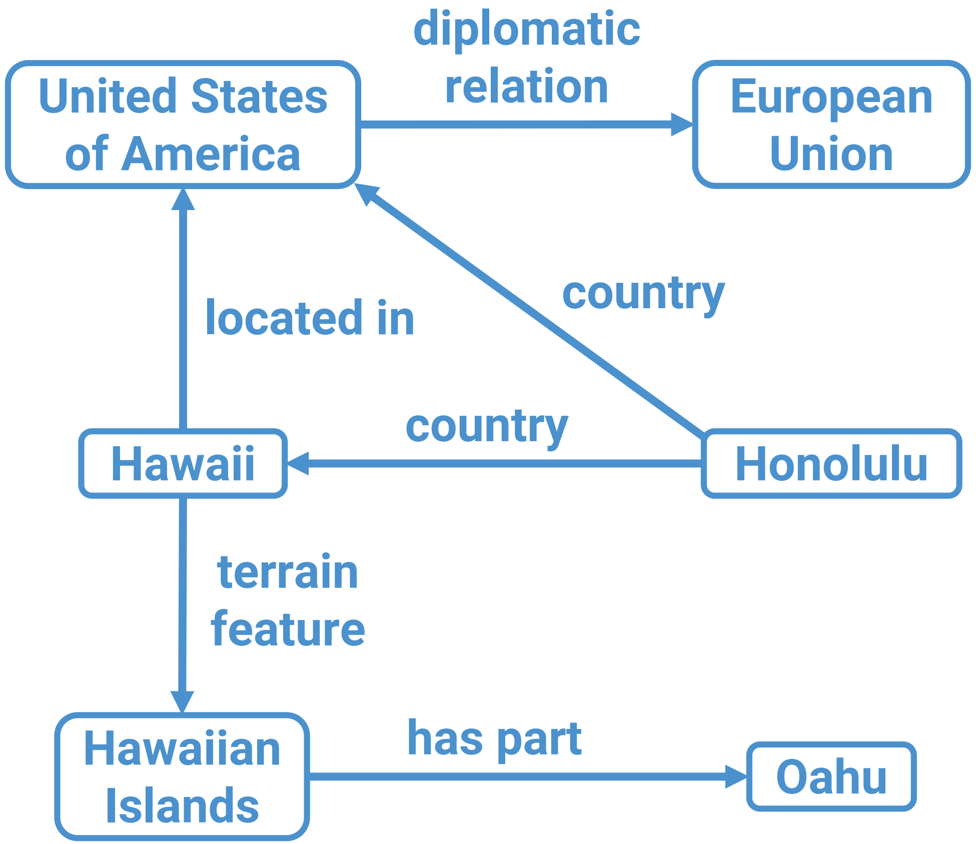
InGram: Inductive Knowledge Graph Embedding via Relation Graphs

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang*
School of Computing, KAIST
* Corresponding Author

The 40th International Conference on Machine Learning (ICML 2023)

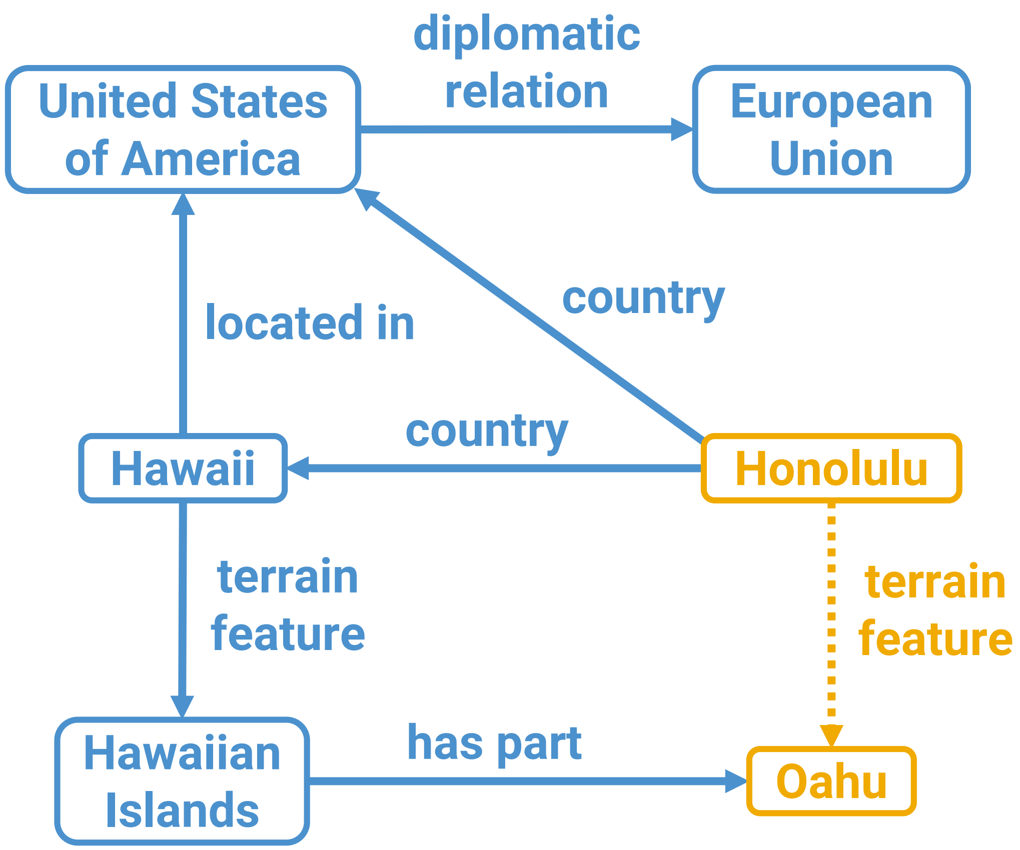
BDILab
BIG DATA INTELLIGENCE

Knowledge Graph



Training Graph

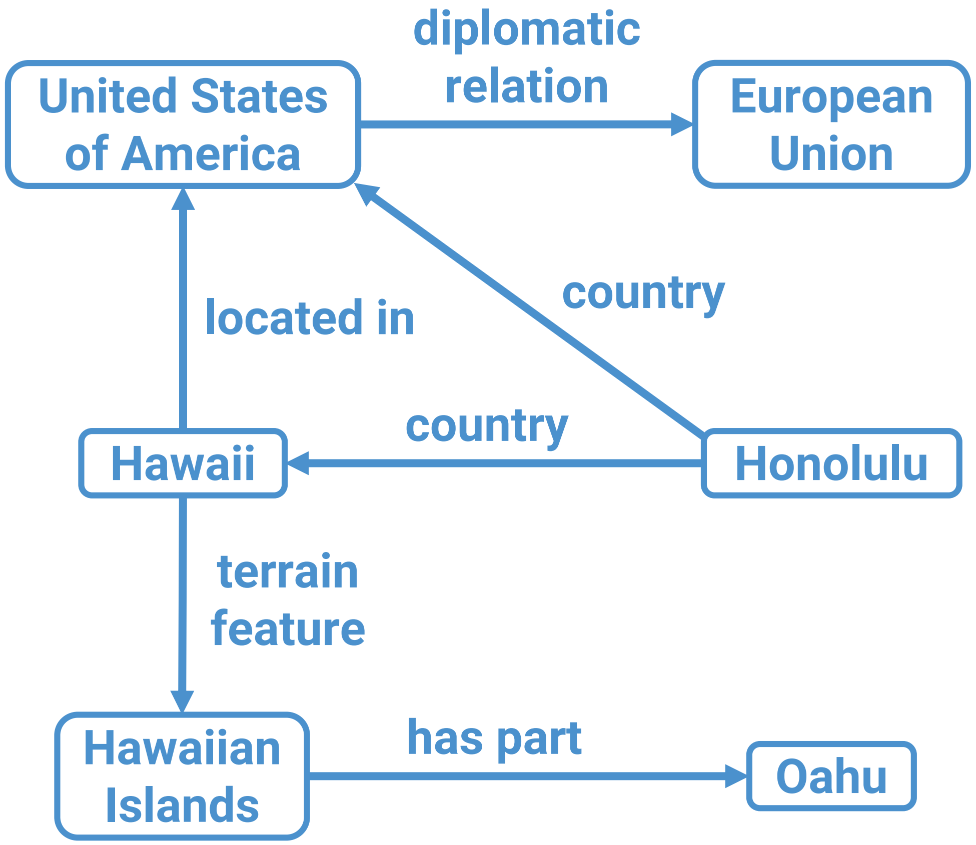
Transductive Knowledge Graph Completion



Training Graph

(Honolulu, terrain feature, ?)

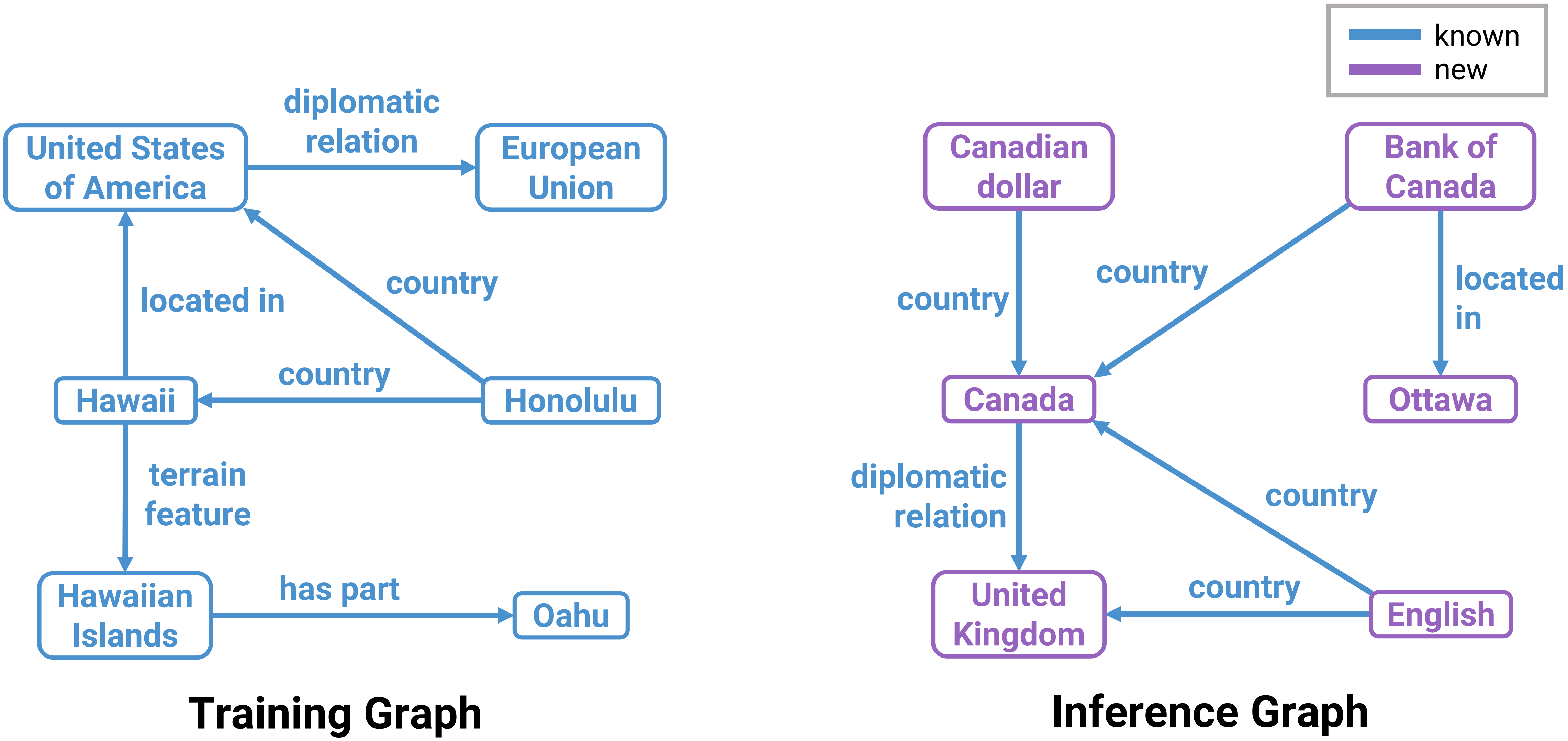
Existing Inductive Knowledge Graph Completion



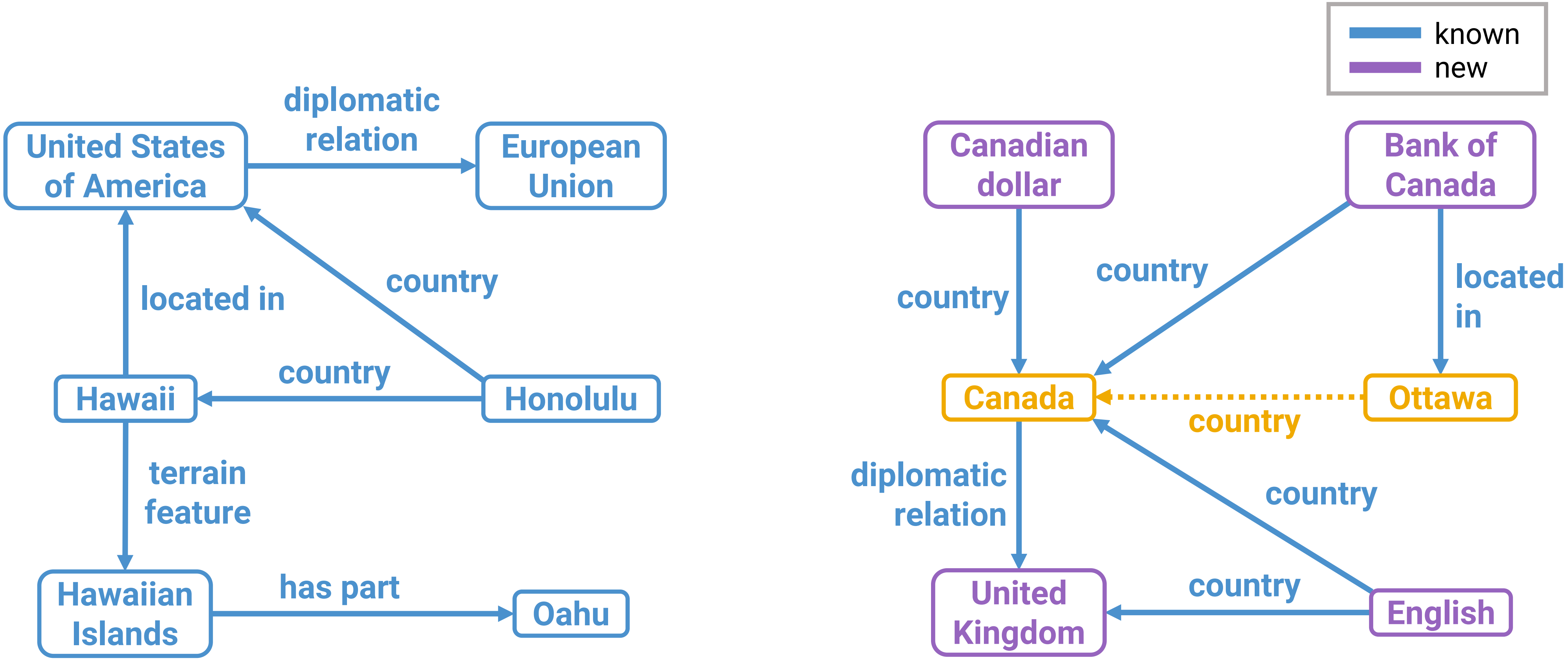
Training Graph

(Ottawa, Country, ?)

Existing Inductive Knowledge Graph Completion



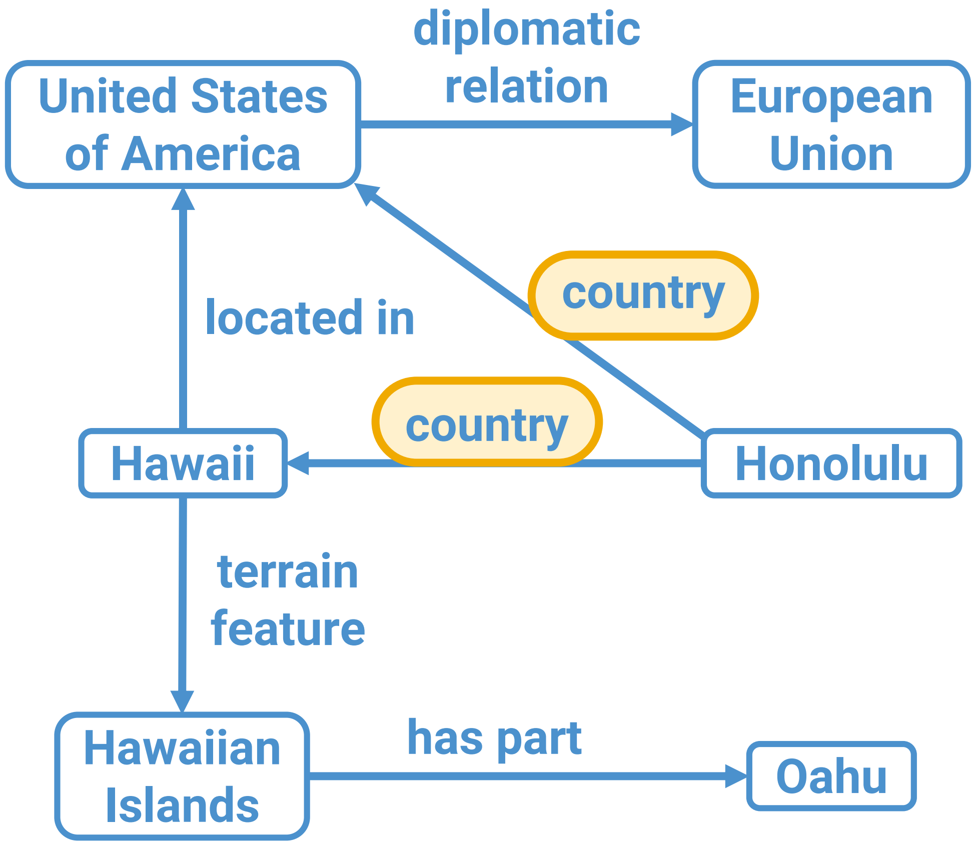
Existing Inductive Knowledge Graph Completion



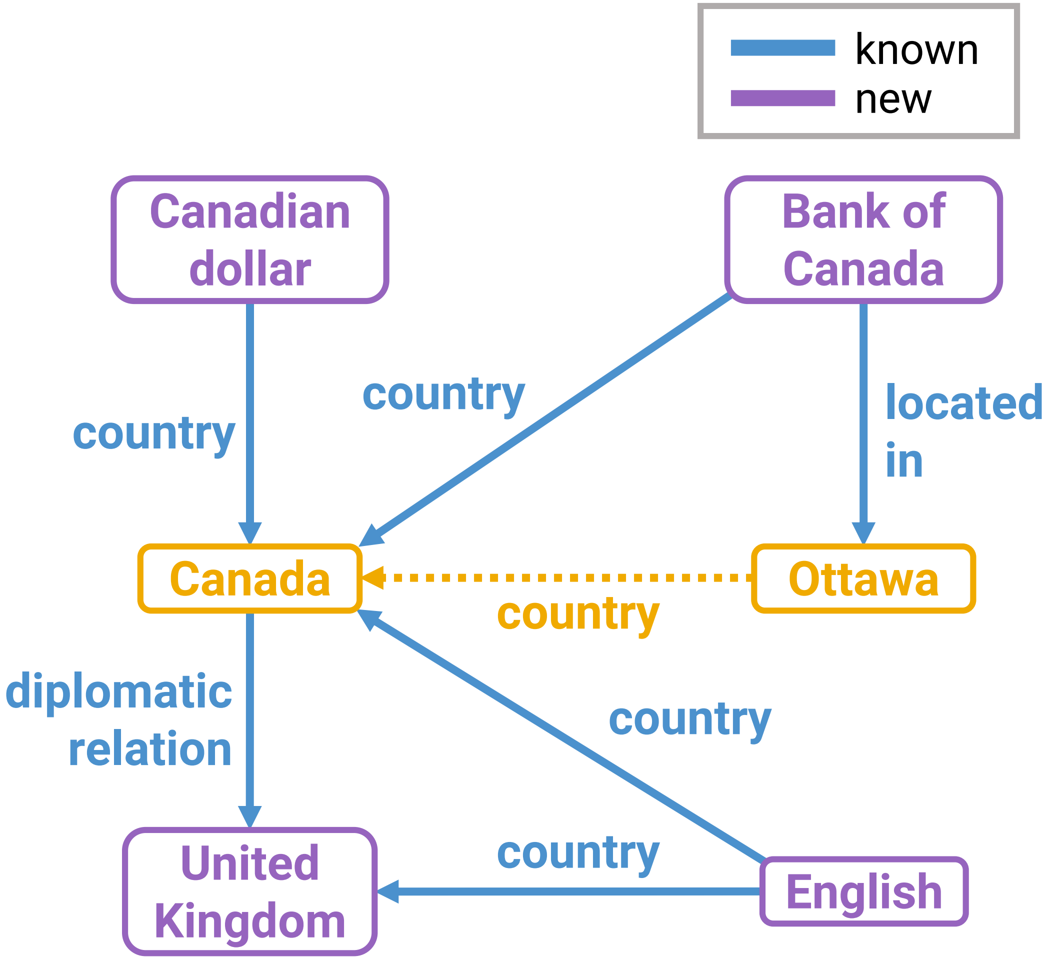
Training Graph

Inference Graph

Transductive Inference for Relations

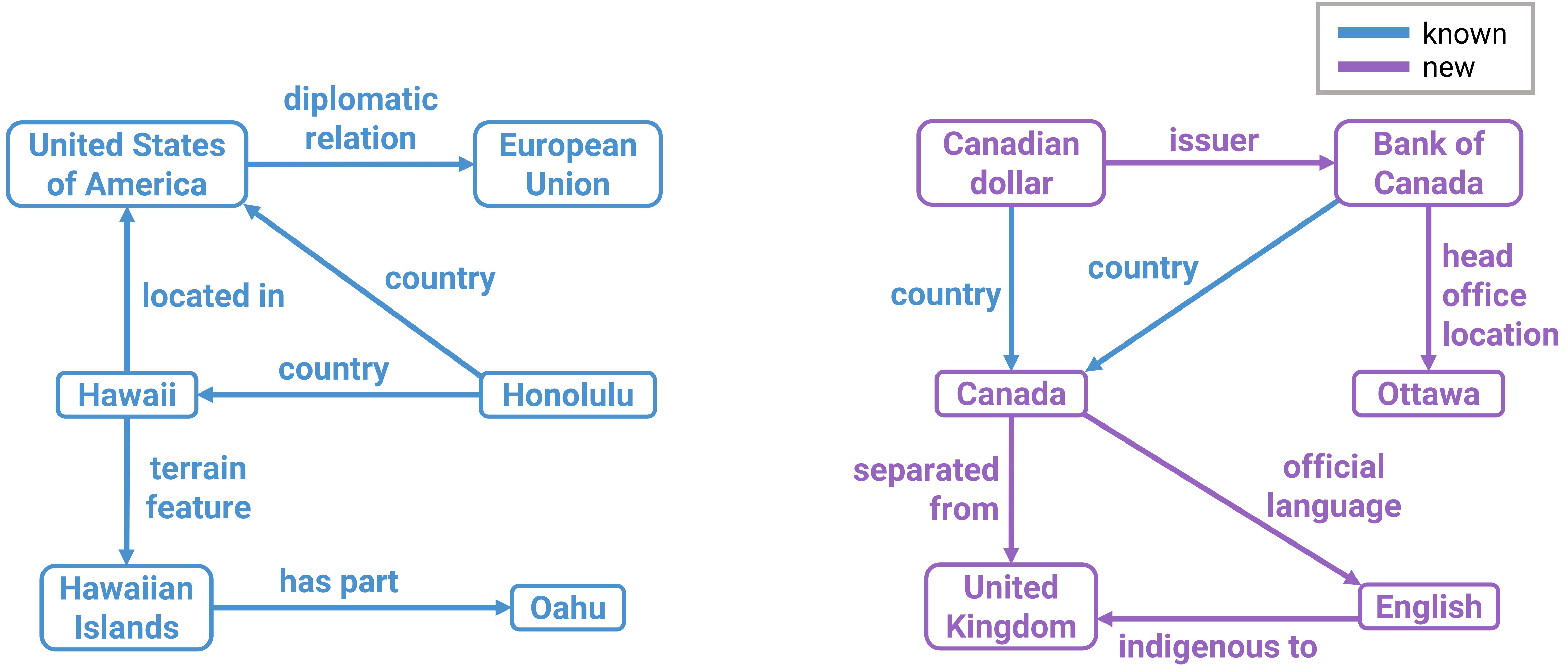


Training Graph



Inference Graph

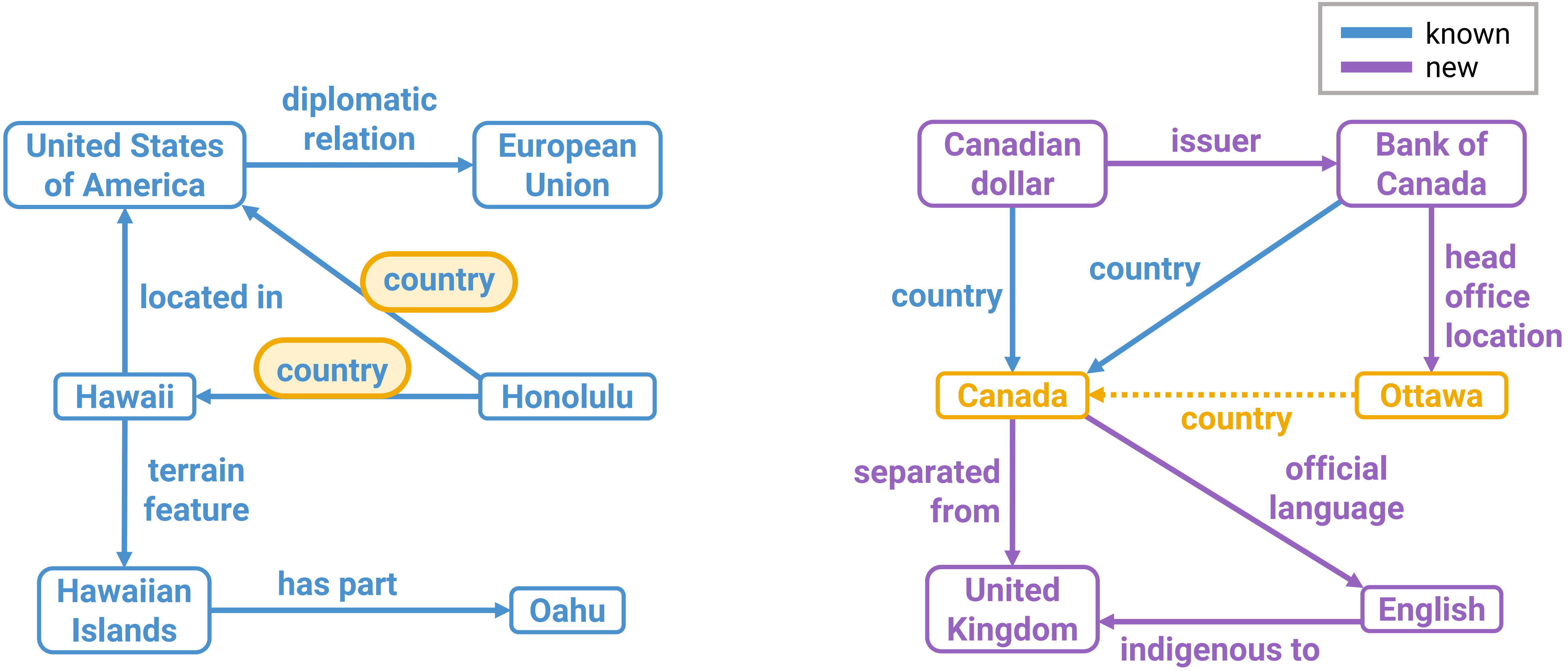
Semi-Inductive Inference for Relations



Training Graph

Inference Graph

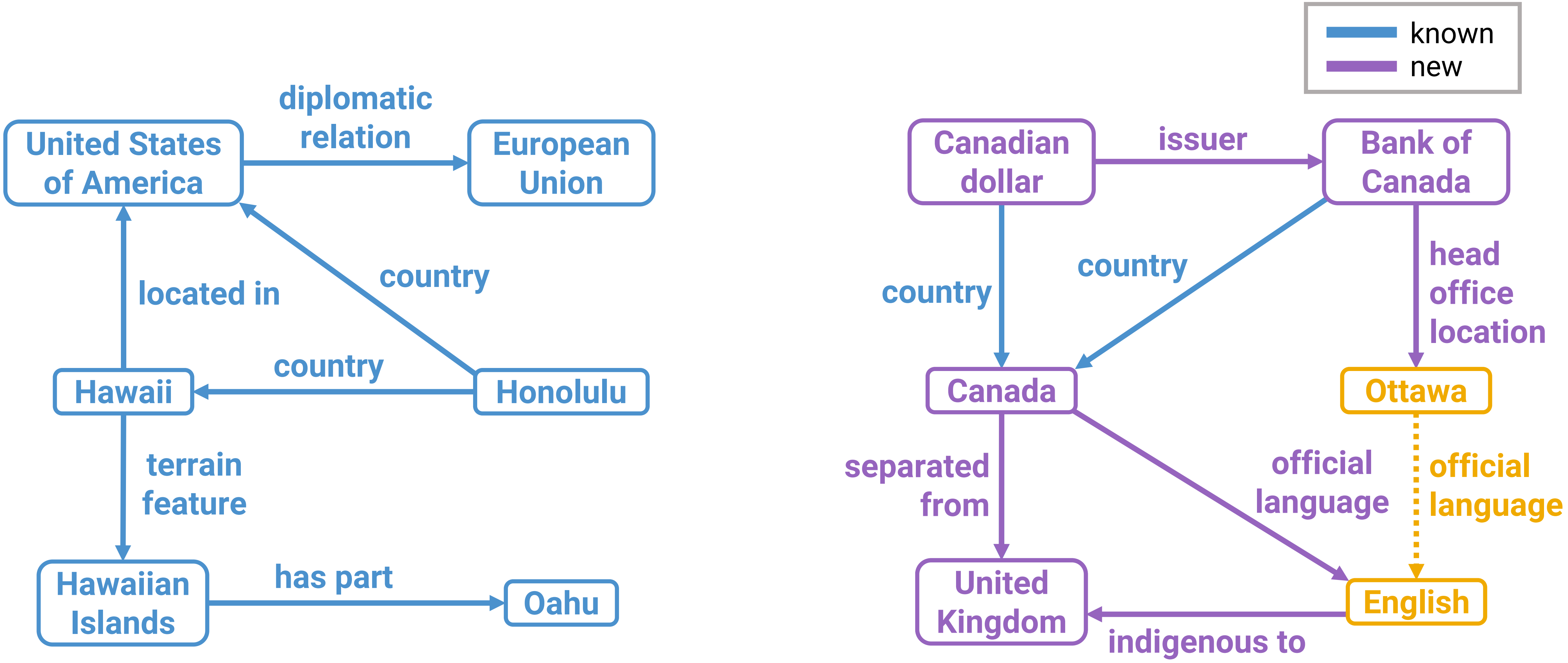
Semi-Inductive Inference for Relations



Training Graph

Inference Graph

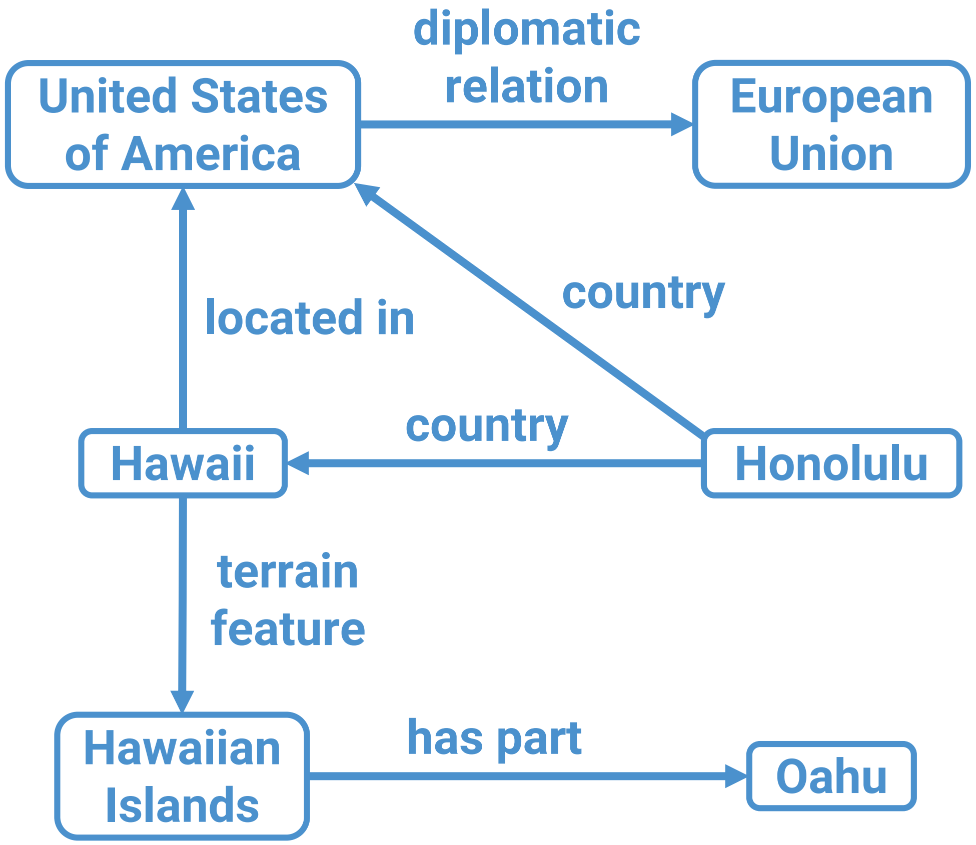
Semi-Inductive Inference for Relations



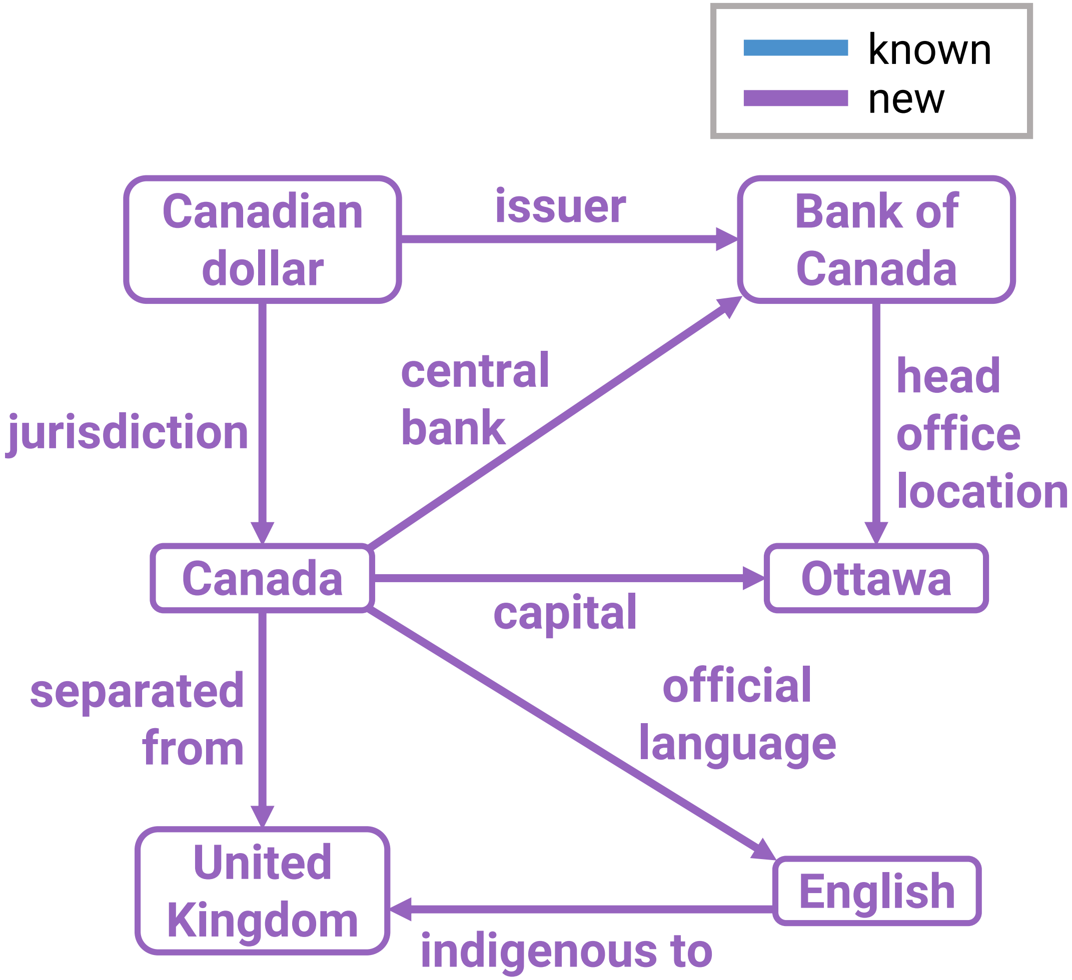
Training Graph

Inference Graph

Inductive Inference for Relations

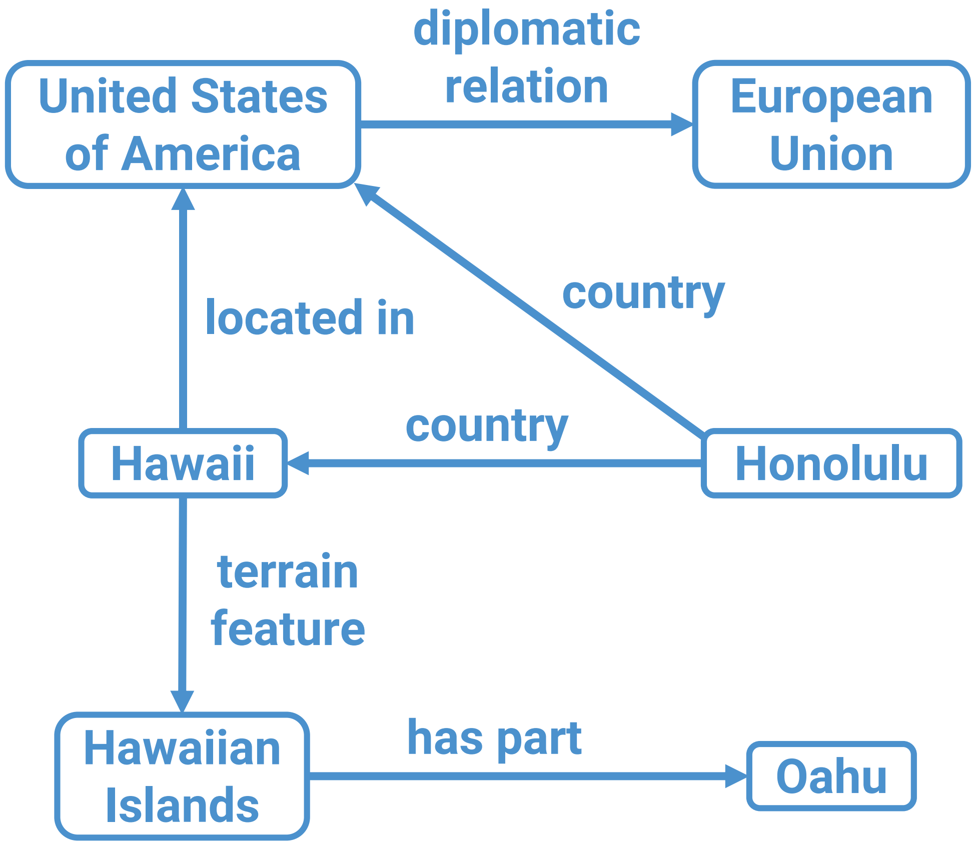


Training Graph

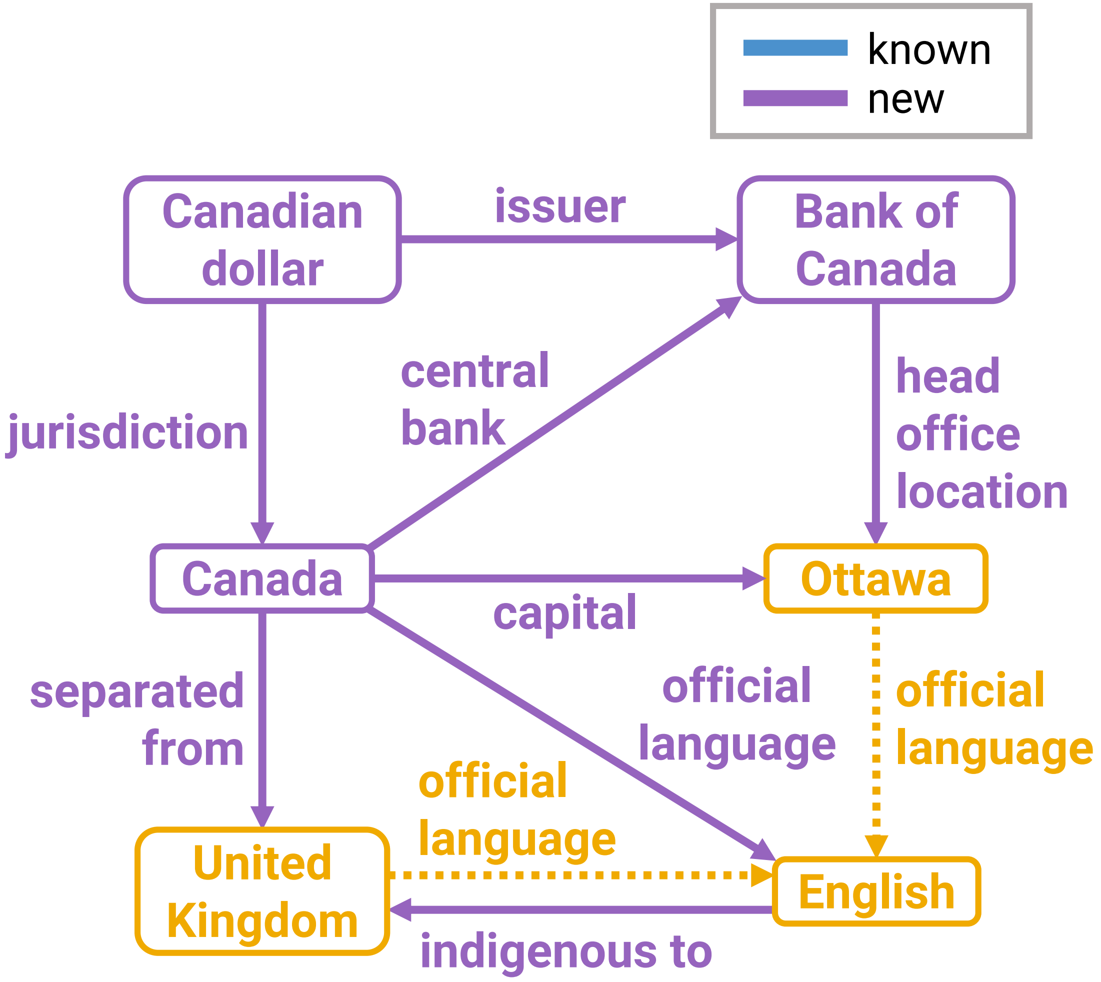


Inference Graph

Inductive Inference for Relations



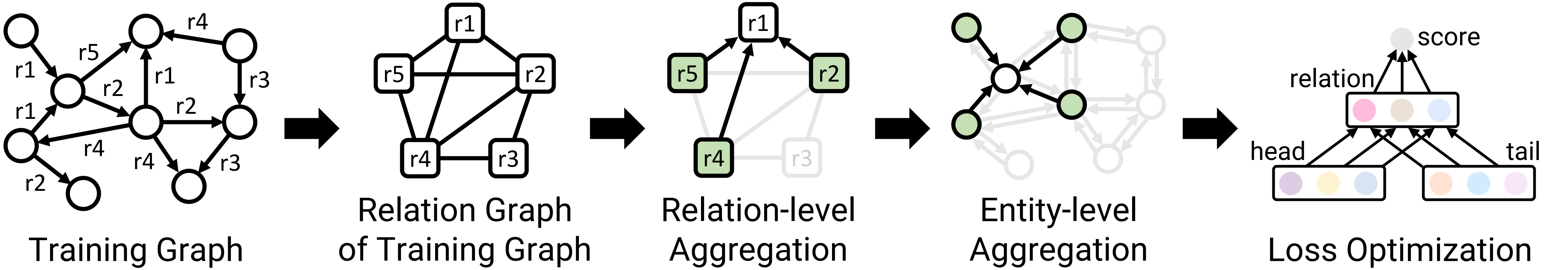
Training Graph



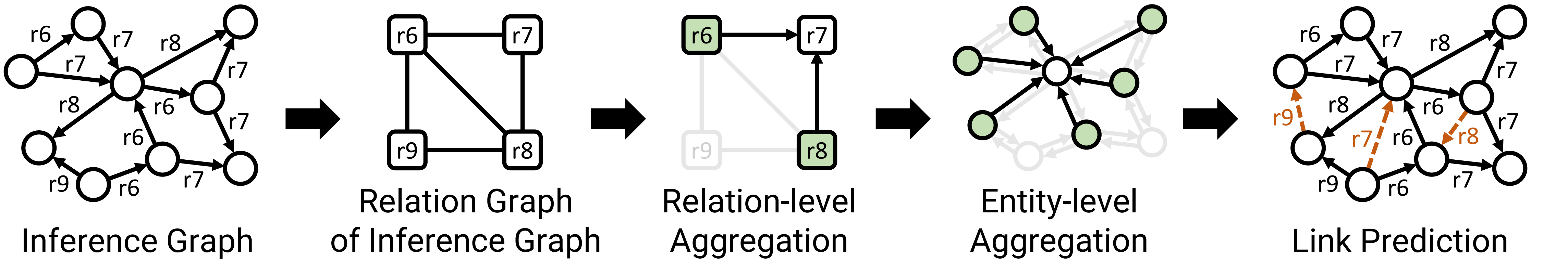
Inference Graph

Overview

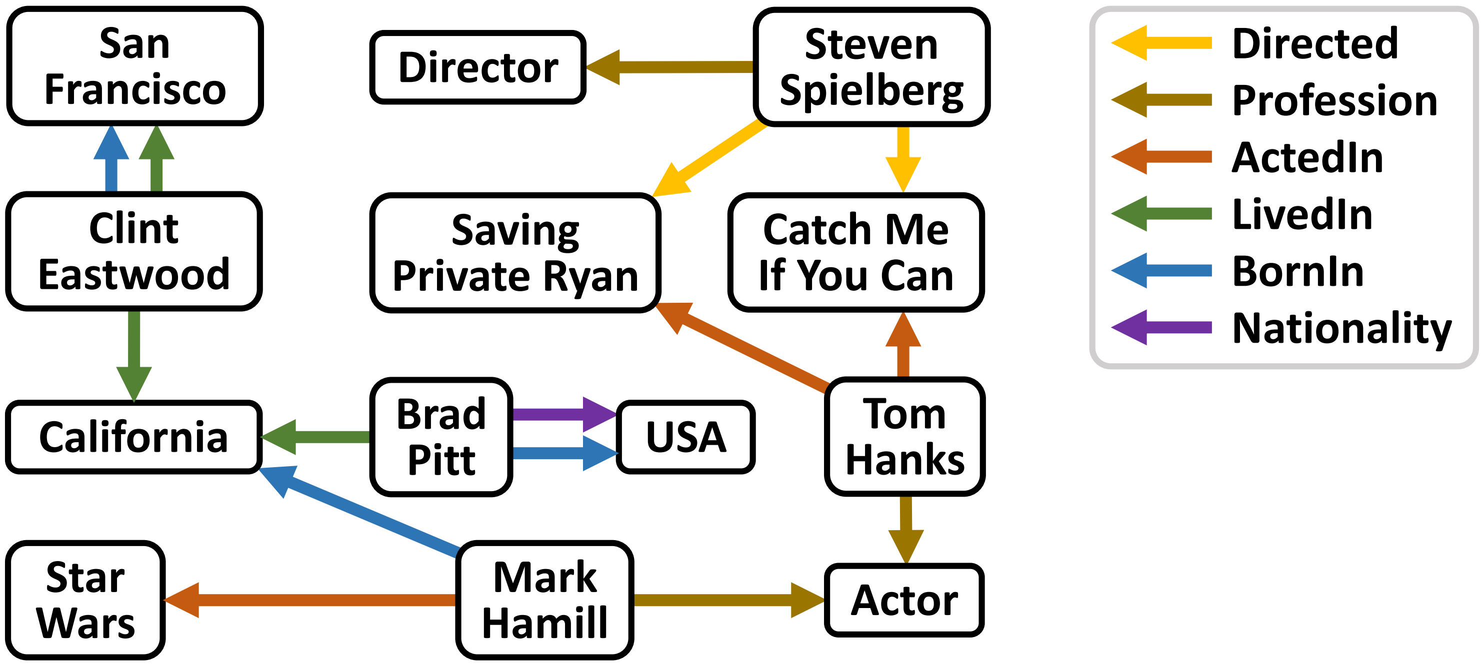
Training Time:



Inference Time:

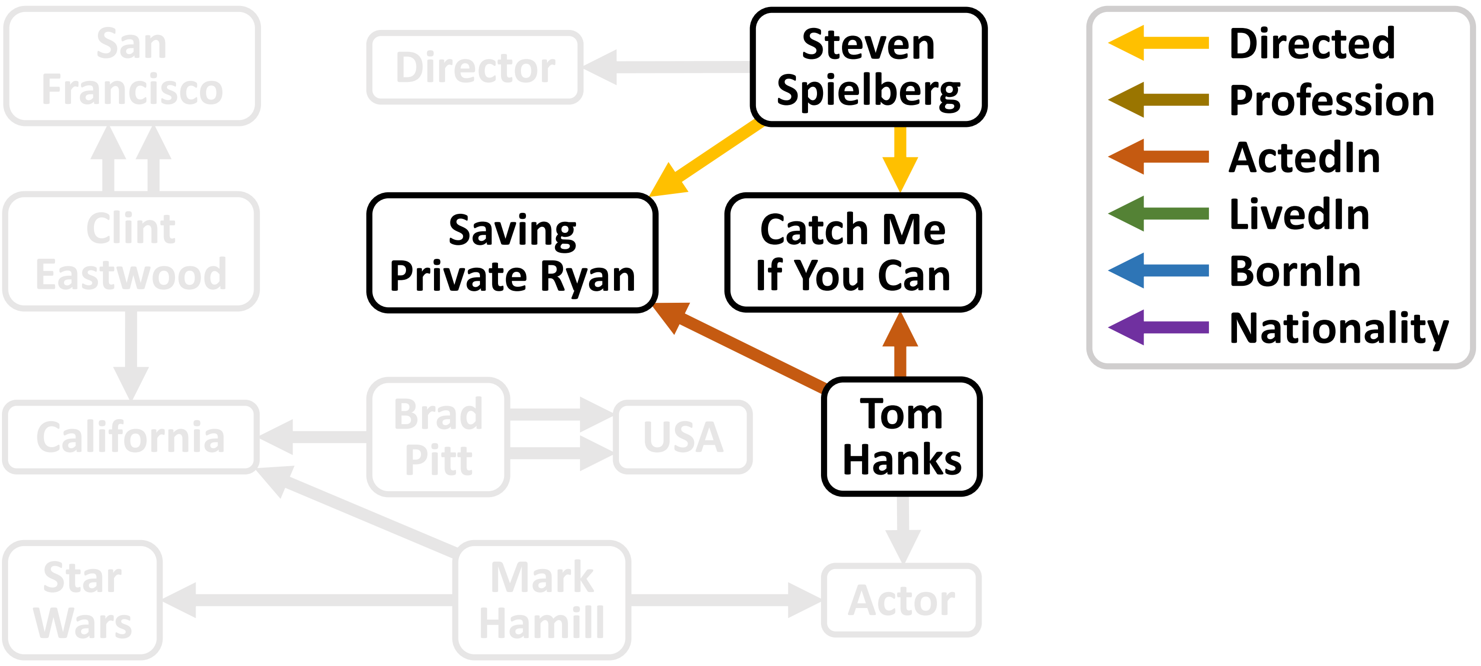


Relation Graph



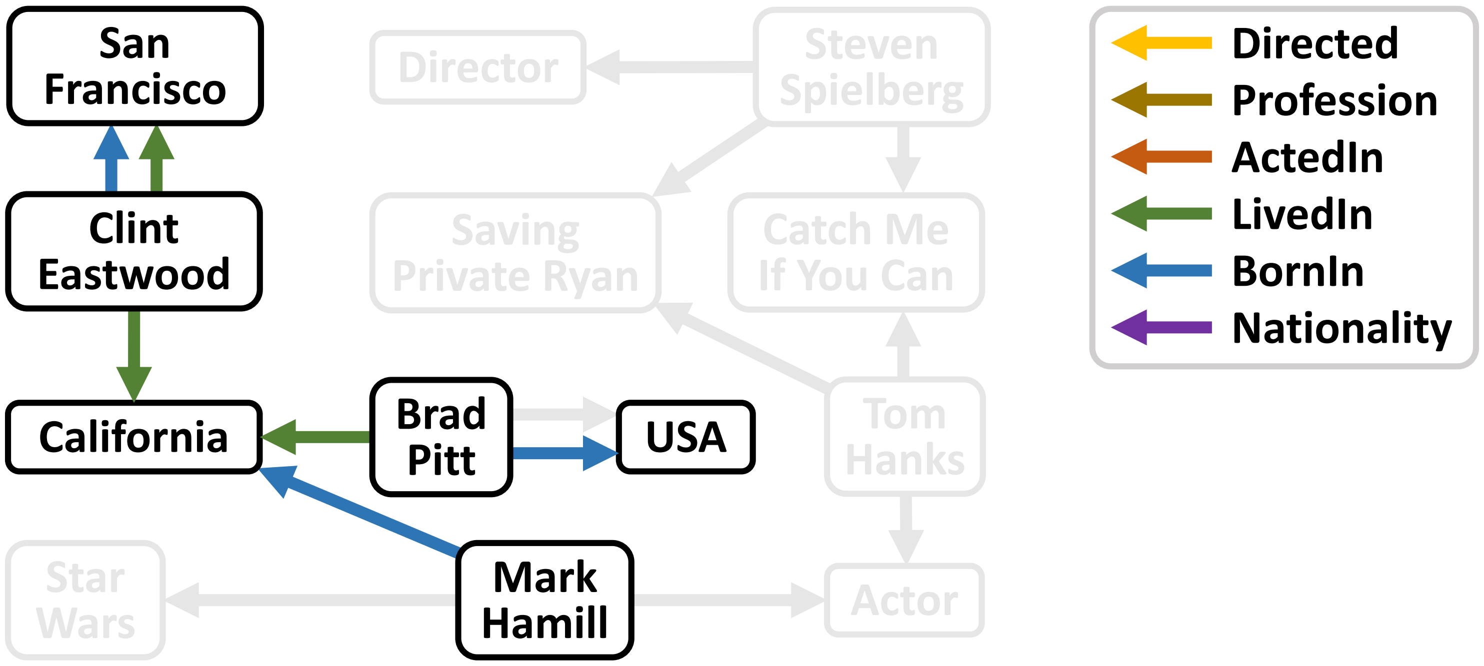
Knowledge Graph

Relation Graph



Knowledge Graph

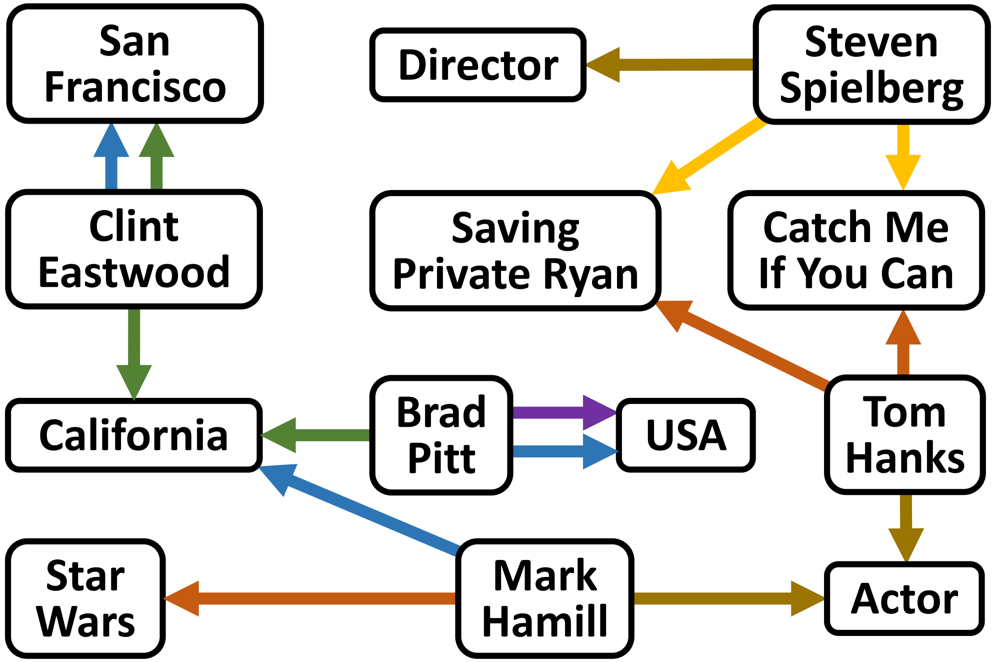
Relation Graph



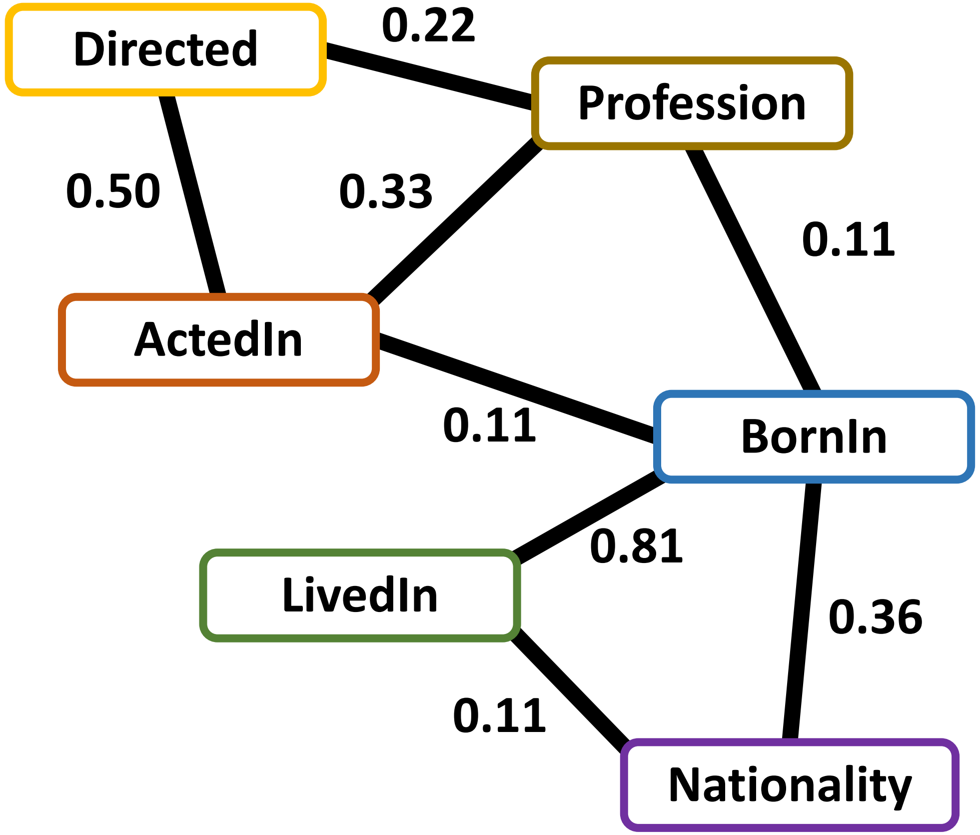
Knowledge Graph

Relation Graph

$$A = E_h^\top D_h^{-2} E_h + E_t^\top D_t^{-2} E_t$$

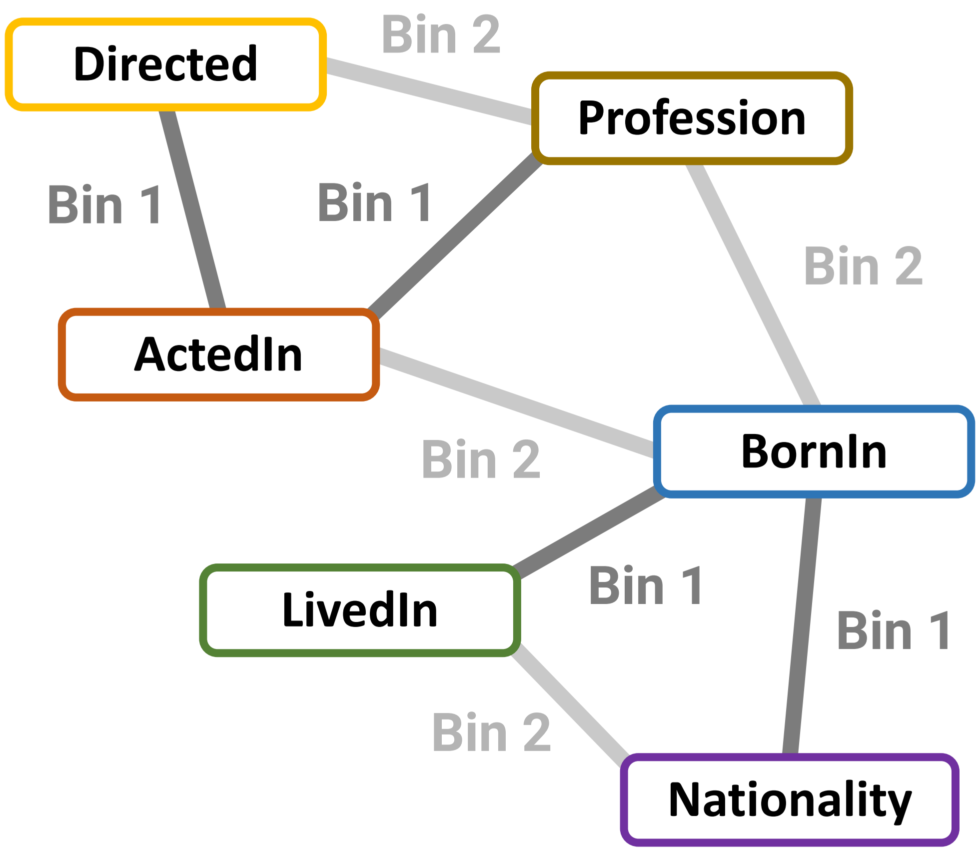


Knowledge Graph



Relation Graph

Relation Graph with Binning



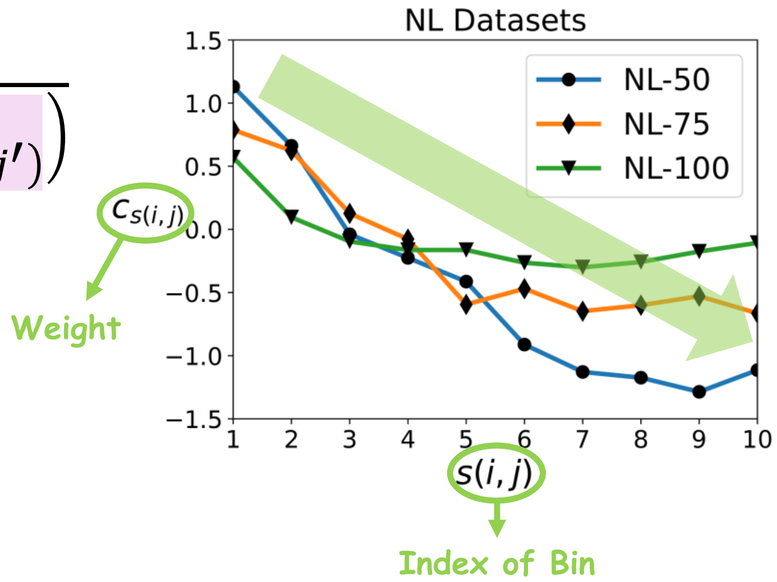
Relation Graph with Binning

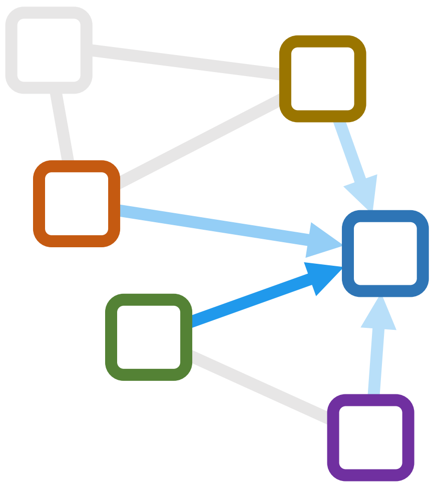
Relation Pairs	Weights	
(BornIn, LivedIn)	0.81	
(ActedIn, Directed)	0.50	Bin 1
(BornIn, Nationality)	0.36	
(ActedIn, Profession)	0.33	
(Directed, Profession)	0.22	
(ActedIn, BornIn)	0.11	Bin 2
(BornIn, Profession)	0.11	
(LivedIn, Nationality)	0.11	

Relation Pairs Sorted by Weights

Relation-level Aggregation

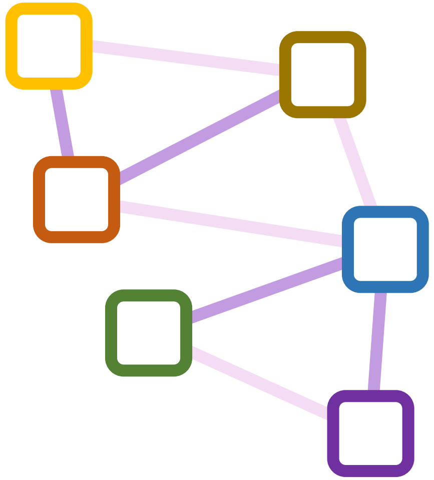
- Aggregate neighboring relations' embedding vectors
  - $\boldsymbol{z}_i^{(l+1)} = \sigma\left(\sum_{r_j \in \mathcal{N}_i} \alpha_{ij}^{(l)} \boldsymbol{W}^{(l)} \boldsymbol{z}_j^{(l)}\right)$
- Consider the **relative importance** and the **affinity weight**
  - $\alpha_{ij}^{(l)} = \dfrac{\exp\left(\psi^{(l)}\left(\left[\boldsymbol{z}_i^{(l)} \| \boldsymbol{z}_j^{(l)}\right]\right) + c_{s(i,j)}^{(l)}\right)}{\sum_{r_{j'} \in \mathcal{N}_i} \exp\left(\psi^{(l)}\left(\left[\boldsymbol{z}_i^{(l)} \| \boldsymbol{z}_{j'}^{(l)}\right]\right) + c_{s(i,j')}^{(l)}\right)}$
  - $\psi^{(l)}(\boldsymbol{x}) = \boldsymbol{y}^{(l)} \sigma\left(\boldsymbol{P}^{(l)} \boldsymbol{x}\right)$





Local structure

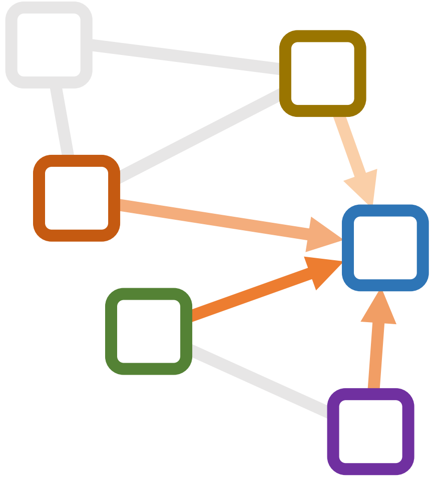
+

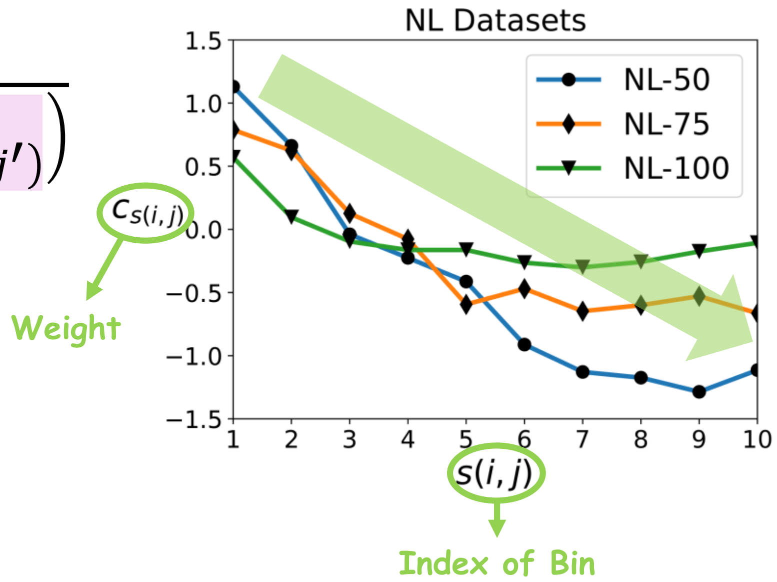


Global level of affinity

Relation-level Aggregation

- Aggregate neighboring relations' embedding vectors
  - $\boldsymbol{z}_i^{(l+1)} = \sigma\left(\sum_{r_j \in \mathcal{N}_i} \alpha_{ij}^{(l)} \boldsymbol{W}^{(l)} \boldsymbol{z}_j^{(l)}\right)$
- Consider the **relative importance** and the **affinity weight**
  - $\alpha_{ij}^{(l)} = \dfrac{\exp\left(\psi^{(l)}\left(\left[\boldsymbol{z}_i^{(l)} \| \boldsymbol{z}_j^{(l)}\right]\right) + c_{s(i,j)}^{(l)}\right)}{\sum_{r_{j'} \in \mathcal{N}_i} \exp\left(\psi^{(l)}\left(\left[\boldsymbol{z}_i^{(l)} \| \boldsymbol{z}_{j'}^{(l)}\right]\right) + c_{s(i,j')}^{(l)}\right)}$
  - $\psi^{(l)}(\boldsymbol{x}) = \boldsymbol{y}^{(l)} \sigma\left(\boldsymbol{P}^{(l)} \boldsymbol{x}\right)$

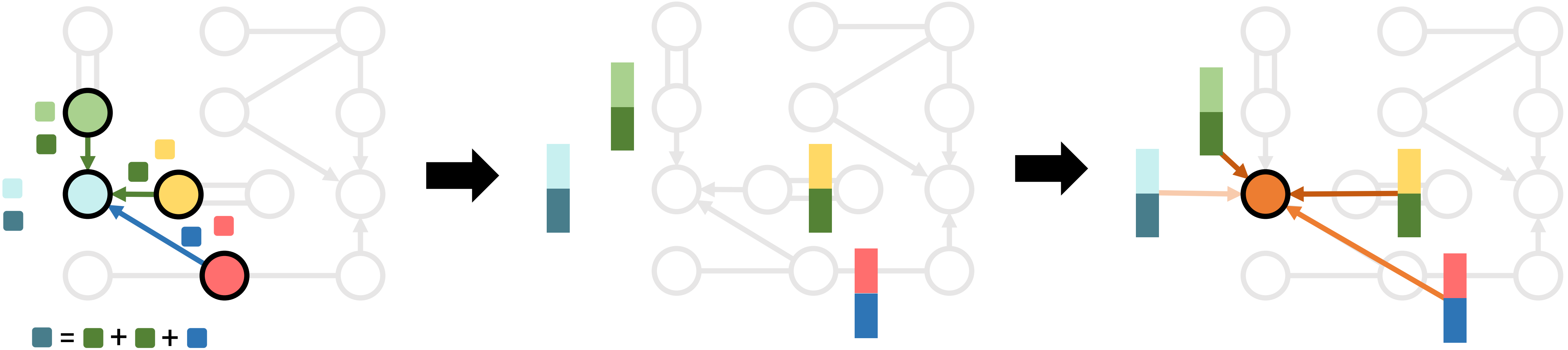




Entity-level Aggregation

- Compute an entity embedding by considering its **own vector**, its **neighbors' embeddings**, and its **adjacent relations**

$$\boldsymbol{h}_i^{(l+1)} = \sigma\left(\beta_{ii}^{(l)}\widehat{\boldsymbol{W}}^{(l)}\left[\boldsymbol{h}_i^{(l)} \| \overline{\boldsymbol{z}}_i^{(L)}\right] + \sum_{v_j \in \widehat{\mathcal{N}}_i} \sum_{r_k \in \mathcal{R}_{ji}} \beta_{ijk}^{(l)}\widehat{\boldsymbol{W}}^{(l)}\left[\boldsymbol{h}_j^{(l)} \| \boldsymbol{z}_k^{(L)}\right]\right)$$



Entity-level Aggregation

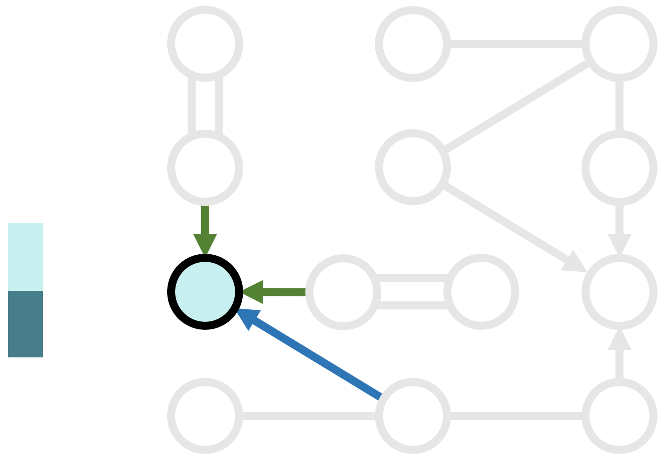
- Consider the **entity itself** and its **adjacent relations**

$$\boldsymbol{h}_i^{(l+1)} = \sigma\left(\beta_{ii}^{(l)}\widehat{\boldsymbol{W}}^{(l)}\left[\boldsymbol{h}_i^{(l)} \| \bar{\boldsymbol{z}}_i^{(L)}\right] + \sum_{v_j \in \widehat{\mathcal{N}}_i} \sum_{r_k \in \mathcal{R}_{ji}} \beta_{ijk}^{(l)}\widehat{\boldsymbol{W}}^{(l)}\left[\boldsymbol{h}_j^{(l)} \| \boldsymbol{z}_k^{(L)}\right]\right)$$

$$\bar{\boldsymbol{z}}_i^{(L)} = \sum_{v_j \in \widehat{\mathcal{N}}_i} \sum_{r_k \in \mathcal{R}_{ji}} \frac{\boldsymbol{z}_k^{(L)}}{\sum_{v_{j'} \in \widehat{\mathcal{N}}_i} |\mathcal{R}_{j'i}|}$$

$$\beta_{ii}^{(l)} = \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_i^{(l)} \| \bar{\boldsymbol{z}}_i^{(L)}\right]\right)\right)/\lambda$$

$$\beta_{ijk}^{(l)} = \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_j^{(l)} \| \boldsymbol{z}_k^{(L)}\right]\right)\right)/\lambda$$



$$\hat{\psi}^{(l)}(\boldsymbol{x}) = \hat{\boldsymbol{y}}^{(l)}\sigma\left(\widehat{\boldsymbol{P}}^{(l)}\boldsymbol{x}\right)$$

$$\lambda = \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_i^{(l)} \| \bar{\boldsymbol{z}}_i^{(L)}\right]\right)\right) + \sum_{v_{j'} \in \widehat{\mathcal{N}}_i} \sum_{r_{k'} \in \mathcal{R}_{j'i}} \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_{j'}^{(l)} \| \boldsymbol{z}_{k'}^{(L)}\right]\right)\right)$$

Entity-level Aggregation

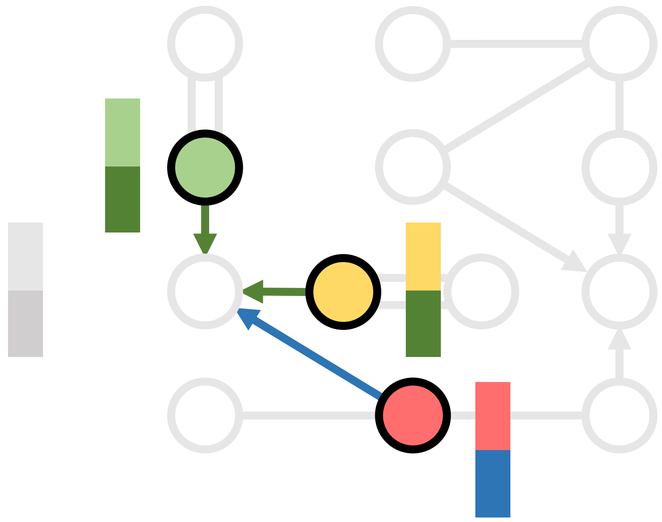
- Consider **neighbors' embedding vectors** and their adjacent **relations**

$$\boldsymbol{h}_i^{(l+1)} = \sigma\left(\beta_{ii}^{(l)}\widehat{\boldsymbol{W}}^{(l)}\left[\boldsymbol{h}_i^{(l)} \| \overline{\boldsymbol{z}}_i^{(L)}\right] + \sum_{v_j \in \widehat{\mathcal{N}}_i} \sum_{r_k \in \mathcal{R}_{ji}} \beta_{ijk}^{(l)} \widehat{\boldsymbol{W}}^{(l)}\left[\boldsymbol{h}_j^{(l)} \| \boldsymbol{z}_k^{(L)}\right]\right)$$

$$\overline{\boldsymbol{z}}_i^{(L)} = \sum_{v_j \in \widehat{\mathcal{N}}_i} \sum_{r_k \in \mathcal{R}_{ji}} \frac{\boldsymbol{z}_k^{(L)}}{\sum_{v_{j'} \in \widehat{\mathcal{N}}_i} |\mathcal{R}_{j'i}|}$$

$$\beta_{ii}^{(l)} = \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_i^{(l)} \| \overline{\boldsymbol{z}}_i^{(L)}\right]\right)\right) / \lambda$$

$$\beta_{ijk}^{(l)} = \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_j^{(l)} \| \boldsymbol{z}_k^{(L)}\right]\right)\right) / \lambda$$



$$\hat{\psi}^{(l)}(\boldsymbol{x}) = \hat{\boldsymbol{y}}^{(l)} \sigma\left(\widehat{\boldsymbol{P}}^{(l)} \boldsymbol{x}\right)$$

$$\lambda = \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_i^{(l)} \| \overline{\boldsymbol{z}}_i^{(L)}\right]\right)\right) + \sum_{v_{j'} \in \widehat{\mathcal{N}}_i} \sum_{r_{k'} \in \mathcal{R}_{j'i}} \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_{j'}^{(l)} \| \boldsymbol{z}_{k'}^{(L)}\right]\right)\right)$$

Entity-level Aggregation

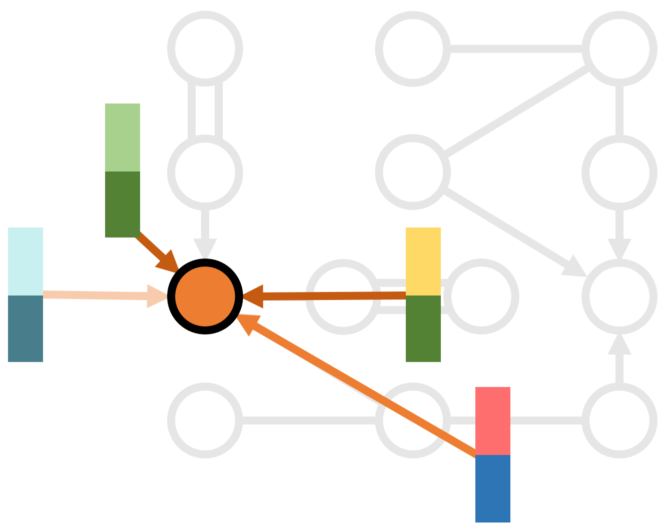
- Consider **the entity itself**, **its neighbors**, and the **relations**

$$\boldsymbol{h}_i^{(l+1)} = \sigma\left(\beta_{ii}^{(l)}\widehat{\boldsymbol{W}}^{(l)}\left[\boldsymbol{h}_i^{(l)} \| \bar{\boldsymbol{z}}_i^{(L)}\right] + \sum_{v_j \in \widehat{\mathcal{N}}_i} \sum_{r_k \in \mathcal{R}_{ji}} \beta_{ijk}^{(l)} \widehat{\boldsymbol{W}}^{(l)}\left[\boldsymbol{h}_j^{(l)} \| \boldsymbol{z}_k^{(L)}\right]\right)$$

$$\bar{\boldsymbol{z}}_i^{(L)} = \sum_{v_j \in \widehat{\mathcal{N}}_i} \sum_{r_k \in \mathcal{R}_{ji}} \frac{\boldsymbol{z}_k^{(L)}}{\sum_{v_{j'} \in \widehat{\mathcal{N}}_i} |\mathcal{R}_{j'i}|}$$

$$\beta_{ii}^{(l)} = \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_i^{(l)} \| \bar{\boldsymbol{z}}_i^{(L)}\right]\right)\right) / \lambda$$

$$\beta_{ijk}^{(l)} = \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_j^{(l)} \| \boldsymbol{z}_k^{(L)}\right]\right)\right) / \lambda$$



$$\hat{\psi}^{(l)}(\boldsymbol{x}) = \hat{\boldsymbol{y}}^{(l)} \sigma\left(\widehat{\boldsymbol{P}}^{(l)} \boldsymbol{x}\right)$$

$$\lambda = \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_i^{(l)} \| \bar{\boldsymbol{z}}_i^{(L)}\right]\right)\right) + \sum_{v_{j'} \in \widehat{\mathcal{N}}_i} \sum_{r_{k'} \in \mathcal{R}_{j'i}} \exp\left(\hat{\psi}^{(l)}\left(\left[\boldsymbol{h}_i^{(l)} \| \boldsymbol{h}_{j'}^{(l)} \| \boldsymbol{z}_{k'}^{(L)}\right]\right)\right)$$

Modeling Relation-Entity Interactions

- Final embedding vectors computation

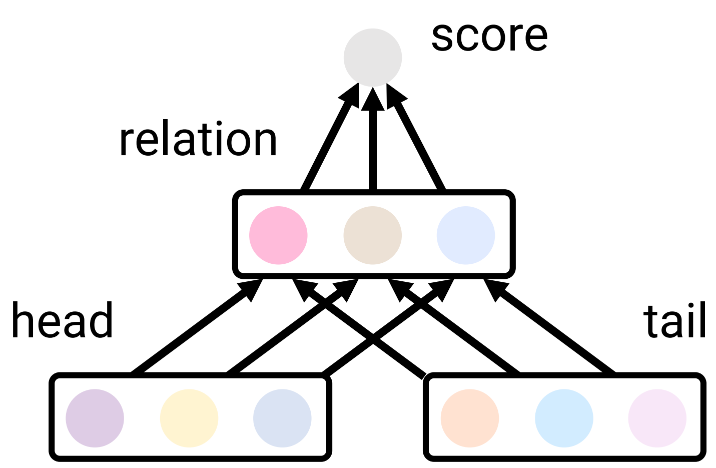
$$\boldsymbol{z}_k = \boldsymbol{M}\boldsymbol{z}_k^{(L)} \text{ and } \boldsymbol{h}_i = \widehat{\boldsymbol{M}}\boldsymbol{h}_i^{(\hat{L})}$$

- Scoring function

$$f(v_i, r_k, v_j) = \boldsymbol{h}_i^{\top} \mathrm{diag}(\overline{\boldsymbol{W}}\boldsymbol{z}_k)\boldsymbol{h}_j$$

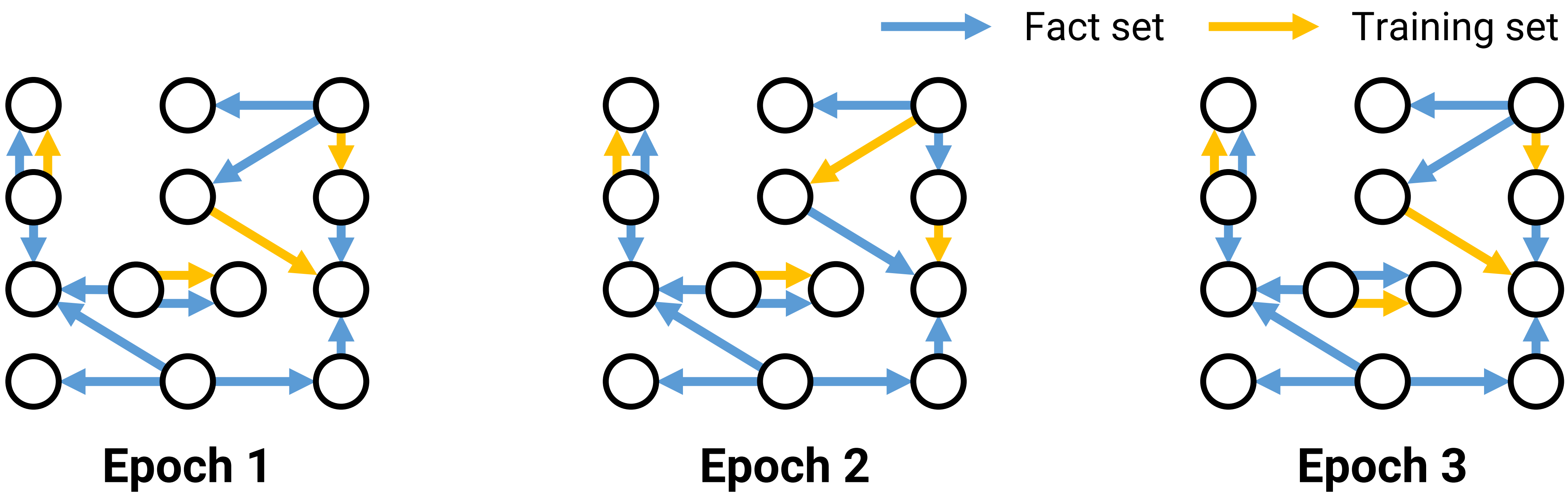
- Loss

$$\sum_{(v_i,r_k,v_j)\in\mathcal{T}_{\mathrm{tr}}} \sum_{(\mathring{v}_i,r_k,\mathring{v}_j)\in\mathring{\mathcal{T}}_{\mathrm{tr}}} \max\left(0, \gamma - f(v_i, r_k, v_j) + f\left(\mathring{v}_i, r_k, \mathring{v}_j\right)\right)$$



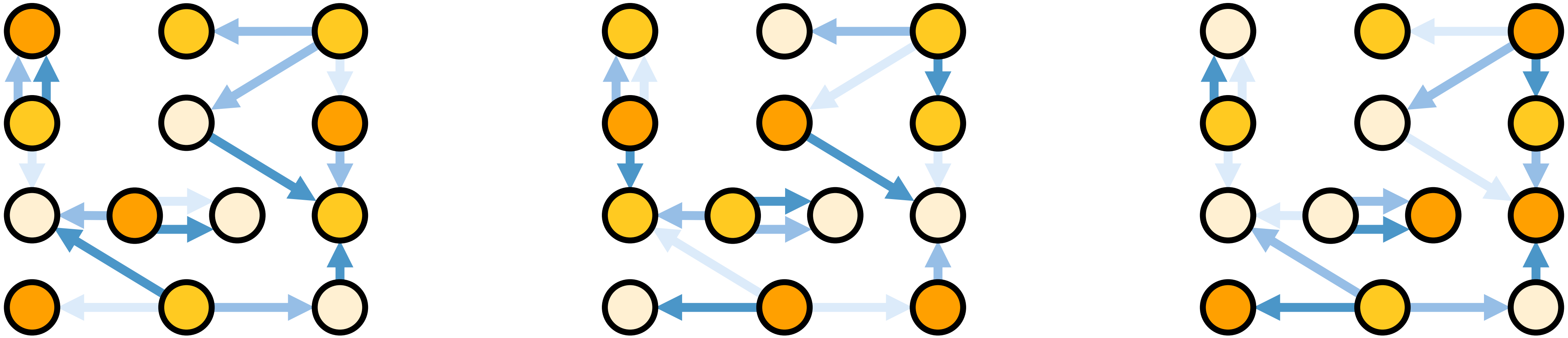
Dynamic split

- Randomly **re-split** the fact set and the training set
  - Fact set: used for aggregating neighboring embeddings
  - Training set: used for calculating the loss



Re-initialization

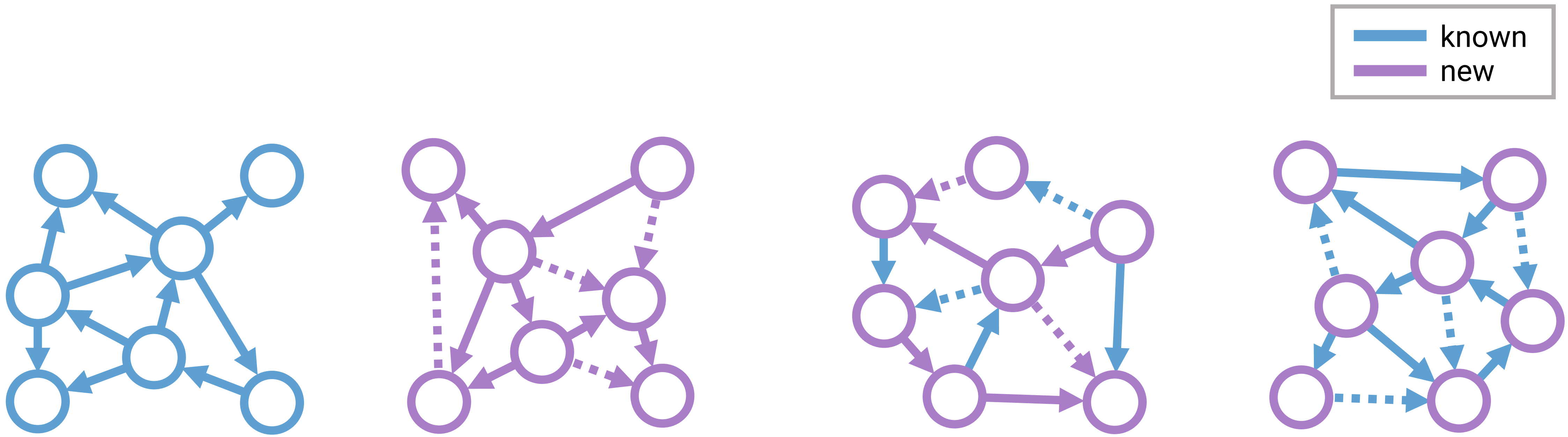
- Randomly **re-initialize** all feature vectors of entities and relations
  - Learns how to compute embedding vectors using random feature vectors
  - Related to the expressive power of GNNs



Epoch 1 Epoch 2 Epoch 3

Experimental Results

- Datasets
  - Based on NELL, Wikidata, and Freebase
  - Create **13 real-world datasets** with various inductive settings



Training Graph

Inductive Inference for Relations

Semi-Inductive Inference for Relations

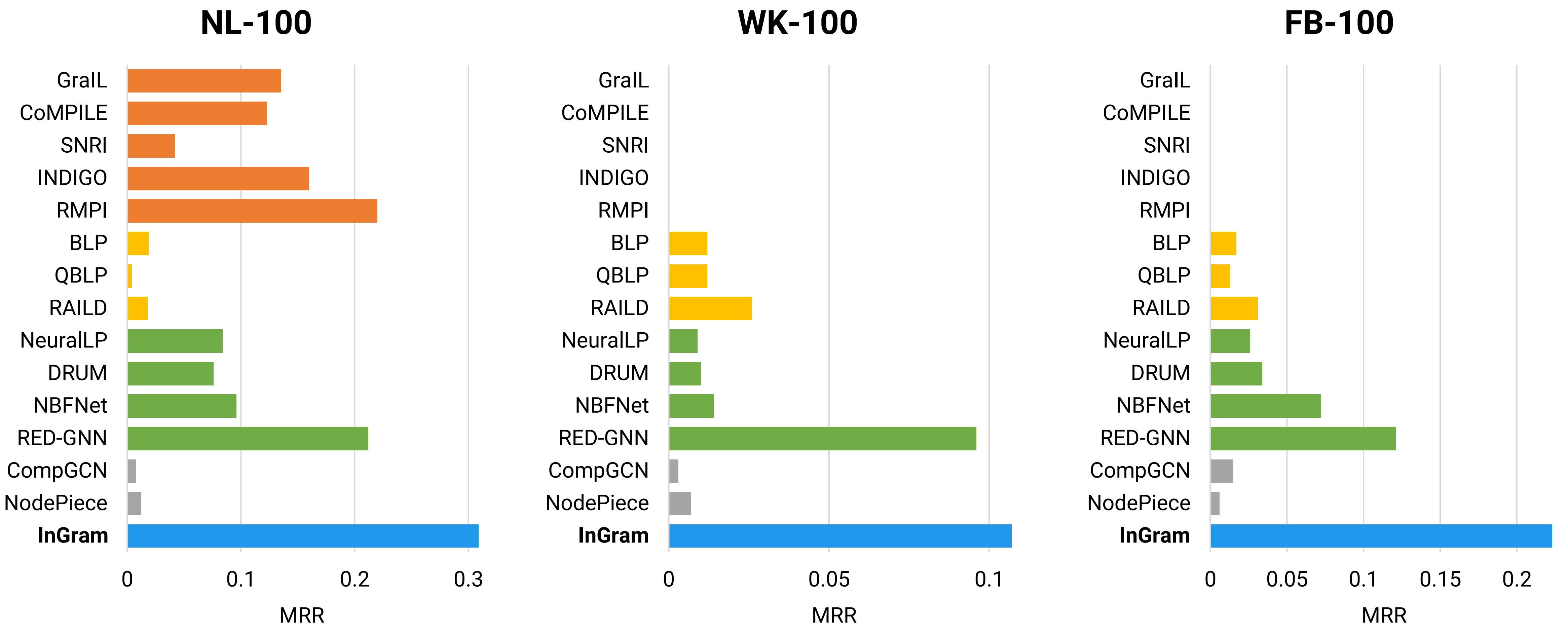
Transductive Inference for Relations

Experimental Results

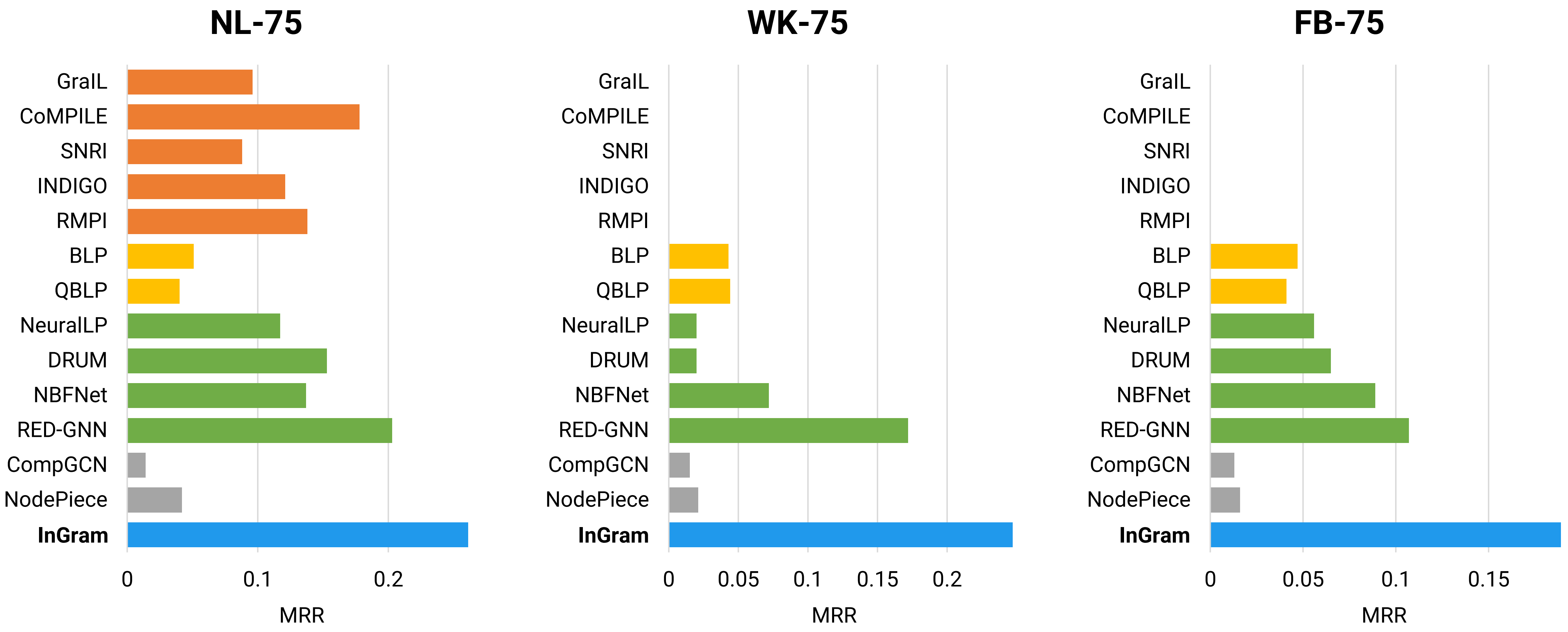
- Datasets
  - Based on NELL, Wikidata, and Freebase
  - Create **13 real-world datasets** with various inductive settings

- Comparison with **14 baselines**
  - **Subgraph sampling**: GraIL, CoMPILE, SNRI, INDIGO, RMPI
  - **BERT-based**: BLP, QBLP, RAILD
  - **Rule-based**: NeuralLP, DRUM, NBFNet, RED-GNN
  - **Others**: CompGCN, NodePiece

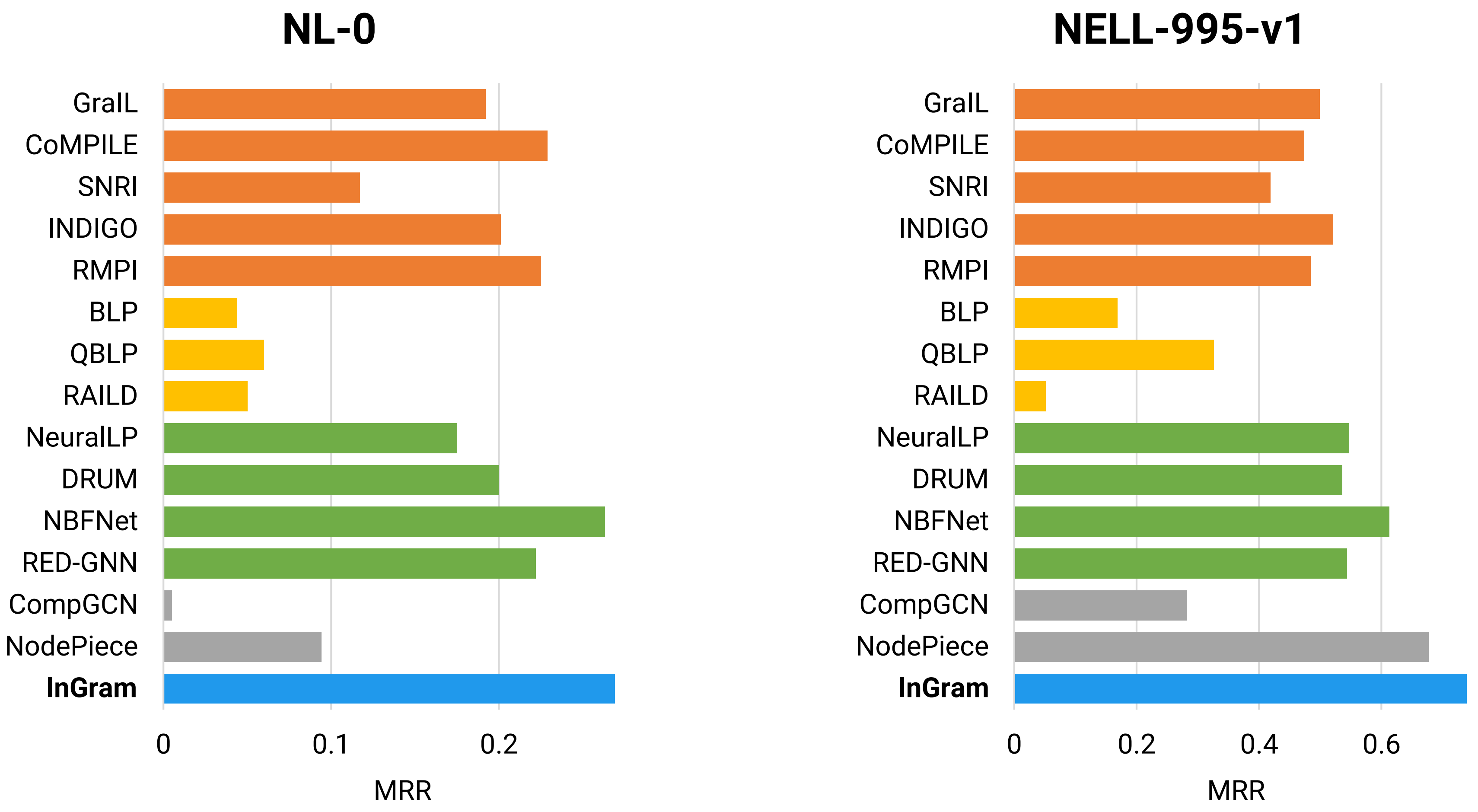
Inductive Inference for Relations



Semi-Inductive Inference for Relations



Transductive Inference for Relations



Conclusion

- Explore various **inductive settings**
- Define the **relation graph** to handle new relations at inference time
- Propose **InGram**, which learns to generate embeddings solely based on the structure of a given knowledge graph
- InGram significantly outperforms state-of-the-art methods for **inductive**, **semi-inductive**, and **transductive** inferences for relations

Our datasets and codes are available at:

https://github.com/bdi-lab/InGram

◀ GitHub

You can find us at:

{jjlee98, chanyoung.chung, jjwhang}@kaist.ac.kr

https://bdi-lab.kaist.ac.kr

◀ BDILab

BDILab

BIG DATA INTELLIGENCE