DeepMind

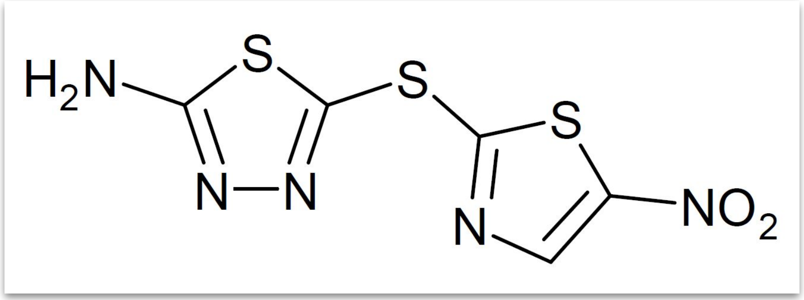
Everything is Connected

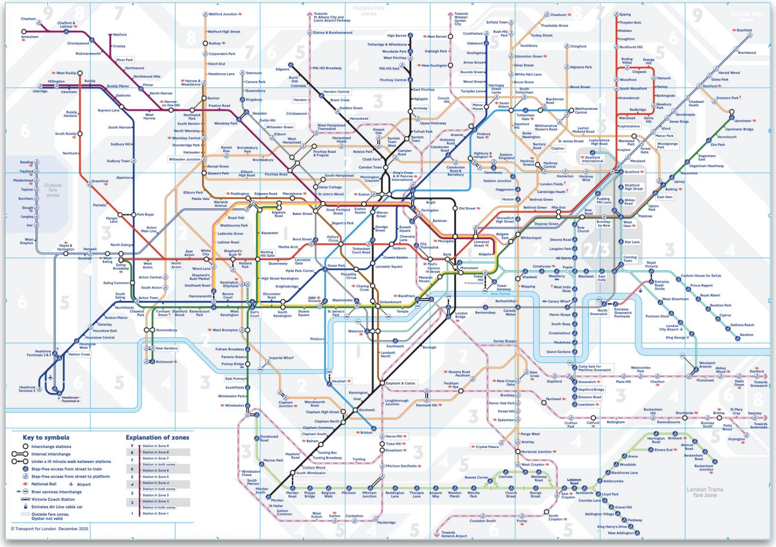
Deep Learning on Graphs

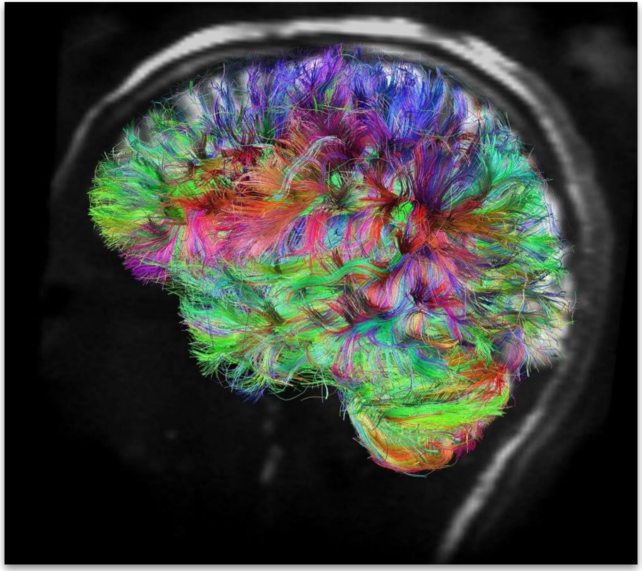
Petar Veličković

MLinPL x CUAI Cambridge Pre-meeting
30 October 2021

Graphs are **everywhere!**





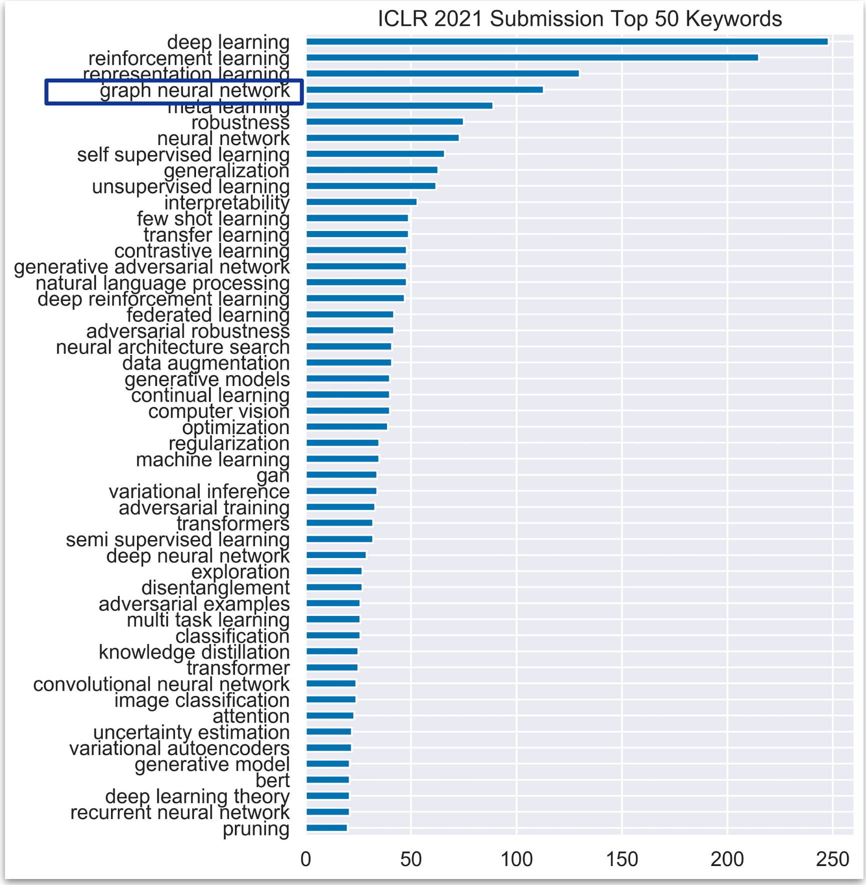


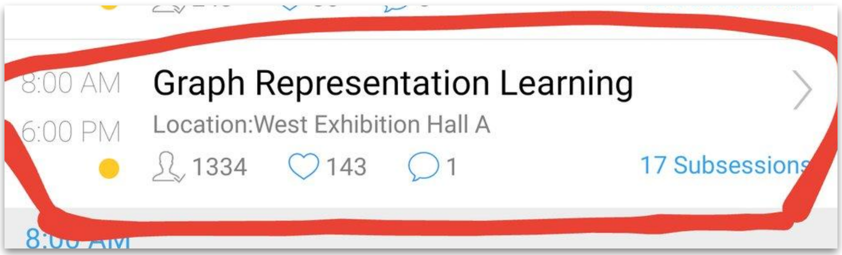
In many ways, graphs are the main modality of data we receive from **nature.**

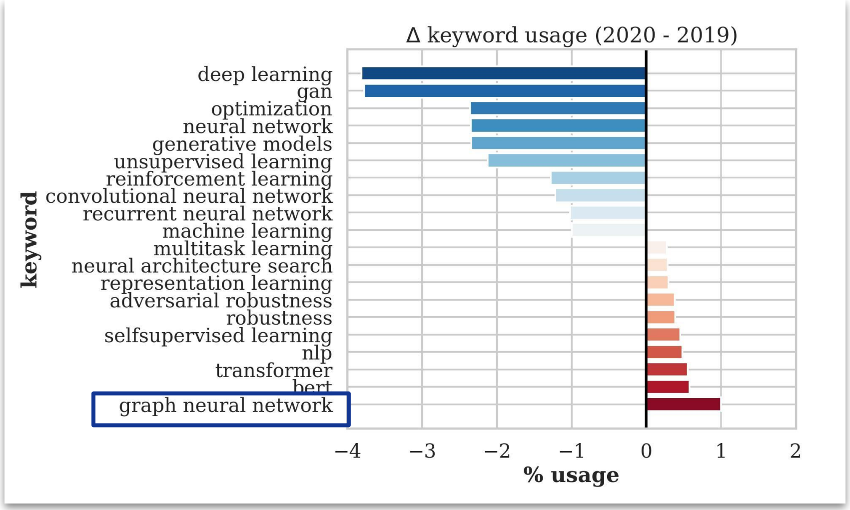
DeepMind

Graph representation learning is likely **critical** on the path to AGI.

A very hot research topic

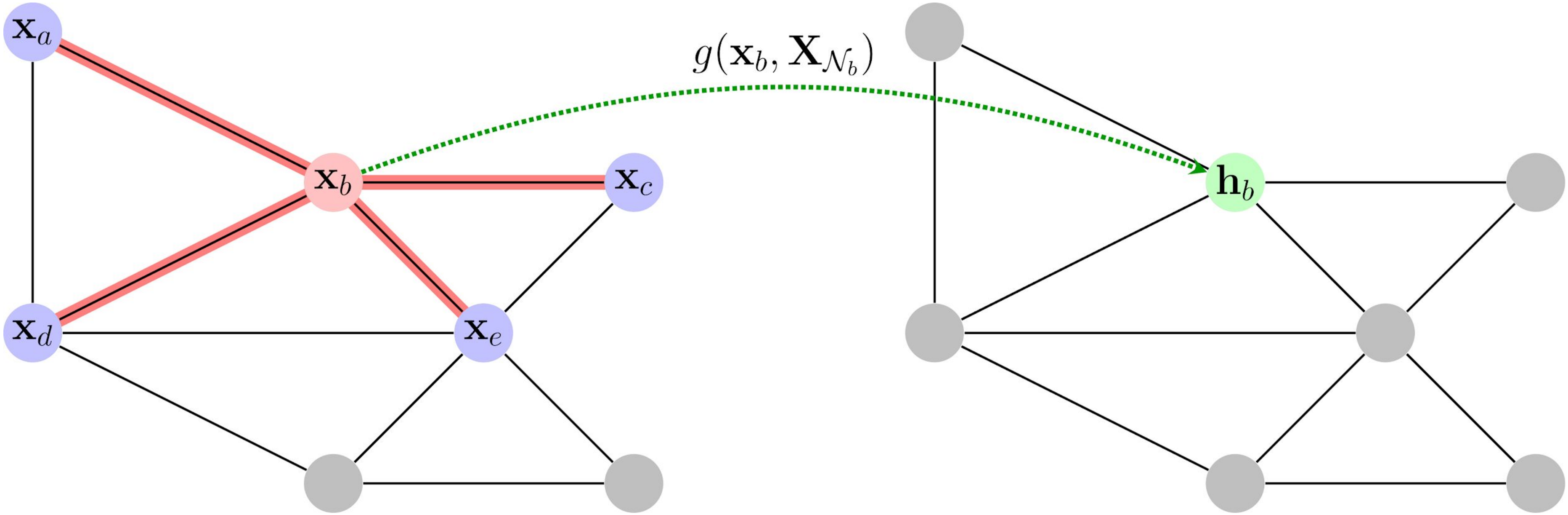






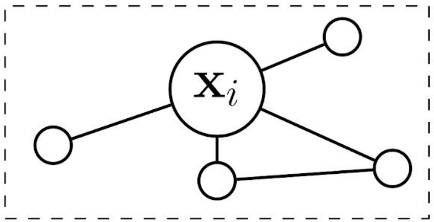
GRL is currently experiencing its "ImageNet" moment

Graph neural networks (GNNs)



$$\mathbf{X}_{\mathcal{N}_b} = \{\!\{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d, \mathbf{x}_e\}\!\}$$

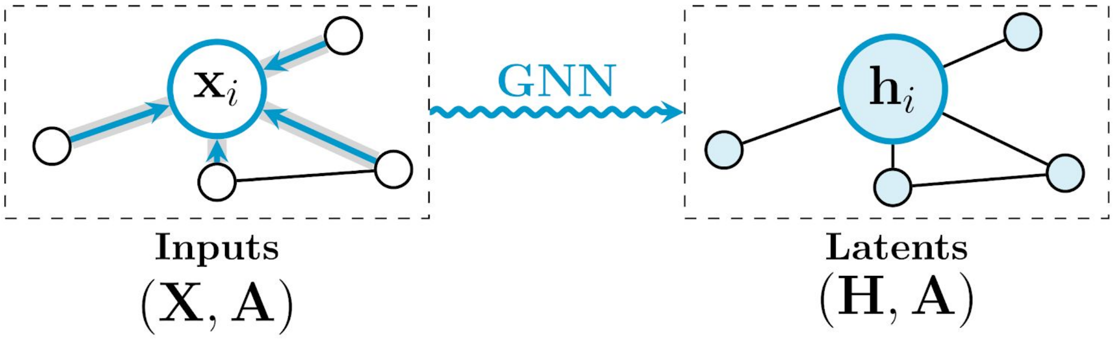
General blueprint for learning on **graphs**



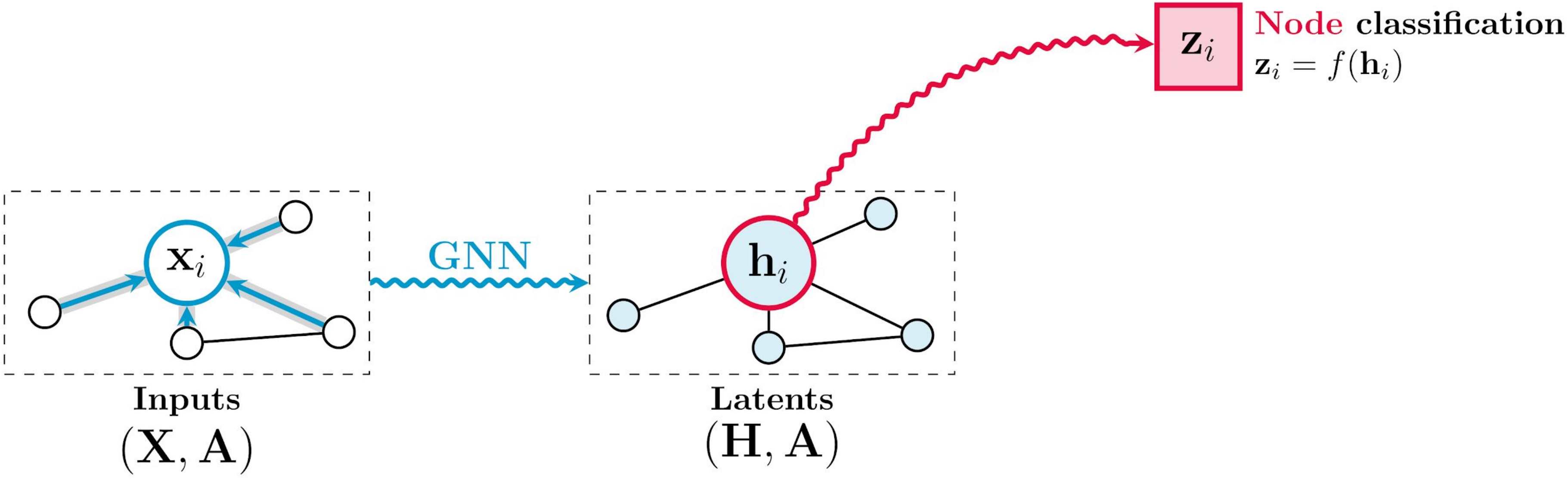
Inputs

$(\mathbf{X}, \mathbf{A})$

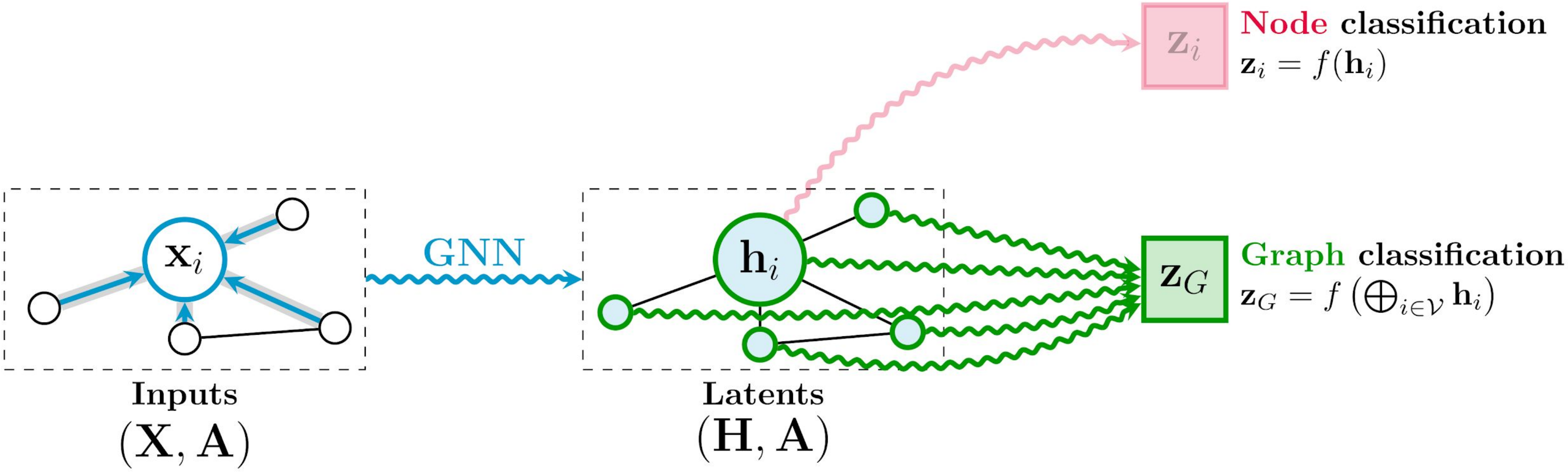
General blueprint for learning on **graphs**



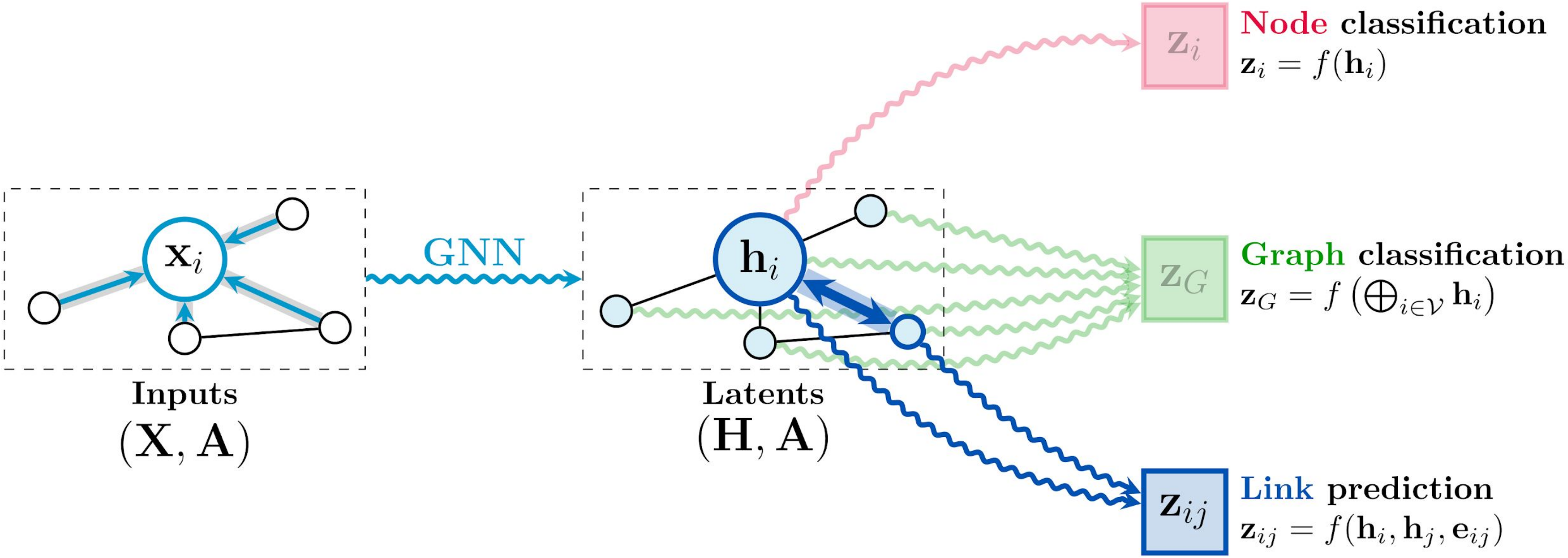
General blueprint for learning on **graphs**



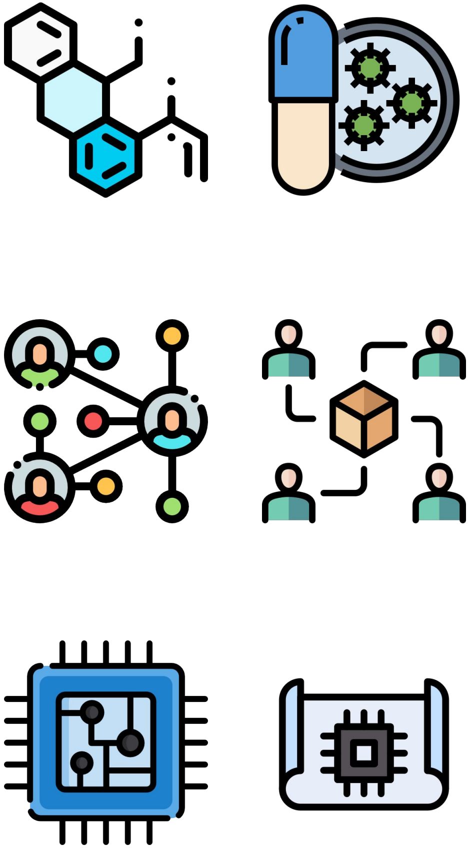
General blueprint for learning on **graphs**



General blueprint for learning on **graphs**

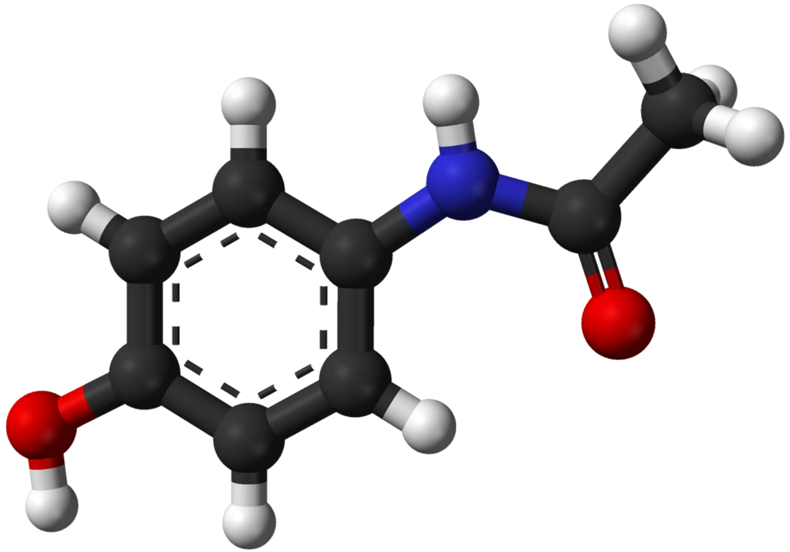


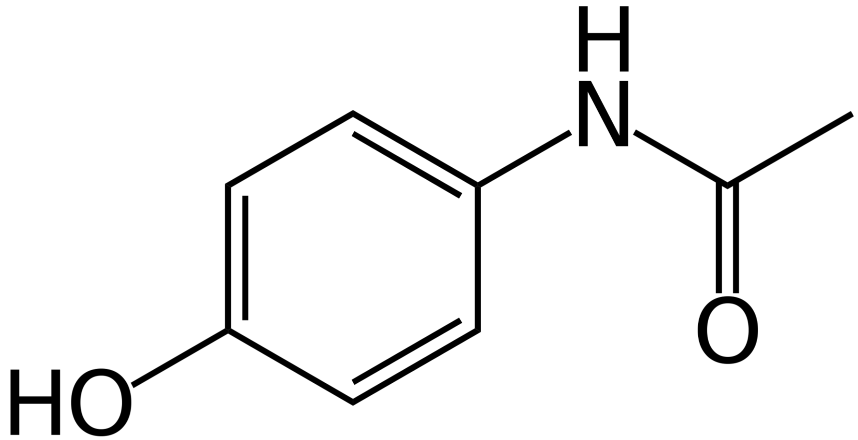
Impactful applications in **science** and **industry**



Molecules are graphs!

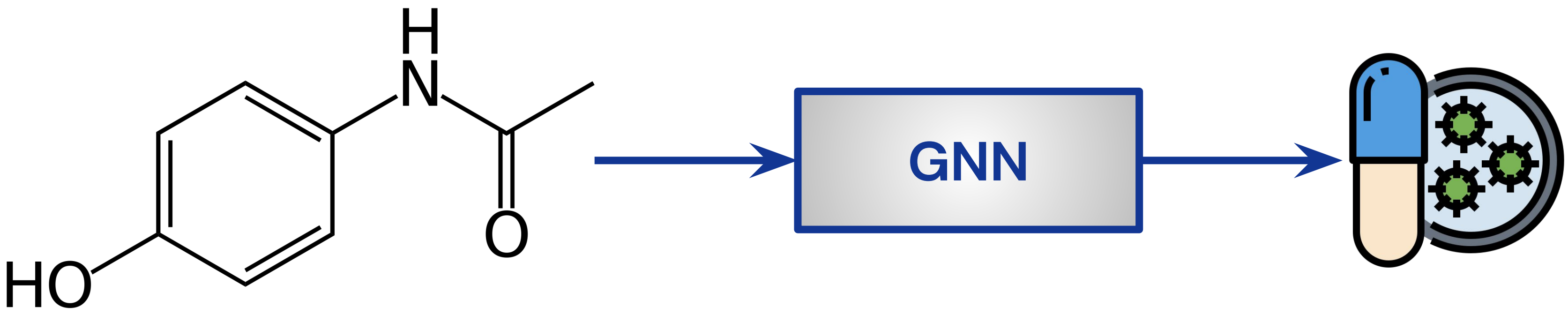
- A very natural way to represent molecules is as a **graph**
  - **Atoms** as nodes, **bonds** as edges
  - Features such as **atom type**, **charge**, **bond type**...





GNNs for molecule classification

- Interesting task to predict is, for example, whether the molecule is a potent **drug**.
  - Can do *binary classification* on whether the drug will inhibit certain bacteria. (*E.coli*)
  - Train on a **curated dataset** for compounds where response is known.



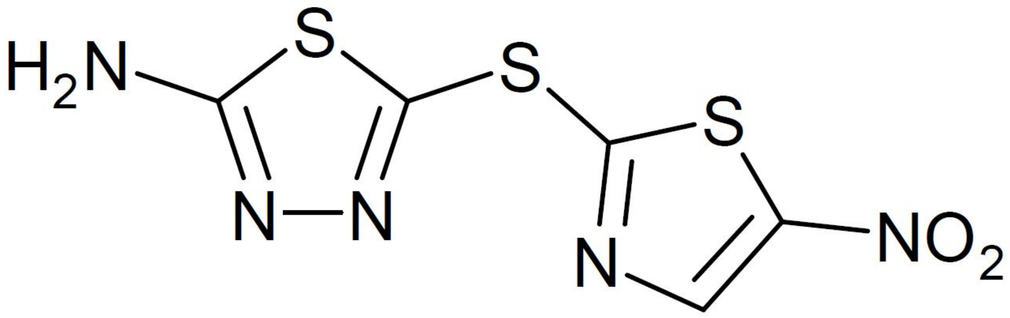
Molecule

Inhibits E.coli?

Follow-up study

- Once trained, the model can be applied to *any* molecule.
    - Execute on a large dataset of known candidate molecules.
    - Select the *~top-100* candidates from your GNN model.
    - Have chemists thoroughly investigate those (after some additional filtering).

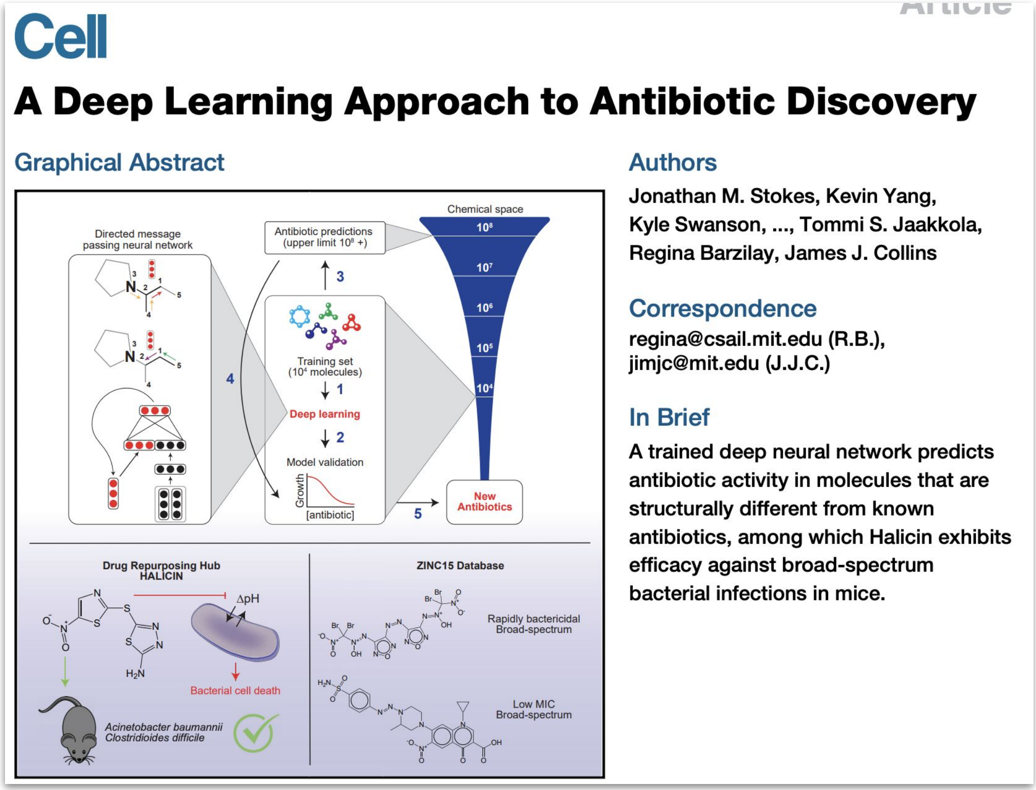
- Discover a previously overlooked compound that is a **highly potent** antibiotic!



Halicin

…Achieve wide acclaim!

Arguably the most popularised **success story** of graph neural networks to date!


Cell

Article

A Deep Learning Approach to Antibiotic Discovery

Graphical Abstract

Authors

Jonathan M. Stokes, Kevin Yang, Kyle Swanson, ..., Tommi S. Jaakkola, Regina Barzilay, James J. Collins

Correspondence

regina@csail.mit.edu (R.B.), jimjc@mit.edu (J.J.C.)

In Brief

A trained deep neural network predicts antibiotic activity in molecules that are structurally different from known antibiotics, among which Halicin exhibits efficacy against broad-spectrum bacterial infections in mice.

(Stokes *et al.*, Cell'20)

…Achieve wide acclaim!

Arguably the most popularised **success story** of graph neural networks to date!

Cell

Article

nature

Subscribe

NEWS · 20 FEBRUARY 2020

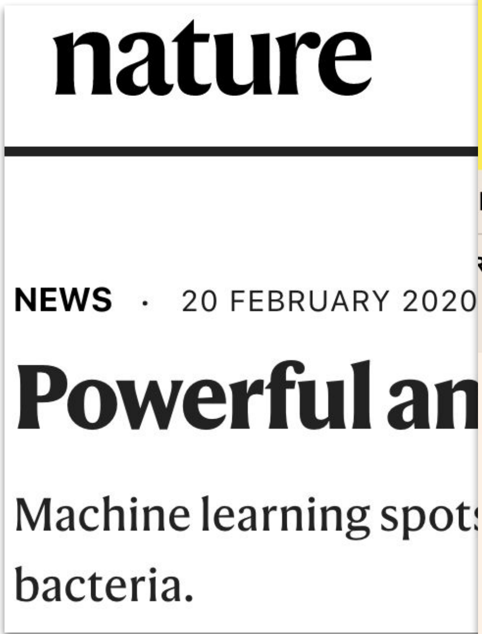
Powerful antibiotics discovered using AI

Machine learning spots molecules that work even against ‘untreatable’ strains of bacteria.

(Stokes *et al.*, Cell’20)

...Achieve wide acclaim!

Arguably the most popula


nature

NEWS · 20 FEBRUARY 2020

Powerful an

Machine learning spot
bacteria.


FINANCIAL TIMES

S COMPANIES TECH MARKETS GRAPHICS OPINION WORK & CAREERS LIFE & ARTS HOW TO SPEND IT

CORONAVIRUS **BUSINESS UPDATE**
Get 30 days' complimentary access to our Coronavirus Business Update newsletter

l intelligence

Robotics

'Death of the office' homeworking claims exaggerated

Anti-social robots harr increase social distanci

Artificial intelligence + Add to myFT

AI discovers antibiotics to treat drug-resistant diseases

Machine learning uncovers potent new drug able to kill 35 powerful bacteria

(Stokes *et al.*, Cell'20)

…Achieve wide acclaim!

Argua

BBC | Sign in | News | Sport | Reel | Worklife | Travel | Future

NEWS

Home | Video | World | UK | Business | Tech | Science | Stories | Entertainment & Arts

BBC WORKLIFE Our new guide for getting ahead

Scientists discover powerful antibiotic using AI

21 February 2020

Share

TIMES

S HOW TO SPEND IT

Anti-social robots harr increase social distanci

g-resistant

Machine learning uncovers potent new drug able to kill 35 powerful bacteria

n

NEWS

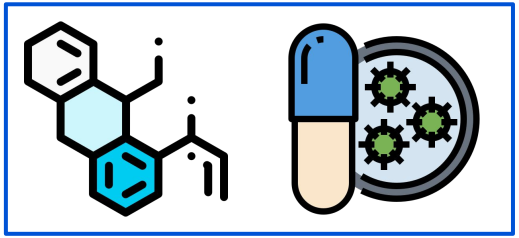
Po

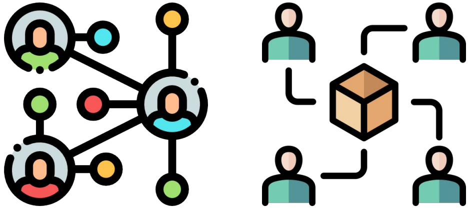
Mach

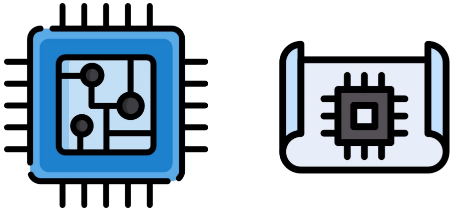
bacte

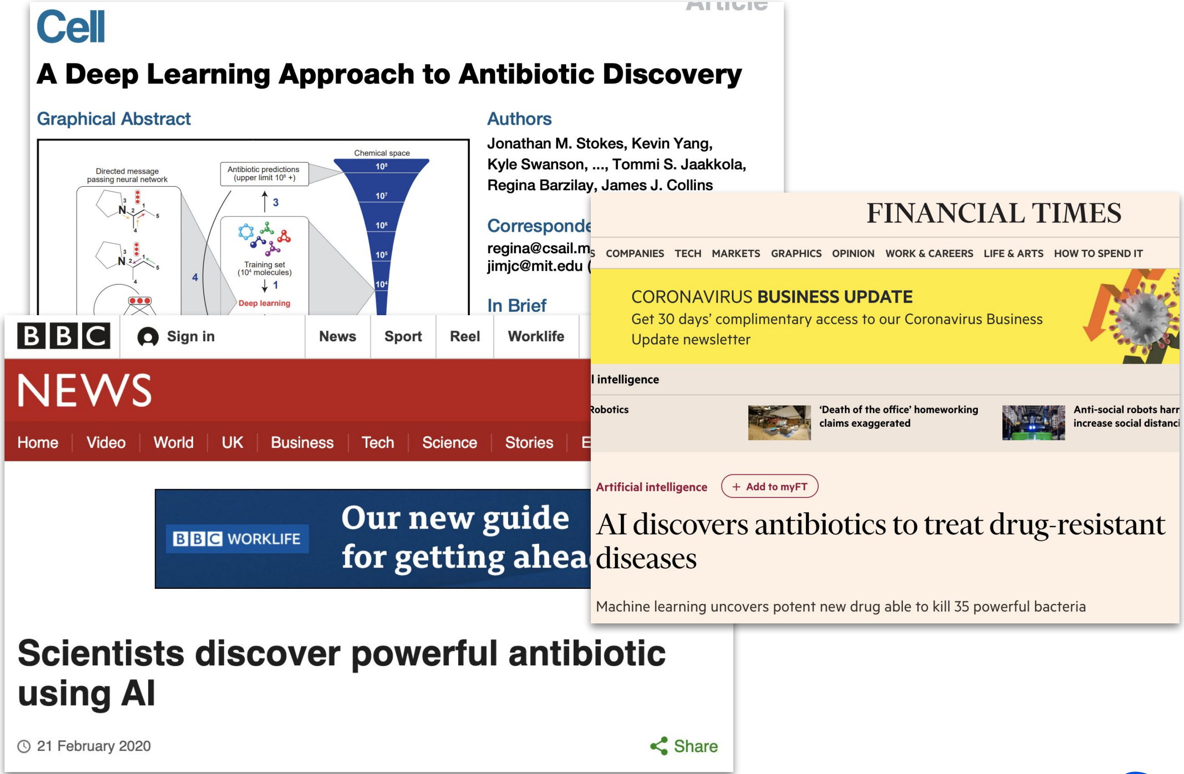
(Stokes *et al.*, Cell'20)

Impactful applications in **science** and **industry**









Cell

Article

A Deep Learning Approach to Antibiotic Discovery

Graphical Abstract

Authors

Jonathan M. Stokes, Kevin Yang, Kyle Swanson, ..., Tommi S. Jaakkola, Regina Barzilay, James J. Collins

Correspondence

regina@csail.m

jimjc@mit.edu

In Brief

BBC | Sign in | News | Sport | Reel | Worklife

NEWS

Home | Video | World | UK | Business | Tech | Science | Stories

BBC WORKLIFE Our new guide for getting ahea

Scientists discover powerful antibiotic using AI

21 February 2020

Share

FINANCIAL TIMES

COMPANIES TECH MARKETS GRAPHICS OPINION WORK & CAREERS LIFE & ARTS HOW TO SPEND IT

CORONAVIRUS **BUSINESS UPDATE**

Get 30 days' complimentary access to our Coronavirus Business Update newsletter

l intelligence

Robotics

'Death of the office' homeworking claims exaggerated

Anti-social robots harr increase social distanci

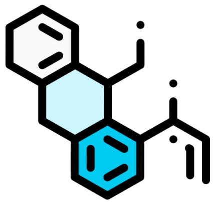
Artificial intelligence + Add to myFT

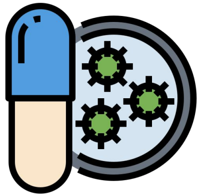
AI discovers antibiotics to treat drug-resistant diseases

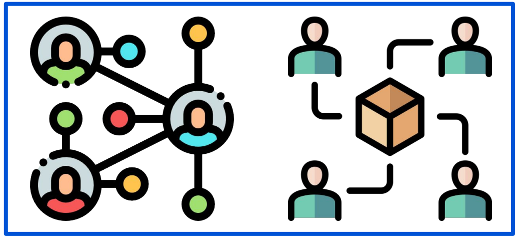
Machine learning uncovers potent new drug able to kill 35 powerful bacteria

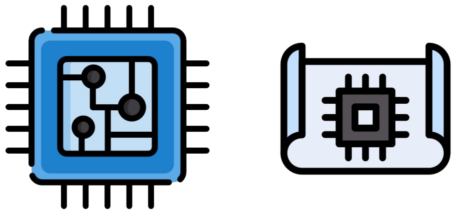
Virtual drug screening

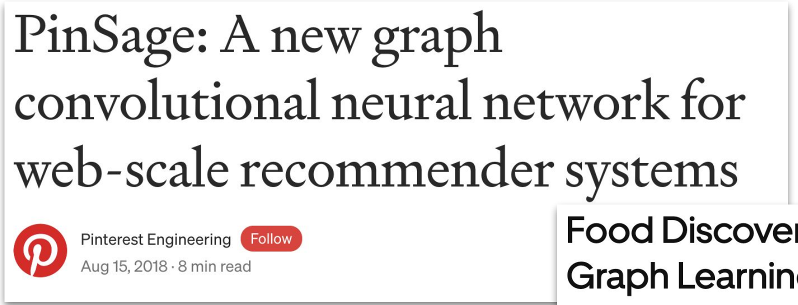
Impactful applications in **science** and **industry**







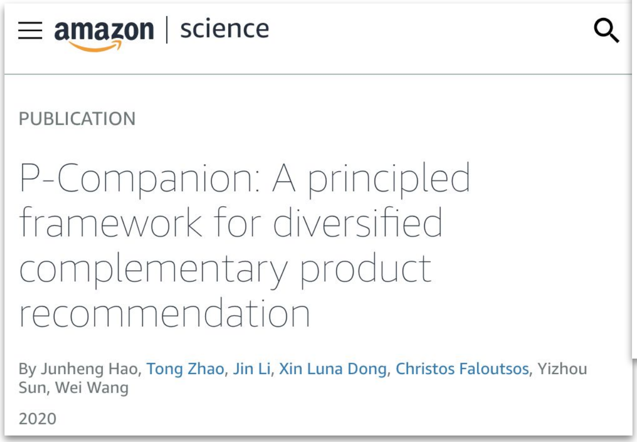




PinSage: A new graph convolutional neural network for web-scale recommender systems

Pinterest Engineering Follow

Aug 15, 2018 · 8 min read



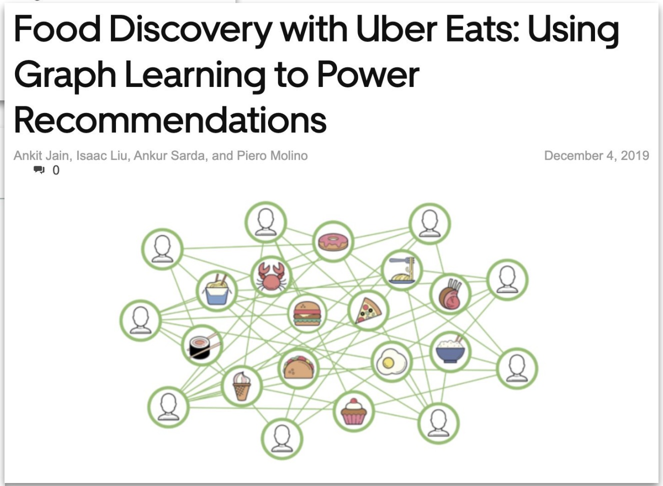
amazon | science

PUBLICATION

P-Companion: A principled framework for diversified complementary product recommendation

By Junheng Hao, Tong Zhao, Jin Li, Xin Luna Dong, Christos Faloutsos, Yizhou Sun, Wei Wang

2020



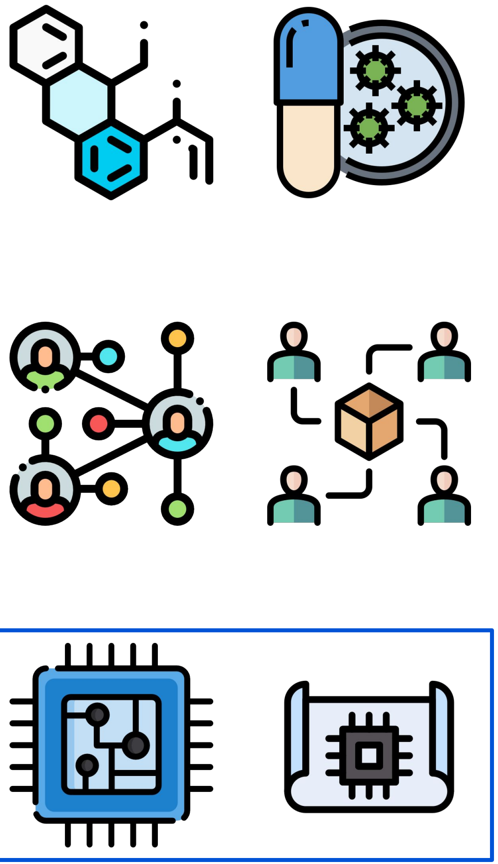
Food Discovery with Uber Eats: Using Graph Learning to Power Recommendations

Ankit Jain, Isaac Liu, Ankur Sarda, and Piero Molino   December 4, 2019

0

Recommender systems

Impactful applications in **science** and **industry**



nature

Explore content ⌄ Journal information ⌄ Publish with us ⌄ Subscribe

nature › articles › article

Article | Published: 09 June 2021

A graph placement methodology for fast chip design

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

GOOGLE \ TECH \ ARTIFICIAL INTELLIGENCE

Google is using AI to design its next generation of AI chips more quickly than humans can

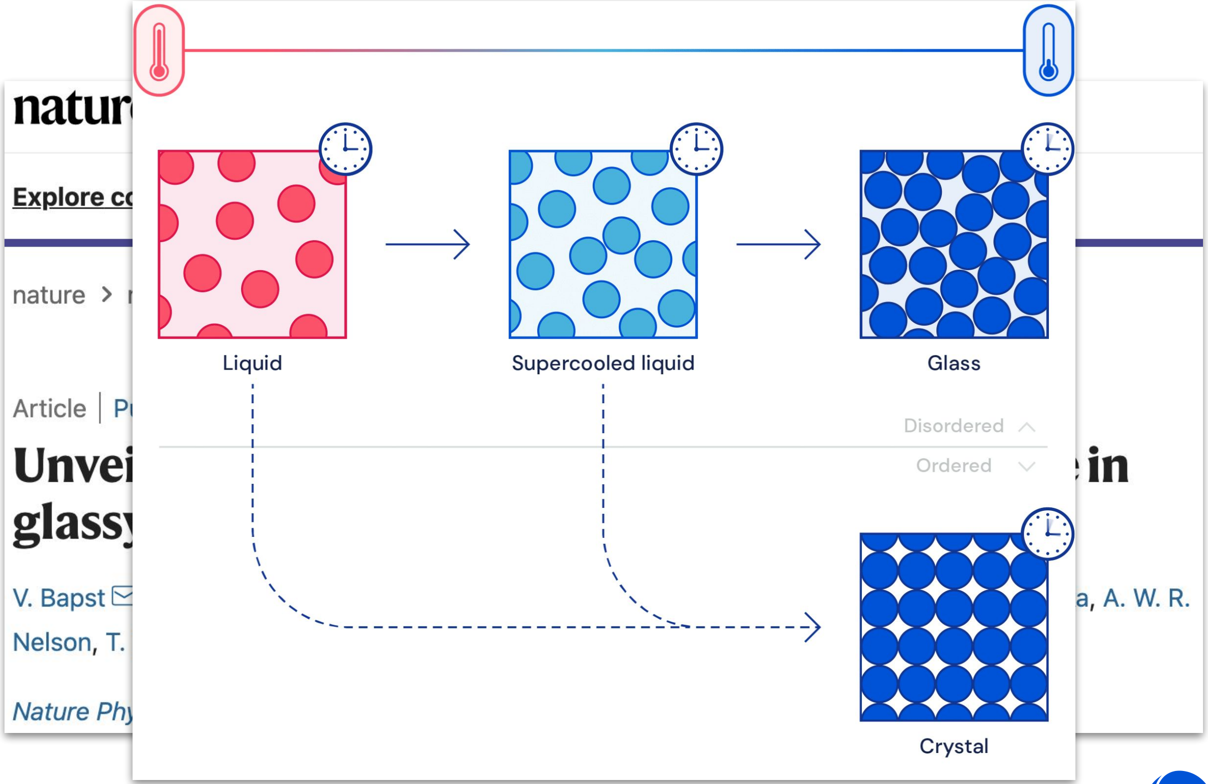
Designs that take humans months can be matched or beaten by AI in six hours

By James Vincent | Jun 10, 2021, 9:13am EDT

Chip design (TPUv5)

Impactful applications **from DeepMind**

Impactful applications **from DeepMind**

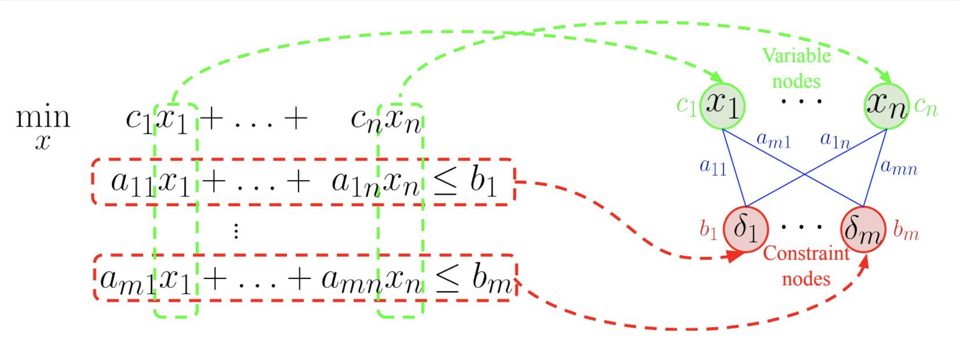


Glassy dynamics

Impactful applications **from DeepMind**

Solving Mixed Integer Programs Using Neural Networks

Vinod Nair[*†1], Sergey Bartunov[*1], Felix Gimeno[*1], Ingrid von Glehn[*1], Pawel Lichocki[*2], Ivan Lobov[*1], Brendan O'Donoghue[*1], Nicolas Sonnerat[*1], Christian Tjandraatmadja[*2], Pengming Wang[*1], Ravichandra Addanki[1], Tharindi Hapuarachchi[1], Thomas Keck[1], James Keeling[1], Pushmeet Kohli[1], Ira Ktena[1], Yujia Li[1], Oriol Vinyals[1], Yori Zwols[1]
[1]DeepMind, [2]Google Research



Combinatorial optimisation

Impactful applications **from DeepMind**

ETA Prediction with Graph Neural Networks in Google Maps

Austin Derrow-Pinion[1], Jennifer She[1], David Wong[2*], Oliver Lange[3], Todd Hester[4*], Luis Perez[5*], Marc Nunkesser[3], Seongjae Lee[3], Xueying Guo[3], Brett Wiltshire[1], Peter W. Battaglia[1], Vishal Gupta[1], Ang Li[1], Zhongwen Xu[6*], Alvaro Sanchez-Gonzalez[1], Yujia Li[1] and Petar Veličković[1]
[1]DeepMind [2]Waymo [3]Google [4]Amazon [5]Facebook AI [6]Sea AI Lab *work done while at DeepMind
{derrowap,jenshe,wongda,petarv}@google.com

VB The Machine GamesBeat Jobs Special Issue Become a Member | S

The Machine
Making sense of AI

DeepMind claims its AI improved Google Maps travel time estimates by up to 50%

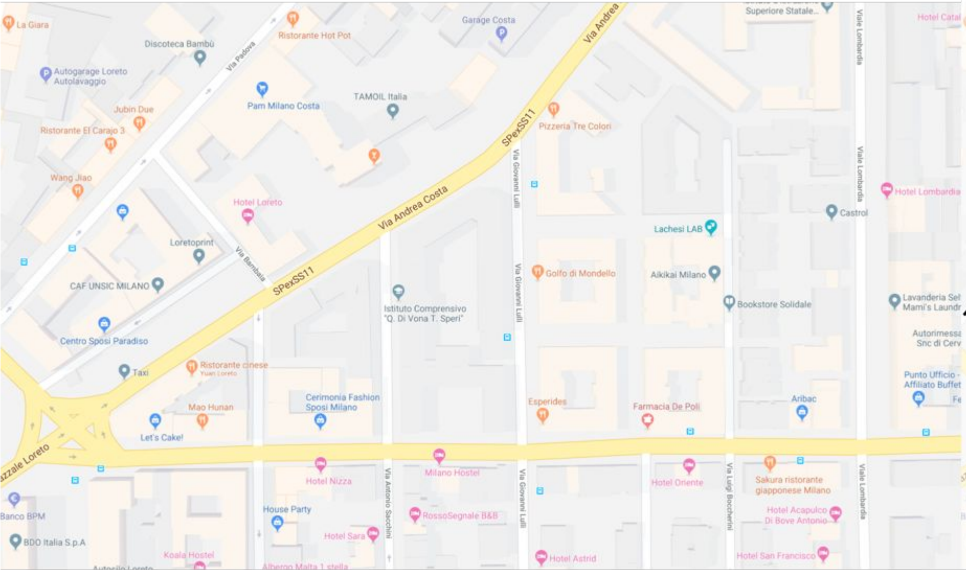
Kyle Wiggers @Kyle_L_Wiggers September 3, 2020 7:00 AM

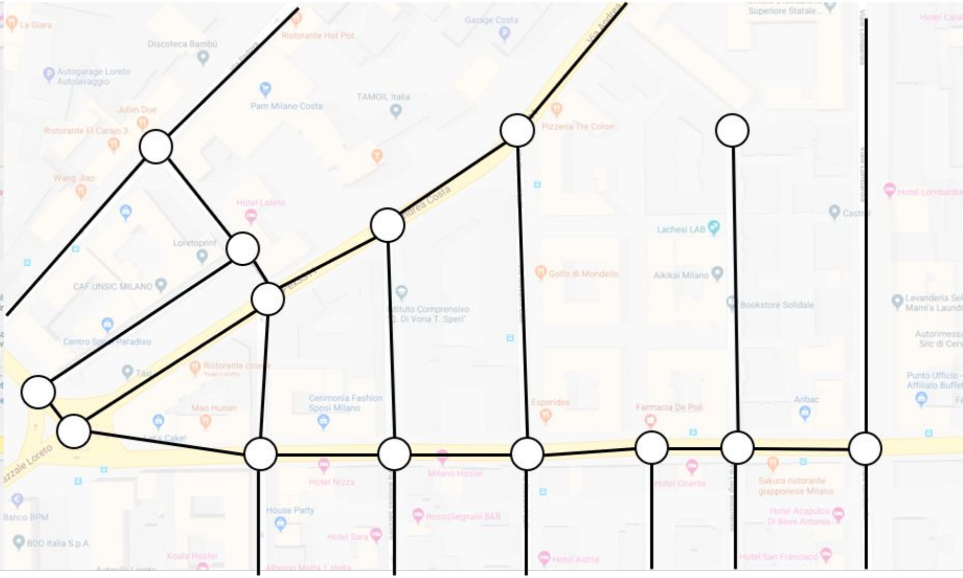
Travel-time Prediction in Google Maps

Enabling scalable traffic predictions with GNNs in Google Maps

Traffic maps are graphs!

Transportation maps (e.g. the ones found on *Google Maps*) naturally modelled as **graphs**.





Nodes could be **intersections**, and edges could be **roads**.

Estimated Time of Arrival (ETA) Prediction

- A critical service provided by Google Maps is **ETA prediction**.
  - Given a start-point and end-point, what is the expected travel time?
  - Important for both **users** and **ride-sharing/delivery** companies (using the Maps API).

- Relevant **node features**: road *length*, *current speeds*, *historical speeds*

- Use anonymised, crowd-sourced real-time / historical traffic data.
  - Not as reliable as e.g. physical speed sensors
  - Traffic conditions change dynamically and unpredictably
  - Most trips between [10min, 1h], requiring **near-future predictions**

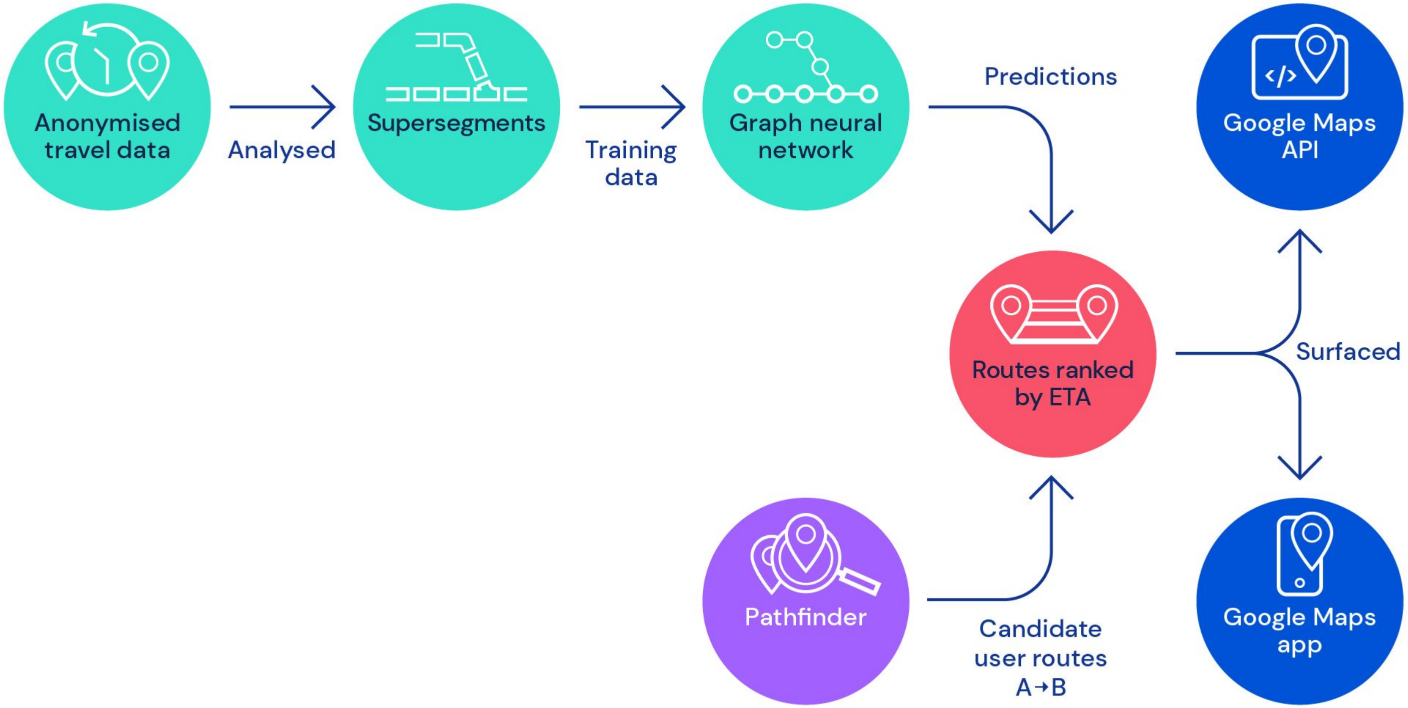
DeepMind's approach: **Graph Nets** on **Supersegments**

Partition candidate route into **supersegments**, sampled proportionally to (est.) **traffic density**.

Run a GNN over **supersegment** graph to estimate ETA (*graph regression*).

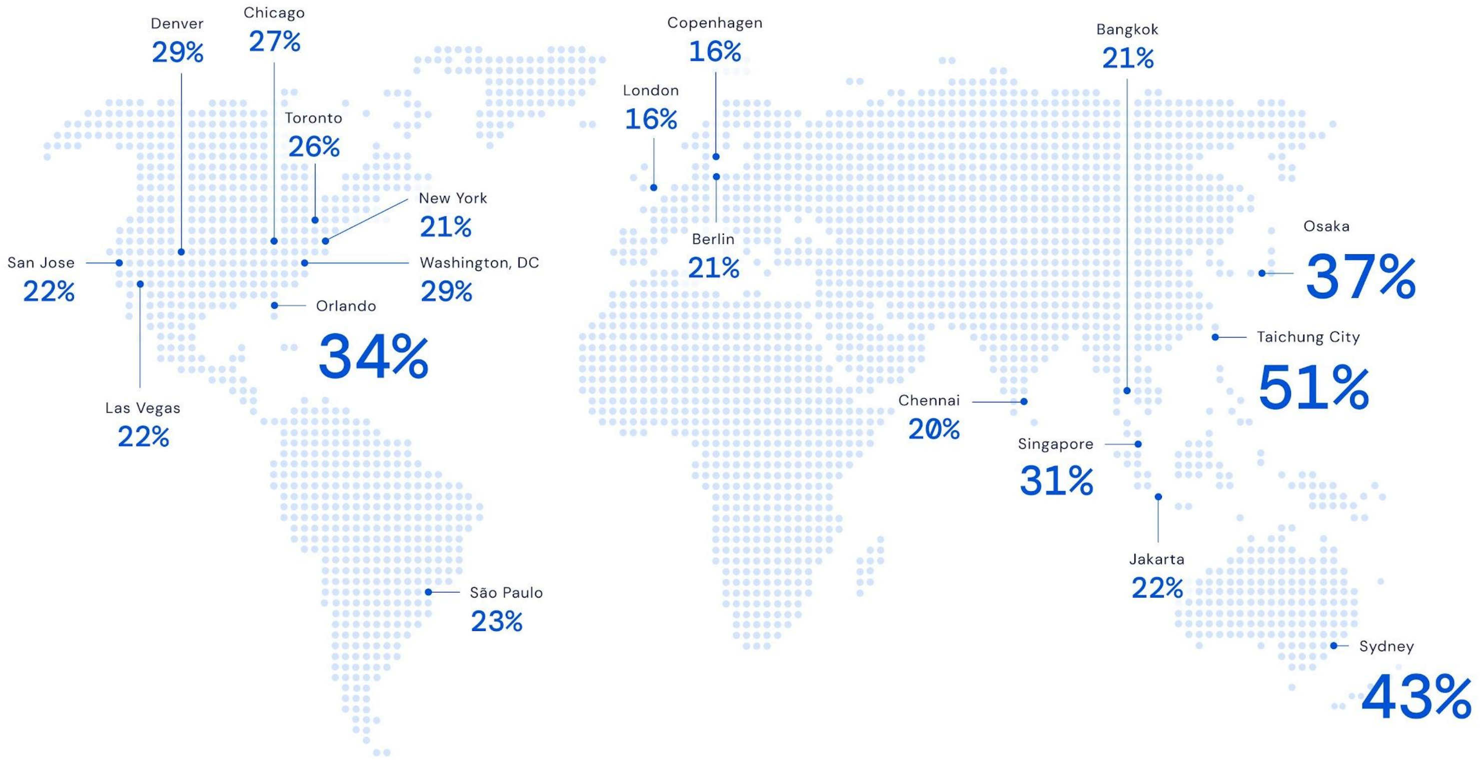
Overall pipeline

Rank candidate routes by predicted ETA, surface information to Google Maps.



Returns

Already **deployed** worldwide, significantly reducing negative ETA outcomes!



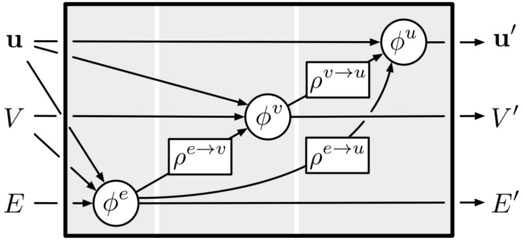
I Getting in on the action

Rich **ecosystem** of **libraries**

github.com/rusty1s/pytorch_geometric

graphneural.network

dgl.ai



github.com/deepmind/graph_nets

github.com/deepmind/jraph

Rich **ecosystem** of **datasets**

OGB

ogb.stanford.edu

PyTorch geometric

https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html

TUDataset

graphlearning.io

Benchmarking Graph Neural Networks

github.com/graphdeeplearning/benchmarking-gnns

Getting into it!

- I recently compiled a list of many useful GNN resources in a **Twitter thread**
  - https://twitter.com/PetarV_93/status/1306689702020382720
- When you feel ready, I **highly** recommend Aleksa Gordić's GitHub repository on GATs:
  - https://github.com/gordicaleksa/pytorch-GAT
  - Arguably the most *gentle* introduction to GNN implementations

DeepMind

II

Graph Isomorphism Testing

How *powerful* are Graph Neural Networks?

- GNNs are a powerful tool for processing real-world graph data
  - But they won't solve ***any*** task specified on a graph accurately!

- Canonical example: deciding *graph isomorphism*
  - Am I able to use my GNN to **distinguish** two *non*-isomorphic graphs? ($\mathbf{h}_{G1} \neq \mathbf{h}_{G2}$)
  - If I can't, any kind of task discriminating them is *hopeless*

- We will assess the **power** of GNNs by *which graphs they are able to **distinguish***.

Weisfeiler-Lehman Test

- Simple but powerful way of distinguishing: pass **random hashes** of **sums** along the edges
  - Iterate until hashes don't change.
  - *"Possibly isomorphic"* if hash histograms are the same.

УДК 519.1

ПРИВЕДЕНИЕ ГРАФА К КАНОНИЧЕСКОМУ ВИДУ И ВОЗНИКАЮЩАЯ ПРИ ЭТОМ АЛГЕБРА

Б. Ю. ВЕЙСФЕЙЛЕР, А. А. ЛЕМАН

Рассматривается алгоритм приведения заданного конечного мультиграфа Г к каноническому виду. В процессе такого приведения возникает новый инвариант графа — алгебра $\mathfrak{A}$ (Г). Изучение свойств алгебры $\mathfrak{A}$ (Г) оказывается полезным при решении некоторых задач теории графов.

Выдвигаются и обсуждаются некоторые предположения относительно связи между свойствами алгебры $\mathfrak{A}$ (Г) и группой автоморфизмов графа Aut (Г). Построен пример неориентированного графа Г, алгебра $\mathfrak{A}$ (Г) которого совпадает с групповой алгеброй некоторой некоммутативной группы.

An algorithm is considered, reducing the specified finite multigraph Г to canonical form. In the course of this reduction, a new invariant of the graph is generated — algebra $\mathfrak{A}$ (Г). Study of the properties of the algebra $\mathfrak{A}$ (Г) proves helpful in solving a number of graph-theoretic problems. Some propositions concerning the relationships between the properties of the algebra $\mathfrak{A}$ (Г) and the graph's automorphism group *Aut* (Г) are discussed. An example of non-oriented graph Г is constructed whose algebra $\mathfrak{A}$ (Г) coincides with the group algebra of a non-commutative group.

1. Рассмотрим произвольный конечный граф Г и его матрицу смежности $A(\Gamma)=\{a_{ij}\}$; здесь a_{ij} — число ребер, ведущих из i-й вершины графа в j-ую; $i, j=1, 2, \ldots, n$. В случае неориентированного графа полагаем $a_{ij}=a_{ji}$. Каноническим видом графа мы будем называть его матрицу смежности при канонической нумерации вершин, т. е. при таком частичном упорядочении множества вершин, при котором из того, что a и b несравнимы следует, что существует автоморфизм графа, переводящий вершину a в b и сохраняющий отношение смежности.

В п. 6, 7 описан процесс приведения графа к каноническому виду, состоящий в поэтапном переупорядочении строк и столбцов матрицы A (Г), который, грубо говоря, сводится к следующему.

Рассмотрим для простоты неориентированный граф без кратных связей. Сначала каждой вершине графа сопоставим характеристический вектор, единственная компонента которого равна числу соседей данной вершины. Затем разобьем вершины на классы, так чтобы вершины с одинаковым характеристическим вектором принадлежали одному классу; классы при этом упорядочим в соответствии с естественным порядком в множестве характеристических векторов. Далее, каждой вершине сопоставим характеристический вектор $v_i=\{l, v_{i1}, v_{i2}, \ldots\}$, где v_{ik} — число соседей k-го класса у i-ой вершины, l — номер класса, к которому принадлежит i-ая вершина. Теперь снова разобьем вершины на классы в соответствии с новыми характеристическими векторами, упорядоченными лексикографически, и т. д. Заметим, что если вершины a и b на некотором шаге принадлежали разным классам и было выполнено условие $a<b$, то и в дальнейшем это условие всегда будет выполнено. Отсюда следует, что описанный процесс обрывается не позже, чем через n шагов — и либо все вершины относятся к разным классам (т. е. построена каноническая нумерация), либо дальнейшего разбиения на классы не происходит.

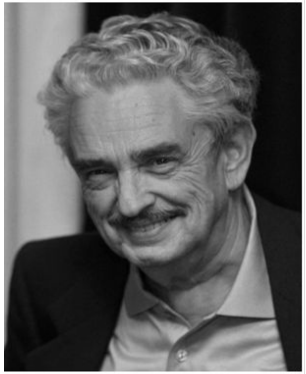
В случае, если Г — ориентированный мультиграф, возьмем в качестве характеристического вектора v_i упорядоченную i-ую строку матрицы A (Г) (считая при этом, что диагональный элемент предшествует всем остальным). Вместо различных элементов a_{ij} введем различные независимые переменные $x_1, x_2, \ldots$, упорядочив их в соответствии с порядком среди a_{ij}. Полученную таким способом матрицу обозначим $X=X(\Gamma)$. При очередном разбиении вершин на классы к одному классу, как и прежде, отнесем вершины с одинаковым характеристическим вектором; при этом k-ая компонента вектора v_i есть по определению сумма элементов i-ой строки матрицы X (Г), соответствующих вершинам k-го класса (предыдущего разбиения). Матрица X (Г) разбивается, таким образом, на блоки, в каждом из которых мы можем ввести свои независимые переменные и т. д. (точное определение этих операций см. п. 6, операции α_1, β_1).

Заметим, что описанная до сих пор процедура аналогична методам, изложенным в [1] и [2].

Для дальнейшего разбиения вершин на классы рассмотрим элемент u_{ij} матрицы $U=X\cdot X'$, где X' — матрица, полученная из X заменой переменных $x_1, x_2, \ldots$ переменными $x_1', x_2', \ldots$, причем все переменные $x_1, x_2, \ldots, x_1', x_2', \ldots$ независимы. Элемент u_{ij} является многочленом второй степени от $x_1, x_2, \ldots$

12

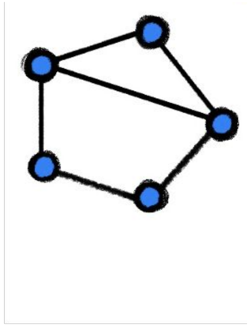
НТИ . СЕР. 2 . № 9 . 1968 . ИНФОРМАЦИОННЫЙ АНАЛИЗ



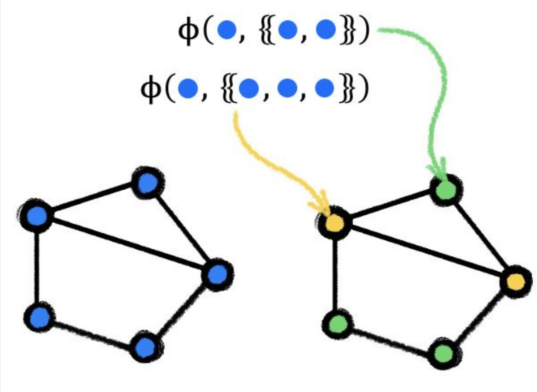
A. Lehman

B. Weisfeiler

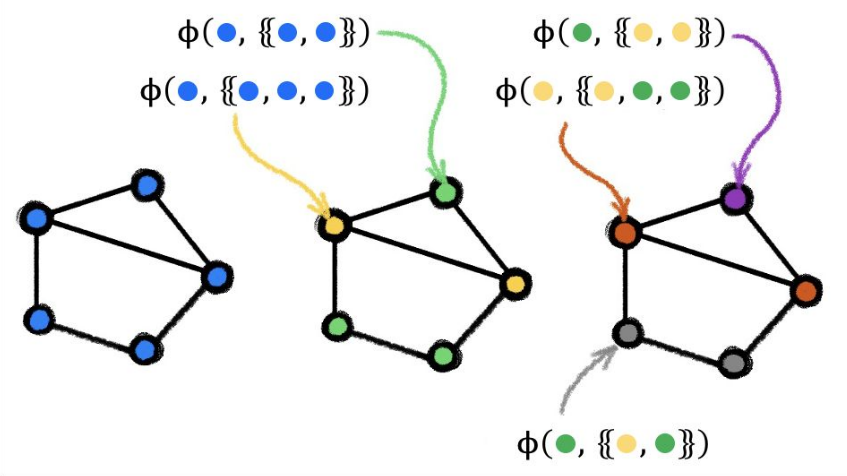
Let’s run the WL Test!



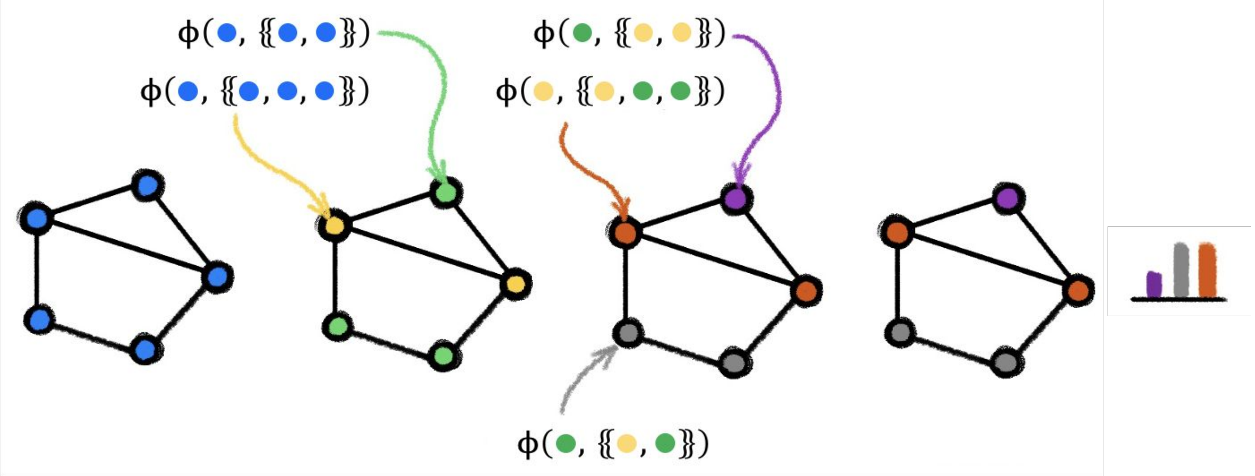
Let’s run the WL Test!



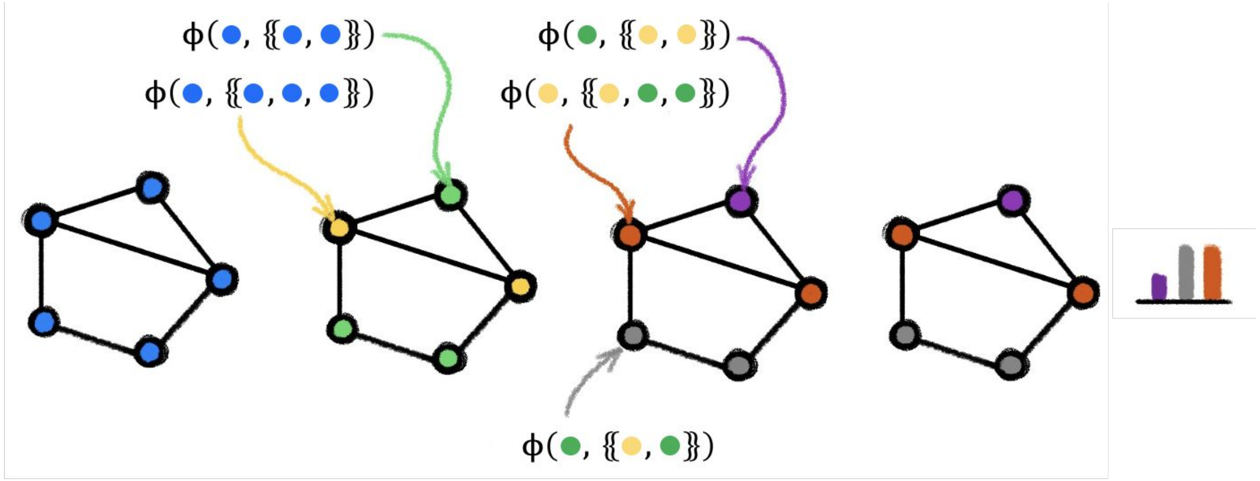
Let’s run the WL Test!

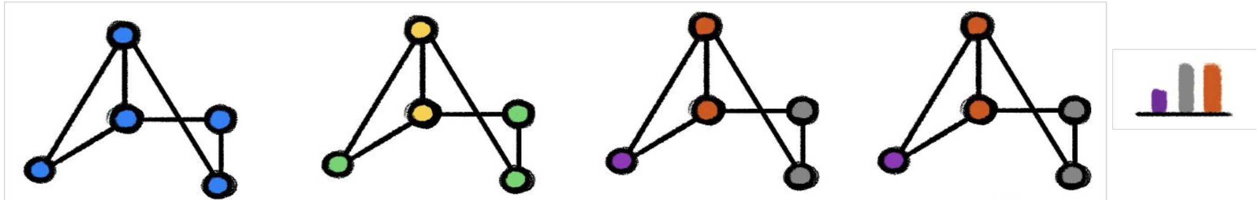


Let’s run the WL Test!



Let’s run the WL Test!





Weisfeiler-Lehman Test

- Connection to conv-GNNs spotted very early; e.g. by GCN (Kipf & Welling, ICLR'17)

- **Untrained** GNNs can hence work very well!
  - Untrained ~ random hash

- The test does **fail** at times, however:

Algorithm 1: WL-1 algorithm (Weisfeiler & Lehmann, 1968)

Input: Initial node coloring $(h_1^{(0)}, h_2^{(0)}, ..., h_N^{(0)})$
Output: Final node coloring $(h_1^{(T)}, h_2^{(T)}, ..., h_N^{(T)})$
$t \leftarrow 0$;
repeat
  for $v_i \in \mathcal{V}$ **do**
    $h_i^{(t+1)} \leftarrow \boxed{\text{hash}\left(\boxed{\sum_{j \in \mathcal{N}_i}} \boxed{h_j^{(t)}}\right)}$;
  $t \leftarrow t + 1$;
until *stable node coloring is reached*;

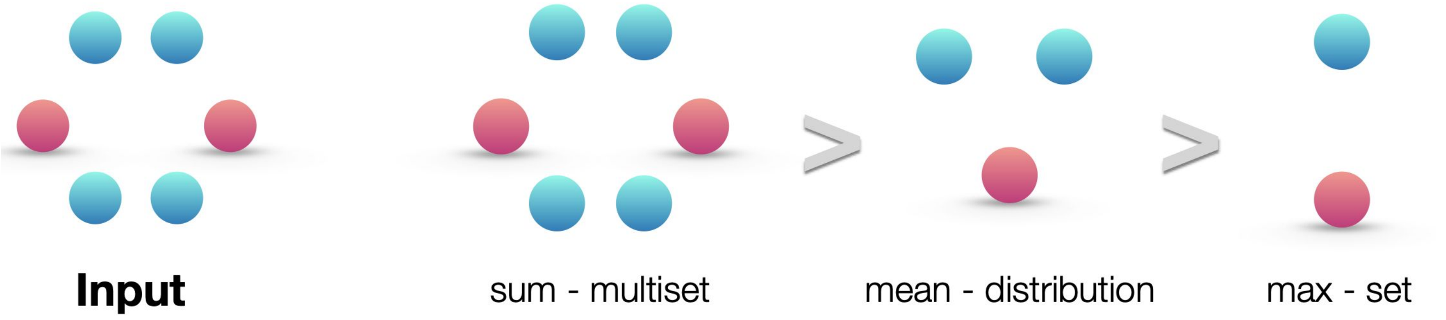
$$\mathbf{h}_i = \boxed{\phi\left(\mathbf{x}_i, \boxed{\bigoplus_{j \in \mathcal{N}_i}} \boxed{c_{ij}\psi(\mathbf{x}_j)}\right)}$$

GNNs are **no more powerful** than 1-WL

- Over ***discrete*** *features*, GNNs can only be **as powerful** as the 1-WL test described before!

GNNs are **no more powerful** than 1-WL

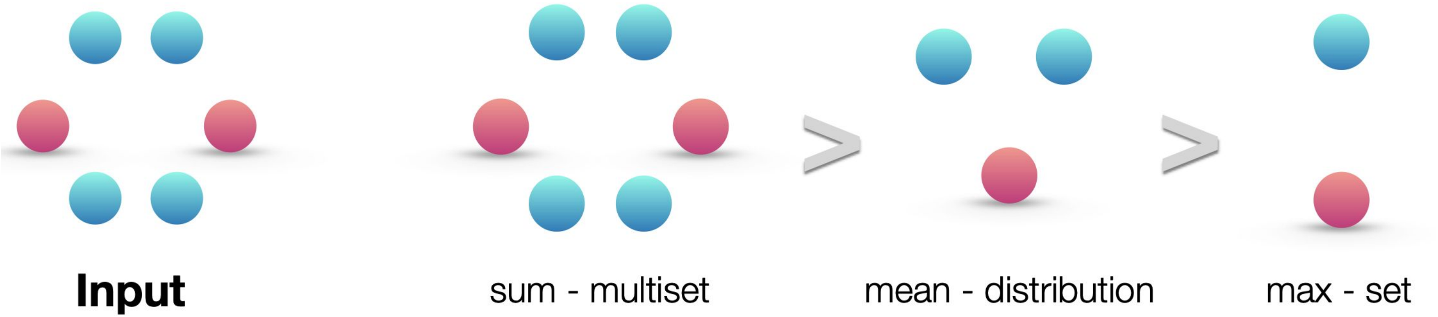
- Over ***discrete*** *features*, GNNs can only be **as powerful** as the 1-WL test described before!
- One important condition for maximal power is an ***injective*** aggregator (e.g. **sum**)



GNNs are **no more powerful** than 1-WL

- Over ***discrete*** *features*, GNNs can only be **as powerful** as the 1-WL test described before!

- One important condition for maximal power is an ***injective*** aggregator (e.g. **sum**)



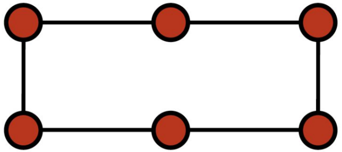
- Graph isomorphism network (**GIN**; Xu *et al.*, ICLR'19) proposes a simple, maximally-expressive GNN, following this principle:

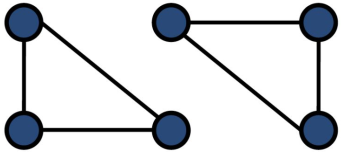
$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1 + \epsilon^{(k)}\right) \cdot h_v^{(k-1)} + \sum\nolimits_{u \in \mathcal{N}(v)} h_u^{(k-1)}\right)$$

Higher-order GNNs

- We can make GNNs stronger by analysing **failure cases** of 1-WL!
    - Very active area, with many open problems!

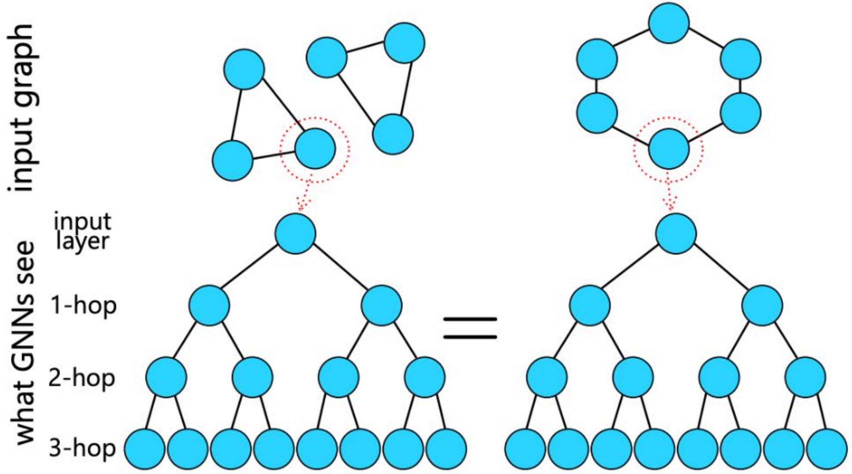
Higher-order GNNs



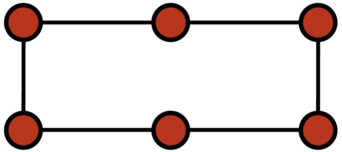


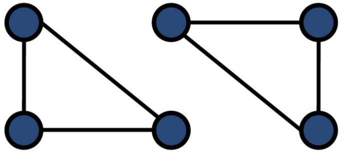
- We can make GNNs stronger by analysing **failure cases** of 1-WL!

- For example, just like 1-WL, GNNs cannot detect **closed triangles**
  - This is because, from a GNN's perspective, all nodes look the same!
  - Can you think of a simple fix?



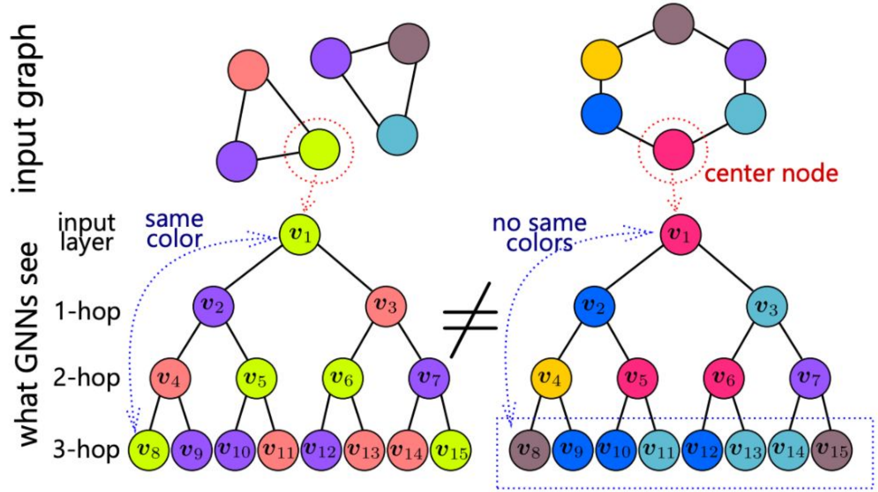
Higher-order GNNs





- We can make GNNs stronger by analysing **failure cases** of 1-WL!

- For example, just like 1-WL, GNNs cannot detect **closed triangles**
  - Augment nodes with **randomised** features (Sato *et al.*, SDM'21)
  - Now a node can "see itself" *k* hops away!



Higher-order GNNs

- We can make GNNs stronger by analysing **failure cases** of 1-WL!

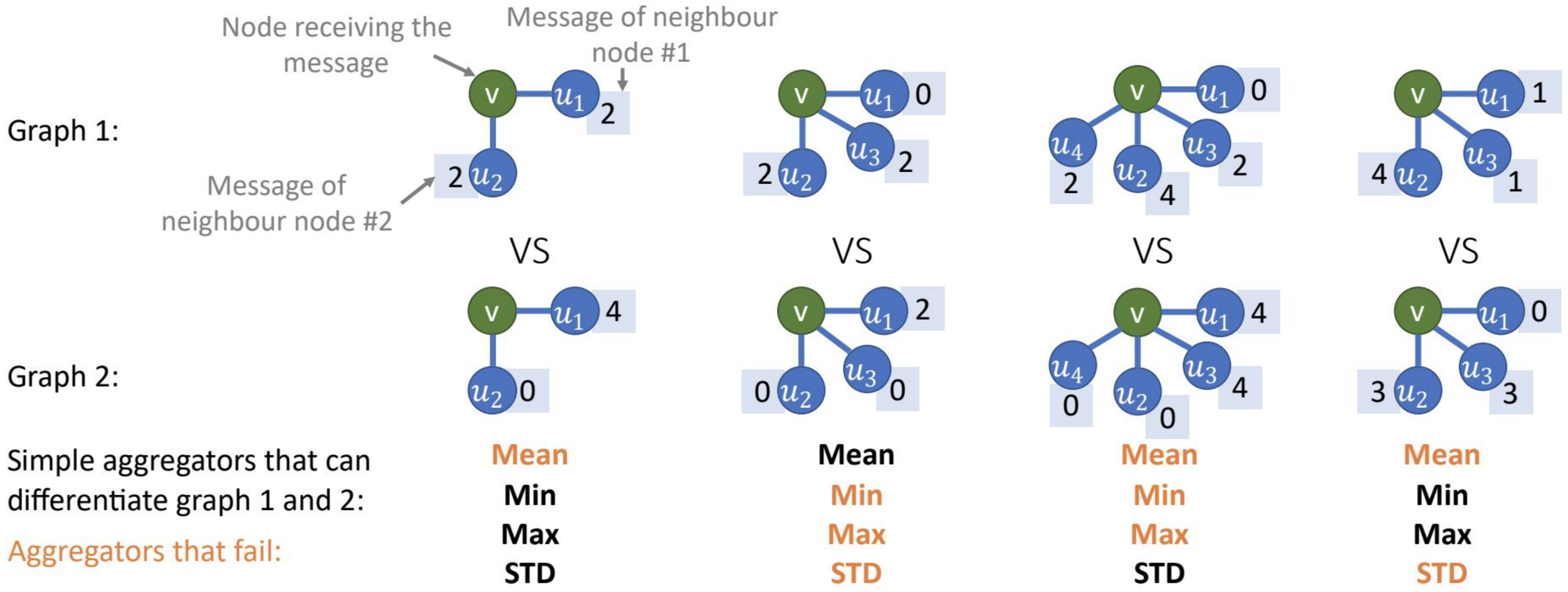
- For example, just like 1-WL, GNNs cannot detect **closed triangles**
  - Augment nodes with randomised/positional features (Sato *et al.*, SDM'21)
    - Explored by RP-GNN (Murphy *et al.*, ICML'19) and P-GNN (You *et al.*, ICML'19)
  - Can also literally **count** interesting subgraphs (Bouritsas *et al.*, 2020)

- Fixing "failure cases" of 1-WL yields many classes of *higher-order GNNs*

- They can broadly be categorised into three groups:
  - Modifying **features** (as above)
  - Modifying the **message passing rule**; e.g. DGN (Beaini, Passaro *et al.* (2020))
  - Modifying the **graph** structure; e.g. 1-2-3-GNNs (Morris *et al.*, AAAI'19)

Going beyond *discrete* features

- What happens when features are **continuous**? (real-world apps / latent GNN states)
  - … the proof for injectivity of sum (hence GINs' expressivity) **falls apart**



Which is best? **Neither.**

- There doesn't seem to be a clear single "winner" aggregator here…

- In fact, we prove in the PNA paper (Corso, Cavalleri *et al.*, NeurIPS'20) that **there isn't one**!

Theorem 1 (Number of aggregators needed). *In order to discriminate between multisets of size n whose underlying set is $\mathbb{R}$, at least n aggregators are needed.*

- The proof is (in my opinion) **really cool**! (relies on **Borsuk-Ulam** theorem)

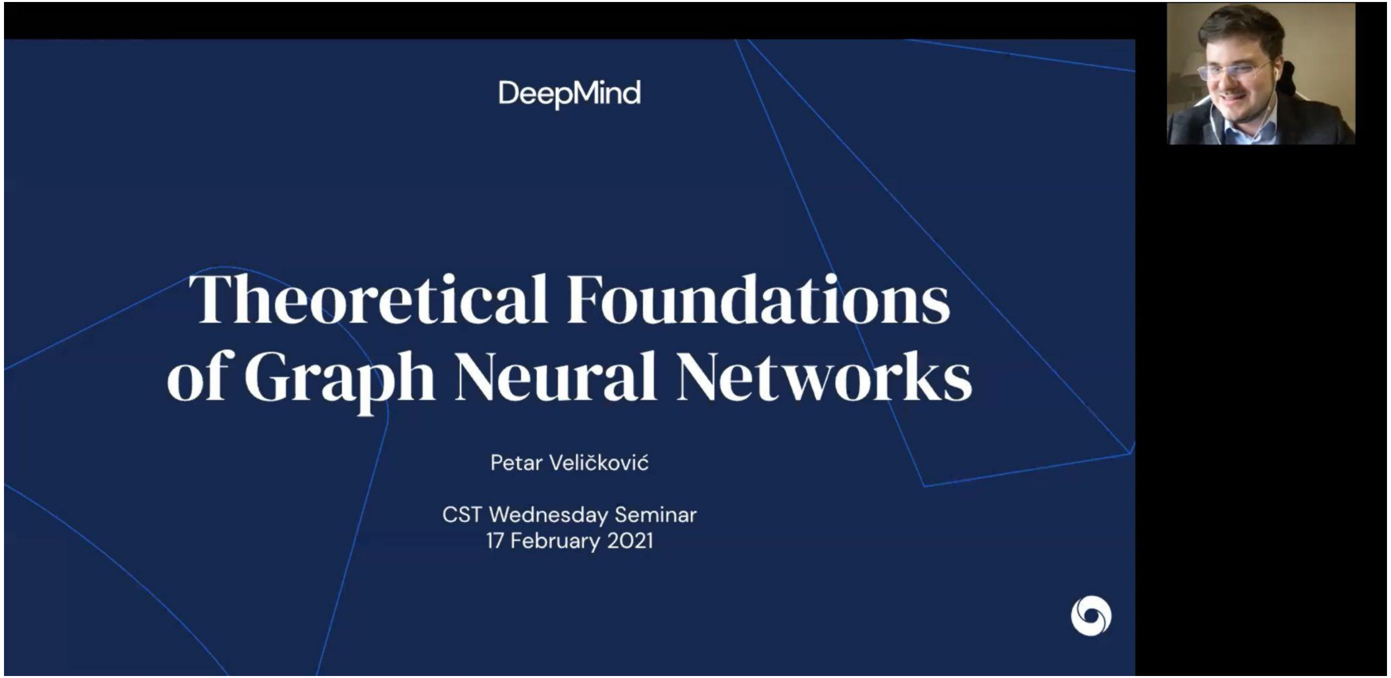
- PNA proposes empirically powerful **combination** of aggregators for general-purpose GNNs:

$$\bigoplus = \underbrace{\begin{bmatrix} I \\ S(D, \alpha = 1) \\ S(D, \alpha = -1) \end{bmatrix}}_{\text{scalers}} \otimes \underbrace{\begin{bmatrix} \mu \\ \sigma \\ \max \\ \min \end{bmatrix}}_{\text{aggregators}}$$

III Further resources

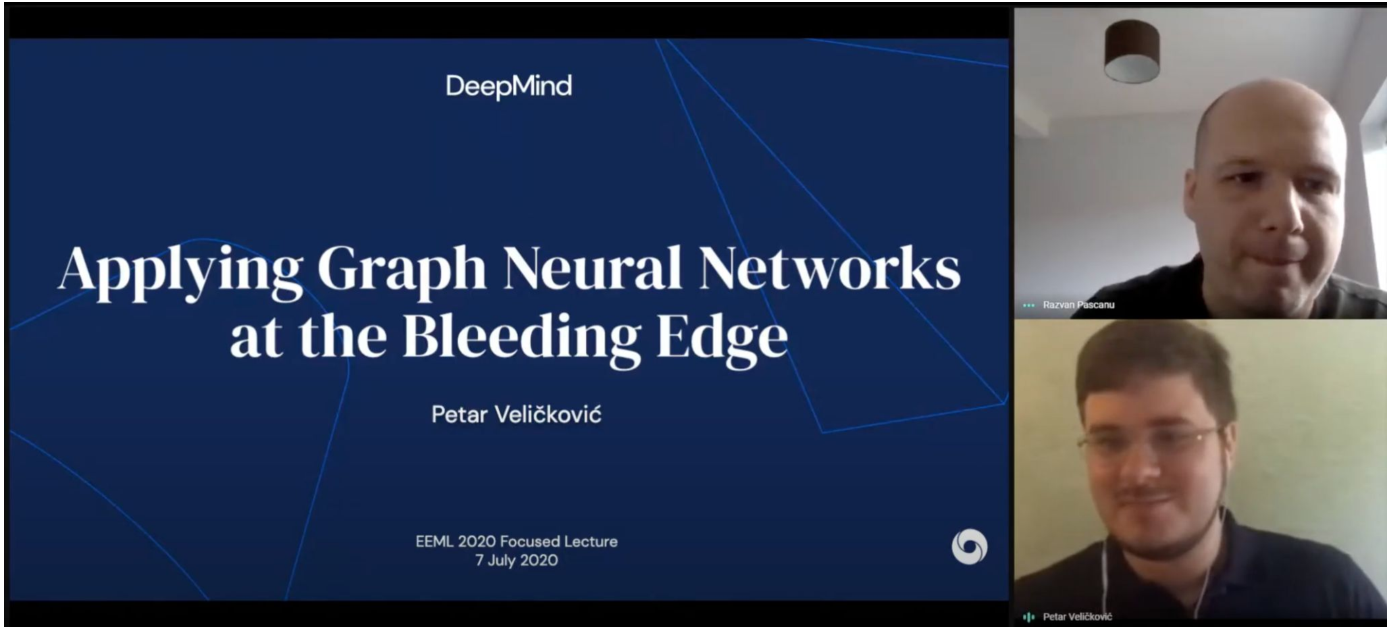
Further insight: graph representation learning

If GNNs are new(ish) to you, I recently gave a useful talk on **theoretical GNN foundations**:
https://www.youtube.com/watch?v=uF53xsT7mjc



Further insight: bleeding-edge applications

For an in-depth view of bleeding edge applications of GNNs, check out my **EEML 2020 talk**:
https://www.youtube.com/watch?v=fpb3j33RfTc



DeepMind

Thank you!

Questions?

petarv@deepmind.com | https://petar-v.com