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

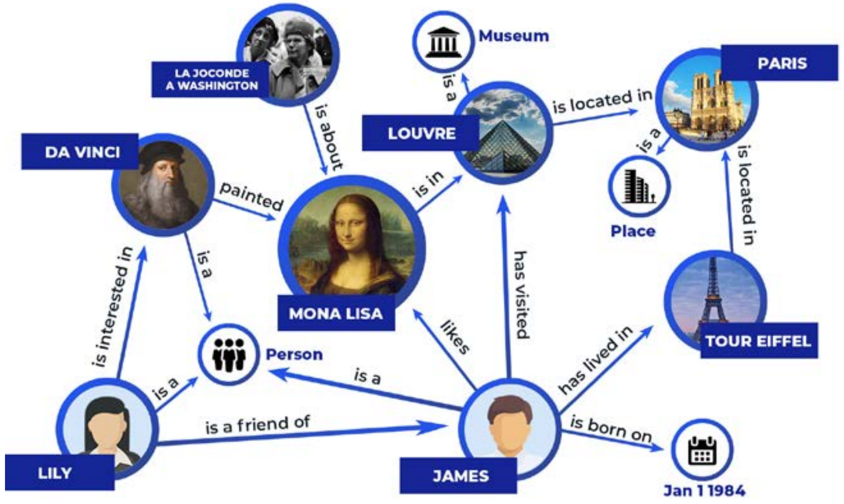
Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, Jian Tang

NeurIPS 2021

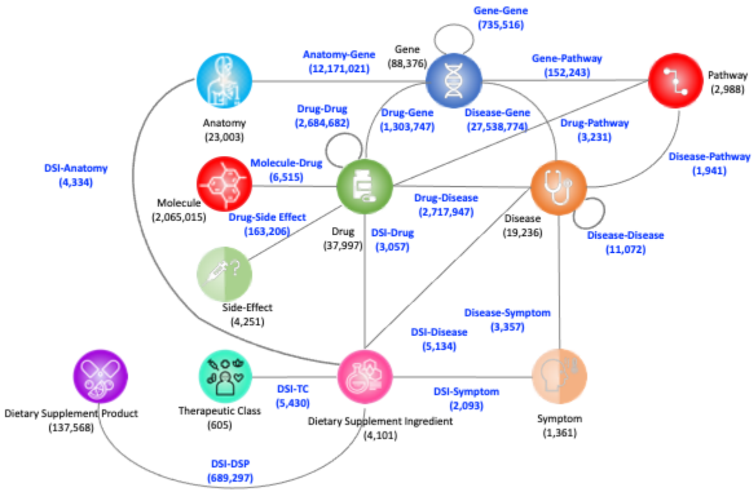
# Link Prediction

- Predict the interactions between nodes

social networks



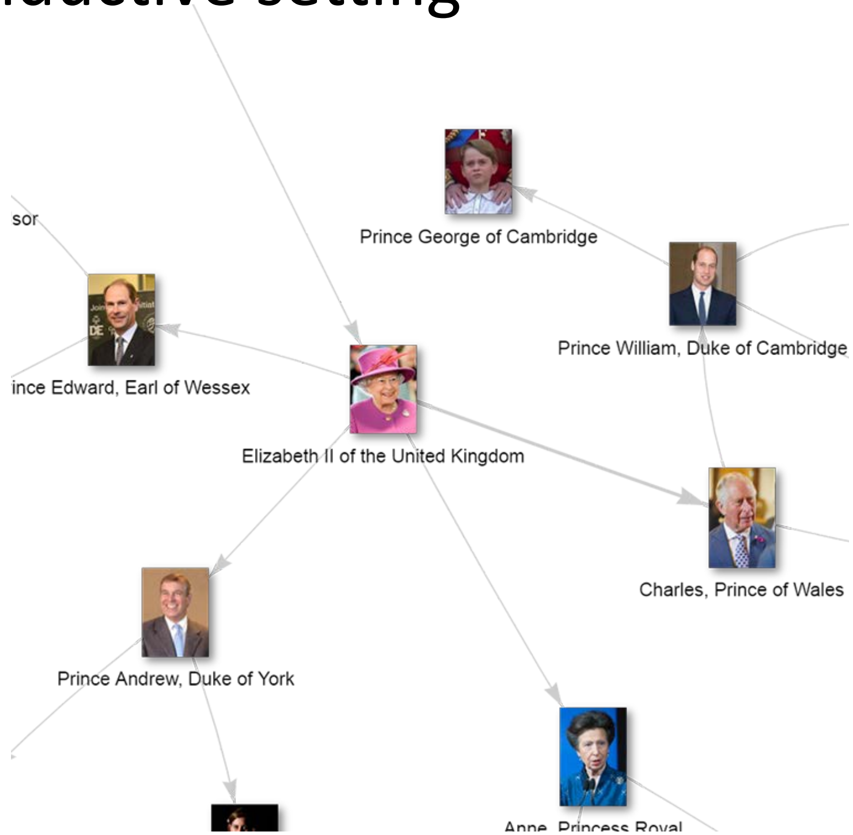

knowledge graph completion



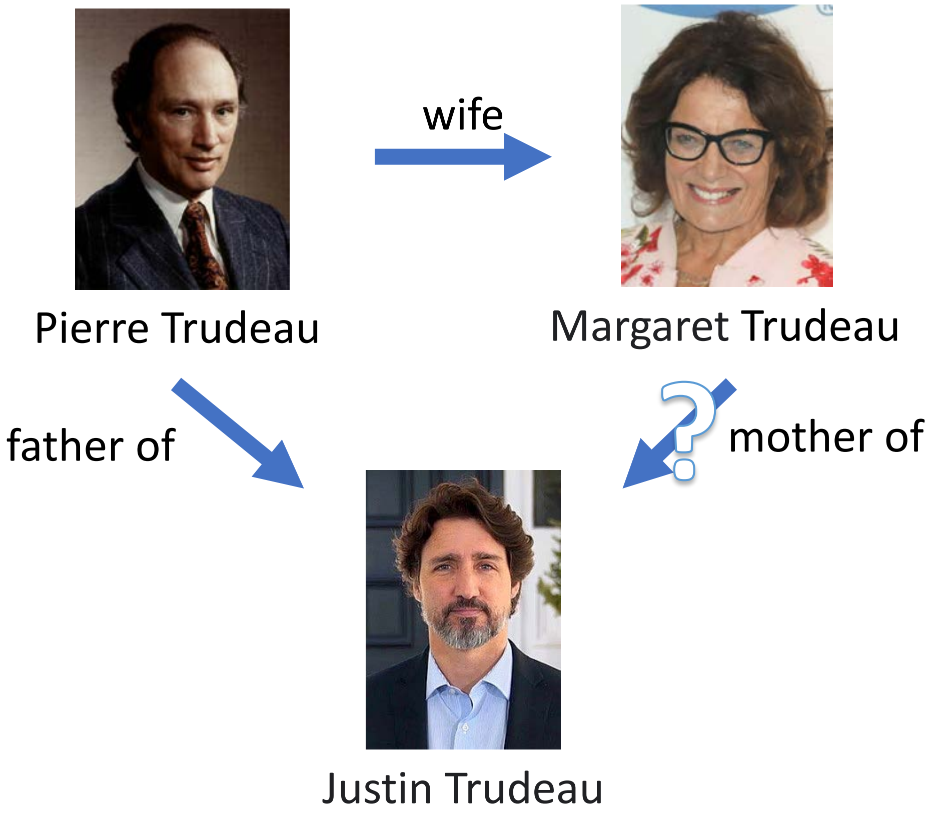

drug repurposing

# Challenges

- Inductive setting




Train



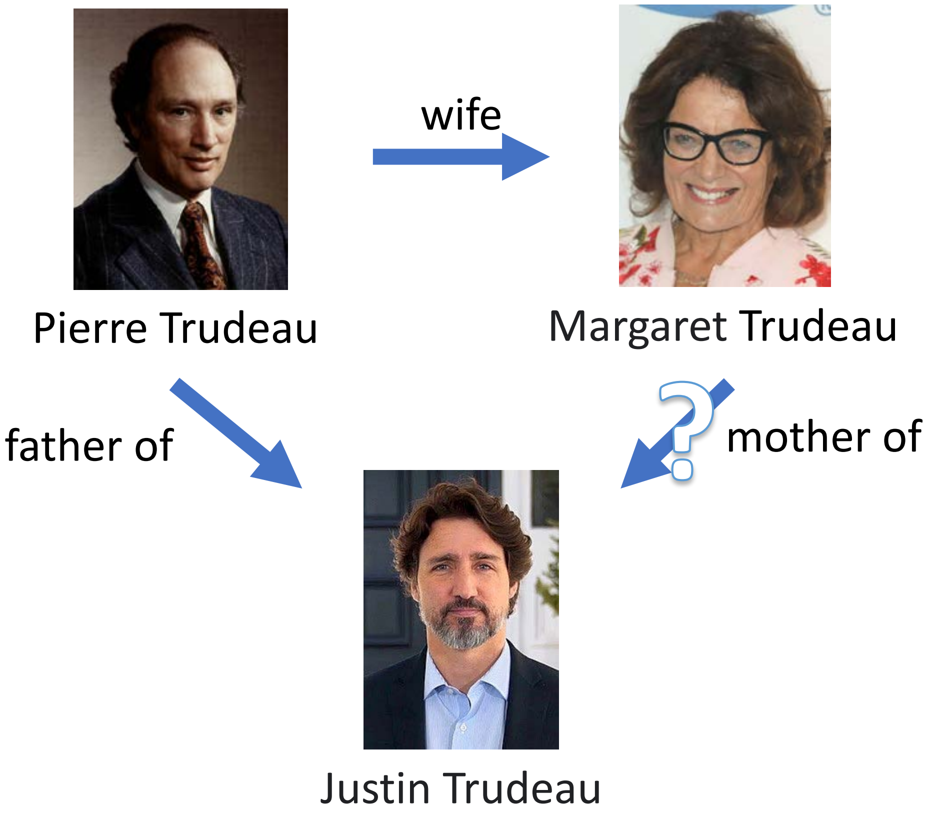

Test (a new graph)

# Challenges

- Interpretability

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

- Why?

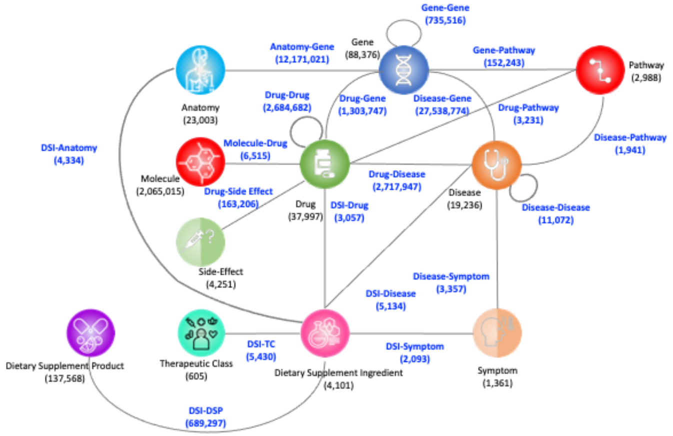


# Challenges

- Large scale

WIKIDATA

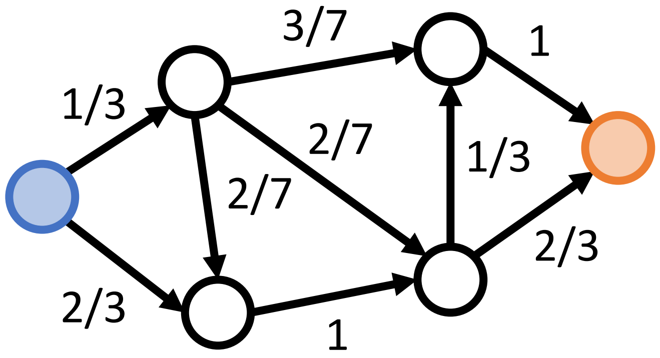
Wikidata
87M entities
504M triplets

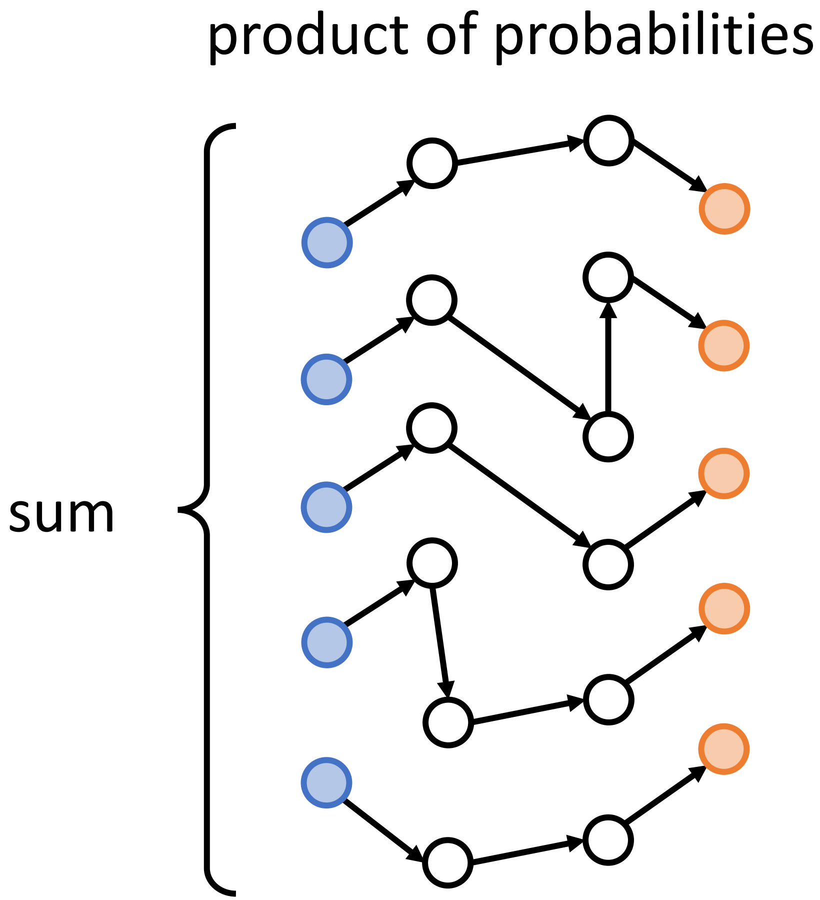



CBKH
2.4M nodes
48M edges

# Traditional Methods

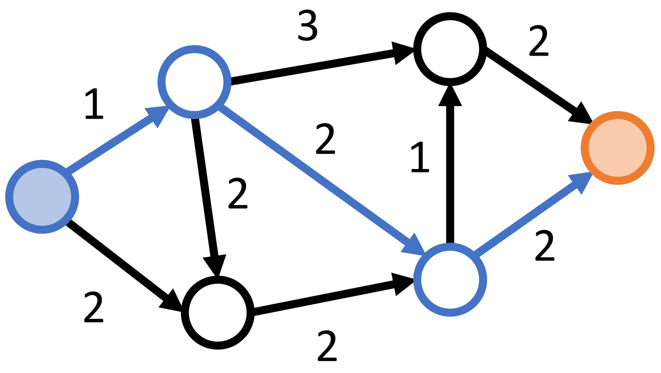
- Personalized PageRank

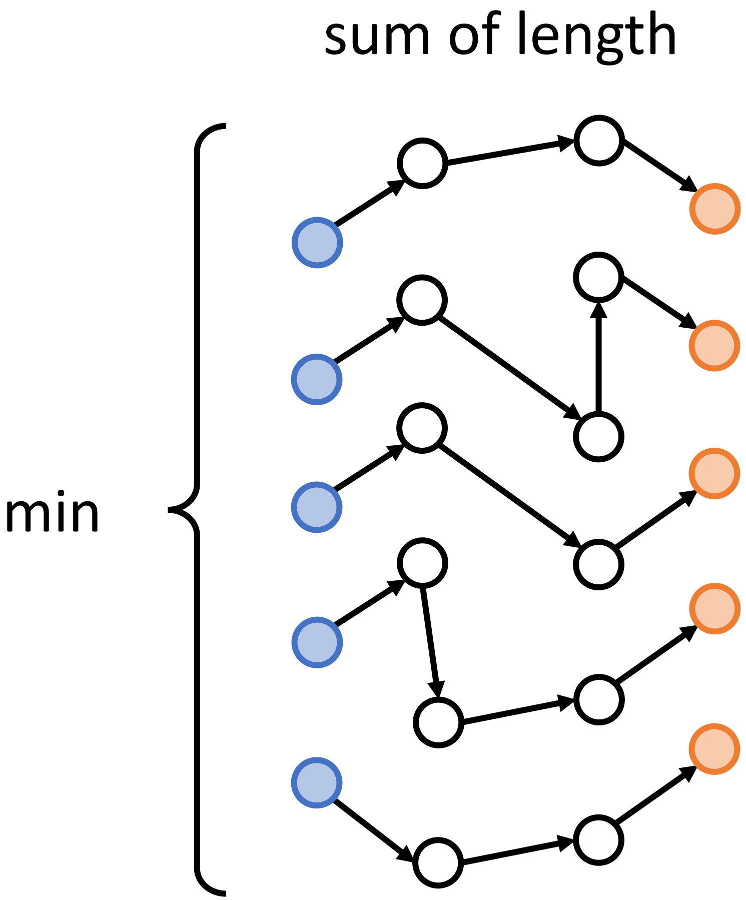


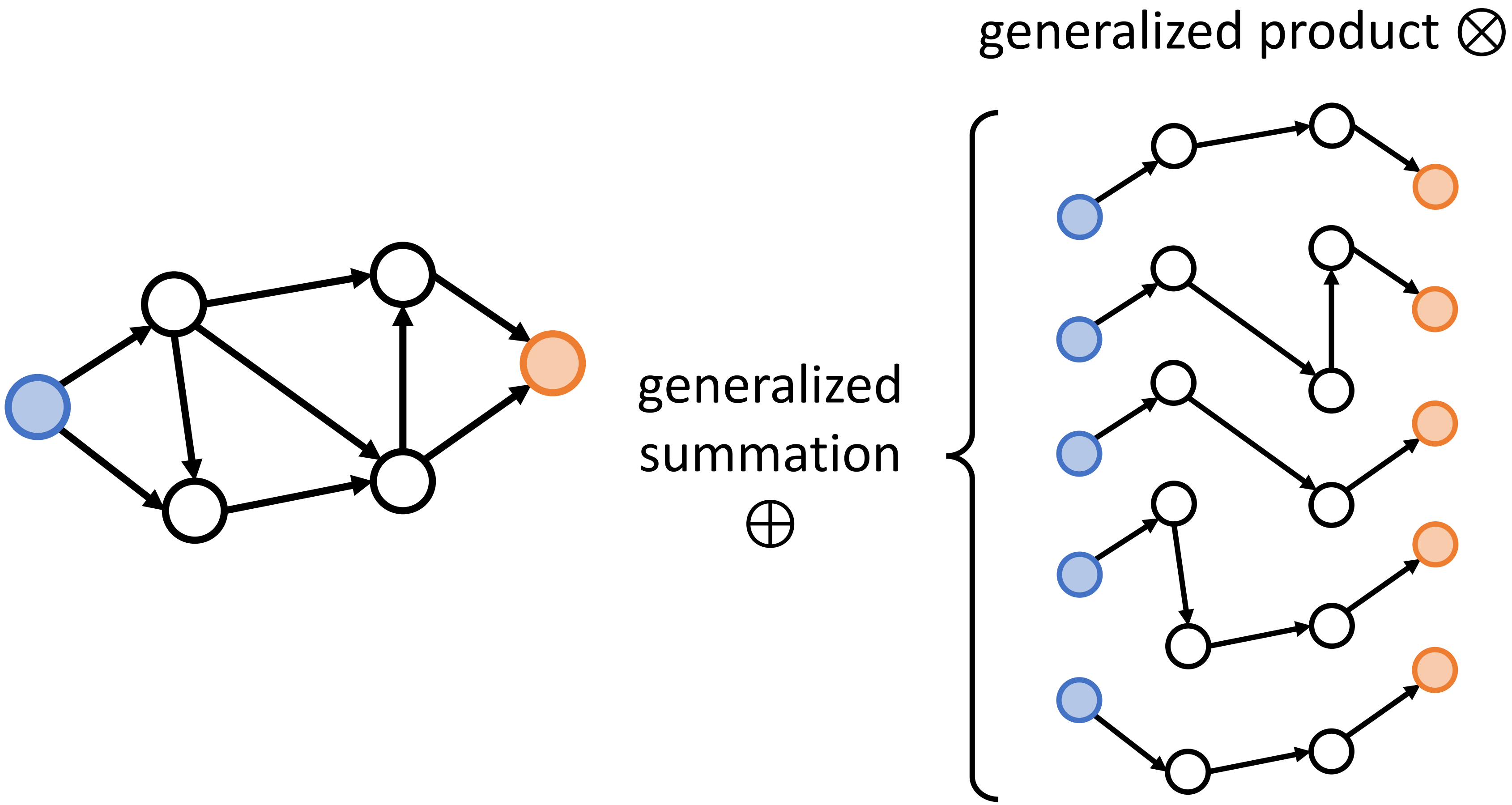
# Traditional Methods

- Graph distance

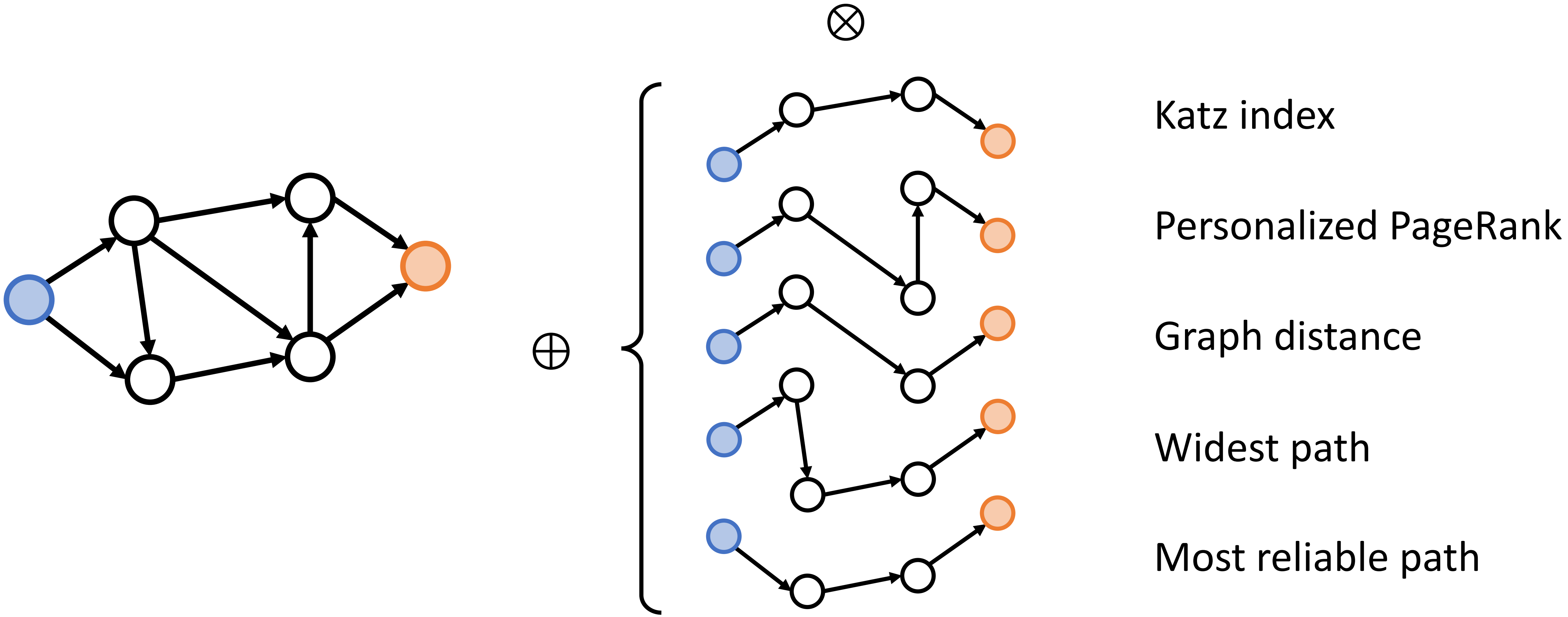


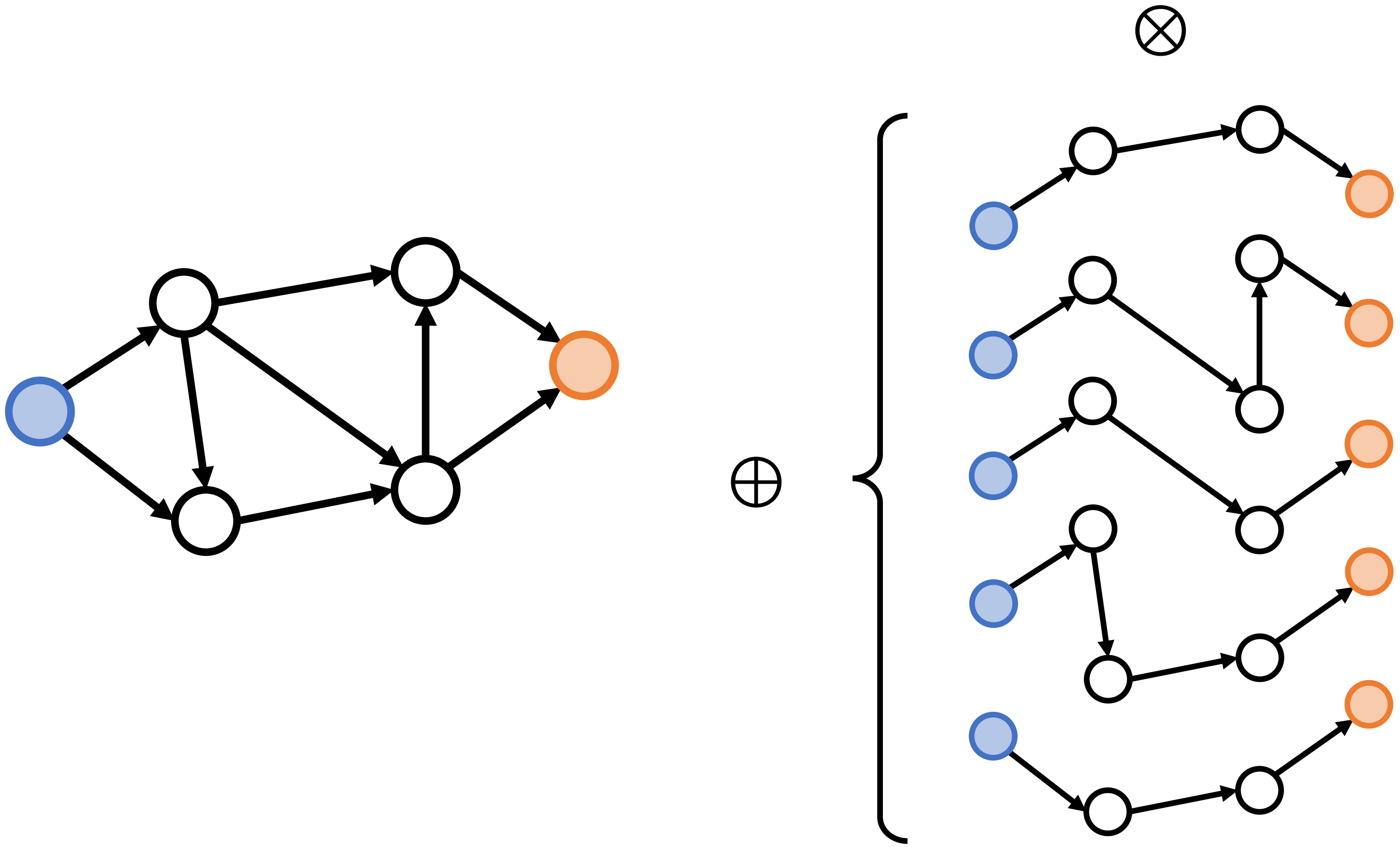
# Path Formulation



# Path Formulation



# Path Formulation



inductive 😄

interpretable 😄

not scalable 🙁

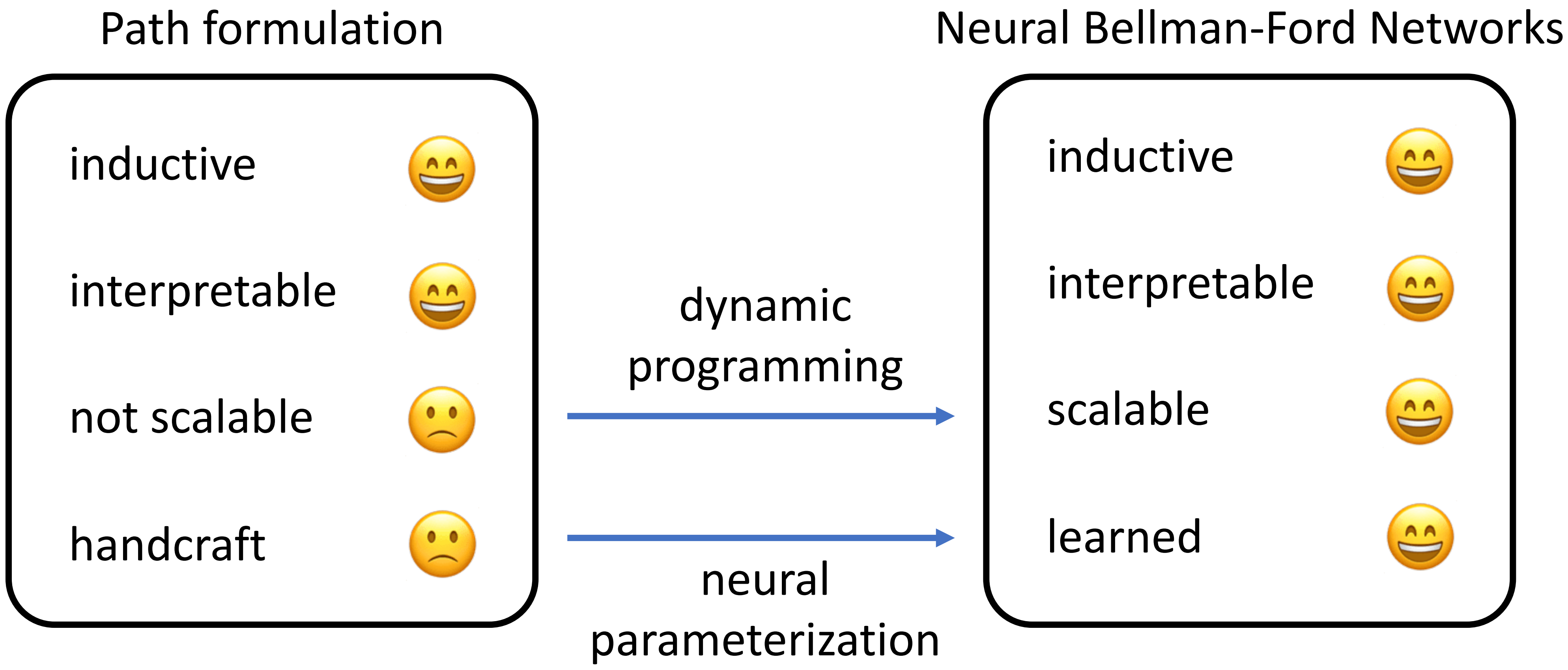
handcraft 🙁

# Make It Better

Path formulation

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

# Make It Better

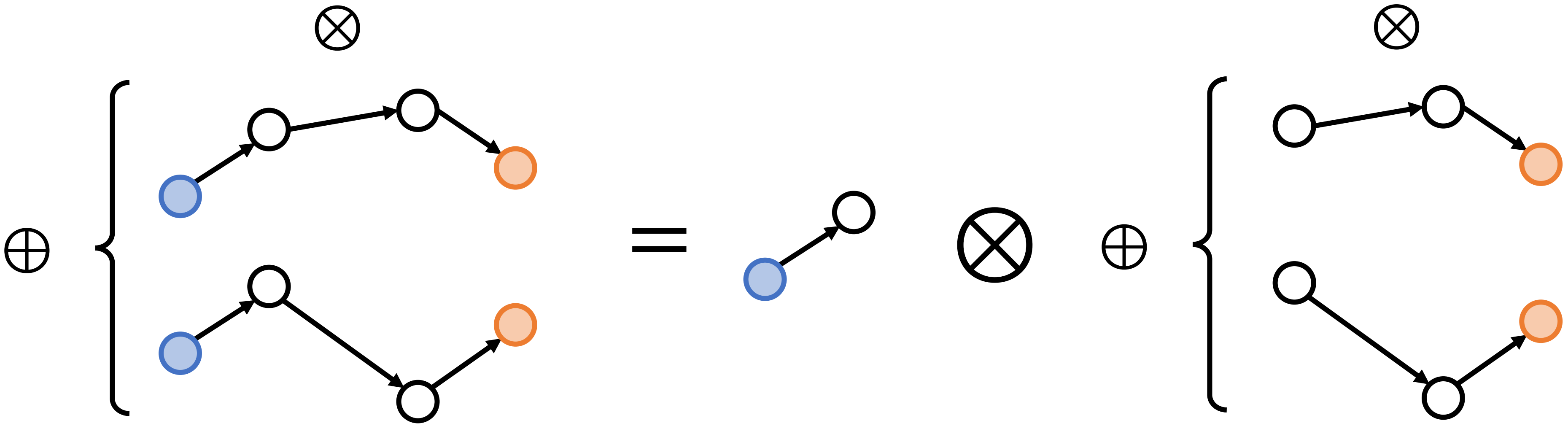


# Dynamic Programming

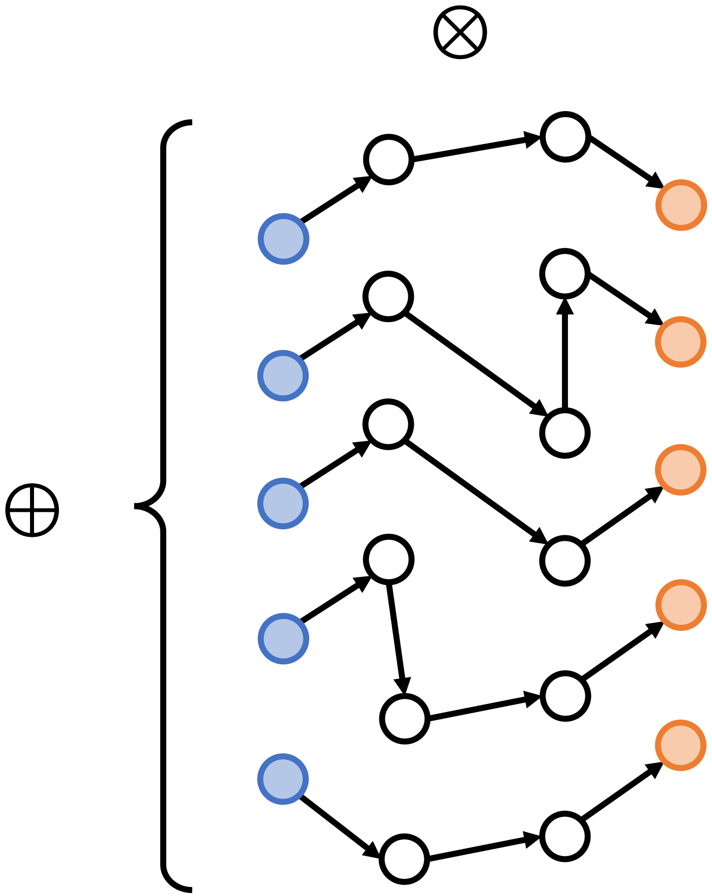
- $ab + ac = a(b + c)$

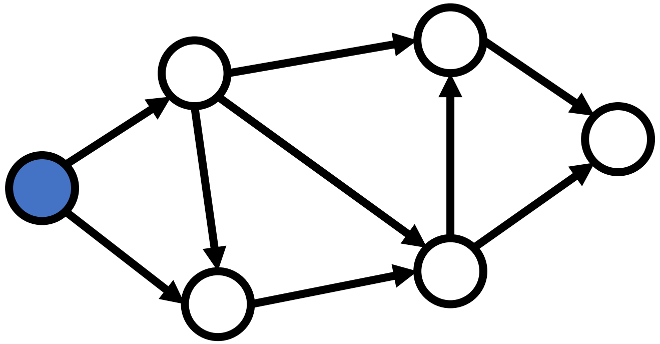
# Dynamic Programming

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

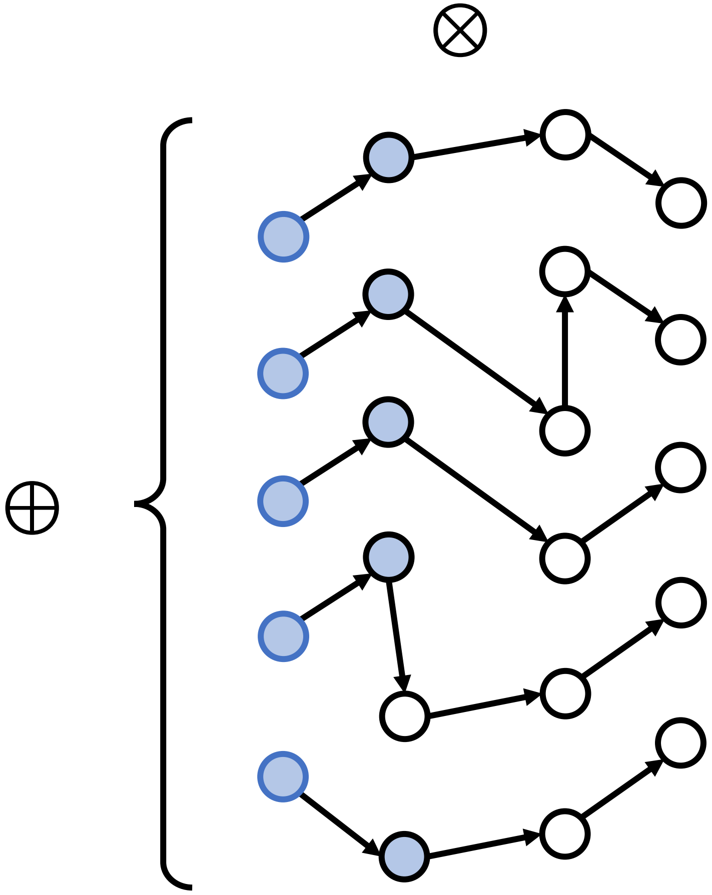


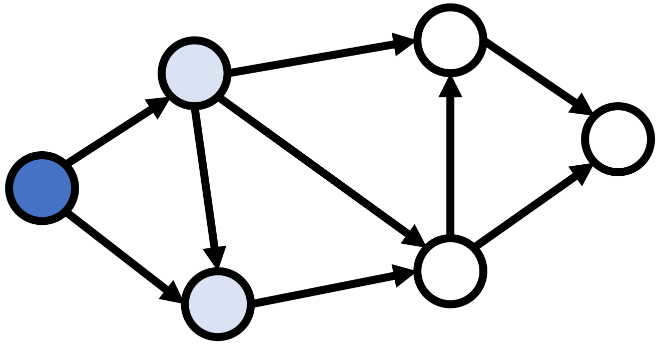
# Dynamic Programming



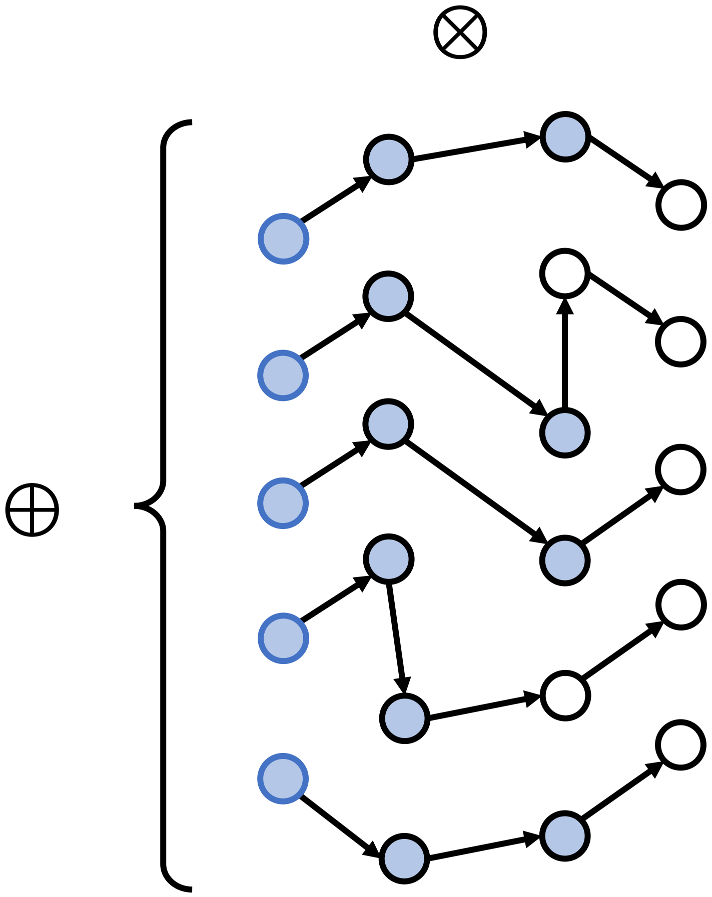



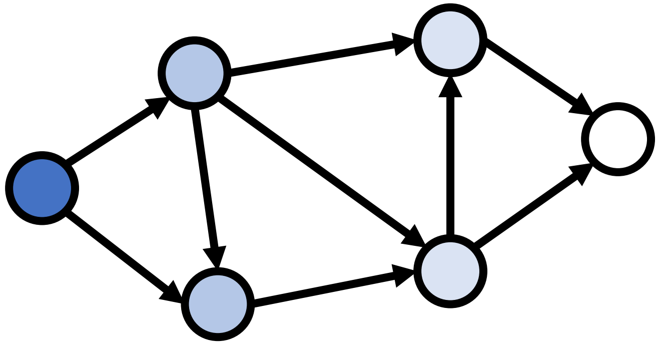
# Dynamic Programming



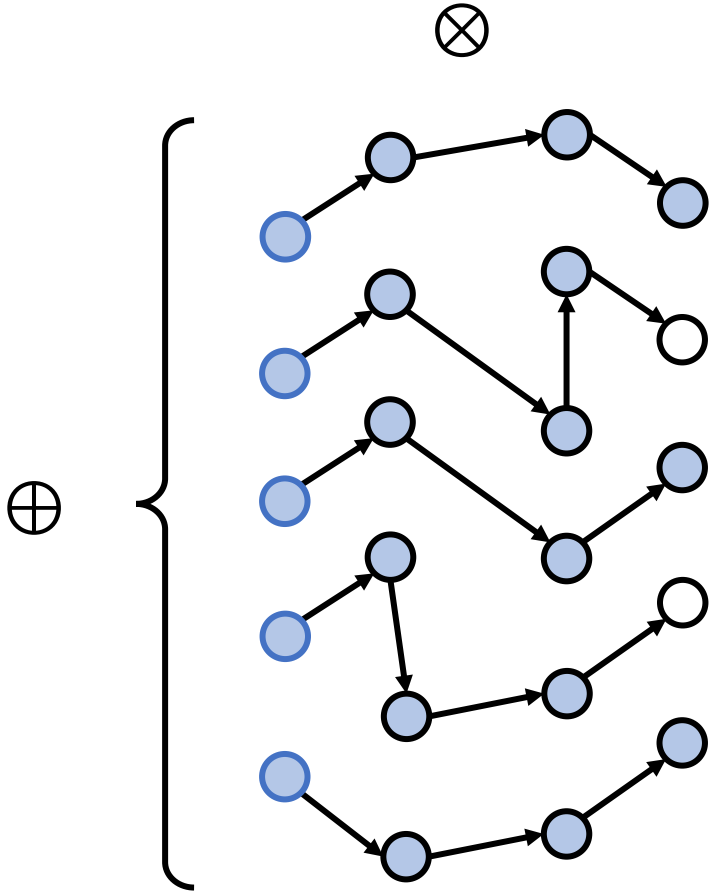



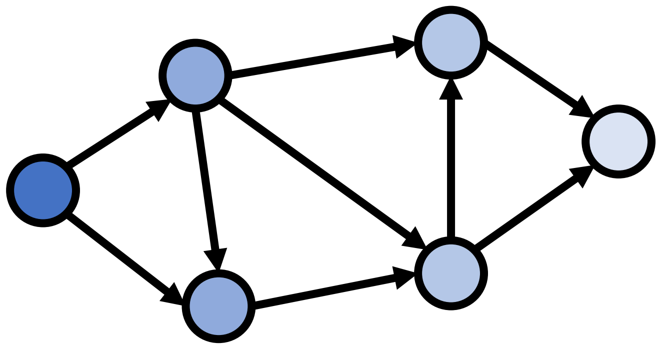
# Dynamic Programming



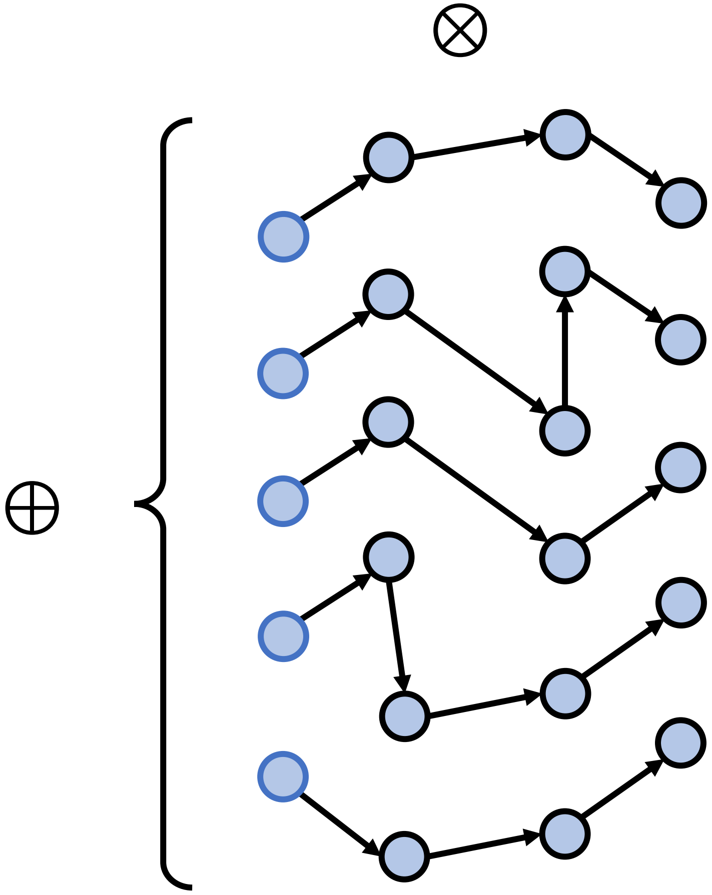



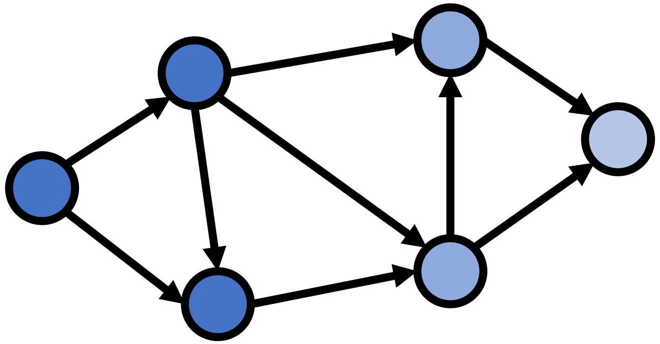
# Dynamic Programming



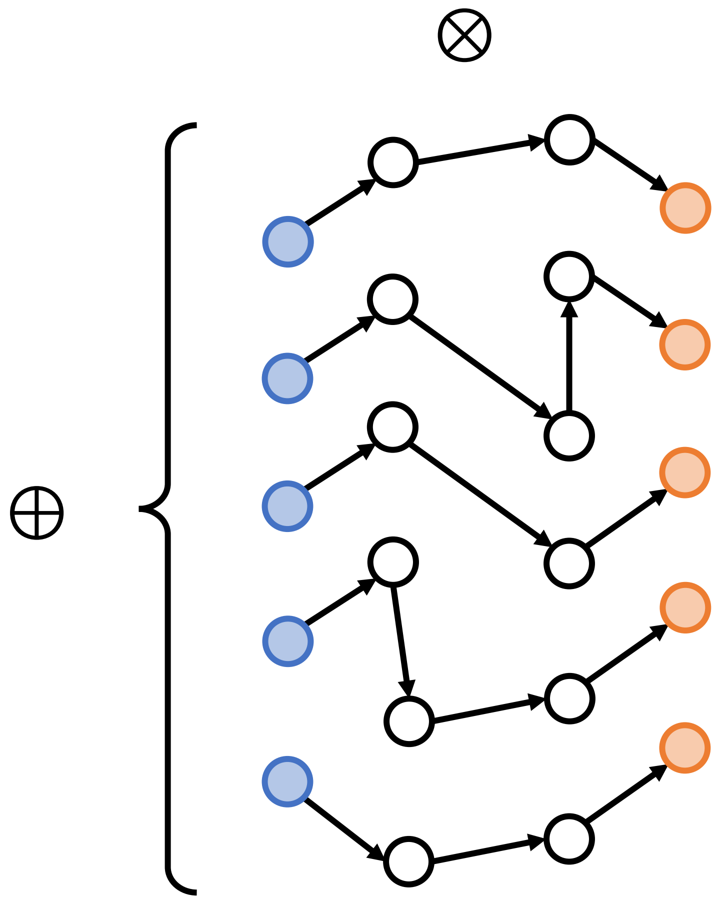



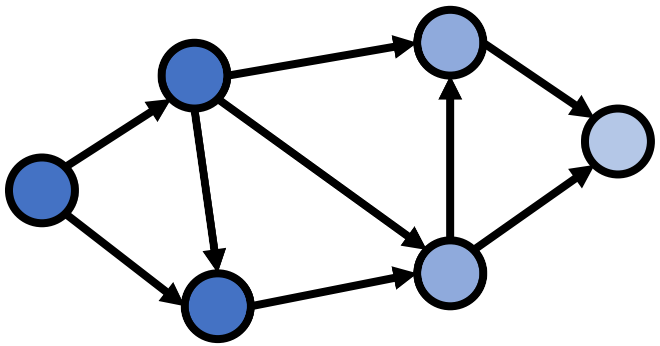
# Dynamic Programming



# Dynamic Programming



power iteration

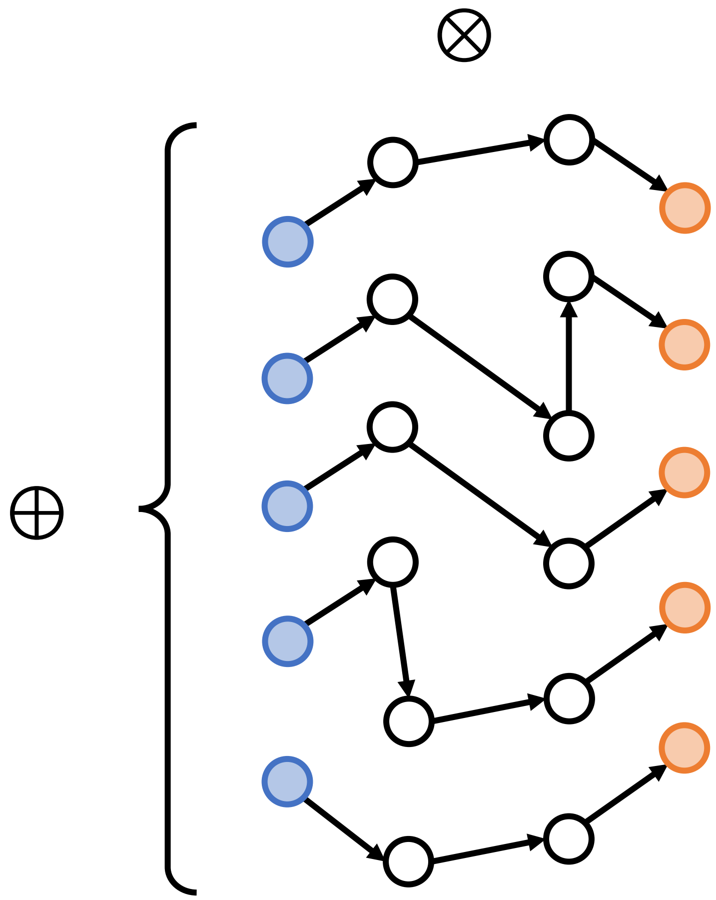
Bellman-Ford algorithm

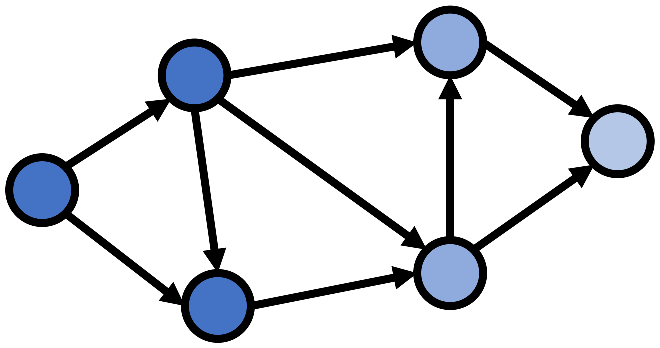
Viterbi algorithm

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

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

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

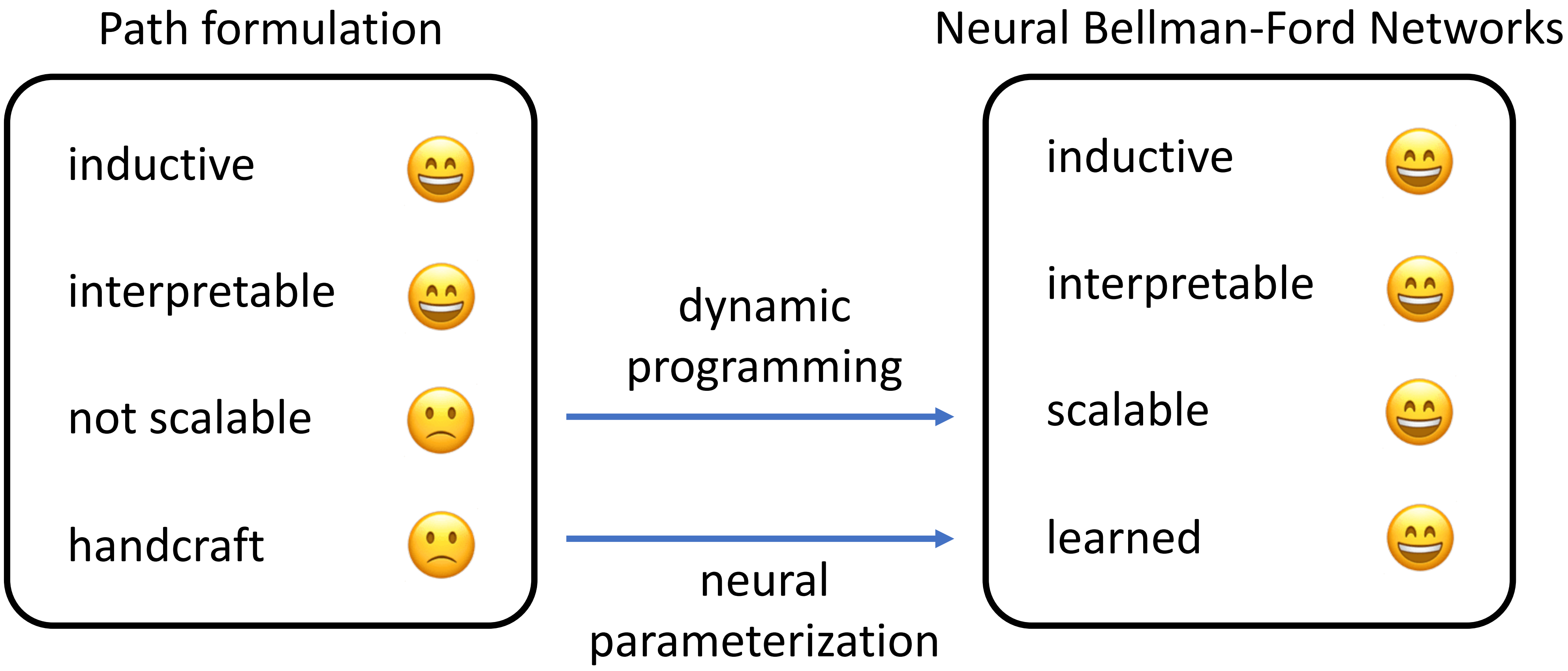
$$\boldsymbol{h}_q^{(0)}(u,v) \leftarrow \begin{cases} ①_q & u = v \\ ⓪_q & u \neq v \end{cases}$$

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

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

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

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

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

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

# Neural Parameterization

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

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

# 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

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

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

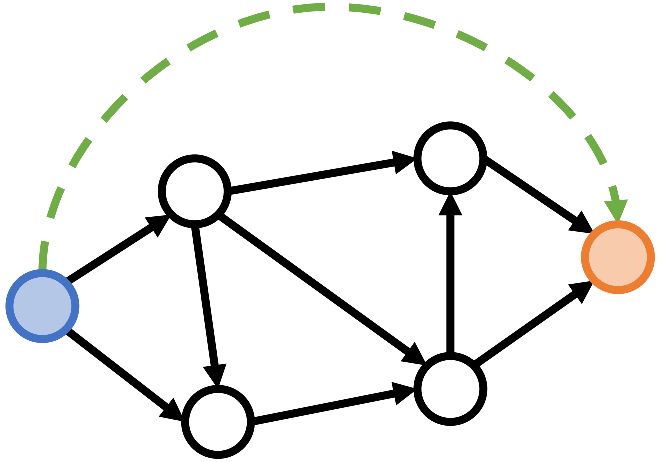
- Path formulation

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

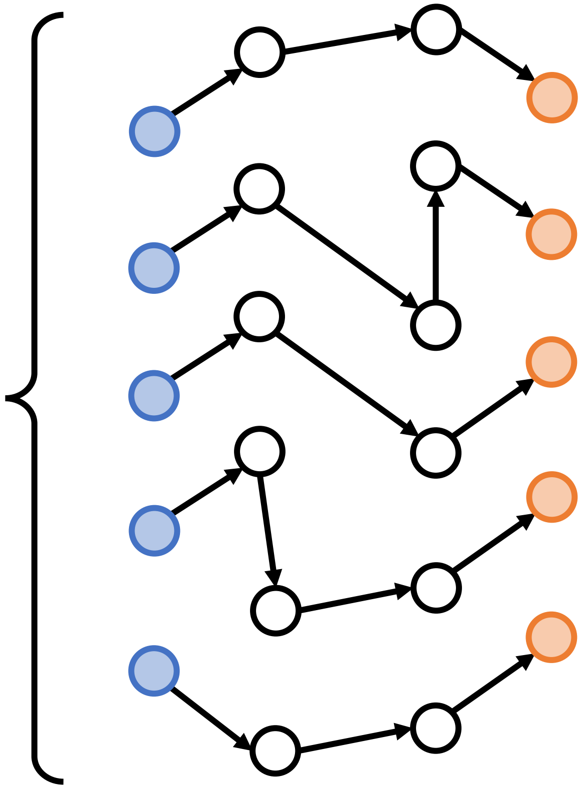
# Revisit Path Formulation

- NBFNet

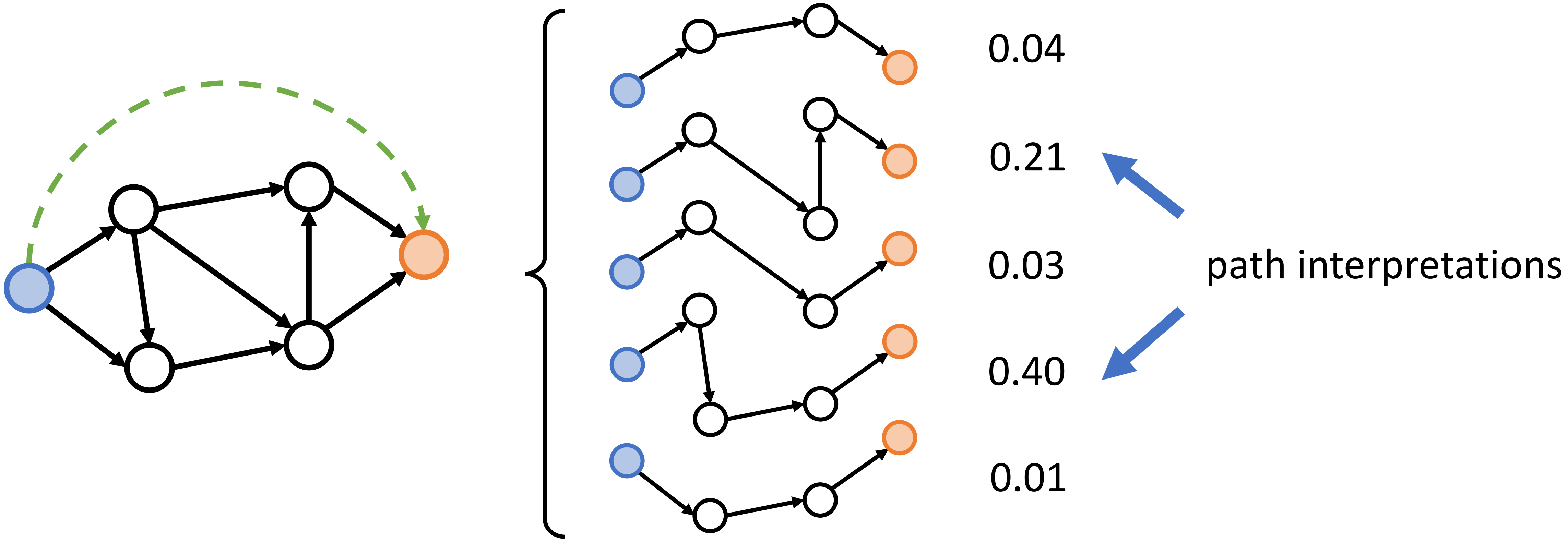
Chain of relational operators



Permutation invariant aggregation



# Interpretation



# 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 [31] | 3521 | 0.174 | 0.119 | 0.186 | 0.285 | 22438 | 0.324 | 0.276 | 0.360 | 0.406 |
| | NeuralLP [64] | - | 0.240 | - | - | 0.362 | - | 0.435 | 0.371 | 0.434 | 0.566 |
| | DRUM [42] | - | 0.343 | 0.255 | 0.378 | 0.516 | - | 0.486 | 0.425 | 0.513 | 0.586 |
| **Embeddings** | TransE [5] | 357 | 0.294 | - | - | 0.465 | 3384 | 0.226 | - | - | 0.501 |
| | DistMult [63] | 254 | 0.241 | 0.155 | 0.263 | 0.419 | 5110 | 0.43 | 0.39 | 0.44 | 0.49 |
| | ComplEx [54] | 339 | 0.247 | 0.158 | 0.275 | 0.428 | 5261 | 0.44 | 0.41 | 0.46 | 0.51 |
| | RotatE [48] | 177 | 0.338 | 0.241 | 0.375 | 0.553 | 3340 | 0.476 | 0.428 | 0.492 | 0.571 |
| **GNNs** | RGCN [44] | 221 | 0.273 | 0.182 | 0.303 | 0.456 | 2719 | 0.402 | 0.345 | 0.437 | 0.494 |
| | GraIL [51] | 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 [26] | 0.834 | 0.889 | 0.768 | 0.810 | 0.757 | 0.856 |
| | Personalized PageRank [38] | 0.845 | 0.899 | 0.762 | 0.814 | 0.763 | 0.860 |
| | SimRank [24] | 0.838 | 0.888 | 0.755 | 0.805 | 0.743 | 0.829 |
| **Embeddings** | DeepWalk [39] | 0.831 | 0.850 | 0.805 | 0.836 | 0.844 | 0.841 |
| | LINE [49] | 0.844 | 0.876 | 0.791 | 0.826 | 0.849 | 0.888 |
| | node2vec [15] | 0.872 | 0.879 | 0.838 | 0.868 | 0.891 | 0.914 |
| **GNNs** | VGAE [28] | 0.914 | 0.926 | 0.908 | 0.920 | 0.944 | 0.947 |
| | S-VGAE [10] | 0.941 | 0.941 | **0.947** | **0.952** | 0.960 | 0.960 |
| | SEAL [68] | 0.933 | 0.942 | 0.905 | 0.924 | 0.978 | 0.979 |
| | NBFNet | **0.956** | **0.962** | 0.923 | 0.936 | **0.983** | **0.982** |

# Results

- Inductive relation prediction

| Class | Method | FB15k-237 | | | | WN18RR | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | v1 | v2 | v3 | v4 | v1 | v2 | v3 | v4 |
| **Path-based** | NeuralLP [14] | 0.529 | 0.589 | 0.529 | 0.559 | 0.744 | 0.689 | 0.462 | 0.671 |
| | DRUM [42] | 0.529 | 0.587 | 0.529 | 0.559 | 0.744 | 0.689 | 0.462 | 0.671 |
| | RuleN [35] | 0.498 | 0.778 | 0.877 | 0.856 | 0.809 | 0.782 | 0.534 | 0.716 |
| **GNNs** | GraIL [51] | 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 on FB15k-237

- **Query:**
- ⟨Florence, vacationer, D.C. Henrie⟩

- **Interpretation:**
- 0.251 ⟨Florence, $\text{contain}^{-1}$, Italy⟩ ∧ ⟨Italy, capital, Rome⟩ ∧ ⟨Rome, vacationer, D.C. Henrie⟩
- 0.183 ⟨Florence, place $\text{live}^{-1}$, G.F. Handel⟩ ∧ ⟨G.F. Handel, live in, Rome⟩ ∧ ⟨Rome, vacationer, D.C. Henrie⟩

# 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)

- Source code: https://github.com/DeepGraphLearning/NBFNet