Relational Reasoning with Graph Neural Networks

Jian Tang

Mila-Quebec AI Institute

HEC Montreal

Email: jian.tang@hec.ca



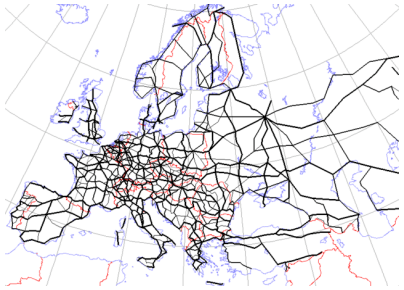


Relational Data/Graphs are Ubiquitous

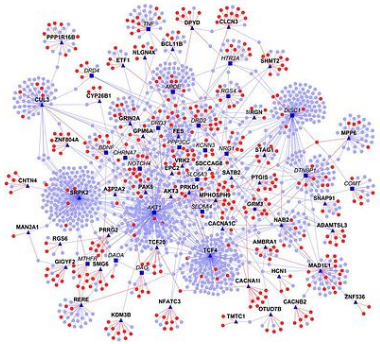
- Graphs: a general and flexible data structure to encode the relations between objects



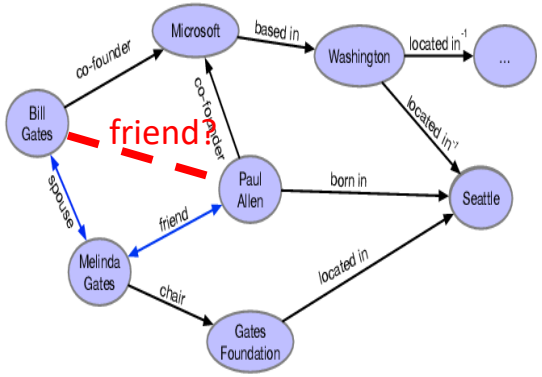
Social Graph



Road Graph



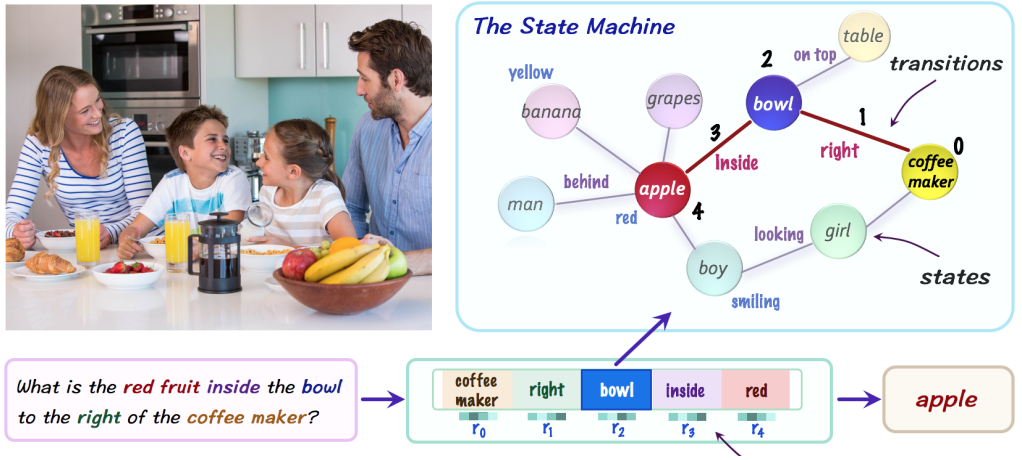
Protein-protein Interaction Graph



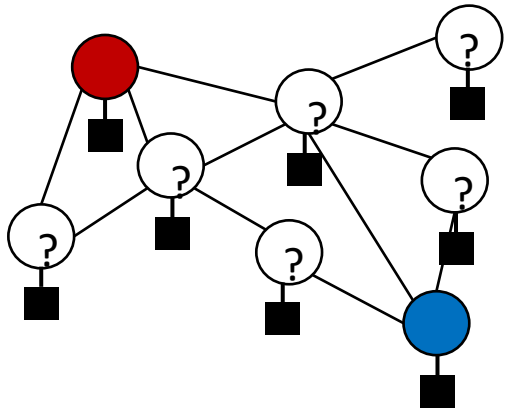
Knowledge Graph

Graphs as Bridges Between System I and System II Reasoning

- Existing deep learning systems are mainly good at **perception**
  - E.g., recognize the objects in images
  - **System I Reasoning**
- Real-world problems are very complex
  - Need to understand the relationships between different facts or high-level semantic variables
  - **System II Reasoning**
- Using graphs to capture the relationship between high-level semantic variables



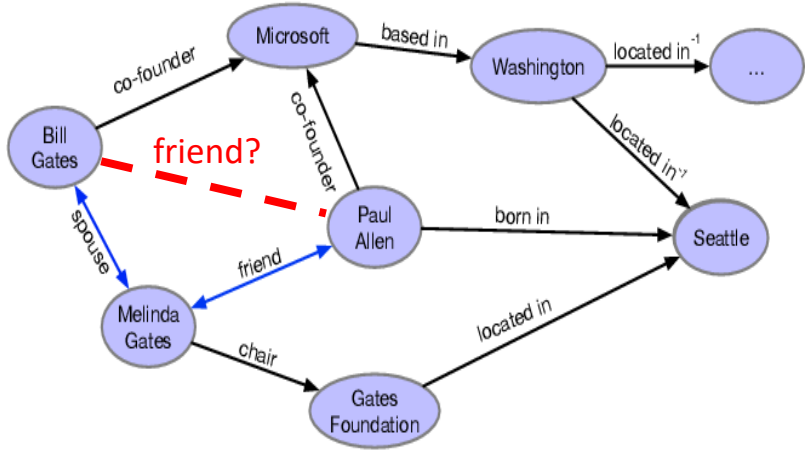
Relational Prediction and Reasoning



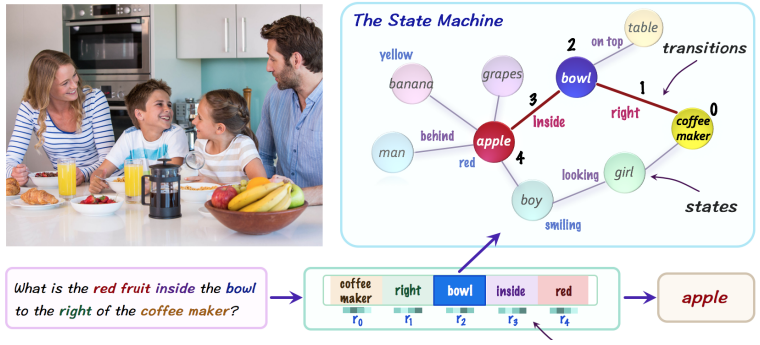
Object labels

Object features

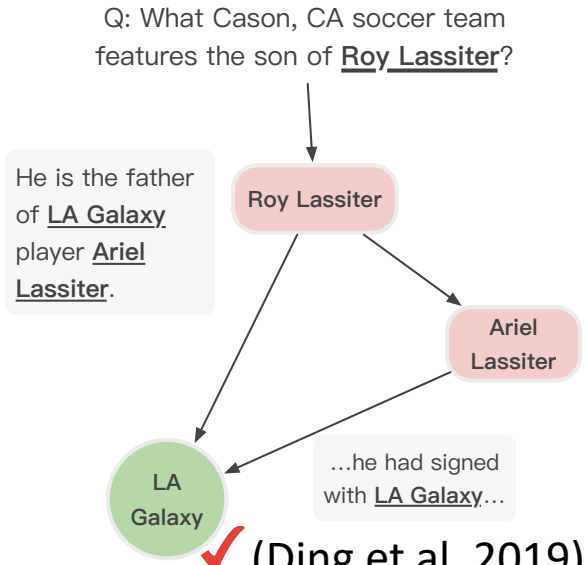
Node classification



Reasoning on knowledge graphs



Visual relational reasoning (Hudson et al. 2019)

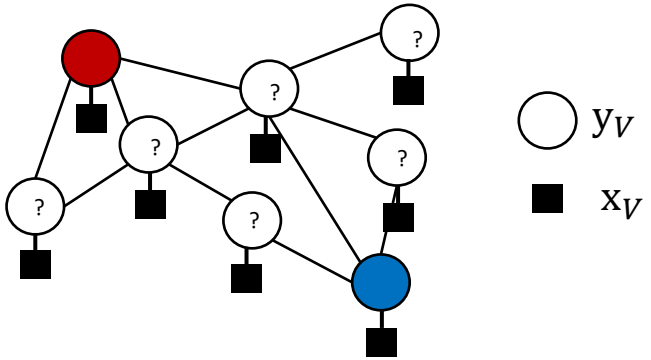


Multi-hop Question answering

(Ding et al. 2019)

Two Types of Learning Paradigms

- **Statistical Relational Learning**
  - Markov Networks (Ross et al. 1980)
  - Conditional Random Fields (Lafferty et al. 2001)
  - Markov Logic Networks (Richardson and Domingos, 2006)
- **Graph Representation Learning**
  - Node Representation Learning (Perozzi et al. 2014, Tang et al. 2015,Grover et al. 2016)
  - Knowledge Graph Embeddings (Bordes et al. 2013, Sun et al. 2019)
  - Graph Neural Networks (Kipf et al. 2016, Veličković et al. 2017)



$$p(\mathbf{y}_V|\mathbf{x}_V) = \frac{1}{Z(\mathbf{x}_V)} \prod_{(i,j)\in E} \psi_{i,j}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{x}_V)$$

Figure 1: Conditional Random Fields

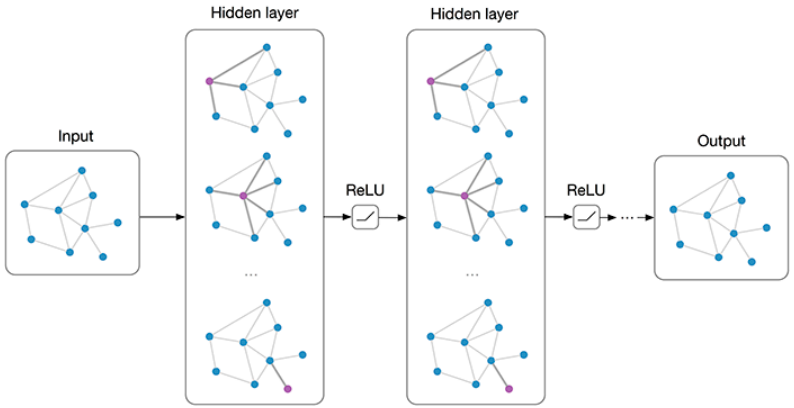
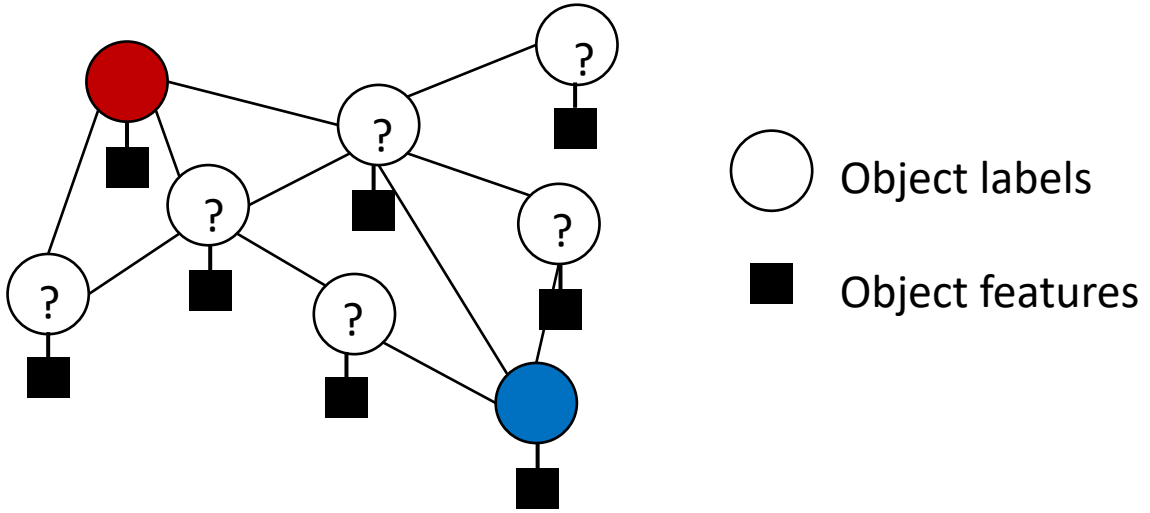


Figure 2: Graph Neural Networks

How to Integrate the Two Learning Paradigms for Relation Reasoning?

Semi-supervised Object Classification

- Given G= (V, E, $\mathbf{x}_V$)
  - $V = V_L \cup V_U$: objects/nodes
  - E : edges
  - $\mathbf{x}_V$: object features

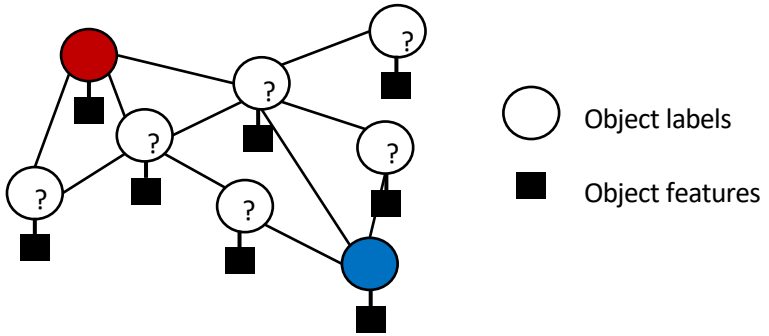


- Give some labeled objects V_L, we want to infer the labels of the rest of objects V_U
- Many other tasks on graphs can be formulated as object classification
  - E.g., link classification

Related Work: Conditional Random Fields

- Models the joint distribution of the object labels given the object features, i.e., $p(\mathbf{y}_V|\mathbf{x}_V)$

$$p(\mathbf{y}_V|\mathbf{x}_V) = \frac{1}{Z(\mathbf{x}_V)} \prod_{(i,j)\in E} \psi_{i,j}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{x}_V)$$



- Pros
  - Capable of modeling the dependency between the object labels
- Cons
  - Some manually defined potential functions
  - Limited model capacity
  - Inference is difficult due to the complicated graph structures

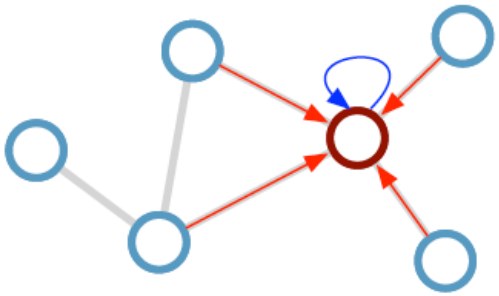
Related Work: Graph Neural Networks

- Learning effective node representations through neural message passing
  - Starting from initial node features $H_0 = X$

$$H_{(l+1)} = f(H_l, A)$$

$$\hat{A} = A + I$$

$$f(H_l, A) = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H_l W_l)$$



Neural Message Passing

- Predicting the node labels independently based on node representations

Related Work: Graph Neural Networks

- Pros
  - Learns effective node representations by feature propagations
  - High model compacity by using multiple non-linear graph convolutional layers
- Cons
  - Ignore the dependency between node labels

GMNN: Graph Markov Neural Networks (Qu, Bengio, and Tang, ICML'19)

- Towards combining statistical relational learning and graph neural networks
- Learning effective node representations for predicting the node labels
- Modeling the label dependencies of nodes
- State-of-the-art performance
  - semi-supervised node classification
  - unsupervised node representation
  - link classification

Meng Qu, Yoshua Bengio, **Jian Tang**. "GMNN: Graph Markov Neural Networks". In ICML'19.

GMNN: Graph Markov Neural Networks

- Model the joint distribution of object labels $\mathbf{y}_V$ conditioned on object attributes $\mathbf{x}_V$, i.e., $\mathrm{p}_\phi(\mathbf{y}_V|\mathbf{x}_V)$
- Learning the model parameters ϕ by maximizing the lower-bound of log-likelihood of the observed data, $\log \mathrm{p}_\phi(\mathbf{y}_L|\mathbf{x}_V)$

$$\log p_\phi(\mathbf{y}_L|\mathbf{x}_V) \geq$$
$$\mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\log p_\phi(\mathbf{y}_L, \mathbf{y}_U|\mathbf{x}_V) - \log q_\theta(\mathbf{y}_U|\mathbf{x}_V)]$$

Optimization with Pseudolikelihood Variational-EM

- E-step: fix p_{ϕ} and update the variational distribution $\mathrm{q}_{\theta}(\mathbf{y}_U|\mathbf{x}_V)$ to approximate the true posterior distribution $\mathrm{p}_{\phi}(\mathbf{y}_U|\mathbf{y}_L, \mathbf{x}_V)$.
- M-step: fix q_{θ} and update p_{ϕ} to maximize the lower bound

$$\ell(\phi) = \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\log p_\phi(\mathbf{y}_L, \mathbf{y}_U|\mathbf{x}_V)]$$

- Directly optimize the joint likelihood is difficult due to the partition function in p_{ϕ}, instead we optimize the pseudolikelihood function

$$\ell_{PL}(\phi) \triangleq \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n \in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{V\setminus n}, \mathbf{x}_V)]$$

$$= \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n \in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x}_V)]$$

Inference/E-step: approximate $\mathrm{p}_{\phi}(\mathbf{y}_U|\mathbf{y}_L, \mathbf{x}_V)$

- Approximate it with variational distribution $\mathrm{q}_{\theta}(\mathbf{y}_U|\mathbf{x}_V)$. Specifically we use mean-field method:

$$q_{\theta}(\mathbf{y}_U|\mathbf{x}_V) = \prod_{n \in U} q_{\theta}(\mathbf{y}_n|\mathbf{x}_V)$$

- We parametrize each variational distribution with a Graph Neural Network

$$q_{\theta}(\mathbf{y}_n|\mathbf{x}_V) = \mathrm{Cat}(\mathbf{y}_n|\mathrm{softmax}(W_{\theta}\mathbf{h}_{\theta,n}))$$

Object representations learned by GNN

Learning/M-step:

- The log-pseudo likelihood:

$$\ell_{PL}(\phi) \triangleq \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n\in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{V\setminus n}, \mathbf{x}_V)]$$

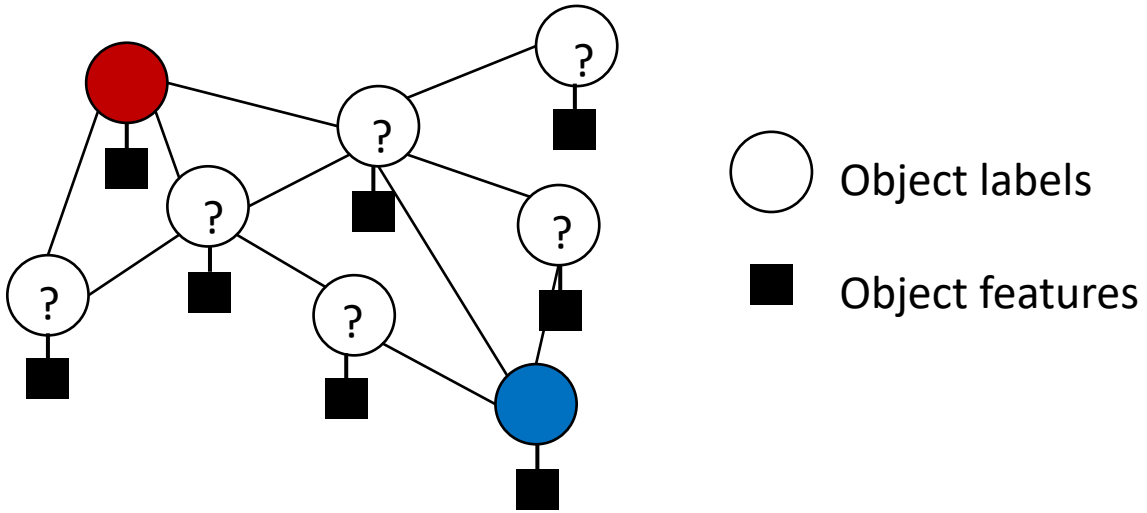
$$= \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n\in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x}_V)]$$

- According to the inference, only the $p_\phi(\mathbf{y_n}|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x_V})$ is required
- Parametrize $p_\phi(\mathbf{y_n}|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x_V})$ with another GCN

$$p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB}(n)}, \mathbf{x}_V) = \mathrm{Cat}(\mathbf{y}_n|\mathrm{softmax}(W_\phi \mathbf{h}_{\phi,n}))$$

Overall Optimization Procedure

- Two Graph Neural Networks Collaborate with each other
  - p_ϕ: learning network, modeling the label dependency
  - q_θ: inference network, learning the object representations
- q_θ infer the labels of unlabeled objects trained with supervision from p_ϕ and labeled objects
- p_ϕ is trained with a fully labeled graph, where the unlabeled objects are labeled by q_θ



Experiments

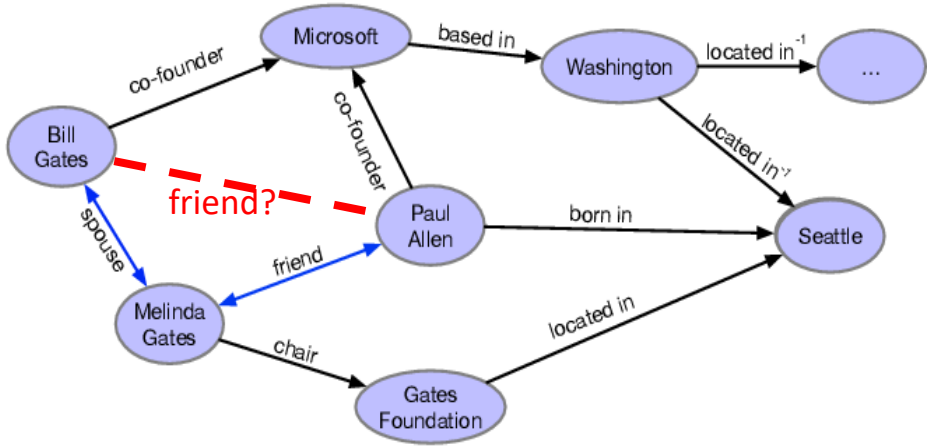
- State-of-the-art performance in node classification

Category	Algorithm	Cora	Citeseer	Pubmed
SSL	LP	74.2	56.3	71.6
SRL	PRM	77.0	63.4	68.3
	RMN	71.3	68.0	70.7
	MLN	74.6	68.0	75.3
GNN	Planetoid *	75.7	64.7	77.2
	GCN *	81.5	70.3	79.0
	GAT *	83.0	72.5	79.0
GMNN	W/o Attr. in p_ϕ	83.4	**73.1**	81.4
	With Attr. in p_ϕ	**83.7**	72.9	**81.8**

- Unsupervised node representation learning
- Link prediction

Reasoning on Knowledge Graphs

- A set of facts $\mathrm{K}G = \{(h, r, t)\}$ represented as triplets
  - (Bill_Gates, Co_Founder, Microsoft)
- A variety of applications
  - Question answering
  - Search
  - Recommender Systems
  - Natural language understanding
  - …



- A fundamental problem: **predicting the missing facts by reasoning with existing facts**

Traditional Symbolic Logic-Rule based approaches

- Expert systems: hard logic rules
  - E.g., $\forall X, Y, \mathrm{Husband}(\mathrm{X}, Y) => \mathrm{Wife}(\mathrm{Y}, \mathrm{X})$
  - ∀ X, Y, Live(X, Y) => Nationality (X, Y)

- Problematic as logic rules can be imperfect or contradictory
- We must handle the uncertainty of logic rules

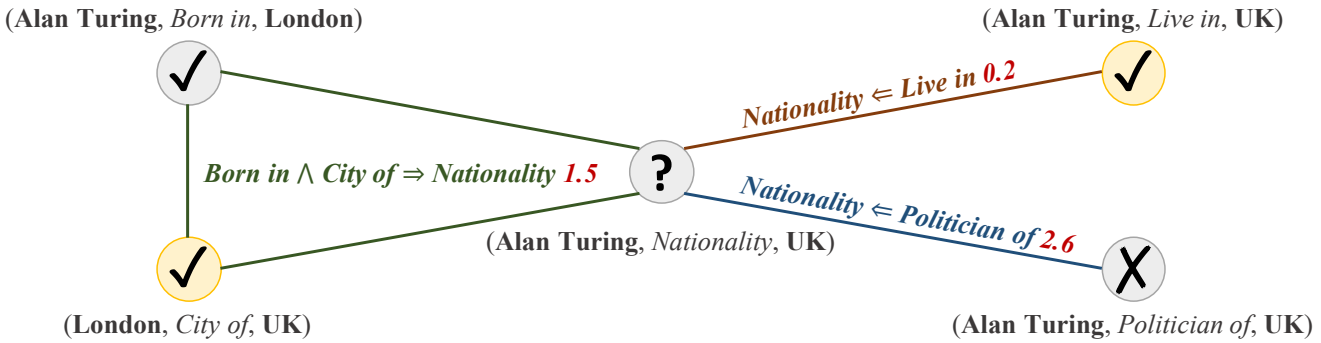
Markov Logic Networks (Richardson and Domingos, 2006)

- Combines first-order logic and probabilistic graphical models

0.2 Live(X, Y) => Nationality (X, Y)

2.6 Politician_of(X, Y) => Nationality (X, Y)

1.5 Born(X,Y)∧City_of (Y,Z) => Nationality(X, Z)



$$p(\mathbf{v}_O, \mathbf{v}_H) = \frac{1}{Z} \exp\left(\sum_{l \in L} w_l \sum_{g \in G_l} \mathbb{1}\{g \text{ is true}\}\right) = \frac{1}{Z} \exp\left(\sum_{l \in L} w_l n_l(\mathbf{v}_O, \mathbf{v}_H)\right)$$

V_O: observed facts

V_H: unobserved/hidden facts

w_l: weight of logic rule l

$n_l(\mathrm{V}_O, \mathrm{V}_H)$: number of true grounds of the logic rule l

Pros and Cons of Markov Logic Networks

- Pros
  - Effectively leverage domain knowledge with logic rules
  - Handle the uncertainty
- Limitation
  - Inference is difficult due to complicated graph structures
  - Recall is low since many facts are not covered by any logic rules

Knowledge Graph Embeddings

- Learning the entity and relation embeddings for predicting the missing facts (e.g., TransE, ComplEx, DisMult, RotatE)
- Defining the joint distribution of all the facts

$$p(\mathbf{v}_O, \mathbf{v}_H) = \prod_{(h,r,t)\in O\cup H} \mathrm{Ber}(\mathbf{v}_{(h,r,t)} | f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t)),$$

An example:

$\mathrm{Ber}(\mathbf{v}_{(h,r,t)} | f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t)) = \sigma(\gamma - ||\mathbf{x}_h + \mathbf{x}_r - \mathbf{x}_t||)$ σ is the sigmoid function, γ is a fixed margin

- Trained by treating V_O as positive facts and V_H as negative facts

Pros and Cons

- Pros
  - Can be effectively and efficiently trained by SGD
  - High recall of missing link prediction with entity and relation embeddings
- Cons
  - Hard to leverage domain knowledge (logic rules)

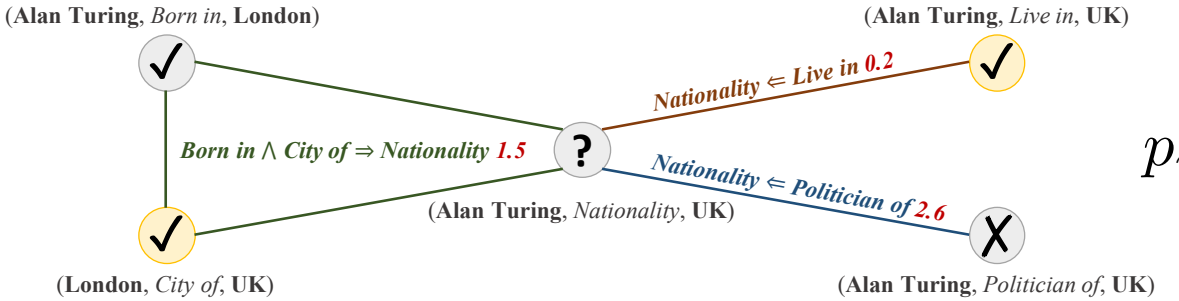
Probabilistic Logic Neural Networks for Reasoning (Qu and Tang, NeurIPS'19.)

- Towards combining Markov Logic Networks and knowledge graph embedding
  - Leverage logic rules and handling their uncertainty
  - Effective and efficient inference
- Define the joint distribution of facts with Markov Logic Network
- Optimization with variational EM
  - Parametrize the variational distribution with knowledge graph embedding methods

Meng Qu and Jian Tang. "Probabilistic Logic Neural Networks for Reasoning." In NeurIPS'2019.

pLogicNet

- Define the joint distribution of facts with an MLN



$$p_w(\mathbf{v}_O, \mathbf{v}_H) = \frac{1}{Z} \exp\left(\sum_l w_l n_l(\mathbf{v}_O, \mathbf{v}_H)\right)$$

- Learning by maximizing the variational lower-bound of the log-likelihood of observed facts

$$\log p_w(\mathbf{v}_O) \geq \mathcal{L}(q_\theta, p_w) = \mathbb{E}_{q_\theta(\mathbf{v}_H)}[\log p_w(\mathbf{v}_O, \mathbf{v}_H) - \log q_\theta(\mathbf{v}_H)]$$

Inference

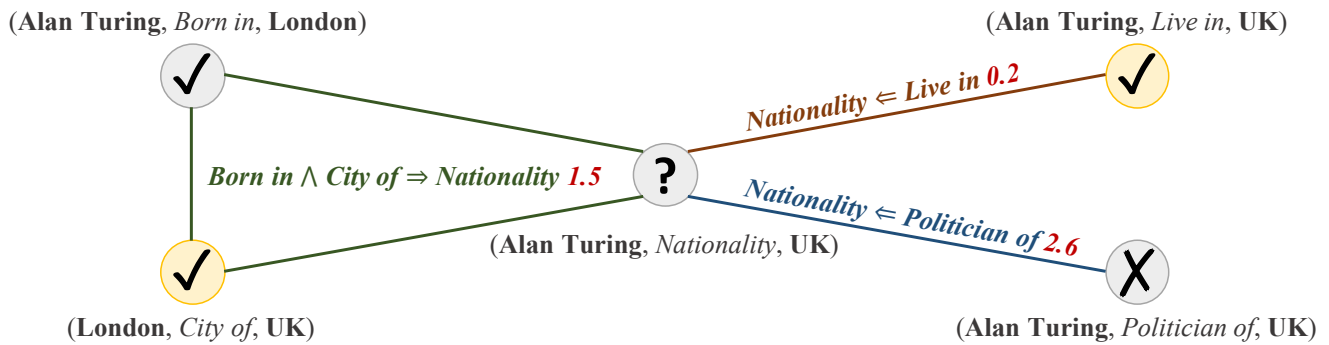
- Amortized mean-field variational inference
  - Use knowledge graph embedding model to parameterize the variational distribution

$$q_\theta(\mathbf{v}_H) = \prod_{(h,r,t)\in H} q_\theta(\mathbf{v}_{(h,r,t)}) = \prod_{(h,r,t)\in H} \mathrm{Ber}(\mathbf{v}_{(h,r,t)}|f(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t)),$$

Learning

- Optimize pseudo-likelihood function
  - Update the weights of logic rules

$$\ell_{PL}(w) \triangleq \mathbb{E}_{q_\theta(\mathbf{v}_H)}[\sum_{h,r,t} \log p_w(\mathbf{v}_{(h,r,t)}|\mathbf{v}_{O\cup H\setminus(h,r,t)})] = \mathbb{E}_{q_\theta(\mathbf{v}_H)}[\sum_{h,r,t} \log p_w(\mathbf{v}_{(h,r,t)}|\mathbf{v}_{\mathrm{MB}(h,r,t)})],$$



Performance of Link Prediction

- **Datasets:** benchmark knowledge graphs
  - FB15K, WN18, FB15K-237, WN18-RR
- Logic rules:
  - Composition rules (e.g., Father of Father is GrandFather)
  - Inverse rules (e.g., Husband and Wife)
  - Symmetric rules (e.g., Similar)
  - Subrelation rules (e.g., Man => Person)

Category	Algorithm	FB15k					WN18				
		MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
KGE	TransE [3]	40	0.730	64.5	79.3	86.4	272	0.772	70.1	80.8	92.0
	DistMult [17]	42	0.798	-	-	89.3	655	0.797	-	-	94.6
	HolE [26]	-	0.524	40.2	61.3	73.9	-	0.938	93.0	94.5	94.9
	ComplEx [41]	-	0.692	59.9	75.9	84.0	-	0.941	93.6	94.5	94.7
	ConvE [8]	51	0.657	55.8	72.3	83.1	374	0.943	93.5	94.6	95.6
Rule-based	BLP [7]	415	0.242	15.1	26.9	42.4	736	0.643	53.7	71.7	83.0
	MLN [32]	352	0.321	21.0	37.0	55.0	717	0.657	55.4	73.1	83.9
Hybrid	RUGE [15]	-	0.768	70.3	81.5	86.5	-	-	-	-	-
	NNE-AER [9]	-	0.803	76.1	83.1	87.4	-	0.943	**94.0**	94.5	94.8
Ours	pLogicNet	**33**	0.792	71.4	85.7	90.1	255	0.832	71.6	94.4	95.7
	pLogicNet*	**33**	**0.844**	**81.2**	**86.2**	**90.2**	**254**	**0.945**	93.9	**94.7**	**95.8**

ICML 2020 Workshop

- **Bridge Between Perception and Reasoning: Graph Neural Networks & Beyond**
  - Website: https://logicalreasoninggnn.github.io/
  - Deadline: 31st, May
- **Organizers:**
  - Jian Tang, Le Song, Jure Leskovec, Renjie Liao, Yujia Li, Sanja Fidler, Richard Zemel, Ruslan Salakhutdinov

Invited Speakers:



YOSHUA BENGIO
UNIVERSITY OF MONTREAL & MILA



PETER BATTAGLIA
DEEPMIND



ZICO KOLTER
CMU



LESLIE KAELBLING
MIT



TOMMI JAAKKOLA
MIT



LUC DE RAEDT
KU LEUVEN



KRISTIAN KERSTING
TU DARMSTADT

Graph Representation Learning for Drug Discovery

The Process of Drug Discovery

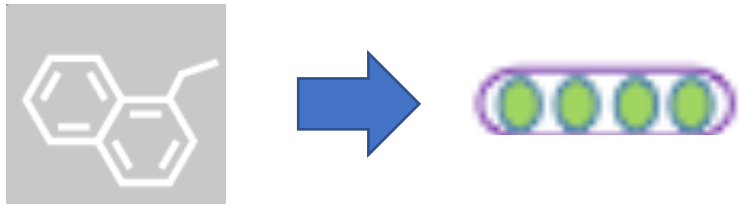
- A very long and costly process
  - On average takes more than 10 years and $2.5B to get a drug approved
- Big opportunities for AI to accelerate this process



Screen millions of functional molecules; Found by serendipity: Penicillin	Modify the molecule to improve specific properties. e.g. toxicity, SA	In-vitro and in-vivo experiments; synthesis	Multiple Phases

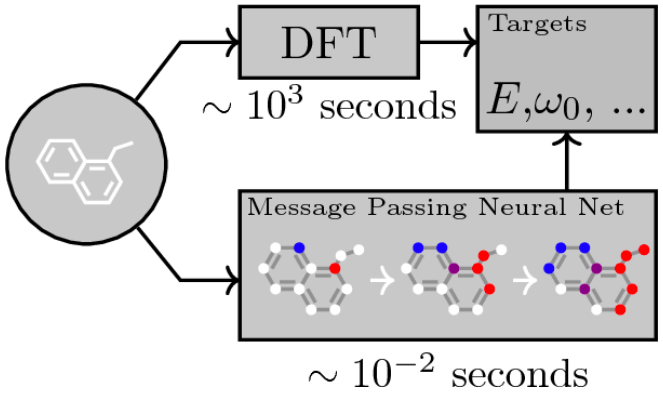
Molecule Properties Prediction

- Predicting the properties of molecules is very important in many stages of drug discovery
  - Virtual screening
- Represent the whole molecule (graph) as a feature vector



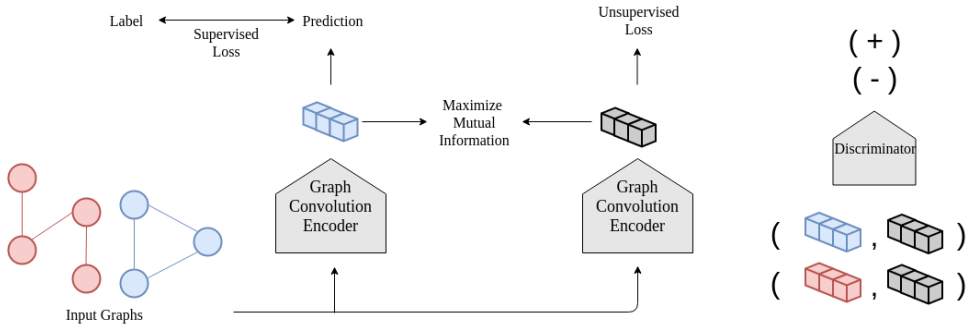
Unsupervised and Semi-supervised Learning for Molecular Graph Representation (Sun et al. ICLR 20)

- Most existing work on molecular representation are based on supervised learning with graph neural networks
  - Require a large number of labeled data
- However, the number of labeled data is very limited
- Leverage the unlabeled data!!



Supervised Methods (Gilmer et al. 17)

Gilmer et al. Neural Message Passing for Quantum Chemistry. ICML'17.



Unsupervised and semi-supervised methods (Sun et al. 19)

Sun et al. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. ICLR'20

Application: Finding Effective Antibiotics for Secondary Infections in COVID-19 (Ongoing Research)

- A high proportion of non-surviving patients of COVID-19 developed a secondary infection (Zhou et al. 2020)
- Finding effective antibiotics
- Predict antibacterial properties
  - Collaboration with MIT
  - https://www.aicures.mit.edu/tasks

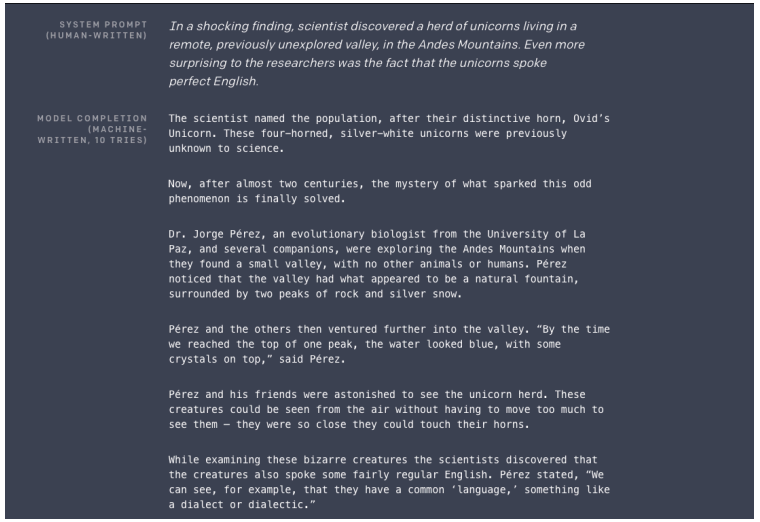
Rank	Model	Author	10-fold CV ROC-AUC	10-fold CV PRC-AUC	Test ROC-AUC	Test PRC-AUC
1	Pre-trained OGB-GIN (ensemble)	Weihua Hu@Stanford	0.905 +/- 0.133	0.494 +/- 0.333	0.837	0.651
2	Chemprop ++	AICures@MIT	0.810 +/- 0.160	0.423 +/- 0.332	0.891	0.641
3	Graph Self-supervised Learning	SJTU_NRC_Mila	0.825 +/- 0.210	0.530 +/- 0.342	0.800	0.622
4	Pre-trained OGB-GIN	Weihua Hu@Stanford	0.907 +/- 0.100	0.464 +/- 0.328	0.870	0.616
5		zhenxingwu	0.813 +/- 0.165	0.465 +/- 0.332	0.73	0.594
6	MLP	IITM			0.777	0.59
7		Gianluca Bontempi			0.848	0.584
8	Chemprop	IITKGP	0.820	0.602	0.818	0.58
9		Gianluca Bontempi			0.858	0.578
10		Gianluca Bontempi			0.789	0.569
11	Random forest	Gianluca Bontempi	0.852		0.832	0.561
12		Aymen Waheb			0.721	0.556
13		zhenxingwu			0.705	0.539
14		Apoorv Umang			0.84	0.513
15	XGBOOST	Andrea Loreggia	0.705 +/- 0.189		0.714	0.301
16	Chemprop	AICures@MIT	0.725 +/- 0.207	0.310 +/- 0.271	0.716	0.183

De Novo Molecule Design and Optimization

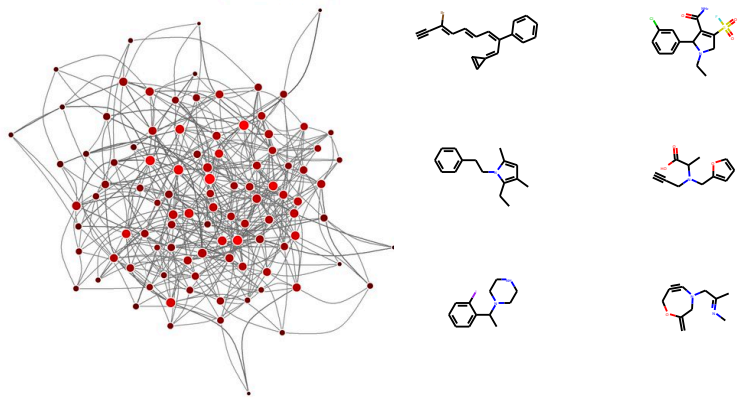
- Deep generative models for data generation



Image generation
(by StyleGAN, From Internet)



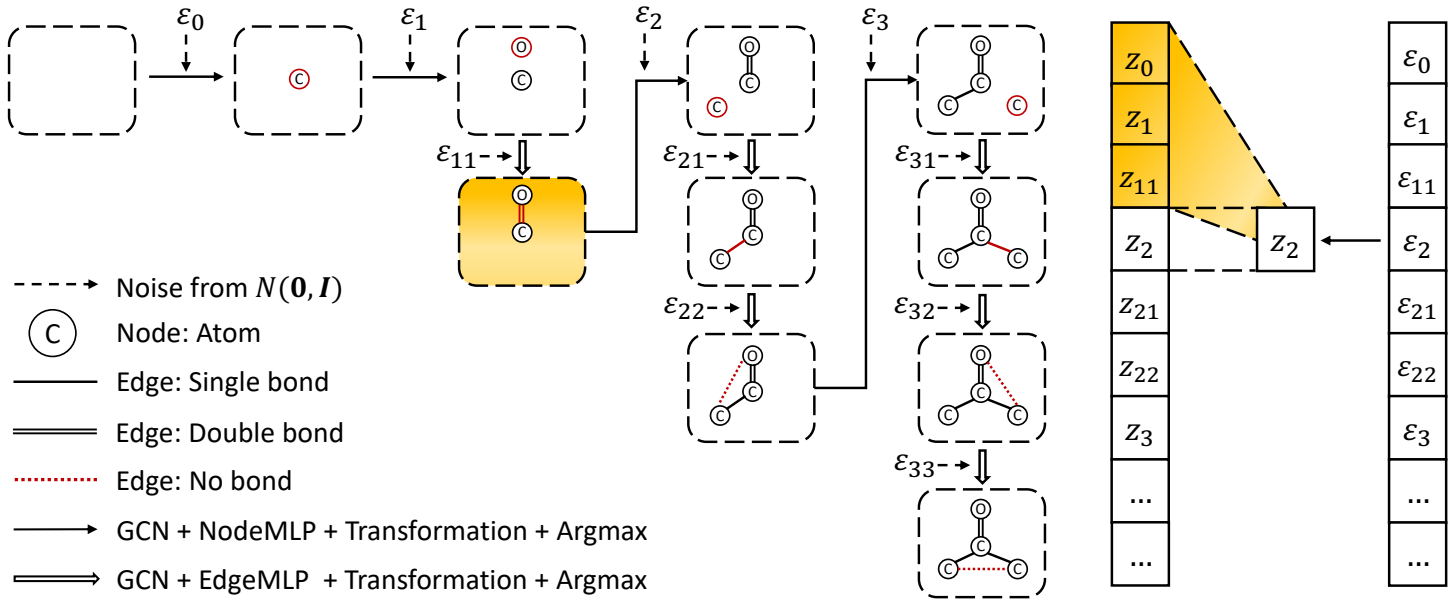
Text generated by by GPT-2,
Examples from Internet



Graphs?

GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation (Shi & Xu et al. ICLR'20)

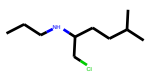
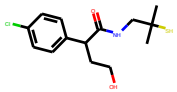
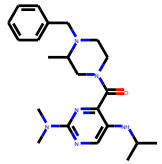
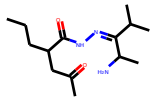
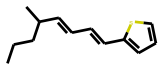
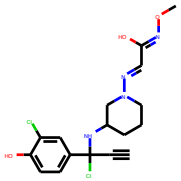
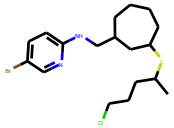
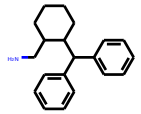
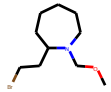
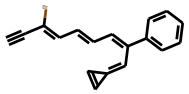
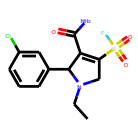
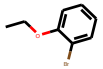
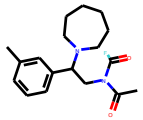
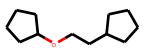
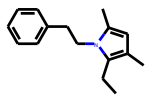
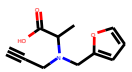
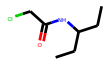
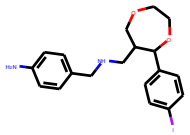
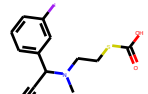
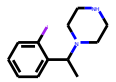
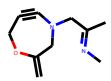
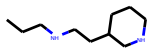
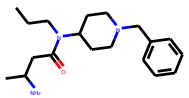
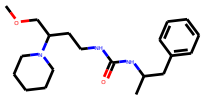
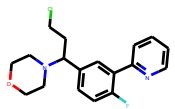
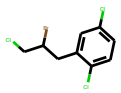
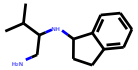
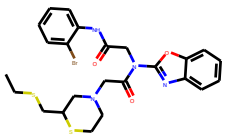
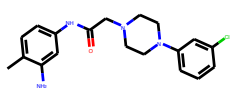
- Formulate graph generation as a sequential decision process
  - In each step, generate a new atom
  - Determine the bonds between the new atoms and existing atoms



(a) Sampling Framework (b) Autoregressive Flow

Molecule Generation

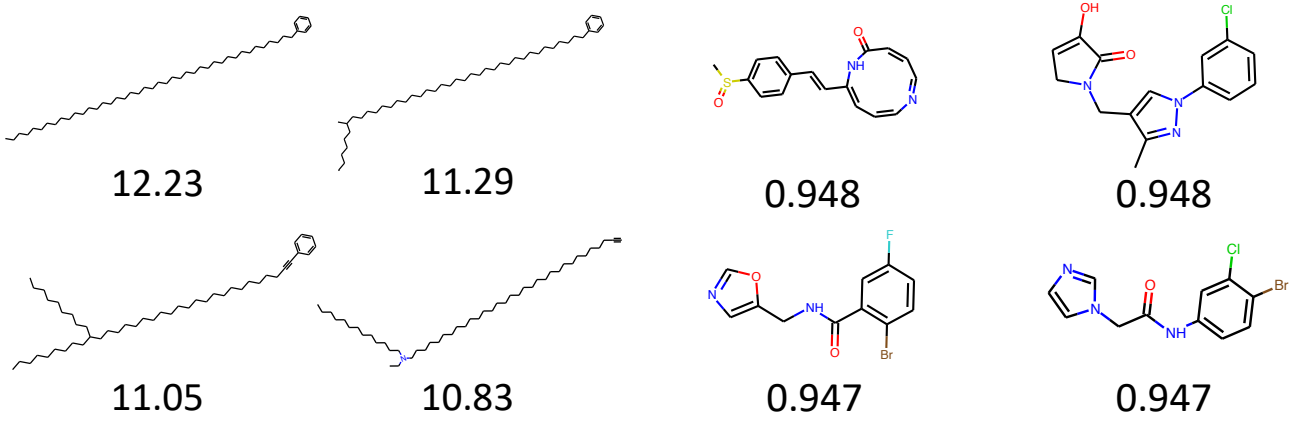
Method	Validity	Validity w/o check	Uniqueness	Novelty	Reconstruction
JT-VAE	100%	—	100%‡	100%‡	76.7%
GCPN	100%	20%†	99.97%‡	100%‡	—
MRNN	100%	65%	99.89%	100%	—
GraphNVP	42.60%	—	94.80%	100%	100%
GraphAF	100%	68%	99.10%	100%	100%

Molecule Optimization

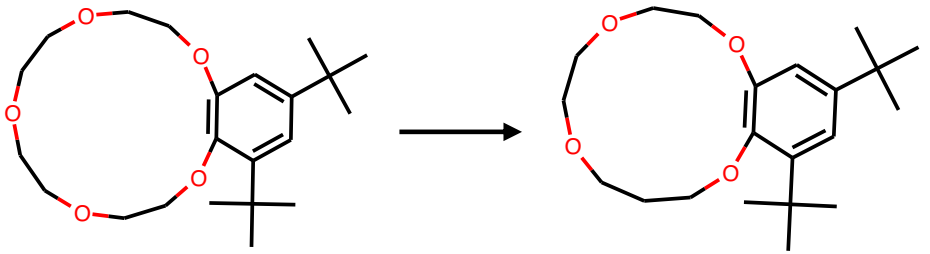
- Properties
  - Penalized logP
  - QED (druglikeness)

Method	Penalized logP				QED			
	1st	2nd	3rd	Validity	1st	2nd	3rd	Validity
ZINC (Dataset)	4.52	4.30	4.23	100.0%	0.948	0.948	0.948	100.0%
JT-VAE (Jin et al., 2018)	5.30	4.93	4.49	100.0%	0.925	0.911	0.910	100.0%
GCPN (You et al., 2018a)	7.98	7.85	7.80	100.0%	**0.948**	0.947	0.946	100.0%
MRNN[1] (Popova et al., 2019)	8.63	6.08	4.73	100.0%	0.844	0.796	0.736	100.0%
GraphAF	**12.23**	**11.29**	**11.05**	100.0%	**0.948**	**0.948**	**0.947**	100.0%

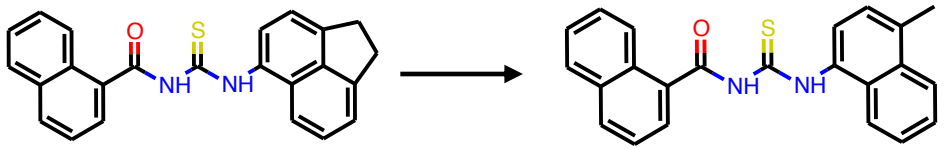


(a) Penalized logP optimization (b) QED optimization

Constrained Optimization



-30.21 -22.87

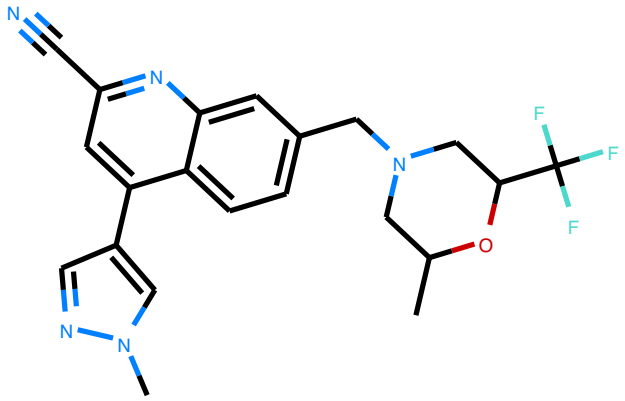


-14.32 3.58

(c) Constrained optimization

Retrosynthesis Prediction

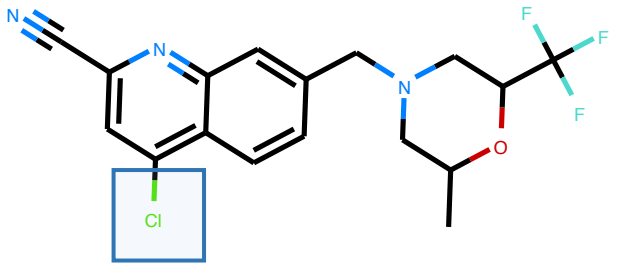
- Once a molecular structure is designed, how to synthesize it?
- Retrosynthesis planning/prediction
  - Identify a set of reactants to synthesize a target molecule



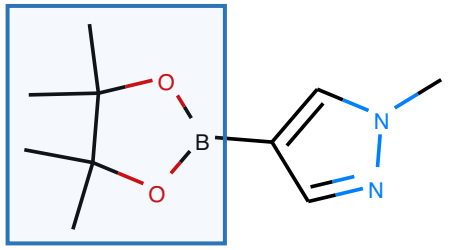
Product (Given)

Predict Reactants

Reaction Type
(optional)



Reactant A



Reactant B

…

…

A Graph to Graphs Framework for Retrosynthesis Prediction (Shi et al. 2020)

- Each molecule is represented as a molecular graph
- Formulate the problem as a graph (product molecule) to a set of graphs (reactants)
- The whole framework are divided into two stages
  - Reaction center identification
  - Graph Translation

Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang and Jian Tang. **A Graph to Graphs Framework for Retrosynthesis Prediction**. In Submission, 2020.

Experiments

- Experiment Setup
  - Benchmark data set USPTO-50K, containing 50k atom-mapped reactions
  - Evaluation metrics: top-*k* exact match (based on canonical SMILES) accuracy

Table 1. Top-*k* exact match accuracy when reaction class is given. Results of all baselines are directly taken from (Dai et al., 2019).

Methods	Top-*k* accuracy %			
	1	3	5	10
Template-free				
Seq2seq	37.4	52.4	57.0	61.7
G2Gs	**61.0**	**81.3**	**86.0**	**88.7**
Template-based				
Retrosim	52.9	73.8	81.2	88.1
Neuralsym	55.3	76.0	81.4	85.1
GLN	**64.2**	**79.1**	**85.2**	**90.0**

Table 2. Top-*k* exact match accuracy when reaction class is unknown. Results of all baselines are taken from (Dai et al., 2019).

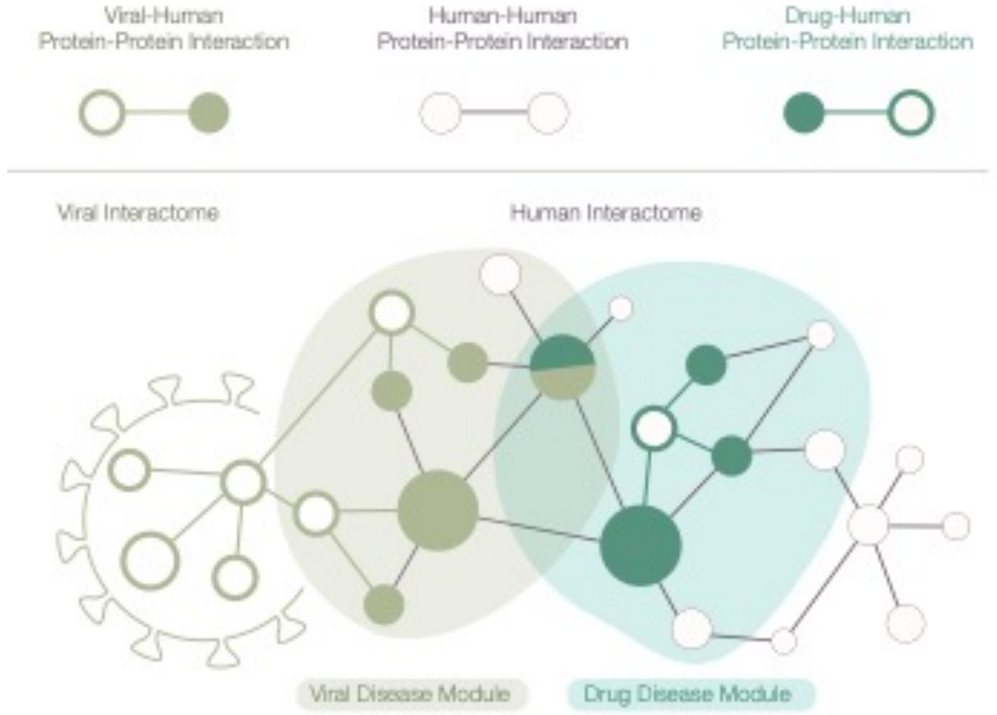
Methods	Top-*k* accuracy %			
	1	3	5	10
Template-free				
Transformer	37.9	57.3	62.7	/
G2Gs	**48.9**	**67.6**	**72.5**	**75.5**
Template-based				
Retrosim	37.3	54.7	63.3	74.1
Neuralsym	44.4	65.3	72.4	78.9
GLN	**52.5**	**69.0**	**75.6**	**83.7**

We're Hiring!!

- Multiple positions in my group
  - Postdocs, PhDs, MSc
- Topics
  - Graph Representation Learning, Graph Neural Networks, Knowledge Graphs
  - Symbolic Logical Reasoning
  - Deep generative Models
  - Drug Discovery, bioinformatics
- Send an email to tangjianpku@gmail.com or jian.tang@hec.ca

Drug Repurposing with Medical Knowledge Graph for COVID-19

- Learning the representations of diseases, proteins, and drugs



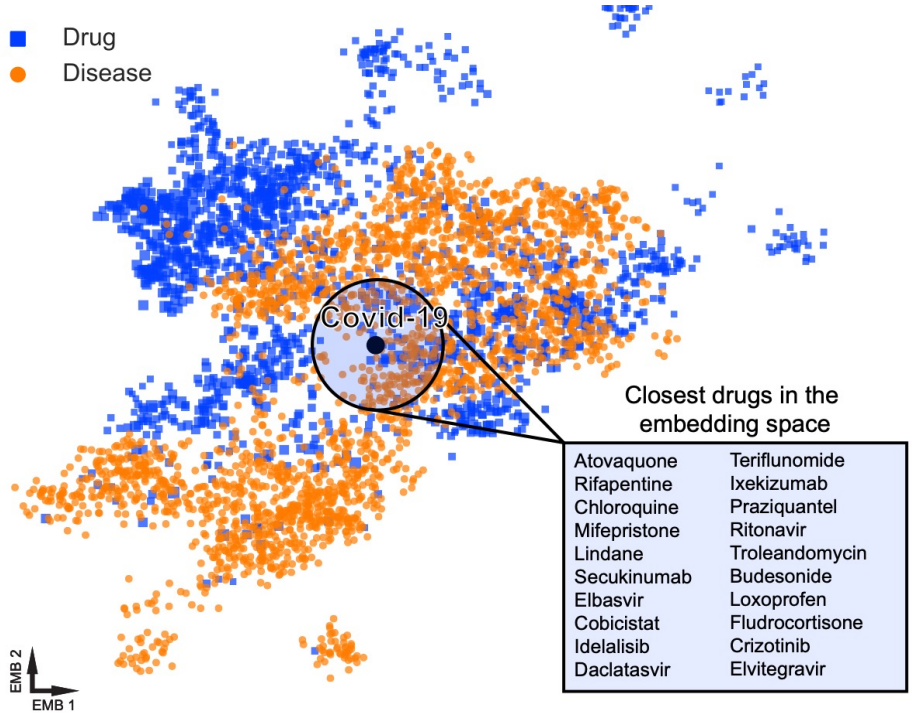


Figure from Gysi et al. 2020

Take Away

- Relational reasoning is important to a variety of applications
  - Node classification, link prediction on knowledge graphs, question answering
  - From System I reasoning to System II Reasoning
- Towards combining two learning frameworks
  - Statistical Relational Learning
  - Graph Representation Learning
- Looking Forward:
  - How to integrate System I and II reasoning?
  - How to automatically learn logical rules?

Thanks!


NSERC
CRSNG


CIFAR


amazon


Microsoft®
Research


IVADO
INSTITUT DE VALORISATION DES DONNÉES


Canada
NRC·CNRC