# Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction

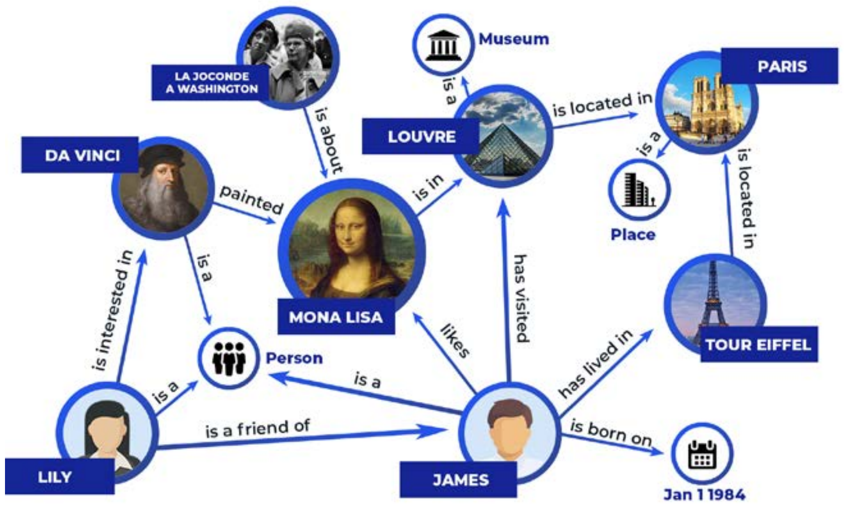
Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, Jian Tang

2022/1/4

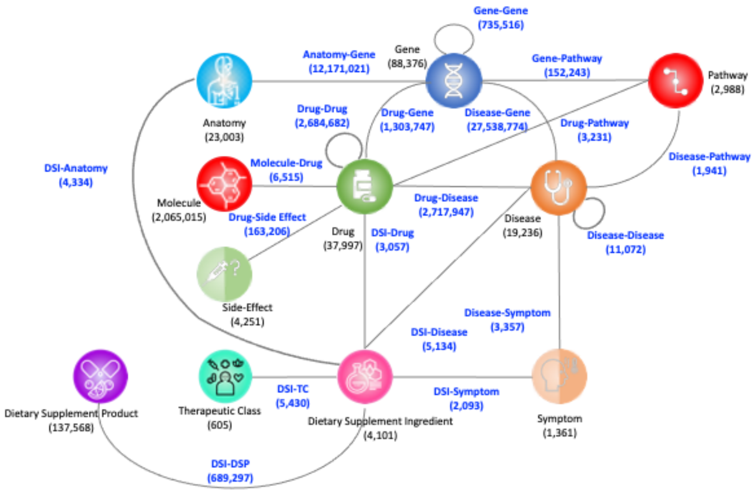
# Link Prediction

- Predict the interactions between nodes

social networks



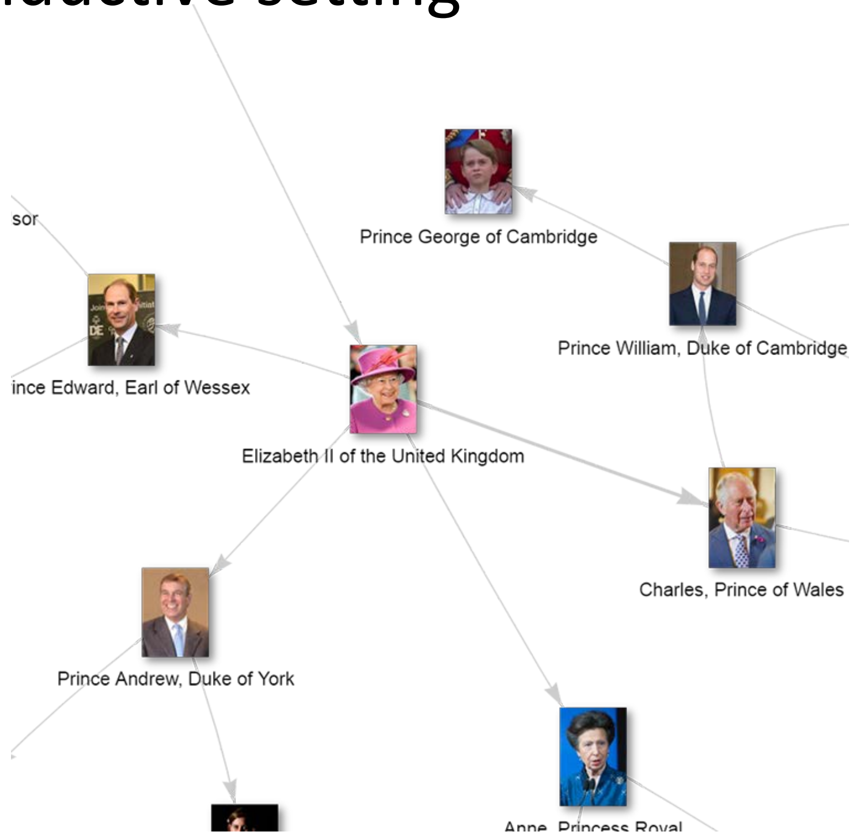

knowledge graph completion



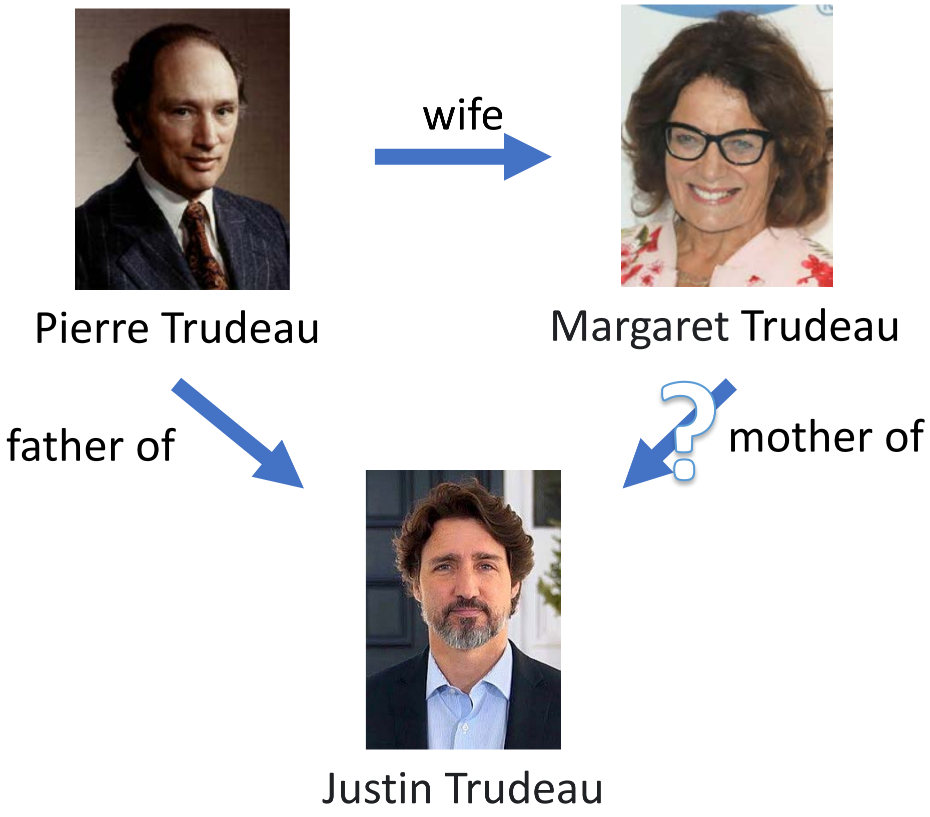

drug repurposing

# Challenges

- Inductive setting




Train



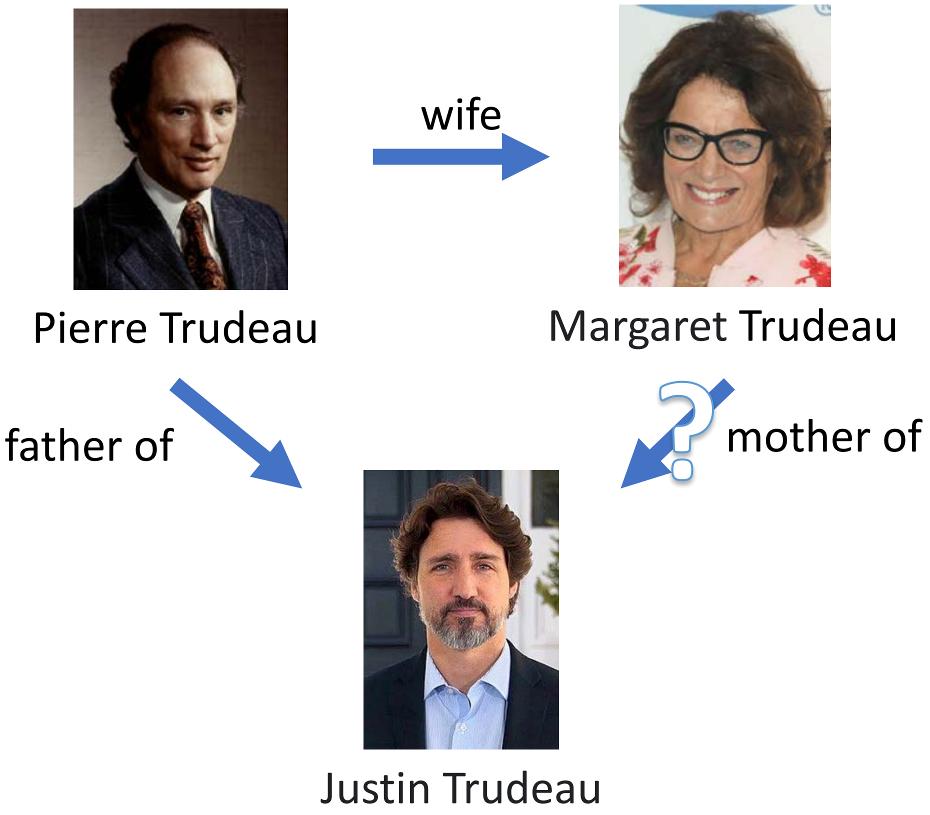

Test (a new graph)

# Challenges

- Interpretability

- **Query:** Who is Justin Trudeau's mother?
- **Answer:** Margaret Trudeau

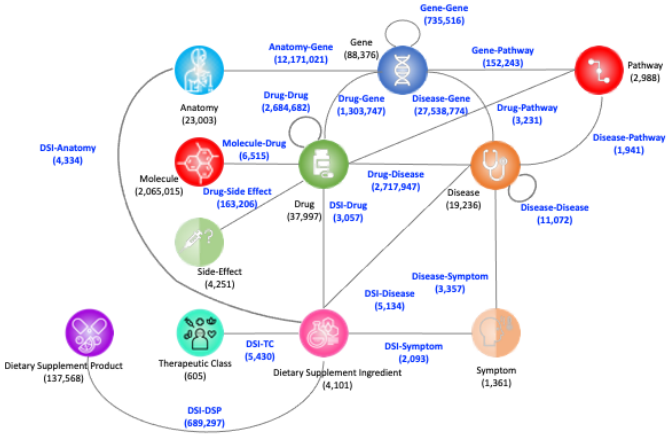
- Why?



# Challenges

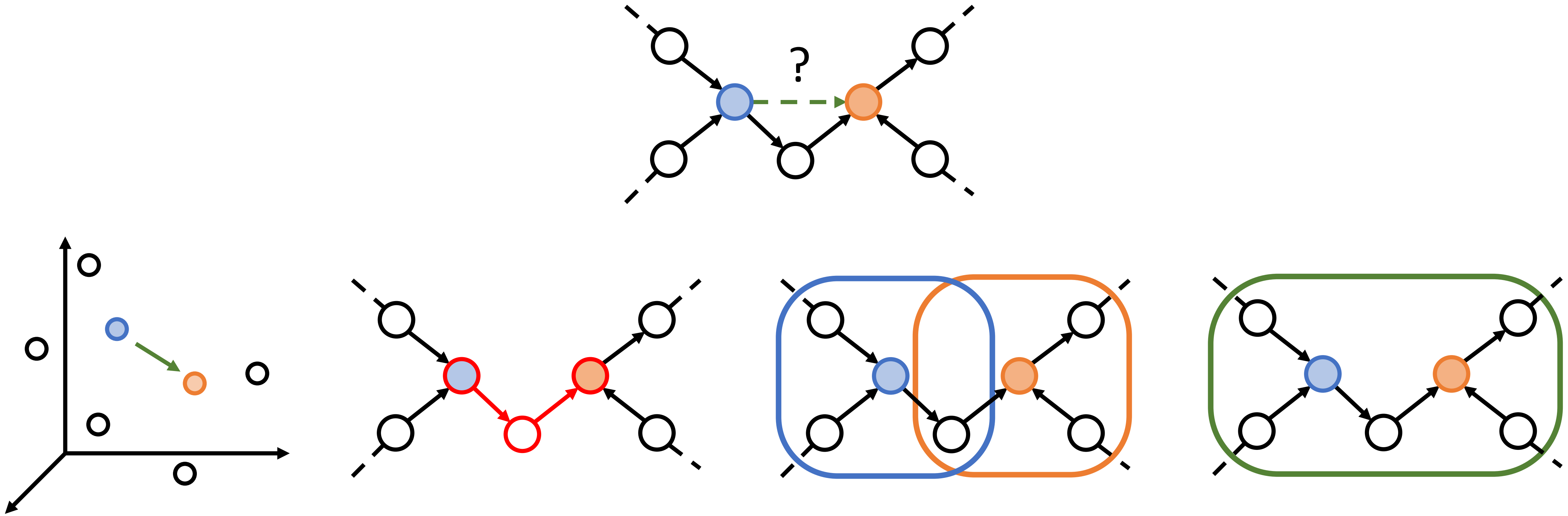
- Large scale

Wikidata
87M entities
504M triplets




CBKH
2.4M nodes
48M edges

# Machine Learning Methods




Embeddings

TransE, DistMult, ComplEx, etc.

Path-based

PathRanking, PathRNN, DeepPath, etc.

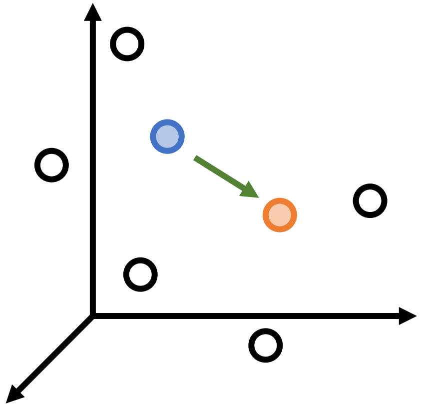
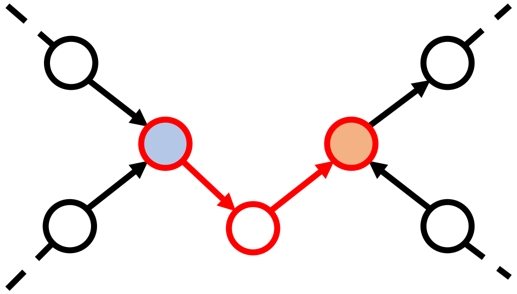
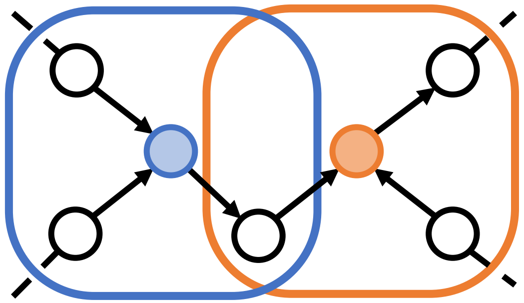
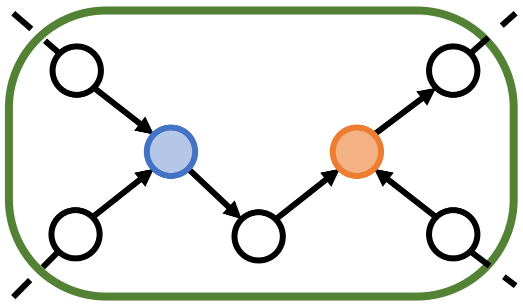
Node GNN encoders

GCN, RGCN, CompGCN, etc.

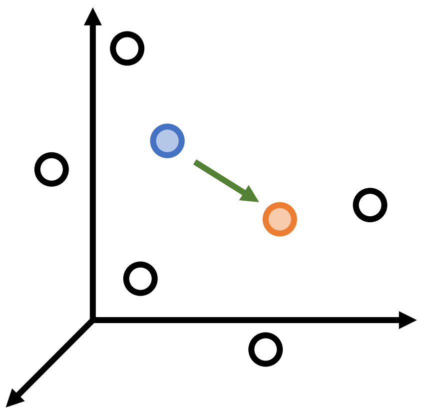
Subgraph GNN encoders

SEAL, GraIL

# Machine Learning Methods

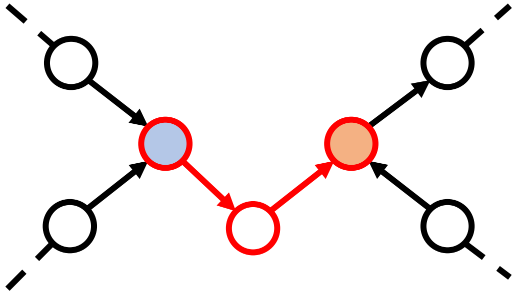
| |  |  |  |  |
|---|---|---|---|---|
| | Embeddings | Path-based | Node GNN encoders | Subgraph GNN encoders |
|  | strong performance<br>scalable | interpretability<br>inductive | good performance<br>flexibility<br>scalable<br>(pseudo) inductive | strong performance<br>flexibility<br>inductive |
|  | no interpretability<br>transductive only | weak performance<br>sometimes not scalable | no interpretability | no interpretability<br>not scalable |

# Machine Learning Methods



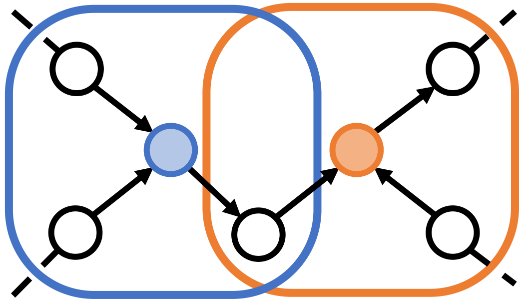
Embeddings

strong performance
scalable



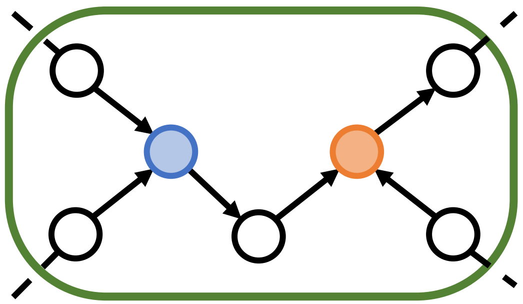
Path-based

interpretability
inductive



Node GNN encoders

good performance
flexibility
scalable
(pseudo) inductive



Subgraph GNN encoders

strong performance
flexibility
inductive

Can we achieve the best of all worlds?

# Traditional Methods

- Traditional methods
  - Katz index: weighted count of paths between two nodes
  - Personalized PageRank: random walk probability from one to the other
  - Graph distance: the shortest path between two nodes

- Lessons
  - Interpretability via paths
  - Inductive
  - Scalability via dynamic programming

# Traditional Methods

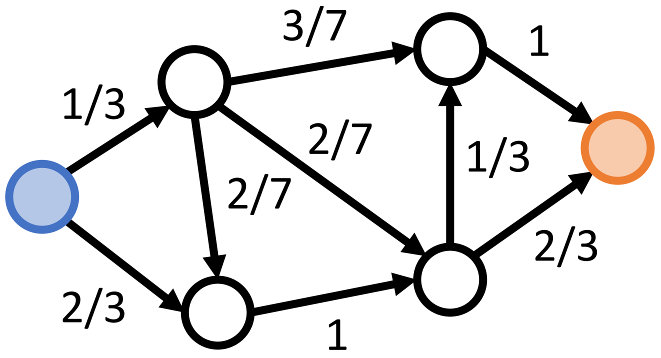
- Traditional methods
  - Katz index: weighted count of paths between two nodes
  - Personalized PageRank: random walk probability from one to the other
  - Graph distance: the shortest path between two nodes

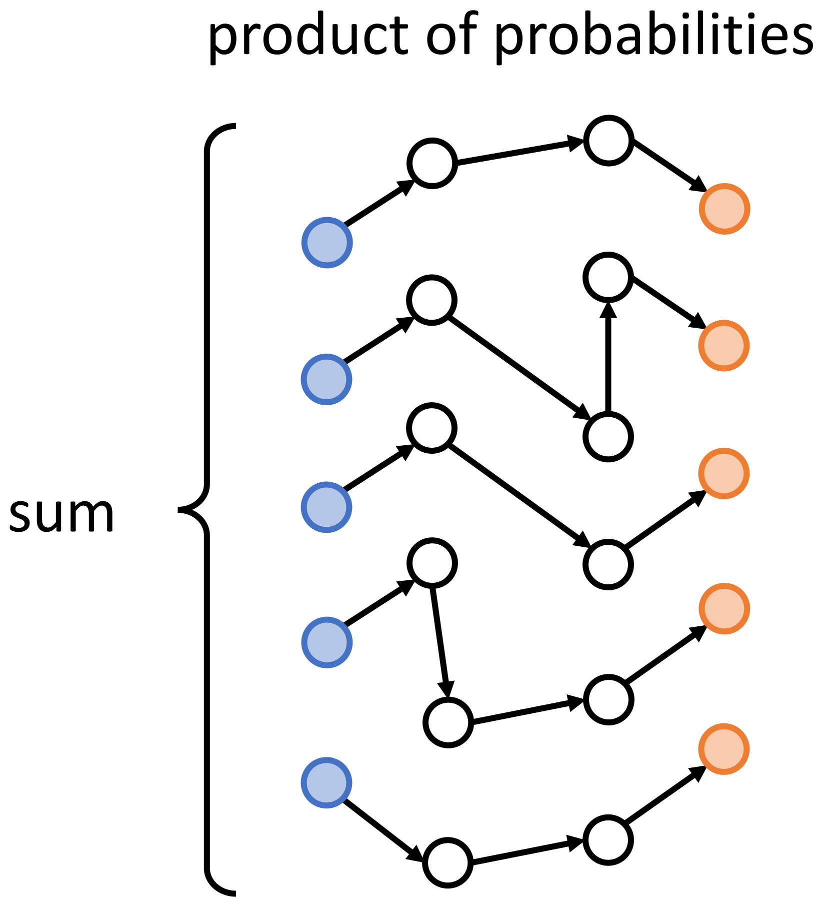
- Lessons
  - Interpretability via paths
  - Inductive
  - Scalability via dynamic programming

Can we parameterize them with NNs?

# Traditional Methods

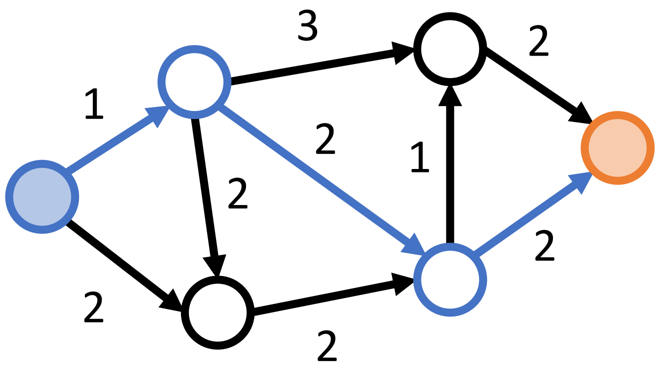
- Personalized PageRank

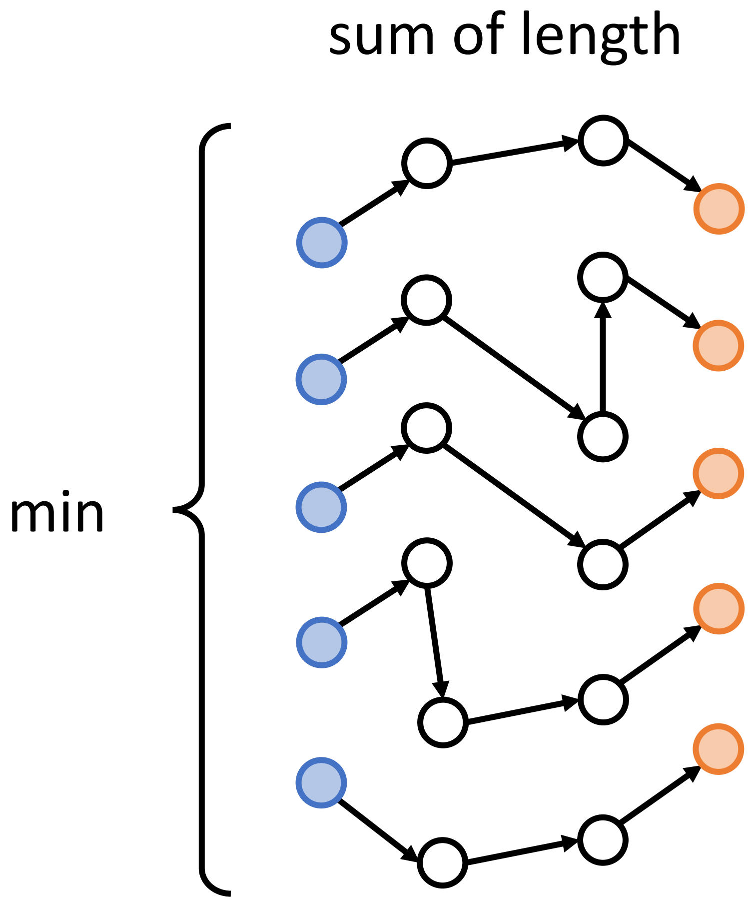


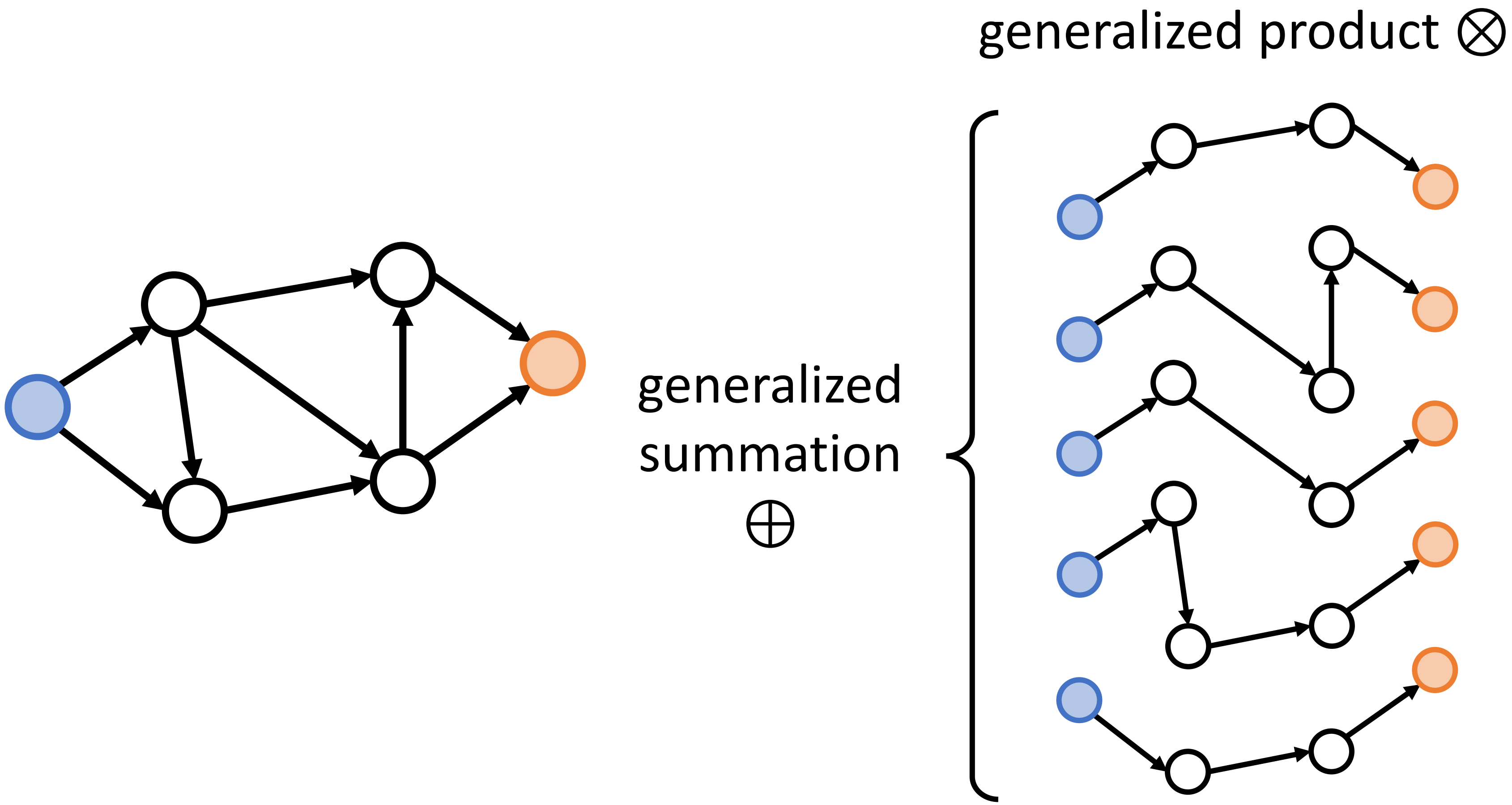
# Traditional Methods

- Graph distance

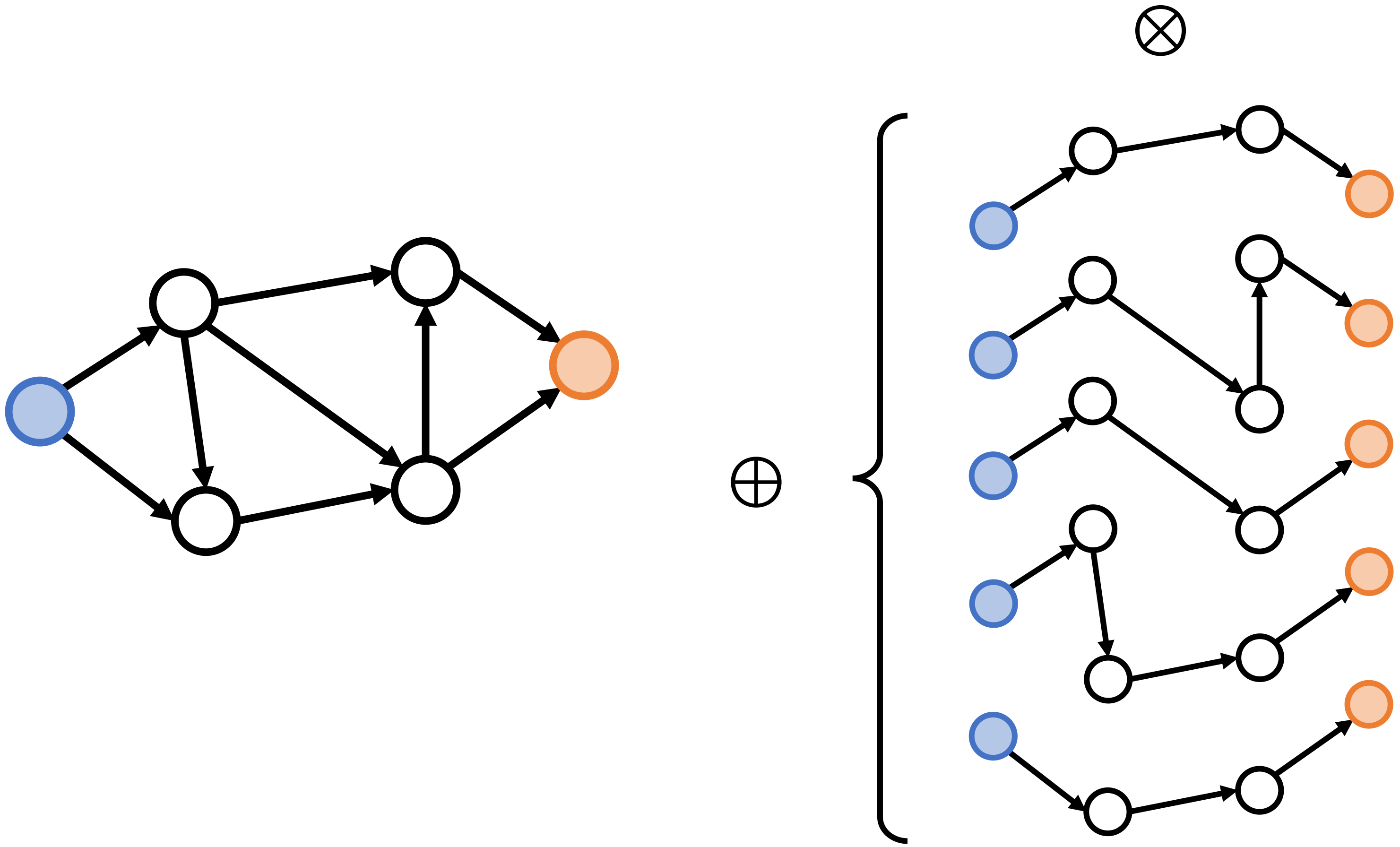


# Path Formulation



# Path Formulation




Katz index

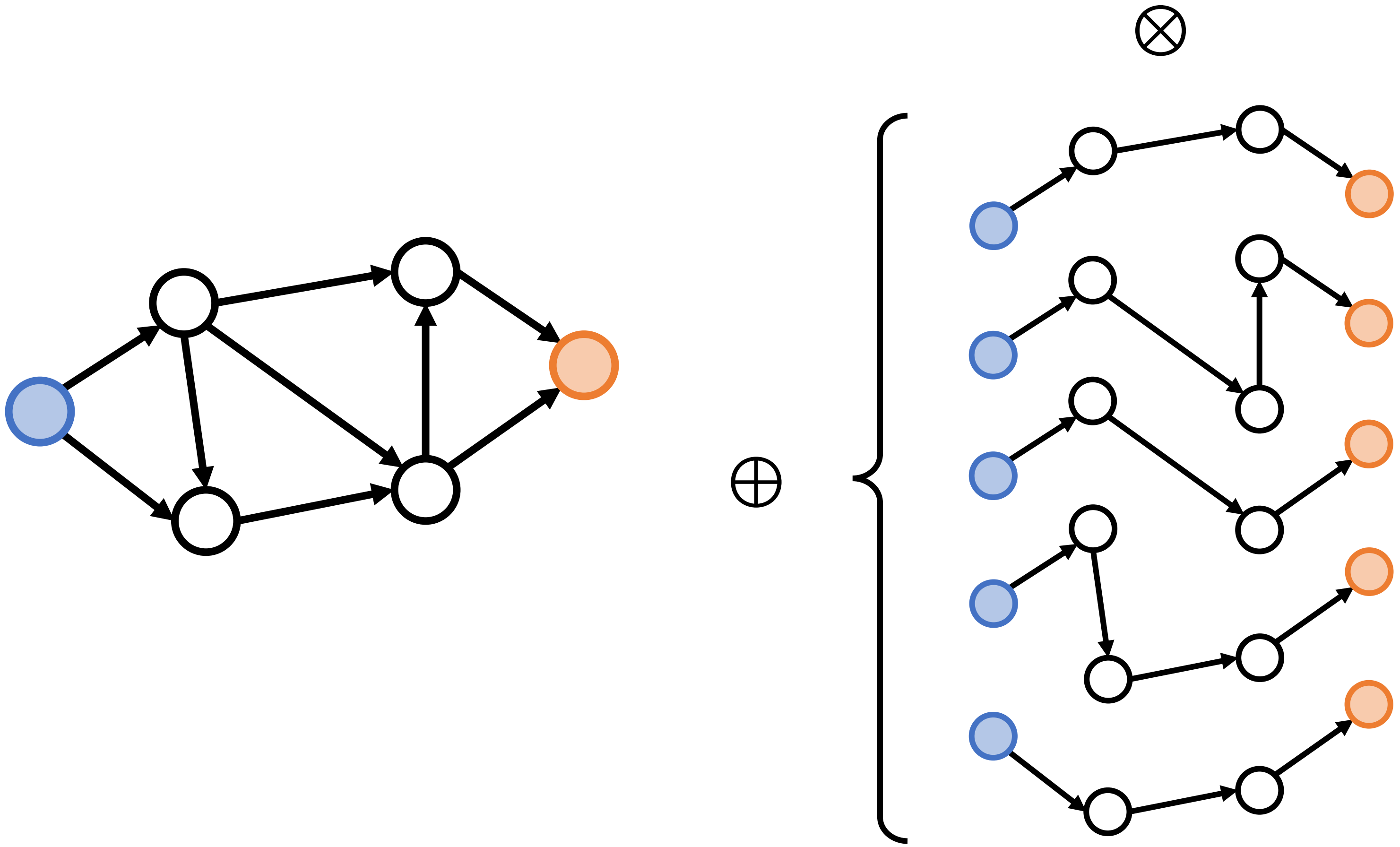
Personalized PageRank

Graph distance

Widest path

Most reliable path

# Path Formulation



inductive

interpretable

not scalable

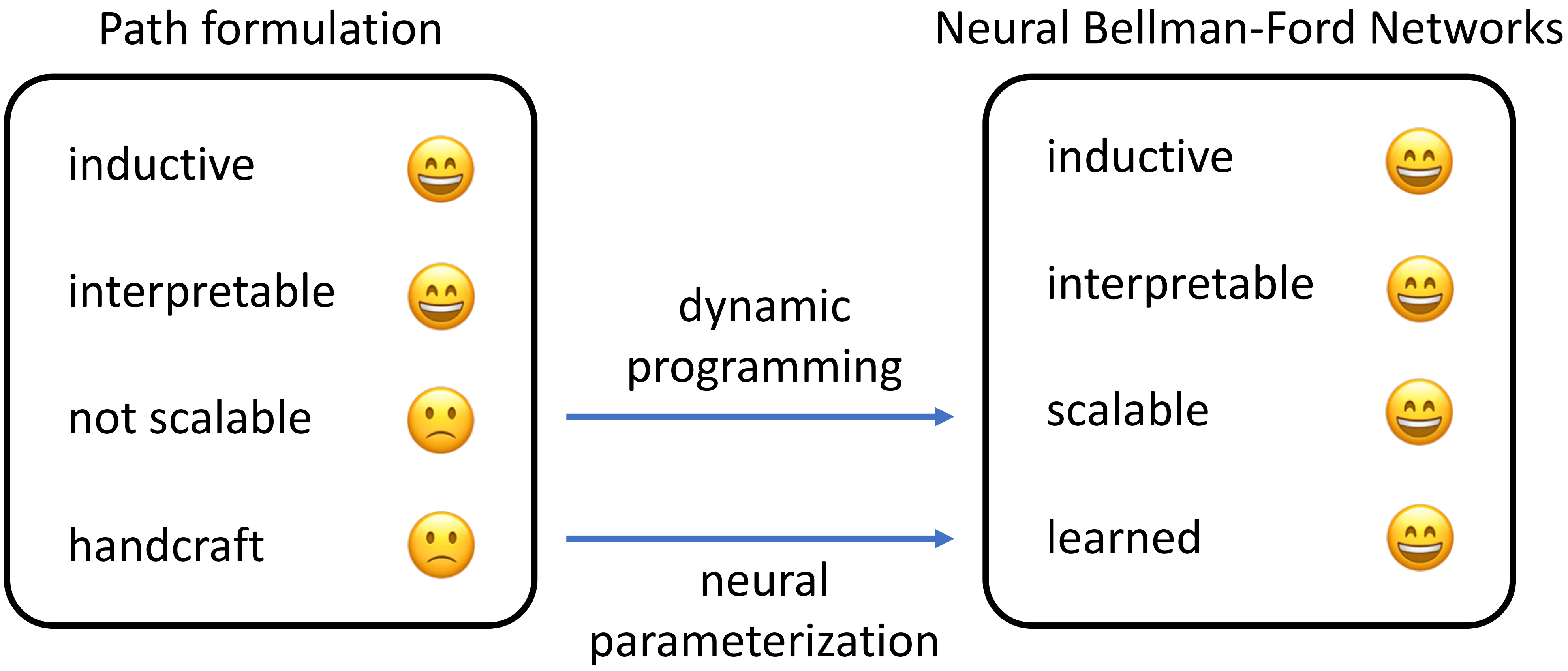
handcraft

# Make It Better

Path formulation

| | |
|---|---|
| inductive | 😄 |
| interpretable | 😄 |
| not scalable | 🙁 |
| handcraft | 🙁 |

# Make It Better

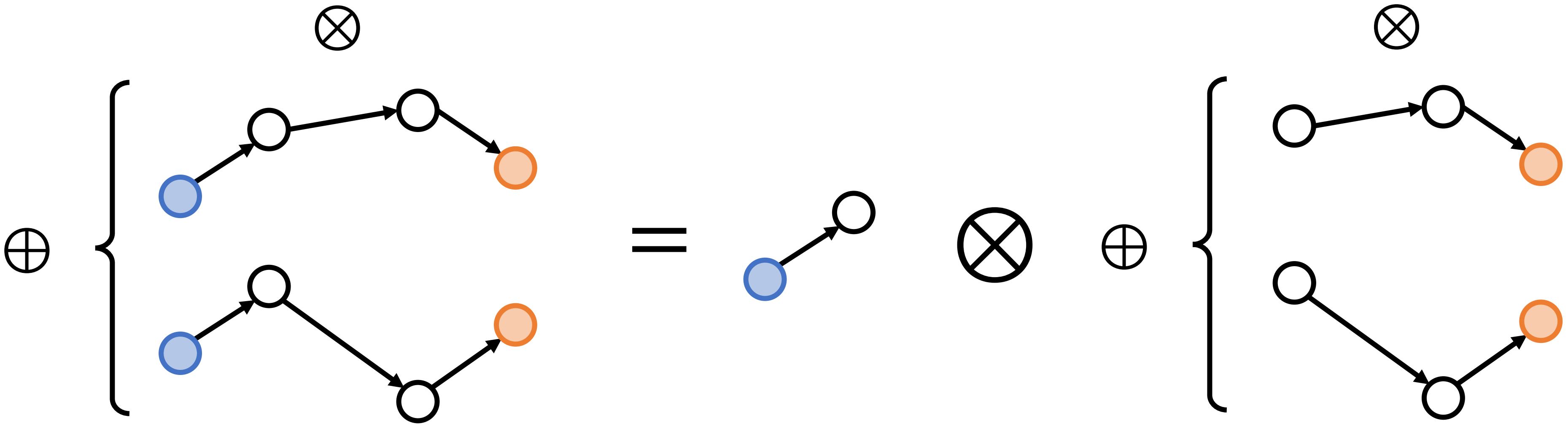


# Dynamic Programming

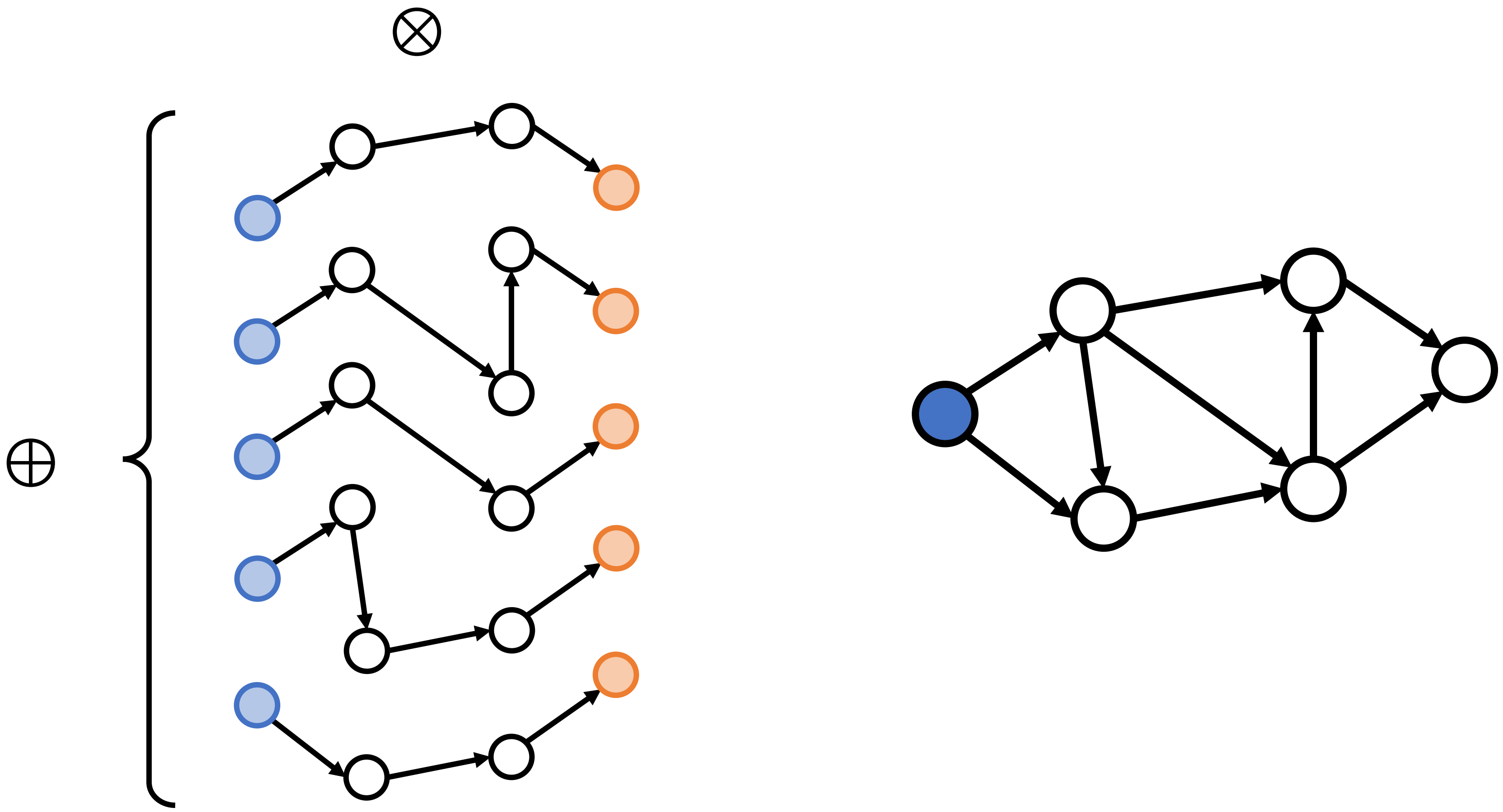
- $ab + ac = a(b + c)$

# Dynamic Programming

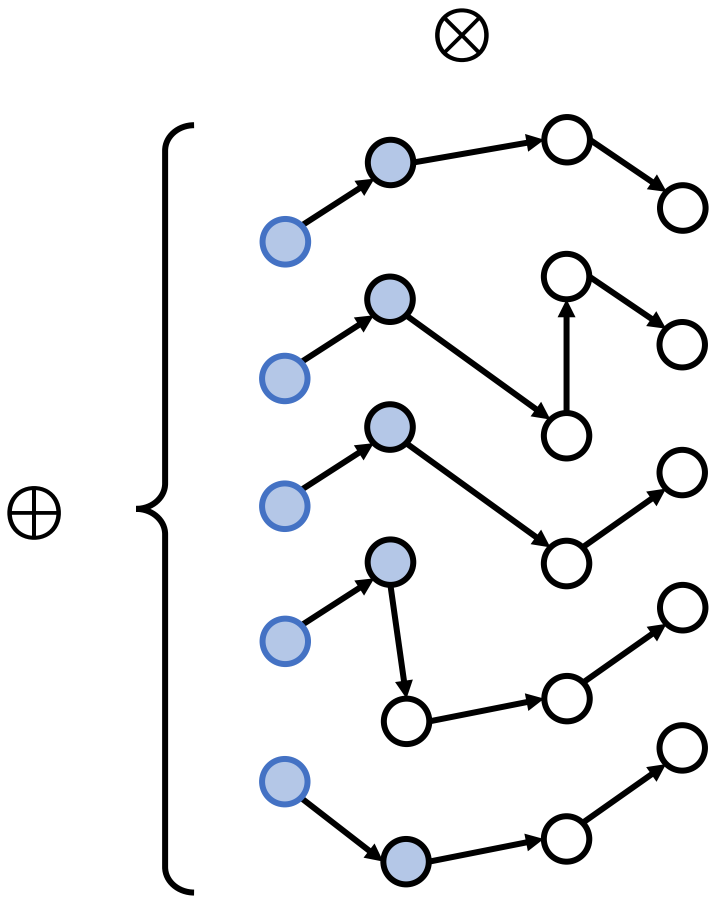
- $ab + ac = a(b + c)$
- $a \otimes b \oplus a \otimes c = a \otimes (b \oplus c)$

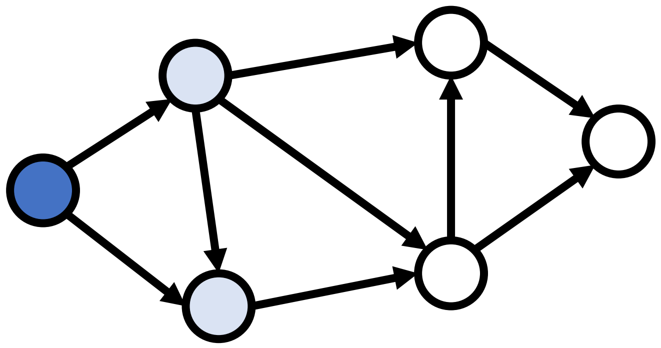


# Dynamic Programming

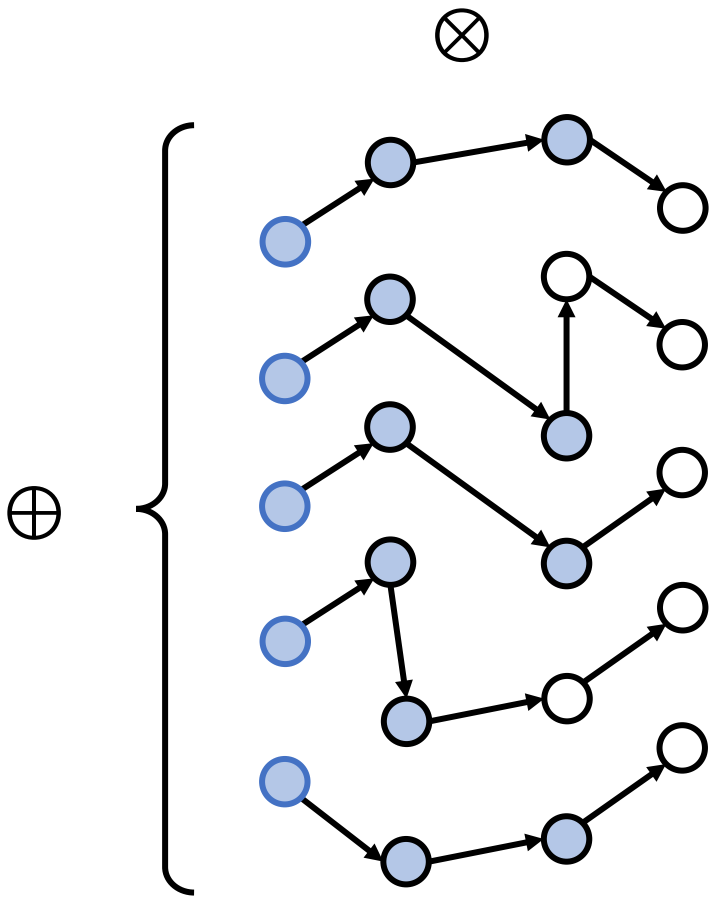


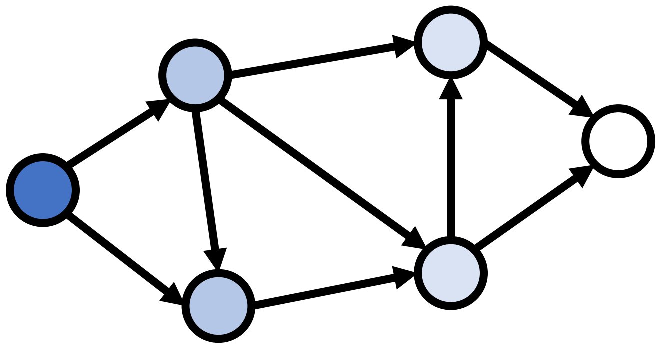
# Dynamic Programming



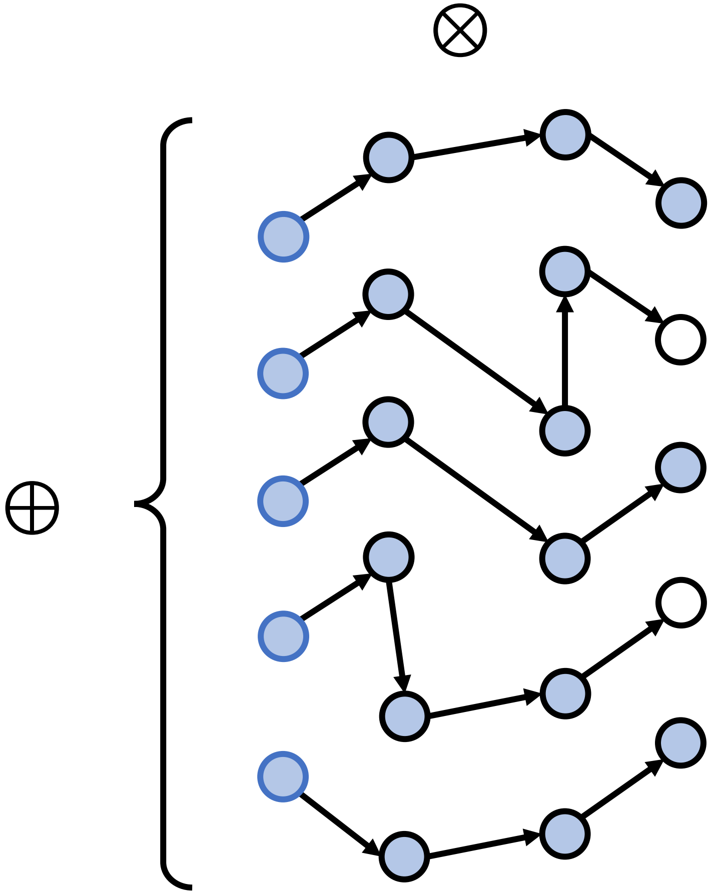



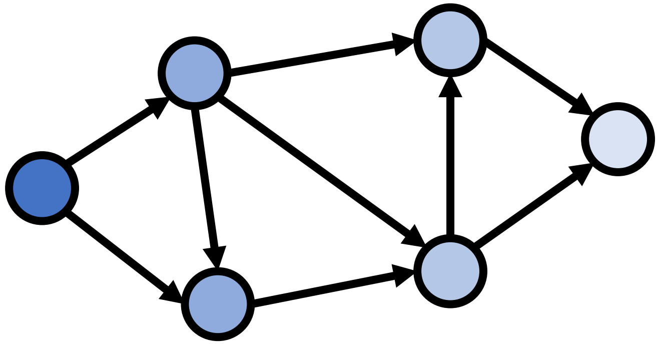
# Dynamic Programming



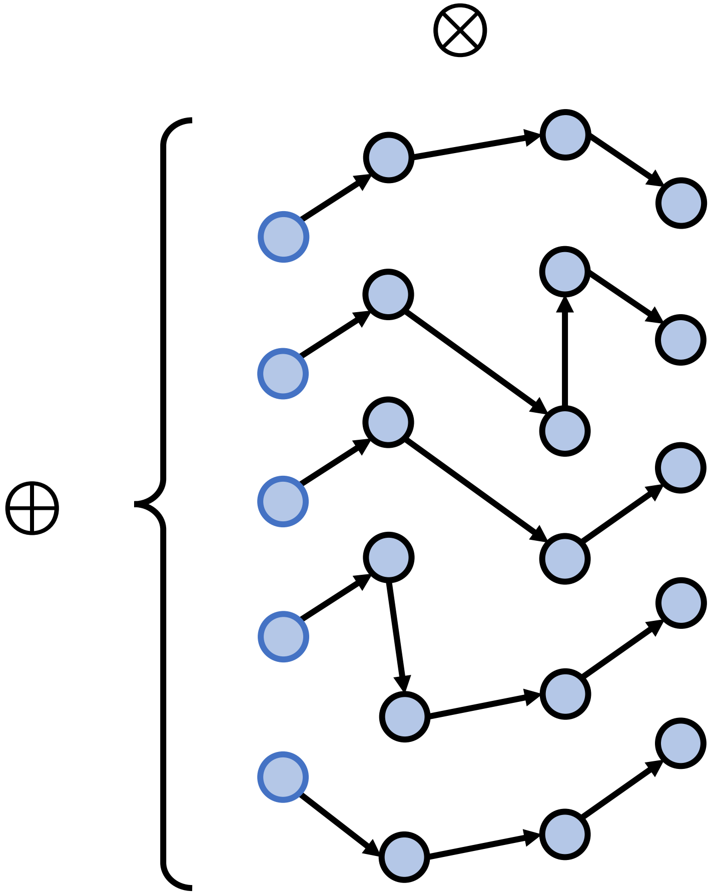



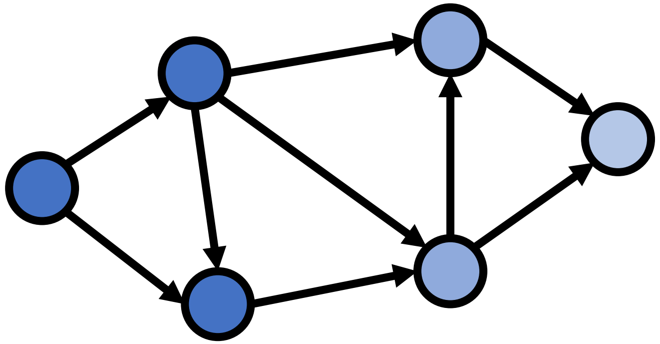
# Dynamic Programming



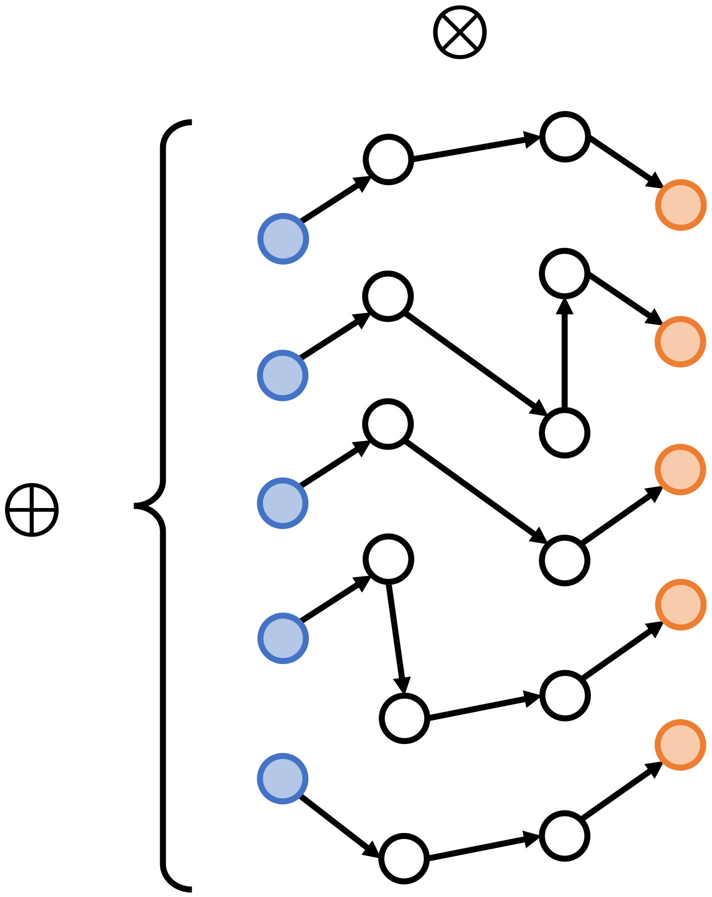



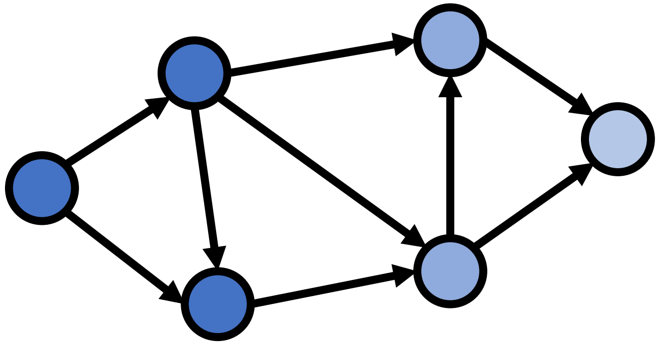
# Dynamic Programming



# Dynamic Programming



power iteration

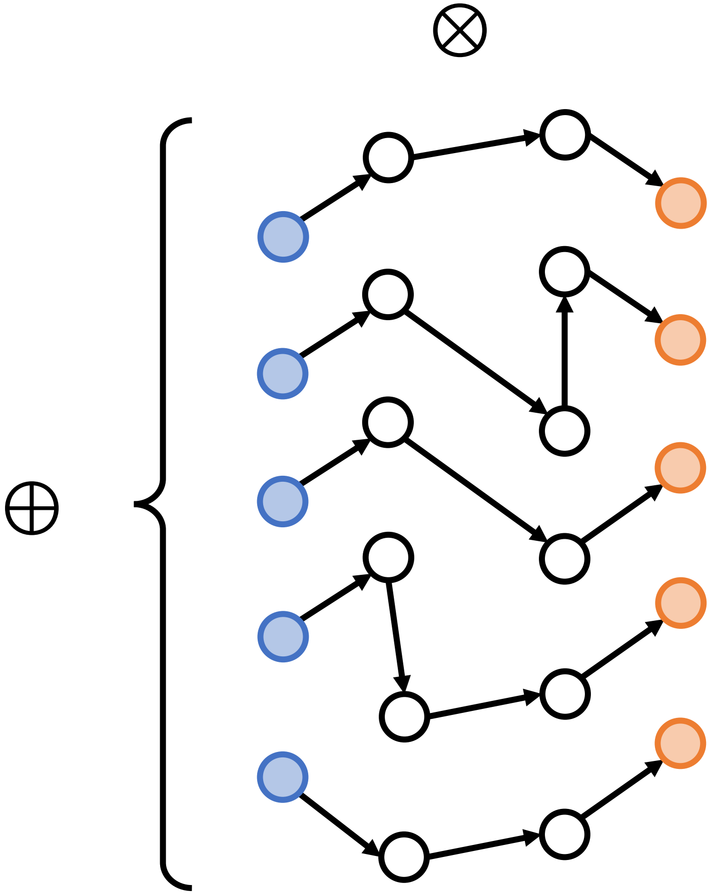
Bellman-Ford algorithm

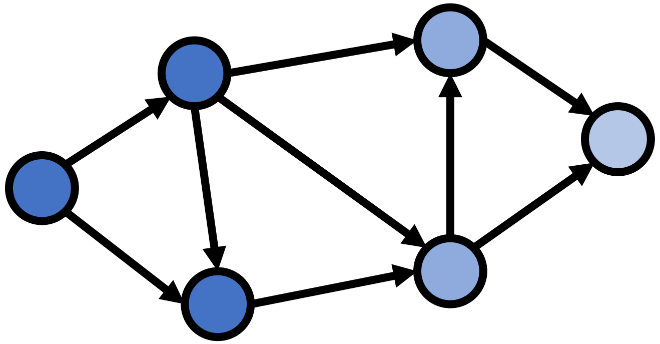
Viterbi algorithm

etc.

scalable 😄

# Dynamic Programming



Generalized
Bellman-Ford algorithm

power iteration

Bellman-Ford algorithm

Viterbi algorithm

etc.

scalable 😄

# Generalized Bellman-Ford Algorithm

- Path formulation

$$\boldsymbol{h}_q(u,v) = \bigoplus_{P \in \mathcal{P}_{uv}} \bigotimes_{e \in P} \boldsymbol{w}_q\,(e)$$

# Generalized Bellman-Ford Algorithm

- Path formulation

$$\boldsymbol{h}_q(u,v) = \bigoplus_{P \in \mathcal{P}_{uv}} \bigotimes_{e \in P} \boldsymbol{w}_q(e)$$

- Generalized Bellman-Ford algorithm

$$\boldsymbol{h}_q^{(0)}(u,v) \leftarrow \begin{cases} \textcircled{1}_q & u = v \\ \textcircled{0}_q & u \neq v \end{cases}$$

boundary condition

# Generalized Bellman-Ford Algorithm

- Path formulation

$$\boldsymbol{h}_q(u,v) = \bigoplus_{P \in \mathcal{P}_{uv}} \bigotimes_{e \in P} \boldsymbol{w}_q(e)$$

- Generalized Bellman-Ford algorithm

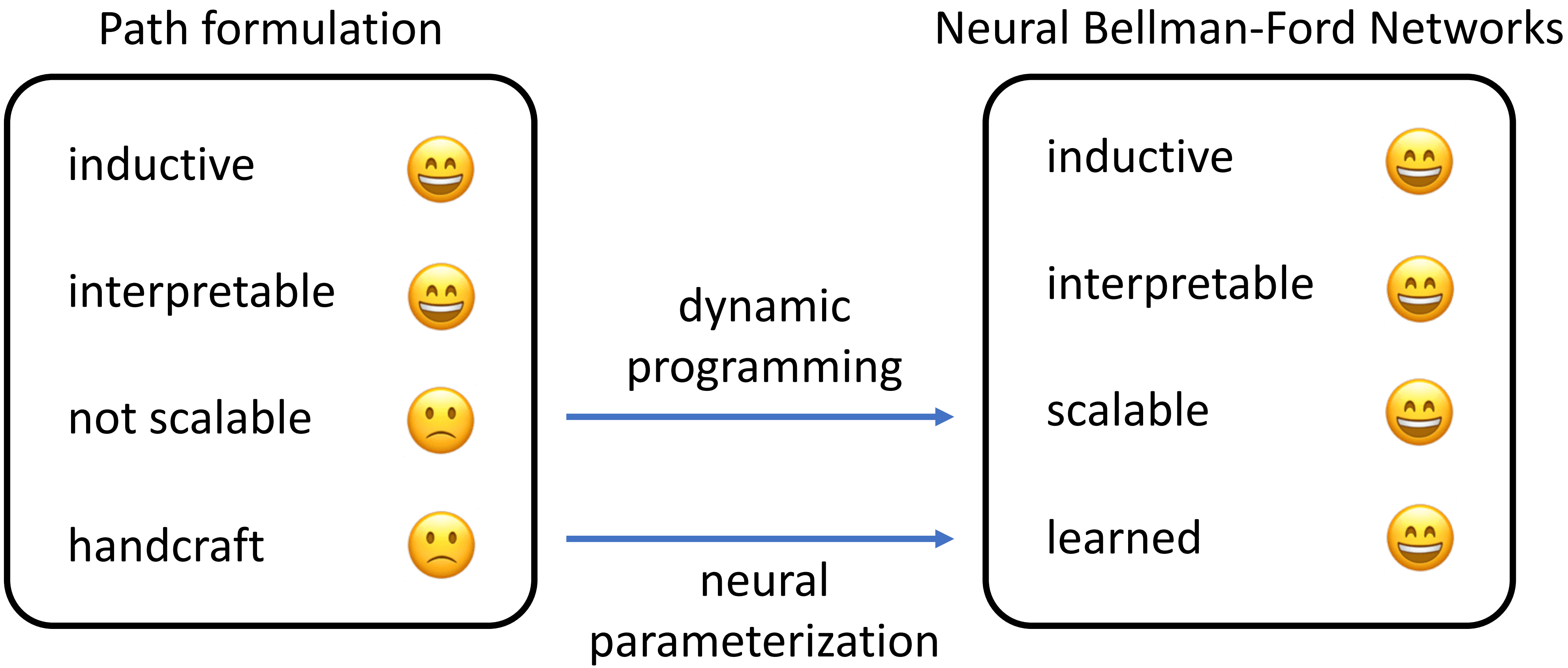
$$\boldsymbol{h}_q^{(0)}(u,v) \leftarrow \begin{cases} ①_q & u = v \\ ⓪_q & u \neq v \end{cases}$$

boundary condition

$$\boldsymbol{h}_q^{(t)}(u,v) \leftarrow \left( \bigoplus_{(x,v) \in \mathcal{E}(v)} \boldsymbol{h}^{(t-1)}(u,x) \otimes \boldsymbol{w}(x,r,v) \right) \oplus \boldsymbol{h}^{(0)}(u,v)$$

Bellman-Ford iteration

# Make It Better



# Neural Parameterization

- Generalized Bellman-Ford algorithm

$$\boldsymbol{h}_q^{(0)}(u, v) \leftarrow 1_q(u = v)$$

$$\boldsymbol{h}_q^{(t)}(u, v) \leftarrow \left( \bigoplus_{(x,v) \in \mathcal{E}(v)} \boldsymbol{h}_q^{(t-1)}(u, x) \otimes \boldsymbol{w}_q(x, r, v) \right) \oplus \boldsymbol{h}_q^{(0)}(u, v)$$

# Neural Parameterization

- Generalized Bellman-Ford algorithm

$$\boldsymbol{h}_q^{(0)}(u,v) \leftarrow 1_q(u=v)$$

$$\boldsymbol{h}_q^{(t)}(u,v) \leftarrow \left( \bigoplus_{(x,v)\in\mathcal{E}(v)} \boldsymbol{h}_q^{(t-1)}(u,x) \otimes \boldsymbol{w}_q(x,r,v) \right) \oplus \boldsymbol{h}_q^{(0)}(u,v)$$

- Neural parameterization
  - Single-source case: abbreviate $\boldsymbol{h}_q^{(t)}(u,v)$ as $\boldsymbol{h}_v^{(t)}$

$$\boldsymbol{h}_v^{(0)} \leftarrow \mathrm{Indicator}(u,v,q)$$

$$\boldsymbol{h}_v^{(t)} \leftarrow \mathrm{Aggregate}\left(\left\{\mathrm{Message}\left(\boldsymbol{h}_x^{(t-1)}, \boldsymbol{w}_q(x,r,v)\right) \middle| (x,r,v)\in\mathcal{E}(v)\right\} \cup \left\{\boldsymbol{h}_v^{(0)}\right\}\right)$$

# Neural Parameterization

$$\boldsymbol{h}_v^{(0)} \leftarrow \text{Indicator}(u, v, q)$$

$$\boldsymbol{h}_v^{(t)} \leftarrow \text{Aggregate}\left(\left\{\text{Message}\left(\boldsymbol{h}_x^{(t-1)}, \boldsymbol{w}_q(x, r, v)\right) \middle| (x, r, v) \in \mathcal{E}(v)\right\} \cup \left\{\boldsymbol{h}_v^{(0)}\right\}\right)$$

# Neural Parameterization

$$\boldsymbol{h}_v^{(0)} \leftarrow \text{Indicator}(u, v, q)$$

$$\boldsymbol{h}_v^{(t)} \leftarrow \text{Aggregate}\left(\left\{\text{Message}\left(\boldsymbol{h}_x^{(t-1)}, \boldsymbol{w}_q(x, r, v)\right) \middle| (x, r, v) \in \mathcal{E}(v)\right\} \cup \left\{\boldsymbol{h}_v^{(0)}\right\}\right)$$

- Indicator: learned embeddings for $q$
- Message: relational operators of KG embeddings (TransE/DistMult/RotatE etc.)
- Aggregate: permutation invariant function (sum/mean/max/PNA etc.)

# Revisit Path Formulation

- Generalized Bellman-Ford algorithm

$$\boldsymbol{h}_q^{(0)}(u,v) \leftarrow \begin{cases} \boldsymbol{q} & u = v \\ 0 & u \neq v \end{cases}$$

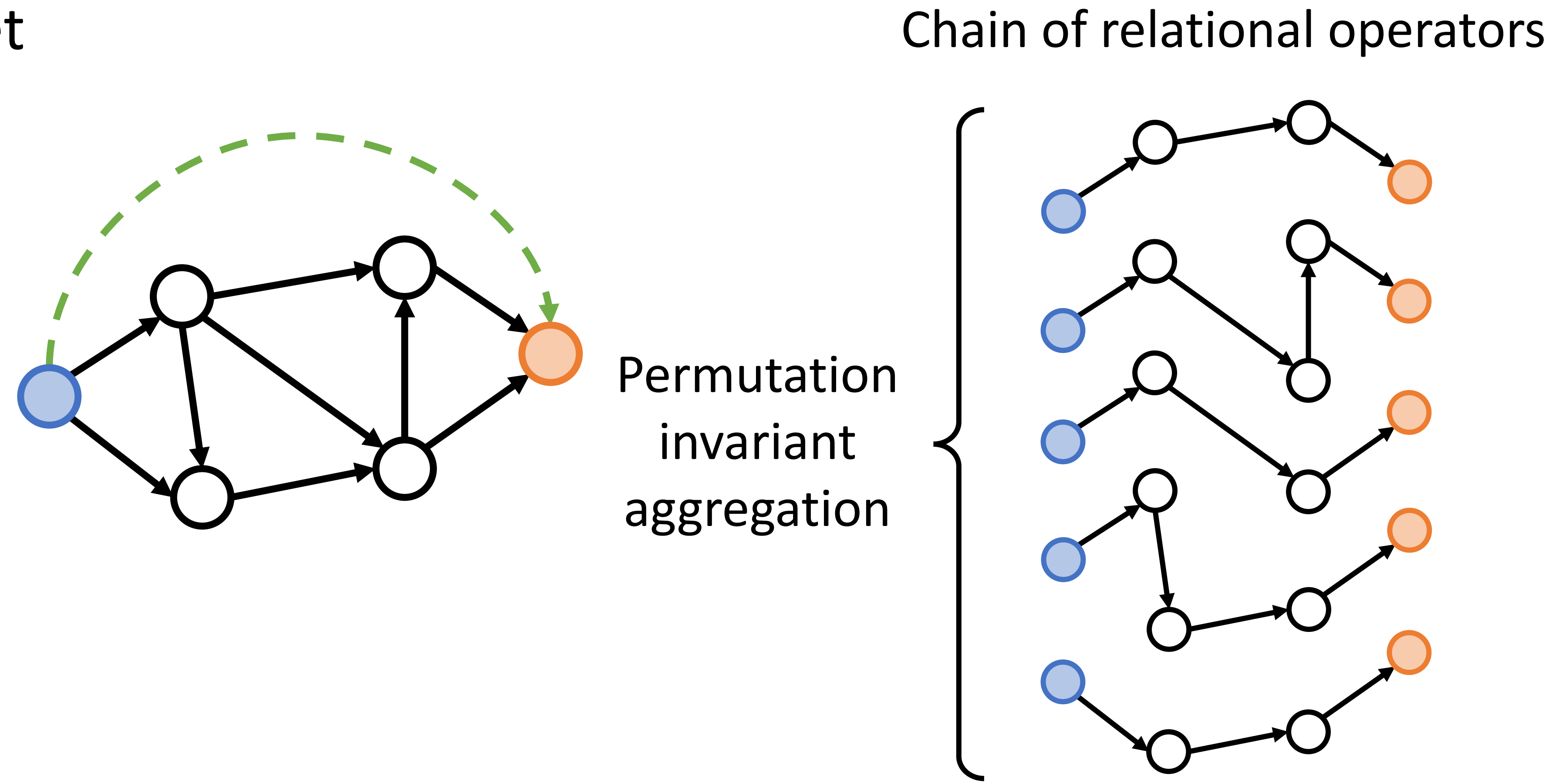
$$\boldsymbol{h}_q^{(t)}(u,v) \leftarrow \mathrm{Aggregate}\left(\left\{\mathrm{Message}\left(\boldsymbol{h}_q^{(t-1)}(u,x), \boldsymbol{w}_q(x,r,v)\right)\right\} \cup \left\{\boldsymbol{h}_v^{(0)}\right\}\right)$$

- Path formulation

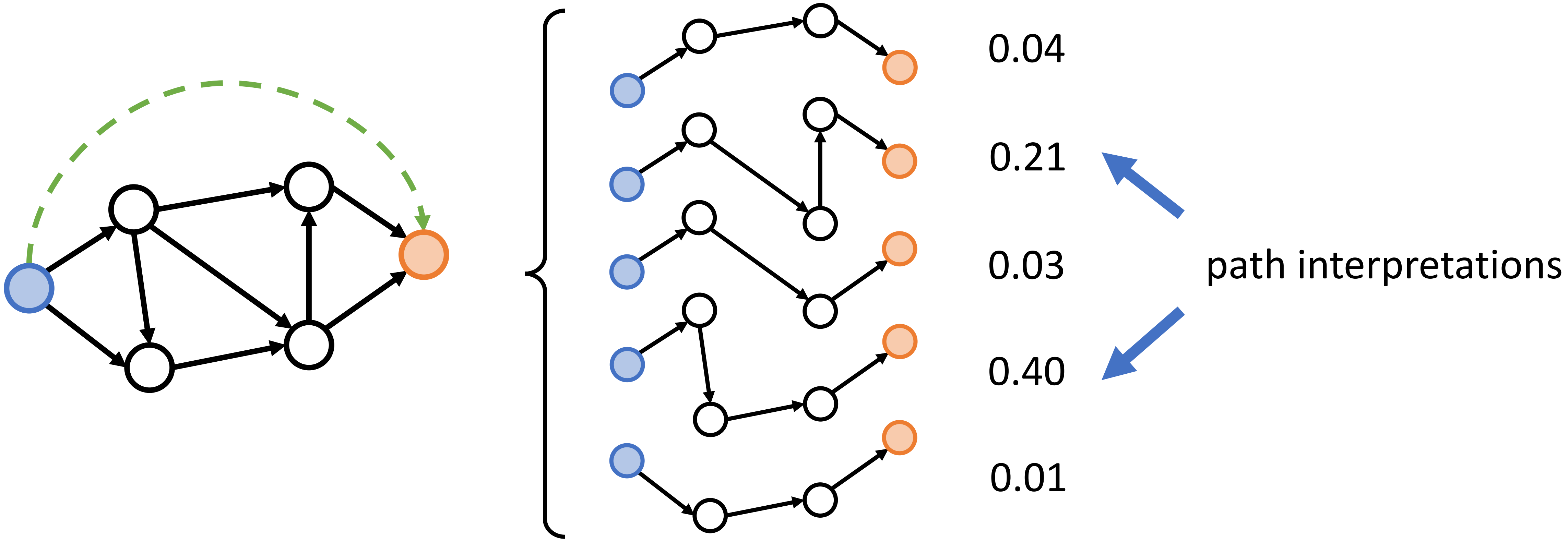
$$\boldsymbol{h}_q(u,v) = \underset{t=0}{\overset{\infty}{\mathrm{Aggregate}}}(\underbrace{\{\mathrm{Message}(\ldots \mathrm{Message}(\boldsymbol{q}, r_1), \ldots r_t)\}}_{t\ \mathrm{Message}})$$

# Revisit Path Formulation

- NBFNet



# Interpretation



# Link Prediction

- Apply a MLP over $\boldsymbol{h}_q^{(t)}(u, v)$
  - $p(v|u, q) = \sigma\left(f\left(\boldsymbol{h}_q(u, v)\right)\right)$
- Homogeneous variant
  - Use the same relation for every $q$ and $r$
- Undirected homogeneous variant
  - $p(u, v) = \sigma\left(f\left(\boldsymbol{h}(u, v) + \boldsymbol{h}(v, u)\right)\right)$

- Train with standard negative sampling
  - Time complexity is independent of #negative
  - Can use arbitrary large #negative or full softmax

# Results

- Knowledge graph completion

| Class | Method | FB15k-237 | | | | | WN18RR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MR | MRR | H@1 | H@3 | H@10 | MR | MRR | H@1 | H@3 | H@10 |
| **Path-based** | Path Ranking [35] | 3521 | 0.174 | 0.119 | 0.186 | 0.285 | 22438 | 0.324 | 0.276 | 0.360 | 0.406 |
| | NeuralLP [69] | - | 0.240 | - | - | 0.362 | - | 0.435 | 0.371 | 0.434 | 0.566 |
| | DRUM [46] | - | 0.343 | 0.255 | 0.378 | 0.516 | - | 0.486 | 0.425 | 0.513 | 0.586 |
| **Embeddings** | TransE [6] | 357 | 0.294 | - | - | 0.465 | 3384 | 0.226 | - | - | 0.501 |
| | DistMult [68] | 254 | 0.241 | 0.155 | 0.263 | 0.419 | 5110 | 0.43 | 0.39 | 0.44 | 0.49 |
| | ComplEx [58] | 339 | 0.247 | 0.158 | 0.275 | 0.428 | 5261 | 0.44 | 0.41 | 0.46 | 0.51 |
| | RotatE [52] | 177 | 0.338 | 0.241 | 0.375 | 0.553 | 3340 | 0.476 | 0.428 | 0.492 | 0.571 |
| | HAKE [76] | - | 0.346 | 0.250 | 0.381 | 0.542 | - | 0.497 | 0.452 | 0.516 | 0.582 |
| | LowFER [1] | - | 0.359 | 0.266 | 0.396 | 0.544 | - | 0.465 | 0.434 | 0.479 | 0.526 |
| **GNNs** | RGCN [48] | 221 | 0.273 | 0.182 | 0.303 | 0.456 | 2719 | 0.402 | 0.345 | 0.437 | 0.494 |
| | GraIL [55] | 2053 | - | - | - | - | 2539 | - | - | - | - |
| | NBFNet | **114** | **0.415** | **0.321** | **0.454** | **0.599** | **636** | **0.551** | **0.497** | **0.573** | **0.666** |

# Results

- Homogeneous link prediction

| Class | Method | Cora | | Citeseer | | PubMed | |
|---|---|---|---|---|---|---|---|
| | | AUROC | AP | AUROC | AP | AUROC | AP |
| **Path-based** | Katz Index [30] | 0.834 | 0.889 | 0.768 | 0.810 | 0.757 | 0.856 |
| | Personalized PageRank [42] | 0.845 | 0.899 | 0.762 | 0.814 | 0.763 | 0.860 |
| | SimRank [28] | 0.838 | 0.888 | 0.755 | 0.805 | 0.743 | 0.829 |
| **Embeddings** | DeepWalk [43] | 0.831 | 0.850 | 0.805 | 0.836 | 0.844 | 0.841 |
| | LINE [53] | 0.844 | 0.876 | 0.791 | 0.826 | 0.849 | 0.888 |
| | node2vec [17] | 0.872 | 0.879 | 0.838 | 0.868 | 0.891 | 0.914 |
| **GNNs** | VGAE [32] | 0.914 | 0.926 | 0.908 | 0.920 | 0.944 | 0.947 |
| | S-VGAE [12] | 0.941 | 0.941 | **0.947** | **0.952** | 0.960 | 0.960 |
| | SEAL [73] | 0.933 | 0.942 | 0.905 | 0.924 | 0.978 | 0.979 |
| | TLC-GNN [67] | 0.934 | 0.931 | 0.909 | 0.916 | 0.970 | 0.968 |
| | NBFNet | **0.956** | **0.962** | 0.923 | 0.936 | **0.983** | **0.982** |

# Results

- Inductive relation prediction

| Class | Method | FB15k-237 v1 | FB15k-237 v2 | FB15k-237 v3 | FB15k-237 v4 | WN18RR v1 | WN18RR v2 | WN18RR v3 | WN18RR v4 |
|---|---|---|---|---|---|---|---|---|---|
| **Path-based** | NeuralLP [16] | 0.529 | 0.589 | 0.529 | 0.559 | 0.744 | 0.689 | 0.462 | 0.671 |
| | DRUM [46] | 0.529 | 0.587 | 0.529 | 0.559 | 0.744 | 0.689 | 0.462 | 0.671 |
| | RuleN [39] | 0.498 | 0.778 | 0.877 | 0.856 | 0.809 | 0.782 | 0.534 | 0.716 |
| **GNNs** | GraIL [55] | 0.642 | 0.818 | 0.828 | 0.893 | 0.825 | 0.787 | 0.584 | 0.734 |
| | NBFNet | **0.834** | **0.949** | **0.951** | **0.960** | **0.948** | **0.905** | **0.893** | **0.890** |

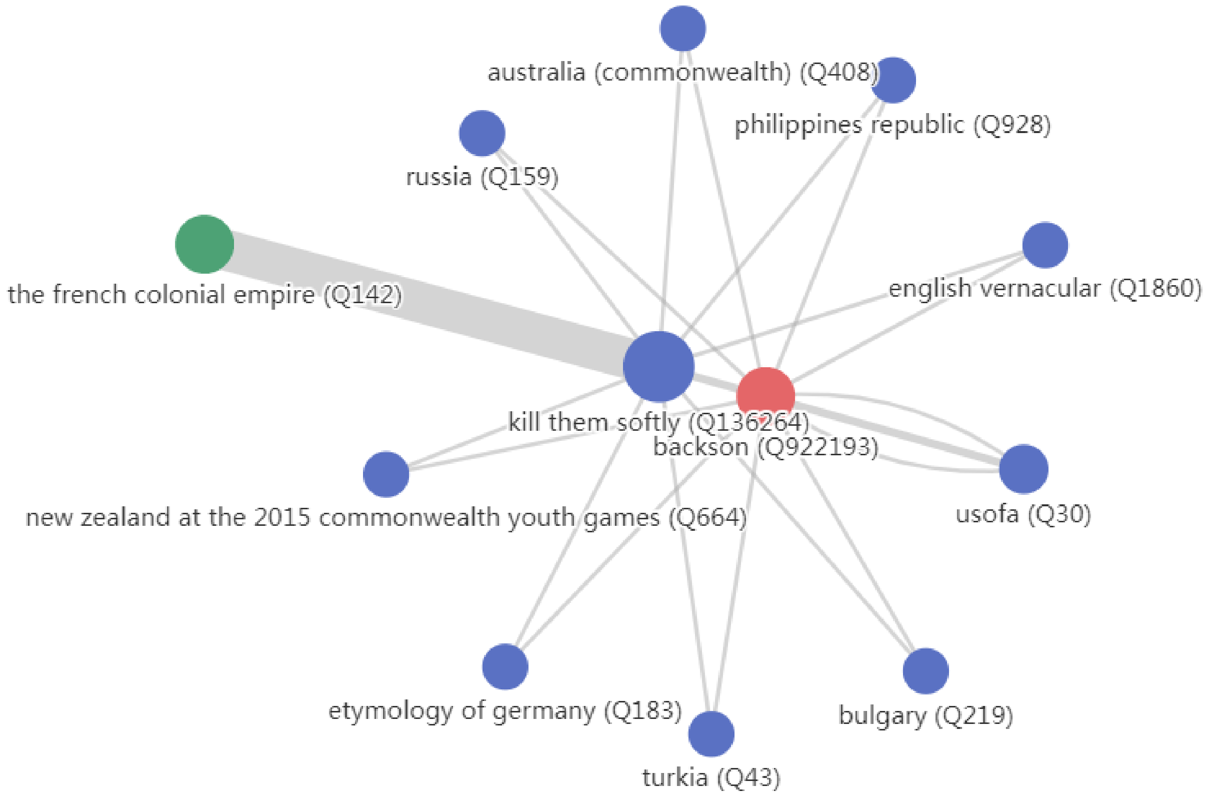
# Interpretation

- FB15k-237

| | |
|---|---|
| **Query** | $\langle$*O. Hardy, nationality, U.S.*$\rangle$ |
| 0.243 | $\langle$*O. Hardy, impersonate*$^{-1}$*, R. Little*$\rangle \wedge \langle$*R. Little, nationality, U.S.*$\rangle$ |
| 0.224 | $\langle$*O. Hardy, ethnicity*$^{-1}$*, Scottish American*$\rangle \wedge \langle$*Scottish American, distribution, U.S.*$\rangle$ |
| **Query** | $\langle$*Florence, vacationer, D.C. Henrie*$\rangle$ |
| 0.251 | $\langle$*Florence, contain*$^{-1}$*, Italy*$\rangle \wedge \langle$*Italy, capital, Rome*$\rangle \wedge \langle$*Rome, vacationer, D.C. Henrie*$\rangle$ |
| 0.183 | $\langle$*Florence, place live*$^{-1}$*, G.F. Handel*$\rangle \wedge \langle$*G.F. Handel, place live, Rome*$\rangle \wedge \langle$*Rome, vacationer, D.C. Henrie*$\rangle$ |
| **Query** | $\langle$*Pearl Harbor (film), language, Japanese*$\rangle$ |
| 0.211 | $\langle$*Pearl Harbor (film), film actor, C.-H. Tagawa*$\rangle \wedge \langle$*C.-H. Tagawa, nationality, Japan*$\rangle$ $\wedge \langle$*Japan, country of origin, Yu-Gi-Oh!*$\rangle \wedge \langle$*Yu-Gi-Oh!, language, Japanese*$\rangle$ |
| 0.208 | $\langle$*Pearl Harbor (film), film actor, C.-H. Tagawa*$\rangle \wedge \langle$*C.-H. Tagawa, nationality, Japan*$\rangle$ $\wedge \langle$*Japan, official language, Japanese*$\rangle$ |

# Interpretation

- p(France | Winnie the Pooh, release region)




https://deepgraphlearning.github.io/project/reasoning

# OGB-LSC @ KDD Cup 2021

Large - Scale Challenge

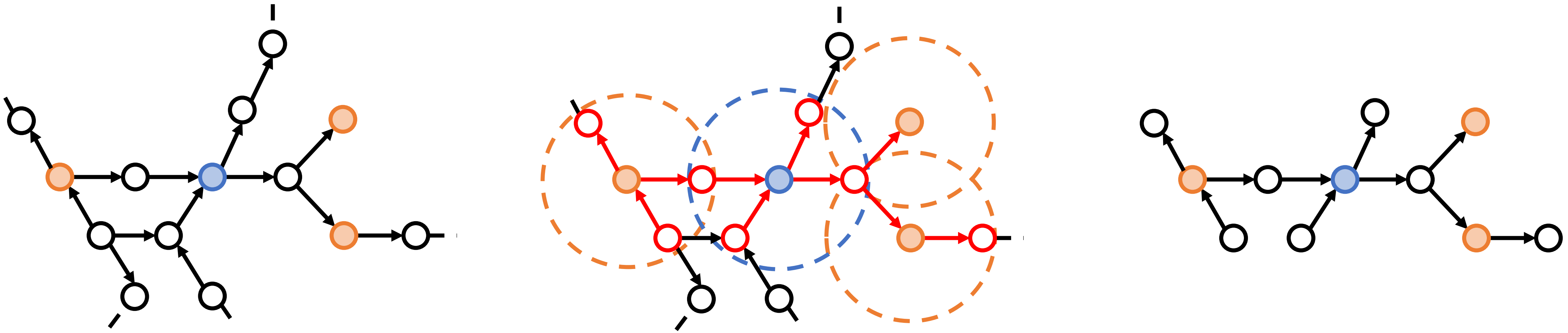
- Setting
  - Rank 1 positive tail entity against 1000 negative tail entities

- Dataset: 90M entity, 1.3K relation, 0.5B triplets
  - 0.5B triplets = 12GB memory
  - Impossible to train such a large graph on GPUs

- How to scale up NBFNet?
  - Sample small graphs like other inductive GNNs

# Bi-directional BFS sampling

- Intuition: long paths are not important for prediction
  - Sample nodes and edges that can be reached by BFS
  - Randomly down sample nodes with large neighborhoods
  - Bi-directional BFS to further reduce the size



# OGB-LSC @ KDD Cup 2021

Large - Scale Challenge

- Results (MRR)
  - Single model: valid: 0.9237
  - Ensemble of 6 models: valid: 0.9304, test: 0.9178

- Rank 12 out of 39 teams, but ...
  - Stronger than all single models reported by the winner
  - Probably the most parameter efficient

# Summary

- Generalize/transfer to unseen graphs with the same semantics
- Interpret predictions via top weighted paths
- Scalable compared to path-based methods and GNNs
- Super parameter efficient compared to popular embedding methods
- Verified on several link prediction tasks and datasets
- Verified in OGB large scale competition (rank 12 out of 39 teams)

- Code at: https://github.com/DeepGraphLearning/NBFNet

# Thanks for your attention!

- Q & A