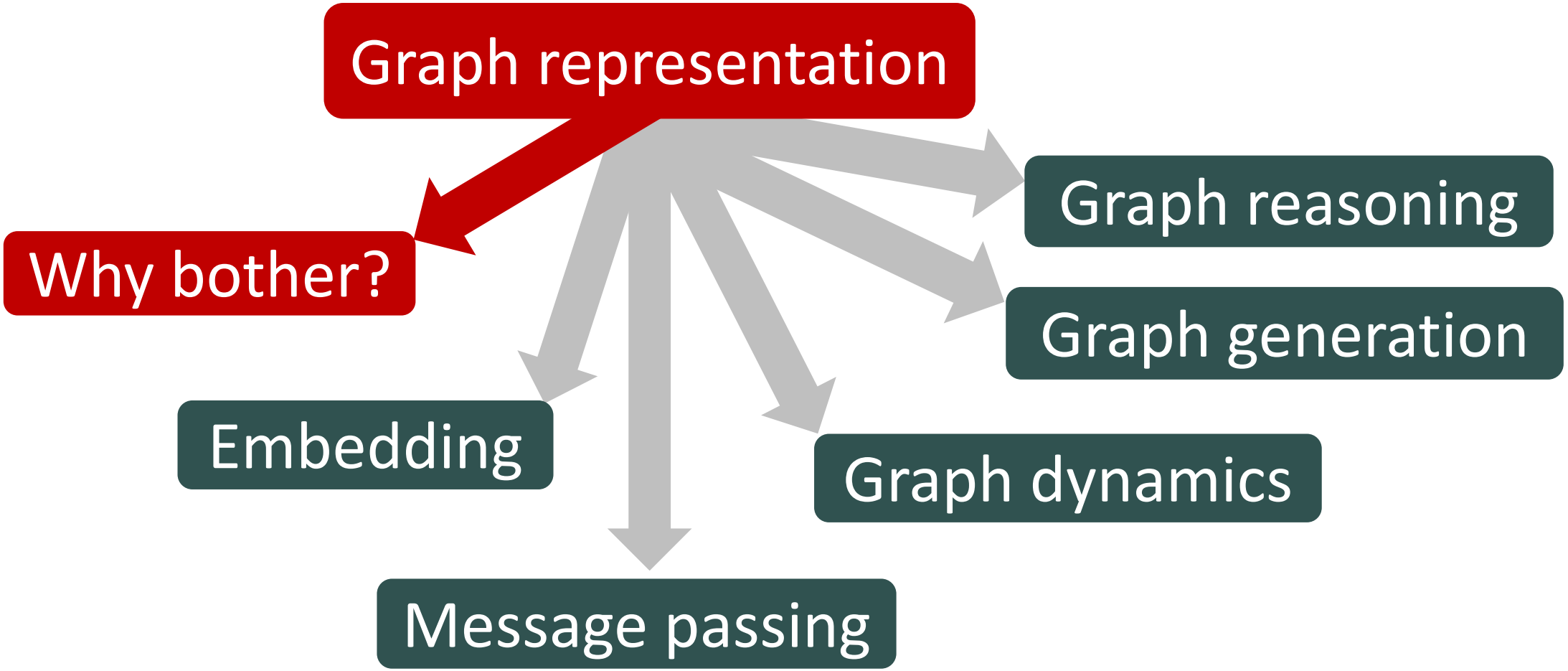
Representation learning on graphs

A/Prof Truyen Tran
Deakin University

truyen.tran@deakin.edu.au
truyentran.github.io
@truyenoz
letdataspeak.blogspot.com
goo.gl/3jJ1O0

HCMC, May 2019


Graph representation
Why bother?
Graph reasoning
Graph generation
Embedding
Graph dynamics
Message passing

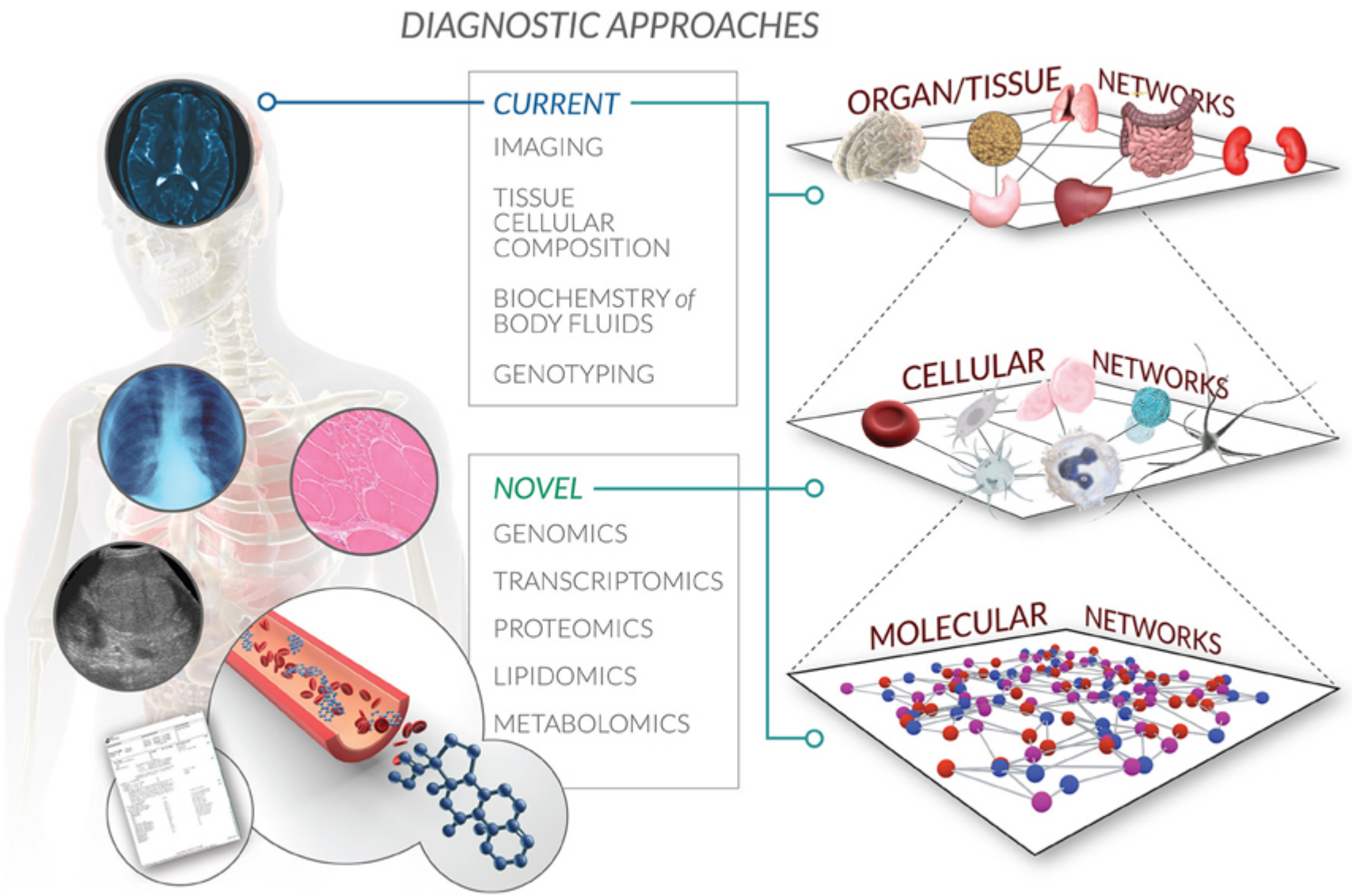
Why learning of graph representation?

Graphs are pervasive in many scientific disciplines.

The sub-area of graph representation has reached a certain maturity, with multiple reviews, workshops and papers at top AI/ML venues.

Deep learning needs to move beyond vector, fixed-size data.

Learning representation as a powerful way to discover hidden patterns making learning, inference and planning easier.



System medicine

https://www.frontiersin.org/articles/10.3389/fphys.2015.00225/full

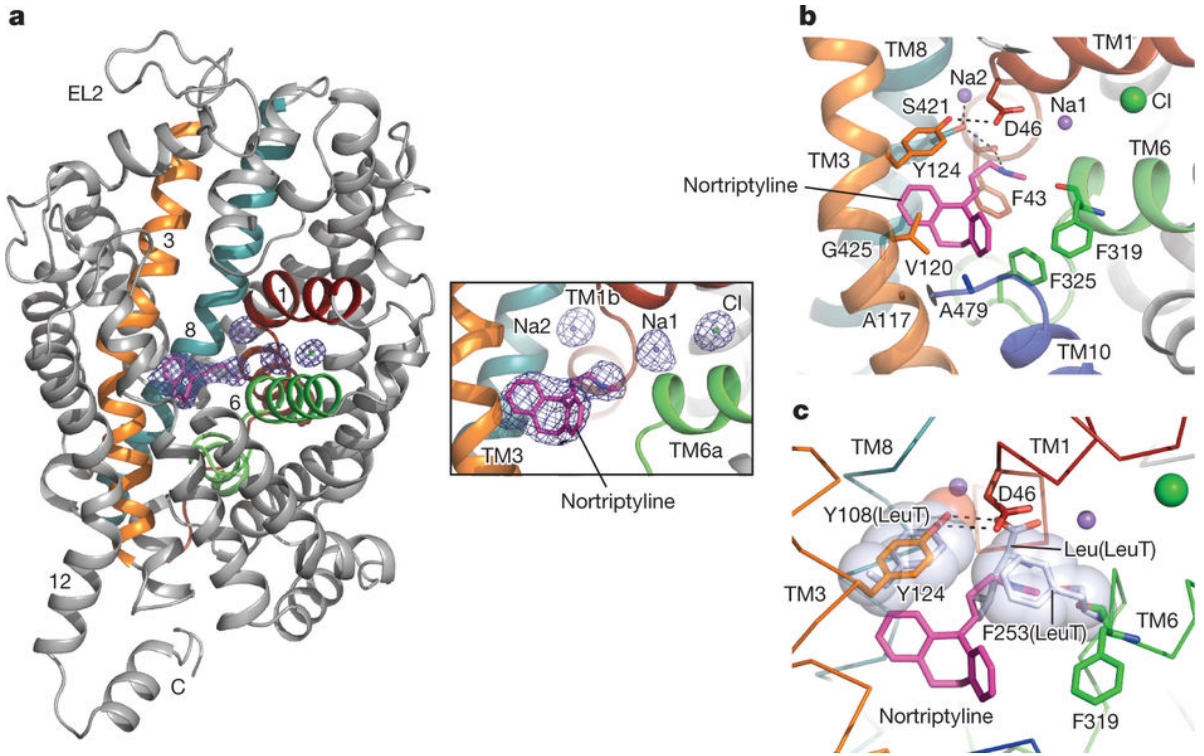
Biology & pharmacy

Traditional techniques:

- Graph kernels (ML)
- Molecular fingerprints (Chemistry)

Modern techniques

- Molecule as graph: atoms as nodes, chemical bonds as edges

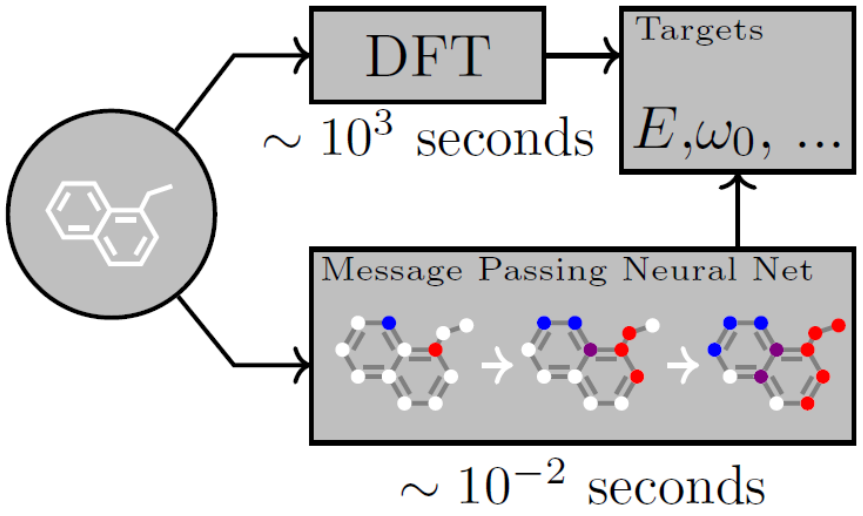


#REF: Penmatsa, Aravind, Kevin H. Wang, and Eric Gouaux. "X-ray structure of dopamine transporter elucidates antidepressant mechanism." *Nature* 503.7474 (2013): 85-90.

Chemistry

DFT = Density Functional Theory

Gilmer, Justin, et al. "Neural message passing for qu
chemistry." *arXiv preprint arXiv:1704.01212* (2017)



- Molecular properties
- Chemical-chemical interaction
- Chemical reaction
- Synthesis planning

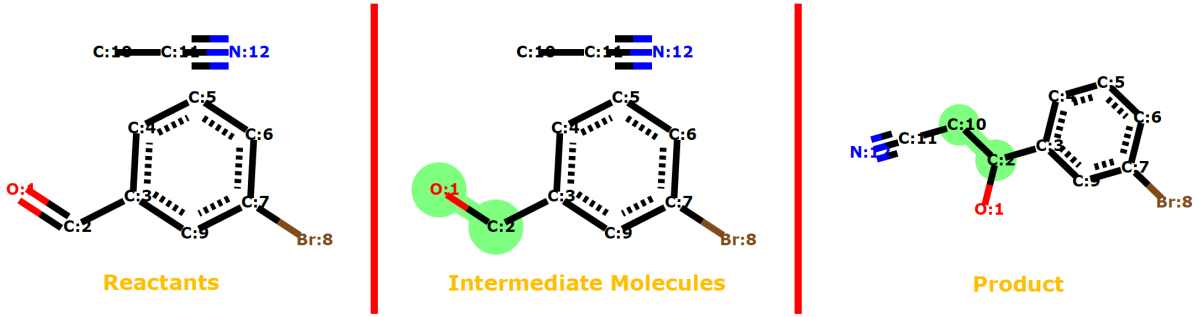
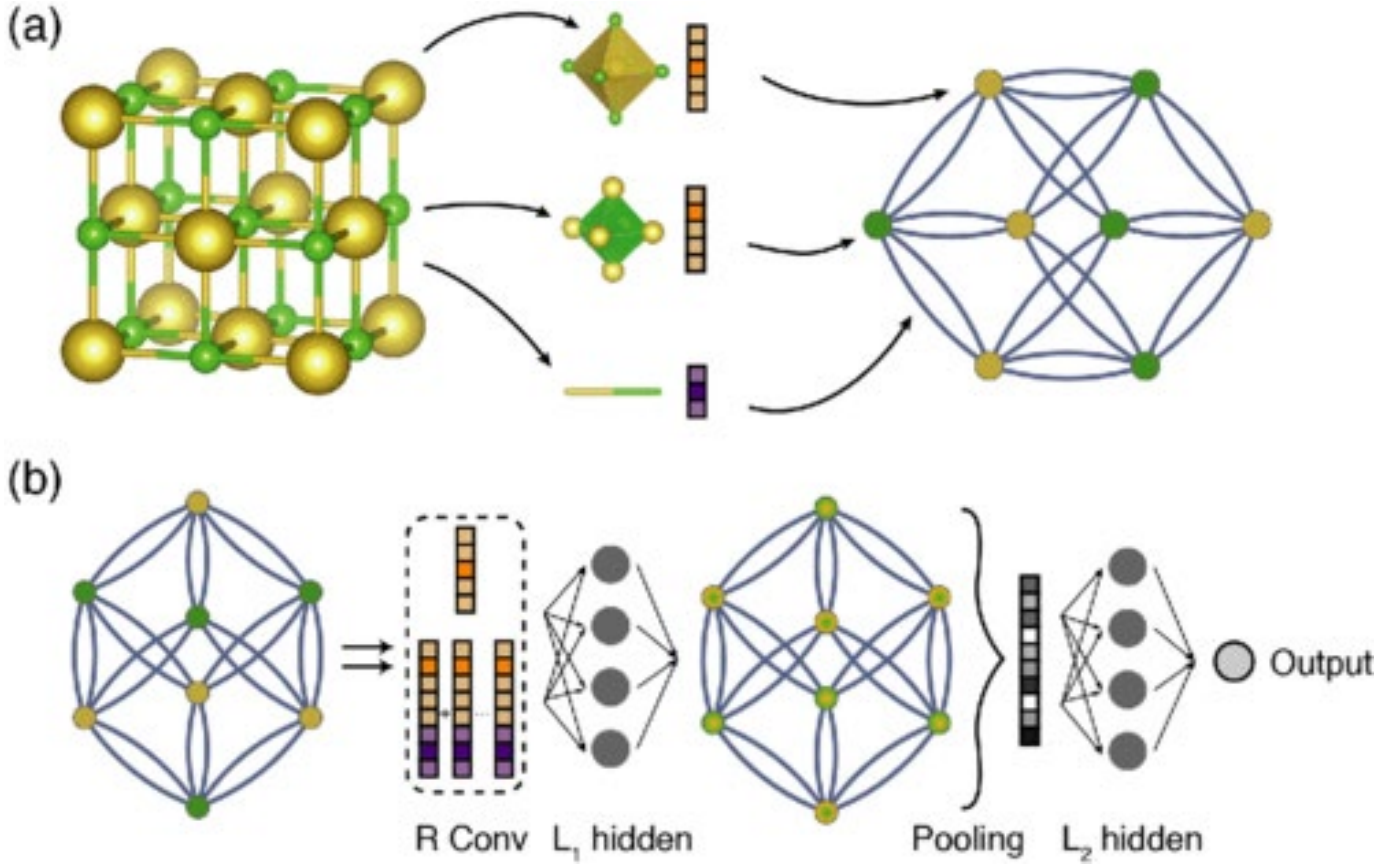


Figure 1: A sample reaction represented as a set of graph transformations from reactants (leftmost) to products (rightmost). Atoms are labeled with their type (Carbon, Oxygen,...) and their index (1, 2,...) in the molecular graph. The atom pairs that change connectivity and their new bonds (if existed) are highlighted in green. There are two bond changes in this case: 1) The double bond between O:1 and C:2 becomes single. 2) A new single bond between C:2 and C:10 is added.

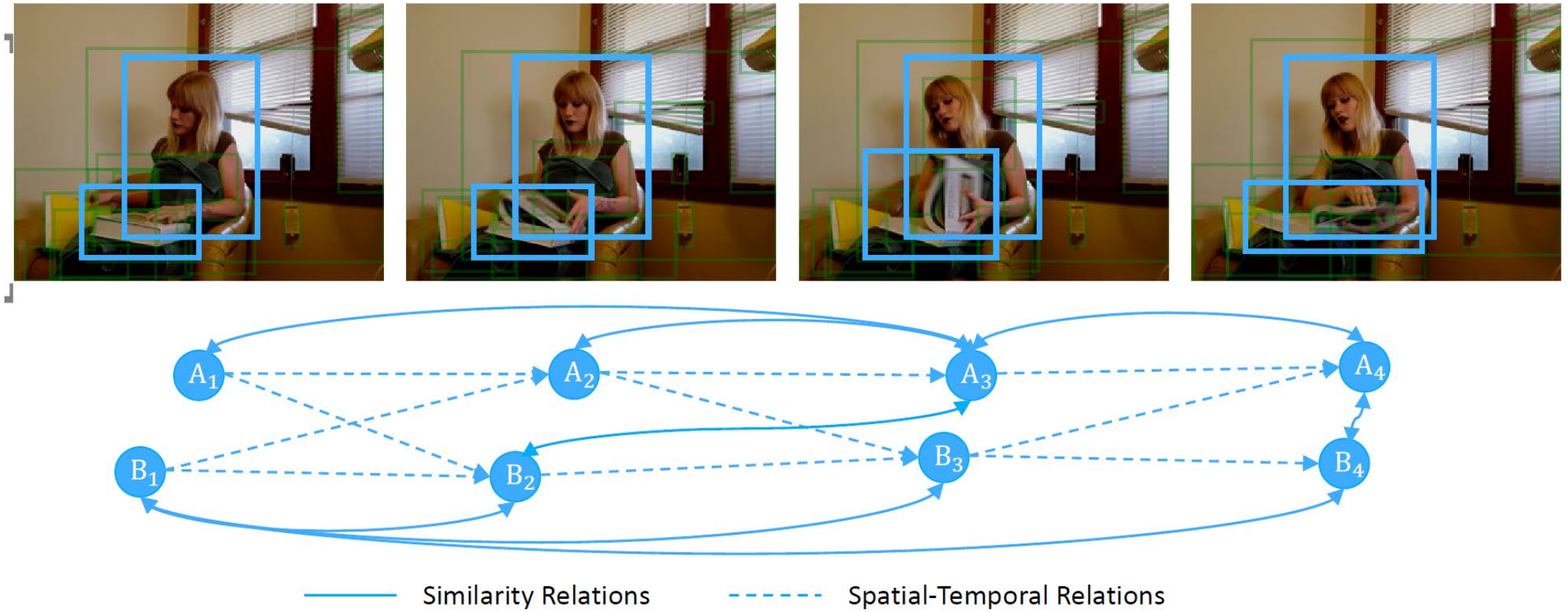
Materials science

- Crystal properties
- Exploring/generating solid structures
- Inverse design



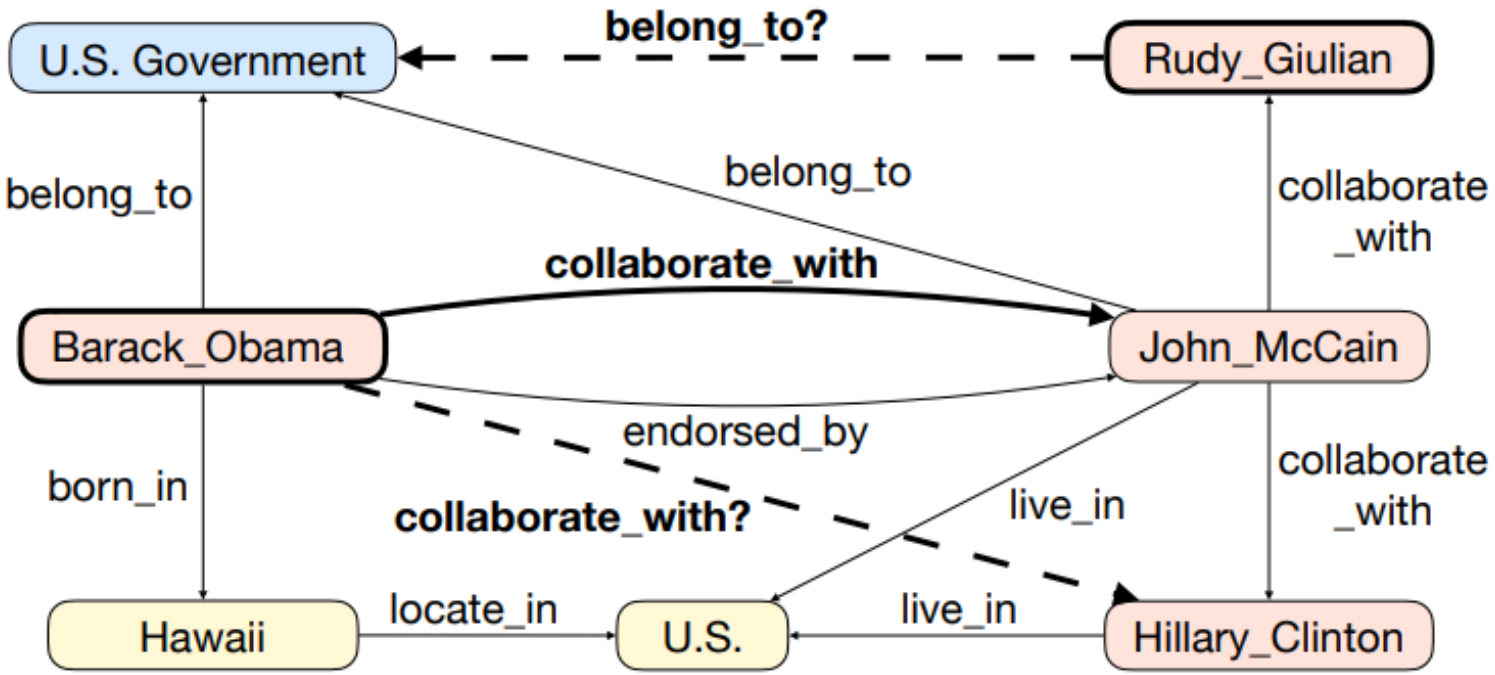
Xie, Tian, and Jeffrey C. Grossman. "Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties." *Physical review letters* 120.14 (2018): 145301.

Videos as space-time region graphs

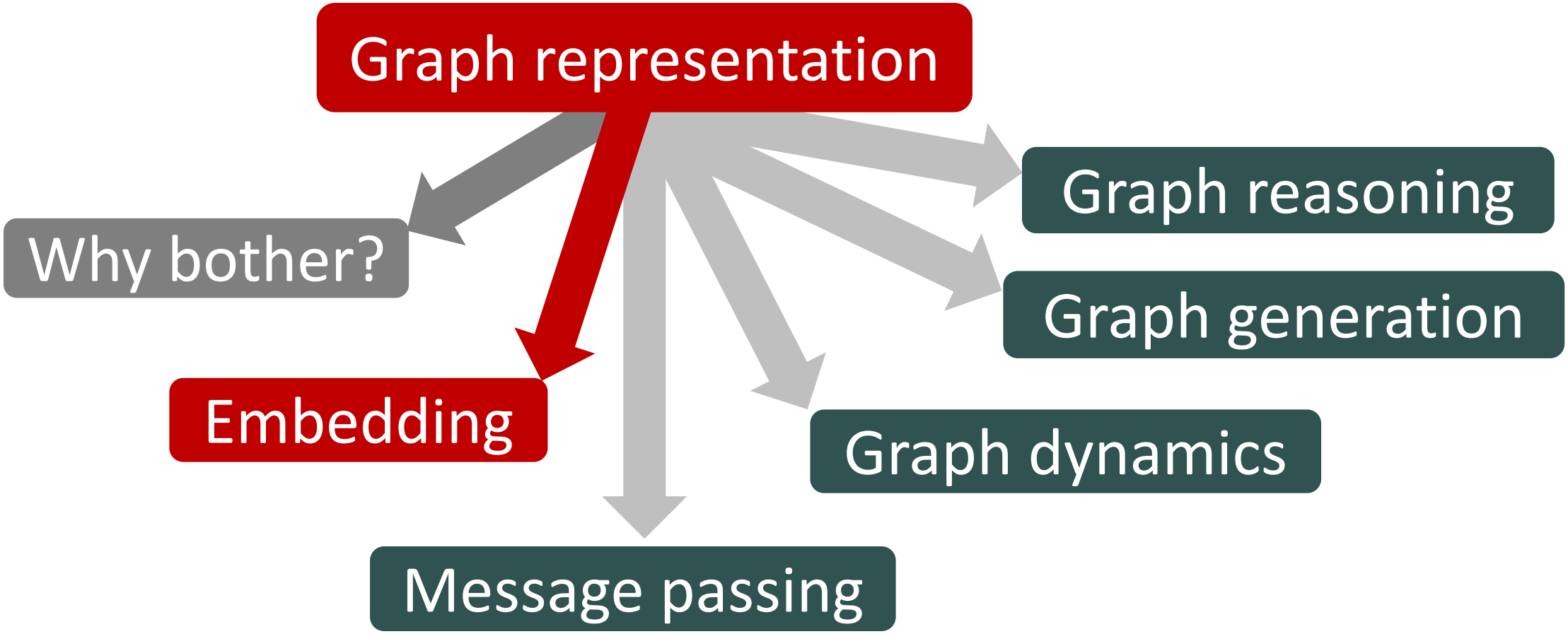


(Abhinav Gupta et al, ECCV'18)

Knowledge graphs

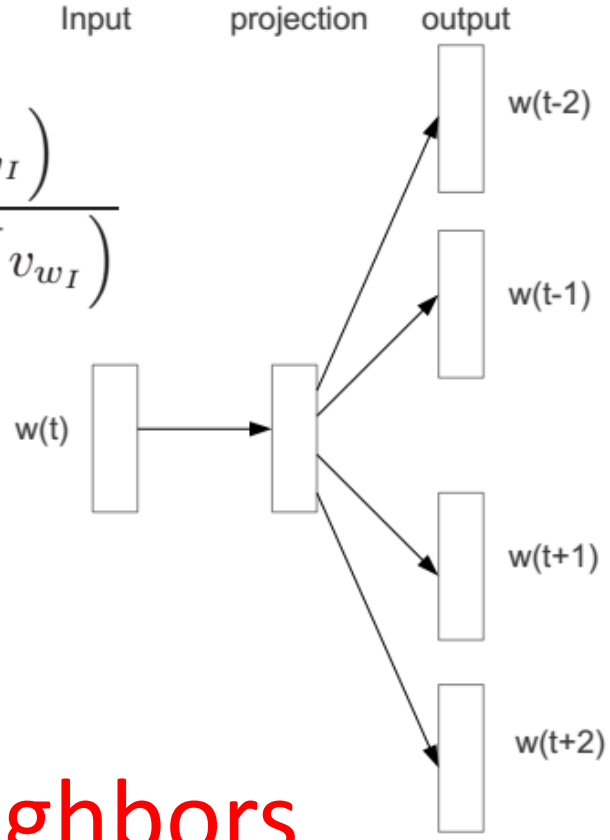


https://www.zdnet.com/article/salesforce-research-knowledge-graphs-and-machine-learning-to-power-einstein/


Graph representation
Why bother?
Embedding
Graph reasoning
Graph generation
Graph dynamics
Message passing

Skip-gram

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^\top v_{w_I}\right)}$$



Loss function

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0} \log p(w_{t+j}|w_t)$$

Predicting neighbors

Negative sampling

$$\log\sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i\sim P_n(w)}\left[\log\sigma(-{v'_{w_i}}^\top v_{w_I})\right]$$

DeepWalk (KDD-2014)

Algorithm 1 DeepWalk(G, w, d, γ, t)

Input: graph $G(V, E)$
window size w
embedding size d
walks per vertex γ
walk length t
Output: matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$

1: Initialization: Sample Φ from $\mathcal{U}^{|V| \times d}$
2: Build a binary Tree T from V
3: **for** $i = 0$ to γ **do**
4: $\mathcal{O} = \text{Shuffle}(V)$
5: **for each** $v_i \in \mathcal{O}$ **do**
6: $\mathcal{W}_{v_i} = RandomWalk(G, v_i, t)$
7: SkipGram(Φ, $\mathcal{W}_{v_i}$, w)
8: **end for**
9: **end for**

Considered as #epochs

Embedding matrix

For Hierarchical Softmax

Iterate over each epoch

Finding neighbours of each node

Update embedding of this node

Neighbour nodes

Window size

Node2Vec (KDD-2016)

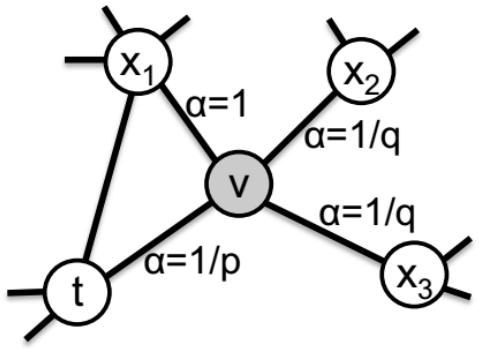
Similar to DeepWalk in using Skip-gram model for unsupervised learning.

Only modifies the search for neighboring nodes that balance between BFS and DFS.

Defines edge embedding based on node embedding

- Can solve link prediction problem

2nd order Random Walk



Consider random walk that just travelled edge **(t, v)**. The walk will decide which is the next node **x** that it should go from **v** by computing π_{vx}

$$\pi_{vx} = \alpha_{pq}(t,x) \cdot w_{vx}$$

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

p and **q** are hyper-parameters

Node2Vec (Cont.)

Algorithm 1 The *node2vec* algorithm.

LearnFeatures (Graph $G = (V, E, W)$, Dimensions d, Walks per node r, Walk length l, Context size k, Return p, In-out q)

$\pi = \text{PreprocessModifiedWeights}(G, p, q)$
$G' = (V, E, \pi)$
Initialize $walks$ to Empty
for $iter = 1$ **to** r **do**
  for all nodes $u \in V$ **do**
    $walk = \text{node2vecWalk}(G', u, l)$
    Append $walk$ to $walks$
$f = \text{StochasticGradientDescent}(k, d, walks)$
return f

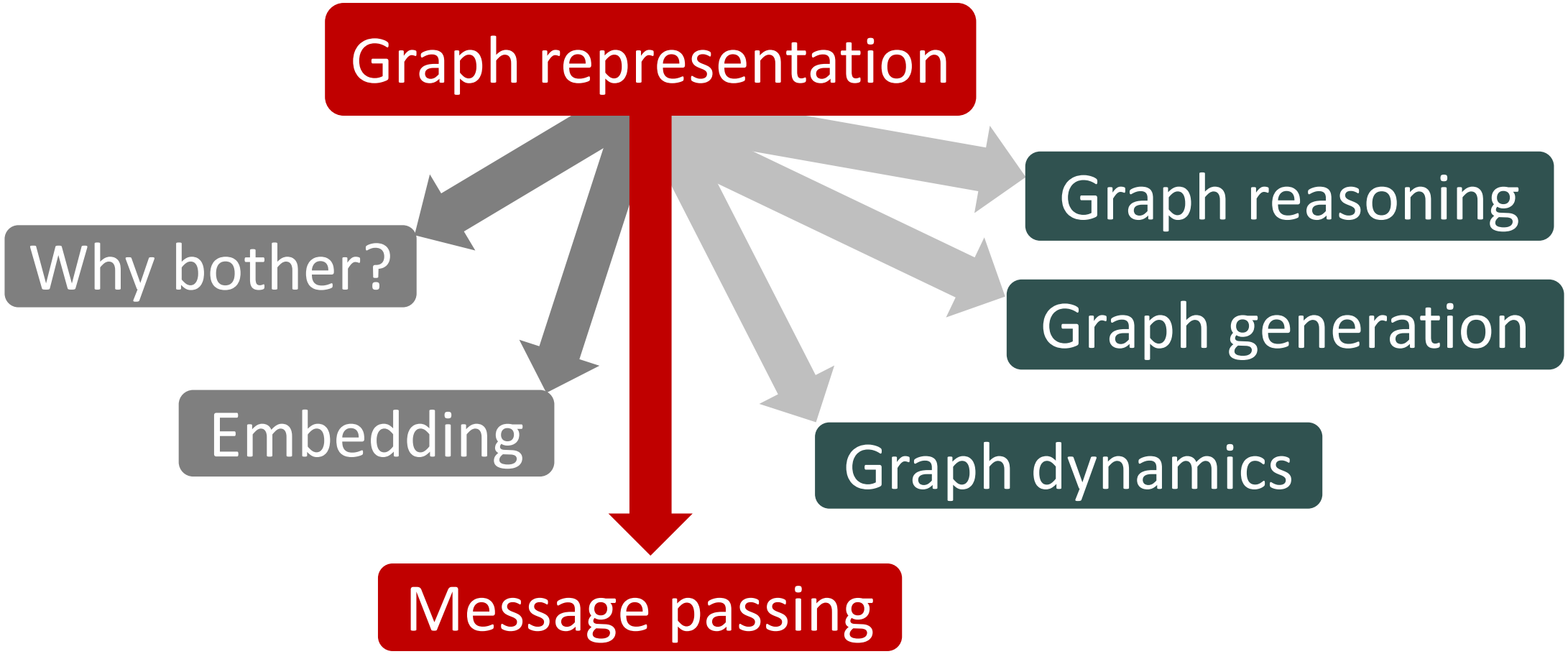
node2vecWalk (Graph $G' = (V, E, \pi)$, Start node u, Length l)

Inititalize $walk$ to $[u]$
for $walk\_iter = 1$ **to** l **do**
  $curr = walk[-1]$
  $V_{curr} = \text{GetNeighbors}(curr, G')$
  $s = \text{AliasSample}(V_{curr}, \pi)$
  Append s to $walk$
return $walk$

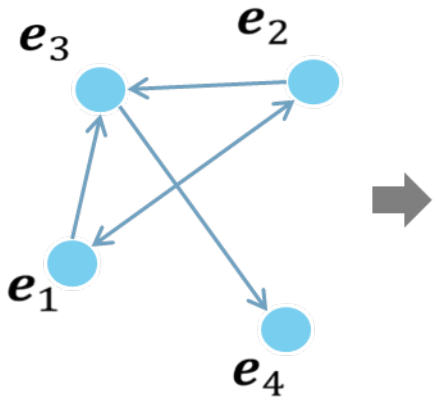
Why it scales over DeepWalk?

Use Negative Sampling instead of Hierarchical Softmax

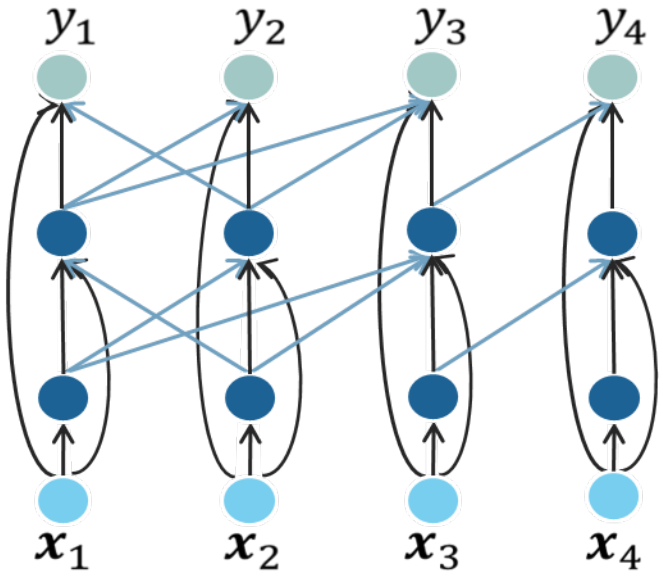
Batch learning


Graph representation
Why bother?
Embedding
Graph reasoning
Graph generation
Graph dynamics
Message passing

Message passing

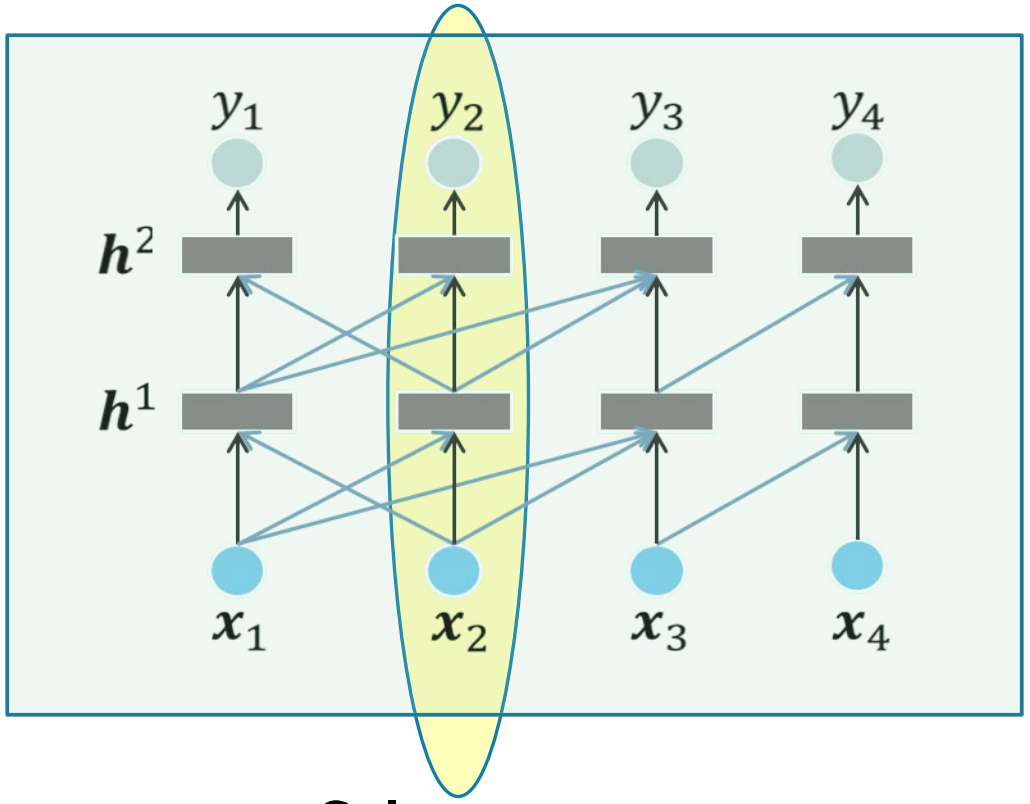


Relation graph



Stacked learning

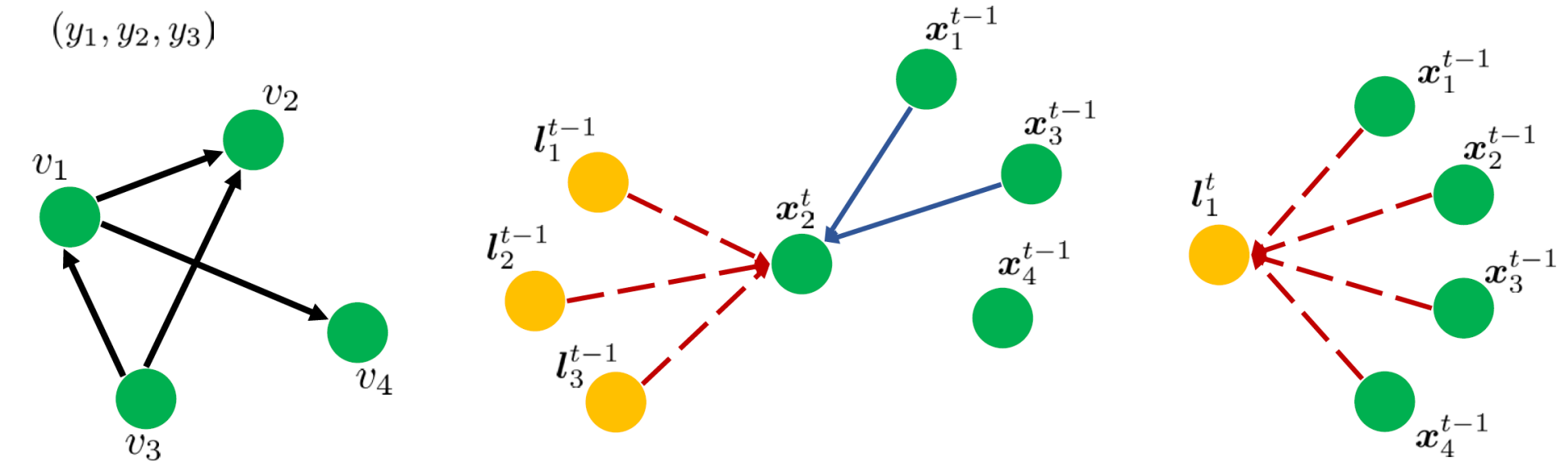
Thin column



Column nets

#REF: Pham, Trang, et al. "Column Networks for Collective Classification." *AAAI*. 2017.

Factored message passing



(a) A input graph with 4 nodes and 3 labels

(b) Input node update

(c) Label node update

#REF: Do, Kien, et al. "Attentional Multilabel Learning over Graphs-A message passing approach." *Machine Learning, 2019*.

Learning deep matrix representations, K Do, **T Tran**, S Venkatesh, *arXiv preprint arXiv:1703.01454*

Graph attention
(Do et al arXiv's17, Veličković et al ICLR' 18)

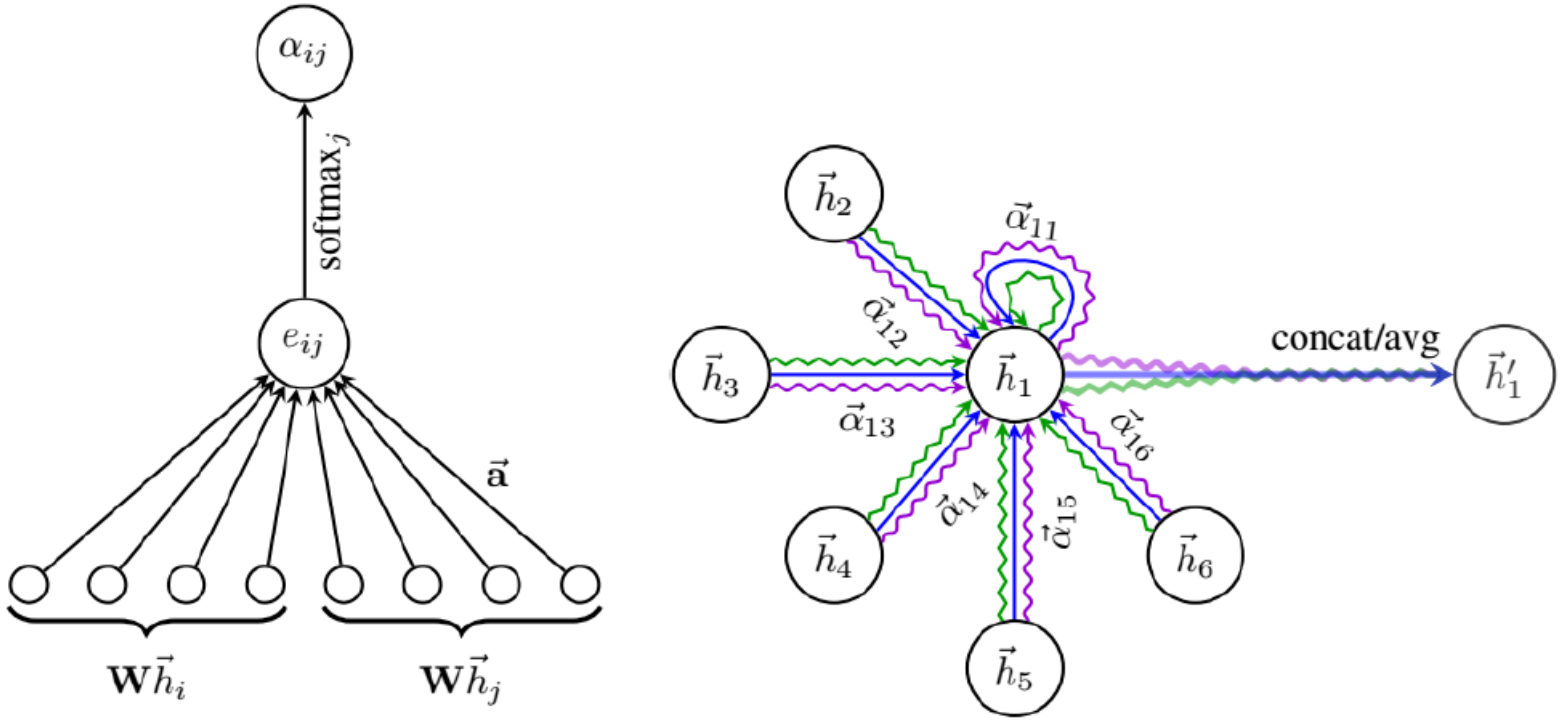
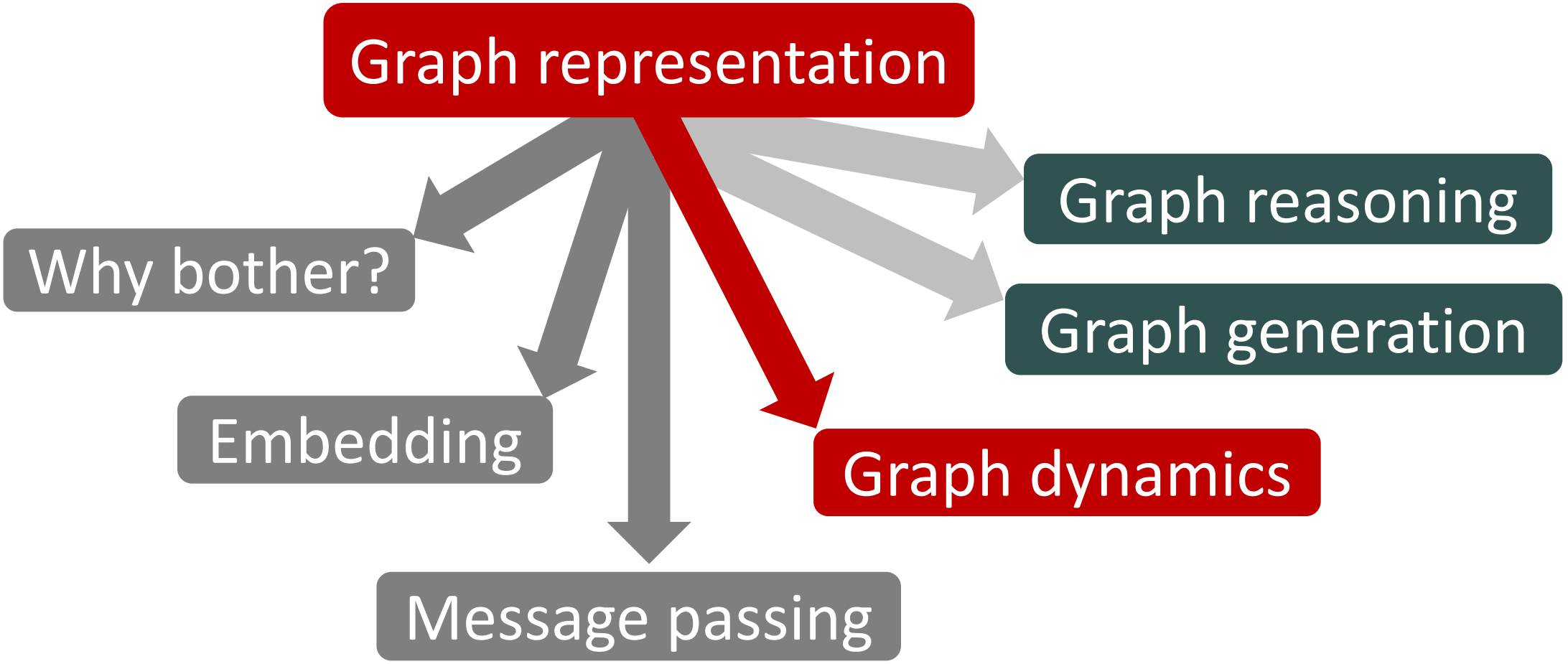


Figure 1: **Left:** The attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ employed by our model, parametrized by a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$. **Right:** An illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or average to obtain $\vec{h}'_1$.


Graph representation
Why bother?
Embedding
Message passing
Graph dynamics
Graph reasoning
Graph generation

Graph morphism

Input: Graph

Output: A new graph. Same nodes, different edges.

Model: Graph morphism

Method: Graph transformation policy network (GTPN)

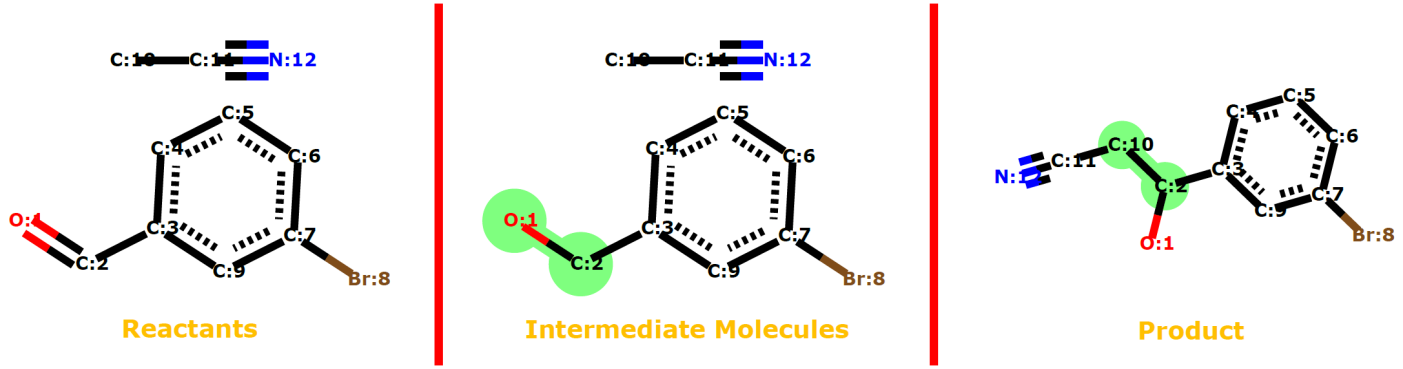


Figure 1: A sample reaction represented as a set of graph transformations from reactants (leftmost) to products (rightmost). Atoms are labeled with their type (Carbon, Oxygen,...) and their index (1, 2,...) in the molecular graph. The atom pairs that change connectivity and their new bonds (if existed) are highlighted in green. There are two bond changes in this case: 1) The double bond between O:1 and C:2 becomes single. 2) A new single bond between C:2 and C:10 is added.

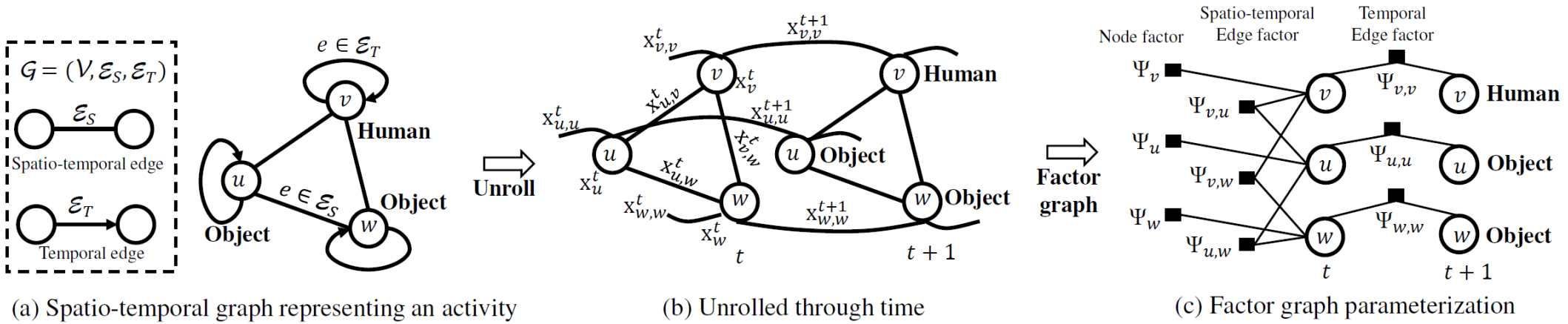
Kien Do, Truyen Tran, and Svetha Venkatesh. "Graph Transformation Policy Network for Chemical Reaction Prediction." *KDD'19*.

Graph recurrence

Graphs that represent interaction between entities through time

Spatial edges are node interaction at a time step

Temporal edges are consistency relationship through time



(a) Spatio-temporal graph representing an activity

(b) Unrolled through time

(c) Factor graph parameterization

Challenges

The addition of temporal edges make the graphs bigger, more complex

- Relying on context specific constraints to reduce the complexity by approximations

Through time, structures of the graph may change

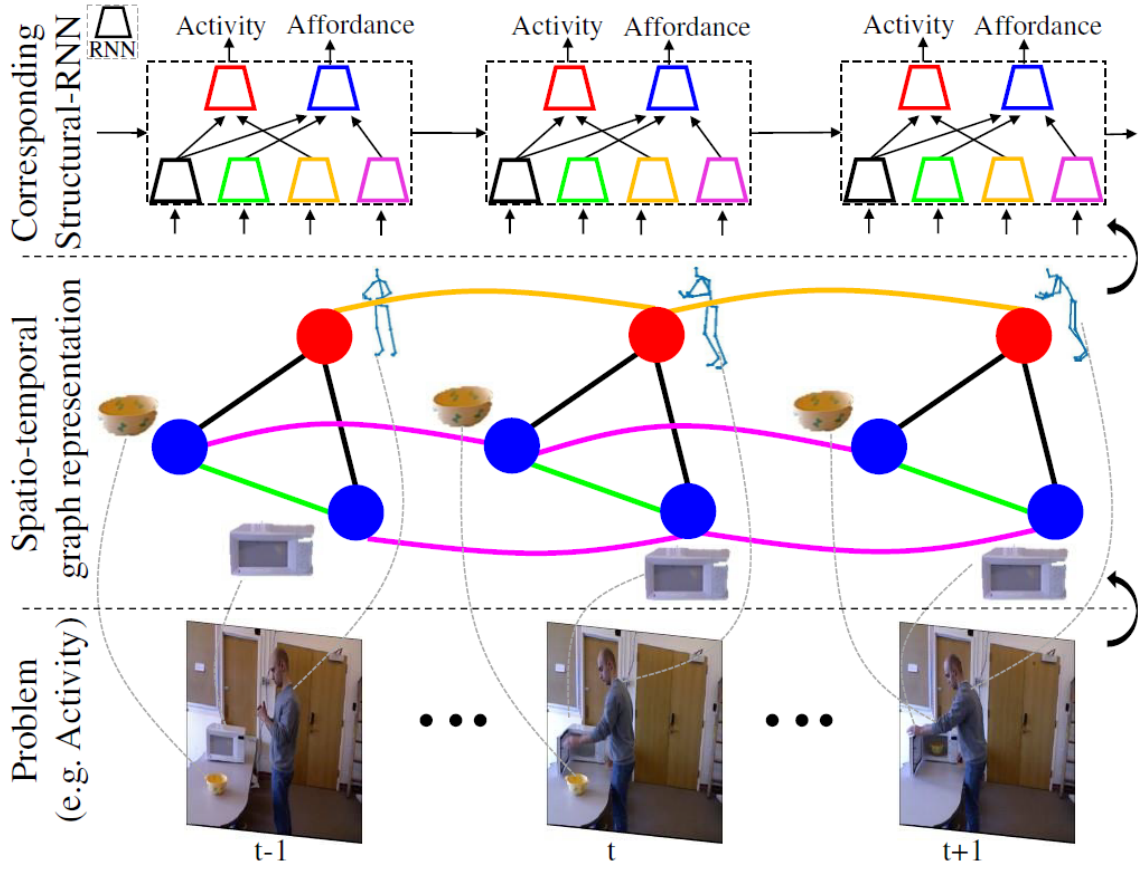
- Hard to solve, most methods model short sequences to avoid this

Structural RNN

CVPR16 best student paper Saxena group, Cornell

Example: human microwaving food

- Middle: s-t graph capturing spatial and temporal interactions between the human and the objects.
- Top: Schematic representation of structural-RNN architecture



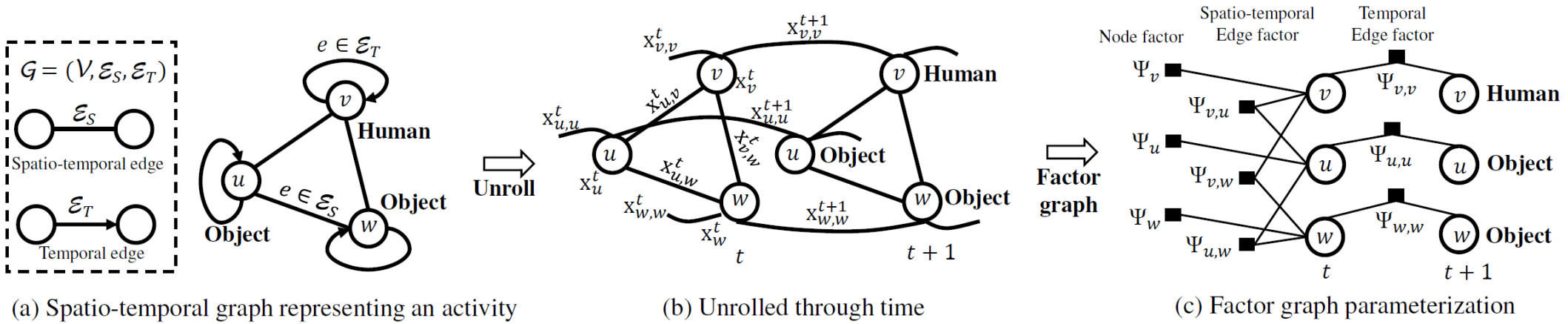
Structural RNN

Node features: human and object poses,

Edge features: relative orientation

Node labels: human activity and object affordance.

- affected by both its node and its interactions with other nodes (edges)



(a) Spatio-temporal graph representing an activity

(b) Unrolled through time

(c) Factor graph parameterization

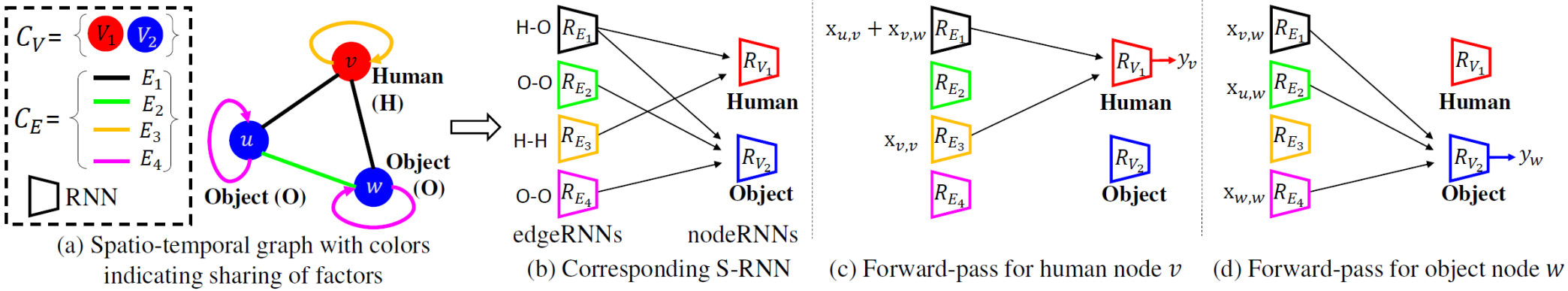
Structural RNN – from s-t graph

Each factor is represented as a RNN

Form a feed forward bipartite graph from edge factors to node factors

Sharing factors between node and edge of same semantics type

- More compact
- Support node # changes



(a) Spatio-temporal graph with colors indicating sharing of factors

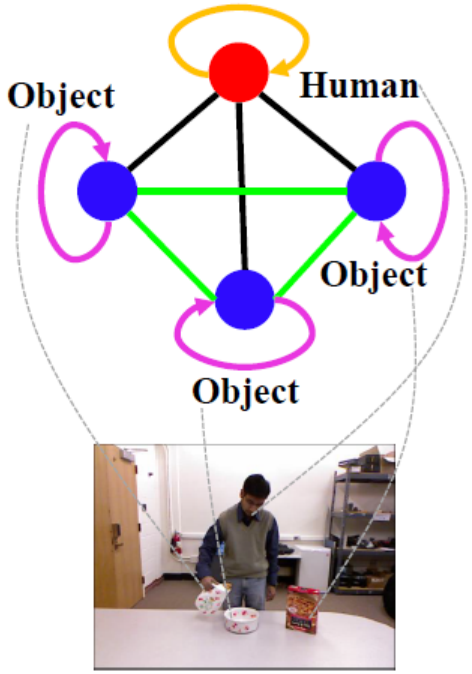
(b) Corresponding S-RNN

(c) Forward-pass for human node v

(d) Forward-pass for object node w

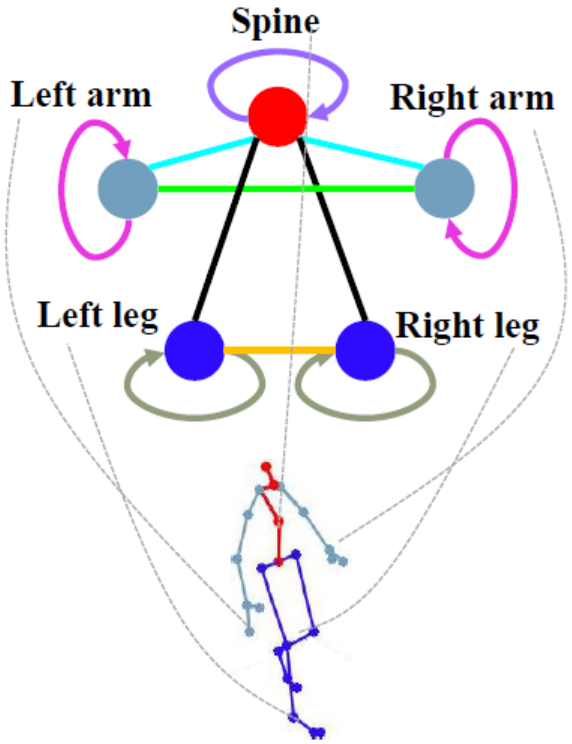
Structural RNN - Applications

Activity detection and anticipation

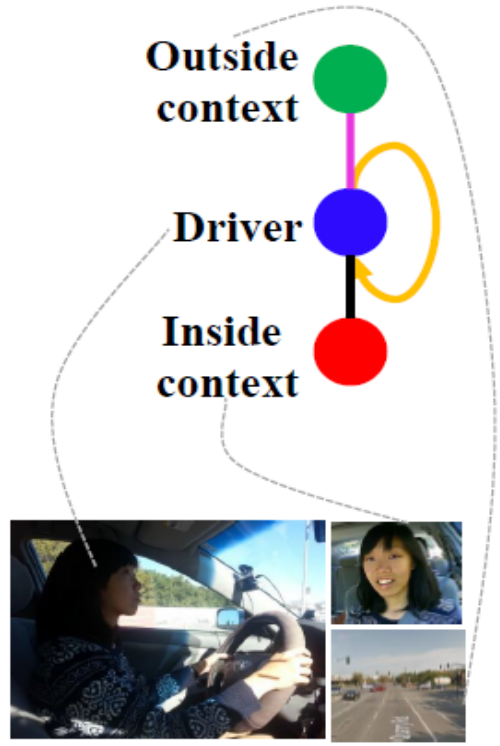


tivity detection and anticipation

Skeleton tracking

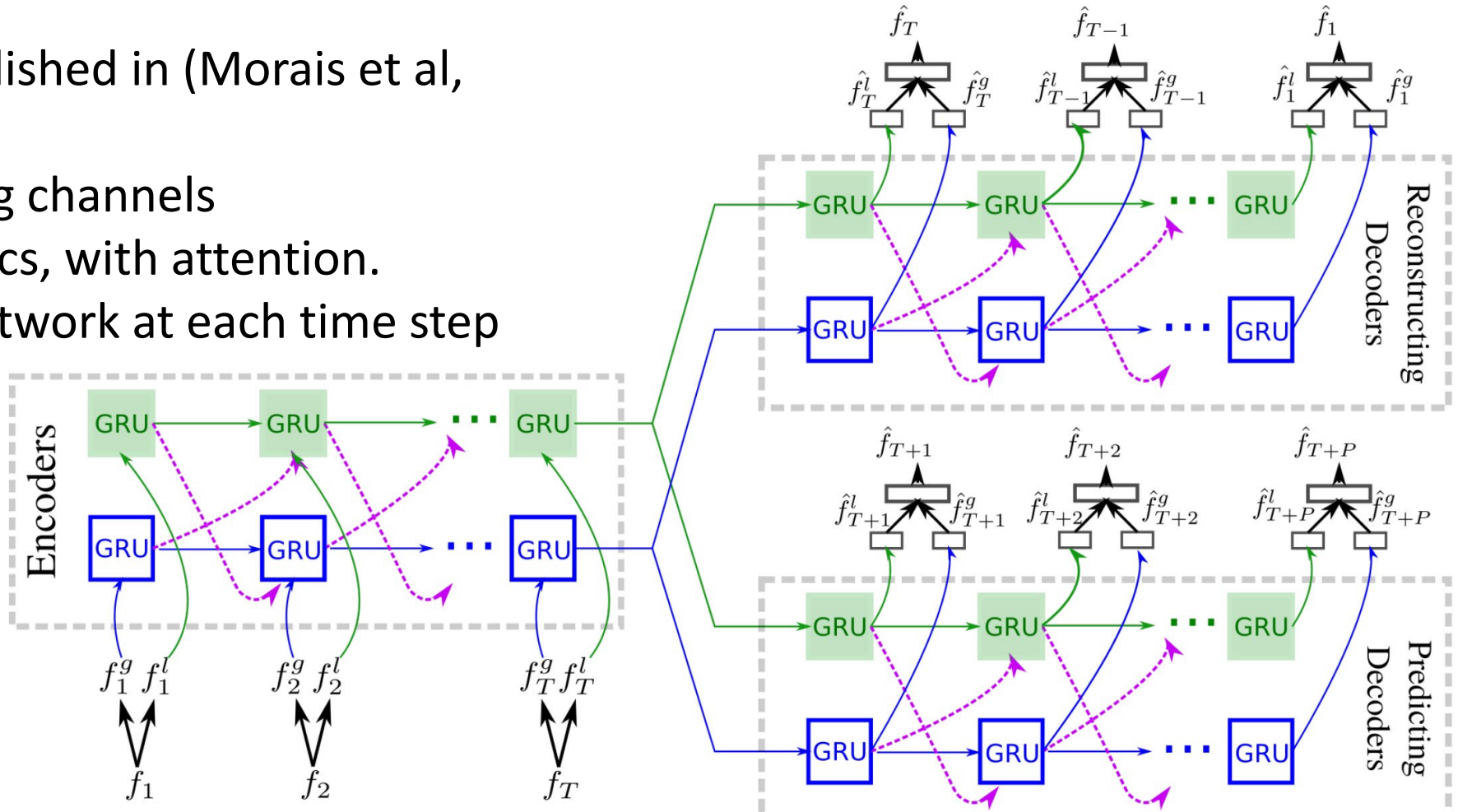


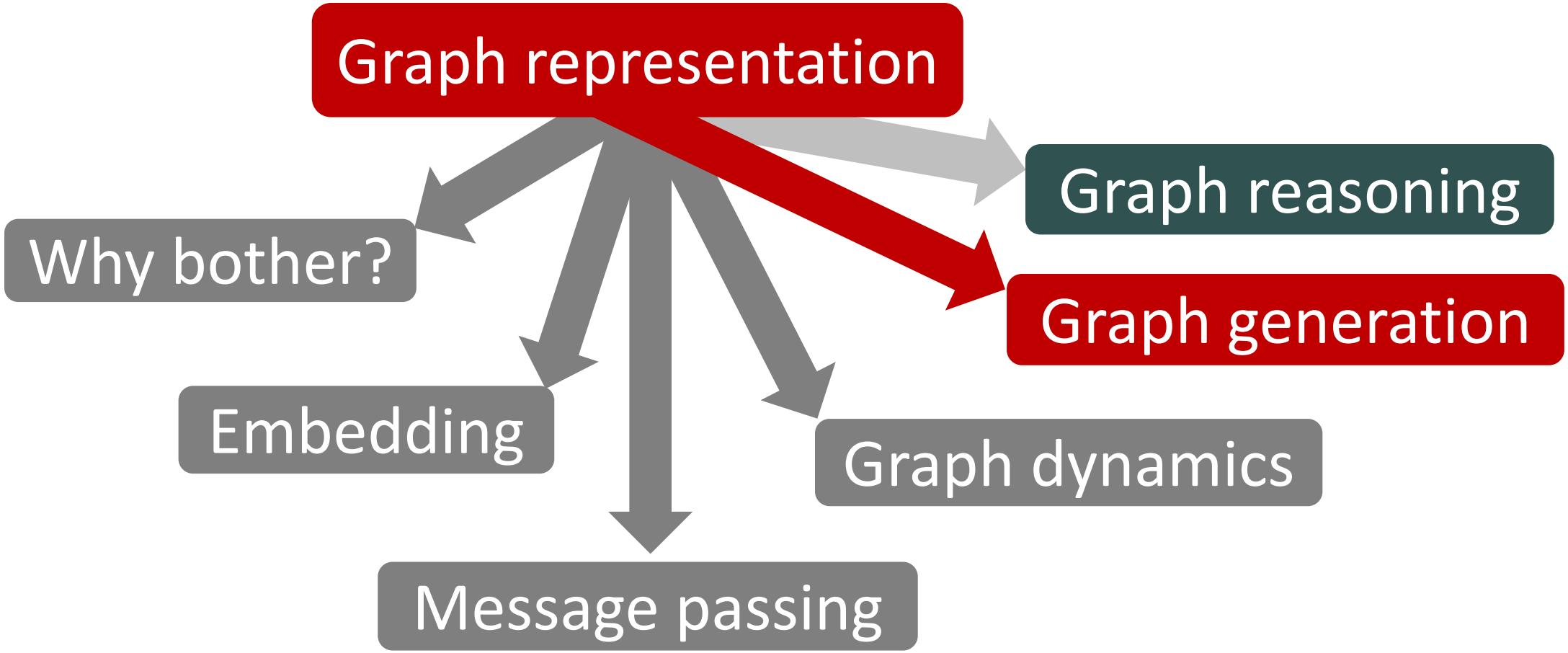
Driver maneuver prediction



Message-Passing Encoder-Decoder Recurrent Net

- Prelim version published in (Morais et al, CVPR'19)
- Multiple interacting channels
- Graphs are dynamics, with attention.
- Mixture density network at each time step




Graph representation
Why bother?
Graph reasoning
Graph generation
Embedding
Graph dynamics
Message passing

Technical challenges

No regular structures (e.g. grid, sequence,…)

Graphs are ***permutation invariant***:

- #permutations are exponential function of #nodes
- The probability of a generated graph G need to be marginalized over all possible permutations

Generating graphs with **variable size**

Diversity of generated graphs

Smoothness of latent space

Generation methods

$$\mathbf{v} \sim P_{model}(\mathbf{v})$$
$$P_{model}(\mathbf{v}) \approx P_{data}(\mathbf{v})$$

Classical random graph models, e.g., An exponential family of probability distributions for directed graphs (Holland and Leinhardt, 1981)

Deep Generative Model Methods:

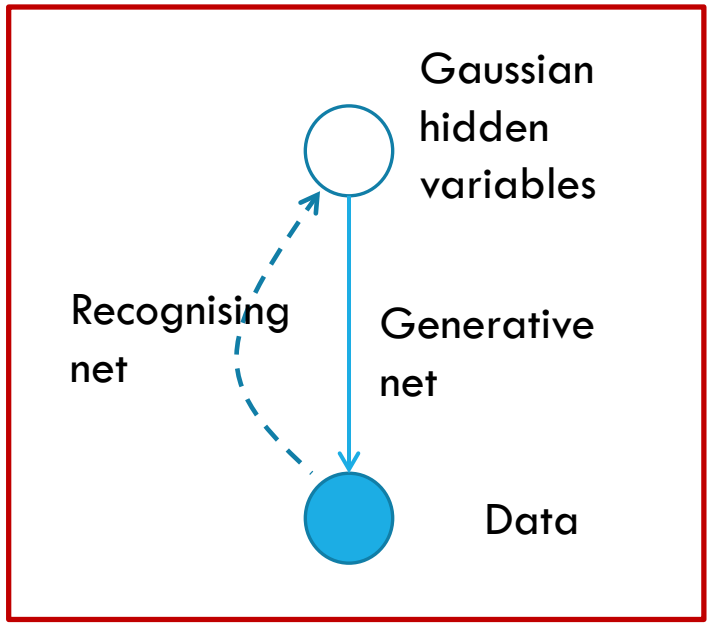
- Variational Graph AutoEncoders
- Graphite: Iterative Generative Modeling of Graphs
- GraphVAE: Towards Generation of Small Graph using Variational AutoEncoder
- Junction Tree Variational AutoEncoder for Molecular Graph Generation

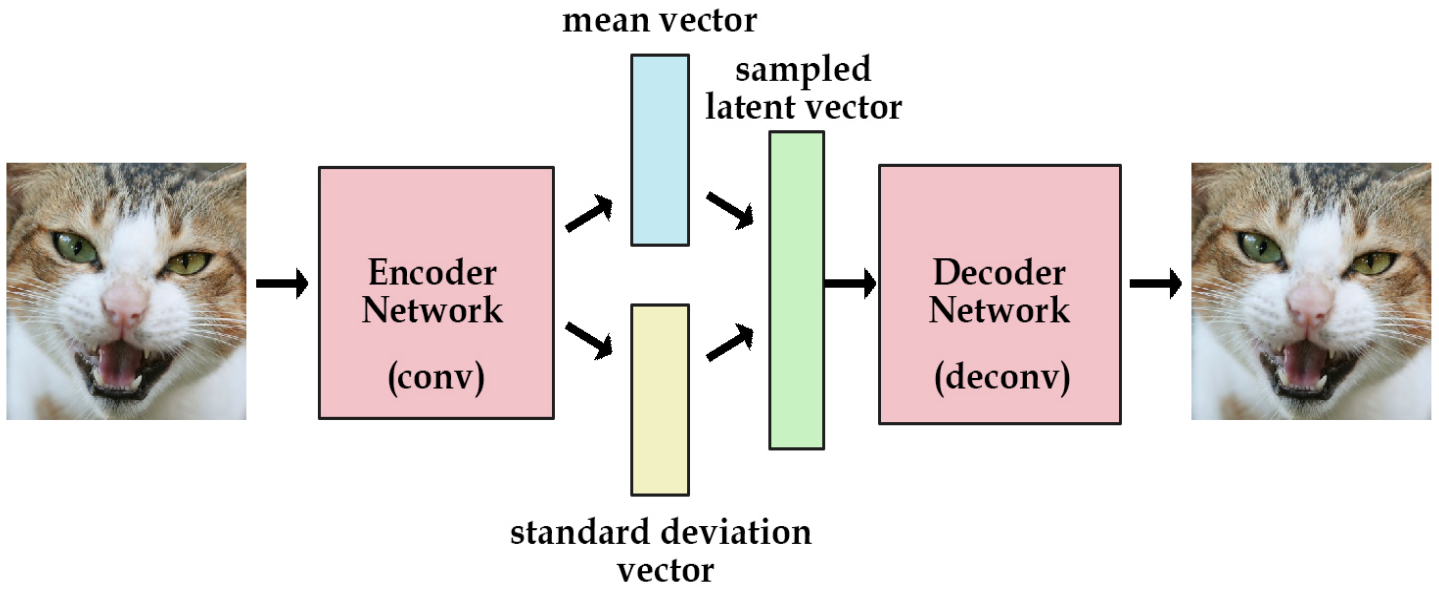
Sequence-based & RL method:

- GraphRNN - A Deep Generative Model for Graphs
- Multi-Objective *De Novo* Drug Design with Conditional Graph Generative Model

Variational Autoencoder
(Kingma & Welling, 2013)

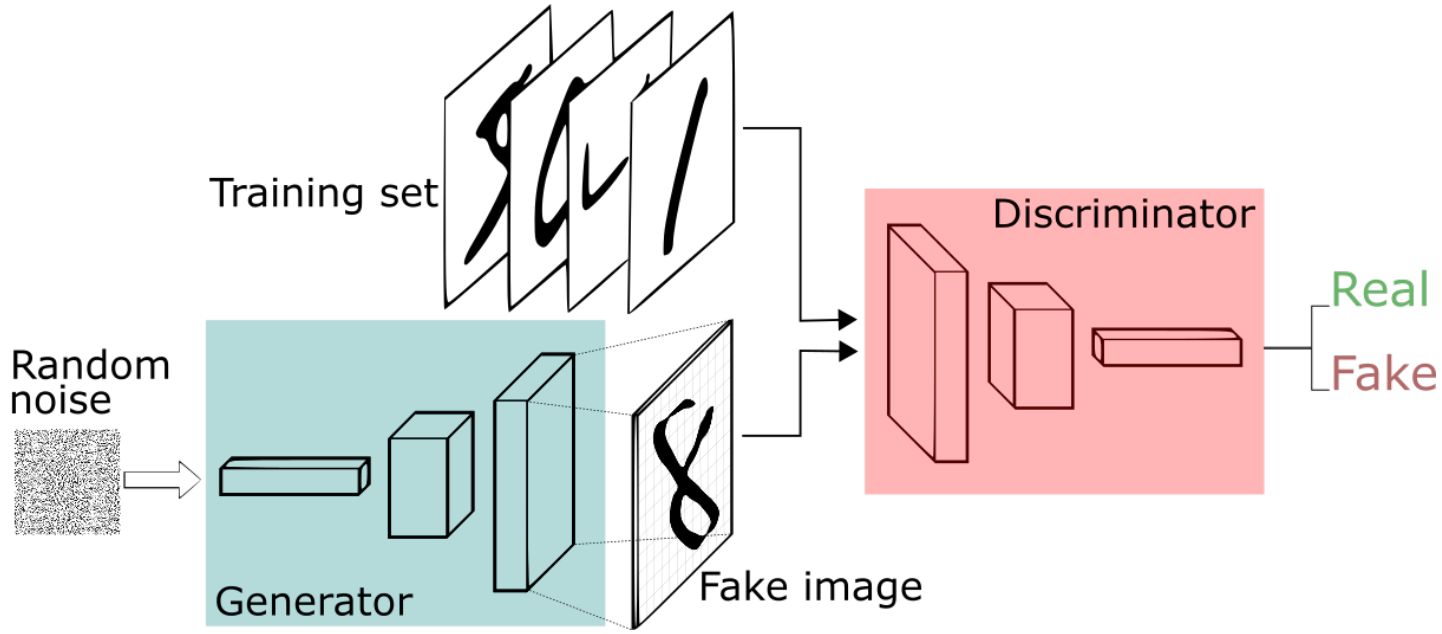
Two separate processes: generative (hidden → visible) versus recognition (visible → hidden)



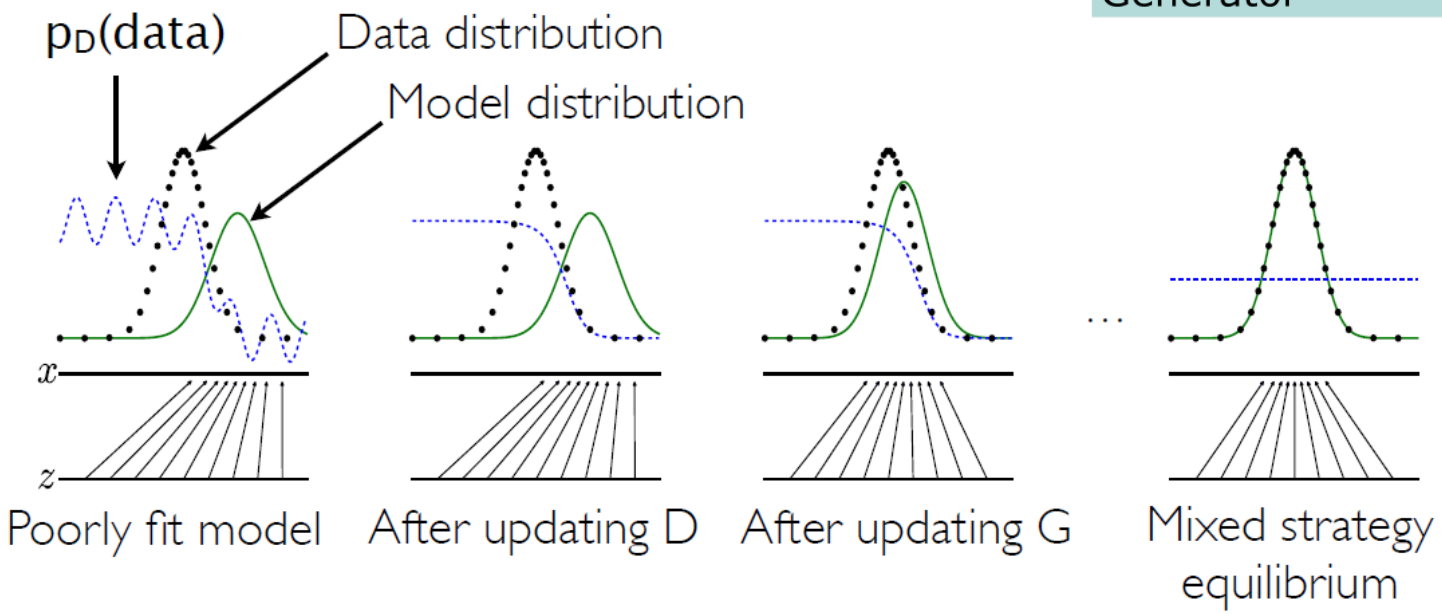


http://kvfrans.com/variational-autoencoders-explained/

Generative Adversarial Networks (Goodfellow et al,2014)



GAN architecture. Source: DL4J



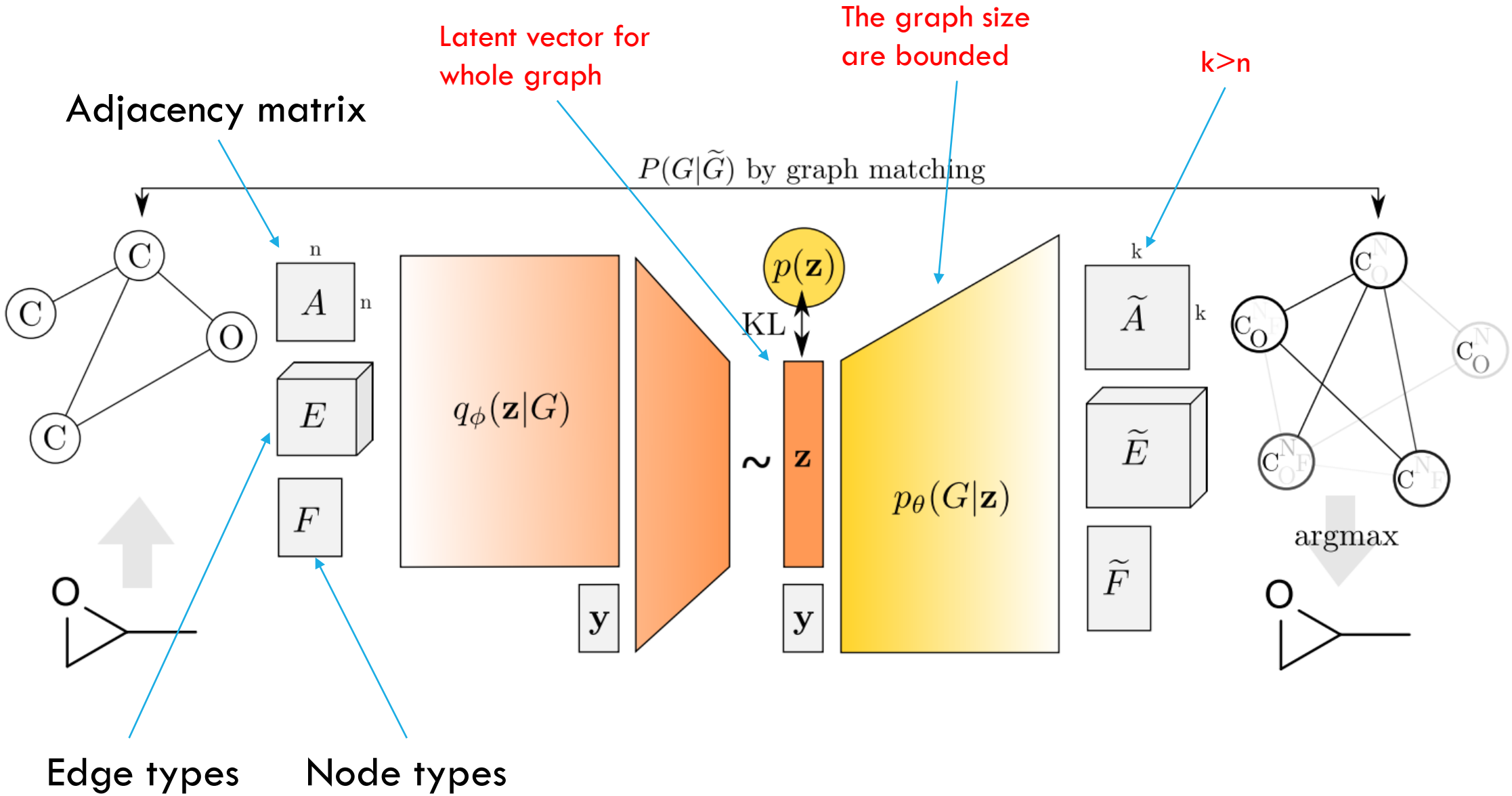
(Goodfellow's, NIPS 2014)

Variational methods

Minimize the upper bound on the negative log-likelihood, equivalent to maximizing the ELBO:

$$\mathcal{L}(\phi, \theta; G) = \mathbb{E}_{q_\phi(\mathbf{z}|G)}[-\log p_\theta(G|\mathbf{z})] + \mathrm{KL}[q_\phi(\mathbf{z}|G)||p(\mathbf{z})]$$

GraphVAE (Simonovsky and Komodakis)



Junction Tree VAE (Jin et. al.)

Algorithm 2 Tree decomposition of molecule $G = (V, E)$

$V_1 \leftarrow$ the set of bonds $(u, v) \in E$ that do not belong to any rings.
$V_2 \leftarrow$ the set of simple rings of G.
for r_1, r_2 **in** V_2 **do**
  Merge rings r_1, r_2 into one ring if they share more than two atoms (bridged rings).
end for
$V_0 \leftarrow$ atoms being the intersection of three or more clusters in $V_1 \cup V_2$.
$\mathcal{V} \leftarrow V_0 \cup V_1 \cup V_2$
$\mathcal{E} \leftarrow \{(i, j, c) \in \mathcal{V} \times \mathcal{V} \times \mathbb{R} \mid |i \cap j| > 0\}$. Set $c = \infty$ if $i \in V_0$ or $j \in V_0$, and $c = 1$ otherwise.
Return The maximum spanning tree over cluster graph $(\mathcal{V}, \mathcal{E})$.

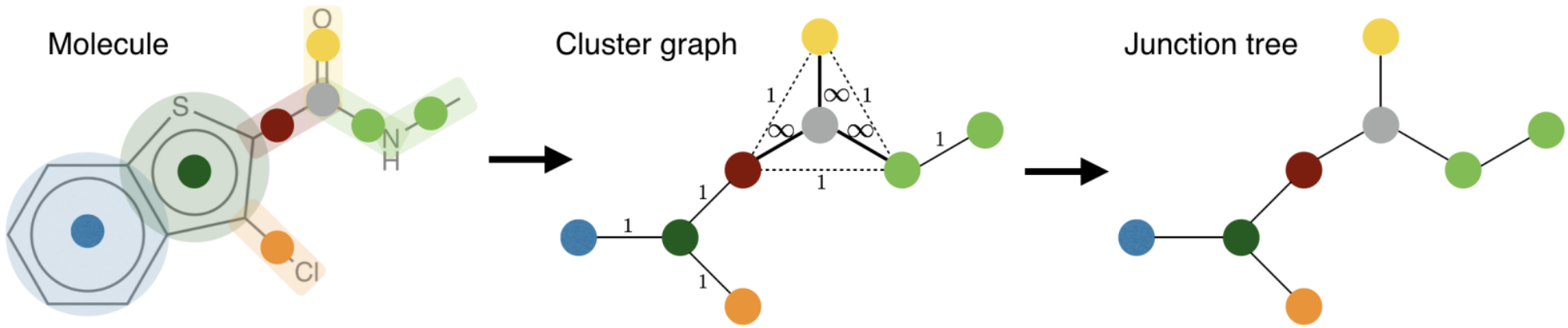
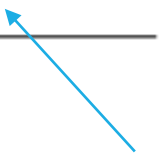


Table 1. Reconstruction accuracy and prior validity results. Baseline results are copied from Kusner et al. (2017); Dai et al. (2018); Simonovsky & Komodakis (2018).

Method	Reconstruction	Validity
CVAE	44.6%	0.7%
GVAE	53.7%	7.2%
SD-VAE[2]	76.2%	43.5%
GraphVAE	-	13.5%
JT-VAE	**76.7%**	**100.0%**



Constraint during adding nodes

GraphRNN

A case of graph dynamics: nodes and edges are added sequentially.

Solve tractability using BFS

You, Jiaxuan, et al. "GraphRNN: Generating realistic graphs with deep auto-regressive models." *ICML* (2018).

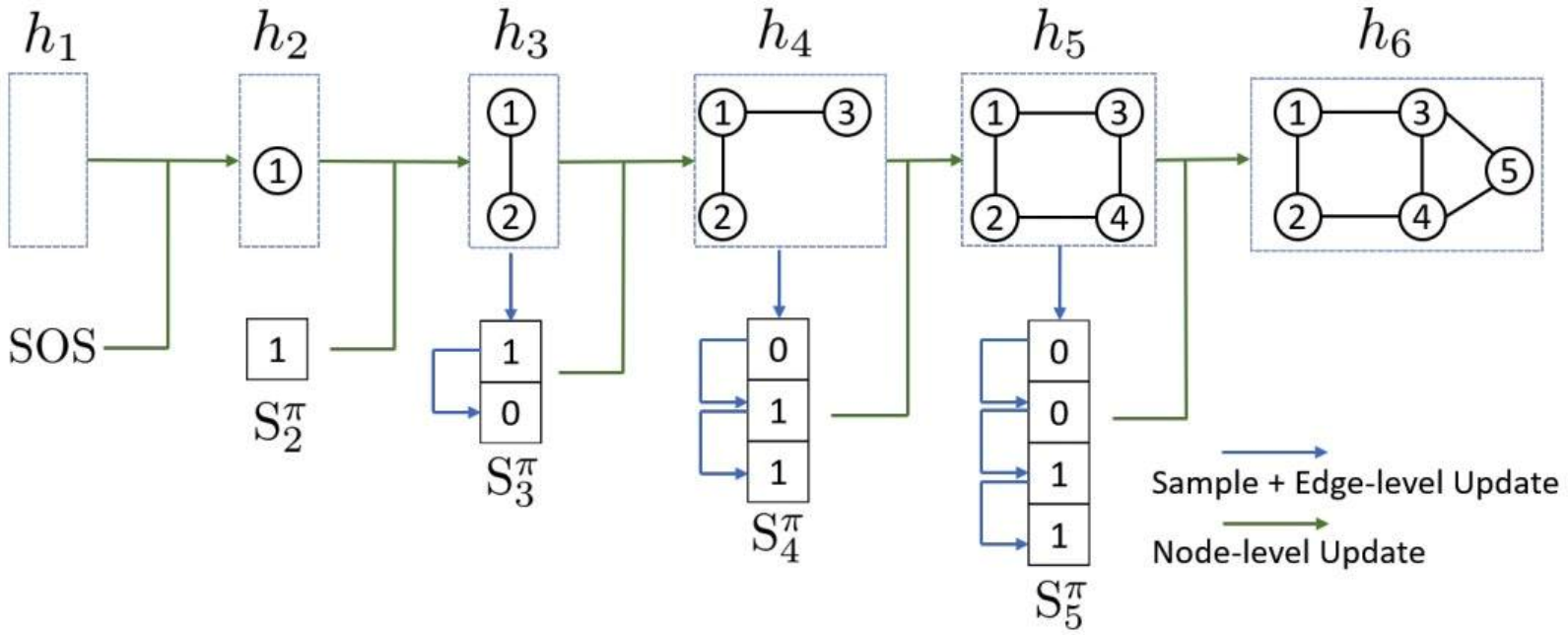
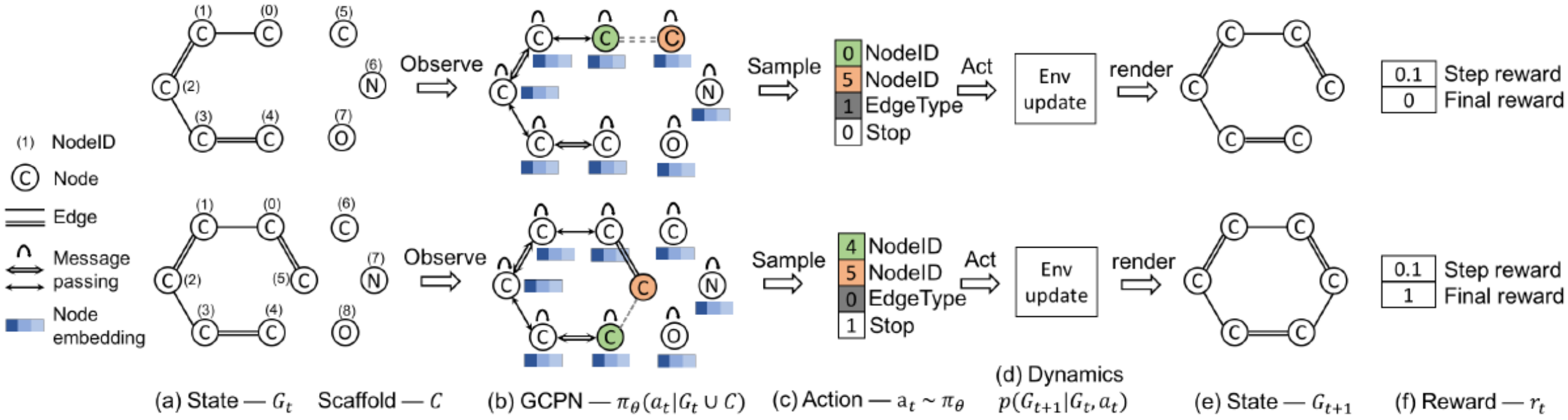


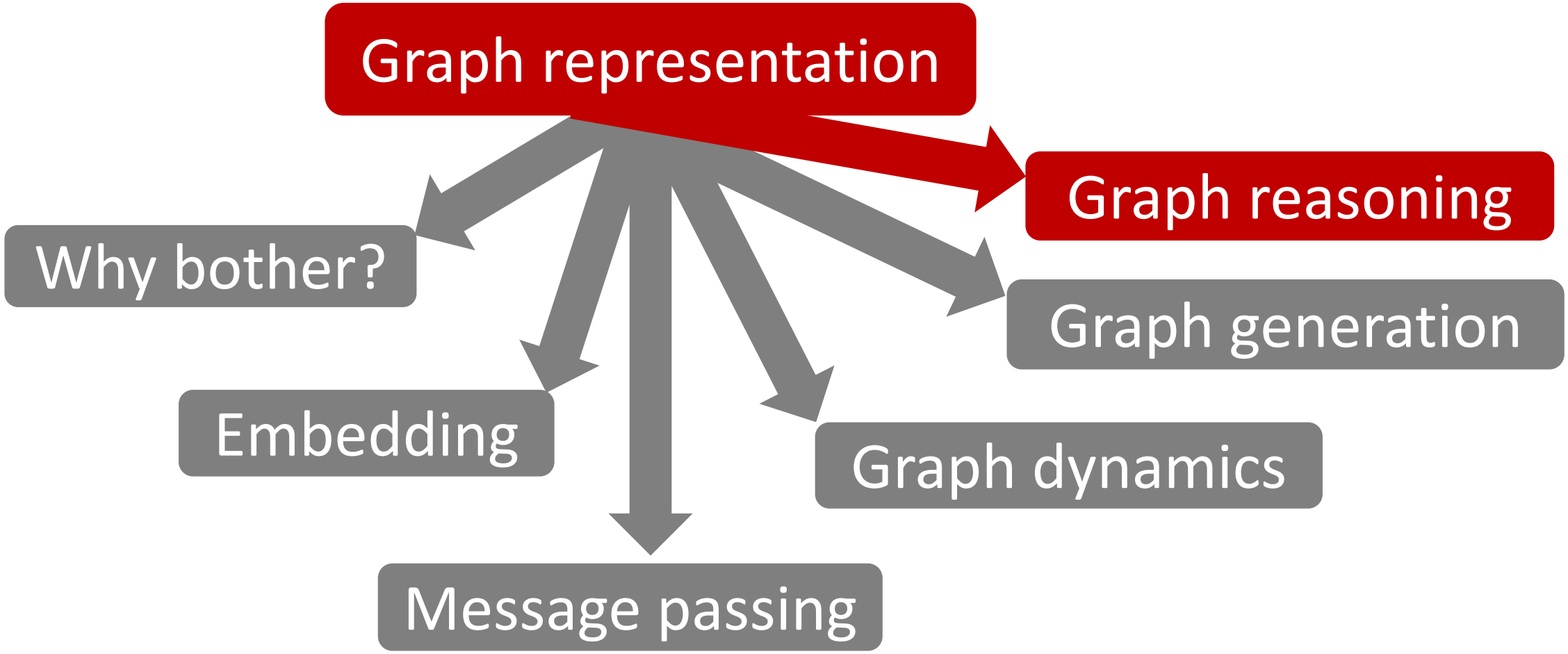
Figure 1. GraphRNN at inference time. Green arrows denote the graph-level RNN that encodes the "graph state" vector h_i in its hidden state, updated by the predicted adjacency vector S_i^π for node $\pi(v_i)$. Blue arrows represent the edge-level RNN, whose hidden state is initialized by the graph-level RNN, that is used to predict the adjacency vector S_i^π for node $\pi(v_i)$.

Graphs step-wise construction using reinforcement learning

Graph rep (message passing) | graph validation (RL) | graph faithfulness (GAN)



You, Jiaxuan, et al. "Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation." *NeurIPS* (2018).

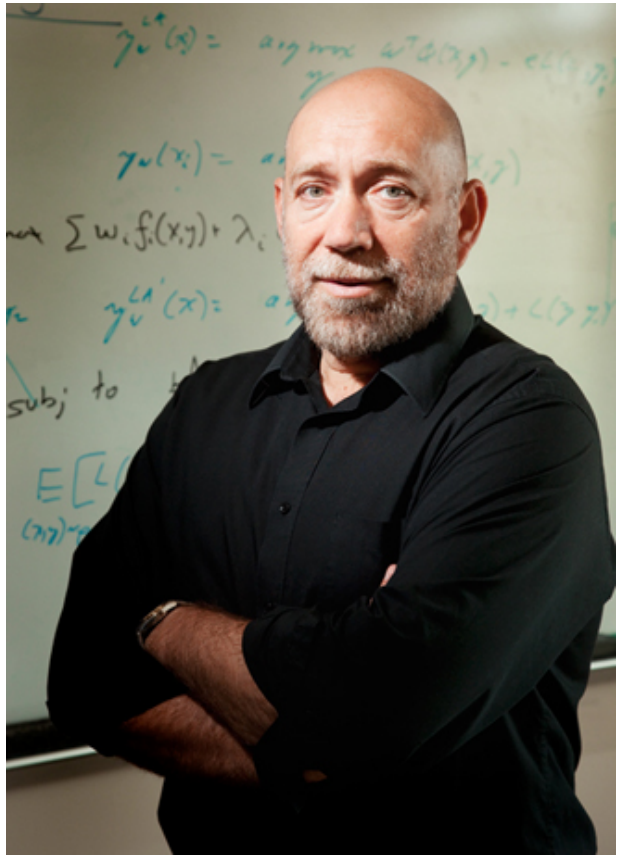

Graph representation
Graph reasoning
Why bother?
Graph generation
Embedding
Graph dynamics
Message passing

Reasoning

Reasoning is to deduce knowledge from previously acquired knowledge in response to a query (or a cue)

Early theories of intelligence:

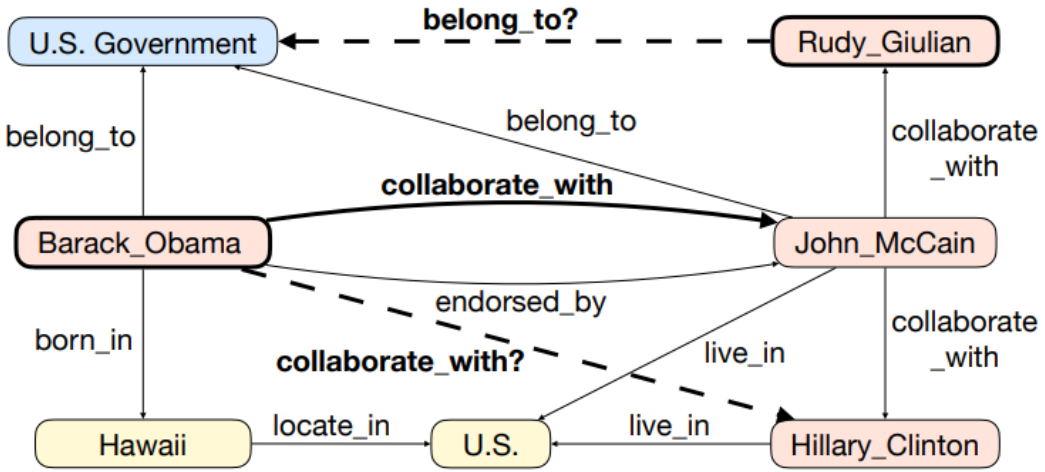
- focuses solely on reasoning,
- learning can be added separately and later! (Khardon & Roth, 1997).



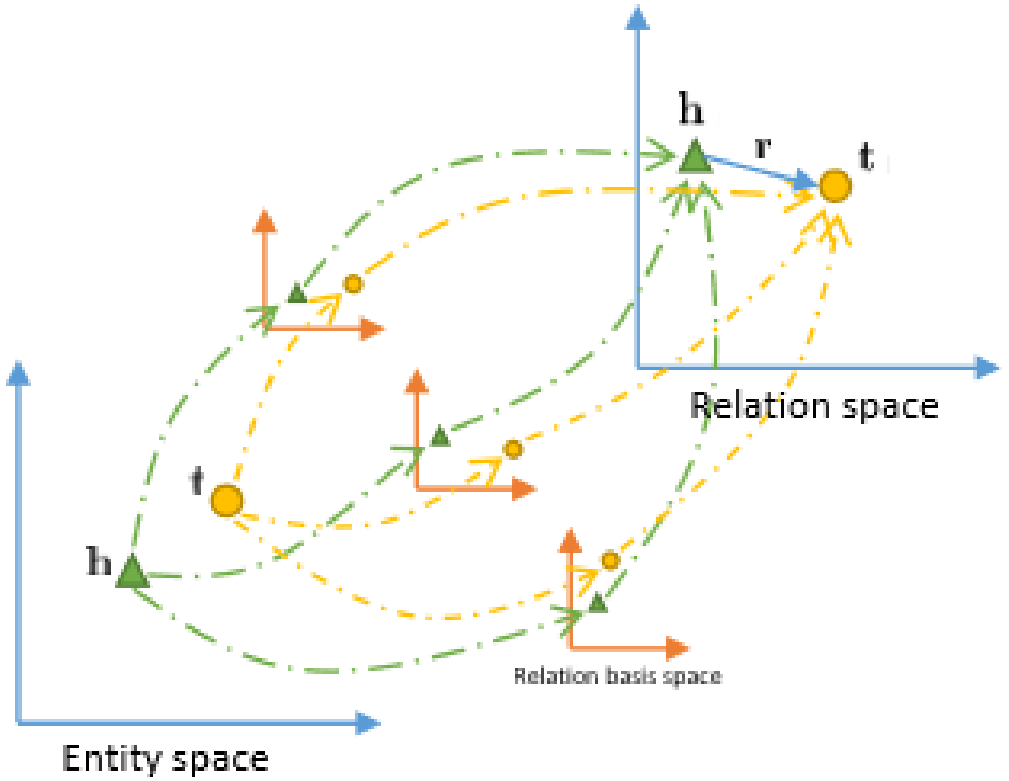
(Dan Roth; *ACM Fellow; IJCAI John McCarthy Award*)

Khardon, Roni, and Dan Roth. "Learning to reason." *Journal of the ACM (JACM)* 44.5 (1997): 697-725.

Inferring relations

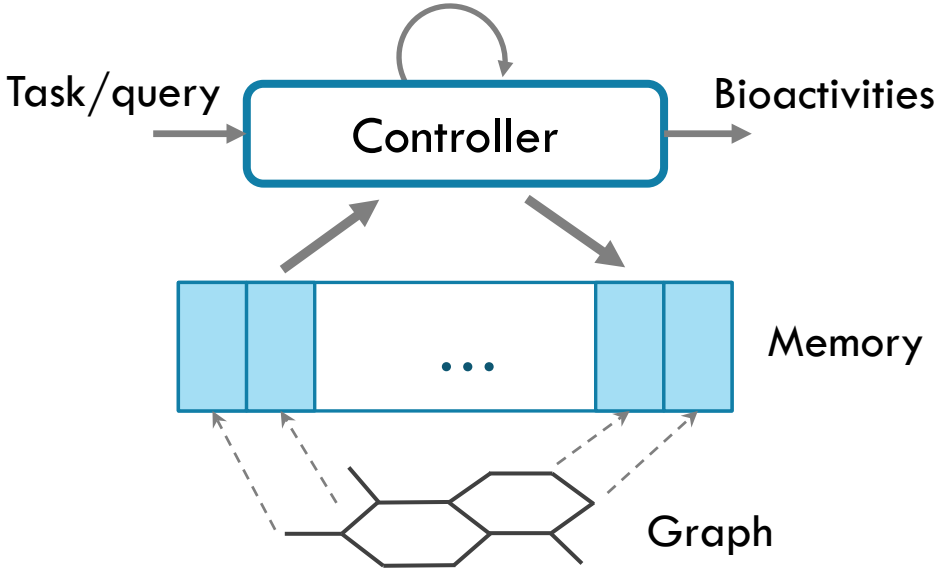


https://www.zdnet.com/article/salesforce-research-knowledge-graphs-and-machine-learning-to-power-einstein/

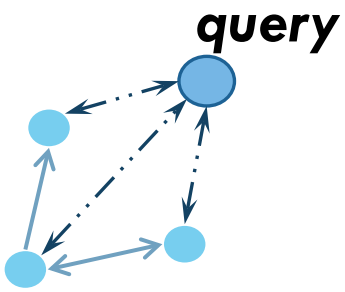


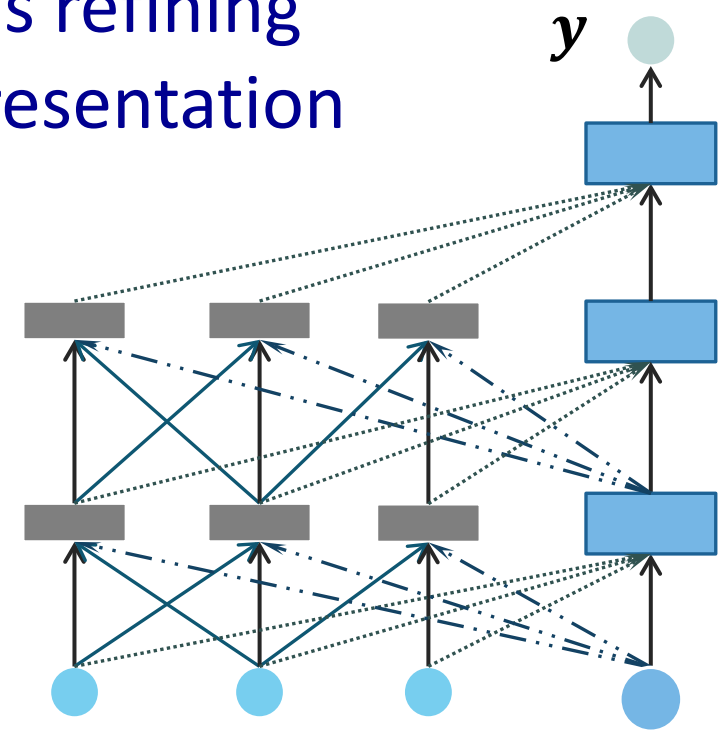
Do, Kien, Truyen Tran, and Svetha Venkatesh. "Knowledge graph embedding with multiple relation projections." *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018.

Querying a graph



Message passing as refining node & query representation

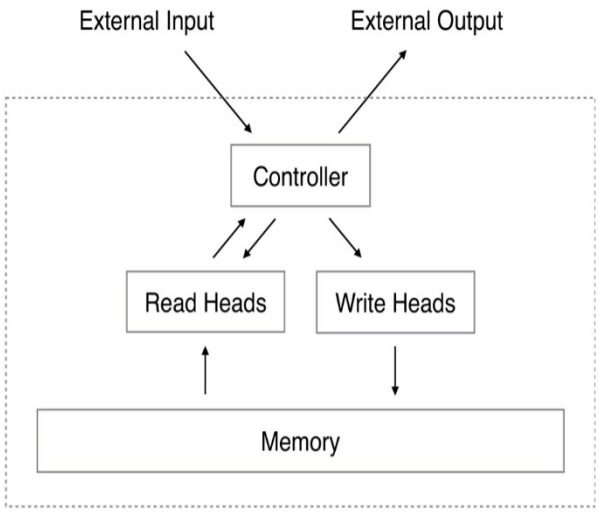




#Ref: Pham, Trang, Truyen Tran, and Svetha Venkatesh. "Graph Memory Networks for Molecular Activity Prediction." *ICPR'18*.

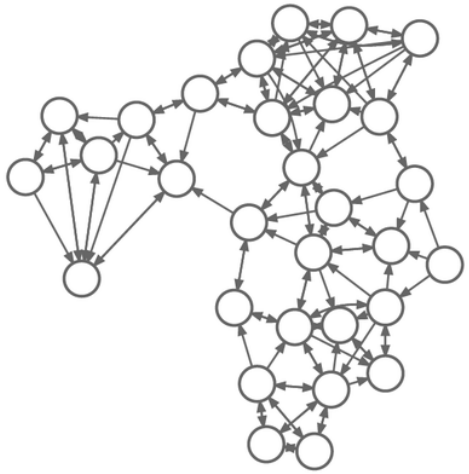
Querying a graph (2)

Neural Turing Machine (Grave et al, 2014)

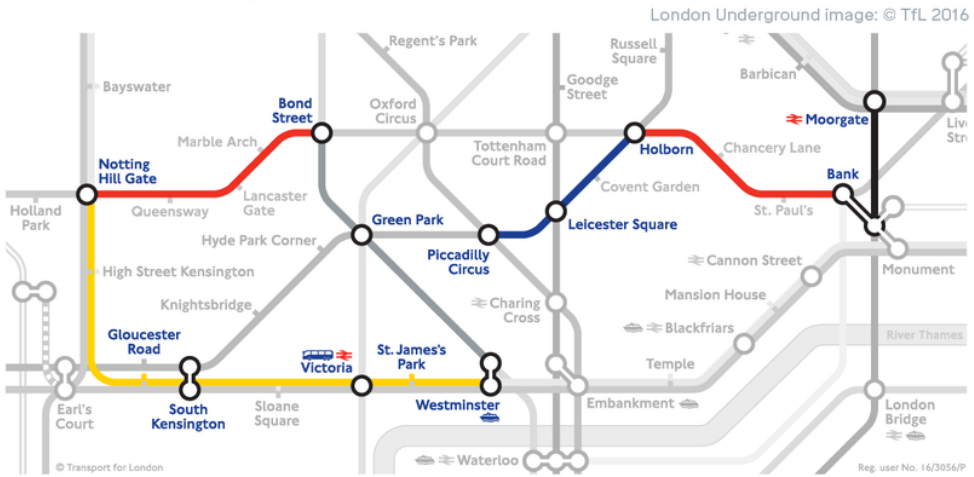


https://rylanschaeffer.github.io/content/research/neural_turing_machine/main.html

Random Training Graph



London Underground



Traversal

Shortest

Underground Input:

(OxfordCircus, TottenhamCtRd, Central)
(TottenhamCtRd, OxfordCircus, Central)
(BakerSt, Marylebone, Circle)
(BakerSt, Marylebone, Bakerloo)
(BakerSt, OxfordCircus, Bakerloo)
...
(LeicesterSq, CharingCross, Northern)
(TottenhamCtRd, LeicesterSq, Northern)
(OxfordCircus, PiccadillyCircus, Bakerloo)
(OxfordCircus, NottingHillGate, Central)
(OxfordCircus, Euston, Victoria)

- 84 edges in total

Traversal Question:

(BondSt, _, Central),
(_, _, Circle), (_, _, Circle),
(_, _, Circle), (_, _, Circle),
(_, _, Jubilee), (_, _, Jubilee),

Answer:

(BondSt, NottingHillGate, Central)
(NottingHillGate, GloucesterRd, Circle)
...
(Westminster, GreenPark, Jubilee)
(GreenPark, BondSt, Jubilee)

Shortest Path Question:

(Moorgate, PiccadillyCircus, _)

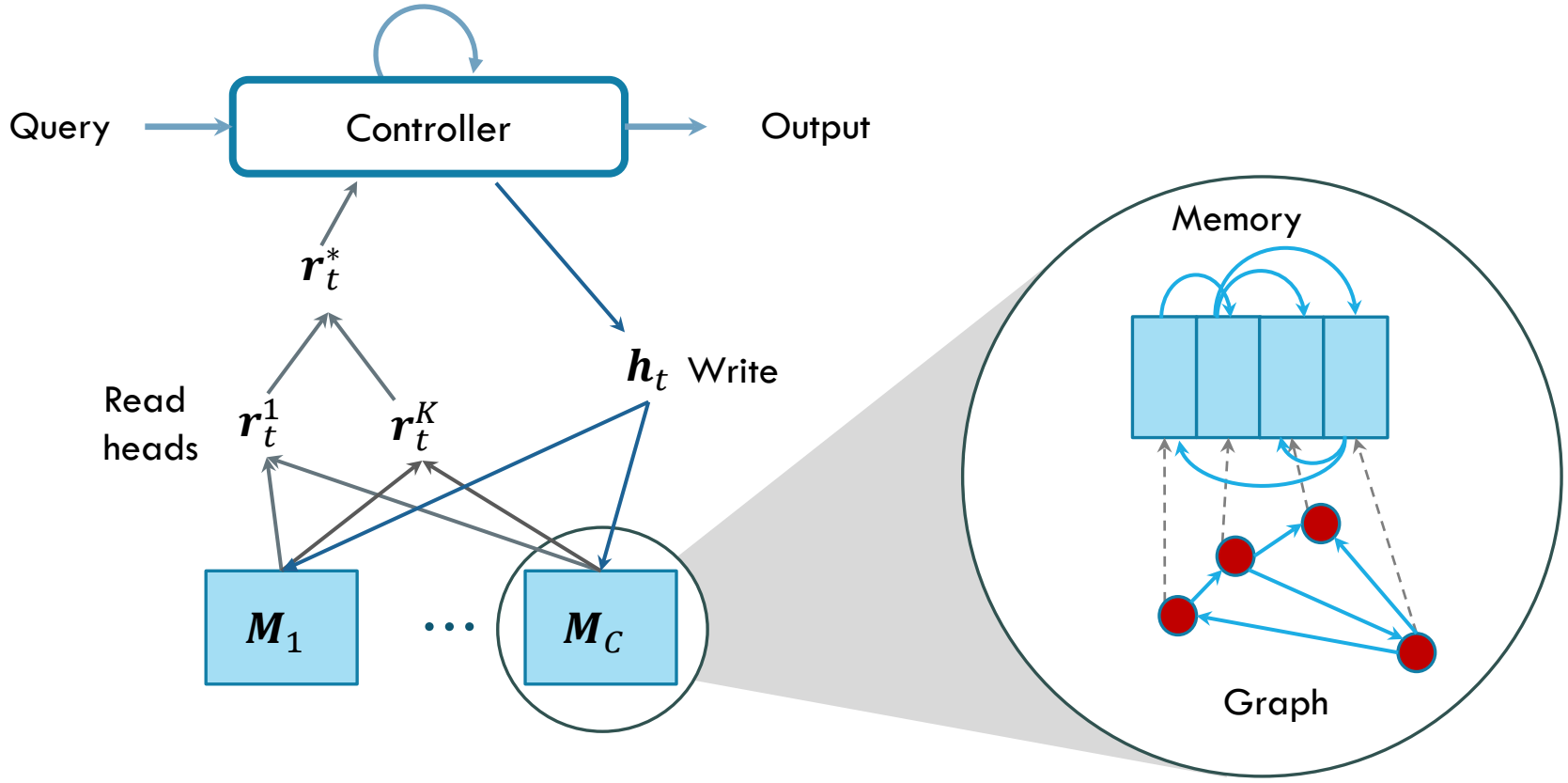
Answer:

(Moorgate, Bank, Northern)
(Bank, Holborn, Central)
(Holborn, LeicesterSq, Piccadilly)
(LeicesterSq, PiccadillyCircus, Piccadilly)

Differentiable Neural Computer (Grave et al, 2016)

Querying multiple graphs

- Detects high-order patterns from disconnected paths
- Learns graph similarity
- Models graph-graph interaction
- Supports structured queries



Pham, Trang, Truyen Tran, and Svetha Venkatesh. "Relational dynamic memory networks." *arXiv preprint arXiv:1808.04247*(2018).

On going and future work

Graph translation, graph2graph, graph2seq

Graphs in context (e.g., crystals, human activities in scene)

Semi-supervised learning

Manifolds learning

Graph as hidden layers (e.g., see Ying et al, NIPS'19)

Generative models of sequence of graphs

Theoretical properties of graphs (e.g., Xu et al, ICLR'19)

Graph matching

Higher-order graph neural networks (e.g., Morris, AAAI'19)

Graphs for logical inference

Graphs of multi-agent communication

Continuous time graph dynamics

Graph density – anomaly detection

For more refs

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2019). A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., ... & Gulcehre, C. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint* arXiv:1806.01261.

Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151, 78-94.

Lee, J. B., Rossi, R. A., Kim, S., Ahmed, N. K., & Koh, E. (2018). Attention models in graphs: A survey. *arXiv preprint arXiv:1807.07984*.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4), 18-42.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*.

Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*. 104.1, 11-33.

The graph team @

A/Prof. Truyen Tran

Dr Vuong Le

Dr Thin Nguyen

Dr Trang Pham
(Now @Google)

Mr Tin Pham

Mr Kien Do

Mr Thao Minh Le

Thank you!

We're hiring

PhD & Postdocs

truyen.tran@deakin.edu.au

https://truyentran.github.io/scholarship.html

http://ahsanqawl.com/2015/10/qa/