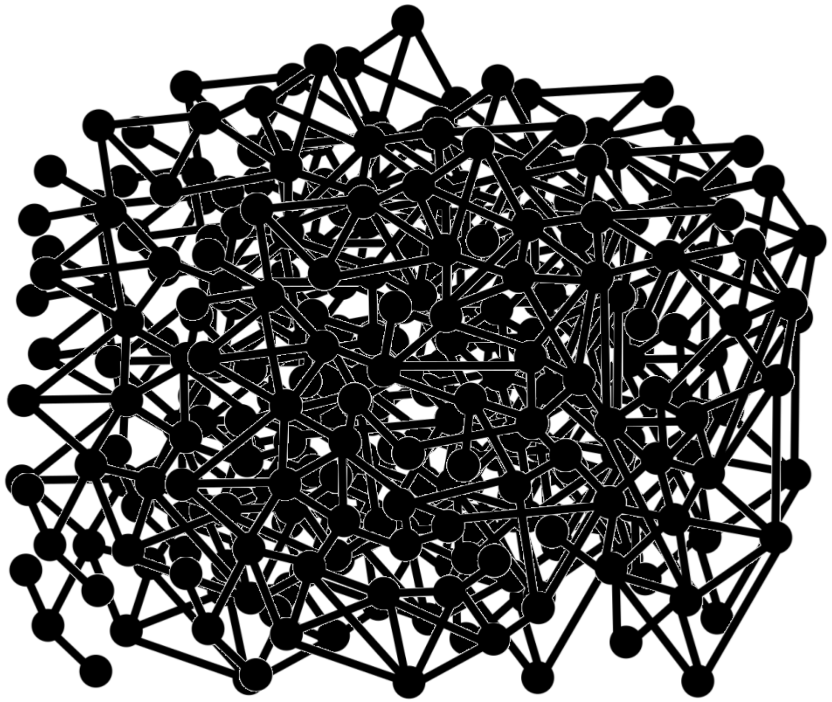
In parallel to the development of graph databases, there has been an ongoing revolution in graph machine learning...

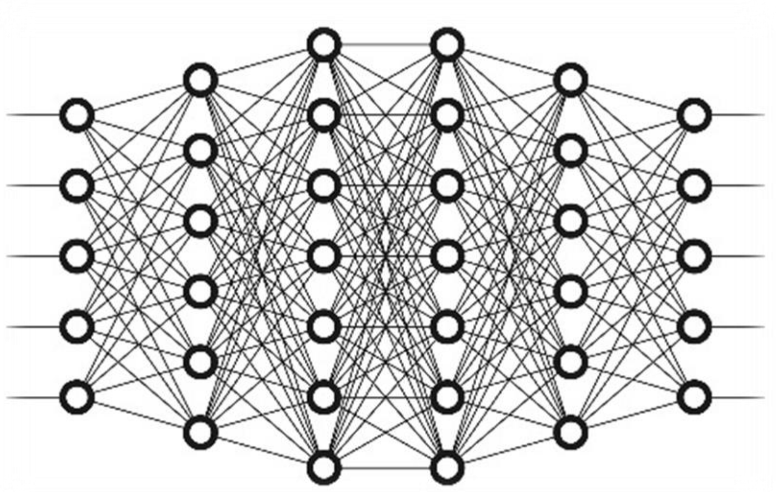


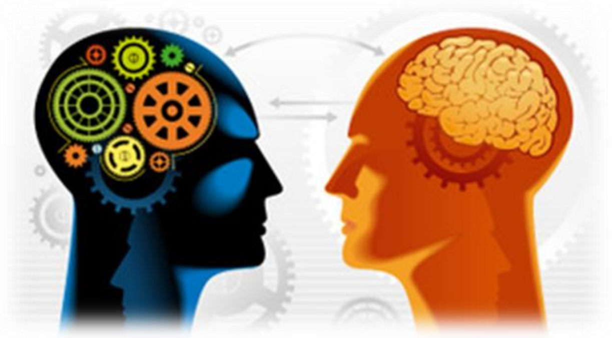


In parallel to the development of graph databases, there has been an ongoing revolution in graph machine learning...

In the last 5 years learning on graphs exploded







Graphs + Deep Learning = Graphs Neural Networks (GNNs)

Graphs + Deep Learning = Graphs Neural Networks (GNNs)

Article

A graph placement methodology for fast chip design

https://doi.org/10.1038/s41586-021-03544-w

Received: 3 November 2020

Accepted: 13 April 2021

Published online: 9 June 2021

Check for updates

Azalia Mirhoseini[1,4]✉, Anna Goldie[1,3,4]✉, Mustafa Yazgan[2], Joe Wenjie Jiang[1], Ebrahim Songhori[1], Shen Wang[1], Young-Joon Lee[2], Eric Johnson[1], Omkar Pathak[2], Azade Nazi[1], Jiwoo Pak[2], Andy Tong[2], Kavya Srinivasa[2], William Hang[3], Emre Tuncer[2], Quoc V. Le[1], James Laudon[1], Richard Ho[2], Roger Carpenter[2] & Jeff Dean[1]

Chip floorplanning is the engineering task of designing the physical layout of a computer chip. Despite five decades of research[1], chip floorplanning has defied

Graphs + Deep Learning = Graphs Neural Networks (GNNs)

Article

A graph placement methodology for fast chip design

https://doi.org/10.1038/s41586-021-03
Received: 3 November 2020
Accepted: 13 April 2021
Published online: 9 June 2021
Check for updates

Article

Advancing mathematics by guiding human intuition with AI

https://doi.org/10.1038/s41586-021-04086-x
Received: 10 July 2021
Accepted: 30 September 2021
Published online: 1 December 2021
Open access

Alex Davies[1✉], Petar Veličković[1], Lars Buesing[1], Sam Blackwell[1], Daniel Zheng[1], Nenad Tomašev[1], Richard Tanburn[1], Peter Battaglia[1], Charles Blundell[1], András Juhász[2], Marc Lackenby[2], Geordie Williamson[3], Demis Hassabis[1] & Pushmeet Kohli[1✉]

The practice of mathematics involves discovering patterns and using these to formulate and prove conjectures, resulting in theorems. Since the 1960s,

Graphs + Deep Learning = Graphs Neural Networks (GNNs)

Article

A graph placement methodology for fast chip design

https://doi.org/10.1038/s41586-021-03
Received: 3 November 2020
Accepted: 13 April 2021
Published online: 9 June 2021
Check for updates

Article

Advancing mathematics by guiding human intuition with AI

https://doi.org/10.1038/s41586-021-04086-x
Received: 10 July 2021
Accepted: 30 September 2021
Published online: 1 December 2021
Open access

Alex Da
Nenad
Marc La

The pra
formul

Article

Highly accurate protein structure prediction with AlphaFold

https://doi.org/10.1038/s41586-021-03819-2
Received: 11 May 2021
Accepted: 12 July 2021
Published online: 15 July 2021
Open access
Check for updates

John Jumper[1,4✉], Richard Evans[1,4], Alexander Pritzel[1,4], Tim Green[1,4], Michael Figurnov[1,4], Olaf Ronneberger[1,4], Kathryn Tunyasuvunakool[1,4], Russ Bates[1,4], Augustin Žídek[1,4], Anna Potapenko[1,4], Alex Bridgland[1,4], Clemens Meyer[1,4], Simon A. A. Kohl[1,4], Andrew J. Ballard[1,4], Andrew Cowie[1,4], Bernardino Romera-Paredes[1,4], Stanislav Nikolov[1,4], Rishub Jain[1,4], Jonas Adler[1], Trevor Back[1], Stig Petersen[1], David Reiman[1], Ellen Clancy[1], Michal Zielinski[1], Martin Steinegger[2,3], Michalina Pacholska[1], Tamas Berghammer[1], Sebastian Bodenstein[1], David Silver[1], Oriol Vinyals[1], Andrew W. Senior[1], Koray Kavukcuoglu[1], Pushmeet Kohli[1] & Demis Hassabis[1,4✉]

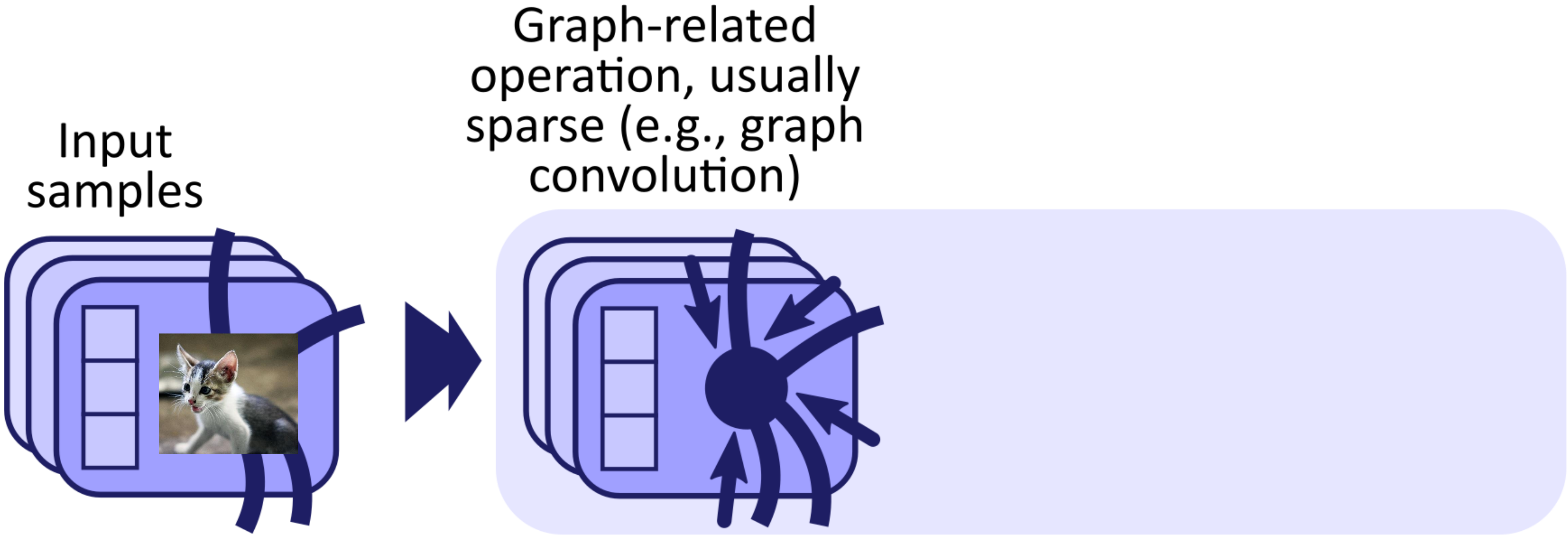
Proteins are essential to life, and understanding their structure can facilitate a

A Single GNN Layer

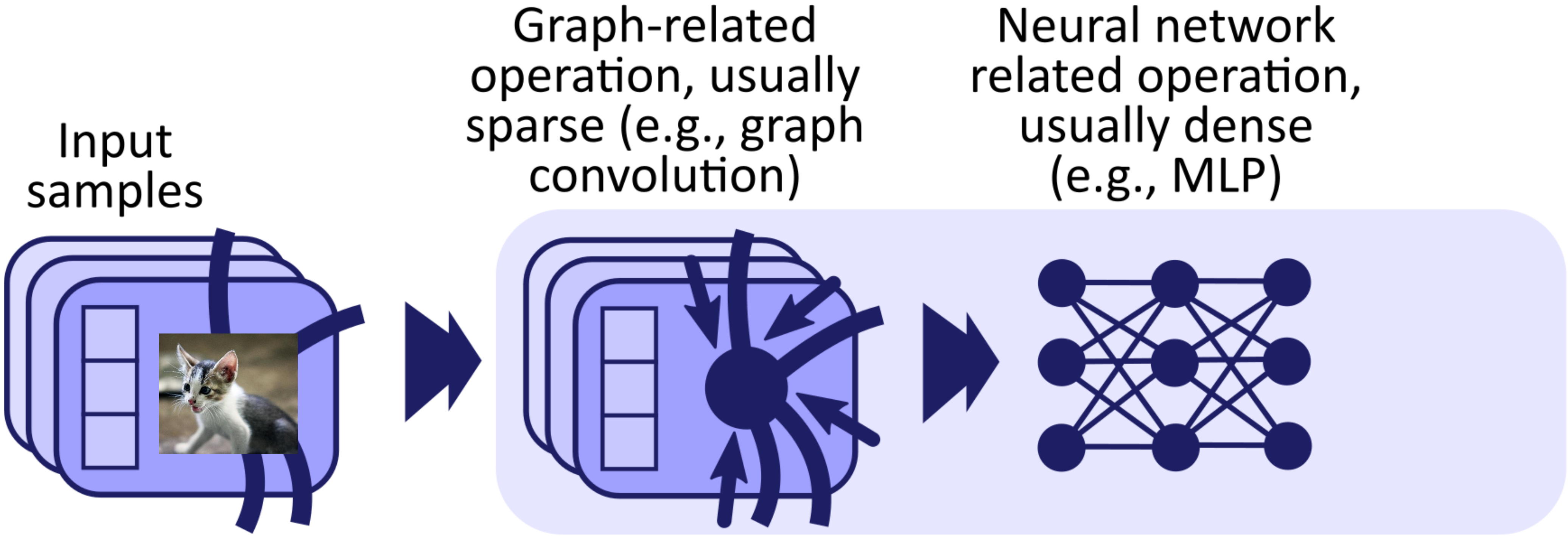
Input
samples

A Single GNN Layer

A Single GNN Layer



A Single GNN Layer



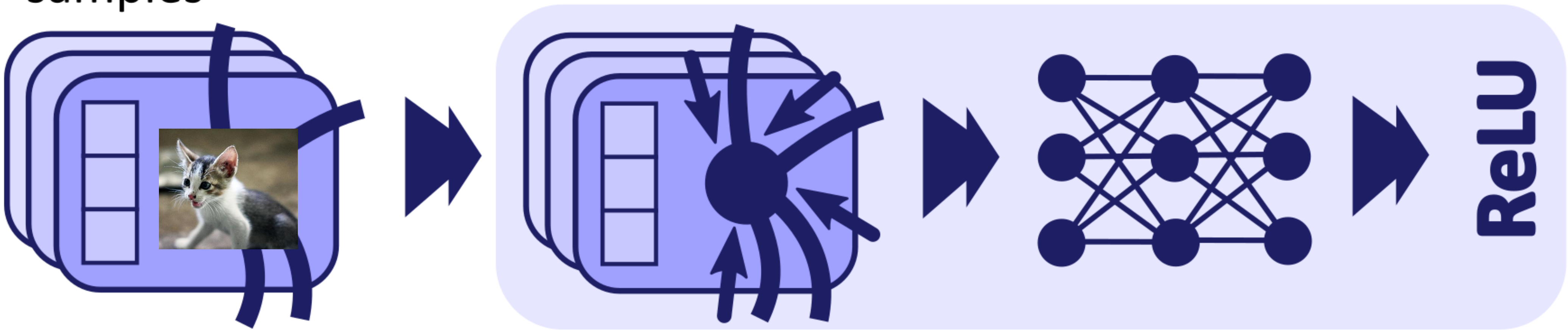
A Single GNN Layer

Input samples

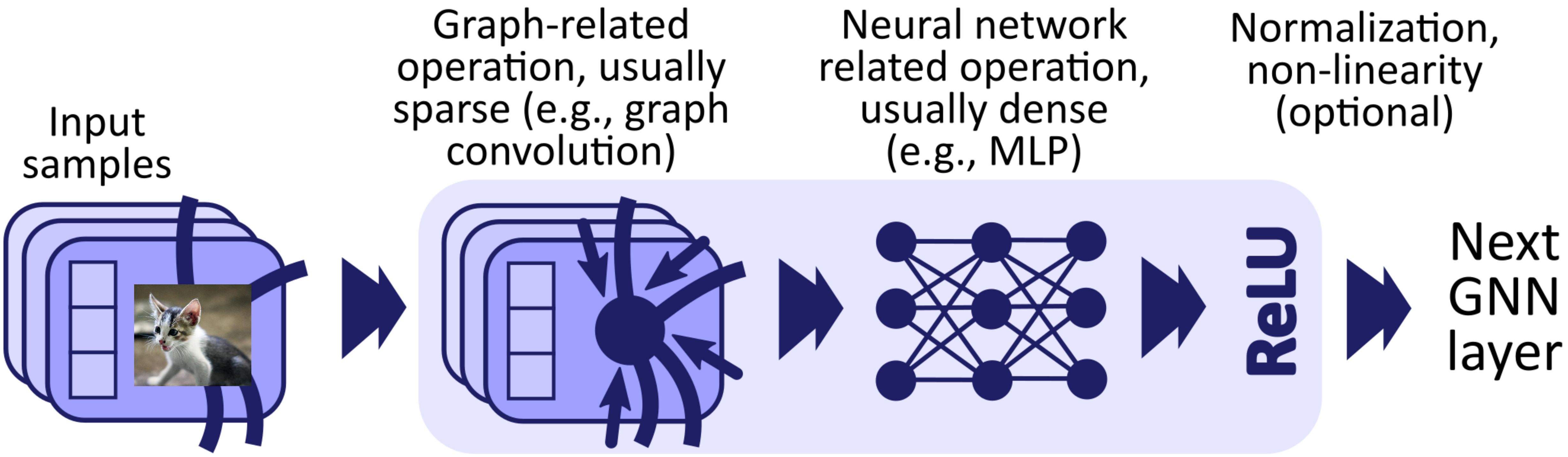
Graph-related operation, usually sparse (e.g., graph convolution)

Neural network related operation, usually dense (e.g., MLP)

Normalization, non-linearity (optional)

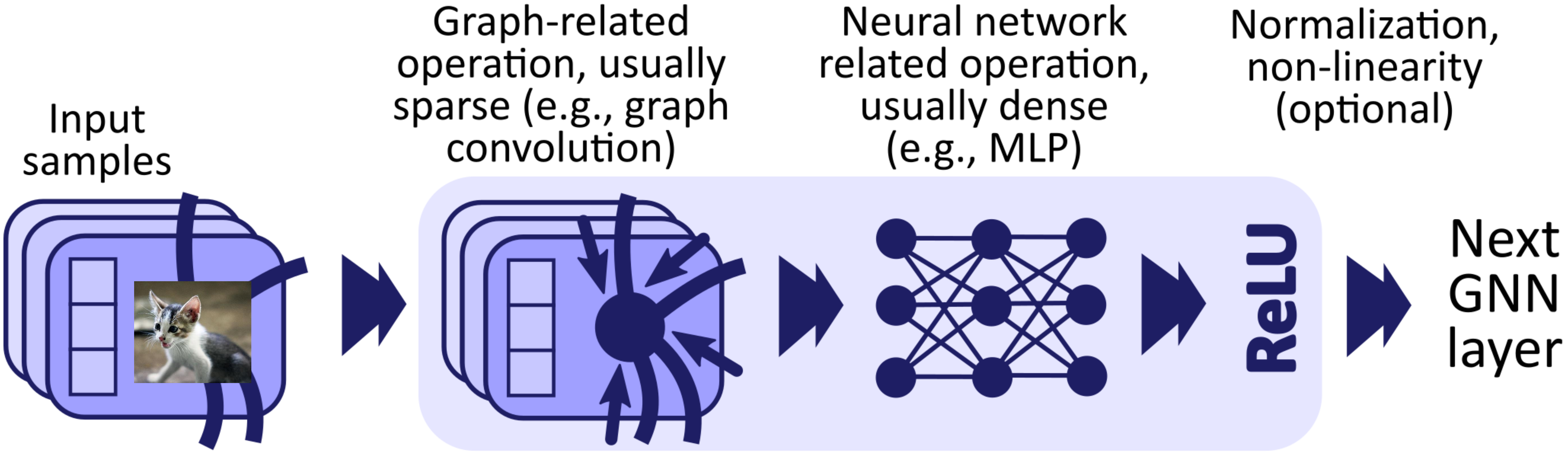


A Single GNN Layer



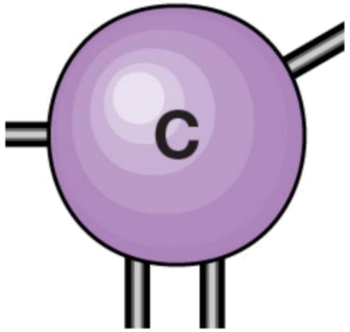
A Single GNN Layer

Classification or regression of nodes



A Single GNN Layer

Classification or regression of nodes

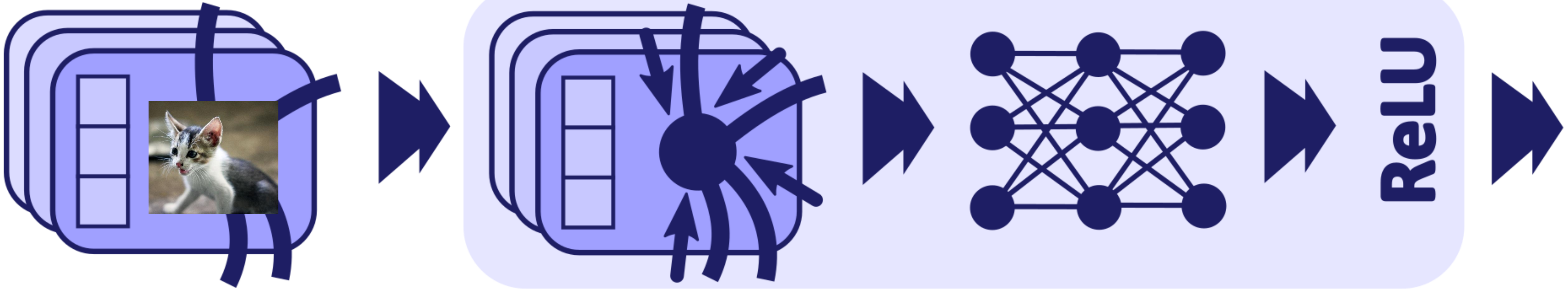


Input samples

Graph-related operation, usually sparse (e.g., graph convolution)

Neural network related operation, usually dense (e.g., MLP)

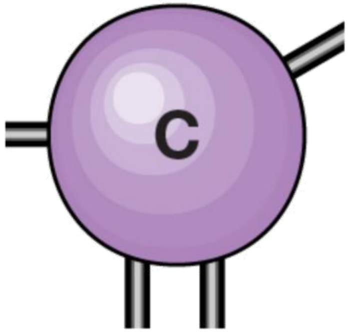
Normalization, non-linearity (optional)



Next GNN layer

A Single GNN Layer

Classification or regression of nodes



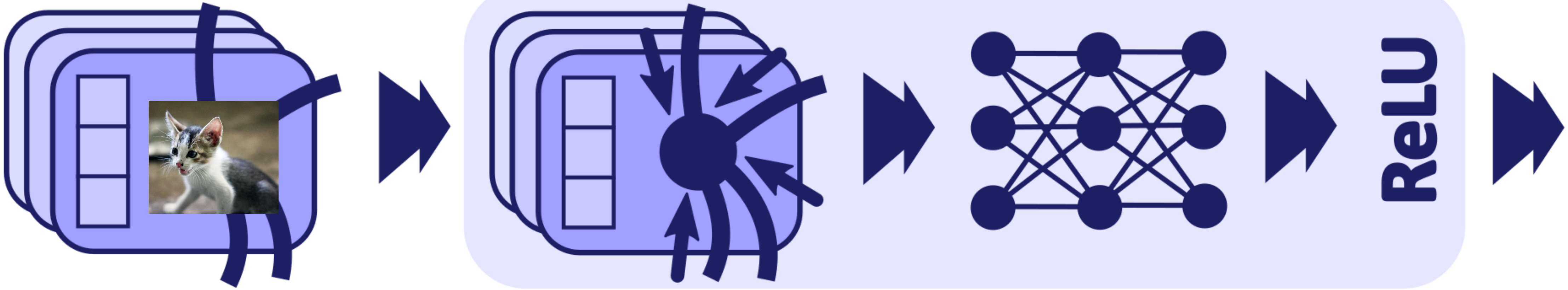
Classification or regression of edges

Input samples

Graph-related operation, usually sparse (e.g., graph convolution)

Neural network related operation, usually dense (e.g., MLP)

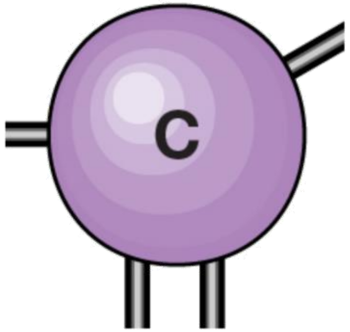
Normalization, non-linearity (optional)



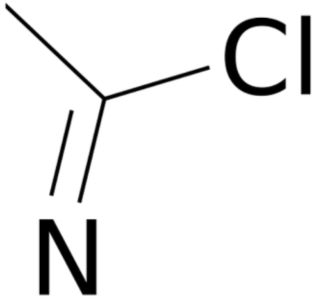
Next GNN layer

A Single GNN Layer

Classification or regression of nodes



Classification or regression of edges

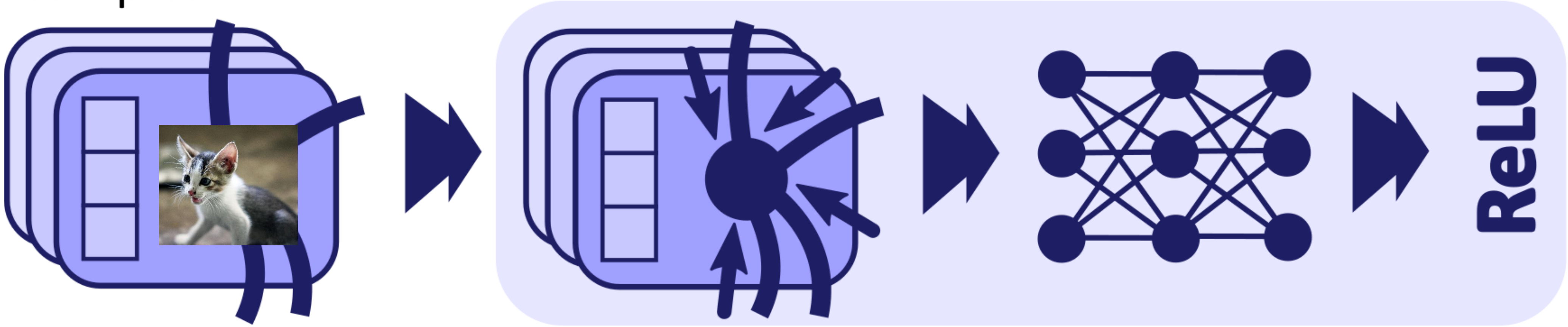


Input samples

Graph-related operation, usually sparse (e.g., graph convolution)

Neural network related operation, usually dense (e.g., MLP)

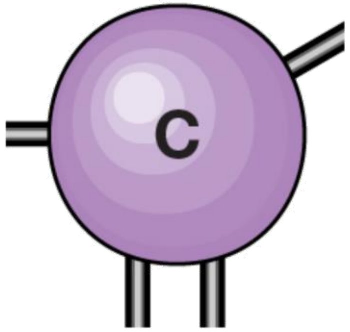
Normalization, non-linearity (optional)



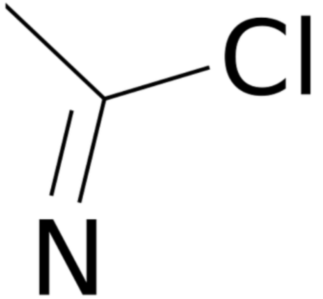
Next GNN layer

A Single GNN Layer

Classification or regression of nodes



Classification or regression of edges



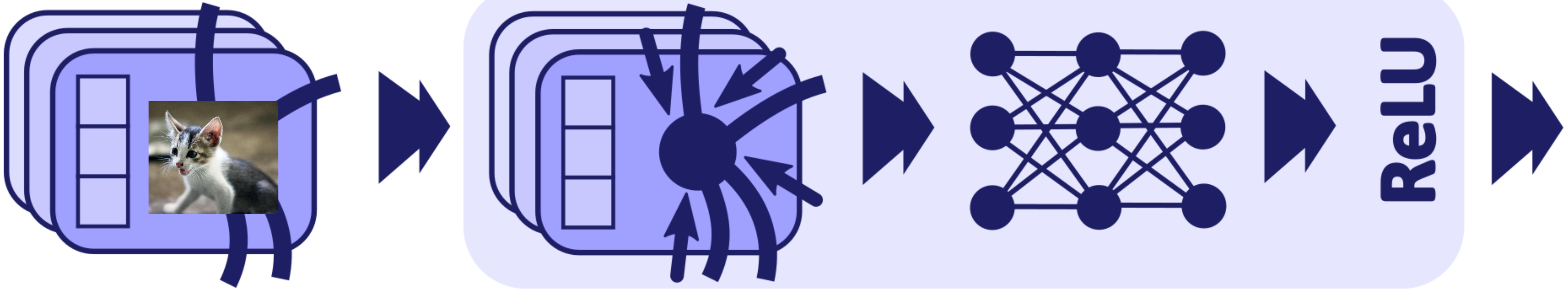
Classification or regression of graphs

Input samples

Graph-related operation, usually sparse (e.g., graph convolution)

Neural network related operation, usually dense (e.g., MLP)

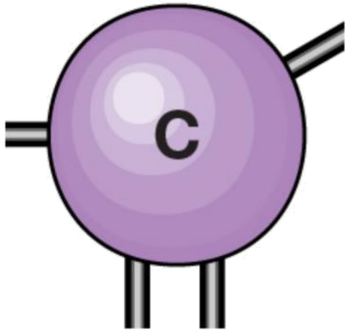
Normalization, non-linearity (optional)



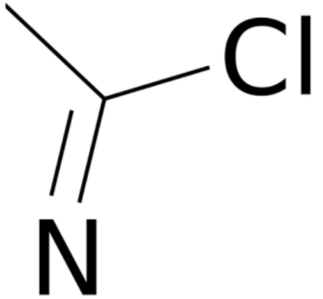
Next GNN layer

A Single GNN Layer

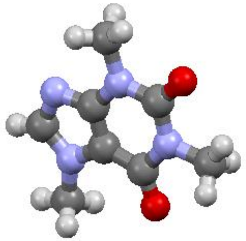
Classification or regression of nodes



Classification or regression of edges



Classification or regression of graphs

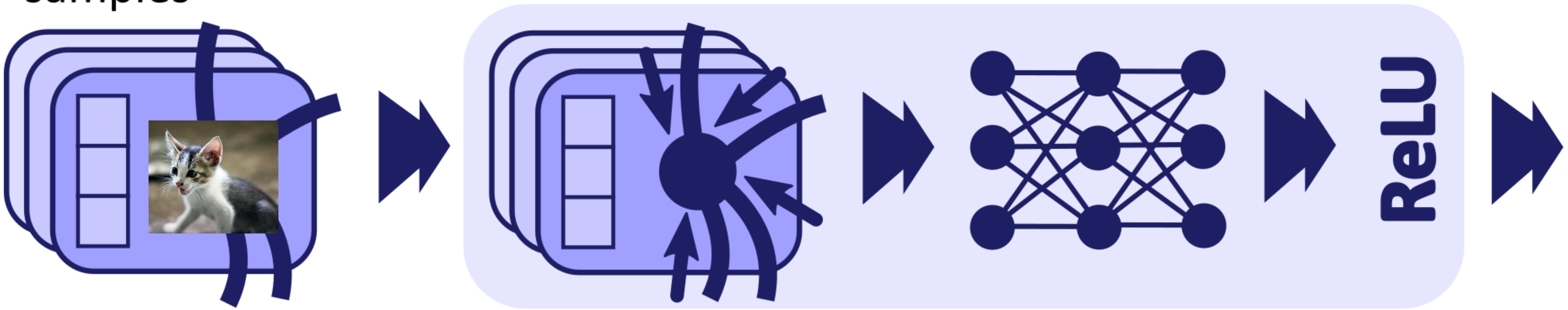


Input samples

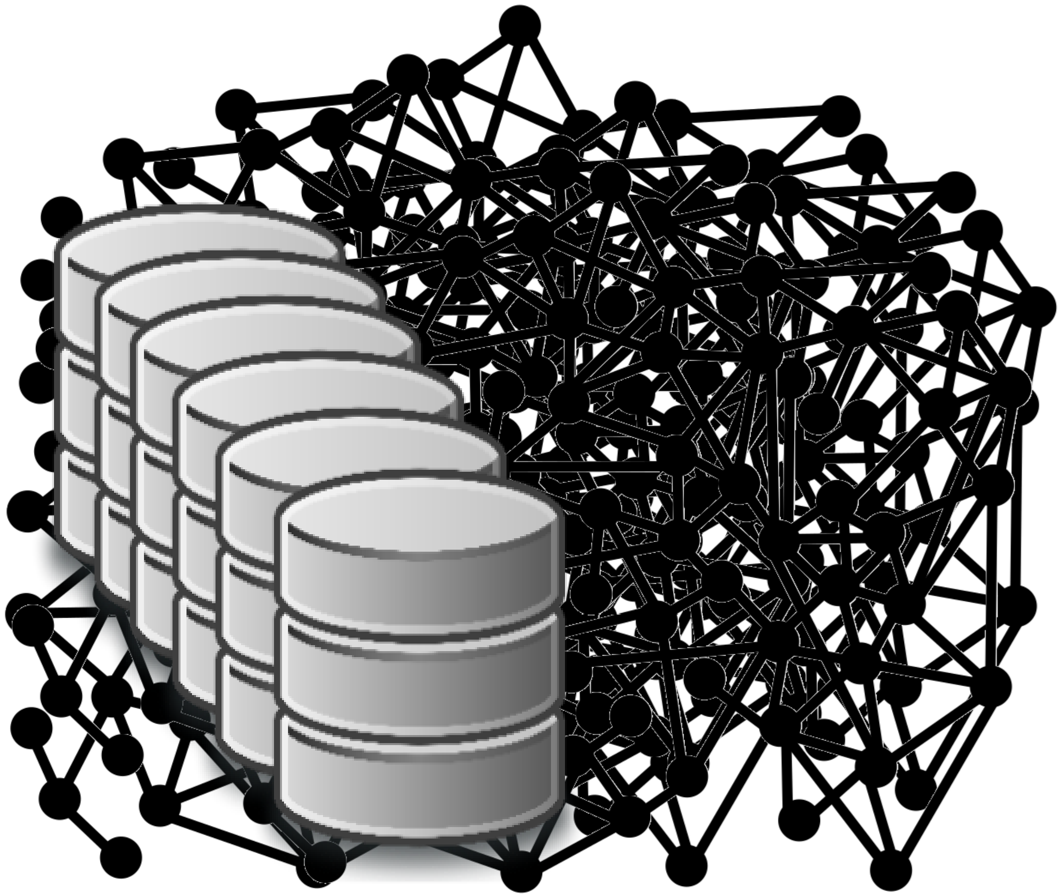
Graph-related operation, usually sparse (e.g., graph convolution)

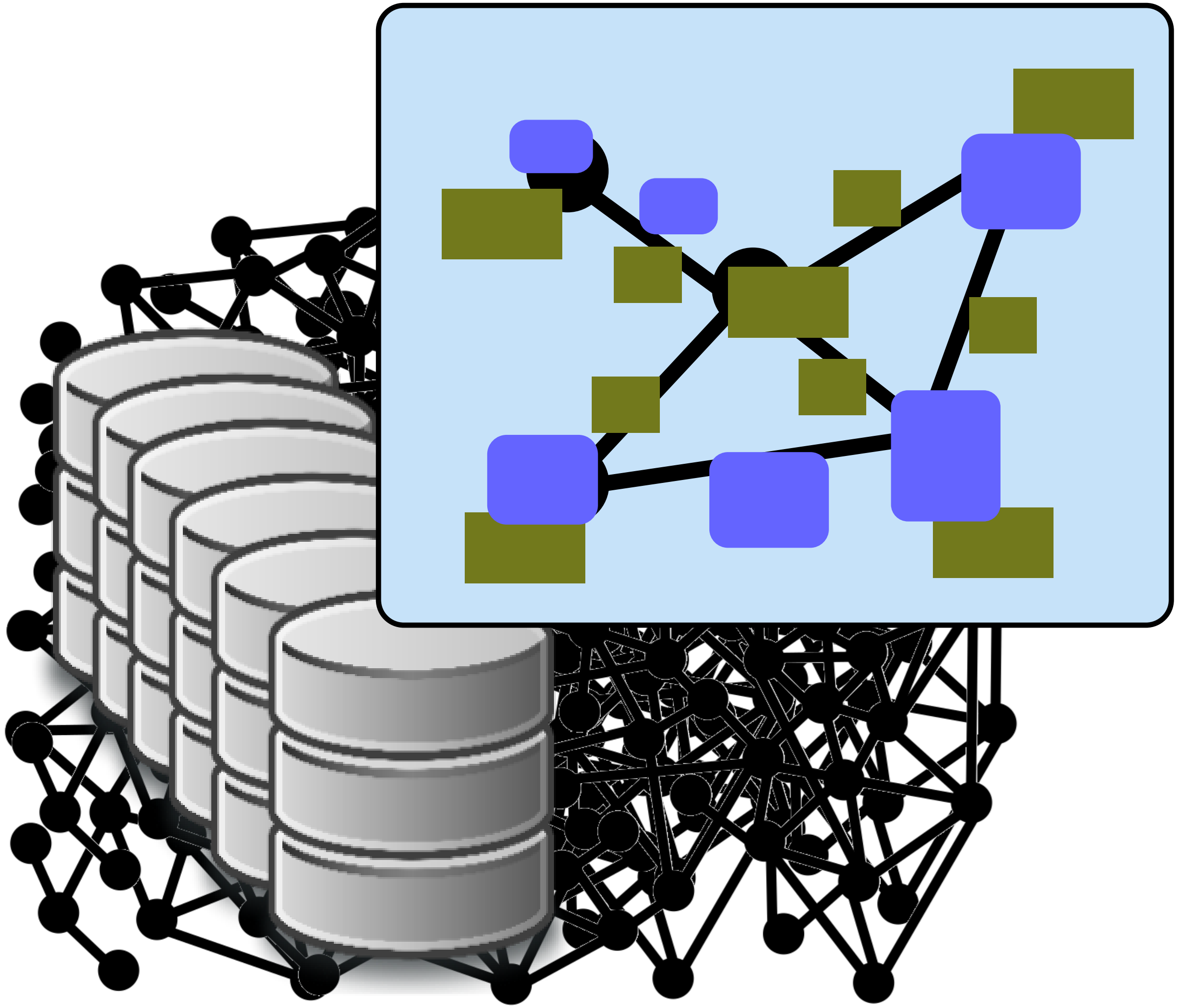
Neural network related operation, usually dense (e.g., MLP)

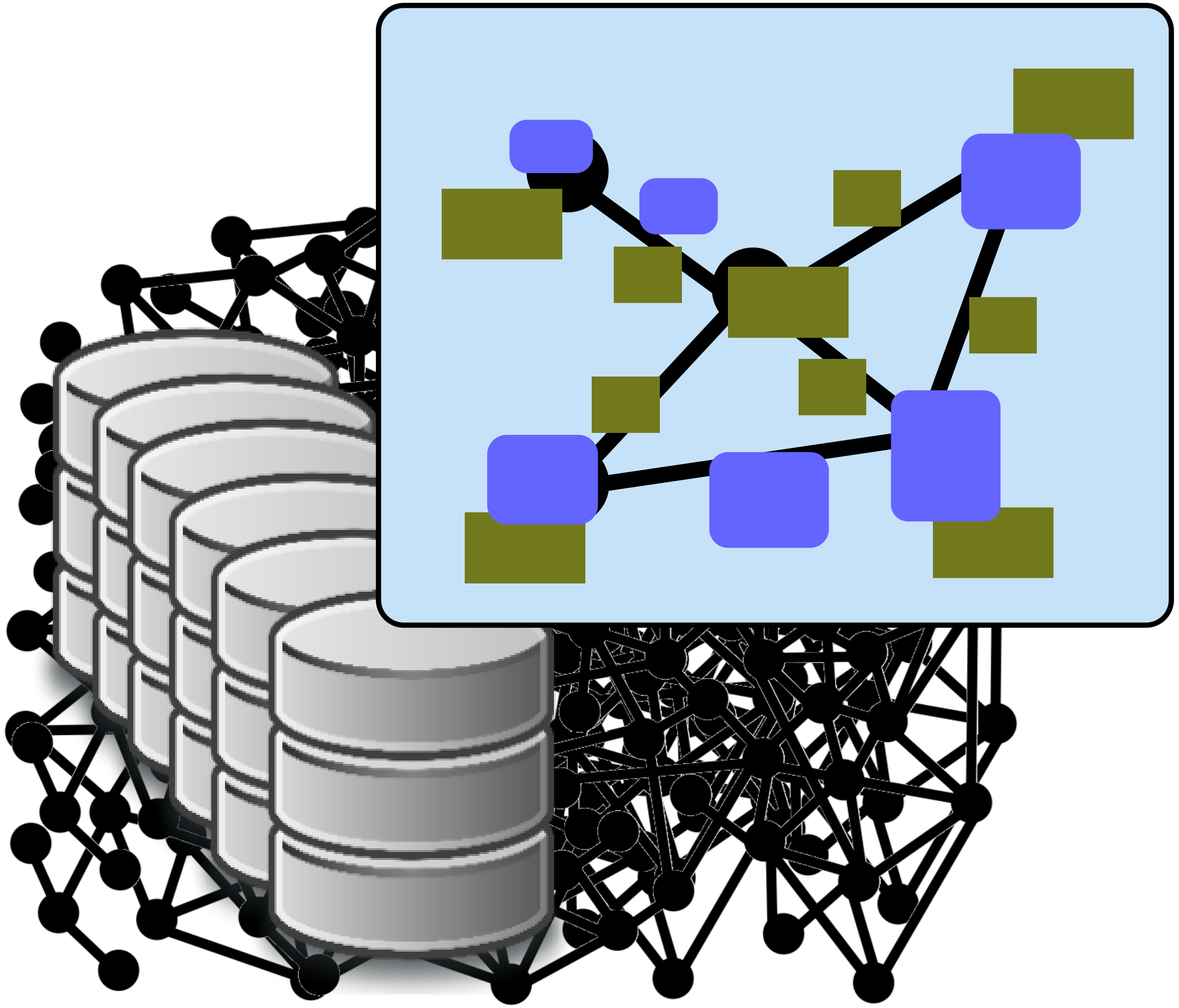
Normalization, non-linearity (optional)

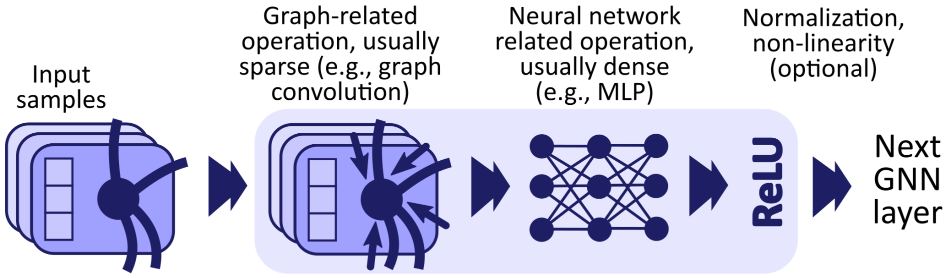


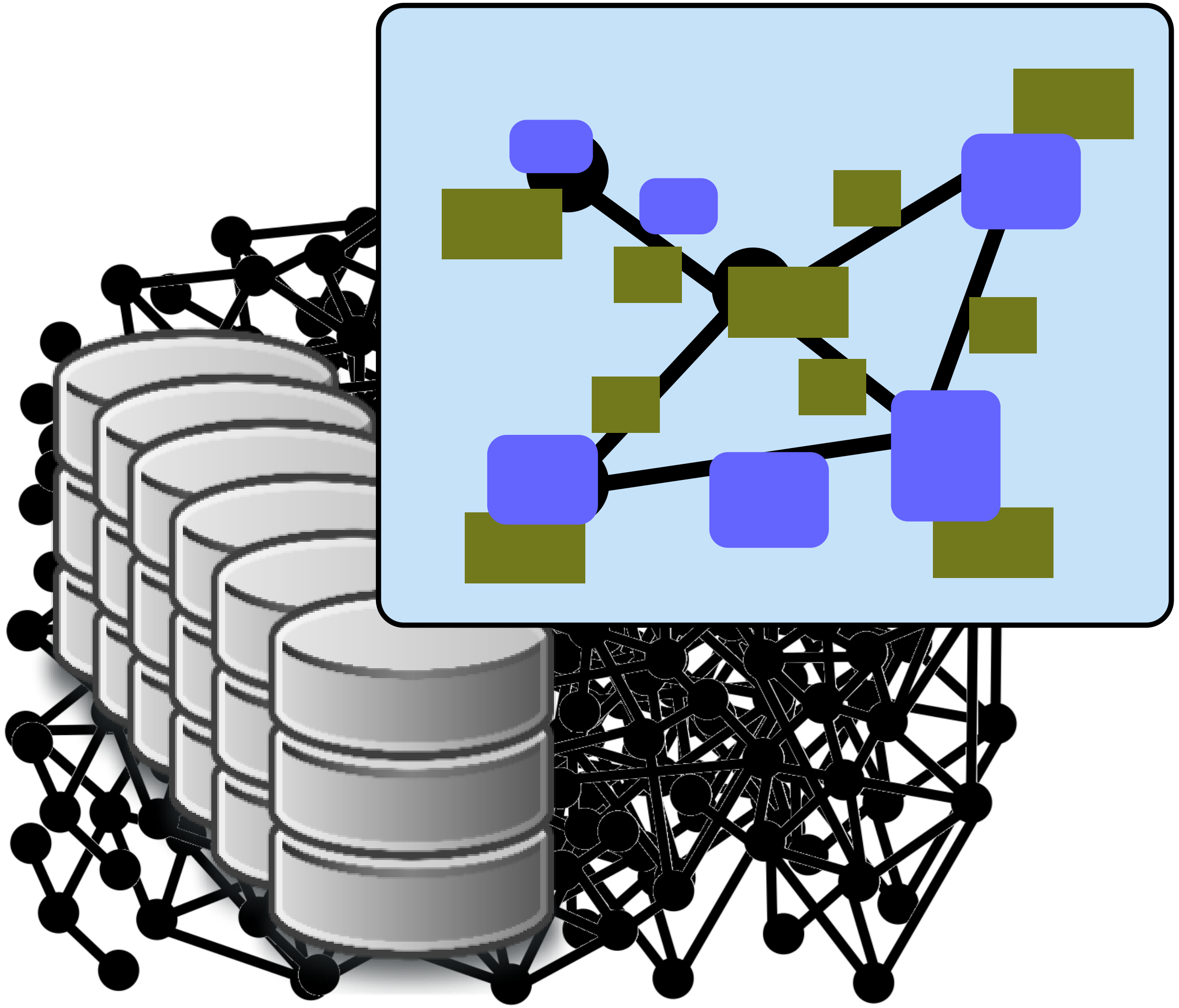
Next GNN layer

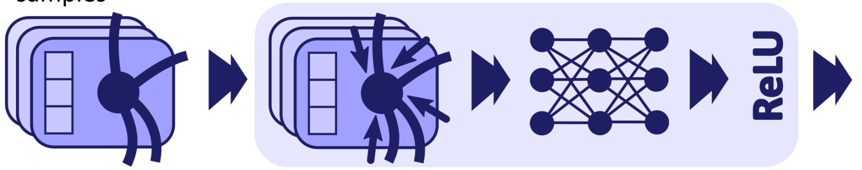




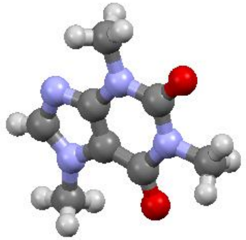


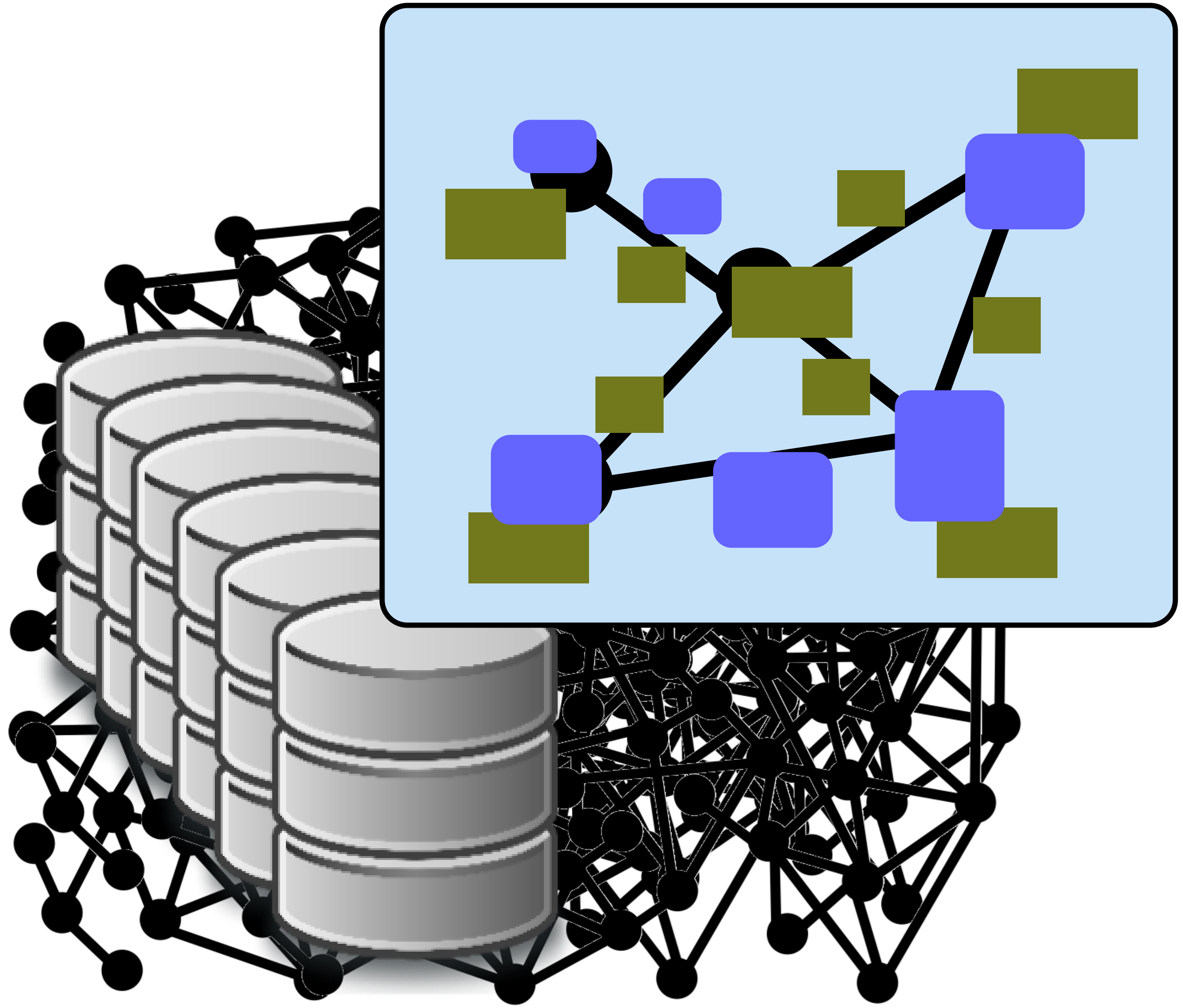

Input samples
Graph-related operation, usually sparse (e.g., graph convolution)
Neural network related operation, usually dense (e.g., MLP)
Normalization, non-linearity (optional)
ReLU
Next GNN layer

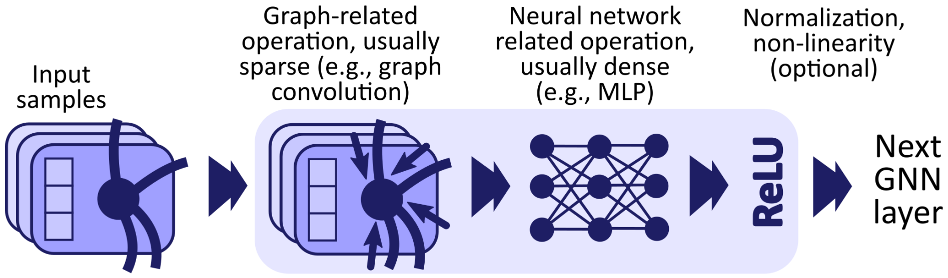




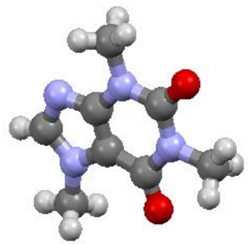
Classification or regression of graphs



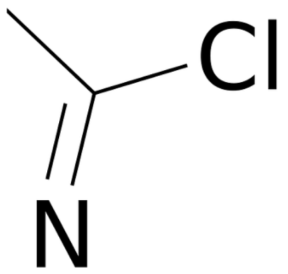


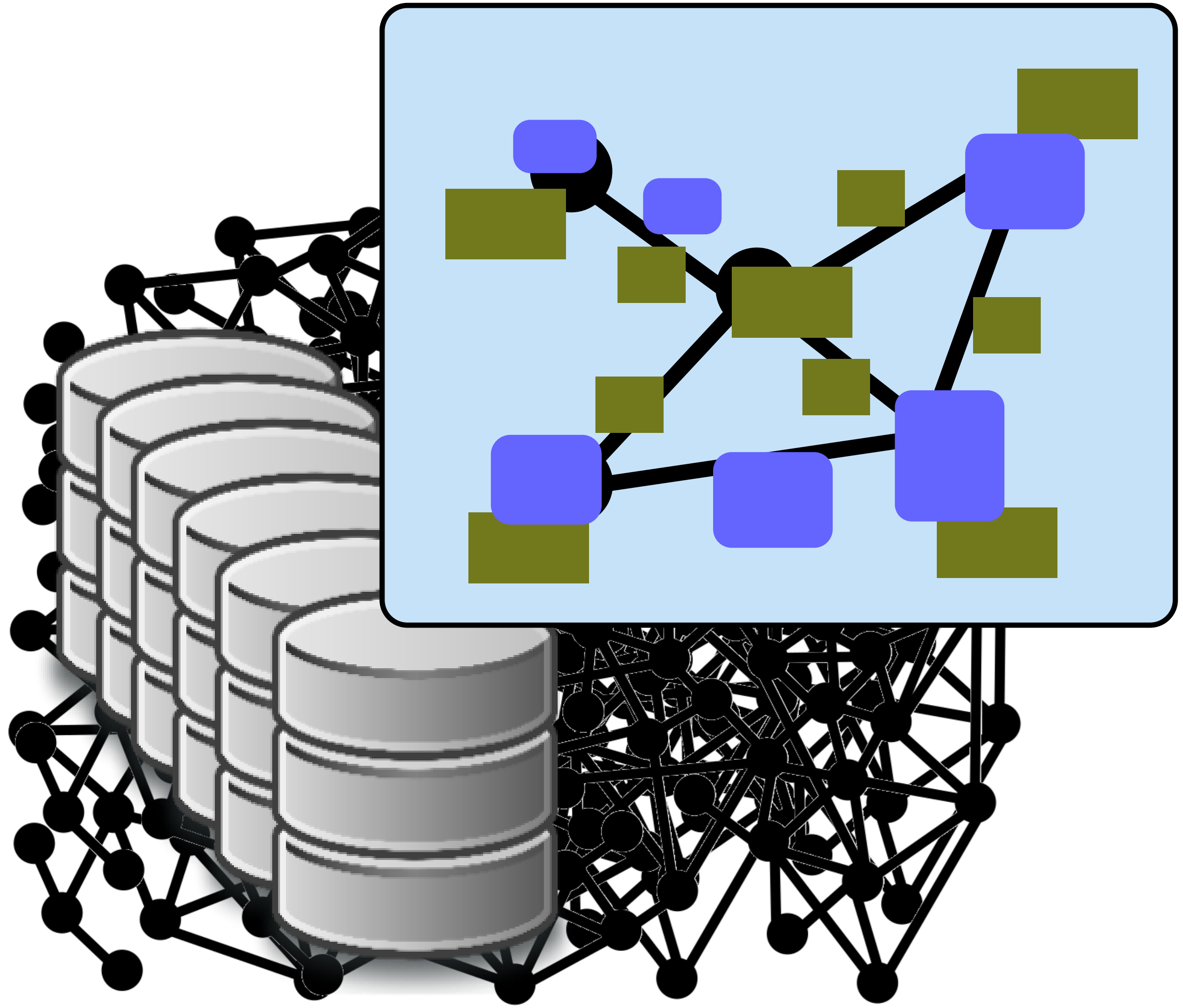


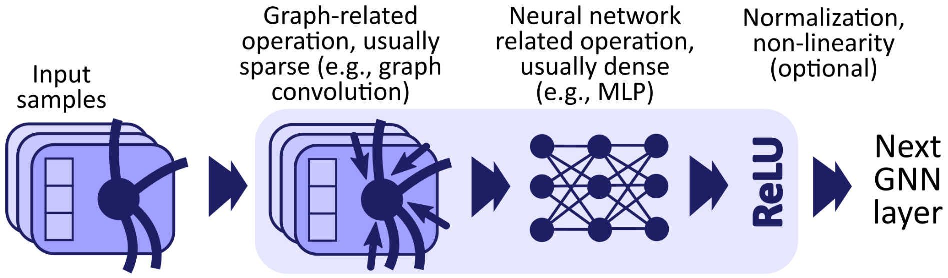
Classification or regression of graphs



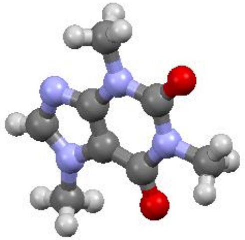
Classification or regression of edges



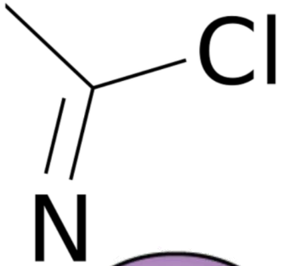




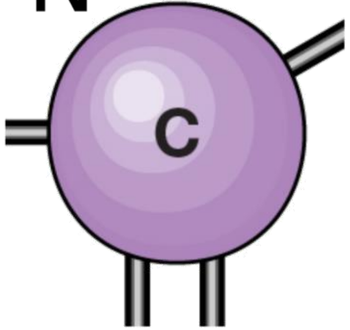
Classification or regression of graphs



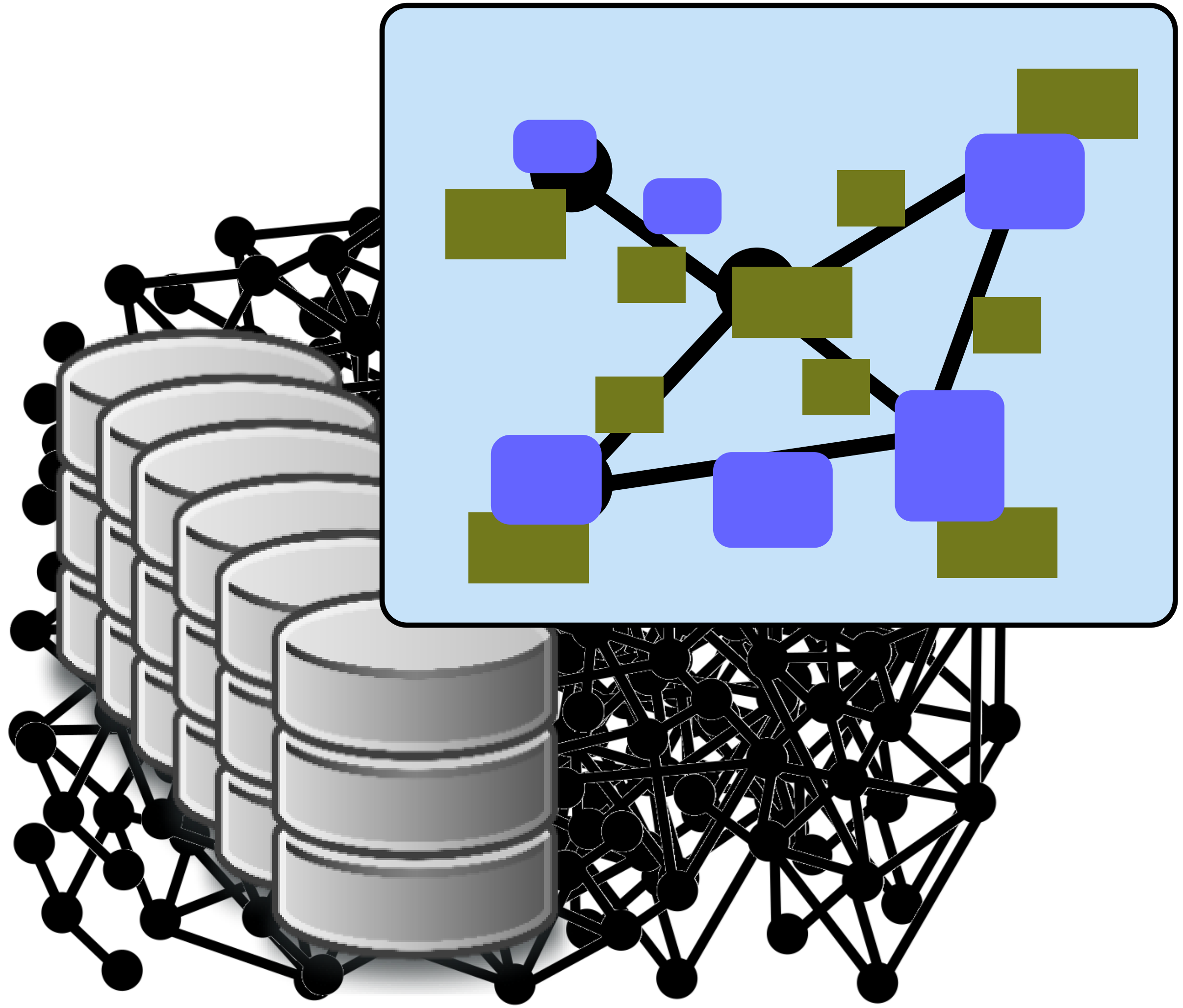
Classification or regression of edges



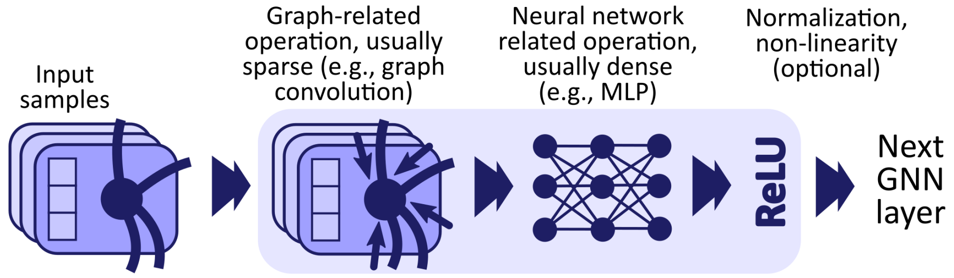
Classification or regression of nodes



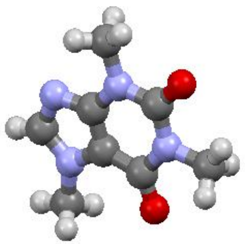
How to marry these two?



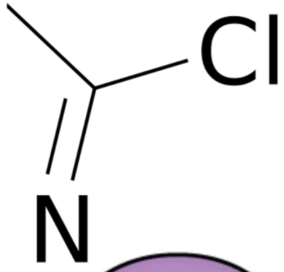
+



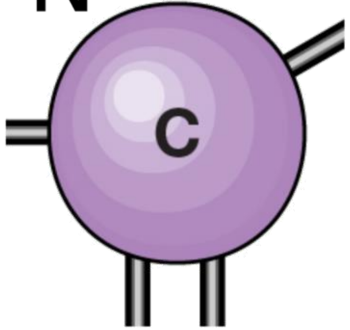
Classification or regression of graphs



Classification or regression of edges

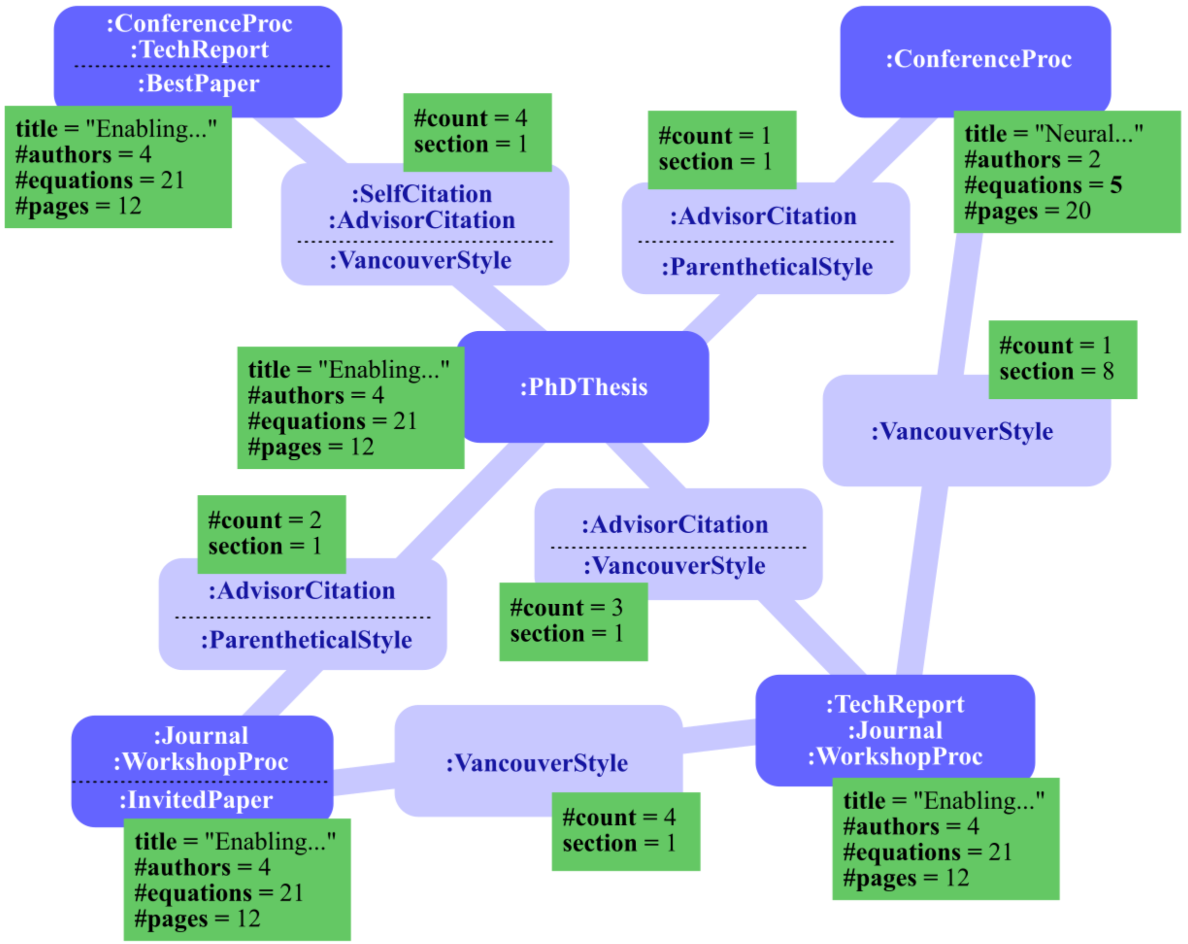


Classification or regression of nodes

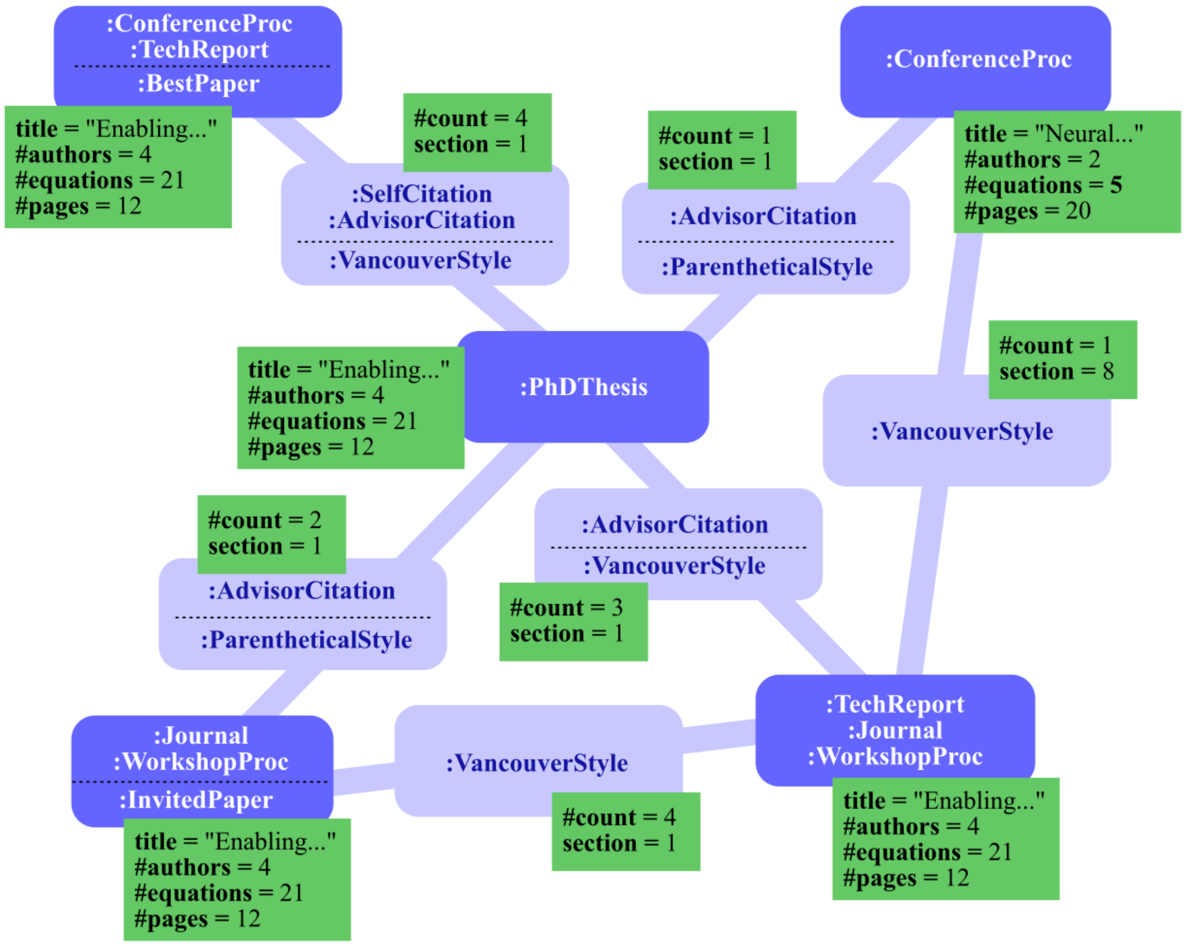


Main challenge: understanding how to merge these two conceptually

Marrying Graph Databases and GNNs

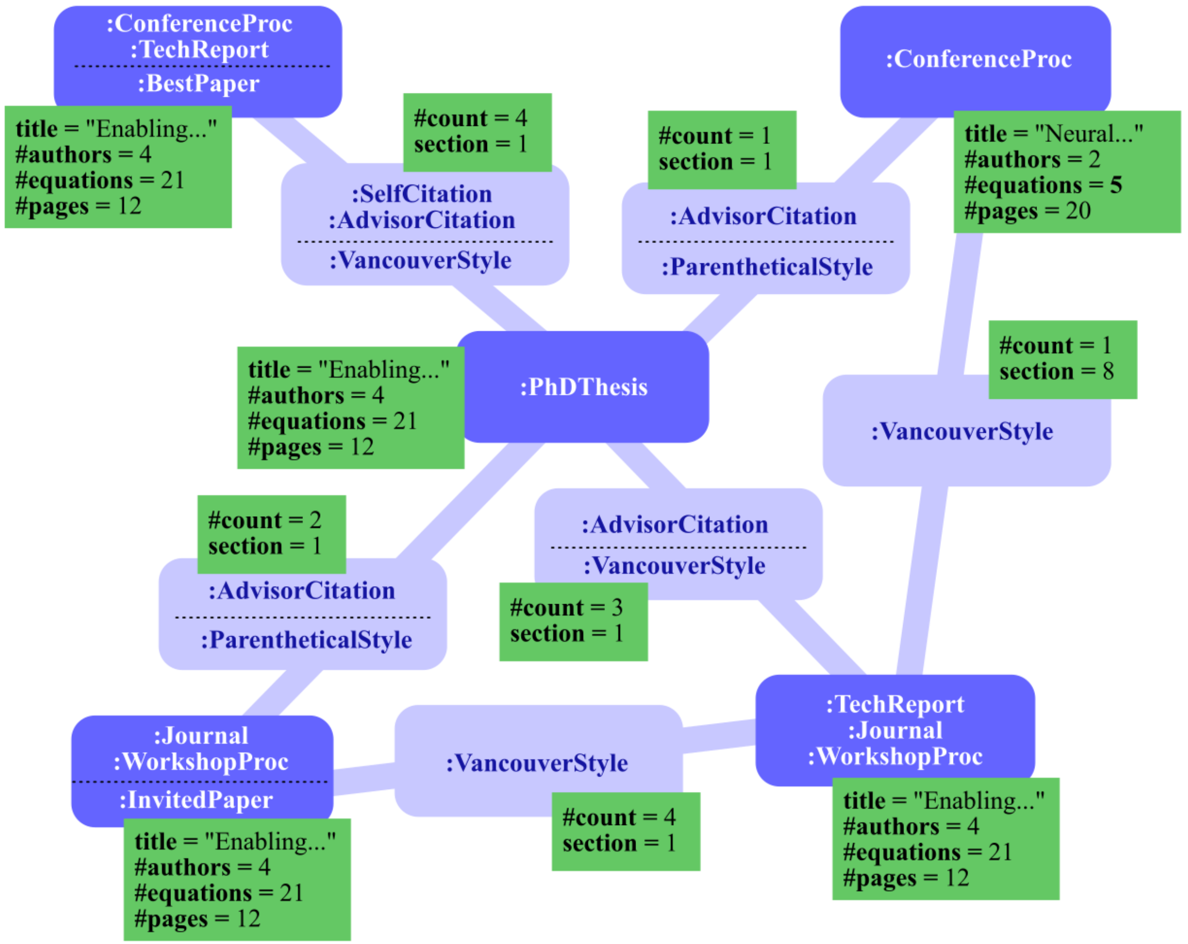


Marrying Graph Databases and GNNs



There is a neat way to think about the GNN workloads within the LPG model

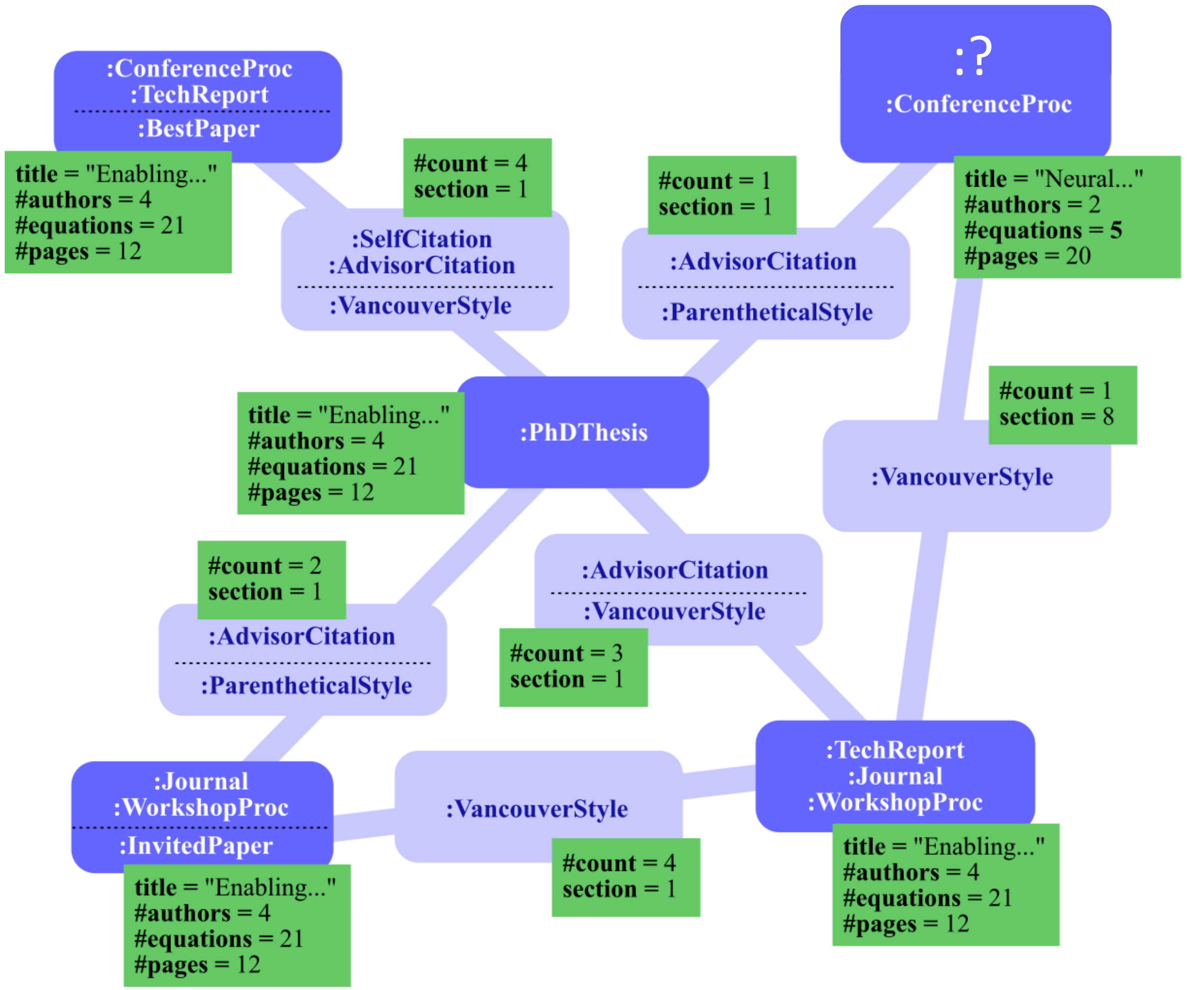
Marrying Graph Databases and GNNs



There is a neat way to think about the GNN workloads within the LPG model

Node/edge/graph classification becomes ***label prediction***

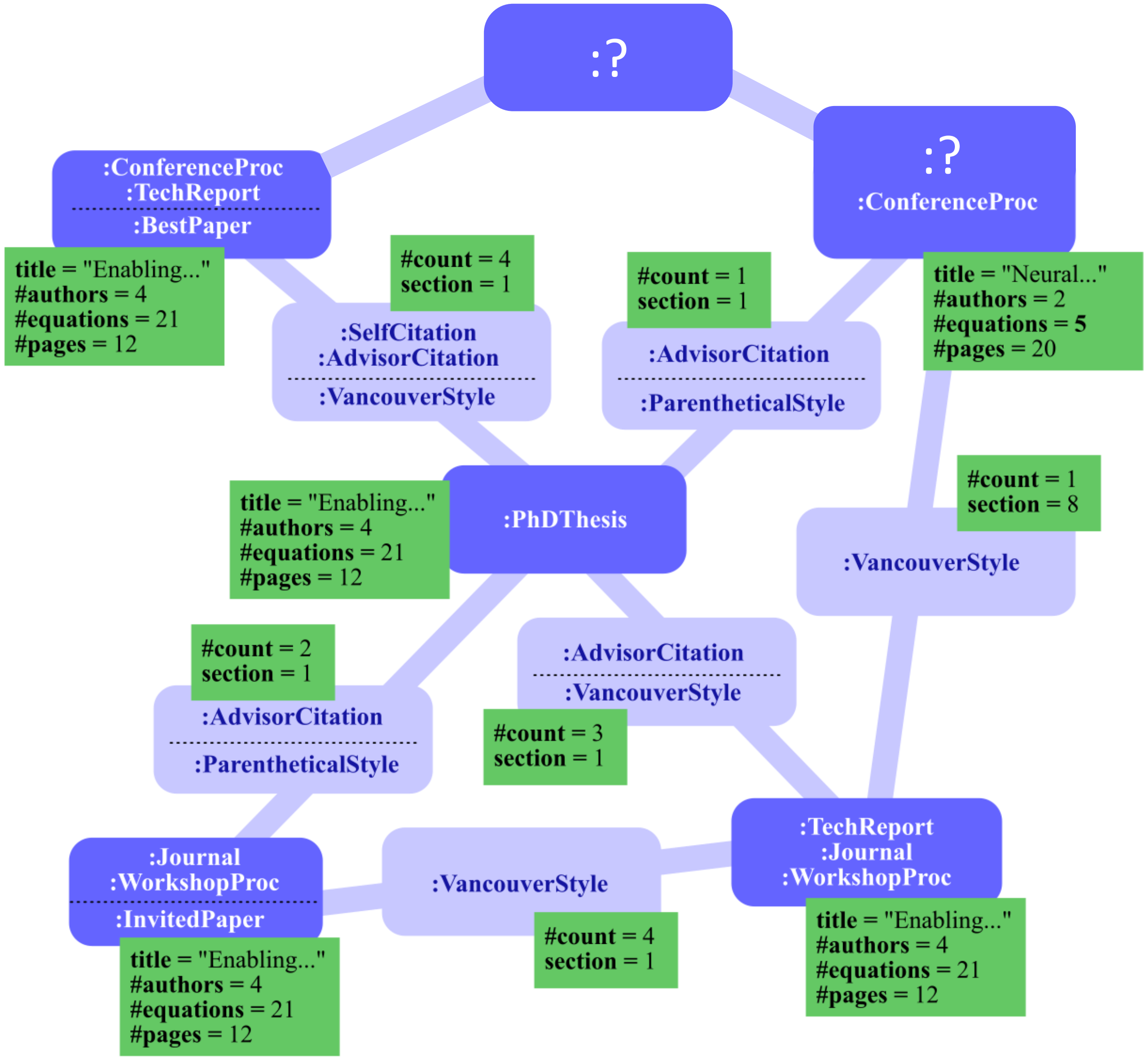
Marrying Graph Databases and GNNs



There is a neat way to think about the GNN workloads within the LPG model

Node/edge/graph classification becomes ***label prediction***

Marrying Graph Databases and GNNs



There is a neat way to think about the GNN workloads within the LPG model

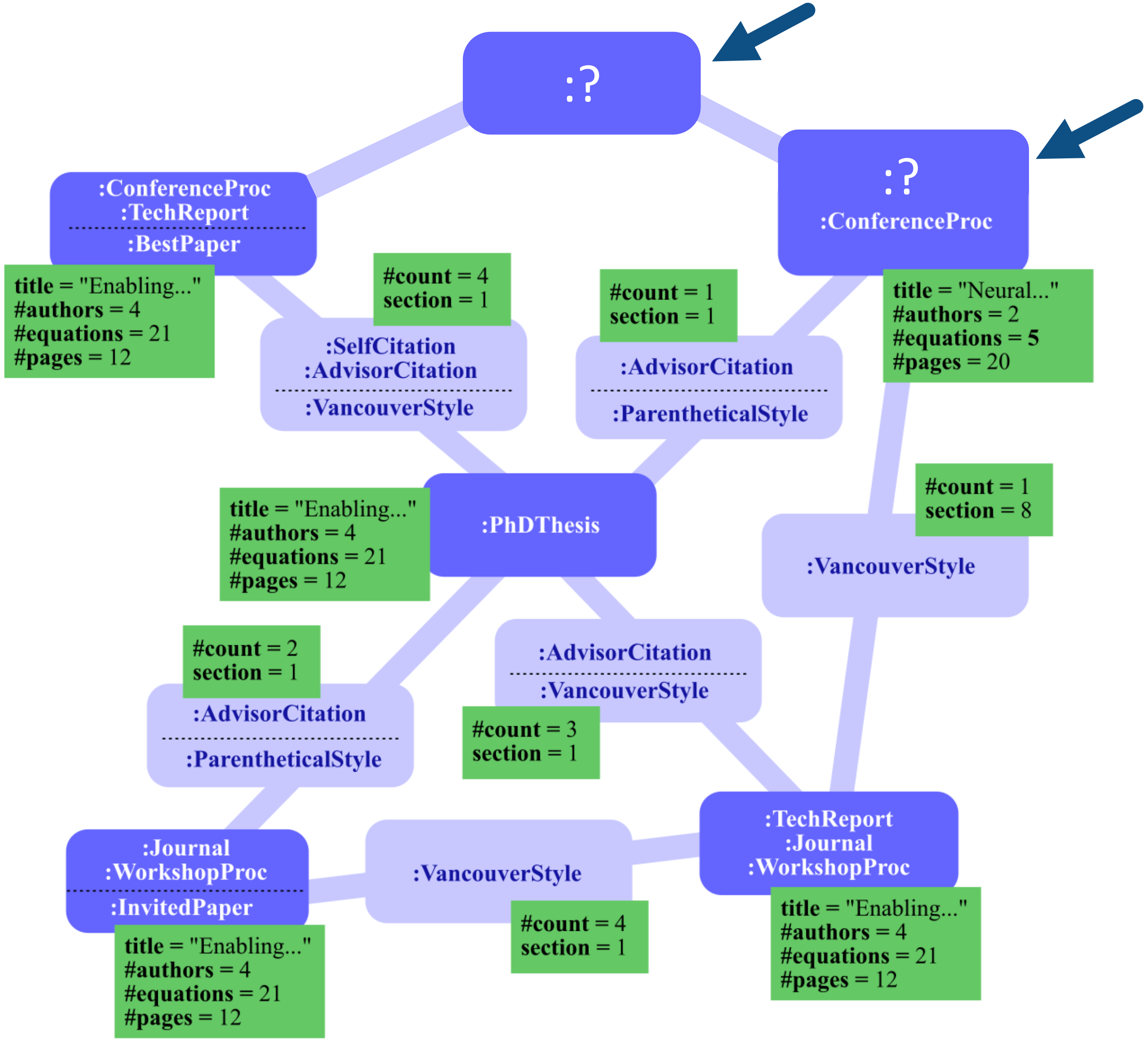
Node/edge/graph classification becomes ***label prediction***

Marrying Graph Databases and GNNs

There is a neat way to think about the GNN workloads within the LPG model

Node/edge/graph classification becomes ***label prediction***

Marrying Graph Databases and GNNs



There is a neat way to think about the GNN workloads within the LPG model

Node/edge/graph classification becomes ***label prediction***

Node/edge/graph regression becomes ***property prediction***

Marrying Graph Databases and GNNs

There is a neat way to think about the GNN workloads within the LPG model

Node/edge/graph classification becomes ***label prediction***

Node/edge/graph regression becomes ***property prediction***

Marrying Graph Databases and GNNs

There is a neat way to think about the GNN workloads within the LPG model

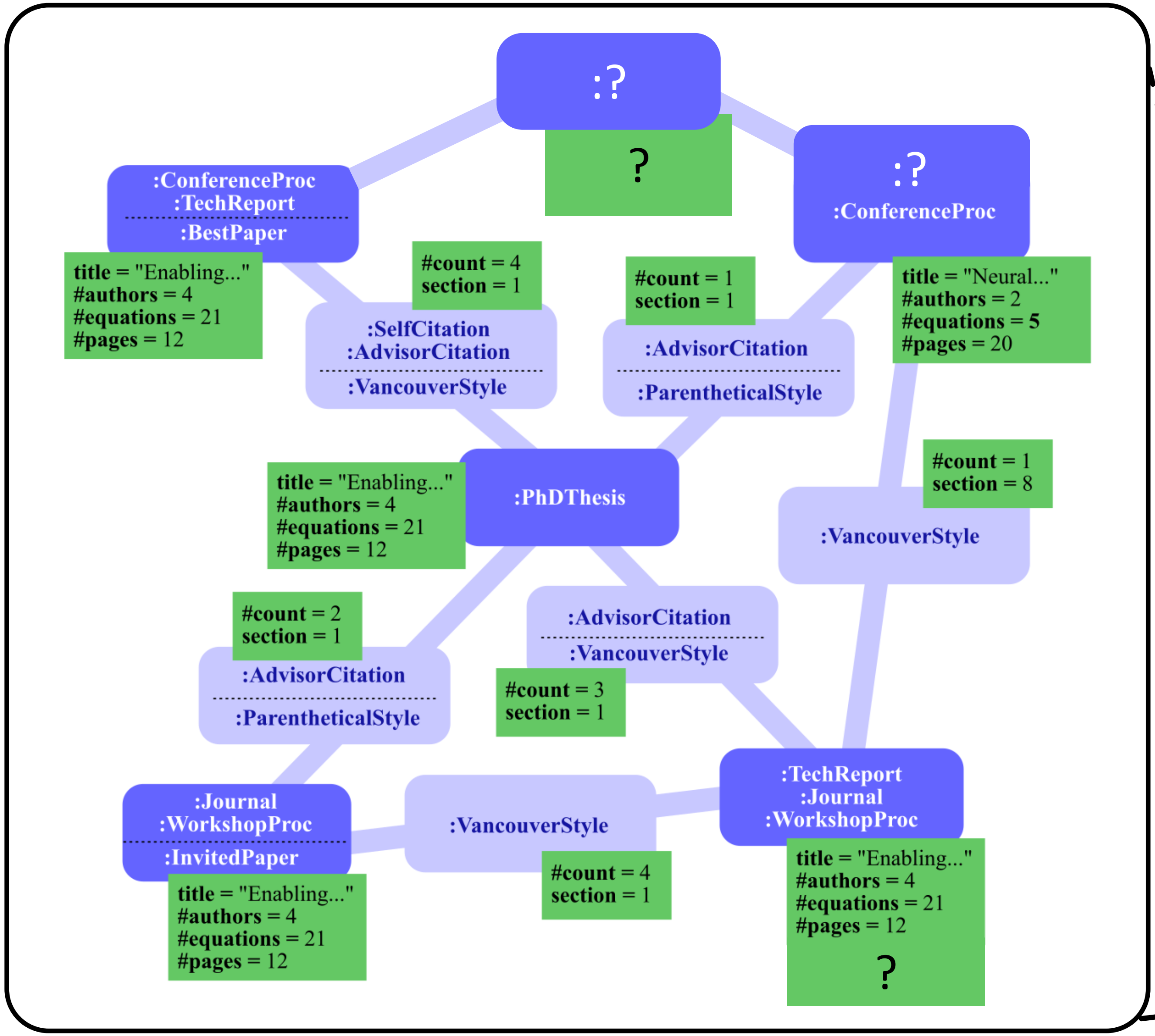
Node/edge/graph classification becomes ***label prediction***

Node/edge/graph regression becomes ***property prediction***

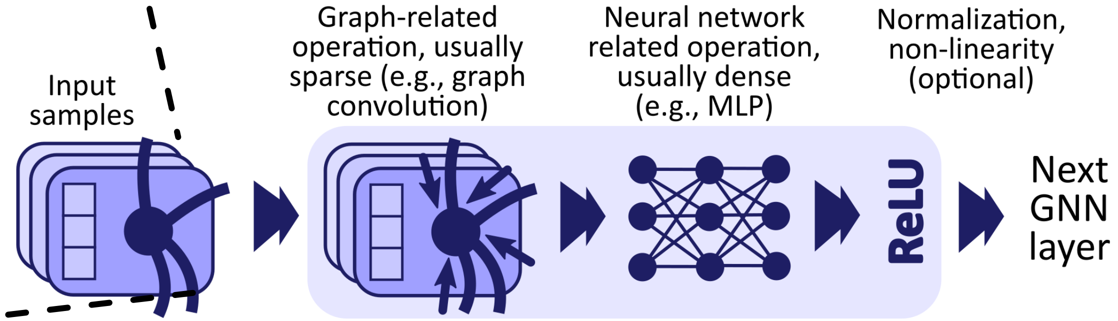
Marrying Graph Databases and GNNs

How do we enable these GNN workloads on the LPG graph datasets **while harnessing all the existing label/property information**?

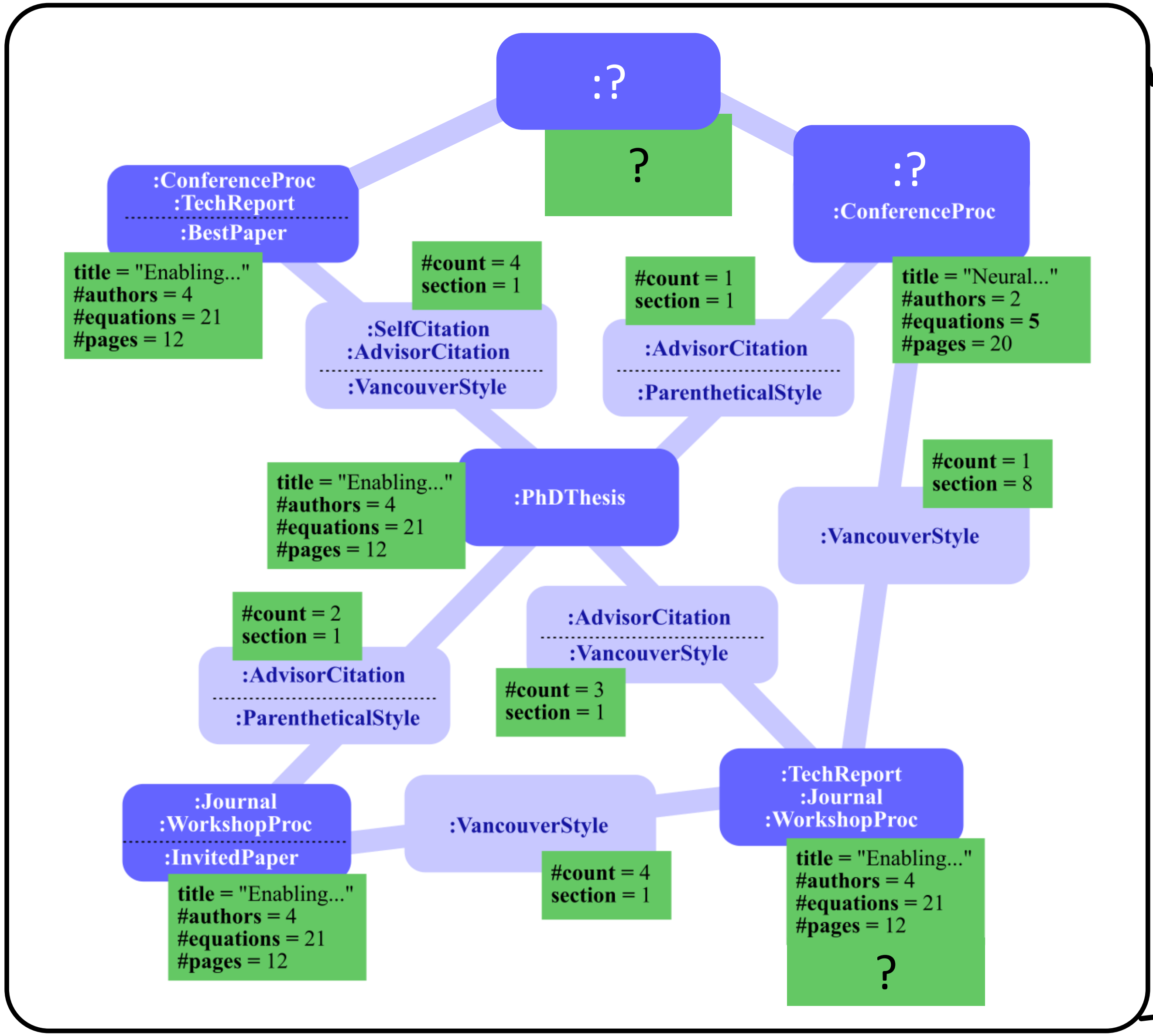
Marrying Graph Databases and GNNs



How do we enable these GNN workloads on the LPG graph datasets **while harnessing all the existing label/property information**?

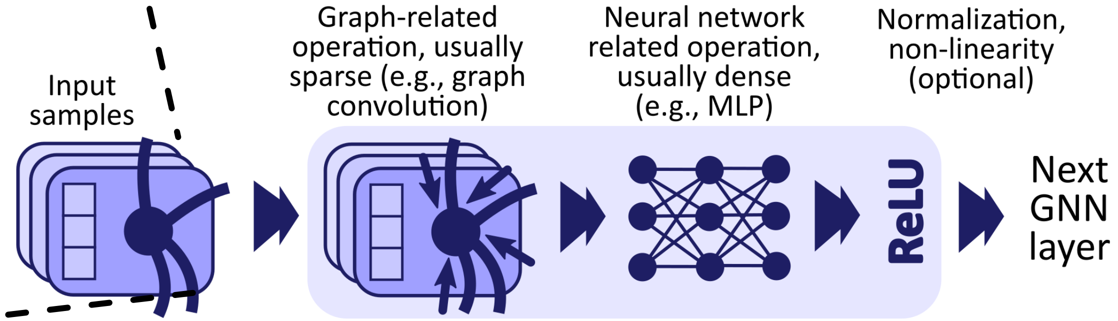


Marrying Graph Databases and GNNs



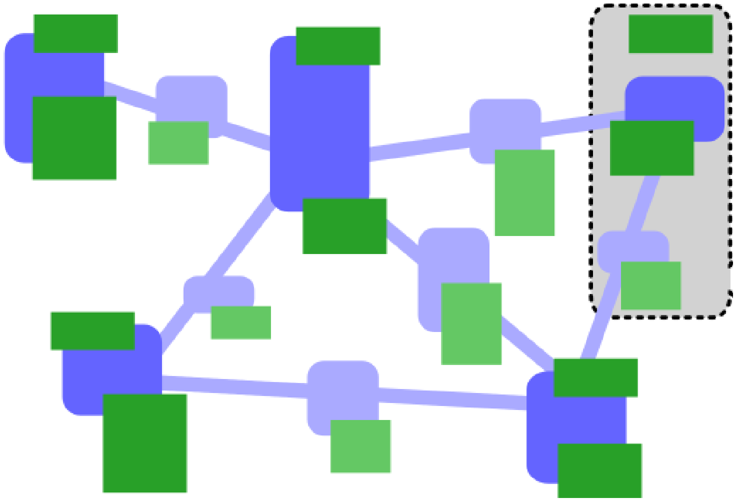
How do we enable these GNN workloads on the LPG graph datasets **while harnessing all the existing label/property information?**

We need the right encoder!

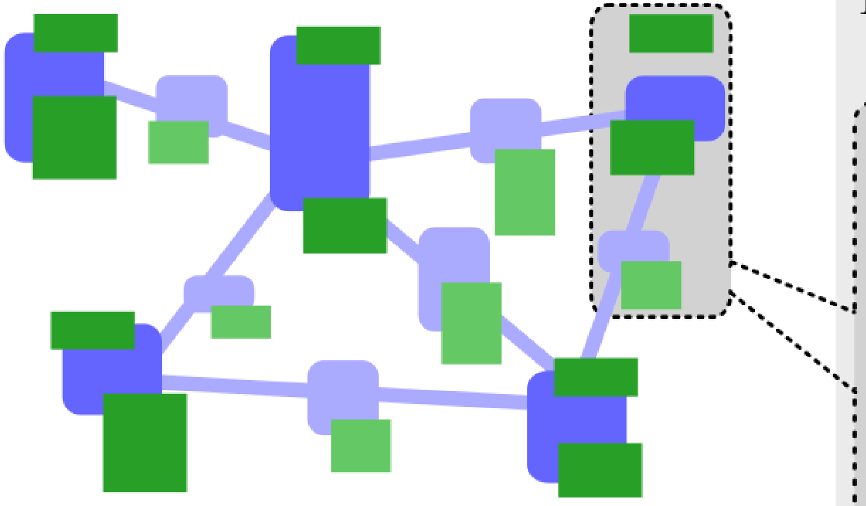


LPG2vec: Encoding LPG Datasets into a Format Digestible by GNNs Pipelines

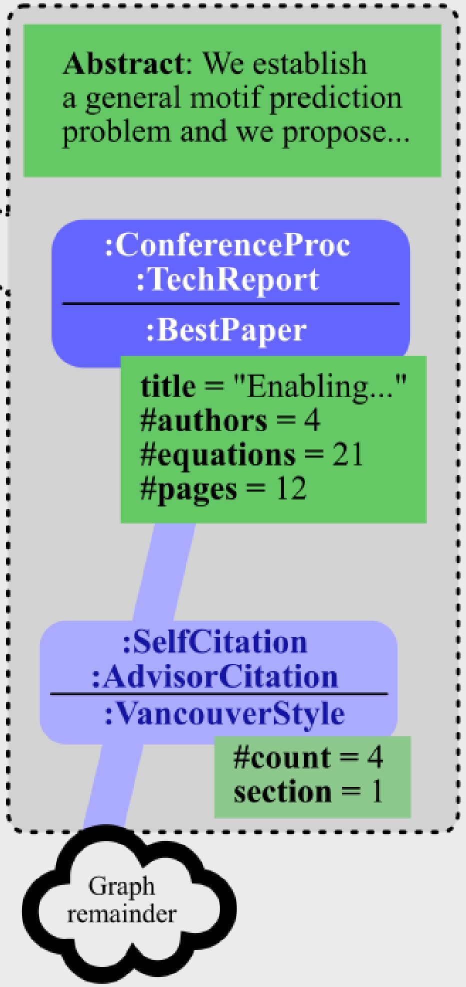
LPG2vec: Encoding LPG Datasets into a Format Digestible by GNNs Pipelines



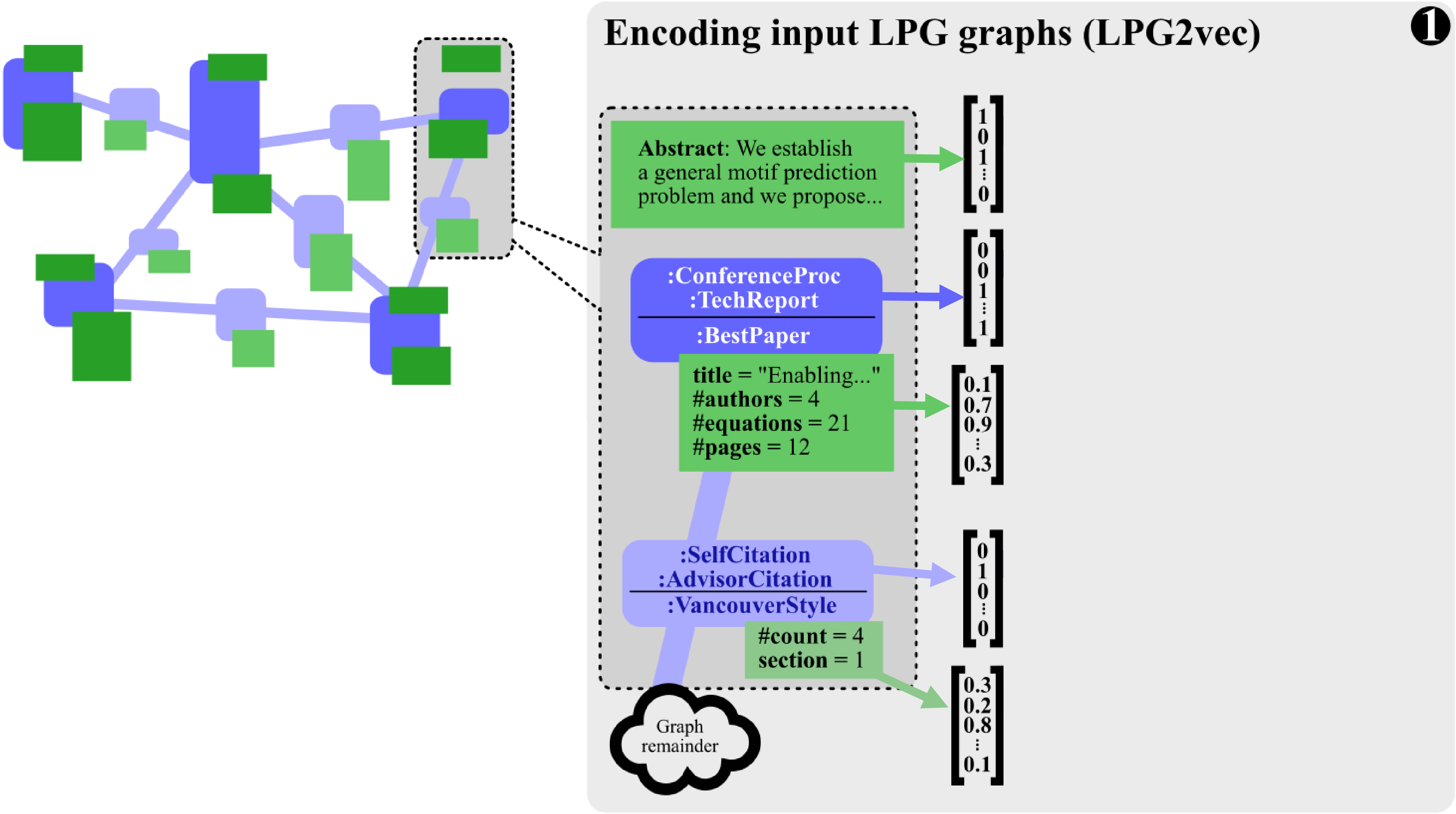
LPG2vec: Encoding LPG Datasets into a Format Digestible by GNNs Pipelines



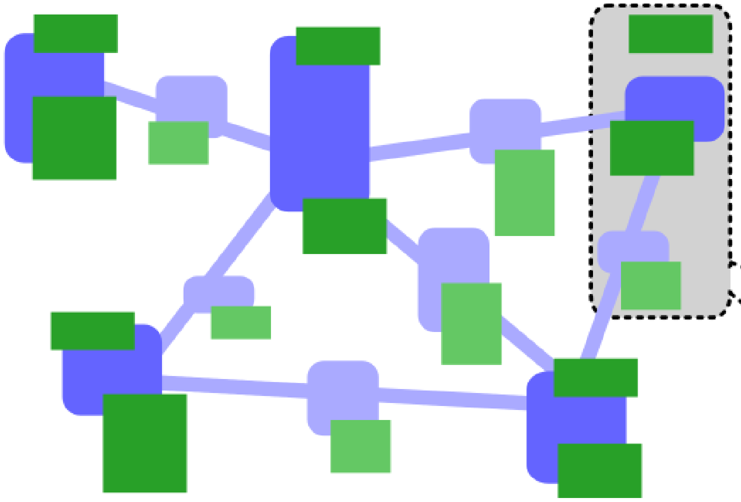
Encoding input LPG graphs (LPG2vec) ❶

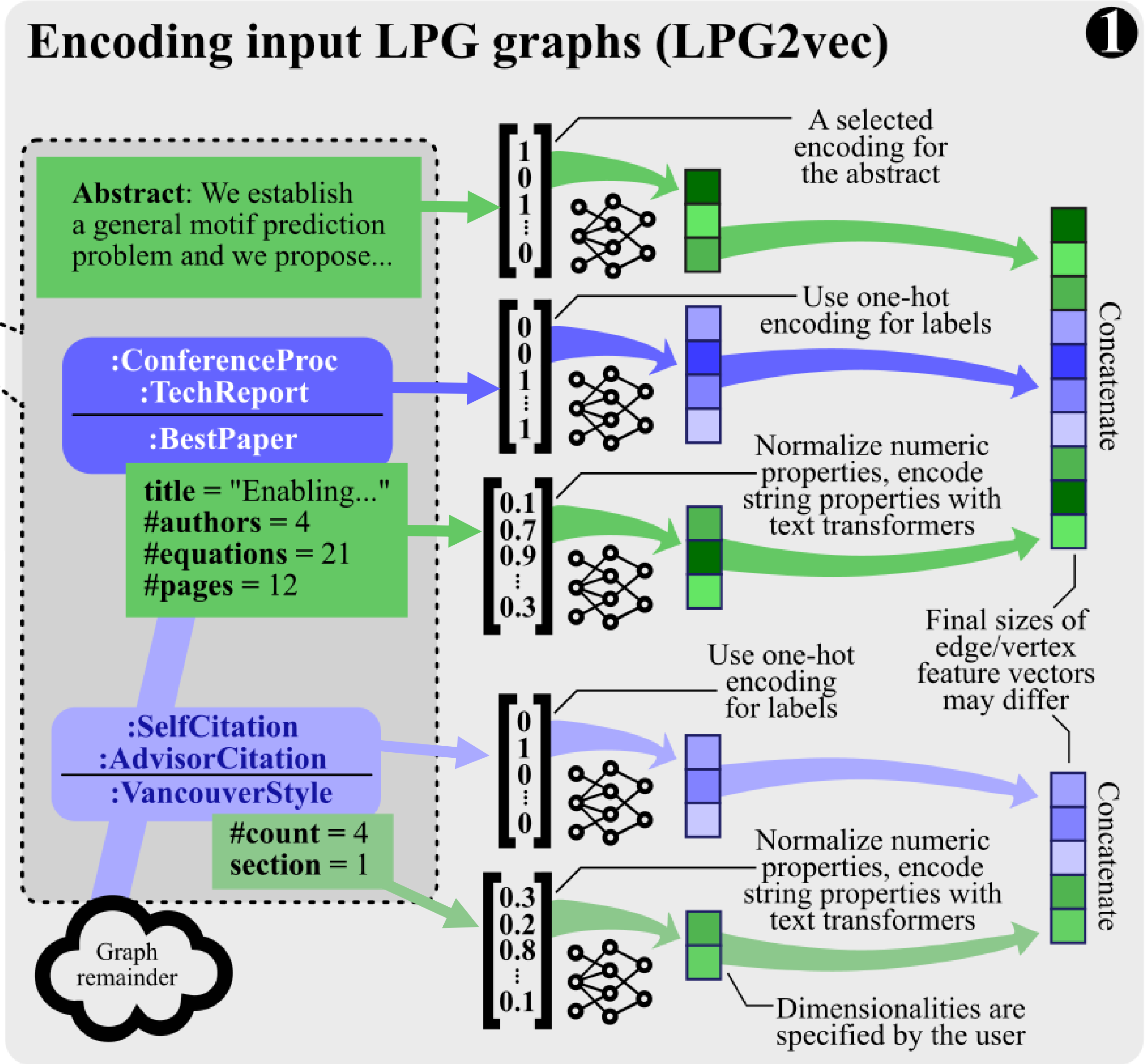


LPG2vec: Encoding LPG Datasets into a Format Digestible by GNNs Pipelines

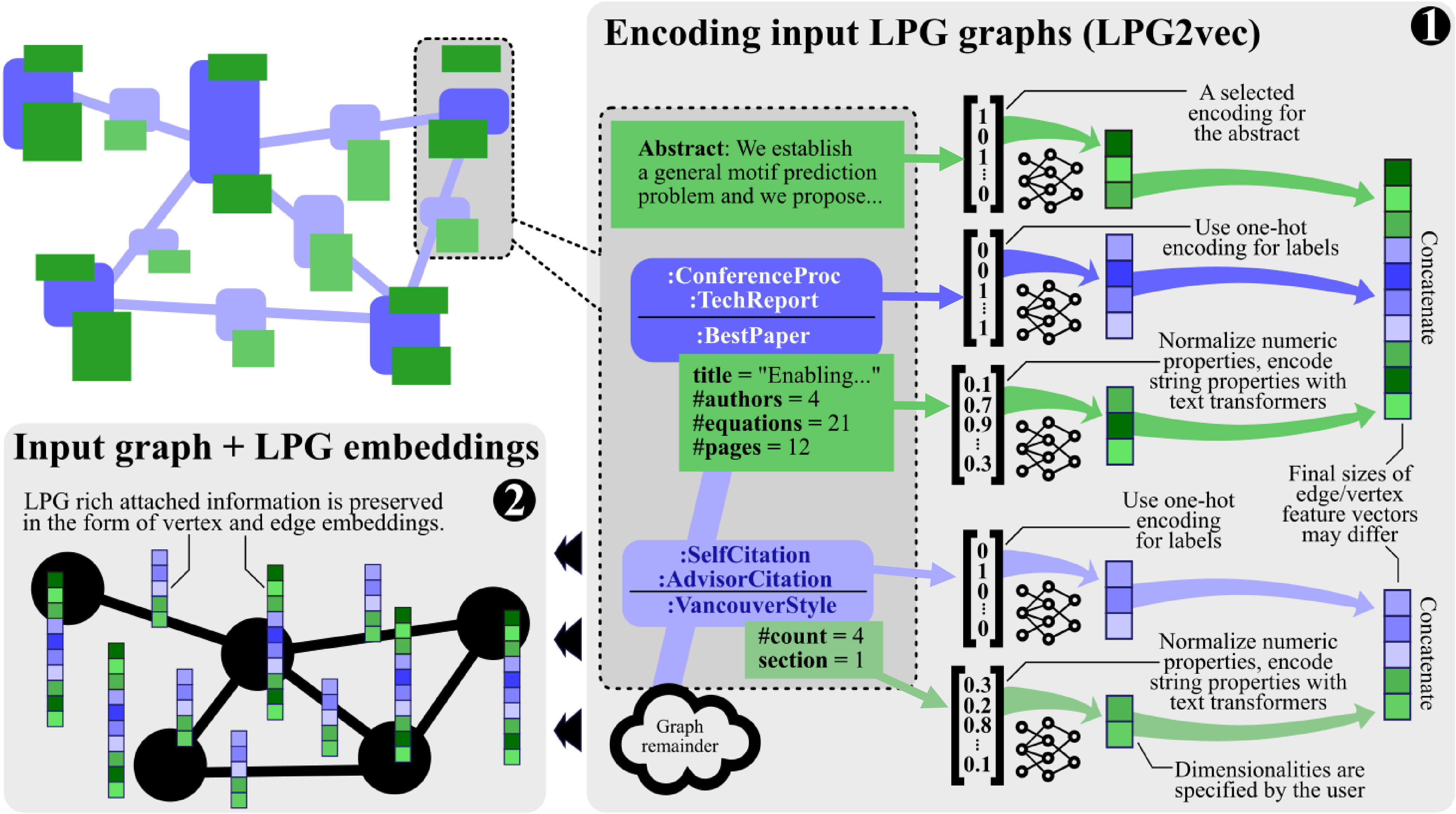


LPG2vec: Encoding LPG Datasets into a Format Digestible by GNNs Pipelines

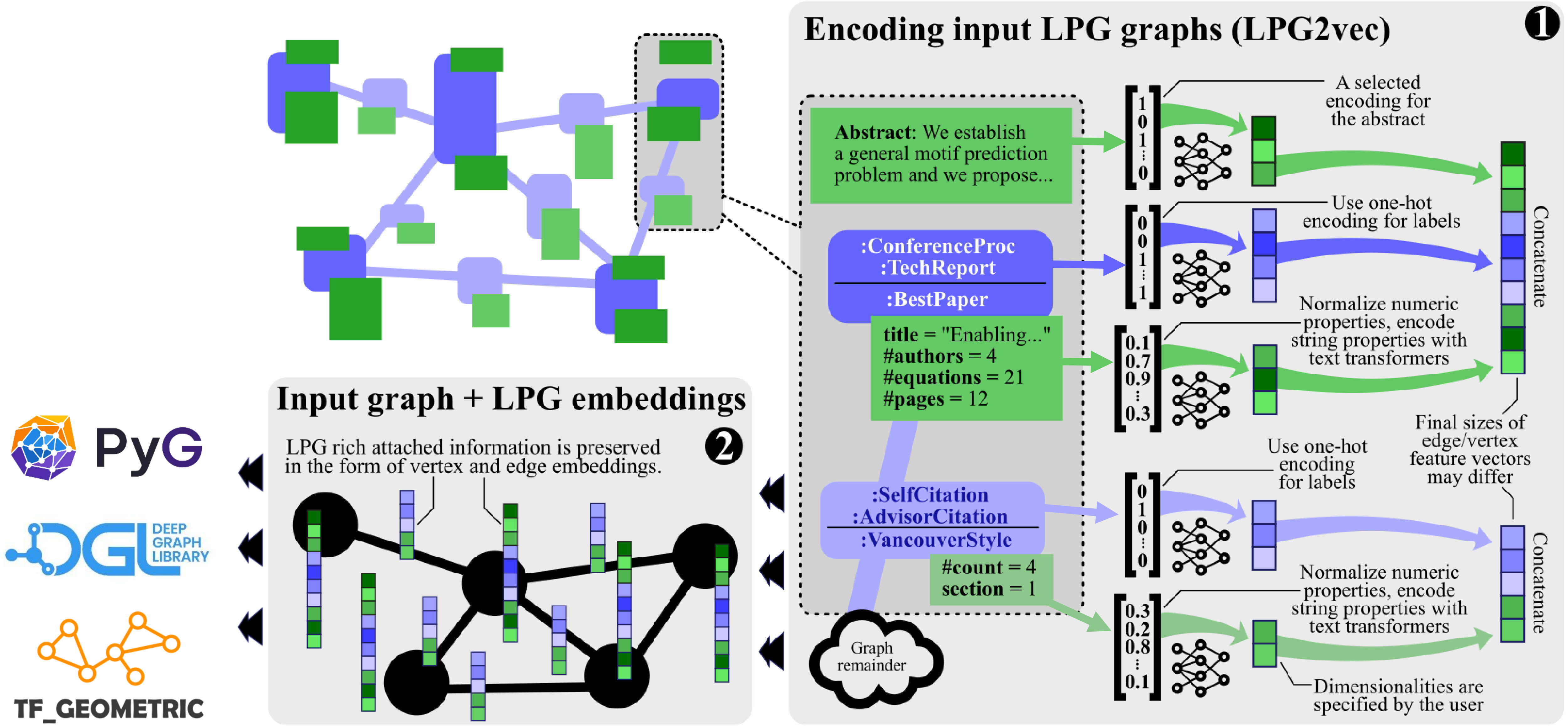




LPG2vec: Encoding LPG Datasets into a Format Digestible by GNNs Pipelines



LPG2vec: Encoding LPG Datasets into a Format Digestible by GNNs Pipelines



Evaluation: Used Machine & Objectives

Evaluation: Used Machine & Objectives

CSCS Cray Piz Daint & Ault
64GB – 2TB memory per server

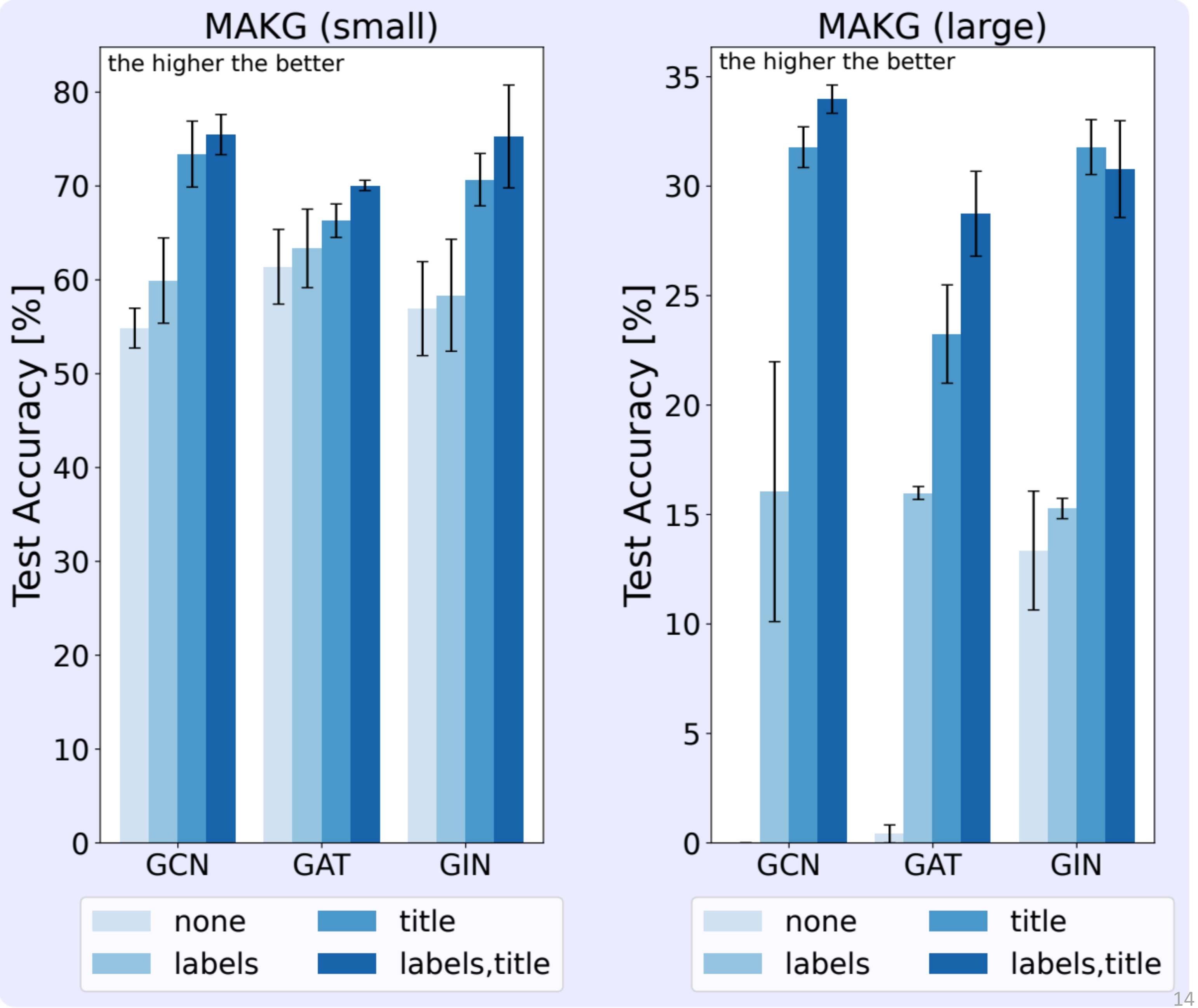
Evaluation: Used Machine & Objectives

CSCS Cray Piz Daint & Ault
64GB – 2TB memory per server

Main goal: show that **LPG2vec successfully harnesses the label and property information** from the LPG graph datasets to offer more accurate predictions in graph ML tasks

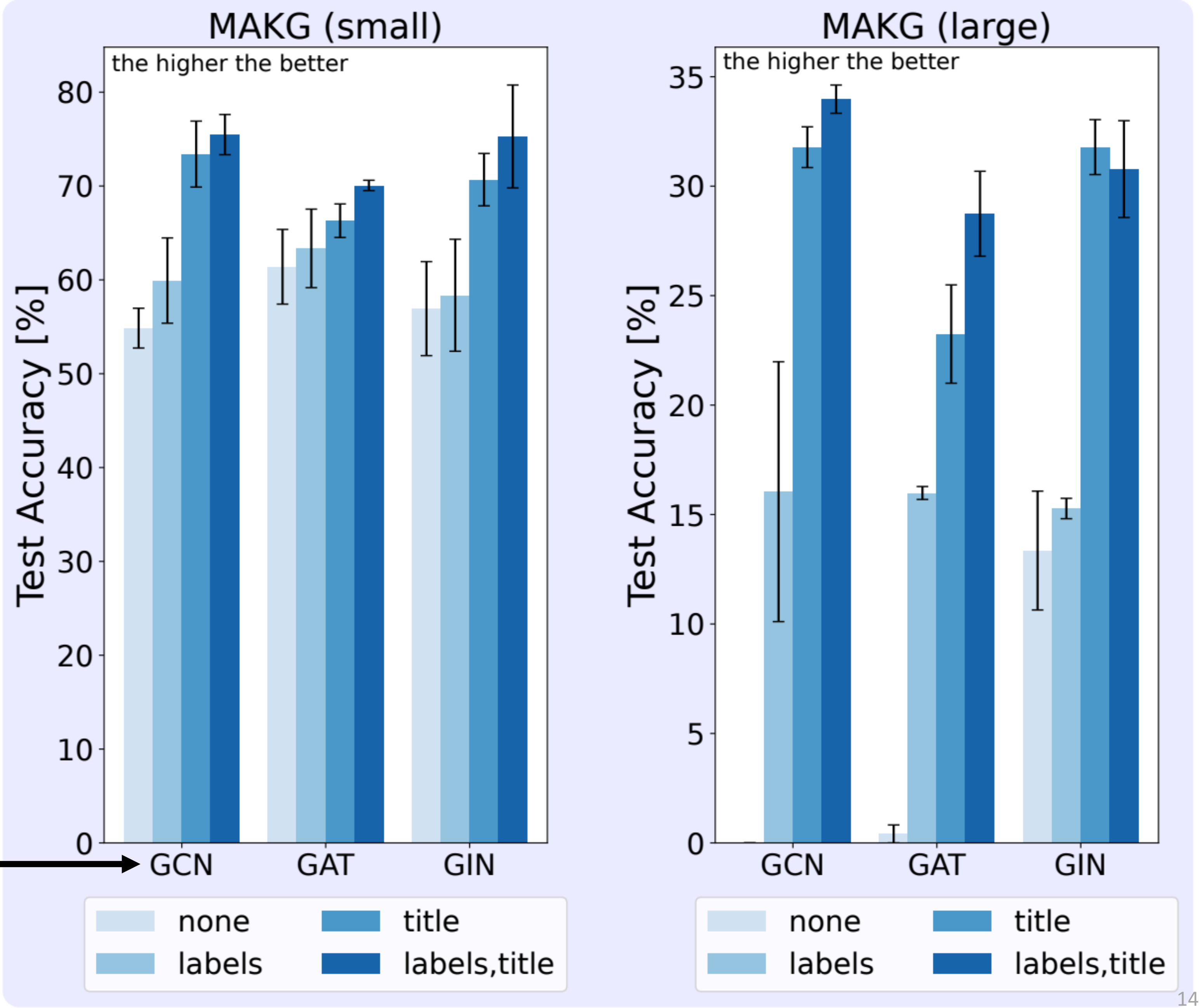
Node Classification, aka Label Prediction

Task: predict the research area of the publication



Node Classification, aka Label Prediction

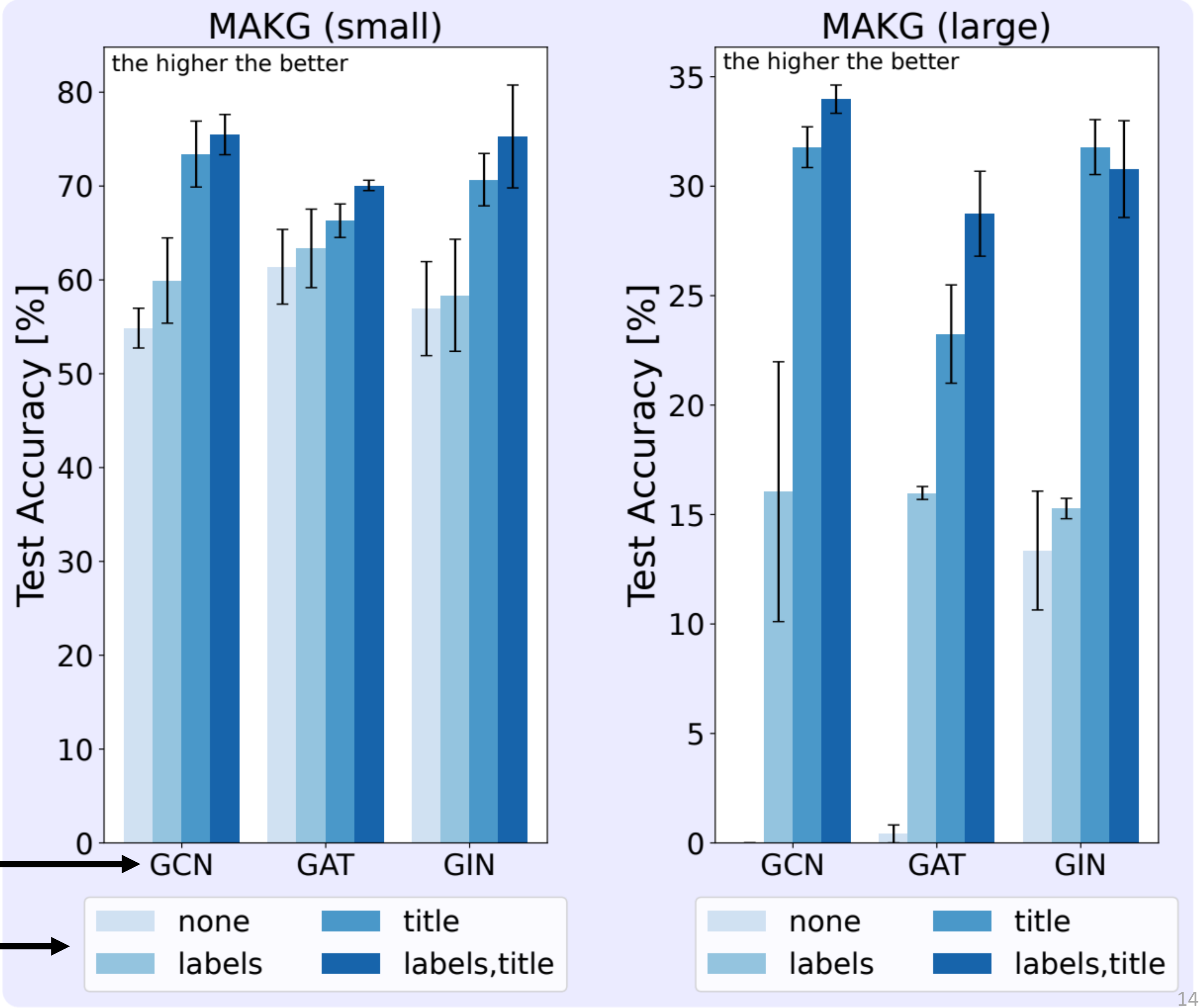
Task: predict the research area of the publication



What GNN model is considered?

Node Classification, aka Label Prediction

Task: predict the research area of the publication



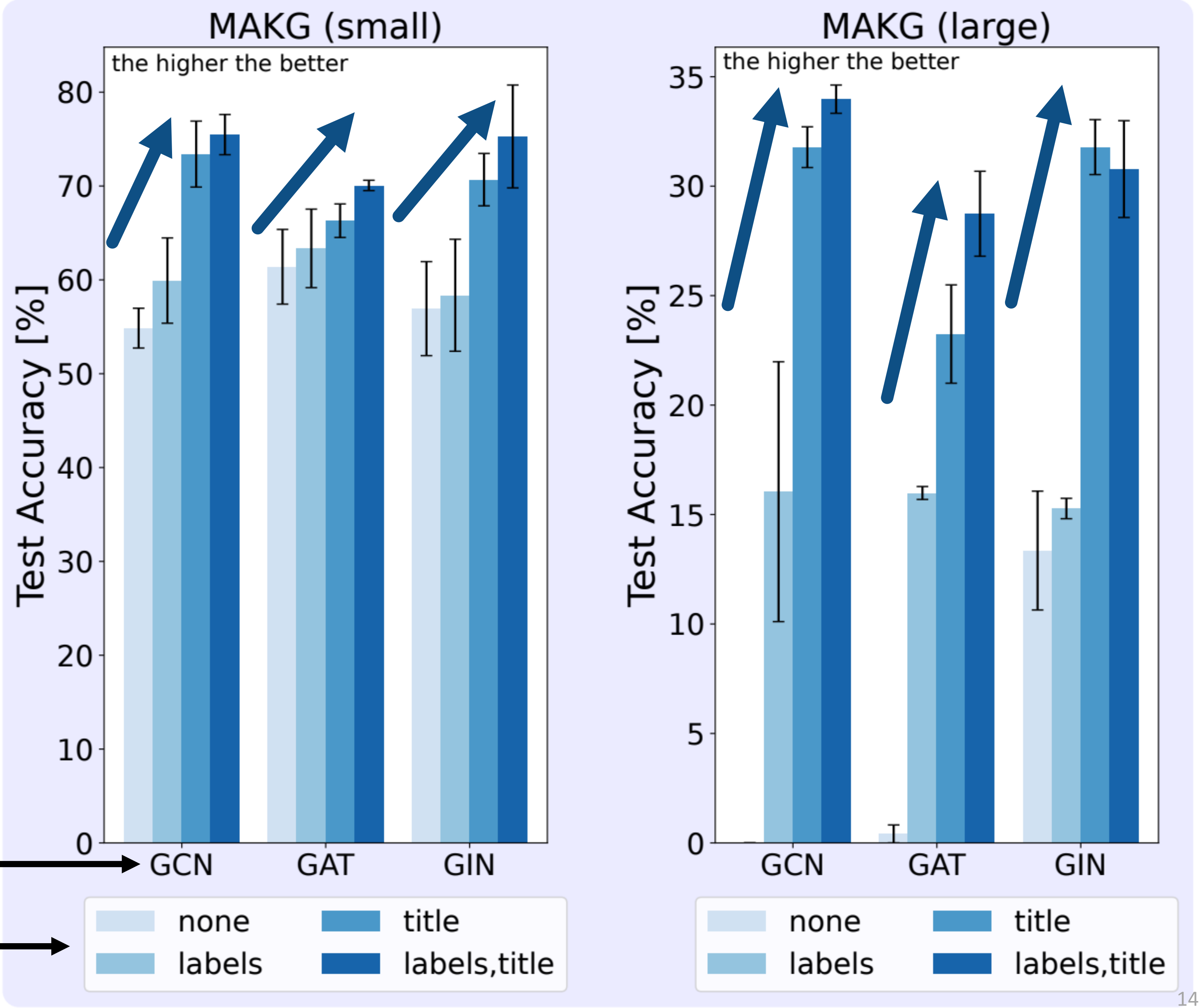
What GNN model is considered?

What label/property data is considered?

Node Classification, aka Label Prediction

Task: predict the research area of the publication

Better accuracy with labels/properties, it is important to use them both



What GNN model is considered?

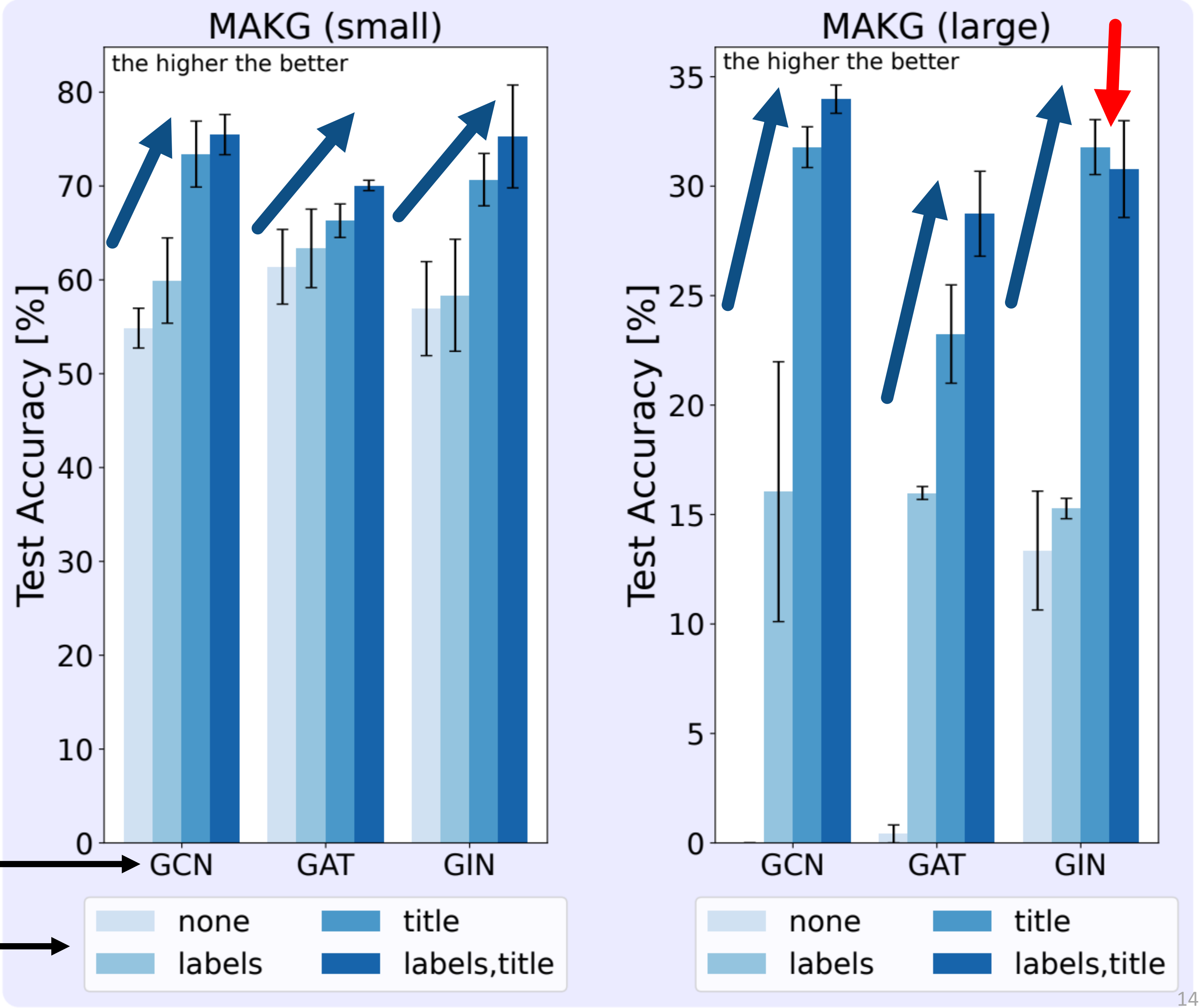
What label/property data is considered?

Node Classification, aka Label Prediction

Task: predict the research area of the publication

Better accuracy with labels/properties, it is important to use them both

...not always?



What GNN model is considered?

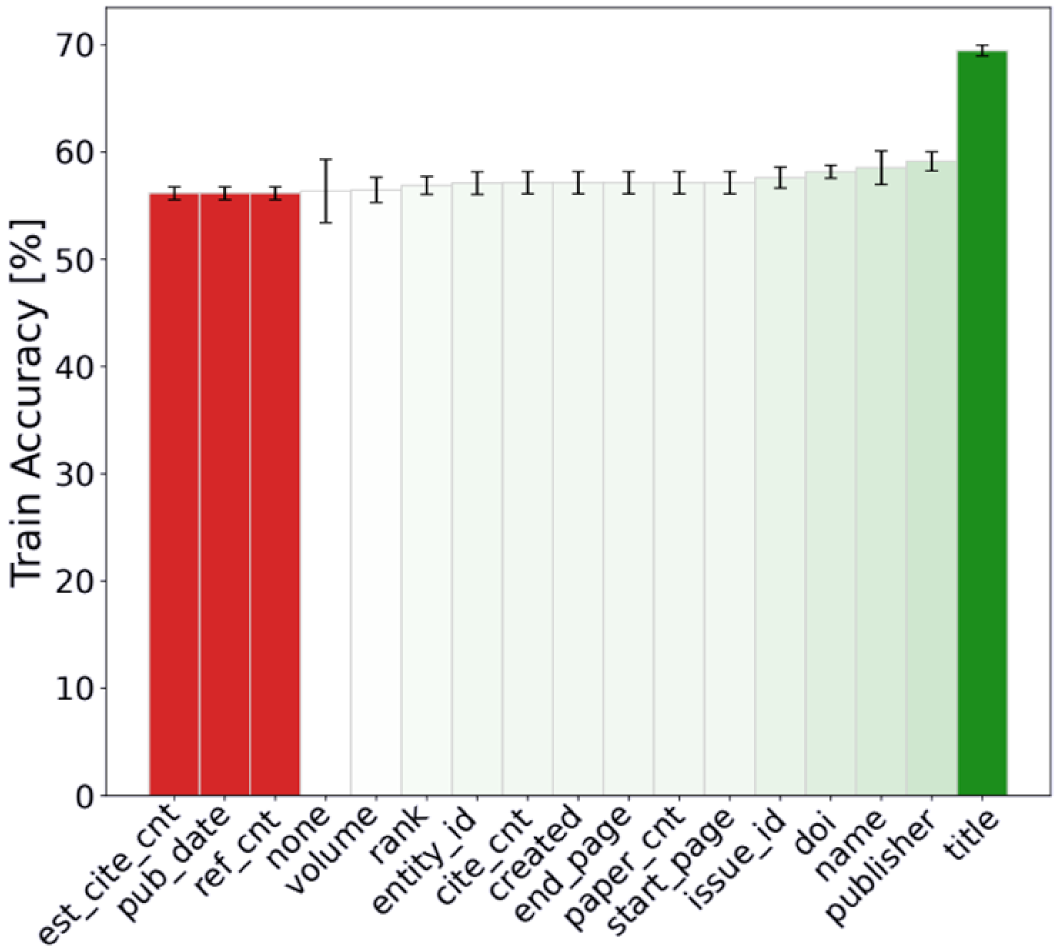
What label/property data is considered?

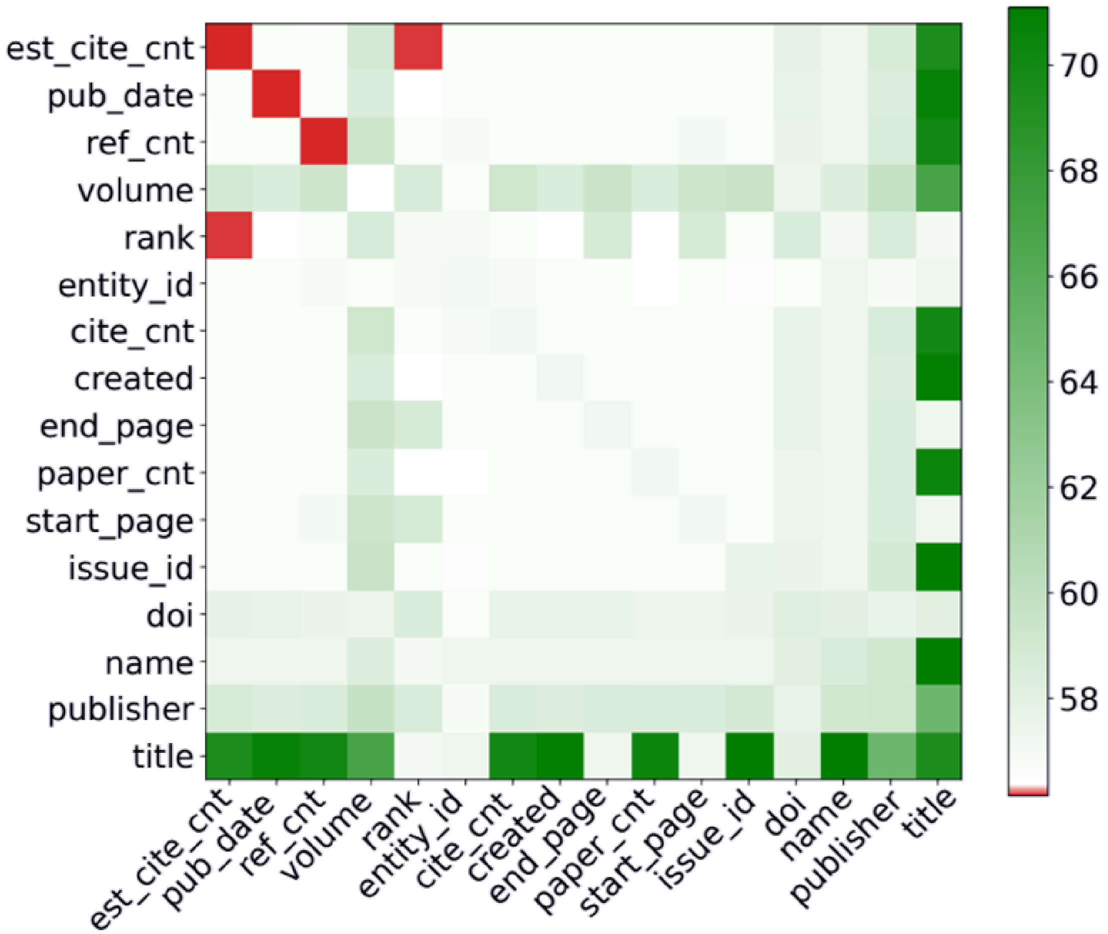
Node Classification, aka Label Prediction

Task: predict the research area of the publication

Node Classification, aka Label Prediction

Task: predict the research area of the publication

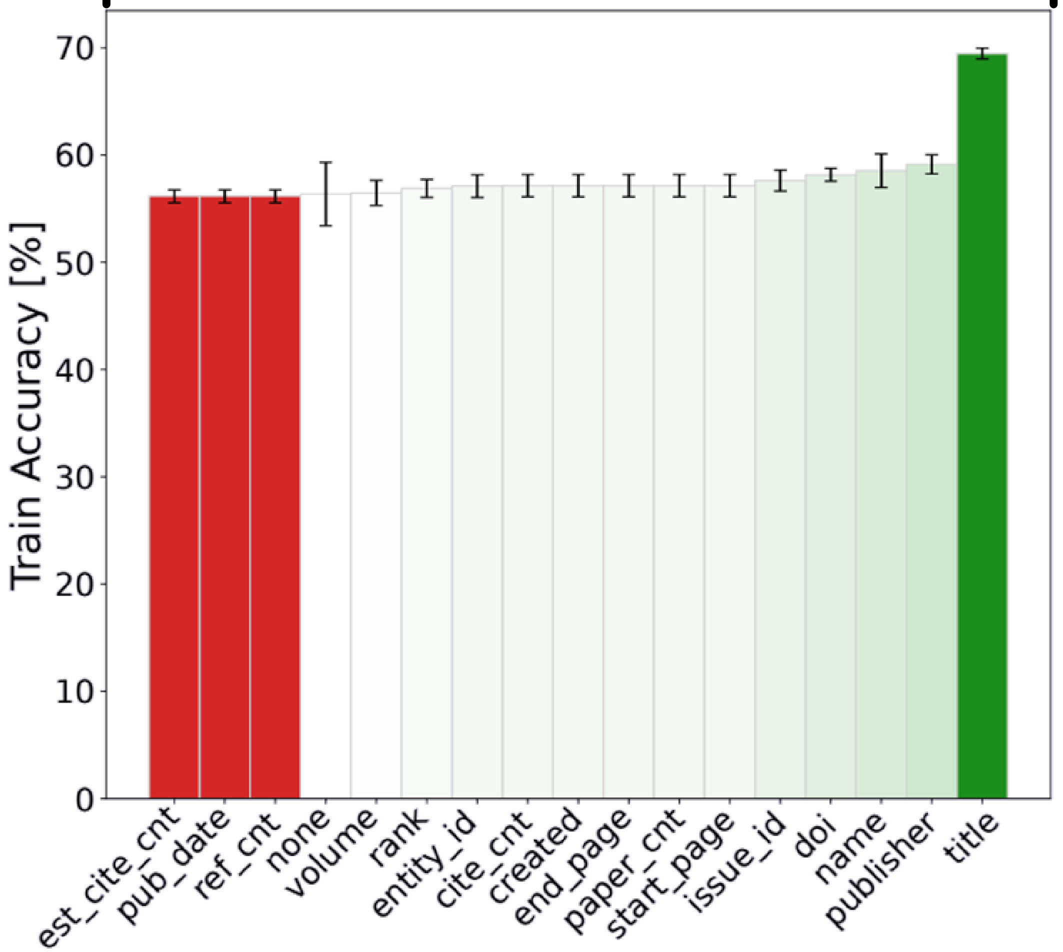


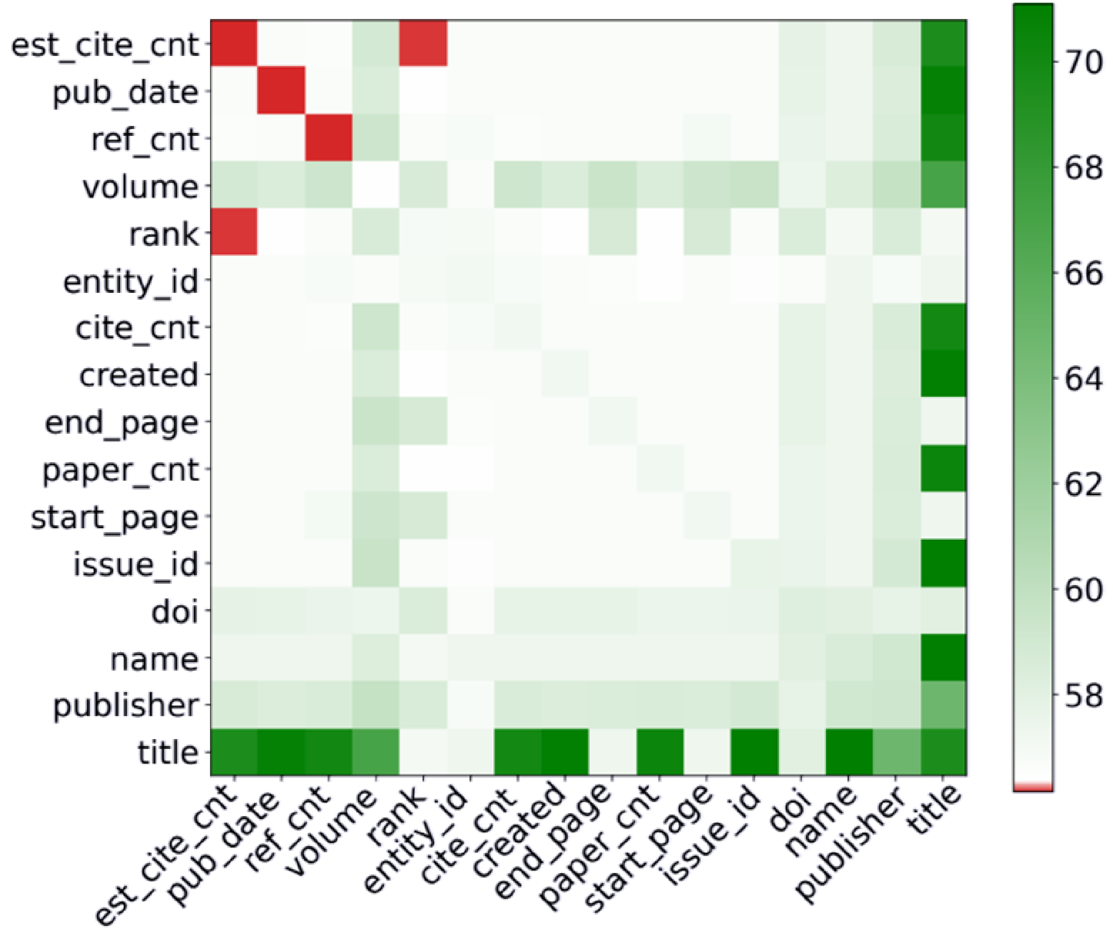


Node Classification, aka Label Prediction

Task: predict the research area of the publication

Impact from each label/property

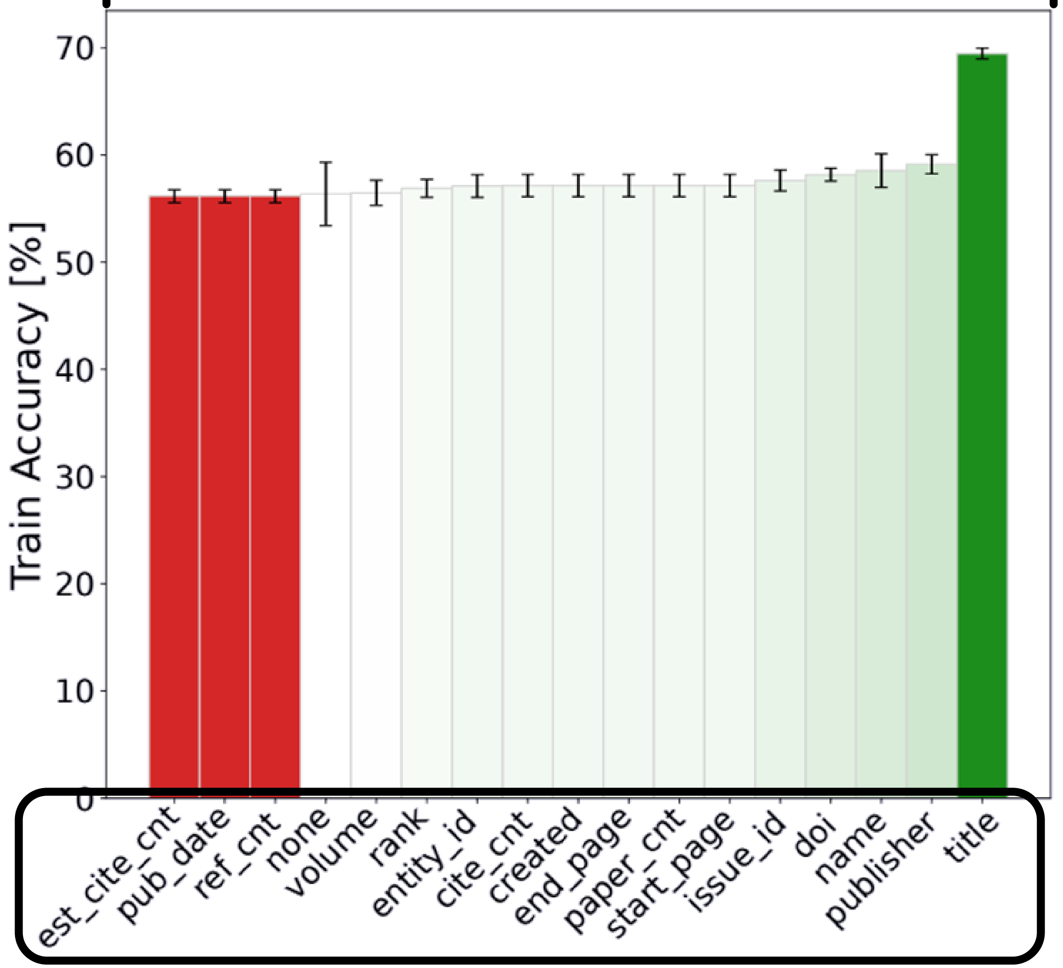


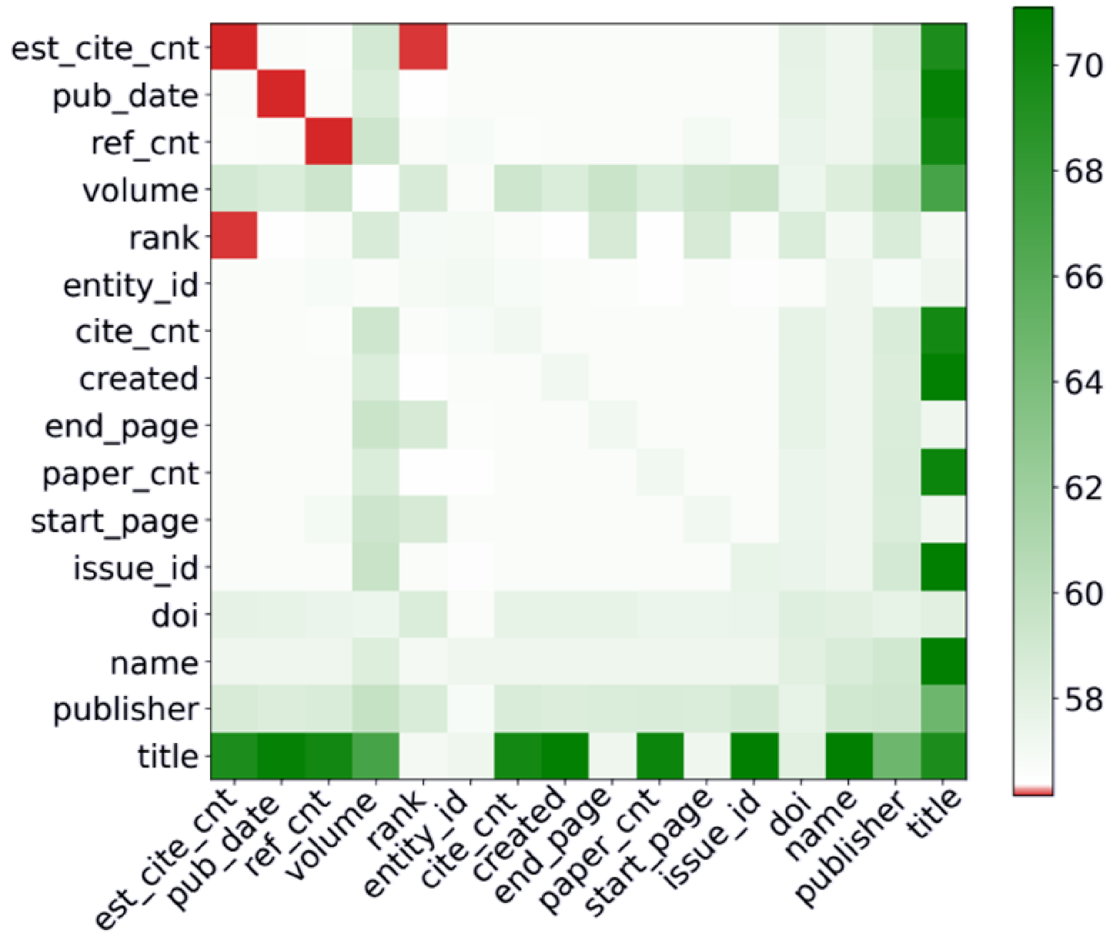


Node Classification, aka Label Prediction

Task: predict the research area of the publication

Impact from each label/property

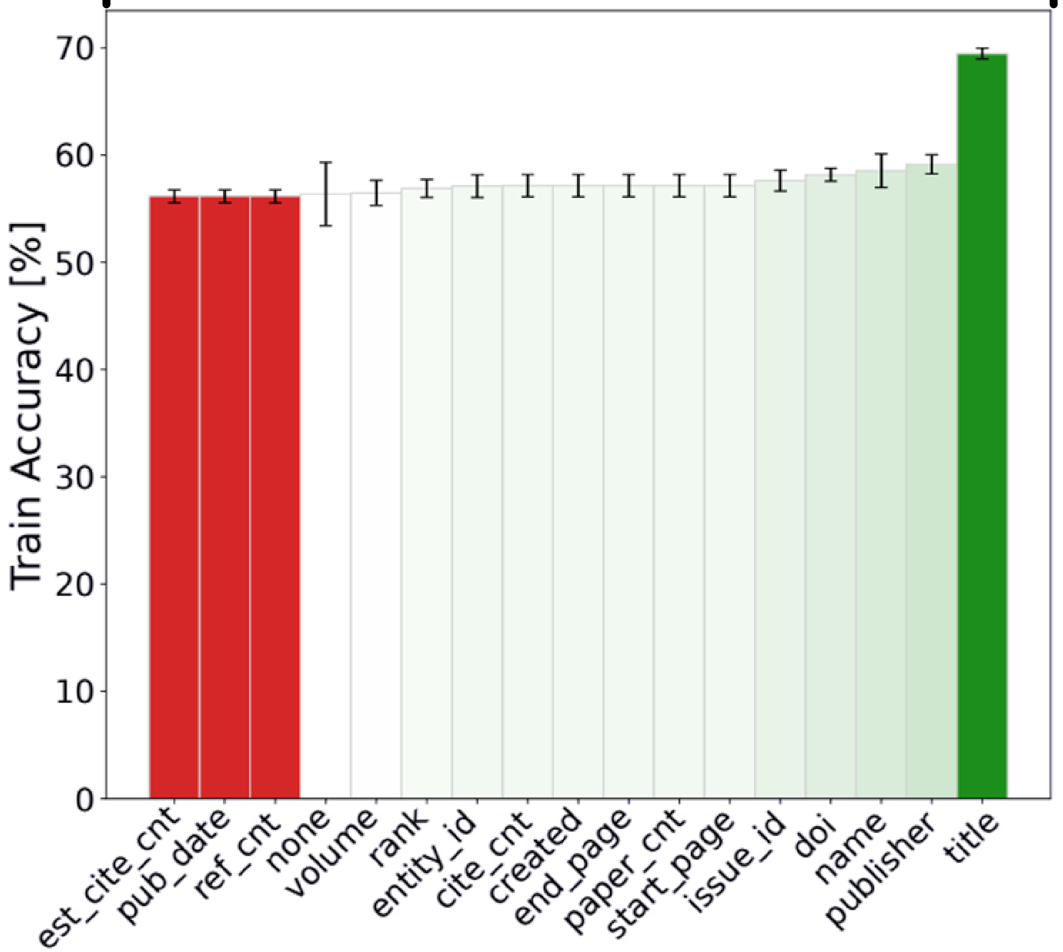


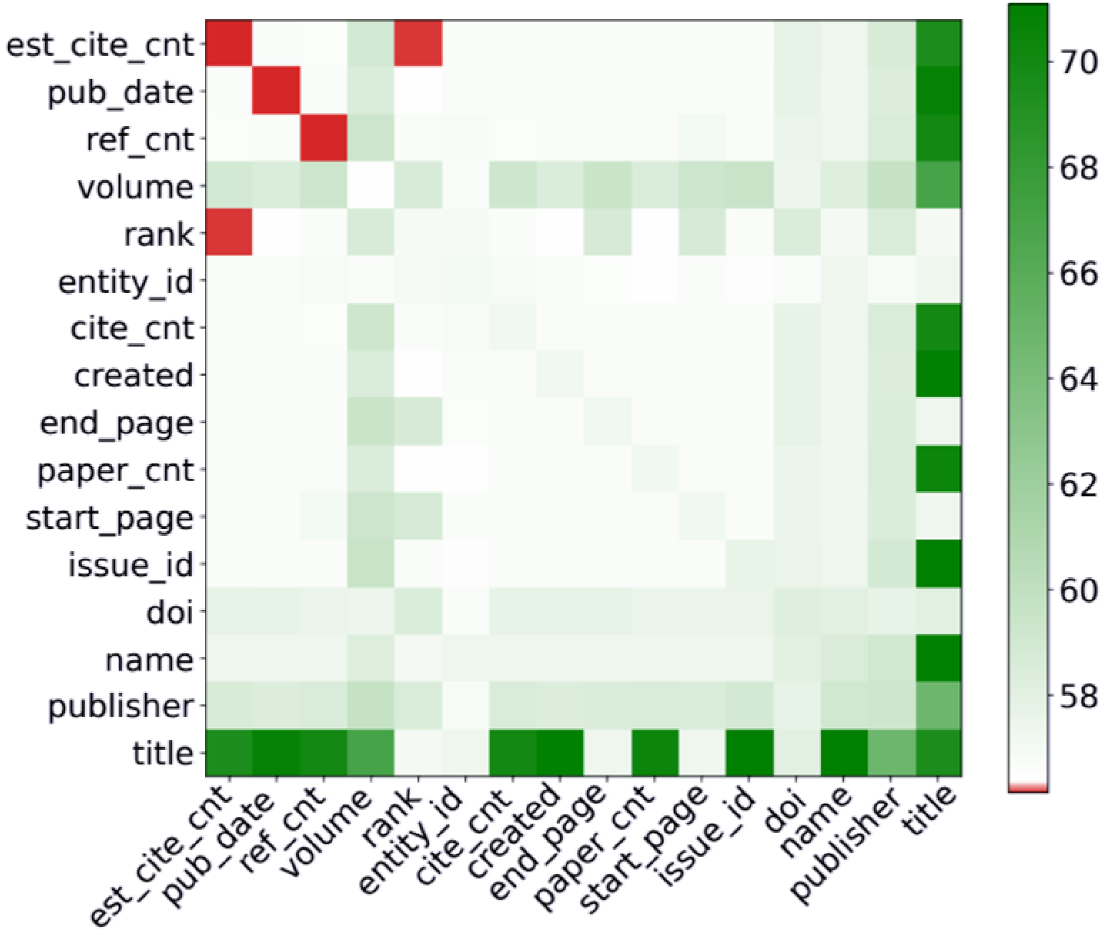


Node Classification, aka Label Prediction

Task: predict the research area of the publication

Impact from each label/property

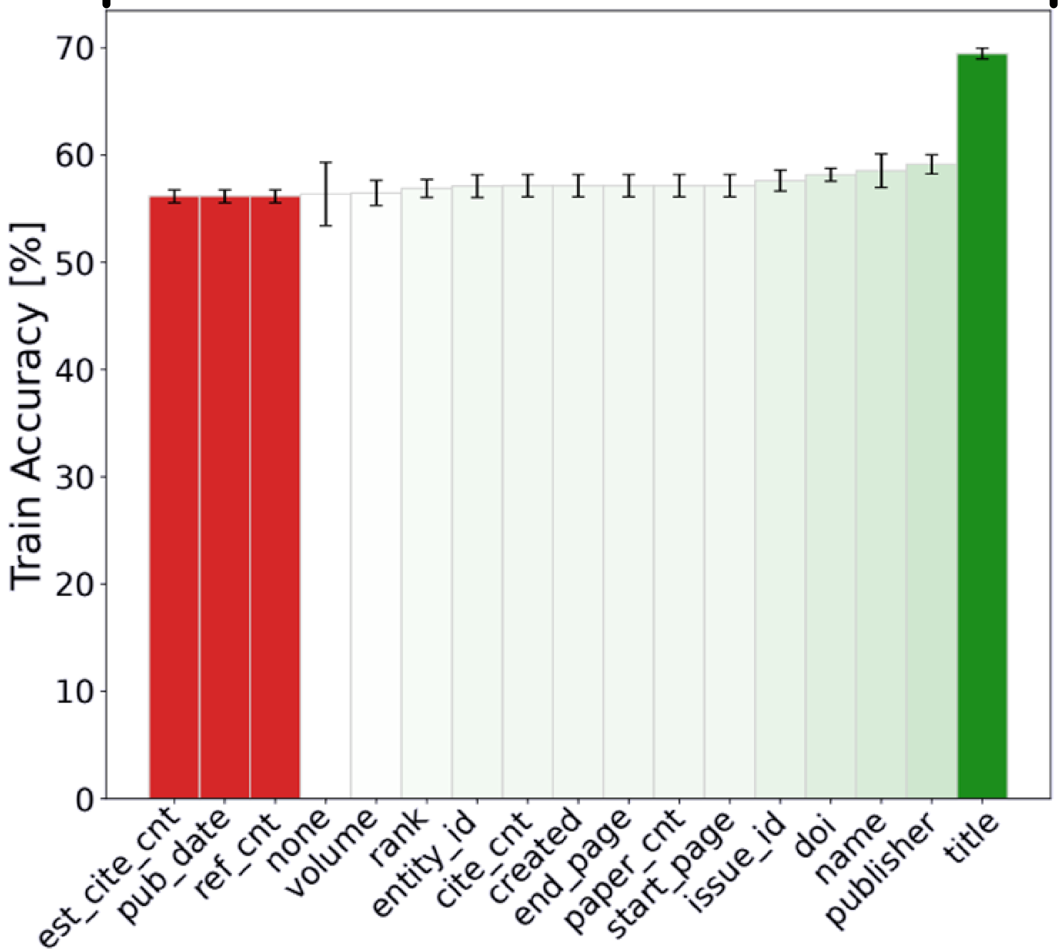




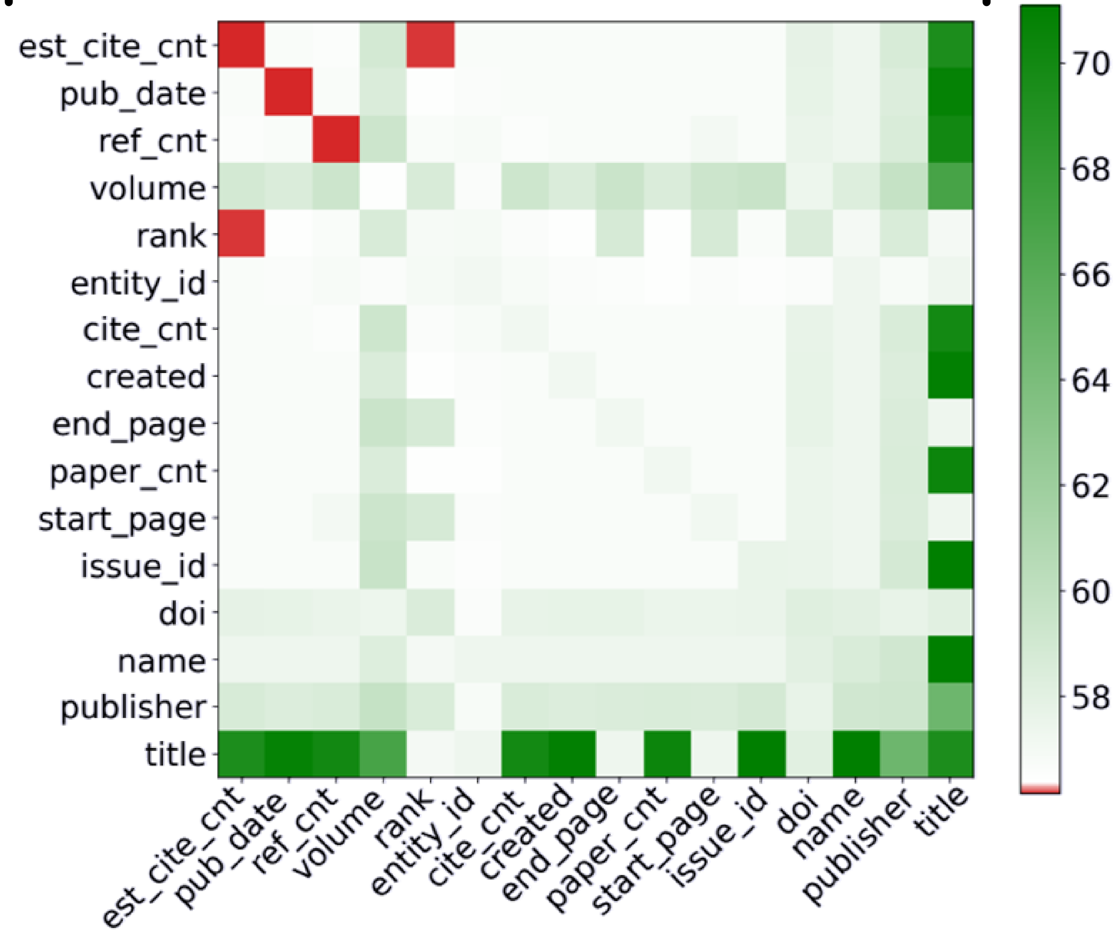
Node Classification, aka Label Prediction

Task: predict the research area of the publication

Impact from each label/property



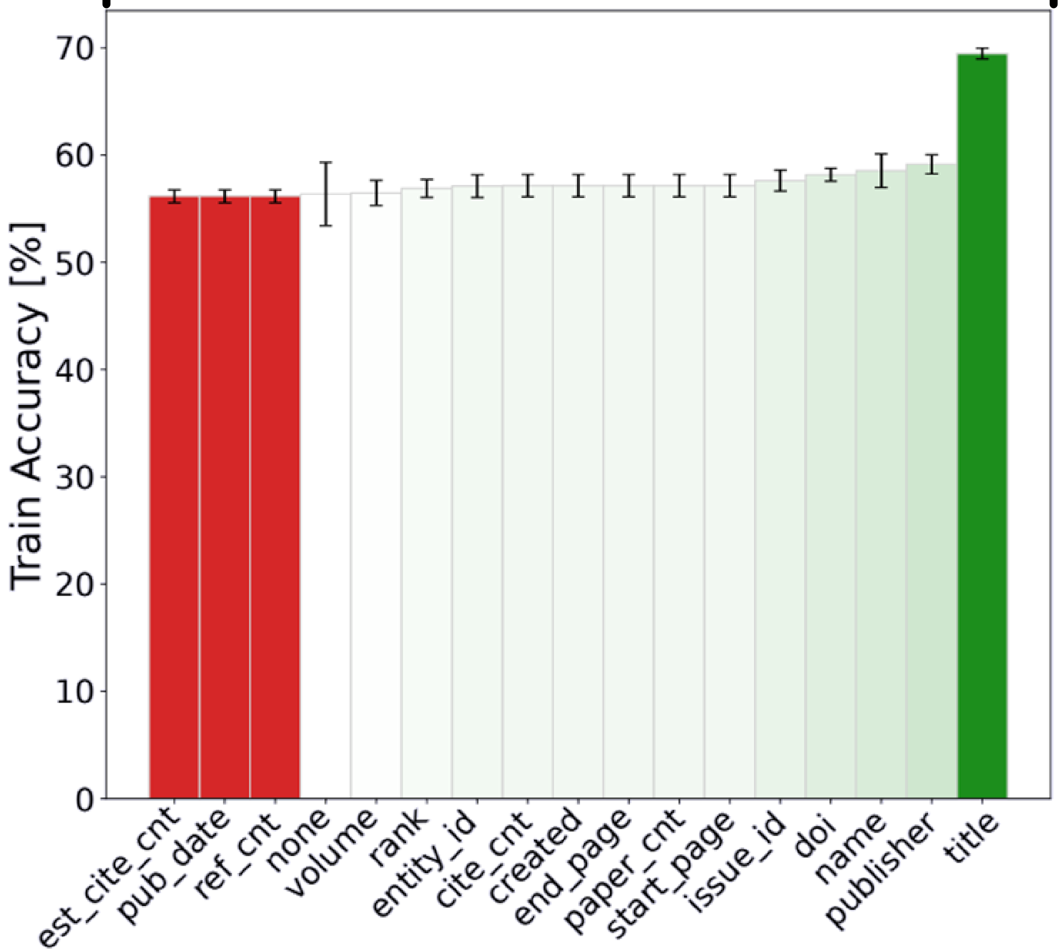
Impact from each label/property **pair**



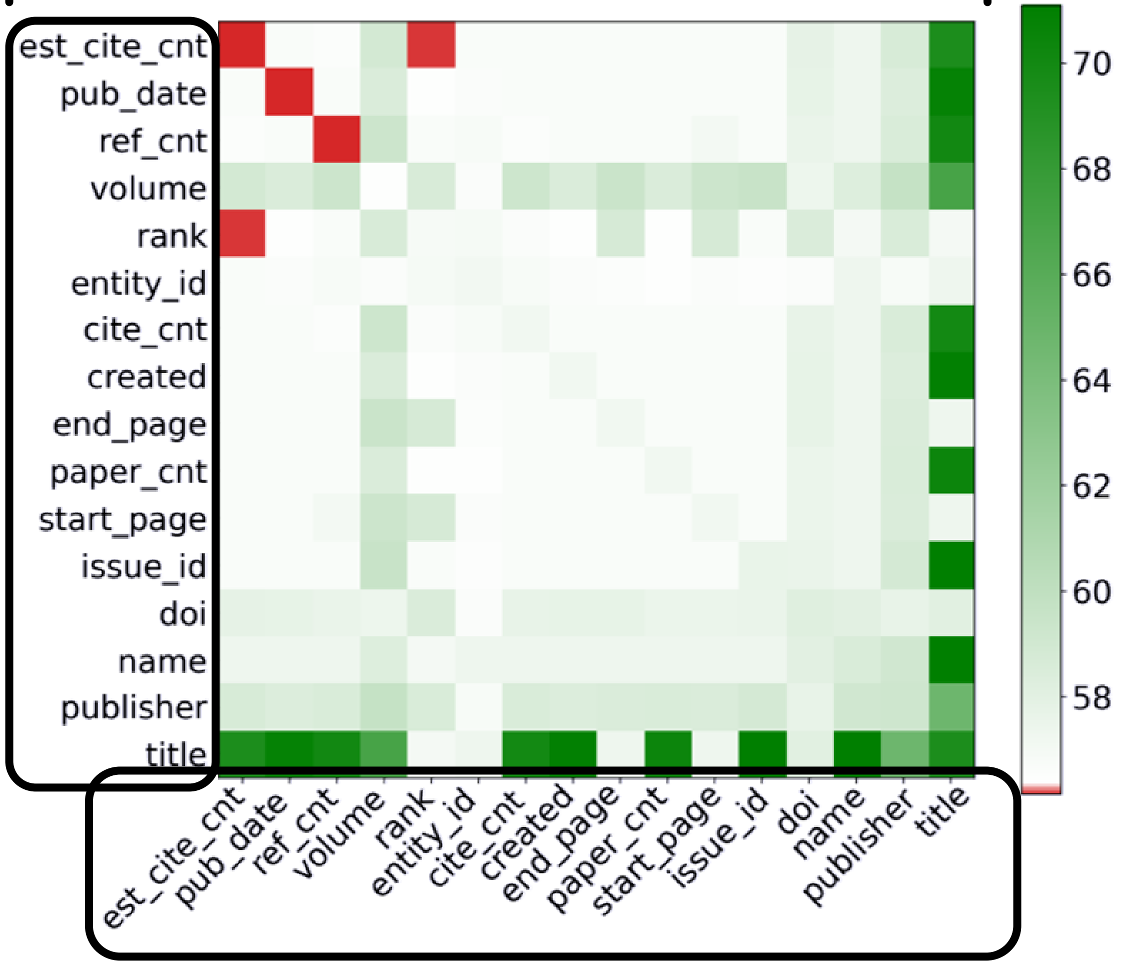
Node Classification, aka Label Prediction

Task: predict the research area of the publication

Impact from each label/property



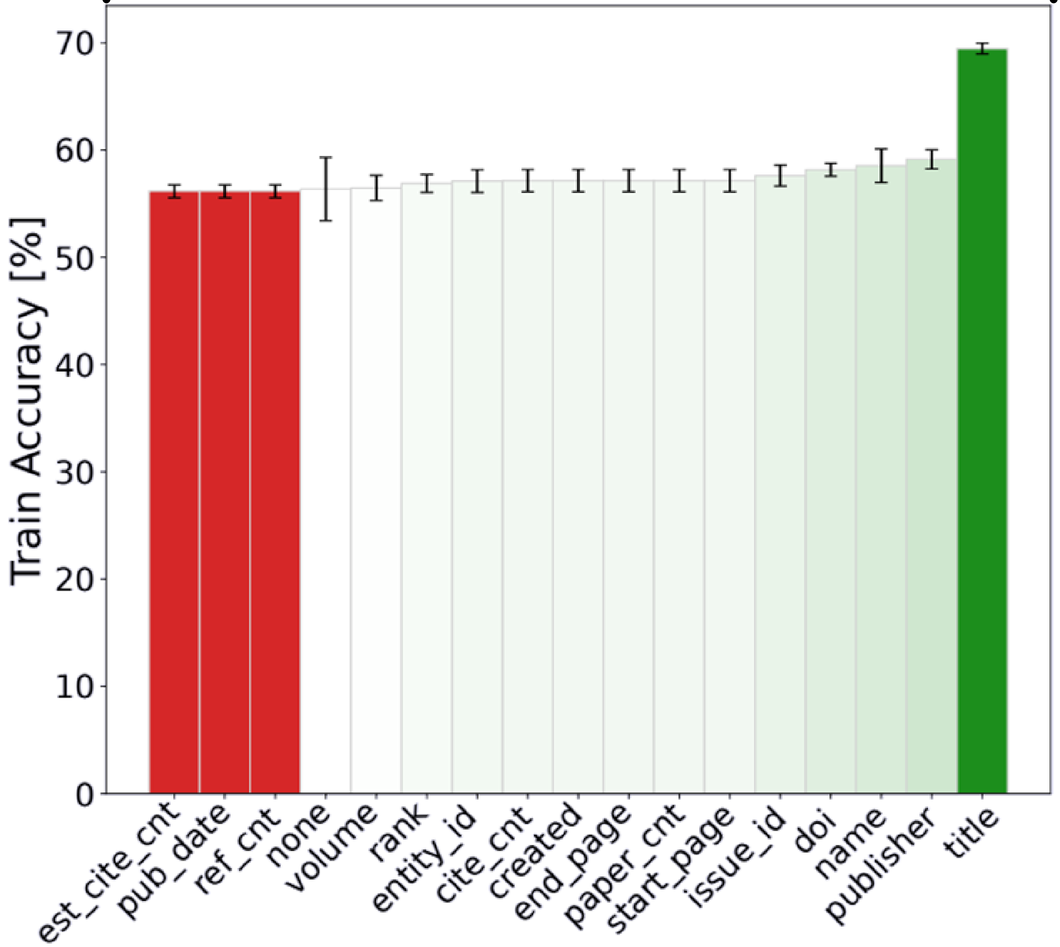
Impact from each label/property **pair**



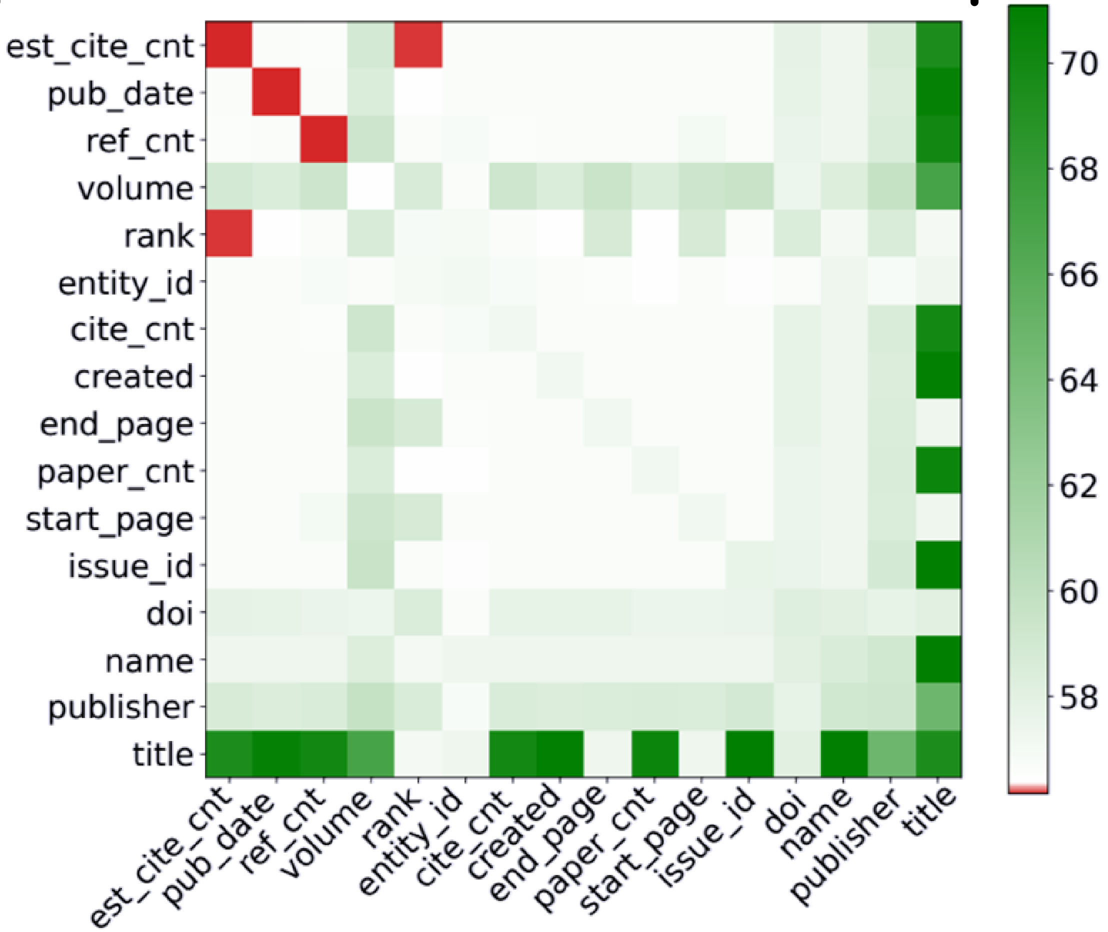
Node Classification, aka Label Prediction

Task: predict the research area of the publication

Impact from each label/property



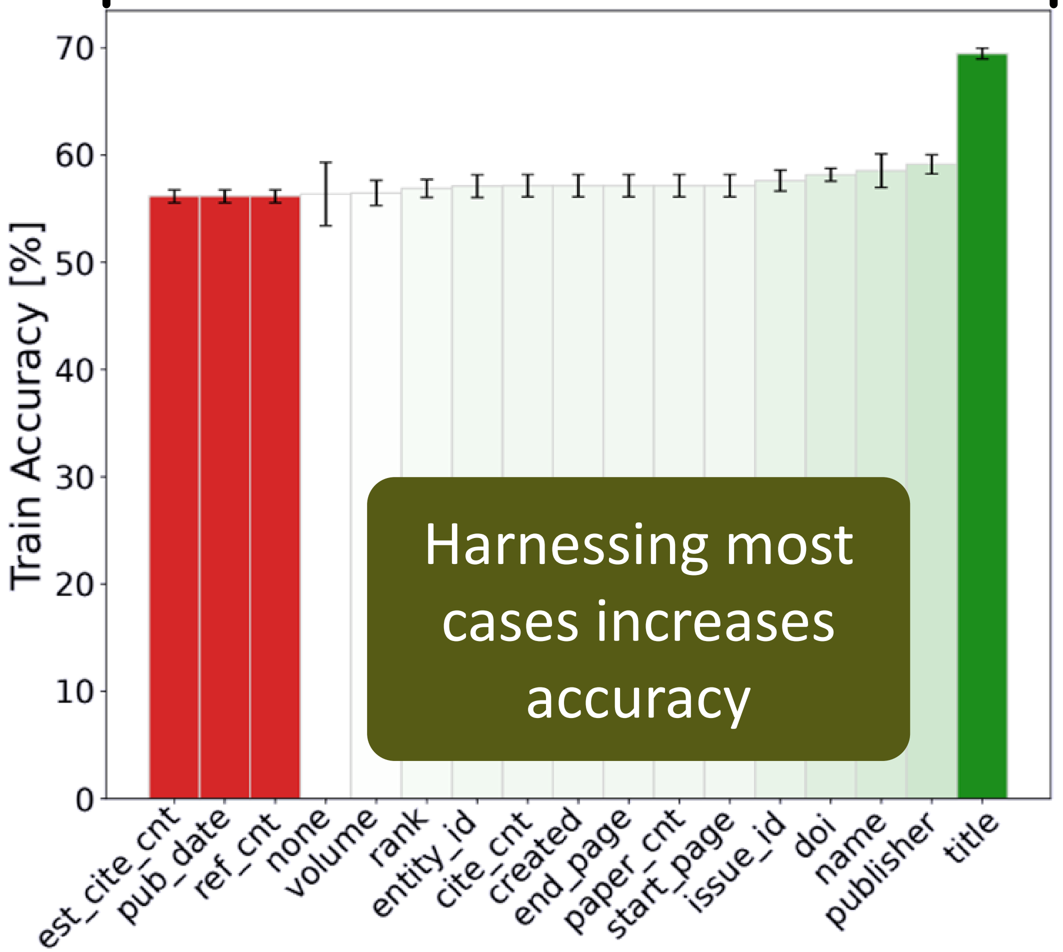
Impact from each label/property **pair**



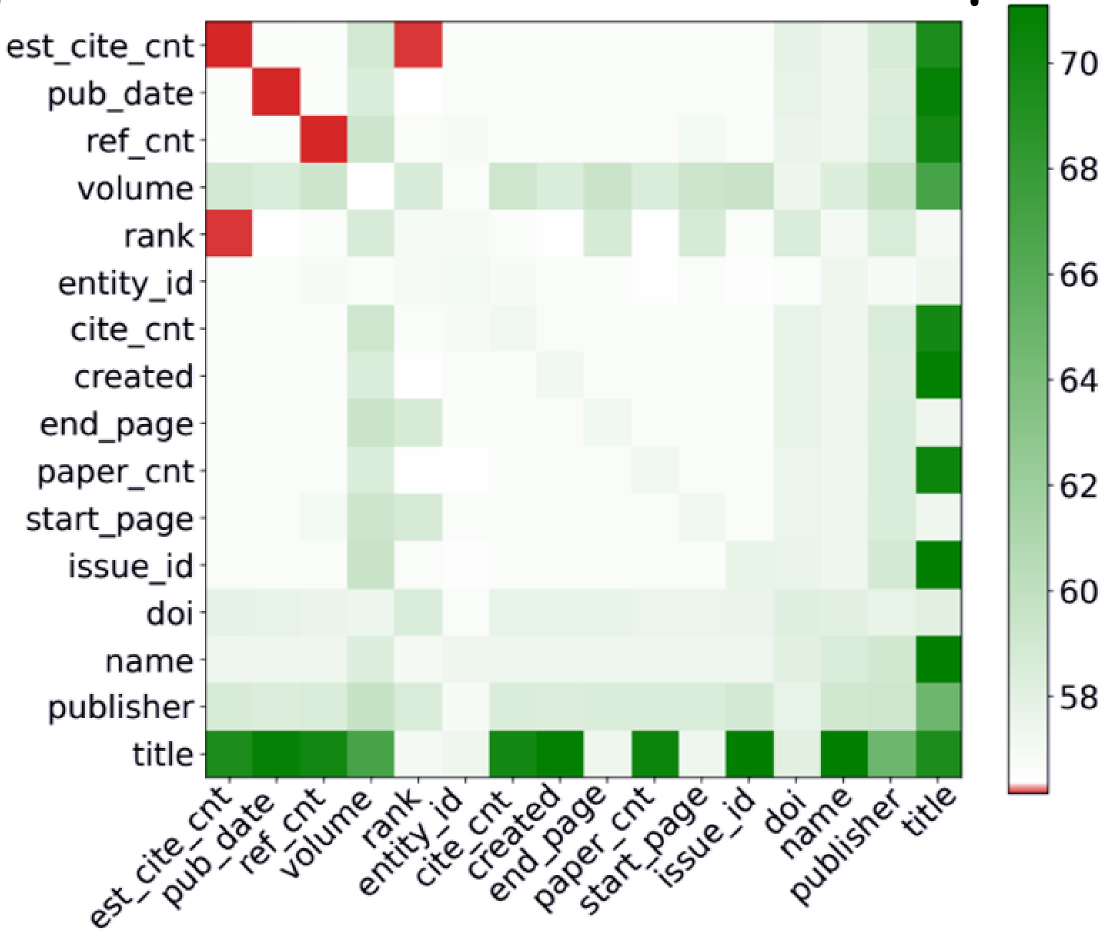
Node Classification, aka Label Prediction

Task: predict the research area of the publication

Impact from each label/property



Impact from each label/property **pair**

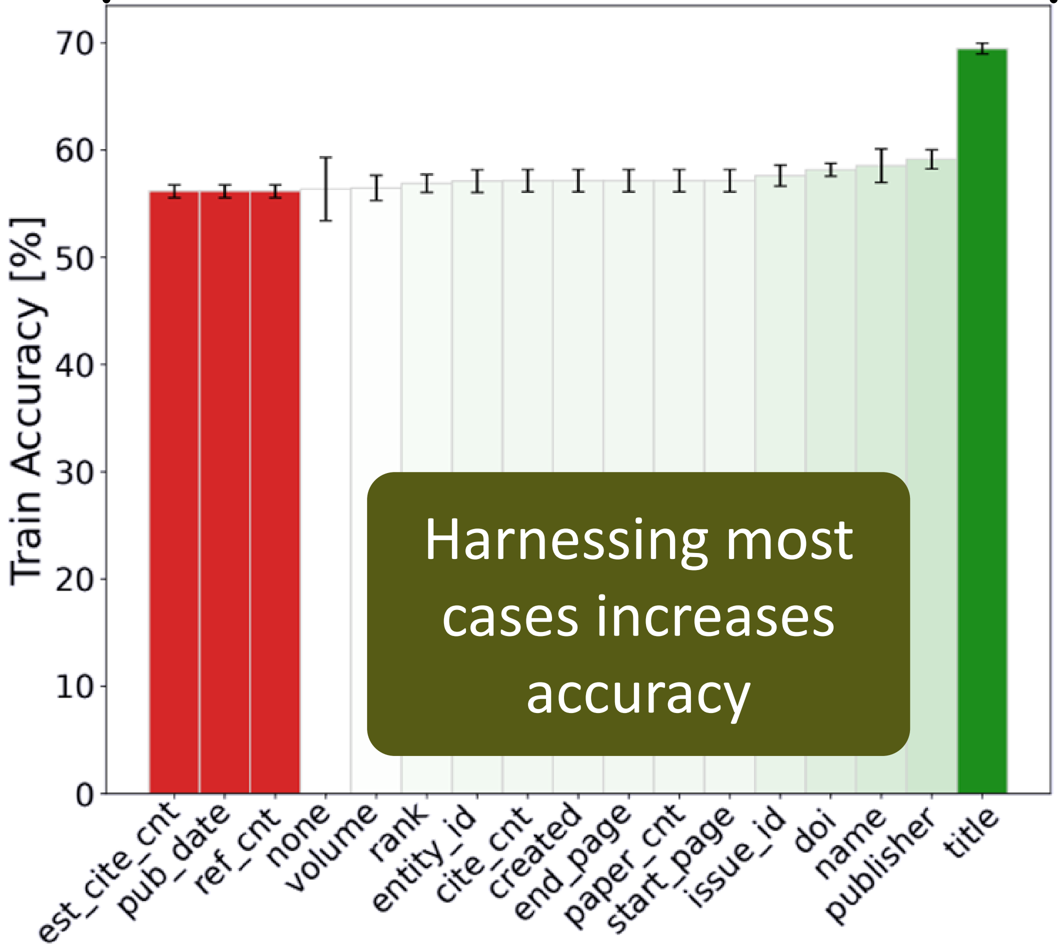


Node Classification, aka Label Prediction

Task: predict the research area of the publication

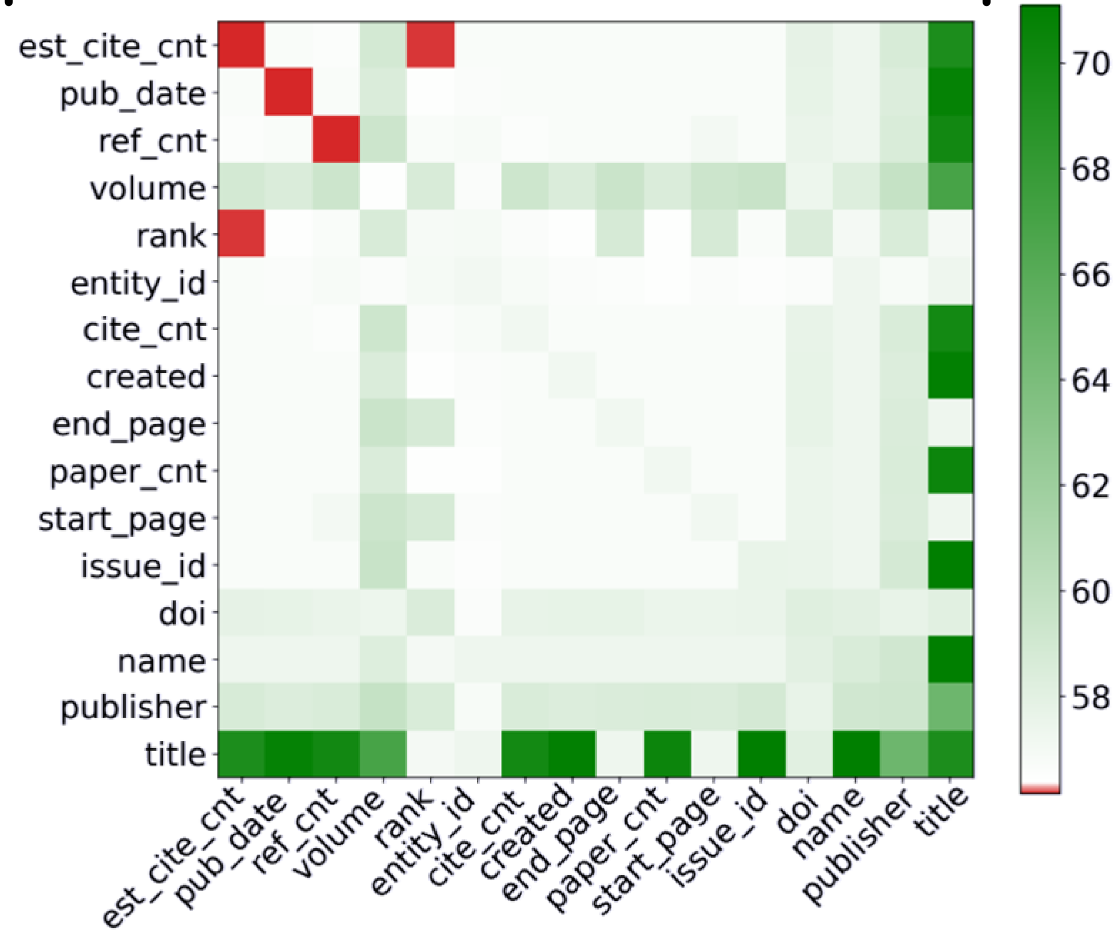
In same cases, accuracy decreases

Impact from each label/property



Harnessing most cases increases accuracy

Impact from each label/property **pair**



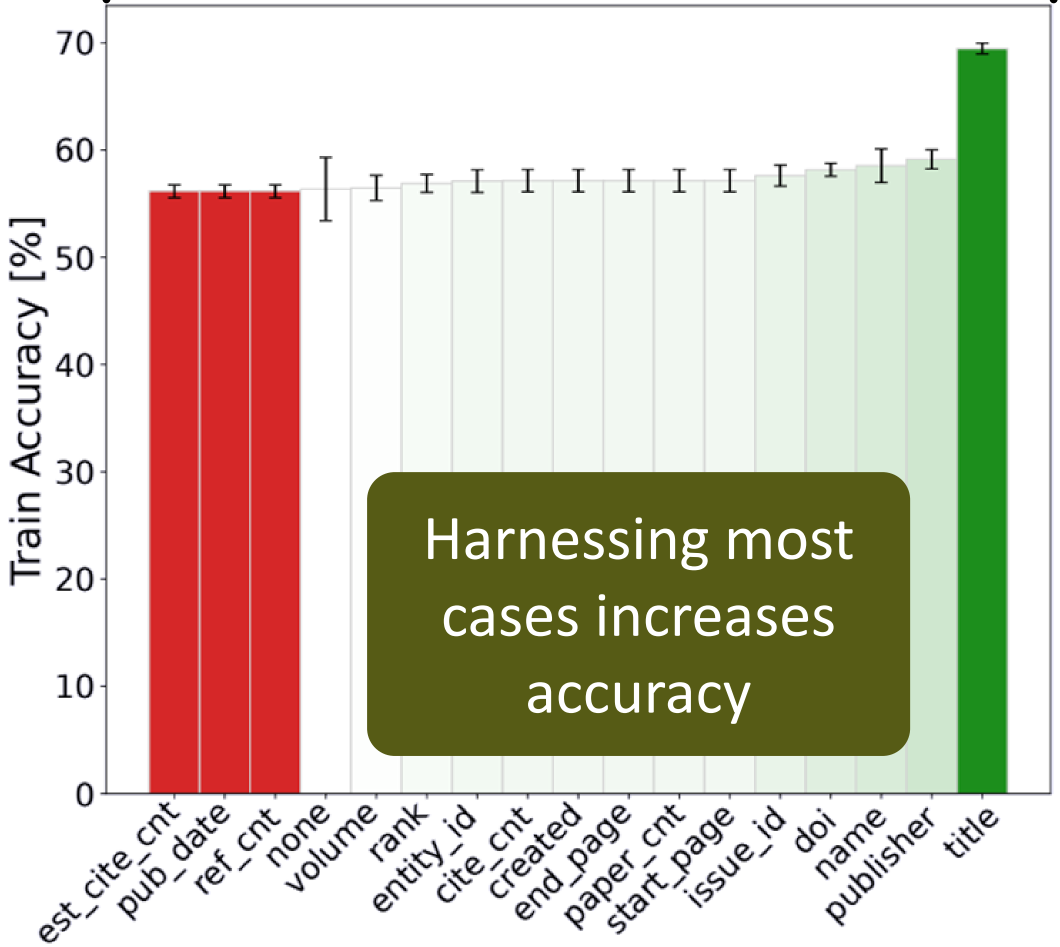
Node Classification, aka Label Prediction

Task: predict the research area of the publication

In same cases, accuracy decreases

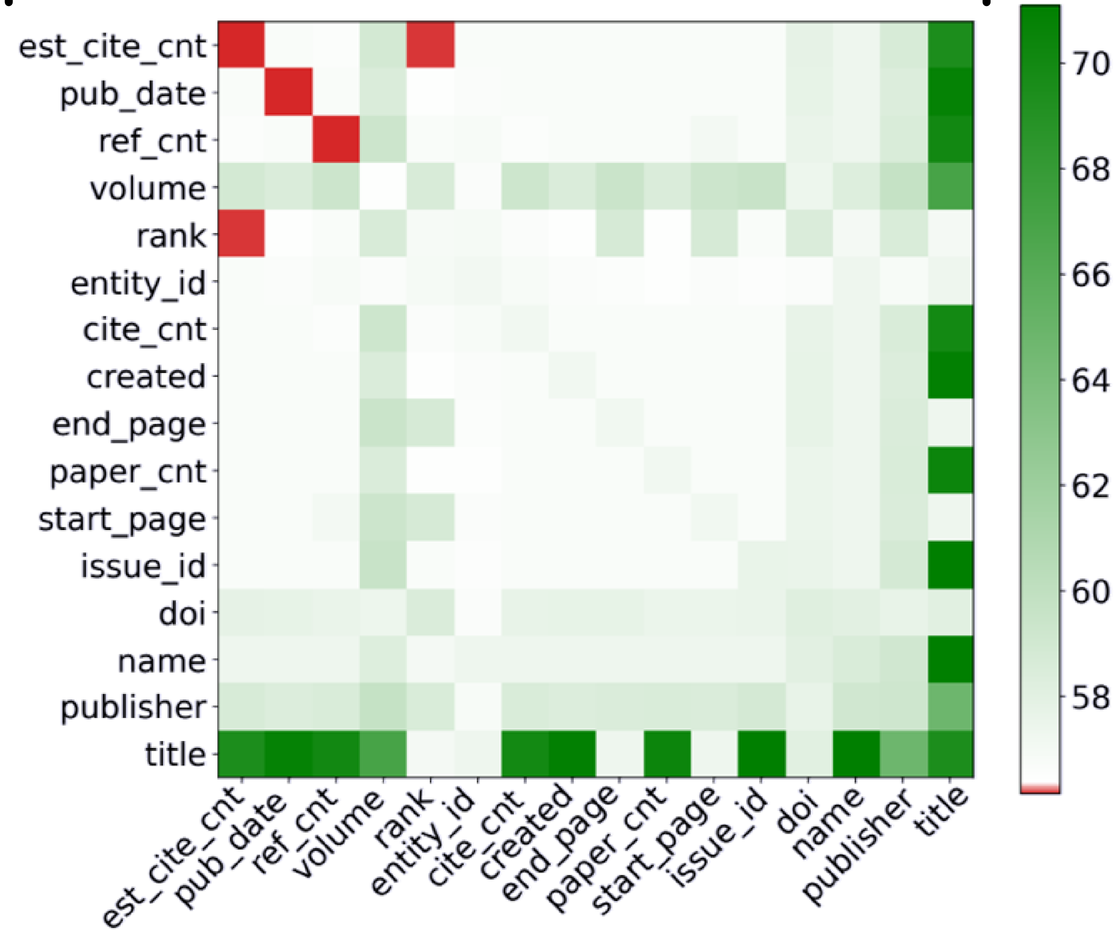
It is important to understand the data well and select the right encoded LPG information

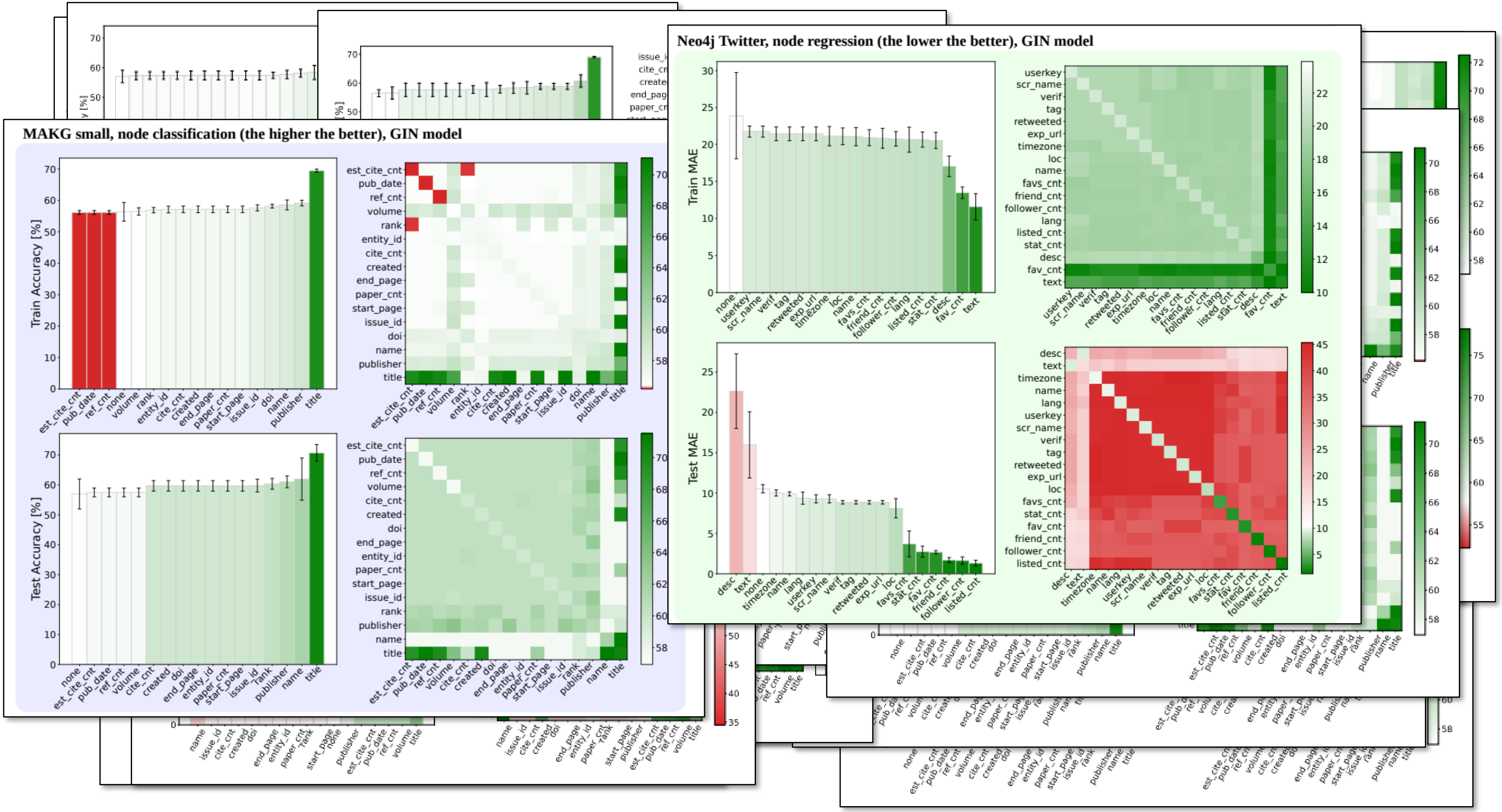
Impact from each label/property

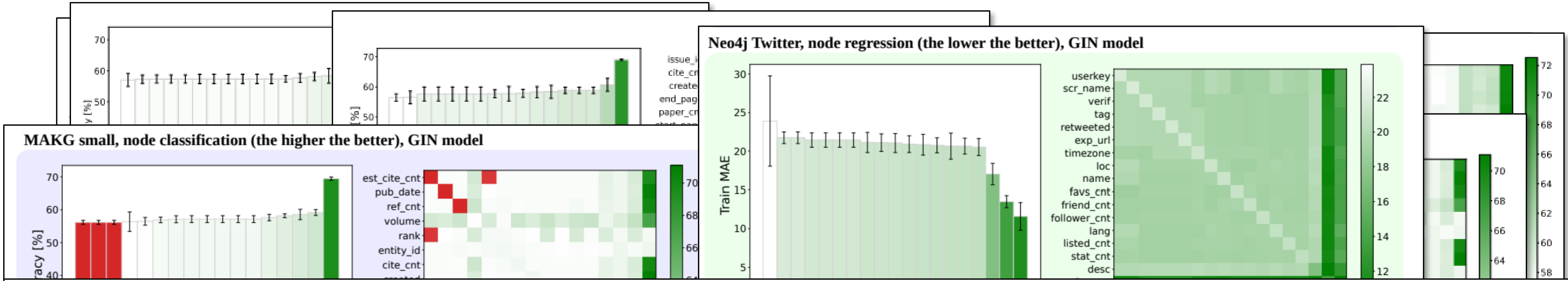


Harnessing most cases increases accuracy

Impact from each label/property **pair**




MAKG small, node classification (the higher the better), GIN model
Train Accuracy [%]
Test Accuracy [%]
est_cite_cnt
pub_date
ref_cnt
none
volume
rank
entity_id
cite_cnt
created
end_page
paper_cnt
start_page
issue_id
doi
name
publisher
title
Neo4j Twitter, node regression (the lower the better), GIN model
Train MAE
Test MAE
userkey
scr_name
verif
tag
retweeted
exp_url
timezone
loc
favs_cnt
friend_cnt
follower_cnt
lang
listed_cnt
stat_cnt
desc
fav_cnt
text



https://arxiv.org/abs/2209.09732 @ LoG'22 (Learning on Graphs'22)

Neural Graph Databases

Maciej Besta[1,†] Patrick Iff[1] Florian Scheidl[1] Kazuki Osawa[1] Nikoli Dryden[1]
Michal Podstawski[2,3] Tiancheng Chen[1] Torsten Hoefler[1,†]

[1]Department of Computer Science, ETH Zurich
[2]Warsaw University of Technology, Warsaw, Poland

Evaluation: Used Machine & Objectives

CSCS Cray Piz Daint & Ault

64GB – 2TB memory per server

Evaluation: Used Machine & Objectives

CSCS Cray Piz Daint & Ault
64GB – 2TB memory per server

How to scale these computations to really big graphs (hundreds of billions of edges) using a lot of parallelism (>a hundred thousand cores)?

Evaluation: Used Machine & Objectives

CSCS Cray Piz Daint & Ault
64GB – 2TB memory per server

How to scale these computations to really big graphs (hundreds of billions of edges) using a lot of parallelism (>a hundred thousand cores)?

https://arxiv.org/abs/2305.11162 **@ ACM/IEEE Supercomputing'23, Best Paper Finalist**

High-Performance Graph Databases That Are Portable, Programmable, and Scale to Hundreds of Thousands of Cores

Maciej Besta[1*#], Robert Gerstenberger[1*#], Marc Fischer[2], Michał Podstawski[3,4], Jürgen Müller[5], Nils Blach[1], Berke Egeli[1], George Mitenkov[1], Wojciech Chlapek[6], Marek Michalewicz[7], Torsten Hoefler[1*]
[1]ETH Zurich; [2]PRODYNA (Schweiz) AG; [3]Warsaw University of Technology; [4]TCL Research Europe; [5]BASF SE; [6]ICM UW; [7]Sano Centre for Computational Medicine; [*]Corresponding authors, [#]alphabetical order

Conclusions

Thank you

Want to know more?

youtube.com/@spcl

twitter.com/spcl_eth

spcl.inf.ethz.ch

github.com/spcl

Conclusions

LPG2vec enables encoding arbitrary LPG datasets and their seamless analysis within an arbitrary GNN processing pipeline

Thank you

Want to know more?

youtube.com/@spcl

twitter.com/spcl_eth

spcl.inf.ethz.ch

github.com/spcl

Conclusions

LPG2vec enables encoding arbitrary LPG datasets and their seamless analysis within an arbitrary GNN processing pipeline

This introduces & lays the foundation for **Neural Graph Databases**

Thank you

Want to know more?

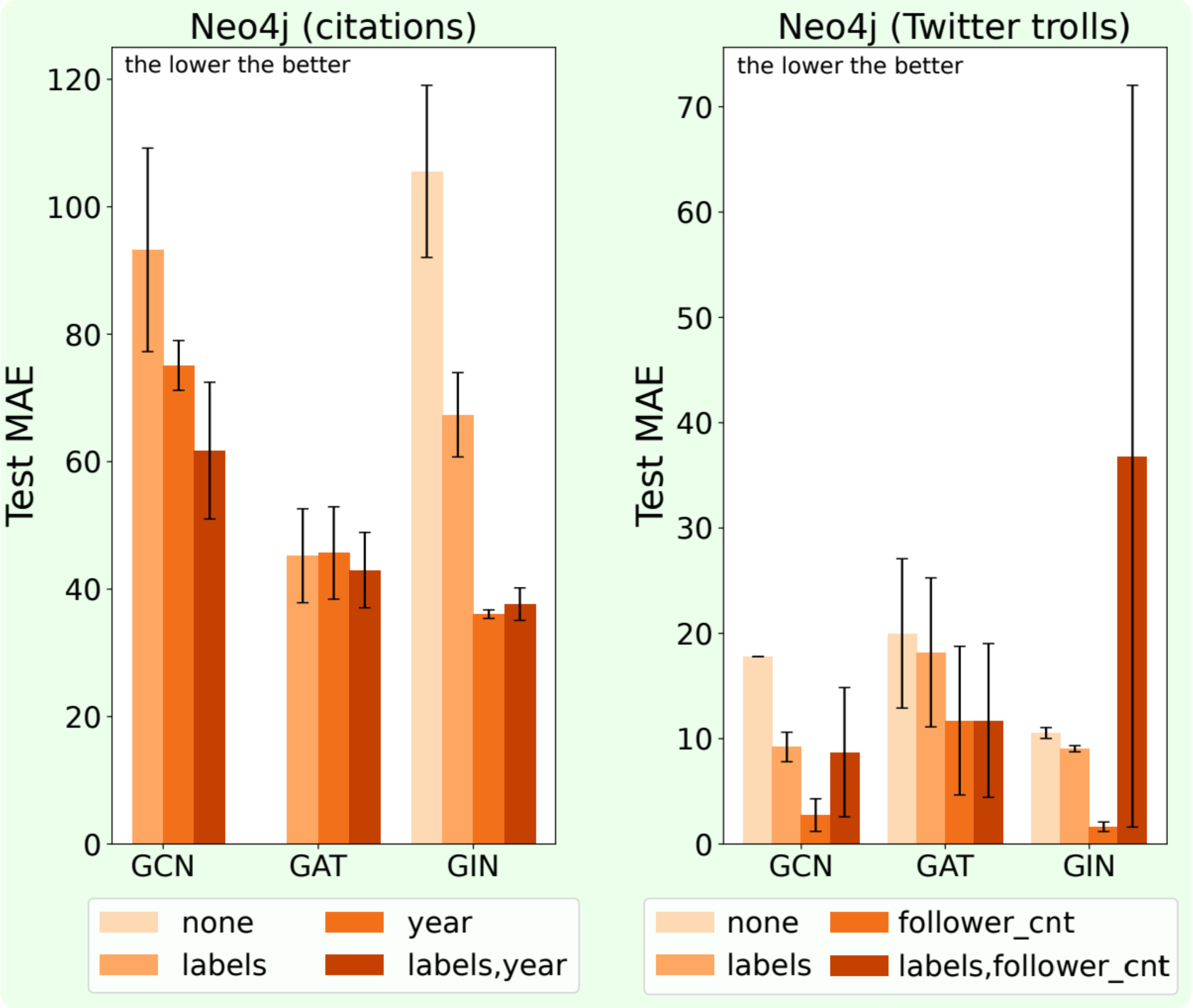
youtube.com/@spcl

twitter.com/spcl_eth

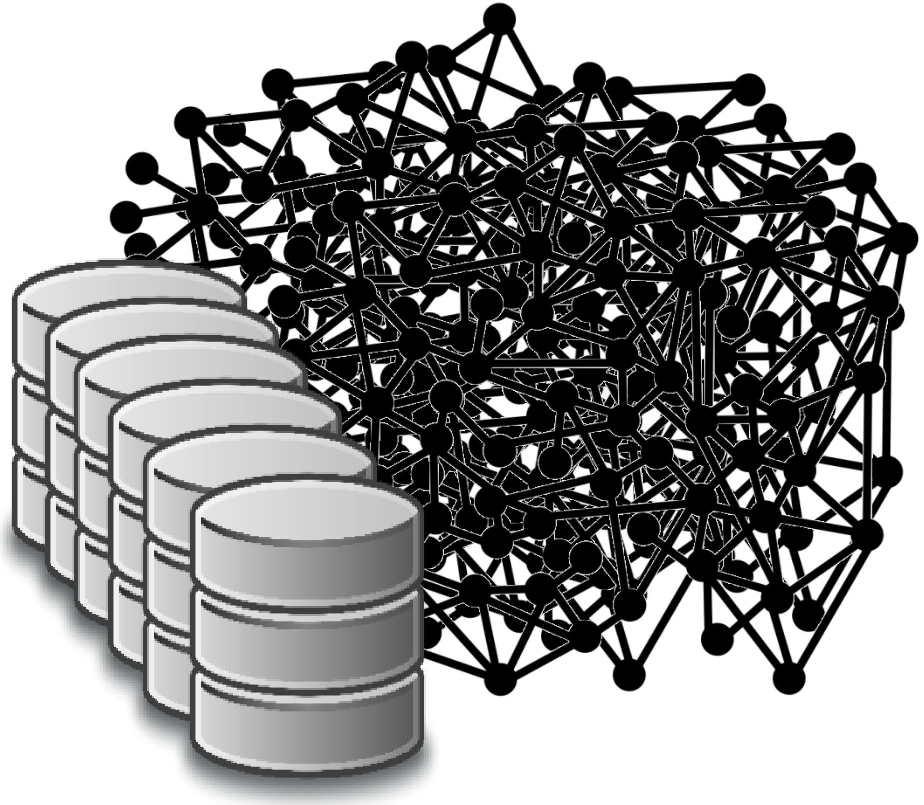
spcl.inf.ethz.ch

github.com/spcl

Node Regression

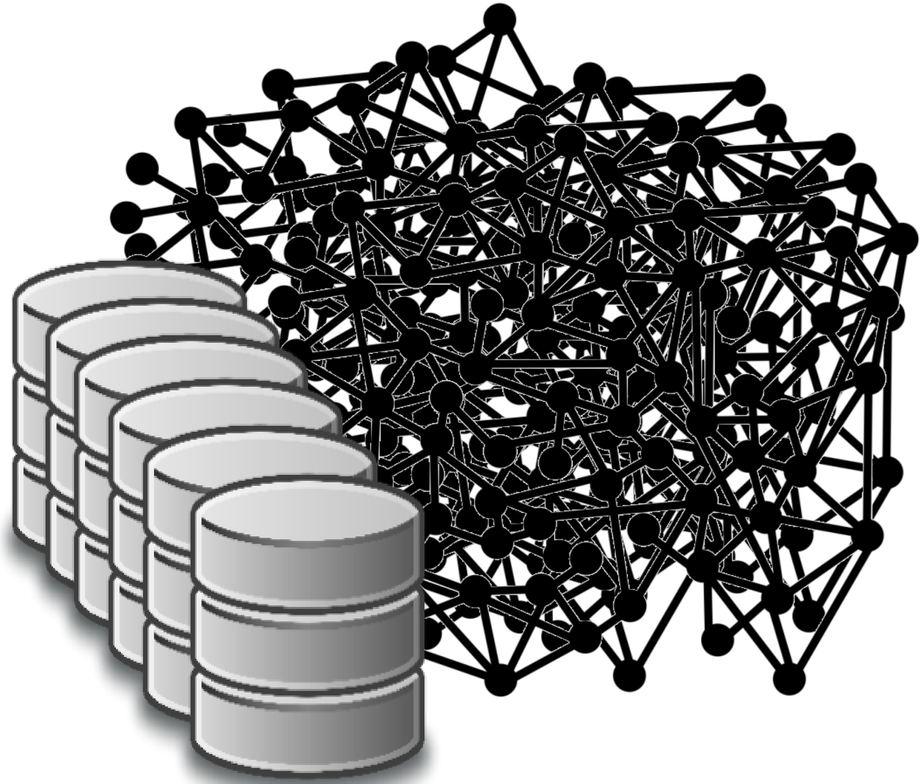


Graph Databases: Major Workloads



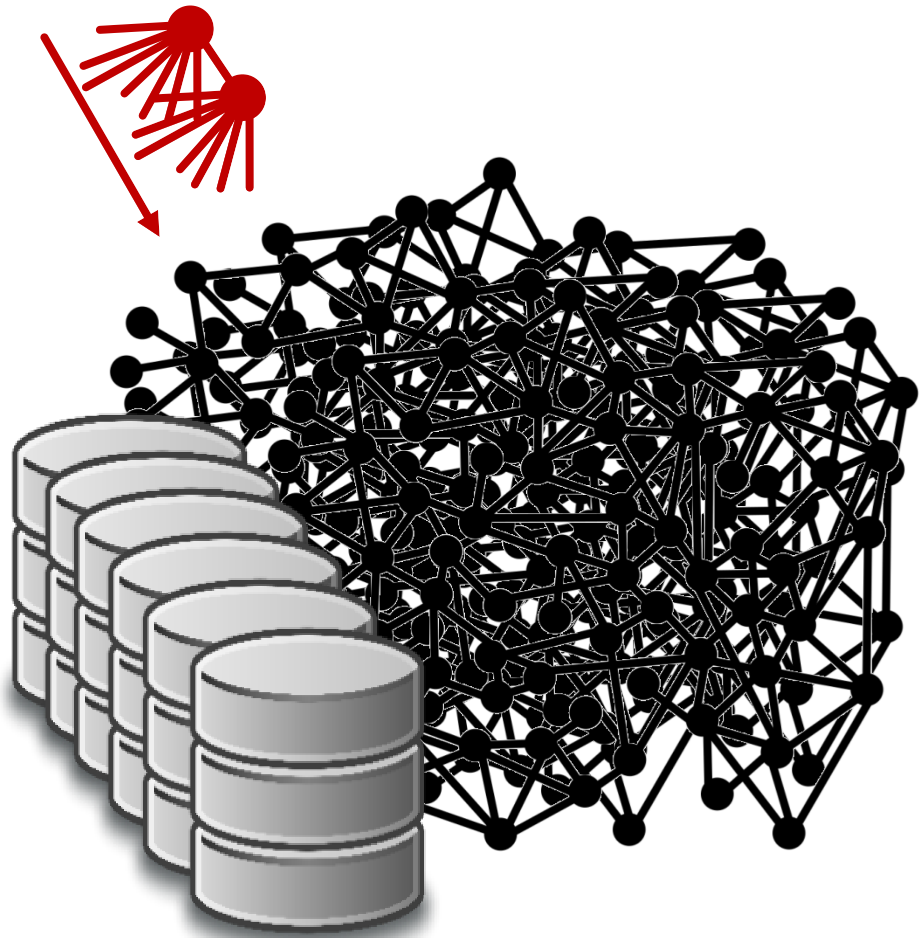
Graph Databases: Major Workloads

Online Transactional Processing **(OLTP)**



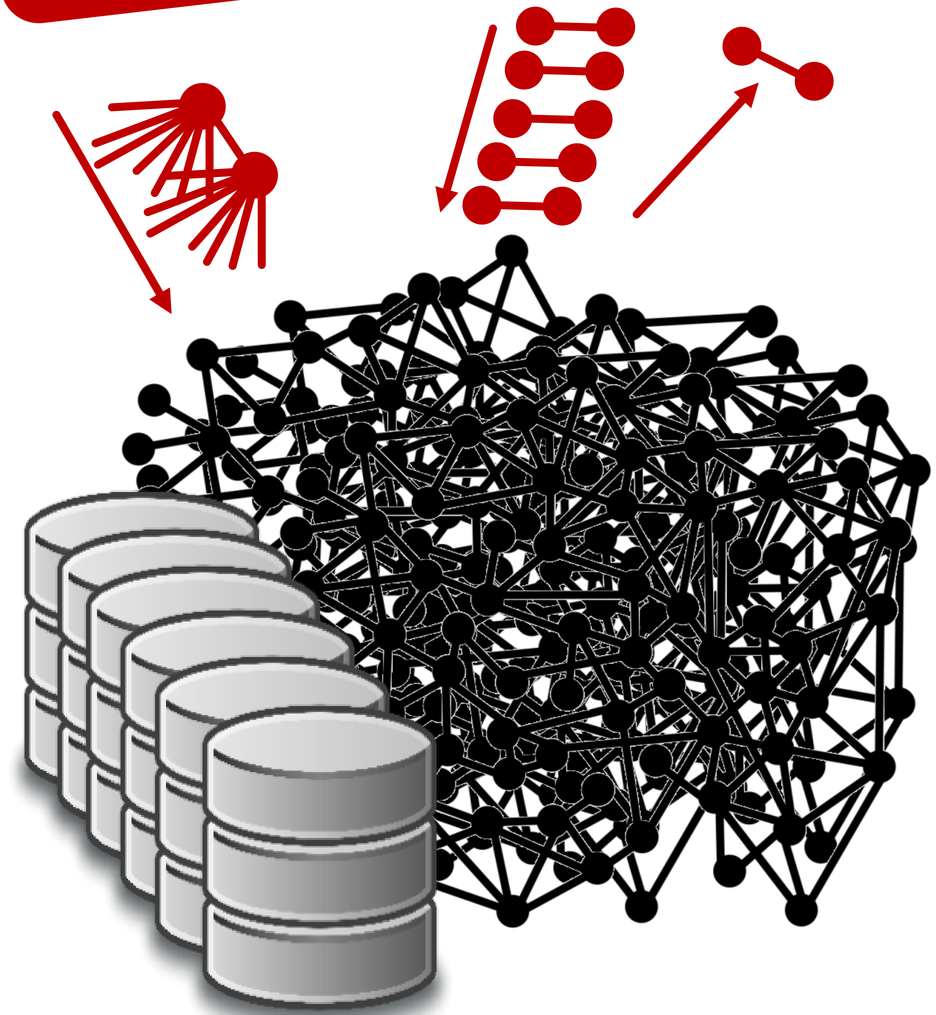
Graph Databases: Major Workloads

Online Transactional Processing **(OLTP)**



Graph Databases: Major Workloads

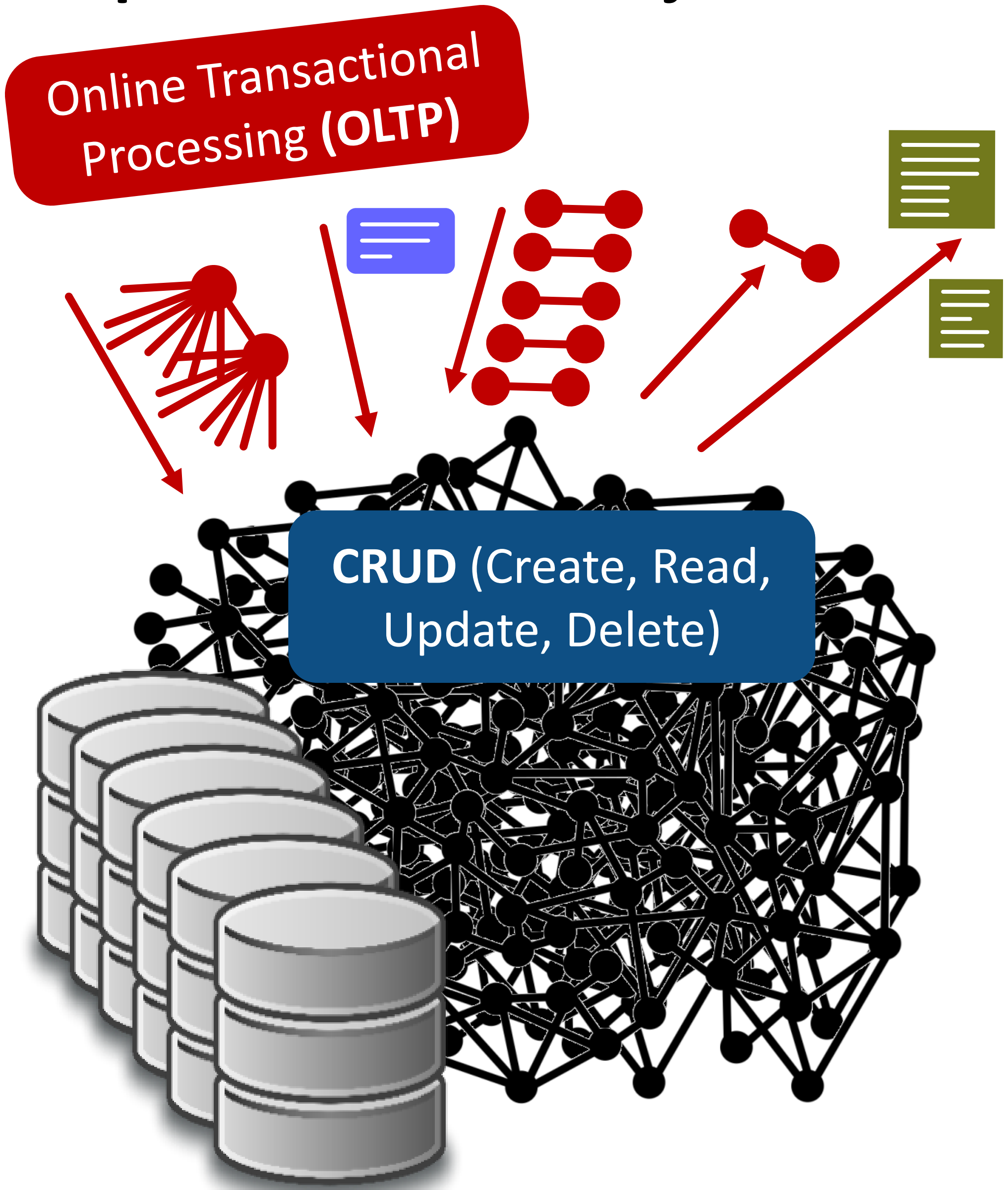
Online Transactional Processing **(OLTP)**



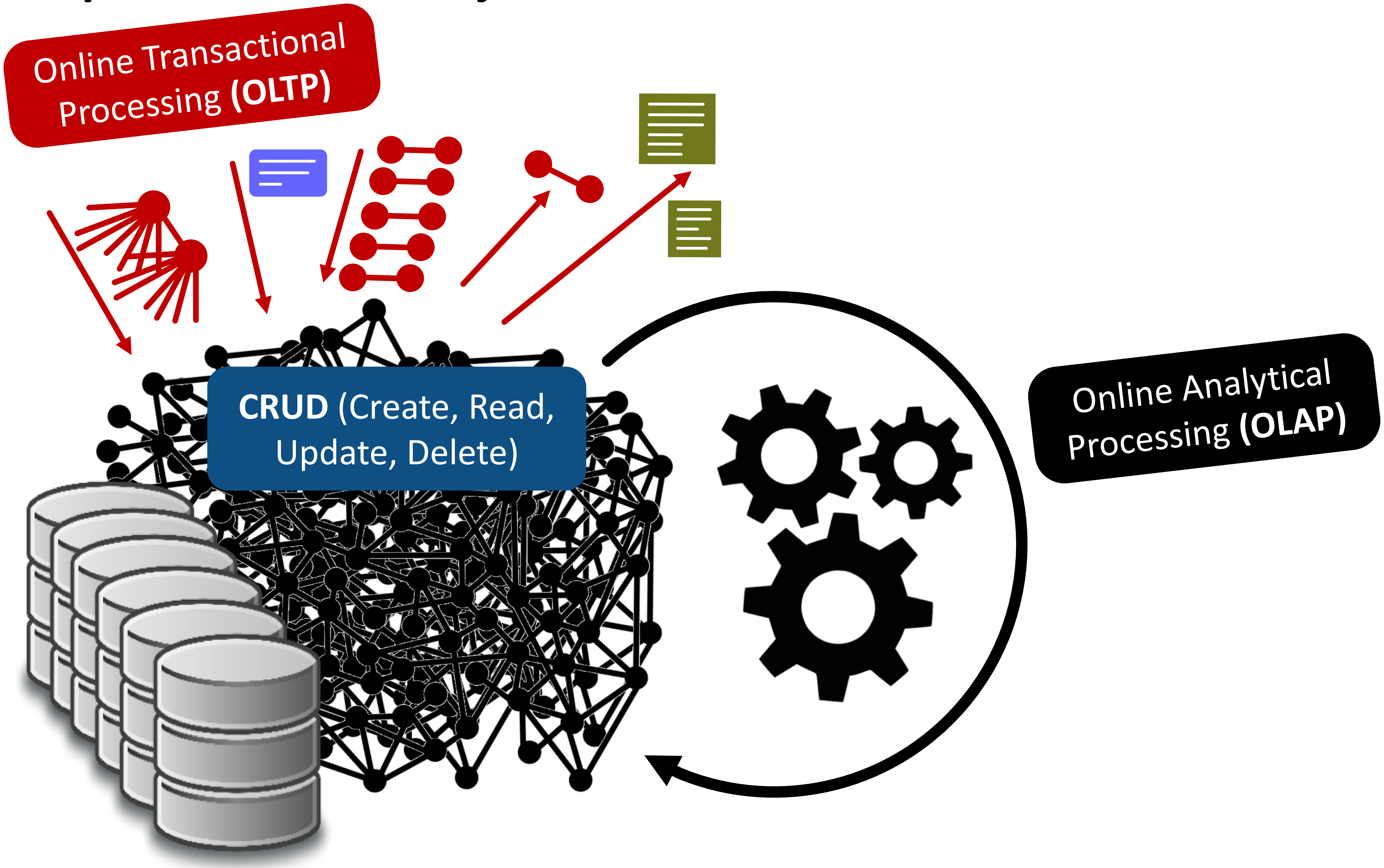
Graph Databases: Major Workloads

Online Transactional Processing **(OLTP)**

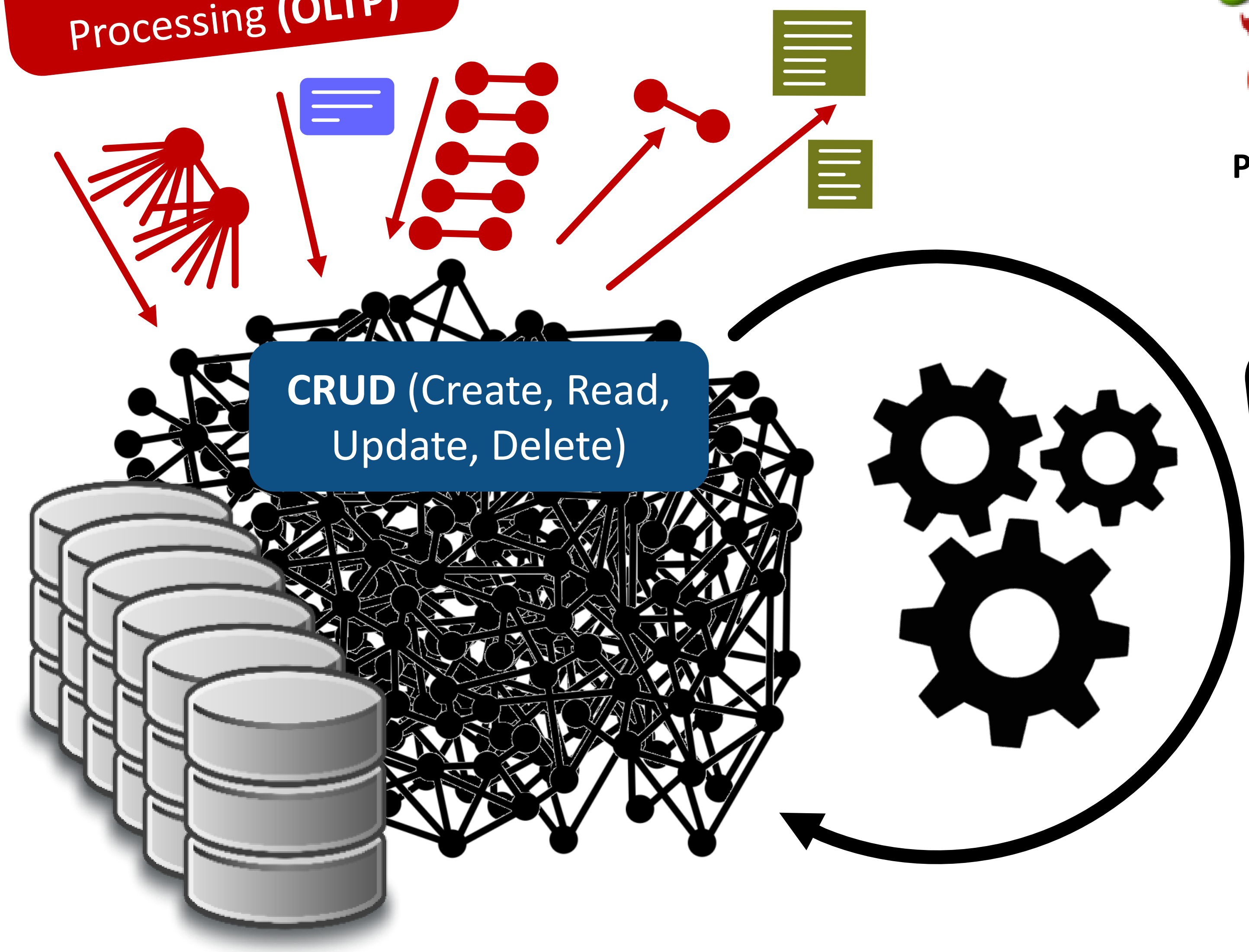
Graph Databases: Major Workloads

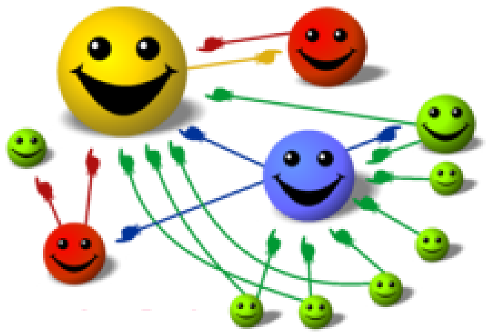


Graph Databases: Major Workloads

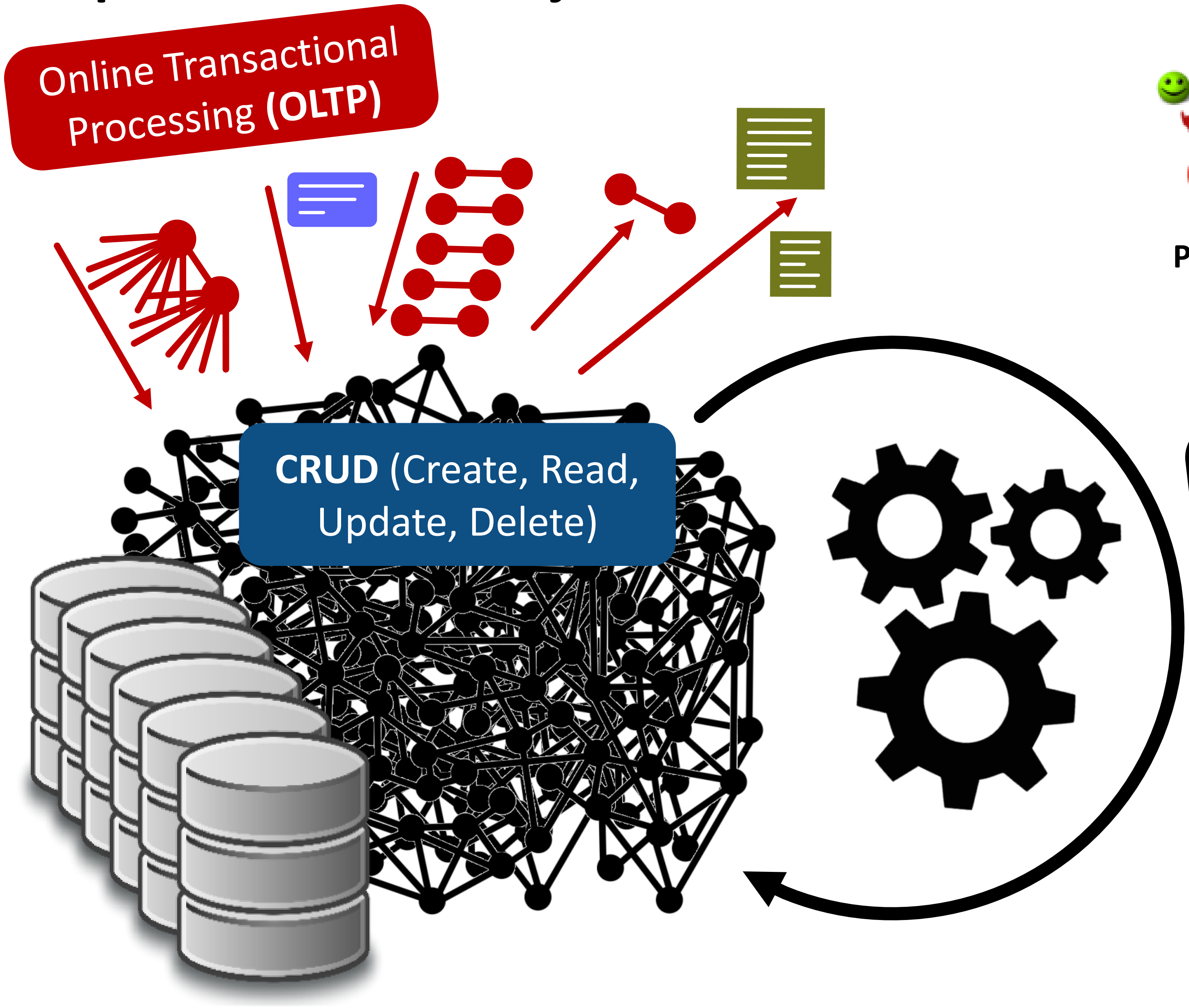


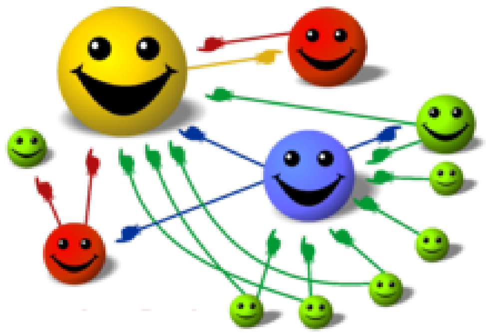
Graph Databases: Major Workloads





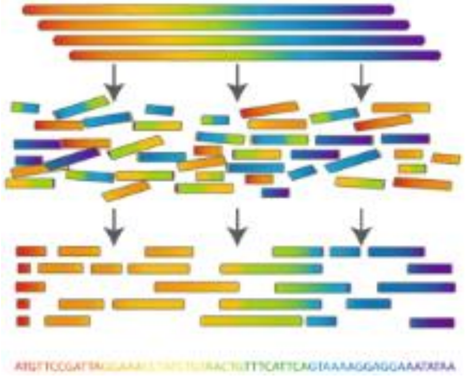
Graph Databases: Major Workloads



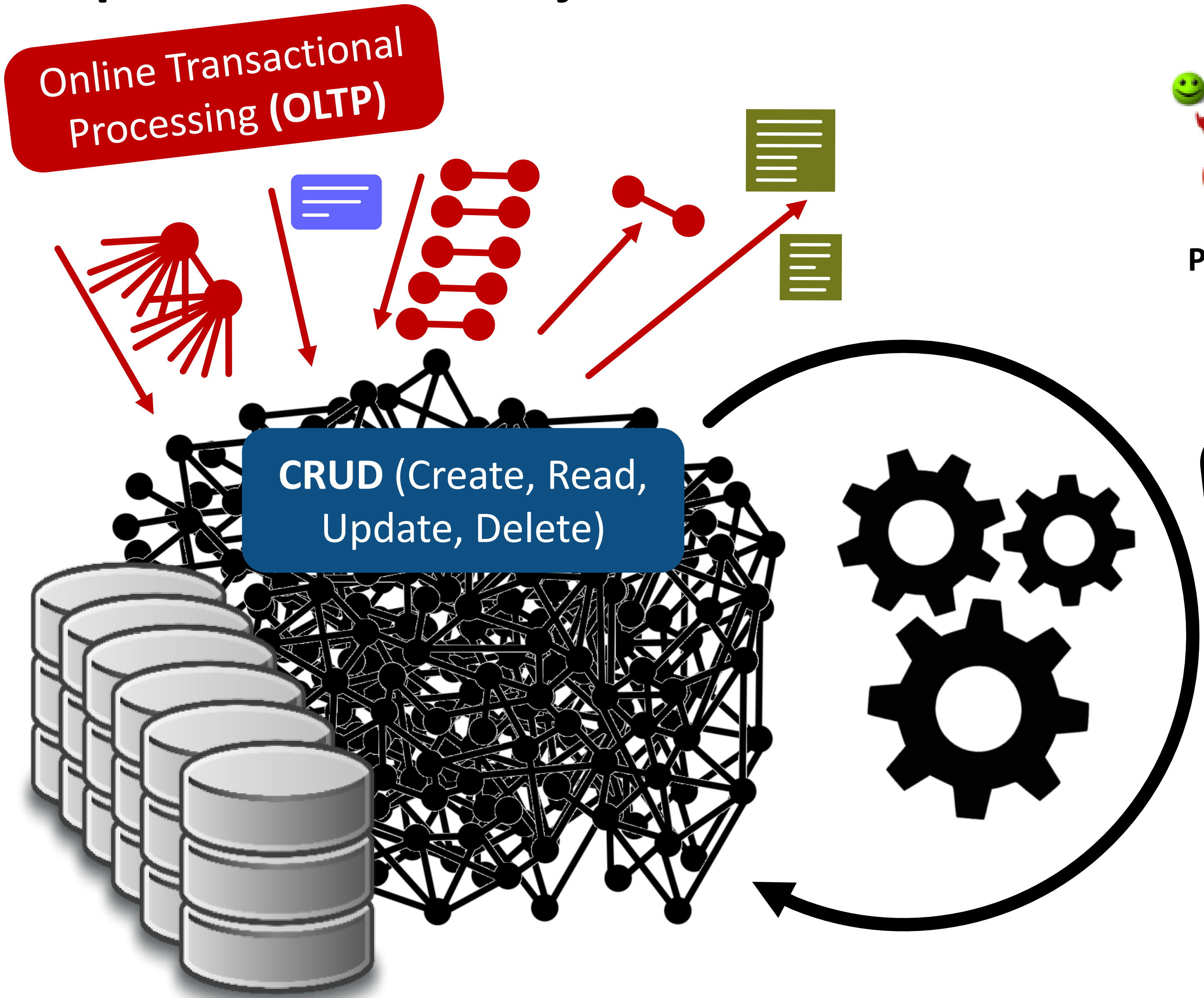


PageRank

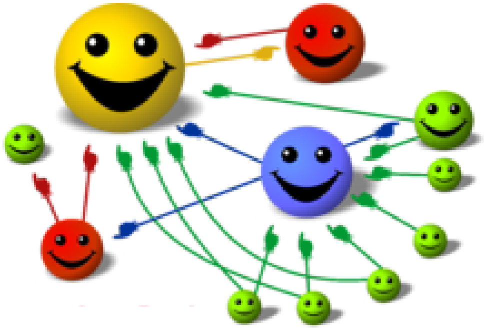
Genome (BFS) traversals



Graph Databases: Major Workloads

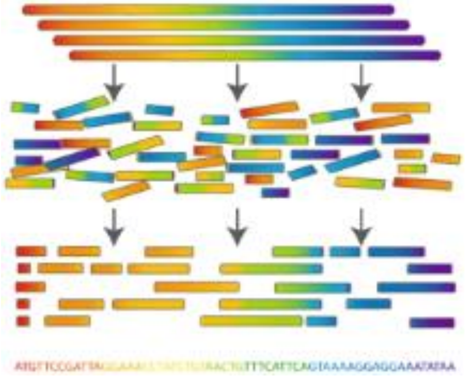


Online Analytical Processing (OLAP)

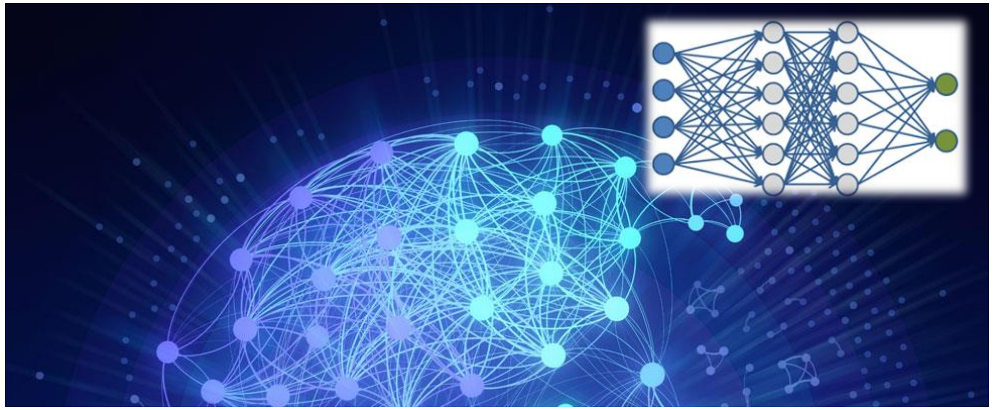


PageRank

Genome (BFS) traversals



Graph Neural Networks



How Does Deep Learning (DL) Work?

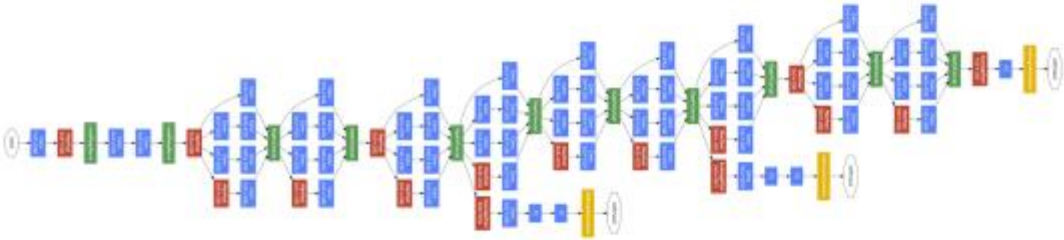
How Does Deep Learning (DL) Work?

How Does Deep Learning (DL) Work?

Samples

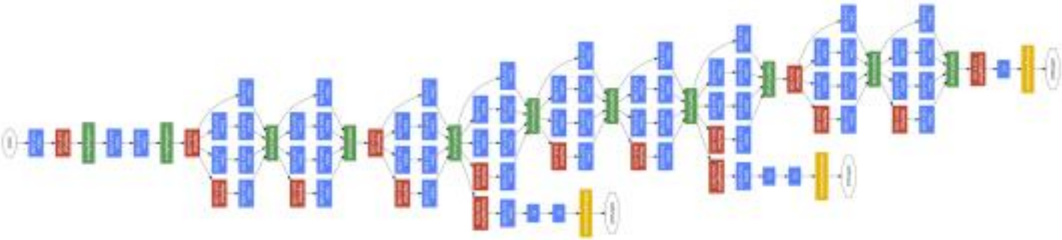
How Does Deep Learning (DL) Work?

Samples



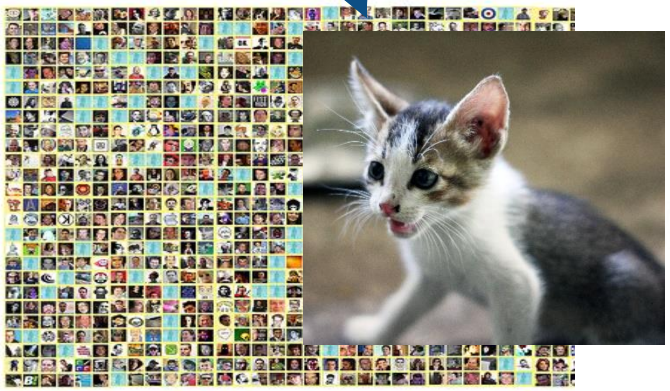
How Does Deep Learning (DL) Work?

Samples

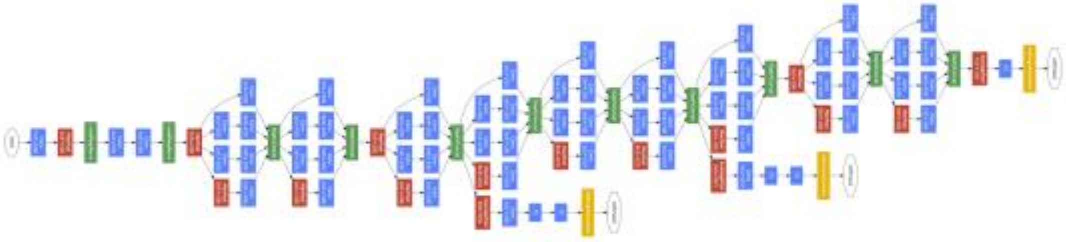


How Does Deep Learning (DL) Work?

Samples



f(x)

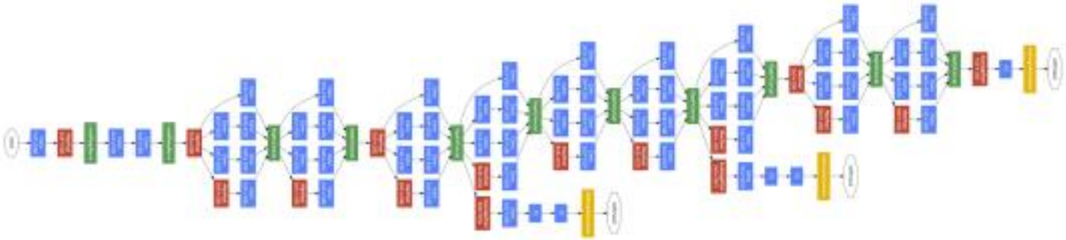


How Does Deep Learning (DL) Work?

Samples

Learning a heuristic

f(x)

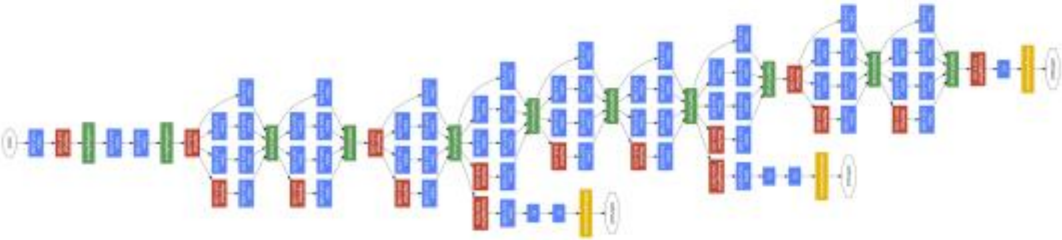


How Does Deep Learning (DL) Work?

Samples

Learning a heuristic

f(x)



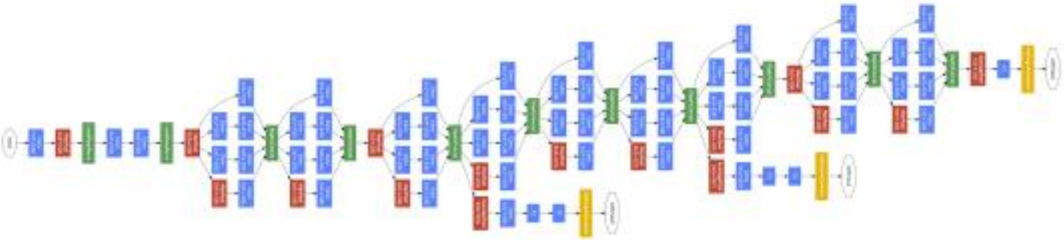
Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

How Does Deep Learning (DL) Work?

Samples

Learning a heuristic

f(x)



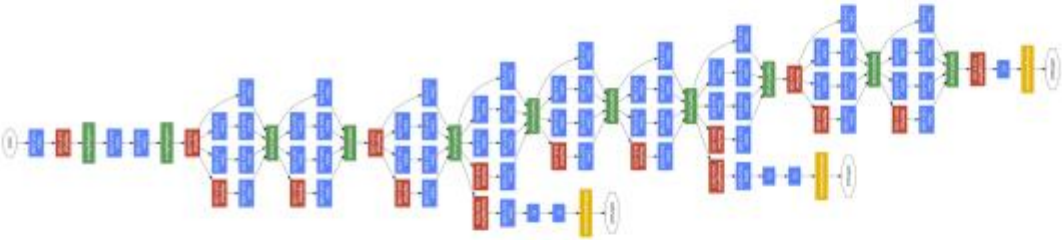
Cat	0.54	Cat	1.00
Dog	0.28	Dog	0.00
Airplane	0.07	Airplane	0.00
	0.04		0.00
Horse	0.33	Horse	0.00
Bicycle	0.02	Bicycle	0.00
Truck	0.02	Truck	0.00

How Does Deep Learning (DL) Work?

Samples

Learning a heuristic

f(x)



layer-wise weight update

Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

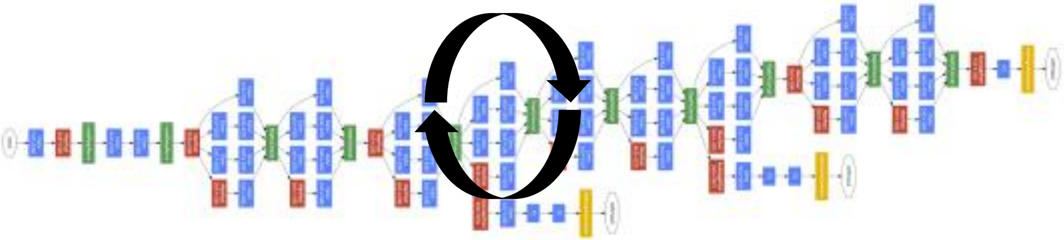
Cat	1.00
Dog	0.00
Airplane	0.00
	0.00
Horse	0.00
Bicycle	0.00
Truck	0.00

How Does Deep Learning (DL) Work?

Samples

Learning a heuristic

f(x)



layer-wise weight update

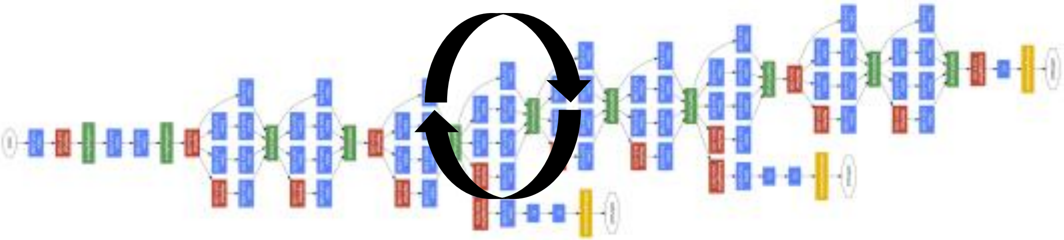
Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

Cat	1.00
Dog	0.00
Airplane	0.00
	0.00
Horse	0.00
Bicycle	0.00
Truck	0.00

Graph Neural Networks (GNNs)

Samples

f(x)



layer-wise weight update

Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

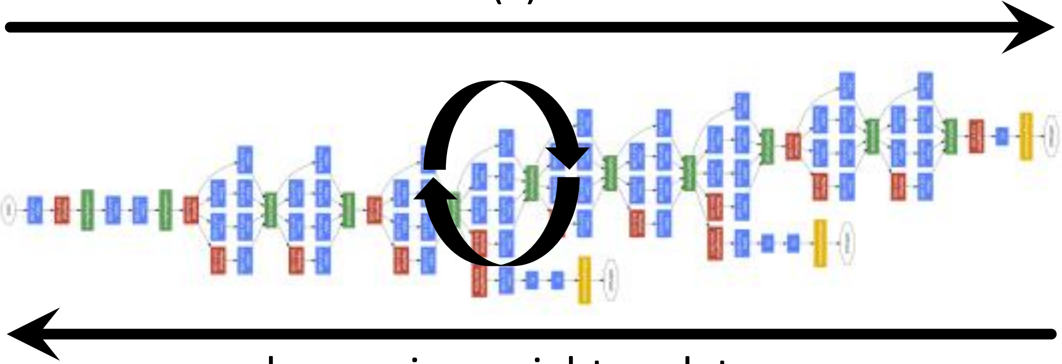
Cat	1.00
Dog	0.00
Airplane	0.00
	0.00
Horse	0.00
Bicycle	0.00
Truck	0.00

Graph Neural Networks (GNNs)

Samples

These could still be photos, but now forming **explicit relations**, e.g., two photos are related if they were taken within an hour

f(x)



layer-wise weight update

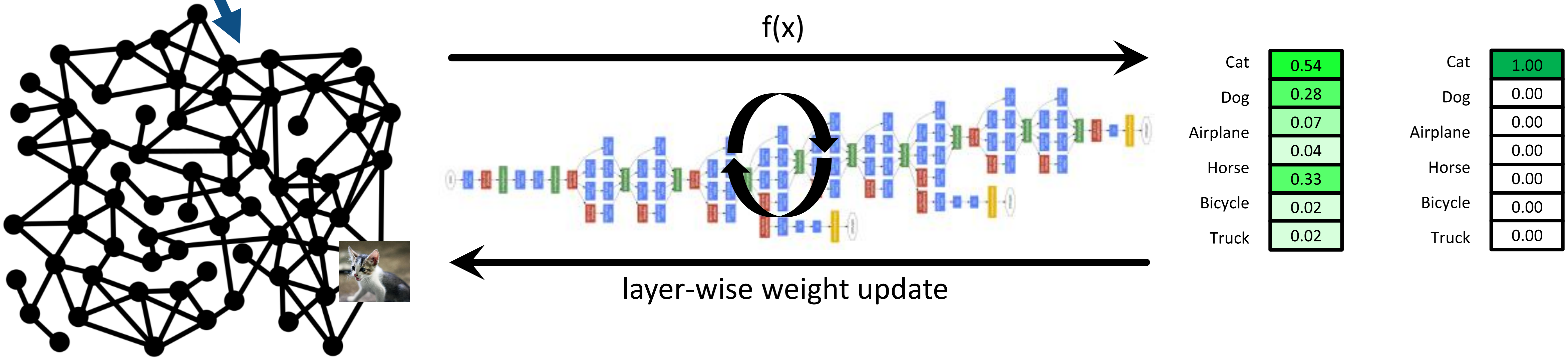
Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

Cat	1.00
Dog	0.00
Airplane	0.00
	0.00
Horse	0.00
Bicycle	0.00
Truck	0.00

Graph Neural Networks (GNNs)

Samples

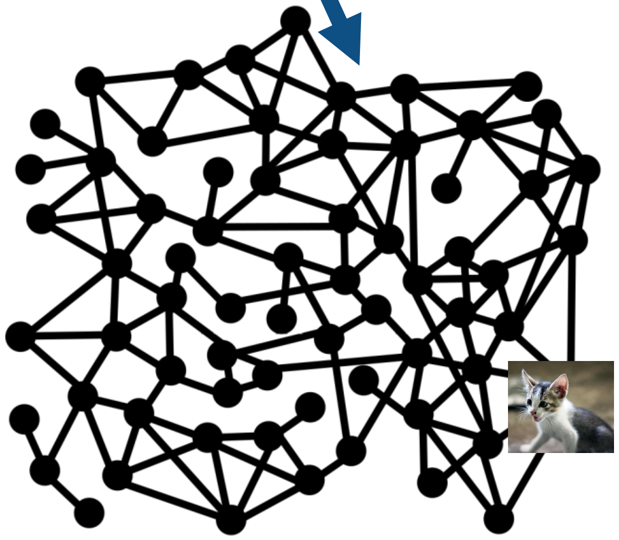
These could still be photos, but now forming **explicit relations**, e.g., two photos are related if they were taken within an hour



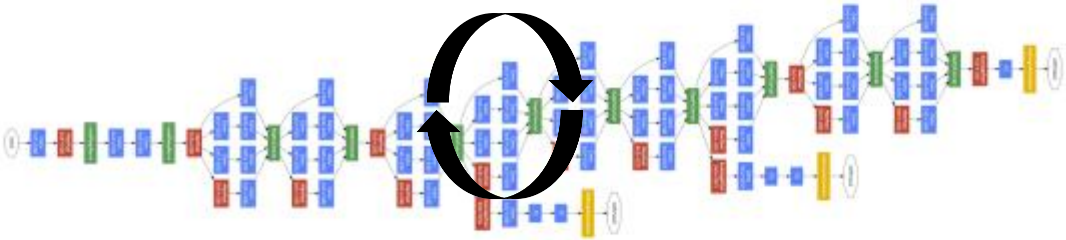
Graph Neural Networks (GNNs)

Samples

These could still be photos, but now forming **explicit relations**, e.g., two photos are related if they were taken within an hour



f(x)



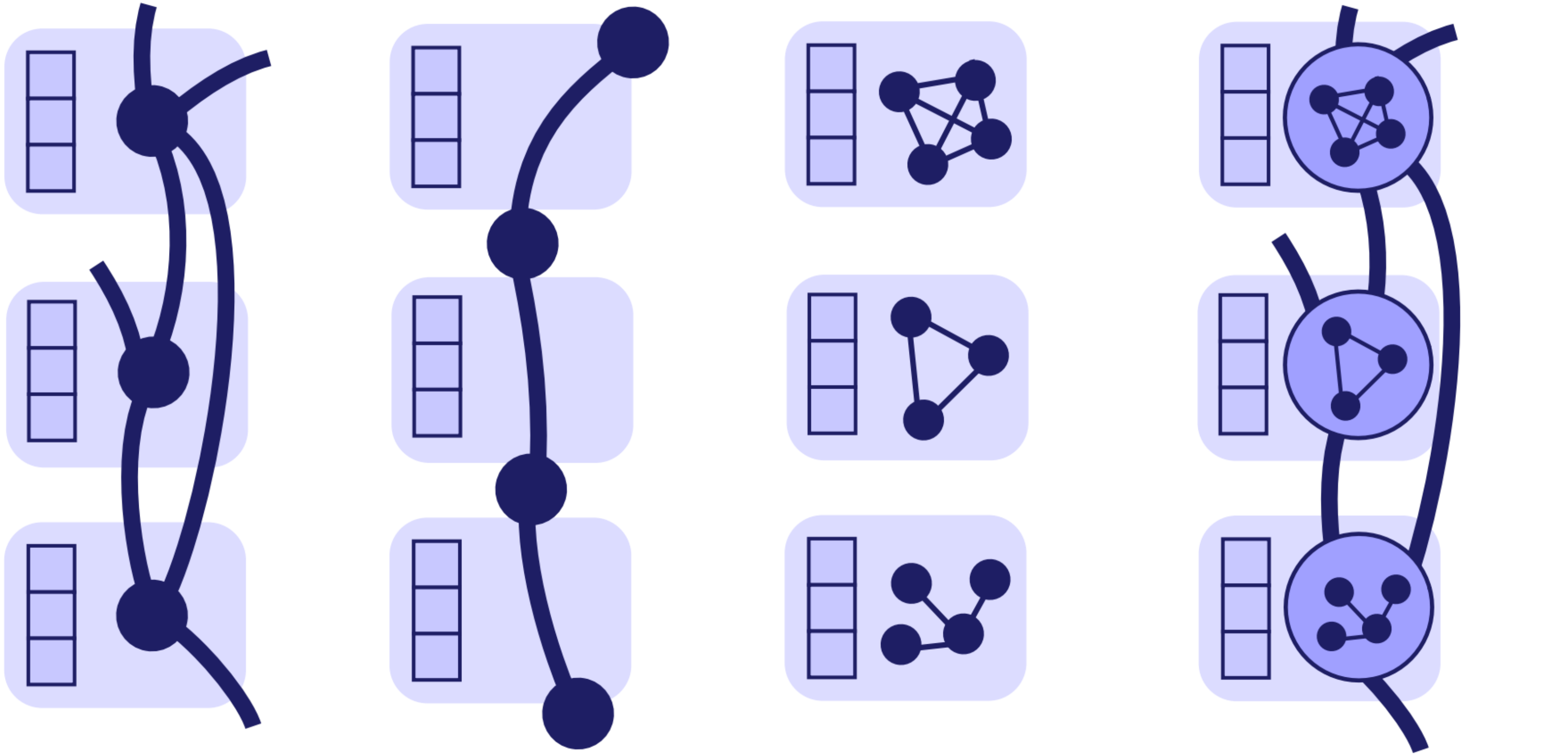
layer-wise weight update

Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

Cat	1.00
Dog	0.00
Airplane	0.00
	0.00
Horse	0.00
Bicycle	0.00
Truck	0.00

These dependencies make efficient processing of GNNs **much more complex** than in traditional DL

Types of Samples & Downstream Tasks: GNNs vs. Traditional DL



Dependencies between samples in GNNs

Even in independent graph case, there are intra-sample dependencies

M. Besta, T. Hoefler. *Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis*. https://arxiv.org/abs/2205.09702, 2022.