Neural-symbolic Knowledge Representation and Reasoning

Jiaoyan Chen
Lecturer in Department of Computer Science
University of Manchester

ESSAI 2024 Athens

This course is

- introductory
- aimed at general computer scientist

- taught by
  - Uli Sattler - days 1-2
  - **Jiaoyan Chen - days 3-5**

(Hiking in Egina, Greece, 11/2023)

- explores combination/integration/collaboration of
  - Symbolic &
  - **Neural**
    - approaches to knowledge representation, reasoning, ML, …

Overview of this course

Day	Topic	Concepts	Technologies
1	Knowledge Graphs	parsing/serialisation, queries, schemas, validation & reasoning	RDF(S), SPARQL, SHACL,
2	Ontologies	Facts & background knowledge, entailments, reasoning & materialisation	OWL, OWL API, Owlready, Protégé
3	Knowledge Graph Embeddings	Classis Es, variants, inductive inference, literal-aware Es, incremental Es, application	TransE, TransH, TransR, GCN, R-GCN, OntoZSL, RMPI
4	Ontology Embeddings	Geometric embeddings, literal-aware OEs, faithfulness, evaluation & applications	ELEm, $\mathrm{Box}^2\mathrm{EL}$, OWL2Vec*, LogMap-ML, ZSL, mOWL
5	Language Models & KR, Discussion & Outlook	LM for KR, ontology & KG for LLM	BERTMap, BERTSubs, DeepOnto, ICON, BLINKOut, GraphRAG

Day 3 Knowledge Graph Embeddings

Part I: Foundations

What is semantic embedding?

- Motivation:
  - Feed symbols such as letters, words and entities into some statistical processing (e.g., machine learning and data mining)
- Represent the symbols by vectors with their **relationships** (**semantics**) kept in the vector space

  E.g.,
  - V(queen) – V(king) ≈ V(mother) – V(father)
  - There is some partnership between queen and king, and between father and mother
- **Sub-symbolic or neural knowledge representation**

One-hot Representation

Vocabulary: (cat, mat, on, sat, the)

=>

cat: [1,0,0,0,0]   mat: [0,1,0,0,0]   on: [0,0,1,0,0]
sat: [0,0,0,1,0]   the: [0,0,0,0,1]

"The cat sat on the mat"

		cat	mat	on	sat	the
the	=>	0	0	0	0	1
cat	=>	1	0	0	0	0
sat	=>	0	0	0	1	0
…				…		

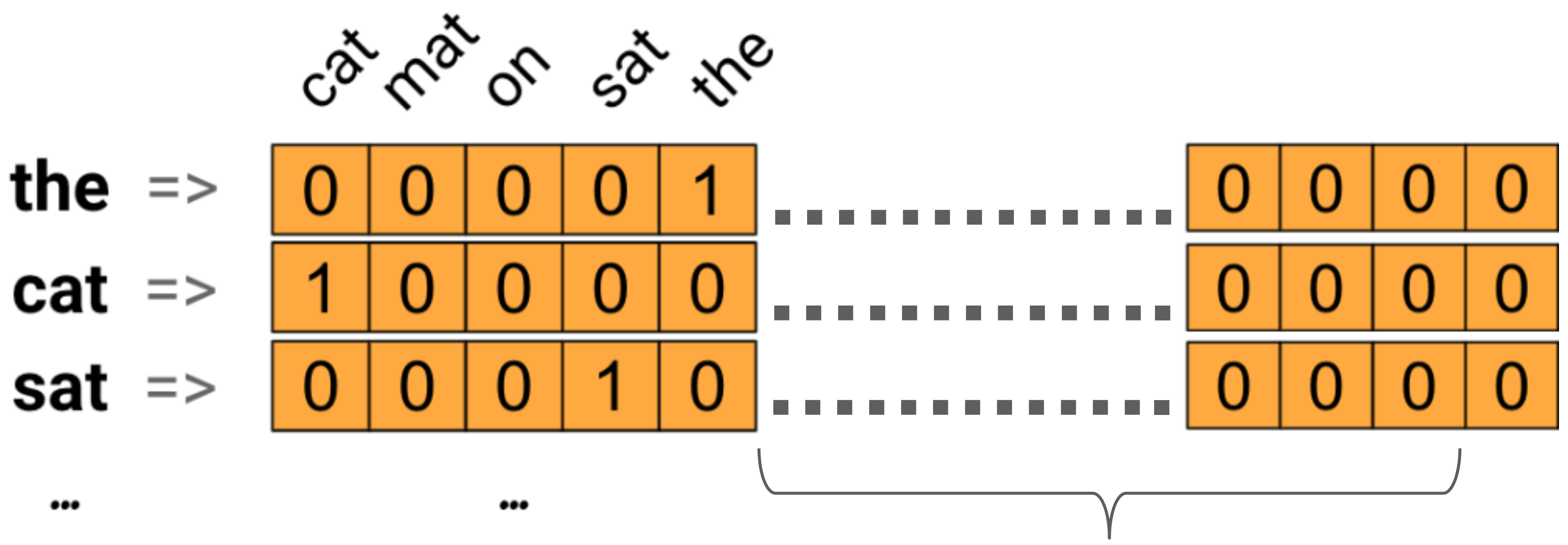
One-hot Representation

Vocabulary: (cat, mat, on, sat, the, …) (>10000)

=> cat: [1,0,0,0,0, ………… 0]

Dimension > 10000

"The cat sat on the mat"



All remaining slots are zero

One-hot Representation

- **Could lead to a very high dimension**
  - Consider a supervised learning problem with no enough samples …
- **Cannot keep the semantics**

$$\text{star} \quad [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, \cdots]$$
$$\text{sun} \quad [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, \cdots]$$

$$\text{sim(star, sun)} = 0$$ ☹

But they can act as the initial input of machine learning models with big data for training e.g., word embedding models

Word Embedding

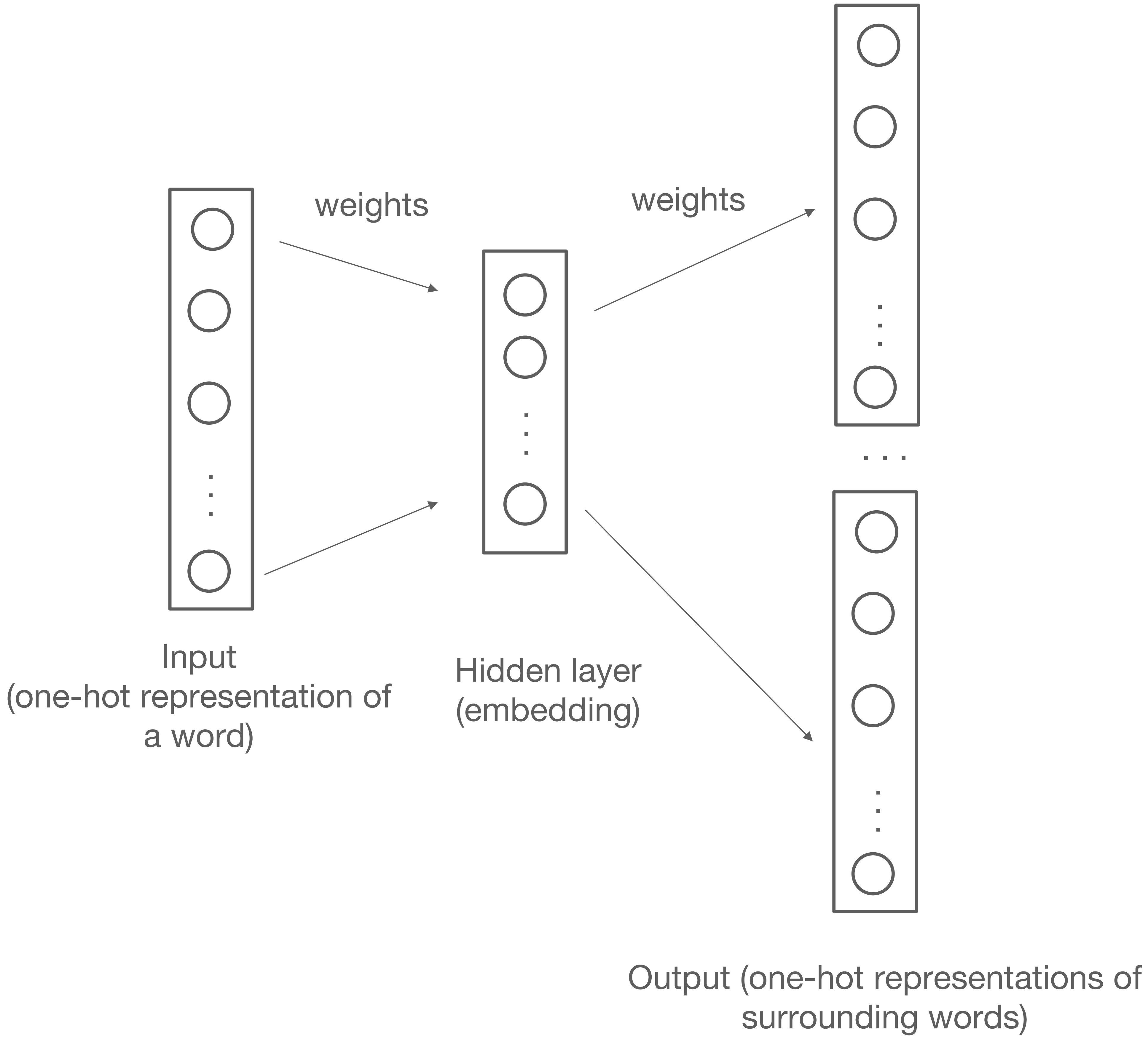
- Distributed representation of words via learning from a large text corpus
  - Represent a word by a low-dimension (e.g., 500) dense vector with its correlation and co-occurrence with **other words appearing in its contexts** kept

- E.g., Word2Vec -- neural network model by Google

Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." NIPS (2013)

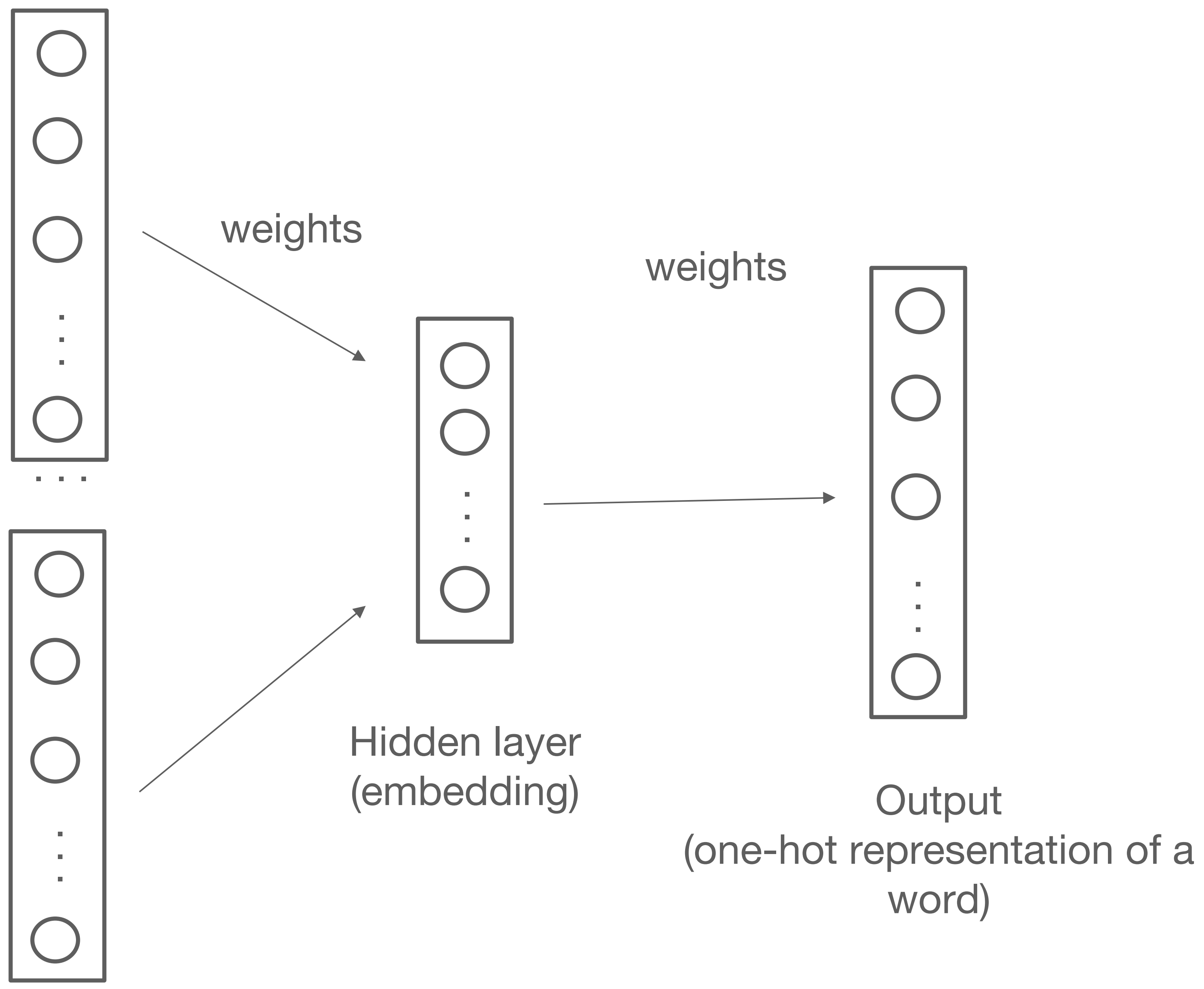
Word Embedding (Word2Vec)

- Model #1: Continuous Skip-gram
  - Training Insight: given a word, predict the surrounding words in a sentence
  - Minimize the loss
    - on a large text corpus (big data)

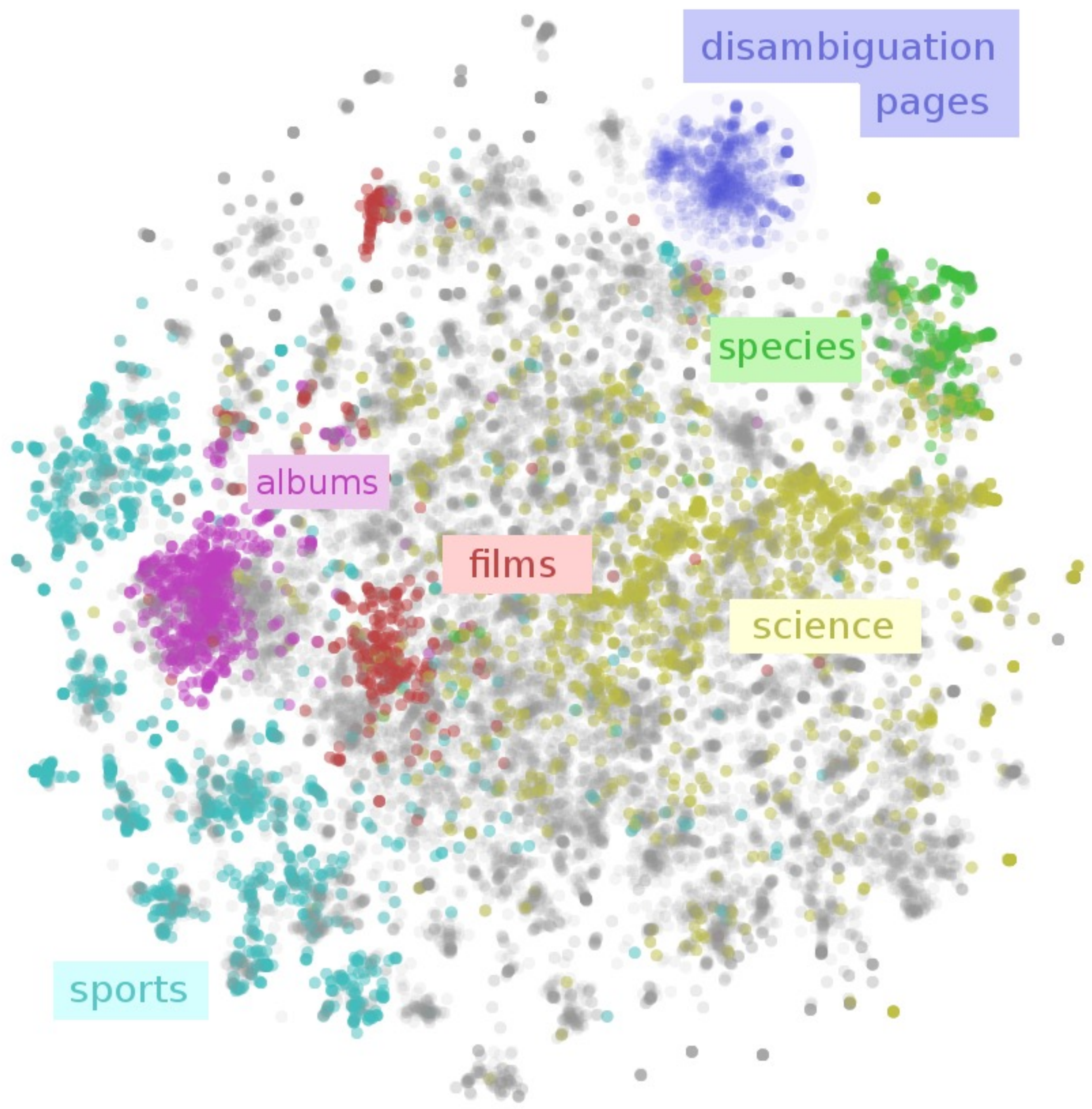


Word Embedding (Word2Vec)

- Model #2: Continuous Bag of Word (CBOW)
  - Training Insight: mask a word in a sentence, predict this word with its surrounding words
  - Minimize the loss on
    - a large text corpus



Word Embedding



Similarity:

$$\textbf{euclidean}(u, v) = \sqrt{\sum_{i=1}^{n} |u_i - v_i|^2}$$

$$\textbf{cosine}(u, v) = 1 - \frac{\sum_{i=1}^{n} u_i \times v_i}{||u||_2 \times ||v||_2}$$

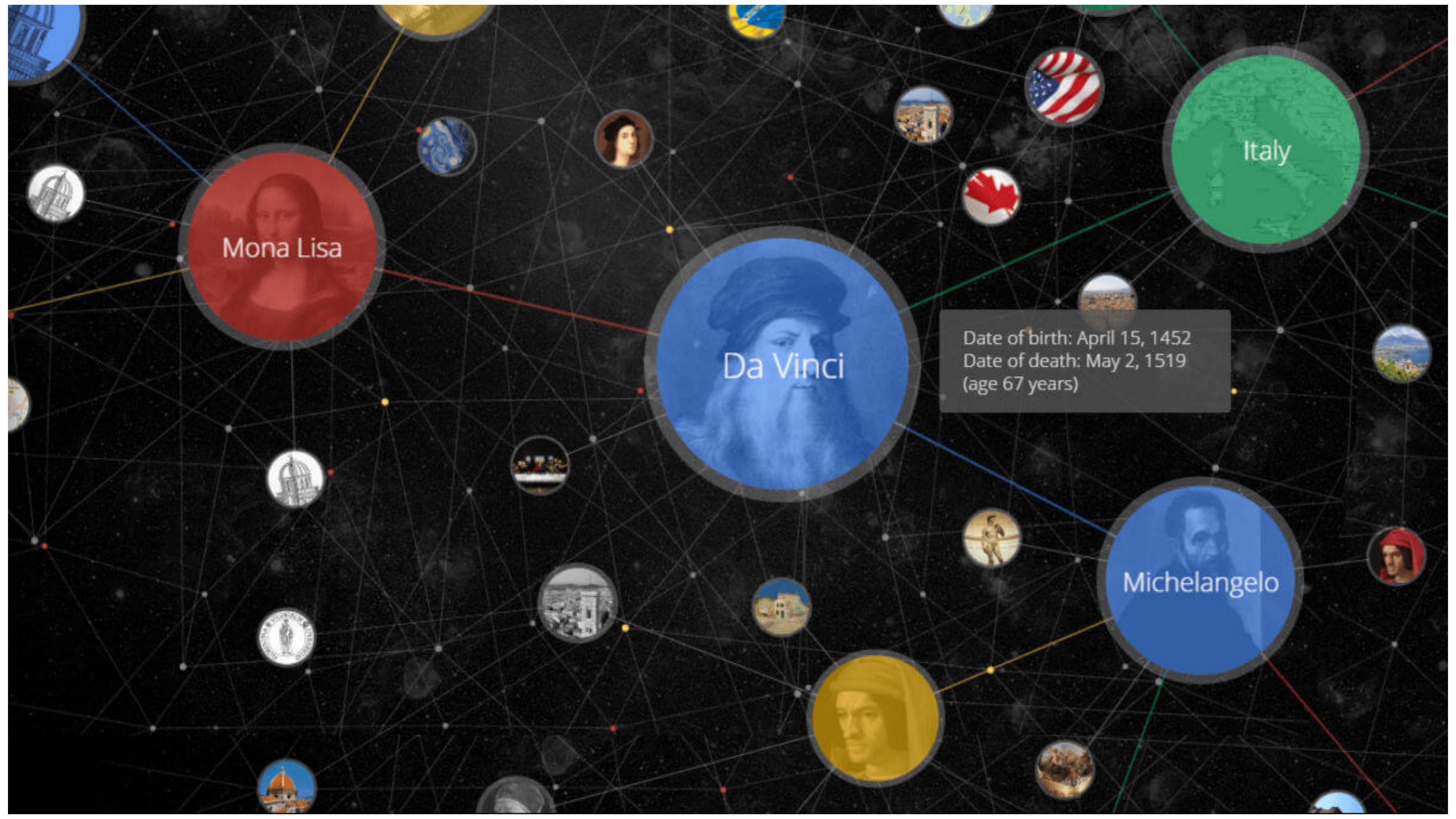
Wide application and great success in NLP

Contextual and Non-contextual Word Embeddings

- E.g., “the bank robber was seen on the river bank”
- For non-contextual word embedding e.g., Word2Vec
  - **V(bank) = V(bank)**
  - One word one vector; ignore the context of a word
- For contextual word embedding e.g., BERT
  - **V(bank) ≠ V(bank)**
  - A word’s vector varies from context to context
  - Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018)

What is knowledge graph?

- “Knowledge Graph” was proposed by Google in 2012, referring to its services to enhance its search engine’s results with knowledge gathered from a variety of sources



- Knowledge ≈ Instances + Facts, represented as RDF triples e.g., <Box, hasParent, Alex>
- Linked and graph structured data

In this lecture we distinguish KG (Day 3) with ontology (Day 4). KGs are in form of RDF data, RDF Data + Literals, RDF data + schema/constraints/rules

TransE: Take relation as translation

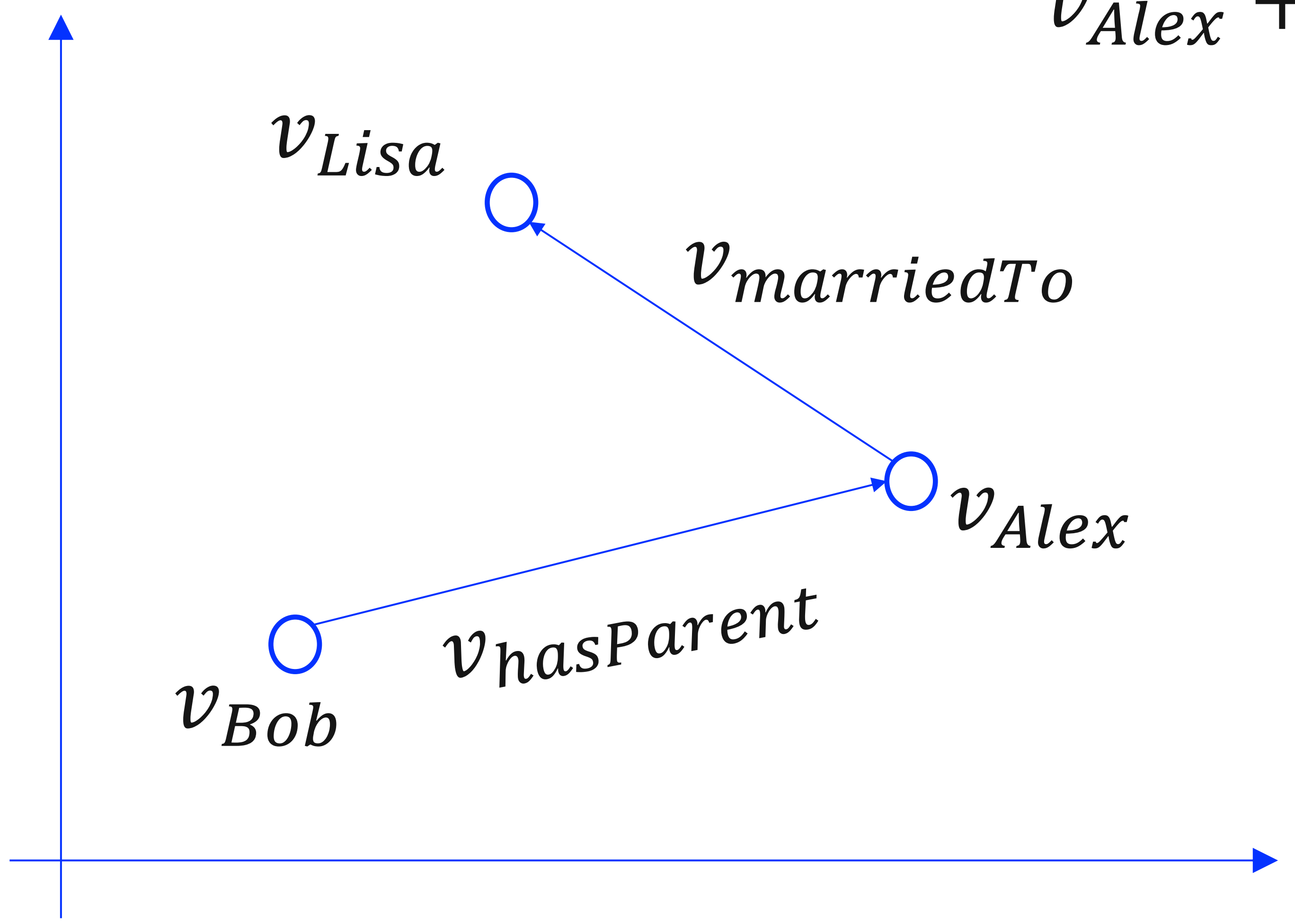
- For each triple <*h*, *r*, *t*>, *h* is translated to *t* by *r* *(denoted by vector l)*

$$v_{Bob} + v_{hasParent} \approx v_{Alex}$$
$$v_{Alex} + v_{marriedTo} \approx v_{Lisa}$$
$$v_i \in R^n$$

RDF triples

<Bob, hasParent, Alex>
<Alex, marriedTo, Lisa>
…

Sampling and Learning



Bordes, A., et al. "Translating embeddings for modeling multi-relational data." *Advances in neural information processing systems* 26 (2013).

TransE: Take relation as translation

- Score function for a triple

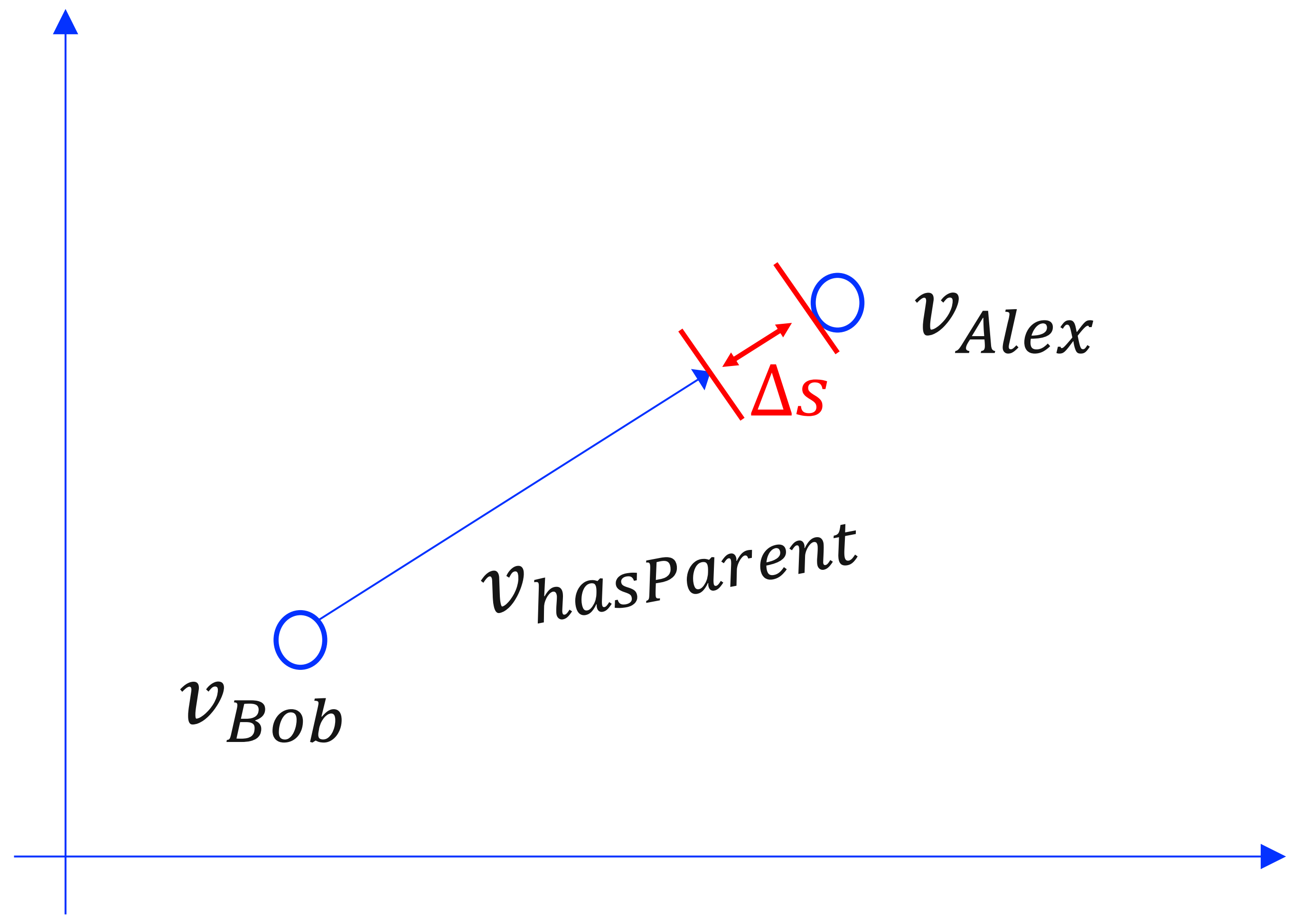
$$f(h, l, t) = |h + l - t|_{L_1/L_2}$$

L_1 (Manhattan distance):

$$\mathbf{d}_1(a, b) = \|a - b\|_1 = \sum_{i=1} |a_i - b_i|.$$

L_2 (Euclidean distance):

$$\mathbf{d}_2(a, b) = \|a - b\| = \|a - b\|_2 = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2}$$



TransE: Take relation as translation

- Negative Sampling
  - Corrupting the head or tail
    - E.g., <Bob, hasParent, Lisa>, <Tom, hasParent, Alex>

Loss

Margin

$$\mathcal{L} = \sum_{(h,l,t)\in S} \sum_{(h',l,t')\in S'_{(h,l,t)}} [\lambda + d(h+l,t) - d(h'+l,t')]_+$$

Positive triples

Negative triples

Distance function

TransE: Take relation as translation

Algorithm 1 Learning TransE

input Training set $S = \{(h, \ell, t)\}$, entities and rel. sets E and L, margin γ, embeddings dim. k.

initialize $\boldsymbol{\ell} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$
$\boldsymbol{\ell} \leftarrow \boldsymbol{\ell} / \|\boldsymbol{\ell}\|$ for each $\ell \in L$
$\mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$

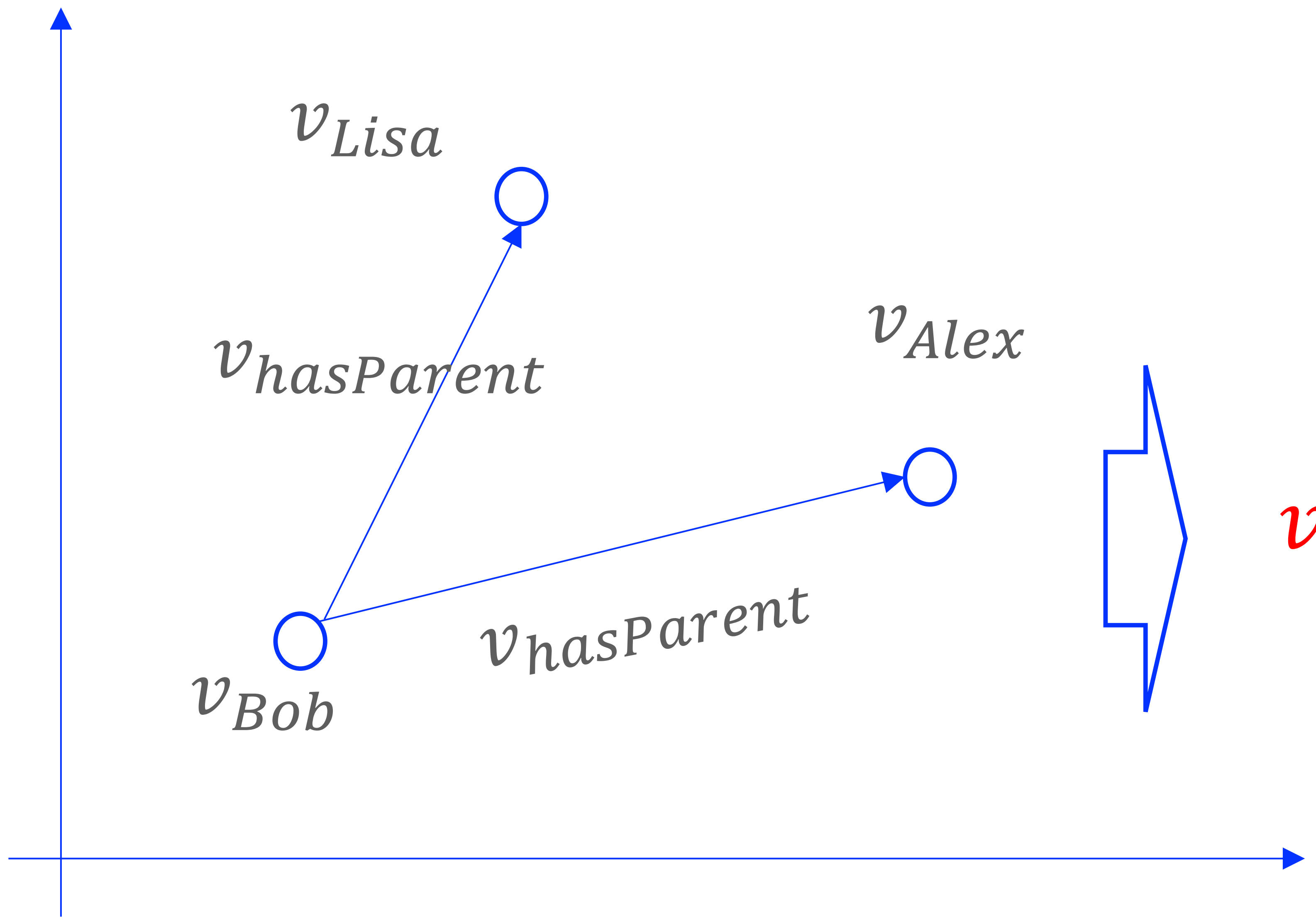
loop
$\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$
$S_{batch} \leftarrow \text{sample}(S, b)$ // sample a minibatch of size b
$T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets
for $(h, \ell, t) \in S_{batch}$ **do**
$(h', \ell, t') \leftarrow \text{sample}(S'_{(h,\ell,t)})$ // sample a corrupted triplet
$T_{batch} \leftarrow T_{batch} \cup \{((h, \ell, t), (h', \ell, t'))\}$
end for
Update embeddings w.r.t. $\displaystyle\sum_{((h,\ell,t),(h',\ell,t')) \in T_{batch}} \nabla\left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})\right]_+$
end loop

Entities and relations are initialized uniformly, and normalized

Favors lower distance (or higher score) for true triplets, high distance (or lower score) for false ones

TransE: Take relation as translation

Limitation: Cannot deal with **one-to-many, many-to-one and many-to-many relations**

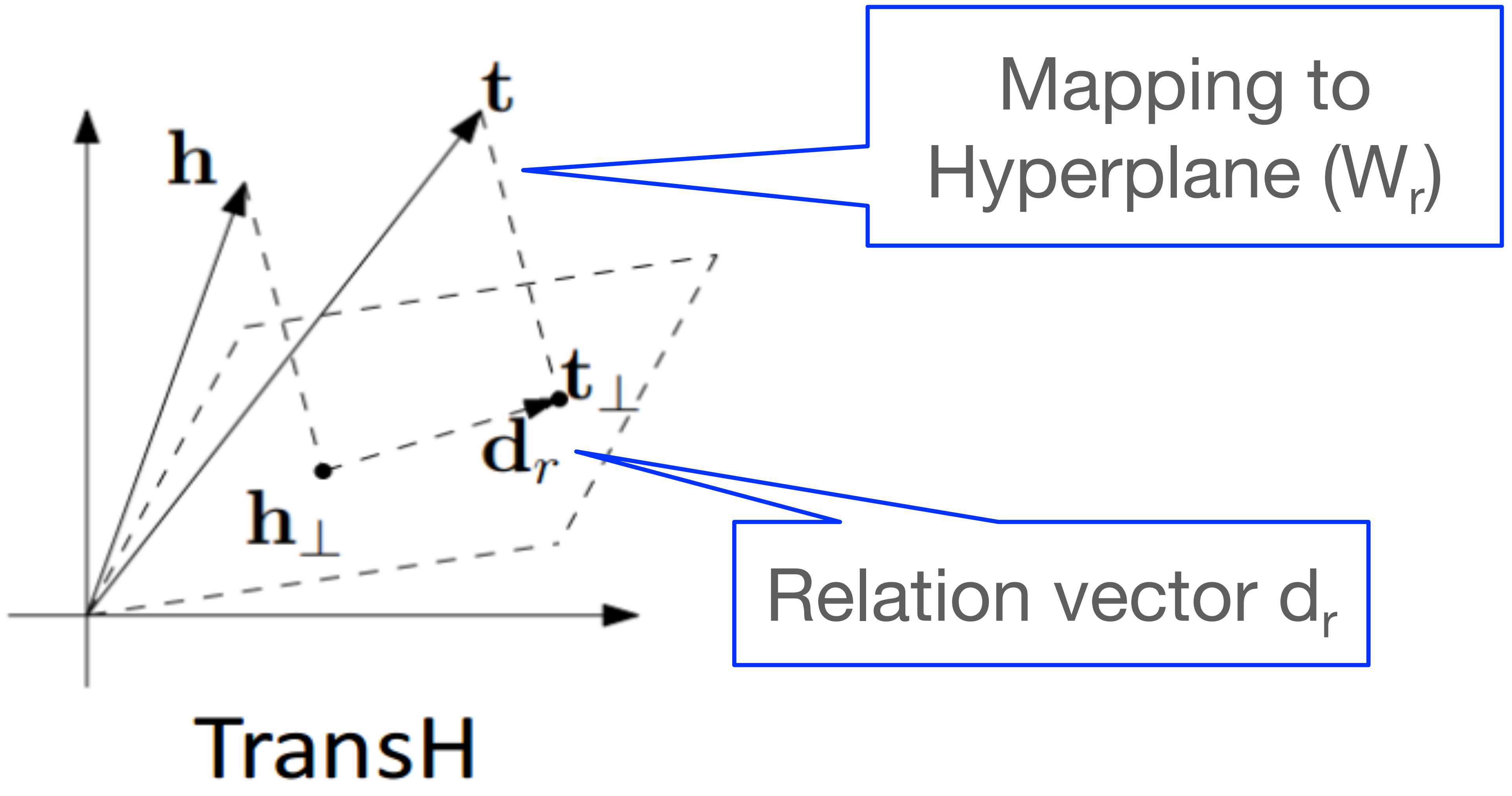


$v_{Alex} = v_{Lisa}$!!

Variants of TransE: TransH

- To address the limitation of failing to model one-to-many, many-to-one and many-to-many relations
- TransH: model a relation as a hyperplane together with a translation operation on it

$$h_{\perp} = h - w_r^T h w_r$$
$$t_{\perp} = t - w_r^T t w_r$$

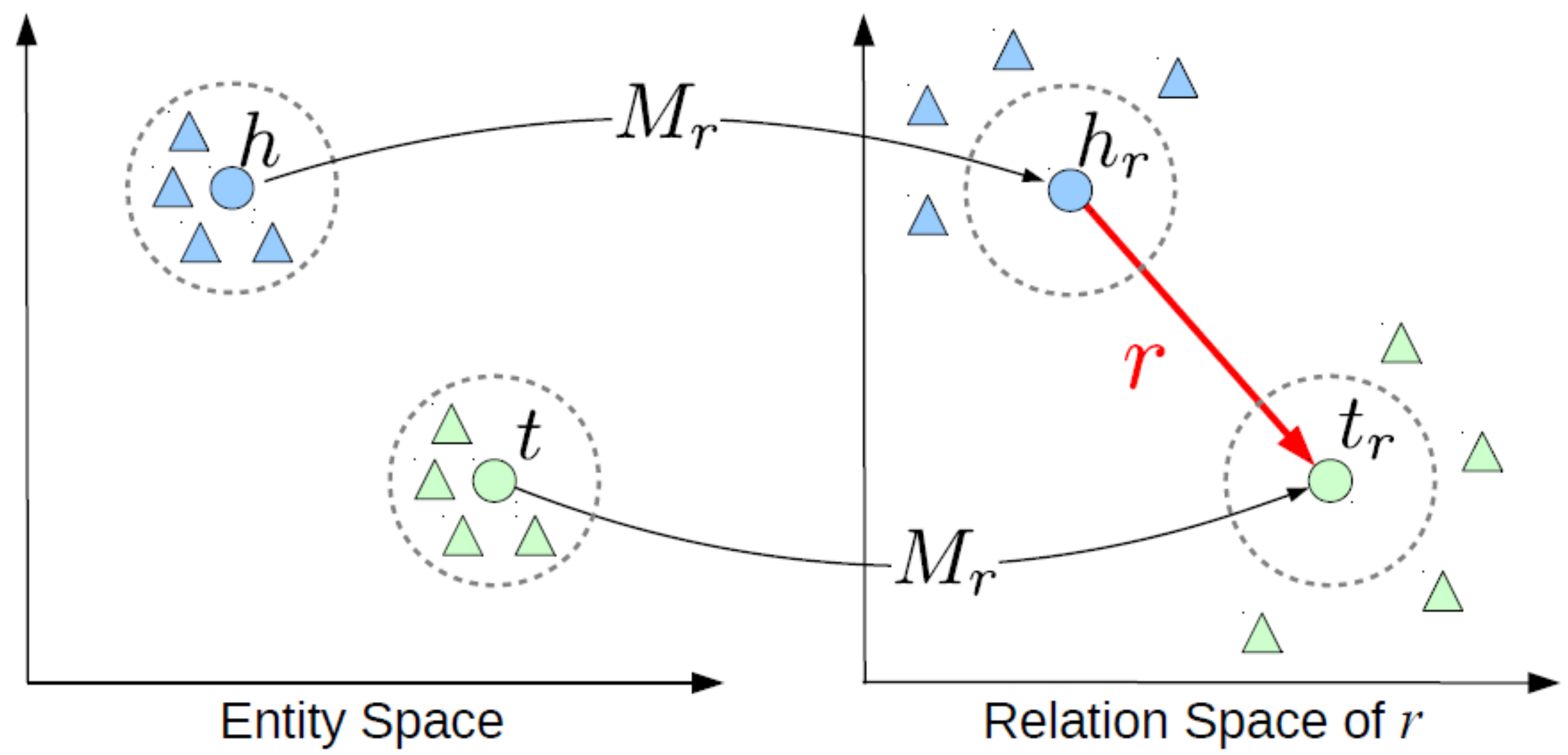


In <Bob, hasParent, Lisa> and <Bob, hasParent, Alex>, Lisa and Alex can have different embeddings, even they become the same when mapped to the hyperplane of hasParent.

Wang, et al. (2014). Knowledge graph embedding by translating on hyperplanes, AAAI.

Variants of TransE: TransR

- Assume the head and tail could lie in different spaces; map them into the same space where the relation lies before calculating the triple score



$$f_r(h,t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_2^2.$$

$$\mathbf{h}_r = \mathbf{h}\mathbf{M}_r, \quad \mathbf{t}_r = \mathbf{t}\mathbf{M}_r.$$

Lin, et al. (2015). Learning entity and relation embeddings for knowledge graph completion, AAAI.

Knowledge Graph Embedding Paradigms

- End-to-end geometric modelling (e.g., TransE)
  - Steps: Define score functions to model the likelihood of triples; define loss functions; learn the embeddings by minimizing the losses
  - Translation-based & decomposition-based
  - Many others: TransD, DistMult, ComplEx, HolE, etc.

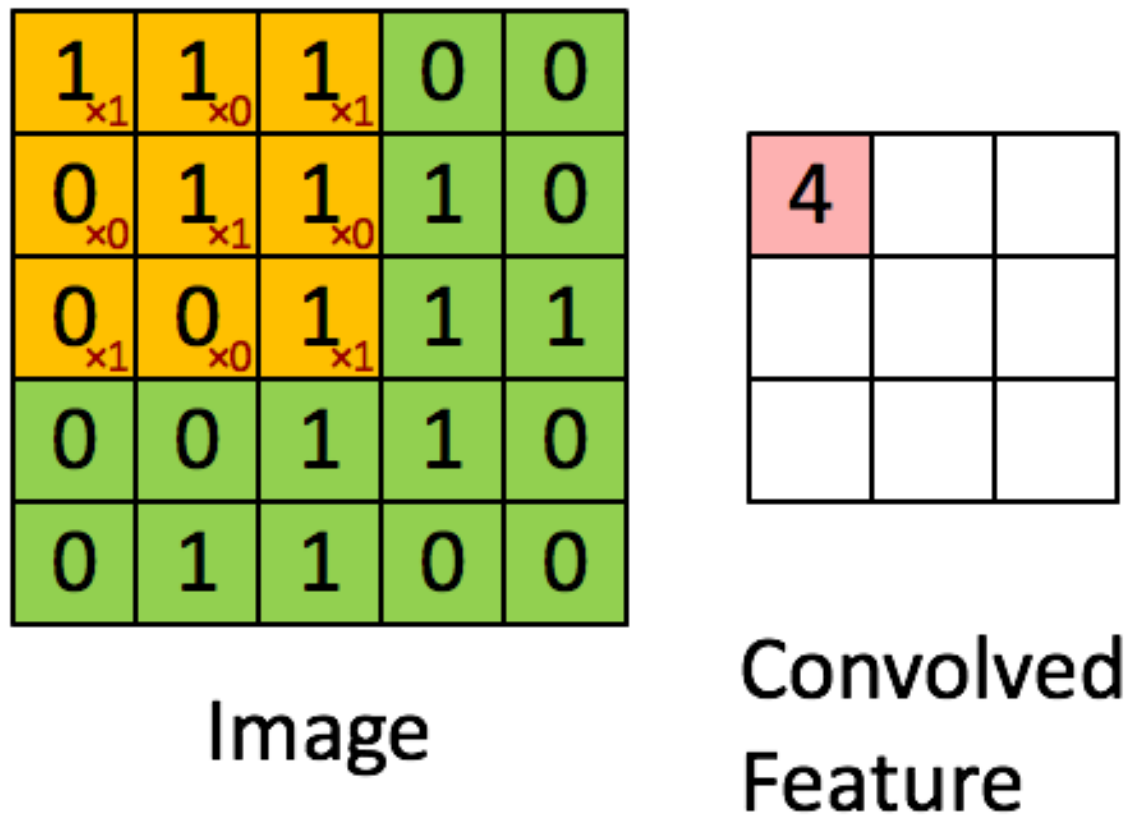
(Revisit)

Knowledge Graph Embedding Paradigms

- Graph Neural Networks
  - A function of representations of neighbors and itself from previous layers
    - **Aggregation** of neighbors
    - **Transformation** to a different space
    - **Combination** of neighbors and the node itself

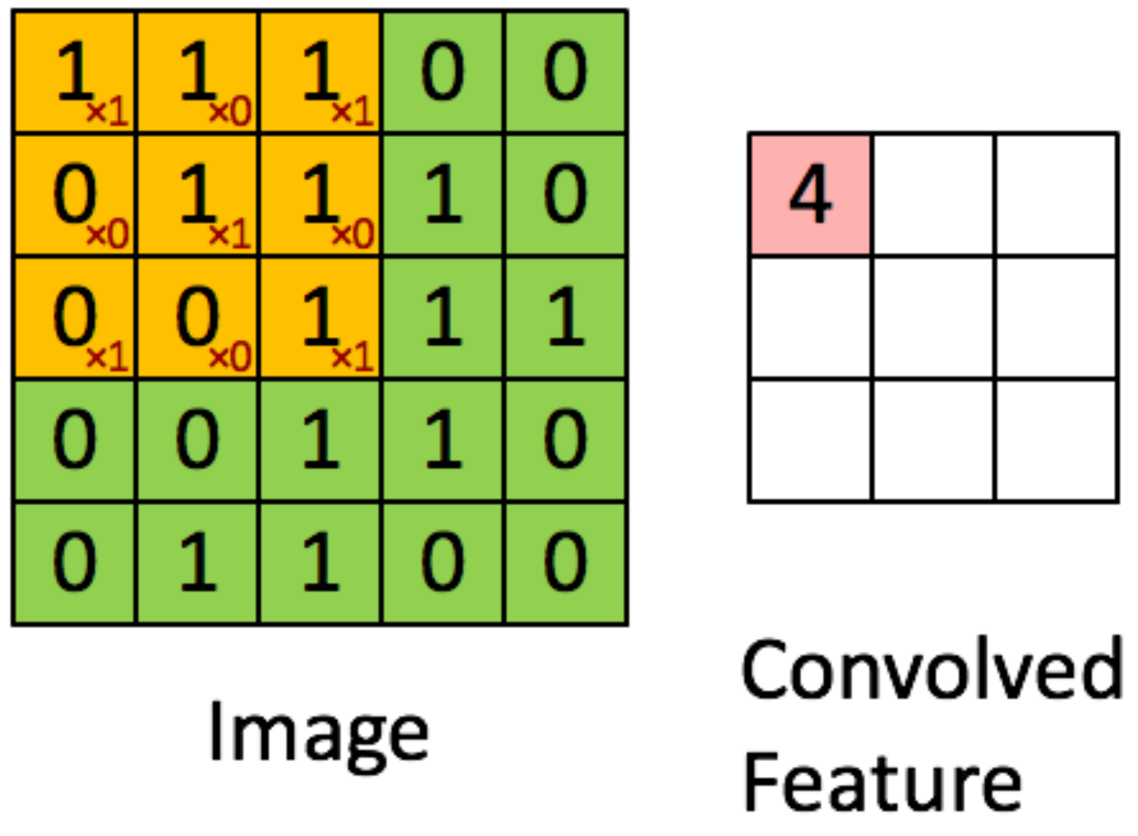
Convolutional Neural Network (CNN) vs GNN

- CNN

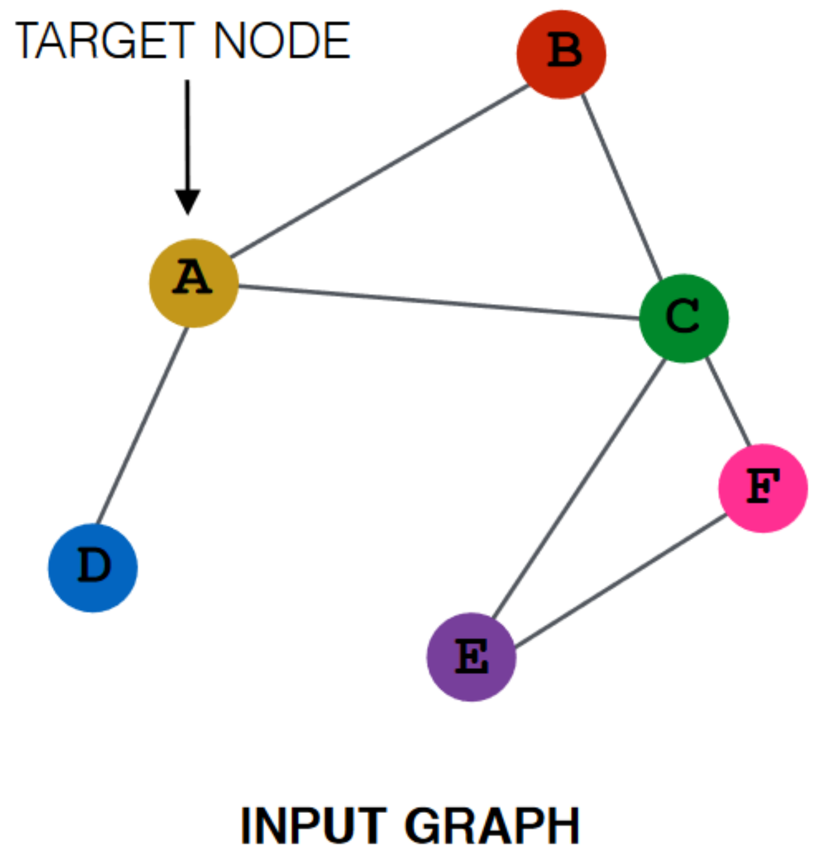


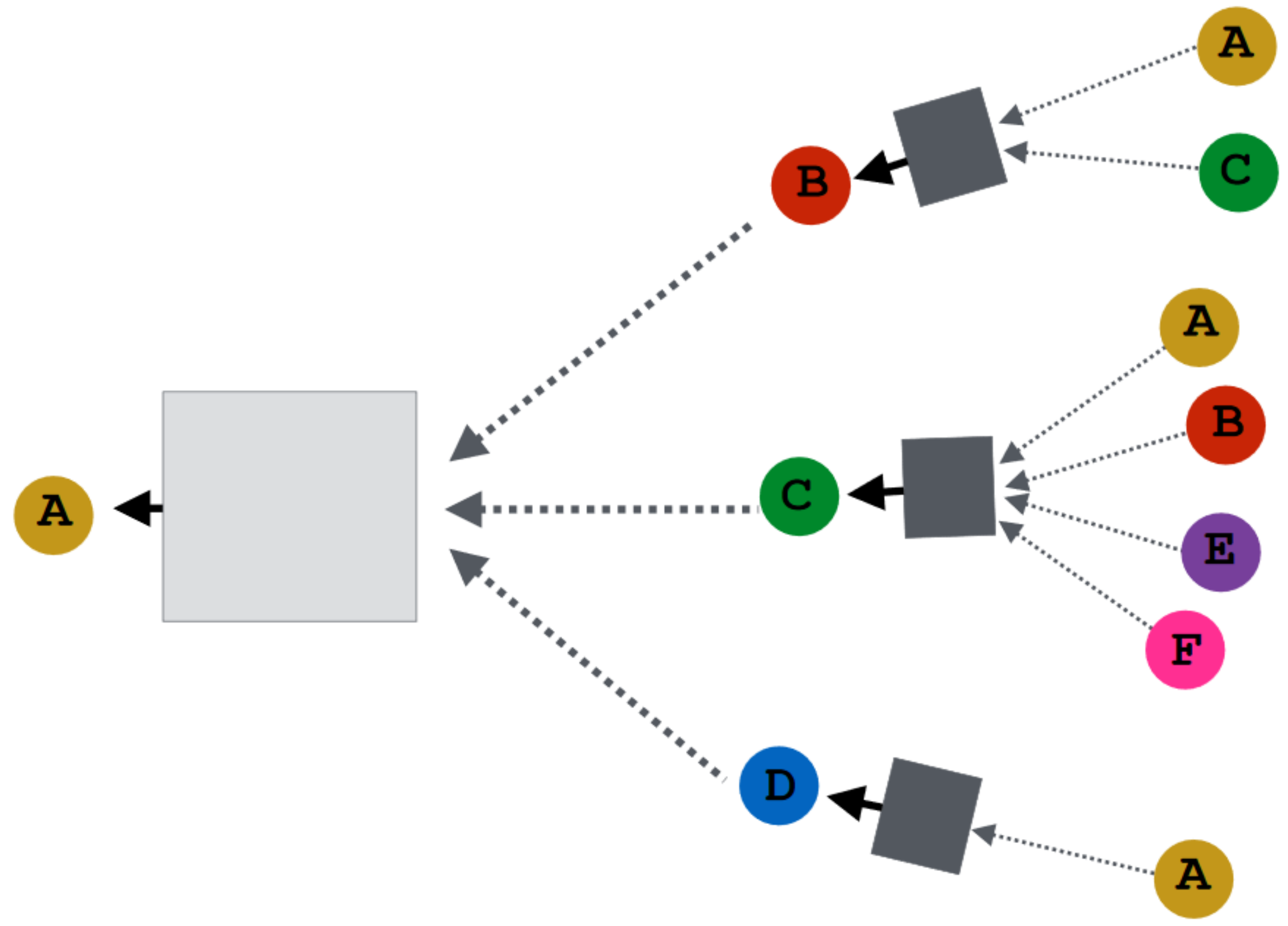
Convolutional Neural Network (CNN) vs GNN

- CNN



- GNN: Extend to irregular graph structure





Graph Neural Network

- KG embedding with GNN
  - Train a GNN unit the loss converges
  - Use final layer output as the embedding

Output of a node v at layer t

$$h_v^{(t)} = f\left(\underbrace{h_v^{(t-1)}}_{\text{Representation vector from previous layer for node } v}, \left\{\underbrace{h_u^{(t-1)} | u \in \mathcal{N}(v)}_{\text{representation vectors from previous layer for node v's neighbors}}\right\}\right)$$

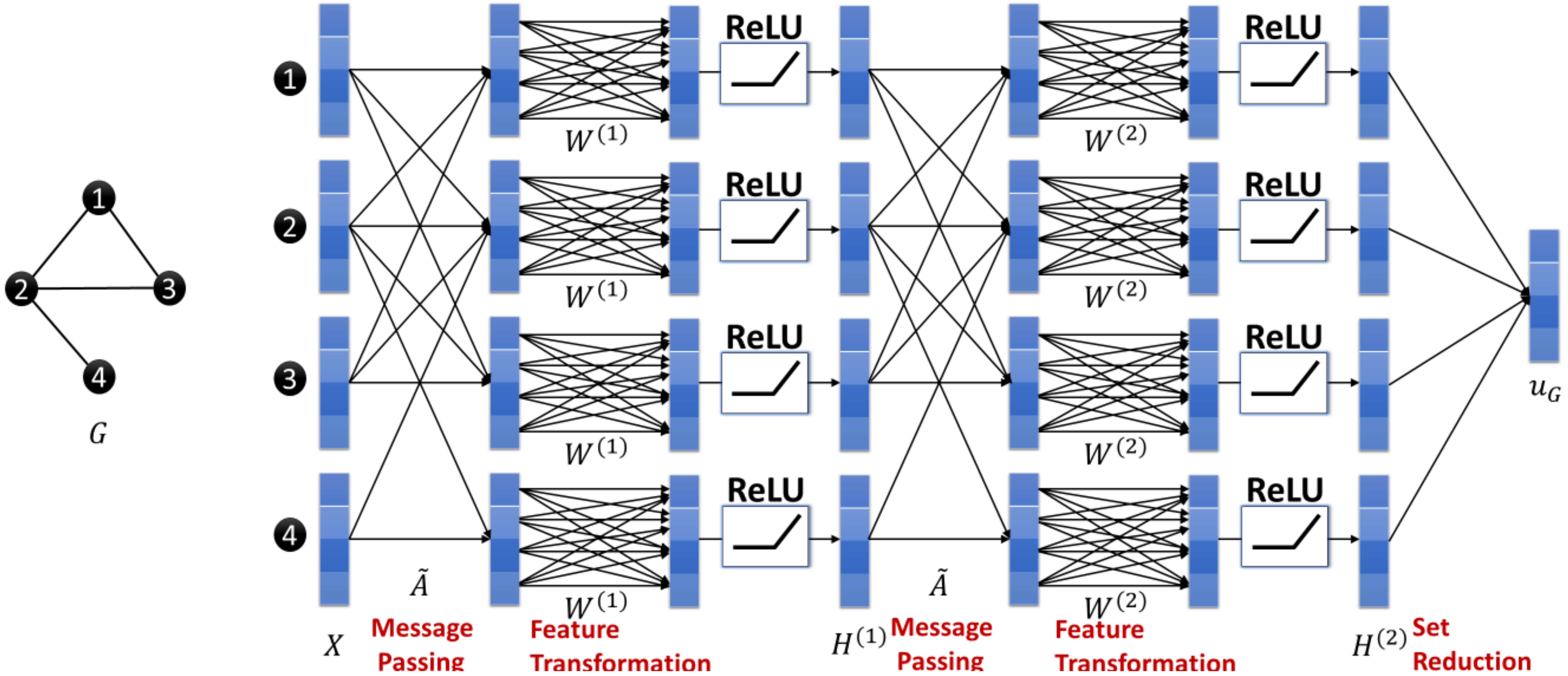
Graph Convolutional Network

- GCN

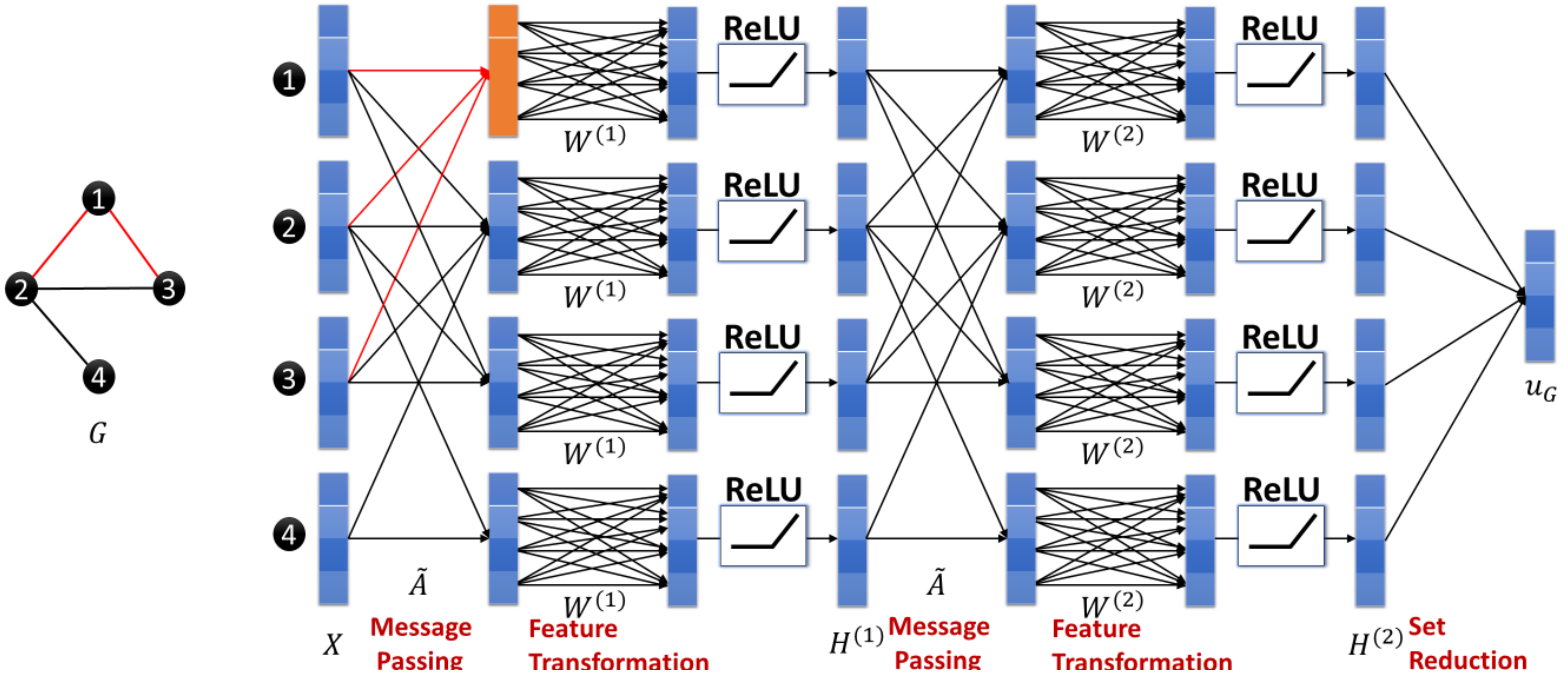
$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_k \sum_{u \in N(v) \cup v} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}}\right)$$

W_k: weight matrix at Layer k, shared across different nodes

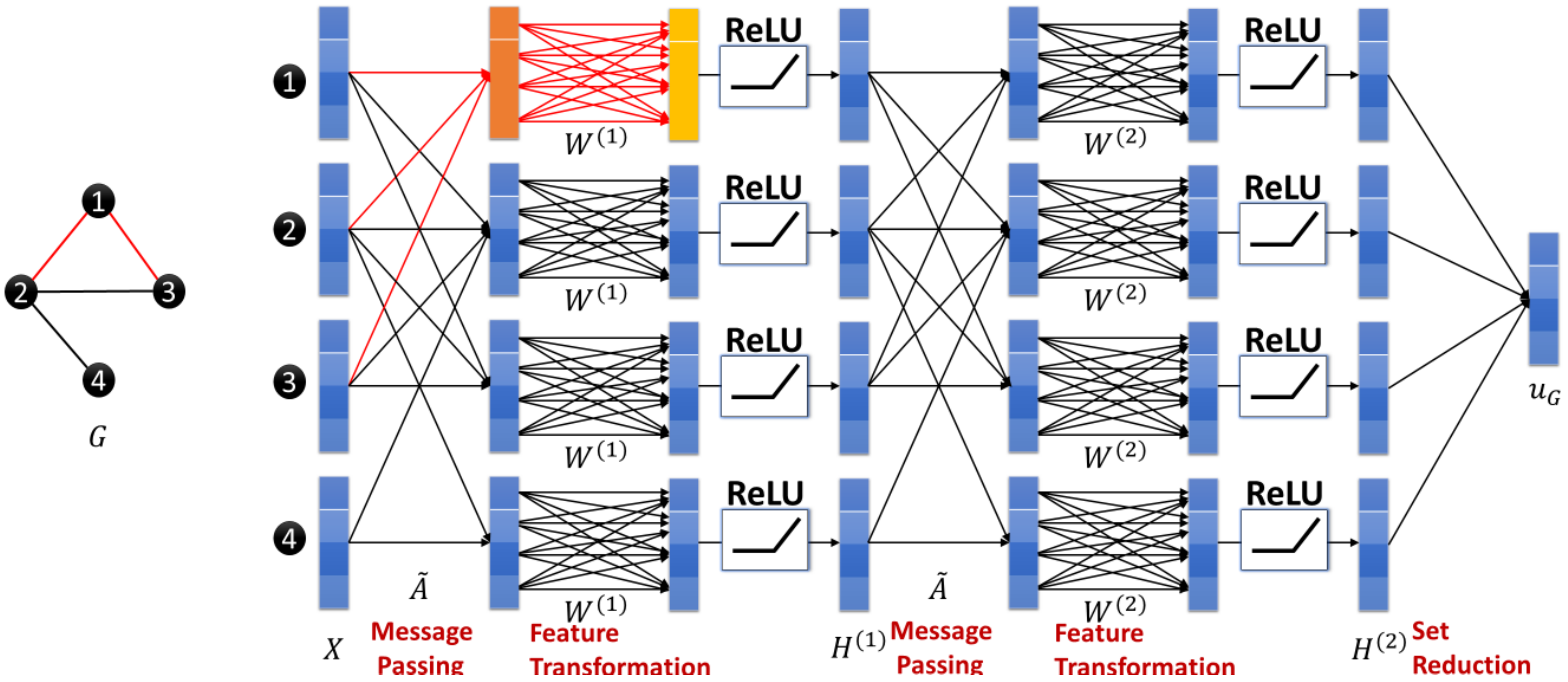
Graph Convolutional Network



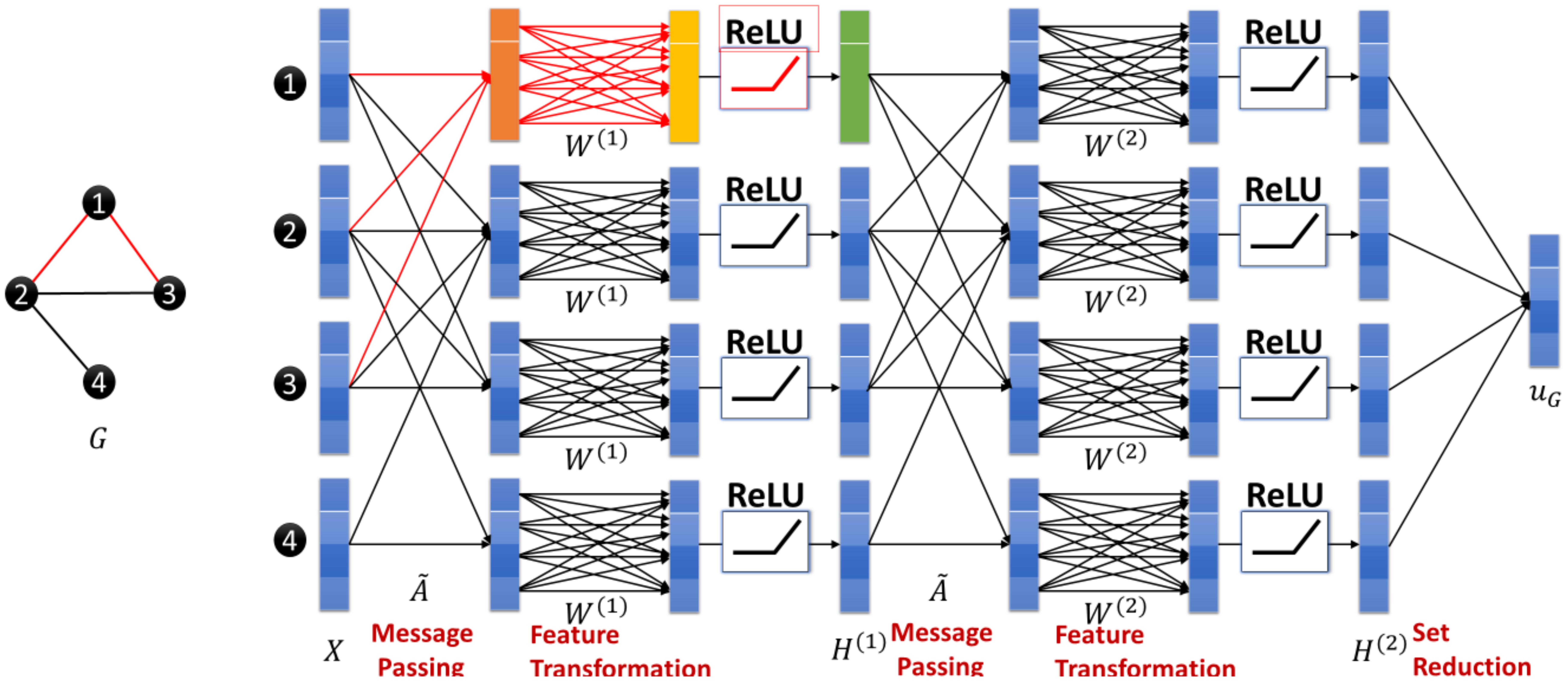
Graph Convolutional Network



Graph Convolutional Network



Graph Convolutional Network



R-GCN

To deal with **a graph with different relations**

Output of a node i at layer $l+1$ of **R-GCN**

Relation-aware
transformation weights

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

Relation-aware
normalization constant

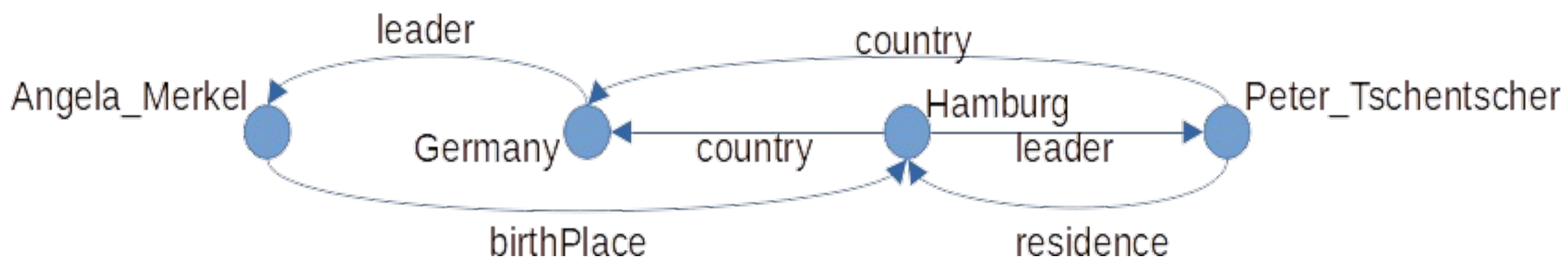
Schlichtkrull, Michael, et al. "Modeling relational data with graph convolutional networks." *The semantic web: 15th international conference, ESWC 2018*.

Knowledge Graph Embedding Paradigms

- Pipeline (e.g., RDF2Vec)
  - Extract sentences (sequences of entities) from the KG, with the relationship between entities kept in the sentences
  - Learn a word embedding model

RDF2Vec

- A variant of node2vec and Deep Graph Kernel which originally support undirected graphs
- Pipeline:
  - Random walk over a KG for entity and relation sentences




```
Hamburg -> country -> Germany            -> leader     -> Angela_Merkel
Germany -> leader  -> Angela_Merkel      -> birthPlace -> Hamburg
Hamburg -> leader  -> Peter_Tschentscher -> residence  -> Hamburg
```

Ristoski, Petar, and Heiko Paulheim. "Rdf2vec: Rdf graph embeddings for data mining." *International semantic web conference*. Springer, Cham, 2016.

RDF2Vec

- Pipeline:
  - Random walk over a KG for entity and relation sentences
  - Learn a CBOW or Skip-gram model (recall Word2Vec) with the sentences

Ristoski, Petar, and Heiko Paulheim. "Rdf2vec: Rdf graph embeddings for data mining." *International semantic web conference*. Springer, Cham, 2016.

RDF2Vec

- Pipeline:
  - Random walk over a KG for entity and relation sentences
  - Learn a CBOW or Skip-gram model (recall Word2Vec) with the sentences

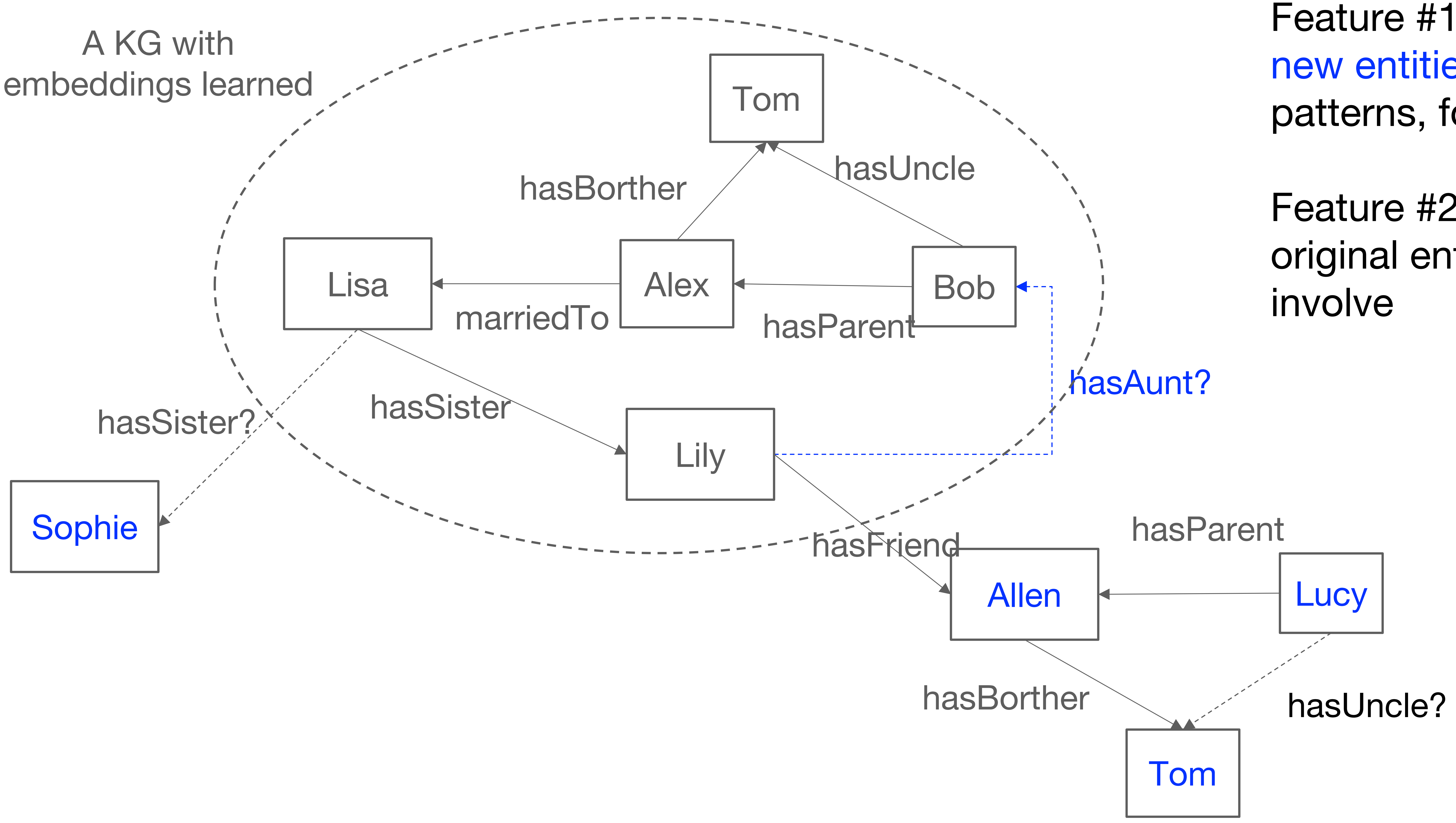
The sentences (walks) mainly keep the correlation between entities!

Ristoski, Petar, and Heiko Paulheim. "Rdf2vec: Rdf graph embeddings for data mining." *International semantic web conference*. Springer, Cham, 2016.

Day 3 Knowledge Graph Embeddings

Part II: Advanced Topics

Embedding for Inductive KG Inference



Feature #1: Learn representation of the new entities and relations, or their graph patterns, for link prediction

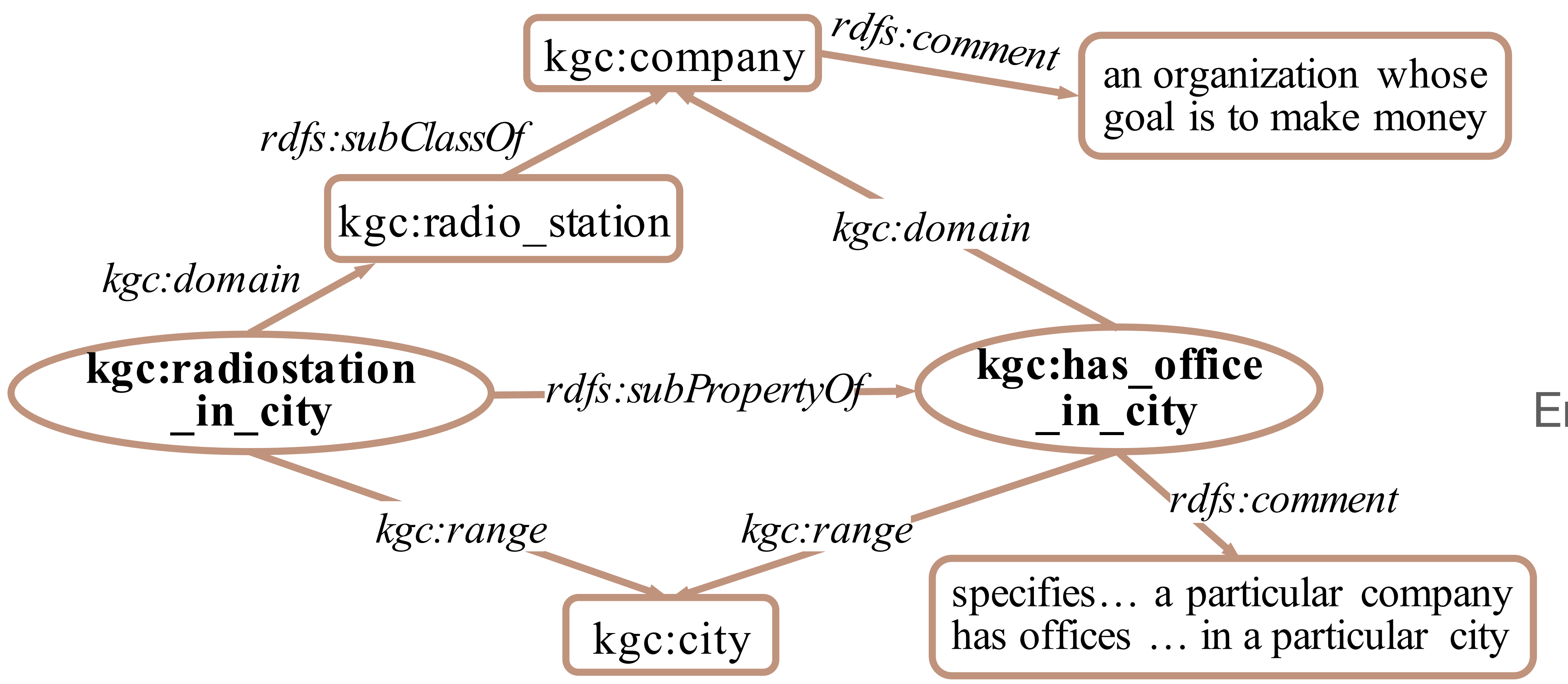
Feature #2: The embeddings of the original entities and relations will not involve

Solution #1: Utilizing Side Information

- The new relation hasAunt
  - Textual description: "An aunt is a woman who is a sibling of a parent or married to a sibling of a parent. Aunts who are related by birth are second-degree relatives. Alternate terms include auntie or aunty" (from Wikipedia)
  - Schema (a meta graph): domain (human), range (woman), the class hierarchies of the domain and range, super-property (hasRelative), etc.

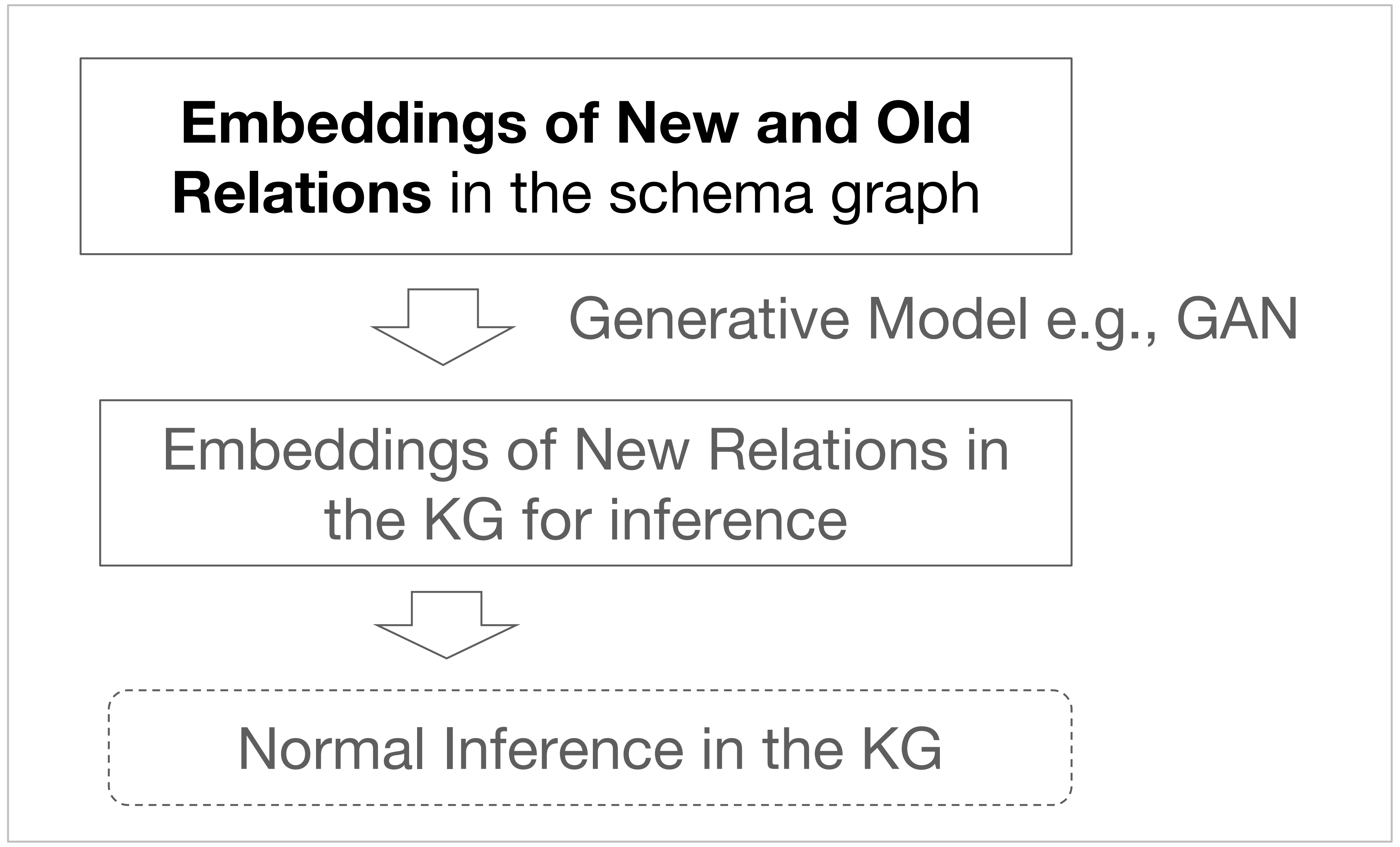
OntoZSL: Ontology Enhanced Zero-shot Learning

- Inductive KG inference for new relations with an ontological schema



Literal-aware schema graph

Embeddings

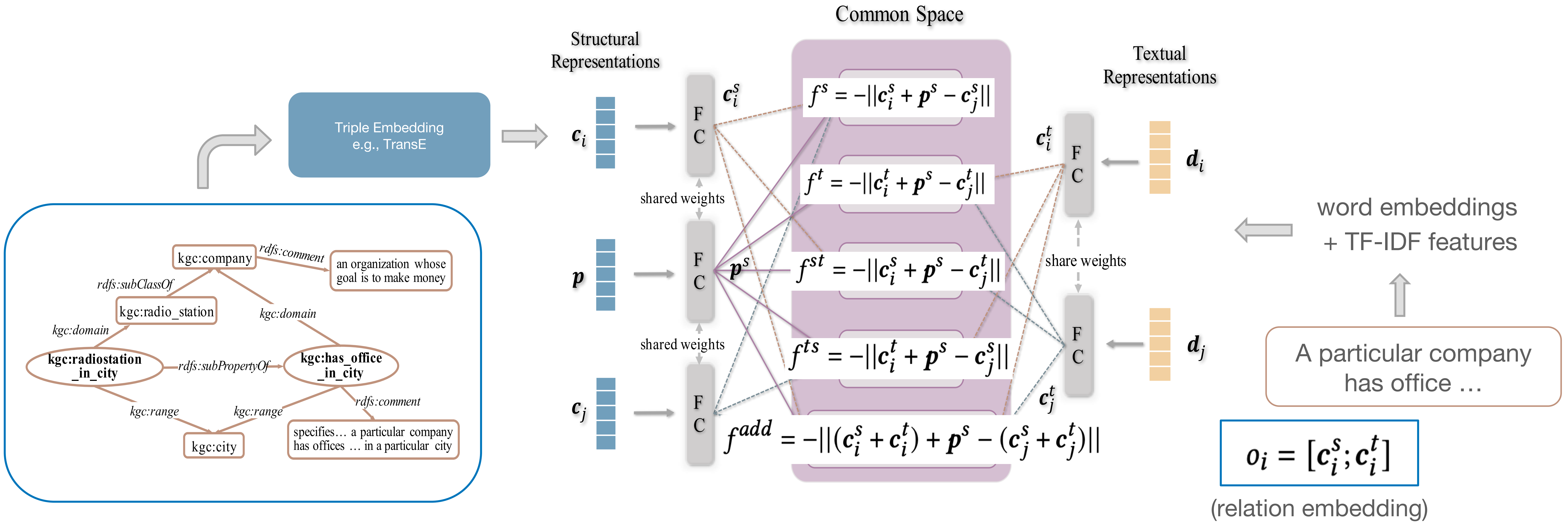


Generative Paradigm of ZSL

Geng, Yuxia, et al. "Ontozsl: Ontology-enhanced zero-shot learning." *Proceedings of the Web Conference 2021*.

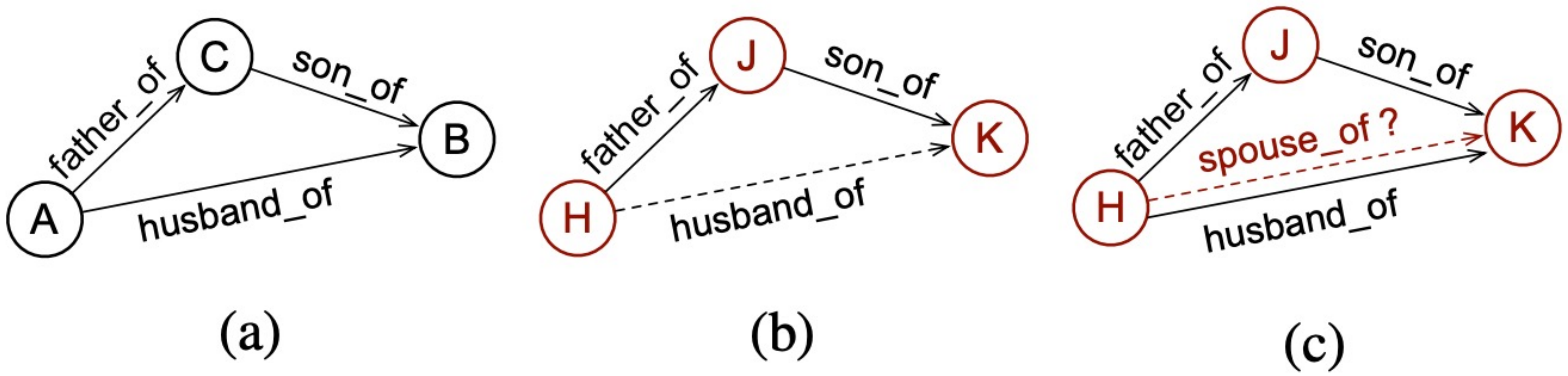
OntoZSL: Ontology Enhanced Zero-shot Learning

- Embedding the literal-aware schema graph



Solution #2: Utilizing the graph pattern

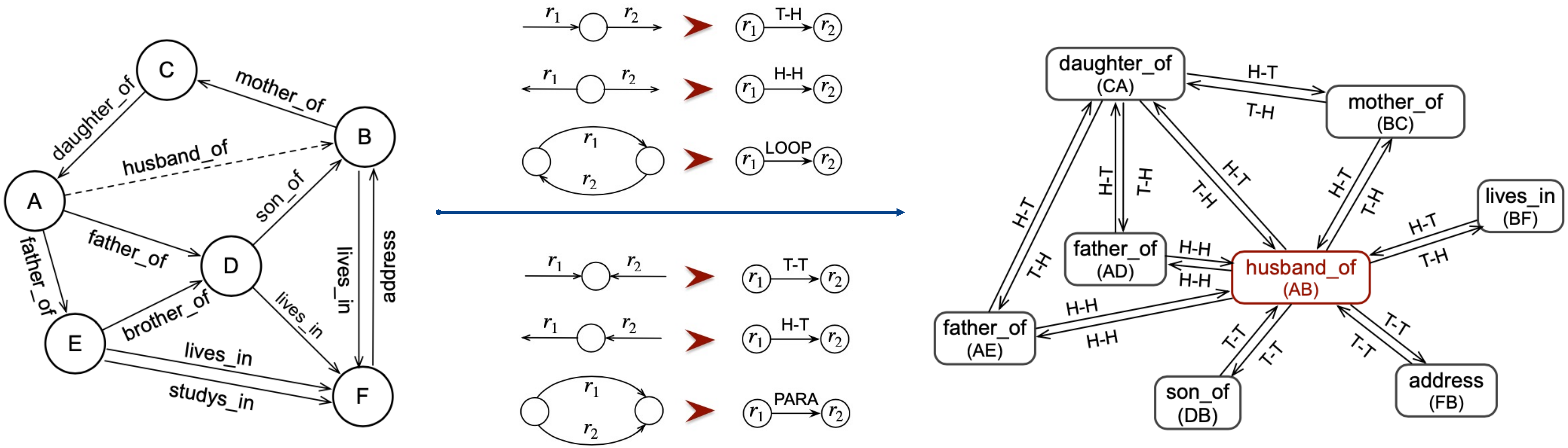
- RMPI: Relational message passing for fully inductive KG completion
  - A testing graph with both unseen entities and unseen relations (c)
  - Basic idea:
    - Learn graph patterns over **local subgraphs** with Graph Neural Networks (GNNs) in an entity-independent manner, i.e., in a view of relation



Geng, Yuxia, et al. "Relational message passing for fully inductive knowledge graph completion." 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE, 2023.

RMPI

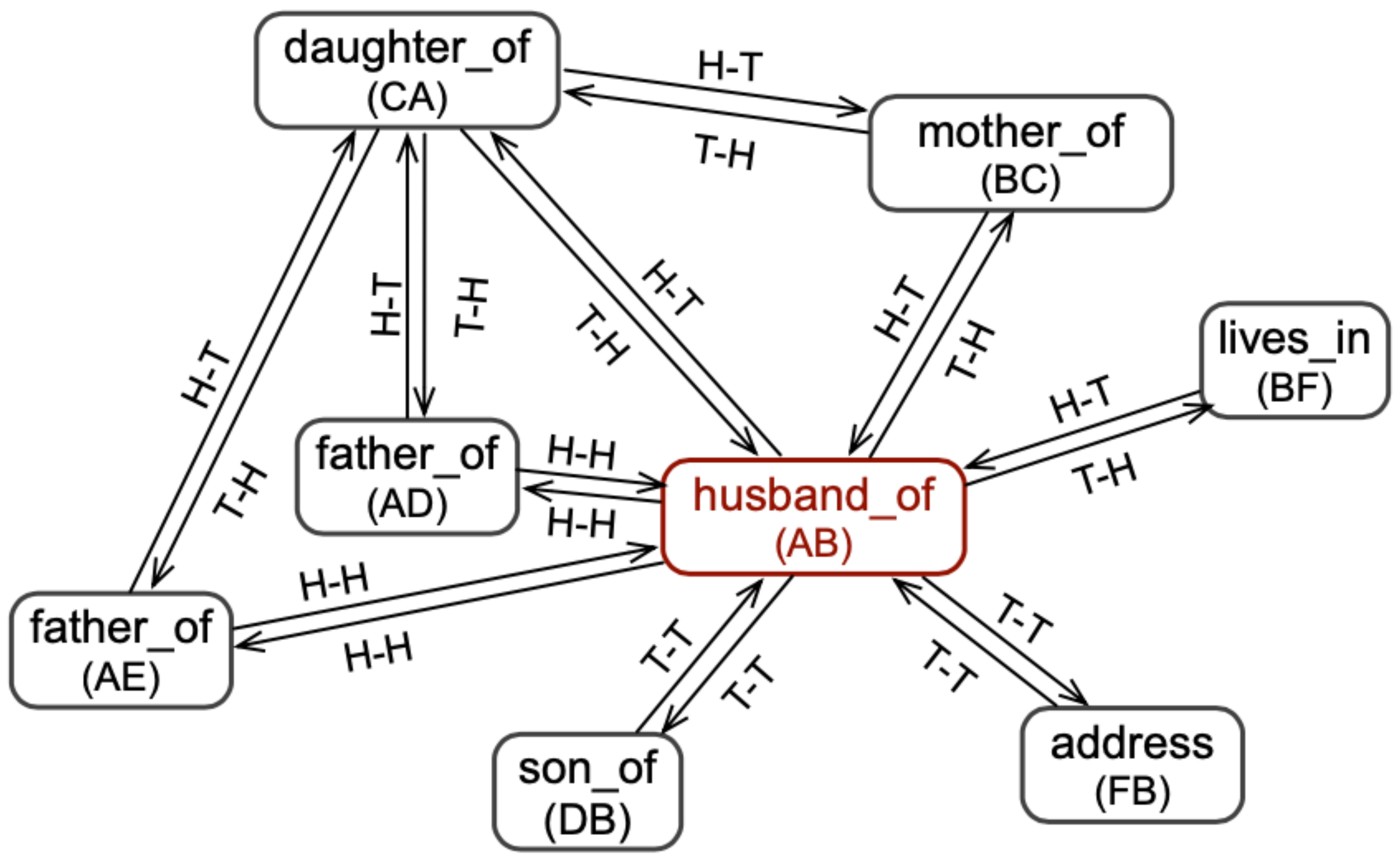
- Subgraph extraction and transformation

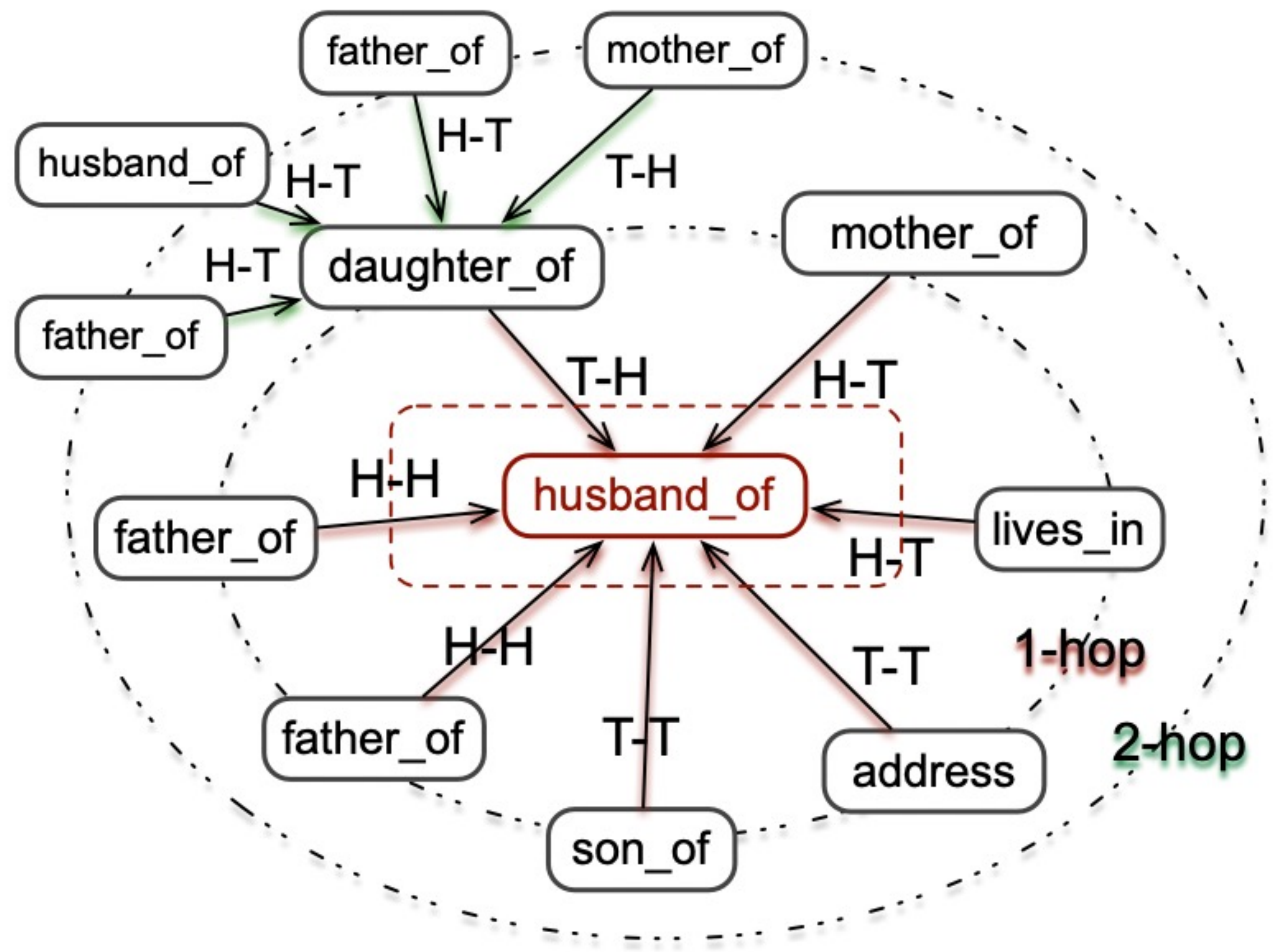


6 Meta Relations represent
connection patterns of relations in the original graph

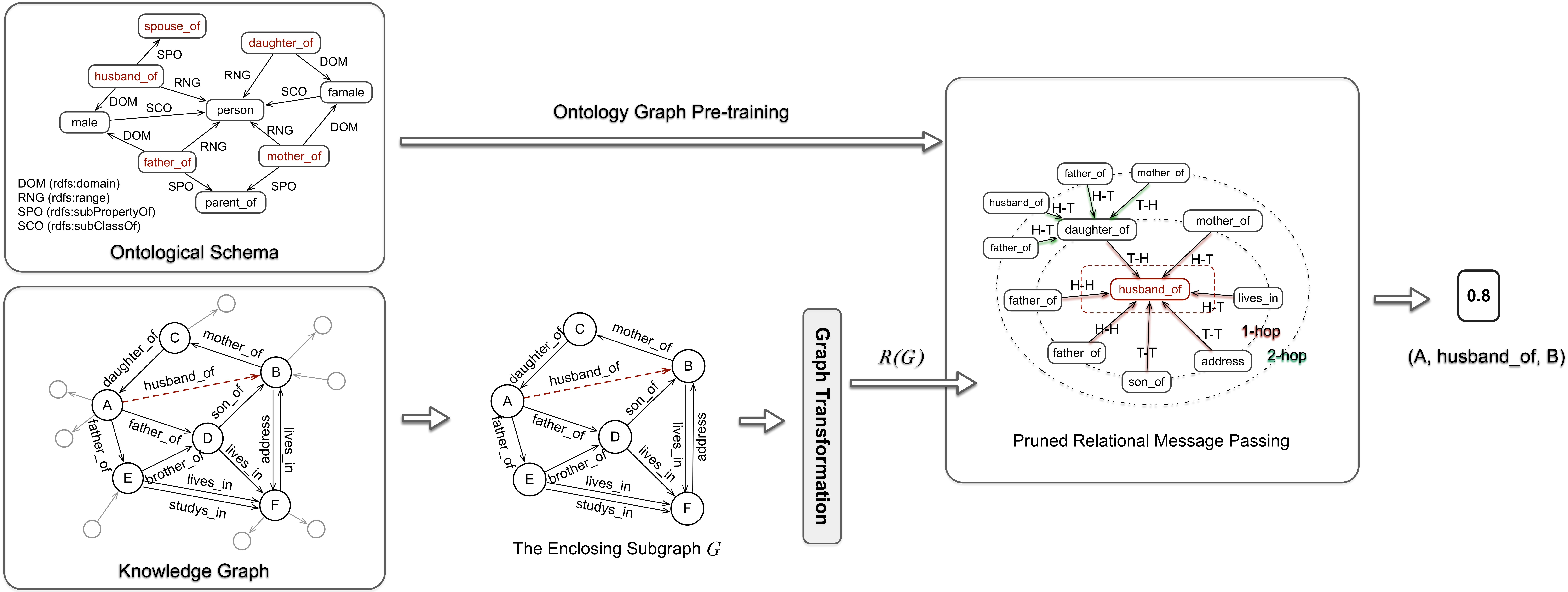
RMPI

- Graph pruning for optimization and prediction of target relation embedding by neighborhood aggregation (GNN)





RMPI (overall framework)



Incremental Learning of KG Embeddings

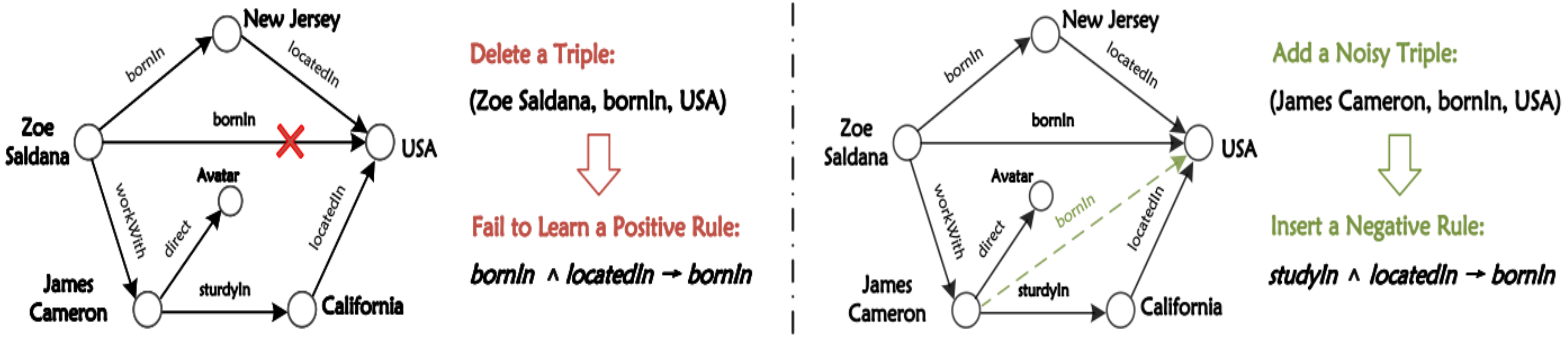
- Relatively less attention, but there are some works

Cui, Yuanning, et al., “Lifelong Embedding Learning and Transfer for Growing Knowledge Graphs”, AAAI 2023

- Challenges
  - Consider training efficiency (instead of re-training)
  - Detect what graph patterns are changed (similar to “Concept Drift” in stream learning).
  - Good testing performance on not only the new added part, but on the original part

Robustness of KG Embeddings

- Motivation: untargeted adversarial attack towards KG embeddings
  - Adversarial attack is to change the least number of facts for training that have the largest negative impact during testing
  - E.g., Horn rules learned from embeddings for getting the facts to attack



Zhao, Tianzhe, et al. "Untargeted Adversarial Attack on Knowledge Graph Embeddings." SIGIR 2024.

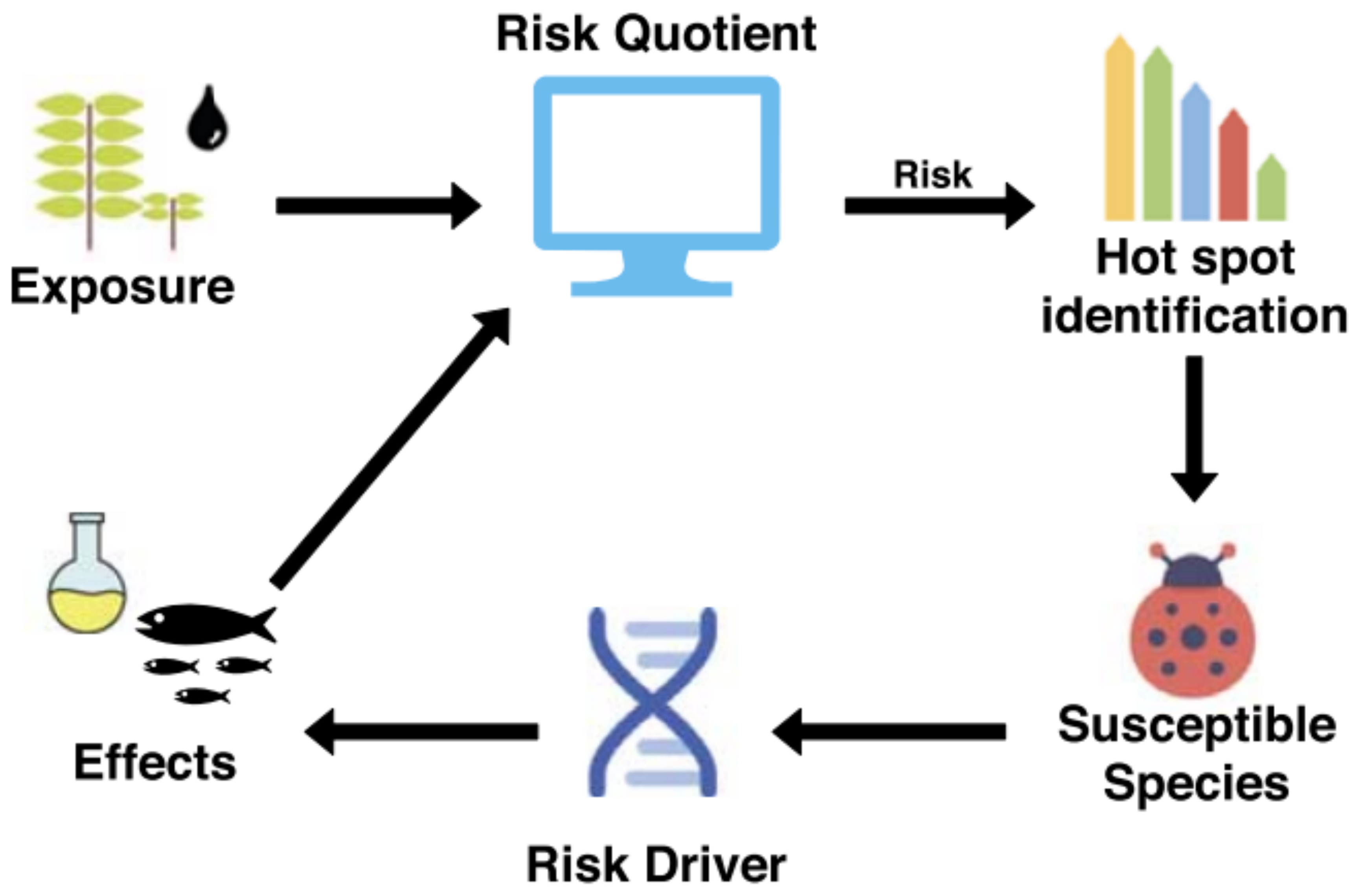
Other Advanced Topics

- Embedding KGs with schemas/rules/constraints
  - RMPI & OntoZSL belong to this type, but there are many more …

Zhang, Wen, et al. "Knowledge graph reasoning with logics and embeddings: Survey and perspective." *arXiv preprint arXiv:2202.07412* (2022).

Application of KG Embedding

- Ecotoxicological effect analysis

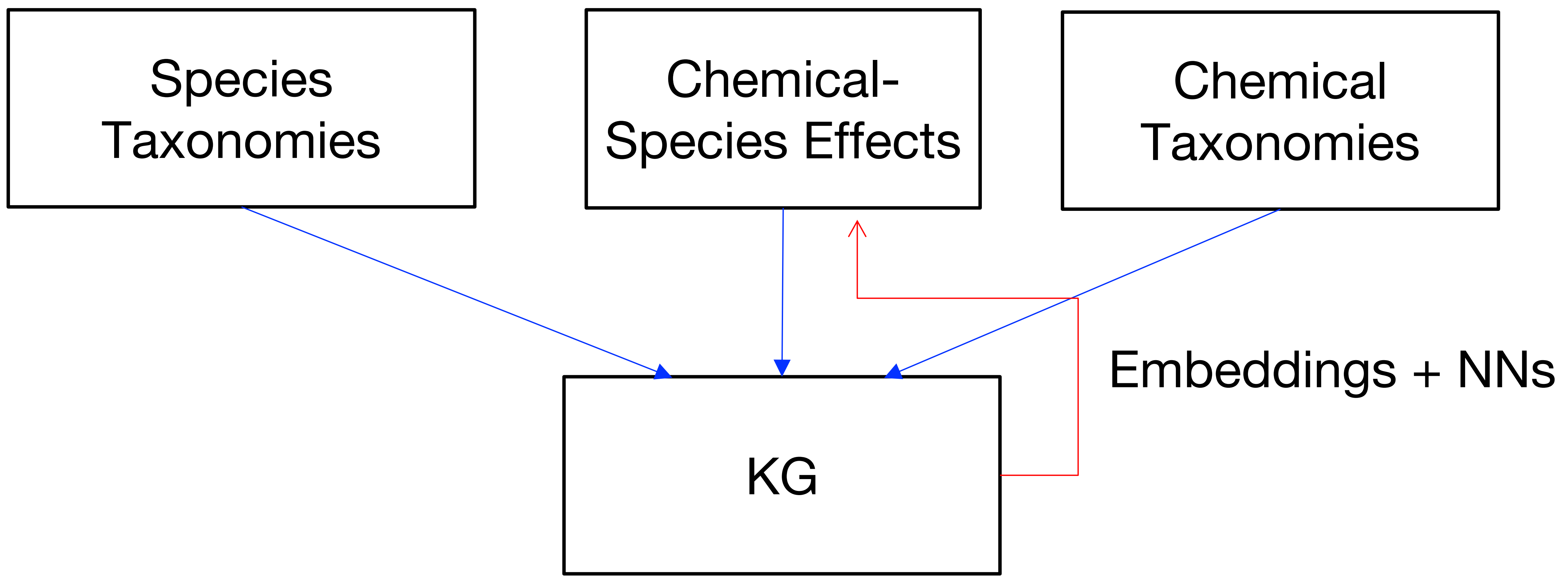


← Simplified ecotoxicological effect analysis pipeline with **experiments**

Myklebust, Erik B., et al. "Prediction of adverse biological effects of chemicals using knowledge graph embeddings." *Semantic Web* 13.3 (2022): 299-338.

Application of KG Embedding

- Ecotoxicological effect analysis

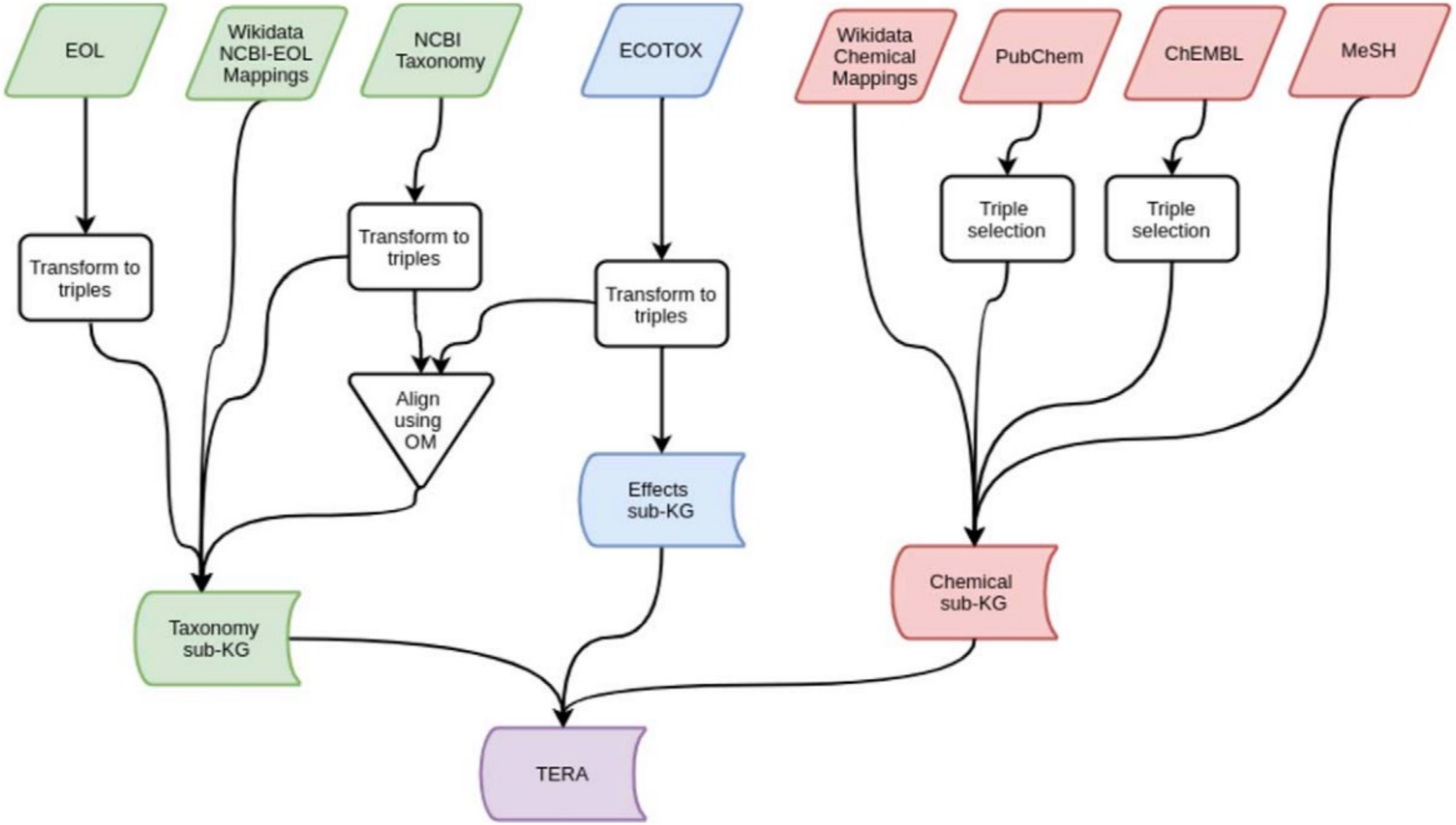


← Simplified idea of using KG embedding for ecotoxicological effect prediction

Application of KG Embedding

- Ecotoxicological effect analysis

Toxicological effect and risk assessment (TERA) KG construction →



Summary

- Knowledge Graph & Semantic embedding
  - One-hot, word embedding
- Knowledge graph embedding
  - Geometric modeling: TransE, TransH, TransR
  - GNNs: GCN, R-GCN
  - Sequence learning: RDF2Vec
- Advanced topics
  - Inductive inference: OntoZSL, RMPI; Incremental learning; Robustness
- Applications
  - Ecotoxicological effect analysis

The End of Day 3