Graph Networks

Vitaly Kurin

University of Oxford

June 7, 2019

Disclaimer

I am not a Graph Neural Networks expert. I want to share my excitement and make more people aware of this amazing research direction.

Graphs are everywhere

- physics

Graphs are everywhere

- physics
- chemistry

Graphs are everywhere

- physics
- chemistry
- computer science

Graphs are everywhere

- physics
- chemistry
- computer science
- biology

Graphs are everywhere

- physics
- chemistry
- computer science
- biology
- ...

It's a two way street

- What can ML do for graphs?

It's a two way street

- What can ML do for graphs?
- What can graphs do for ML

It's a two way street

- What can ML do for graphs?
- What can graphs do for ML

Inductive biases examples

- Locality in CNN

Inductive biases examples

- Locality in CNN
- Sequential structure in RNN

Inductive biases examples

- Locality in CNN
- Sequential structure in RNN
- Locality in Nearest Neighbours

Inductive biases examples

- Locality in CNN
- Sequential structure in RNN
- Locality in Nearest Neighbours
- ...

Graph Neural Networks (GNN)

- GNN originated in [GMS05] and [SGT$^+$08];

Graph Neural Networks (GNN)

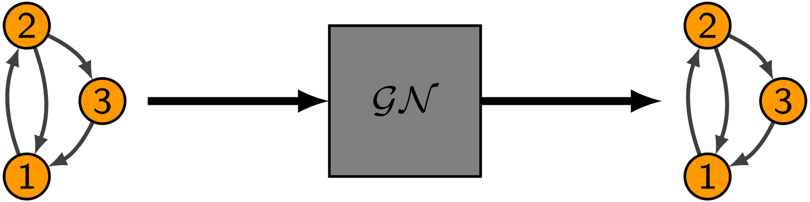
- GNN originated in [GMS05] and [SGT$^+$08];
- [BHB$^+$18] unifies a lot of types into Graph Networks (GN);

Graph Neural Networks (GNN)

- GNN originated in [GMS05] and [SGT+08];
- [BHB+18] unifies a lot of types into Graph Networks (GN);

This talk focuses on GN only!

What is a Graph Network?



What is a Graph?

- Directed;

What is a Graph?

- Directed;
- Labeled;

What is a Graph?

- Directed;
- Labeled;
- With the global attribute $\mathbf{u}$;

What is a Graph?

- Directed;
- Labeled;
- With the global attribute $\mathbf{u}$;
- tuple $\langle \mathcal{V}, \mathcal{E}, u, \rangle$;

What is a Graph?

- Directed;
- Labeled;
- With the global attribute $\mathbf{u}$;
- tuple $\langle \mathcal{V}, \mathcal{E}, u, \rangle$;
- I will call all of them *entities*;

What is a Graph?

- Directed;
- Labeled;
- With the global attribute $\mathbf{u}$;
- tuple $\langle \mathcal{V}, \mathcal{E}, u, \rangle$;
- I will call all of them *entities*;
- $e = (s, r) \forall e \in \mathcal{E}$;

What is a Graph?

- Directed;
- Labeled;
- With the global attribute $\mathbf{u}$;
- tuple $\langle \mathcal{V}, \mathcal{E}, u, \rangle$;
- I will call all of them *entities*;
- $e = (s, r) \forall e \in \mathcal{E}$;
- $\mathcal{E}_{v_i}^{in}$ is the set of all incoming edges to v_i;

Two main components of a GN

- updaters ϕ^e, ϕ^v, ϕ^u

Two main components of a GN

- updaters ϕ^e, ϕ^v, ϕ^u
- aggregators $\rho^{e\to v}, \rho^{v\to u}, \rho^{e\to u}$

Updater is a function which updates entities features

- edge updater: $e_i' = \phi^e(e_i, s_i, r_i, u)$

Updater is a function which updates entities features

- edge updater: $e_i' = \phi^e(e_i, s_i, r_i, u)$
- vertex updater: $v_i' = \phi^v(v_i, \rho^{e \to v}(\mathcal{E}_{v_i}^{in}), u)$

Updater is a function which updates entities features

- edge updater: $e_i' = \phi^e(e_i, s_i, r_i, u)$
- vertex updater: $v_i' = \phi^v(v_i, \rho^{e \to v}(\mathcal{E}_{v_i}^{in}), u)$
- global updater: $u' = \phi^u(\rho^{e \to u}(\mathcal{E}), \rho^{v \to u}(\mathcal{V}), u)$

What is an aggregator?

- a function working on sets

What is an aggregator?

- a function working on sets
- the main trick of a GN;

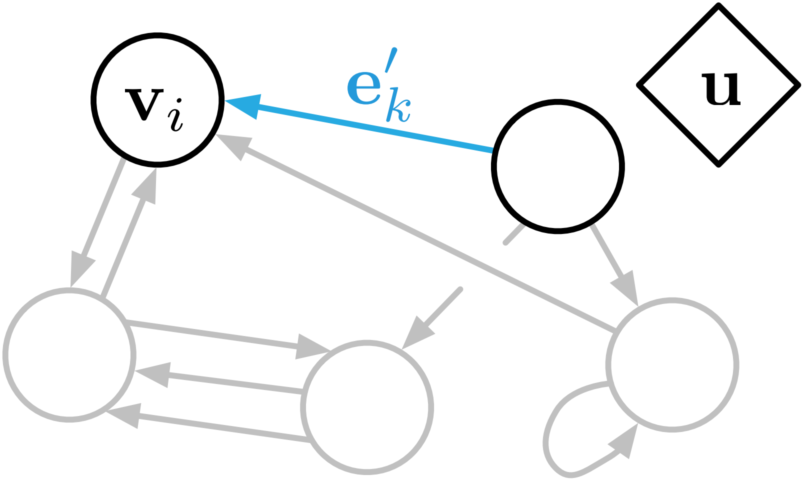
What is an aggregator?

- a function working on sets
- the main trick of a GN;
- enables GN to work on different graph topologies;

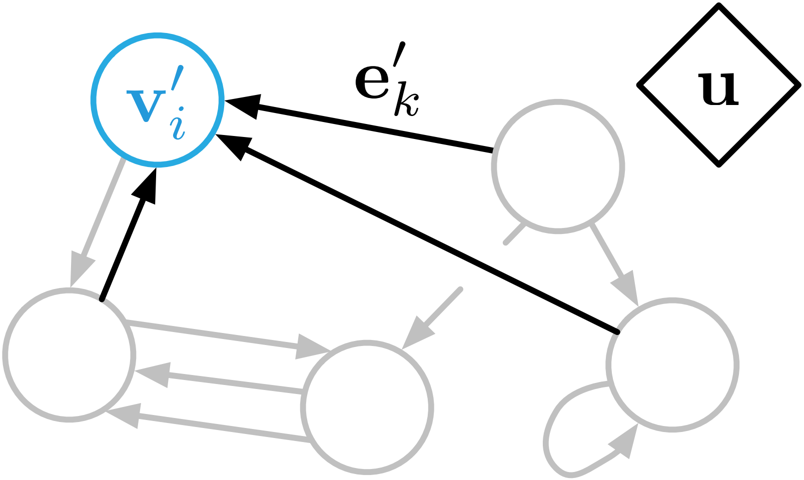
What is an aggregator?

- a function working on sets
- the main trick of a GN;
- enables GN to work on different graph topologies;
- examples?

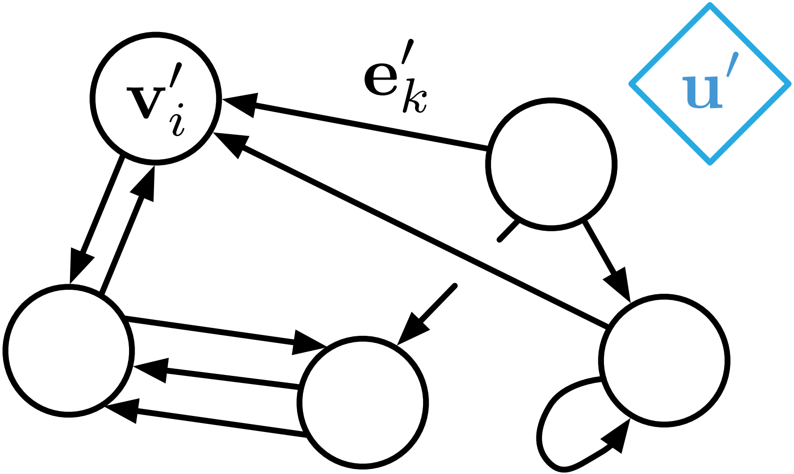
Edge forward step, [BHB+18]



Vertex forward step, [BHB+18]



Global forward step, [BHB+18]



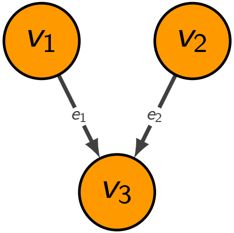
GN forward step

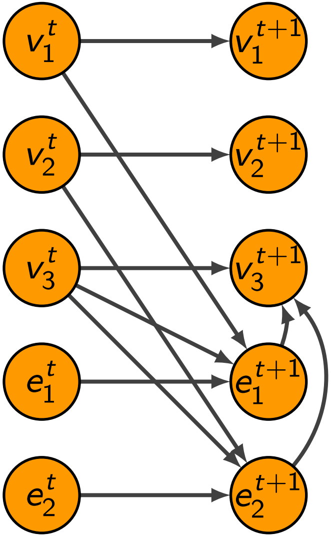
Algorithm 1 Steps of computation in a full GN block. [BHB+18]

---

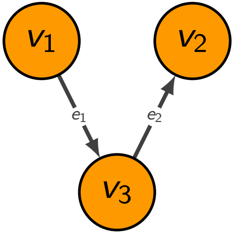
function $\textsc{GraphNetwork}(\mathcal{E}, \mathcal{V}, \mathbf{u})$
  for $k \in \{1 \dots N^e\}$ **do**
    $\mathbf{e}'_k \leftarrow \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}, \right)$ ▷ 1. Compute updated edge attributes
  end for
  for $i \in \{1 \dots N^n\}$ **do**
    let $\mathcal{E}'_i = \{(\mathbf{e}'_k, r_k, s_k)\}_{r_k=i,\ k=1:N^e}$
    $\bar{\mathbf{e}}'_i \leftarrow \rho^{e \to v}\left(\mathcal{E}'_i\right)$ ▷ 2. Aggregate edge attributes per node
    $\mathbf{v}'_i \leftarrow \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right)$ ▷ 3. Compute updated node attributes
  end for
  let $V' = \{\mathbf{v}'\}_{i=1:N^v}$
  let $E' = \{(\mathbf{e}'_k, r_k, s_k)\}_{k=1:N^e}$
  $\bar{\mathbf{e}}' \leftarrow \rho^{e \to u}\left(\mathcal{E}'\right)$ ▷ 4. Aggregate edge attributes globally
  $\bar{\mathbf{v}}' \leftarrow \rho^{v \to u}\left(\mathcal{V}'\right)$ ▷ 5. Aggregate node attributes globally
  $\mathbf{u}' \leftarrow \phi^u\left(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}\right)$ ▷ 6. Compute updated global attribute
  return $(\mathcal{E}', \mathcal{V}', \mathbf{u}')$
end function

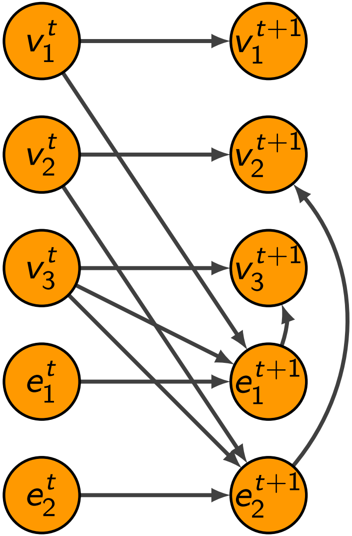
GN computation graph





Input graph defines the computation graph!



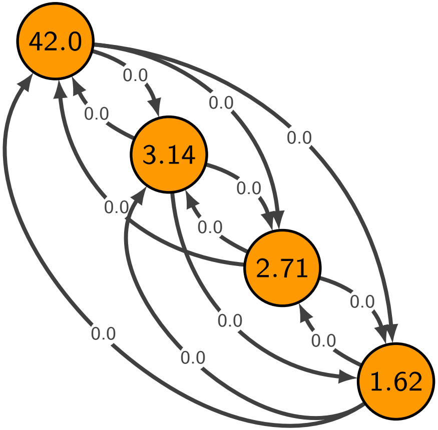


Let's see Graph Networks in action!

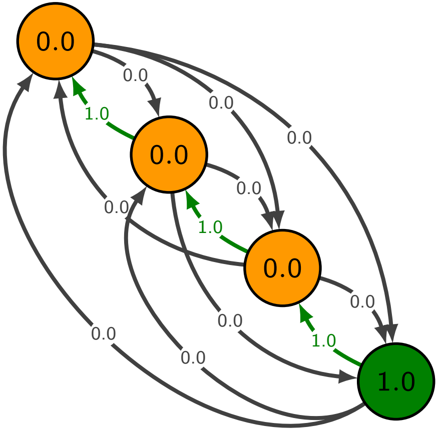
Sort an array of real numbers

42.0	3.14	2.71	1.62

Graph Encoding

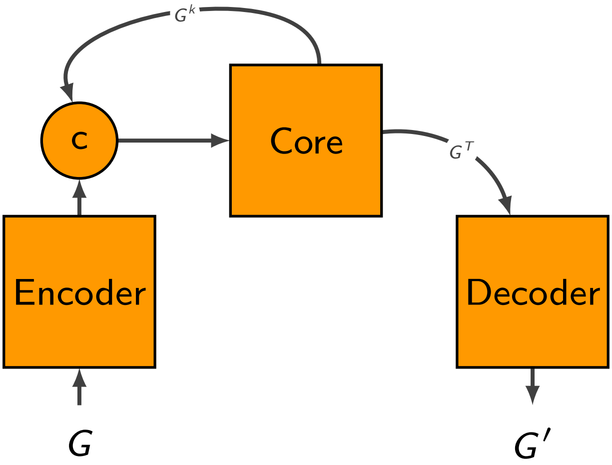


(a) Input Graph

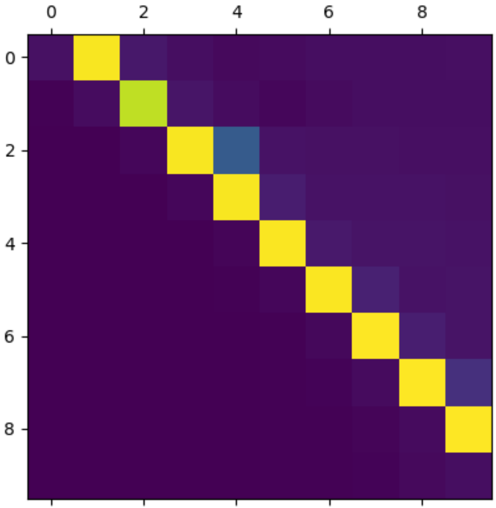


(b) Target Graph

Model Architecture



Sorting task results



Graph Networks as a tool

Graph Networks for Supervised Learning

Neural Message Passing for Quantum Chemistry

G] 12 Jun 2017

Neural Message Passing for Quantum Chemistry

Justin Gilmer [1] **Samuel S. Schoenholz** [1] **Patrick F. Riley** [2] **Oriol Vinyals** [3] **George E. Dahl** [1]

Abstract

Supervised learning on molecules has incredible potential to be useful in chemistry, drug discovery, and materials science. Luckily, several promising and closely related neural network models invariant to molecular symmetries have already been described in the literature. These models learn a message passing algorithm and aggregation procedure to compute a function of their entire input graph. At this point, the next step is to find a particularly effective variant of this general approach and apply it to chemical prediction benchmarks until we either solve them or reach the limits of the approach. In this pa-

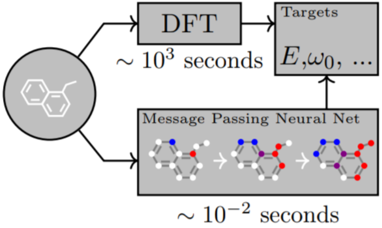
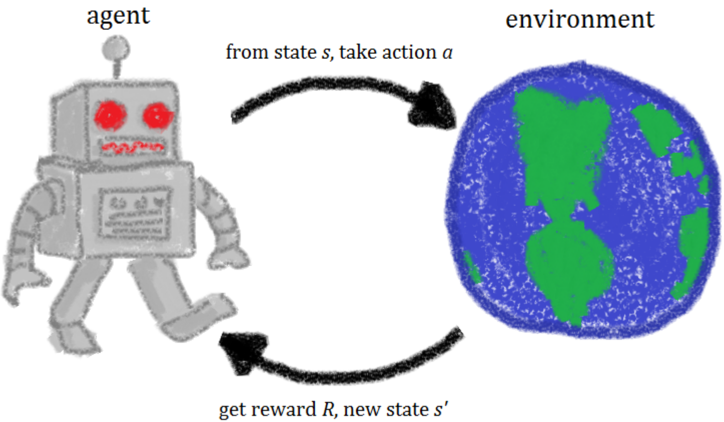


Figure 1. A Message Passing Neural Network predicts quantum properties of an organic molecule by modeling a computationally

Graph Networks for RL

Reinforcement Learning in a nutshell



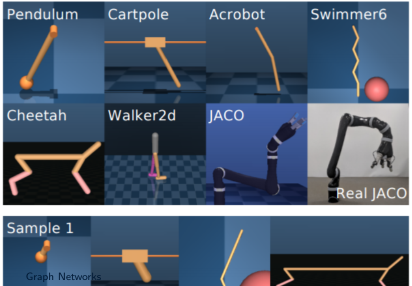
source: `https://commons.wikimedia.org/wiki/File:Rl_agent.png`

Graph Networks as Learnable Physics Engines for Inference and Control

Alvaro Sanchez-Gonzalez [1] Nicolas Heess [1] Jost Tobias Springenberg [1] Josh Merel [1] Martin Riedmiller [1]
Raia Hadsell [1] Peter Battaglia [1]

Abstract

Understanding and interacting with everyday physical scenes requires rich knowledge about the structure of the world, represented either implicitly in a value or policy function, or explicitly in a transition model. Here we introduce a new class of learnable models—based on *graph networks*—which implement an inductive bias for object- and relation-centric representations of complex, dynamical systems. Our results show that as a forward model, our approach supports accurate predictions from real and simulated data, and surprisingly strong and efficient generaliza-



NerveNet: Learning Structured Policy with Graph Neural Networks

Published as a conference paper at ICLR 2018

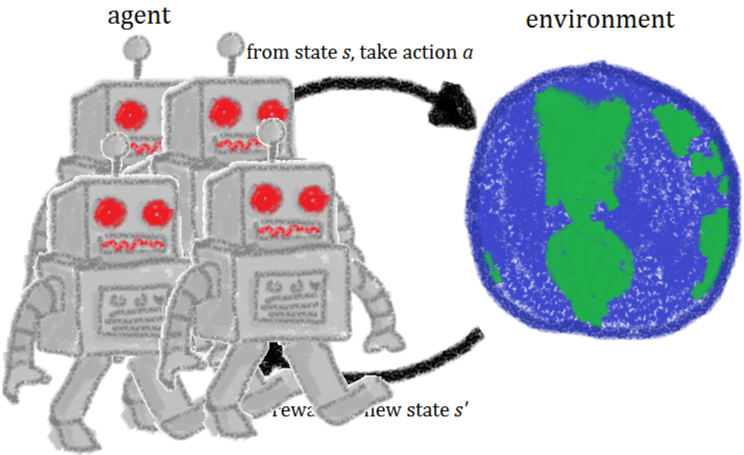
NerveNet: Learning Structured Policy with Graph Neural Networks

Tingwu Wang[*], Renjie Liao[*], Jimmy Ba & Sanja Fidler
Department of Computer Science
University of Toronto
Vector Institute
{tingwuwang,rjliao}@cs.toronto.edu,
jimmy@psi.toronto.edu, fidler@cs.toronto.edu

Abstract

We address the problem of learning structured policies for continuous control. In traditional reinforcement learning, policies of agents are learned by multi-layer perceptrons (MLPs) which take the concatenation of all observations from the environment as input for predicting actions. In this work, we propose NerveNet to explicitly model the structure of an agent, which naturally takes the form of a graph. Specifically, serving as the agent's policy network, NerveNet first propagates information over the structure of the agent and then predict actions for different parts of the agent. In the experiments, we first show that our NerveNet is comparable to state-of-the-art methods on standard MuJoCo environments. We further

Multi-Agent Reinforcement Learning



source: `https://commons.wikimedia.org/wiki/File:Rl_agent.png`

Relational Forward Models

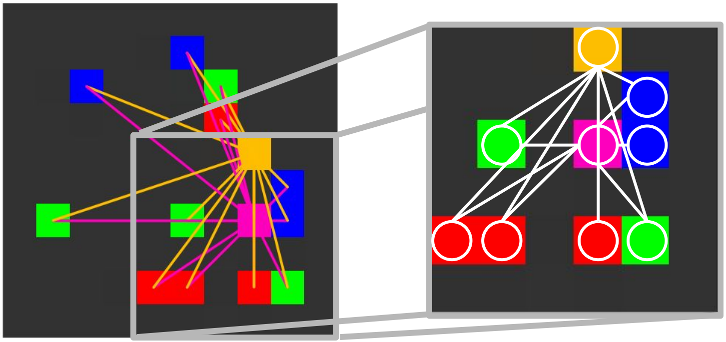
RELATIONAL FORWARD MODELS FOR MULTI-AGENT LEARNING

Andrea Tacchetti*[1], **H. Francis Song***[1], **Pedro A. M. Mediano***[1,2], **Vinicius Zambaldi**[1],
Neil C. Rabinowitz[1], **Thore Graepel**[1], **Matthew Botvinick**[1] **& Peter W. Battaglia**[1]
* denotes equal contribution
[1] DeepMind
[2] Imperial College London
{atacchet,songf,pmediano,vzambaldi
ncr,thore,botvinick,peterbattaglia}@google.com

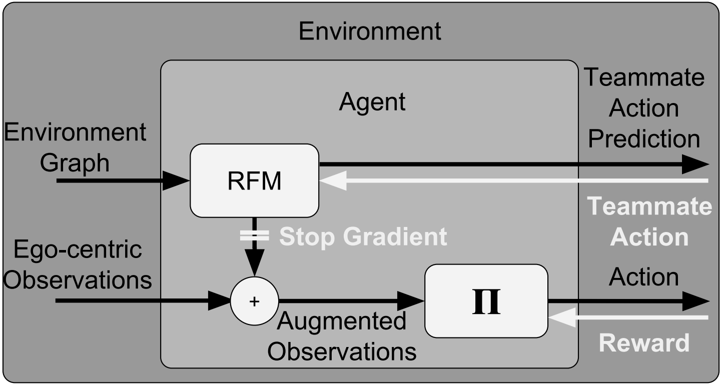
ABSTRACT

The behavioral dynamics of multi-agent systems have a rich and orderly structure, which can be leveraged to understand these systems, and to improve how artificial agents learn to operate in them. Here we introduce Relational Forward Models (RFM) for multi-agent learning, networks that can learn to make accurate predictions of agents' future behavior in multi-agent environments. Because these models operate on the discrete entities and relations present in the environment, they produce interpretable intermediate representations which offer insights into what drives agents' behavior, and what events mediate the intensity and valence of social interactions. Furthermore, we show that embedding RFM modules inside agents results in faster learning systems compared to non-augmented baselines. As more and more of the autonomous systems we develop and interact with become multi-agent in nature, developing richer analysis tools for characterizing how and why agents make decisions is increasingly necessary. Moreover, developing artificial agents that quickly and safely learn to coordinate with one another, and with humans in shared environments, is crucial.

Each state of a MARL problem can be represented as a graph



Policies conditioned on graphs



Not all data is ready available in the form of a graph

Computational efficiency?

- Hard to batch with dynamic graphs;

Theoretical limitations?

- Some of the phenomena are not well suited for graph representation
- Graph Networks are unable to solve some classes of problems, i.e. discriminating between some non-isomorphic graphs.

Practical considerations

- Think of graphs as of data with some feeding mechanism
- Batch
- Padding becomes expensive;
- `torch.split hurts the backward pass A LOT;`
- Have to be integrated with traditional pipelines;
- Visualisation is a Thing!

To sum up

- There's Plenty of Room at the Bottom
- GN is a flexible instrument for injection of inductive biases
- Graph Networks are turning into a tool for supervised learning, unsupervised learning and RL
- Jump on the bandwagon!

Thanks!

- `https://twitter.com/y0b1byte`
- `https://yobibyte.github.io/pages/paper-notes.html`
- `vitaly.kurin@magd.ox.ac.uk`

References I

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al.
Relational inductive biases, deep learning, and graph networks.
arXiv preprint arXiv:1806.01261, 2018.

Marco Gori, Gabriele Monfardini, and Franco Scarselli.
A new model for learning in graph domains.
In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia.
Graph networks as learnable physics engines for inference and control.
arXiv preprint arXiv:1806.01242, 2018.

References II

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini.
The graph neural network model.
IEEE Transactions on Neural Networks, 20(1):61–80, 2008.

Andrea Tacchetti, H Francis Song, Pedro AM Mediano, Vinicius Zambaldi, Neil C Rabinowitz, Thore Graepel, Matthew Botvinick, and Peter W Battaglia.
Relational forward models for multi-agent learning.
arXiv preprint arXiv:1809.11044, 2018.

Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler.
Nervenet: Learning structured policy with graph neural networks.
2018.