MACIEJ BESTA, TORSTEN HOEFLER

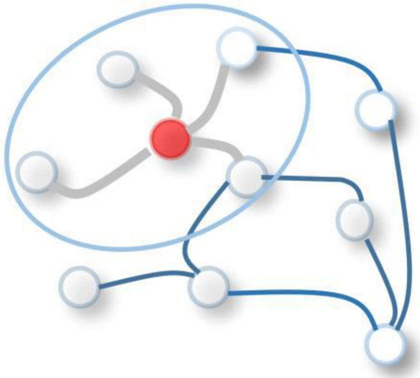
Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

SPCL

Future Computing Laboratory

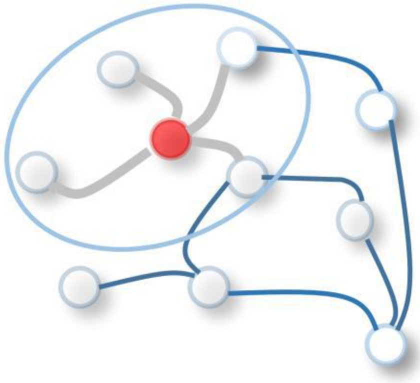
Overview of My Research: High-Performance Irregular Workloads & Interconnects

Overview of My Research: High-Performance Irregular Workloads & Interconnects

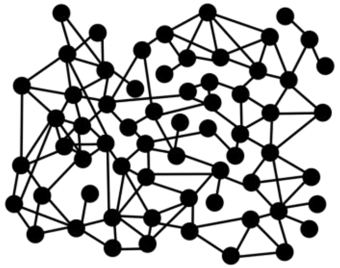


Graph neural networks

Overview of My Research: High-Performance Irregular Workloads & Interconnects

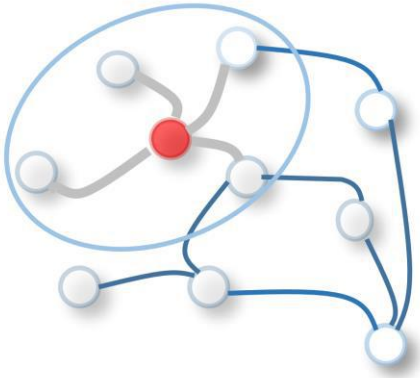


Graph neural networks

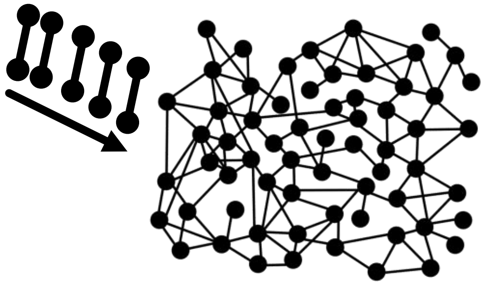


Graph streaming

Overview of My Research: High-Performance Irregular Workloads & Interconnects

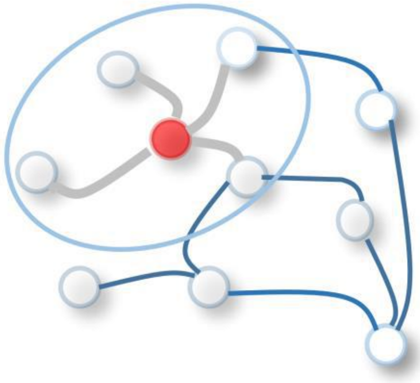


Graph neural networks

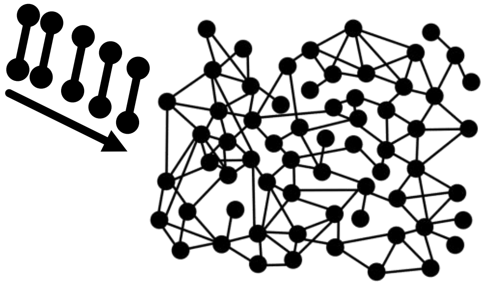


Graph streaming

Overview of My Research: High-Performance Irregular Workloads & Interconnects



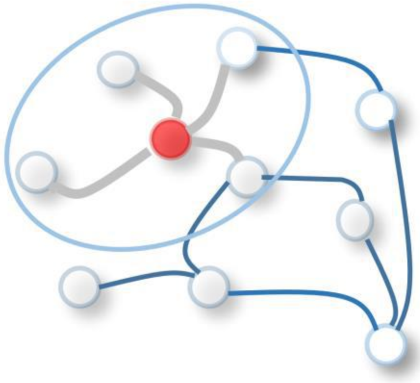
Graph neural networks



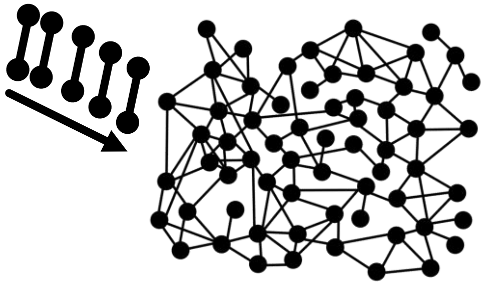
Graph streaming

Graph databases

Overview of My Research: High-Performance Irregular Workloads & Interconnects

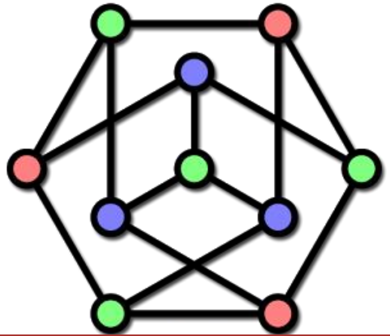


Graph neural networks



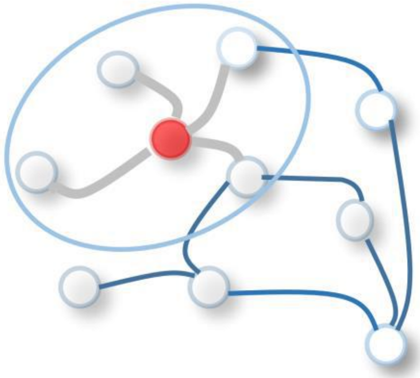
Graph streaming

Graph databases

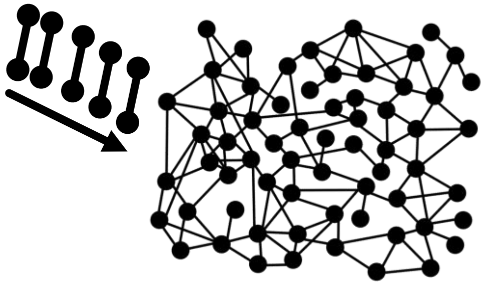


Graph algorithms

Overview of My Research: High-Performance Irregular Workloads & Interconnects

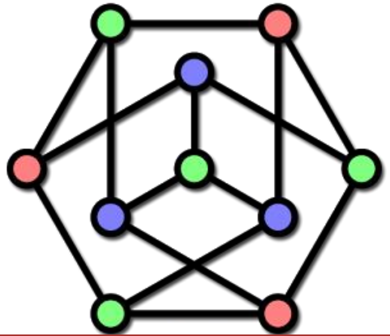


Graph neural networks

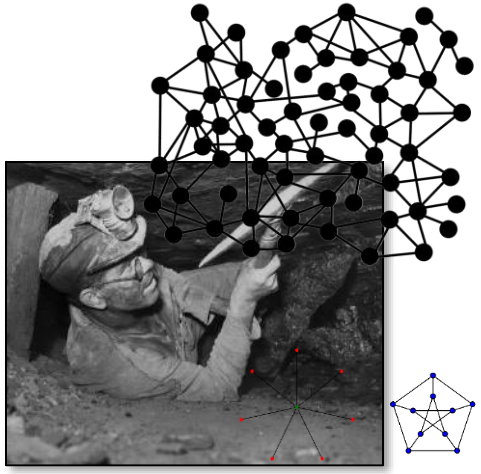


Graph streaming

Graph databases



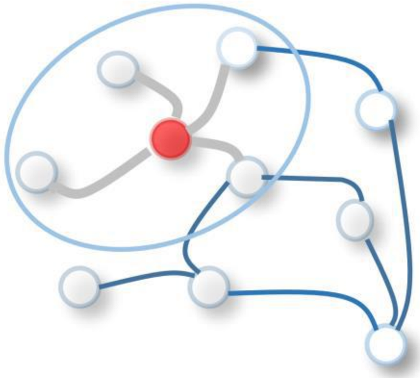
Graph algorithms



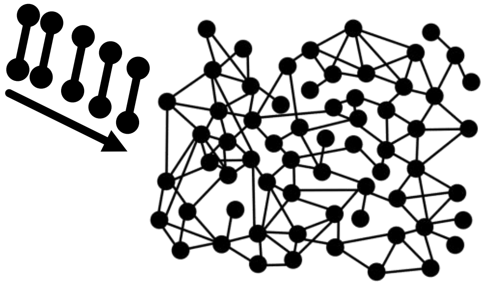
Graph mining

Overview of My Research: High-Performance Irregular Workloads & Interconnects

…

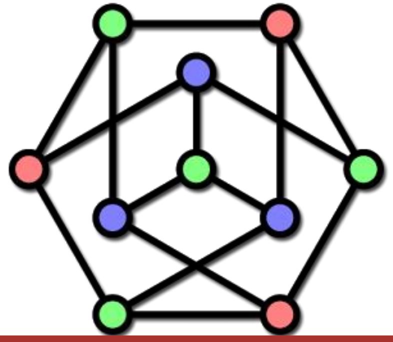


Graph neural networks

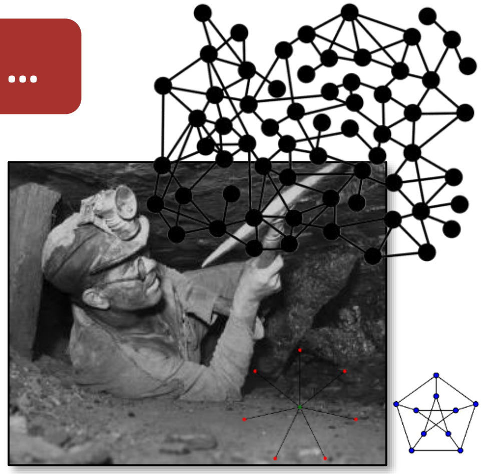


Graph streaming

Graph databases



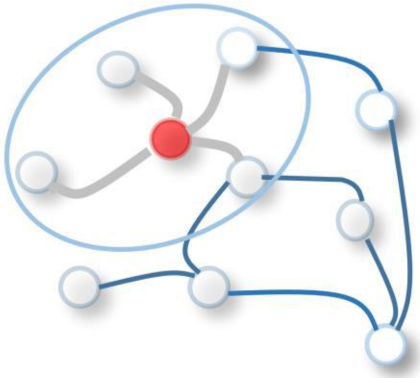
Graph algorithms



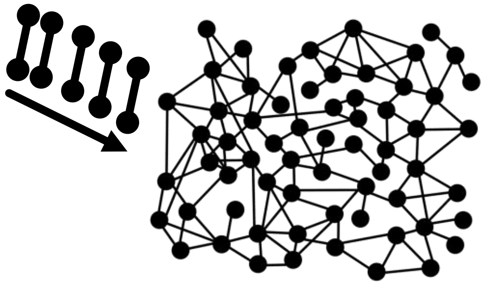
Graph mining

Overview of My Research: High-Performance Irregular Workloads & Interconnects

...

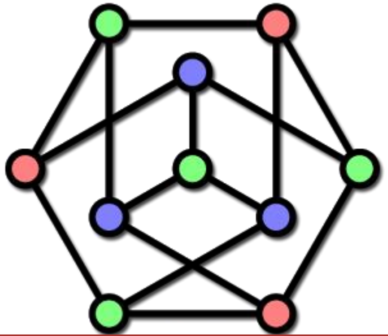


Graph neural networks

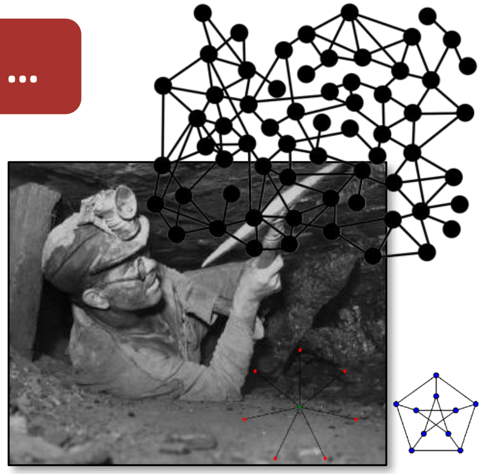


Graph streaming

Graph databases

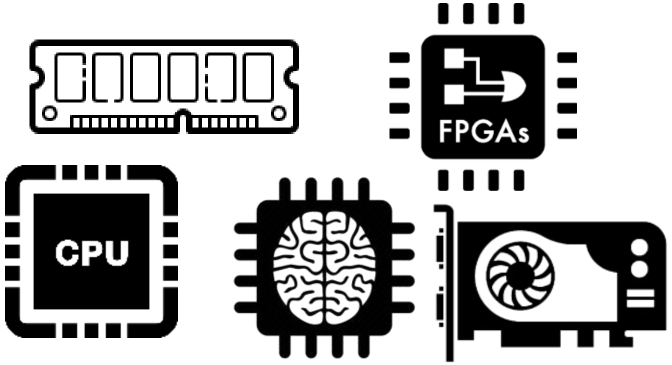


Graph algorithms



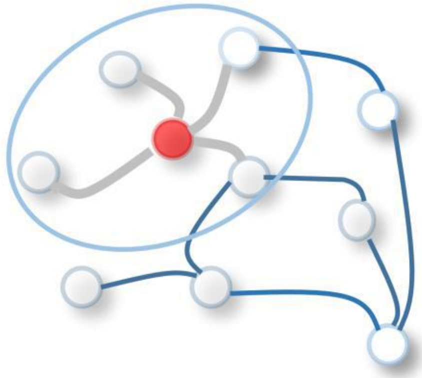
Graph mining

Heterogeneous resources

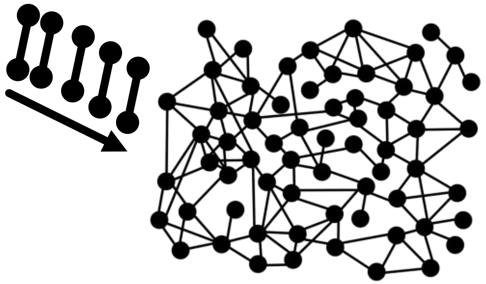


Overview of My Research: High-Performance Irregular Workloads & Interconnects

...

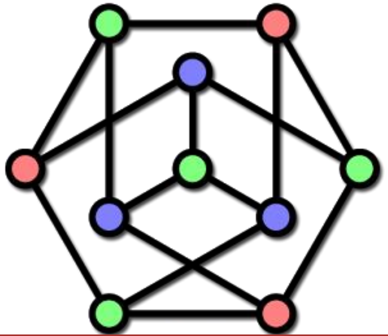


Graph neural networks

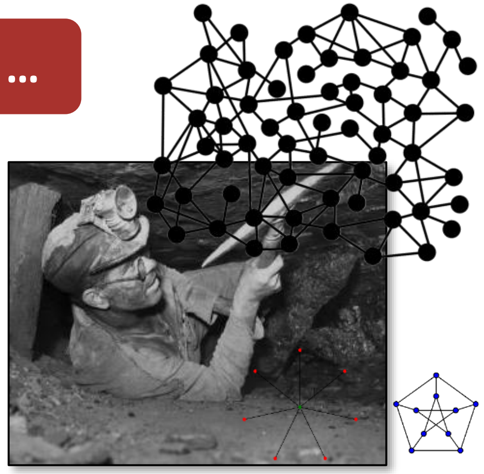


Graph streaming

Graph databases

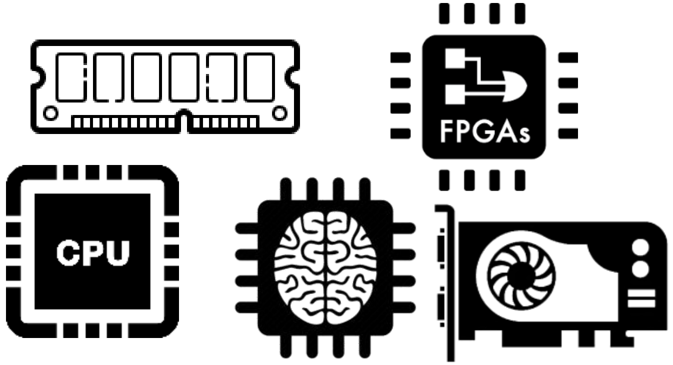


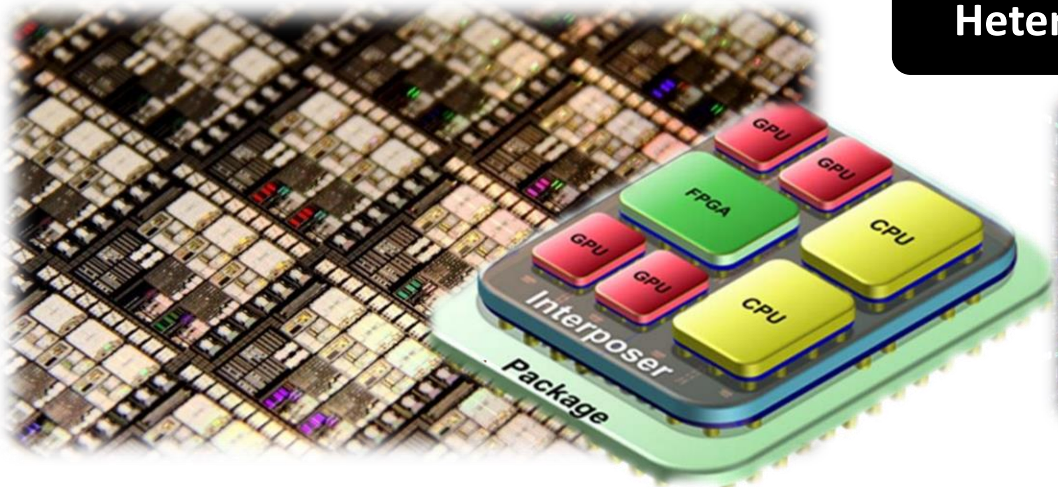
Graph algorithms



Graph mining

Heterogeneous resources

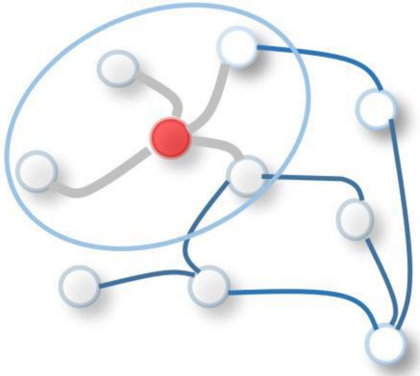




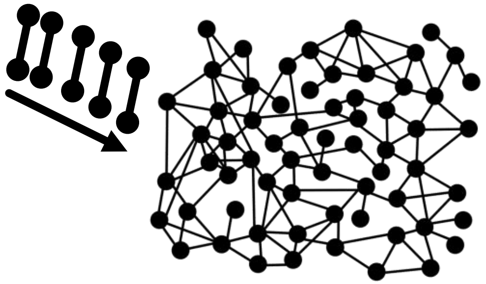
Heterogeneous settings

Overview of My Research: High-Performance Irregular Workloads & Interconnects

...

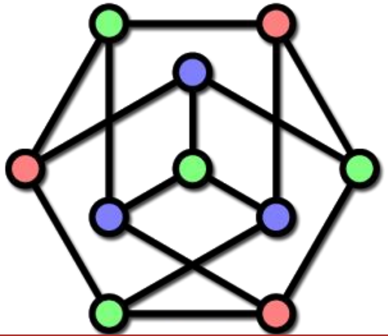


Graph neural networks

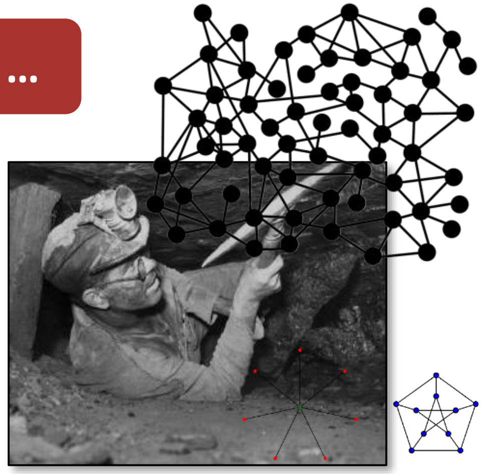


Graph streaming

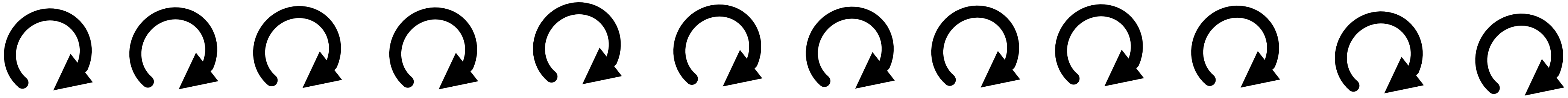
Graph databases



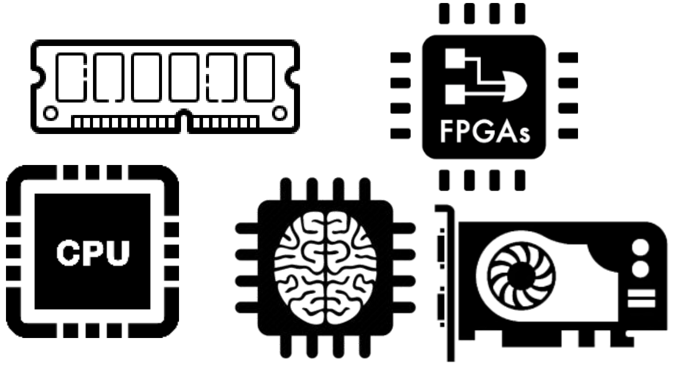
Graph algorithms

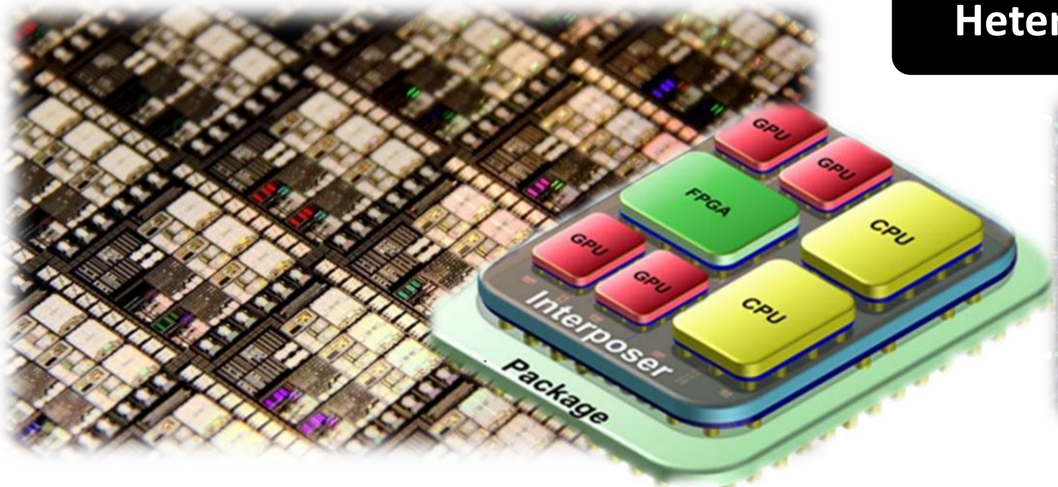


Graph mining



Heterogeneous resources

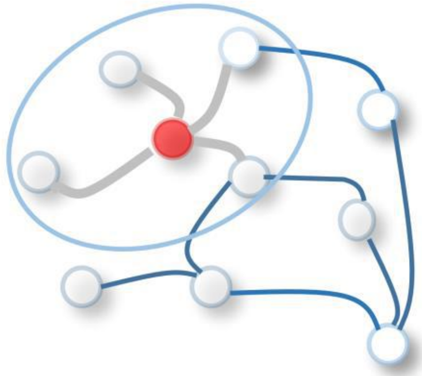


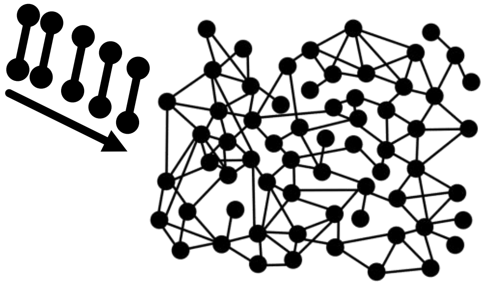


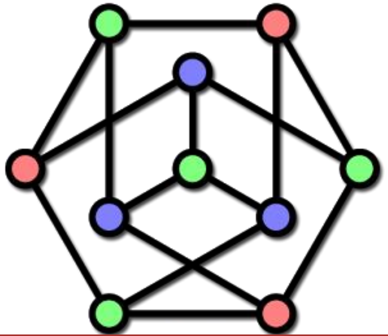
Heterogeneous settings

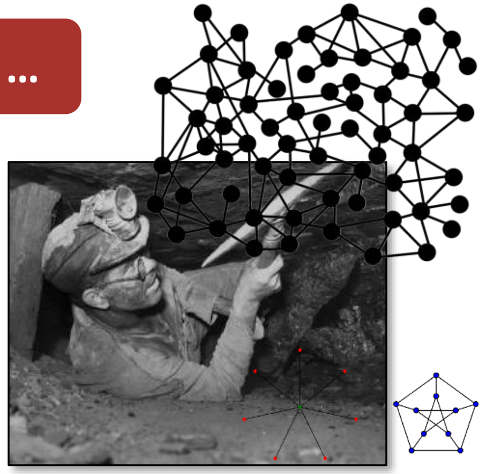
Overview of My Research: High-Performance Irregular Workloads & Interconnects

...









Graph neural networks

Graph streaming

Graph databases

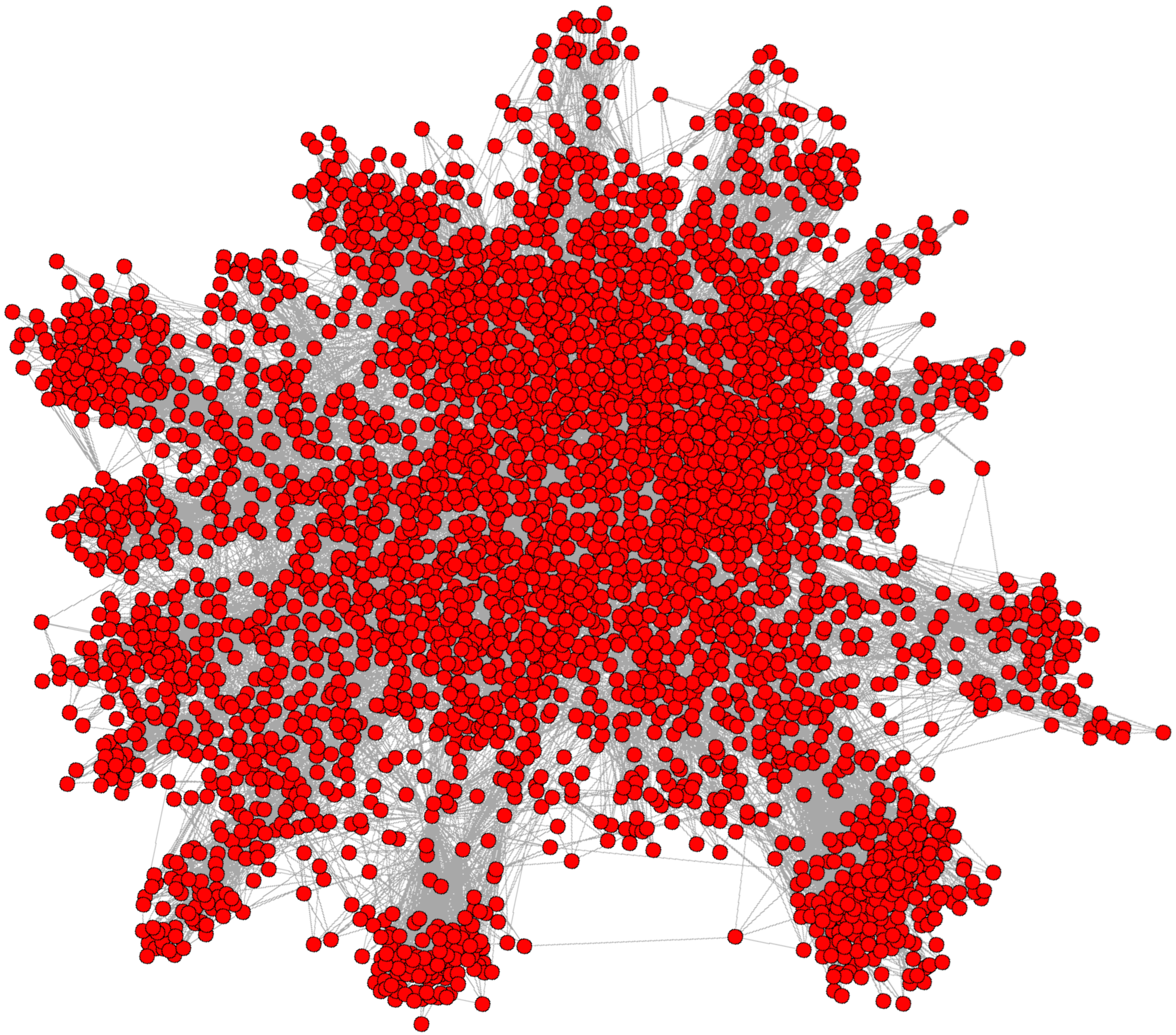
Graph algorithms

Graph mining

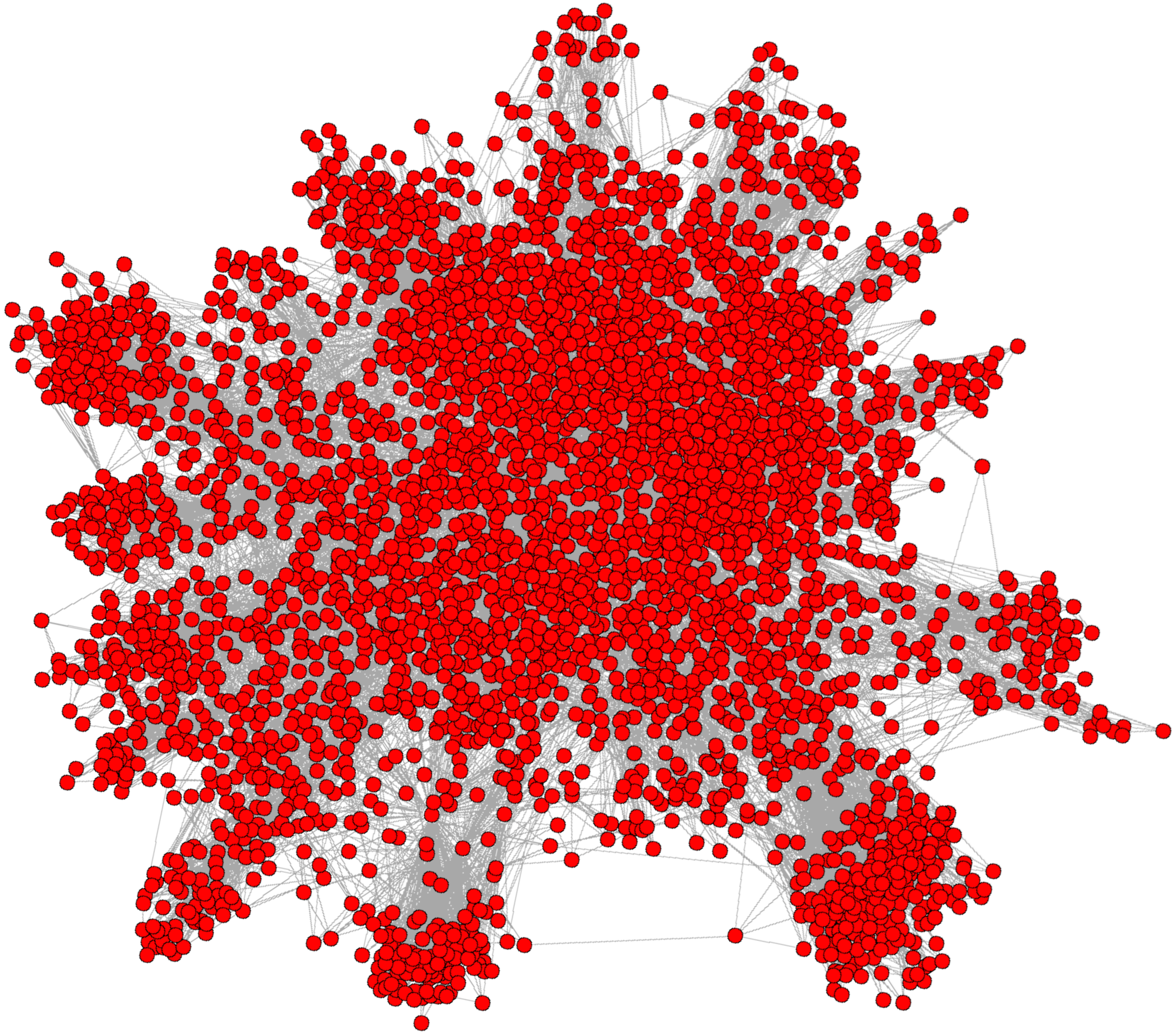
Heterogeneous resources

Heterogeneous settings

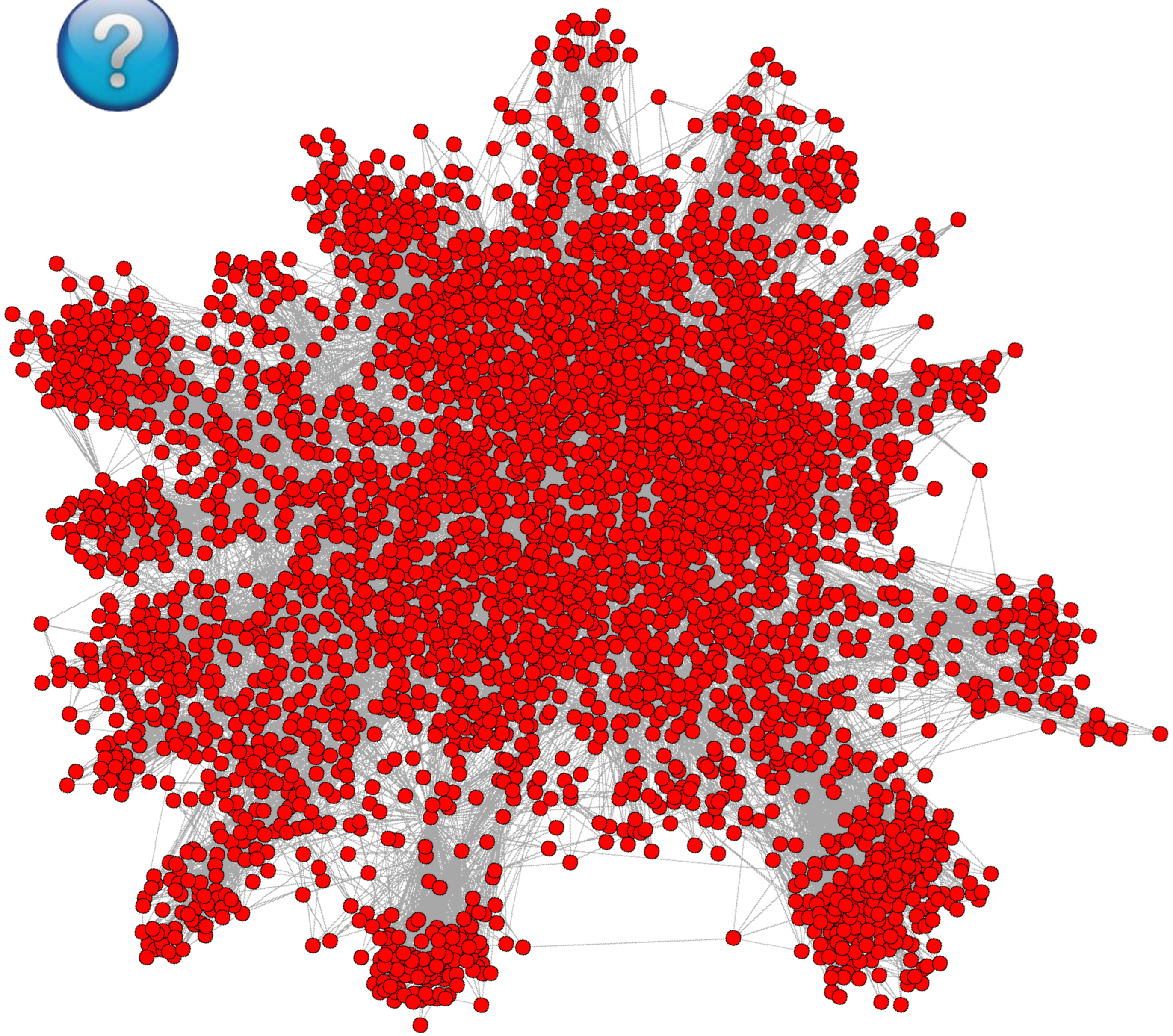
High-Performance Irregular Workloads



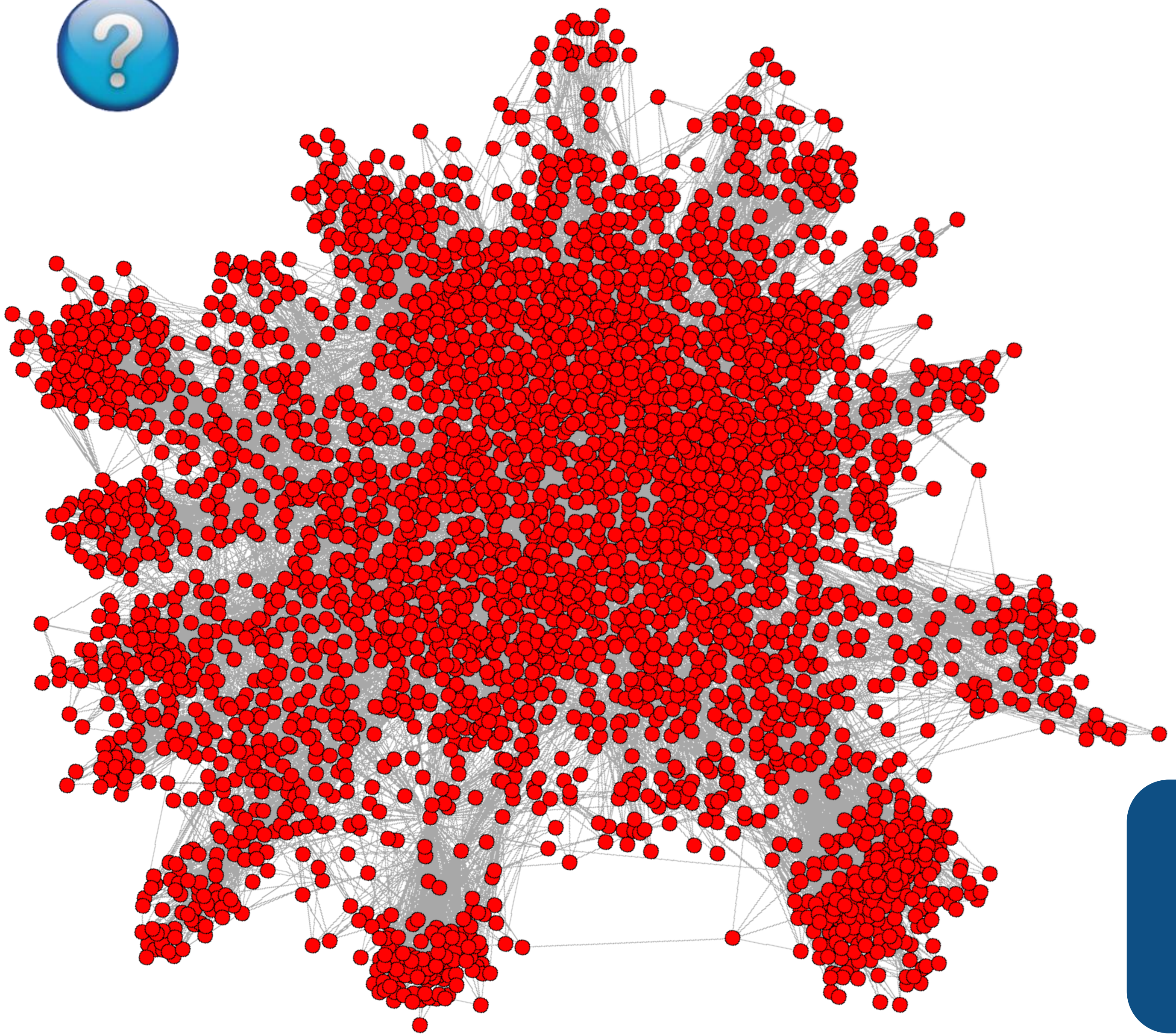
High-Performance Irregular Workloads



High-Performance Irregular Workloads

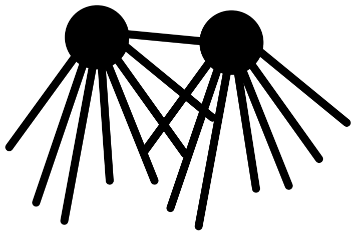


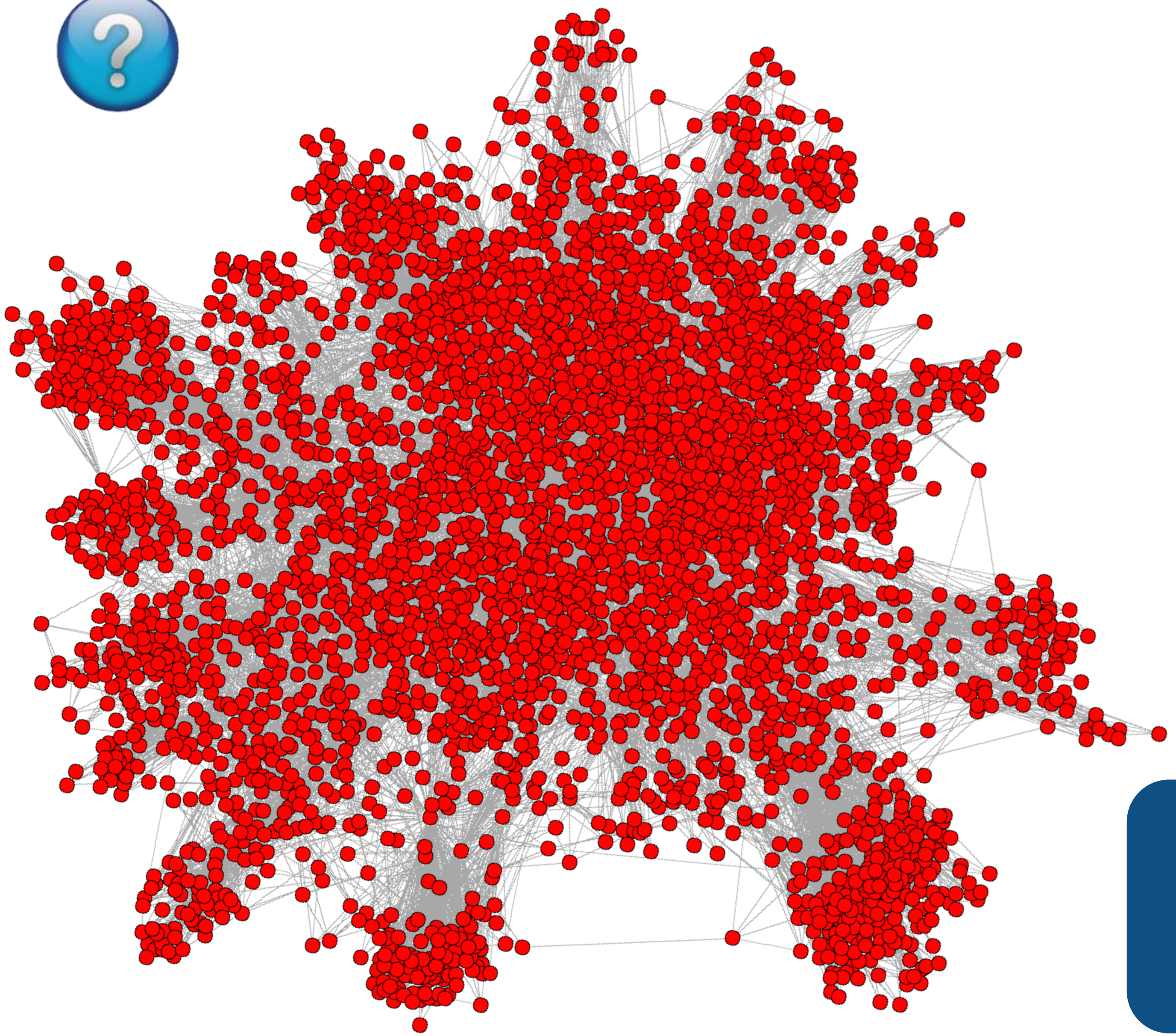
High-Performance Irregular Workloads



For example:
"Irregularity" of input datasets

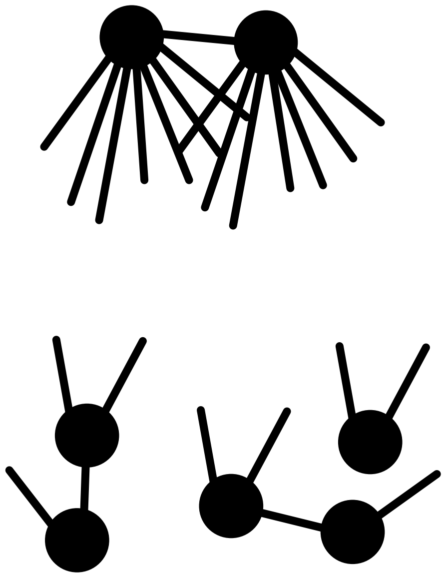
High-Performance Irregular Workloads

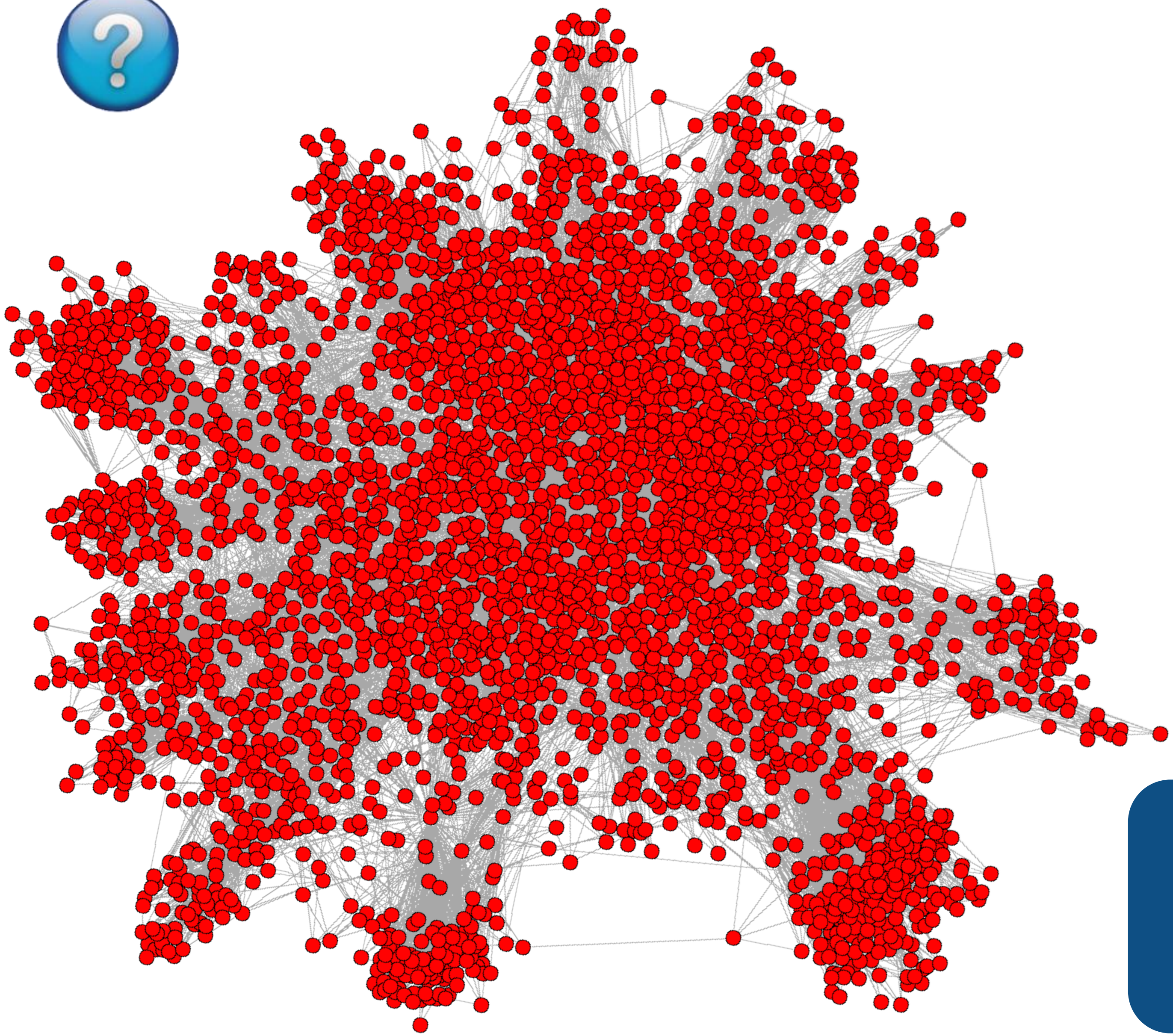




For example:
"Irregularity" of input datasets

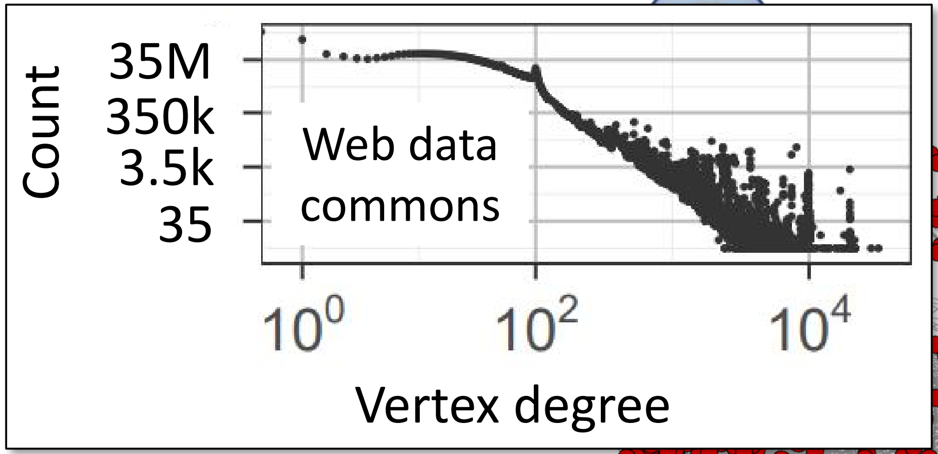
High-Performance Irregular Workloads

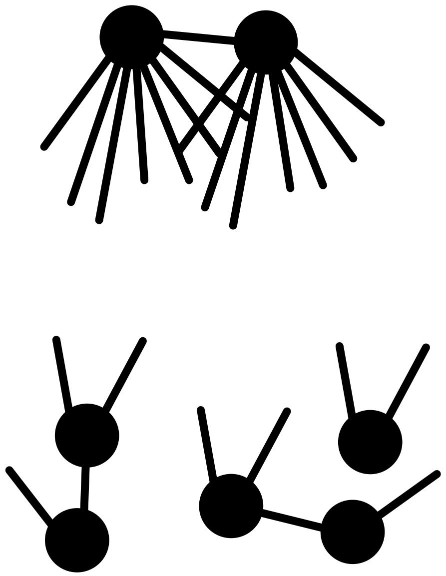


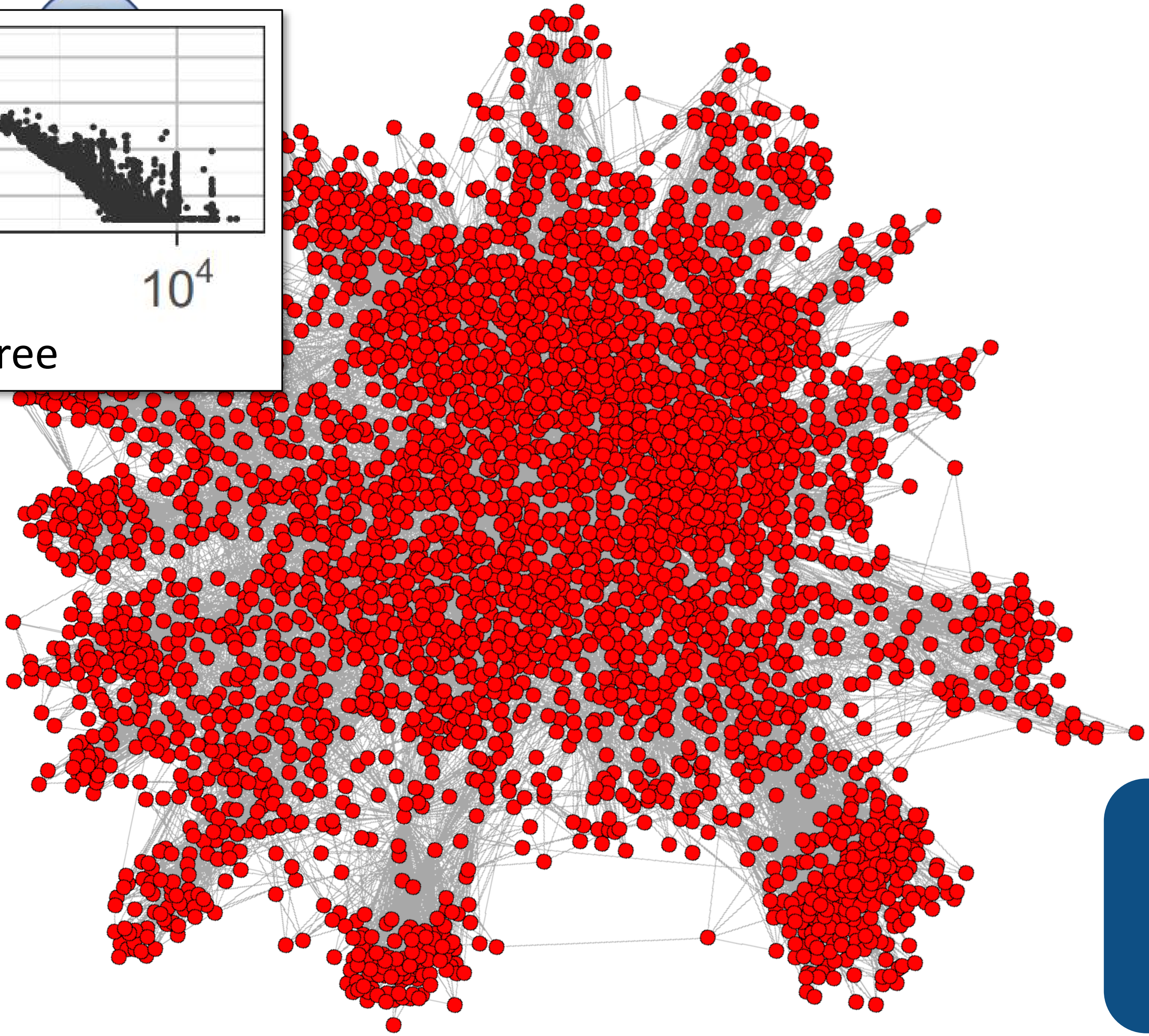


For example:
"Irregularity" of input datasets

High-Performance Irregular Workloads

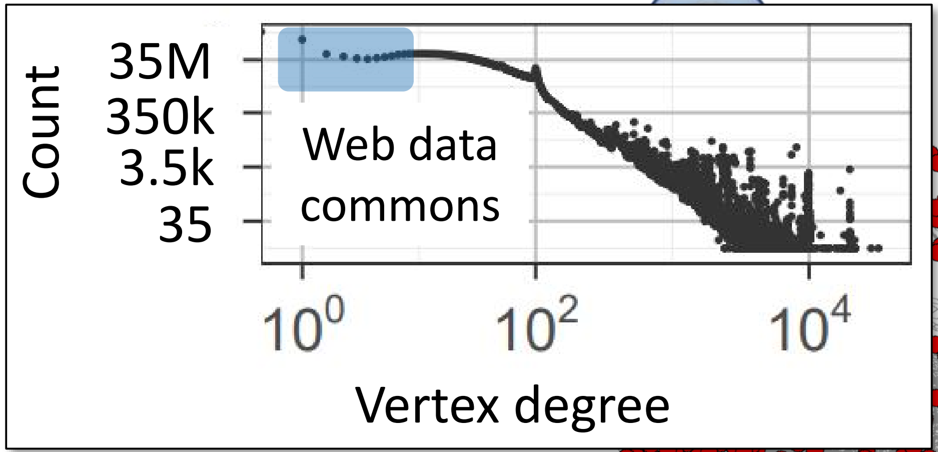


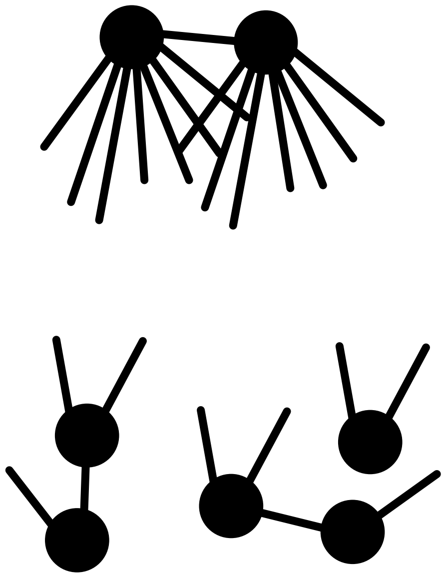


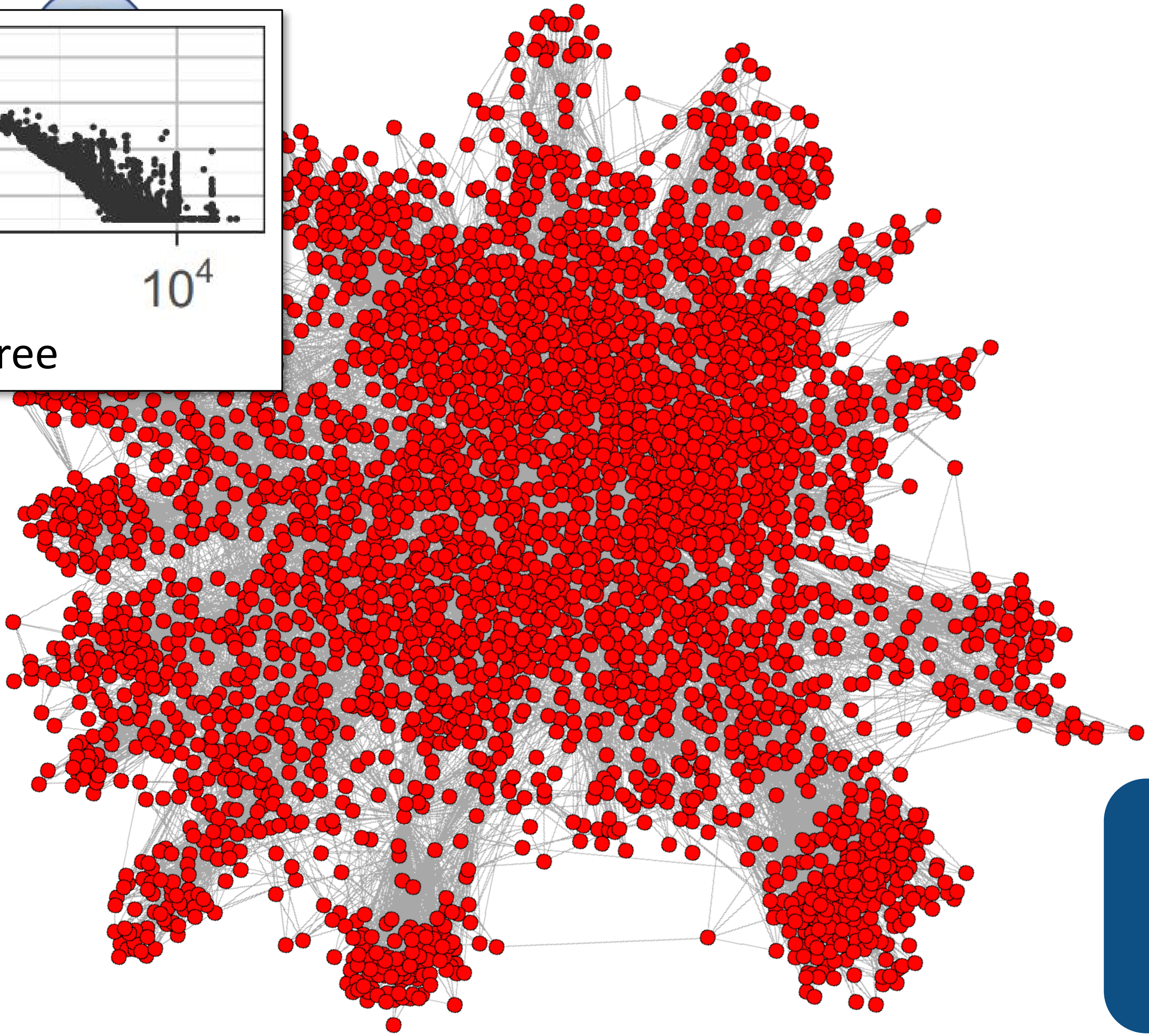


For example:
"Irregularity" of input datasets

High-Performance Irregular Workloads

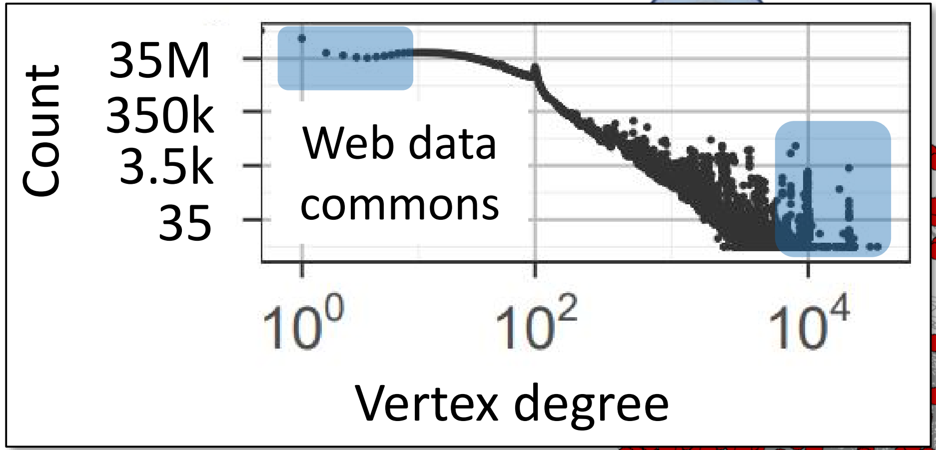


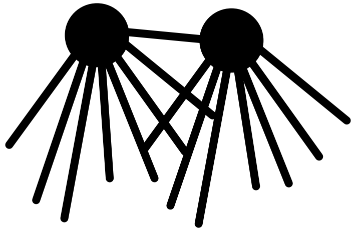


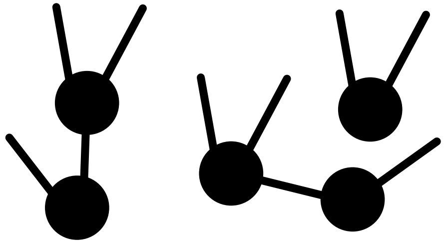


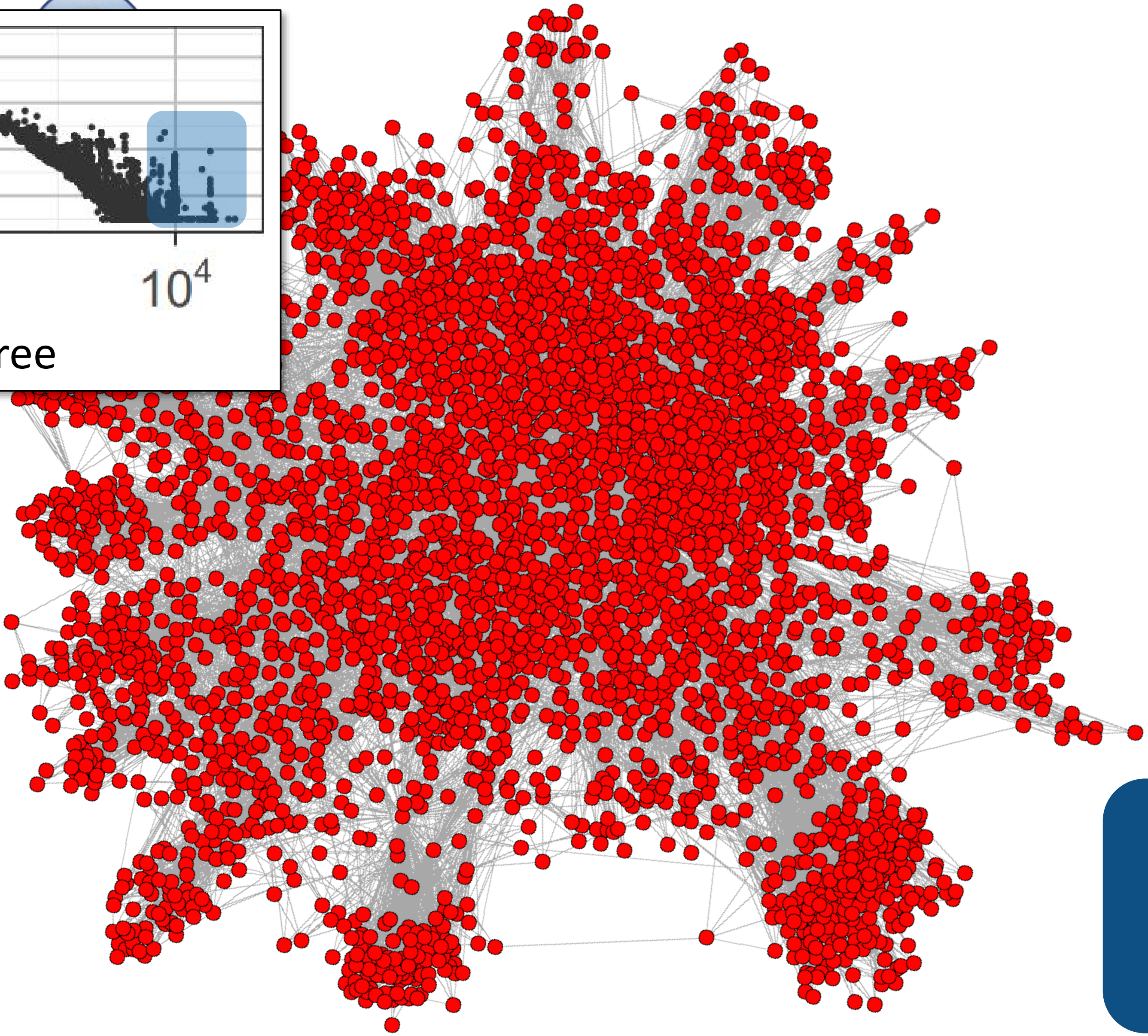
For example:
"Irregularity" of input datasets

High-Performance Irregular Workloads



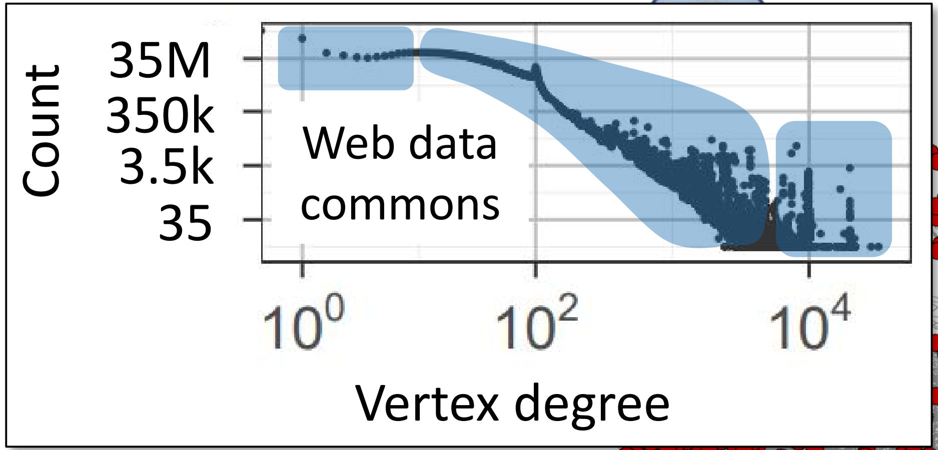


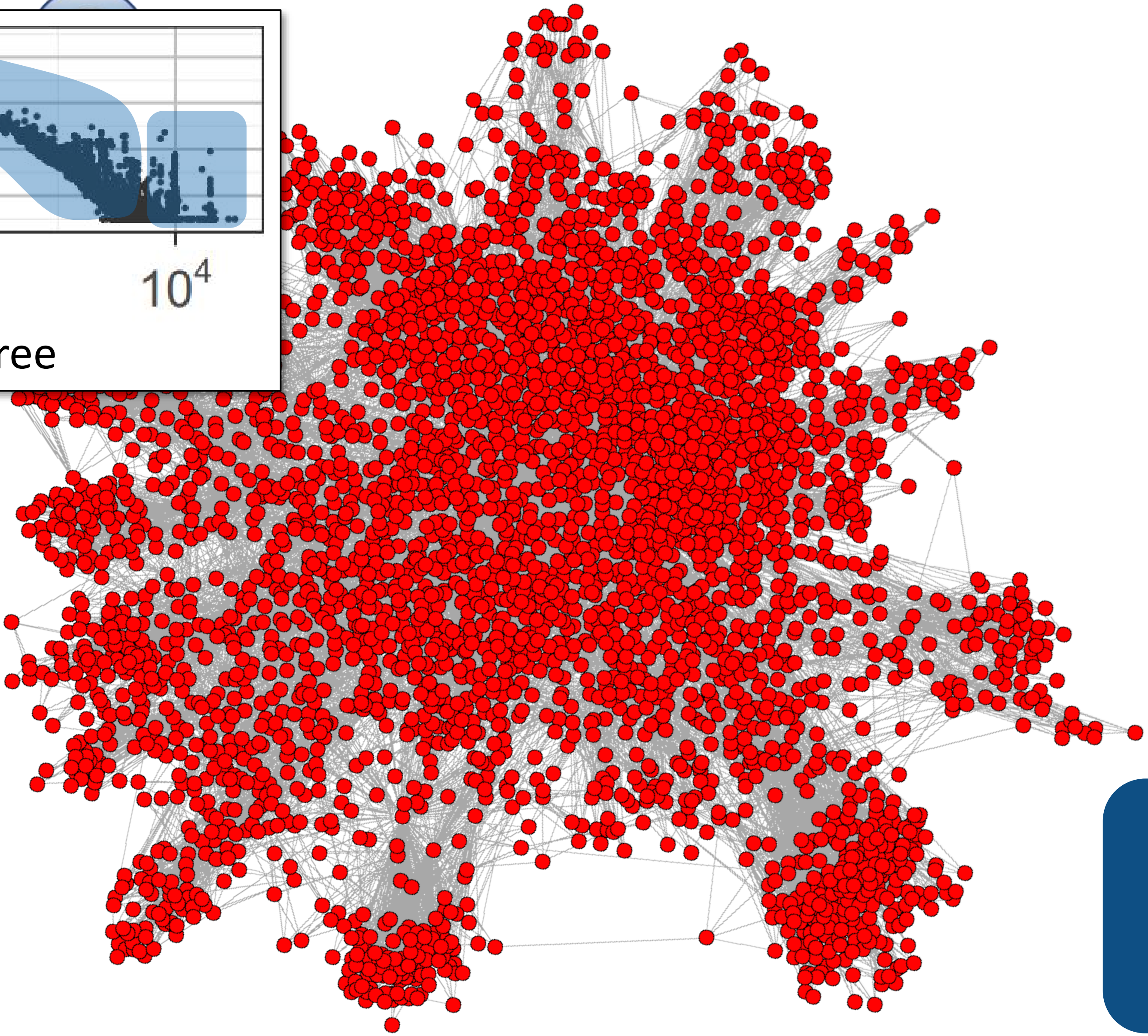


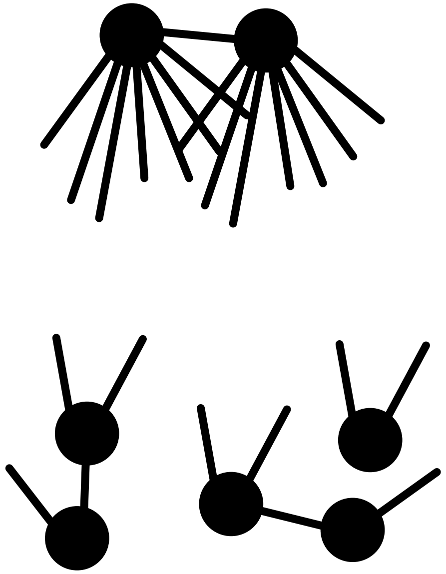


For example:
“Irregularity” of input datasets

High-Performance Irregular Workloads

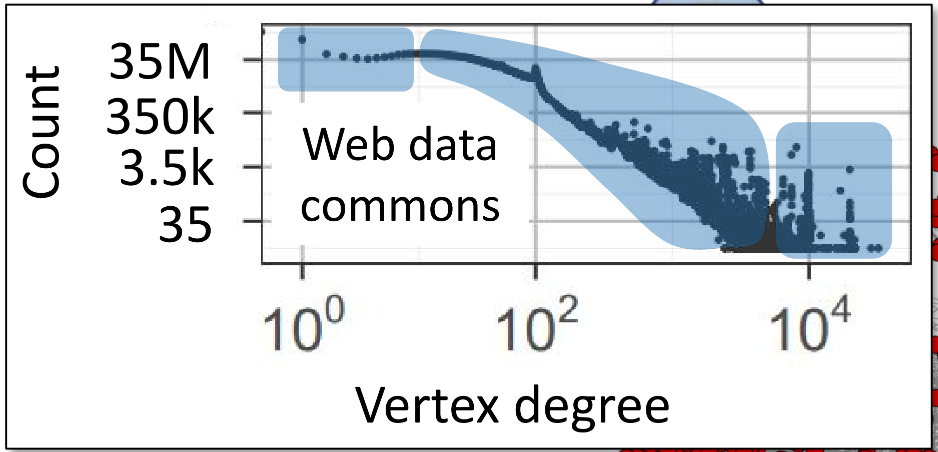


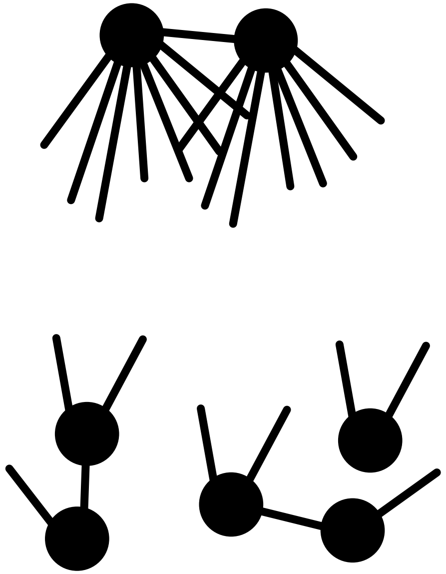


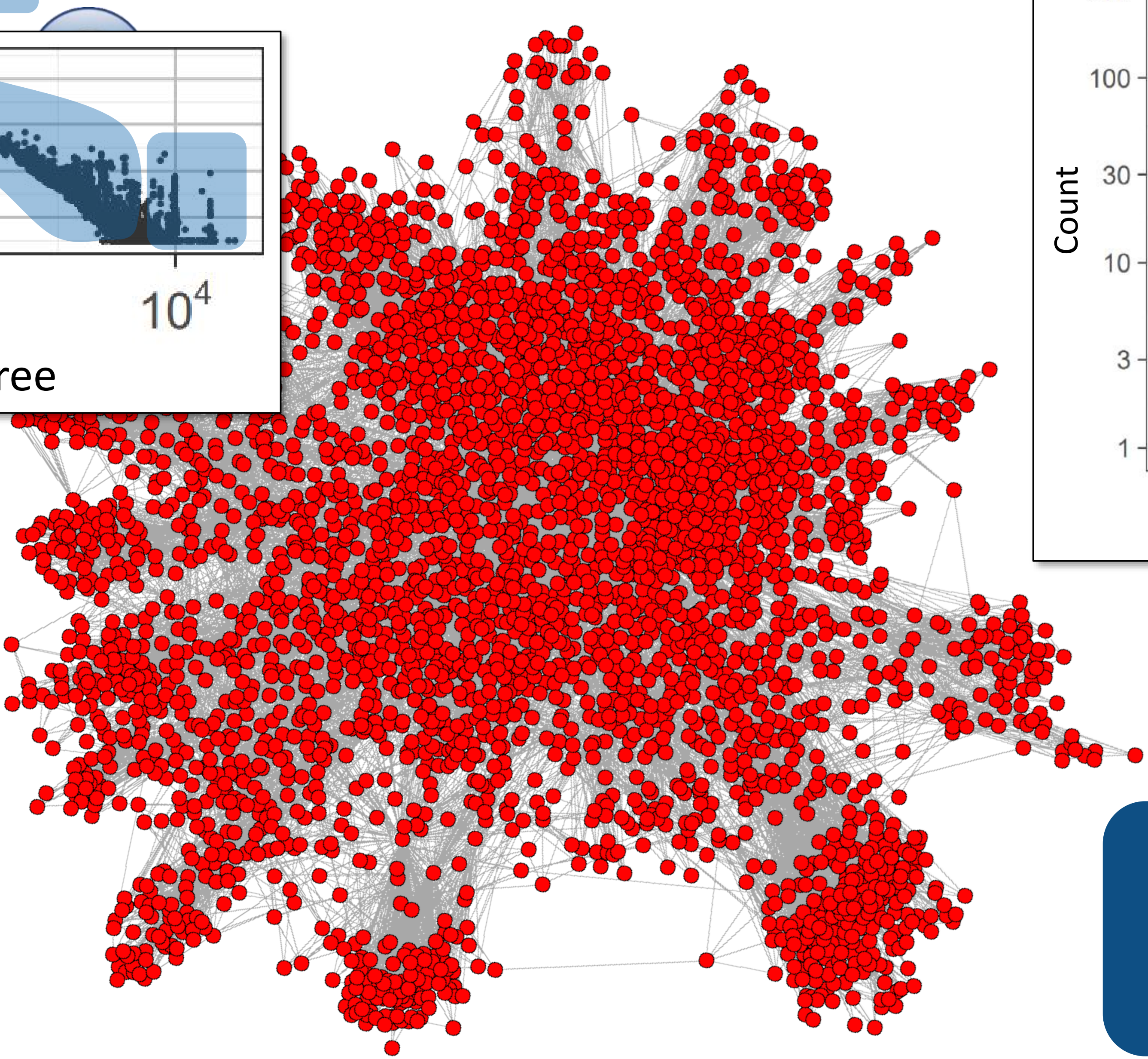


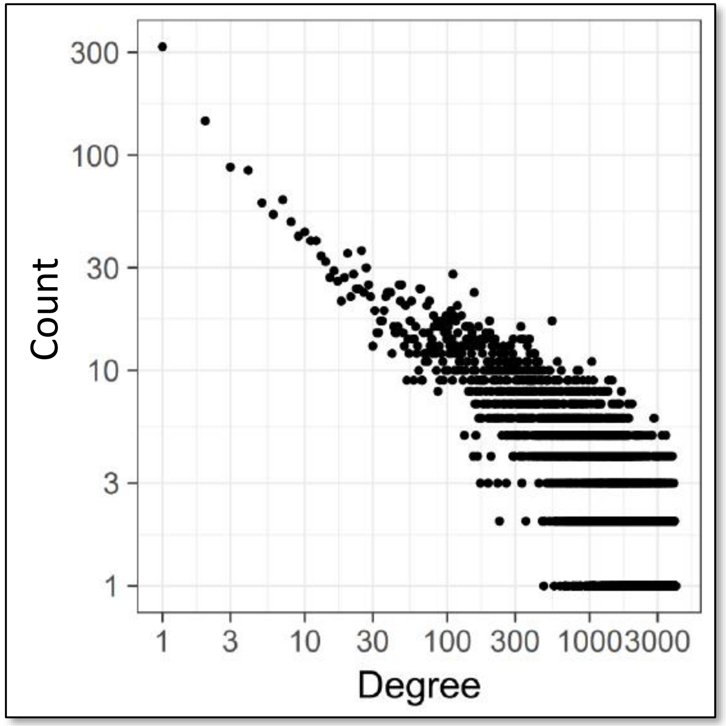
For example:
"Irregularity" of input datasets

High-Performance Irregular Workloads



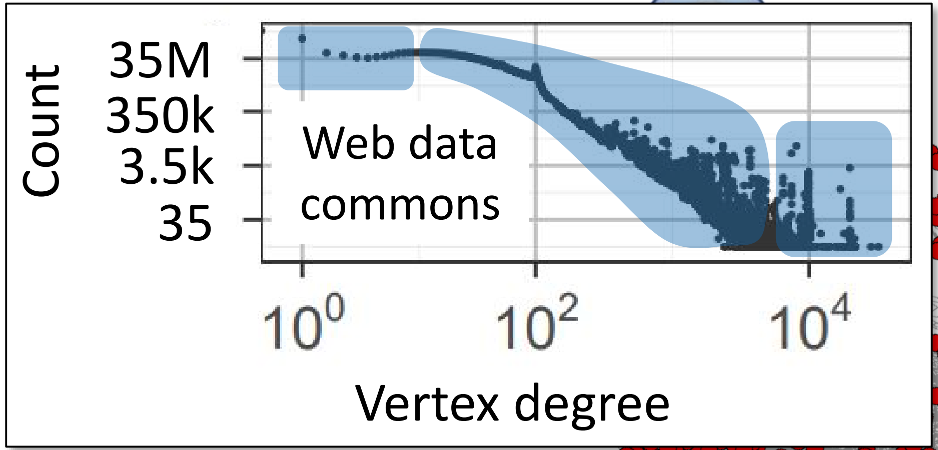


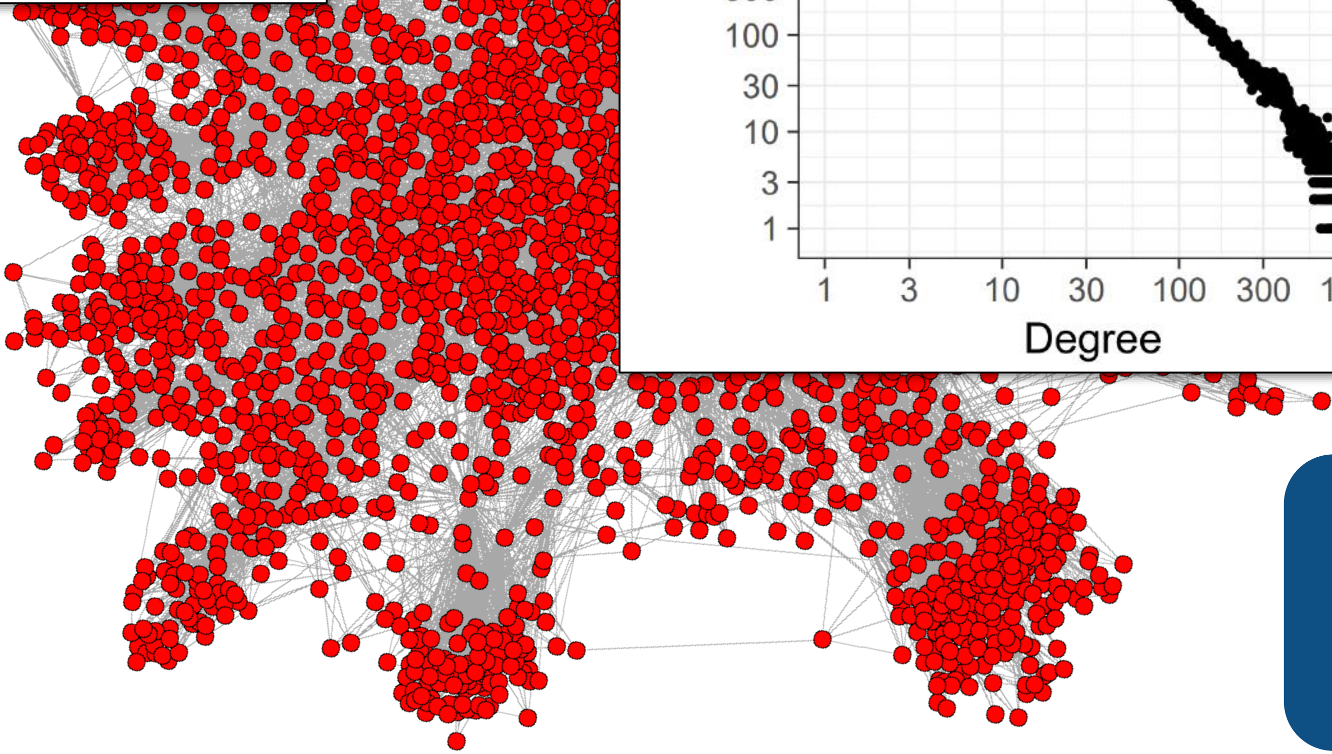


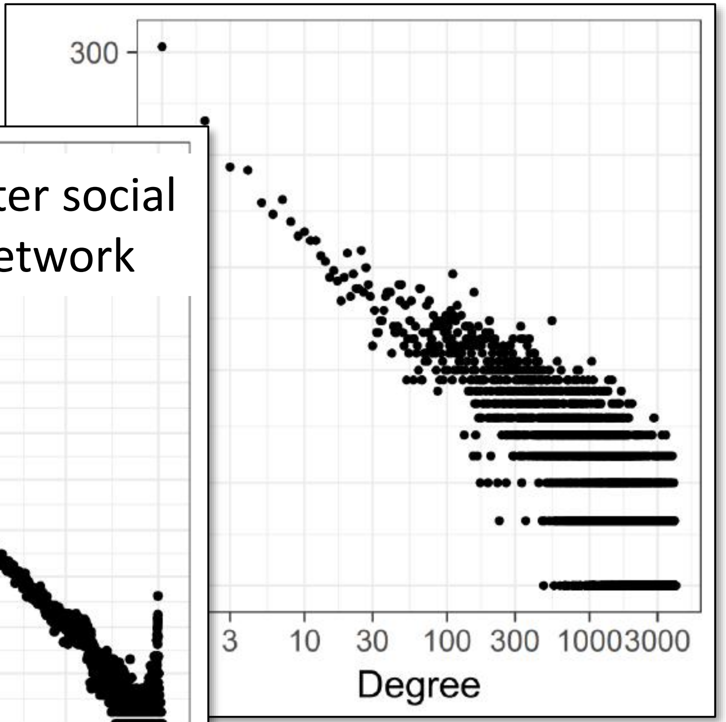


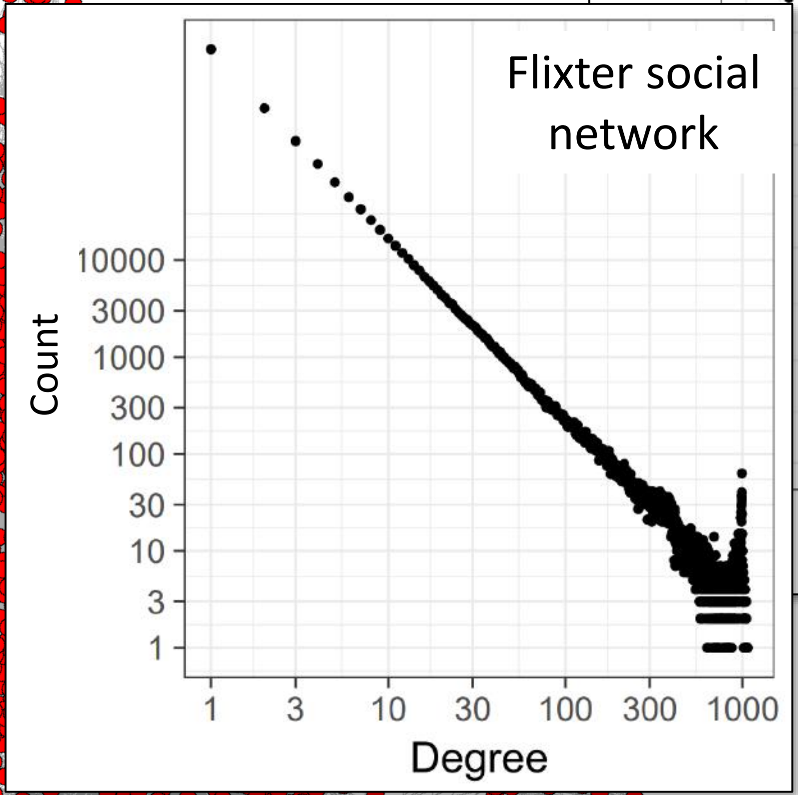
For example:
"Irregularity" of input datasets

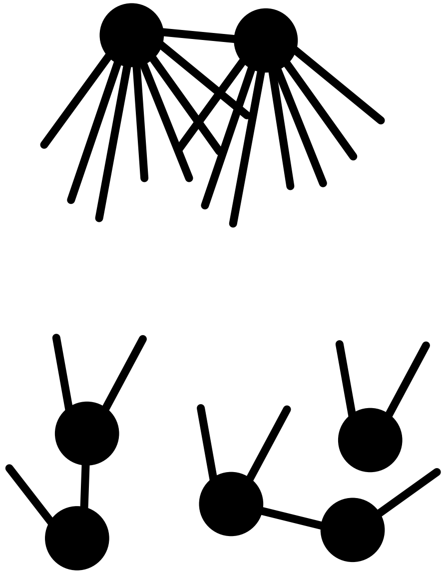
High-Performance Irregular Workloads





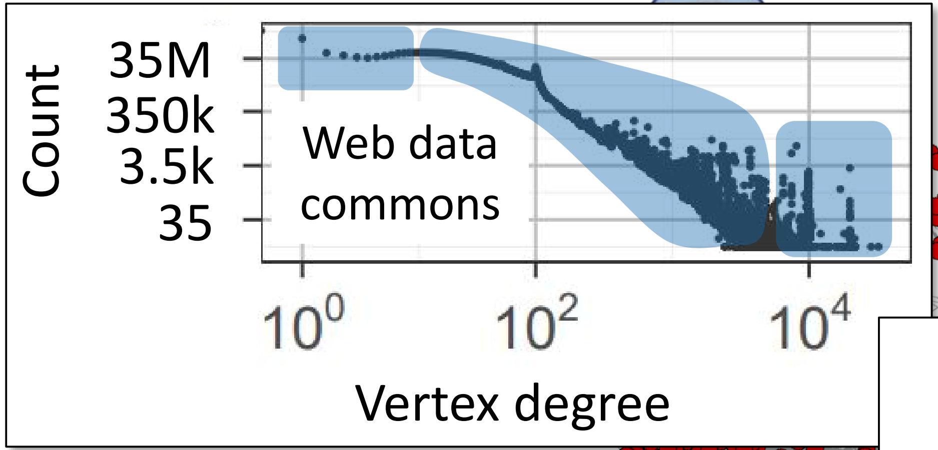


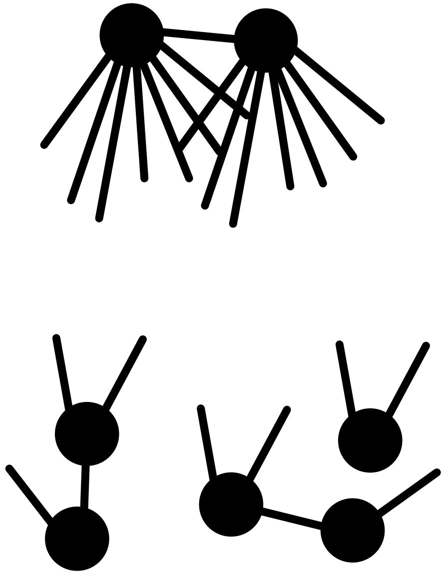


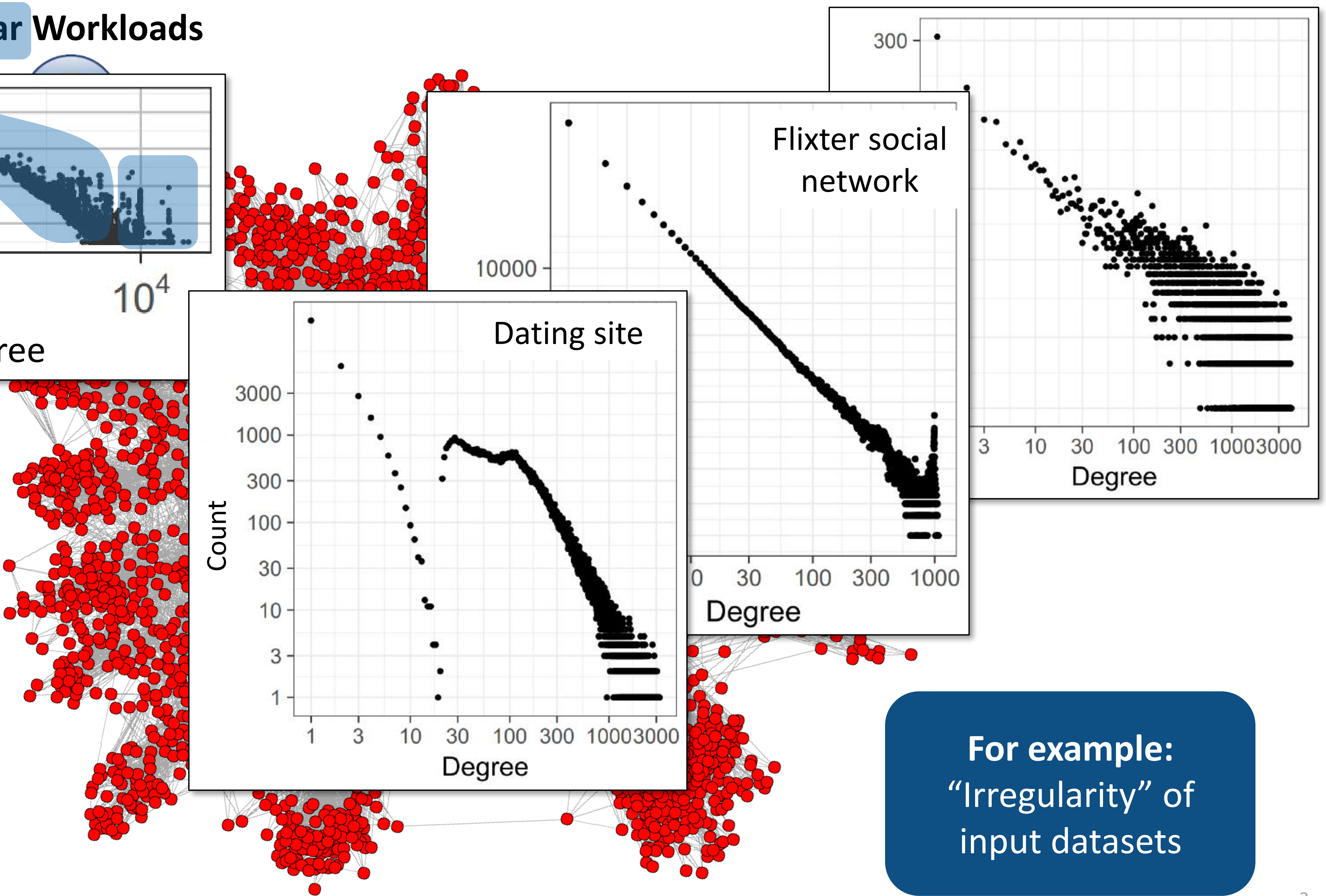


For example:
“Irregularity” of input datasets

High-Performance Irregular Workloads

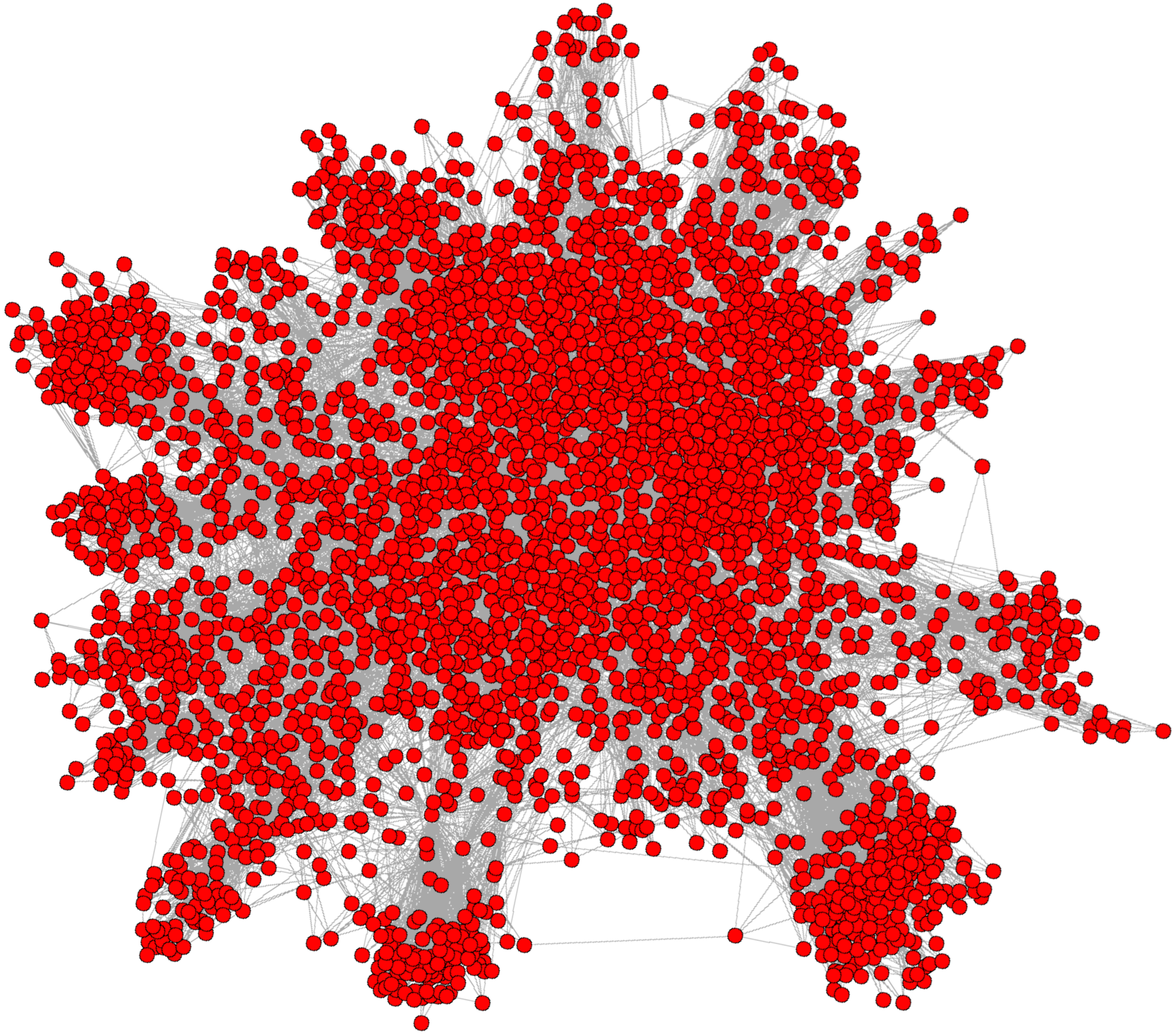




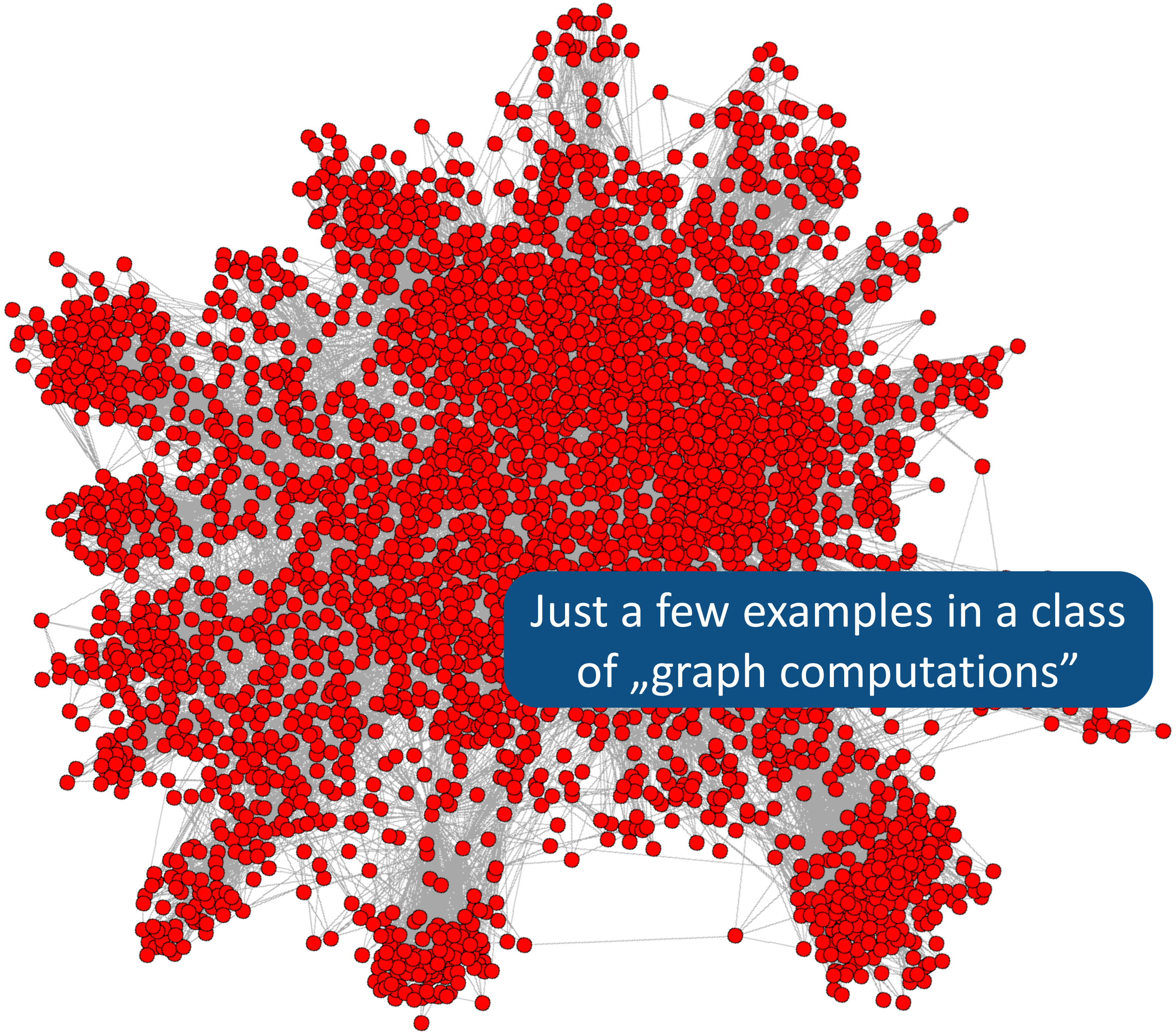


For example:
“Irregularity” of input datasets

High-Performance Irregular Workloads: Does Anyone Care?

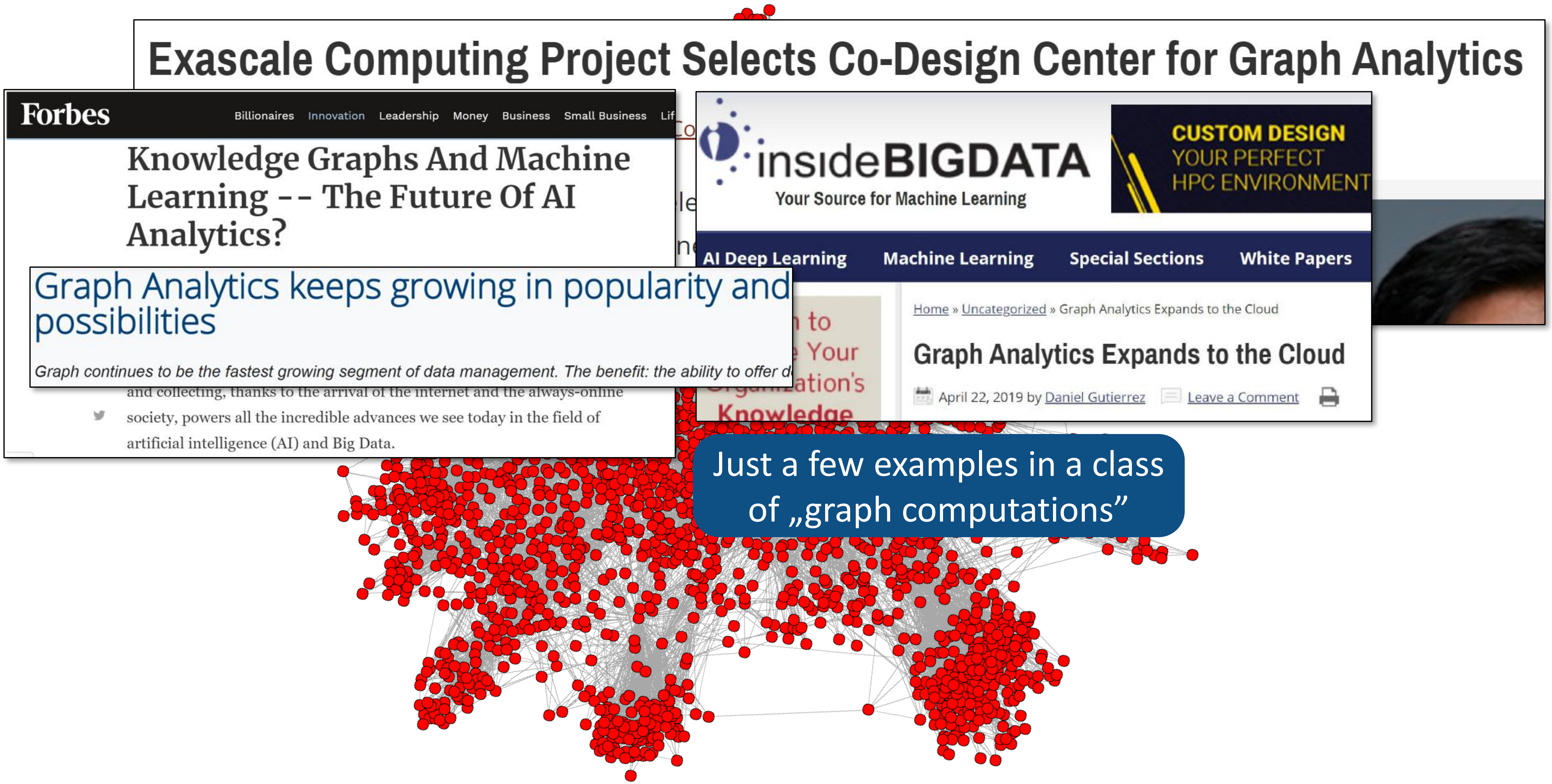


High-Performance Irregular Workloads: Does Anyone Care?



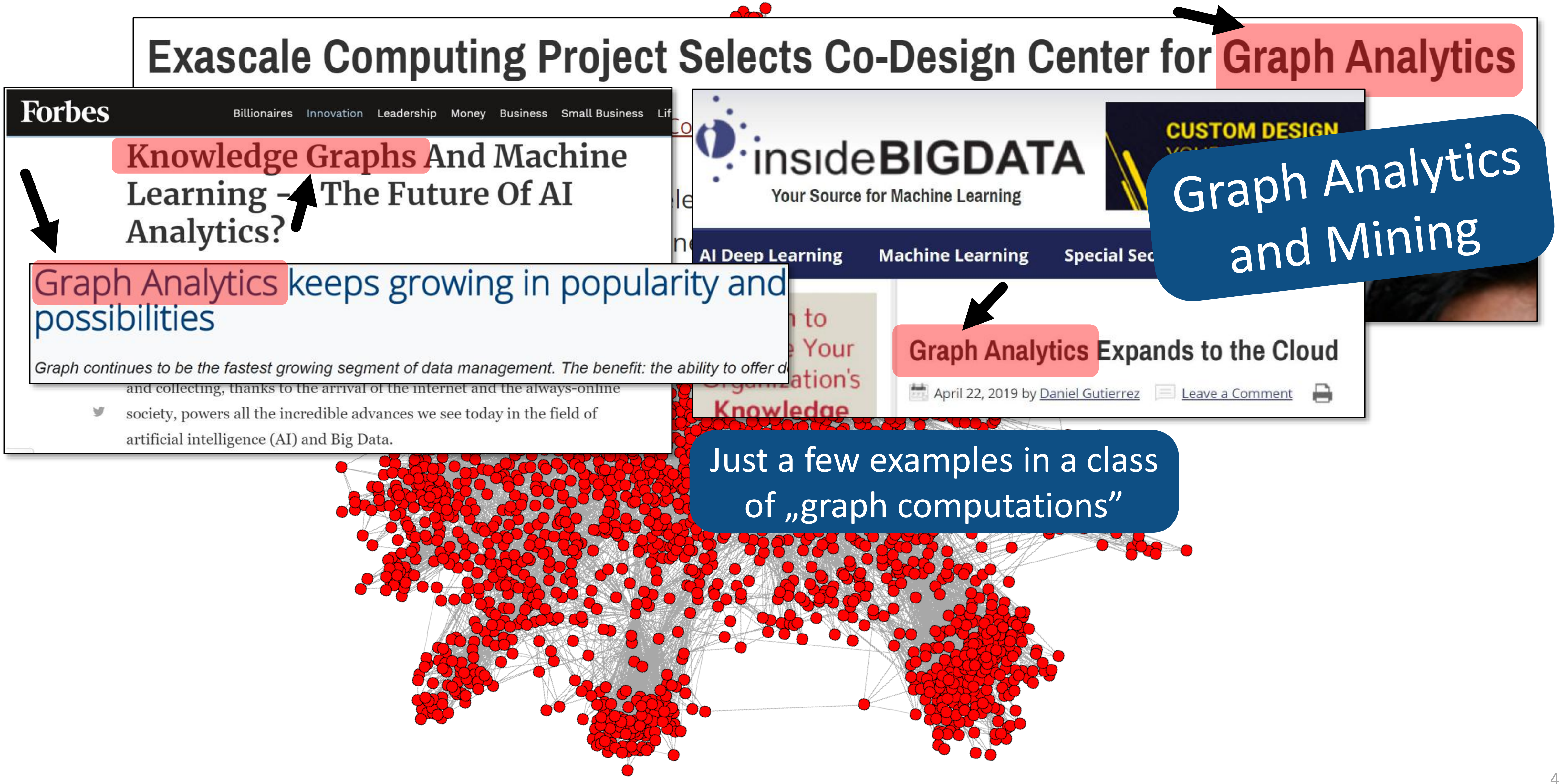
Just a few examples in a class of „graph computations"

High-Performance Irregular Workloads: Does Anyone Care?

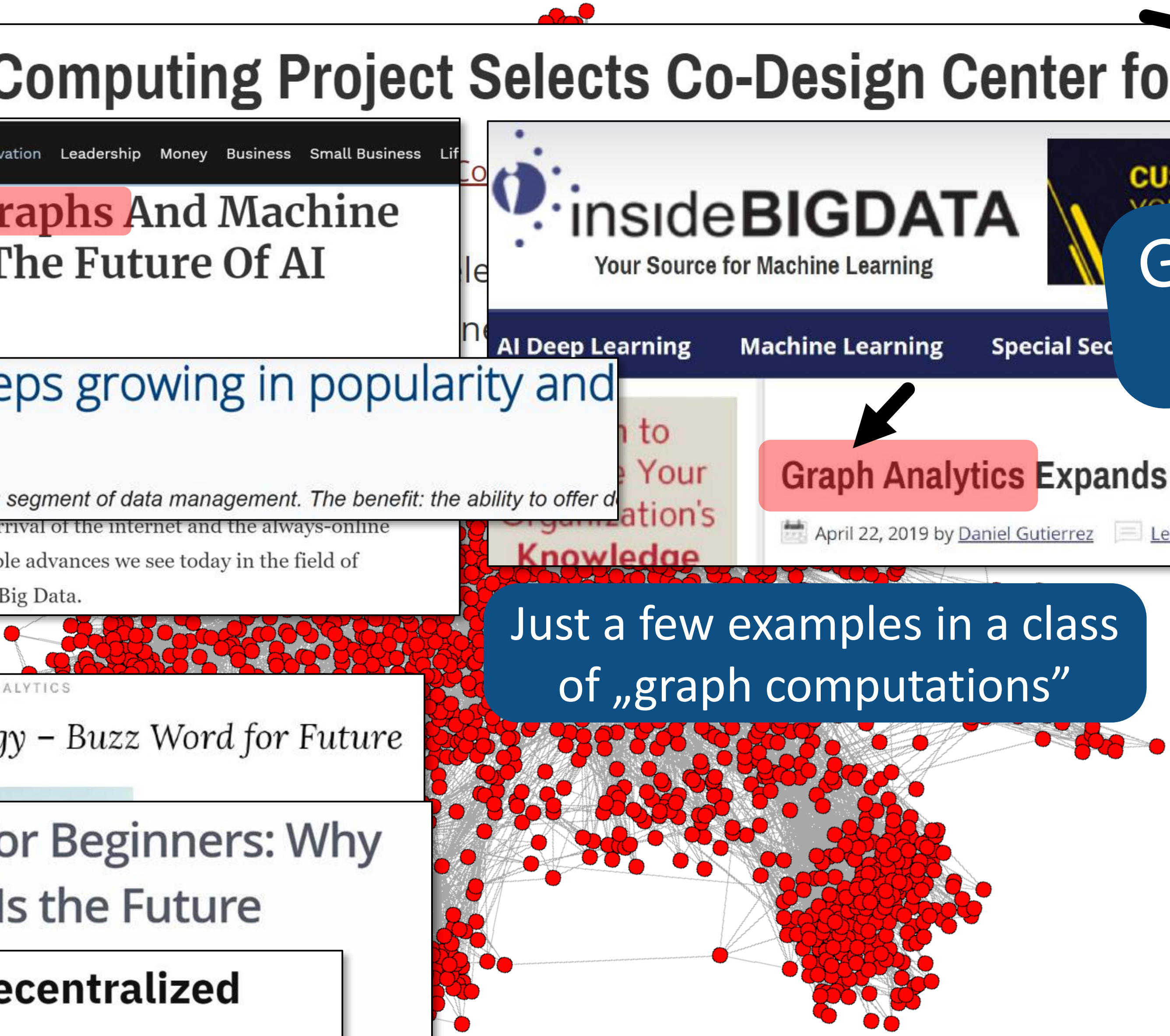


Just a few examples in a class of „graph computations”

High-Performance Irregular Workloads: Does Anyone Care?



High-Performance Irregular Workloads: Does Anyone Care?



Exascale Computing Project Selects Co-Design Center for Graph Analytics

Forbes Billionaires Innovation Leadership Money Business Small Business Lif

Knowledge Graphs And Machine Learning – The Future Of AI Analytics?

and collecting, thanks to the arrival of the internet and the always-online society, powers all the incredible advances we see today in the field of artificial intelligence (AI) and Big Data.

Graph Analytics keeps growing in popularity and possibilities

Graph continues to be the fastest growing segment of data management. The benefit: the ability to offer d

insideBIGDATA Your Source for Machine Learning

CUSTOM DESIGN

AI Deep Learning | Machine Learning | Special Sec

Graph Analytics Expands to the Cloud

April 22, 2019 by Daniel Gutierrez | Leave a Comment

Graph Analytics and Mining

Just a few examples in a class of „graph computations"

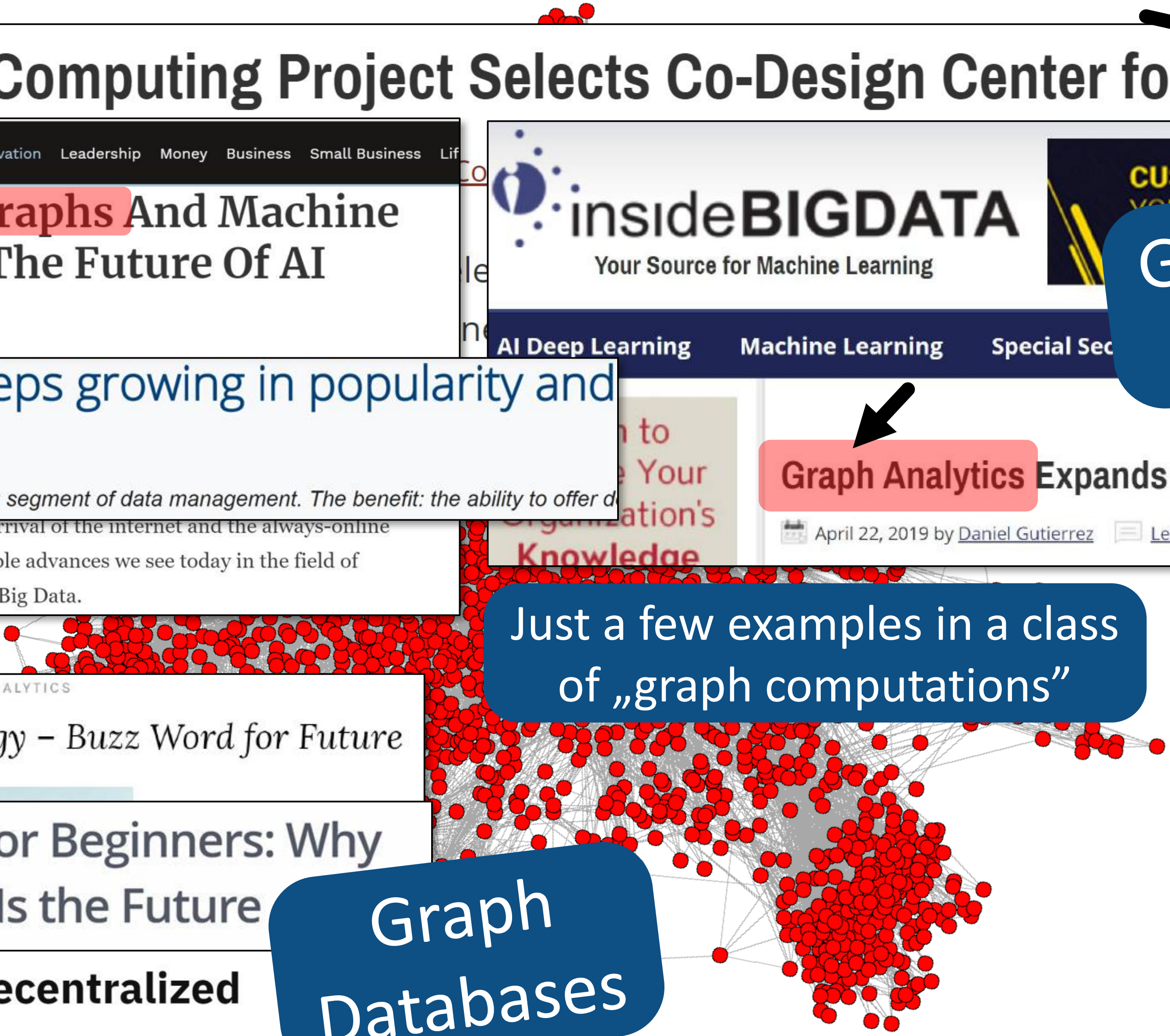
GRAPH ANALYTICS

Graph Database Technology – Buzz Word for Future

Graph Databases for Beginners: Why Graph Technology Is the Future

The Future of Data: A Decentralized Graph Database

High-Performance Irregular Workloads: Does Anyone Care?



Exascale Computing Project Selects Co-Design Center for Graph Analytics

Forbes
Billionaires Innovation Leadership Money Business Small Business Lif

Knowledge Graphs And Machine Learning -- The Future Of AI Analytics?

Graph Analytics keeps growing in popularity and possibilities

Graph continues to be the fastest growing segment of data management. The benefit: the ability to offer d

and collecting, thanks to the arrival of the internet and the always-online society, powers all the incredible advances we see today in the field of artificial intelligence (AI) and Big Data.

insideBIGDATA
Your Source for Machine Learning
CUSTOM DESIGN
AI Deep Learning Machine Learning Special Sec

Graph Analytics Expands to the Cloud

April 22, 2019 by Daniel Gutierrez Leave a Comment

Graph Analytics and Mining

Just a few examples in a class of „graph computations"

GRAPH ANALYTICS

Graph Database Technology – Buzz Word for Future

Graph Databases for Beginners: Why Graph Technology Is the Future

The Future of Data: A Decentralized Graph Database

Graph Databases

High-Performance Irregular Workloads: Does Anyone Care?

Exascale Computing Project Selects Co-Design Center for Graph Analytics

Forbes — Billionaires, Innovation, Leadership, Money, Business, Small Business, Lif

Knowledge Graphs And Machine Learning – The Future Of AI Analytics?

and collecting, thanks to the arrival of the internet and the always-online society, powers all the incredible advances we see today in the field of artificial intelligence (AI) and Big Data.

Graph Analytics keeps growing in popularity and possibilities

Graph continues to be the fastest growing segment of data management. The benefit: the ability to offer d

insideBIGDATA — Your Source for Machine Learning

CUSTOM DESIGN

AI Deep Learning | Machine Learning | Special Sec

Graph Analytics Expands to the Cloud

April 22, 2019 by Daniel Gutierrez — Leave a Comment

Graph Analytics and Mining

Just a few examples in a class of „graph computations"

GRAPH ANALYTICS

Graph Database Technology – Buzz Word for Future

Graph Databases for Beginners: Why Graph Technology Is the Future

The Future of Data: A Decentralized Graph Database

Graph Databases

Home › Architectures › DARPA ERI: HIVE and Intel PUMA Graph Processor

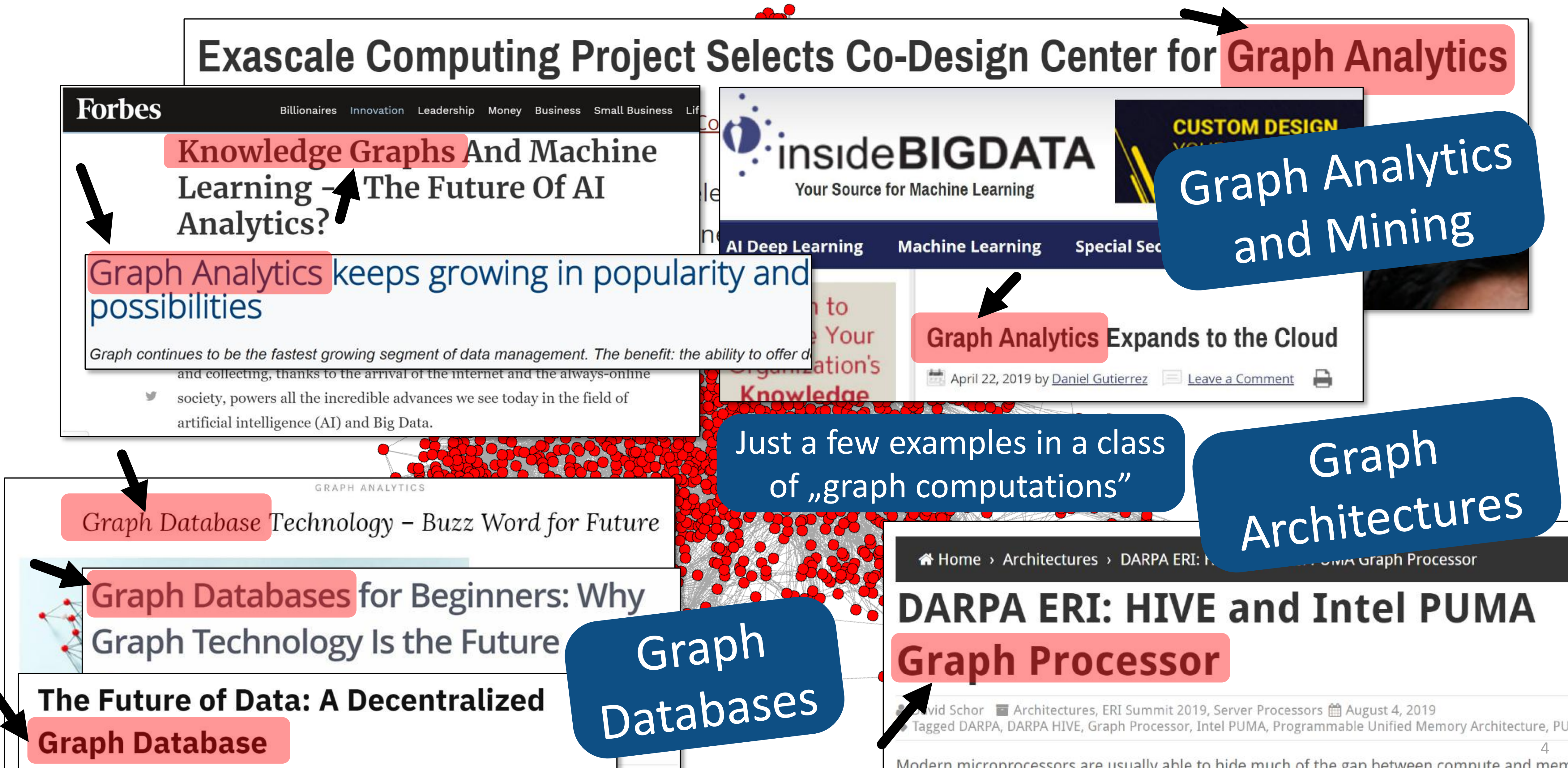
DARPA ERI: HIVE and Intel PUMA Graph Processor

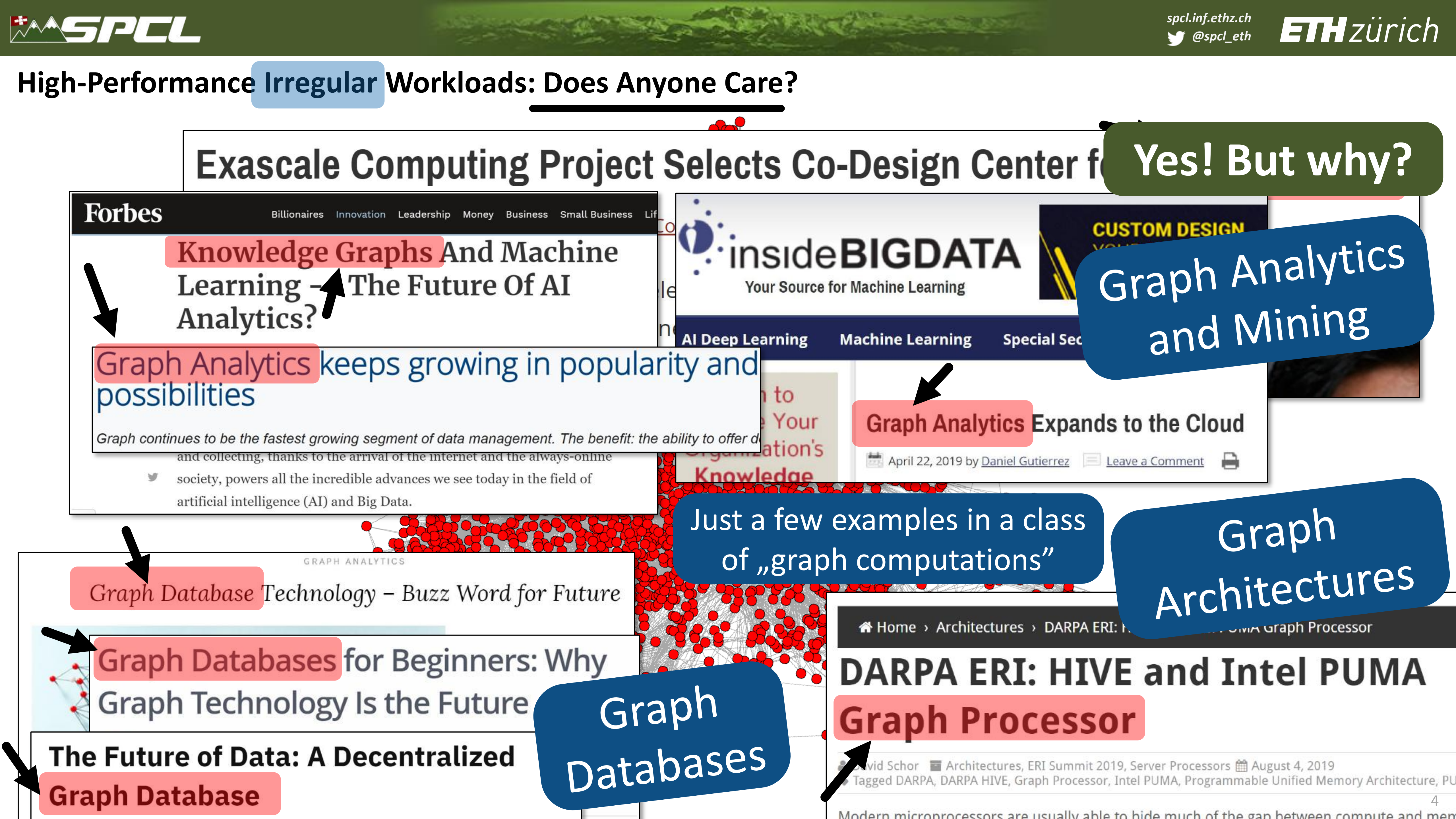
David Schor — Architectures, ERI Summit 2019, Server Processors — August 4, 2019

Tagged DARPA, DARPA HIVE, Graph Processor, Intel PUMA, Programmable Unified Memory Architecture, PU

Modern microprocessors are usually able to hide much of the gap between compute and mem

High-Performance Irregular Workloads: Does Anyone Care?



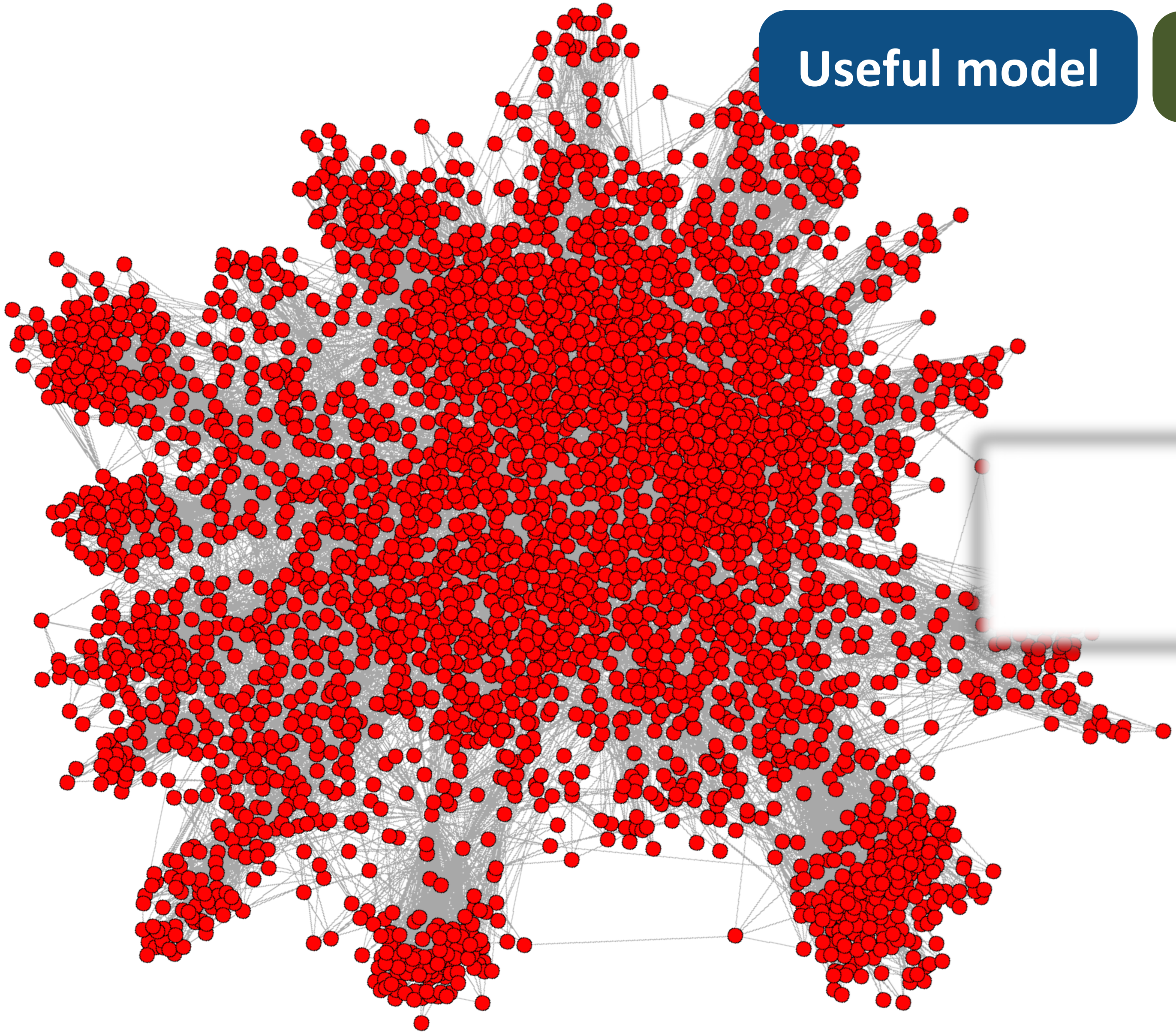


SPCL
spcl.inf.ethz.ch
@spcl_eth
ETH zürich
High-Performance Irregular Workloads: Does Anyone Care?
Yes! But why?
Exascale Computing Project Selects Co-Design Center f
Forbes
Billionaires Innovation Leadership Money Business Small Business
Knowledge Graphs And Machine Learning – The Future Of AI Analytics?
Graph Analytics keeps growing in popularity and possibilities
Graph continues to be the fastest growing segment of data management. The benefit: the ability to offer d
and collecting, thanks to the arrival of the internet and the always-online society, powers all the incredible advances we see today in the field of artificial intelligence (AI) and Big Data.
insideBIGDATA
Your Source for Machine Learning
CUSTOM DESIGN
AI Deep Learning
Machine Learning
Graph Analytics Expands to the Cloud
April 22, 2019 by Daniel Gutierrez
Leave a Comment
Graph Analytics and Mining
Just a few examples in a class of „graph computations"
Graph Architectures
GRAPH ANALYTICS
Graph Database Technology – Buzz Word for Future
Graph Databases for Beginners: Why Graph Technology Is the Future
The Future of Data: A Decentralized Graph Database
Graph Databases
Home › Architectures › DARPA ERI:
DARPA ERI: HIVE and Intel PUMA Graph Processor
Architectures, ERI Summit 2019, Server Processors
August 4, 2019
Tagged DARPA, DARPA HIVE, Graph Processor, Intel PUMA, Programmable Unified Memory Architecture

High-Performance Irregular Workloads: Does Anyone Care?

Useful model

Yes! But why?

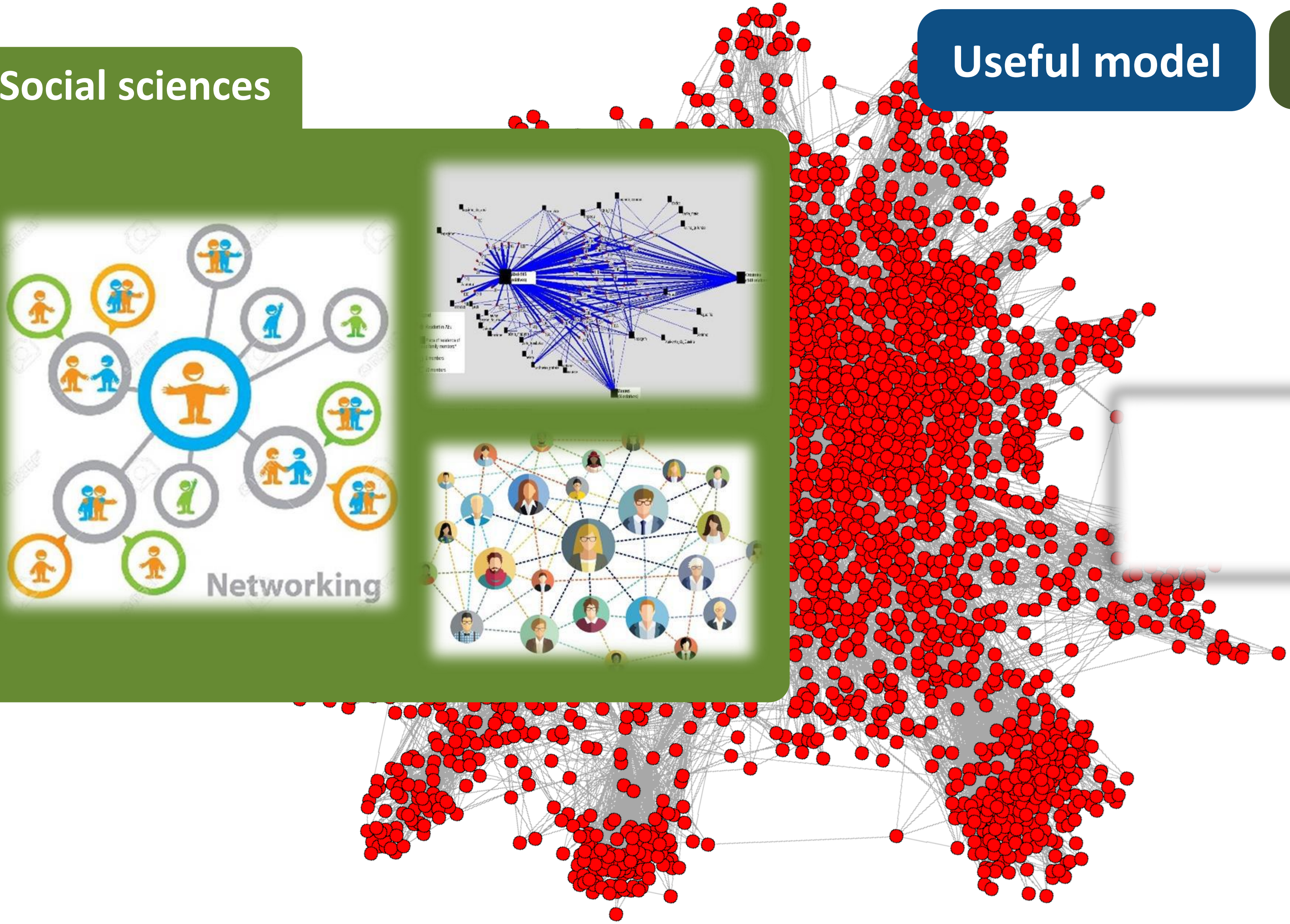


High-Performance Irregular Workloads: Does Anyone Care?

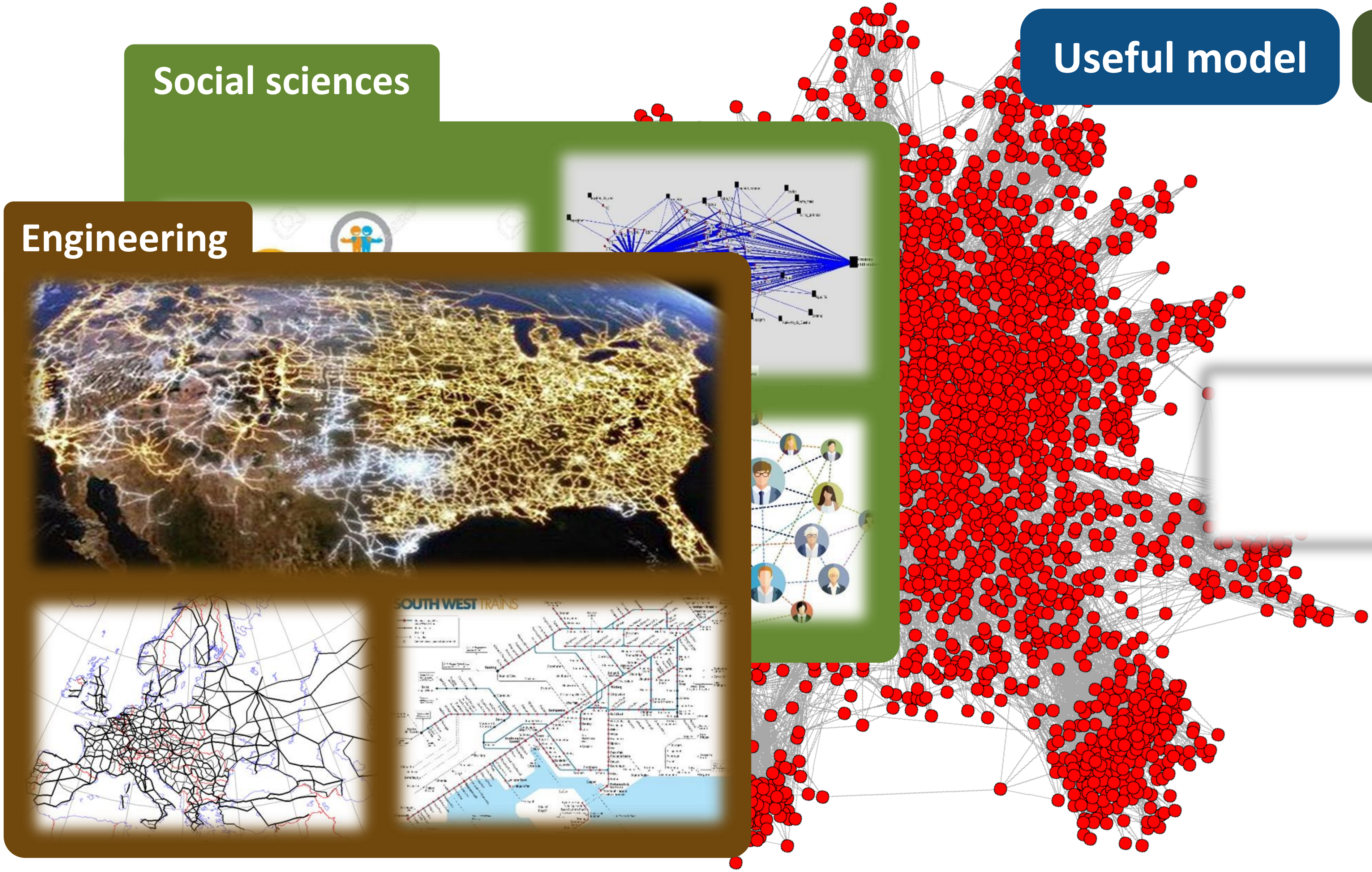
Useful model

Yes! But why?

Social sciences



High-Performance Irregular Workloads: Does Anyone Care?



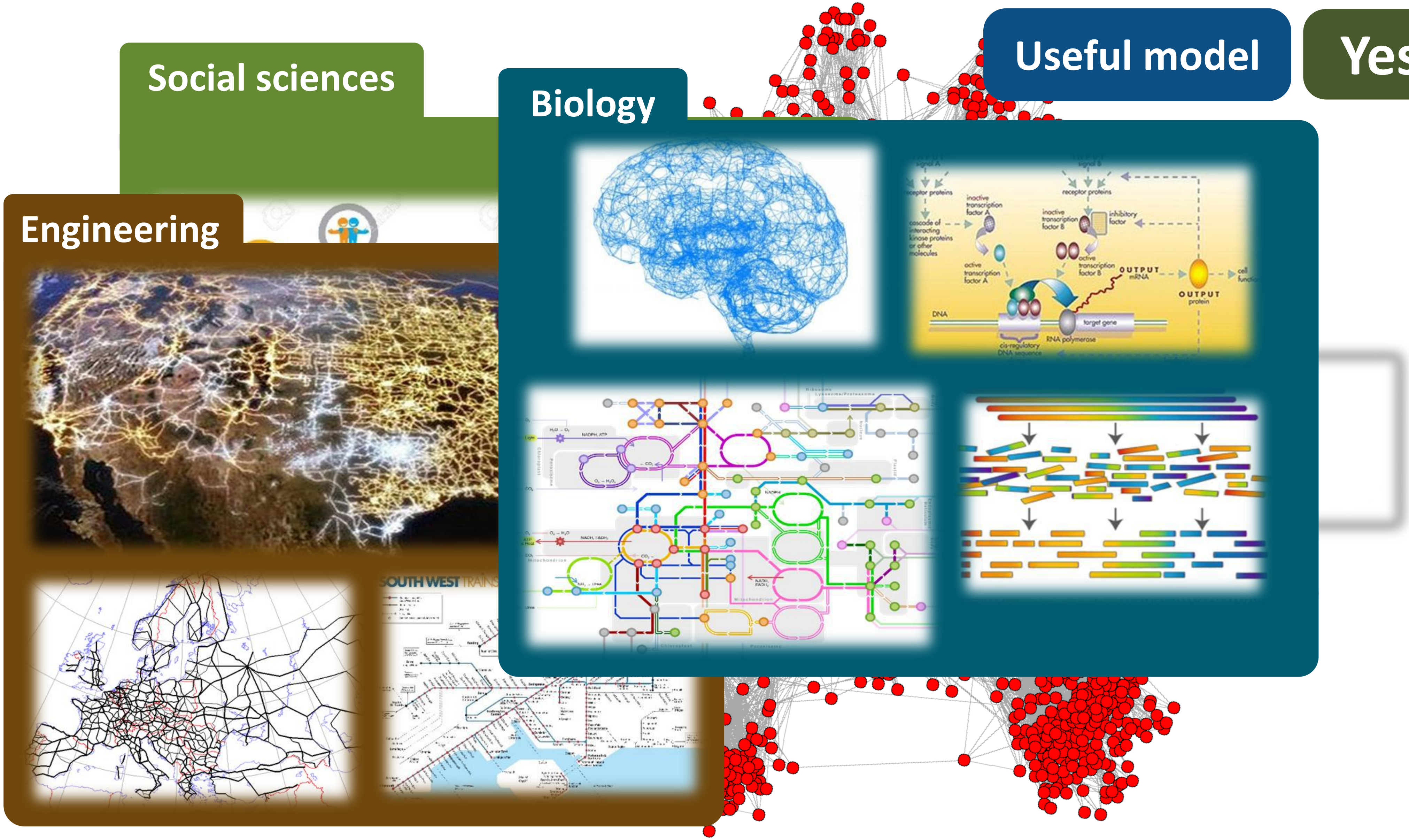
Useful model

Yes! But why?

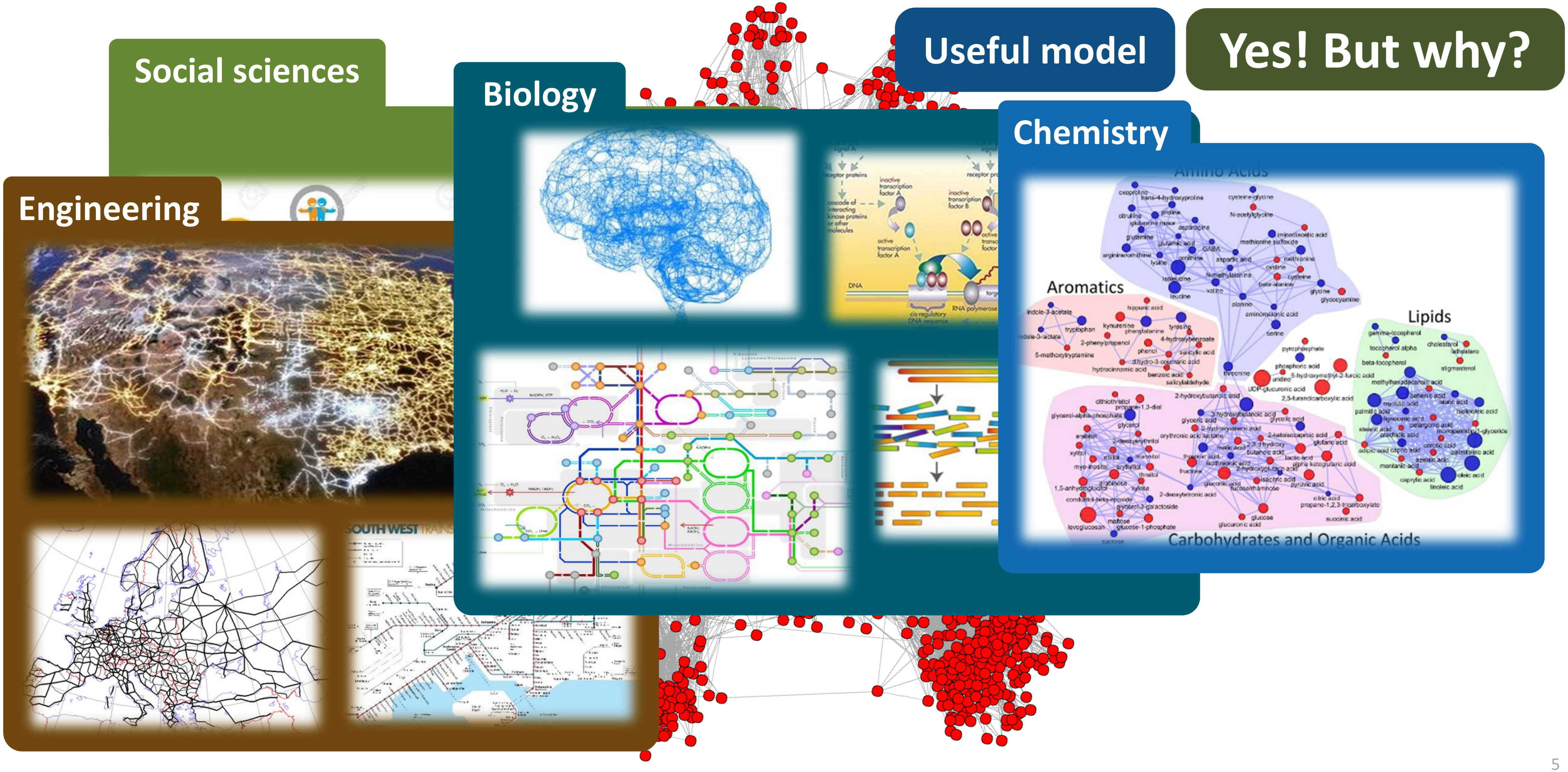
High-Performance Irregular Workloads: Does Anyone Care?

Useful model

Yes! But why?



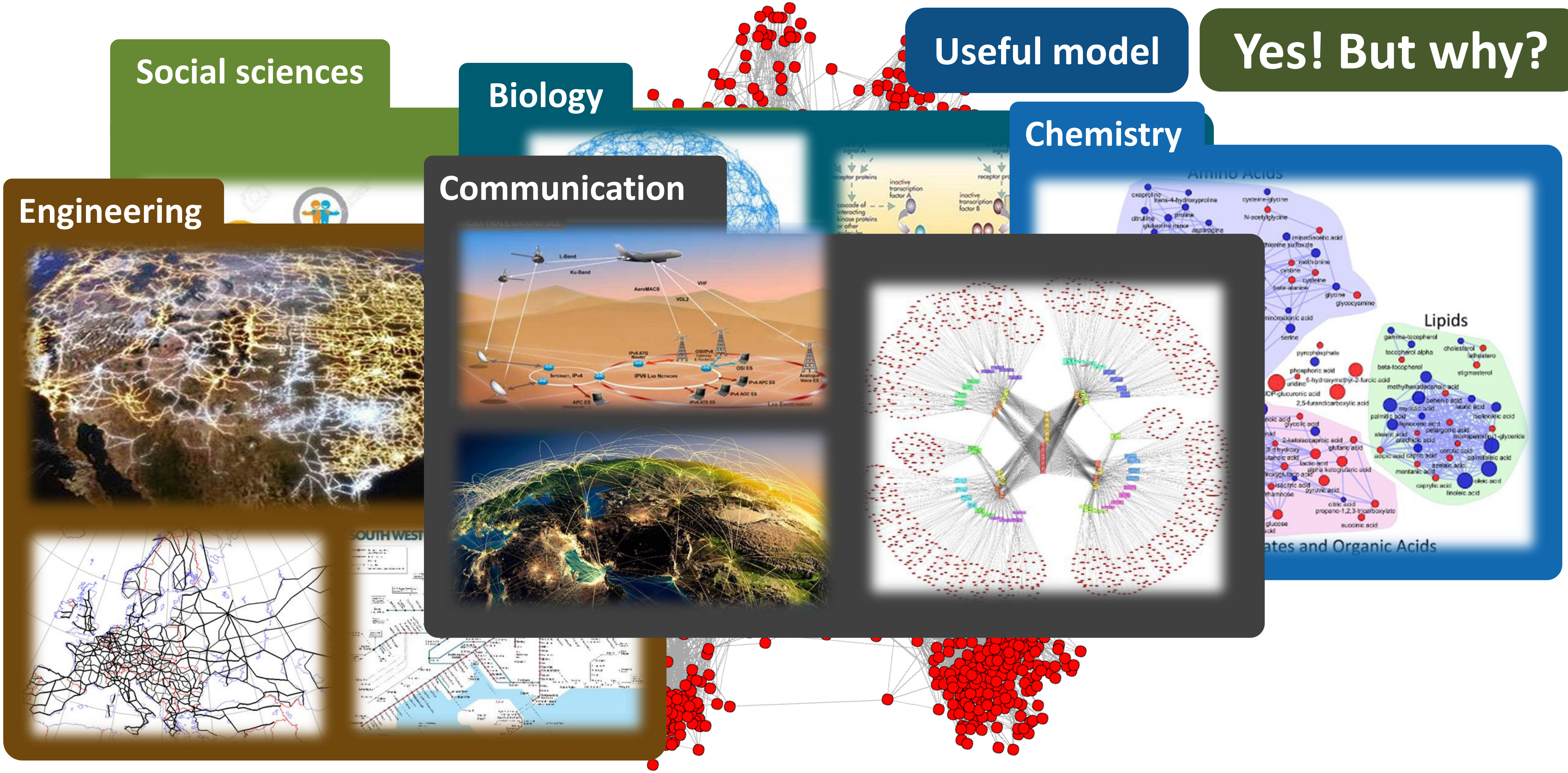
High-Performance Irregular Workloads: Does Anyone Care?



High-Performance Irregular Workloads: Does Anyone Care?

Useful model

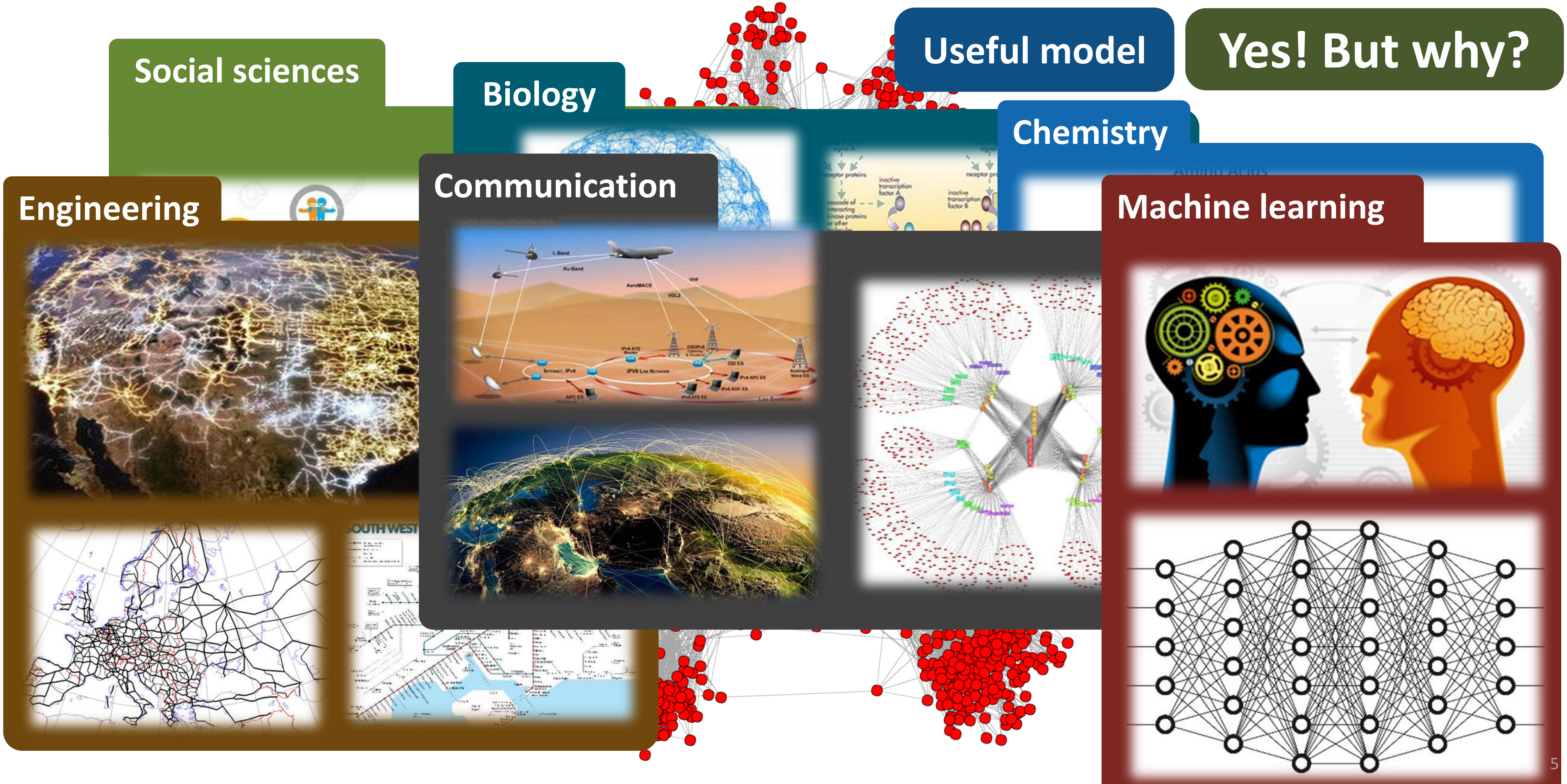
Yes! But why?



High-Performance Irregular Workloads: Does Anyone Care?

Useful model

Yes! But why?



High-Performance Irregular Workloads: Does Anyone Care?

Useful model

Yes! But why?

Social sciences

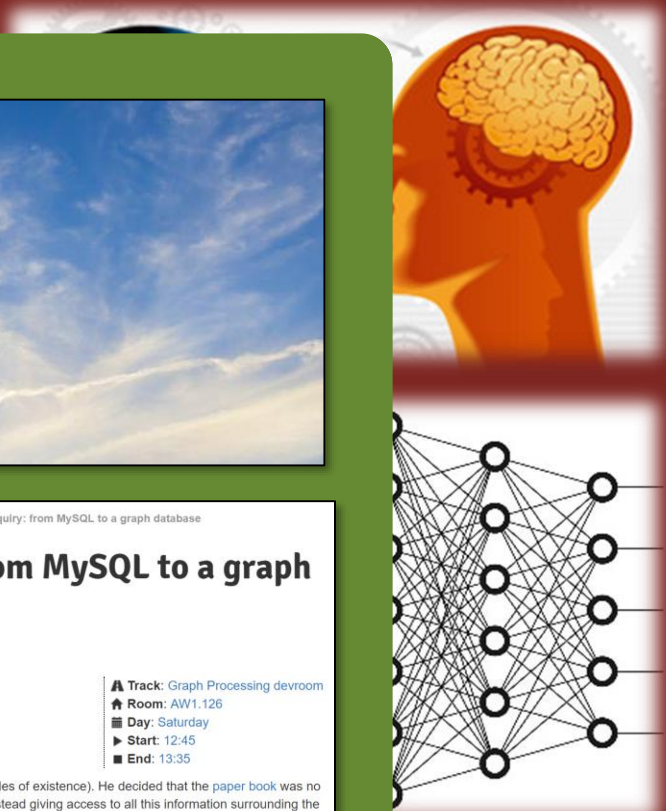
Biology

Chemistry

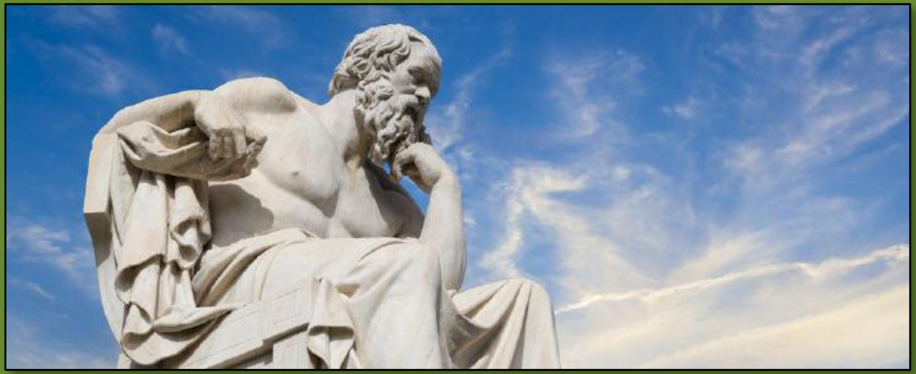
Engineering

Communication

Machine learning



...even philosophy ☺



FOSDEM 2016 / Schedule / Events / Developer rooms / Graph Processing / Modeling a Philosophical Inquiry: from MySQL to a graph database

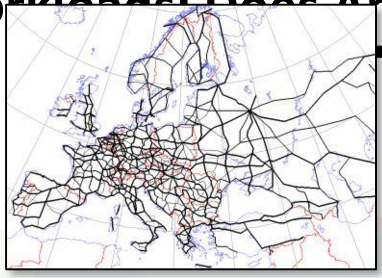
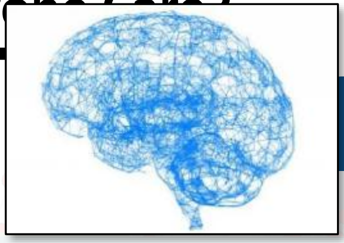
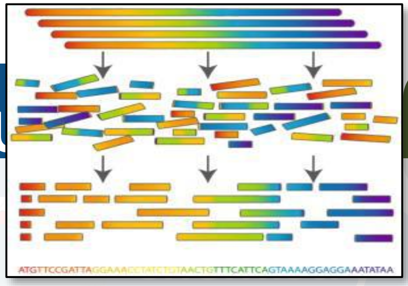
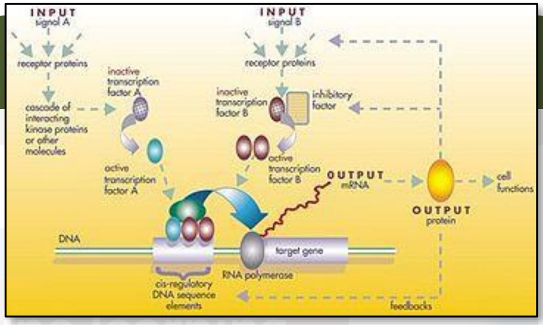
Modeling a Philosophical Inquiry: from MySQL to a graph database

The short story of a long refactoring process

Track: Graph Processing devroom
Room: AW1.126
Day: Saturday
Start: 12:45
End: 13:35

Bruno Latour wrote a book about philosophy (an inquiry into modes of existence). He decided that the paper book was no place for the numerous footnotes, documentation or glossary, instead giving access to all this information surrounding the book through a web application which would present itself as a reading companion. He also offered to the community of

High-Performance Irregular Workloads: Does Anyone Care?

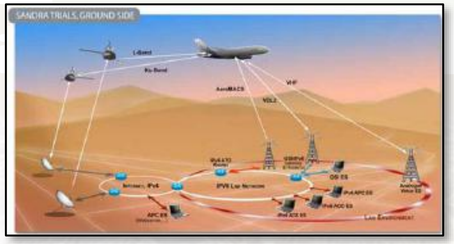






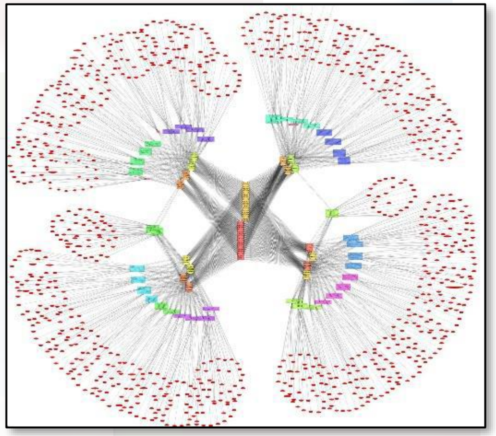
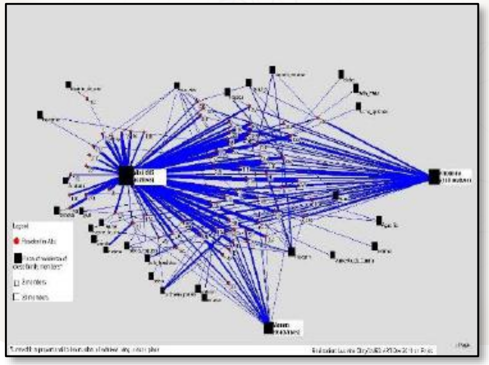
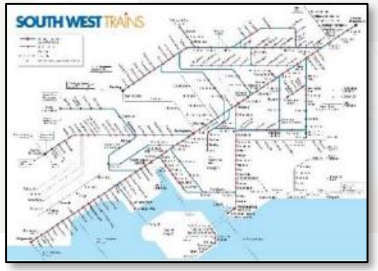
Engi…

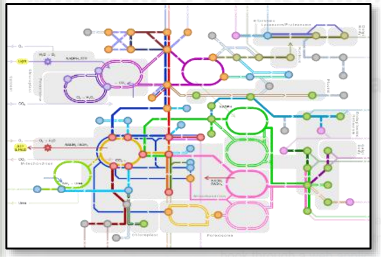
Communication

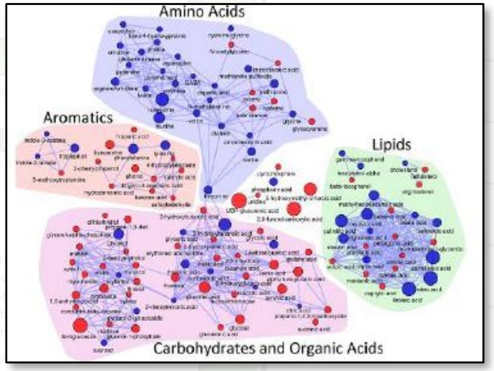
…even philosophy ☺

Machine learning

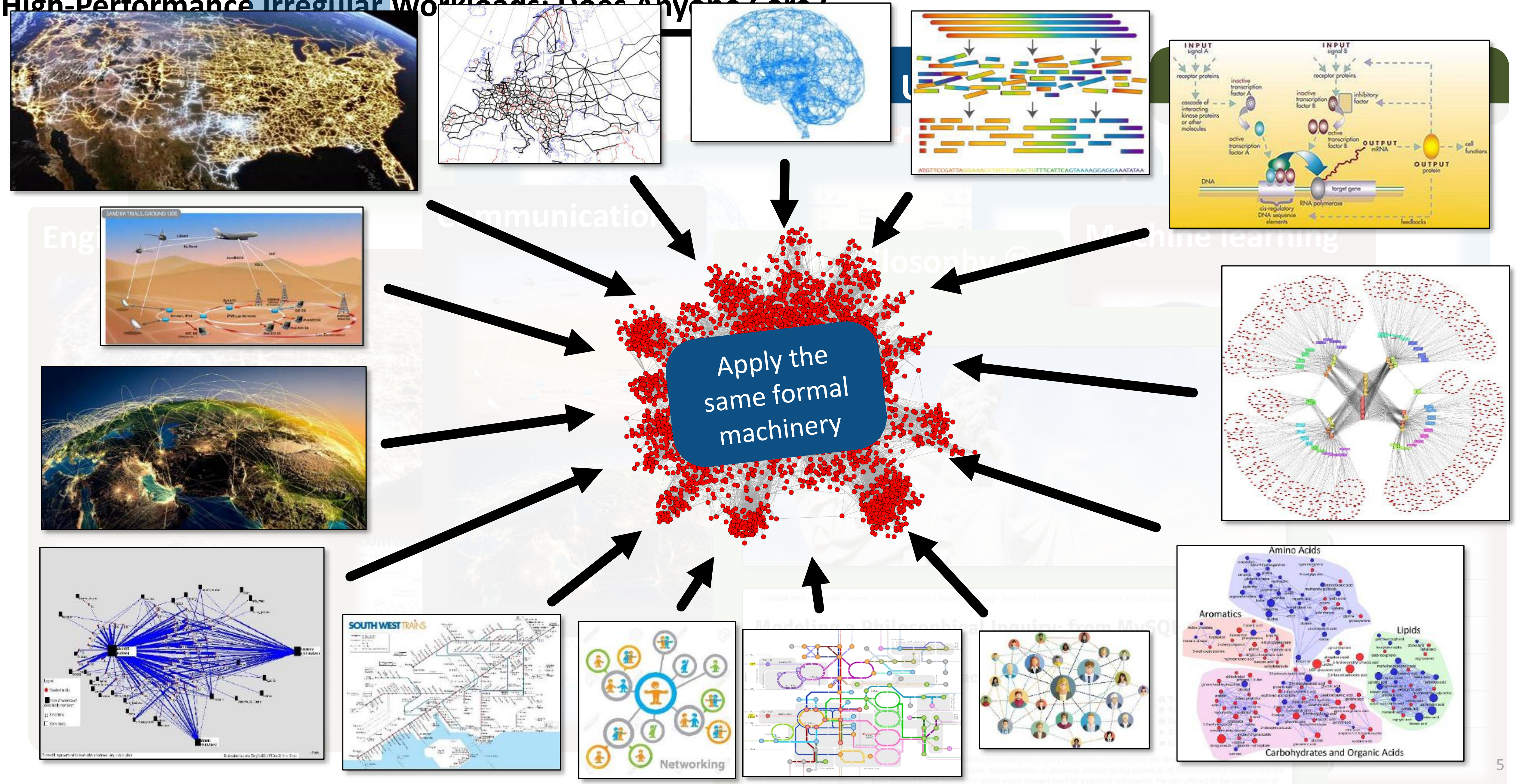






High-Performance Irregular Workloads: Does Anyone Care?



Overview of My Research (Perspective of Compute Stack Layers)

Overview of My Research (Perspective of Compute Stack Layers)

High

"Level of abstraction"

Low

Overview of My Research (Perspective of Compute Stack Layers)

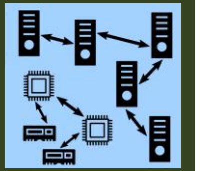
Overview of My Research (Perspective of Compute Stack Layers)

High

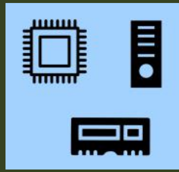
"Level of abstraction"

Low

Networking



Hardware



Overview of My Research (Perspective of Compute Stack Layers)

High

"Level of abstraction"

Low

Middleware, Frameworks, Runtimes, OS

Networking

Hardware

Overview of My Research (Perspective of Compute Stack Layers)

High

"Level of abstraction"

Low

Algorithms

Middleware, Frameworks, Runtimes, OS

Networking

Hardware

Overview of My Research (Perspective of Compute Stack Layers)

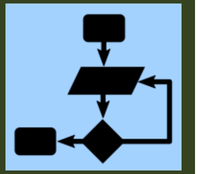
High

"Level of abstraction"

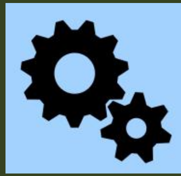
Low

Paradigms, Abstractions, Programming Models

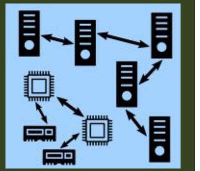
Algorithms



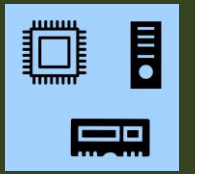
Middleware, Frameworks, Runtimes, OS



Networking



Hardware



Overview of My Research (Perspective of Compute Stack Layers)

Many projects are touching on multiple levels. For clarity, we assign one project to one most related (roughly) level

High

"Level of abstraction"

Low

Paradigms, Abstractions, Programming Models

[SC13] [HPDC'14] [CACM'18] [SC19a] [arXiv'21c] [PADAL'14] [KDD'22]

Algorithms

[PACT'18] [IPDPS'20] [SC20a] [IPDPS'22b] [PPoPP'18] [SPAA'21a] [SPAA'21b]
[HPDC'16] [SC17] [IPDPS'17] [HPDC'17] [arXiv'18] [SC19c] [arXiv'20] [PPoPP'21] [arXiv'22]

Middleware, Frameworks, Runtimes, OS

[IPDPS'22a] [CACM'21]
[HPDC'15] [ICS'15] [arXiv'19] [IPDPS'19] [Middleware'21] [VLDB'21] [TPDS'22]

Networking

[SC14] [Scientific'14] [ASPLOS'18] [SC20b] [arXiv'21b] [TPDS'21a]

Hardware

[PACT'15] [FPGA'19] [arXiv'19] [TRETS'20] [TPDS'21b] [SC19b] [MICRO'21]

Overview of My Research (Perspective of Compute Stack Layers)

Many projects are touching on multiple levels. For clarity, we assign one project to one most related (roughly) level

High

"Level of abstraction"

Low

Paradigms, Abstractions, Programming Models

[SC13] [HPDC'14] [CACM'18] [SC19a] [arXiv'21c] [PADAL'14] [KDD'22]

Algorithms

[PACT'18] [IPDPS'20] [SC20a] [IPDPS'22b] [PPoPP'18] [SPAA'21a] [SPAA'21b]

[HPDC'16] [SC17] [IPDPS'17] [HPDC'17] [arXiv'18] [SC19c] [arXiv'20] [PPoPP'21] [arXiv'22]

Middleware, Frameworks, Runtimes, OS

[IPDPS'22a] [CACM'21]

[HPDC'15] [ICS'15] [arXiv'19] [IPDPS'19] [Middleware'21] [VLDB'21] [TPDS'22]

1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

Overview of My Research (Perspective of Compute Stack Layers)

Many projects are touching on multiple levels. For clarity, we assign one project to one most related (roughly) level

High

"Level of abstraction"

Low

Paradigms, Abstractions, Programming Models

[SC13] [HPDC'14] [CACM'18] [SC19a] [arXiv'21c] [PADAL'14] [KDD'22]

1a] [SPAA'21b]

PoPP'21 [arXiv'22]

PDS'22]

Motif Prediction with Graph Neural Networks

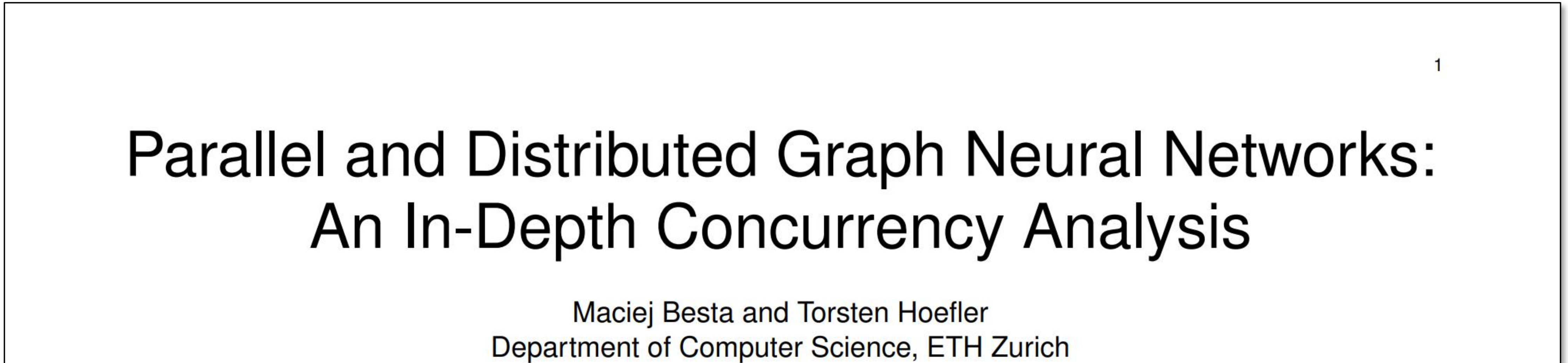
Maciej Besta[1†], Raphael Grob[1], Cesare Miglioli[2], Nicola Bernold[1], Grzegorz Kwasniewski[1], Gabriel Gjini[1], Raghavendra Kanakagiri[3], Saleh Ashkboos[1], Lukas Gianinazzi[1], Nikoli Dryden[1], Torsten Hoefler[1†]

1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

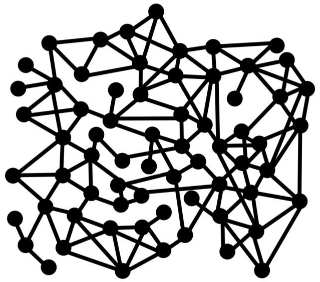
Presentation Overview

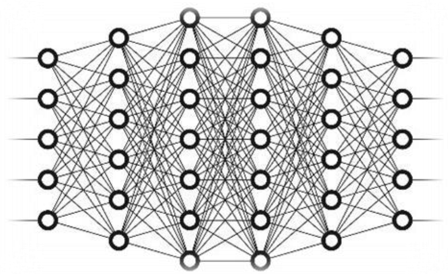

1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

Presentation Overview



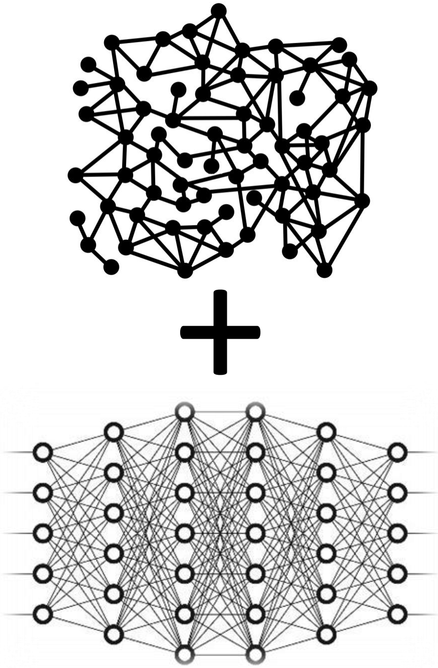


1

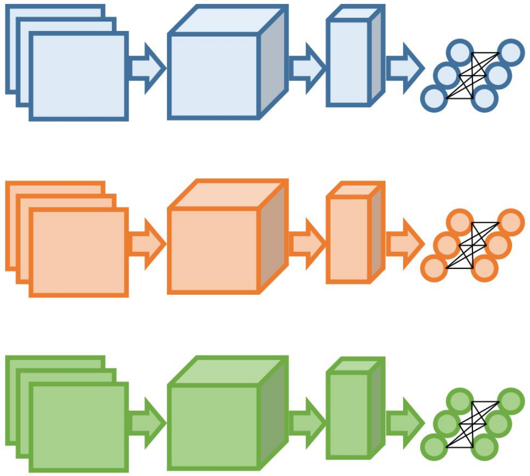
Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

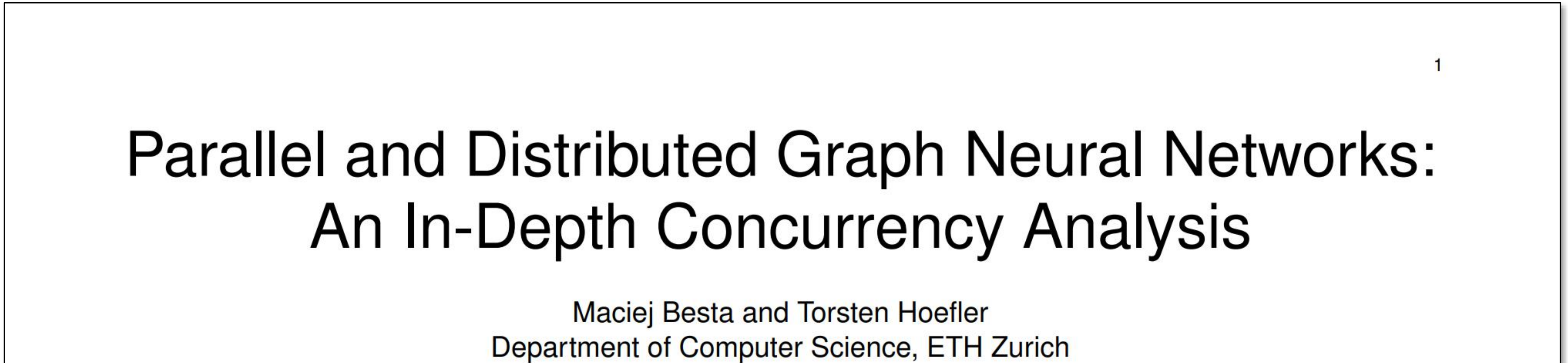
Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

Presentation Overview



Data parallelism



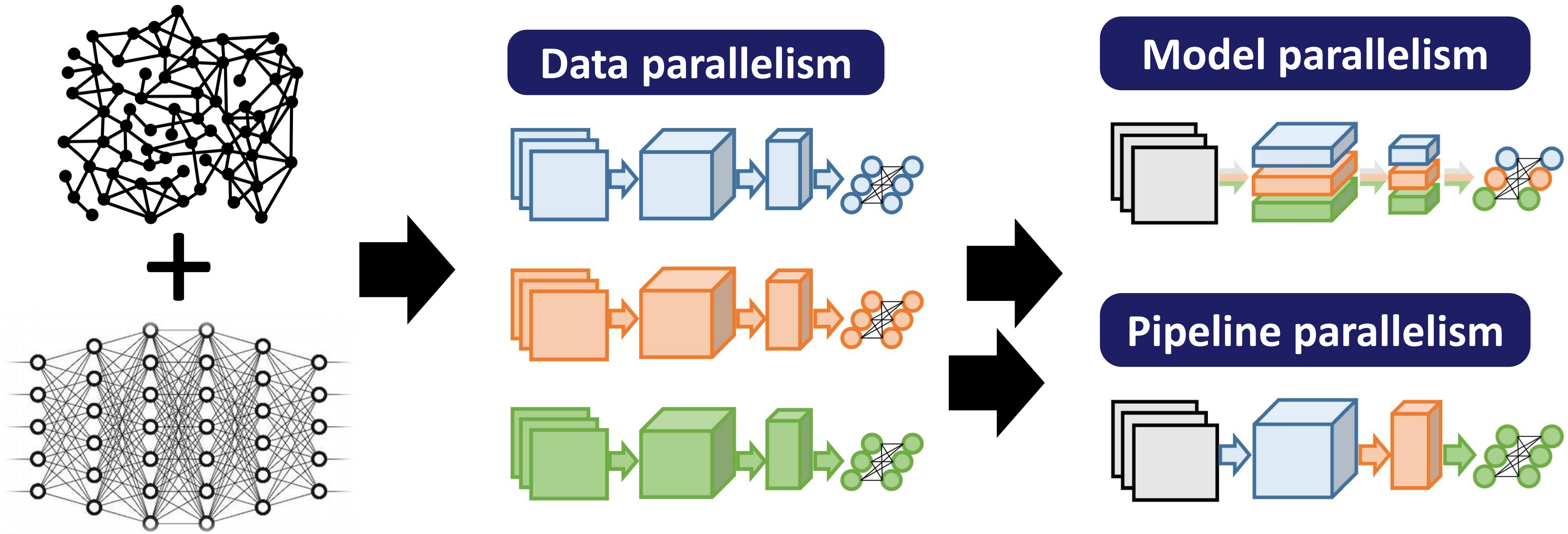


1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

Presentation Overview

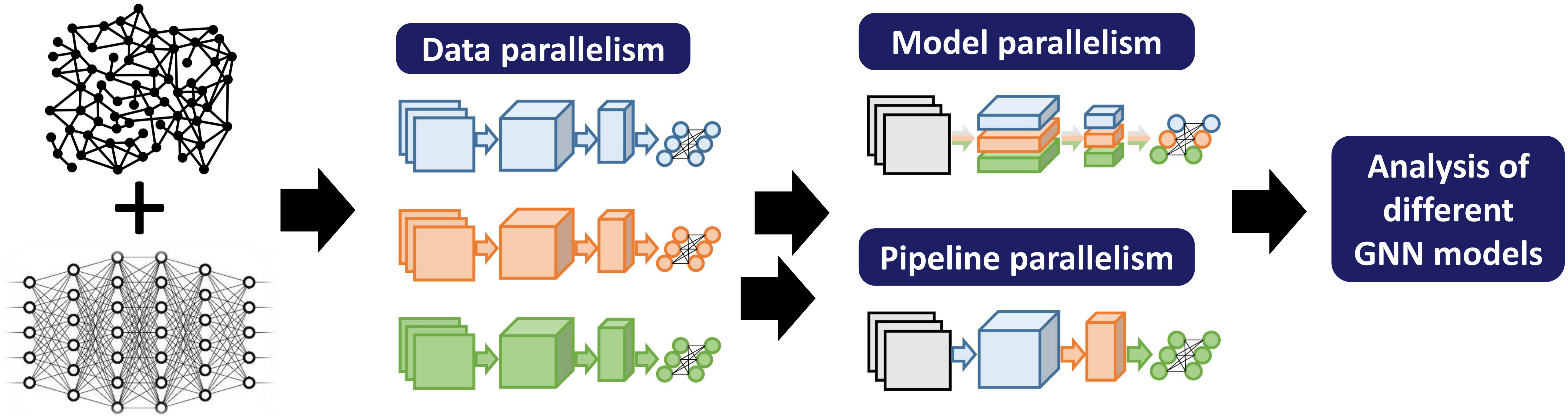


1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

Presentation Overview

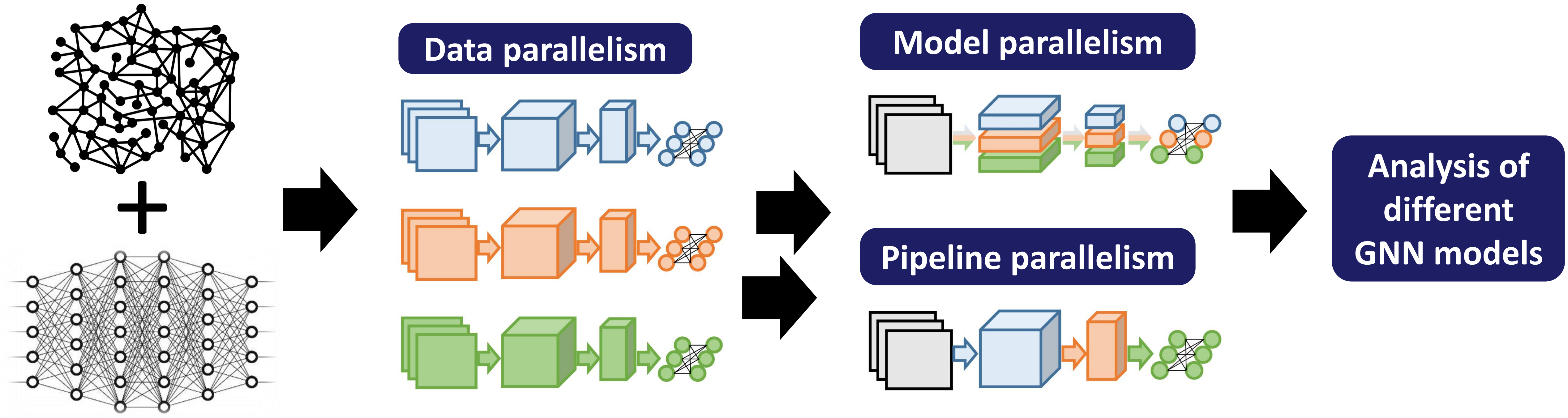


1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

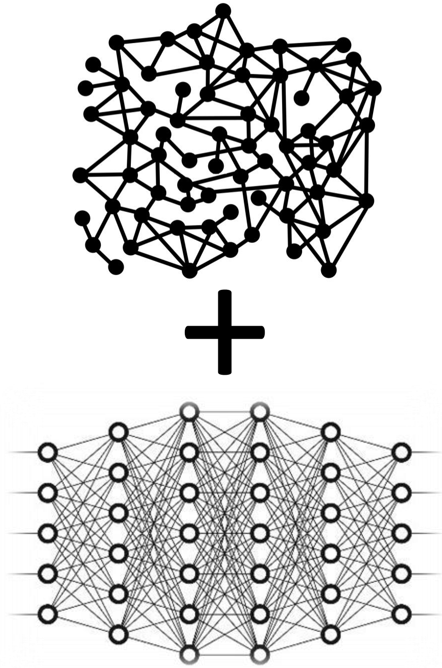
Presentation Overview



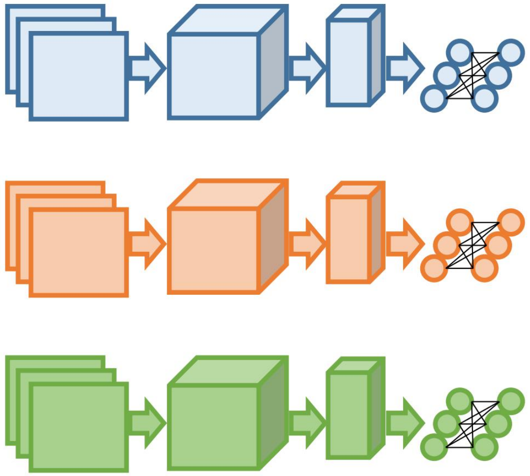
Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

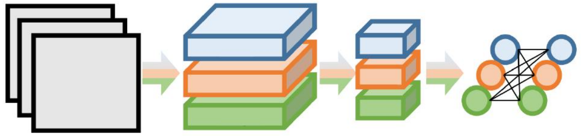
Presentation Overview



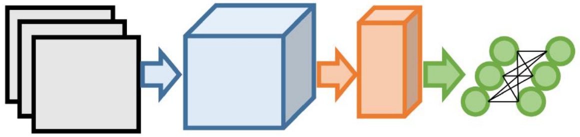
Data parallelism



Model parallelism



Pipeline parallelism



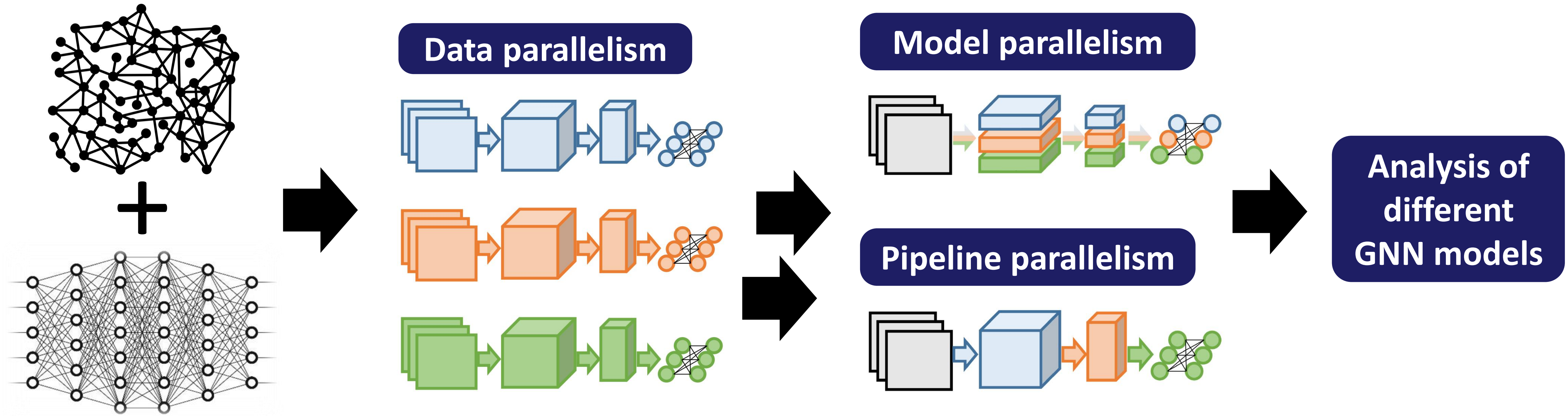
Analysis of different GNN models

Traditional Deep Learning (DL) vs. Graph Neural Networks (GNNs)

Graph Neural Networks: ...currency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

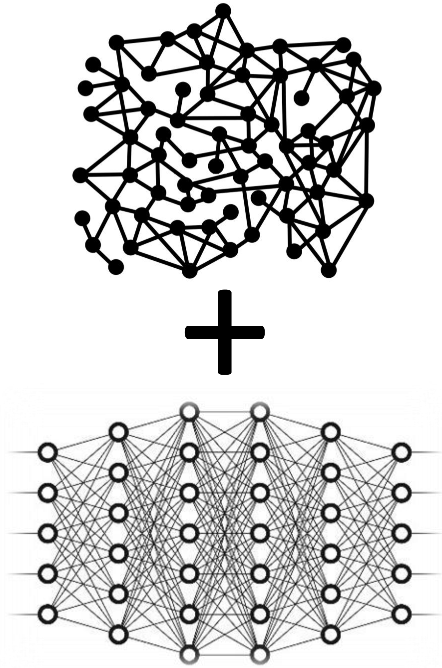
Presentation Overview



Traditional Deep Learning (DL) vs. Graph Neural Networks (GNNs)

Insights about amount of parallelism in GNN computations

Presentation Overview



Traditional Deep Learning (DL) vs. Graph Neural Networks (GNNs)

Insights about amount of parallelism in GNN computations

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

How Does Deep Learning (DL) Work?

Parts of this slide are borrowed from T. Hoefler

How Does Deep Learning (DL) Work?

Parts of this slide are borrowed from T. Hoefler

How Does Deep Learning (DL) Work?

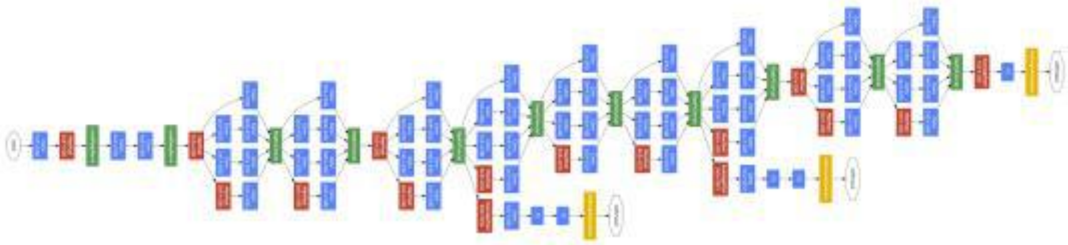
Parts of this slide are borrowed from T. Hoefler

Samples

How Does Deep Learning (DL) Work?

Parts of this slide are borrowed from T. Hoefler

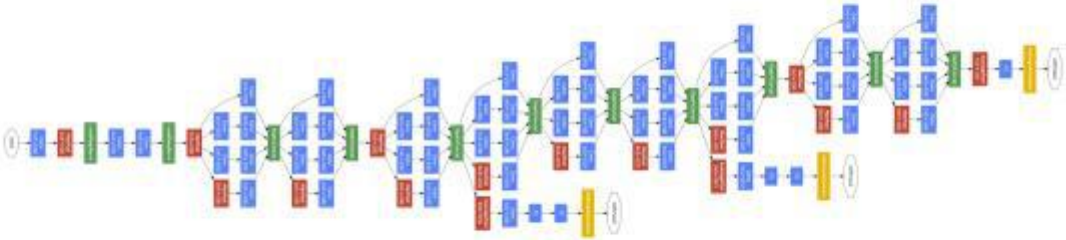
Samples



How Does Deep Learning (DL) Work?

Parts of this slide are borrowed from T. Hoefler

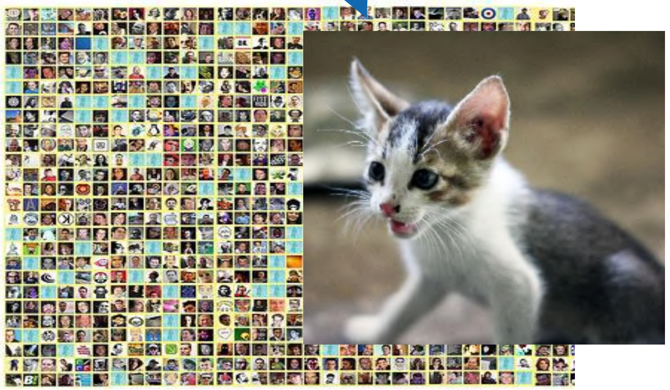
Samples



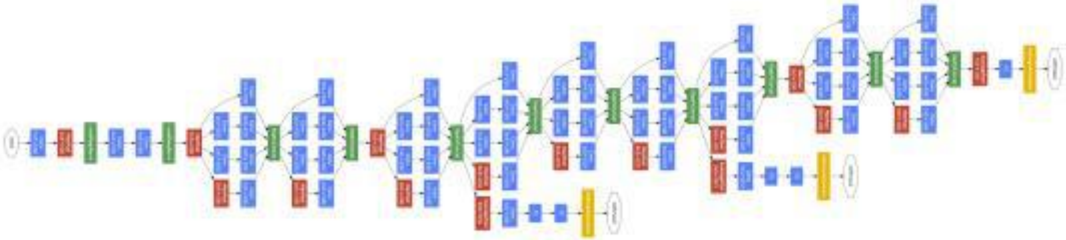
Parts of this slide are borrowed from T. Hoefler

How Does Deep Learning (DL) Work?

Samples



f(x)



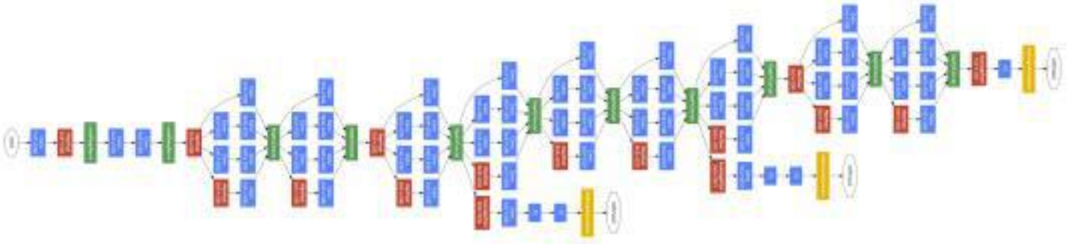
How Does Deep Learning (DL) Work?

Parts of this slide are borrowed from T. Hoefler

Samples

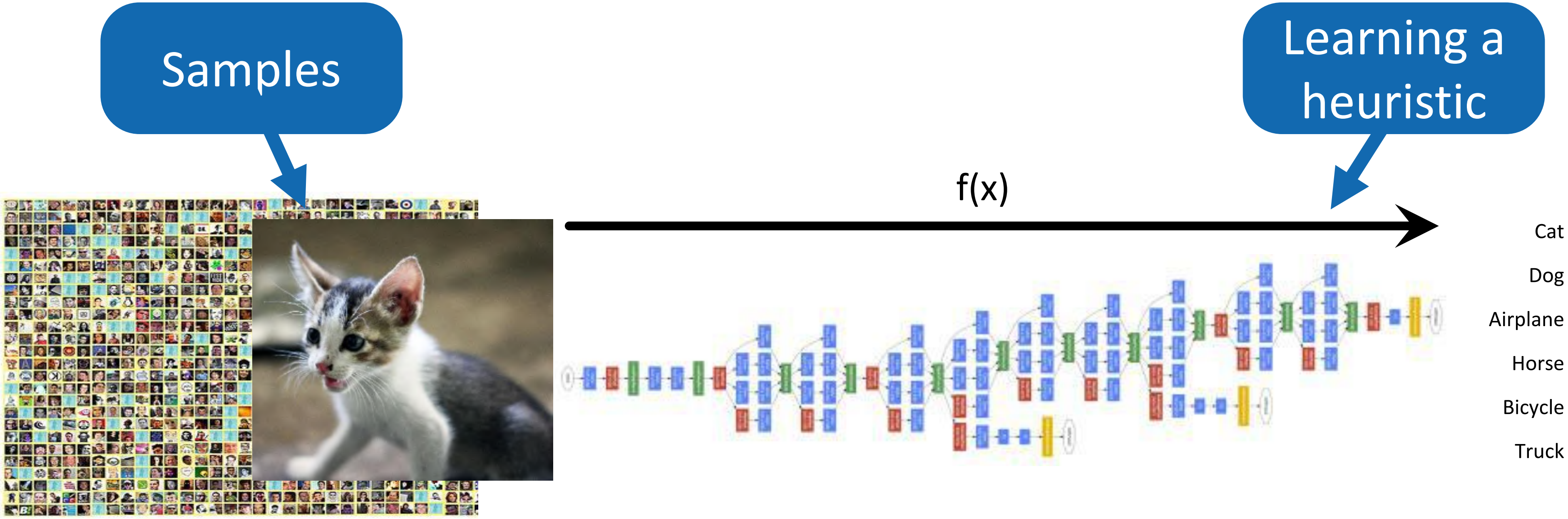
Learning a heuristic

f(x)



Parts of this slide are borrowed from T. Hoefler

How Does Deep Learning (DL) Work?



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

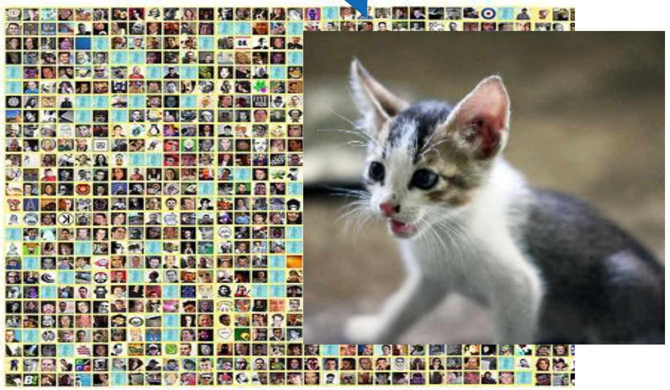
Parts of this slide are borrowed from T. Hoefler

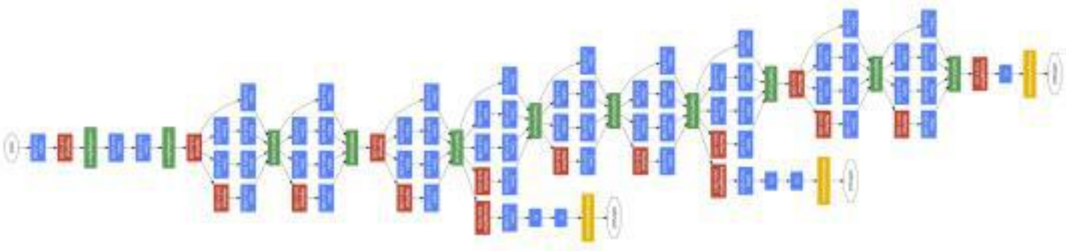
How Does Deep Learning (DL) Work?

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

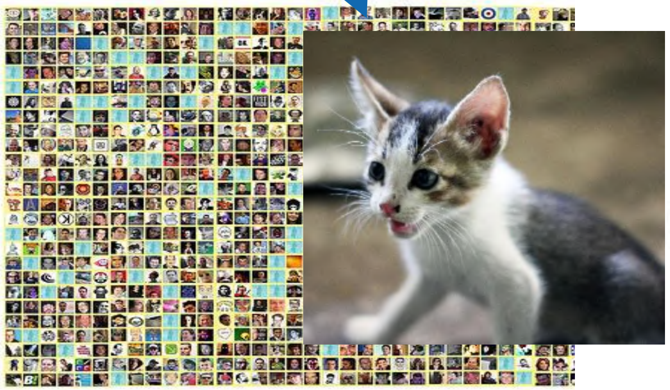
Cat	1.00
Dog	0.00
Airplane	0.00
	0.00
Horse	0.00
Bicycle	0.00
Truck	0.00

Parts of this slide are borrowed from T. Hoefler

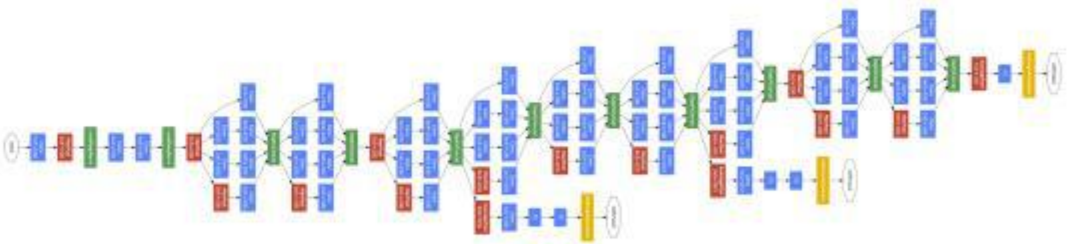
How Does Deep Learning (DL) Work?

Samples

Learning a heuristic



f(x)



layer-wise weight update

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

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

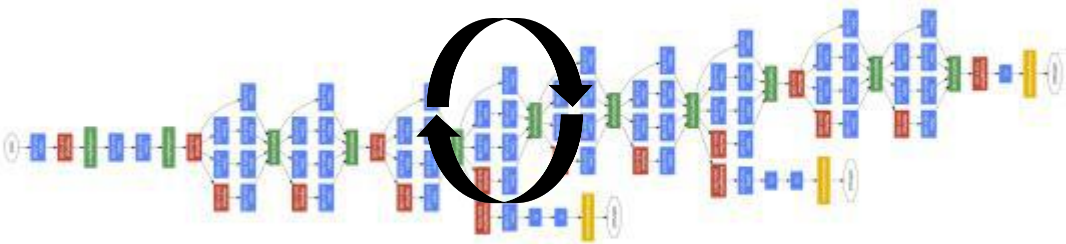
Parts of this slide are borrowed from T. Hoefler

How Does Deep Learning (DL) Work?

Samples

Learning a heuristic

f(x)



layer-wise weight update

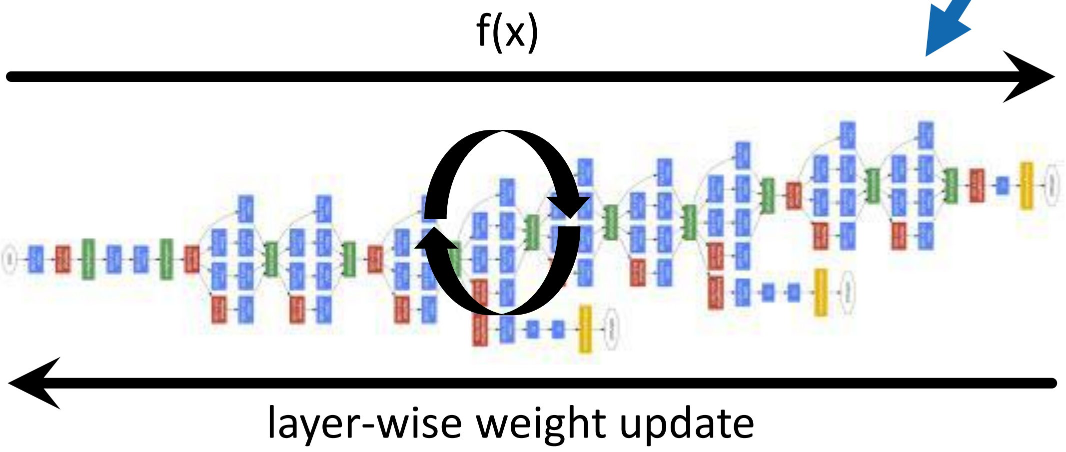
Cat	0.54	Cat	1.00
Dog	0.28	Dog	0.00
Airplane	0.07	Airplane	0.00
	0.04		0.00
Horse	0.33	Horse	0.00
Bicycle	0.02	Bicycle	0.00
Truck	0.02	Truck	0.00

How Does Deep Learning (DL) Work?

Parts of this slide are borrowed from T. Hoefler

Samples

Learning a heuristic



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

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

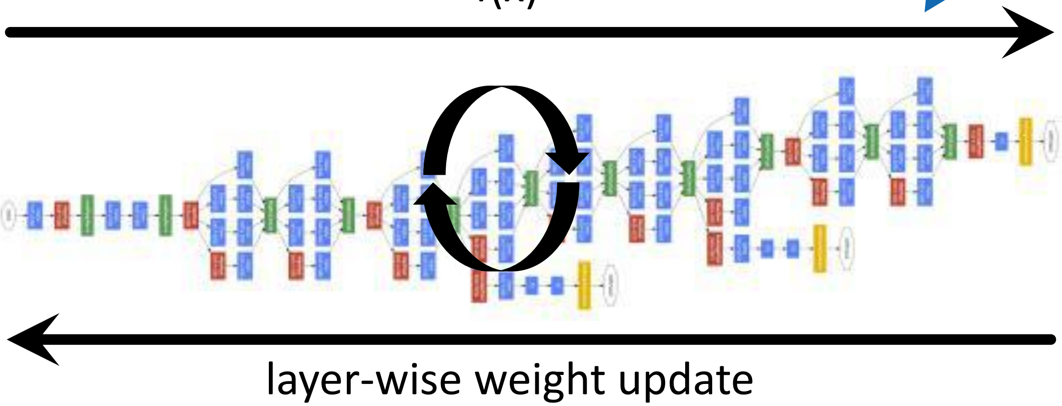
Full-batch: accurate weight updates, but slow convergence

Parts of this slide are borrowed from T. Hoefler

How Does Deep Learning (DL) Work?

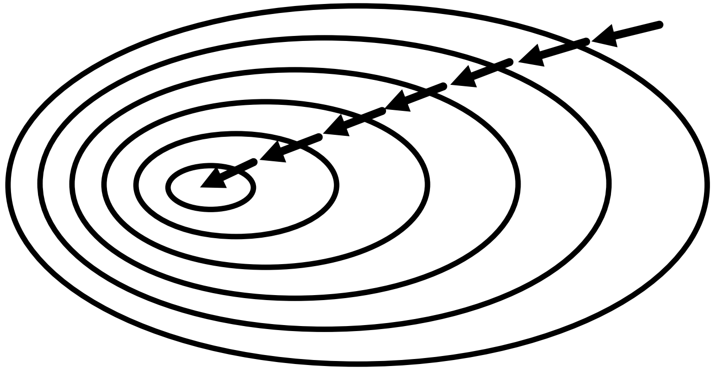
Samples

Learning a heuristic



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

Full-batch: accurate weight updates, but slow convergence

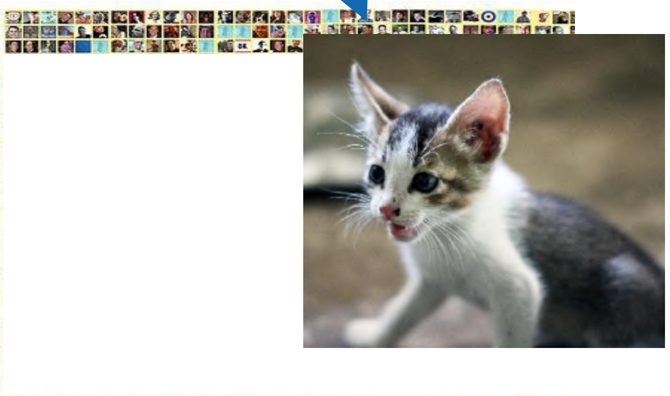


Parts of this slide are borrowed from T. Hoefler

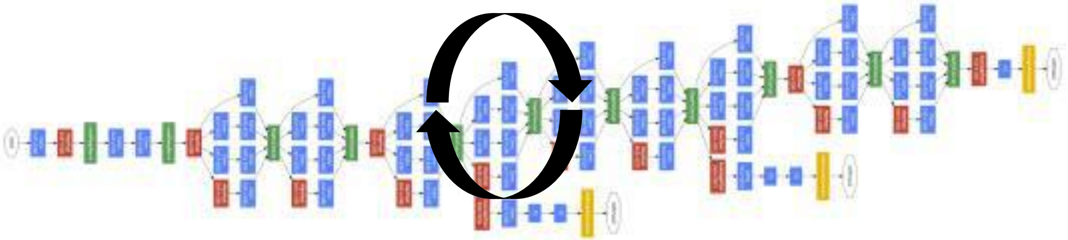
How Does Deep Learning (DL) Work?

Samples

Learning a heuristic



f(x)

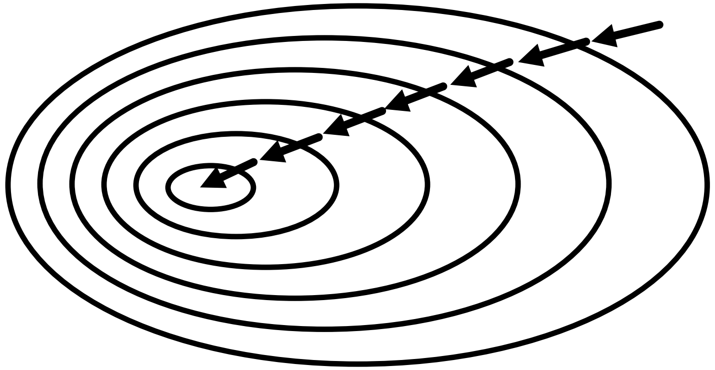


layer-wise weight update

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

Full-batch: accurate weight updates, but slow convergence

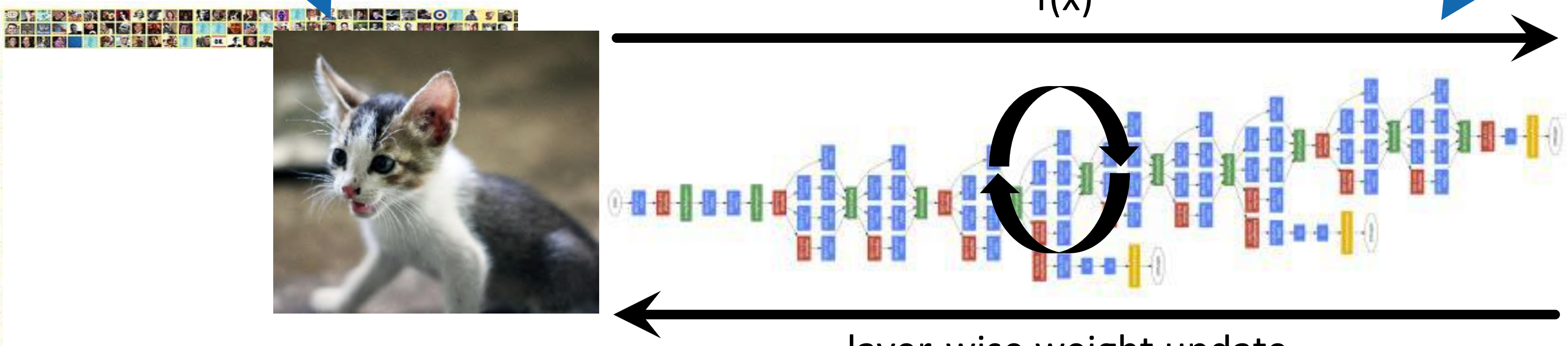


Parts of this slide are borrowed from T. Hoefler

How Does Deep Learning (DL) Work?

Samples

Learning a heuristic

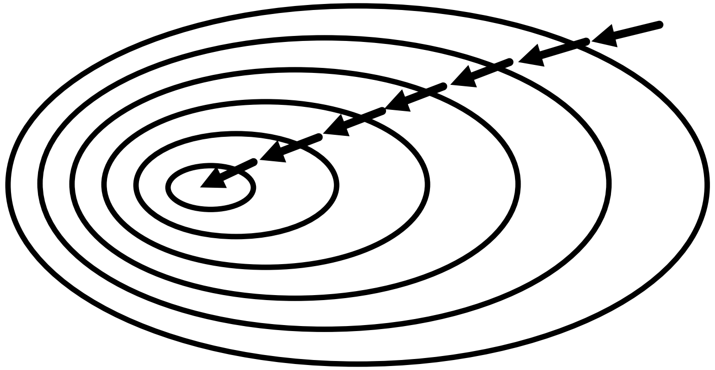


Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

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

Full-batch: accurate weight updates, but slow convergence

Mini-batch: less accurate weight updates, but faster convergence

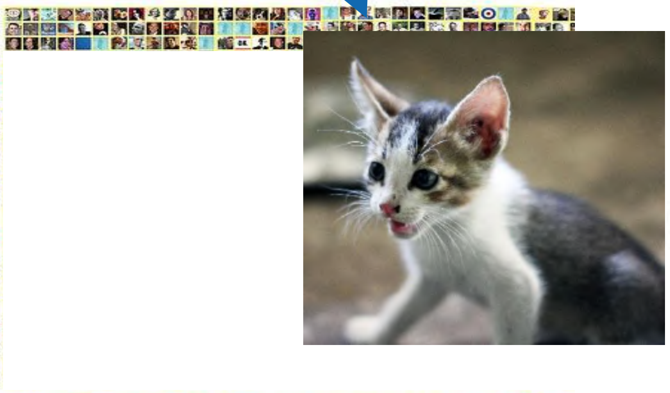


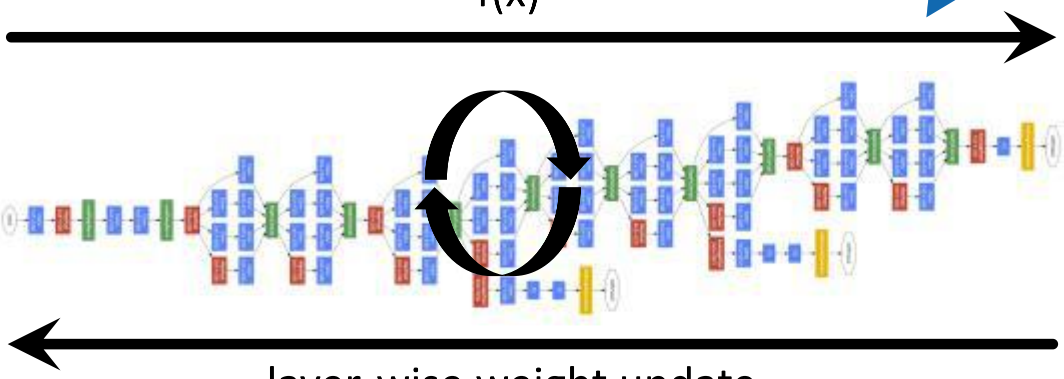
How Does Deep Learning (DL) Work?

Parts of this slide are borrowed from T. Hoefler

Samples

Learning a heuristic

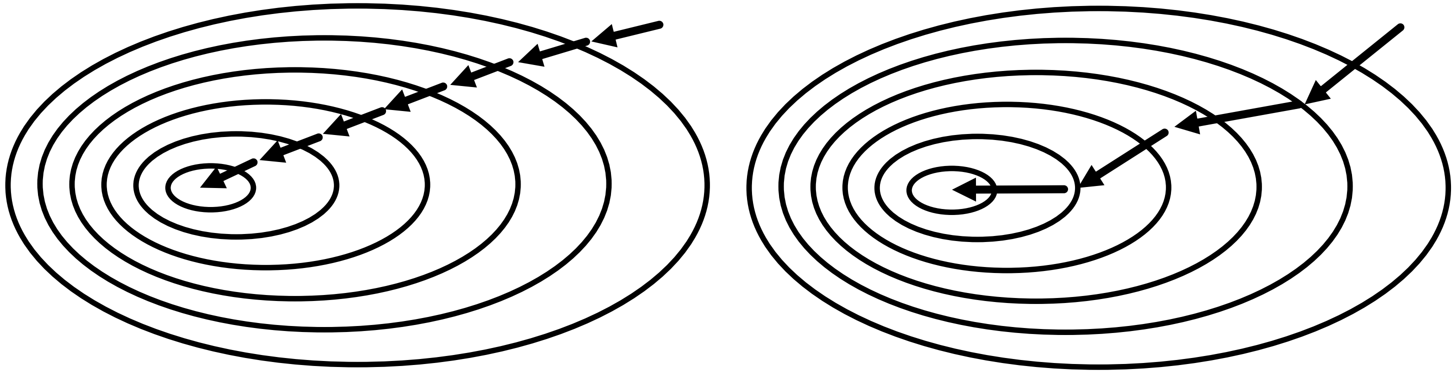




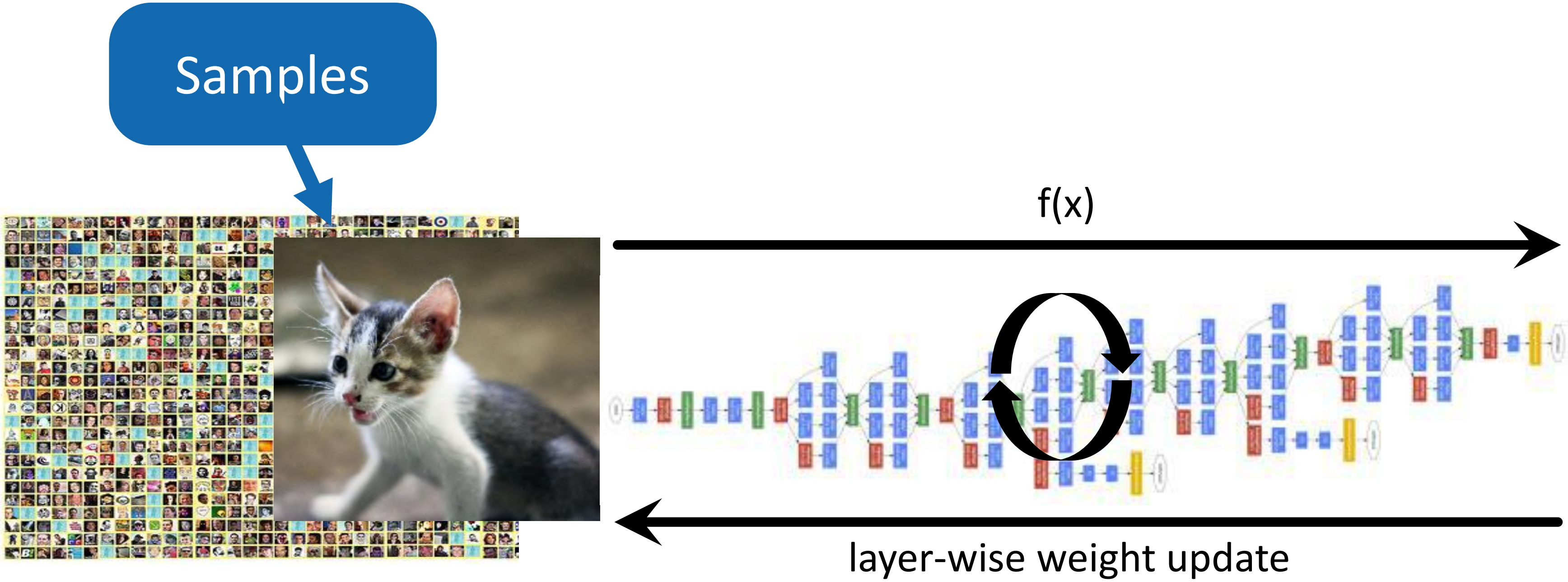
Cat	0.54	Cat	1.00
Dog	0.28	Dog	0.00
Airplane	0.07	Airplane	0.00
	0.04		0.00
Horse	0.33	Horse	0.00
Bicycle	0.02	Bicycle	0.00
Truck	0.02	Truck	0.00

Full-batch: accurate weight updates, but slow convergence

Mini-batch: less accurate weight updates, but faster convergence



New Form of Deep Learning: Graph Neural Networks (GNNs)



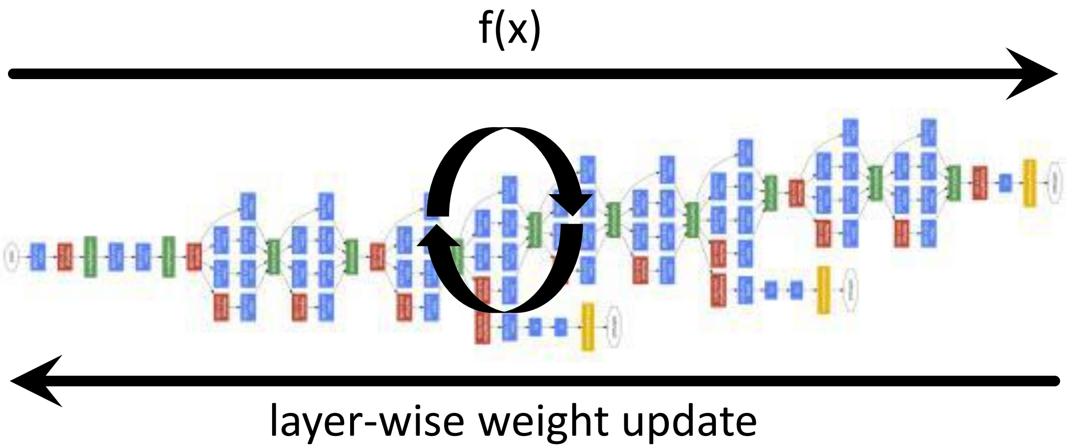
Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

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

New Form of Deep Learning: 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



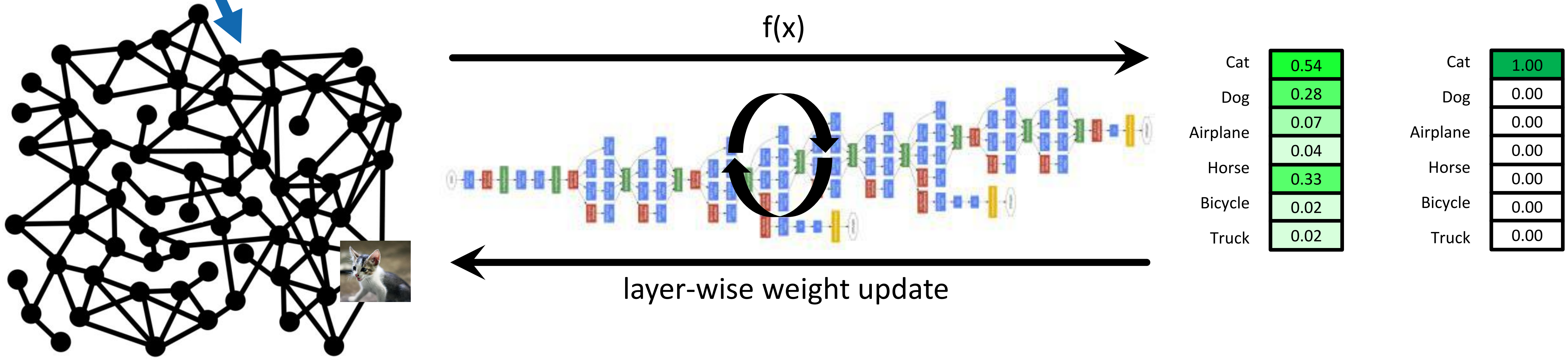
Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

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

New Form of Deep Learning: Graph Neural Networks (GNNs)

Samples

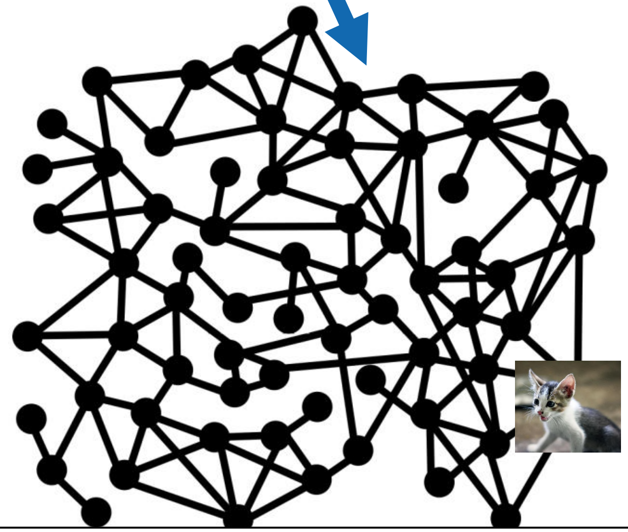
These could still be photos, but now forming **explicit relations**, e.g., two photos are related if they were taken within an hour



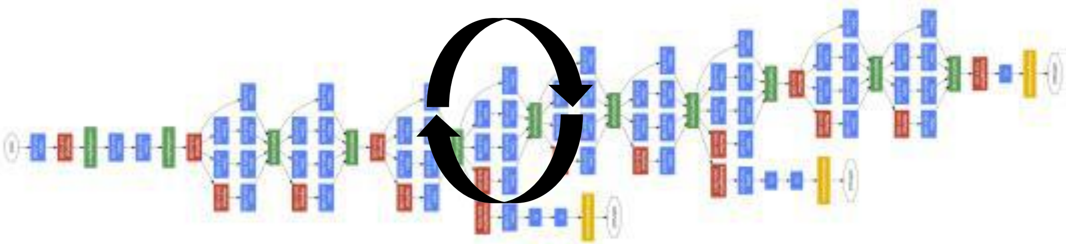
New Form of Deep Learning: Graph Neural Networks (GNNs)

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

The **graph neural network model** [PDF] ieee.org
F Scarselli, M Gori, AC Tsoi… - … on **neural networks**, 2008 - ieeexplore.ieee.org
Many underlying relationships among data in several areas of science and engineering, eg, computer vision, molecular chemistry, molecular biology, pattern recognition, and data …
☆ Save Cite Cited by 3880 Related articles All 16 versions

New Form of Deep Learning: 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)

Cat 0.54

Cat 1.00 0.00 0.00 0.00 0.00 0.00 0.00

Semi-supervised classification with graph convolutional networks — **[PDF]** arxiv.org

TN **Kipf**, M **Welling** - arXiv preprint arXiv:1609.02907, 2016 - arxiv.org

We present a scalable approach for semi-supervised learning on graph-structured data that is based on an efficient variant of convolutional neural networks which operate directly on …

☆ Save 99 Cite Cited by 15270 Related articles All 19 versions

The **graph neural network model** — **[PDF]** ieee.org

F Scarselli, M Gori, AC Tsoi… - … on **neural networks**, 2008 - ieeexplore.ieee.org

Many underlying relationships among data in several areas of science and engineering, eg, computer vision, molecular chemistry, molecular biology, pattern recognition, and data …

☆ Save 99 Cite Cited by 3880 Related articles All 16 versions

New Form of Deep Learning: Graph Neural Networks (GNNs)

Samp

Graph attention networks **[PDF]** arxiv.org

P Veličković, G Cucurull, A Casanova… - arXiv preprint arXiv …, 2017 - arxiv.org

We present graph attention networks (GATs), novel neural network architectures that operate on graph-structured data, leveraging masked self-attentional layers to address the …

☆ Save 99 Cite Cited by 3447 Related articles All 8 versions

Semi-supervised classification with graph convolutional networks **[PDF]** arxiv.org

TN **Kipf**, M **Welling** - arXiv preprint arXiv:1609.02907, 2016 - arxiv.org

We present a scalable approach for semi-supervised learning on graph-structured data that is based on an efficient variant of convolutional neural networks which operate directly on …

☆ Save 99 Cite Cited by 15270 Related articles All 19 versions

0.00
0.00
0.00
0.00
0.00
0.00

The **graph neural network model** **[PDF]** ieee.org

F Scarselli, M Gori, AC Tsoi… - … on **neural networks**, 2008 - ieeexplore.ieee.org

Many underlying relationships among data in several areas of science and engineering, eg, computer vision, molecular chemistry, molecular biology, pattern recognition, and data …

☆ Save 99 Cite Cited by 3880 Related articles All 16 versions

New Form of Deep Learning: Graph Neural Networks (GNNs)

Samp

Graph attention networks [PDF] arxiv.org

P Veličković, G Cucurull, A Casanova… - arXiv preprint arXiv …, 2017 - arxiv.org

We present graph attention networks (GATs), novel neural network architectures that operate on graph-structured data, leveraging masked self-attentional layers to address the …

☆ Save Cite Cited by 3447 Related articles All 8 versions

Semi-supervised classification with graph convolutional networks [PDF] arxiv.org

TN **Kipf**, M **Welling** - arXiv preprint arXiv:1609.02907, 2016 - arxiv.org

We present a scalable approach for semi-supervised learning on graph-structured data that is based on an efficient variant of convolutional neural networks which operate directly on …

☆ Save Cite Cited by 15270 Related articles All 19 versions

0.00
0.00
0.00
0.00
0.00
0.00

The **graph neural network model** [PDF] ieee.org

F Scarselli, M Gori, AC Tsoi… - … on **neural networks**, 2008 - ieeexplore.ieee.org

Many underlying relationships among data in several areas of science and engineering, eg, computer vision, molecular chemistry, molecular biology, pattern recognition, and data …

☆ Save Cite Cited by 3880 Related articles All 16 versions

Let’s see some recent success stories of GNNs

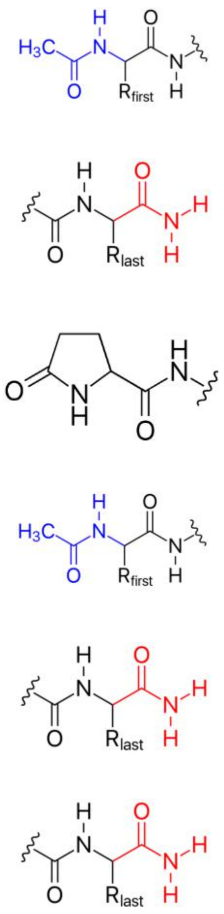
EXAMPLE: PROTEIN FOLDING

Example: Protein Folding

Given this…

EXAMPLE: PROTEIN FOLDING

Given this…

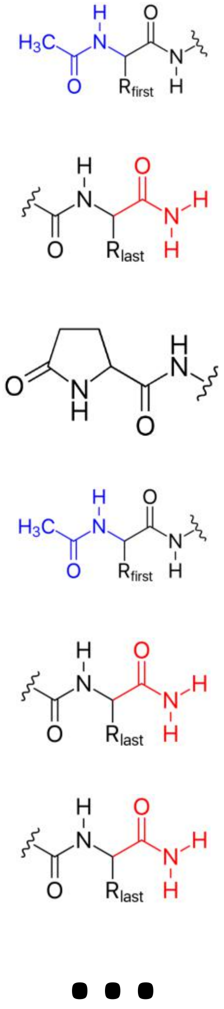


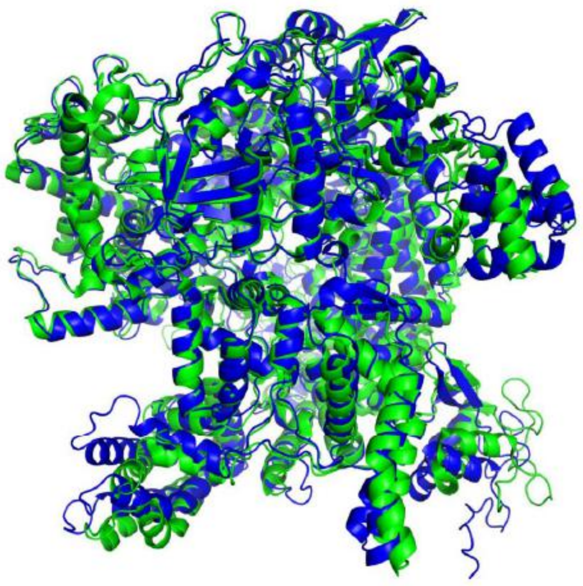
…

Example: Protein Folding

Given this…

…predict this:

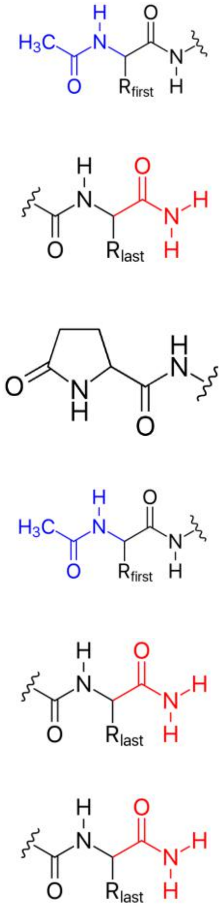




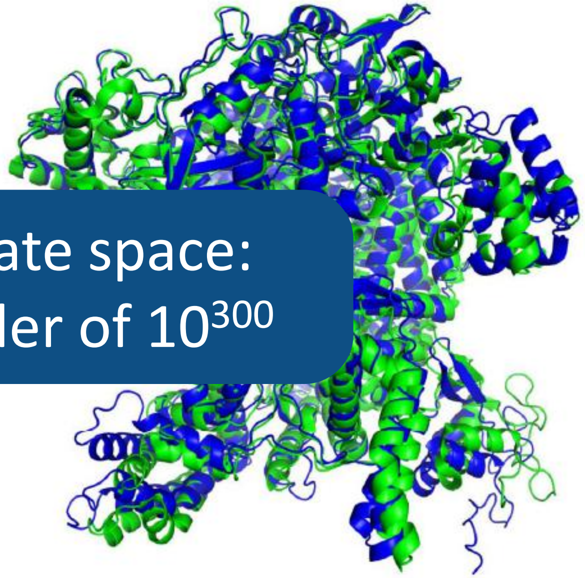
EXAMPLE: PROTEIN FOLDING

Given this…

…predict this:



…

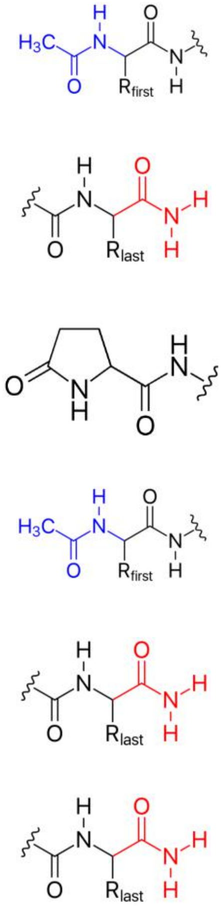


State space:
order of 10^{300}

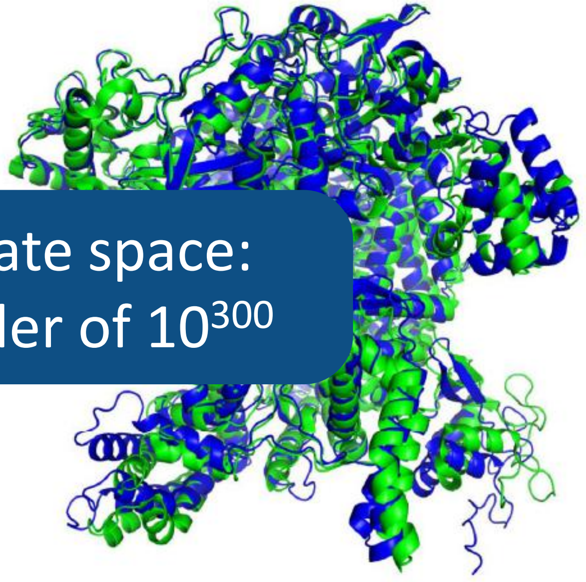
Example: Protein Folding

Given this…

…predict this:



…



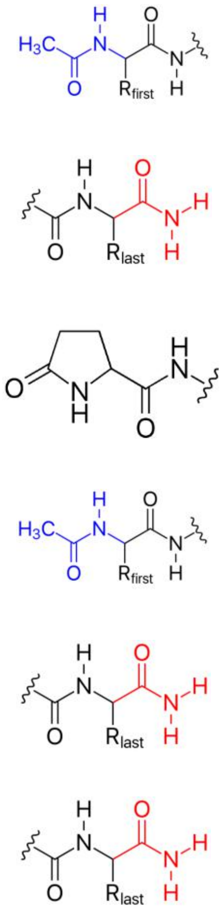
State space:
order of 10^{300}

"In theory, a protein's amino acid sequence should fully determine its structure"
(Christian Anfinsen, 1972 Nobel Prize in Chemistry)

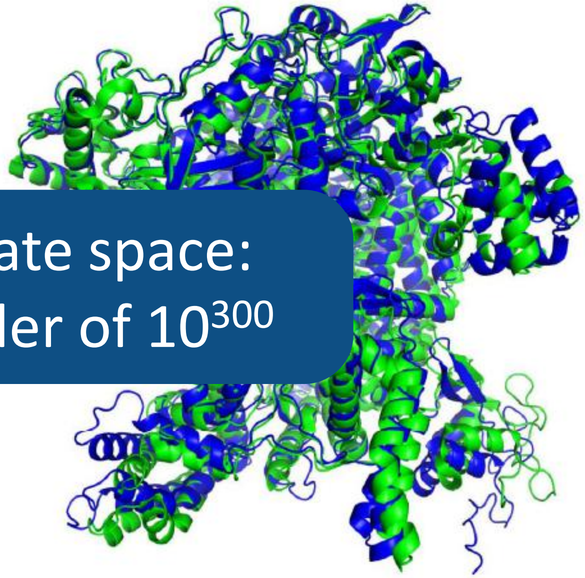
EXAMPLE: PROTEIN FOLDING

Given this…

…predict this:



…



State space: order of 10^{300}

"In theory, a protein's amino acid sequence should fully determine its structure" (Christian Anfinsen, 1972 Nobel Prize in Chemistry)

@ Nature 2021

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

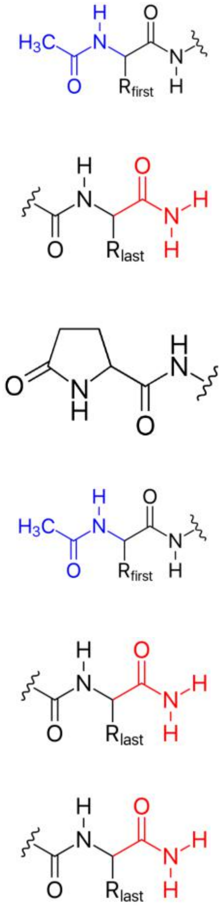
John Jumper[1,4], Richard Evans[1,4], Alexander Pritzel[1,4], Tim Green[1,4], Michael Figurnov[1,4], Olaf Ronneberger[1,4], Kathryn Tunyasuvunakool[1,4], Russ Bates[1,4], Augustin Žídek[1,4], Anna Potapenko[1,4], Alex Bridgland[1,4], Clemens Meyer[1,4], Simon A. A. Kohl[1,4], Andrew J. Ballard[1,4], 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

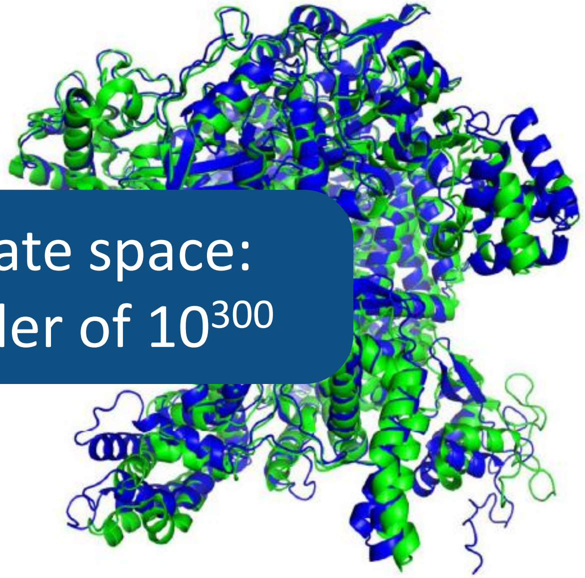
Example: Protein Folding

Given this...

...predict this:

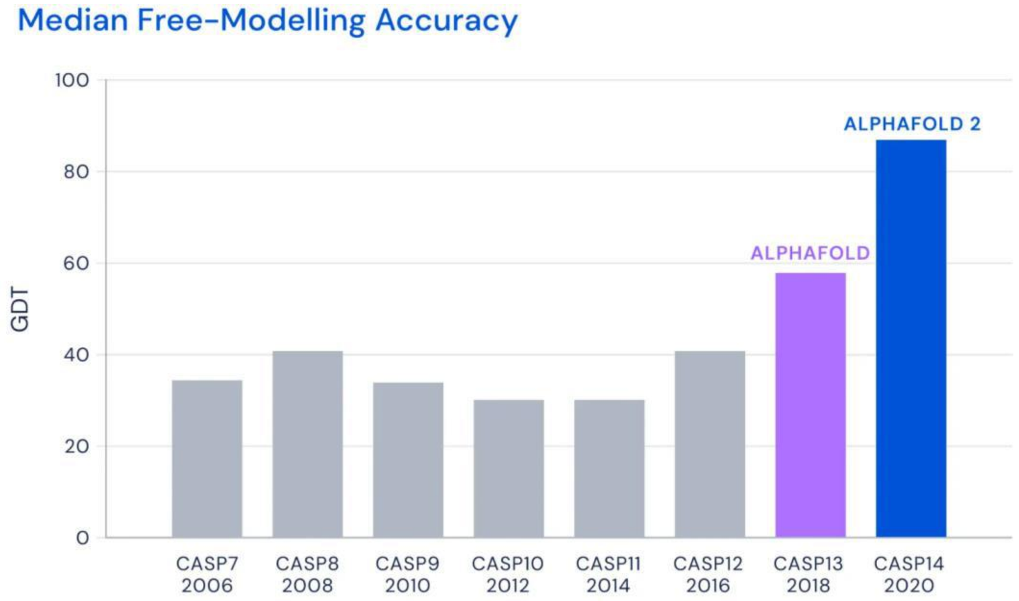


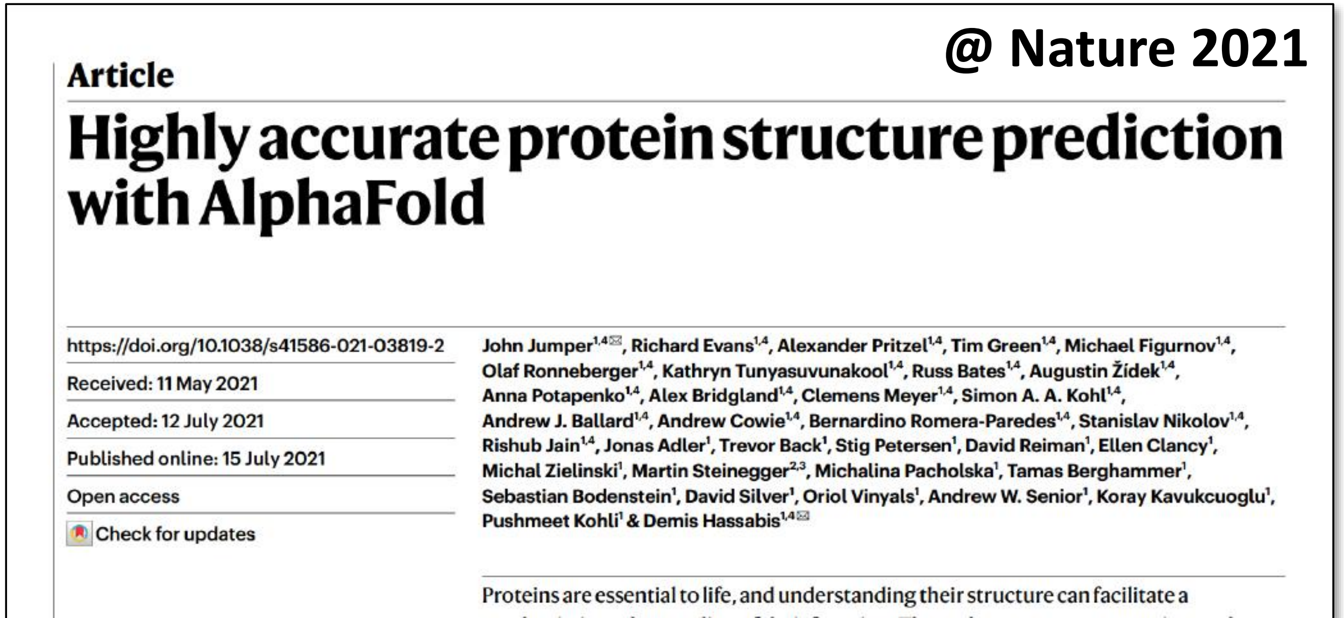
...



State space: order of 10^{300}

"In theory, a protein's amino acid sequence should fully determine its structure" (Christian Anfinsen, 1972 Nobel Prize in Chemistry)





@ Nature 2021

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

Example: Protein Folding

Forbes

Oct 3, 2021, 07:34pm EDT | 56,236 views

AlphaFold Is The Most Important Achievement In AI—Ever

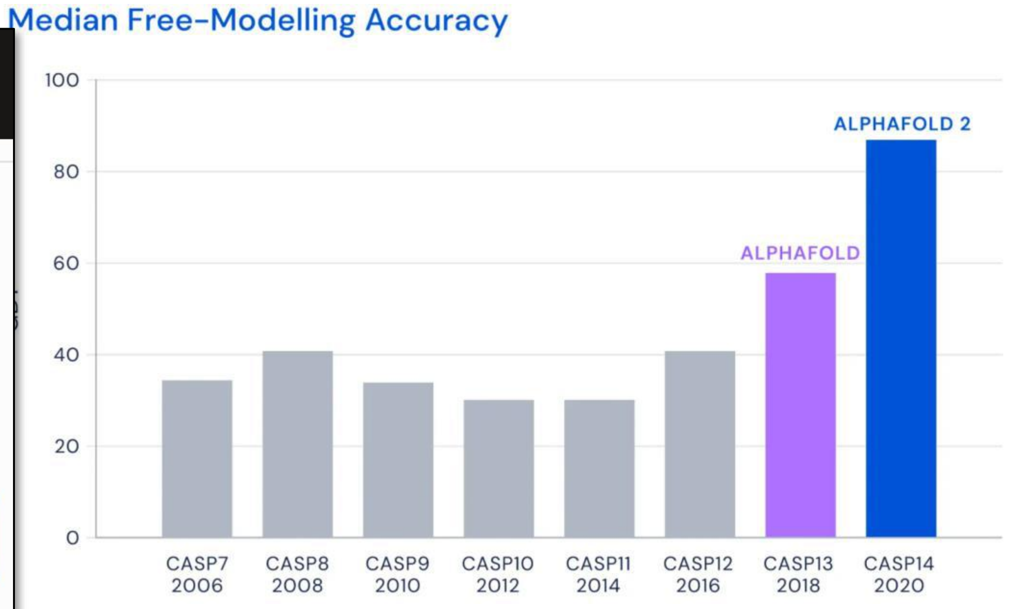
Rob Toews Contributor

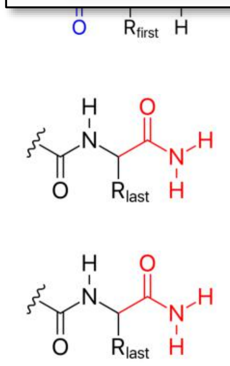
AI

I write about the big picture of artificial intelligence.

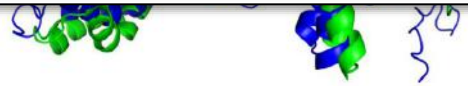
Follow

Listen to article 19 minutes





...



"In theory, a protein's amino acid sequence should fully determine its structure" (Christian Anfinsen, 1972 Nobel Prize in Chemistry)

@ Nature 2021

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

Example: Protein Folding

Forbes

Oct 3, 2021, 07:34pm EDT | 56,236 views

AlphaFold Is The Most Important Achievement In AI—Ever

Rob Toews Contributor
AI
I write about the big picture of artificial intelligence.

Follow

Listen to article 19 minutes

"In theory, a protein's amino acid sequence should fully determine its structure" (Christian Anfinsen, 1972 Nobel Prize in Chemistry)

...

@ Nature 2021

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

Example: Protein Folding

Forbes

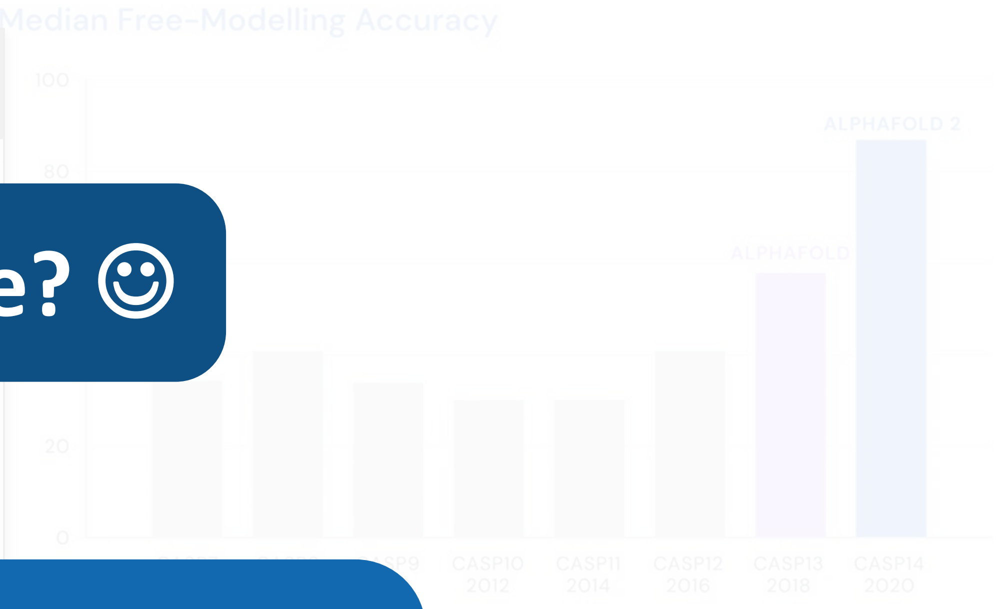
Oct 3, 2021, 07:34pm EDT | 56,236 views

AlphaFold Is The M… Achievement In AI—Ever

Rob Toews Contributor
AI
I write about the big picture of …

Follow

Listen to article 19 minutes



Key technique? ☺

Graph Attention Networks (an important GNN model)

"In theory, a protein's amino acid sequence should fully determine its structure" (Christian Anfinsen, 1972 Nobel Prize in Chemistry)

@ Nature 2021

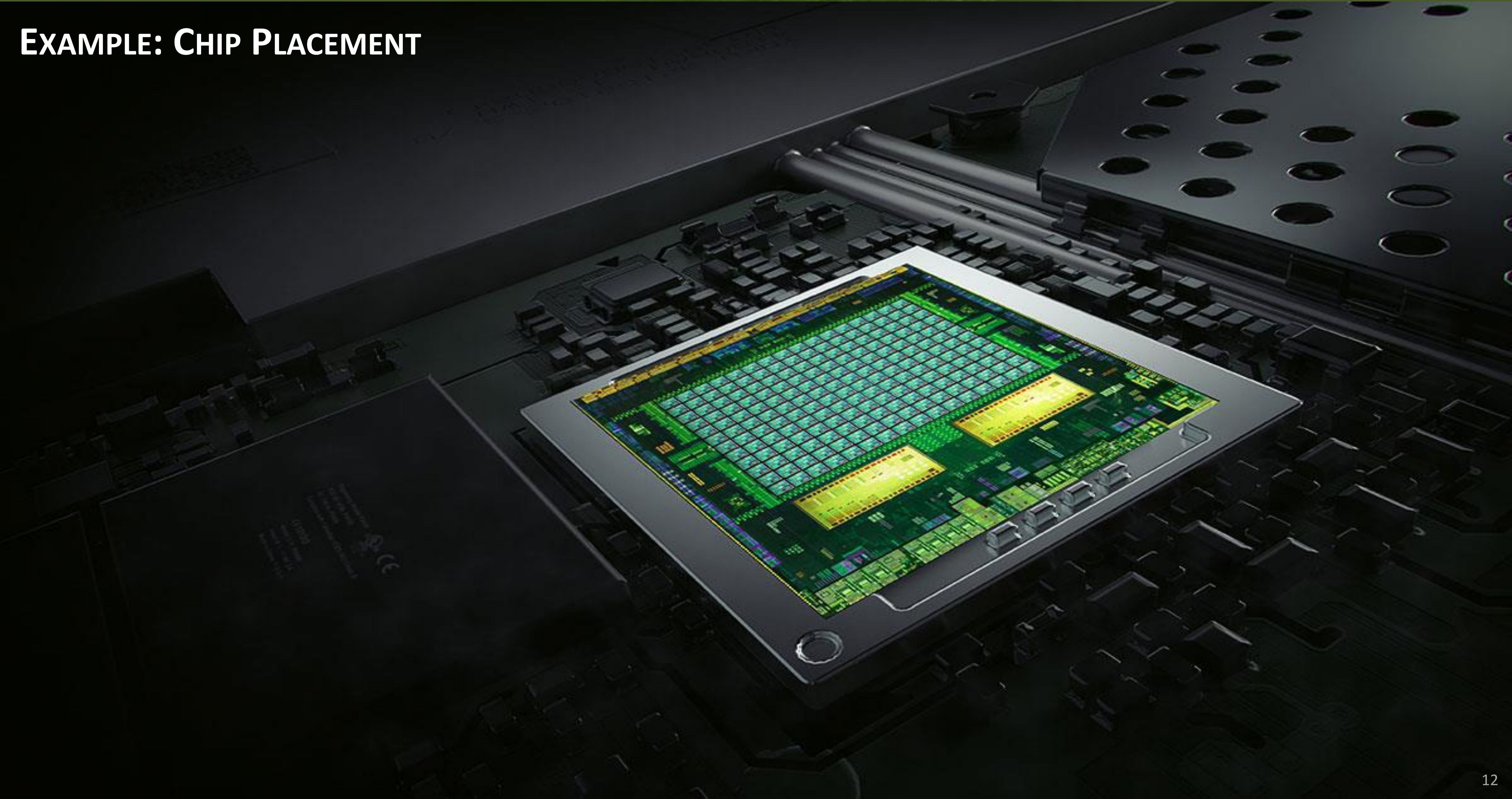
…tructure prediction

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

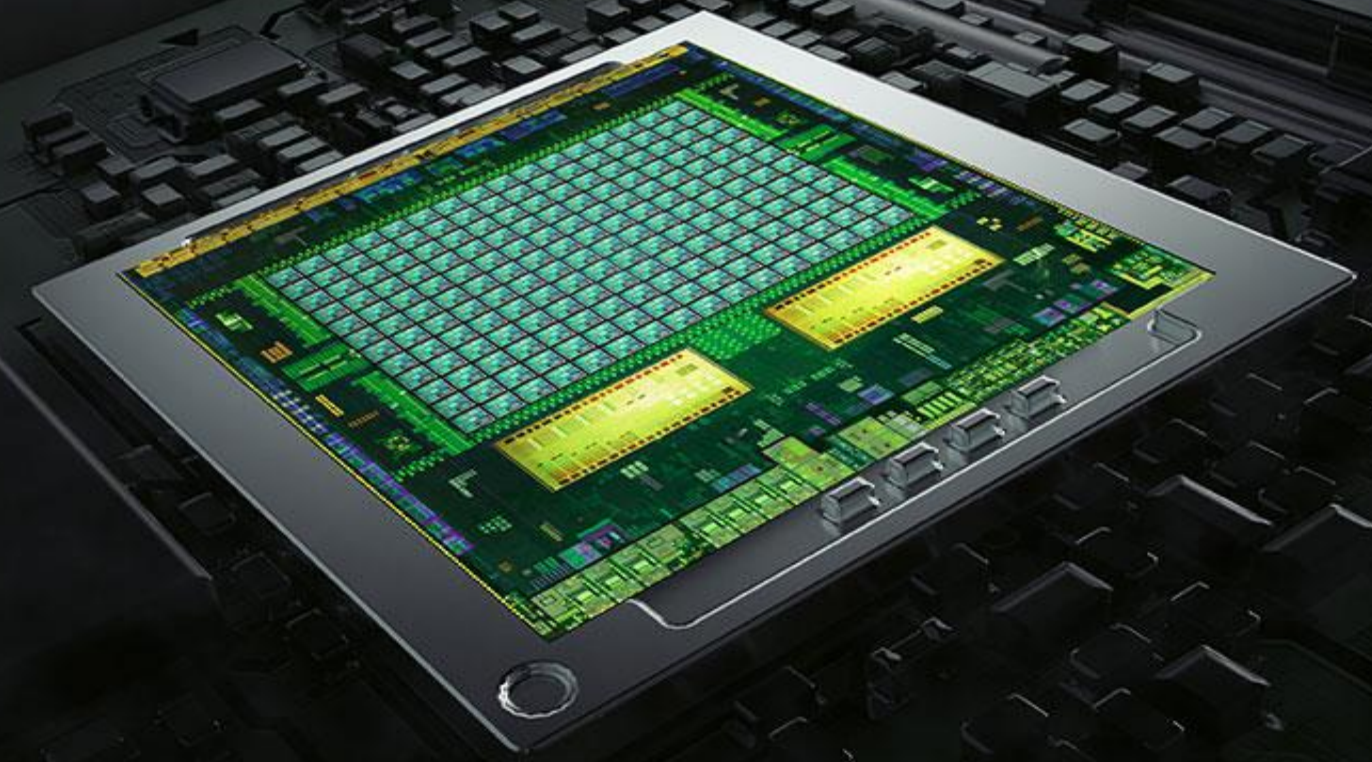
Proteins are essential to life, and understanding their structure can facilitate a…

Example: Chip Placement



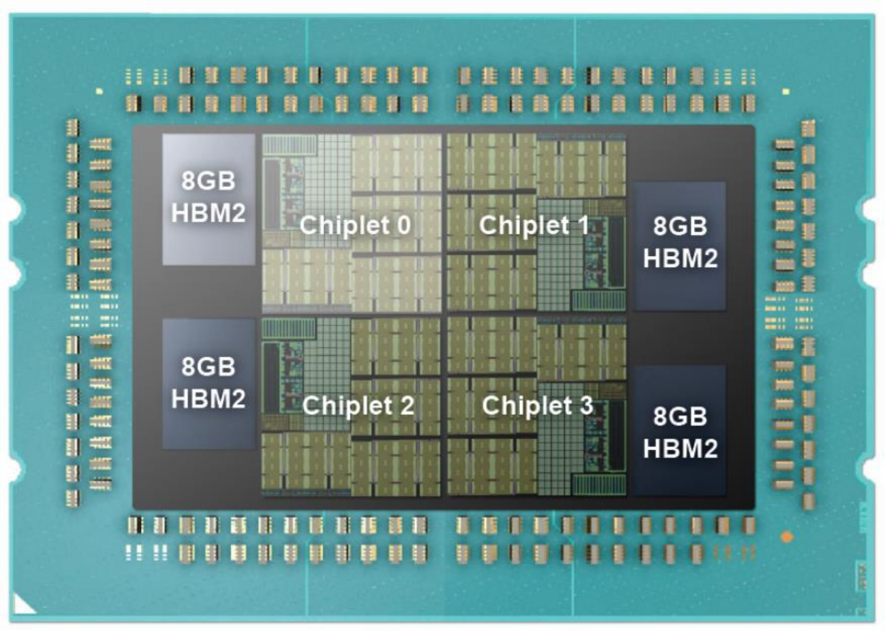
Example: Chip Placement

Find a physical layout for this:



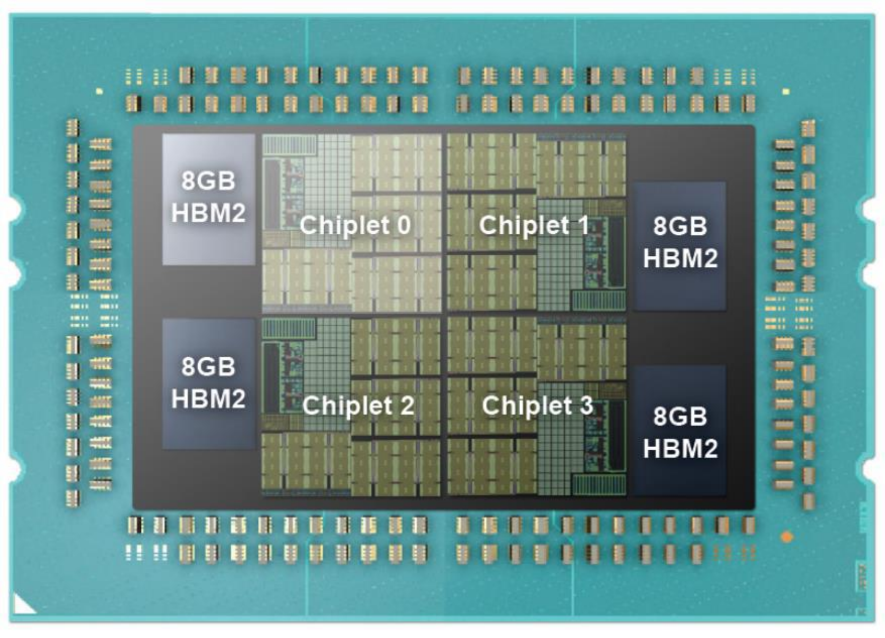
Example: Chip Placement

Find a physical layout for this:

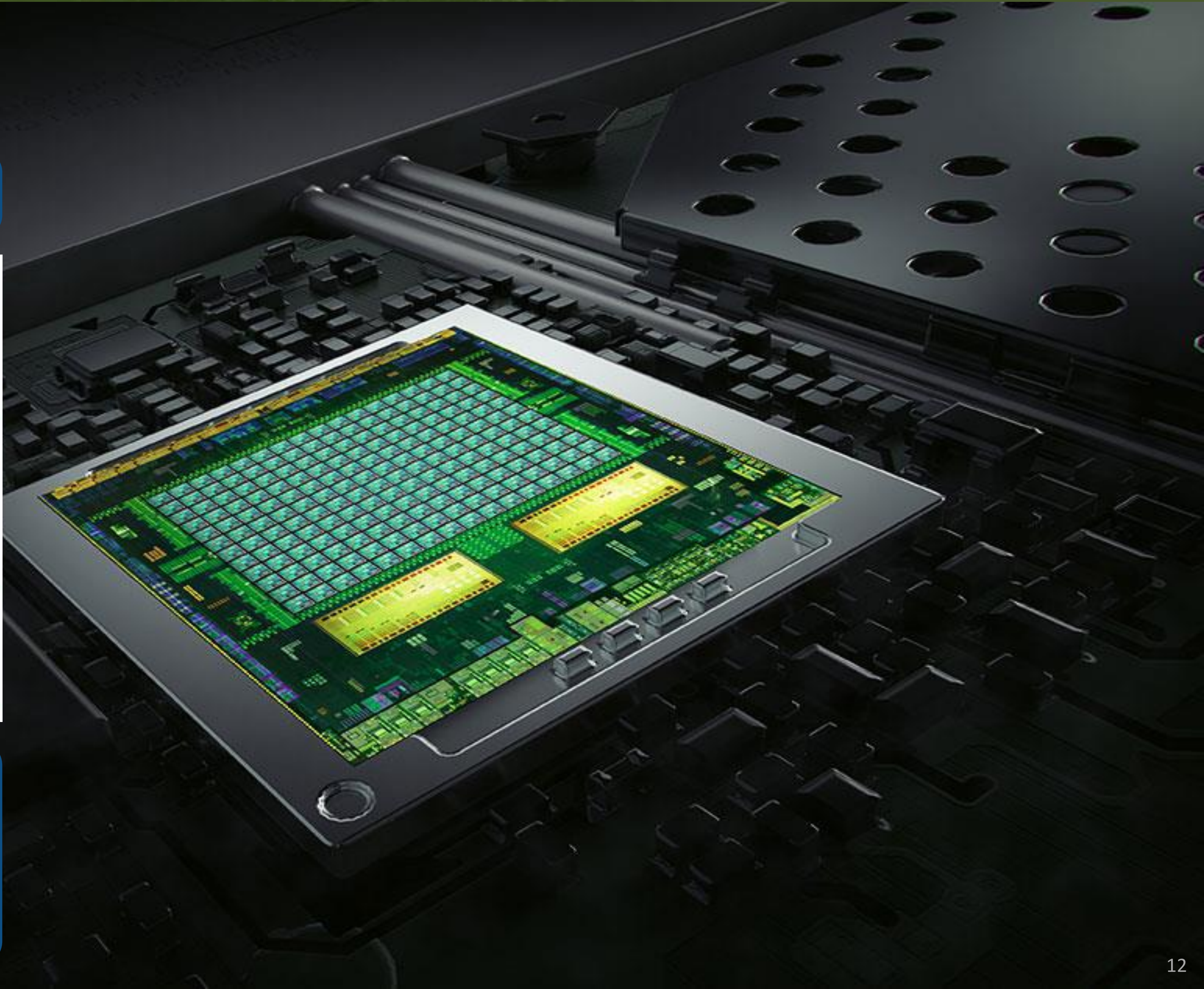


Example: Chip Placement

Find a physical layout for this:



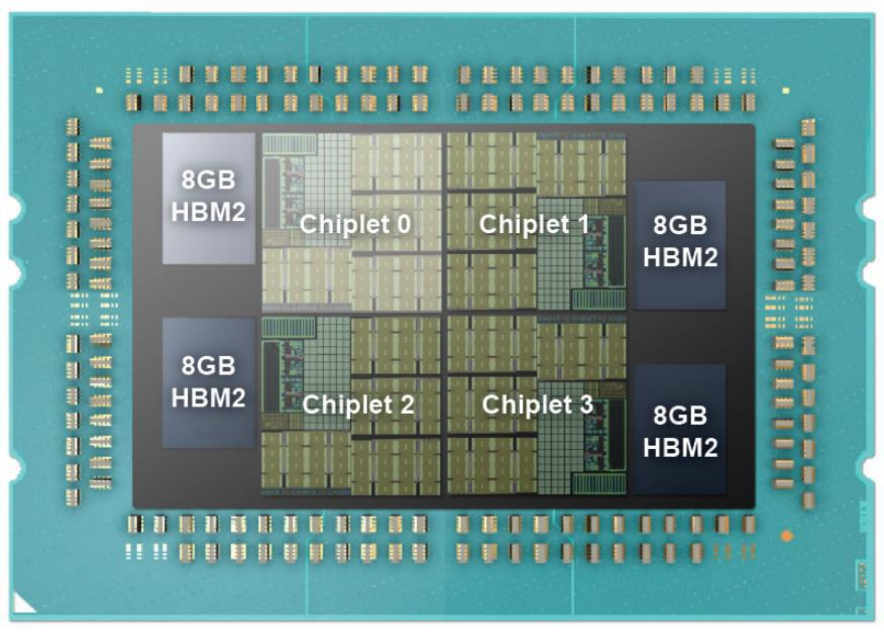
Physical layout: placement of up to millions of (highly heterogeneous) elements such as memory banks, cores, caches, gates, …



Example: Chip Placement

While ensuring…

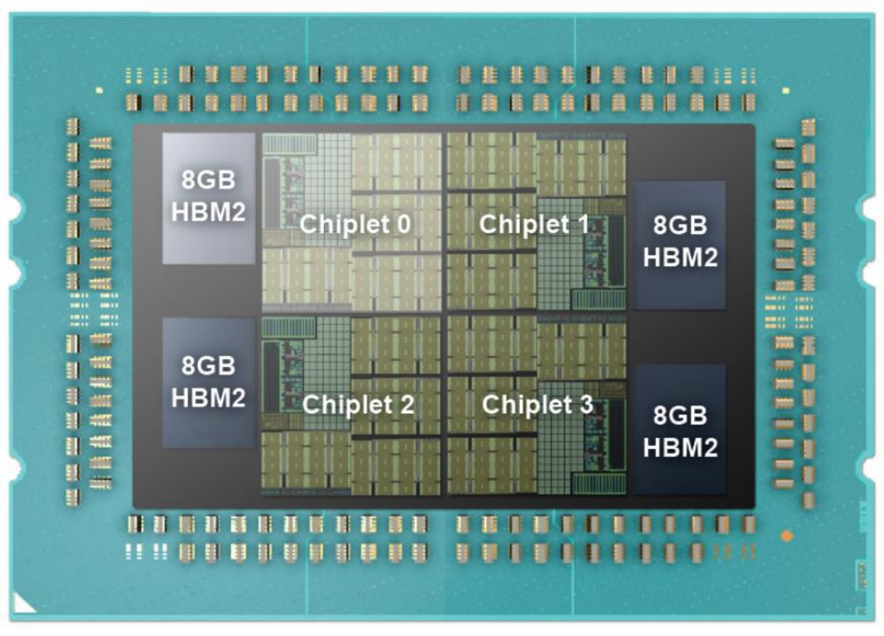
Find a physical layout for this:



Physical layout: placement of up to millions of (highly heterogeneous) elements such as memory banks, cores, caches, gates, …

Example: Chip Placement

Find a physical layout for this:



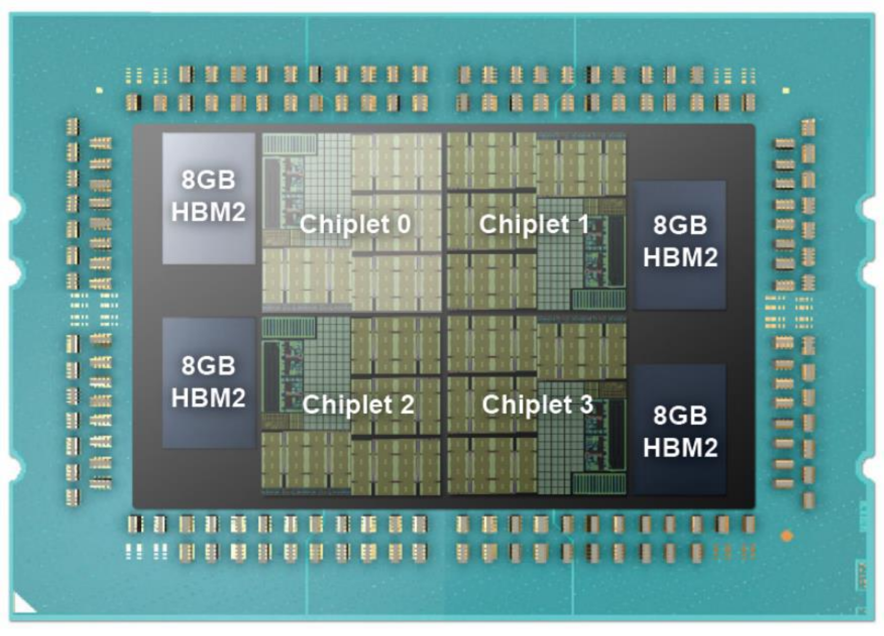
Physical layout: placement of up to millions of (highly heterogeneous) elements such as memory banks, cores, caches, gates, …

While ensuring…

…

Example: Chip Placement

Find a physical layout for this:



Physical layout: placement of up to millions of (highly heterogeneous) elements such as memory banks, cores, caches, gates, …

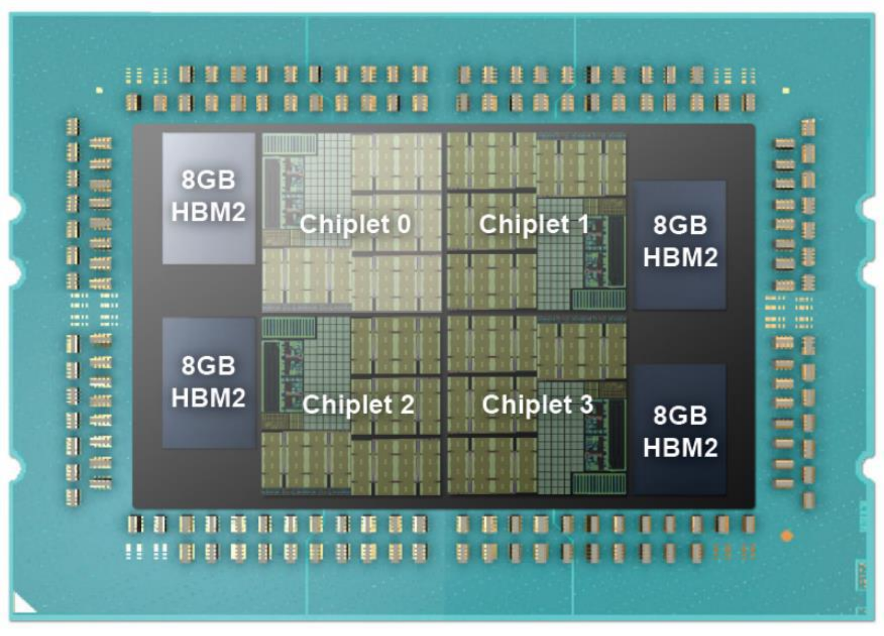
While ensuring…

…

State space: order of 10^{2500}

Example: Chip Placement

Find a physical layout for this:



Physical layout: placement of up to millions of (highly heterogeneous) elements such as memory banks, cores, caches, gates, …

While ensuring…

…

State space: order of 10^{2500}

@ Nature 2021

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

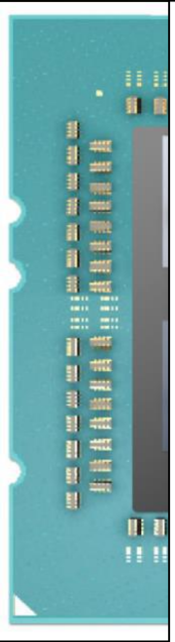
Check for updates

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

Example: Chip Placement

Find



nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature › nature podcast › article

NATURE PODCAST | 09 June 2021

Google AI beats humans at designing computer chips

An AI that designs computer chips in hours, and zooming in on DNA's complex 3D structures.

…

e space:
of 10^{2500}

@ Nature 2021

ethodology for fast

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

Physical layout: placement of up to millions of (highly heterogeneous) elements such as memory banks, cores, caches, gates, …

Example: Chip Placement

Find

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature > nature podcast > article

NATURE PODCAST | 09 June 2021

Google AI beats humans at designing computer chips

An AI that designs computer chips in hours, and zooming in on DNA's complex 3D structures.

…e space:
…of 10^{2500}

@ Nature 2021

…methodology for fast

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, chip floorplanning has defied

Physical layout: placement of up to millions of (highly heterogeneous) elements such as memory banks, cores, caches, gates, …

Example: Chip Placement

Find

nature

Explore content ⌄ About the journal

nature › nature podcast › article

NATURE PODCAST | 09 June 2021

Google AI ... designing ...

An AI that designs compu... structures.

Key technique? ☺

Graph Embeddings / Graph Neural Networks + Reinforcement Learning

...e space: ...of 10^{2500}

@ Nature 2021

...dology for fast

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

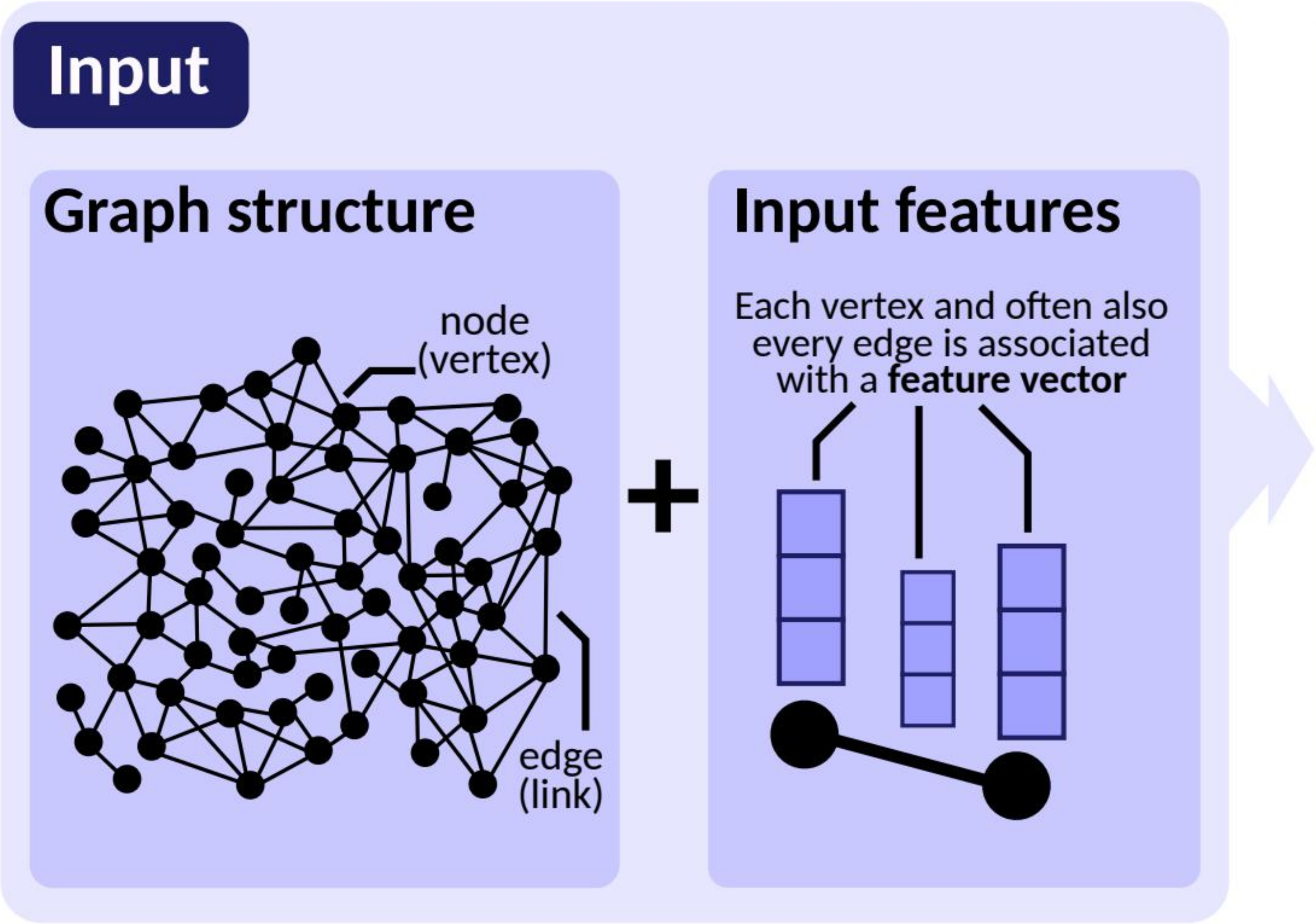
Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, Jiwoo Pak, Andy Tong, Kavya Srinivasa, William Hang, Emre Tuncer, Quoc V. Le, James Laudon, Richard Ho, Roger Carpenter & Jeff Dean

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

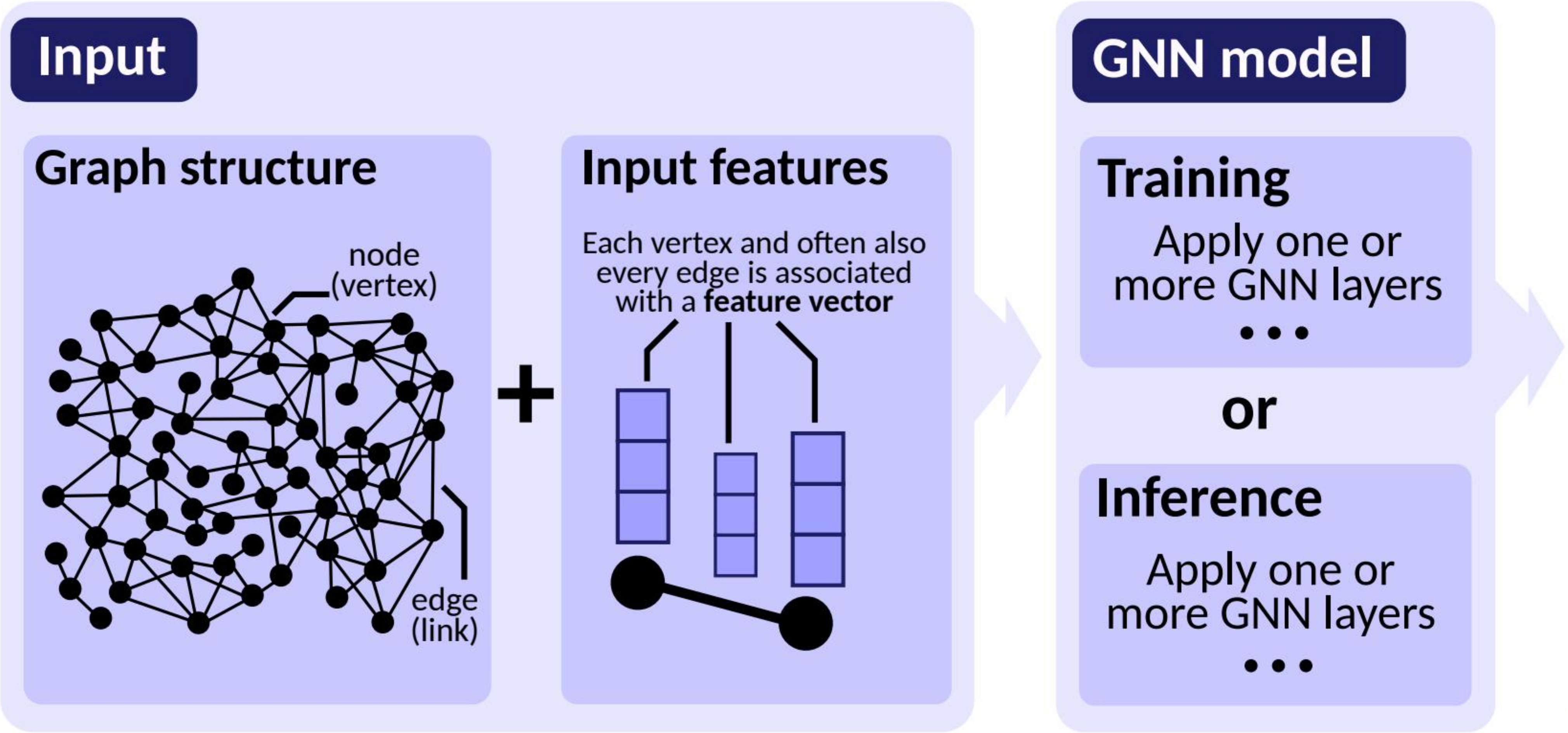
Physical layout: placement of up to millions of (highly heterogeneous) elements such as memory banks, cores, caches, gates, ...

Overview of a GNN Computation

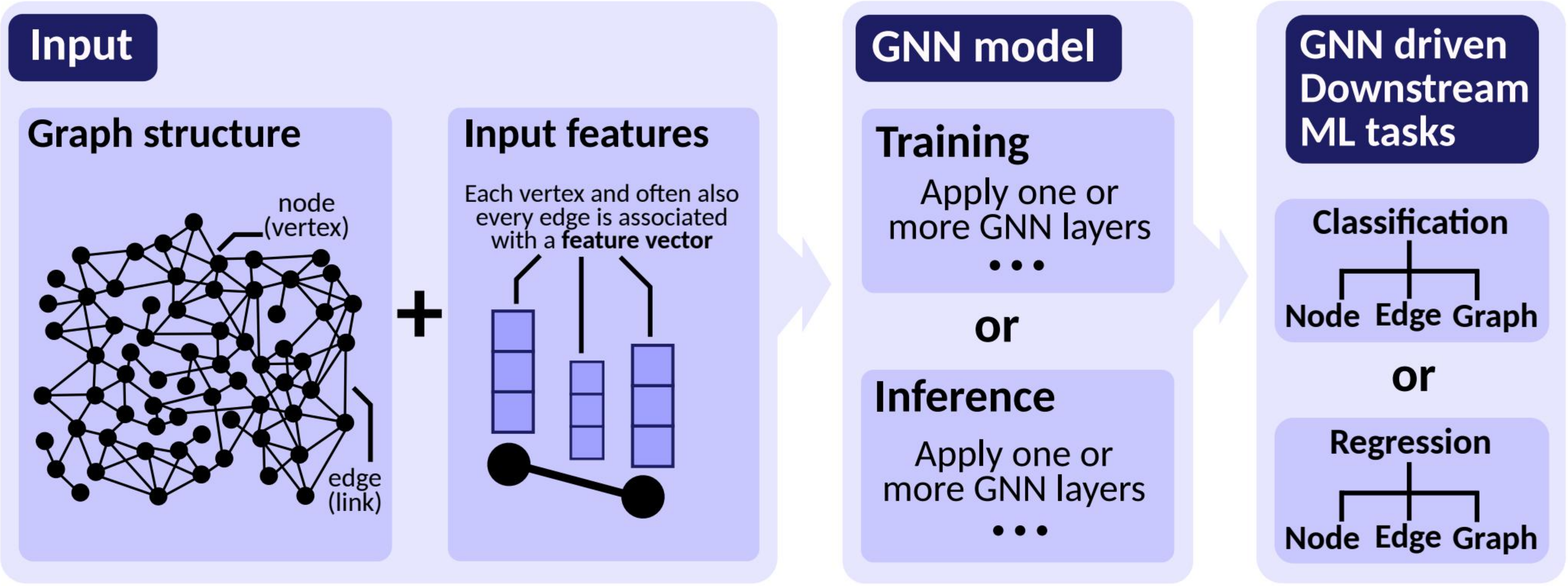
Overview of a GNN Computation



Overview of a GNN Computation



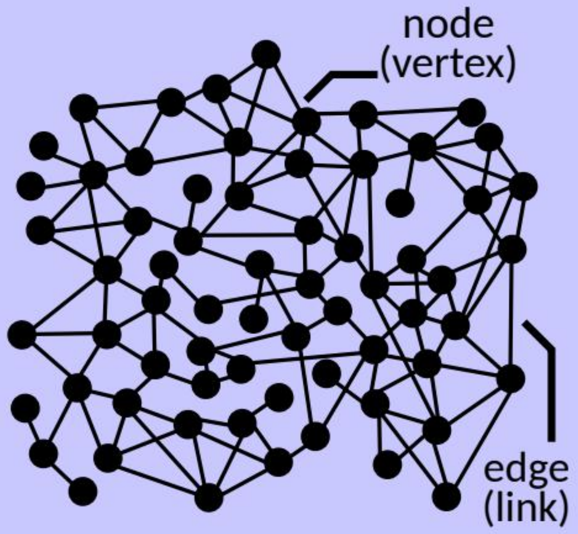
Overview of a GNN Computation



Overview of a GNN Computation

Input

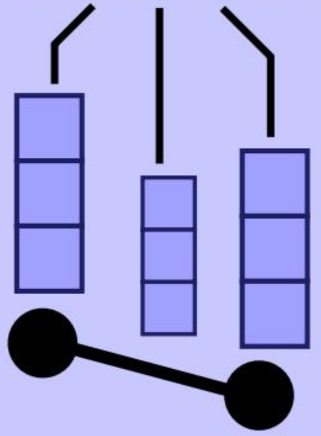
Graph structure



+

Input features

Each vertex and often also every edge is associated with a **feature vector**



GNN model

Training

Apply one or more GNN layers
• • •

or

Inference

Apply one or more GNN layers
• • •

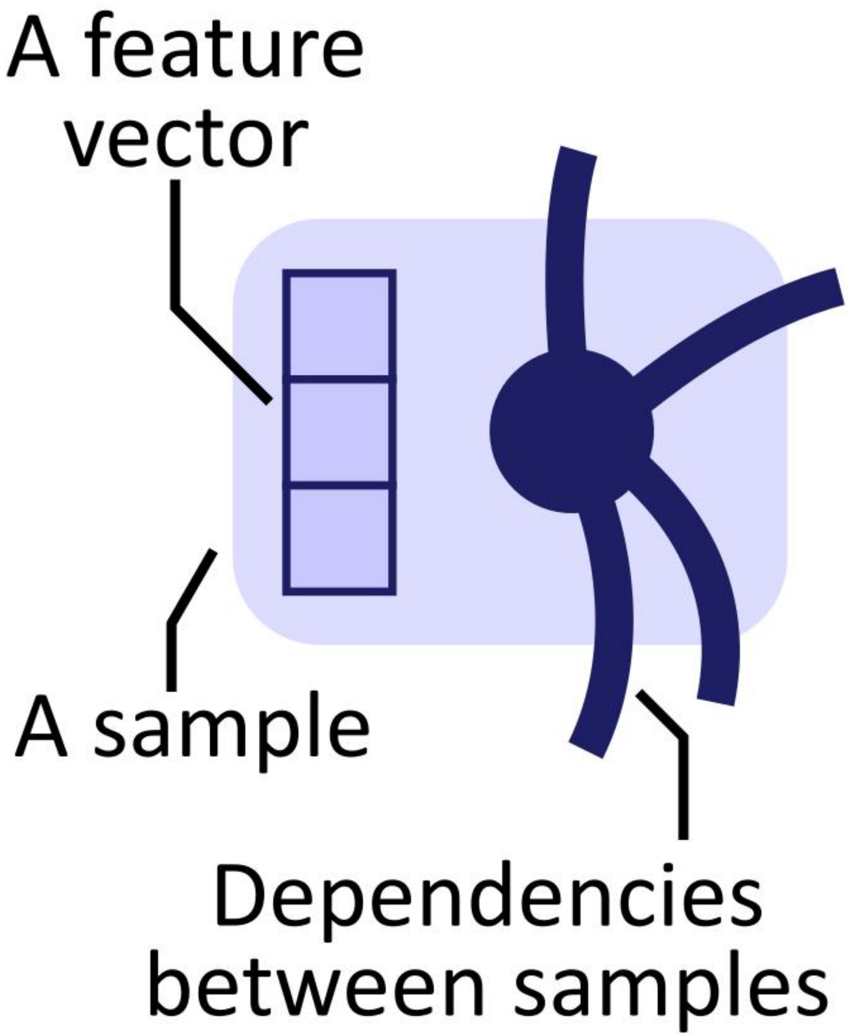
GNN driven Downstream ML tasks

Classification
Node Edge Graph

or

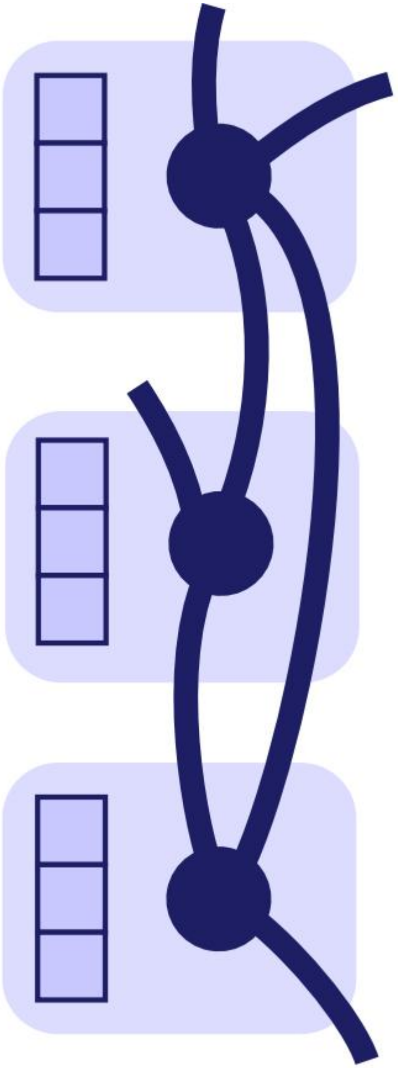
Regression
Node Edge Graph

Types of Samples & Downstream Tasks in GNNs



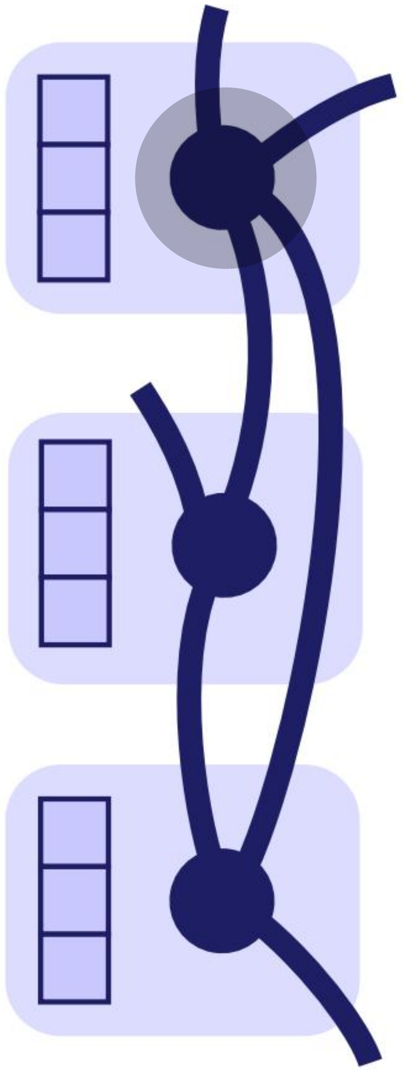
Types of Samples & Downstream Tasks in GNNs

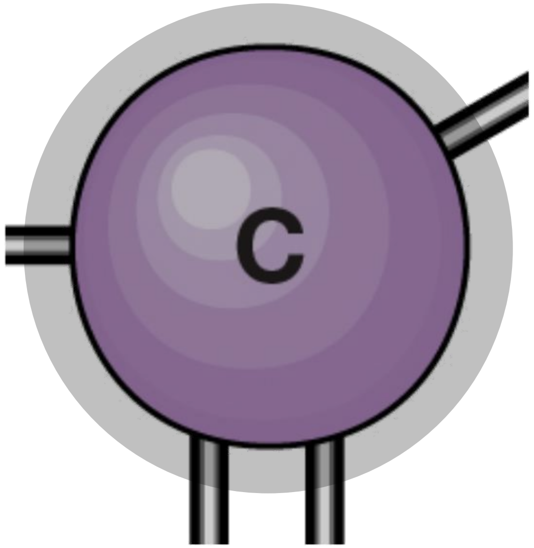
Vertices (dependent)



Types of Samples & Downstream Tasks in GNNs

Vertices (dependent)

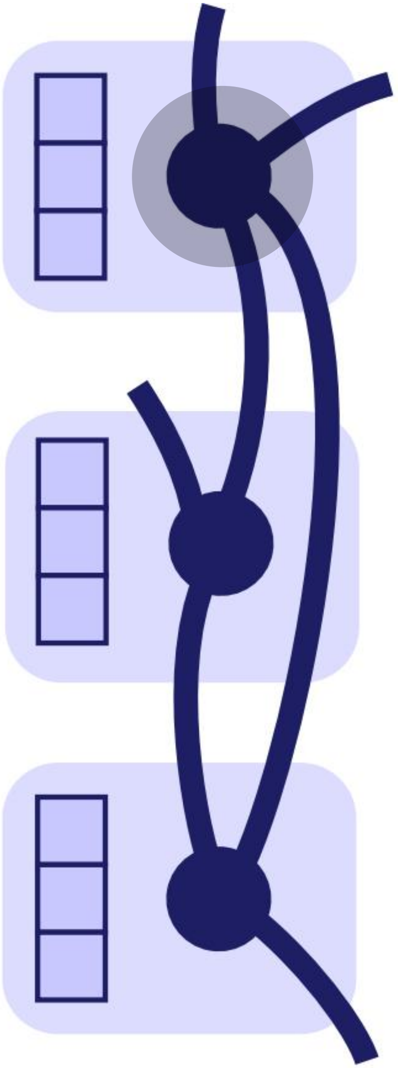


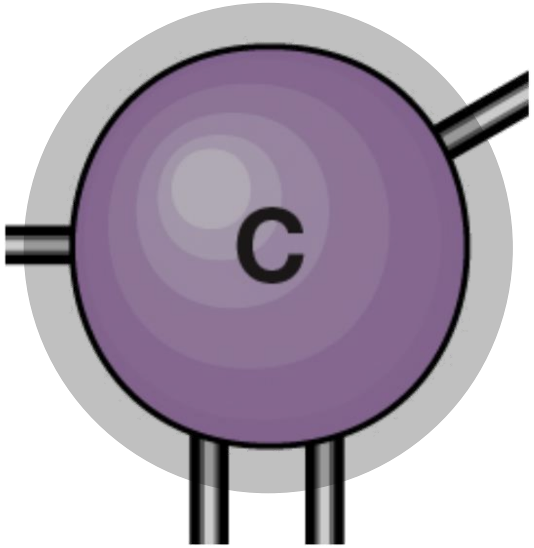


Example: atom

Types of Samples & Downstream Tasks in GNNs

Vertices (dependent)





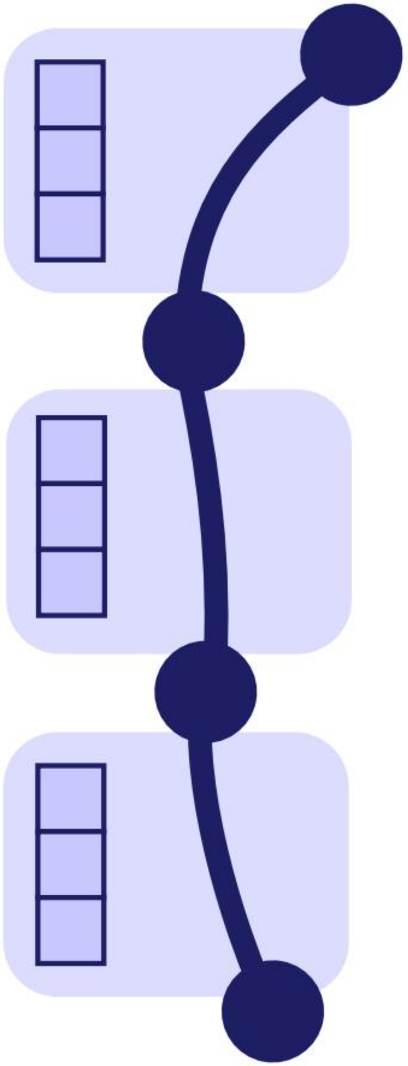
Example: atom

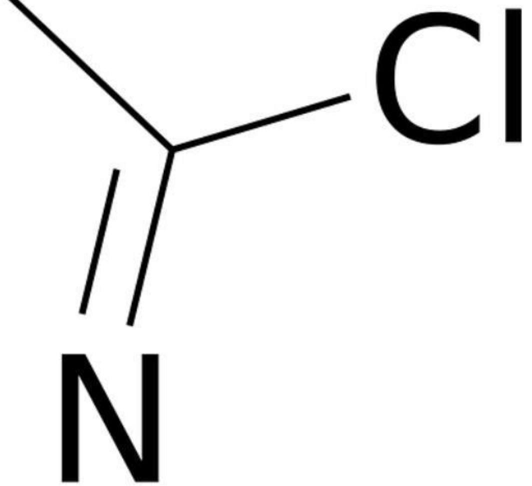
Example classification task: predict the atom element

Example regression task: predict the atom charge

Types of Samples & Downstream Tasks in GNNs

Edges (dependent)





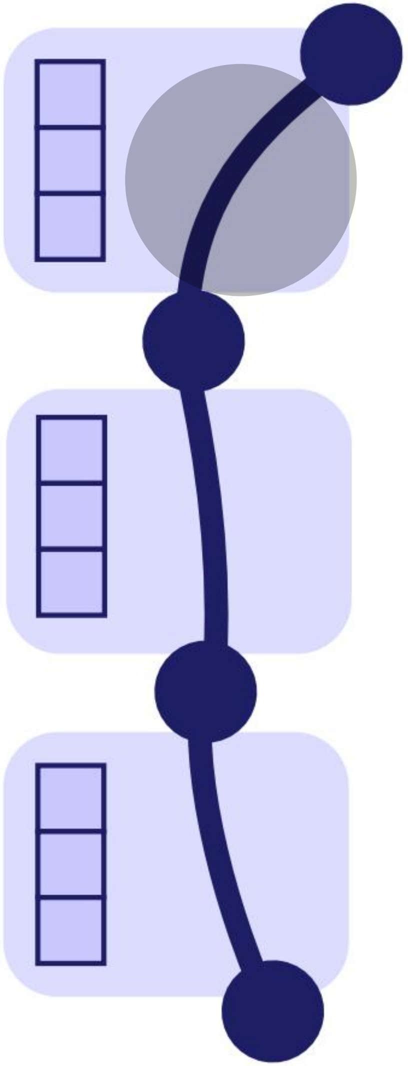
Example: atomic bond

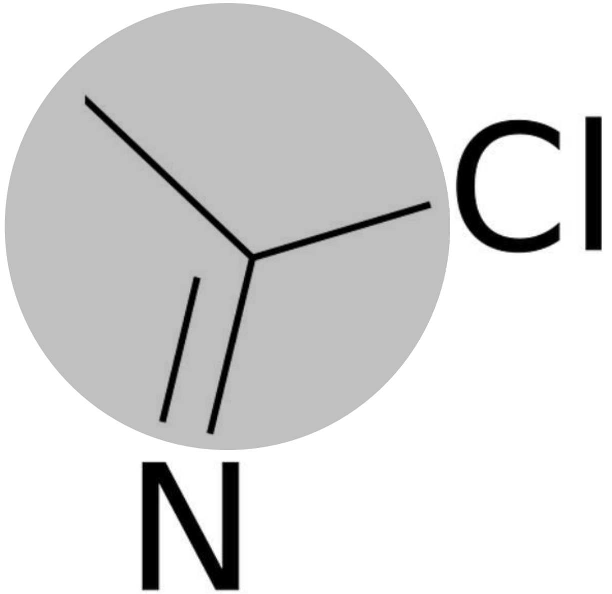
Example classification task: predict the bond type

Example regression task: predict the bond valence

Types of Samples & Downstream Tasks in GNNs

Edges (dependent)





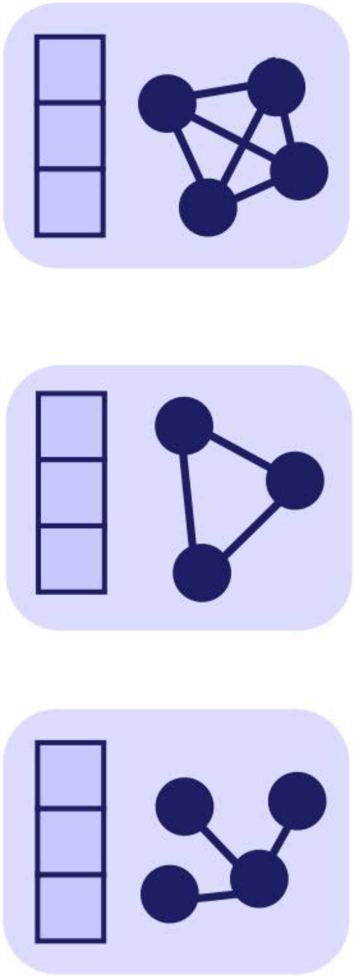
Example: atomic bond

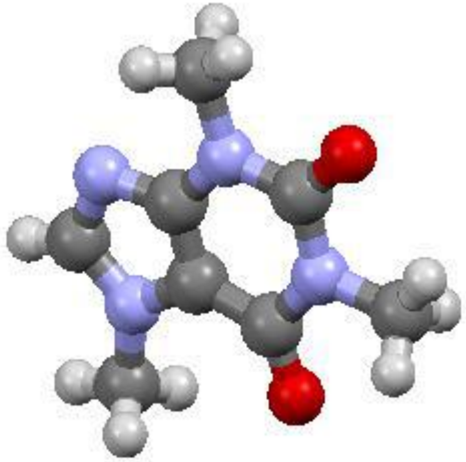
Example classification task: predict the bond type

Example regression task: predict the bond valence

Types of Samples & Downstream Tasks in GNNs

Graphs (independent)





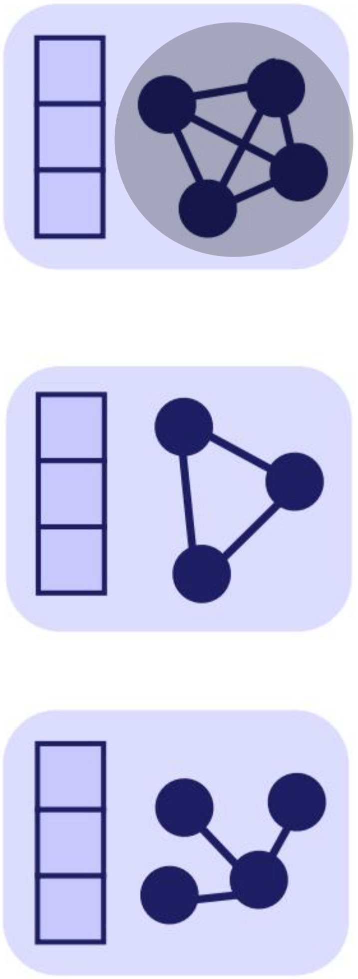
Example: chemical molecule

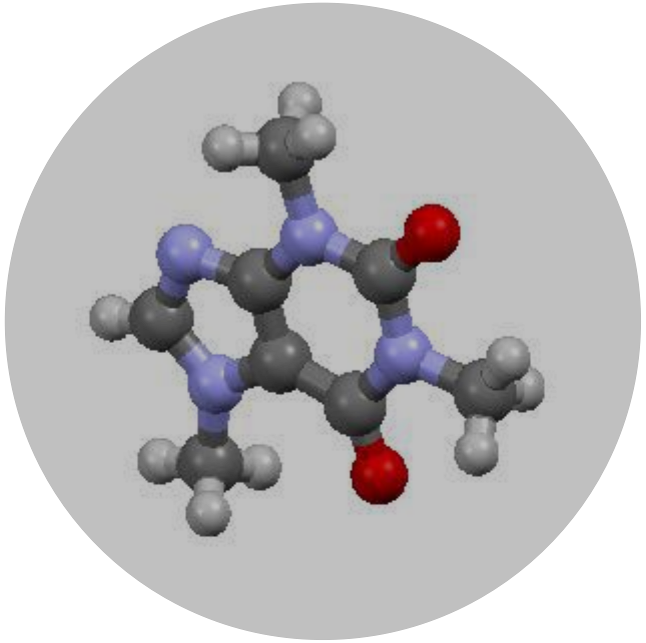
Example classification task: predict the class of a molecule

Example regression task: predict the solubility of a molecule

Types of Samples & Downstream Tasks in GNNs

Graphs (independent)





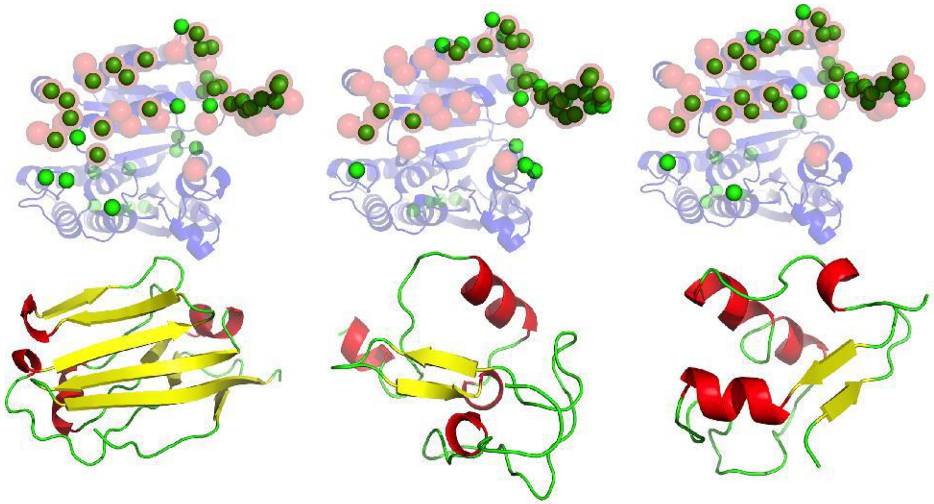
Example: chemical molecule

Example classification task: predict the class of a molecule

Example regression task: predict the solubility of a molecule

Types of Samples & Downstream Tasks in GNNs

Graphs (dependent)



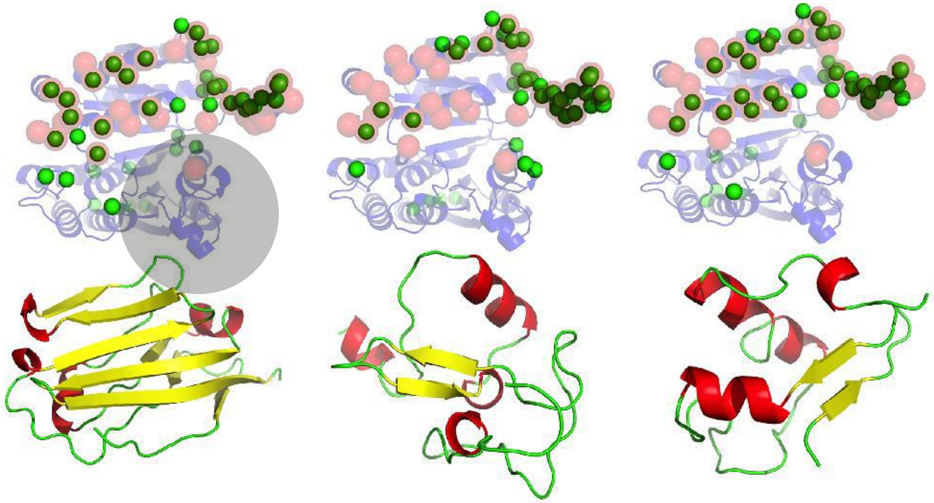
Example: interacting protein

Example classification task: predict the protein type

Example regression task: predict molecular weight

Types of Samples & Downstream Tasks in GNNs

Graphs (dependent)

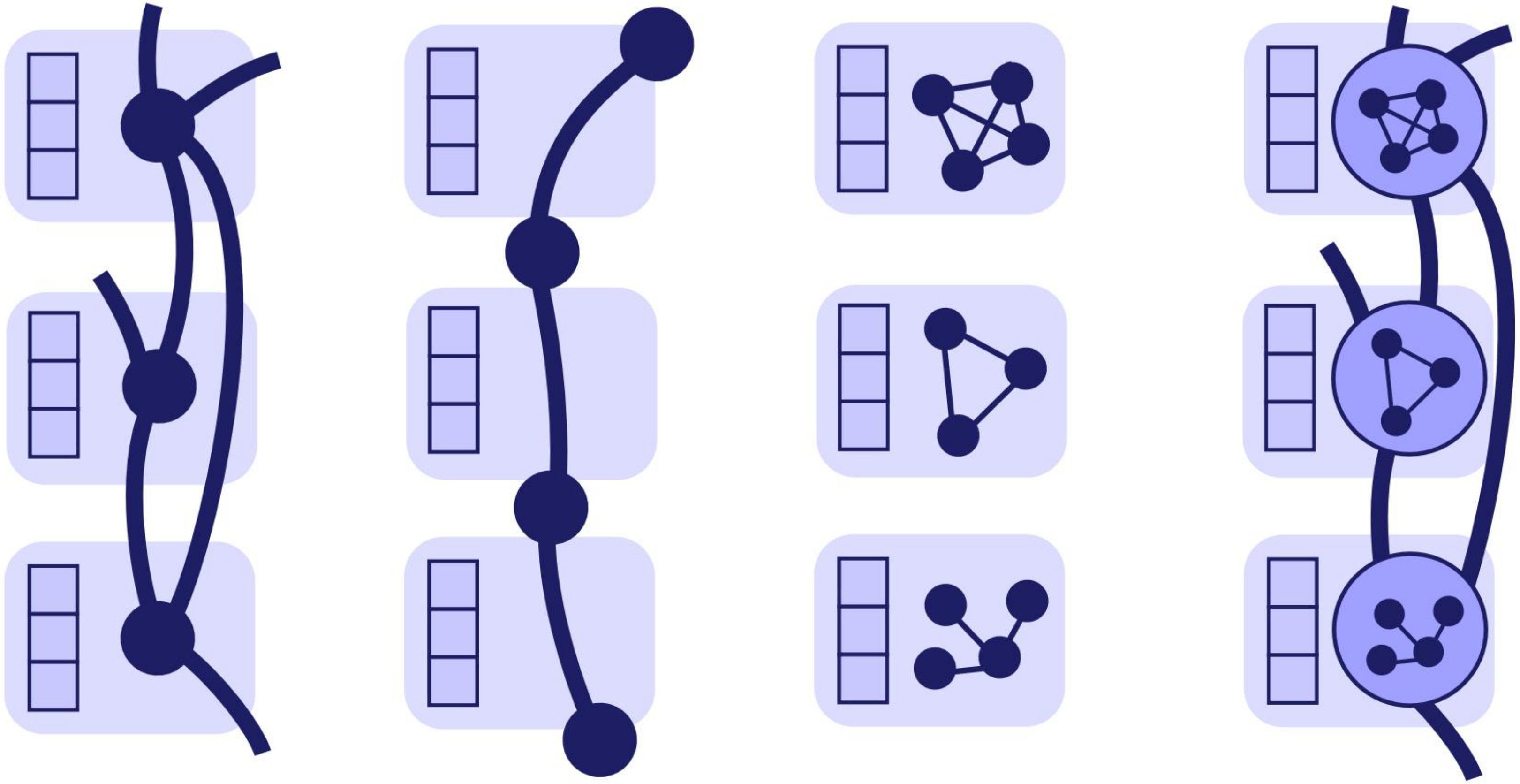


Example: interacting protein

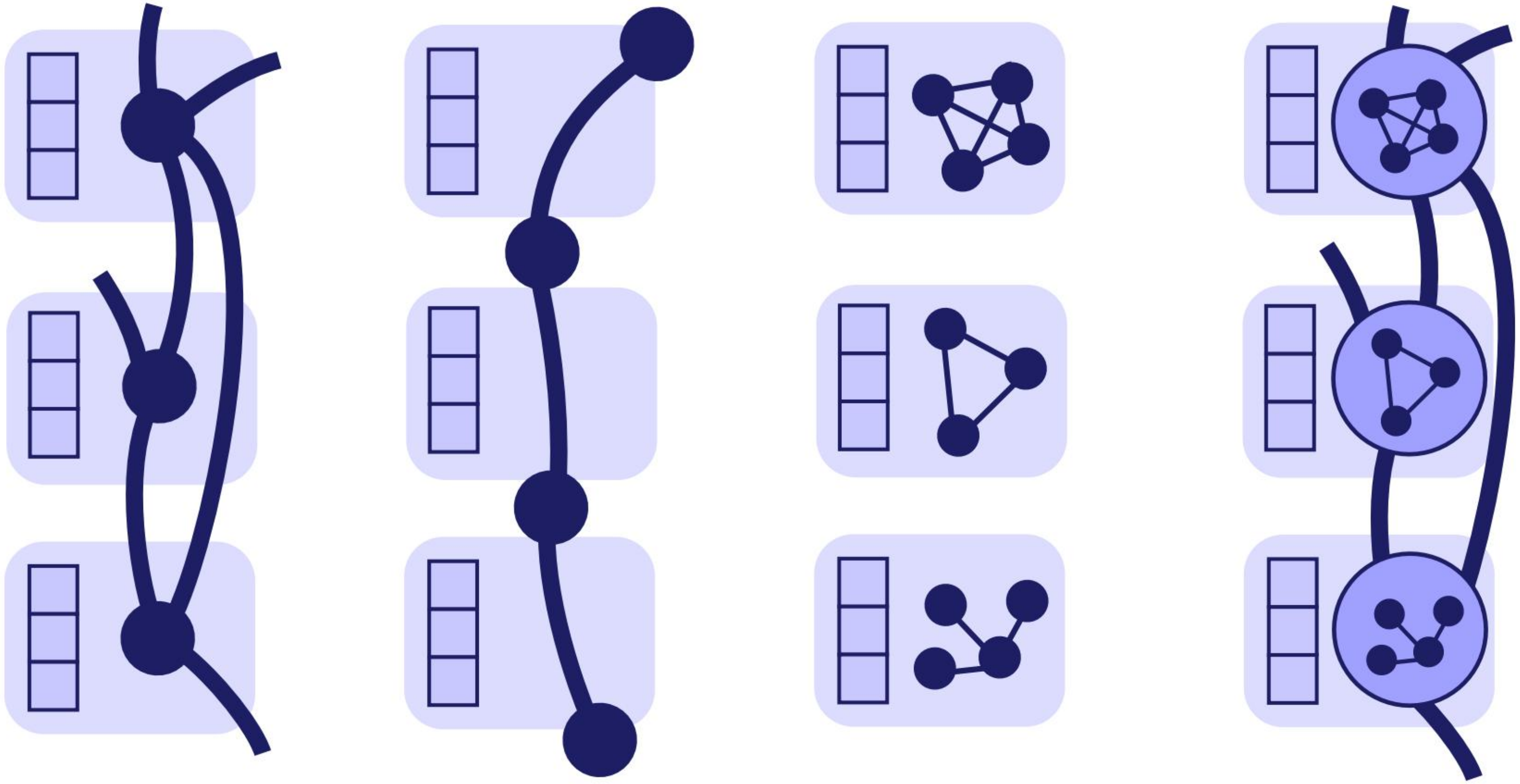
Example classification task: predict the protein type

Example regression task: predict molecular weight

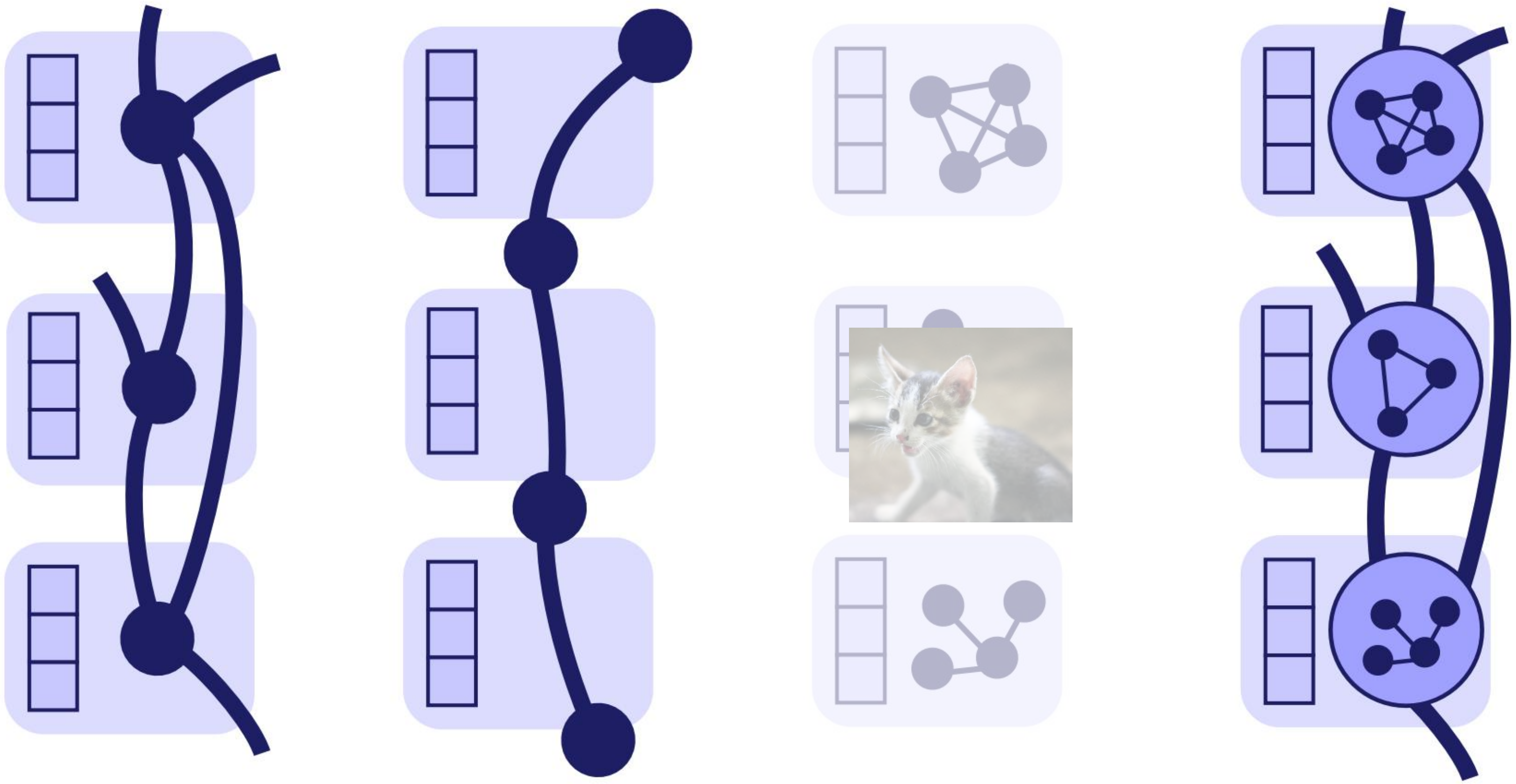
Types of Samples & Downstream Tasks: GNNs vs. Traditional DL



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

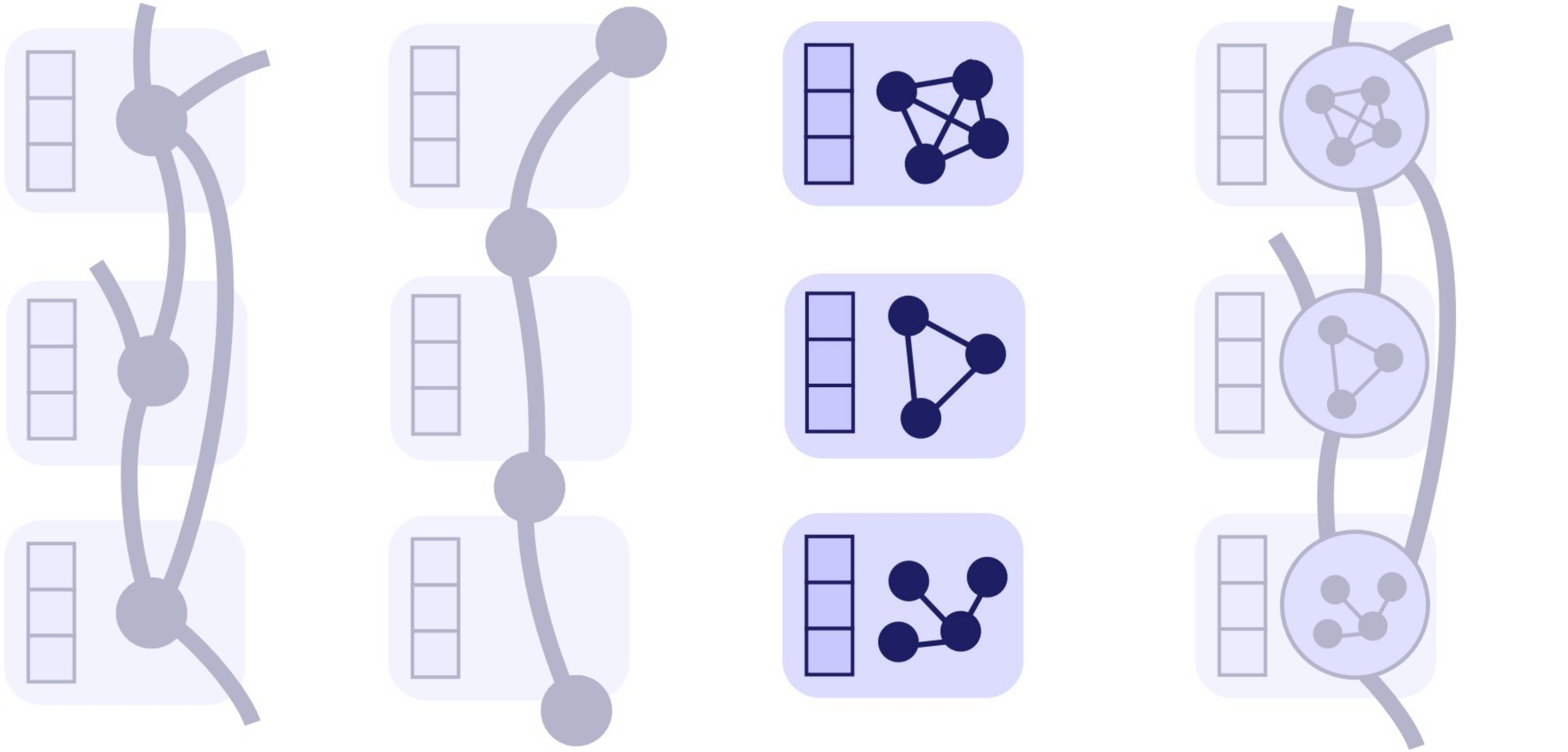


Types of Samples & Downstream Tasks: GNNs vs. Traditional DL



Dependencies between samples in GNNs

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



Dependencies between samples in GNNs

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

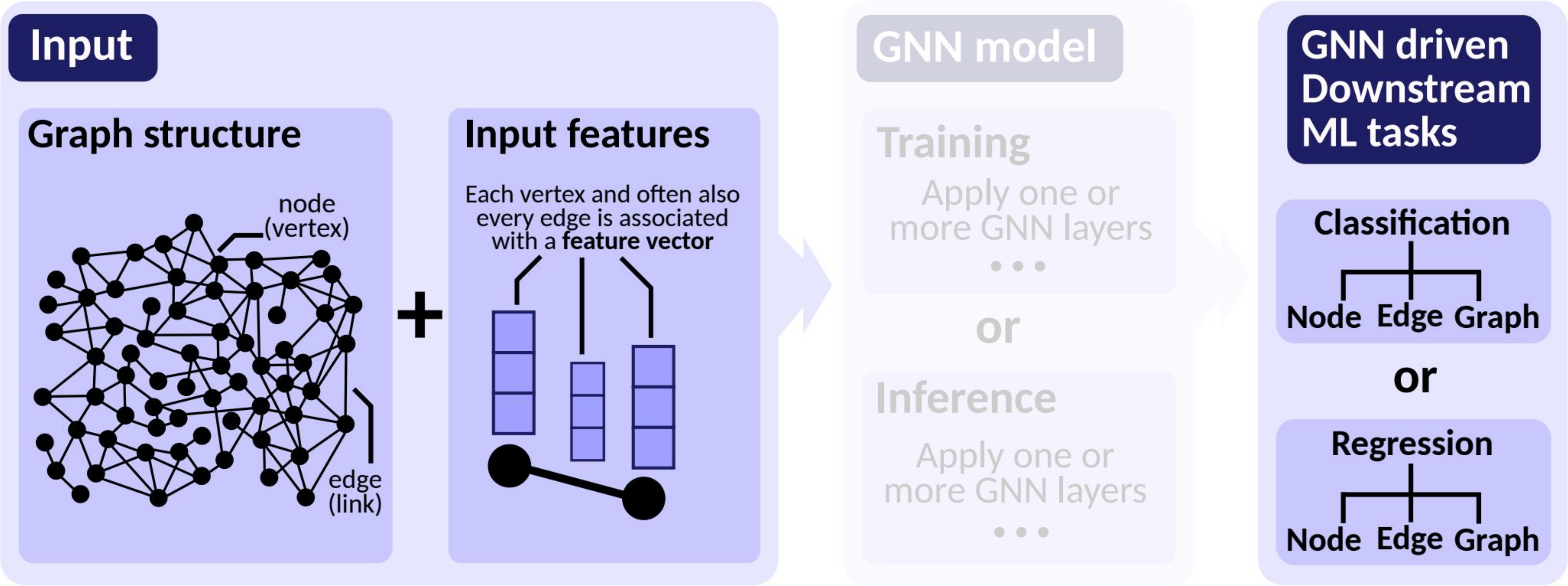
Types of Samples & Downstream Tasks: GNNs vs. Traditional DL

As we'll see later, these dependencies make parallelism in GNNs **much more complex** than in traditional DL

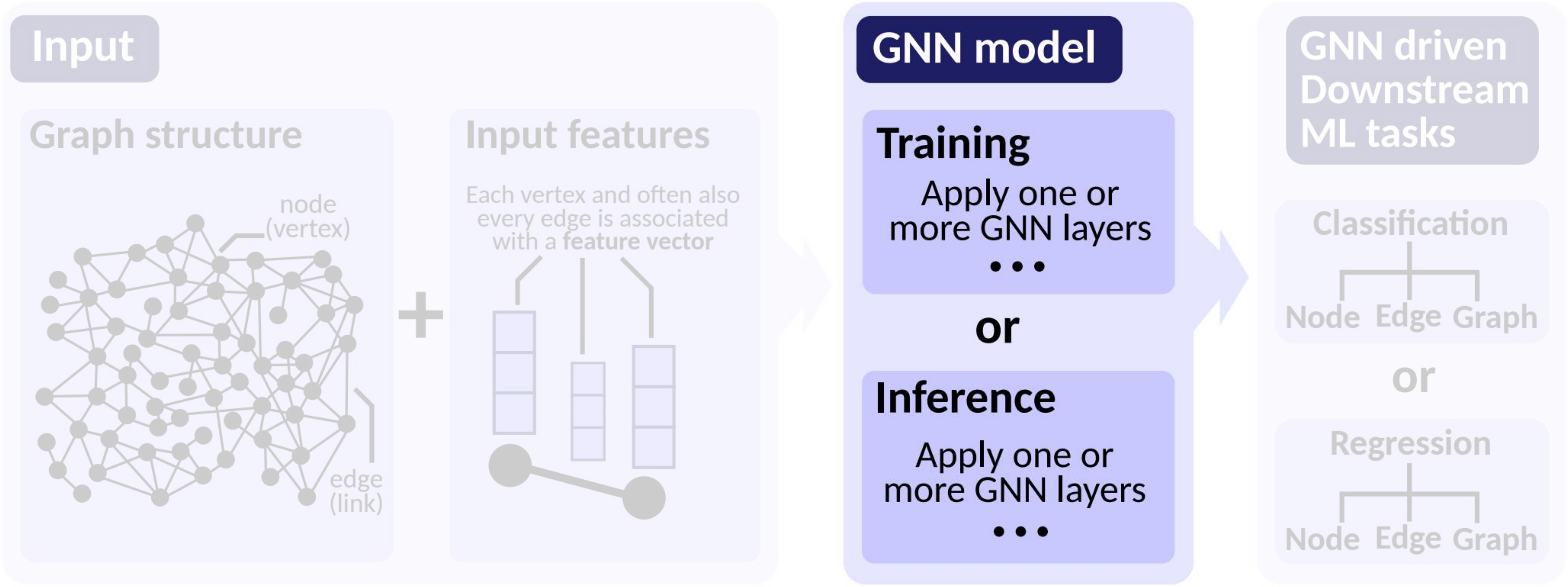
Dependencies between samples in GNNs

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

Overview of a GNN Computation

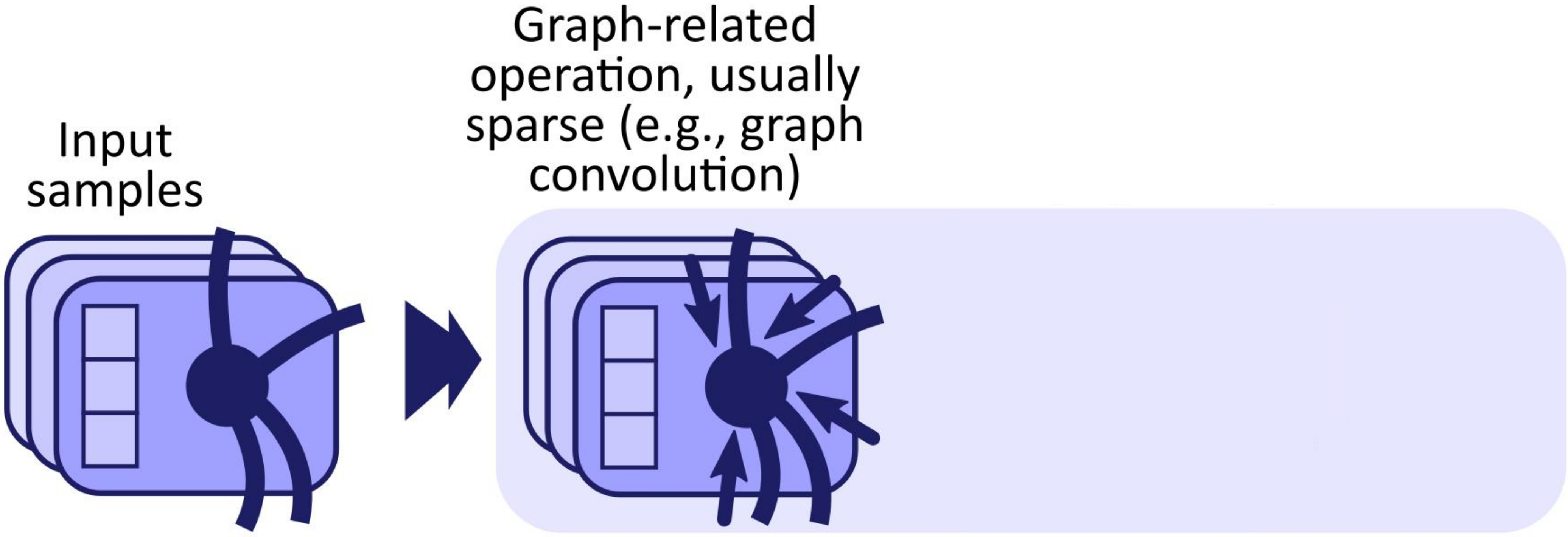


Overview of a GNN Computation

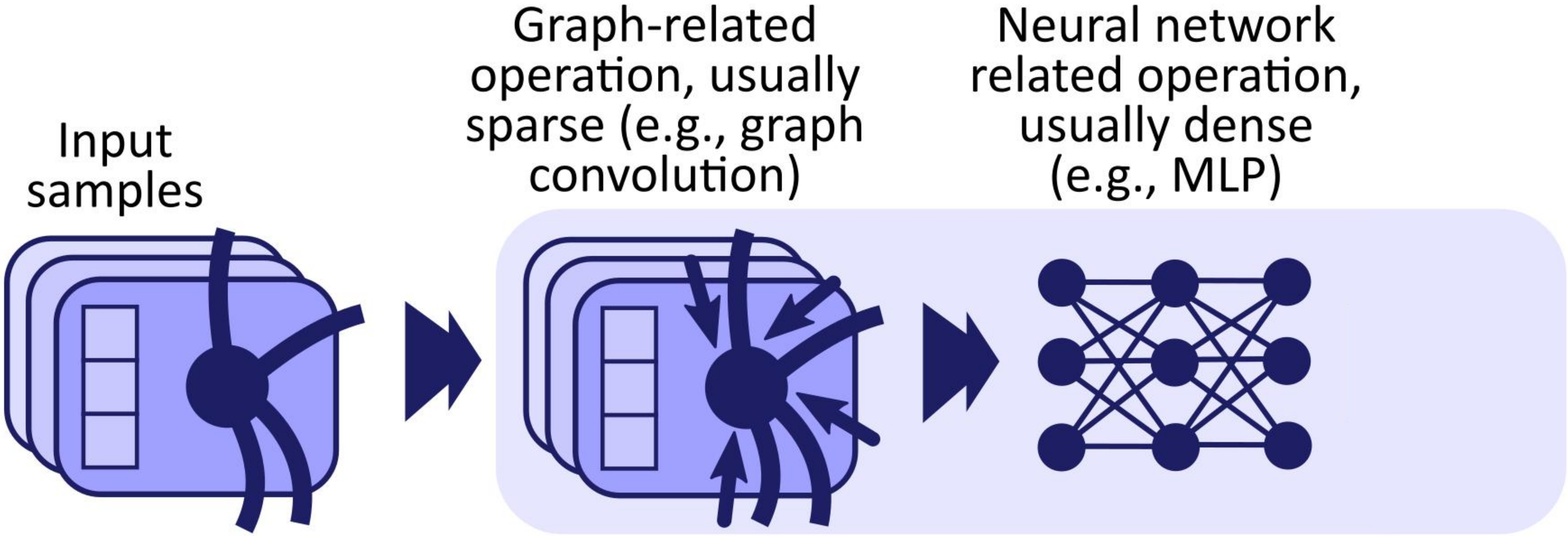


A Single GNN Layer

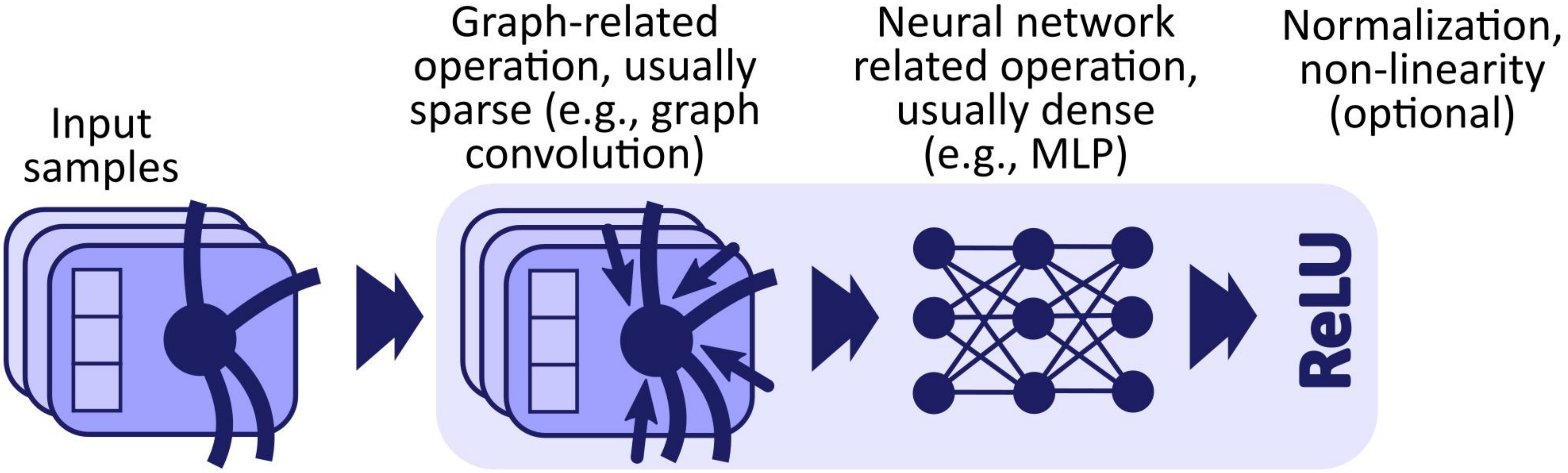
A Single GNN Layer



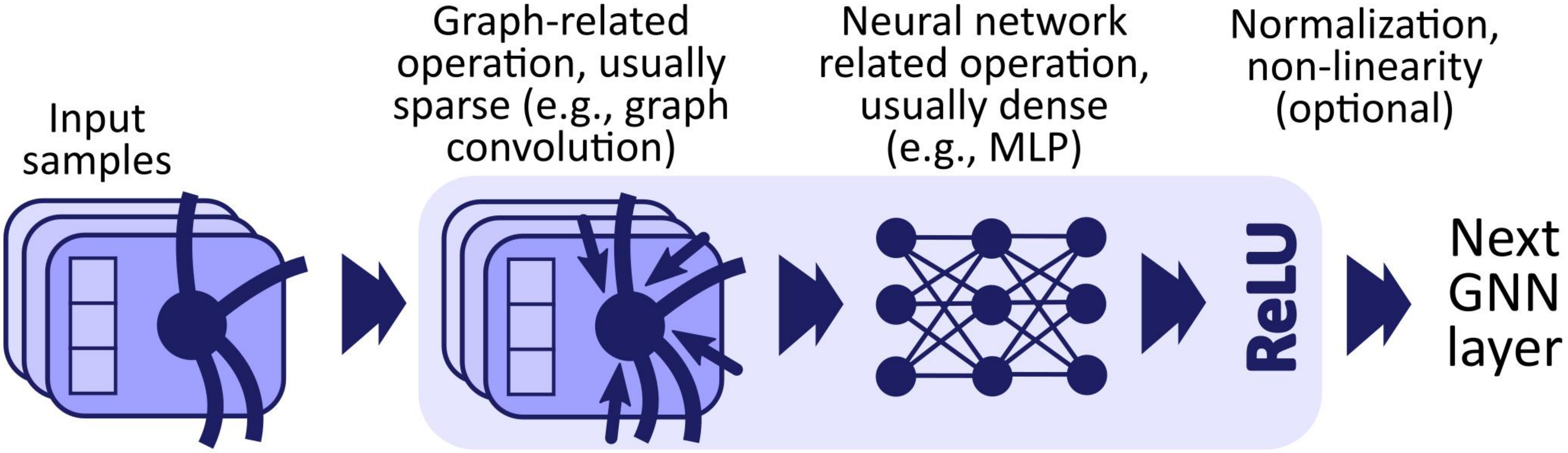
A Single GNN Layer



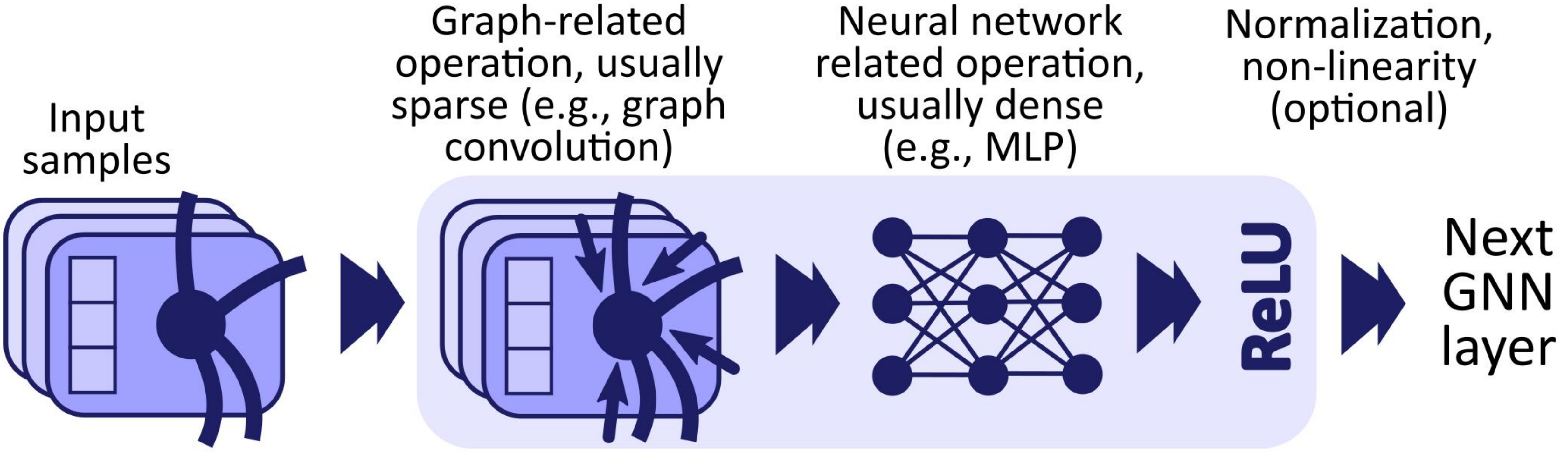
A Single GNN Layer



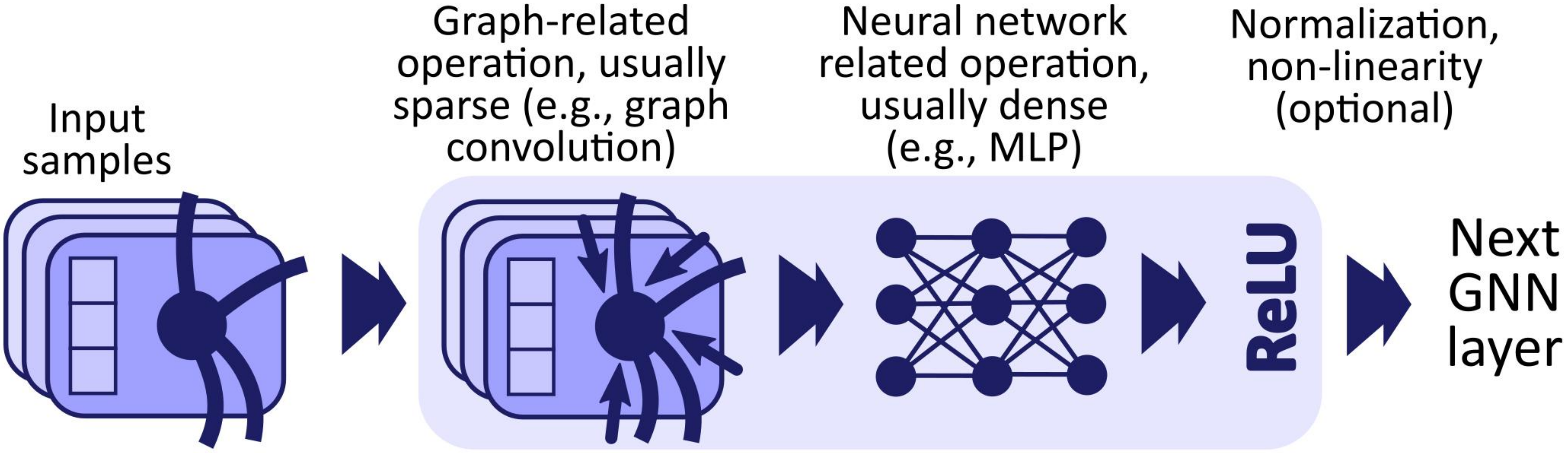
A Single GNN Layer

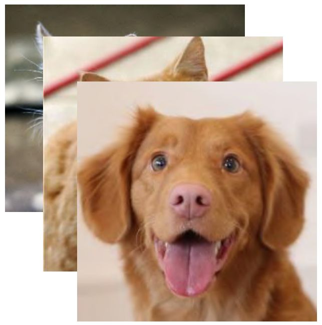


Layers: GNNs vs. Traditional DL



Layers: GNNs vs. Traditional DL





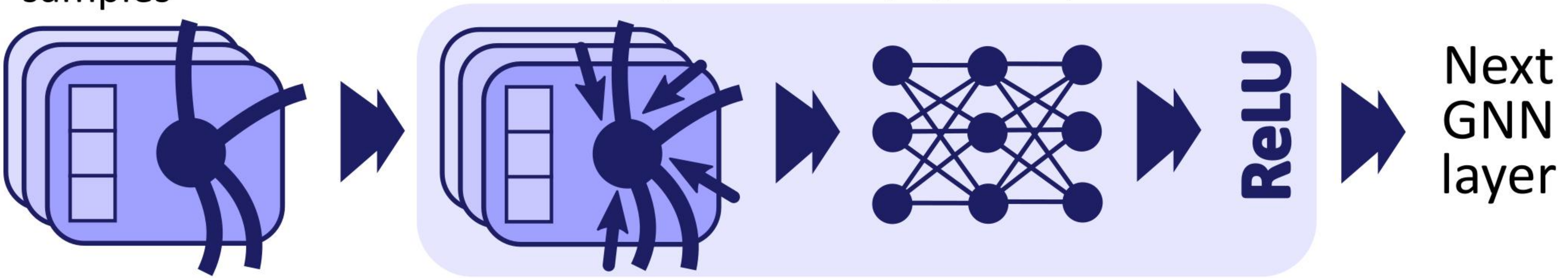
Layers: GNNs vs. Traditional DL

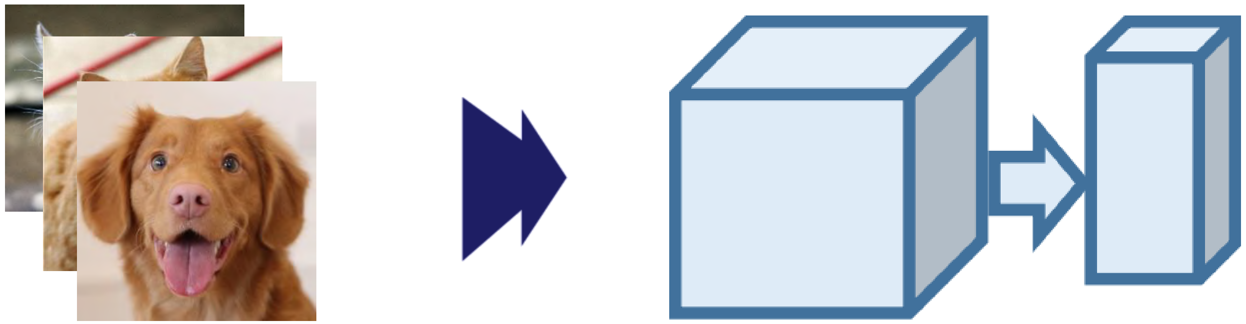
Input samples

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

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

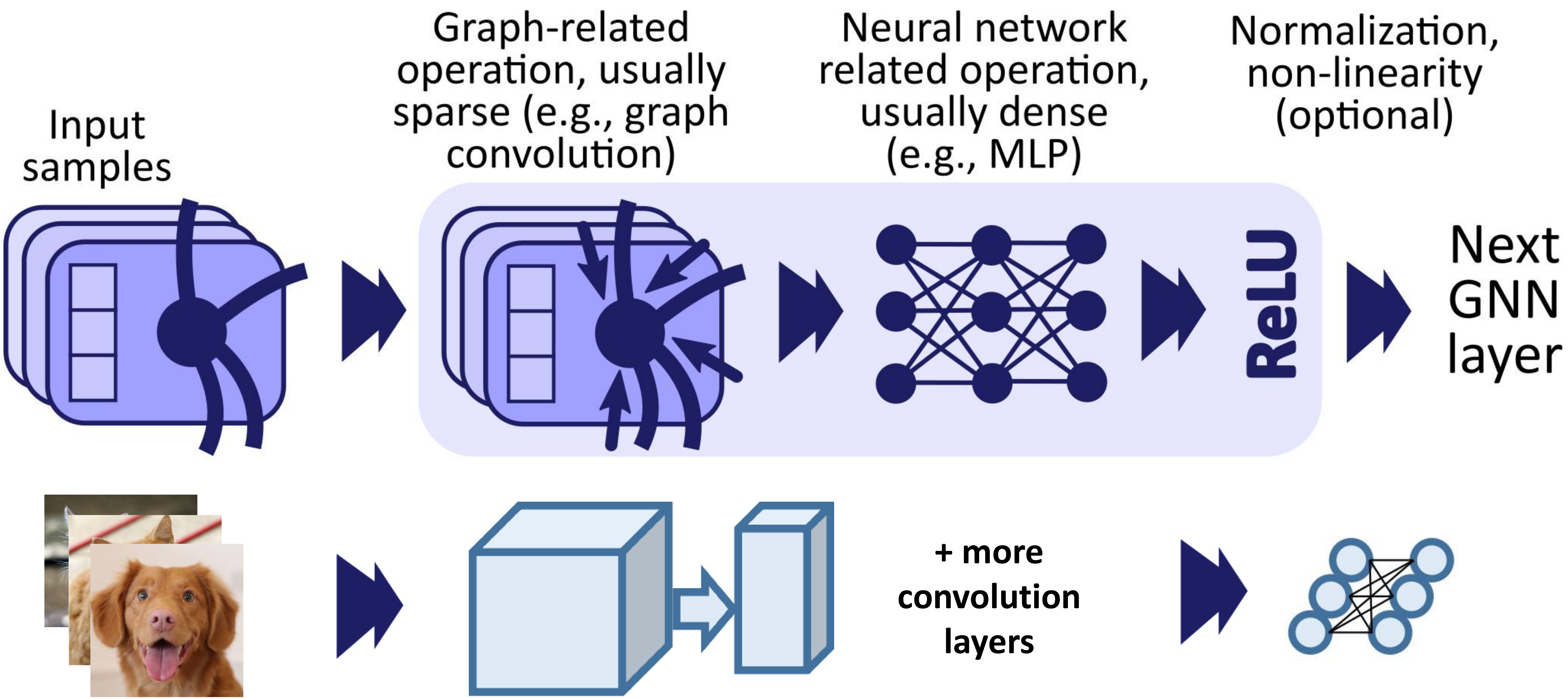
Normalization, non-linearity (optional)



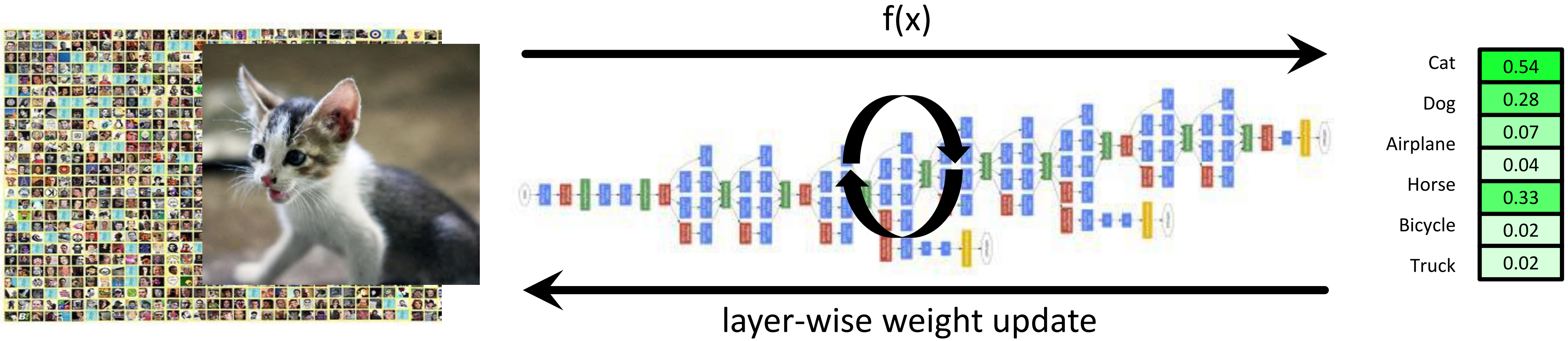


+ more convolution layers

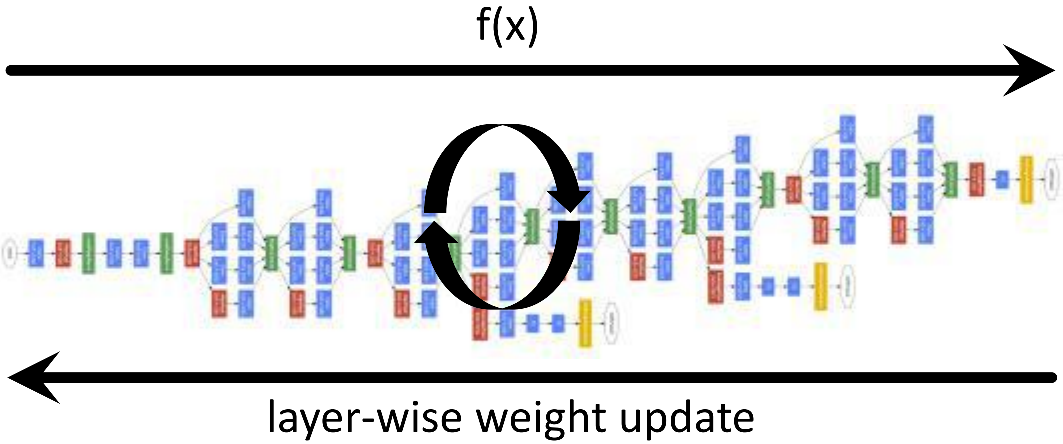
Layers: GNNs vs. Traditional DL



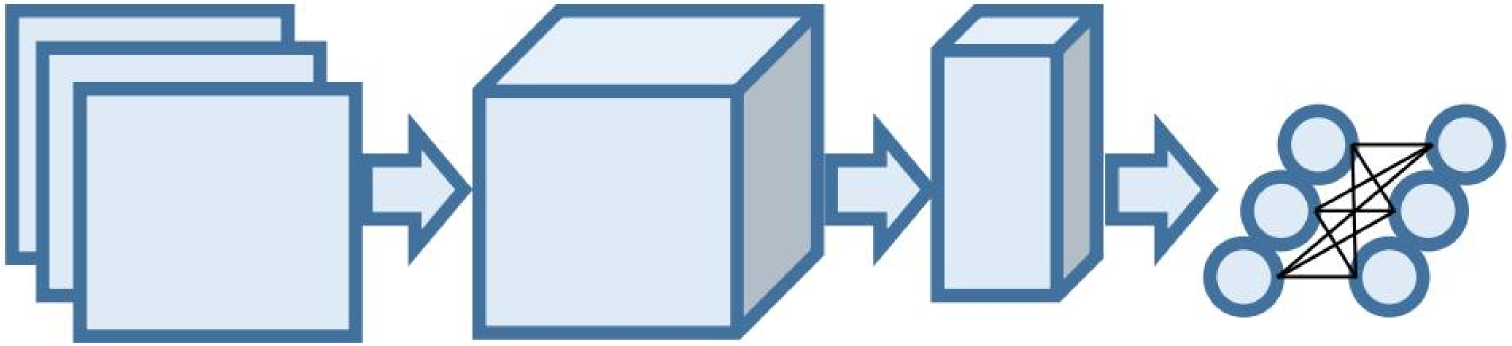
Parallelism in Traditional Deep Learning



Parallelism in Traditional Deep Learning



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



Parallelism in Traditional Deep Learning

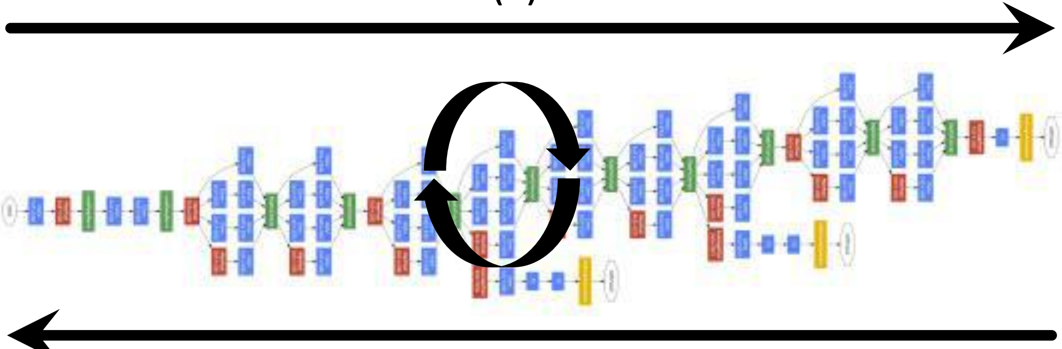
Data

Parallelism in Traditional Deep Learning

Data

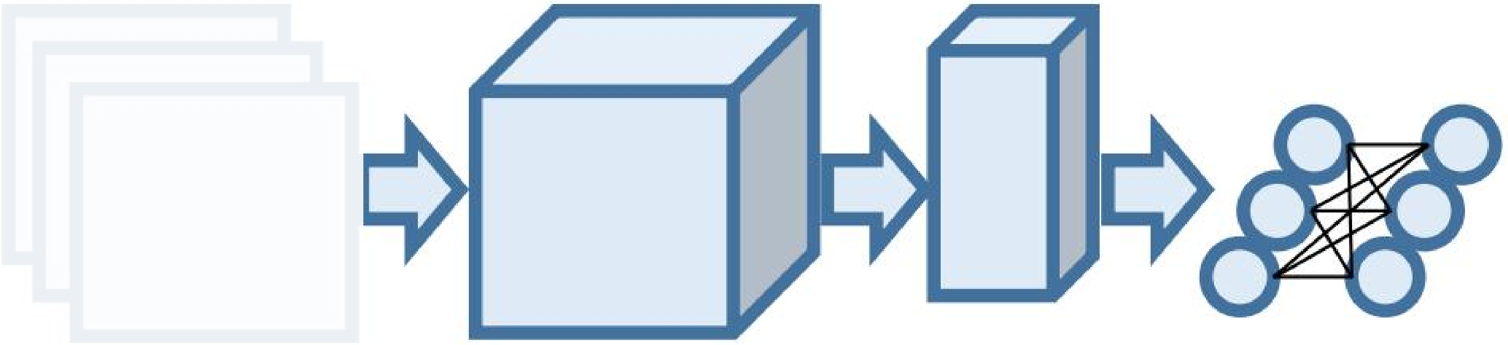
Model

f(x)



layer-wise weight update

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



Parallelism in Traditional Deep Learning

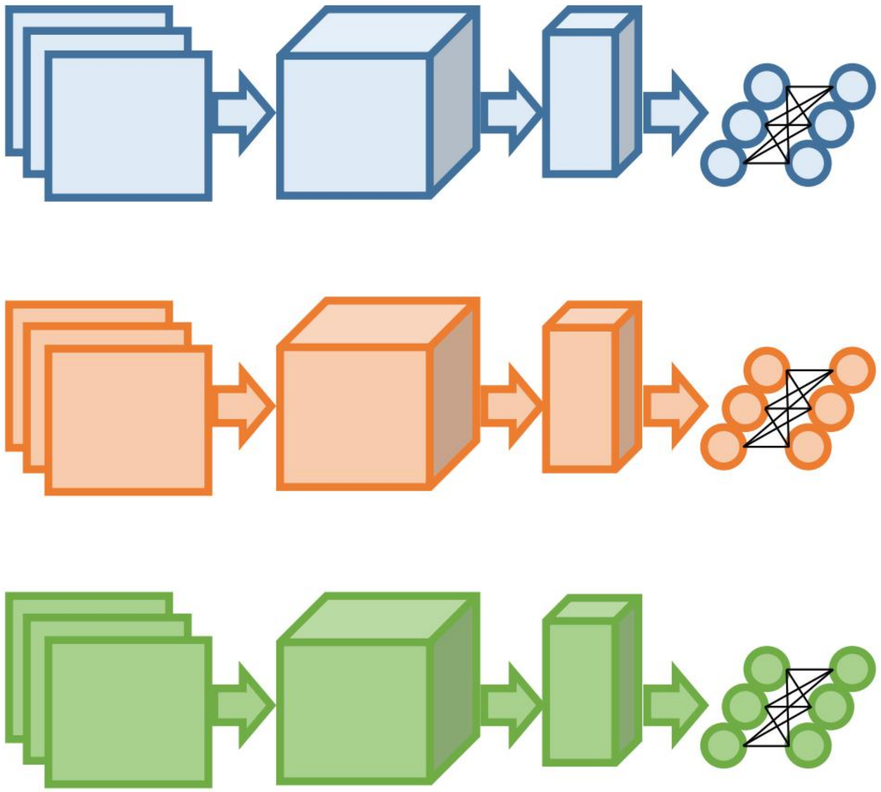
Different colors correspond to different (parallel) workers

Pictures taken from: “Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis”, T. Ben-Nun, T. Hoefler, ACM CSUR, 2018

Parallelism in Traditional Deep Learning

Different colors correspond to different (parallel) workers

Data parallelism

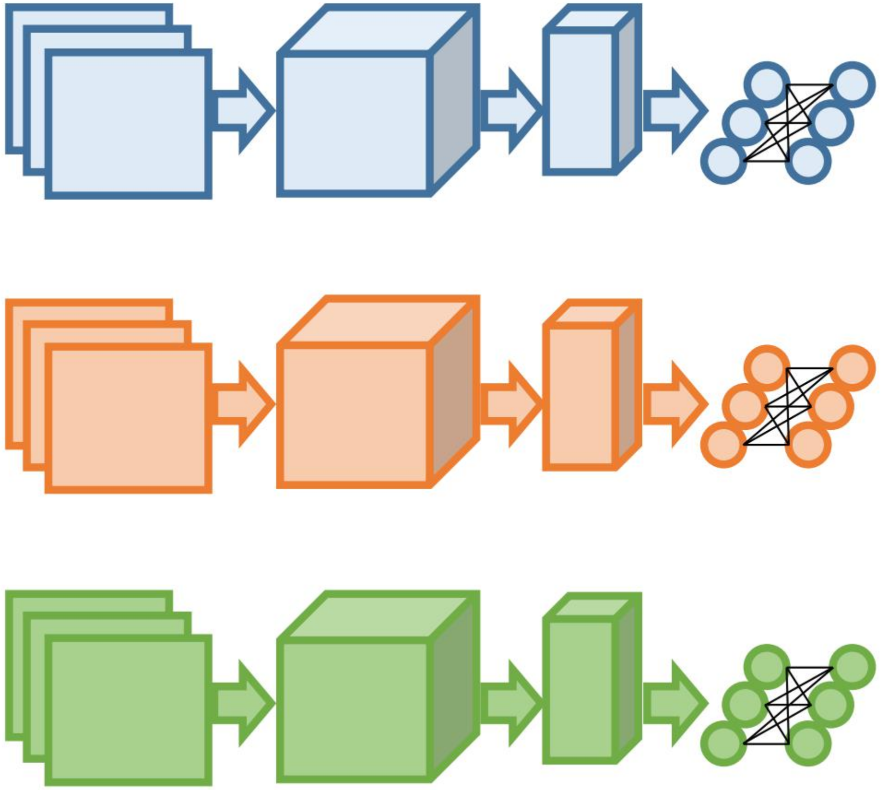


Pictures taken from: “Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis”, T. Ben-Nun, T. Hoefler, ACM CSUR, 2018

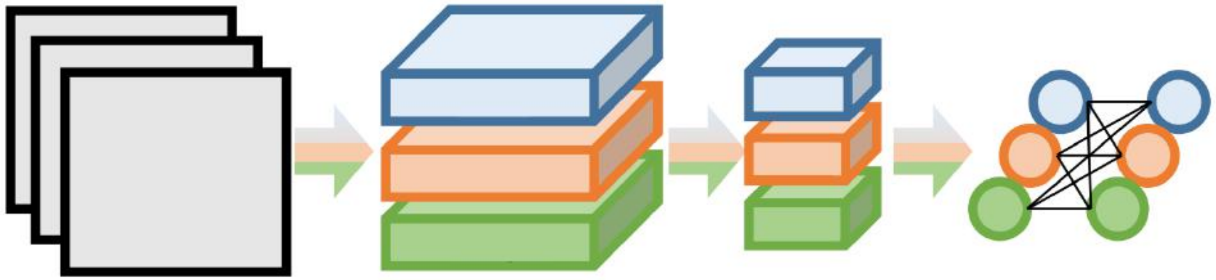
Parallelism in Traditional Deep Learning

Different colors correspond to different (parallel) workers

Data parallelism



Model parallelism

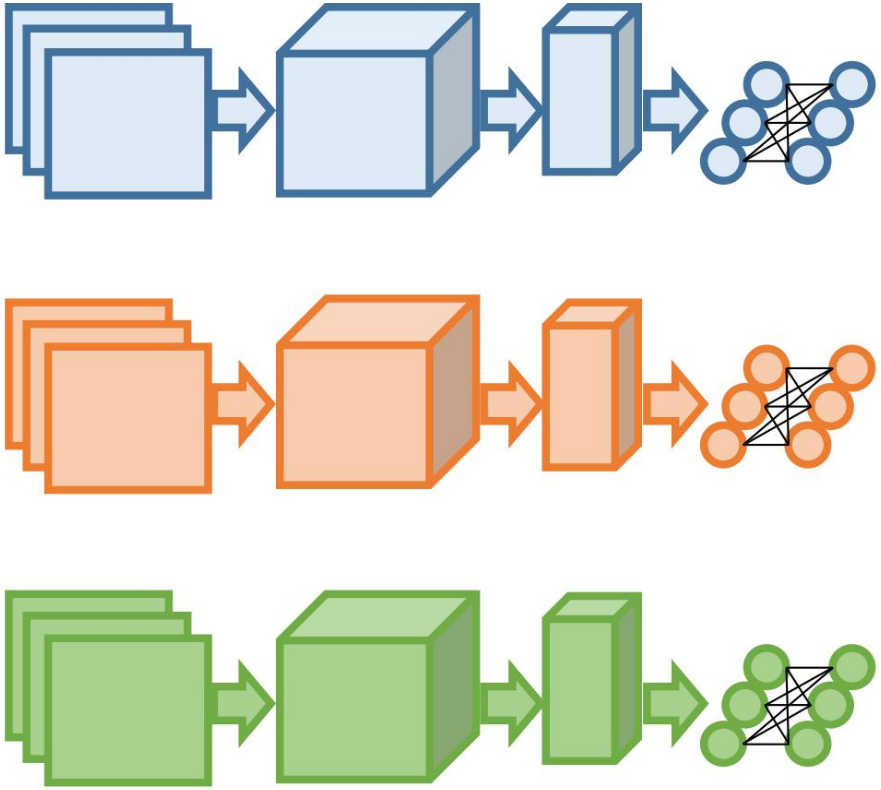


Pictures taken from: "Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis", T. Ben-Nun, T. Hoefler, ACM CSUR, 2018

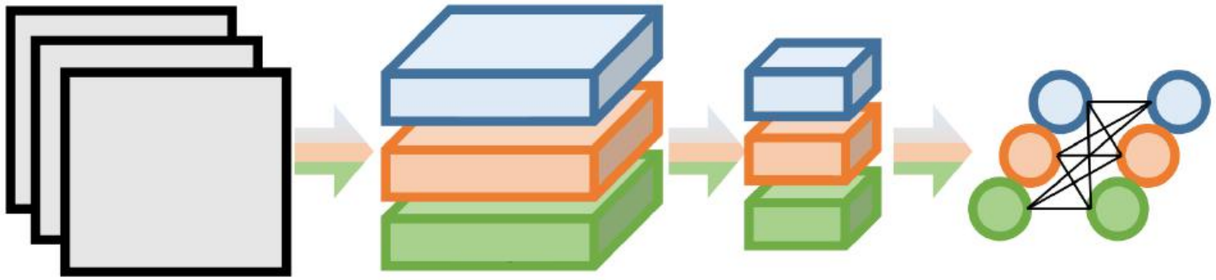
Parallelism in Traditional Deep Learning

Different colors correspond to different (parallel) workers

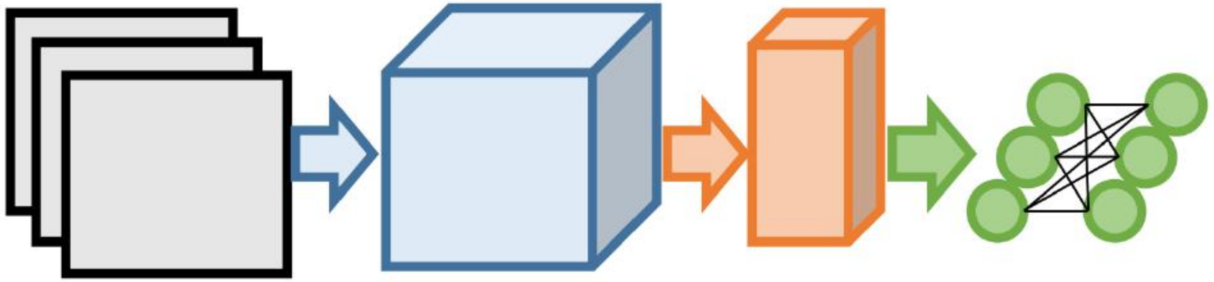
Data parallelism



Model parallelism



Pipeline parallelism

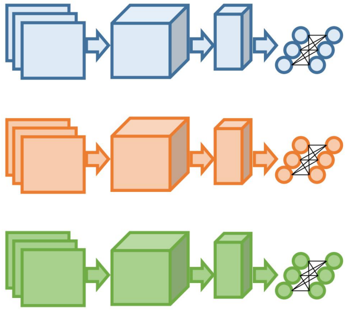


Pictures taken from: "Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis", T. Ben-Nun, T. Hoefler, ACM CSUR, 2018

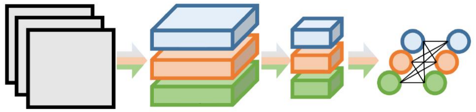
Parallelism in Traditional Deep Learning vs. GNNs

Different colors correspond to different (parallel) workers

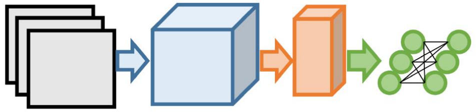
Data parallelism



Model parallelism



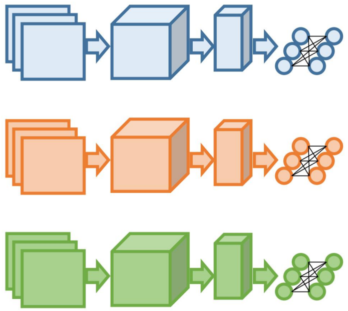
Pipeline parallelism



Parallelism in Traditional Deep Learning vs. GNNs

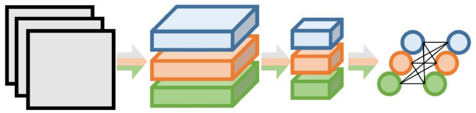
Different colors correspond to different (parallel) workers

Data parallelism



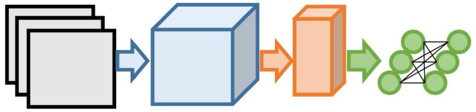
Data parallelism

Model parallelism



Model parallelism

Pipeline parallelism

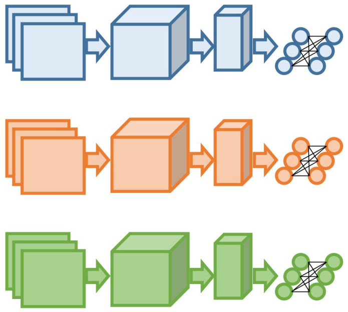


Pipeline parallelism

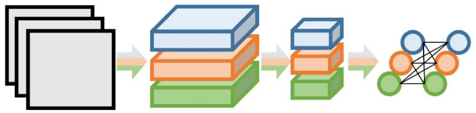
Parallelism in Traditional Deep Learning vs. GNNs

Different colors correspond to different (parallel) workers

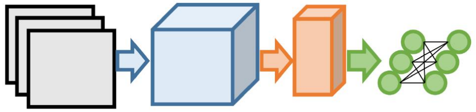
Data parallelism

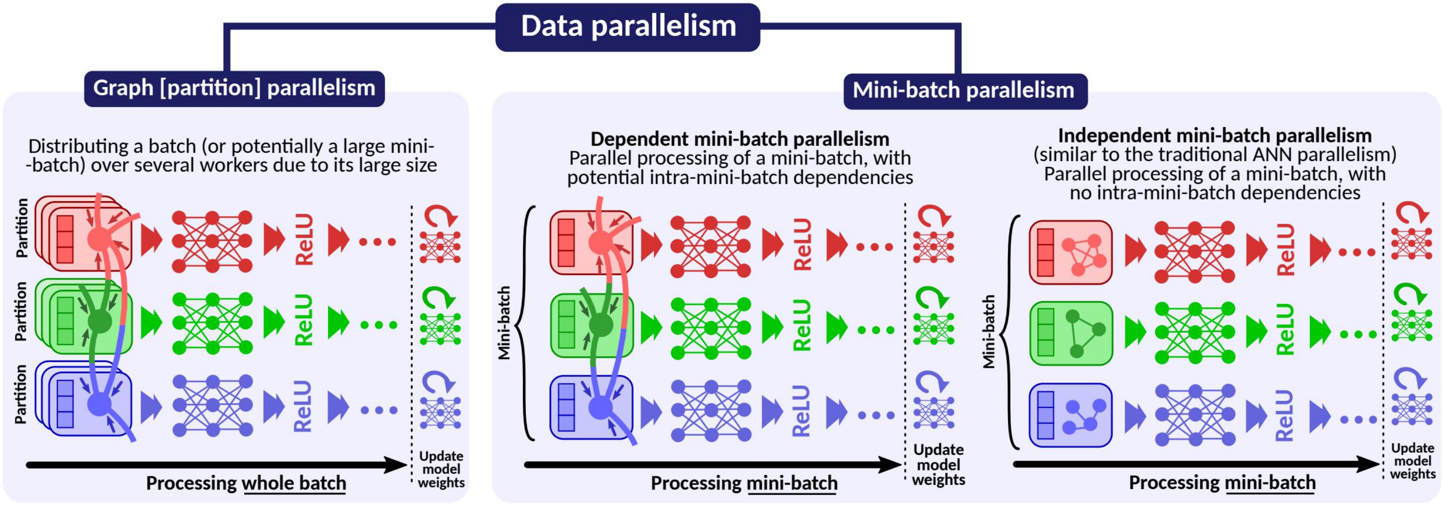


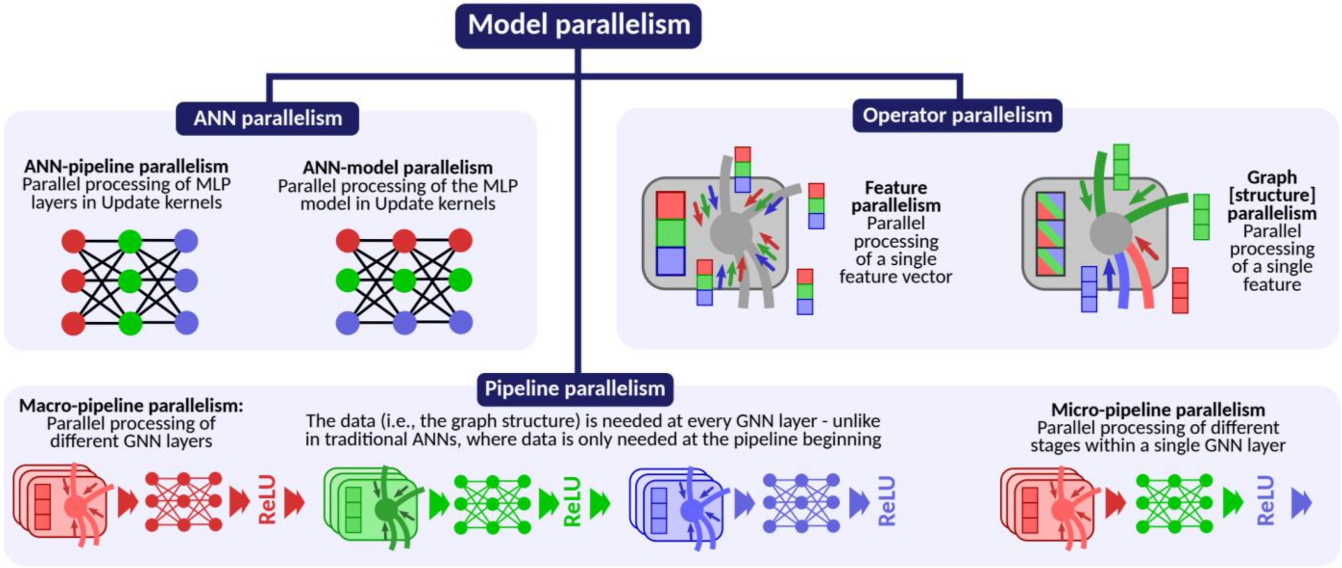
Model parallelism



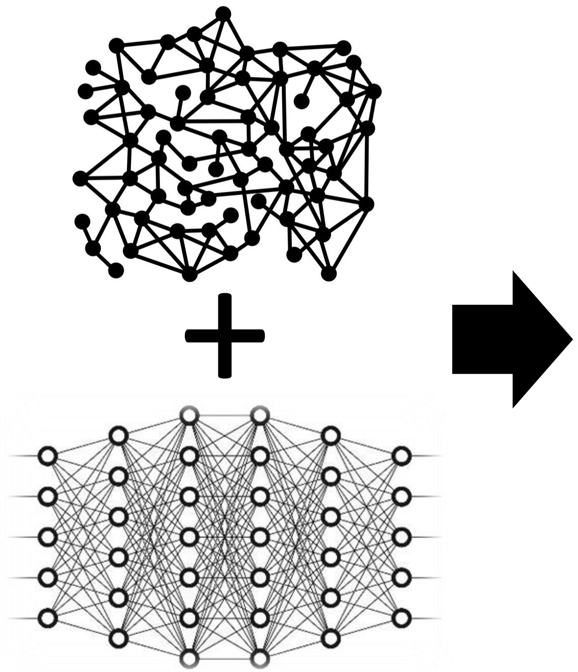
Pipeline parallelism







Presentation Overview

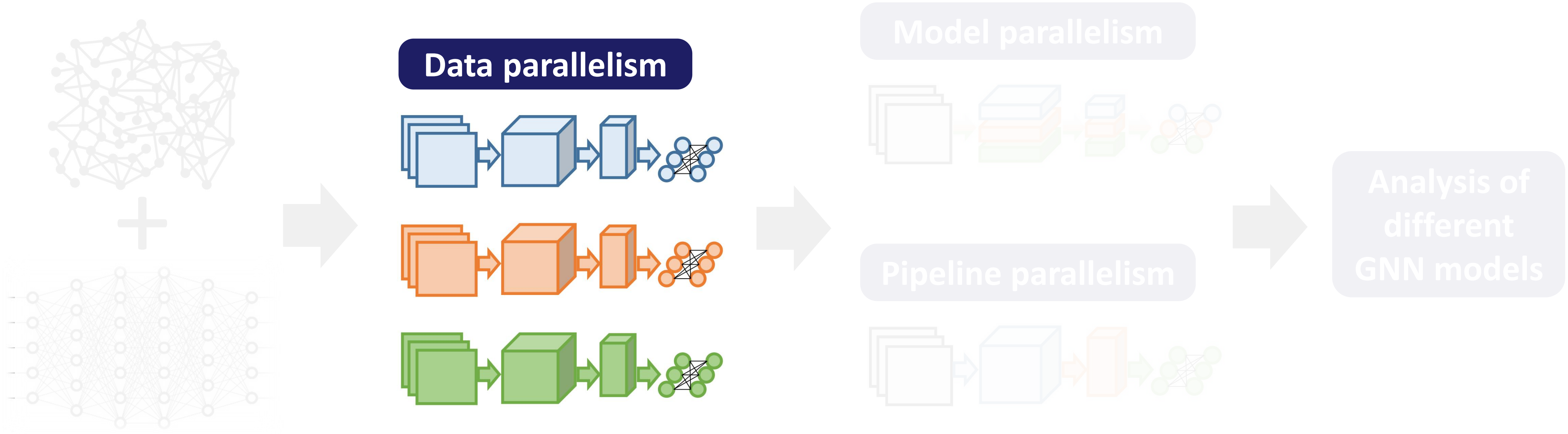


1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

Presentation Overview

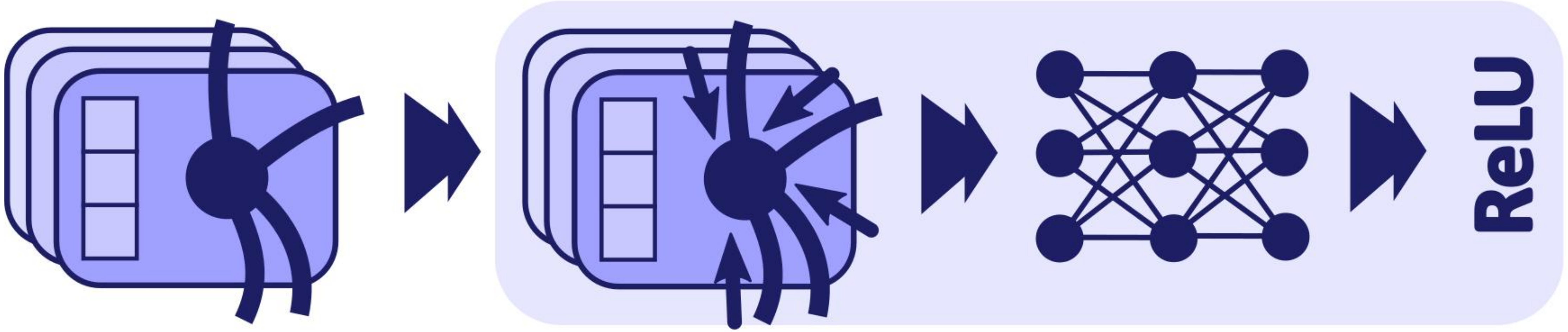


1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

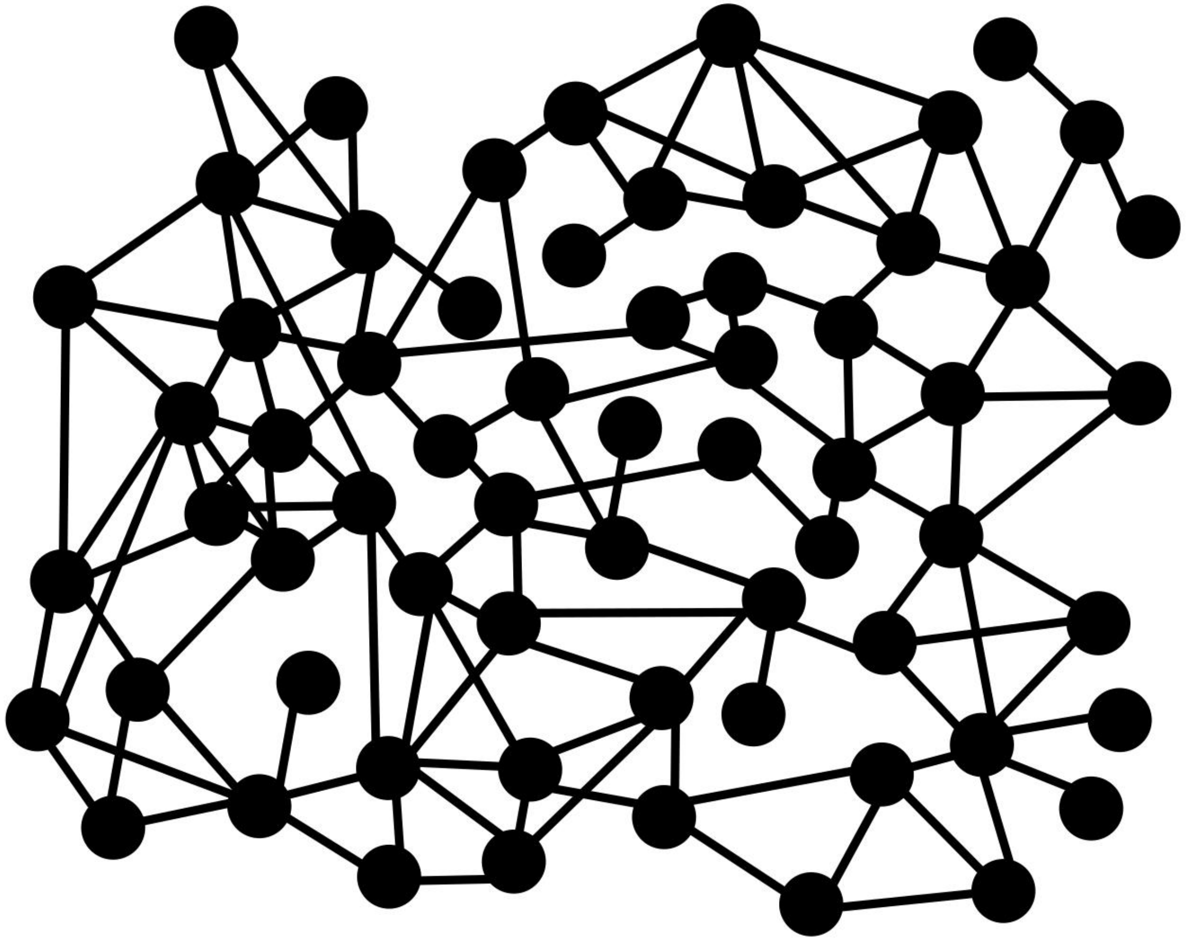
- Data parallelism
  - Graph [partition] parallelism
  - Mini-batch parallelism
    - Independent mini-batch parallelism
    - Dependent mini-batch parallelism



- **Data parallelism**
  - **Graph [partition] parallelism**
  - **Mini-batch parallelism**
    - **Independent mini-batch parallelism**
    - **Dependent mini-batch parallelism**

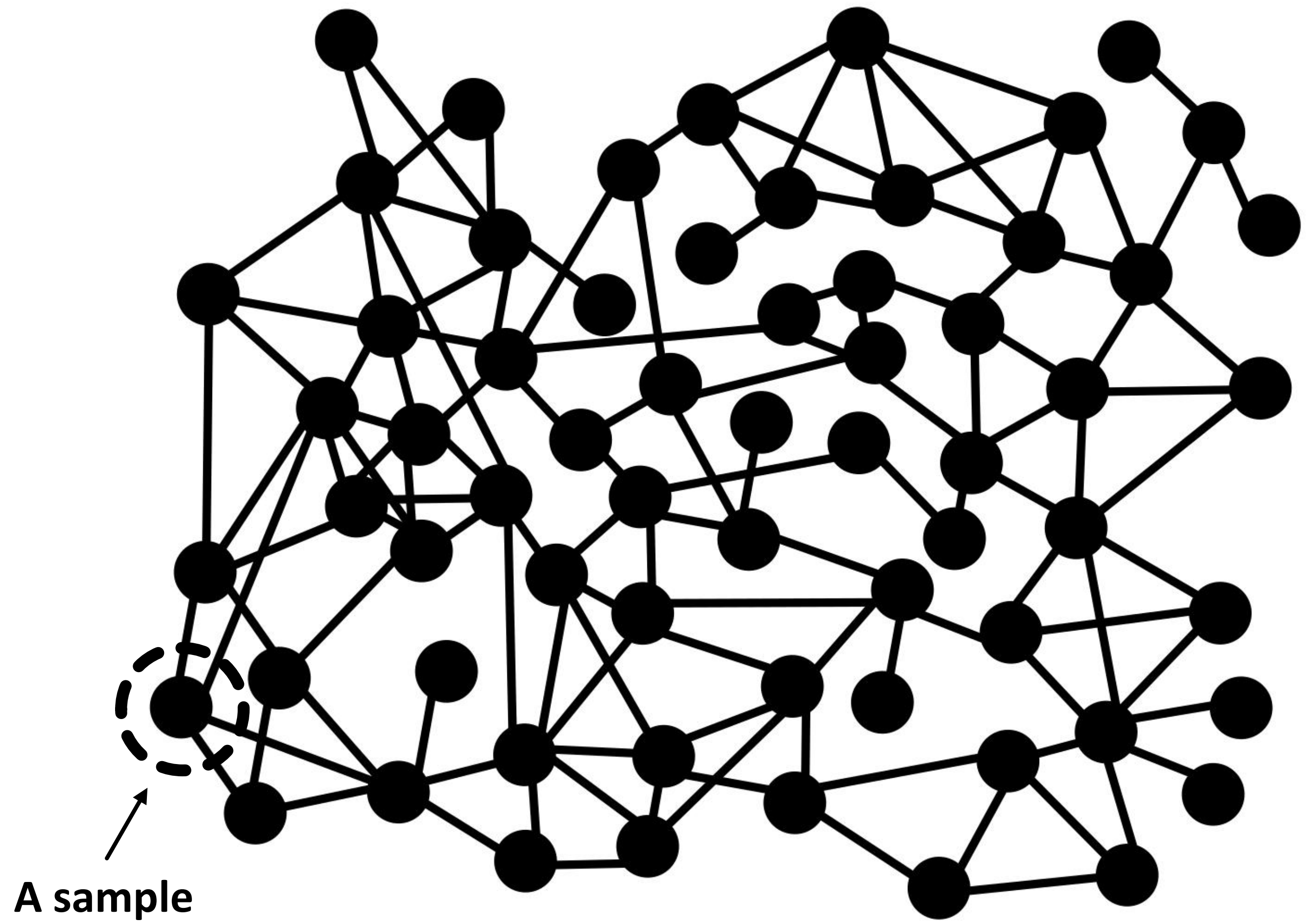
Graph [partition] parallelism

Different colors correspond to different (parallel) workers



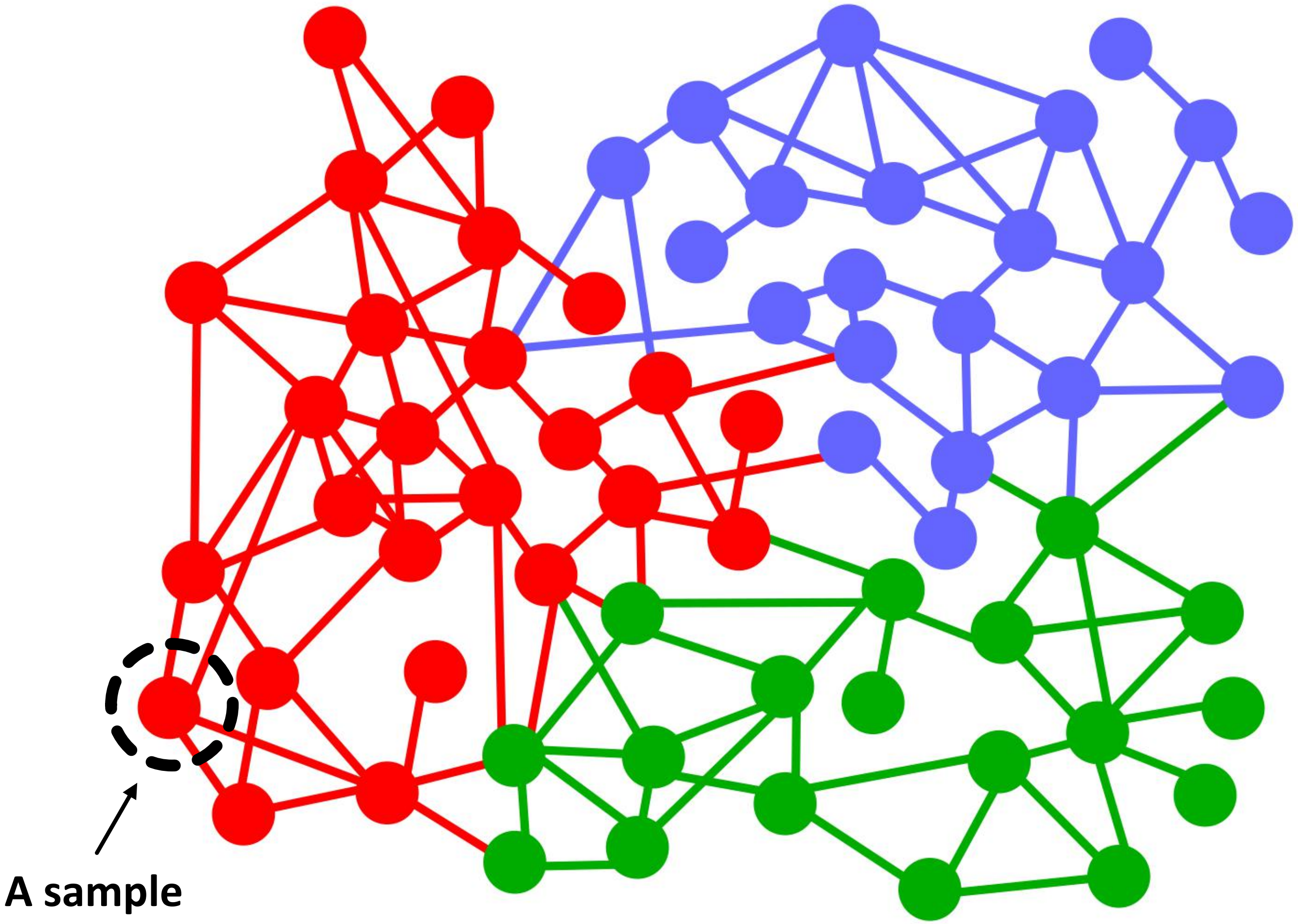
Graph [partition] parallelism

Different colors correspond to different (parallel) workers



Graph [partition] parallelism

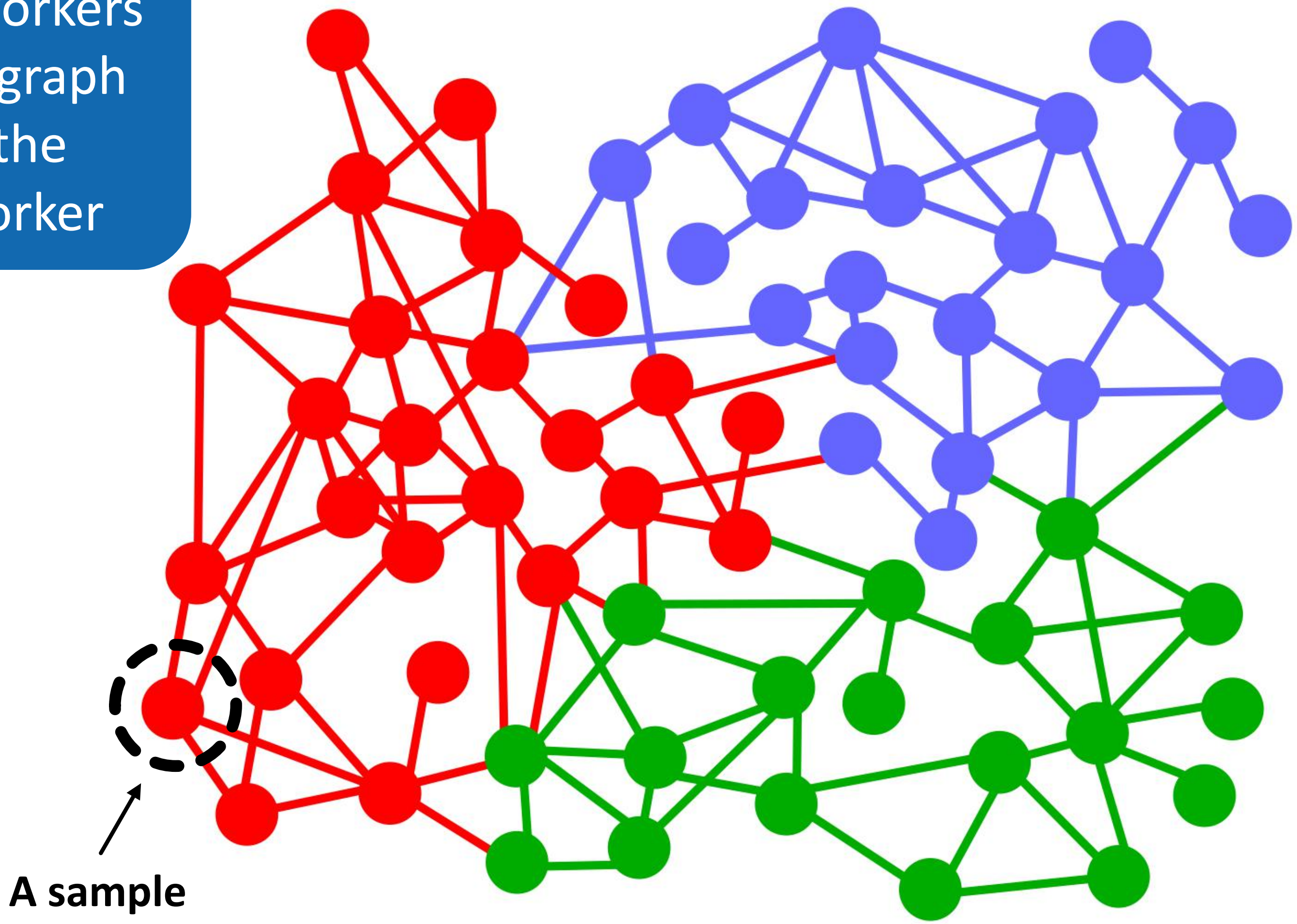
Different colors correspond to different (parallel) workers



Graph [partition] parallelism

Different colors correspond to different (parallel) workers

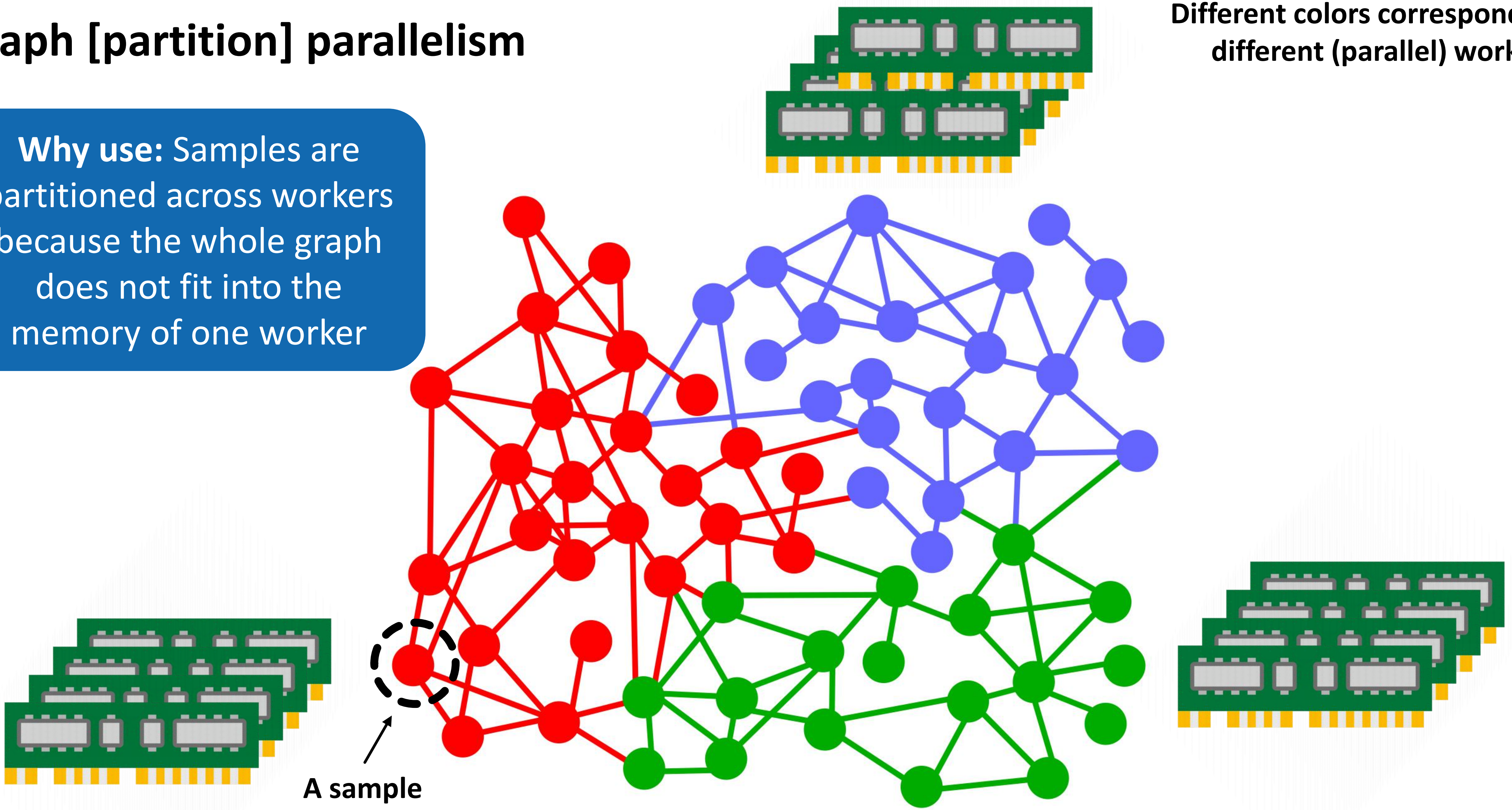
Why use: Samples are partitioned across workers because the whole graph does not fit into the memory of one worker



Graph [partition] parallelism

Why use: Samples are partitioned across workers because the whole graph does not fit into the memory of one worker

Different colors correspond to different (parallel) workers

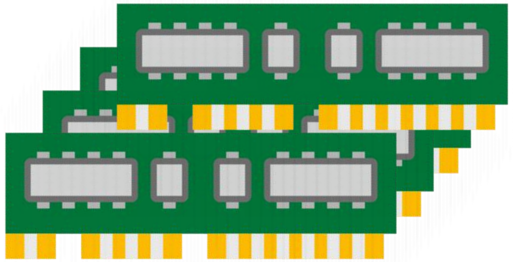
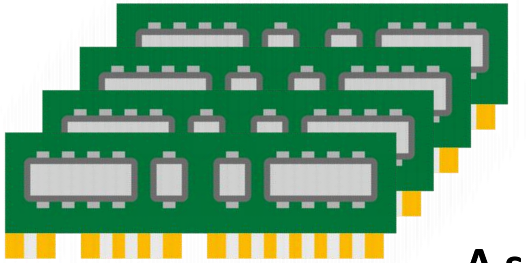
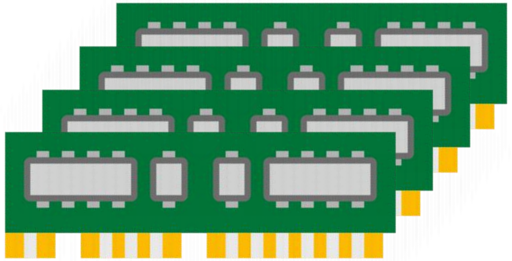
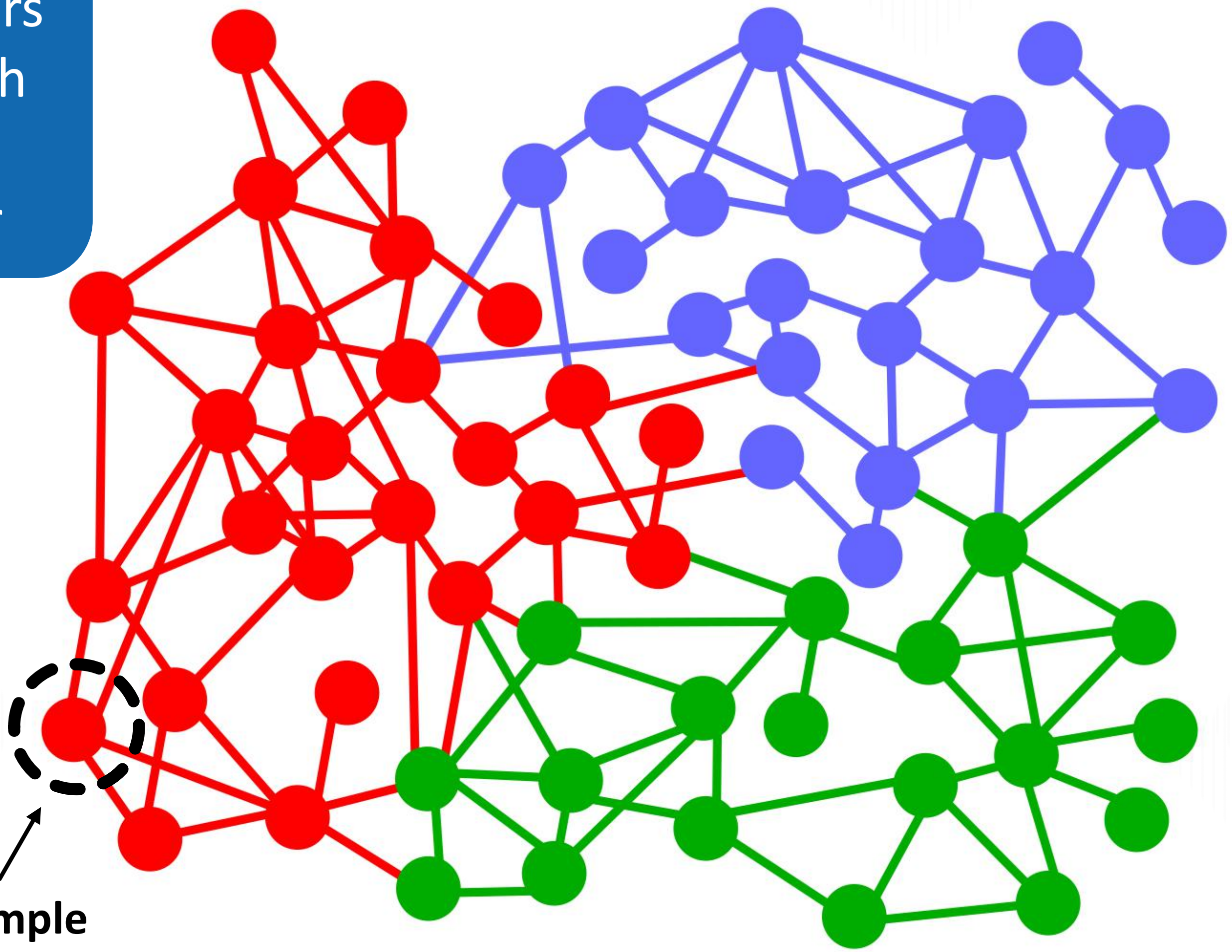


Graph [partition] parallelism

Different colors correspond to different (parallel) workers

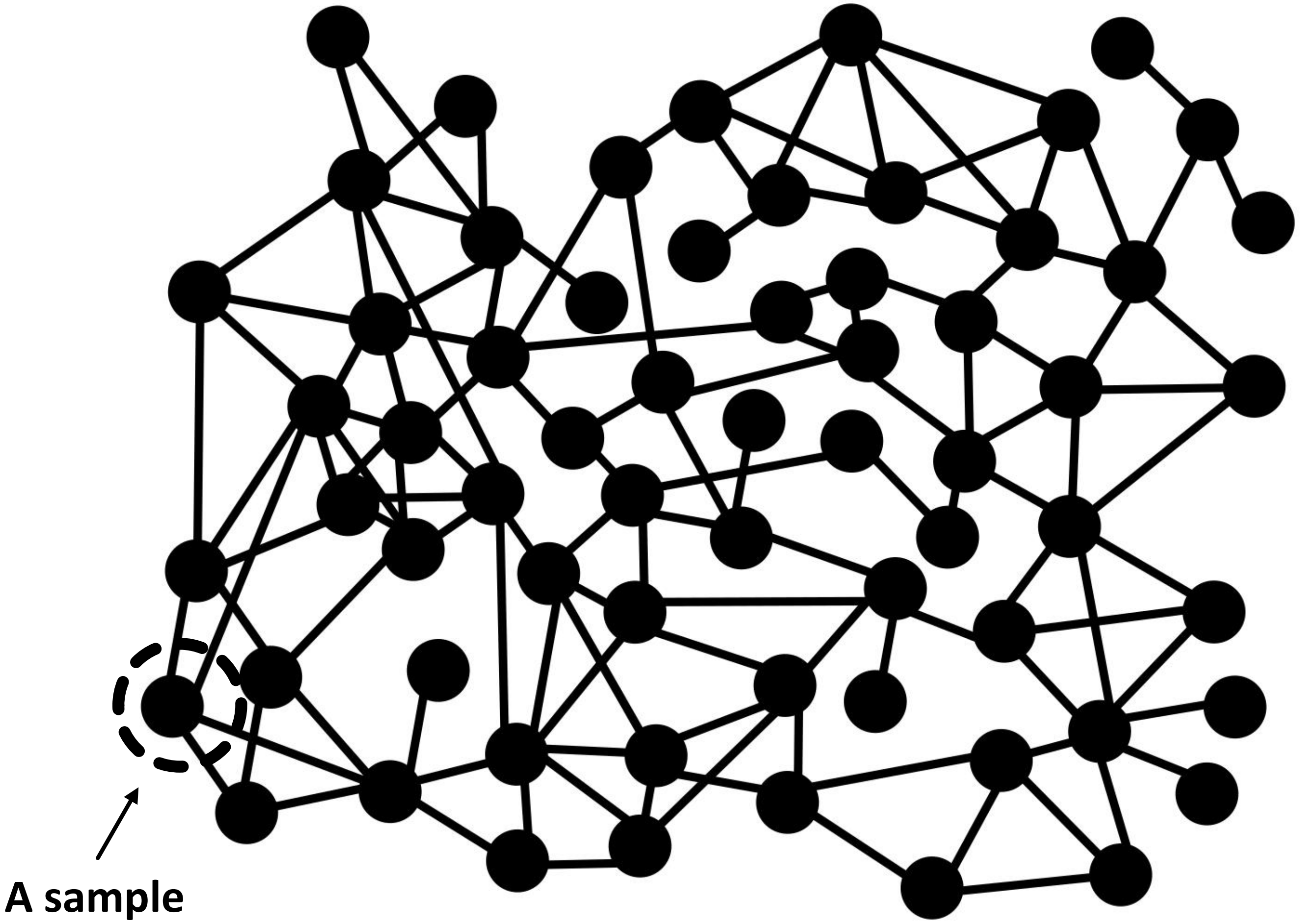
Why use: Samples are partitioned across workers because the whole graph does not fit into the memory of one worker

Each vertex and/or edge falls into some partition

Dependent mini-batch parallelism

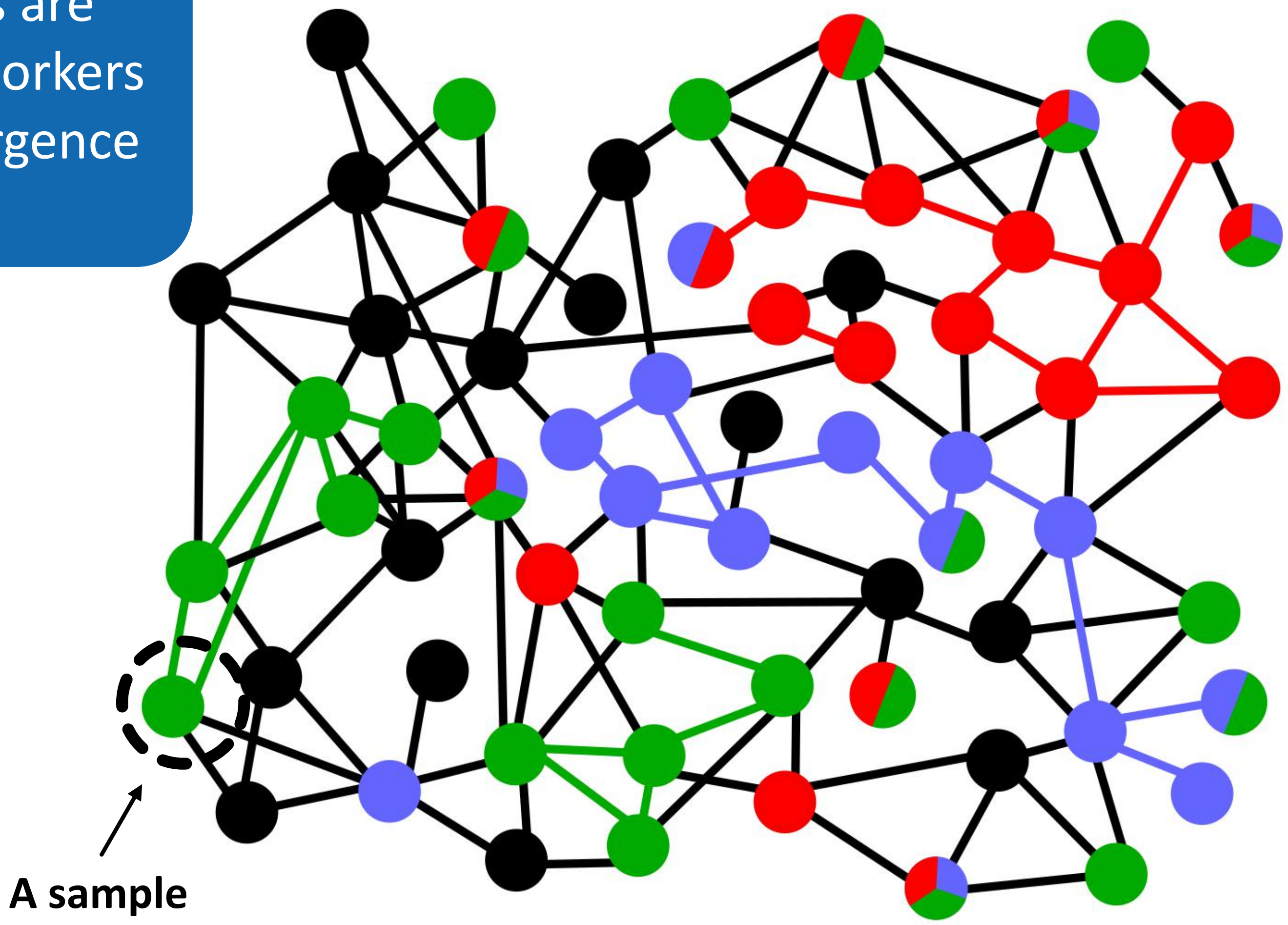
Different colors correspond to different (parallel) workers



Dependent mini-batch parallelism

Different colors correspond to different (parallel) workers

Why use: Samples are partitioned across workers to accelerate convergence

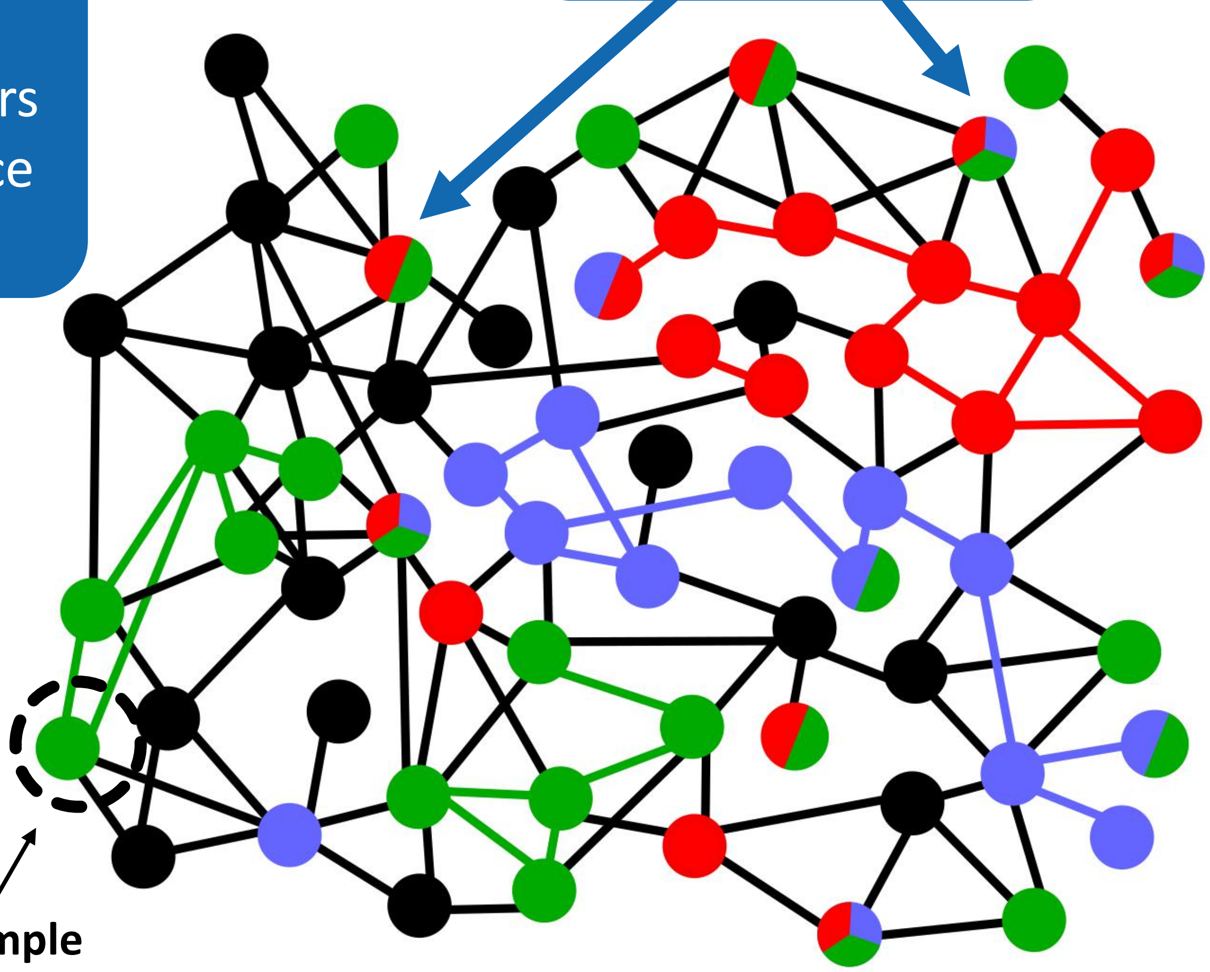


Dependent mini-batch parallelism

Why use: Samples are partitioned across workers to accelerate convergence

Samples may fall into more than one mini-batch

Different colors correspond to different (parallel) workers



A sample

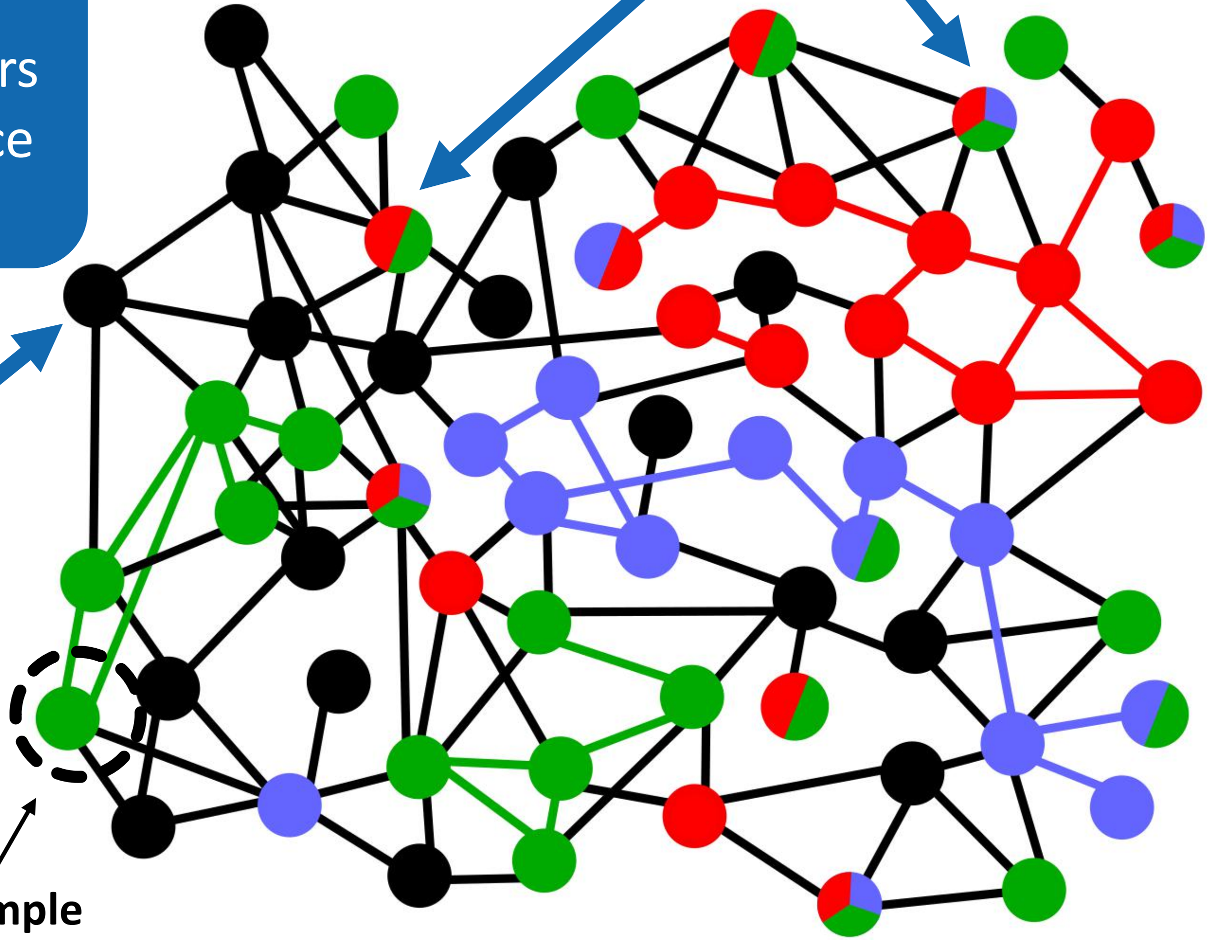
Dependent mini-batch parallelism

Different colors correspond to different (parallel) workers

Samples may fall into more than one mini-batch

Why use: Samples are partitioned across workers to accelerate convergence

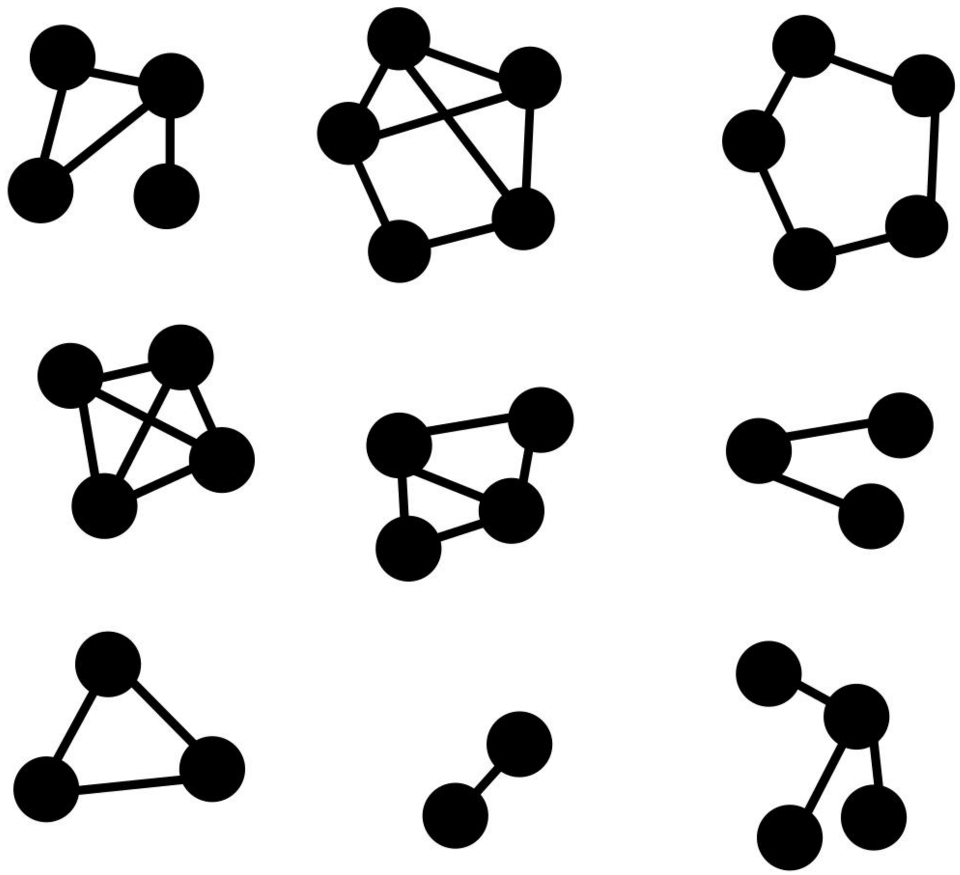
Not all samples necessarily belong to a mini-batch



A sample

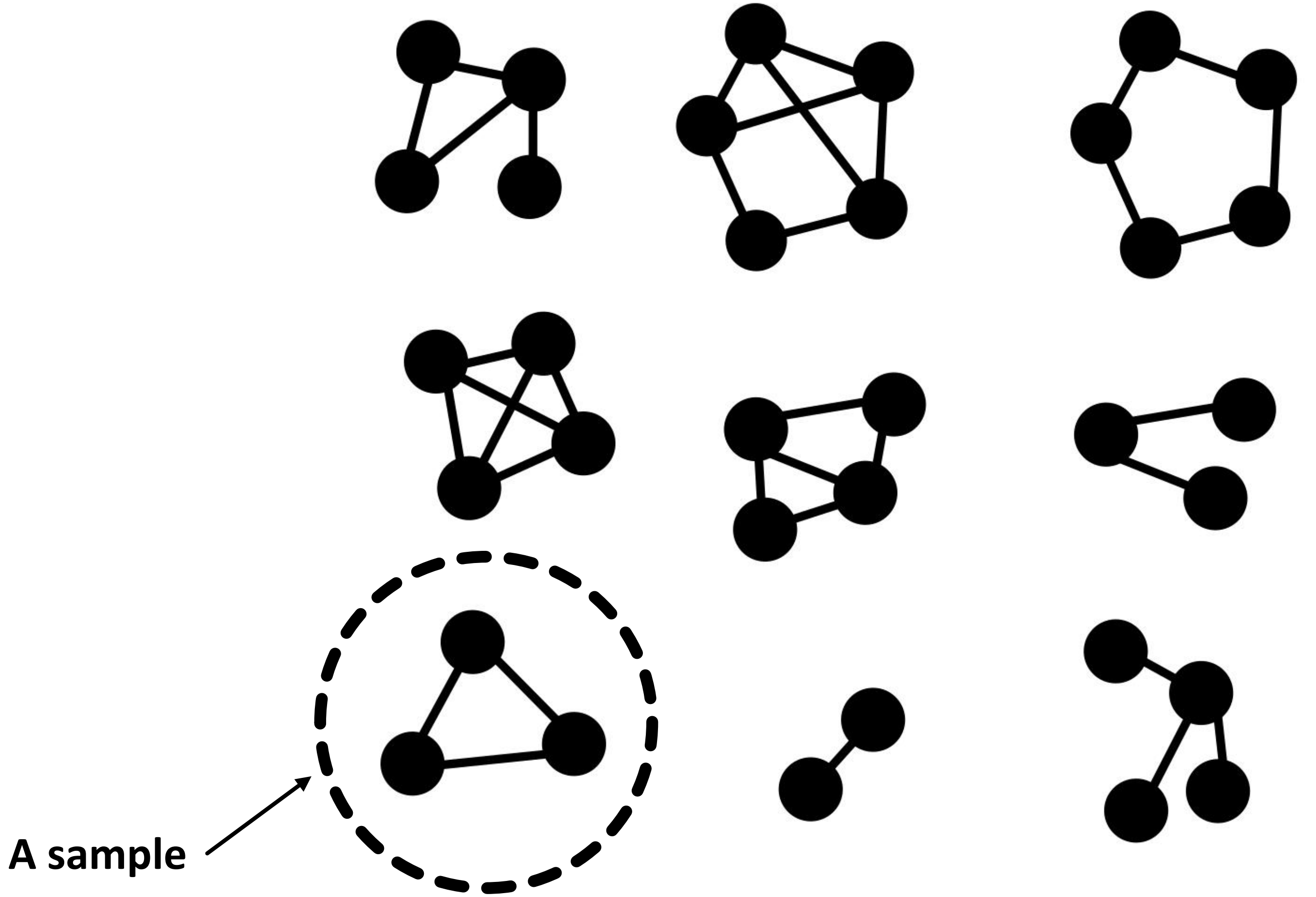
Independent mini-batch parallelism
(cf. stochastic mini-batch training)

Different colors correspond to different (parallel) workers



Independent mini-batch parallelism
(cf. stochastic mini-batch training)

Different colors correspond to different (parallel) workers

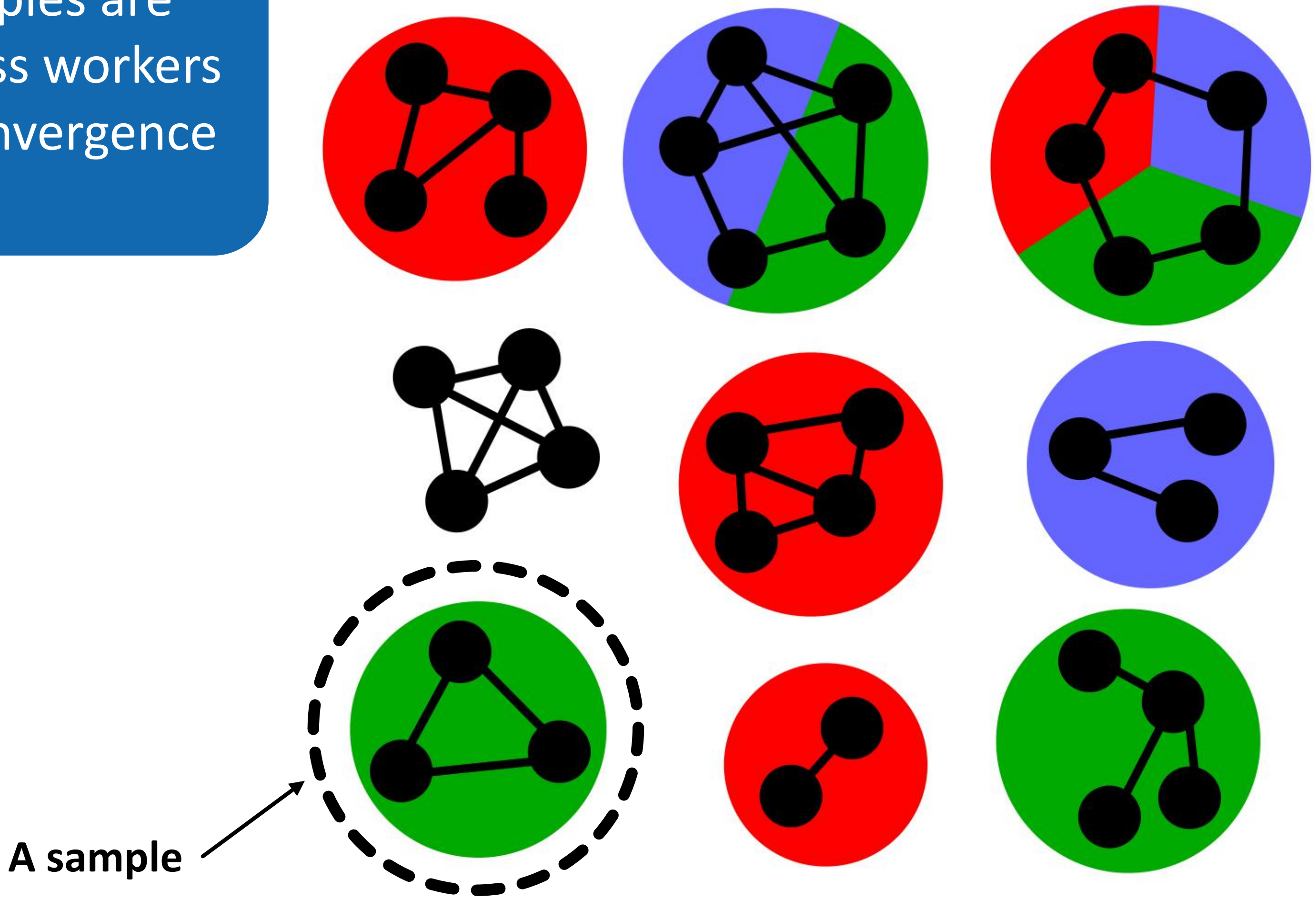


Independent mini-batch parallelism

(cf. stochastic mini-batch training)

Different colors correspond to different (parallel) workers

Why use: Samples are partitioned across workers to accelerate convergence



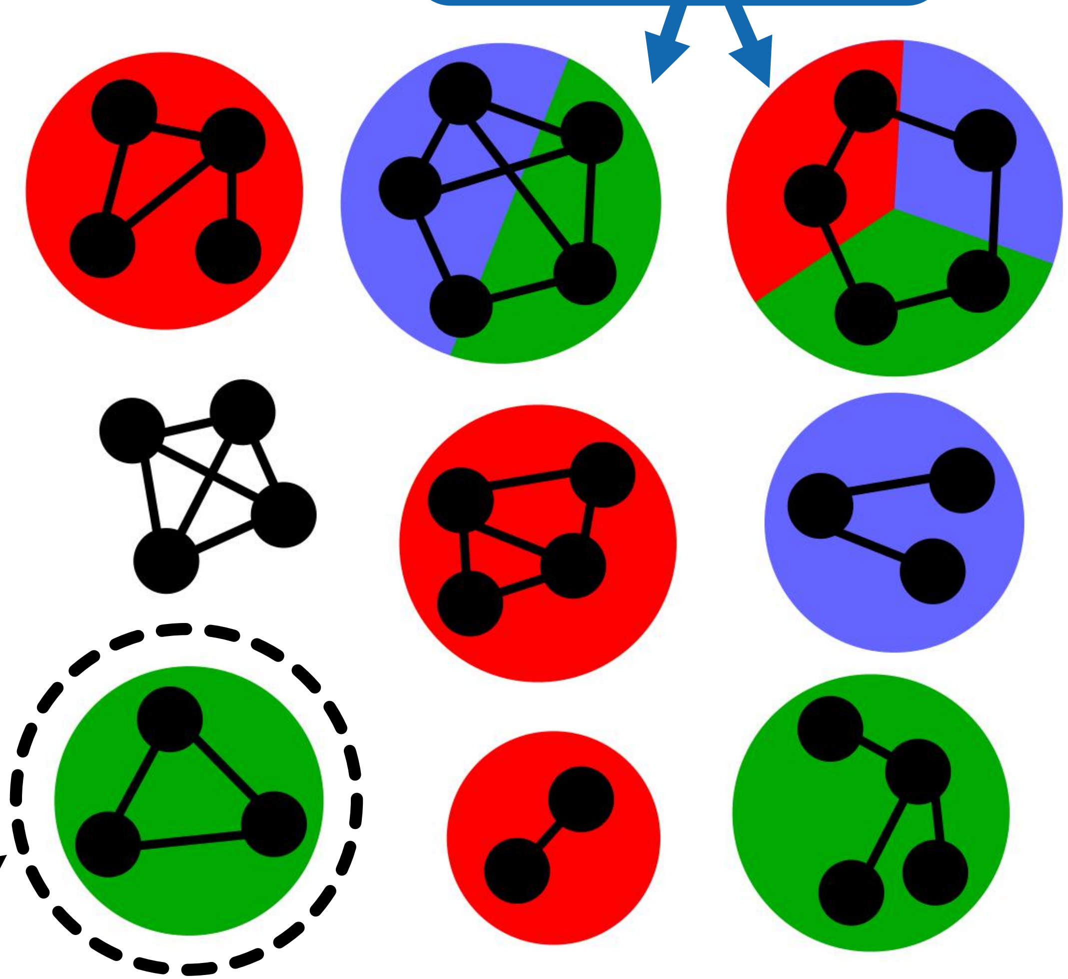
Independent mini-batch parallelism

(cf. stochastic mini-batch training)

Why use: Samples are partitioned across workers to accelerate convergence

Samples may fall into more than one mini-batch

Different colors correspond to different (parallel) workers



A sample

Independent mini-batch parallelism

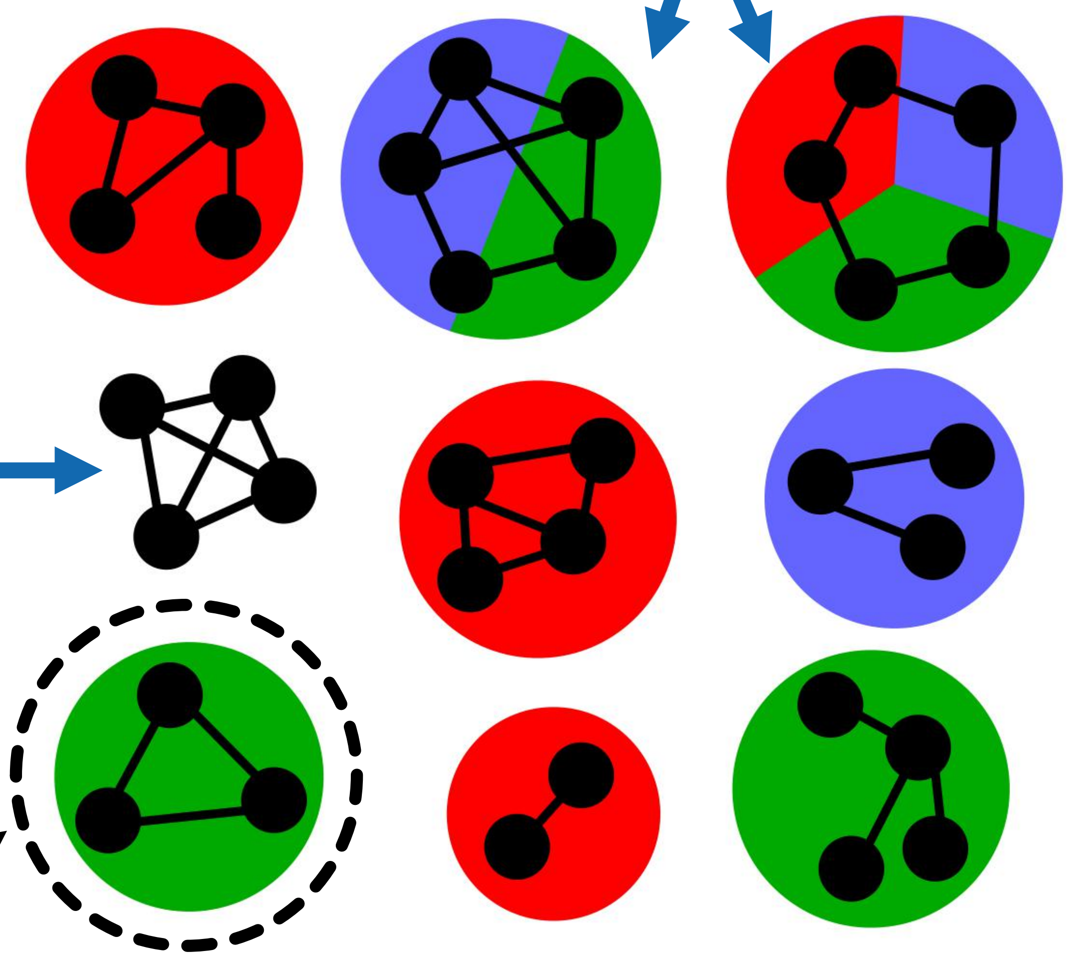
(cf. stochastic mini-batch training)

Different colors correspond to different (parallel) workers

Why use: Samples are partitioned across workers to accelerate convergence

Samples may fall into more than one mini-batch

Not all samples necessarily belong to a mini-batch

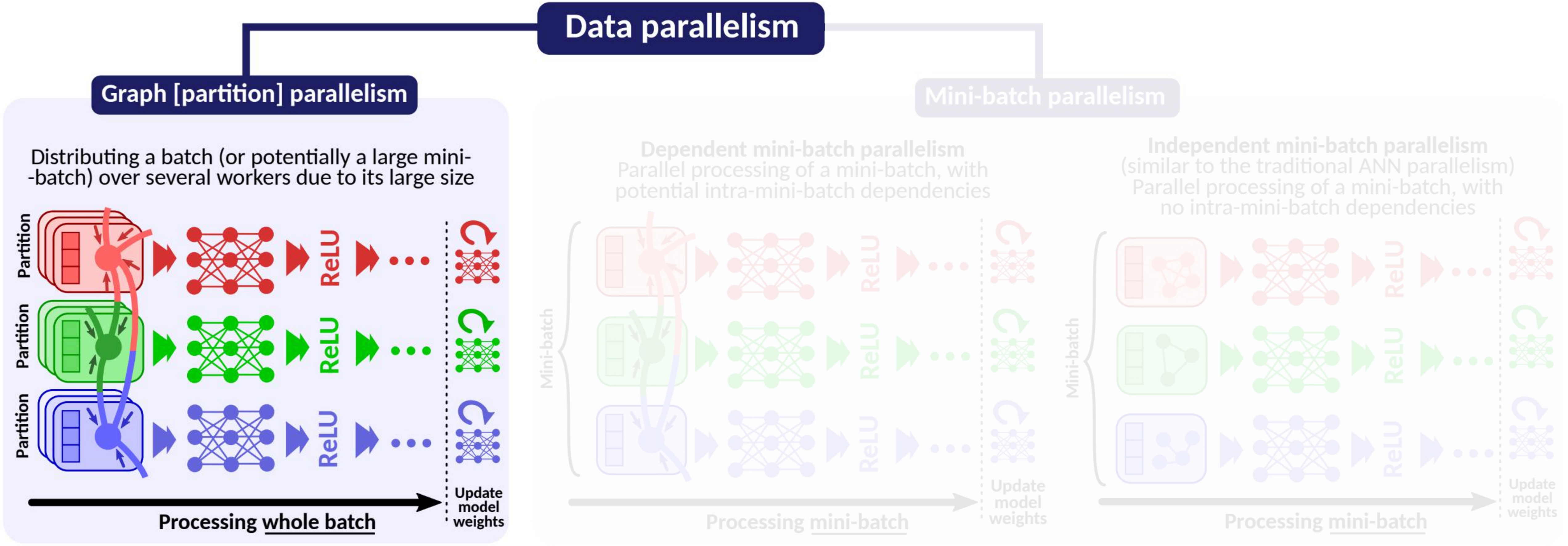


Graph partition vs. mini-batch parallelism

Different colors correspond to different (parallel) workers

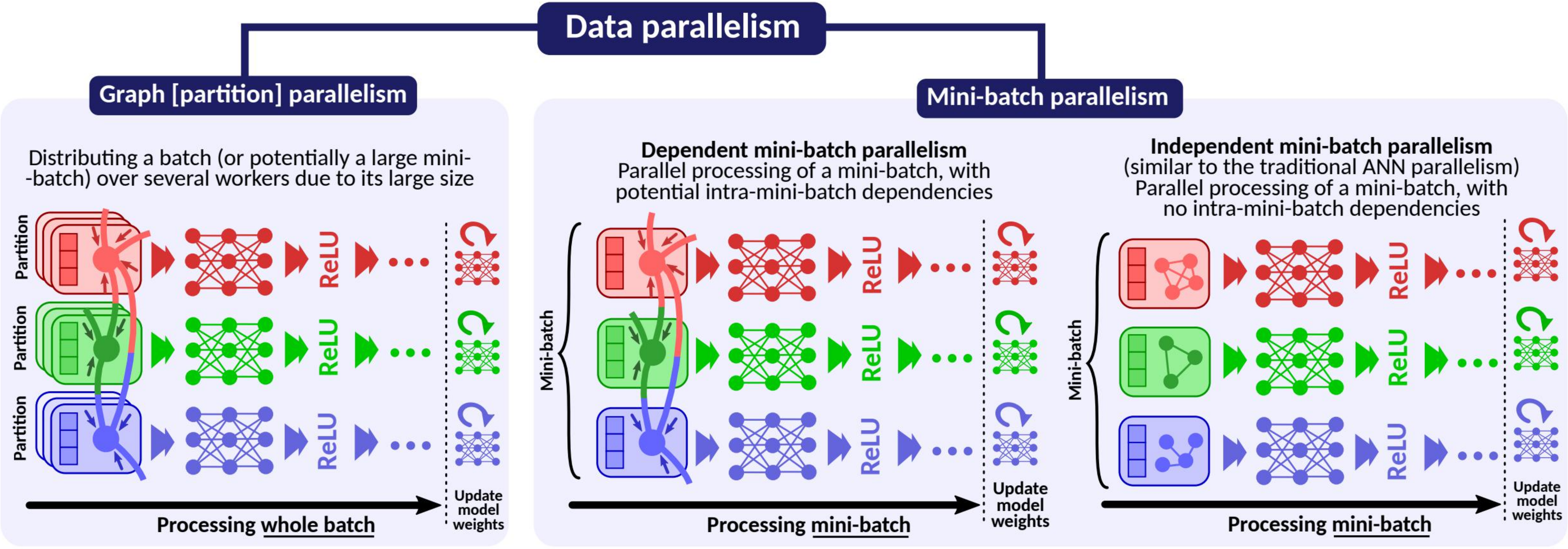
Graph partition vs. mini-batch parallelism

Different colors correspond to different (parallel) workers



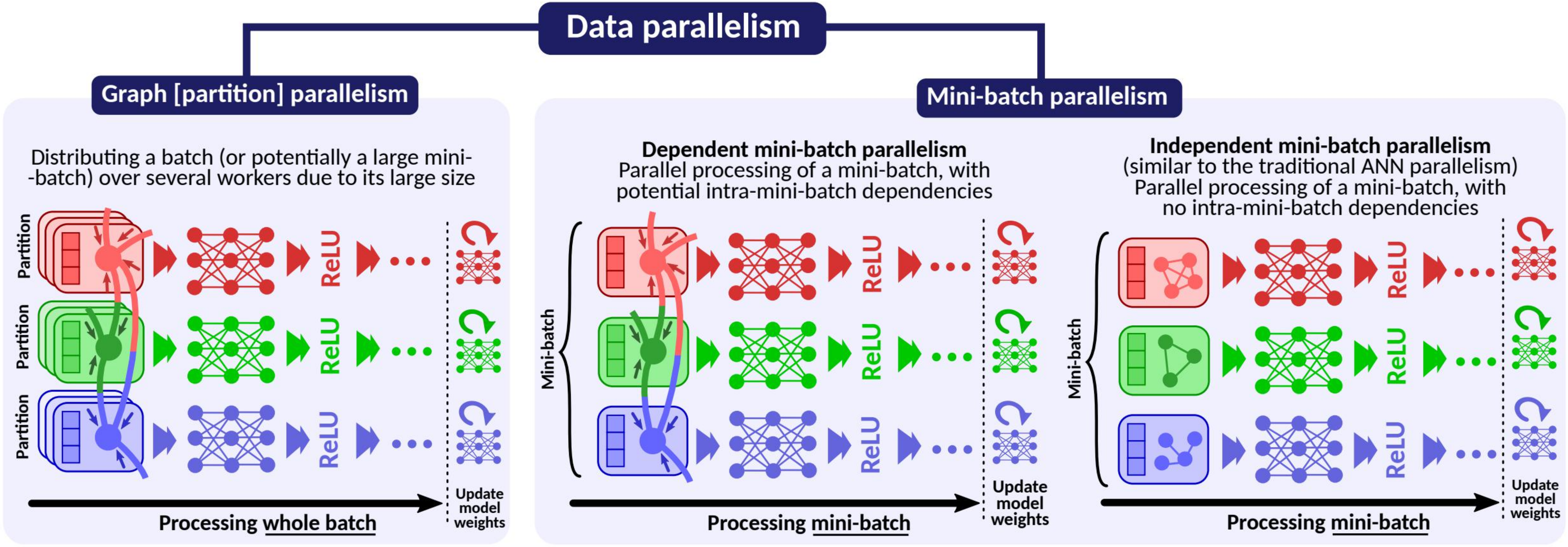
Graph partition vs. mini-batch parallelism

Different colors correspond to different (parallel) workers



Graph partition vs. mini-batch parallelism

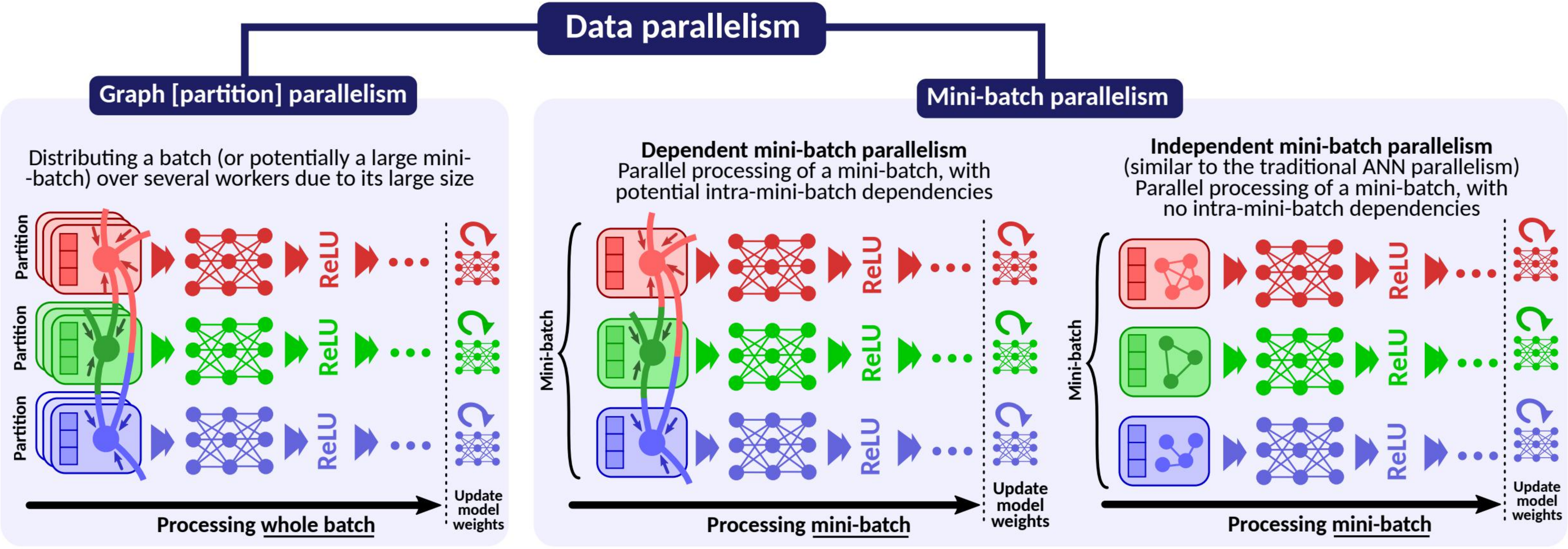
Different colors correspond to different (parallel) workers



Timing of updates to model weights

Graph partition vs. mini-batch parallelism

Different colors correspond to different (parallel) workers

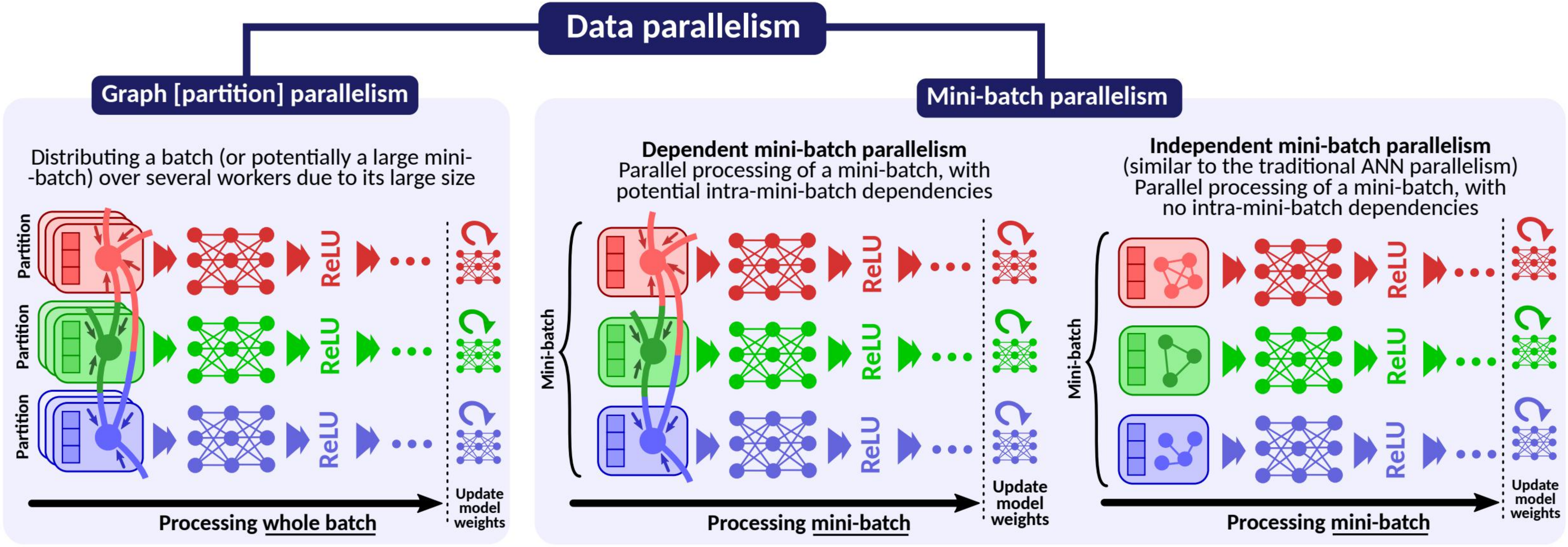


Timing of updates to model weights

Reason for being incorporated

Graph partition vs. mini-batch parallelism

Different colors correspond to different (parallel) workers



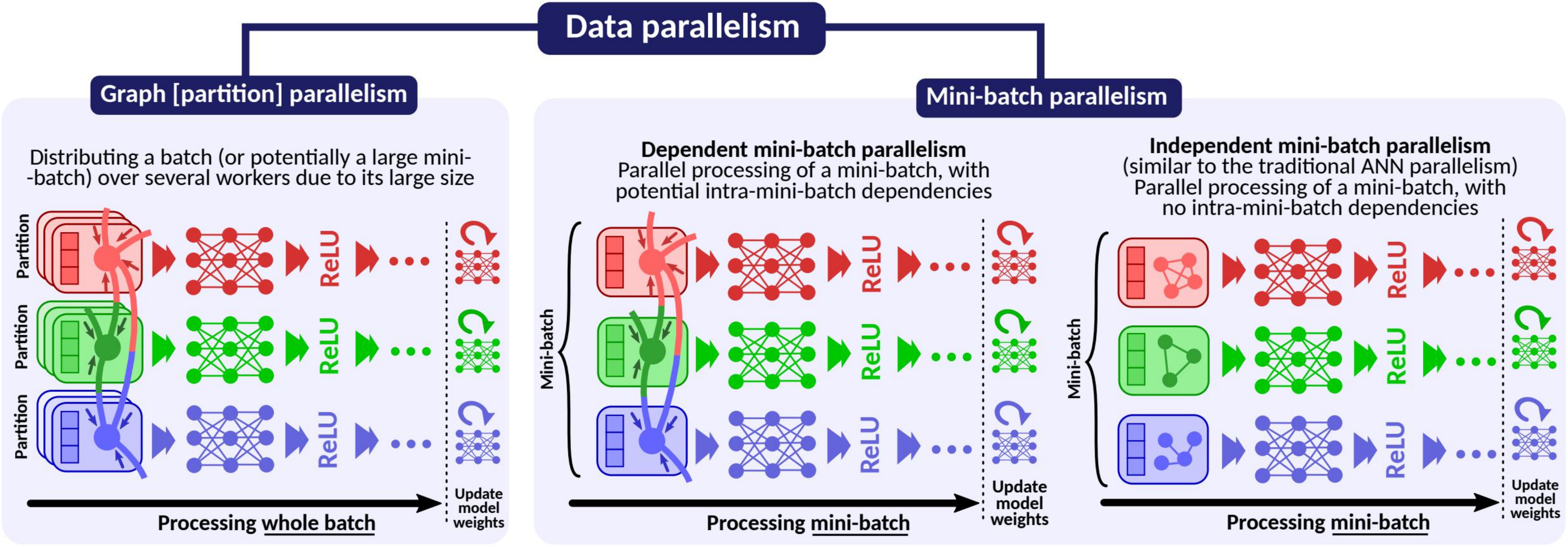
Timing of updates to model weights

Reason for being incorporated

Partitioning across workers

Graph partition vs. mini-batch parallelism

Different colors correspond to different (parallel) workers



Timing of updates to model weights

Reason for being incorporated

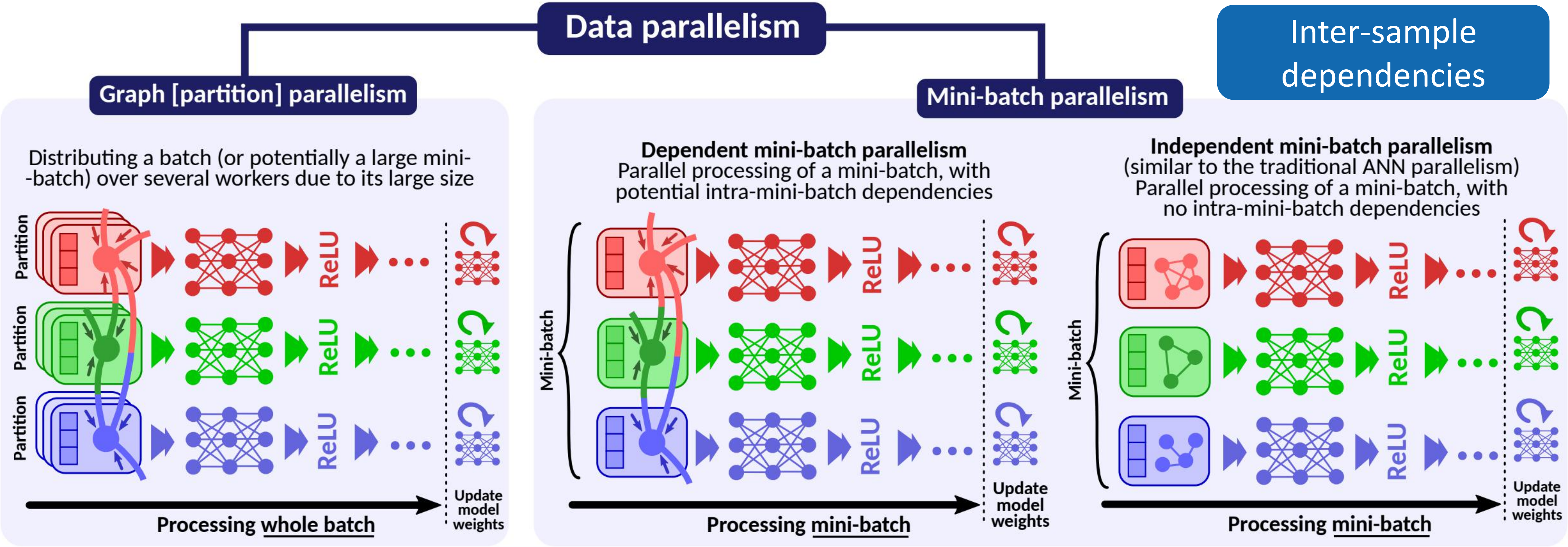
Partitioning across workers

Primary objective when partitioning

Graph partition vs. mini-batch parallelism

Different colors correspond to different (parallel) workers

Inter-sample dependencies



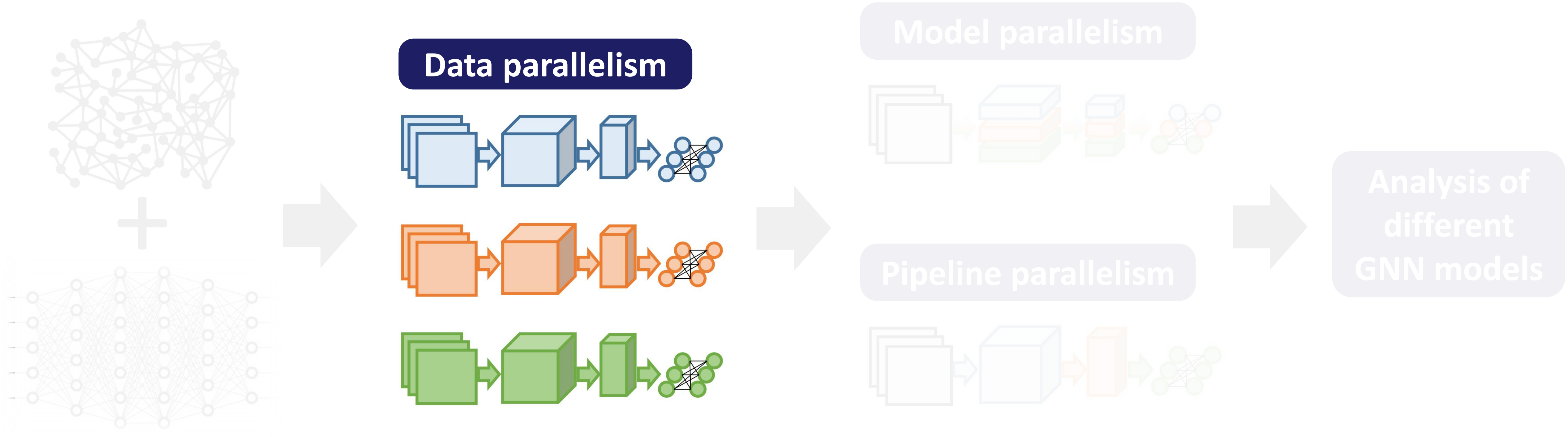
Timing of updates to model weights

Reason for being incorporated

Partitioning across workers

Primary objective when partitioning

Presentation Overview



1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

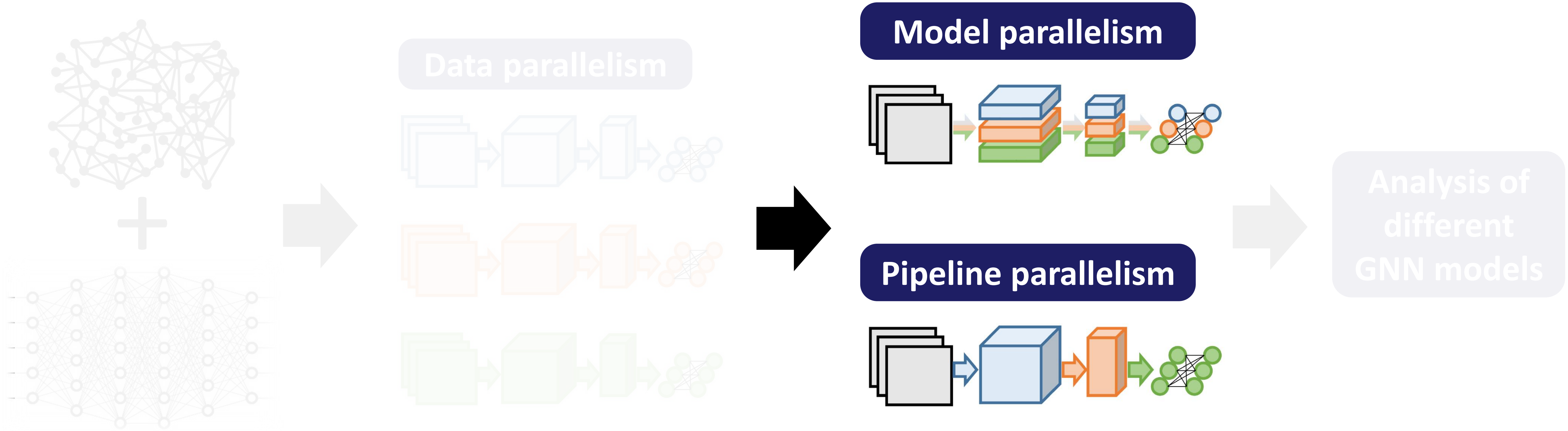
Presentation Overview

1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

Presentation Overview

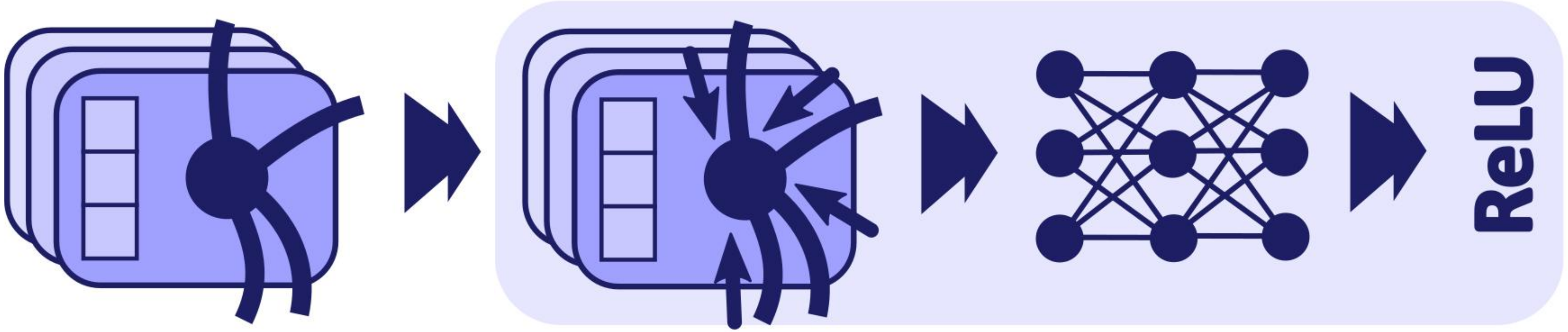


1

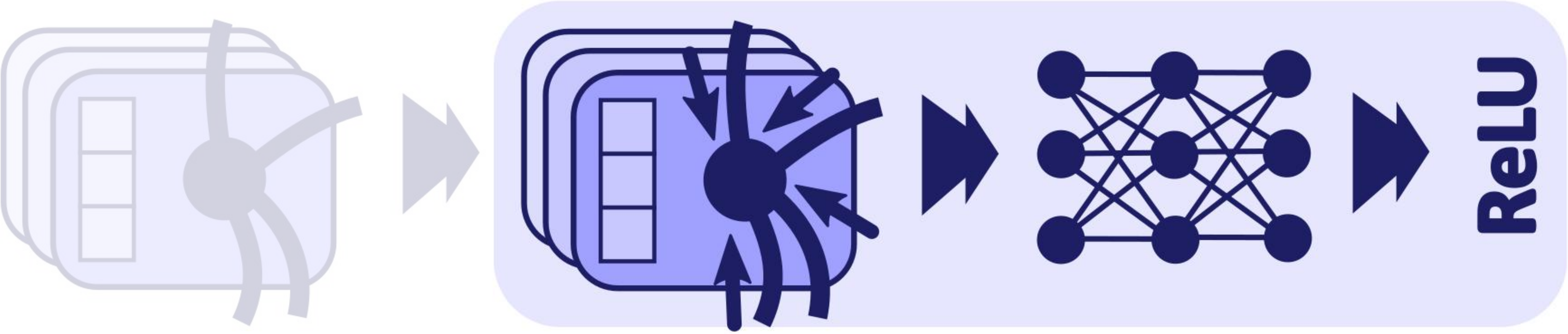
Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

- **Model parallelism**
  - **Operator parallelism**
    - **Feature parallelism**
    - **Structure parallelism**
  - **Pipeline parallelism**
    - **Macro-pipeline parallelism**
    - **Micro-pipeline parallelism**
  - **ANN parallelism**
    - **ANN-model parallelism**
    - **ANN-pipeline parallelism**

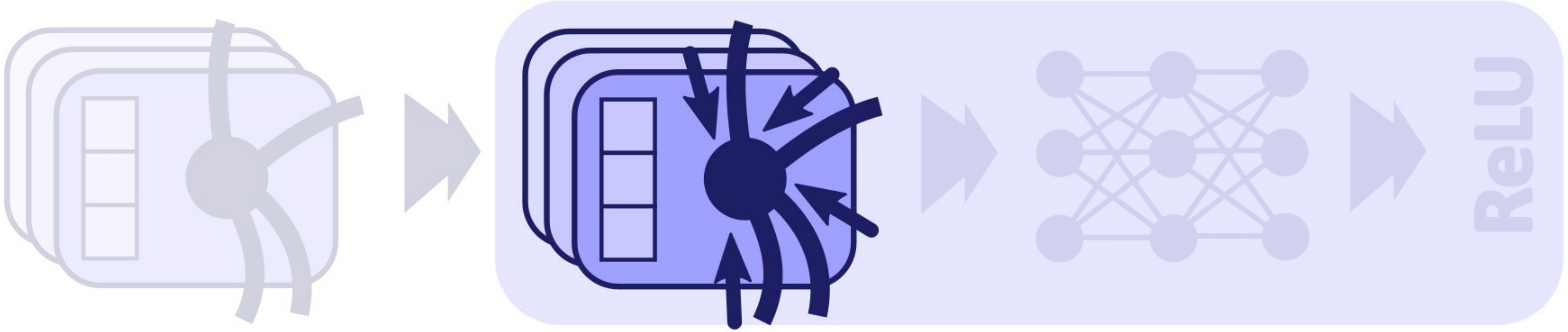


- Model parallelism
  - Operator parallelism
    - Feature parallelism
    - Structure parallelism
  - Pipeline parallelism
    - Macro-pipeline parallelism
    - Micro-pipeline parallelism
  - ANN parallelism
    - ANN-model parallelism
    - ANN-pipeline parallelism



Model parallelism

- **Operator parallelism**
  - **Feature parallelism**
  - **Structure parallelism**
- Pipeline parallelism
  - Macro-pipeline parallelism
  - Micro-pipeline parallelism
- ANN parallelism
  - ANN-model parallelism
  - ANN-pipeline parallelism



Operator parallelism

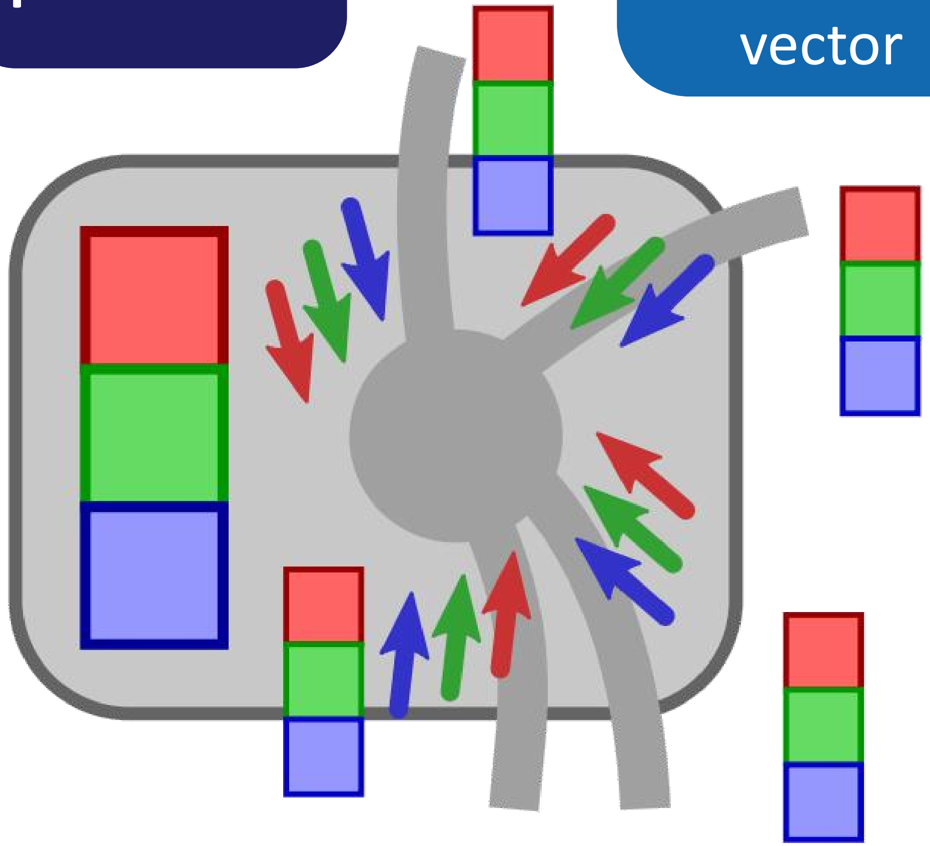
Different colors correspond to different (parallel) workers

Operator parallelism

Different colors correspond to different (parallel) workers

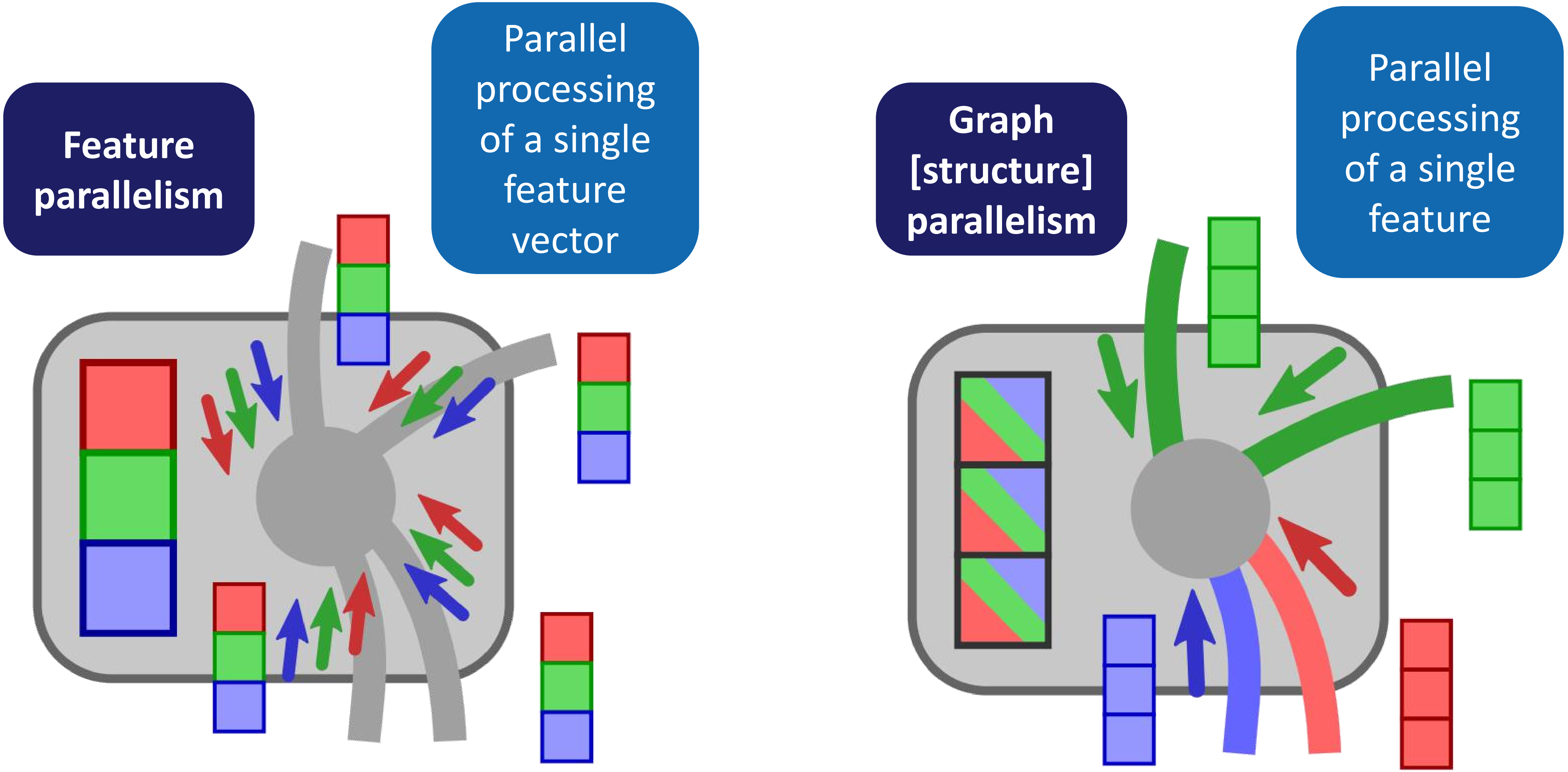
Feature parallelism

Parallel processing of a single feature vector

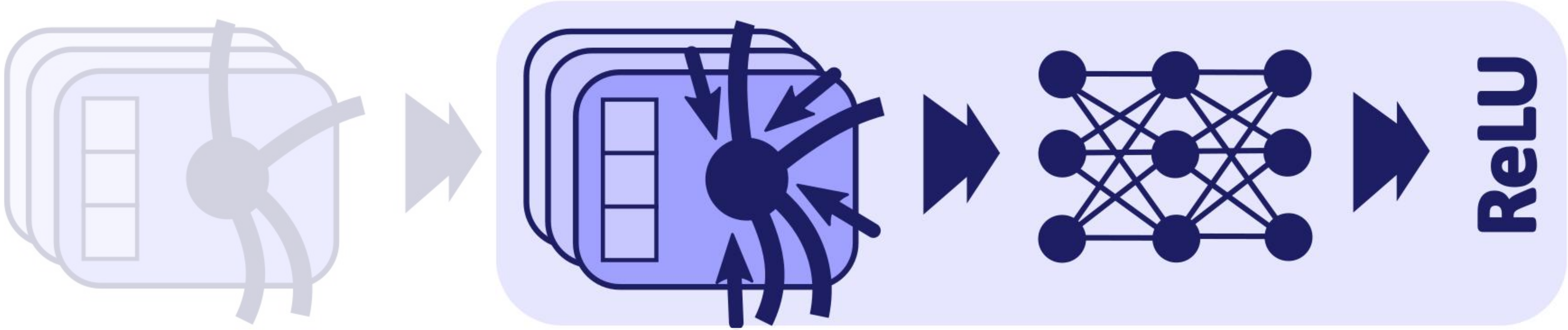


Operator parallelism

Different colors correspond to different (parallel) workers



Model parallelism
- Operator parallelism
  - Feature parallelism
  - Structure parallelism
- Pipeline parallelism
  - Macro-pipeline parallelism
  - Micro-pipeline parallelism
- ANN parallelism
  - ANN-model parallelism
  - ANN-pipeline parallelism



Model parallelism

- Operator parallelism
  - Feature parallelism
  - Structure parallelism
- Pipeline parallelism
  - Macro-pipeline parallelism
  - Micro-pipeline parallelism
- **ANN parallelism**
  - **ANN-model parallelism**
  - **ANN-pipeline parallelism**

ANN parallelism

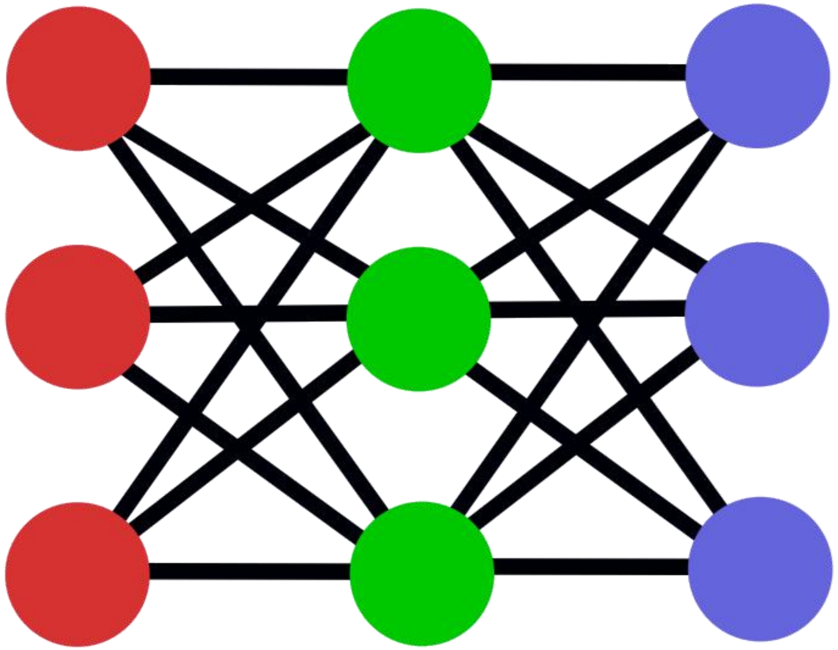
Different colors correspond to different (parallel) workers

ANN parallelism

Different colors correspond to different (parallel) workers

ANN-pipeline parallelism

Parallel processing of MLP layers in Update kernels

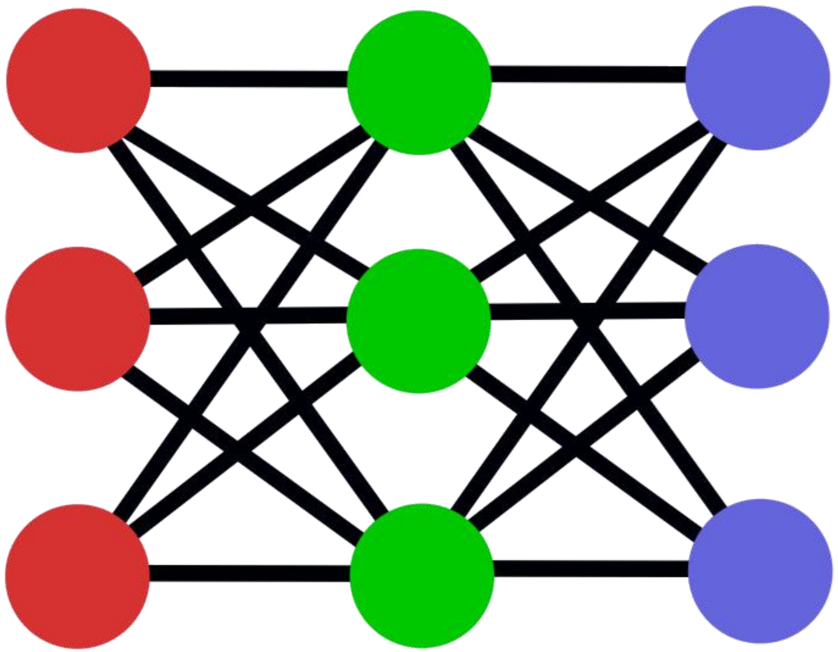


ANN parallelism

Different colors correspond to different (parallel) workers

ANN-pipeline parallelism

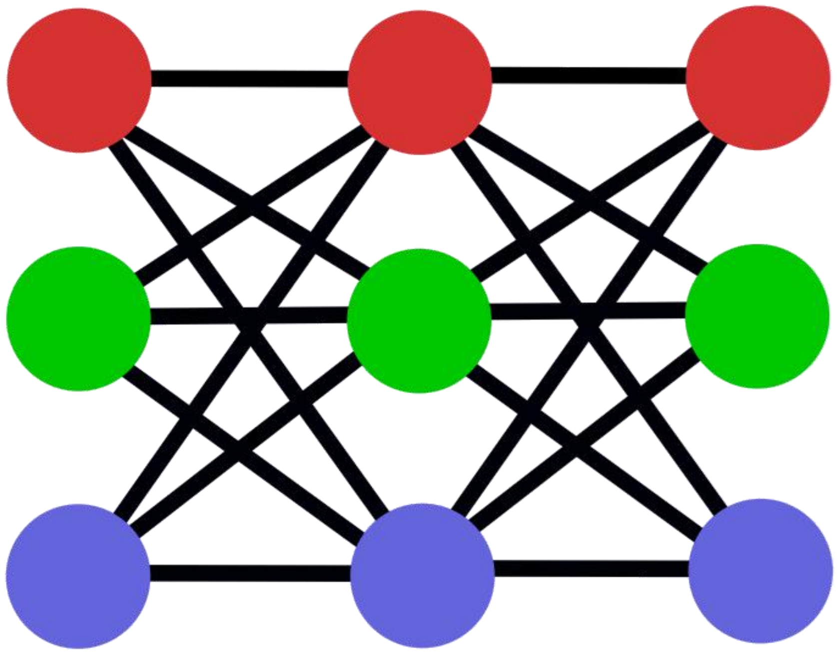
Parallel processing of MLP layers in Update kernels



ANN-model parallelism

ANN-model parallelism

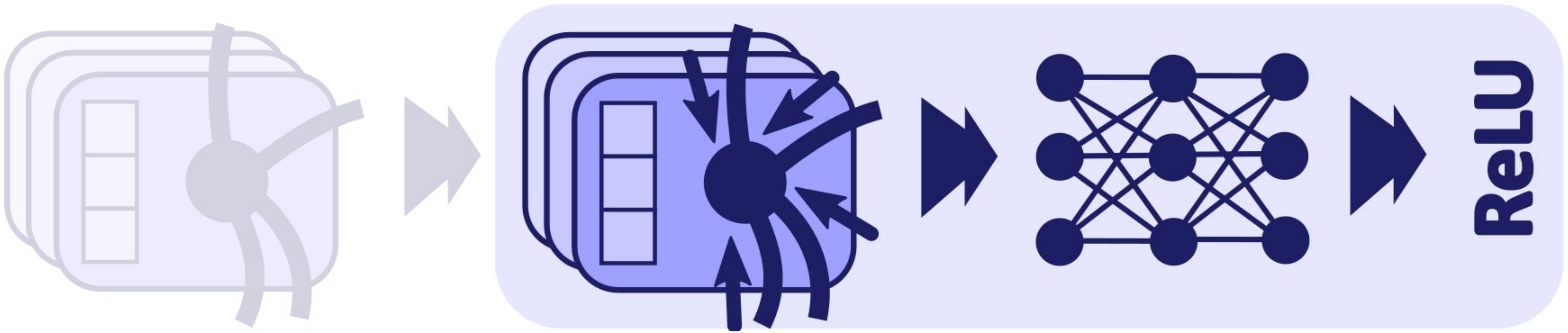
Parallel processing of the MLP model in Update kernels



Model parallelism

- Operator parallelism
  - Feature parallelism
  - Structure parallelism
- Pipeline parallelism
  - Macro-pipeline parallelism
  - Micro-pipeline parallelism
- **ANN parallelism**
  - **ANN-model parallelism**
  - **ANN-pipeline parallelism**

Model parallelism
- Operator parallelism
  - Feature parallelism
  - Structure parallelism
- **Pipeline parallelism**
  - Macro-pipeline parallelism
  - **Micro-pipeline parallelism**
- ANN parallelism
  - ANN-model parallelism
  - ANN-pipeline parallelism

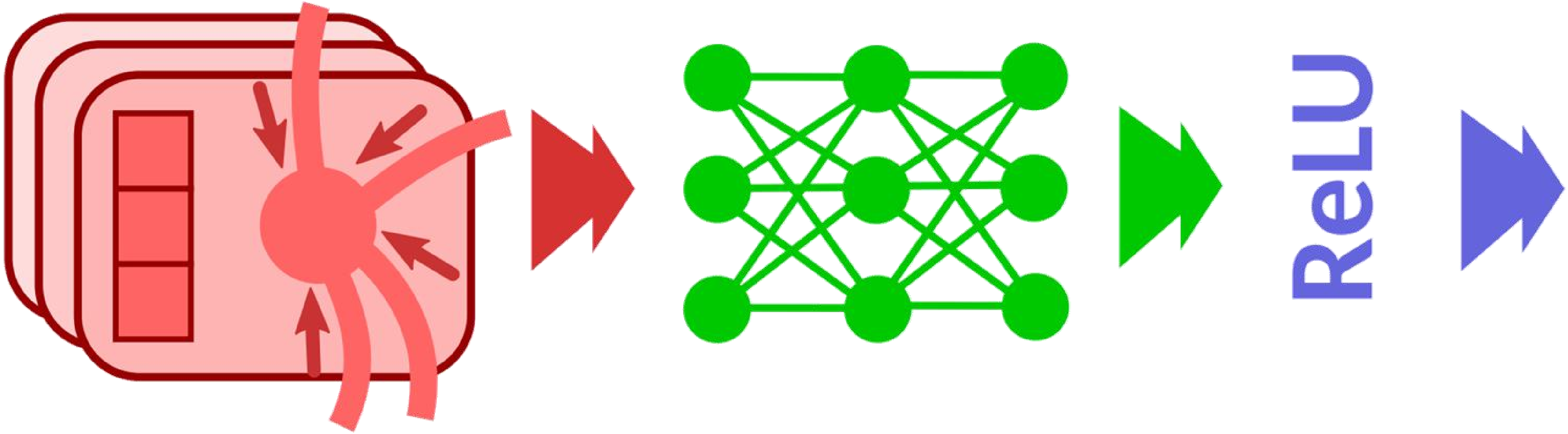


Pipeline parallelism

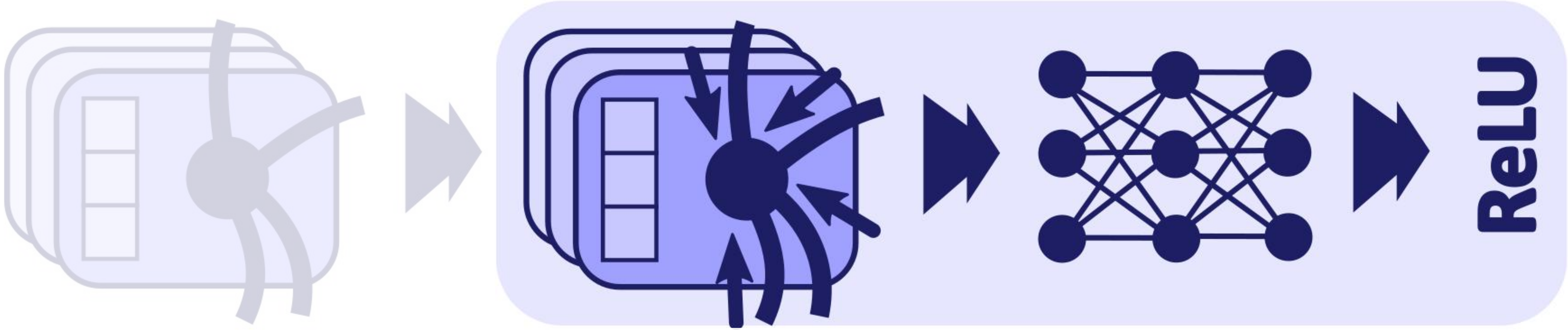
Different colors correspond to different (parallel) workers

Micro-pipeline parallelism

Parallel processing of different stages within a single GNN layer

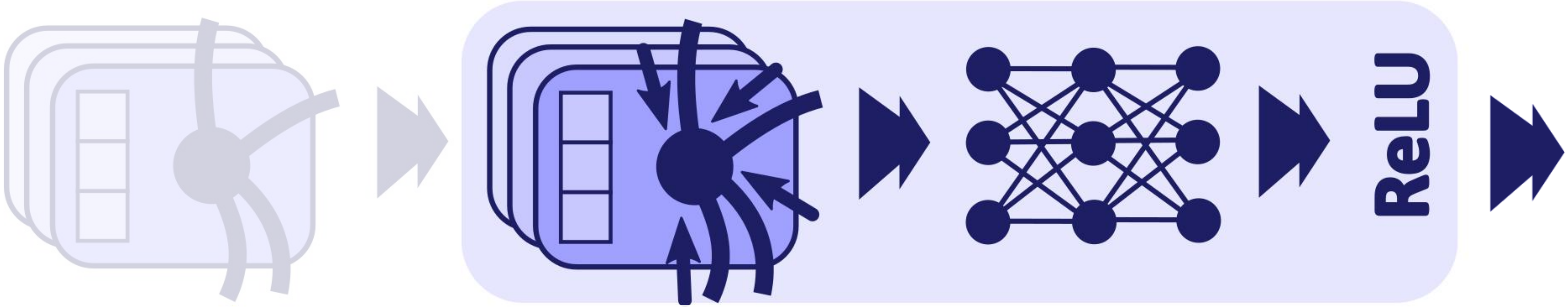


Model parallelism
- Operator parallelism
  - Feature parallelism
  - Structure parallelism
- Pipeline parallelism
  - Macro-pipeline parallelism
  - **Micro-pipeline parallelism**
- ANN parallelism
  - ANN-model parallelism
  - ANN-pipeline parallelism



Model parallelism

- Operator parallelism
  - Feature parallelism
  - Structure parallelism
- Pipeline parallelism
  - **Macro-pipeline parallelism**
  - Micro-pipeline parallelism
- ANN parallelism
  - ANN-model parallelism
  - ANN-pipeline parallelism

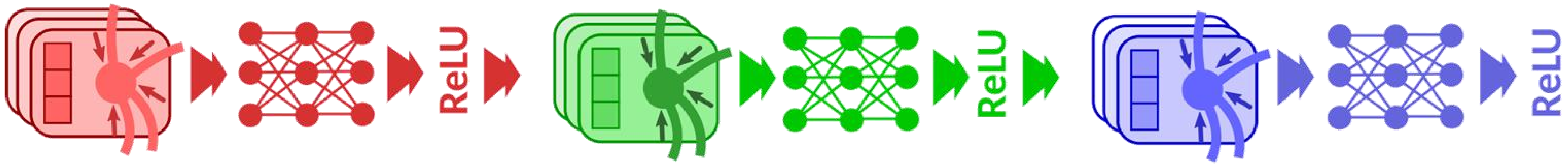


Pipeline parallelism

Different colors correspond to different (parallel) workers

Macro-pipeline parallelism

Parallel processing of different GNN layers

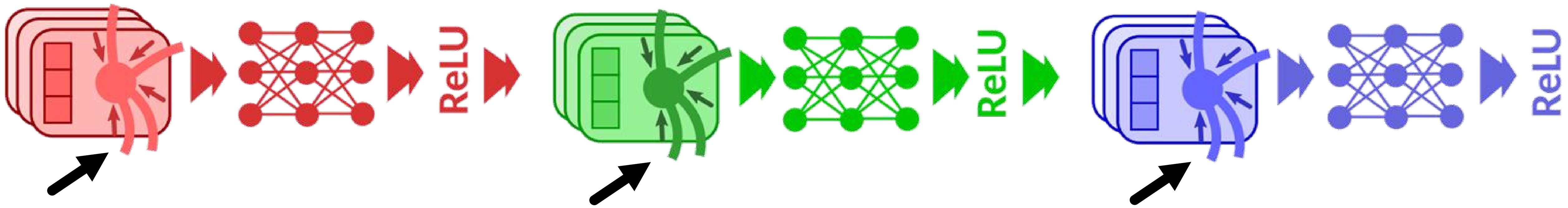


Pipeline parallelism

Different colors correspond to different (parallel) workers

Macro-pipeline parallelism

Parallel processing of different GNN layers



The data (i.e., the graph structure) is needed at every GNN layer - unlike in traditional ANNs, where data is only needed at the pipeline beginning

Forms of Parallelism in Traditional DL vs. GNNs

Different colors correspond to different (parallel) workers

Forms of Parallelism in Traditional DL vs. GNNs

Different colors correspond to different (parallel) workers

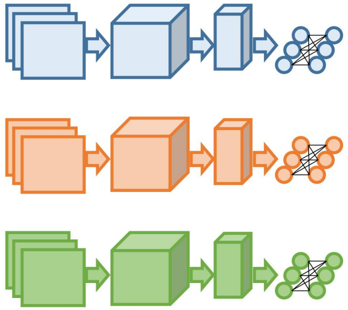
Parallelism in GNNs is **more complex** than in Traditional DL (dependencies!)

Forms of Parallelism in Traditional DL vs. GNNs

Different colors correspond to different (parallel) workers

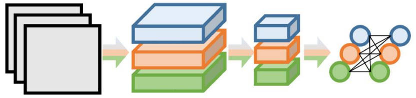
Parallelism in GNNs is **more complex** than in Traditional DL (dependencies!)

Data parallelism

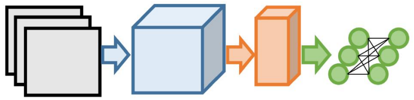


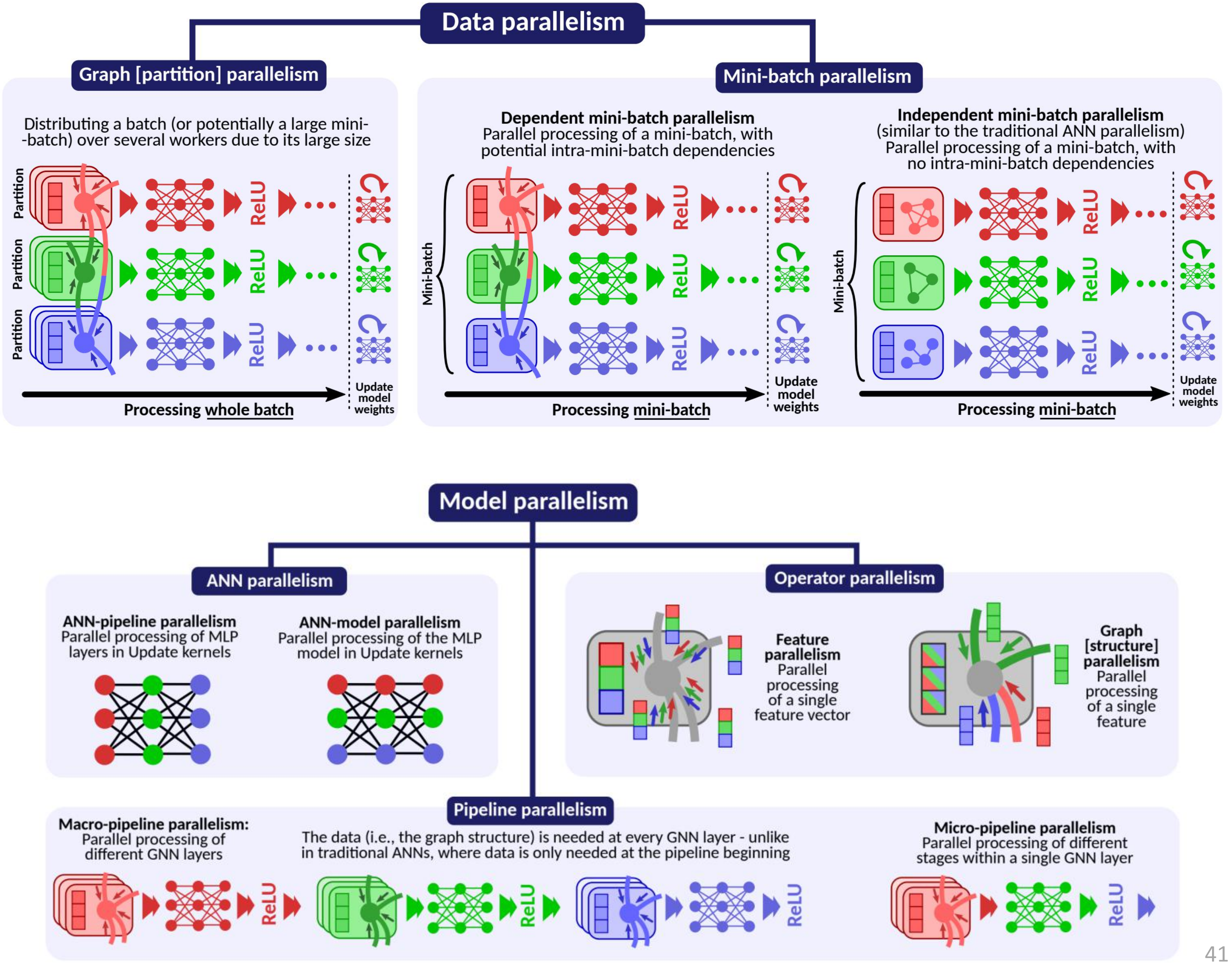
vs.

Model parallelism



Pipeline parallelism





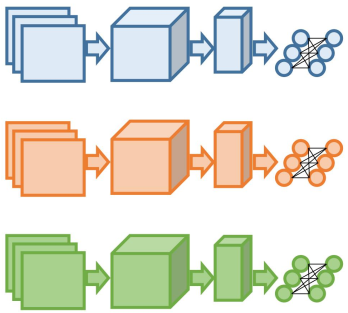
Forms of Parallelism in Traditional DL vs. GNNs

Different colors correspond to different (parallel) workers

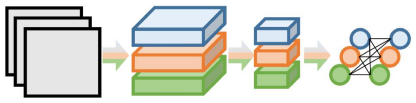
Parallelism in GNNs is **more complex** than in Traditional DL (dependencies!)

New forms of parallelism in GNNs (again dependencies!)

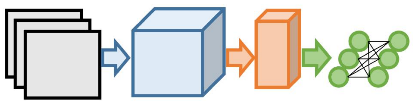
Data parallelism



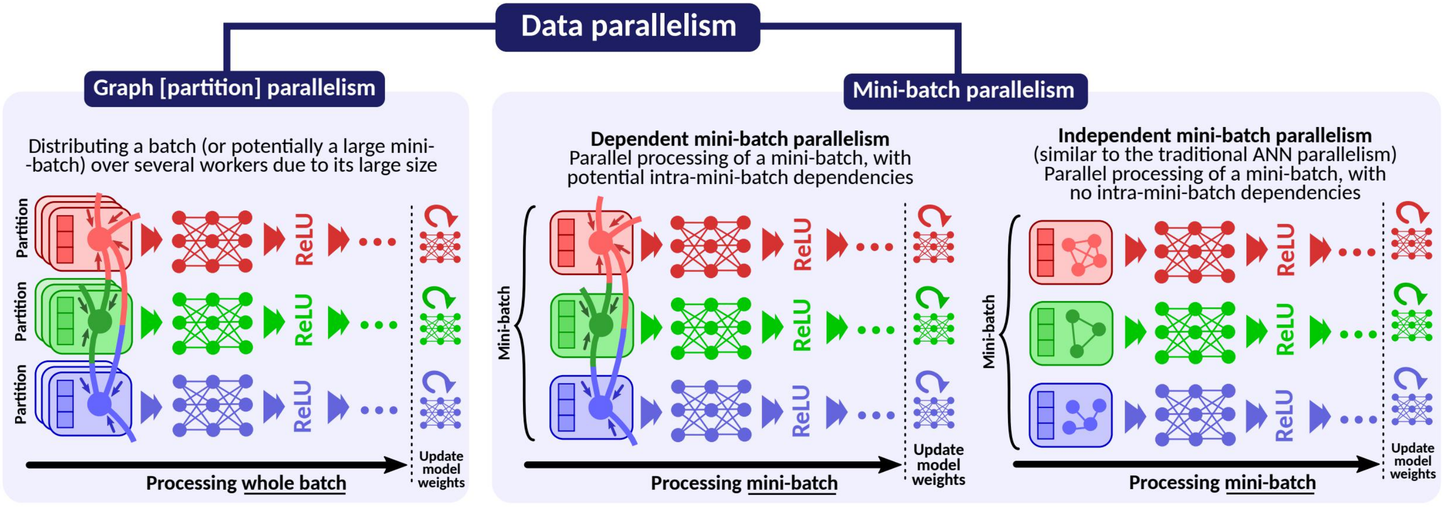
Model parallelism

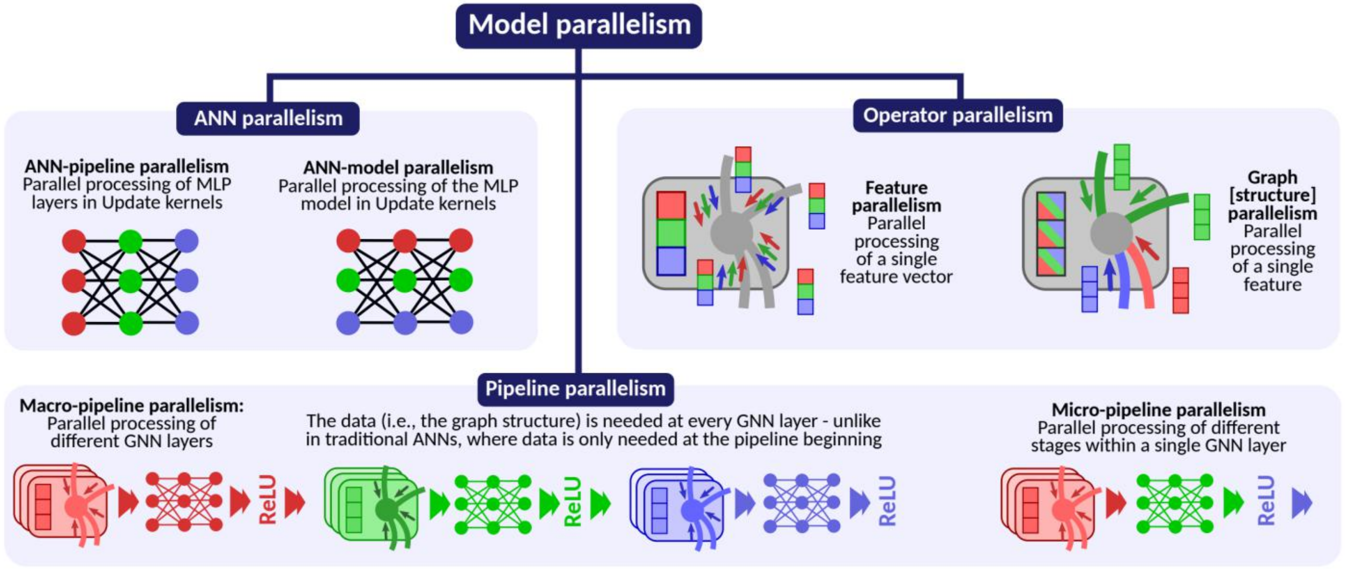


Pipeline parallelism



vs.





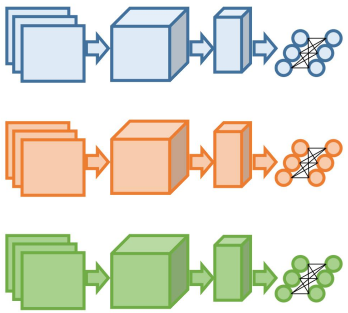
Forms of Parallelism in Traditional DL vs. GNNs

Different colors correspond to different (parallel) workers

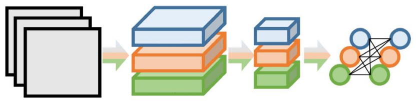
Parallelism in GNNs is **more complex** than in Traditional DL (dependencies!)

New forms of parallelism in GNNs (again dependencies!)

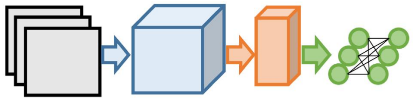
Data parallelism



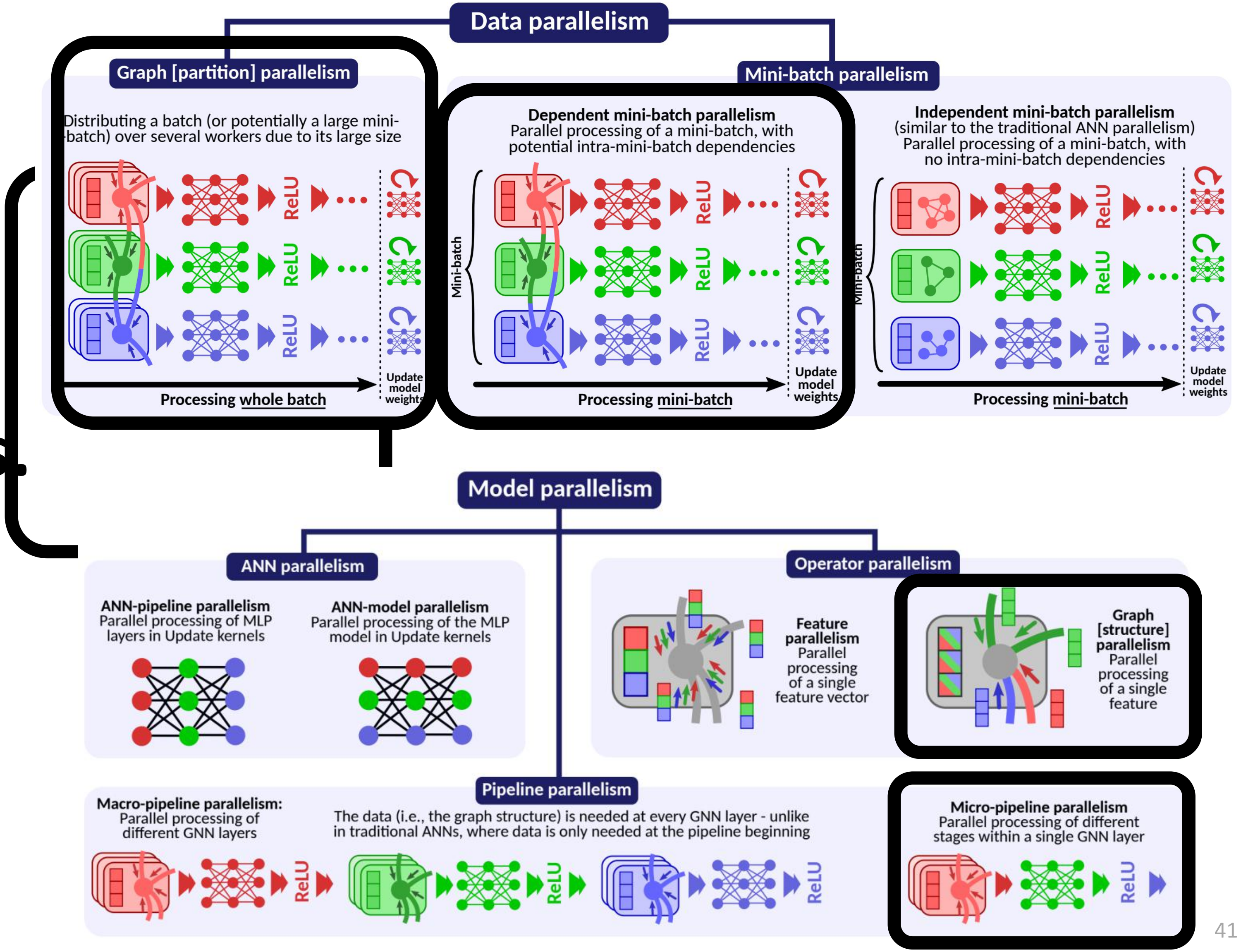
Model parallelism



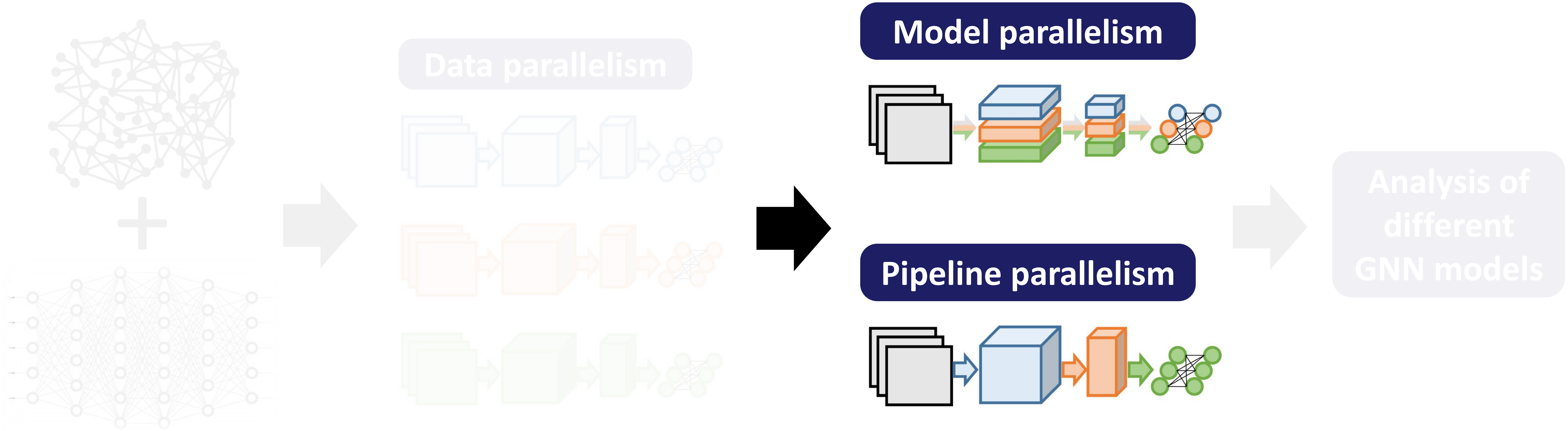
Pipeline parallelism



vs.



Presentation Overview



1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

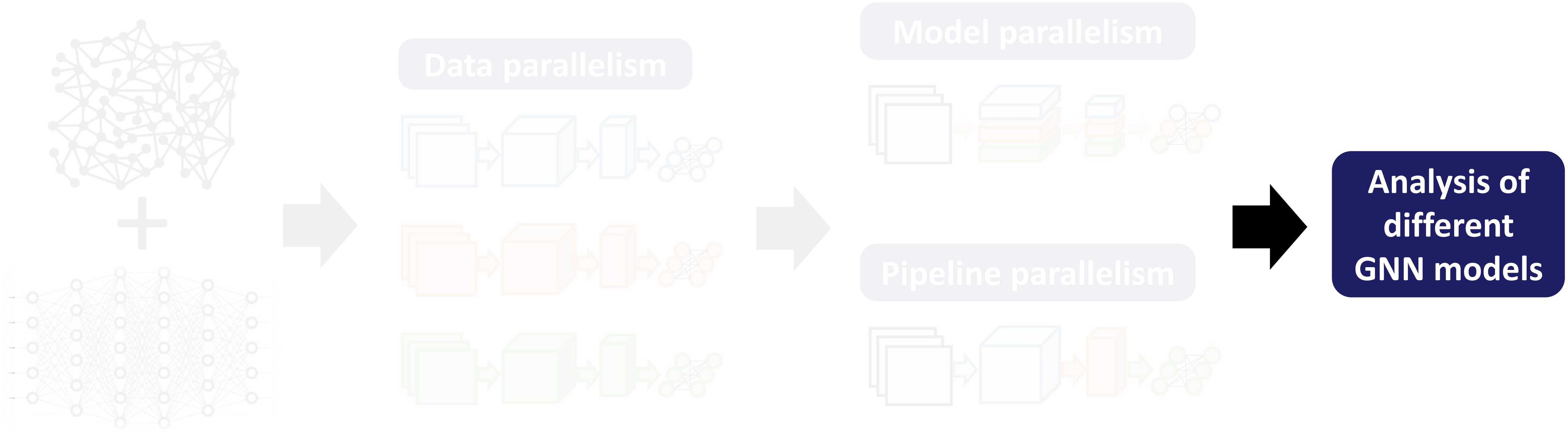
Presentation Overview

1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

Presentation Overview



1

Parallel and Distributed Graph Neural Networks: An In-Depth Concurrency Analysis

Maciej Besta and Torsten Hoefler
Department of Computer Science, ETH Zurich

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Global GNN formulations

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Formulations based on functions operating on single vertices & edges

Global GNN formulations

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Global GNN formulations

Formulations based on functions operating on single vertices & edges

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Global GNN formulations

Formulations based on functions operating on single vertices & edges

Vertex feature vector

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Global GNN formulations

Formulations based on functions operating on single vertices & edges

Vertex feature vector

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Global GNN formulations

Formulations based on functions operating on single vertices & edges

Vertex feature vector

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID

Neighbors

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Global GNN formulations

Formulations based on functions operating on single vertices & edges

Vertex feature vector

GNN layer

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID

Neighbors

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Global GNN formulations

Formulations based on functions operating on single vertices & edges

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex feature vector

GNN layer

Functions that specify the model details

Vertex ID

Neighbors

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Formulations based on functions operating on single vertices & edges

Global GNN formulations

Formulations based on operations on matrices grouping all vertex & edge related vectors

Vertex feature vector

GNN layer

Functions that specify the model details

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID

Neighbors

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Formulations based on functions operating on single vertices & edges

Global GNN formulations

Formulations based on operations on matrices grouping all vertex & edge related vectors

Vertex feature vector

GNN layer

Functions that specify the model details

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID

Neighbors

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Formulations based on functions operating on single vertices & edges

Vertex feature vector

GNN layer

Functions that specify the model details

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID

Neighbors

Global GNN formulations

Formulations based on operations on matrices grouping all vertex & edge related vectors

GNN layer

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Formulations based on functions operating on single vertices & edges

Vertex feature vector

GNN layer

Functions that specify the model details

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID

Neighbors

Global GNN formulations

Formulations based on operations on matrices grouping all vertex & edge related vectors

All vertex feature vectors grouped together

GNN layer

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Formulations based on functions operating on single vertices & edges

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

- Vertex feature vector: $\mathbf{h}_i^{(l+1)}$
- Vertex ID: i
- GNN layer: $(l+1)$, (l)
- Functions that specify the model details: ϕ, $\bigoplus$, ψ
- Neighbors: $j \in N(i)$

Global GNN formulations

Formulations based on operations on matrices grouping all vertex & edge related vectors

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

- All vertex feature vectors grouped together: $\mathbf{H}^{(l+1)}$
- Adjacency matrix: $\mathbf{A}$
- GNN layer: $(l+1)$, (l)

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Formulations based on functions operating on single vertices & edges

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

- Vertex feature vector: $\mathbf{h}_i^{(l+1)}$
- Vertex ID: i
- GNN layer: $(l+1)$, (l)
- Functions that specify the model details: ϕ, $\bigoplus$, ψ
- Neighbors: $j \in N(i)$

Global GNN formulations

Formulations based on operations on matrices grouping all vertex & edge related vectors

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\dots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

- All vertex feature vectors grouped together: $\mathbf{H}^{(l+1)}$
- Adjacency matrix: $\mathbf{A}$
- GNN layer: $(l+1)$, (l)
- Model parameters: $\mathbf{W}^{(l)}$

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Formulations based on functions operating on single vertices & edges

Vertex feature vector — GNN layer — Functions that specify the model details

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID — Neighbors

Global GNN formulations

Formulations based on operations on matrices grouping all vertex & edge related vectors

All vertex feature vectors grouped together — Adjacency matrix — GNN layer — Model parameters

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Functions that specify the model details

Taxonomy of Mathematical Formulations of GNNs

Local GNN formulations

Formulations based on functions operating on si…

Global GNN formulations

Formulations based on operations on matrices … …lated vectors

These formulations are not fully equivalent, i.e., some models only have known local formulations, and vice versa

Vertex feature vector

GNN laye…

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID

Neighbors

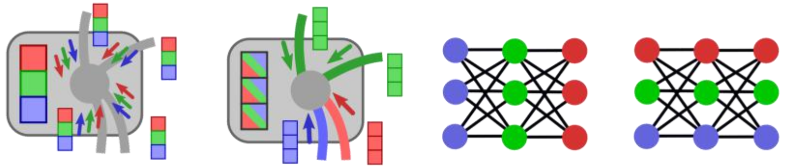
…er

Model parameters

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Functions that specify the model details

Taxonomy of Mathematical Formulations of GNNs



Local GNN formulations

Formulations based on functions operating on si...

Global GNN formulations

Formulations based on operations on matrices ...lated vectors

These formulations are not fully equivalent, i.e., some models only have known local formulations, and vice versa

Vertex feature vector

GNN laye...

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j\in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Vertex ID

Neighbors

...er

Model parameters

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Functions that specify the model details

Local Formulations of GNN Models

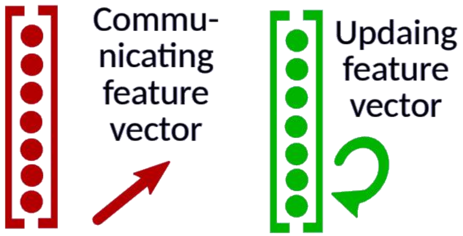
$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Local Formulations of GNN Models

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

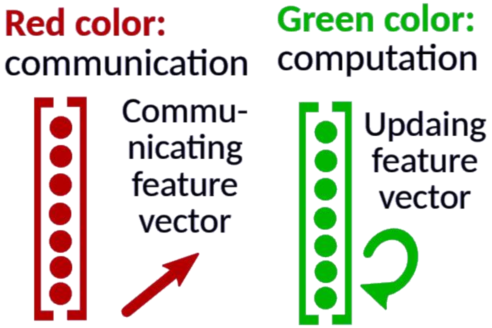
Red color: communication

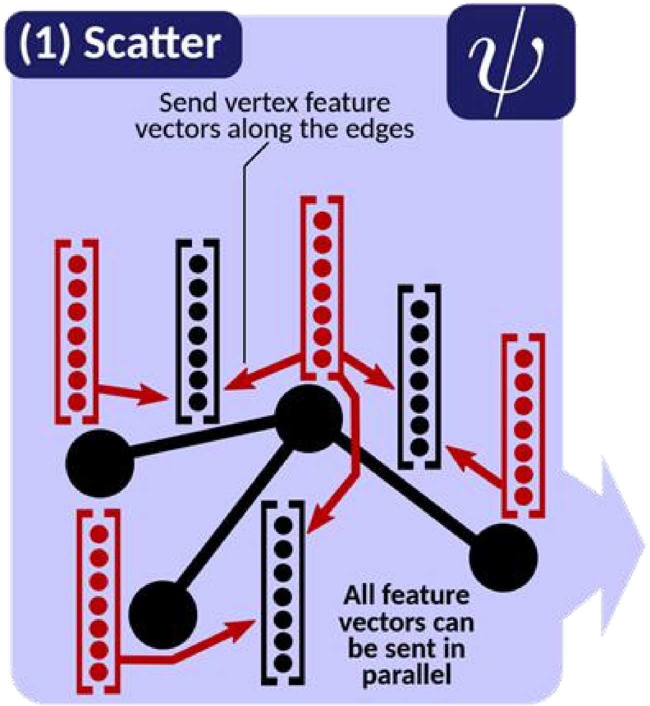
Green color: computation



Local Formulations of GNN Models

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$



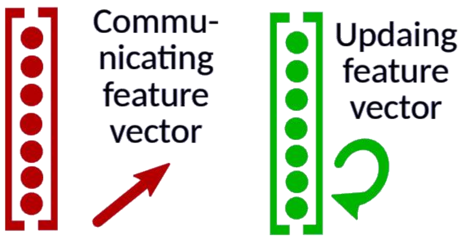


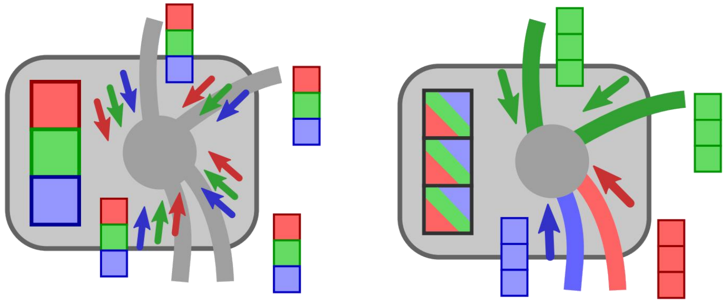
Local Formulations of GNN Models

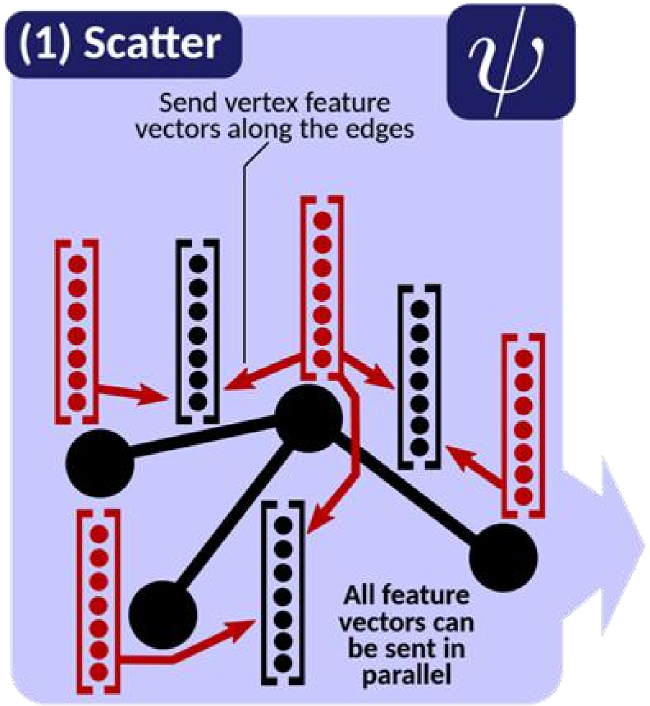
$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Red color: communication

Green color: computation





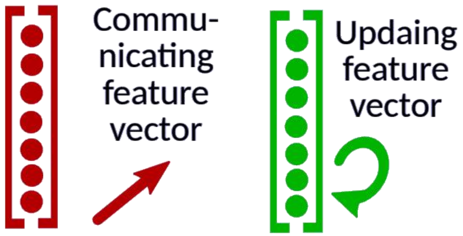


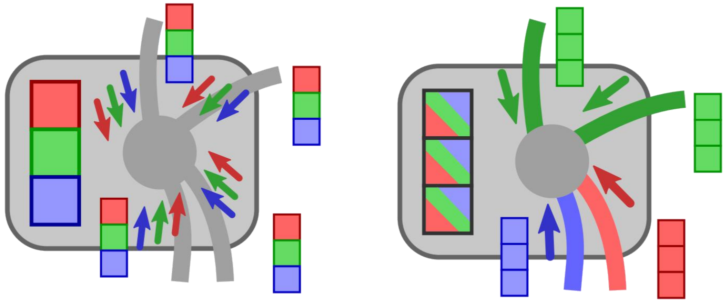
Local Formulations of GNN Models

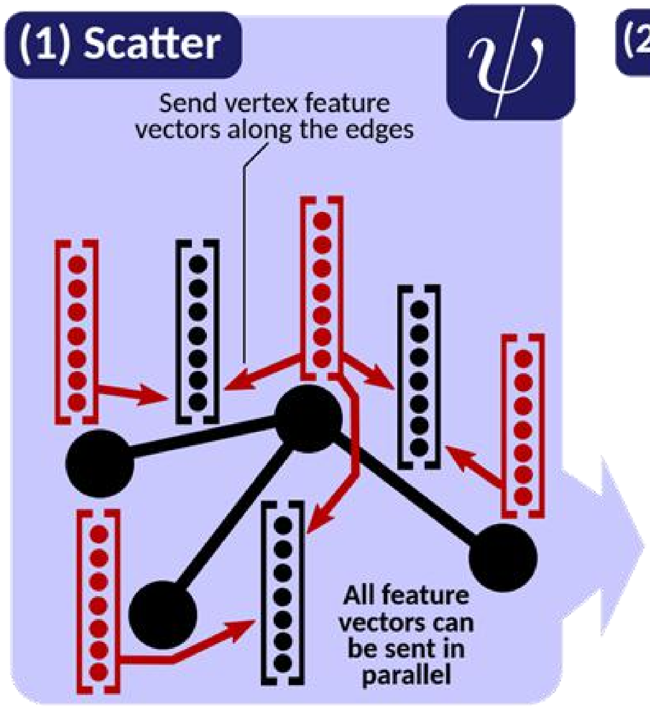
$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

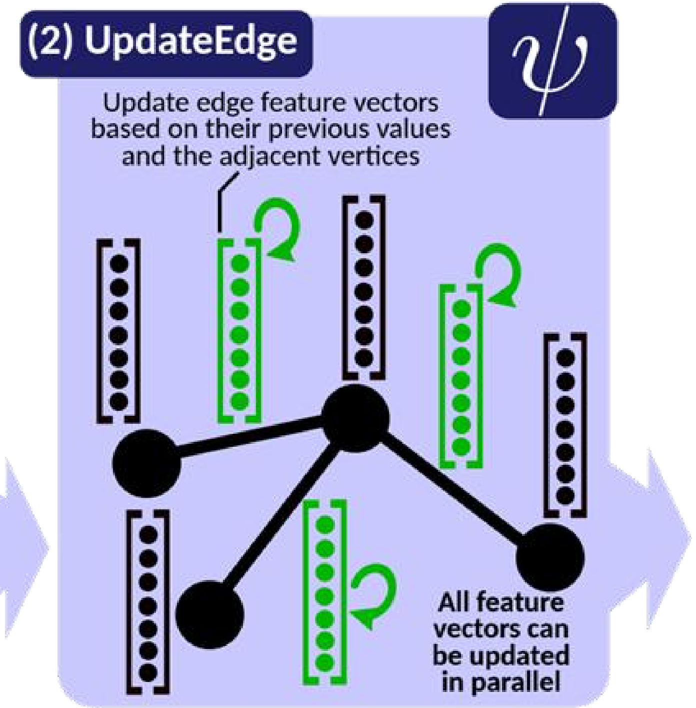
Red color: communication

Green color: computation



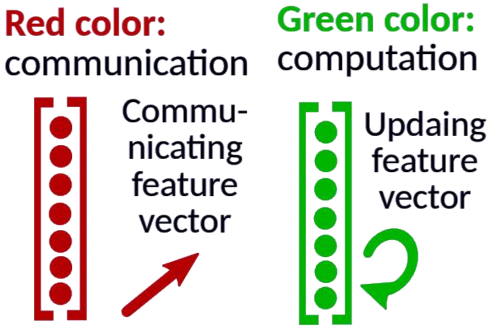


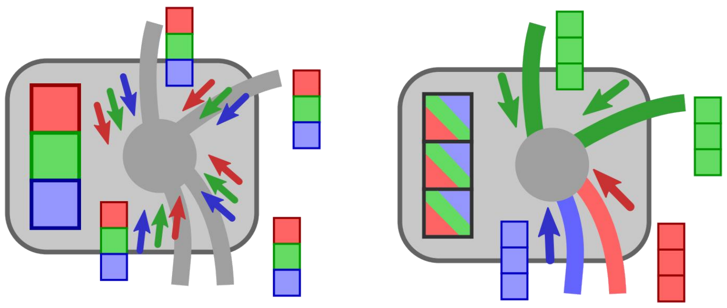


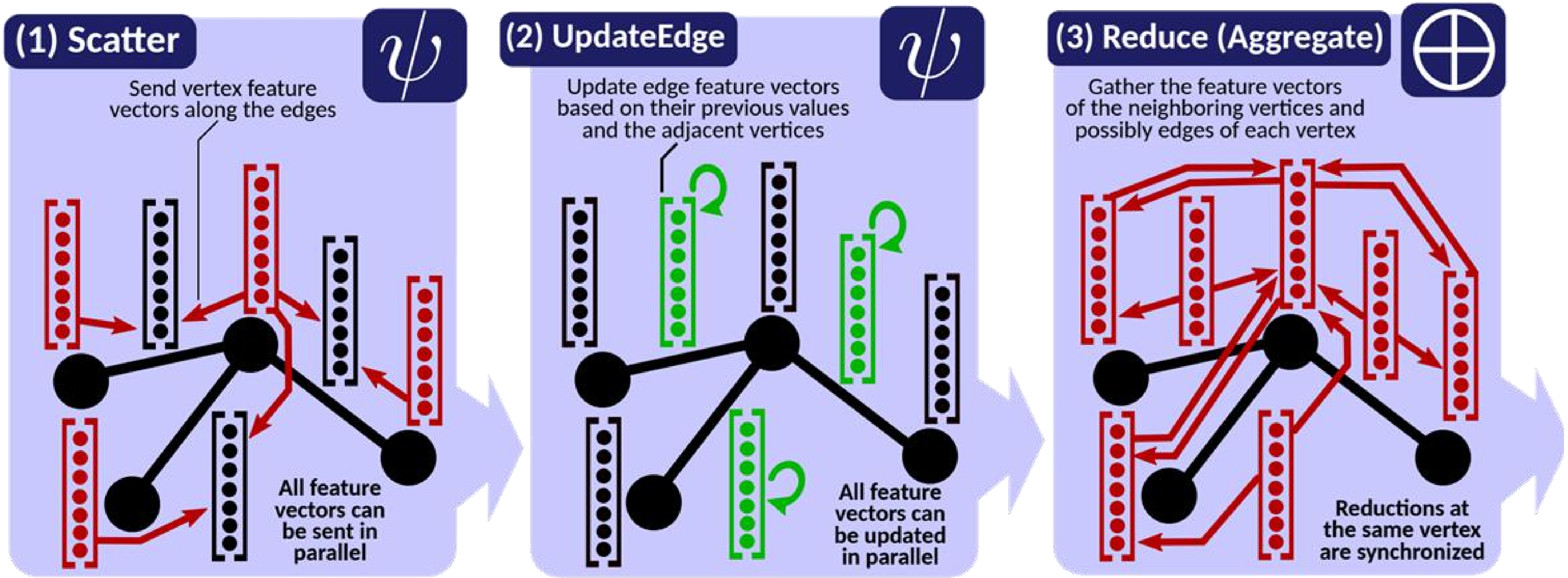


Local Formulations of GNN Models

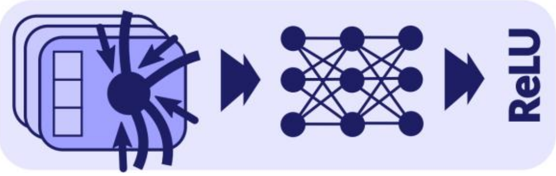
$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$







Local Formulations of GNN Models



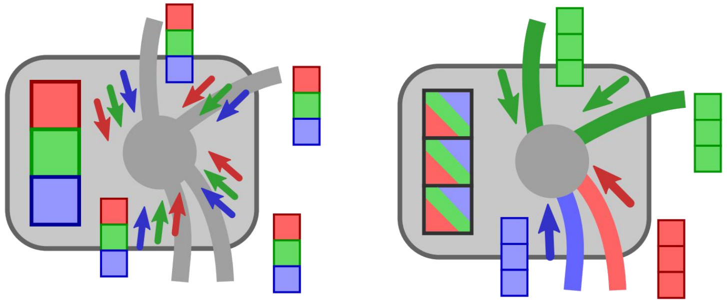
$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

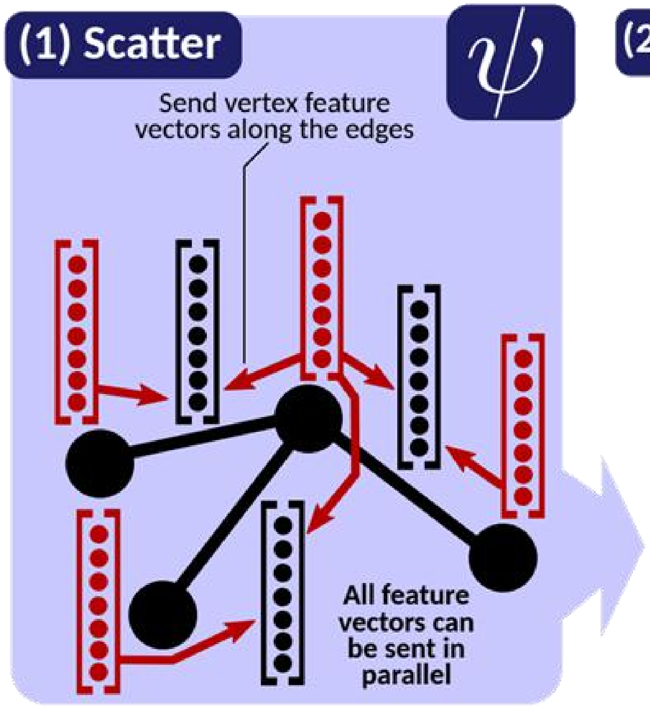
Red color: communication

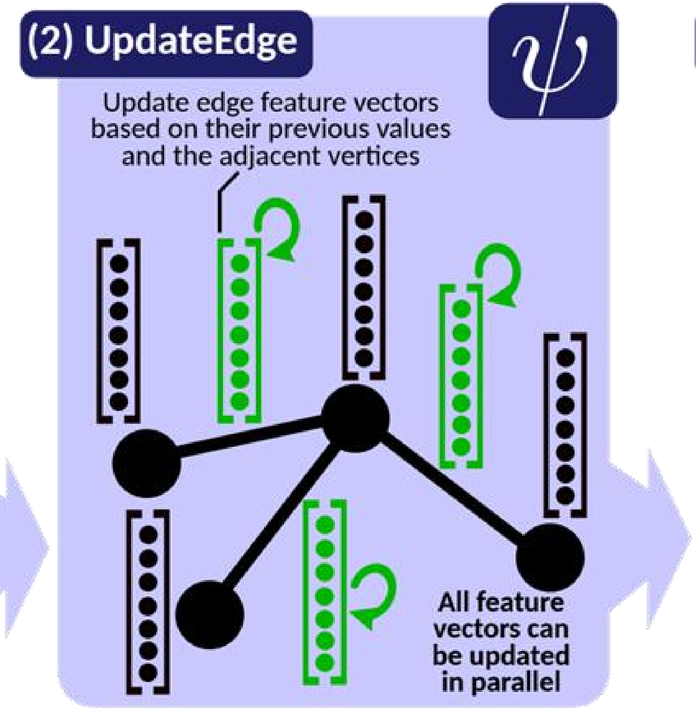
Commu-nicating feature vector

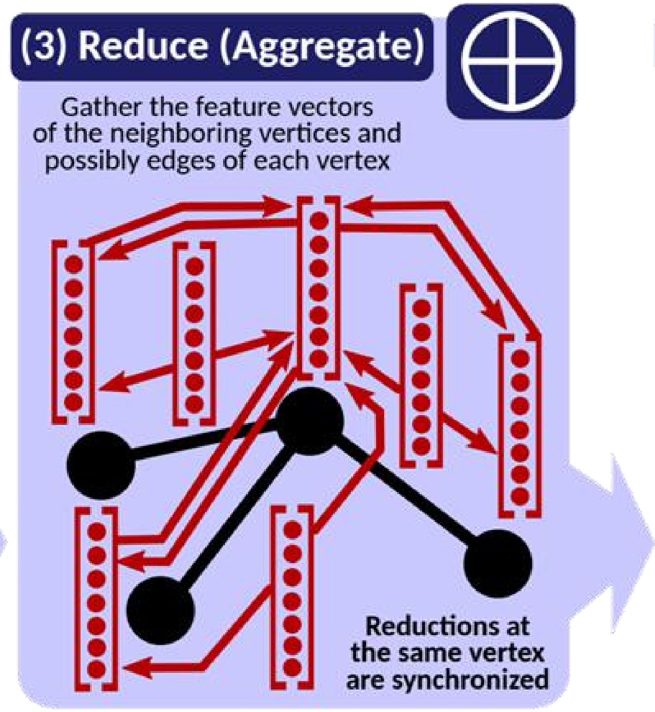
Green color: computation

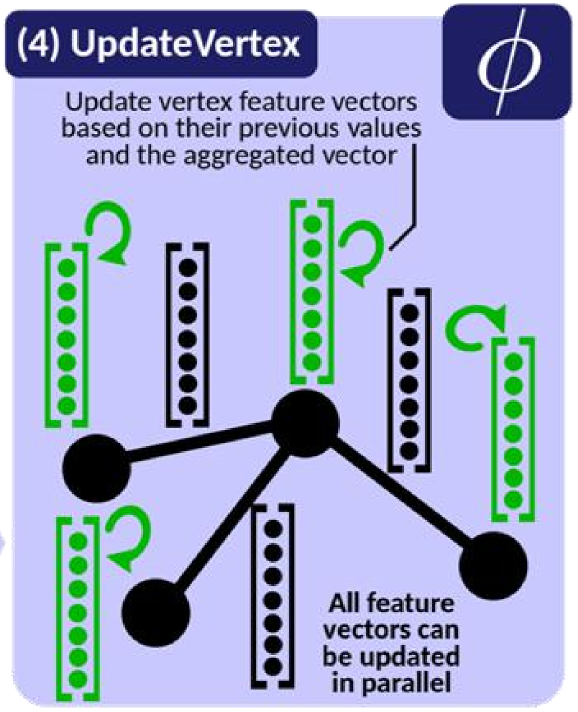
Updaing feature vector



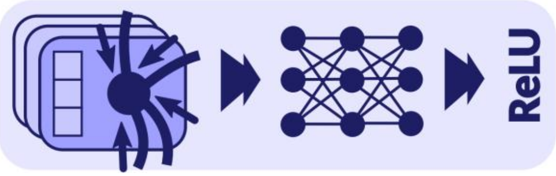




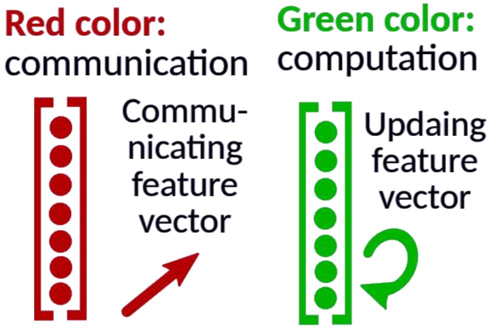


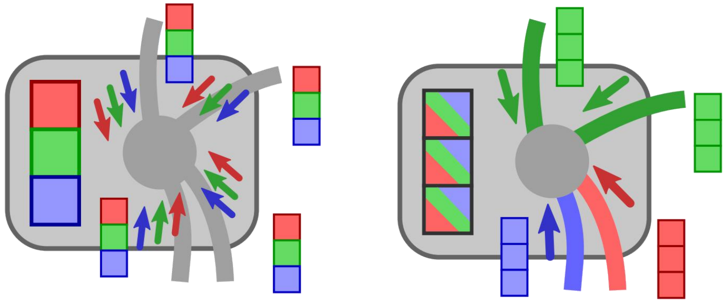


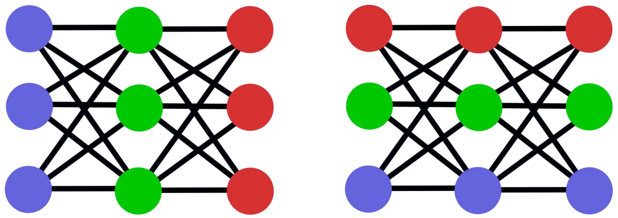
Local Formulations of GNN Models

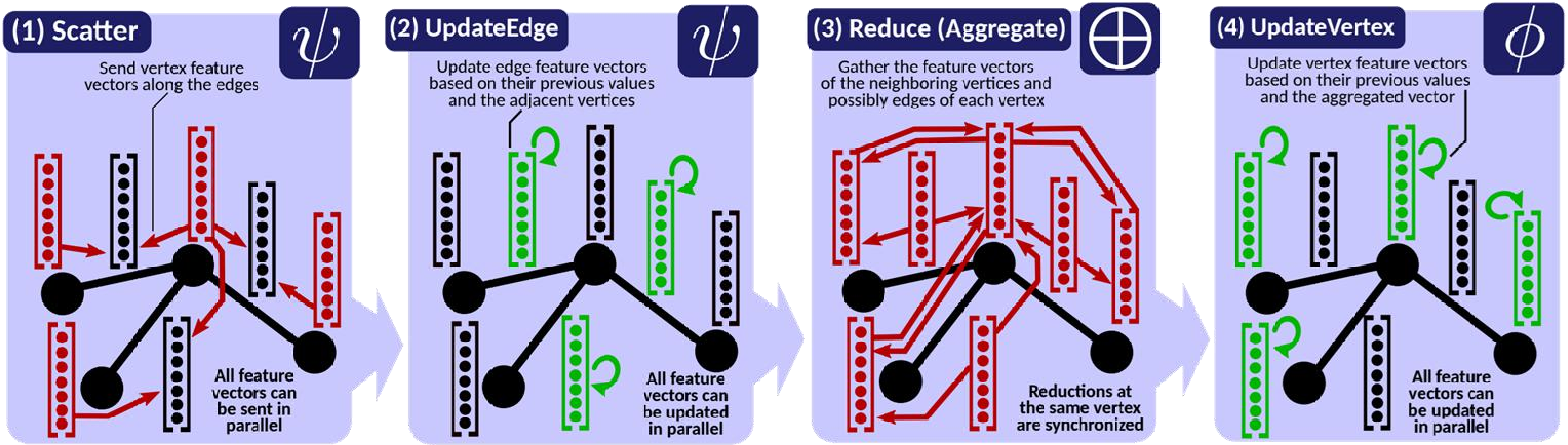


$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$









Parallel Analysis of ψ

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference
GCN [128]
GraphSAGE [101] (mean)
GIN [226]
CommNet [192]
Vanilla attention [201]
MoNet [158]
GAT [202]
Attention-based GNNs [196]
G-GCN [47]
GraphSAGE [101] (pooling)
EdgeConv [216] "choice 1"
EdgeConv [216] "choice 5"

Parallel Analysis of ψ

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference	Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$
GCN [128]	$\frac{1}{\sqrt{d_i d_j}}\mathbf{h}_j$
GraphSAGE [101] (mean)	$\mathbf{h}_j$
GIN [226]	$\mathbf{h}_j$
CommNet [192]	$\mathbf{h}_j$
Vanilla attention [201]	$\left(\mathbf{h}_i^T \cdot \mathbf{h}_j\right)\mathbf{h}_j$
MoNet [158]	$\exp\left(-\frac{1}{2}\left(\mathbf{h}_j - \mathbf{w}_j\right)^T \mathbf{W}_j^{-1}\left(\mathbf{h}_j - \mathbf{w}_j\right)\right)$
GAT [202]	$\frac{\exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \middle\| \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{y \in \widehat{N}(i)} \exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \middle\| \mathbf{W}\mathbf{h}_y\right]\right)\right)}\mathbf{h}_j$
Attention-based GNNs [196]	$w\frac{\mathbf{h}_i^T \cdot \mathbf{h}_j}{\lVert\mathbf{h}_i\rVert \lVert\mathbf{h}_j\rVert}\mathbf{h}_j$
G-GCN [47]	$\sigma\left(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{h}_j\right) \odot \mathbf{h}_j$
GraphSAGE [101] (pooling)	$\sigma\left(\mathbf{W}\mathbf{h}_j + \mathbf{w}\right)$
EdgeConv [216] "choice 1"	$\mathbf{W}\mathbf{h}_j$
EdgeConv [216] "choice 5"	$\sigma\left(\mathbf{W}_1\left(\mathbf{h}_j - \mathbf{h}_i\right) + \mathbf{W}_2\mathbf{h}_i\right)$

Parallel Analysis of ψ

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference	Class	Formulation for $\psi\left(\mathbf{h}_i, \mathbf{h}_j\right)$
GCN [128]	C-GNN	$\frac{1}{\sqrt{d_i d_j}} \mathbf{h}_j$
GraphSAGE [101] (mean)	C-GNN	$\mathbf{h}_j$
GIN [226]	C-GNN	$\mathbf{h}_j$
CommNet [192]	C-GNN	$\mathbf{h}_j$
Vanilla attention [201]	A-GNN	$\left(\mathbf{h}_i^T \cdot \mathbf{h}_j\right) \mathbf{h}_j$
MoNet [158]	A-GNN	$\exp\left(-\frac{1}{2}\left(\mathbf{h}_j - \mathbf{w}_j\right)^T \mathbf{W}_j^{-1}\left(\mathbf{h}_j - \mathbf{w}_j\right)\right)$
GAT [202]	A-GNN	$\frac{\exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{y \in \widehat{N}(i)} \exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_y\right]\right)\right)} \mathbf{h}_j$
Attention-based GNNs [196]	A-GNN	$w \frac{\mathbf{h}_i^T \cdot \mathbf{h}_j}{\Vert\mathbf{h}_i\Vert \Vert\mathbf{h}_j\Vert} \mathbf{h}_j$
G-GCN [47]	MP-GNN	$\sigma\left(\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \mathbf{h}_j\right) \odot \mathbf{h}_j$
GraphSAGE [101] (pooling)	MP-GNN	$\sigma\left(\mathbf{W}\mathbf{h}_j + \mathbf{w}\right)$
EdgeConv [216] "choice 1"	MP-GNN	$\mathbf{W}\mathbf{h}_j$
EdgeConv [216] "choice 5"	MP-GNN	$\sigma\left(\mathbf{W}_1\left(\mathbf{h}_j - \mathbf{h}_i\right) + \mathbf{W}_2 \mathbf{h}_i\right)$

Parallel Analysis of ψ

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Output of ψ

● **(static)**

Reference	Class	Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$
GCN [128]	C-GNN	$\frac{1}{\sqrt{d_i d_j}}\mathbf{h}_j$
GraphSAGE [101] (mean)	C-GNN	$\mathbf{h}_j$
GIN [226]	C-GNN	$\mathbf{h}_j$
CommNet [192]	C-GNN	$\mathbf{h}_j$
Vanilla attention [201]	A-GNN	$(\mathbf{h}_i^T \cdot \mathbf{h}_j)\,\mathbf{h}_j$
MoNet [158]	A-GNN	$\exp\left(-\frac{1}{2}(\mathbf{h}_j - \mathbf{w}_j)^T \mathbf{W}_j^{-1} (\mathbf{h}_j - \mathbf{w}_j)\right)$
GAT [202]	A-GNN	$\frac{\exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{y \in \widehat{N}(i)} \exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_y\right]\right)\right)}\mathbf{h}_j$
Attention-based GNNs [196]	A-GNN	$w\frac{\mathbf{h}_i^T \cdot \mathbf{h}_j}{\Vert\mathbf{h}_i\Vert \Vert\mathbf{h}_j\Vert}\mathbf{h}_j$
G-GCN [47]	MP-GNN	$\sigma(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{h}_j) \odot \mathbf{h}_j$
GraphSAGE [101] (pooling)	MP-GNN	$\sigma(\mathbf{W}\mathbf{h}_j + \mathbf{w})$
EdgeConv [216] "choice 1"	MP-GNN	$\mathbf{W}\mathbf{h}_j$
EdgeConv [216] "choice 5"	MP-GNN	$\sigma(\mathbf{W}_1(\mathbf{h}_j - \mathbf{h}_i) + \mathbf{W}_2\mathbf{h}_i)$

Parallel Analysis of ψ

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Output of ψ	Reference	Class	Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$
(static)	GCN [128]	C-GNN	$\frac{1}{\sqrt{d_i d_j}}\mathbf{h}_j$
(static)	GraphSAGE [101] (mean)	C-GNN	$\mathbf{h}_j$
(static)	GIN [226]	C-GNN	$\mathbf{h}_j$
(static)	CommNet [192]	C-GNN	$\mathbf{h}_j$
(learnt)	Vanilla attention [201]	A-GNN	$\left(\mathbf{h}_i^T \cdot \mathbf{h}_j\right)\mathbf{h}_j$
(learnt)	MoNet [158]	A-GNN	$\exp\left(-\frac{1}{2}\left(\mathbf{h}_j - \mathbf{w}_j\right)^T \mathbf{W}_j^{-1}\left(\mathbf{h}_j - \mathbf{w}_j\right)\right)$
(learnt)	GAT [202]	A-GNN	$\frac{\exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{y \in \widehat{N}(i)} \exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_y\right]\right)\right)}\mathbf{h}_j$
(learnt)	Attention-based GNNs [196]	A-GNN	$w \frac{\mathbf{h}_i^T \cdot \mathbf{h}_j}{\lVert\mathbf{h}_i\rVert \lVert\mathbf{h}_j\rVert}\mathbf{h}_j$
(learnt)	G-GCN [47]	MP-GNN	$\sigma\left(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{h}_j\right) \odot \mathbf{h}_j$
(learnt)	GraphSAGE [101] (pooling)	MP-GNN	$\sigma\left(\mathbf{W}\mathbf{h}_j + \mathbf{w}\right)$
(learnt)	EdgeConv [216] "choice 1"	MP-GNN	$\mathbf{W}\mathbf{h}_j$
(learnt)	EdgeConv [216] "choice 5"	MP-GNN	$\sigma\left(\mathbf{W}_1\left(\mathbf{h}_j - \mathbf{h}_i\right) + \mathbf{W}_2\mathbf{h}_i\right)$

Parallel Analysis of ψ

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Output of ψ	Reference	Class	Formulation for $\psi\left(\mathbf{h}_i, \mathbf{h}_j\right)$
(static)	GCN [128]	C-GNN	$\frac{1}{\sqrt{d_i d_j}}\mathbf{h}_j$
	GraphSAGE [101] (mean)	C-GNN	$\mathbf{h}_j$
	GIN [226]	C-GNN	$\mathbf{h}_j$
	CommNet [192]	C-GNN	$\mathbf{h}_j$
(learnt)	Vanilla attention [201]	A-GNN	$\left(\mathbf{h}_i^T \cdot \mathbf{h}_j\right)\mathbf{h}_j$
	MoNet [158]	A-GNN	$\exp\left(-\frac{1}{2}\left(\mathbf{h}_j - \mathbf{w}_j\right)^T \mathbf{W}_j^{-1}\left(\mathbf{h}_j - \mathbf{w}_j\right)\right)$
	GAT [202]	A-GNN	$\frac{\exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{y \in \widehat{N}(i)} \exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_y\right]\right)\right)}\mathbf{h}_j$
	Attention-based GNNs [196]	A-GNN	$w\frac{\mathbf{h}_i^T \cdot \mathbf{h}_j}{\Vert\mathbf{h}_i\Vert\Vert\mathbf{h}_j\Vert}\mathbf{h}_j$
	G-GCN [47]	MP-GNN	$\sigma\left(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{h}_j\right) \odot \mathbf{h}_j$
	GraphSAGE [101] (pooling)	MP-GNN	$\sigma\left(\mathbf{W}\mathbf{h}_j + \mathbf{w}\right)$
	EdgeConv [216] "choice 1"	MP-GNN	$\mathbf{W}\mathbf{h}_j$
	EdgeConv [216] "choice 5"	MP-GNN	$\sigma\left(\mathbf{W}_1\left(\mathbf{h}_j - \mathbf{h}_i\right) + \mathbf{W}_2\mathbf{h}_i\right)$

Parallel Analysis of ψ

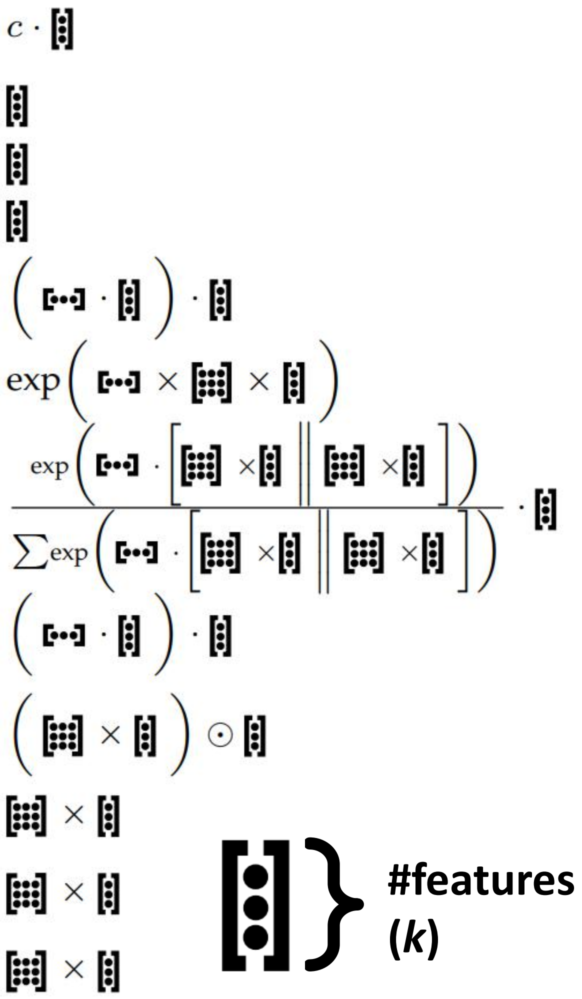
$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Output of ψ	Reference	Class	Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Dimensions & density of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$
● (static)	GCN [128]	C-GNN	$\frac{1}{\sqrt{d_i d_j}} \mathbf{h}_j$	$c \cdot [\vdots]$
	GraphSAGE [101] (mean)	C-GNN	$\mathbf{h}_j$	$[\vdots]$
	GIN [226]	C-GNN	$\mathbf{h}_j$	$[\vdots]$
	CommNet [192]	C-GNN	$\mathbf{h}_j$	$[\vdots]$
● (learnt)	Vanilla attention [201]	A-GNN	$(\mathbf{h}_i^T \cdot \mathbf{h}_j)\, \mathbf{h}_j$	$\left([\cdots] \cdot [\vdots]\right) \cdot [\vdots]$
	MoNet [158]	A-GNN	$\exp\left(-\frac{1}{2}(\mathbf{h}_j - \mathbf{w}_j)^T \mathbf{W}_j^{-1} (\mathbf{h}_j - \mathbf{w}_j)\right)$	$\exp\left([\cdots] \times [\vdots\vdots\vdots] \times [\vdots]\right)$
	GAT [202]	A-GNN	$\frac{\exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{y \in \widehat{N}(i)} \exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_y\right]\right)\right)} \mathbf{h}_j$	$\frac{\exp\left([\cdots] \cdot \left[[\vdots\vdots\vdots] \times [\vdots] \Vert [\vdots\vdots\vdots] \times [\vdots]\right]\right)}{\sum \exp\left([\cdots] \cdot \left[[\vdots\vdots\vdots] \times [\vdots] \Vert [\vdots\vdots\vdots] \times [\vdots]\right]\right)} \cdot [\vdots]$
	Attention-based GNNs [196]	A-GNN	$w \frac{\mathbf{h}_i^T \cdot \mathbf{h}_j}{\Vert\mathbf{h}_i\Vert \Vert\mathbf{h}_j\Vert} \mathbf{h}_j$	$\left([\cdots] \cdot [\vdots]\right) \cdot [\vdots]$
$[\vdots]$	G-GCN [47]	MP-GNN	$\sigma(\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \mathbf{h}_j) \odot \mathbf{h}_j$	$\left([\vdots\vdots\vdots] \times [\vdots]\right) \odot [\vdots]$
	GraphSAGE [101] (pooling)	MP-GNN	$\sigma(\mathbf{W}\mathbf{h}_j + \mathbf{w})$	$[\vdots\vdots\vdots] \times [\vdots]$
	EdgeConv [216] "choice 1"	MP-GNN	$\mathbf{W}\mathbf{h}_j$	$[\vdots\vdots\vdots] \times [\vdots]$
	EdgeConv [216] "choice 5"	MP-GNN	$\sigma(\mathbf{W}_1(\mathbf{h}_j - \mathbf{h}_i) + \mathbf{W}_2 \mathbf{h}_i)$	$[\vdots\vdots\vdots] \times [\vdots]$

Parallel Analysis of ψ

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j\in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Output of ψ	Reference	Class	Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Dimensions & density of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$
(static)	GCN [128]	C-GNN	$\frac{1}{\sqrt{d_i d_j}}\mathbf{h}_j$	
	GraphSAGE [101] (mean)	C-GNN	$\mathbf{h}_j$	
	GIN [226]	C-GNN	$\mathbf{h}_j$	
	CommNet [192]	C-GNN	$\mathbf{h}_j$	
(learnt)	Vanilla attention [201]	A-GNN	$(\mathbf{h}_i^T \cdot \mathbf{h}_j)\,\mathbf{h}_j$	
	MoNet [158]	A-GNN	$\exp\left(-\frac{1}{2}(\mathbf{h}_j - \mathbf{w}_j)^T \mathbf{W}_j^{-1}(\mathbf{h}_j - \mathbf{w}_j)\right)$	
	GAT [202]	A-GNN	$\frac{\exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{y\in\widehat{N}(i)} \exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_y\right]\right)\right)}\mathbf{h}_j$	
	Attention-based GNNs [196]	A-GNN	$w\frac{\mathbf{h}_i^T \cdot \mathbf{h}_j}{\Vert\mathbf{h}_i\Vert\Vert\mathbf{h}_j\Vert}\mathbf{h}_j$	
	G-GCN [47]	MP-GNN	$\sigma(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{h}_j) \odot \mathbf{h}_j$	
	GraphSAGE [101] (pooling)	MP-GNN	$\sigma(\mathbf{W}\mathbf{h}_j + \mathbf{w})$	
	EdgeConv [216] "choice 1"	MP-GNN	$\mathbf{W}\mathbf{h}_j$	
	EdgeConv [216] "choice 5"	MP-GNN	$\sigma(\mathbf{W}_1(\mathbf{h}_j - \mathbf{h}_i) + \mathbf{W}_2\mathbf{h}_i)$	



Parallel Analysis of ψ

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Output of ψ	Reference	Class	Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Dimensions & density of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Work & depth of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$	
● (static)	GCN [128]	C-GNN	$\frac{1}{\sqrt{d_i d_j}}\mathbf{h}_j$	$c \cdot [\vdots]$	$O(k)$	$O(1)$
	GraphSAGE [101] (mean)	C-GNN	$\mathbf{h}_j$	$[\vdots]$	$O(1)$	$O(1)$
	GIN [226]	C-GNN	$\mathbf{h}_j$	$[\vdots]$	$O(1)$	$O(1)$
	CommNet [192]	C-GNN	$\mathbf{h}_j$	$[\vdots]$	$O(1)$	$O(1)$
● (learnt)	Vanilla attention [201]	A-GNN	$(\mathbf{h}_i^T \cdot \mathbf{h}_j)\,\mathbf{h}_j$	$([\cdots] \cdot [\vdots]) \cdot [\vdots]$	$O(k)$	$O(\log k)$
	MoNet [158]	A-GNN	$\exp\left(-\frac{1}{2}(\mathbf{h}_j - \mathbf{w}_j)^T \mathbf{W}_j^{-1} (\mathbf{h}_j - \mathbf{w}_j)\right)$	$\exp([\cdots] \times [\vdots\vdots\vdots] \times [\vdots])$	$O(k^2)$	$O(\log k)$
	GAT [202]	A-GNN	$\frac{\exp\left(\sigma\left(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_j]\right)\right)}{\sum_{y \in \widehat{N}(i)} \exp\left(\sigma\left(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_y]\right)\right)}\mathbf{h}_j$	$\frac{\exp\left([\cdots] \cdot \left[[\vdots\vdots\vdots] \times [\vdots] \Vert [\vdots\vdots\vdots] \times [\vdots]\right]\right)}{\sum \exp\left([\cdots] \cdot \left[[\vdots\vdots\vdots] \times [\vdots] \Vert [\vdots\vdots\vdots] \times [\vdots]\right]\right)} \cdot [\vdots]$	$O(dk^2)$	$O(\log k + \log d)$
	Attention-based GNNs [196]	A-GNN	$w \frac{\mathbf{h}_i^T \cdot \mathbf{h}_j}{\Vert\mathbf{h}_i\Vert \Vert\mathbf{h}_j\Vert}\mathbf{h}_j$	$([\cdots] \cdot [\vdots]) \cdot [\vdots]$	$O(k)$	$O(\log k)$
$[\vdots]$	G-GCN [47]	MP-GNN	$\sigma(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{h}_j) \odot \mathbf{h}_j$	$([\vdots\vdots\vdots] \times [\vdots]) \odot [\vdots]$	$O(k^2)$	$O(\log k)$
	GraphSAGE [101] (pooling)	MP-GNN	$\sigma(\mathbf{W}\mathbf{h}_j + \mathbf{w})$	$[\vdots\vdots\vdots] \times [\vdots]$	$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 1"	MP-GNN	$\mathbf{W}\mathbf{h}_j$	$[\vdots\vdots\vdots] \times [\vdots]$	$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 5"	MP-GNN	$\sigma(\mathbf{W}_1(\mathbf{h}_j - \mathbf{h}_i) + \mathbf{W}_2\mathbf{h}_i)$	$[\vdots\vdots\vdots] \times [\vdots]$	$O(k^2)$	$O(\log k)$

$[\vdots]$ } **#features (*k*)**

Parallel Analysis of ψ

n: #vertices in a graph m: #edges in a graph
L: #layers in a GNN k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j\in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

work: total #operations
depth: longest chain of sequential dependencies

Output of ψ	Reference	Class	Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Dimensions & density of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Work & depth of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$	
(static)	GCN [128]	C-GNN	$\frac{1}{\sqrt{d_i d_j}}\mathbf{h}_j$	$c \cdot$	$O(k)$	$O(1)$
	GraphSAGE [101] (mean)	C-GNN	$\mathbf{h}_j$		$O(1)$	$O(1)$
	GIN [226]	C-GNN	$\mathbf{h}_j$		$O(1)$	$O(1)$
	CommNet [192]	C-GNN	$\mathbf{h}_j$		$O(1)$	$O(1)$
(learnt)	Vanilla attention [201]	A-GNN	$(\mathbf{h}_i^T \cdot \mathbf{h}_j)\,\mathbf{h}_j$	$(\cdot) \cdot$	$O(k)$	$O(\log k)$
	MoNet [158]	A-GNN	$\exp\left(-\frac{1}{2}(\mathbf{h}_j - \mathbf{w}_j)^T \mathbf{W}_j^{-1}(\mathbf{h}_j - \mathbf{w}_j)\right)$	$\exp(\times \times)$	$O(k^2)$	$O(\log k)$
	GAT [202]	A-GNN	$\frac{\exp\left(\sigma\left(\mathbf{a}^T\cdot\left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{y\in\hat{N}(i)}\exp\left(\sigma\left(\mathbf{a}^T\cdot\left[\mathbf{W}\mathbf{h}_i \Vert \mathbf{W}\mathbf{h}_y\right]\right)\right)}\mathbf{h}_j$	$\frac{\exp(\cdot [\times \Vert \times])}{\sum \exp(\cdot [\times \Vert \times])} \cdot$	$O(dk^2)$	$O(\log k + \log d)$
	Attention-based GNNs [196]	A-GNN	$w\frac{\mathbf{h}_i^T\cdot\mathbf{h}_j}{\Vert\mathbf{h}_i\Vert\Vert\mathbf{h}_j\Vert}\mathbf{h}_j$	$(\cdot) \cdot$	$O(k)$	$O(\log k)$
	G-GCN [47]	MP-GNN	$\sigma(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{h}_j) \odot \mathbf{h}_j$	$(\times) \odot$	$O(k^2)$	$O(\log k)$
	GraphSAGE [101] (pooling)	MP-GNN	$\sigma(\mathbf{W}\mathbf{h}_j + \mathbf{w})$	$\times$	$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 1"	MP-GNN	$\mathbf{W}\mathbf{h}_j$	$\times$	$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 5"	MP-GNN	$\sigma(\mathbf{W}_1(\mathbf{h}_j - \mathbf{h}_i) + \mathbf{W}_2\mathbf{h}_i)$	$\times$	$O(k^2)$	$O(\log k)$

#features (k)

Parallel Analysis of ψ

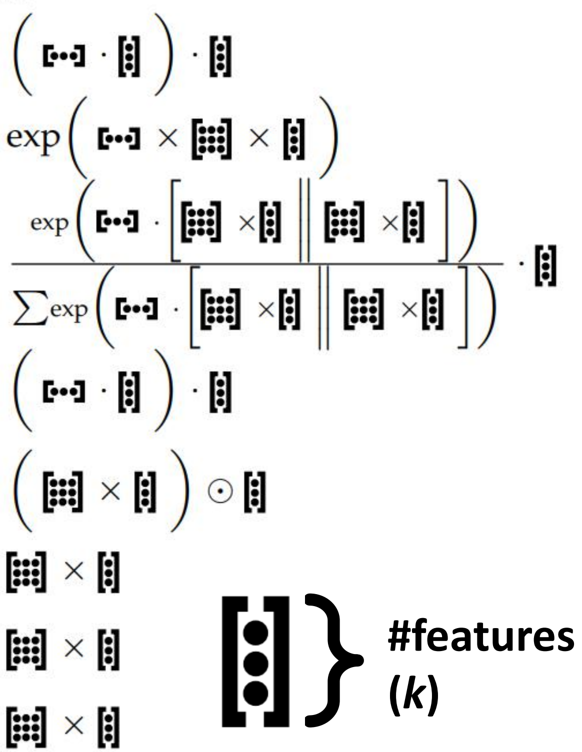
n: #vertices in a graph m: #edges in a graph
L: #layers in a GNN k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

C-GNNs almost always take *O(1)* depth and *O(1)* work

work: total #operations
depth: longest chain of sequential dependencies

Output of ψ	Reference	Class	Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Dimensions & density of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Work & depth of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$	
(static)	GCN [128]	C-GNN		$c \cdot$	$O(k)$	$O(1)$
	GraphSAGE [101] (mean)	C-GNN			$O(1)$	$O(1)$
	GIN [226]	C-GNN			$O(1)$	$O(1)$
	CommNet [192]	C-GNN			$O(1)$	$O(1)$
(learnt)	Vanilla attention [201]	A-GNN			$O(k)$	$O(\log k)$
	MoNet [158]	A-GNN		exp	$O(k^2)$	$O(\log k)$
	GAT [202]	A-GNN		exp / $\sum$exp	$O(dk^2)$	$O(\log k + \log d)$
	Attention-based GNNs [196]	A-GNN			$O(k)$	$O(\log k)$
	G-GCN [47]	MP-GNN			$O(k^2)$	$O(\log k)$
	GraphSAGE [101] (pooling)	MP-GNN			$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 1"	MP-GNN			$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 5"	MP-GNN			$O(k^2)$	$O(\log k)$



Parallel Analysis of ψ

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

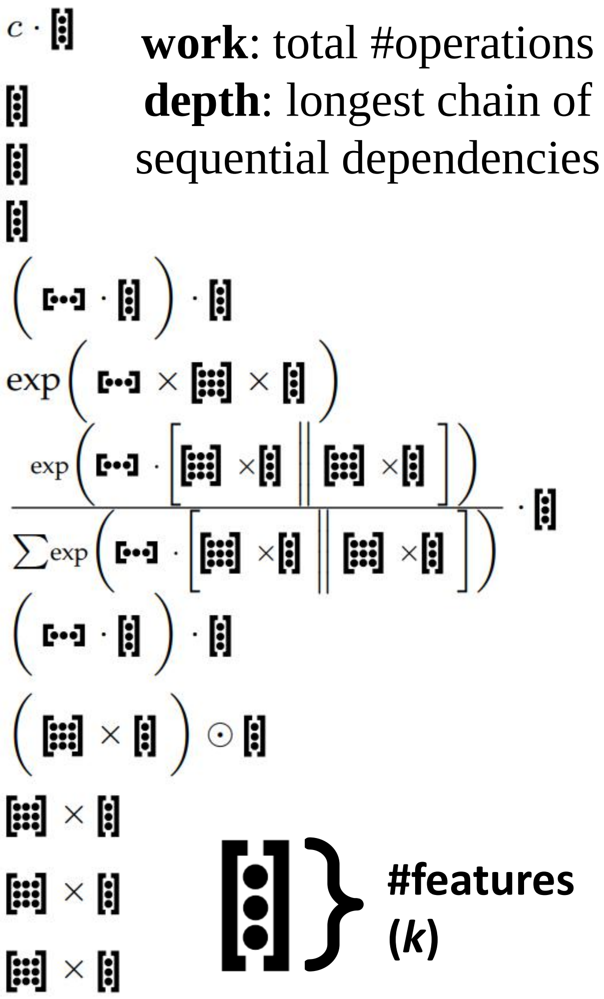
Output of ψ	Reference	Class	Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Dimensions & density of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Work & depth of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$	
(static)	GCN [128]	C-GNN			$O(k)$	$O(1)$
	GraphSAGE [101] (mean)	C-GNN			$O(1)$	$O(1)$
	GIN [226]	C-GNN			$O(1)$	$O(1)$
	CommNet [192]	C-GNN			$O(1)$	$O(1)$
(learnt)	Vanilla attention [201]	A-GNN			$O(k)$	$O(\log k)$
	MoNet [158]	A-GNN			$O(k^2)$	$O(\log k)$
	GAT [202]	A-GNN			$O(dk^2)$	$O(\log k + \log d)$
	Attention-based GNNs [196]	A-GNN			$O(k)$	$O(\log k)$
	G-GCN [47]	MP-GNN			$O(k^2)$	$O(\log k)$
	GraphSAGE [101] (pooling)	MP-GNN			$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 1"	MP-GNN			$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 5"	MP-GNN			$O(k^2)$	$O(\log k)$

C-GNNs almost always take *O(1)* depth and *O(1)* work

A-GNNs and MP-GNNs have much more complex formulations

Nearly all A-GNNs and MP-GNNs have $O(k^2)$ work and $O(\log k)$ depth

work: total #operations
depth: longest chain of sequential dependencies



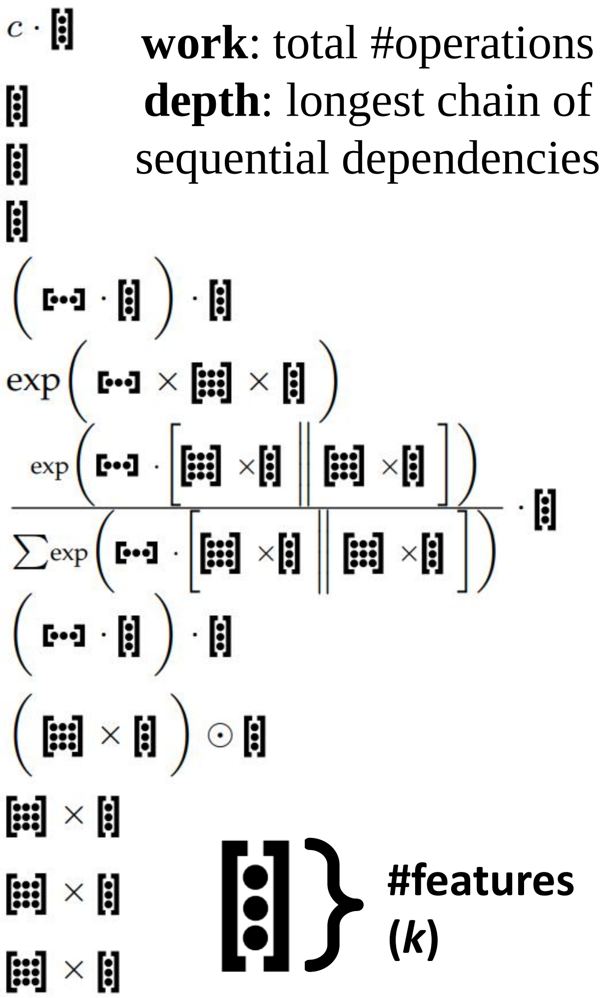
Parallel Analysis of ψ

n: #vertices in a graph m: #edges in a graph
L: #layers in a GNN k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Output of ψ	Reference	Class	Work & depth of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$ (work)	(depth)
(static)	GCN [128]	C-GNN	$O(k)$	$O(1)$
	GraphSAGE [101] (mean)	C-GNN	$O(1)$	$O(1)$
	GIN [226]	C-GNN	$O(1)$	$O(1)$
	CommNet [192]	C-GNN	$O(1)$	$O(1)$
(learnt)	Vanilla attention [201]	A-GNN	$O(k)$	$O(\log k)$
	MoNet [158]	A-GNN	$O(k^2)$	$O(\log k)$
	GAT [202]	A-GNN	$O(dk^2)$	$O(\log k + \log d)$
	Attention-based GNNs [196]	A-GNN	$O(k)$	$O(\log k)$
	G-GCN [47]	MP-GNN	$O(k^2)$	$O(\log k)$
	GraphSAGE [101] (pooling)	MP-GNN	$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 1"	MP-GNN	$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 5"	MP-GNN	$O(k^2)$	$O(\log k)$

Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$ / Dimensions & density of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$:



C-GNNs almost always take *O(1)* depth and *O(1)* work

A-GNNs and MP-GNNs have much more complex formulations

Nearly all A-GNNs and MP-GNNs have *O(k²)* work and *O(log k)* depth

Most computationally intense model, has also logarithmic depth

work: total #operations
depth: longest chain of sequential dependencies

Parallel Analysis of ψ

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Output of ψ	Reference	Class	Work of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$	Depth of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$
(static)	GCN [128]	C-GNN	$O(k)$	$O(1)$
	GraphSAGE [101] (mean)	C-GNN	$O(1)$	$O(1)$
	GIN [226]	C-GNN	$O(1)$	$O(1)$
	CommNet [192]	C-GNN	$O(1)$	$O(1)$
(learnt)	Vanilla attention [201]	A-GNN	$O(k)$	$O(\log k)$
	MoNet [158]	A-GNN	$O(k^2)$	$O(\log k)$
	GAT [202]	A-GNN	$O(dk^2)$	$O(\log k + \log d)$
	Attention-based GNNs [196]	A-GNN	$O(k)$	$O(\log k)$
	G-GCN [47]	MP-GNN	$O(k^2)$	$O(\log k)$
	GraphSAGE [101] (pooling)	MP-GNN	$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 1"	MP-GNN	$O(k^2)$	$O(\log k)$
	EdgeConv [216] "choice 5"	MP-GNN	$O(k^2)$	$O(\log k)$

Formulation for $\psi(\mathbf{h}_i, \mathbf{h}_j)$:

C-GNNs almost always take *O(1)* depth and *O(1)* work

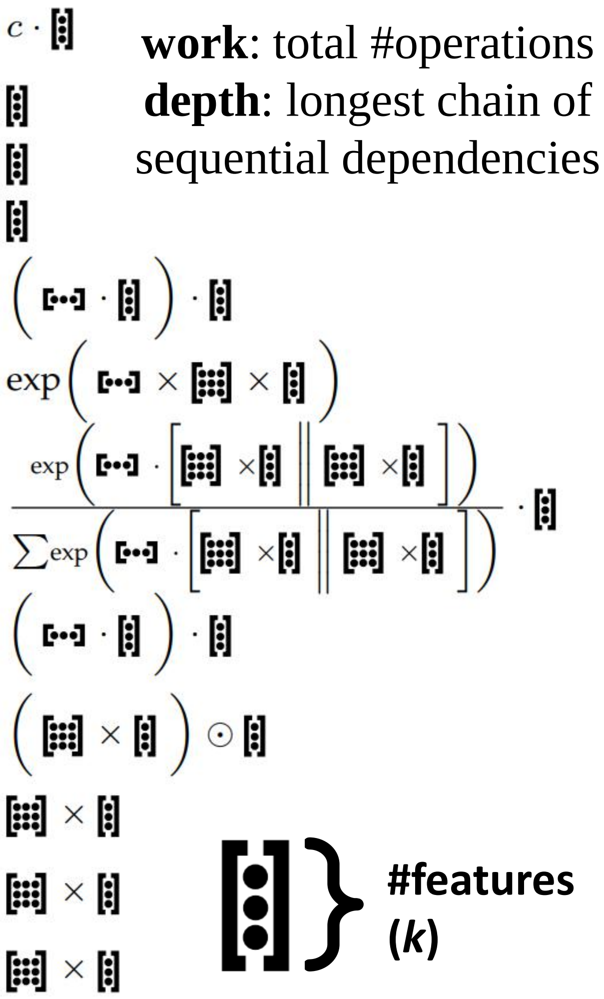
A-GNNs and MP-GNNs have much more complex formulations

Nearly all A-GNNs and MP-GNNs have *O(k²)* work and *O(log k)* depth

Most computationally intense model, has also logarithmic depth

Most models use GEMV; matrices and vectors are dense

Dimensions & density of one execution of $\psi(\mathbf{h}_i, \mathbf{h}_j)$:



work: total #operations
depth: longest chain of sequential dependencies

Parallel Analysis of ϕ

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference	Class
GCN [128]	C-GNN
GraphSAGE [101] (mean)	C-GNN
GIN [226]	C-GNN
CommNet [192]	C-GNN
Vanilla attention [201]	A-GNN
GAT [202]	A-GNN
Attention-based GNNs [196]	A-GNN
MoNet [158]	A-GNN
G-GCN [47]	MP-GNN
GraphSAGE [101] (pooling)	MP-GNN
EdgeConv [216] “choice 1”	MP-GNN
EdgeConv [216] “choice 5”	MP-GNN

Parallel Analysis of ϕ

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference	Class	Formulation of ϕ for $\mathbf{h}_i^{(l)}$; $\psi(\mathbf{h}_i, \mathbf{h}_j)$ are stated in Table 5
GCN [128]	C-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_j)\right)$
GraphSAGE [101] (mean)	C-GNN	$\mathbf{W} \times \left(\frac{1}{d_i} \cdot \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_j)\right)\right)$
GIN [226]	C-GNN	$\mathrm{MLP}\left((1+\epsilon)\mathbf{h}_i + \sum_{j \in N(i)} \psi(\mathbf{h}_j)\right)$
CommNet [192]	C-GNN	$\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \times \left(\sum_{j \in N+(i)} \psi(\mathbf{h}_j)\right)$
Vanilla attention [201]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$
GAT [202]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$
Attention-based GNNs [196]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$
MoNet [158]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_j)\right)$
G-GCN [47]	MP-GNN	$\mathbf{W} \times \left(\sum_{j \in N+(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$
GraphSAGE [101] (pooling)	MP-GNN	$\left(\mathbf{W} \times \left(\mathbf{h}_i \middle\| \left(\max_{j \in N(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)\right)\right)$
EdgeConv [216] "choice 1"	MP-GNN	$\sum_{j \in N+(i)} \psi(\mathbf{h}_j)$
EdgeConv [216] "choice 5"	MP-GNN	$\max_{j \in N+(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)$

Parallel Analysis of ϕ

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference	Class	Formulation of ϕ for $\mathbf{h}_i^{(l)}$; $\psi(\mathbf{h}_i, \mathbf{h}_j)$ are stated in Table 5	Dimensions & density of computing $\phi(\cdot)$, excluding $\psi(\cdot)$
GCN [128]	C-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_j)\right)$	$\times \sum$
GraphSAGE [101] (mean)	C-GNN	$\mathbf{W} \times \left(\frac{1}{d_i} \cdot \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_j)\right)\right)$	$\times \sum$
GIN [226]	C-GNN	$\text{MLP}\left((1+\epsilon)\mathbf{h}_i + \sum_{j \in N(i)} \psi(\mathbf{h}_j)\right)$	K times: $\times \ldots \times$; $\times \sum$
CommNet [192]	C-GNN	$\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \times \left(\sum_{j \in N+(i)} \psi(\mathbf{h}_j)\right)$	$\times \sum$
Vanilla attention [201]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$	$\times \sum$
GAT [202]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$	$\times \sum$
Attention-based GNNs [196]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$	$\times \sum$
MoNet [158]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_j)\right)$	$\times \sum$
G-GCN [47]	MP-GNN	$\mathbf{W} \times \left(\sum_{j \in N+(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$	$\times \sum$
GraphSAGE [101] (pooling)	MP-GNN	$\left(\mathbf{W} \times \left(\mathbf{h}_i \middle\Vert \left(\max_{j \in N(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)\right)\right)$	$\times \left(\Vert \left(\times \sum \right) \right)$
EdgeConv [216] "choice 1"	MP-GNN	$\sum_{j \in N+(i)} \psi(\mathbf{h}_j)$	$\sum$
EdgeConv [216] "choice 5"	MP-GNN	$\max_{j \in N+(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)$	$\sum$

Parallel Analysis of ϕ

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

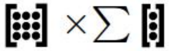
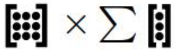
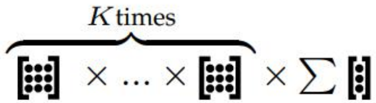
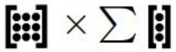
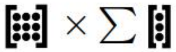
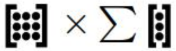
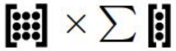
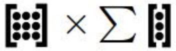
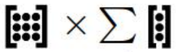
$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference	Class	Formulation of ϕ for $\mathbf{h}_i^{(l)}$; $\psi(\mathbf{h}_i, \mathbf{h}_j)$ are stated in Table 5	Dimensions & density of computing $\phi(\cdot)$, excluding $\psi(\cdot)$	Work & depth (a whole training iteration or inference, including ψ from Table 5)	
GCN [128]	C-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_j)\right)$	▦ $\times \sum$ ▮	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
GraphSAGE [101] (mean)	C-GNN	$\mathbf{W} \times \left(\frac{1}{d_i} \cdot \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_j)\right)\right)$	▦ $\times \sum$ ▮	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
GIN [226]	C-GNN	$\mathrm{MLP}\left((1+\epsilon)\mathbf{h}_i + \sum_{j \in N(i)} \psi(\mathbf{h}_j)\right)$	$\overbrace{▦ \times \ldots \times ▦}^{K \text{ times}}$ $\times \sum$ ▮	$O(Lmk + LKnk^2)$	$O(L \log d + LK \log k)$
CommNet [192]	C-GNN	$\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \times \left(\sum_{j \in N+(i)} \psi(\mathbf{h}_j)\right)$	▦ $\times \sum$ ▮	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
Vanilla attention [201]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$	▦ $\times \sum$ ▮	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
GAT [202]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$	▦ $\times \sum$ ▮	$O(Lmdk^2 + Lnk^2)$	$O(L \log d + L \log k)$
Attention-based GNNs [196]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$	▦ $\times \sum$ ▮	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
MoNet [158]	A-GNN	$\mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi(\mathbf{h}_j)\right)$	▦ $\times \sum$ ▮	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
G-GCN [47]	MP-GNN	$\mathbf{W} \times \left(\sum_{j \in N+(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)$	▦ $\times \sum$ ▮	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
GraphSAGE [101] (pooling)	MP-GNN	$\left(\mathbf{W} \times \left(\mathbf{h}_i \middle\| \left(\max_{j \in N(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right)\right)\right)$	▦ $\times$ (▮ ‖ (▦ $\times \sum$ ▮))	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
EdgeConv [216] "choice 1"	MP-GNN	$\sum_{j \in N+(i)} \psi(\mathbf{h}_j)$	$\sum$ ▮	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
EdgeConv [216] "choice 5"	MP-GNN	$\max_{j \in N+(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)$	$\sum$ ▮	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$

Parallel Analysis of ϕ

n: #vertices in a graph m: #edges in a graph
L: #layers in a GNN k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference	**Class**	**Formulation of ϕ for $\mathbf{h}_i^{(l)}$; $\psi(\mathbf{h}_i, \mathbf{h}_j)$ are stated in Table 5**	**Dimensions & density of computing $\phi(\cdot)$, excluding $\psi(\cdot)$**	**Work & depth** (a whole training iteration or inference, **including ψ from Table 5**)	
GCN [128]	C-GNN			$O(Lmk + Lnk^2)$	$O(L\log d + L\log k)$
GraphSAGE [101] (mean)	C-GNN			$O(Lmk + Lnk^2)$	$O(L\log d + L\log k)$
GIN [226]	C-GNN			$O(Lmk + LKnk^2)$	$O(L\log d + LK\log k)$
CommNet [192]	C-GNN			$O(Lmk + Lnk^2)$	$O(L\log d + L\log k)$
Vanilla attention [201]	A-GNN			$O(Lmk + Lnk^2)$	$O(L\log d + L\log k)$
GAT [202]	A-GNN			$O(Lmdk^2 + Lnk^2)$	$O(L\log d + L\log k)$
Attention-based GNNs [196]	A-GNN			$O(Lmk + Lnk^2)$	$O(L\log d + L\log k)$
MoNet [158]	A-GNN			$O(Lmk^2 + Lnk^2)$	$O(L\log d + L\log k)$
G-GCN [47]	MP-GNN			$O(Lmk^2 + Lnk^2)$	$O(L\log d + L\log k)$
GraphSAGE [101] (pooling)	MP-GNN			$O(Lmk^2 + Lnk^2)$	$O(L\log d + L\log k)$
EdgeConv [216] "choice 1"	MP-GNN			$O(Lmk^2 + Lnk^2)$	$O(L\log d + L\log k)$
EdgeConv [216] "choice 5"	MP-GNN			$O(Lmk^2 + Lnk^2)$	$O(L\log d + L\log k)$

Depth of one GNN layer is almost always logarithmic (nice)

Parallel Analysis of ϕ

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference	Class	Formulation of ϕ for $\mathbf{h}_i^{(l)}$; $\psi(\mathbf{h}_i, \mathbf{h}_j)$ are stated in Table 5	Dimensions & density of computing $\phi(\cdot)$, excluding $\psi(\cdot)$	Work & depth (a whole training iteration or inference, including ψ from Table 5)	
GCN [128]	C-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
GraphSAGE [101] (mean)	C-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
GIN [226]	C-GNN		K times: $\times \ldots \times$, $\times \sum$	$O(Lmk + LKnk^2)$	$O(L \log d + LK \log k)$
CommNet [192]	C-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
Vanilla attention [201]	A-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
GAT [202]	A-GNN		$\times \sum$	$O(Lmdk^2 + Lnk^2)$	$O(L \log d + L \log k)$
Attention-based GNNs [196]	A-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
MoNet [158]	A-GNN		$\times \sum$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
G-GCN [47]	MP-GNN		$\times \sum$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
GraphSAGE [101] (pooling)	MP-GNN		$\times \left(\,\Vert\, \left(\times \sum \right) \right)$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
EdgeConv [216] "choice 1"	MP-GNN		$\sum$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
EdgeConv [216] "choice 5"	MP-GNN		$\sum$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$

Depth of one GNN layer is almost always logarithmic (nice)

Work varies, being the highest for GAT. Depth is still logarithmic ☺

Parallel Analysis of ϕ

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Reference	Class	Formulation of ϕ for $\mathbf{h}_i^{(l)}$; $\psi(\mathbf{h}_i, \mathbf{h}_j)$ are stated in Table 5	Dimensions & density of computing $\phi(\cdot)$, excluding $\psi(\cdot)$	Work & depth (a whole training iteration or inference, including ψ from Table 5)	
GCN [128]	C-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
GraphSAGE [101] (mean)	C-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
GIN [226]	C-GNN		K times: $\times \dots \times$; $\times \sum$	$O(Lmk + LKnk^2)$	$O(L \log d + LK \log k)$
CommNet [192]	C-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
Vanilla attention [201]	A-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
GAT [202]	A-GNN		$\times \sum$	$O(Lmdk^2 + Lnk^2)$	$O(L \log d + L \log k)$
Attention-based GNNs [196]	A-GNN		$\times \sum$	$O(Lmk + Lnk^2)$	$O(L \log d + L \log k)$
MoNet [158]	A-GNN		$\times \sum$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
G-GCN [47]	MP-GNN		$\times \sum$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
GraphSAGE [101] (pooling)	MP-GNN		$\times \left(\Vert \left(\times \sum \right) \right)$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
EdgeConv [216] "choice 1"	MP-GNN		$\sum$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$
EdgeConv [216] "choice 5"	MP-GNN		$\sum$	$O(Lmk^2 + Lnk^2)$	$O(L \log d + L \log k)$

Depth of one GNN layer is almost always logarithmic (nice)

Work varies, being the highest for GAT. Depth is still logarithmic ☺

All the models entail matrix-vector dense products and a sum of up to *d* dense vectors

Parallel Analysis of $\bigoplus$

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Parallel Analysis of $\bigoplus$

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Aggregation is almost always a commutative and associative operation such as min, max, or plain sum

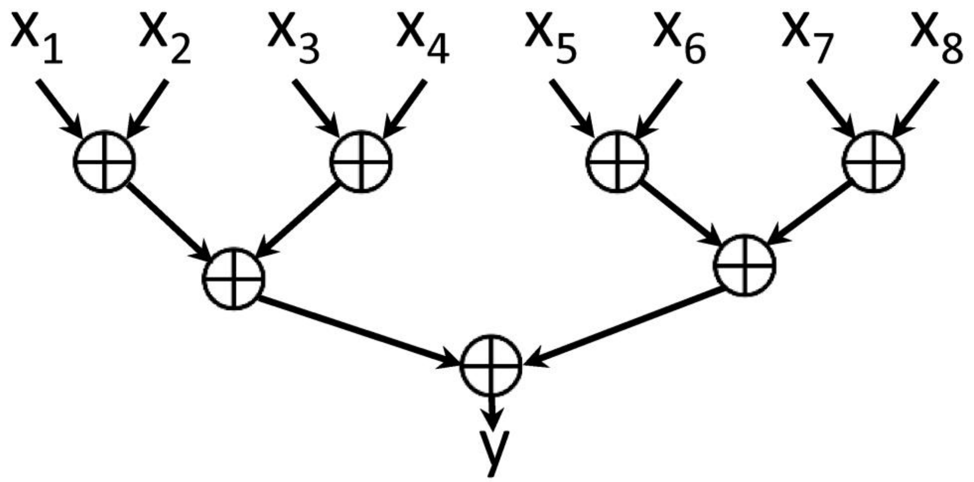
Parallel Analysis of $\bigoplus$

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Aggregation is almost always a commutative and associative operation such as min, max, or plain sum

Using established parallel tree reduction algorithms, it takes *O(log d)* depth and *O(k d)* work.



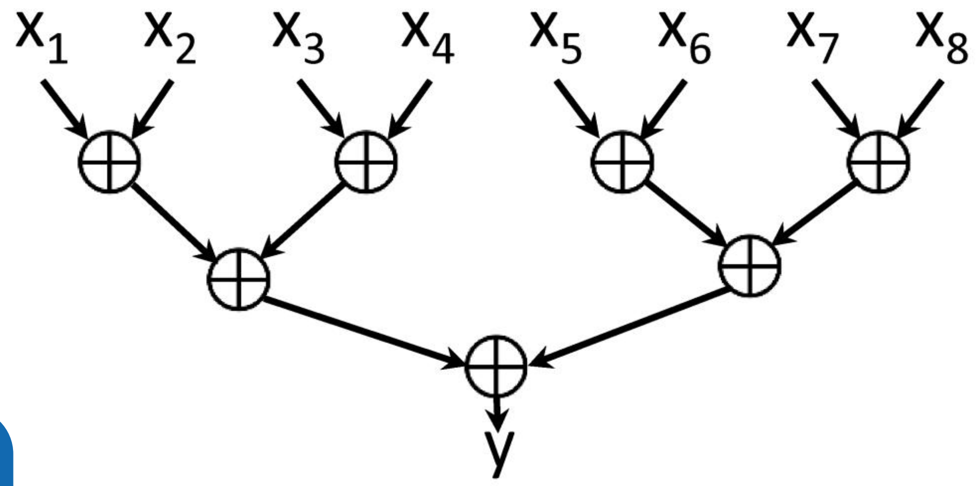
Parallel Analysis of $\bigoplus$

n: #vertices in a graph
m: #edges in a graph
L: #layers in a GNN
k: #features
d: maximum degree

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

Aggregation is almost always a commutative and associative operation such as min, max, or plain sum

Using established parallel tree reduction algorithms, it takes *O(log d)* depth and *O(k d)* work.



Aggregation is the bottleneck in depth in many considered models. This is because *d* (maximum vertex degree) is usually much larger than *k*

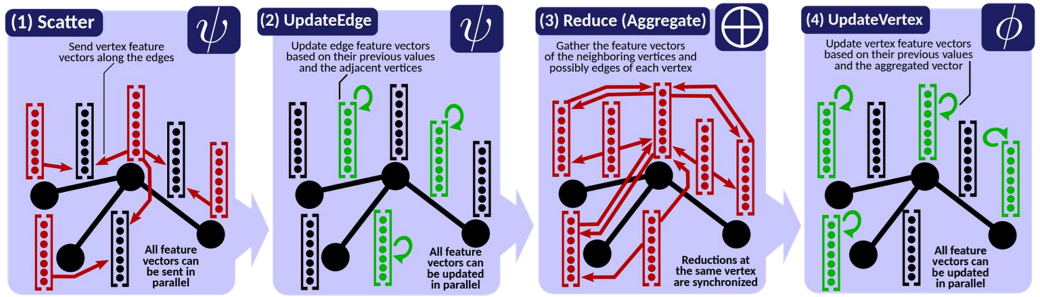
Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

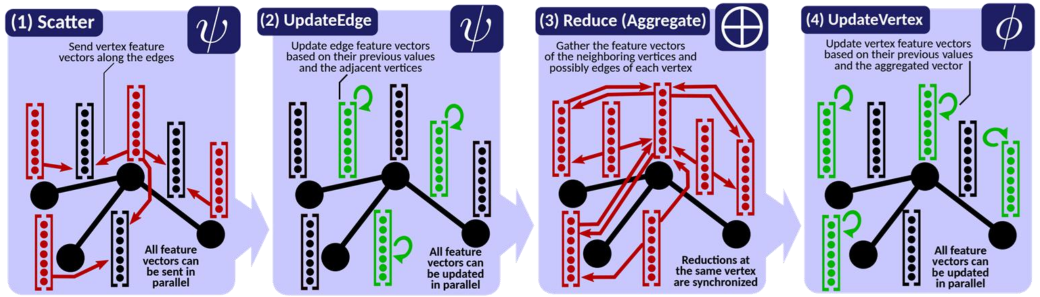
Local formulation cheatsheet:

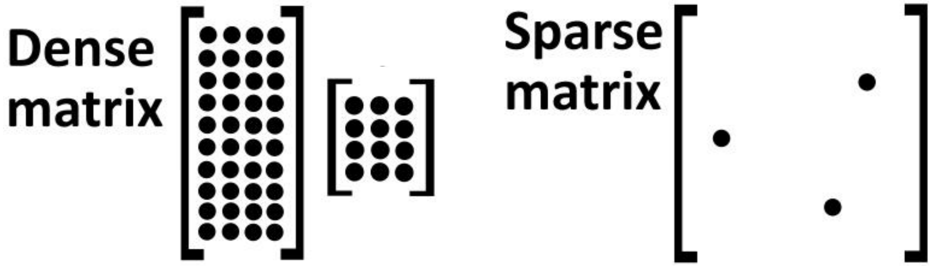


Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:

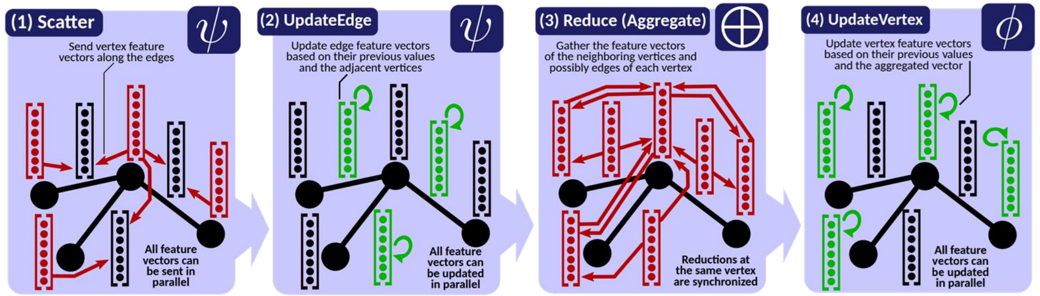


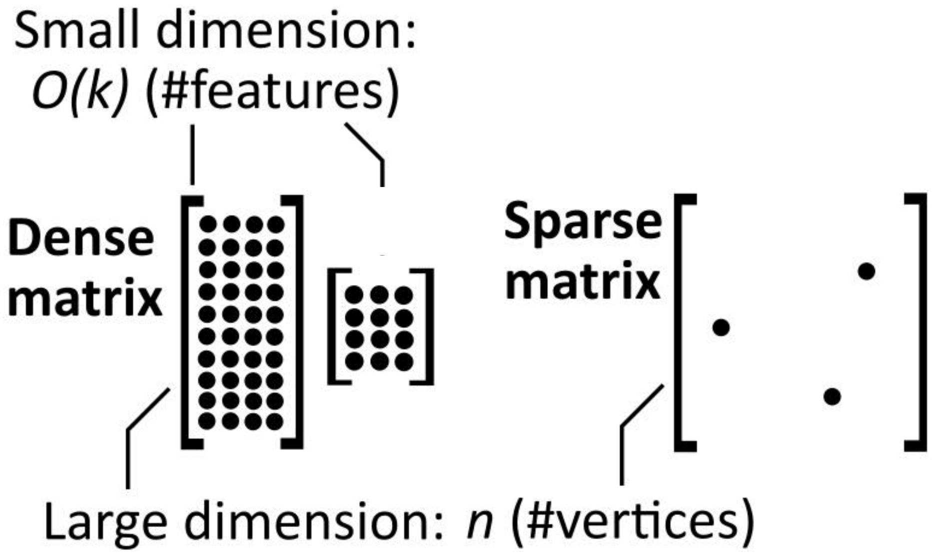


Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:

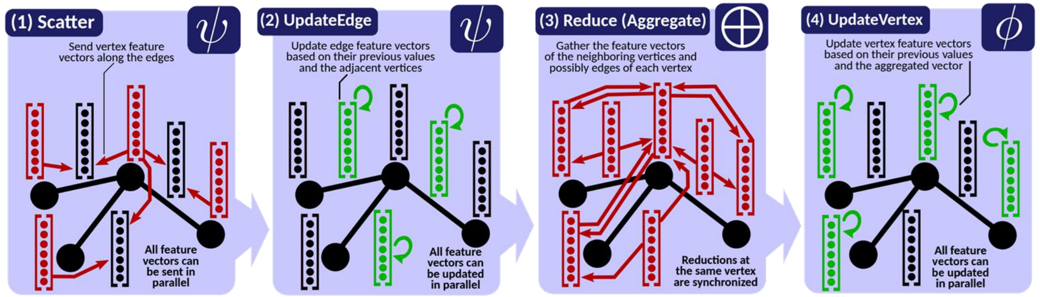


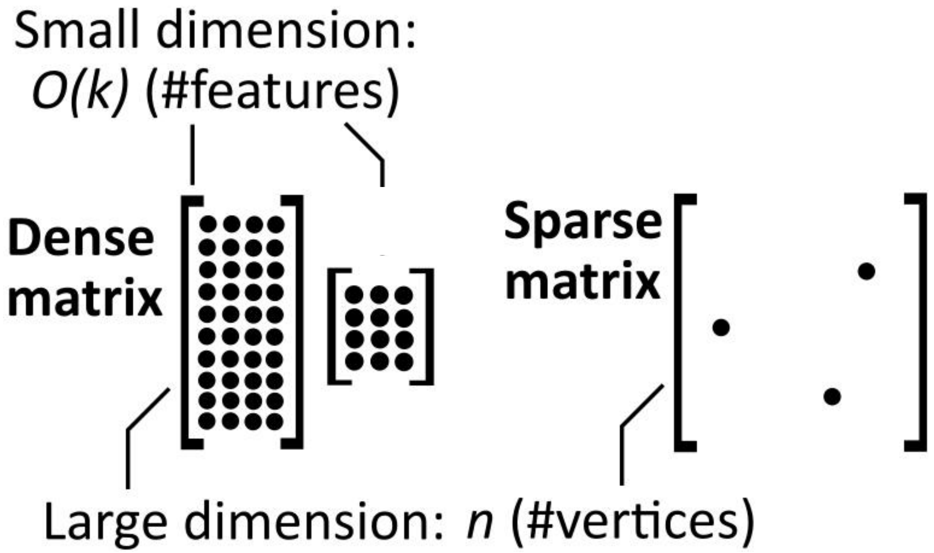


Global Formulations of GNN Models

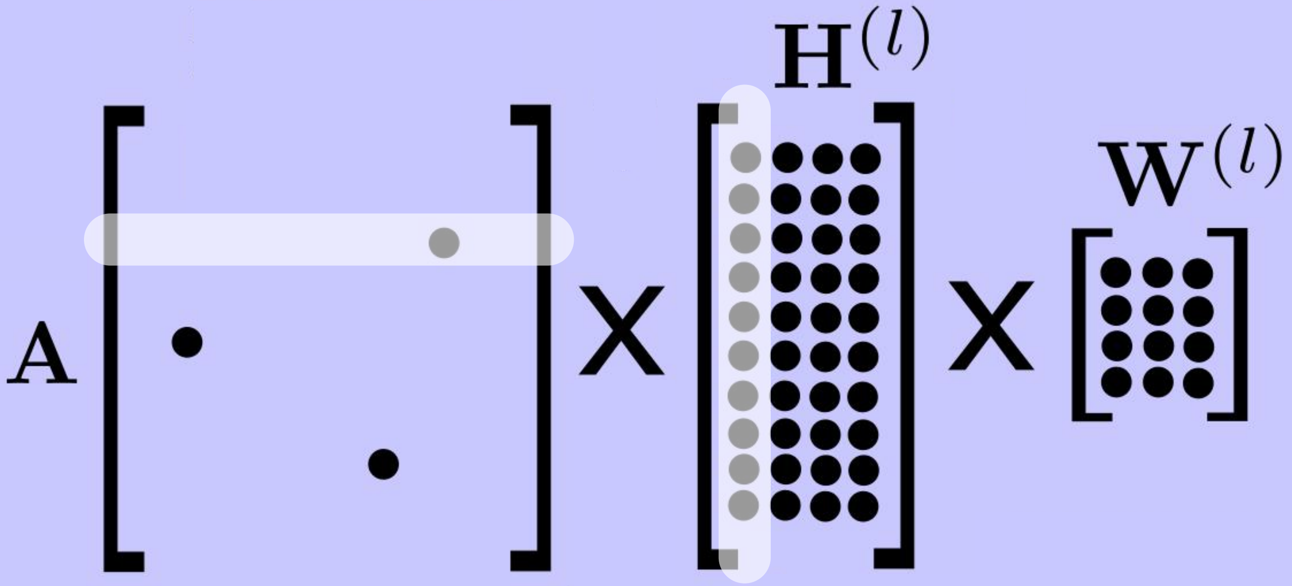
$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:





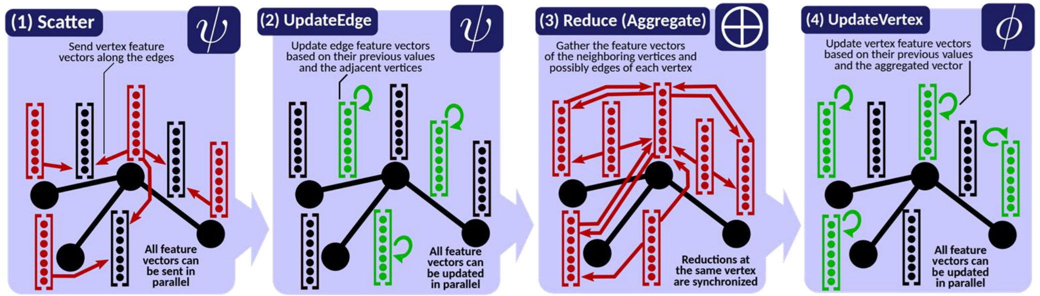
Example model: Graph Convolution Network

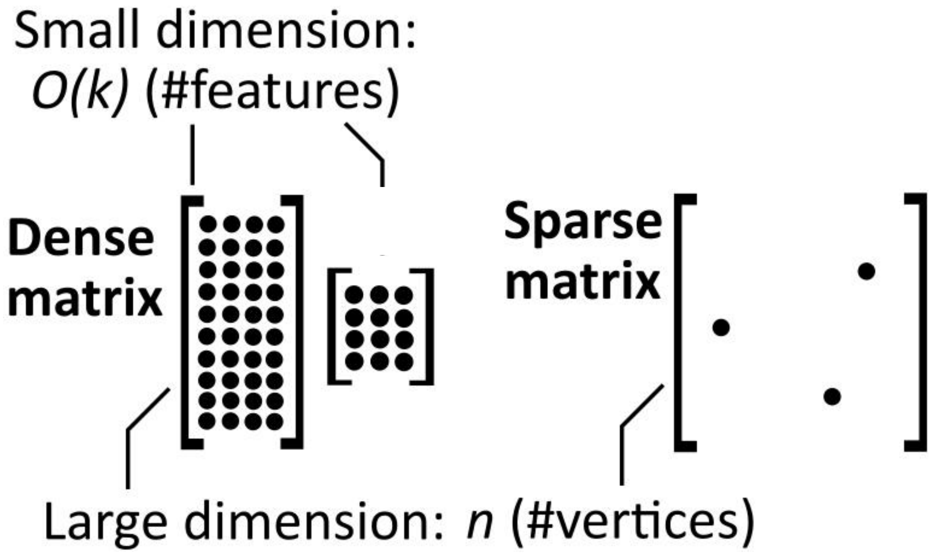


Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

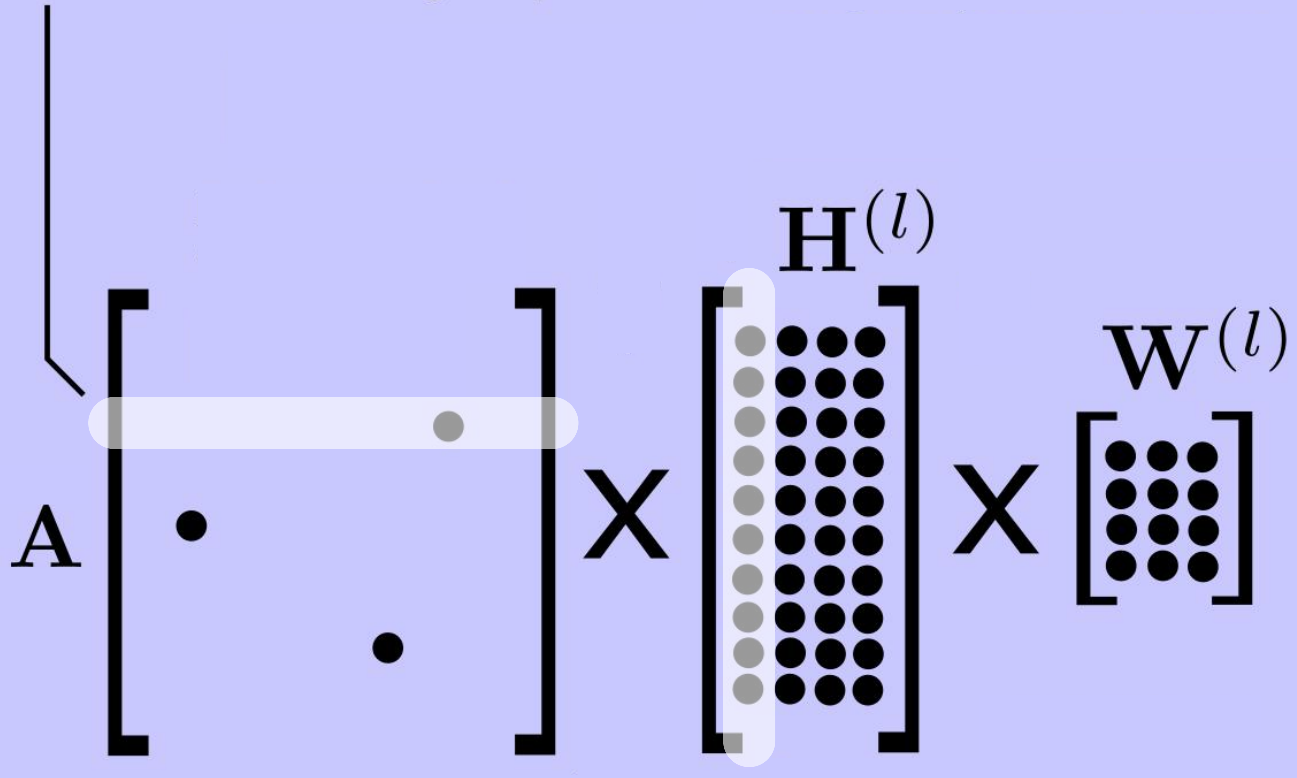
Local formulation cheatsheet:





Example model: Graph Convolution Network

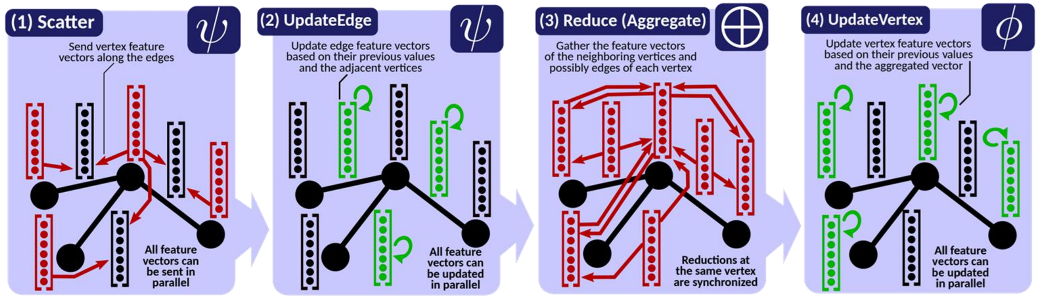
Highlighted row corresponds to the neighbors of a specific vertex *v*, whose feature vector is being computed

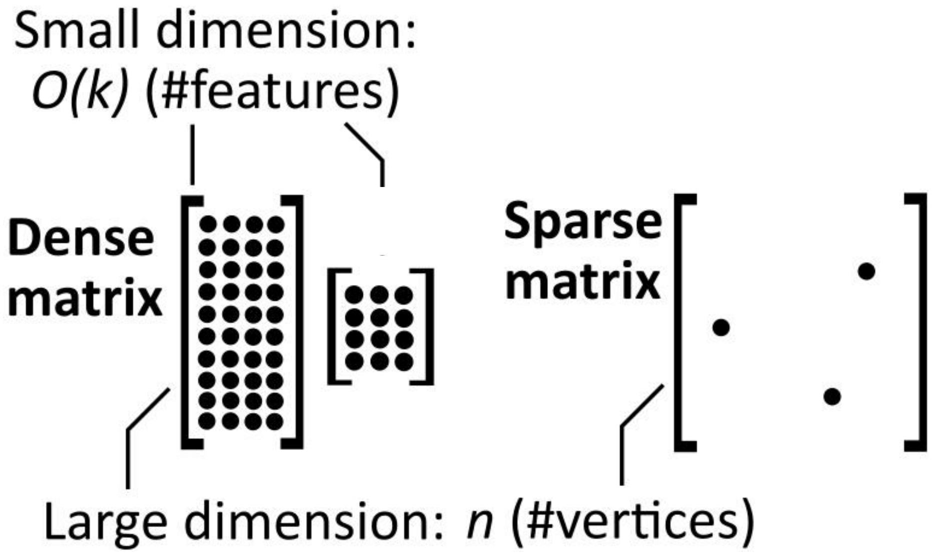


Global Formulations of GNN Models

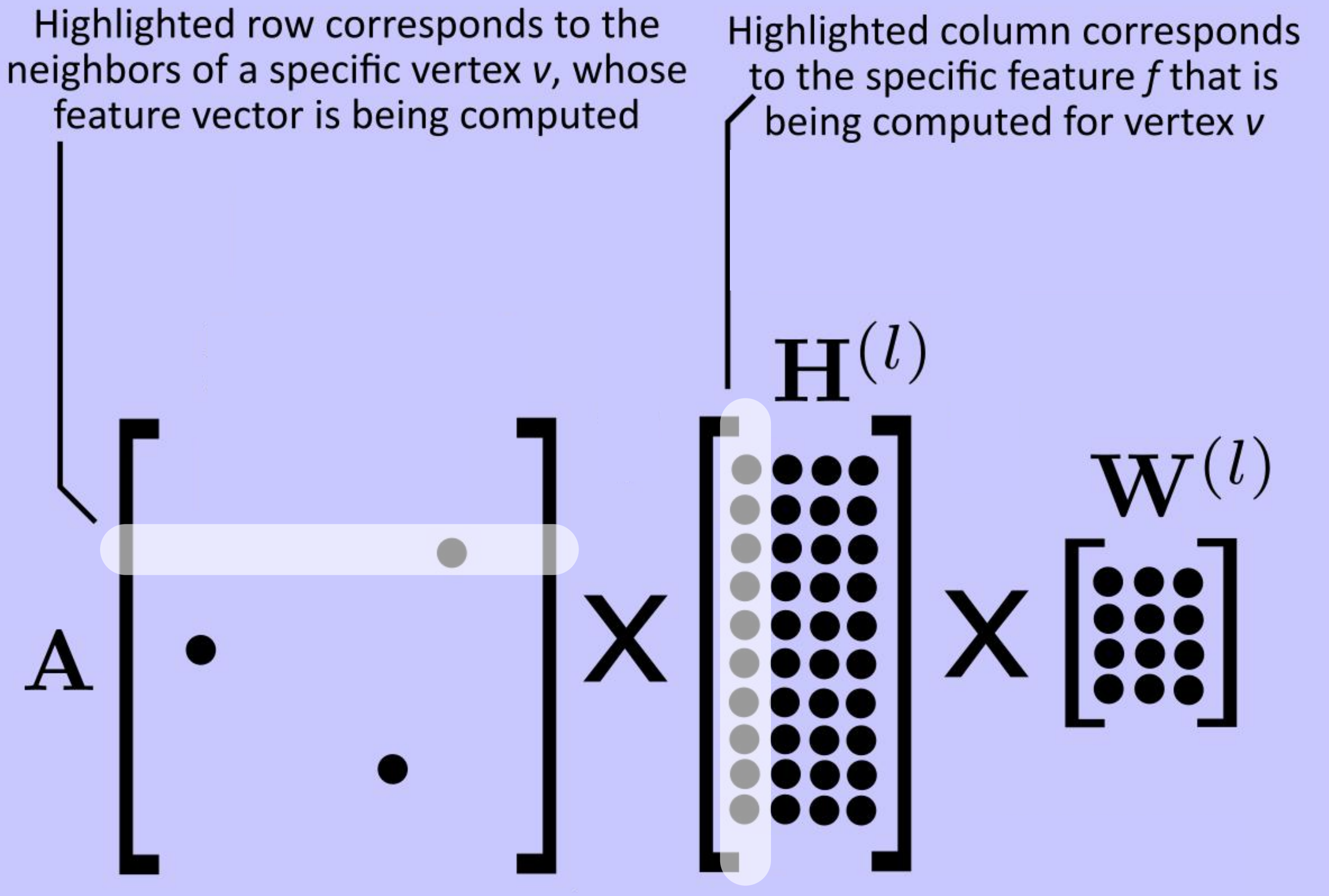
$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:





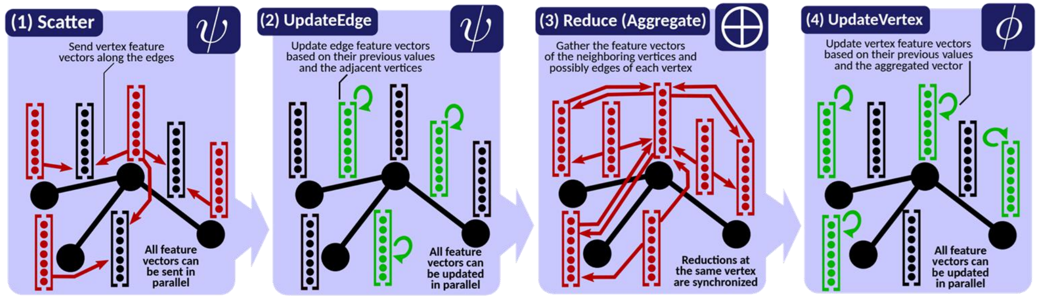
Example model: Graph Convolution Network

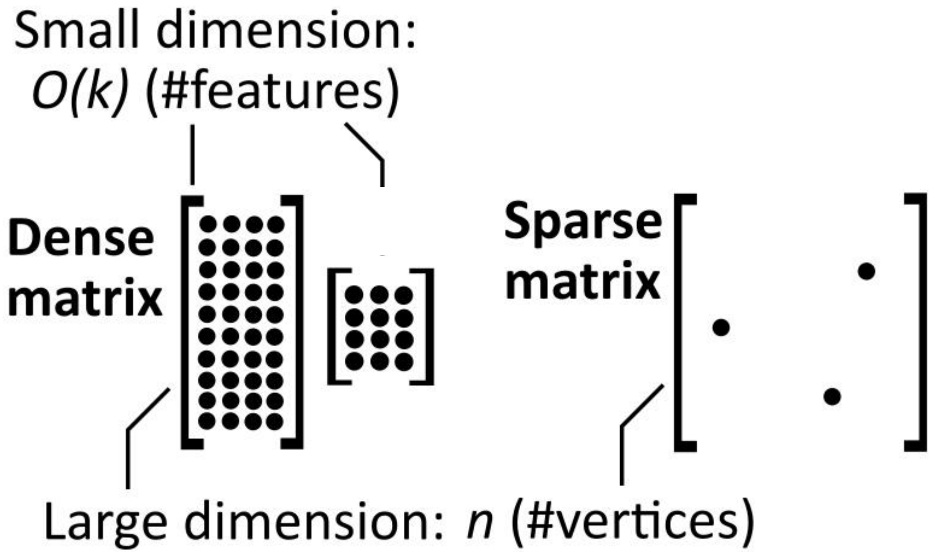


Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:

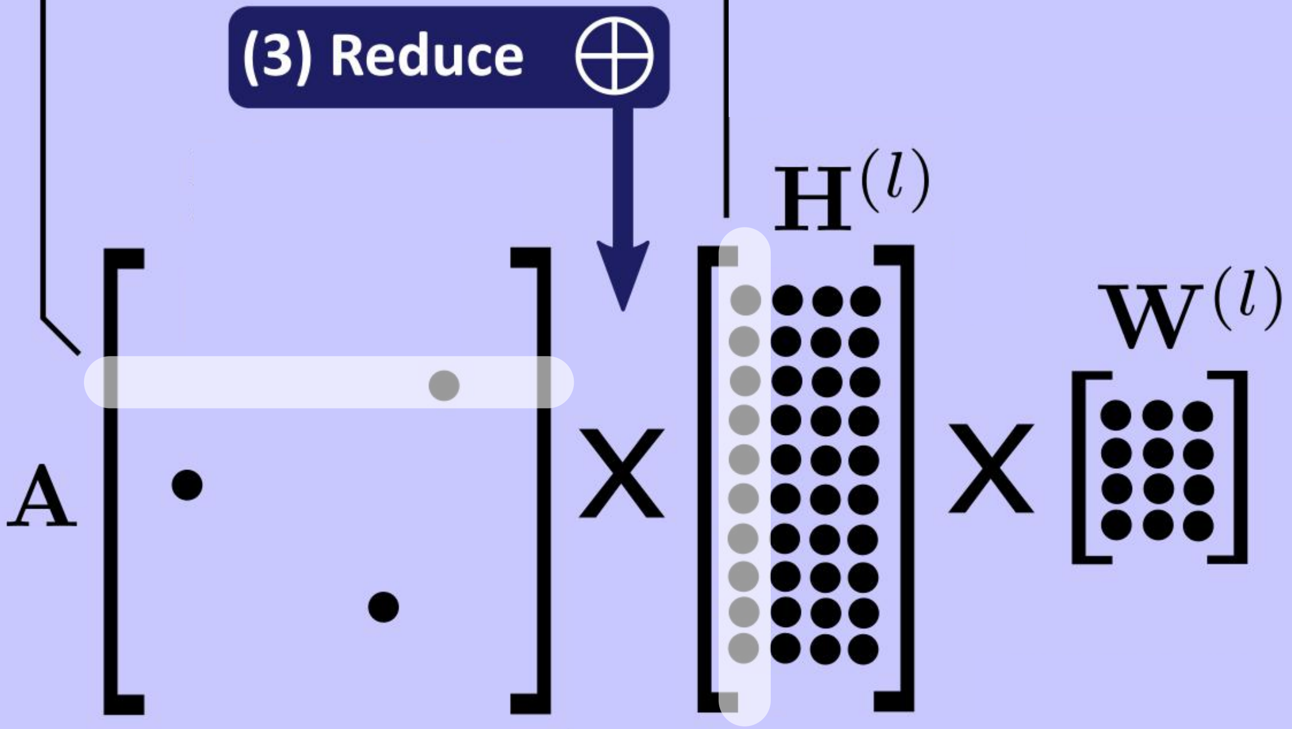




Example model: Graph Convolution Network

Highlighted row corresponds to the neighbors of a specific vertex *v*, whose feature vector is being computed

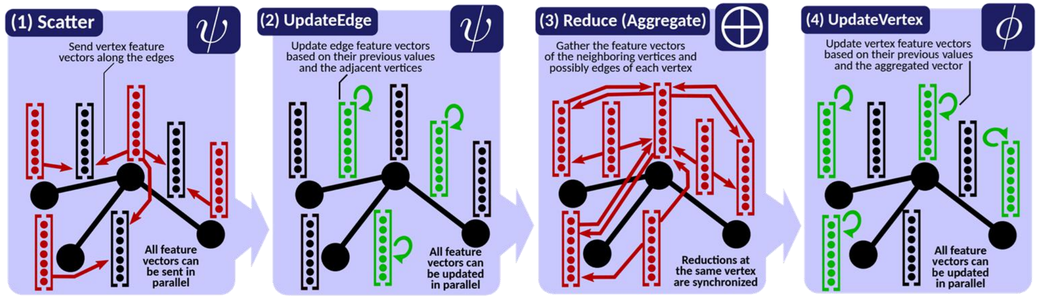
Highlighted column corresponds to the specific feature *f* that is being computed for vertex *v*

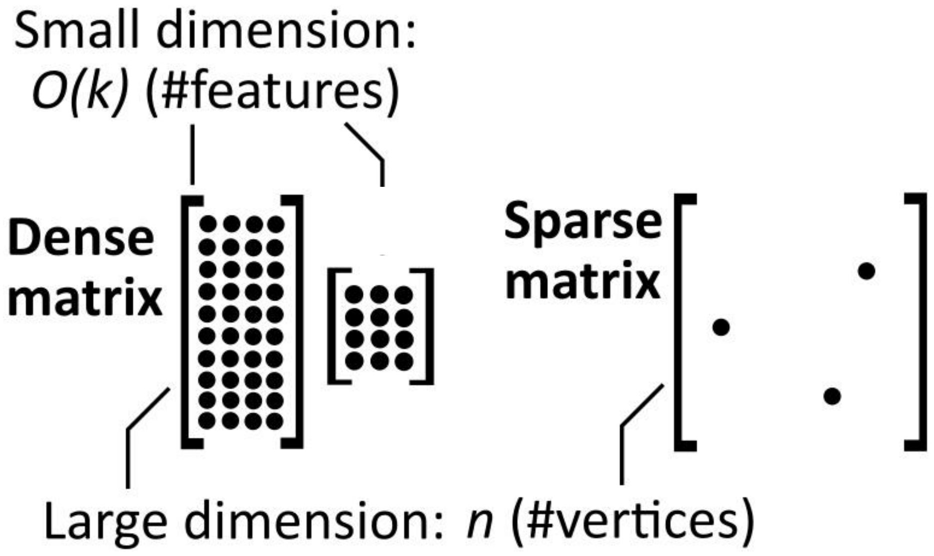


Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:

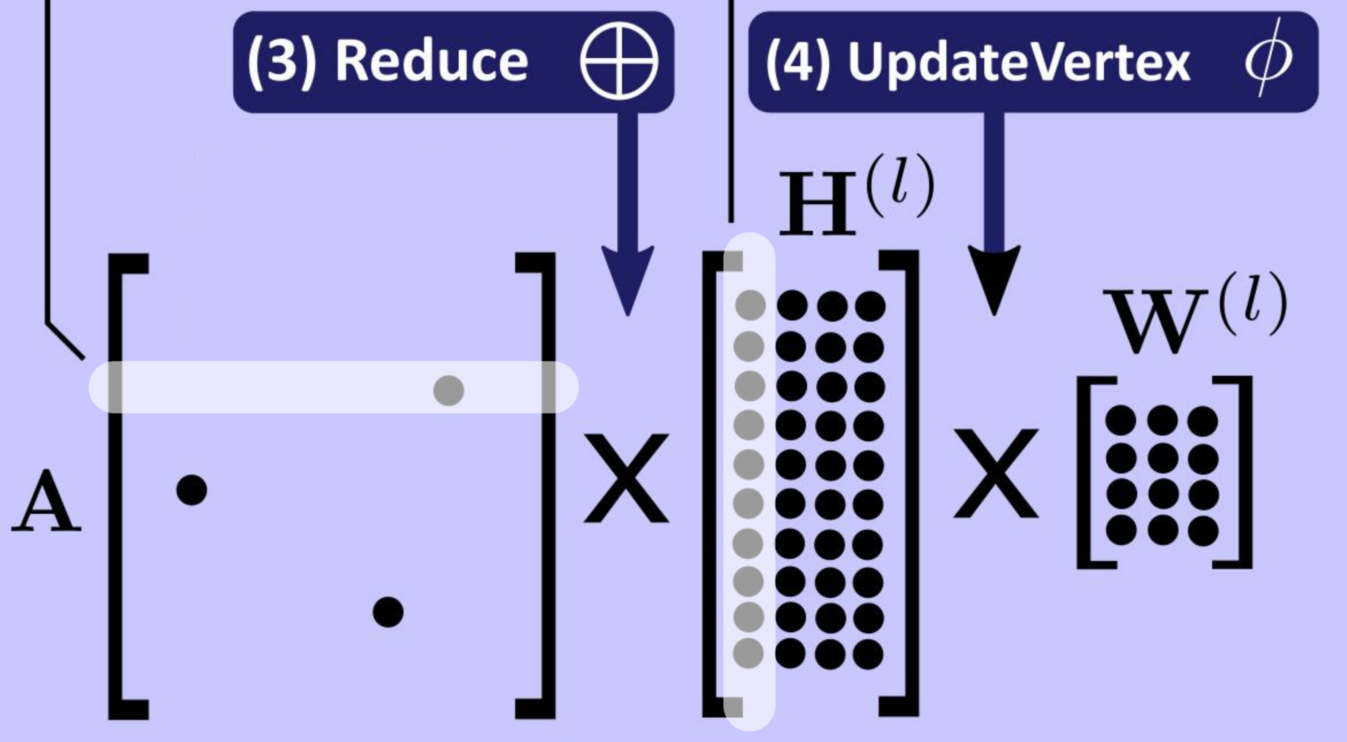




Example model: Graph Convolution Network

Highlighted row corresponds to the neighbors of a specific vertex *v*, whose feature vector is being computed

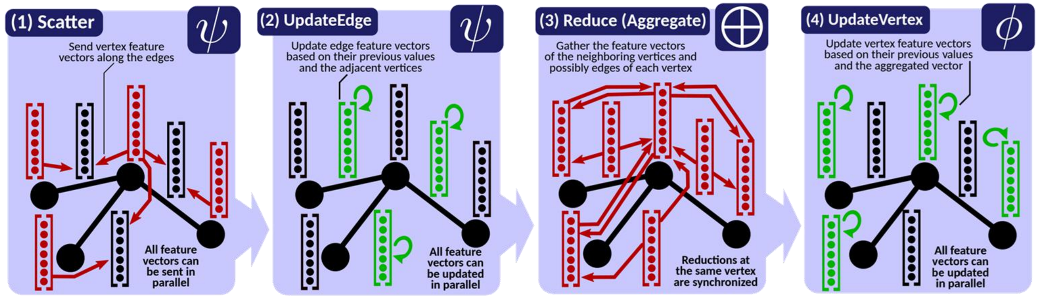
Highlighted column corresponds to the specific feature *f* that is being computed for vertex *v*

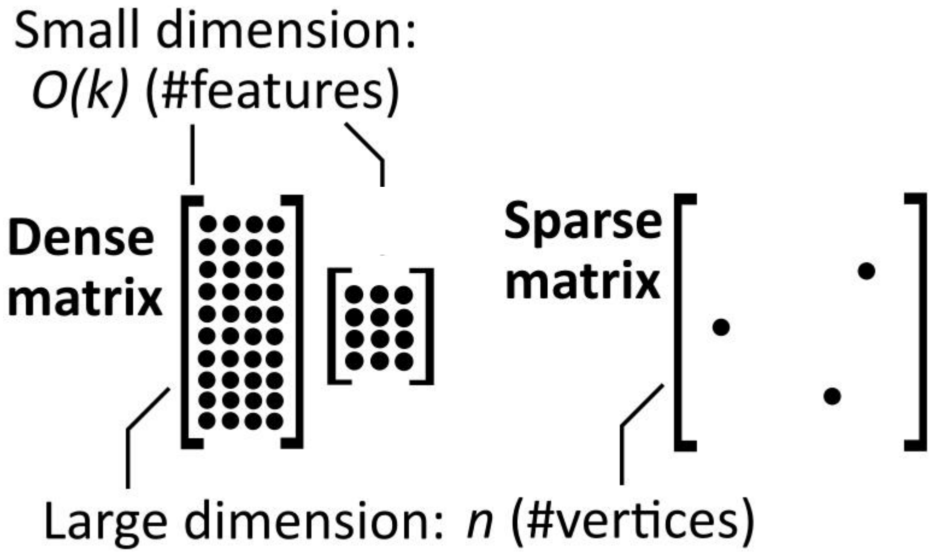


Global Formulations of GNN Models

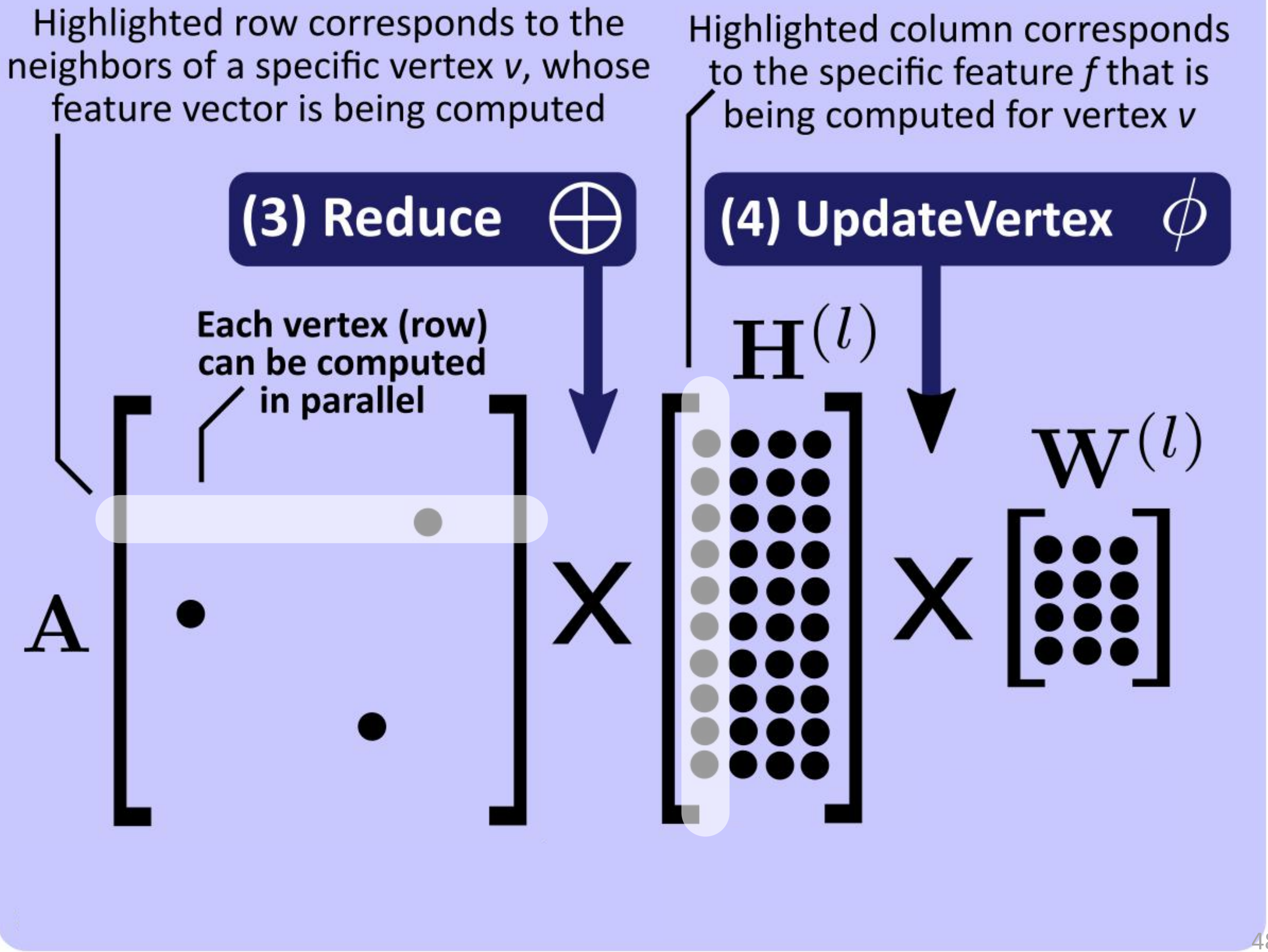
$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:





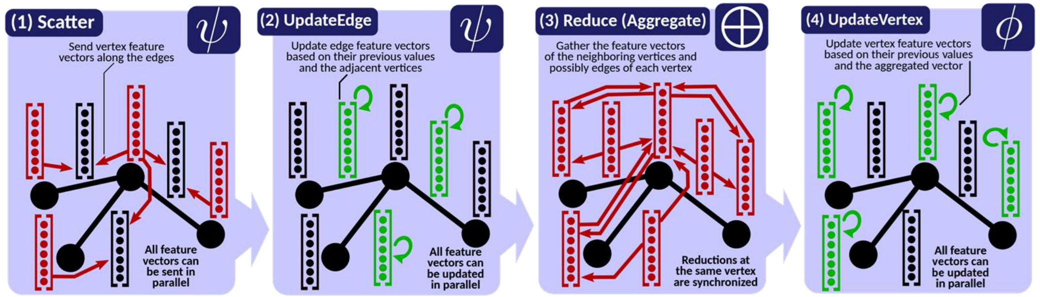
Example model: Graph Convolution Network

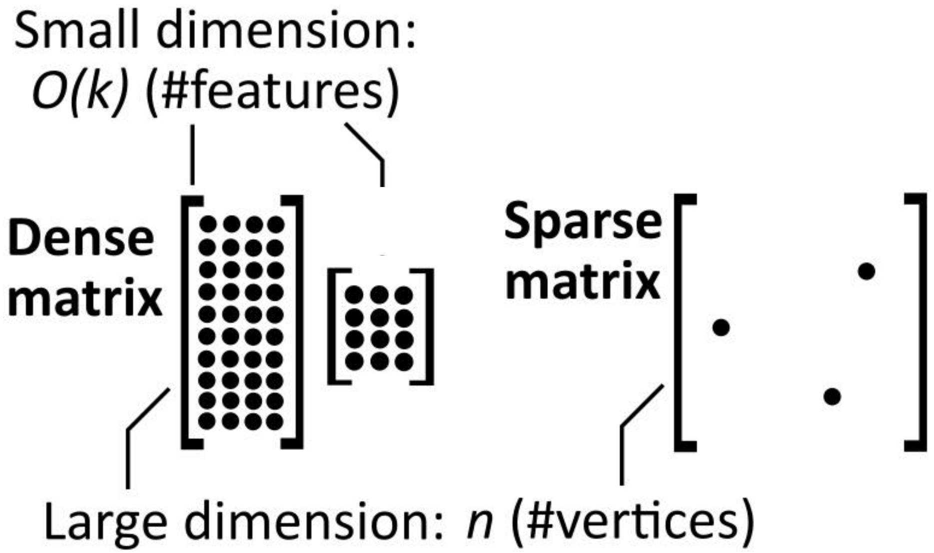


Global Formulations of GNN Models

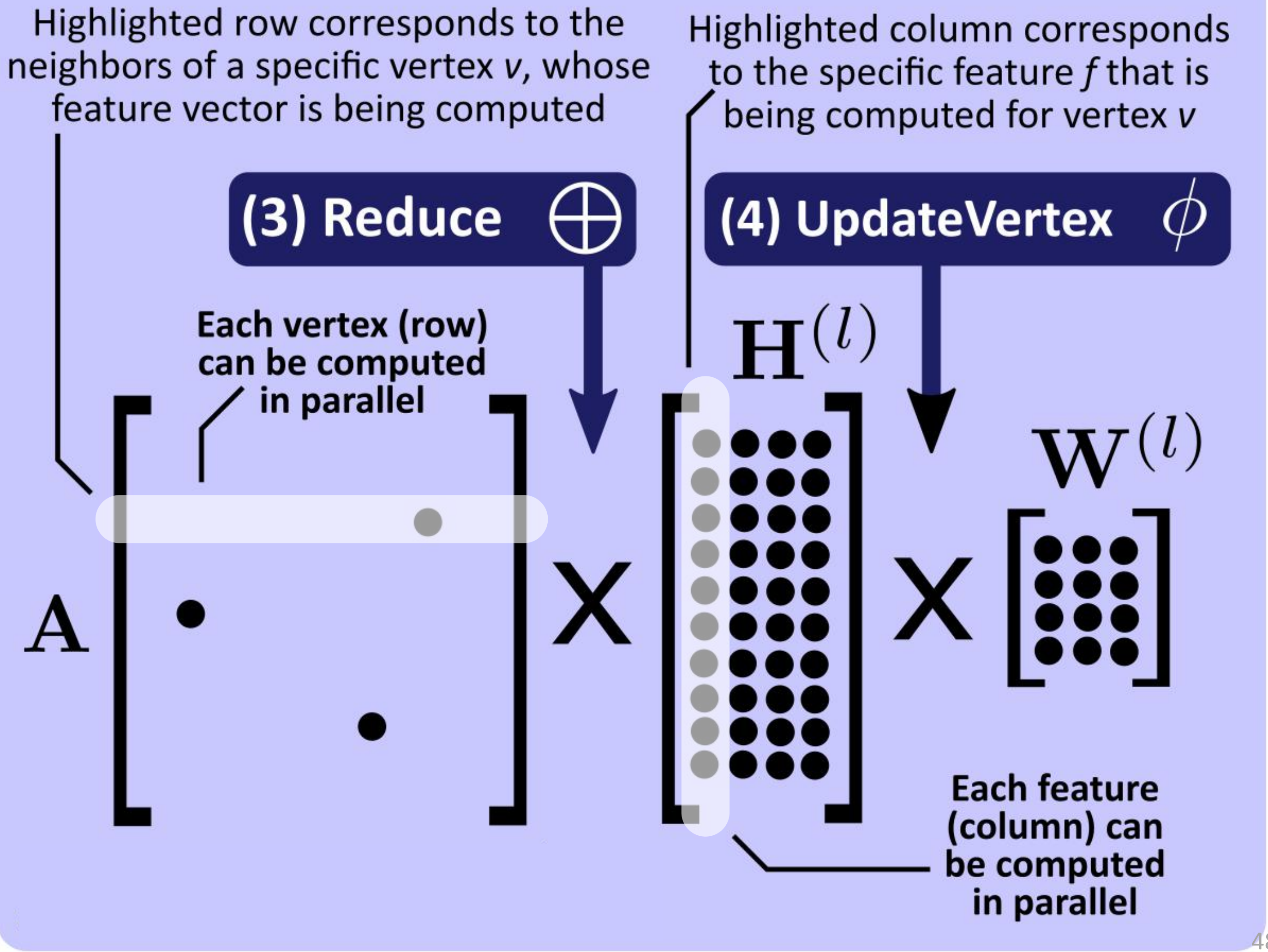
$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:





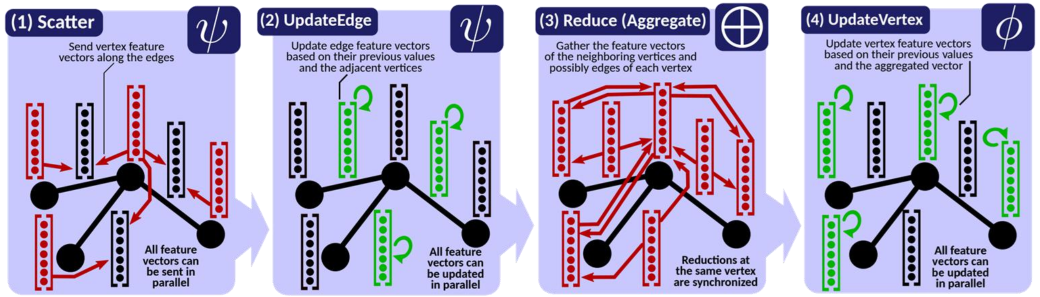
Example model: Graph Convolution Network

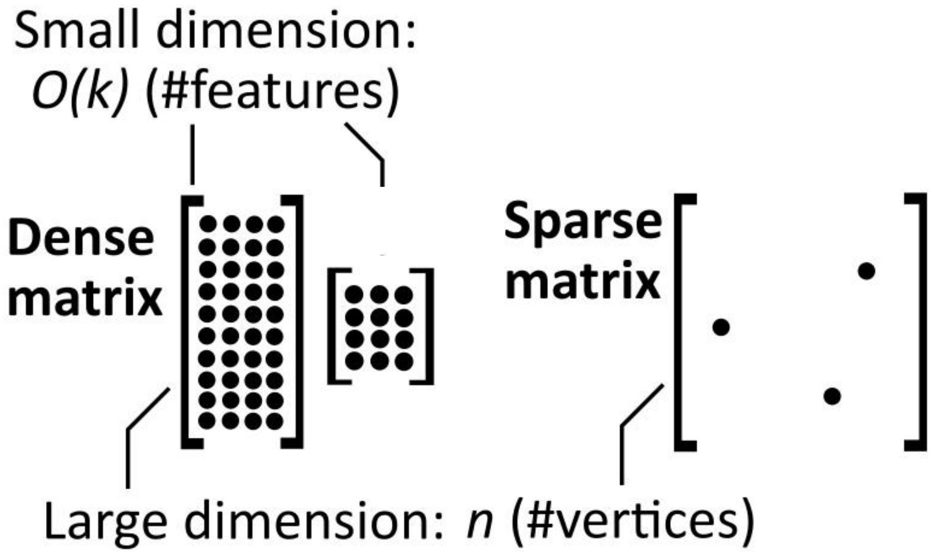


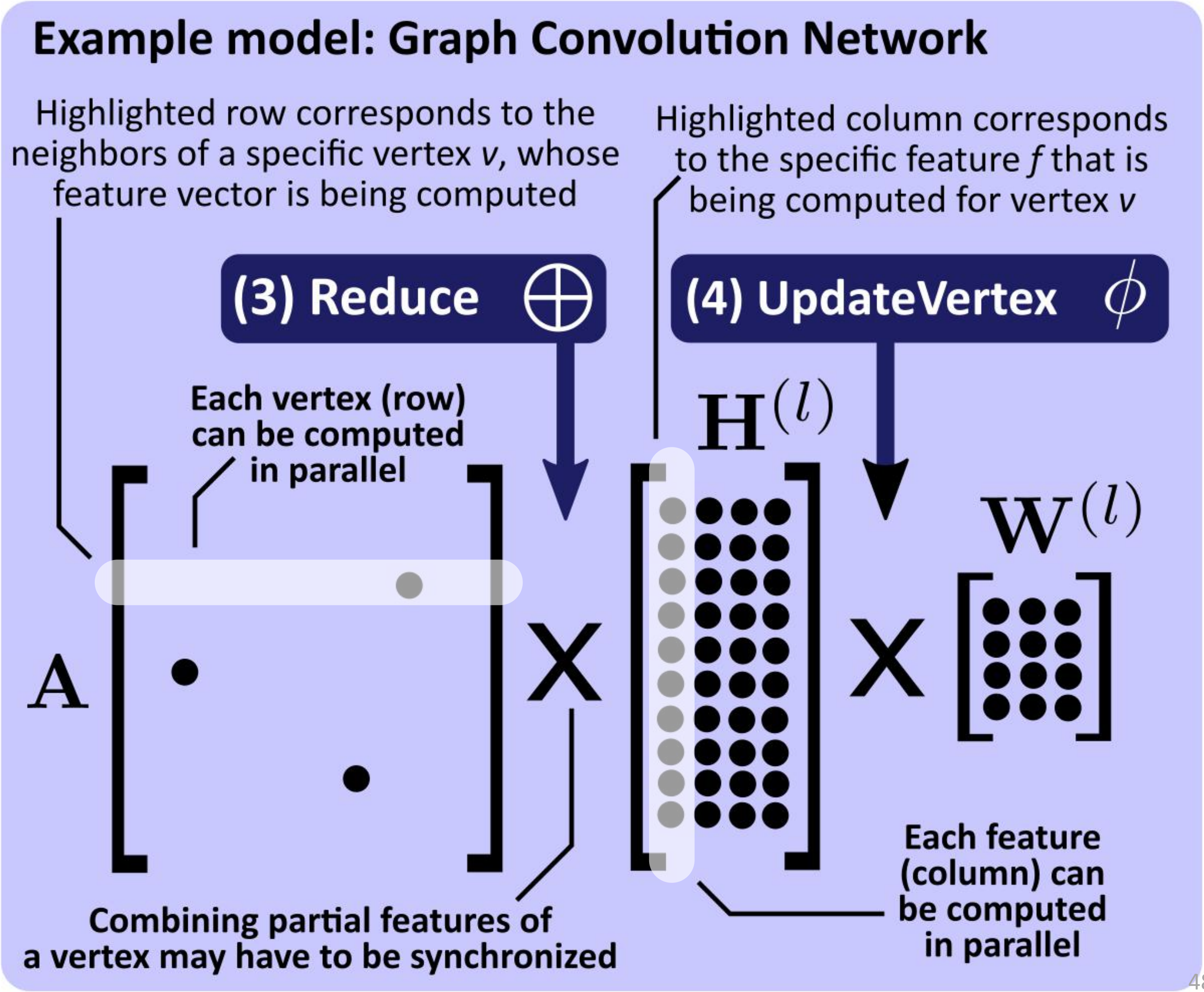
Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:



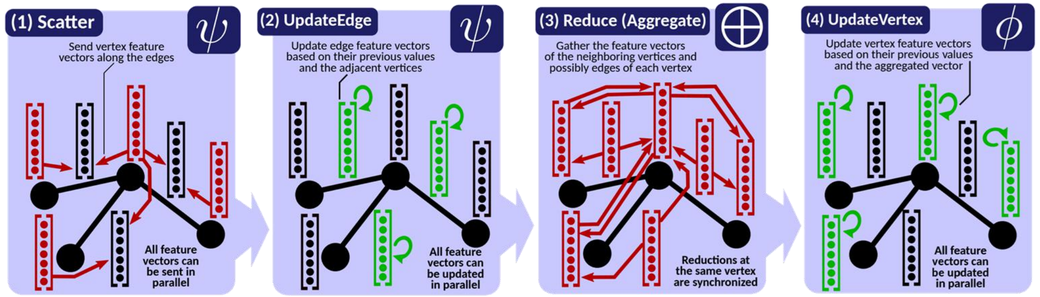


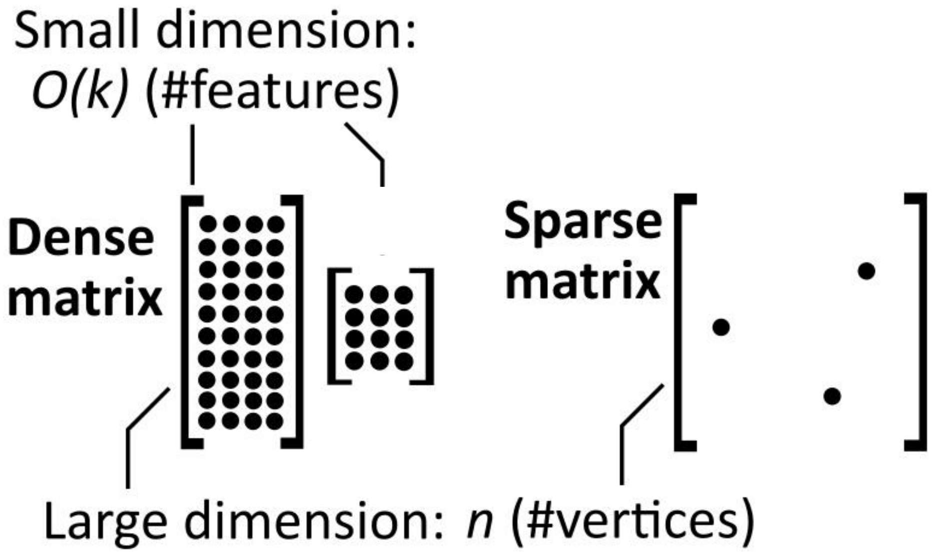


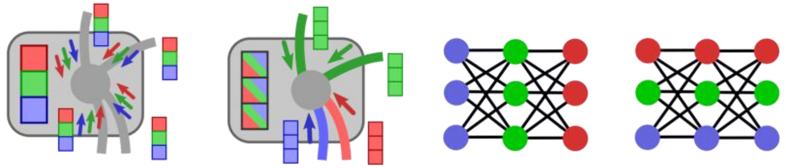
Global Formulations of GNN Models

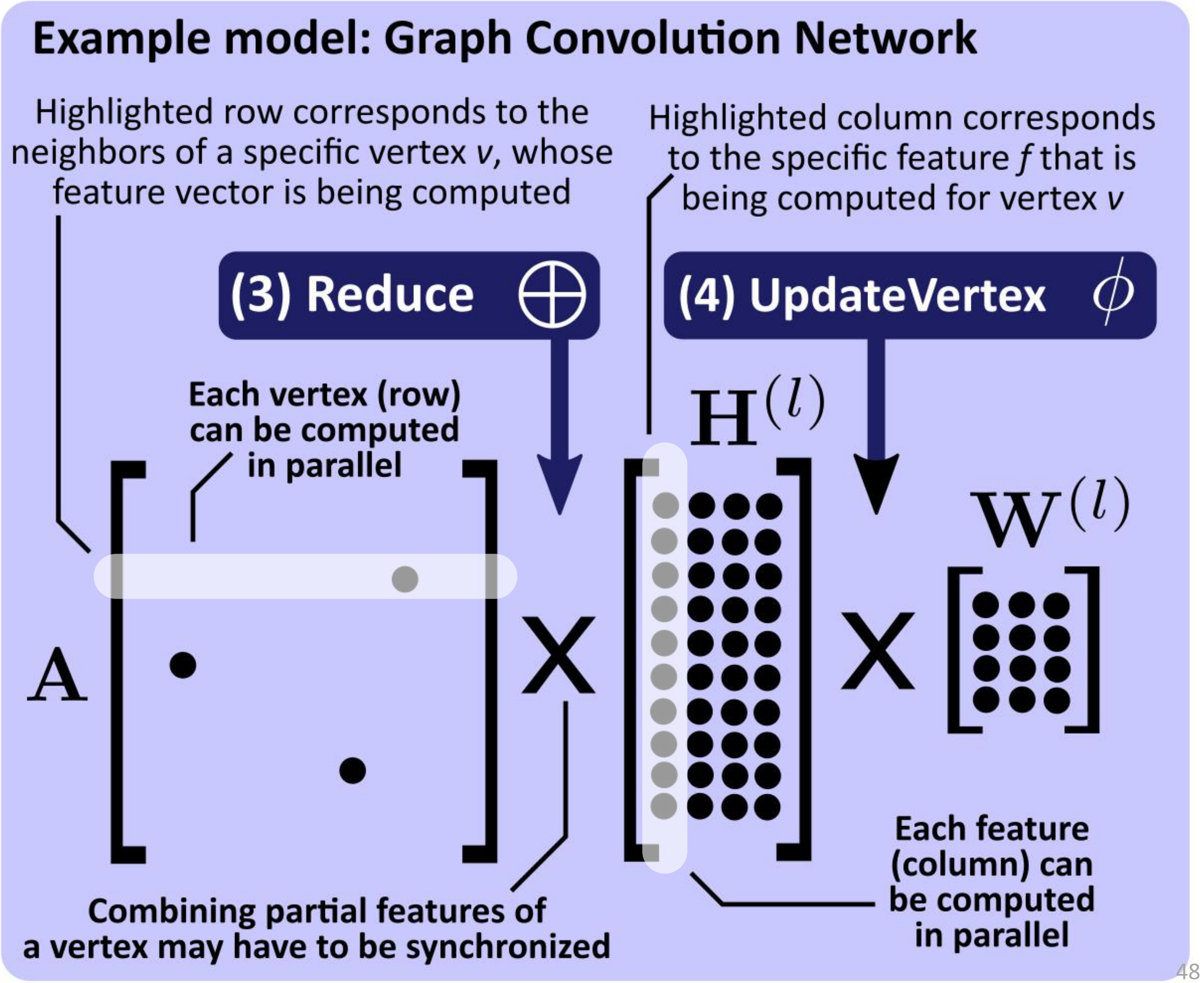
$$\mathbf{H}^{(l+1)} = \mathbf{A} \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local formulation cheatsheet:









Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \left(\mathbf{A} \odot \left(\mathbf{H}^{(l)} \times \mathbf{H}^{(l)}{}^{T}\right)\right) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

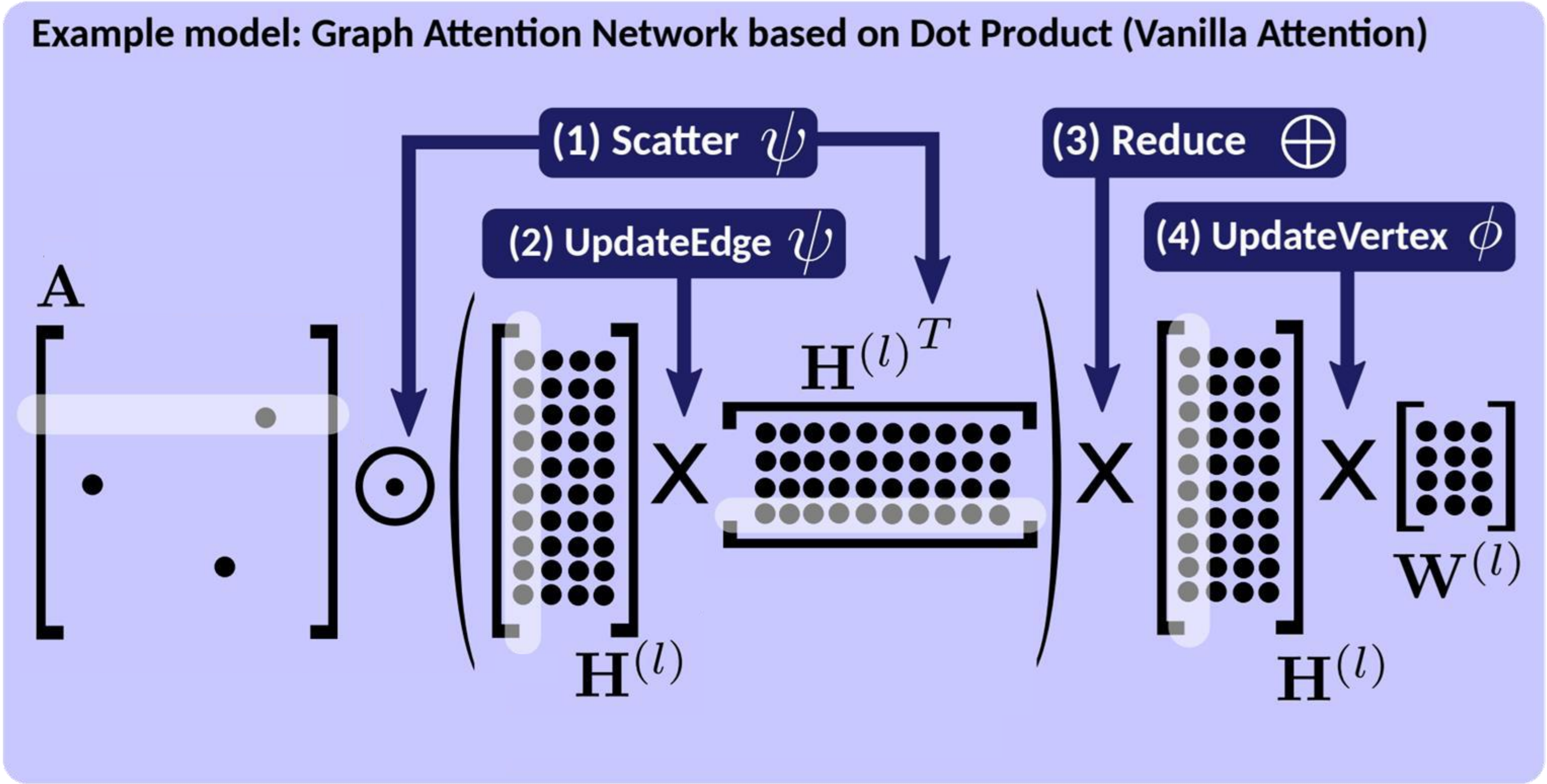
Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \left(\mathbf{A} \odot \left(\mathbf{H}^{(l)} \times \mathbf{H}^{(l)^T}\right)\right) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Example model: Graph Attention Network based on Dot Product (Vanilla Attention)

Global Formulations of GNN Models

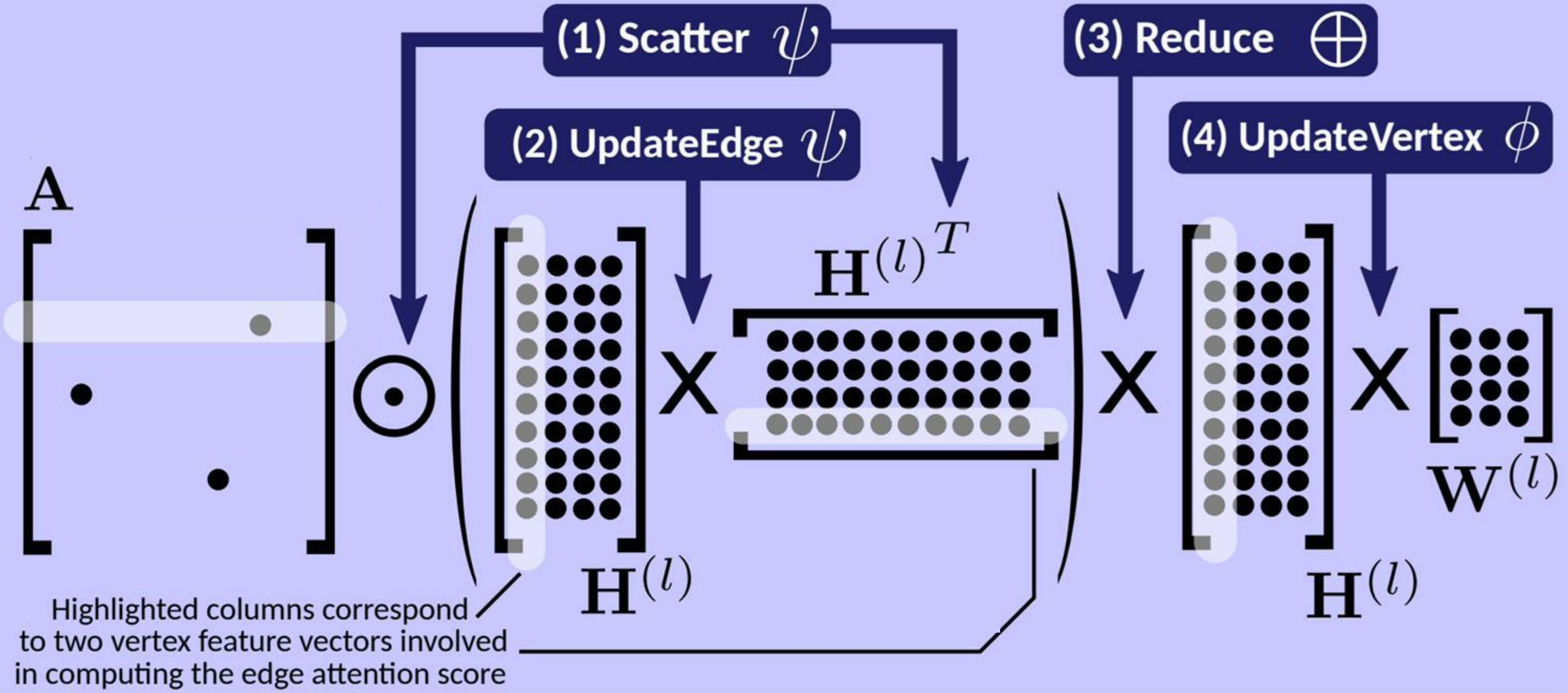
$$\mathbf{H}^{(l+1)} = \left(\mathbf{A} \odot \left(\mathbf{H}^{(l)} \times \mathbf{H}^{(l)^T}\right)\right) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$



Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \left(\mathbf{A} \odot \left(\mathbf{H}^{(l)} \times \mathbf{H}^{(l)^T}\right)\right) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Example model: Graph Attention Network based on Dot Product (Vanilla Attention)

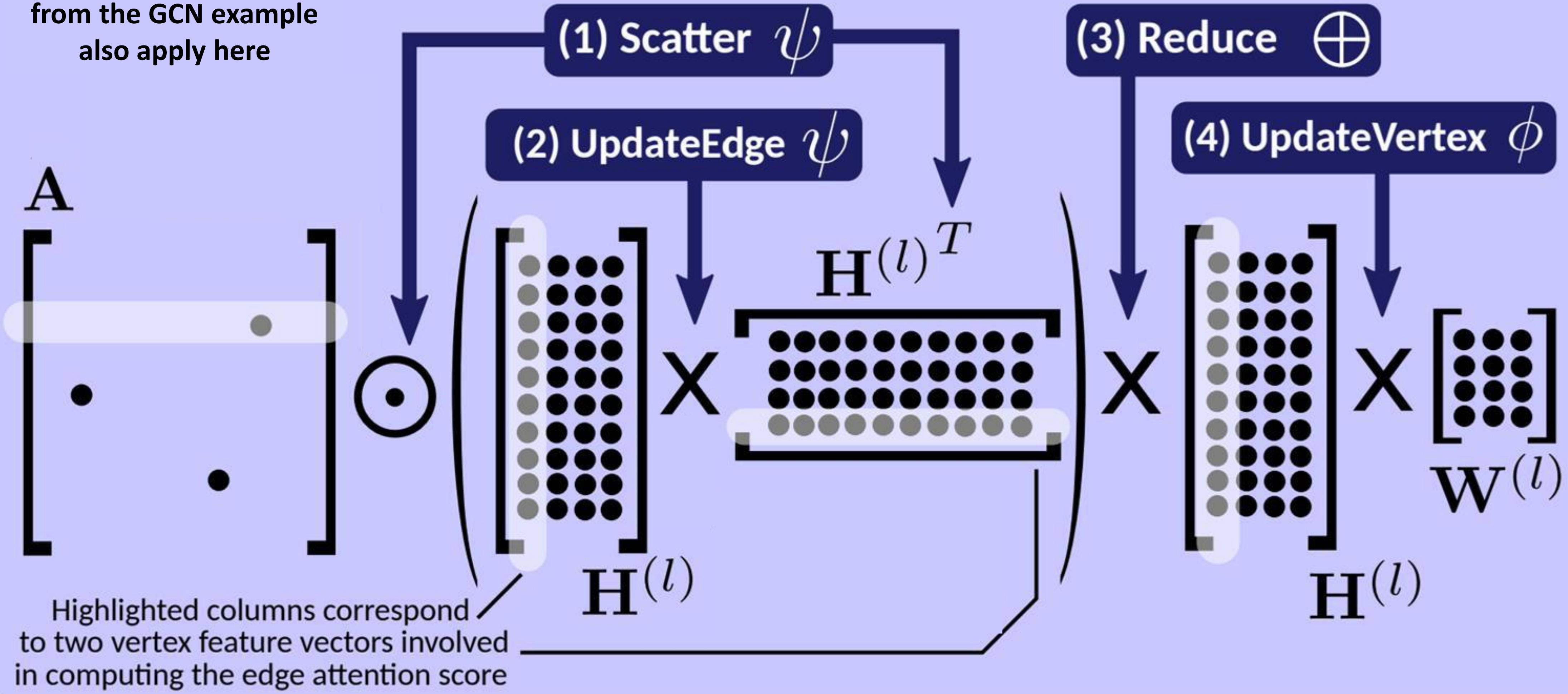


Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \left(\mathbf{A} \odot \left(\mathbf{H}^{(l)} \times \mathbf{H}^{(l)^T}\right)\right) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Example model: Graph Attention Network based on Dot Product (Vanilla Attention)

All forms of parallelism from the GCN example also apply here

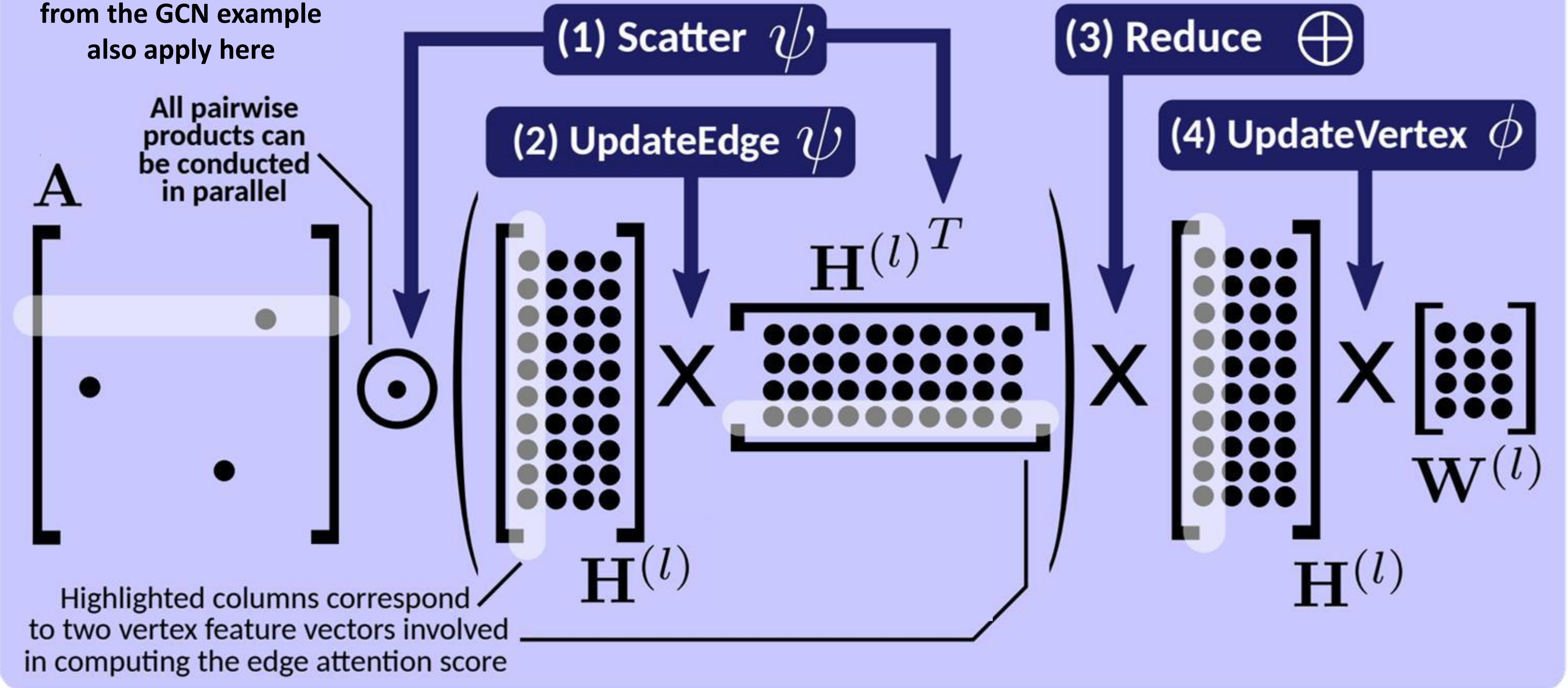


Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \left(\mathbf{A} \odot \left(\mathbf{H}^{(l)} \times \mathbf{H}^{(l)^T}\right)\right) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Example model: Graph Attention Network based on Dot Product (Vanilla Attention)

All forms of parallelism from the GCN example also apply here

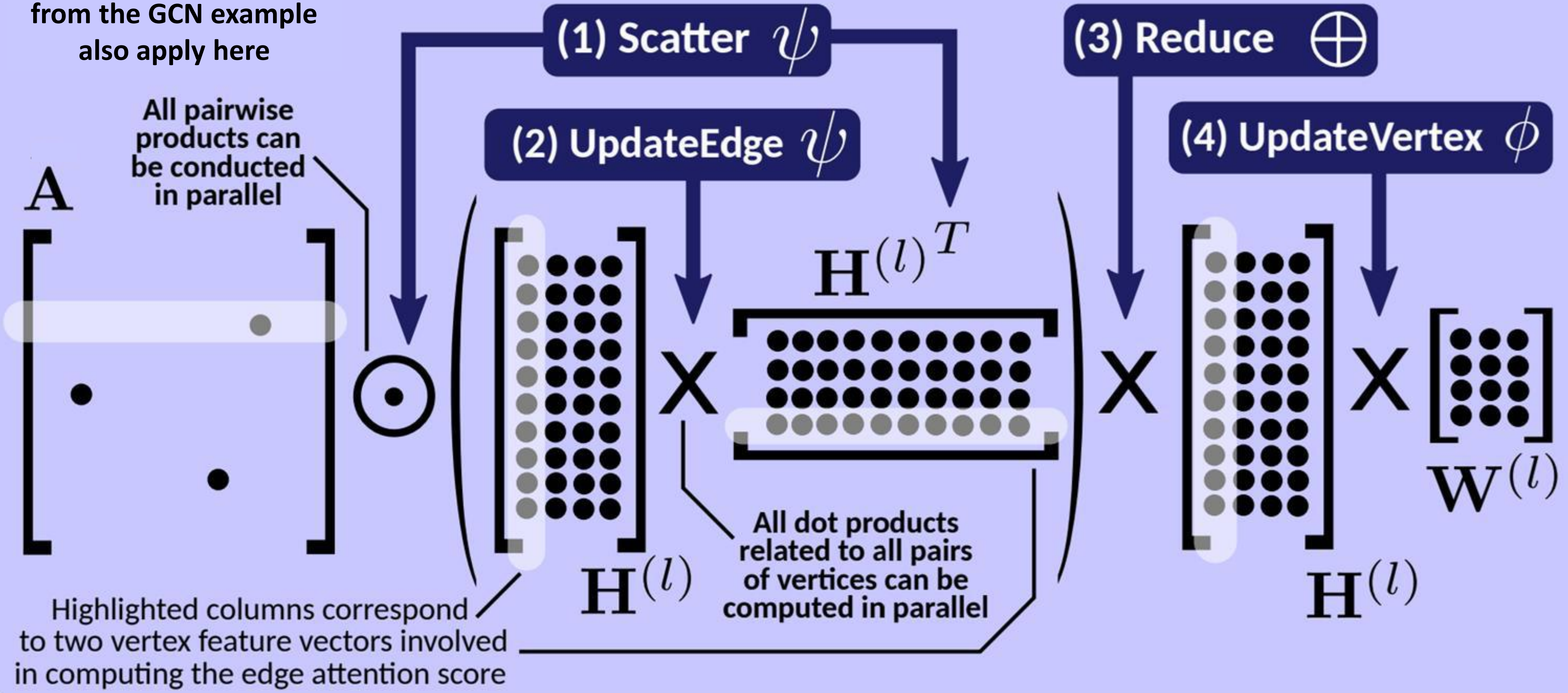


Global Formulations of GNN Models

$$\mathbf{H}^{(l+1)} = \left(\mathbf{A} \odot \left(\mathbf{H}^{(l)} \times \mathbf{H}^{(l)^T}\right)\right) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Example model: Graph Attention Network based on Dot Product (Vanilla Attention)

All forms of parallelism from the GCN example also apply here



Global Formulations: Parallel Analysis

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Reference
GCN [128]
GraphSAGE [101] (mean)
GIN [226]
CommNet [192]
Dot Product [201]
EdgeConv [216] "choice 1"
SGC [219]
DeepWalk [168]
ChebNet [72]
DCNN [6], GDC [130]
Node2Vec [97]
LINE [148], SDNE [207]
Auto-Regress [250], [256]
PPNP [43], [129], [230]
ARMA [38], ParWalks [221]

Global Formulations: Parallel Analysis

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Reference	Algebraic formulation for $\mathbf{H}^{(l+1)}$
GCN [128]	$\widehat{\mathbf{A}}\mathbf{HW}$
GraphSAGE [101] (mean)	$\widehat{\mathbf{A}}\mathbf{HW}$
GIN [226]	$\text{MLP}\left(((1+\epsilon)\mathbf{I} + \widehat{\mathbf{A}})\mathbf{H}\right)$
CommNet [192]	$\mathbf{AHW}_2 + \mathbf{HW}_1$
Dot Product [201]	$\left(\mathbf{A} \odot (\mathbf{HH}^T)\right)\mathbf{HW}$
EdgeConv [216] "choice 1"	$\mathbf{AHW}$
SGC [219]	$\widehat{\mathbf{A}}^s\mathbf{HW}$
DeepWalk [168]	$\left(\sum_{s=0}^{T} \overline{\mathbf{A}}^s\right)\mathbf{HW}$
ChebNet [72]	$\left(\sum_{s=0}^{T} \theta_s \overline{\mathbf{A}}^s\right)\mathbf{HW}$
DCNN [6], GDC [130]	$\left(\sum_{s=1}^{T} w_s \overline{\mathbf{A}}^s\right)\mathbf{HW}$
Node2Vec [97]	$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$
LINE [148], SDNE [207]	$\left(\overline{\mathbf{A}} + \theta\overline{\mathbf{A}}^2\right)\mathbf{HW}$
Auto-Regress [250], [256]	$\left((1+\alpha)\mathbf{I} - \alpha\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$
PPNP [43], [129], [230]	$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$
ARMA [38], ParWalks [221]	$b\left(\mathbf{I} - a\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$

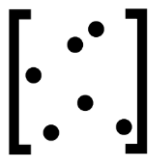
Global Formulations: Parallel Analysis

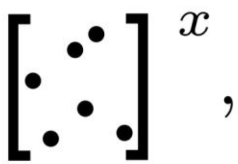
$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\dots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Reference	Type	Algebraic formulation for $\mathbf{H}^{(l+1)}$
GCN [128]	L	$\widehat{\mathbf{A}}\mathbf{H}\mathbf{W}$
GraphSAGE [101] (mean)	L	$\widehat{\mathbf{A}}\mathbf{H}\mathbf{W}$
GIN [226]	L	$\text{MLP}\left(((1+\epsilon)\mathbf{I} + \widehat{\mathbf{A}})\mathbf{H}\right)$
CommNet [192]	L	$\mathbf{A}\mathbf{H}\mathbf{W}_2 + \mathbf{H}\mathbf{W}_1$
Dot Product [201]	L	$(\mathbf{A} \odot (\mathbf{H}\mathbf{H}^T))\,\mathbf{H}\mathbf{W}$
EdgeConv [216] "choice 1"	L	$\mathbf{A}\mathbf{H}\mathbf{W}$
SGC [219]	P	$\widehat{\mathbf{A}}^s\mathbf{H}\mathbf{W}$
DeepWalk [168]	P	$\left(\sum_{s=0}^{T} \overline{\mathbf{A}}^s\right)\mathbf{H}\mathbf{W}$
ChebNet [72]	P	$\left(\sum_{s=0}^{T} \theta_s \overline{\mathbf{A}}^s\right)\mathbf{H}\mathbf{W}$
DCNN [6], GDC [130]	P	$\left(\sum_{s=1}^{T} w_s \overline{\mathbf{A}}^s\right)\mathbf{H}\mathbf{W}$
Node2Vec [97]	P	$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{H}\mathbf{W}$
LINE [148], SDNE [207]	P	$\left(\overline{\mathbf{A}} + \theta\overline{\mathbf{A}}^2\right)\mathbf{H}\mathbf{W}$
Auto-Regress [250], [256]	R	$\left((1+\alpha)\mathbf{I} - \alpha\widehat{\mathbf{A}}\right)^{-1}\mathbf{H}\mathbf{W}$
PPNP [43], [129], [230]	R	$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{H}\mathbf{W}$
ARMA [38], ParWalks [221]	R	$b\left(\mathbf{I} - a\widehat{\mathbf{A}}\right)^{-1}\mathbf{H}\mathbf{W}$

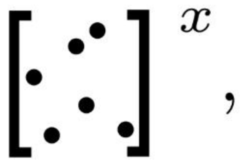
Global Formulations: Parallel Analysis

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\dots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$





$x \in \mathbb{N}$

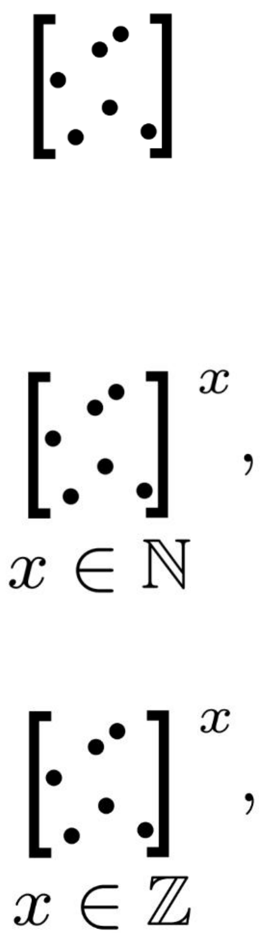


$x \in \mathbb{Z}$

Reference	Type	Algebraic formulation for $\mathbf{H}^{(l+1)}$
GCN [128]	L	$\widehat{\mathbf{A}}\mathbf{H}\mathbf{W}$
GraphSAGE [101] (mean)	L	$\widehat{\mathbf{A}}\mathbf{H}\mathbf{W}$
GIN [226]	L	$\text{MLP}\left(((1+\epsilon)\mathbf{I} + \widehat{\mathbf{A}})\mathbf{H}\right)$
CommNet [192]	L	$\mathbf{A}\mathbf{H}\mathbf{W}_2 + \mathbf{H}\mathbf{W}_1$
Dot Product [201]	L	$(\mathbf{A} \odot (\mathbf{H}\mathbf{H}^T))\,\mathbf{H}\mathbf{W}$
EdgeConv [216] "choice 1"	L	$\mathbf{A}\mathbf{H}\mathbf{W}$
SGC [219]	P	$\widehat{\mathbf{A}}^s\mathbf{H}\mathbf{W}$
DeepWalk [168]	P	$\left(\sum_{s=0}^{T} \overline{\mathbf{A}}^s\right)\mathbf{H}\mathbf{W}$
ChebNet [72]	P	$\left(\sum_{s=0}^{T} \theta_s \overline{\mathbf{A}}^s\right)\mathbf{H}\mathbf{W}$
DCNN [6], GDC [130]	P	$\left(\sum_{s=1}^{T} w_s \overline{\mathbf{A}}^s\right)\mathbf{H}\mathbf{W}$
Node2Vec [97]	P	$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{H}\mathbf{W}$
LINE [148], SDNE [207]	P	$\left(\overline{\mathbf{A}} + \theta\overline{\mathbf{A}}^2\right)\mathbf{H}\mathbf{W}$
Auto-Regress [250], [256]	R	$\left((1+\alpha)\mathbf{I} - \alpha\widehat{\mathbf{A}}\right)^{-1}\mathbf{H}\mathbf{W}$
PPNP [43], [129], [230]	R	$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{H}\mathbf{W}$
ARMA [38], ParWalks [221]	R	$b\left(\mathbf{I} - a\widehat{\mathbf{A}}\right)^{-1}\mathbf{H}\mathbf{W}$

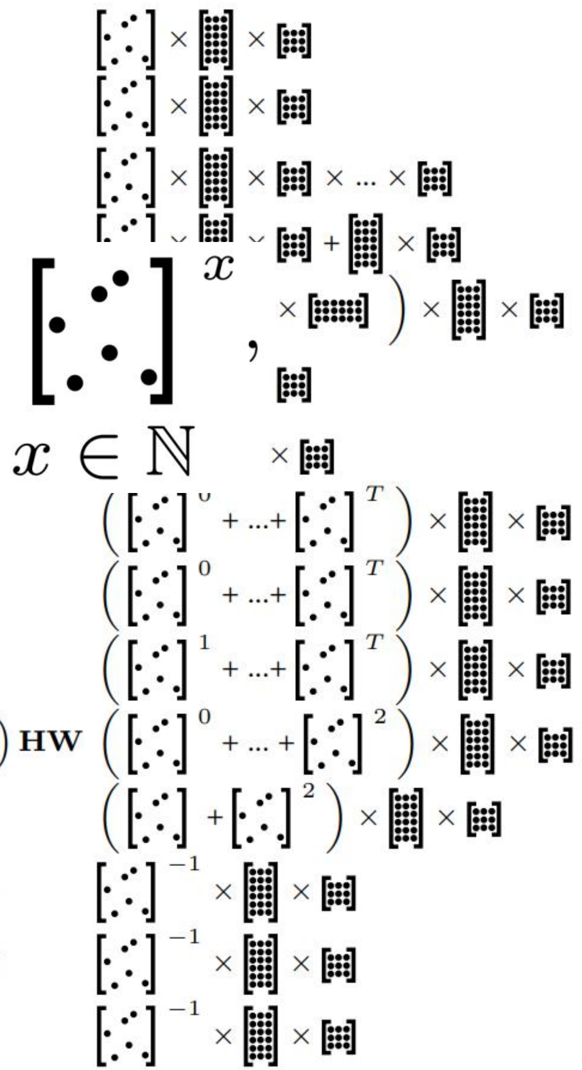
Global Formulations: Parallel Analysis

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$



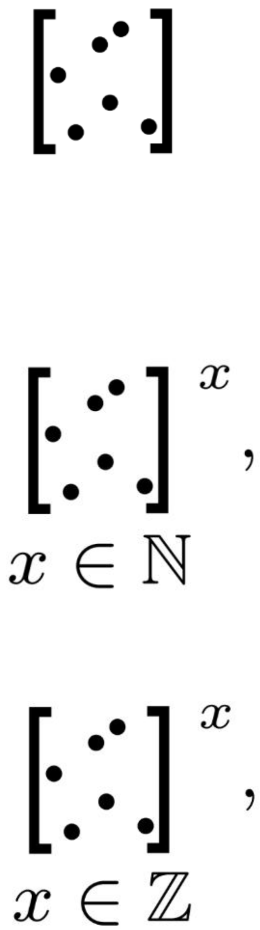
Reference	Type	Algebraic formulation for $\mathbf{H}^{(l+1)}$
GCN [128]	L	$\widehat{\mathbf{A}}\mathbf{HW}$
GraphSAGE [101] (mean)	L	$\widehat{\mathbf{A}}\mathbf{HW}$
GIN [226]	L	MLP $\left(((1+\epsilon)\mathbf{I}+\widehat{\mathbf{A}})\mathbf{H}\right)$
CommNet [192]	L	$\mathbf{AHW}_2 + \mathbf{HW}_1$
Dot Product [201]	L	$\left(\mathbf{A} \odot (\mathbf{HH}^T)\right)\mathbf{HW}$
EdgeConv [216] "choice 1"	L	$\mathbf{AHW}$
SGC [219]	P	$\widehat{\mathbf{A}}^s\mathbf{HW}$
DeepWalk [168]	P	$\left(\sum_{s=0}^{T} \overline{\mathbf{A}}^s\right)\mathbf{HW}$
ChebNet [72]	P	$\left(\sum_{s=0}^{T} \theta_s \overline{\mathbf{A}}^s\right)\mathbf{HW}$
DCNN [6], GDC [130]	P	$\left(\sum_{s=1}^{T} w_s \overline{\mathbf{A}}^s\right)\mathbf{HW}$
Node2Vec [97]	P	$\left(\frac{1}{p}\mathbf{I} + \left(1-\frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$
LINE [148], SDNE [207]	P	$\left(\overline{\mathbf{A}} + \theta\overline{\mathbf{A}}^2\right)\mathbf{HW}$
Auto-Regress [250], [256]	R	$\left((1+\alpha)\mathbf{I} - \alpha\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$
PPNP [43], [129], [230]	R	$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$
ARMA [38], ParWalks [221]	R	$b\left(\mathbf{I} - a\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$

Dimensions & density of deriving $\mathbf{H}^{(l+1)}$

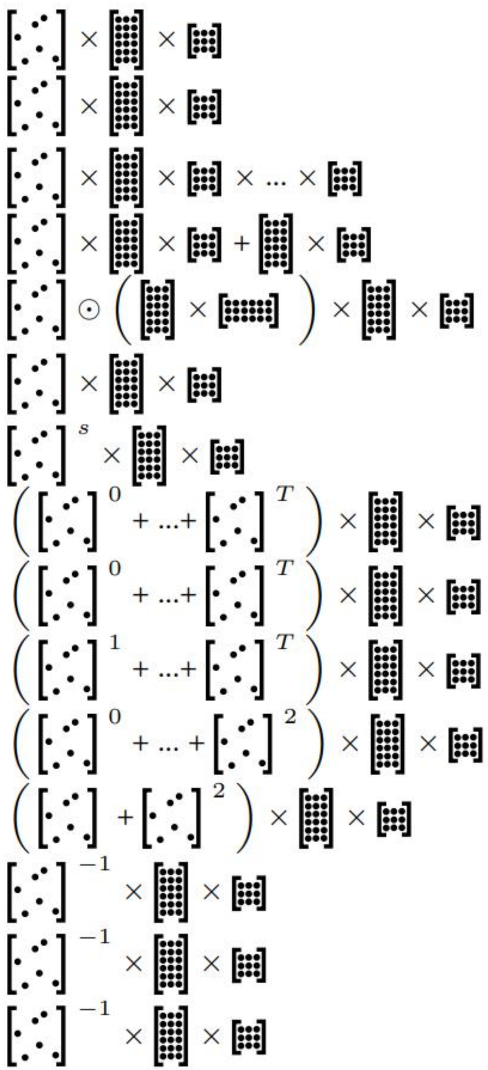


Global Formulations: Parallel Analysis

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

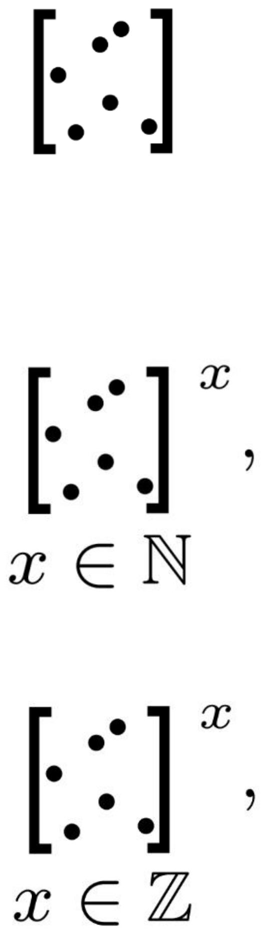


Reference	Type	Algebraic formulation for $\mathbf{H}^{(l+1)}$	Dimensions & density of deriving $\mathbf{H}^{(l+1)}$	Work & depth (one whole training iteration or inference)	
GCN [128]	L	$\widehat{\mathbf{A}}\mathbf{H}\mathbf{W}$		$O(mkL + Lnk^2)$	$O(L\log k + L\log d)$
GraphSAGE [101] (mean)	L	$\widehat{\mathbf{A}}\mathbf{H}\mathbf{W}$		$O(mkL + Lnk^2)$	$O(L\log k + L\log d)$
GIN [226]	L	$\text{MLP}\left(((1+\epsilon)\mathbf{I} + \widehat{\mathbf{A}})\mathbf{H}\right)$		$O(mkL + KLnk^2)$	$O(LK\log k + LK\log d)$
CommNet [192]	L	$\mathbf{A}\mathbf{H}\mathbf{W}_2 + \mathbf{H}\mathbf{W}_1$		$O(mkL + Lnk^2)$	$O(L\log k + L\log d)$
Dot Product [201]	L	$(\mathbf{A} \odot (\mathbf{H}\mathbf{H}^T))\,\mathbf{H}\mathbf{W}$		$O(Lmk + Lnk^2)$	$O(L\log k + L\log d)$
EdgeConv [216] "choice 1"	L	$\mathbf{A}\mathbf{H}\mathbf{W}$		$O(mkL + Lnk^2)$	$O(L\log k + L\log d)$
SGC [219]	P	$\widehat{\mathbf{A}}^s\mathbf{H}\mathbf{W}$		$O(mn\log s + nk^2)$	$O(\log k + \log s\log d)$
DeepWalk [168]	P	$\left(\sum_{s=0}^{T}\overline{\mathbf{A}}^s\right)\mathbf{H}\mathbf{W}$		$O(mn\log T + nk^2)$	$O(\log k + \log T\log d)$
ChebNet [72]	P	$\left(\sum_{s=0}^{T}\theta_s\overline{\mathbf{A}}^s\right)\mathbf{H}\mathbf{W}$		$O(mn\log T + nk^2)$	$O(\log k + \log T\log d)$
DCNN [6], GDC [130]	P	$\left(\sum_{s=1}^{T}w_s\overline{\mathbf{A}}^s\right)\mathbf{H}\mathbf{W}$		$O(mn\log T + nk^2)$	$O(\log k + \log T\log d)$
Node2Vec [97]	P	$\left(\frac{1}{p}\mathbf{I} + \left(1-\frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{H}\mathbf{W}$		$O(mn + nk^2)$	$O(\log k + \log d)$
LINE [148], SDNE [207]	P	$\left(\overline{\mathbf{A}} + \theta\overline{\mathbf{A}}^2\right)\mathbf{H}\mathbf{W}$		$O(mn + nk^2)$	$O(\log k + \log d)$
Auto-Regress [250], [256]	R	$\left((1+\alpha)\mathbf{I} - \alpha\widehat{\mathbf{A}}\right)^{-1}\mathbf{H}\mathbf{W}$		$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
PPNP [43], [129], [230]	R	$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{H}\mathbf{W}$		$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
ARMA [38], ParWalks [221]	R	$b\left(\mathbf{I} - a\widehat{\mathbf{A}}\right)^{-1}\mathbf{H}\mathbf{W}$		$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$

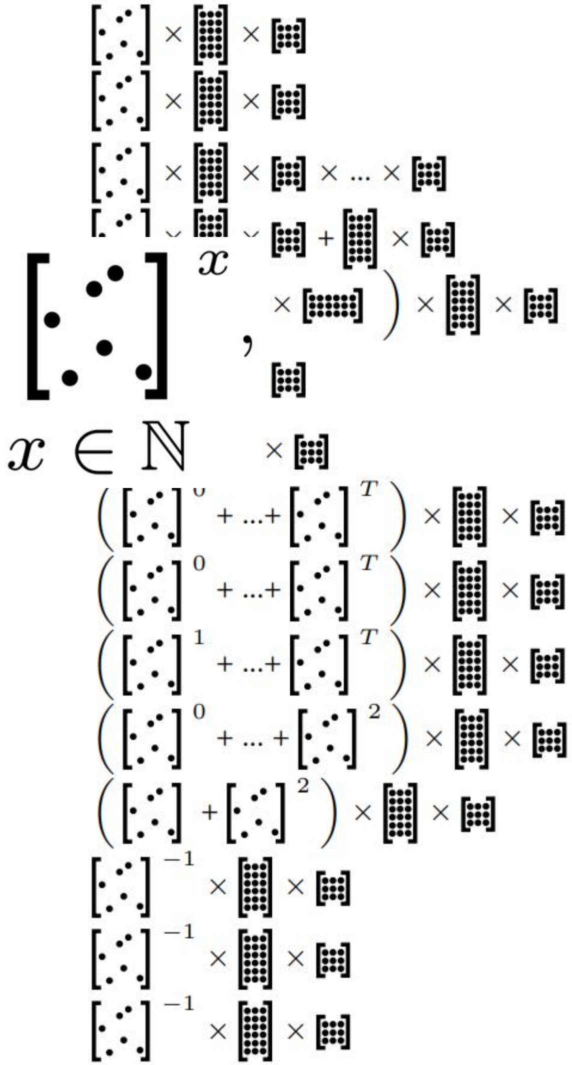


Global Formulations: Parallel Analysis

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

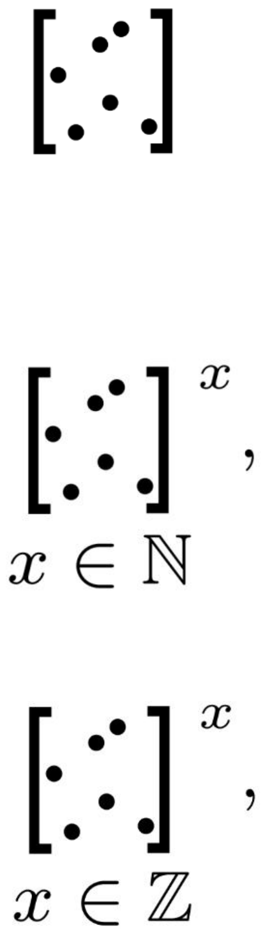


Reference	Type	Algebraic formulation for $\mathbf{H}^{(l+1)}$	Dimensions & density of deriving $\mathbf{H}^{(l+1)}$	Work & depth (one whole training iteration or inference)	
GCN [128]	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
GraphSAGE [101] (mean)	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
GIN [226]	L			$O(mkL + KLnk^2)$	$O(LK \log k + LK \log d)$
CommNet [192]	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
Dot Product [201]	L			$O(Lmk + Lnk^2)$	$O(L \log k + L \log d)$
EdgeConv [216] "choice 1"	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
SGC [219]	P			$O(mn \log s + nk^2)$	$O(\log k + \log s \log d)$
DeepWalk [168]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
ChebNet [72]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
DCNN [6], GDC [130]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
Node2Vec [97]	P			$O(mn + nk^2)$	$O(\log k + \log d)$
LINE [148], SDNE [207]	P			$O(mn + nk^2)$	$O(\log k + \log d)$
Auto-Regress [250], [256]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
PPNP [43], [129], [230]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
ARMA [38], ParWalks [221]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$



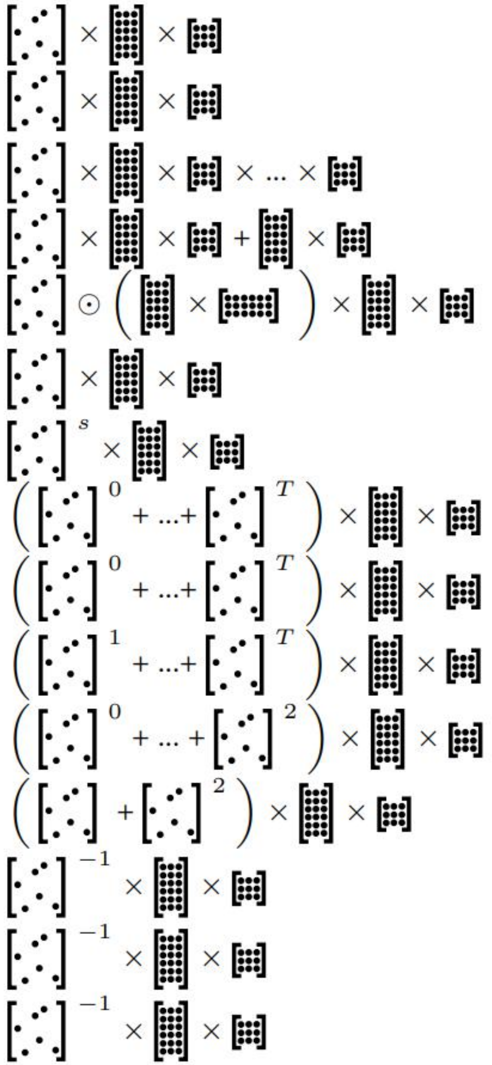
Global Formulations: Parallel Analysis

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$



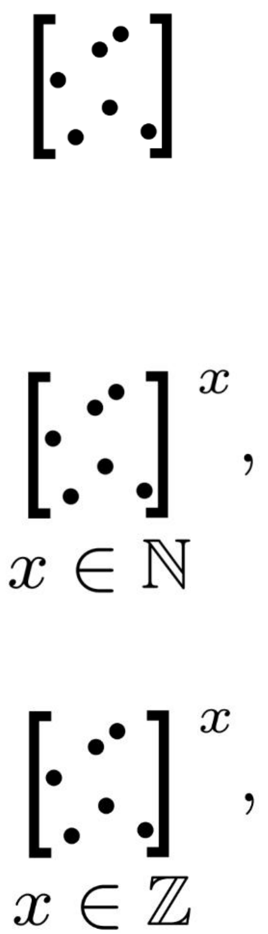
Depth of one layer (in Linear models) is logarithmic

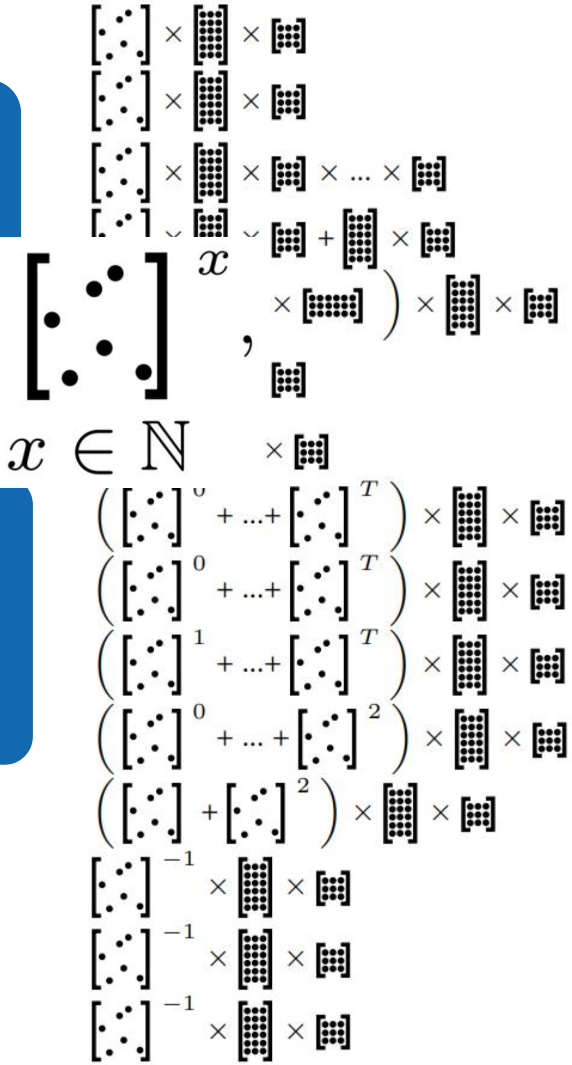
Reference	Type	Algebraic formulation for $\mathbf{H}^{(l+1)}$	Dimensions & density of deriving $\mathbf{H}^{(l+1)}$	Work & depth (one whole training iteration or inference)	
GCN [128]	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
GraphSAGE [101] (mean)	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
GIN [226]	L			$O(mkL + KLnk^2)$	$O(LK \log k + LK \log d)$
CommNet [192]	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
Dot Product [201]	L			$O(Lmk + Lnk^2)$	$O(L \log k + L \log d)$
EdgeConv [216] "choice 1"	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
SGC [219]	P			$O(mn \log s + nk^2)$	$O(\log k + \log s \log d)$
DeepWalk [168]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
ChebNet [72]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
DCNN [6], GDC [130]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
Node2Vec [97]	P			$O(mn + nk^2)$	$O(\log k + \log d)$
LINE [148], SDNE [207]	P			$O(mn + nk^2)$	$O(\log k + \log d)$
Auto-Regress [250], [256]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
PPNP [43], [129], [230]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
ARMA [38], ParWalks [221]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$



Global Formulations: Parallel Analysis

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$





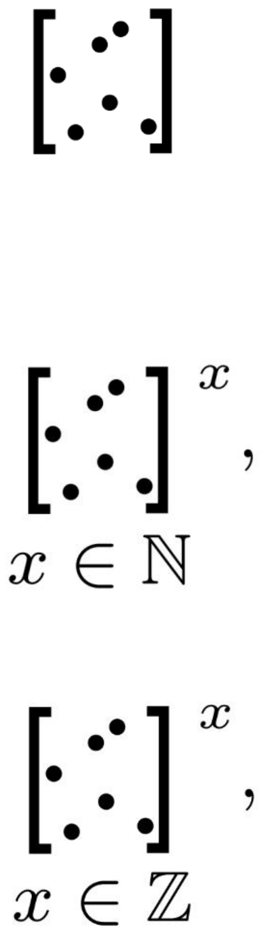
Depth of one layer (in Linear models) is logarithmic

Depth is logarithmic (in Polynomial models)

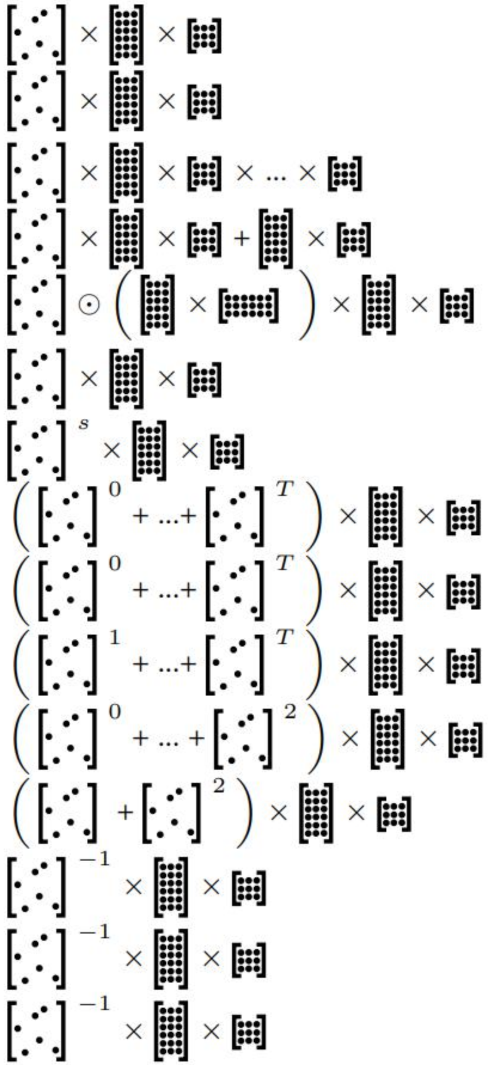
Reference	Type	Algebraic formulation for $\mathbf{H}^{(l+1)}$	Dimensions & density of deriving $\mathbf{H}^{(l+1)}$	Work & depth (one whole training iteration or inference)	
GCN [128]	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
GraphSAGE [101] (mean)	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
GIN [226]	L			$O(mkL + KLnk^2)$	$O(LK \log k + LK \log d)$
CommNet [192]	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
Dot Product [201]	L			$O(Lmk + Lnk^2)$	$O(L \log k + L \log d)$
EdgeConv [216] "choice 1"	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
SGC [219]	P			$O(mn \log s + nk^2)$	$O(\log k + \log s \log d)$
DeepWalk [168]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
ChebNet [72]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
DCNN [6], GDC [130]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
Node2Vec [97]	P			$O(mn + nk^2)$	$O(\log k + \log d)$
LINE [148], SDNE [207]	P			$O(mn + nk^2)$	$O(\log k + \log d)$
Auto-Regress [250], [256]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
PPNP [43], [129], [230]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
ARMA [38], ParWalks [221]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$

Global Formulations: Parallel Analysis

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$



Reference	Type	Algebraic formulation for $\mathbf{H}^{(l+1)}$	Dimensions & density of deriving $\mathbf{H}^{(l+1)}$	Work & depth (one whole training iteration or inference)	
GCN [128]	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
GraphSAGE [101] (mean)	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
GIN [226]	L			$O(mkL + KLnk^2)$	$O(LK \log k + LK \log d)$
CommNet [192]	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
Dot Product [201]	L			$O(Lmk + Lnk^2)$	$O(L \log k + L \log d)$
EdgeConv [216] "choice 1"	L			$O(mkL + Lnk^2)$	$O(L \log k + L \log d)$
SGC [219]	P			$O(mn \log s + nk^2)$	$O(\log k + \log s \log d)$
DeepWalk [168]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
ChebNet [72]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
DCNN [6], GDC [130]	P			$O(mn \log T + nk^2)$	$O(\log k + \log T \log d)$
Node2Vec [97]	P			$O(mn + nk^2)$	$O(\log k + \log d)$
LINE [148], SDNE [207]	P			$O(mn + nk^2)$	$O(\log k + \log d)$
Auto-Regress [250], [256]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
PPNP [43], [129], [230]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$
ARMA [38], ParWalks [221]	R			$O(n^3 + nk^2)$	$O(\log^2 n + \log k)$



Depth of one layer (in Linear models) is logarithmic

Depth is logarithmic (in Polynomial models)

Depth is square logarithmic (in Rational models)

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Some models have both formulations, e.g., Graph Convolution Network.
Such models have the same work/depth in both formulations (i.e., they have fundamentally the same amount of parallelism)

$$\mathbf{h}_i^{(l+1)} = ReLU\left(\mathbf{W}^{(l)} \times \left(\sum_{j \in \widehat{N}(i)} \frac{1}{\sqrt{d_i d_j}} \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = ReLU(\mathbf{A}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$$

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\dots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

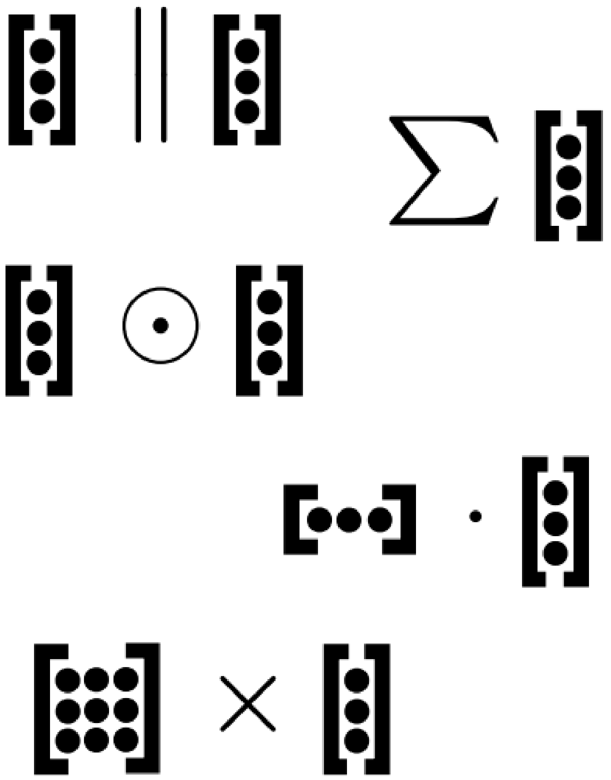
Different linear algebra kernels are used

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Different linear algebra kernels are used

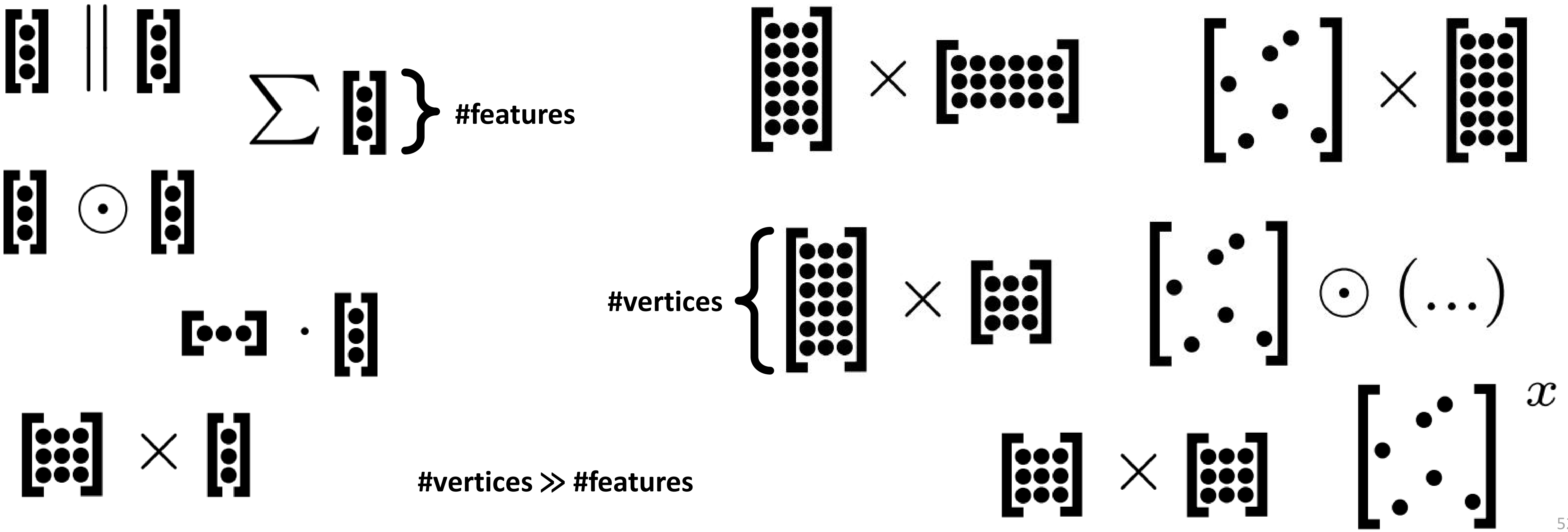


Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\dots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Different linear algebra kernels are used



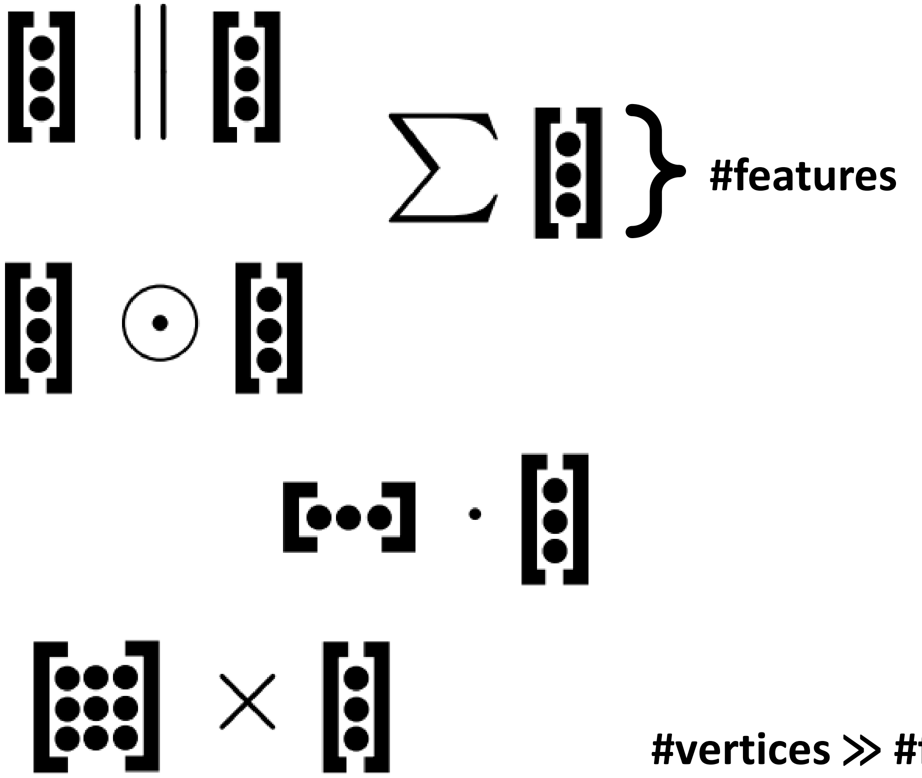
Local vs. Global Formulations

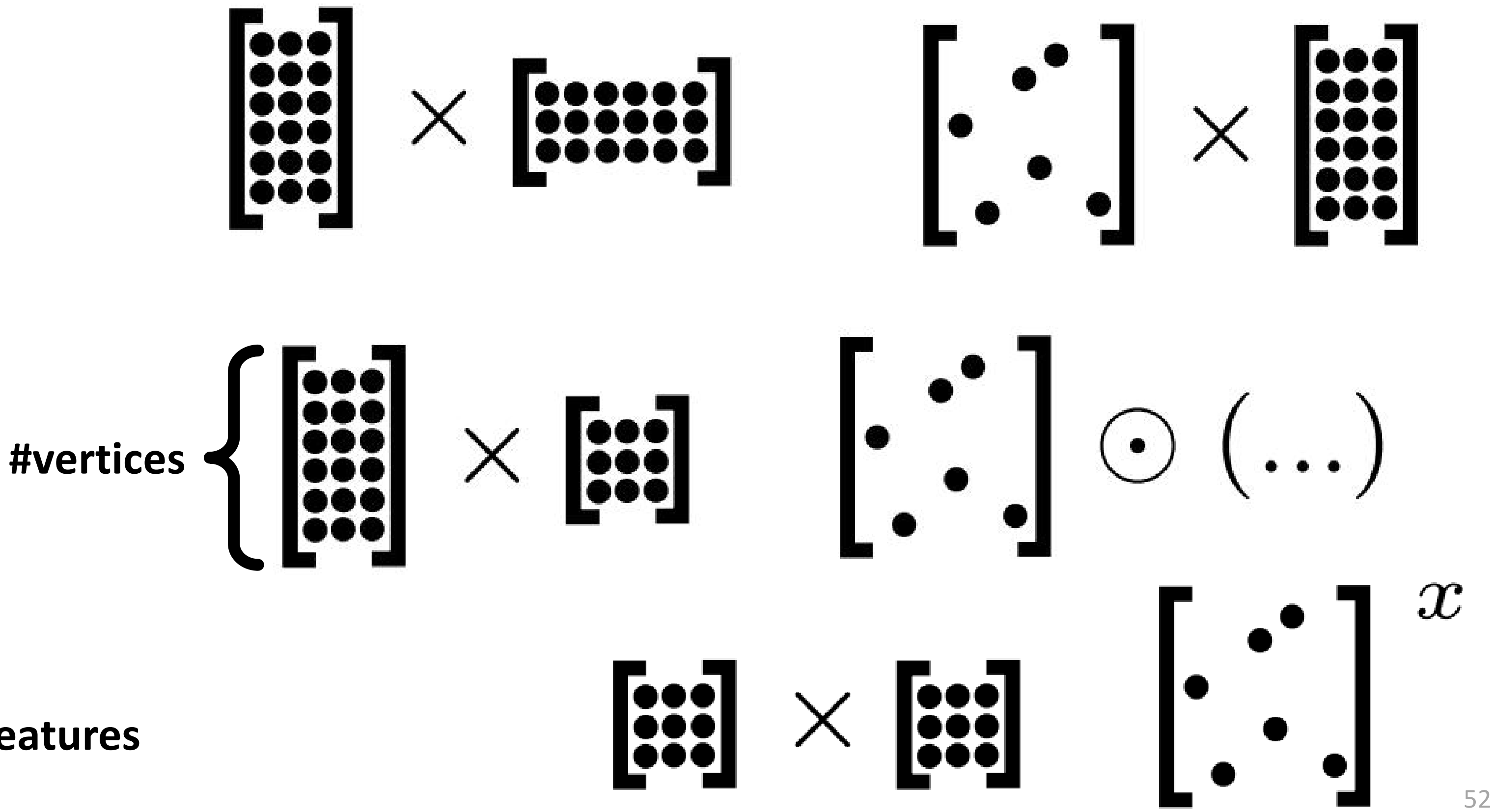
$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Different linear algebra kernels are used

Potential for different optimizations. For example, **there may be more opportunities to use vectorization in the global formulations** (one can vectorize matrices that group all vertices and edges)





#vertices ≫ #features

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Some models (A-GNNs, MP-GNNs) do **not** have known global formulations.
One example is the original Graph Attention (GAT) model

$$\psi: \quad \frac{\exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{y \in \widehat{N}(i)} \exp\left(\sigma\left(\mathbf{a}^T \cdot \left[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_y\right]\right)\right)} \mathbf{h}_j$$

$$\phi: \quad \mathbf{W} \times \left(\sum_{j \in \widehat{N}(i)} \psi\left(\mathbf{h}_i, \mathbf{h}_j\right)\right)$$

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Polynomial & Rational models do **not** have known local formulations (e.g., Node2Vec or PPNP)

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Polynomial & Rational models do **not** have known local formulations (e.g., Node2Vec or PPNP)

Node2Vec:

$$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$$

PPNP:

$$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$$

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Polynomial & Rational models do **not** have known local formulations (e.g., Node2Vec or PPNP)

They still also offer parallelism: *O(log n)* (Polynomial) and *O(log² n)* (Rational) depth

Node2Vec:

$$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$$

PPNP:

$$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$$

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Polynomial & Rational models do **not** have known local formulations (e.g., Node2Vec or PPNP)

They still also offer parallelism: *O(log n)* (Polynomial) and *O(log² n)* (Rational) depth

While they have one iteration, making *L* vanish, they require deriving a given power of ***A***

Node2Vec:

$$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$$

PPNP:

$$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$$

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\dots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Polynomial & Rational models do **not** have known local formulations (e.g., Node2Vec or PPNP)

They still also offer parallelism: *O(log n)* (Polynomial) and *O(log² n)* (Rational) depth

While they have one iteration, making *L* vanish, they require deriving a given power of ***A***

Node2Vec:

$$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$$

PPNP:

$$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$$

As computing powers of ***A*** is not interleaved with non-linearities (as is the case with many local models), the increase in work and depth is **only logarithmic**, indicating **more parallelism**

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Polynomial & Rational models do **not** have known local formulations (e.g., Node2Vec or PPNP)

They still also offer parallelism: *O(log n)* (Polynomial) and *O(log² n)* (Rational) depth

While they have one iteration, making *L* vanish, they require deriving a given power of ***A***

Node2Vec:

$$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$$

PPNP:

$$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$$

As computing powers of ***A*** is not interleaved with non-linearities (as is the case with many local models), the increase in work and depth is **only logarithmic**, indicating **more parallelism**

Still, their representative power **may be lower**, due to the lack of non-linearities

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus_{j \in N(i)} \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Polynomial & Rational models do **not** have known local formulations (e.g., Node2Vec or PPNP)

They still also offer parallelism: *O(log n)* (Polynomial) and *O(log² n)* (Rational) depth

While they have one iteration, making *L* vanish, they require deriving a given power of ***A***

Node2Vec:

$$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$$

PPNP:

$$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$$

As computing powers of ***A*** is not interleaved with non-linearities (as is the case with many local models), the increase in work and depth is **only logarithmic**, indicating **more parallelism**

Still, their representative power **may be lower**, due to the lack of non-linearities

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Models with only local formulations: potential for better representative power

Po... ions (e.g., Node2Vec or PPNP)

They still also offer parallelism: *O(log n)* (Polynomial) and *O(log² n)* (Rational) depth

While they have one iteration, making *L* vanish, they require deriving a given power of ***A***

Node2Vec:

$$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$$

PPNP:

$$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$$

As computing powers of ***A*** is not interleaved with non-linearities (as is the case with many local models), the increase in work and depth is **only logarithmic**, indicating **more parallelism**

Still, their representative power **may be lower**, due to the lack of non-linearities

Local vs. Global Formulations

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \bigoplus \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right)\right)$$

$$\mathbf{H}^{(l+1)} = f(\mathbf{A}) \odot (\ldots) \times \mathbf{H}^{(l)} \times \mathbf{W}^{(l)}$$

Models with only local formulations:
potential for better representative power

...ions (e.g., Node2Vec or PPNP)

They still also offer parallelism: *O(log n)* (Polynomial) and *O(log² n)* (Rational) depth

While they have one iteration, making *L* vanish, they require deriving a given power of **A**

Models with only global formulations:
potential for higher performance

Node2Vec:

$$\left(\frac{1}{p}\mathbf{I} + \left(1 - \frac{1}{q}\right)\overline{\mathbf{A}} + \frac{1}{q}\overline{\mathbf{A}}^2\right)\mathbf{HW}$$

... interleaved with non-linearities (as is the case with many local models), the increase in work and depth is **only logarithmic**, indicating **more parallelism**

PPNP:

$$\alpha\left(\mathbf{I} - (1-\alpha)\widehat{\mathbf{A}}\right)^{-1}\mathbf{HW}$$

Still, their representative power **may be lower**, due to the lack of non-linearities

Local vs. Global Formulations

Models with only local formulations:
potential for better representative power

Models with only global formulations:
potential for higher performance

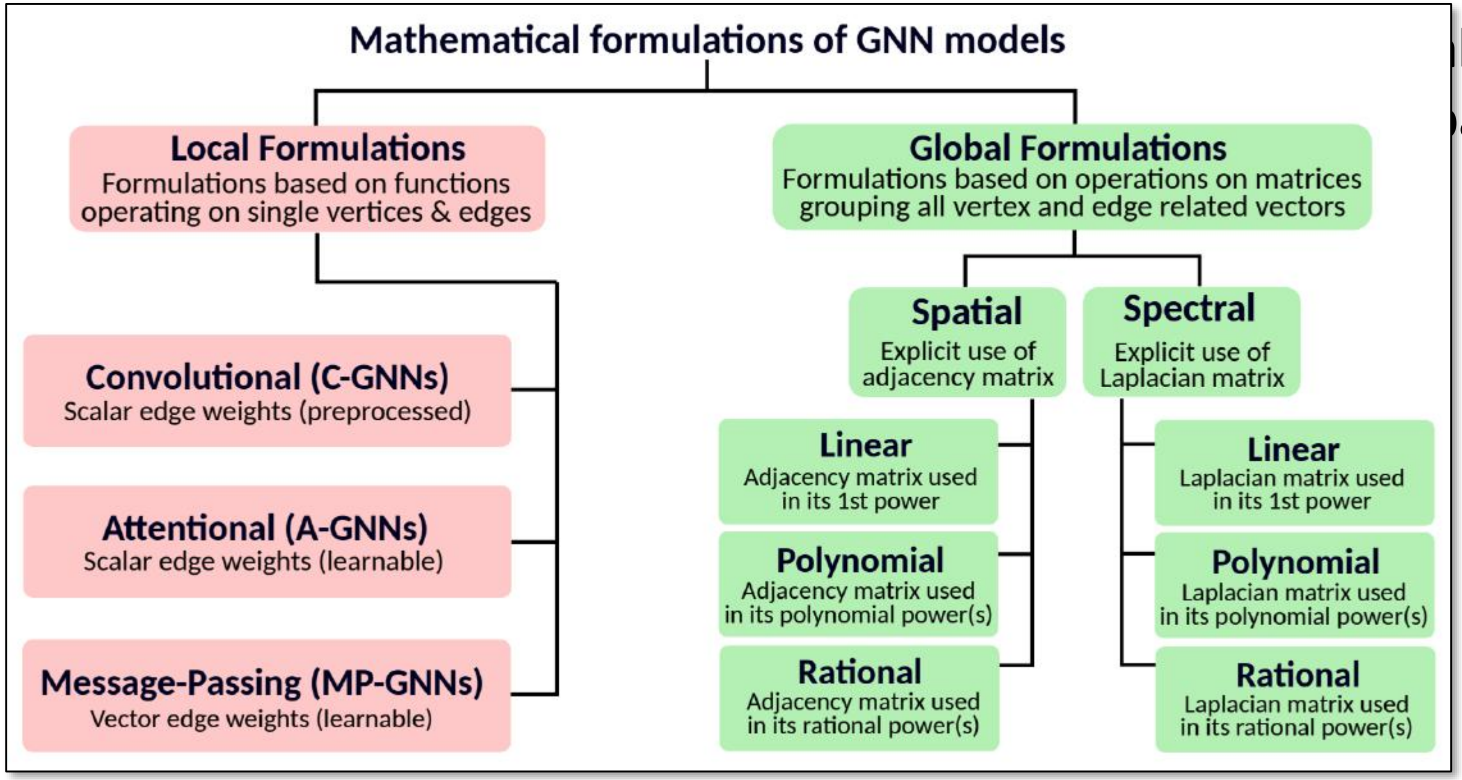
Models with both formulations: **potential for both better representative power and higher performance**

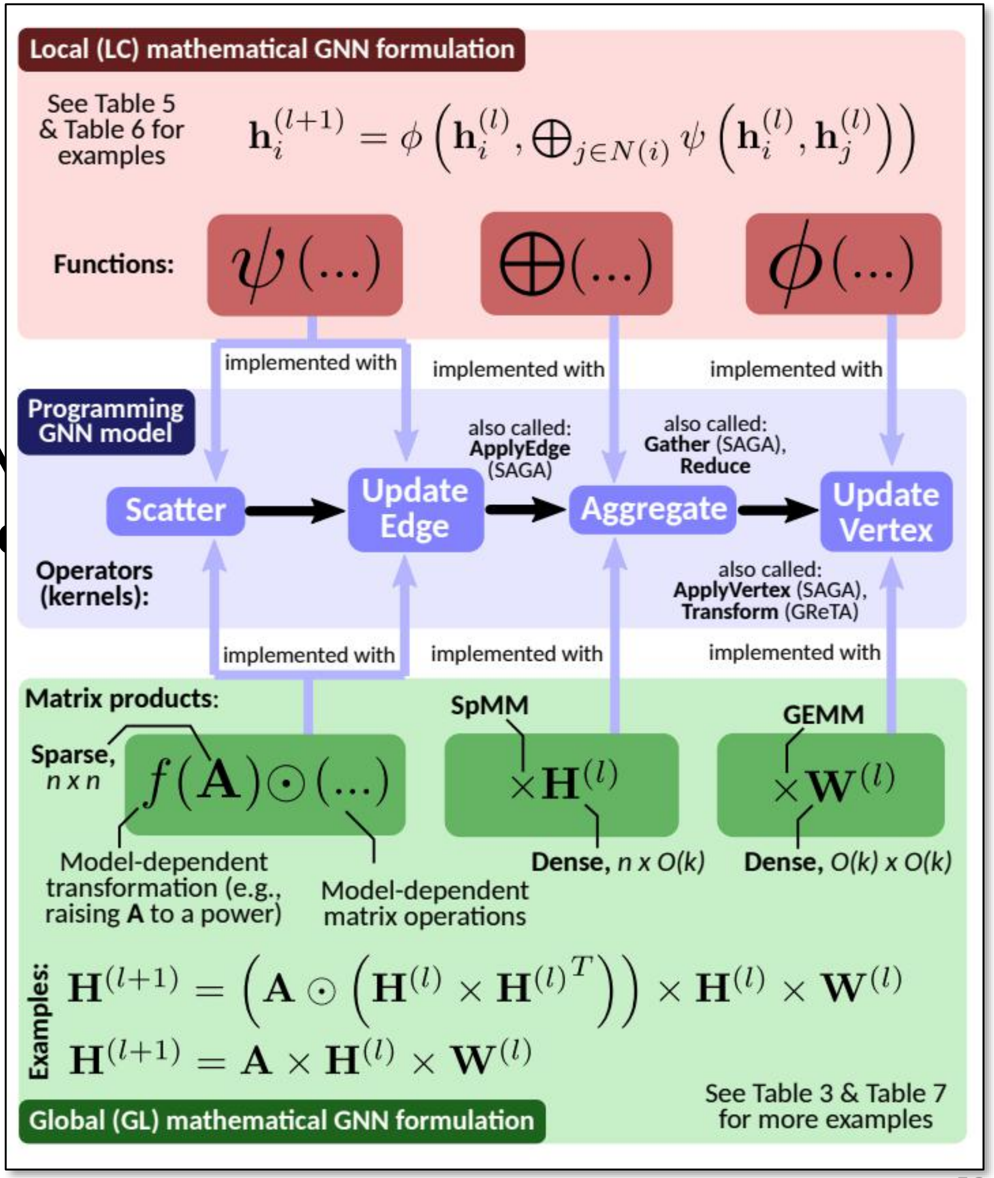
Time for some “Bragging slides” ;-)
(i.e., what’s also in there)

... Check the paper ☺

... Check the paper ☺

More details on mathematical GNN models and GNN programming paradigms



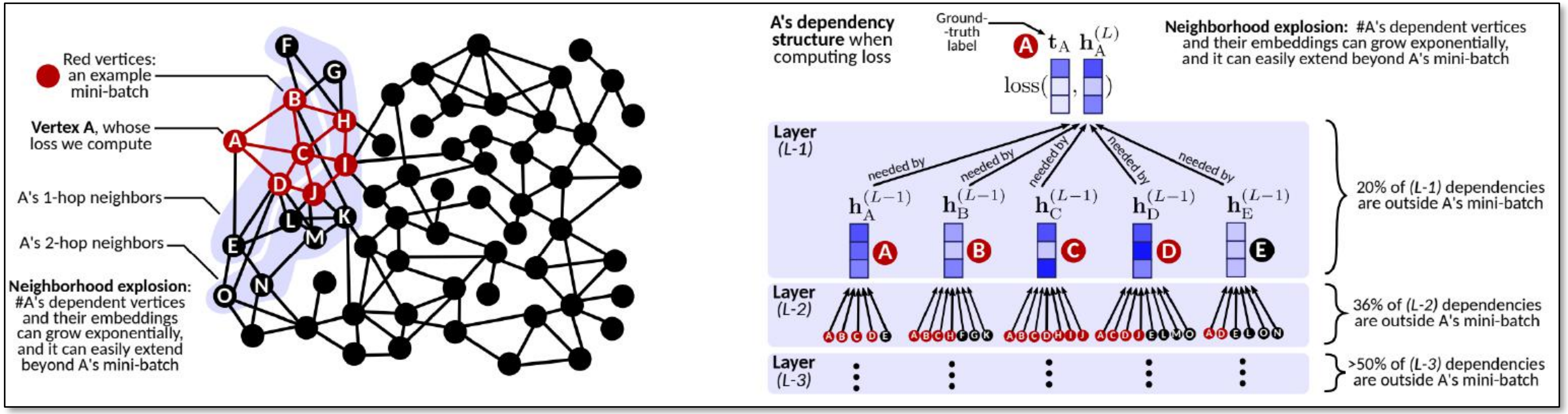


… Check the paper ☺

Different effects and challenges (neighborhood explosion, types of partitioning, types of sampling, …)

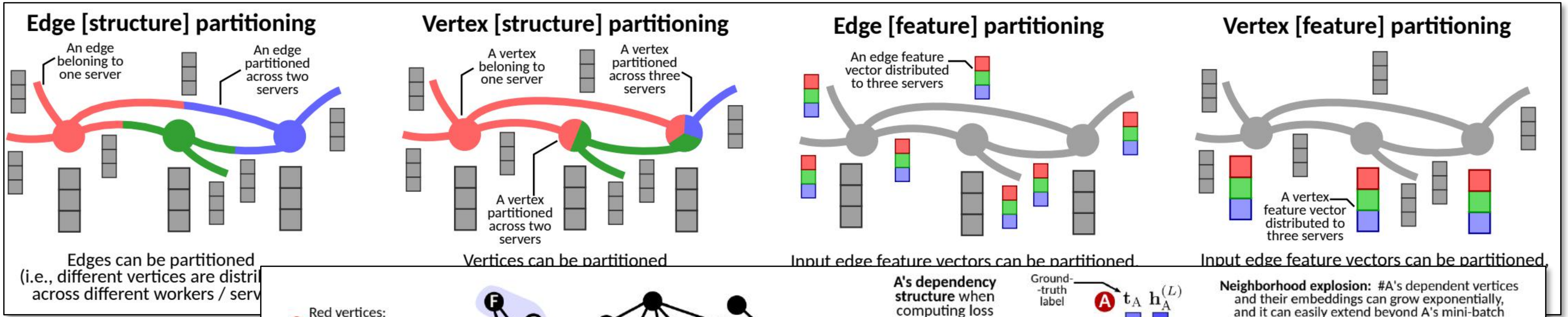
... Check the paper ☺

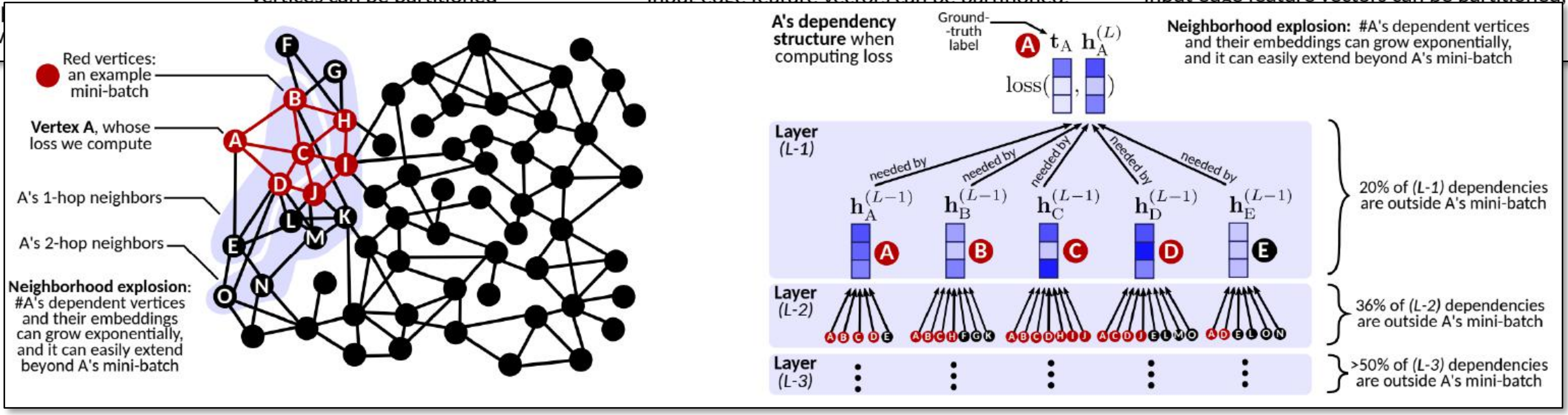
Different effects and challenges (neighborhood explosion, types of partitioning, types of sampling, …)



... Check the paper ☺

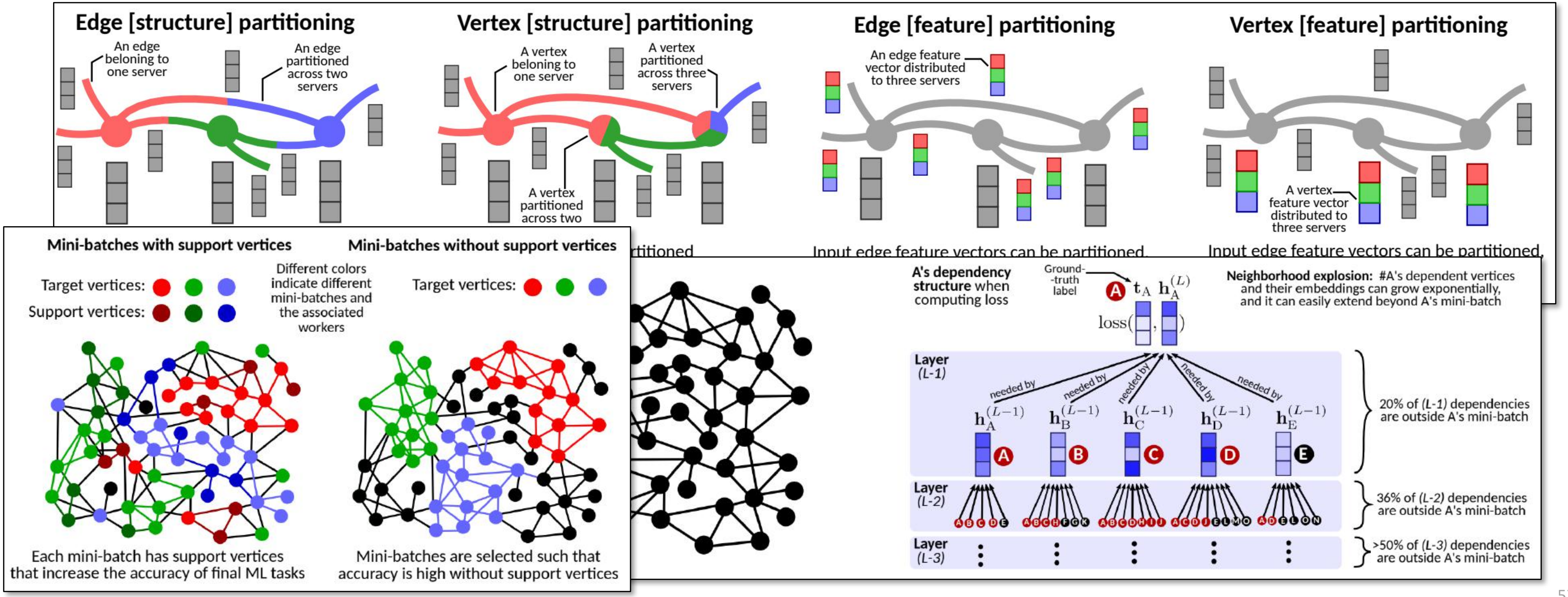
Different effects and challenges (neighborhood explosion, types of partitioning, types of sampling, …)





... Check the paper ☺

Different effects and challenges (neighborhood explosion, types of partitioning, types of sampling, …)



... Check the paper ☺

More work-depth analyses, plus communication & synchronization

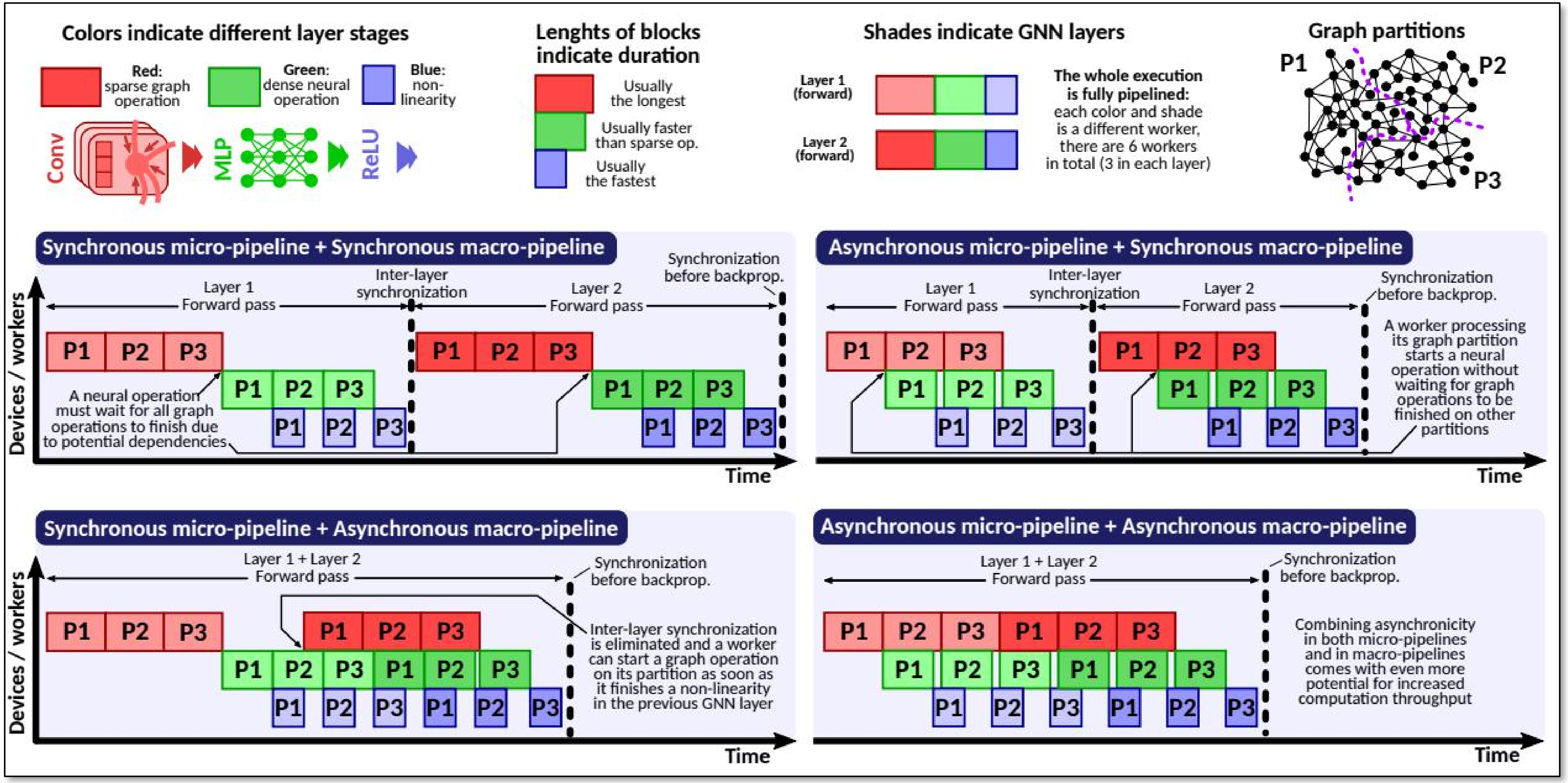
Method	Work & depth in one training iteration	
	Full-batch training schemes:	
Full-batch [128]	$O\left(Lmk + Lnk^2\right)$	$O\left(L\log k + L\log d\right)$
Weight-tying [139]	$O\left(Lmk + Lnk^2\right)$	$O\left(L\log k + L\log d\right)$
RevGNN [139]	$O\left(Lmk + Lnk^2\right)$	$O\left(L\log k + L\log d\right)$
	Mini-batch training schemes:	
GraphSAGE [101]	$O\left(Lmk + Lnk^2 + c^L nk^2\right)$	$O\left(L\log k + L\log c\right)$
VR-GCN [60]	$O\left(Lmk + Lnk^2 + c^L nk^2\right)$	$O\left(L\log k + L\log c\right)$
FastGCN [61]	$O\left(Lmk + Lnk^2 + cLnk^2\right)$	$O\left(L\log k + L\log c\right)$
Cluster-GCN [65]	$O\left(W_{pre} + Lmk + Lnk^2\right)$	$O\left(D_{pre} + L\log k + L\log d\right)$
GraphSAINT [234]	$O\left(W_{pre} + Lmk + Lnk^2\right)$	$O\left(D_{pre} + L\log k + L\log d\right)$

... Check the paper ☺

Asynchronous GNNs

... Check the paper ☺

Asynchronous GNNs



… Check the paper ☺

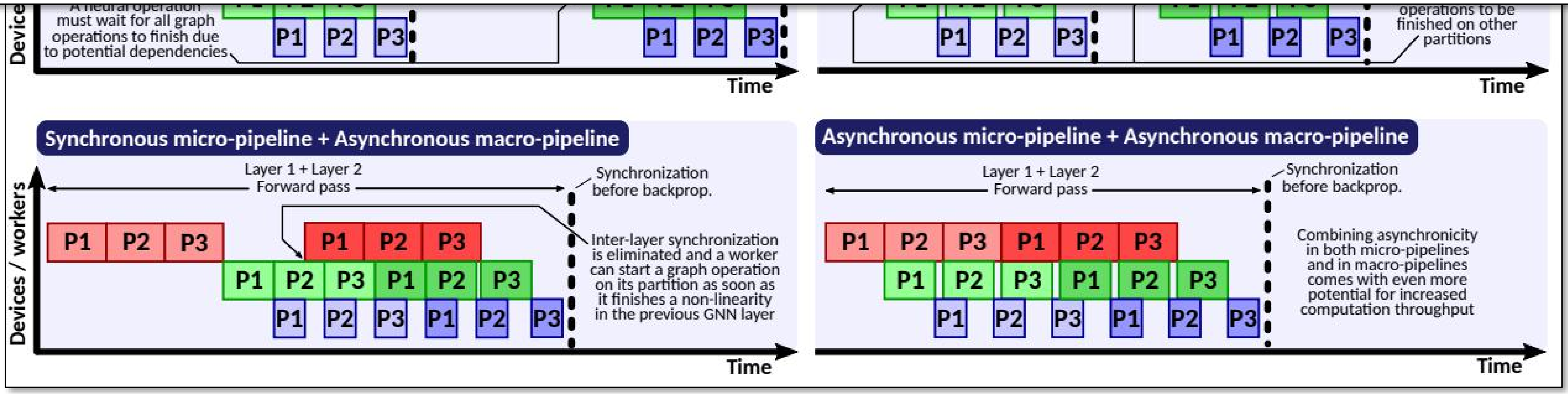
Asynchronous GNNs

Standard computation with graph partition parallelism:

$$\mathbf{h}_i^{(t,l)} = \phi\left(\mathbf{h}_i^{(t,l-1)}, \bigoplus_{j\in N^{\mathcal{L}}(i)} \psi\left(\mathbf{h}_i^{(t,l-1)}, \mathbf{h}_j^{(t,l-1)}\right) \bigoplus_{j\in N^{\mathcal{R}}(i)} \psi\left(\mathbf{h}_i^{(t,l-1)}, \mathbf{h}_j^{(t,l-1)}\right)\right) \quad (2)$$

Using bounded stale feature vectors with graph partition parallelism (worst case):

$$\mathbf{h}_i^{(t,l)} = \phi\left(\mathbf{h}_i^{(t-T_\phi,l-L_\phi)}, \bigoplus_{j\in N^{\mathcal{L}}(i)} \psi\left(\mathbf{h}_i^{(t-T_\phi,l-L_\phi)}, \mathbf{h}_j^{(t-T_\psi^{\mathcal{L}},l-L_\psi^{\mathcal{L}})}\right) \bigoplus_{j\in N^{\mathcal{R}}(i)} \psi\left(\mathbf{h}_i^{(t-T_\phi,l-L_\phi)}, \mathbf{h}_j^{(t-T_\psi^{\mathcal{R}},l-L_\psi^{\mathcal{R}})}\right)\right) \quad (3)$$



... Check the paper ☺

Asynchronous GNNs

Standard computation with graph partition parallelism:

$$\mathbf{h}_i^{(t,l)} = \phi\left(\mathbf{h}_i^{(t,l-1)}, \bigoplus_{j\in N^{\mathcal{L}}(i)} \psi\left(\mathbf{h}_i^{(t,l-1)}, \mathbf{h}_j^{(t,l-1)}\right) \bigoplus_{j\in N^{\mathcal{R}}(i)} \psi\left(\mathbf{h}_i^{(t,l-1)}, \mathbf{h}_j^{(t,l-1)}\right)\right) \quad (2)$$

Using bounded stale feature vectors with graph partition parallelism (worst case):

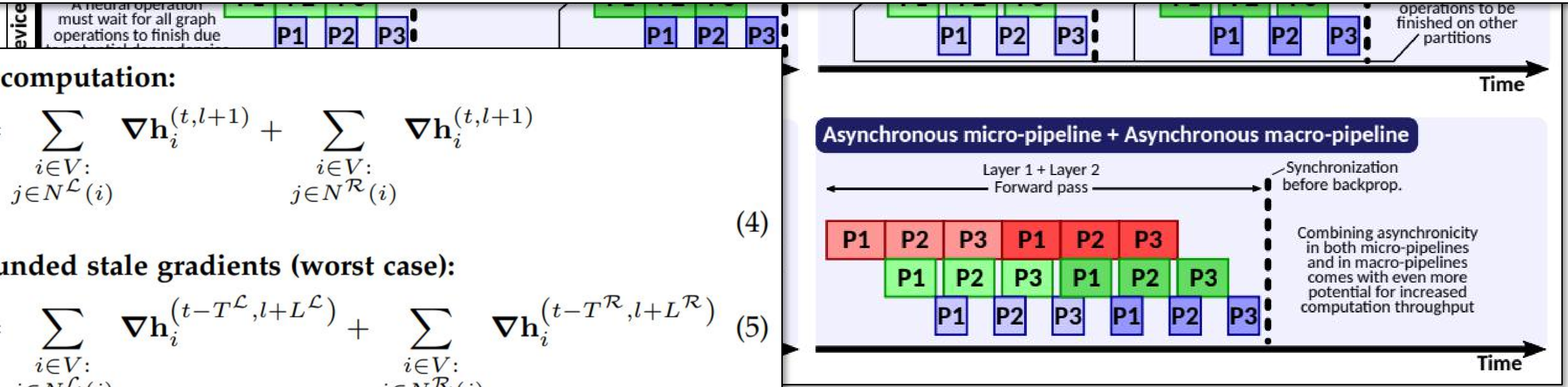
$$\mathbf{h}_i^{(t,l)} = \phi\left(\mathbf{h}_i^{(t-T_\phi,l-L_\phi)}, \bigoplus_{j\in N^{\mathcal{L}}(i)} \psi\left(\mathbf{h}_i^{(t-T_\phi,l-L_\phi)}, \mathbf{h}_j^{(t-T_\psi^{\mathcal{L}},l-L_\psi^{\mathcal{L}})}\right) \bigoplus_{j\in N^{\mathcal{R}}(i)} \psi\left(\mathbf{h}_i^{(t-T_\phi,l-L_\phi)}, \mathbf{h}_j^{(t-T_\psi^{\mathcal{R}},l-L_\psi^{\mathcal{R}})}\right)\right) \quad (3)$$

Standard computation:

$$\nabla\mathbf{h}_j^{(t,l)} = \sum_{\substack{i\in V:\\ j\in N^{\mathcal{L}}(i)}} \nabla\mathbf{h}_i^{(t,l+1)} + \sum_{\substack{i\in V:\\ j\in N^{\mathcal{R}}(i)}} \nabla\mathbf{h}_i^{(t,l+1)} \quad (4)$$

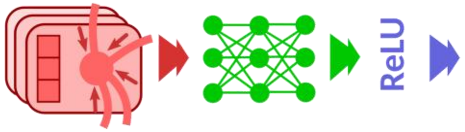
Using bounded stale gradients (worst case):

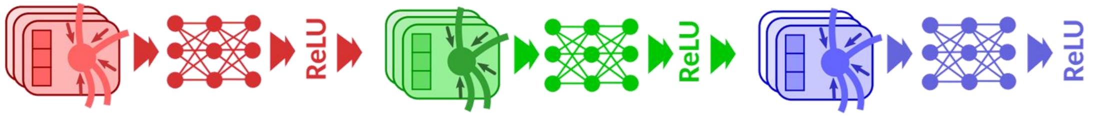
$$\nabla\mathbf{h}_j^{(t,l)} = \sum_{\substack{i\in V:\\ j\in N^{\mathcal{L}}(i)}} \nabla\mathbf{h}_i^{(t-T^{\mathcal{L}},l+L^{\mathcal{L}})} + \sum_{\substack{i\in V:\\ j\in N^{\mathcal{R}}(i)}} \nabla\mathbf{h}_i^{(t-T^{\mathcal{R}},l+L^{\mathcal{R}})} \quad (5)$$



... Check the paper ☺

Asynchronous GNNs



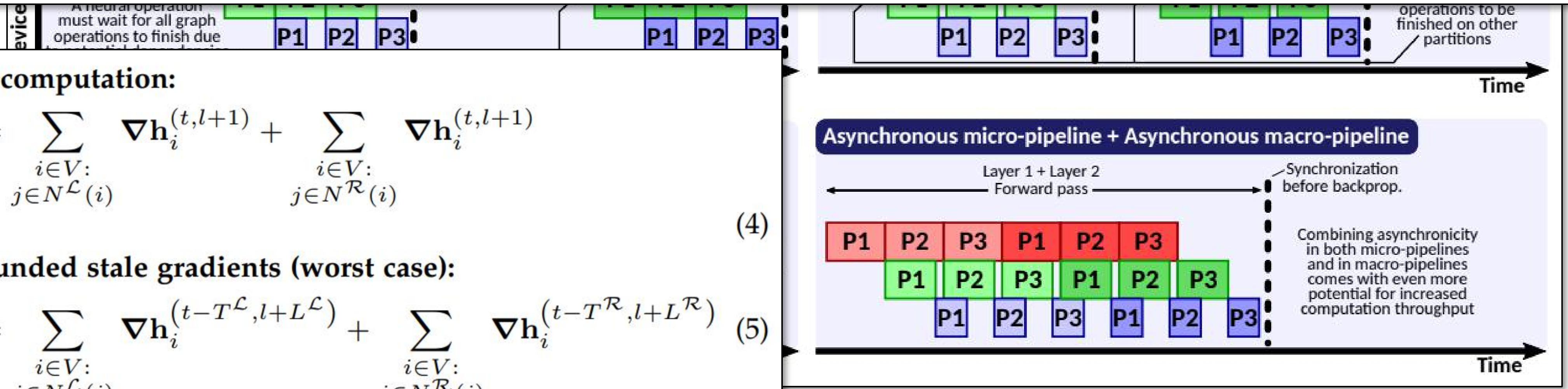


Standard computation with graph partition parallelism:

$$\mathbf{h}_i^{(t,l)} = \phi\left(\mathbf{h}_i^{(t,l-1)}, \bigoplus_{j\in N^{\mathcal{L}}(i)} \psi\left(\mathbf{h}_i^{(t,l-1)}, \mathbf{h}_j^{(t,l-1)}\right) \bigoplus_{j\in N^{\mathcal{R}}(i)} \psi\left(\mathbf{h}_i^{(t,l-1)}, \mathbf{h}_j^{(t,l-1)}\right)\right) \quad (2)$$

Using bounded stale feature vectors with graph partition parallelism (worst case):

$$\mathbf{h}_i^{(t,l)} = \phi\left(\mathbf{h}_i^{(t-T_\phi,l-L_\phi)}, \bigoplus_{j\in N^{\mathcal{L}}(i)} \psi\left(\mathbf{h}_i^{(t-T_\phi,l-L_\phi)}, \mathbf{h}_j^{(t-T_\psi^{\mathcal{L}},l-L_\psi^{\mathcal{L}})}\right) \bigoplus_{j\in N^{\mathcal{R}}(i)} \psi\left(\mathbf{h}_i^{(t-T_\phi,l-L_\phi)}, \mathbf{h}_j^{(t-T_\psi^{\mathcal{R}},l-L_\psi^{\mathcal{R}})}\right)\right) \quad (3)$$



Standard computation:

$$\nabla\mathbf{h}_j^{(t,l)} = \sum_{\substack{i\in V:\\ j\in N^{\mathcal{L}}(i)}} \nabla\mathbf{h}_i^{(t,l+1)} + \sum_{\substack{i\in V:\\ j\in N^{\mathcal{R}}(i)}} \nabla\mathbf{h}_i^{(t,l+1)} \quad (4)$$

Using bounded stale gradients (worst case):

$$\nabla\mathbf{h}_j^{(t,l)} = \sum_{\substack{i\in V:\\ j\in N^{\mathcal{L}}(i)}} \nabla\mathbf{h}_i^{(t-T^{\mathcal{L}},l+L^{\mathcal{L}})} + \sum_{\substack{i\in V:\\ j\in N^{\mathcal{R}}(i)}} \nabla\mathbf{h}_i^{(t-T^{\mathcal{R}},l+L^{\mathcal{R}})} \quad (5)$$

... Check the paper ☺

Parallel analysis of frameworks and accelerators

... Check the paper ☺

Parallel analysis of frameworks and accelerators

Reference	Arch.	Ds?	T?	I?	Op?	mp?	Mp?	Dp?	Dpp	PM	Remarks
[SW] PipeGCN [206]	CPU+GPU	■	■ (fb)	✗	■	■	✗	■	sh	LC	
[SW] BNS-GCN [205]	GPU	■	■ (fb)	✗	?	?	✗	■	sh		
[SW] PaSca [243]	GPU	■	■	■	?	?	?	■			
[SW] Marius++ [204]	CPU	✗	■ (mb)	✗	■ (v)	?	✗	■		LC (SU)	Focus on using disk
[SW] BGL [151]	GPU	■	■ (mb)	✗	■	■	✗	■	sh	—	
[SW] DistDGLv2 [248]	CPU+GPU	■	■ (mb)	✗	■	■	✗	■	sh	—	
[SW] SAR [159]	CPU	■	■ (fb)	✗	✗	✗	✗	■			
[SW] DeepGalois [105]	CPU	■	■ (fb)	✗	?	✗	✗	■ (v)	sh	LC (AU)	
[SW] DistGNN [155]	CPU	■	■ (fb)	✗	?	✗	✗	■ (v)	sh	LC(AU) ?	
[SW] DGCL [54]	GPU	◧*	■	✗	?	✗	✗	■ (v)		LC (AU)	*Only two servers used.
[SW] Seastar [222]	GPU	✗	■	✗	■ (f)	✗	✗	■ (v, t)		LC (VC)	
[SW] Chakaravarthy [56]	GPU	■	■ (fb)	✗	✗	✗	✗	■ (v, sn)		✗	
[SW] Zhou et al. [249]	CPU	✗	✗	■	■ (f)	✗	✗	?		✗	
[SW] MC-GCN [12]	GPU	◧*	■ (fb)	✗	■ (f)	✗	✗	■ (v)		GL	*Multi-GPU within one node.
[SW] Dorylus [197]	CPU	■	■ (fb)	✗	✗	✗	■	■ (v)		LC (SAGA)	
[SW] Min et al. [156]	GPU	◧*	■ (mb)	✗	?	✗	■	■ (v)		GL	*Multi-GPU within one node.
[SW] GNNAdvisor [215]	GPU	✗	■	■	■ (f, s)	✗	✗	?		GL, LC	
[SW] AliGraph [255]	CPU	■	■	■	?	✗	?	■		LC (NAU)	
[SW] FlexGraph [208]	CPU	■	■ (fb)	✗	■ (s)	?	✗	■		LC (NAU)	
[SW] Kim et al. [125]	CPU+GPU	✗	■ (mb)	✗	■ (s)	?	✗	■		LC (AU)	
[SW] AGL [237]	CPU	■	■ (mb)	■	✗	■	■	■		MapReduce	
[SW] ROC [117]	CPU+GPU	■	■ (fb)	■	✗	✗	✗	■		✗	
[SW] DistDGL [247]	CPU	■	■ (mb)	✗	?	✗	✗	■		✗	
[SW] PaGraph [10], [149]	GPU	◧*	■ (mb)	✗	?	✗	✗	■		✗	*Multi-GPU within one node.
[SW] 2PGraph [240]	GPU	■	■ (mb)	✗	?	■	?	■		—	
[SW] GMLP [242]	GPU	■	■ (mb)	✗	?	✗	?	■		LC	
[SW] fuseGNN [64]	GPU	✗	■	✗	■	✗	✗	■		LC (AU)*	*Two aggregation schemes are used.
[SW] P^3 [81]	CPU+GPU	■	■ (mb)	✗	■	■	✗	■		LC (SAGA)*	*A variant called P-TAGS
[SW] QGTC [214]	GPU	✗	■	✗	?	✗	◧	■	sh	GL	
[SW] CAGNET [199]	CPU+GPU	■	■ (fb)	✗	■ (f, s)	✗	✗	■ (v, e)	sh+rep	GL	
[SW] PCGCN [198]	CPU+GPU	✗	■	✗	✗	✗	✗	■ (e)	sh	—	
[SW] FeatGraph [111]	CPU, GPU	✗	■ (fb)	■	■ (f, s)	✗	✗	■ (v)	sh	GL	
[SW] G^3 [150]	GPU	✗	■	■	?	?	?	?		GL	
[SW] NeuGraph [153]	GPU	■	■	✗	◧	■	✗	■ (v, e)	sh	LC (SAGA)	
[SW] PyTorch-Direct [79]	GPU	■	■	■	■	?	?	■ (v, e)		GL, LC	
[SW] PyG [79]	CPU, GPU	■	■*	■	■	?	?	■ (v, e)		GL, LC	*Mini-batching for graph components
[SW] DGL [209]	CPU, GPU	■	■*	■	■	?	◧	■		GL, LC	*Mini-batching for graph components

… Check the paper ☺

Parallel analysis of frameworks and accelerators

Reference	Arch.	Ds?	T?	I?	Op?	mp?	Mp?	Dp?	Dpp	PM	Remarks
[SW] PipeGCN [206]	CPU+GPU	■	■ (fb)	✗	■	■	✗	■	sh	LC	
[SW] BNS-GCN [205]	GPU	■	■ (fb)	✗	?	?	✗	■	sh		
[SW] PaSca [243]	GPU	■	■	■	?	?	?	■			
[SW] Marius++ [204]	CPU	✗	■ (mb)	✗	■ (v)	?	✗	■		LC (SU)	Focus on using disk
[SW] BGL [151]	GPU	■	■ (mb)	✗	■	■	✗	■	sh	—	
[SW] DistDGLv2 [248]	CPU+GPU	■	■ (mb)	✗	■	■	✗	■	sh	—	
[SW] SAR [159]	CPU	■	■ (fb)	✗	✗	✗	✗	■			
[SW] DeepGalois [105]	CPU	■	■ (fb)	✗	?	✗	✗	■ (v)	sh	LC (AU)	
[SW] DistGNN [155]	CPU	■	■ (fb)	✗	?	✗	✗	■ (v)	sh	LC(AU) ?	
[SW] DGCL [54]	GPU	◧*	■	✗	?	✗	✗	■ (v)		LC (AU)	*Only two servers used.
[SW] Seastar [222]	GPU	✗	■	✗	■ (f)	✗	✗	■ (v, t)		LC (VC)	
[SW] Chakaravarthy [56]	GPU	■	■ (fb)	✗	✗	✗	✗	■ (v, sn)		✗	
[SW] Zhou et al. [249]	CPU	✗	✗	■	■ (f)	✗	✗	?		✗	
[SW] MC-GCN [12]	GPU	◧*	■ (fb)	✗	■ (f)	✗	✗	■ (v)		GL	*Multi-GPU within one node.
[SW] Dorylus [197]	CPU	■	■ (fb)	✗							
[SW] Min et al. [156]	GPU	◧*	■ (mb)	✗							
[SW] GNNAdvisor [215]	GPU	✗	■	■							
[SW] AliGraph [255]	CPU	■	■	■							
[SW] FlexGraph [208]	CPU	■	■ (fb)	✗							
[SW] Kim et al. [125]	CPU+GPU	✗	■ (mb)	✗							
[SW] AGL [237]	CPU	■	■ (mb)	■							
[SW] ROC [117]	CPU+GPU	■	■ (fb)	■							
[SW] DistDGL [247]	CPU	■	■ (mb)	✗							
[SW] PaGraph [10], [149]	GPU	◧*	■ (mb)	✗							
[SW] 2PGraph [240]	GPU	■	■ (mb)	✗							
[SW] GMLP [242]	GPU	■	■ (mb)	✗							
[SW] fuseGNN [64]	GPU	✗	■	✗							
[SW] P^3 [81]	CPU+GPU	■	■ (mb)	✗							
[SW] QGTC [214]	GPU	✗	■	✗							
[SW] CAGNET [199]	CPU+GPU	■	■ (fb)	✗							
[SW] PCGCN [198]	CPU+GPU	✗	■	✗							
[SW] FeatGraph [111]	CPU, GPU	✗	■ (fb)	■							
[SW] G^3 [150]	GPU	✗	■	■							
[SW] NeuGraph [153]	GPU	■	■	✗	◧	■	✗	■ (v, e)	sh	LC (SAGA)	
[SW] PyTorch-Direct [79]	GPU	■	■	■	■	?	?	■ (v, e)		GL, LC	
[SW] PyG [79]	CPU, GPU	■	■*	■	■	?	?	■ (v, e)		GL, LC	*Mini-batching for graph components
[SW] DGL [209]	CPU, GPU	■	■*	■	■	?	◧	■		GL, LC	*Mini-batching for graph components

Reference	Arch.	Ds?	T?	I?	Op?	mp?	Mp?	Dp?	Dpp	PM
[HW] ZIPPER [245]	new	✗	✗	■	◧	■	✗	■ (v, e)	sh	GL, LC
[HW] GCNear [253]	new (PIM)	✗	■ (fb)	✗	■	■	✗	■ (v, e)	sh	LC (AU)
[HW] BlockGNN [254]	new	✗	■	■	◧	■	✗	■		—
[HW] TARe [103]	new (ReRAM)	✗	✗	■	◧	■	✗	?		GL
[HW] Rubik [62]	new	✗	■ (mb)	✗	◧	?	✗	■ (v, e)	sh	LC (AU)
[HW] GCNAX [141]	new	✗	✗	■	◧	?	✗	?		GL
[HW] Li et al. [140]	new	✗	✗	■	■	■	■	■		LC (AU)
[HW] GReTA [126]	new	✗	✗	■	?	?	?	■	sh	LC (GReTA)
[HW] GNN-PIM [217]	new (PIM)	✗	✗	■	?	■	✗	■ (v)	sh	LC (SAGA)
[HW] EnGN [145]	new	✗	✗	■	◧	?	✗	■ (v, e)	sh	LC (AU + "feature extraction" stage)
[HW] HyGCN [228]	new	✗	✗	■	■	■	✗	■ (v, e)	sh	LC (AU)
[HW] AWB-GCN [85]	new	✗	✗	■	◧	■	■	■ (v, e)	sh	GL
[HW] GRIP [127]	new	✗	✗	■	?	■	◧	■ (v, e)	sh	GL, LC (GReTA)
[HW] Zhang et al. [235]	new	✗	✗	■	■	■	◧	■ (v, e)	sh	GL
[HW] GraphACT [233]	new	✗	■ (mb)	✗	◧	■	✗	?		GL
[HW] Auten et al. [7]	new	✗	✗	■	?	?	?	?		LC (AU)

... Check the paper ☺

Potential for future research – a lot of ideas on how to move on from here

8 Challenges & Opportunities

Many of the considered parts of the parallel and distributed GNN landscape were not thoroughly researched. Some were not researched at all. We now list such challenges and opportunities for future research.

MACIEJ BESTA, TORSTEN HOEFLER, ET AL.

Motif Prediction with Graph Neural Networks

Thank you for your attention

SPCL

Future Computing Laboratory