

Note to other teachers and users of these slides: We would be delighted if you found our material useful for giving your own lectures. Feel free to use these slides verbatim, or to modify them to fit your own needs. If you make use of a significant portion of these slides in your own lecture, please include this message, or a link to our web site: <http://cs224w.Stanford.edu>

Stanford CS224W: Towards Foundation Models for Knowledge Graphs

CS224W: Machine Learning with Graphs
Charilaos Kanatsoulis and Jure Leskovec, Stanford
University

<http://cs224w.stanford.edu>



Announcements

- **Homework 3** due **today**
 - Gradescope submissions close at 11:59 PM
- **Colab 4** due Tuesday 12/2
- **Exam reminder!**
 - Wednesday 11/19 from 6-8PM
 - See Ed for your assigned room
 - If you're taking OAE/makeup exam, you should have received an email with details
 - Practice Exam (and solutions) posted on Ed
- **Colab 5** released today, due on Thursday 12/4

Stanford CS224W: Foundation Models

CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

<http://cs224w.stanford.edu>



Large Language Models



Large creative AI models will transform life and work. But how exactly do they function? Read more about the promise and peril of artificial intelligence here: <https://econ.trib.al/adS00Nj>

Illustration: George Wylesol



Generative AI

How does ChatGPT actually work?

Despite the feeling of magic, large language models (LLMs) are, in reality, a giant exercise in statistics

Language Models are Few-Shot Learners

Tom B. Brown*	Benjamin Mann*	Nick Ryder*	Melanie Subbiah*	
Jared Kaplan ¹	Prafulla Dhariwal	Arvind Neelakantan	Pranav Shyam	Girish Sastry
Amanda Askell	Sandhini Agarwal	Ariel Herbert-Voss	Gretchen Krueger	Tom Henighan
Rewon Child	Aditya Ramesh	Daniel M. Ziegler	Jeffrey Wu	Clemens Winter
Christopher Hesse	Mark Chen	Eric Sigler	Mateusz Litwin	Scott Gray
Benjamin Chess	Jack Clark	Christopher Berner		
Sam McCandlish	Alec Radford	Ilya Sutskever	Dario Amodei	

OpenAI

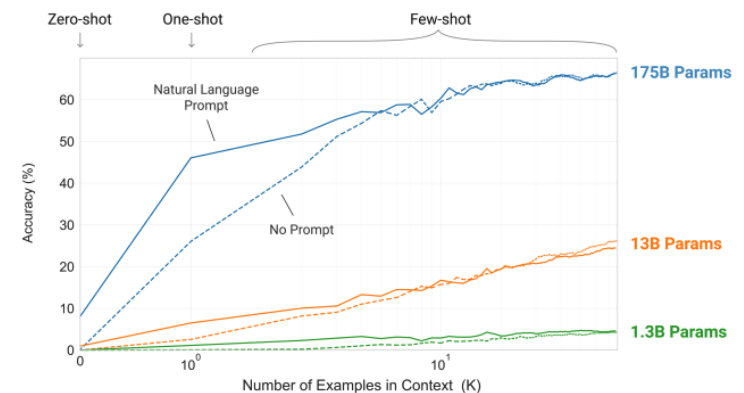


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper “in-context learning curves” for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

Foundation Models for Comp. Vision



Figure 1. A building foundation — **Image: DALLE 2**

Source: https://medium.com/@tenyks_blogger/the-foundation-models-resaping-computer-vision-b299a91527fb

Relational Foundation Model

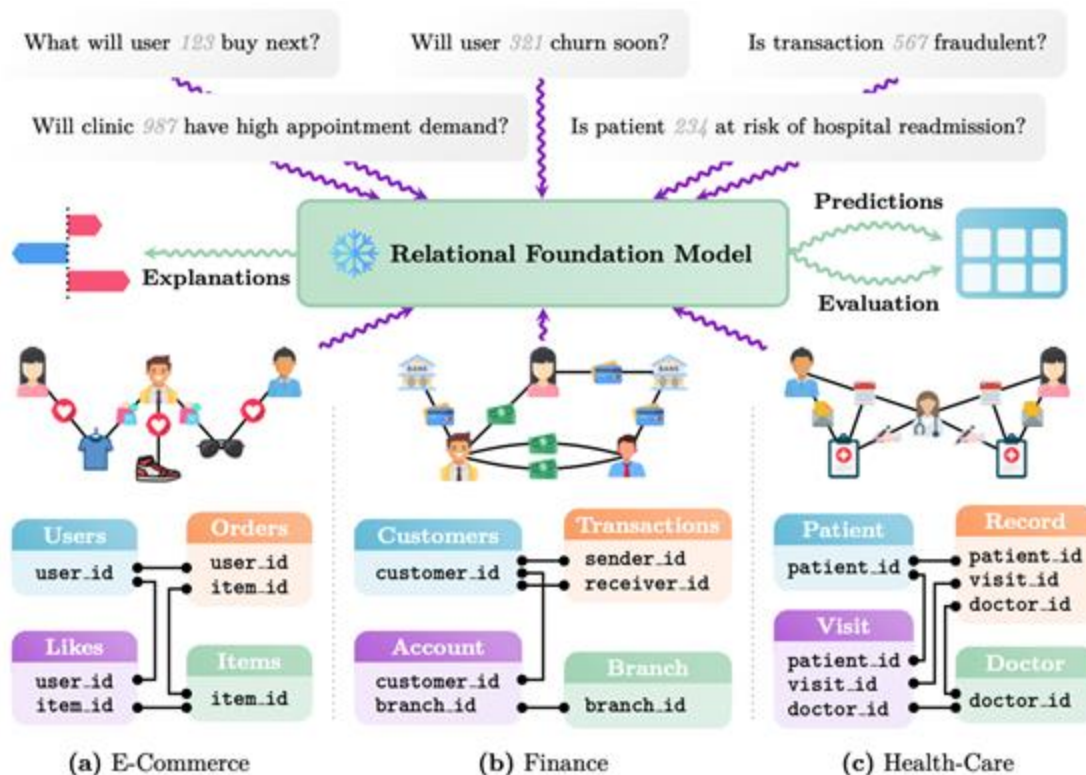


Figure 1: Key capabilities of Relational Foundation Models. RFMs can be applied to new/unseen databases and schemas with highly varying structural characteristics, as found in (a) e-commerce, (b) finance, or (c) health care. Secondly, they can be applied to any predictive task type, ranging from one-off assessments (e.g. entity-level fraud prediction) to temporal predictive queries (e.g. temporal recommendation prediction). Thirdly, they generalize to new predictive tasks and give accurate predictions without any task-specific model tuning. Finally, not only do RFMs support prediction outputs, but they also offer insights into the reasoning processes via explanations, and build trust through extensive quantitative evaluation mechanisms.

Foundation Models for Biology

Search icon

U.S. INTERNATIONAL CANADA LIVING WORLD

SUBSCRIBE FOR \$1 WEEK Account

Monday, March 19, 2024 Today's Paper

U.S. World Business Arts Lifestyle Opinion Audio Games Cooking Wine & Spirits The Athletic

Netanyahu and Biden Escalate Public Feud Over Gaza War

President Biden said Benjamin Netanyahu's military strategy was "hurting Israel more than helping Israel." Mr. Netanyahu dismissed the comments as "wrong."

See more headlines

The United Arab Emirates has maintained its links to Israel throughout the war, but the relationship is under pressure.

8 MIN READ

An Army ship has set sail to help build a pier to deliver aid into Gaza, the U.S. military said.

2 MIN READ

Elon Musk Has a Giant Charity. Its Money Stays Close to Home.

After making billions in tax-deductible donations, Mr. Musk gave away far less than required in some years — and what he did give often supported his own interests.

12 MIN READ

Behind Our Reporting

3:12

Opinion

W.J. HENNINGAN
1 Person, 1 Button, 15 Minutes: Absolute Authority to End the World

ZEBA KLEIN
Fine, Call It a Comeback

DAVID FREDERICK
Why Haley Voters Should Support Biden

PETER WERNER
If There's One Thing Trump Is Right About, It's Republicans

DANIELA J. LAMAS
As a Doctor, I Don't Fear Covid as Once Did. But I Carry Its Grave Lessons Forward.

ROSS PERLIN
America Has No Official Language. Instead It Has Hundreds.

JACOB POLL
There Is a Secret Hamiltonian in the White House

NICHOLAS KRATZOFF
'People Are Hoping That Israel Nukes Us So We Get Rid of This Pain'

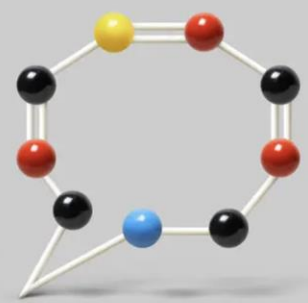
LETTERS FROM OUR READERS
Listening to the

CHRISTOPHER CALDWELL
This Prophetic Academic

A.I. Is Learning What It Means to Be Alive

Given troves of data about biology, A.I. models have made surprising discoveries. What could they teach us someday?

Doug Chayka

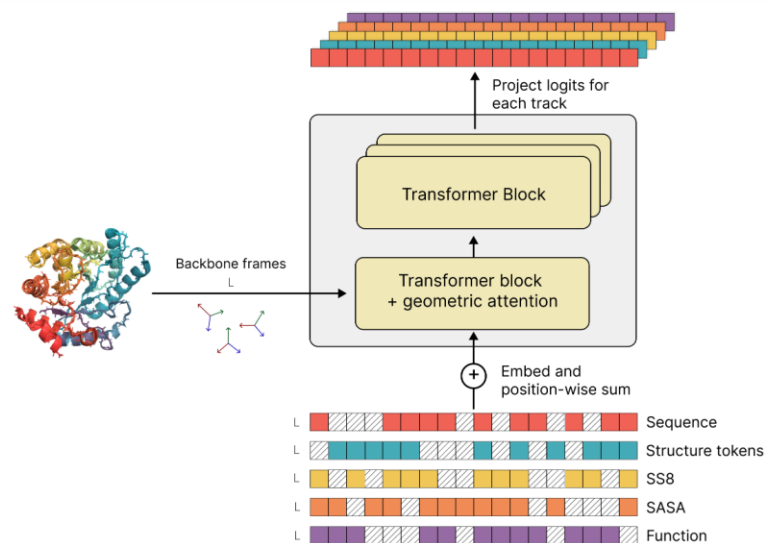


IDEAS

A.I. Is Learning What It Means to Be Alive

Given troves of data about biology, A.I. models have made surprising discoveries. What could they teach us someday?

Doug Chayka



Source: <https://github.com/evolutionaryscale/esm>

Foundation Models for Cell Biology

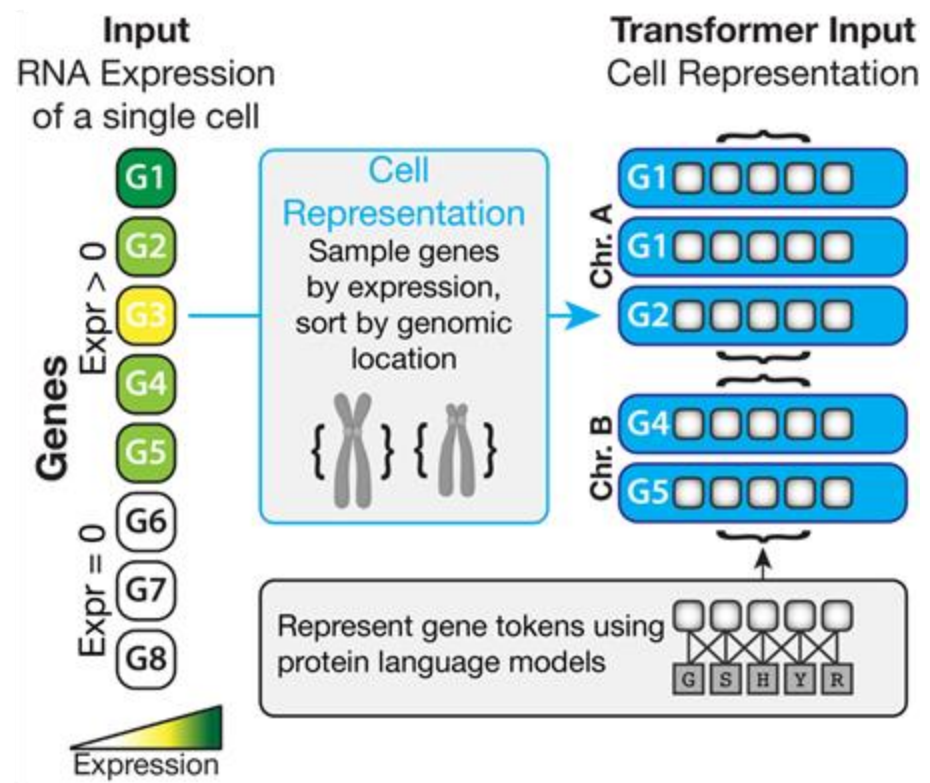
- UCE creates universal representations of cells

Input:

RNA expression of a single cell/nucleus

Output:

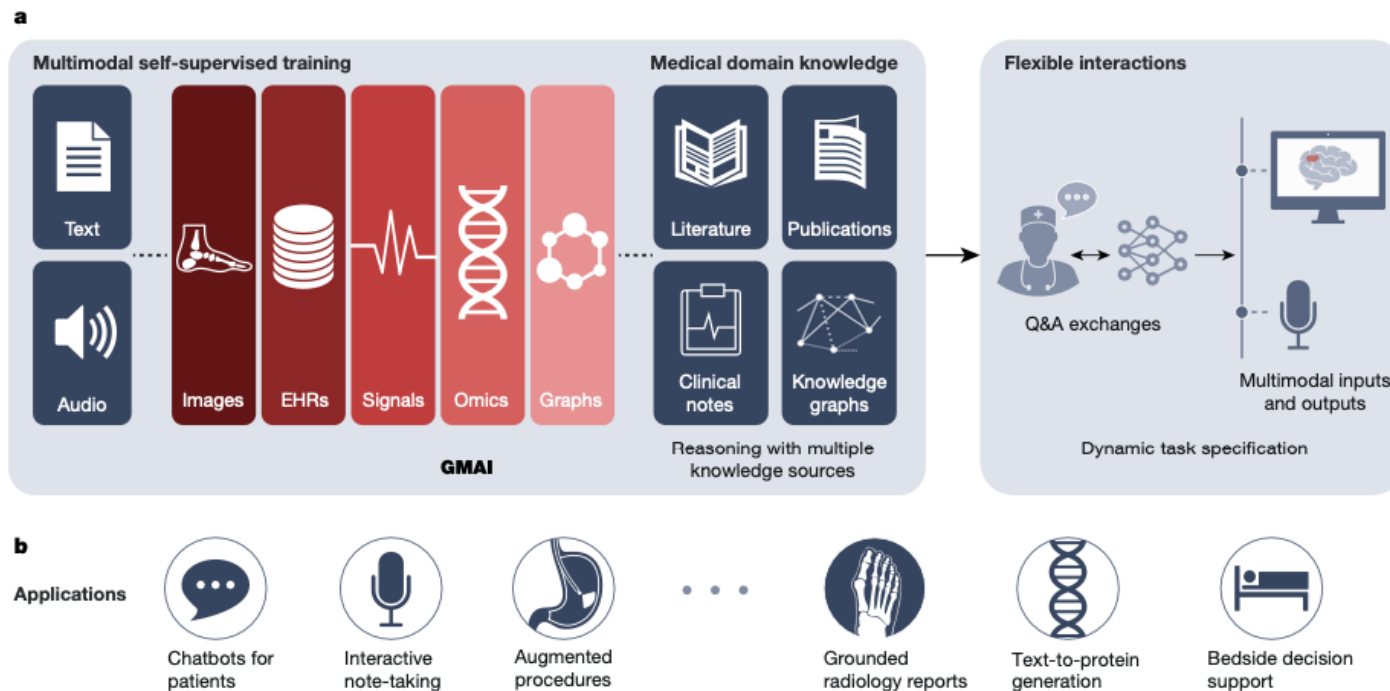
Cell Embedding



Foundation Models for Medicine

Perspective

Foundation models for generalist medical artificial intelligence



Regulations: Application approval; validation; audits; community-based challenges; analyses of biases, fairness and diversity

Fig. 1 | Overview of a GMAI model pipeline. a, A GMAI model is trained on multiple medical data modalities, through techniques such as self-supervised learning. To enable flexible interactions, data modalities such as images or data from EHRs can be paired with language, either in the form of text or speech data. Next, the GMAI model needs to access various sources of medical knowledge to carry out medical reasoning tasks, unlocking a wealth of capabilities that can be used in downstream applications. The resulting GMAI model then carries

out tasks that the user can specify in real time. For this, the GMAI model can retrieve contextual information from sources such as knowledge graphs or databases, leveraging formal medical knowledge to reason about previously unseen tasks. **b,** The GMAI model builds the foundation for numerous applications across clinical disciplines, each requiring careful validation and regulatory assessment.

Foundation Models

define a foundation model



A **foundation model** is a large-scale machine learning model trained on broad data (often unstructured and diverse) that serves as a general-purpose model across multiple tasks. These models, such as GPT-3, BERT, or CLIP, are typically characterized by:

1. **Pretraining:** They are pretrained on extensive datasets using self-supervised or unsupervised learning to learn a wide range of representations and capabilities.
2. **Scalability:** Foundation models leverage large architectures (e.g., deep neural networks) with billions or even trillions of parameters.
3. **Adaptability:** Once pretrained, these models can be fine-tuned or adapted to perform a variety of downstream tasks, such as text generation, image recognition, or translation, with relatively small amounts of task-specific data.
4. **Transfer Learning:** The knowledge learned during pretraining can be transferred effectively to different domains, enabling efficient learning of new tasks.

Foundation models provide a basis for a variety of applications and are central to advances in AI, often serving as the backbone of many AI systems.

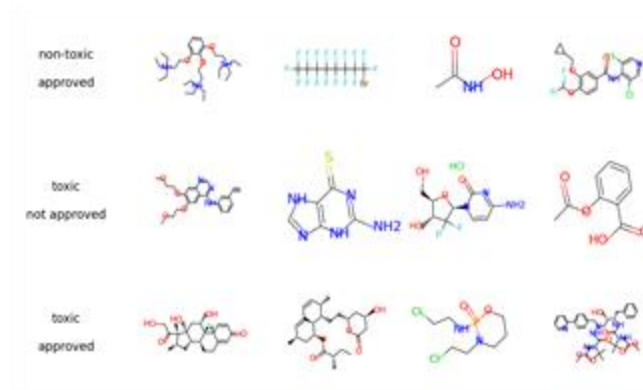
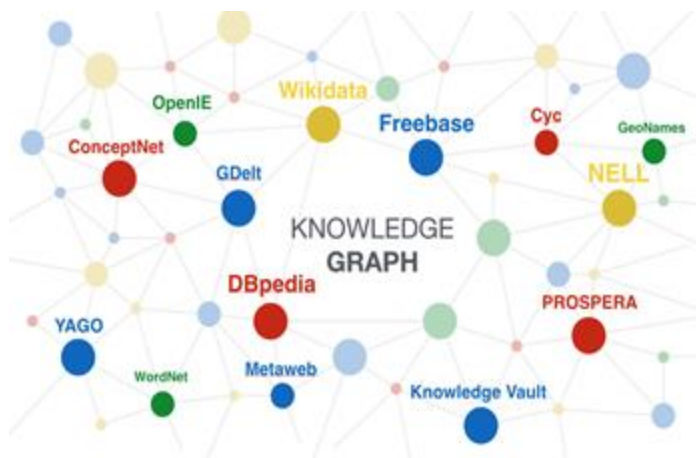
Foundation Models Characteristics

- Foundation models are pretrained architectures
- Modern AI trains very large models with a huge amount of data
- Foundation models can be used as zero-shot or few-shot learners
- They generalize to new entities that have not been observed during training

Graph Foundation Models

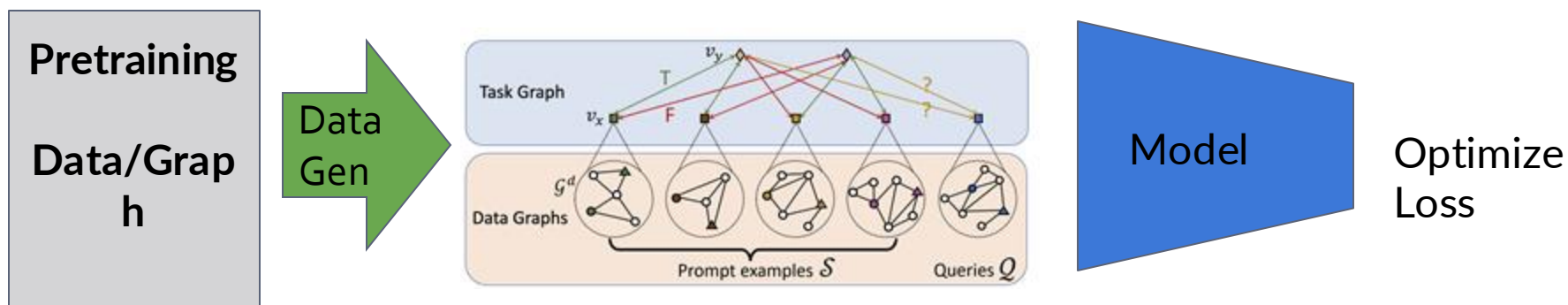
Challenges

- Multiple types of graphs
- Multiple types of tasks



Recap: PRODIGY

- Unified framework to learn graph tasks of multiple levels
- Agnostic of the graph type



Towards Foundation Models for KGs

- Build a model that **learns embeddings** for **entities** and **relations** of **any KG**
- The model will be trained in an **unsupervised fashion**
- The model will **transfer knowledge** between different KGs
- We **will not discuss** scalability of the training data and the model

Stanford CS224W: Knowledge Graphs

CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

<http://cs224w.stanford.edu>

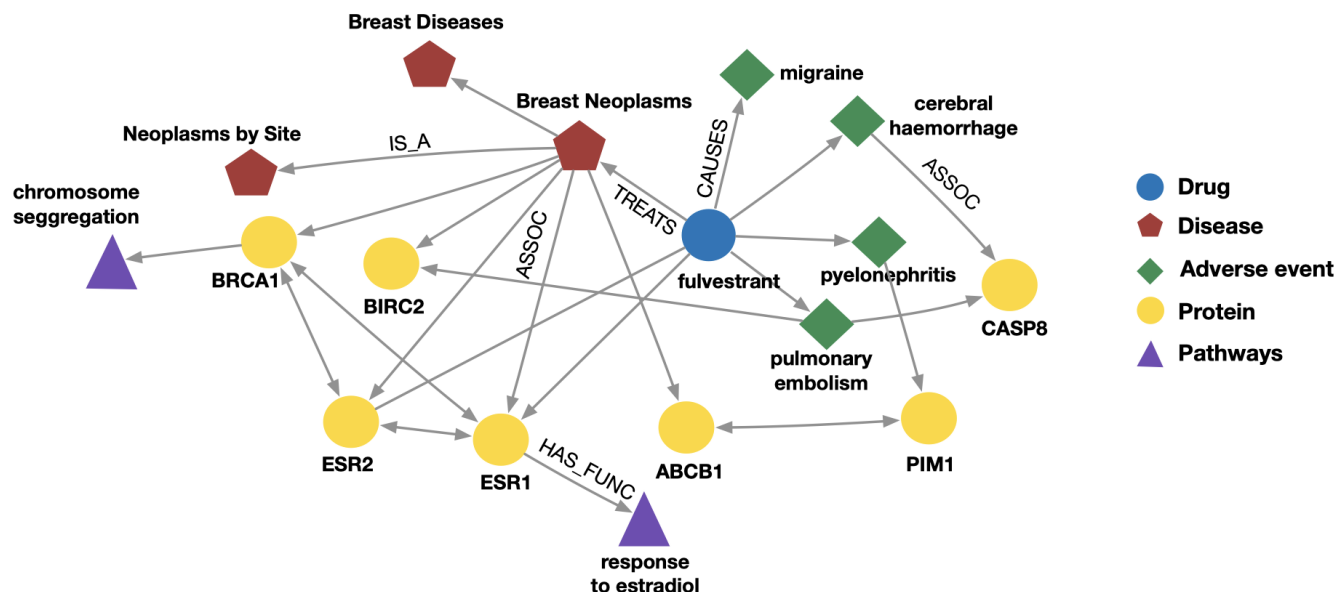


KG Representation

- Edges in KG are represented as **triples** (h, r, t)
 - **head** (h) has **relation** (r) with **tail** (t)
- **Key Idea:**
 - Model entities and relations in embedding space \mathbb{R}^k
 - Associate entities and relations with **shallow embeddings**
 - **Note we do not learn a GNN here!**
 - Given a triple (h, r, t) , the goal is that the **embedding of (h, r) should be close** to the **embedding of t** .
 - How to embed (h, r) ?
 - How to define score $f_r(h, t)$?
 - Score f_r is high if (h, r, t) exists, else f_r is low

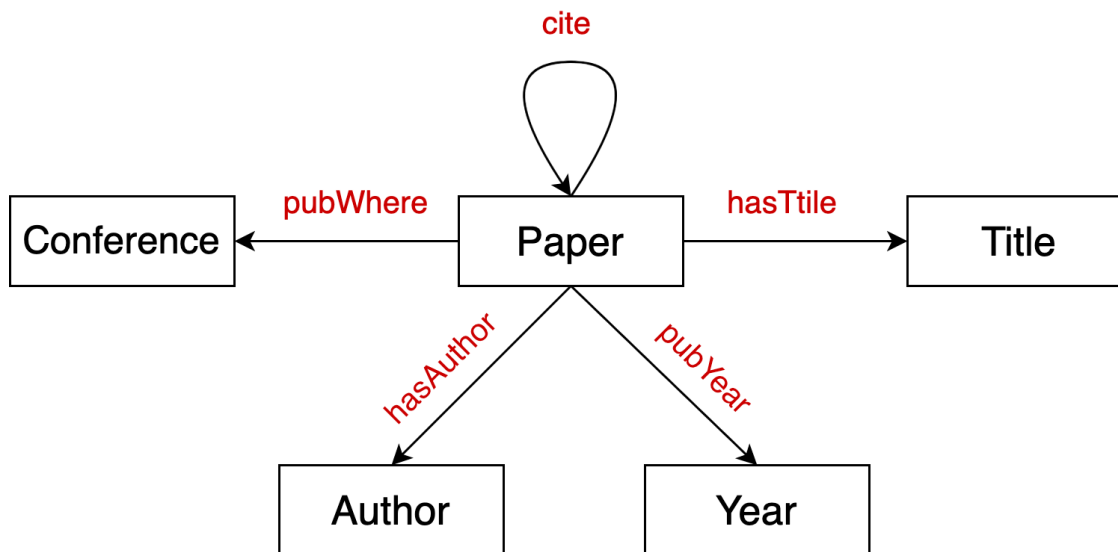
Example: Bio Knowledge Graphs

- **Node types:** drug, disease, adverse event, protein, pathways
- **Relation types:** has_func, causes, assoc, treats, is_a



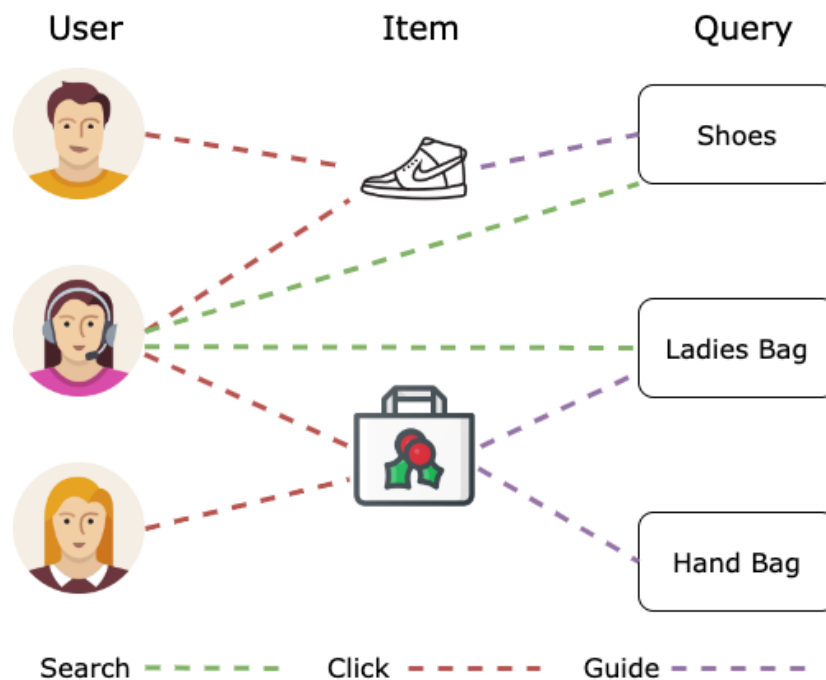
Example: Bibliographic Networks

- **Node types:** paper, title, author, conference, year
- **Relation types:** pubWhere, pubYear, hasTitle, hasAuthor, cite



Example: E-Commerce Networks

- Example: E-Commerce Graph
 - **Node types:** User, Item, Query, Location, ...
 - **Edge types:** Purchase, Visit, Guide, Search, ...
 - Different node type's features spaces can be different!



KG vs Natural Language

- Example: E-Commerce Graph
 - **Natural Language:**
 - User searches for shoes, User buys hand-bag

KG vs Natural Language

- Example: E-Commerce Graph

- **Natural Language:**

- User searches for shoes User buys hand-bag

- **All entities are words or tokens of the same type**

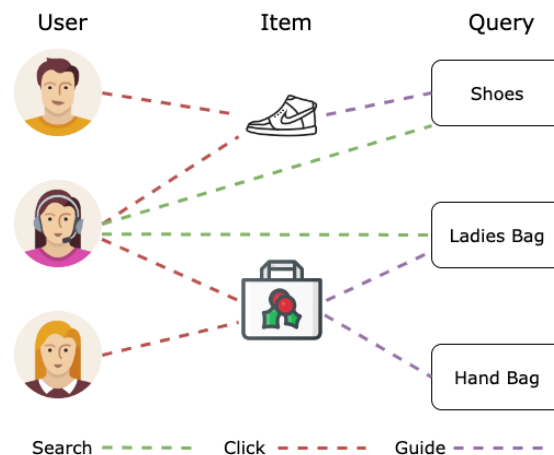
- **Homogeneous vocabulary**

- **Knowledge Graph:**

- User, shoes, hand-bag → entities
 - Searches, buys → relations
 - **Two modalities, heterogeneous vocabulary**

KG vs Natural Language

- Example: E-Commerce Graph
 - **Natural Language:**
 - User searches for shoes User buys hand-bag
 - **Tokens are connected sequentially**
 - **Knowledge Graph:**
 - **Entities have an underlying complex graph structure**



Towards Foundation Models for KGs

- Build a model that **learns embeddings** for **entities** and **relations** of **any KG**
 - **Double equivariant architectures**
- Process the graph structure between entities
 - **Use Graph Neural Networks**
- Find a canonical task for unsupervised training
 - **Knowledge Graph Completion**

Stanford CS224W: Transductive Link prediction

CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

<http://cs224w.stanford.edu>

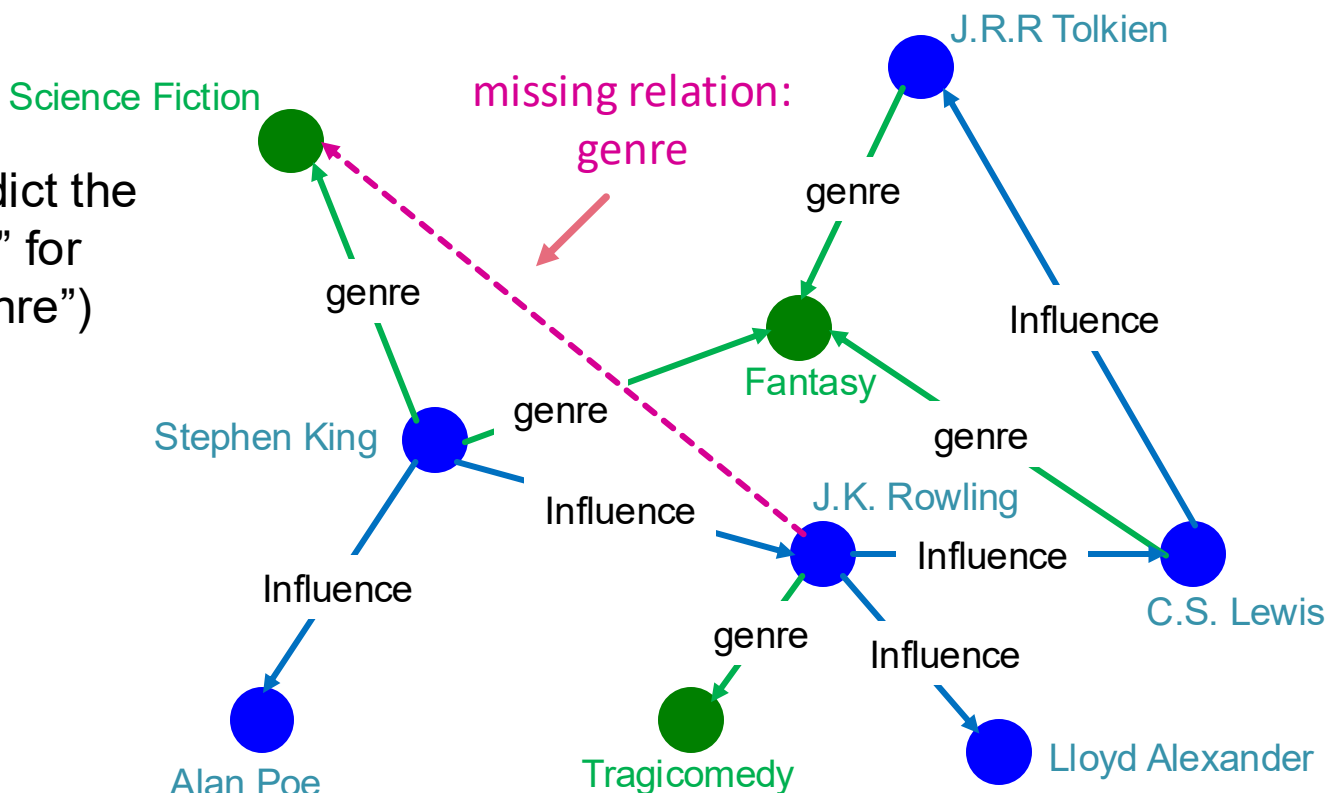


Recap: KG Completion Task

Given an enormous KG, can we complete the KG?

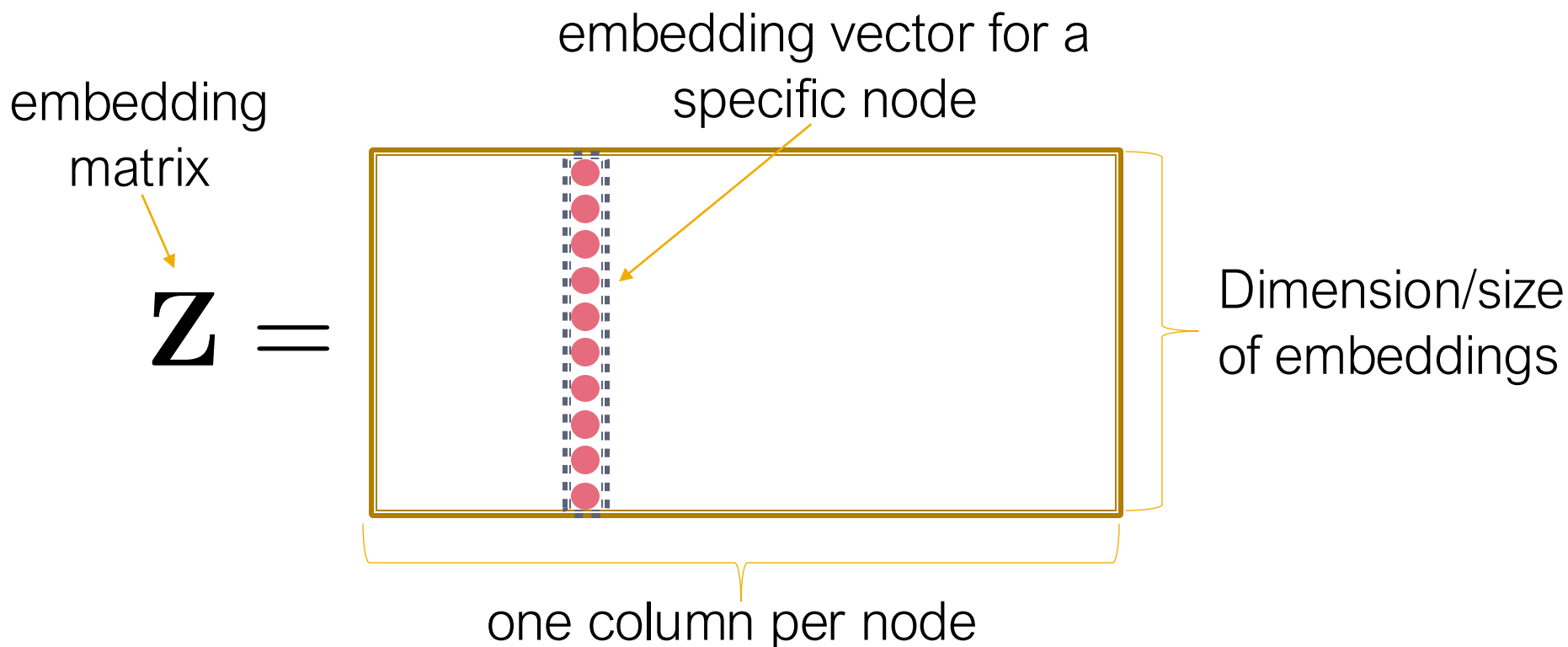
- For a given (**head**, **relation**), we predict missing **tails**.

Example task: predict the tail “Science Fiction” for (“J.K. Rowling”, “genre”)



Recap: “Shallow” Encoding

- Simplest encoding approach: **encoder is just an embedding-lookup**



Recap: KG Completion Models

KG embedding Models:

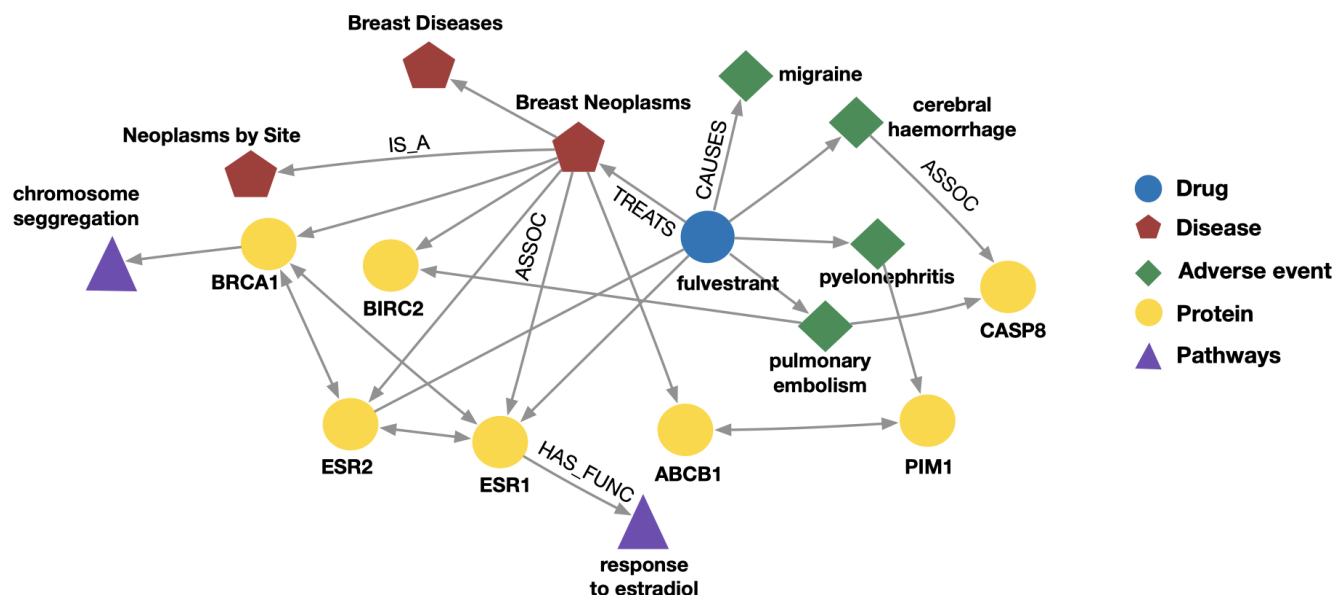
Model	Score	Embedding	Sym.	Antisym.	Inv.	Compos.	1-to-N
TransE	$-\ \mathbf{h} + \mathbf{r} - \mathbf{t}\ $	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^k$	✗	✓	✓	✓	✗
TransR	$-\ M_r \mathbf{h} + \mathbf{r} - M_r \mathbf{t}\ $	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k,$ $\mathbf{r} \in \mathbb{R}^d,$ $M_r \in \mathbb{R}^{d \times k}$	✓	✓	✓	✓	✓
DistMult	$\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^k$	✓	✗	✗	✗	✓
Complex	$\text{Re}(\langle \mathbf{h}, \mathbf{r}, \bar{\mathbf{t}} \rangle)$	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{C}^k$	✓	✓	✓	✗	✓
RotateE	$-\ \mathbf{h} \circ \mathbf{r} - \mathbf{t}\ $	$\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{C}^k$	✓	✓	✓	✓	✗

KG embedding methods

- Structure agnostic shallow encoders **learn embeddings** for **entities** and **relations**
 - The entity and relation embeddings are the trainable weights
- The structure of the KG is utilized in the loss function
- KG embedding methods can only be used on a single KG
 - They cannot transfer knowledge between KGs

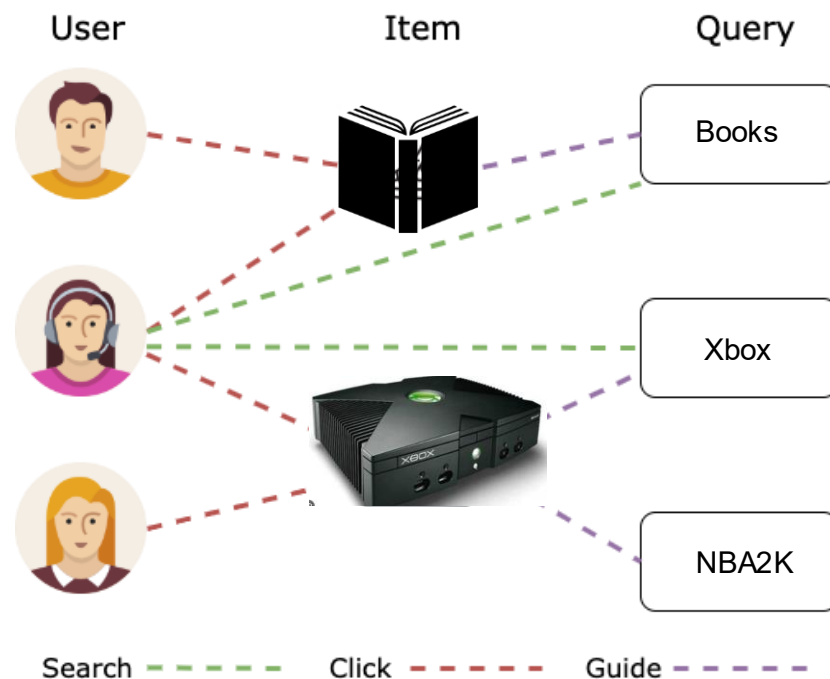
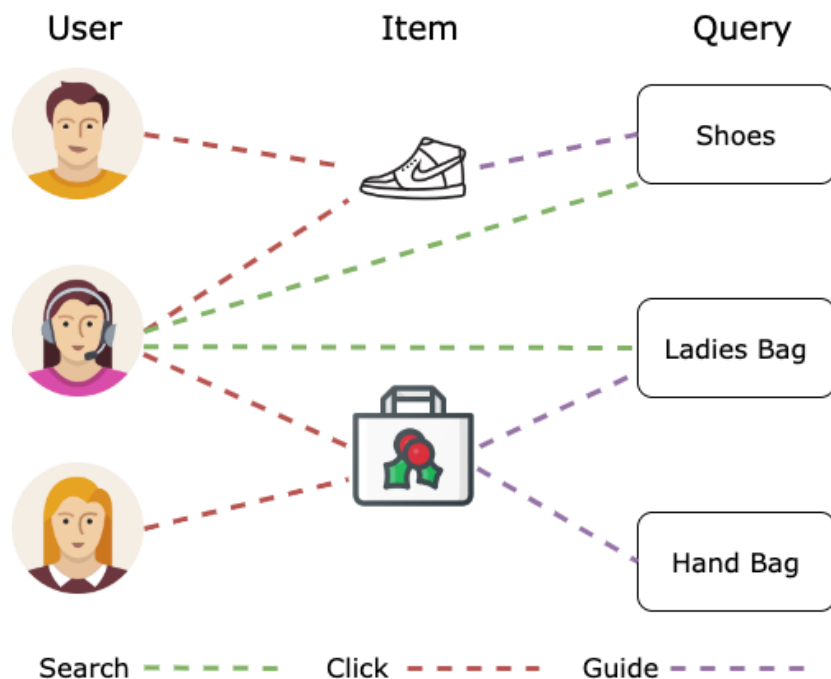
New entities in Bio KGs

- What if a new drug comes in?
- Need to retrain the shallow encoder



Generalize across similar KGs

- What if I want to transfer knowledge from one KG to another?
- The KGs have the same relations



Stanford CS224W: Inductive Link prediction

CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

<http://cs224w.stanford.edu>



Generalize across similar KGs

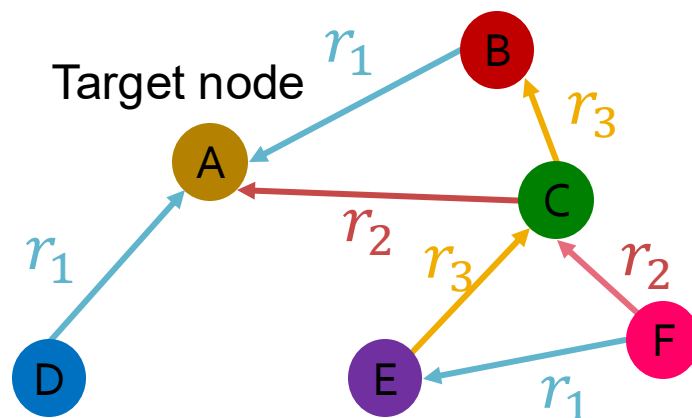
- What if I want to transfer knowledge from one KG to another?
- The KGs have the same relations

Solution:

- Observe that a **KG** is a **heterogeneous graph**
- Use a **heterogenous GNN**
 - E.g., RGCN, Heterogenous Graph Transformer

Relational GCN

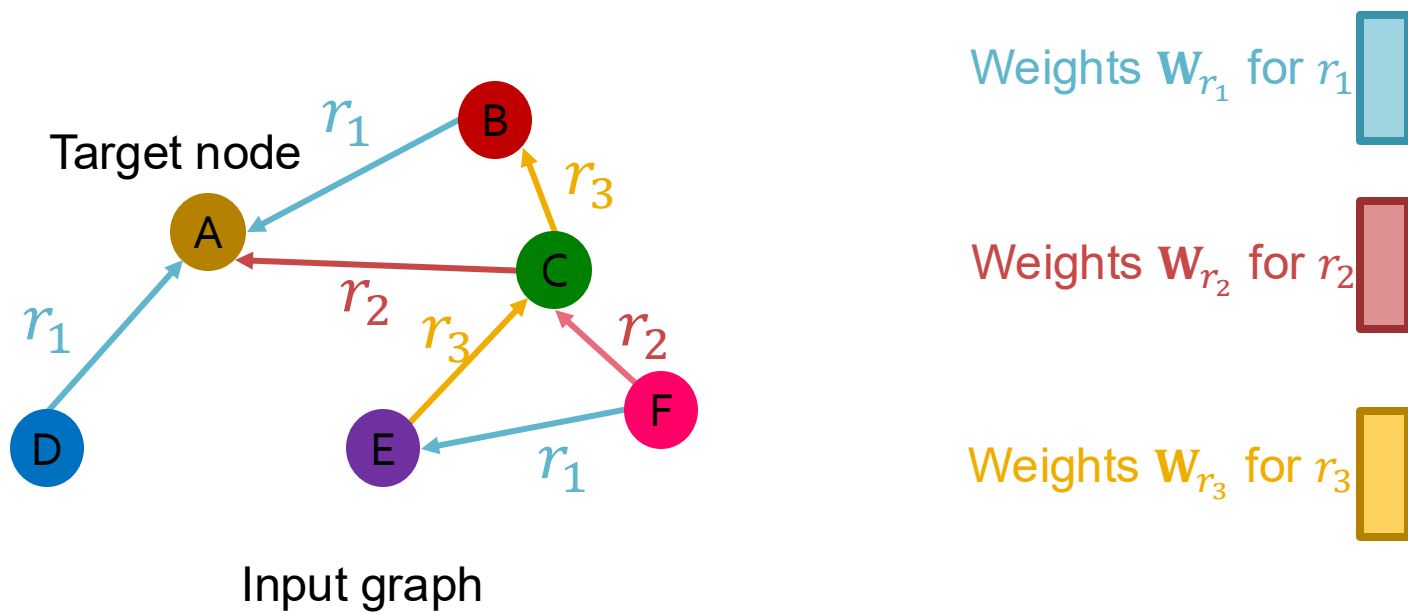
- What if the graph has **multiple relation types**?



Input graph

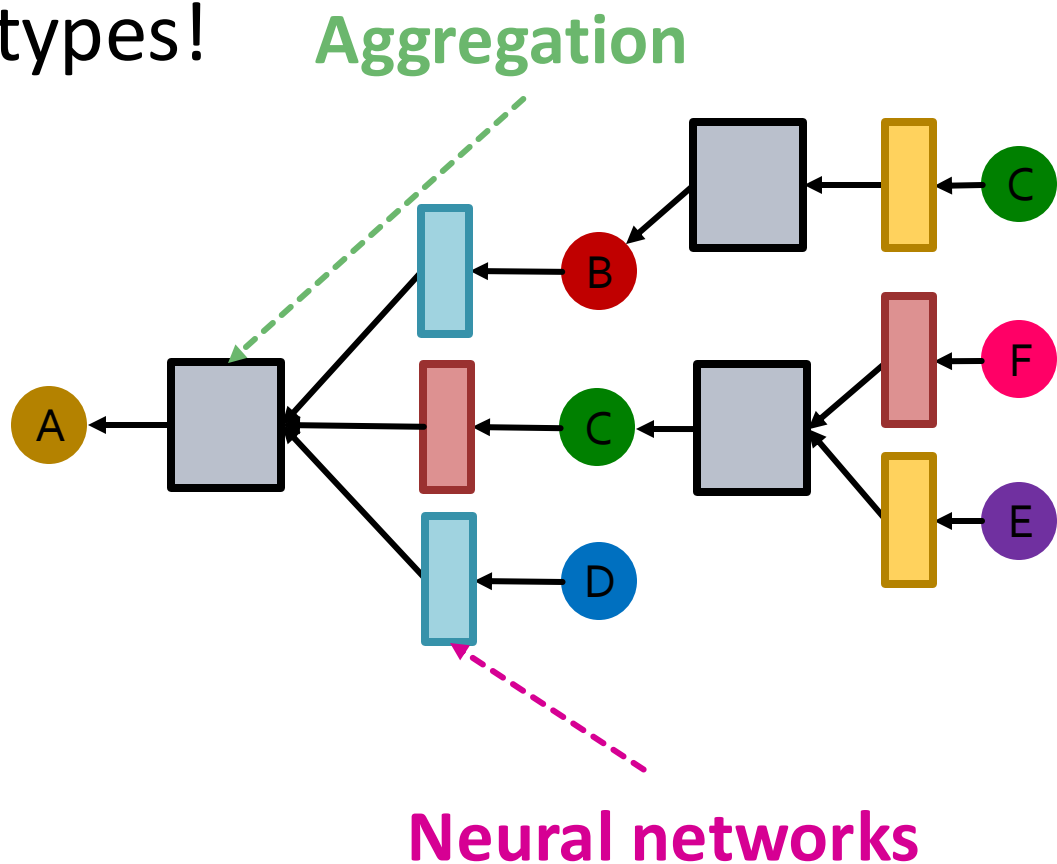
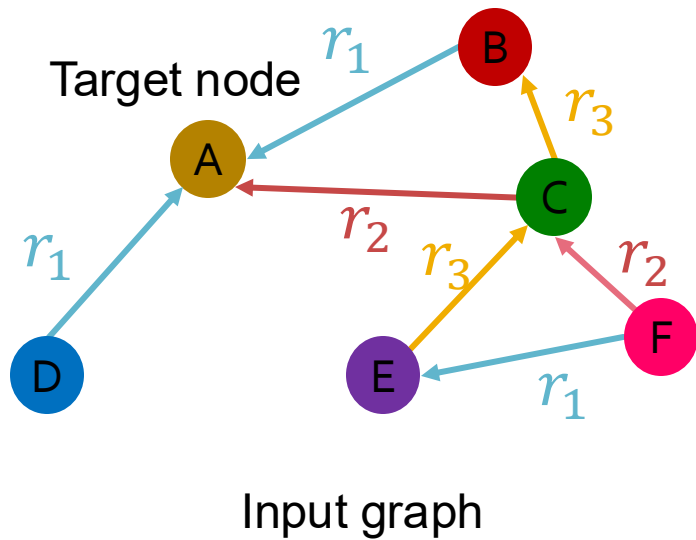
Relational GCN (2)

- What if the graph has **multiple relation types**?
- Use different neural network weights for different relation types.



Relational GCN (3)

- What if the graph has **multiple relation types**?
- Use different neural network weights for different relation types!



Recap: Classical GNN Layers: GCN

■ (1) Graph Convolutional Networks (GCN)

$$\mathbf{h}_v^{(l)} = \sigma \left(\sum_{u \in N(v)} \mathbf{w}^{(l)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|} \right)$$

■ We add a self-loop

$$\mathbf{h}_v^{(l)} = \sigma \left(\sum_{u \in N(v)} \mathbf{w}^{(l)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|} + \mathbf{w}^{(l)} \mathbf{h}_v^{(l-1)} \right)$$

Relational GCN (4)

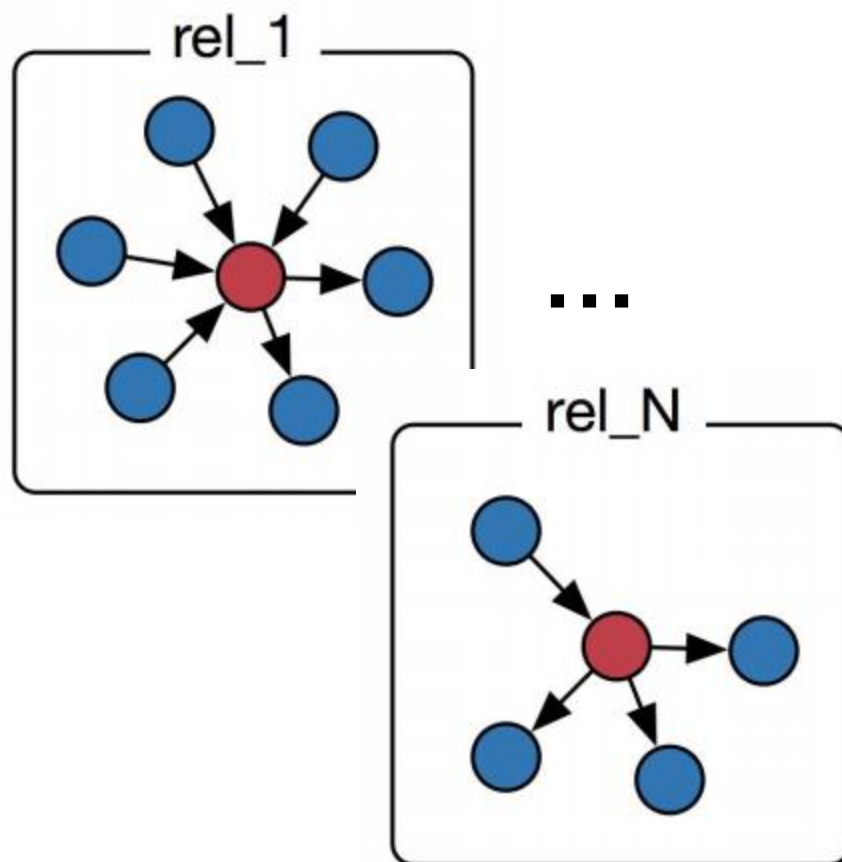
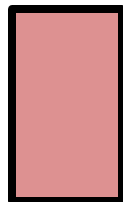
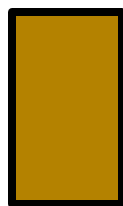
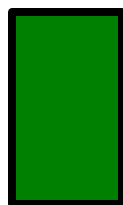
- Introduce a set of neural networks for each relation type!

Weight for rel_1

...

Weight for rel_N

Weight for self-loop



Relational GCN: Definition

- **Relational GCN (RGCN):**

$$\mathbf{h}_v^{(l+1)} = \sigma \left(\sum_{r \in R} \sum_{u \in N_v^r} \frac{1}{c_{v,r}} \mathbf{W}_r^{(l)} \mathbf{h}_u^{(l)} + \mathbf{W}_0^{(l)} \mathbf{h}_v^{(l)} \right)$$

- **How to write this as Message + Aggregation?**

- **Message:**

Normalized by node degree
of the relation $c_{v,r} = |N_v^r|$

- Each neighbor of a given relation:

$$\mathbf{m}_{u,r}^{(l)} = \frac{1}{c_{v,r}} \mathbf{W}_r^{(l)} \mathbf{h}_u^{(l)}$$

- Self-loop:

$$\mathbf{m}_v^{(l)} = \mathbf{W}_0^{(l)} \mathbf{h}_v^{(l)}$$

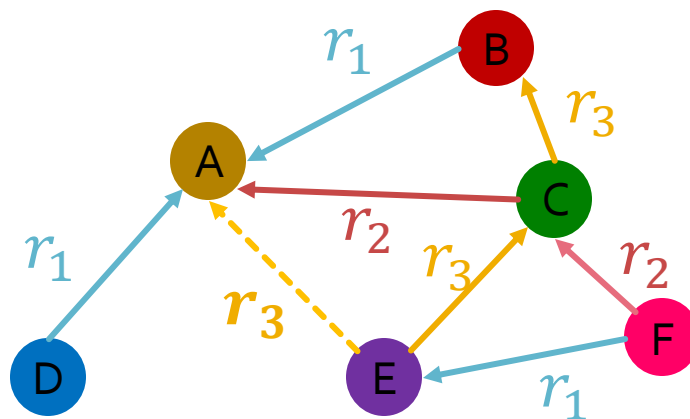
- **Aggregation:**

- Sum over messages from neighbors and self-loop, then apply activation

- $\mathbf{h}_v^{(l+1)} = \sigma \left(\text{Sum} \left(\left\{ \mathbf{m}_{u,r}^{(l)}, u \in N(v) \right\} \cup \left\{ \mathbf{m}_v^{(l)} \right\} \right) \right)$

RGCN for Link Prediction (1)

- Assume (E, r_3, A) is training supervision edge, all the other edges are training message edges
- Use RGCN to score (E, r_3, A) !
 - Take the final layer of E and A : $\mathbf{h}_E^{(L)}$ and $\mathbf{h}_A^{(L)} \in \mathbb{R}^d$
 - Relation-specific score function $f_r: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$
 - One example $f_{r_3}(\mathbf{h}_E, \mathbf{h}_A) = \mathbf{h}_E^T \mathbf{W}_{r_3} \mathbf{h}_A$, $\mathbf{W}_{r_3} \in \mathbb{R}^{d \times d}$



Input Graph

Heterogeneous GNNs

- Structure aware GNN encoders **learn embeddings** for **entities**
- Structure agnostic shallow encoders **learn embeddings** for **relations**
 - **The relation embeddings are trainable weights**
- The structure of the KG is also utilized in the loss function
- RGCN can be used for inductive link prediction across new entities
 - **It cannot transfer knowledge between KGs with different relation types**

Stanford CS224W: Inductive Link prediction for new entity and relation types

CS224W: Machine Learning with Graphs
Jure Leskovec, Stanford University
<http://cs224w.stanford.edu>



Generalize across different KGs

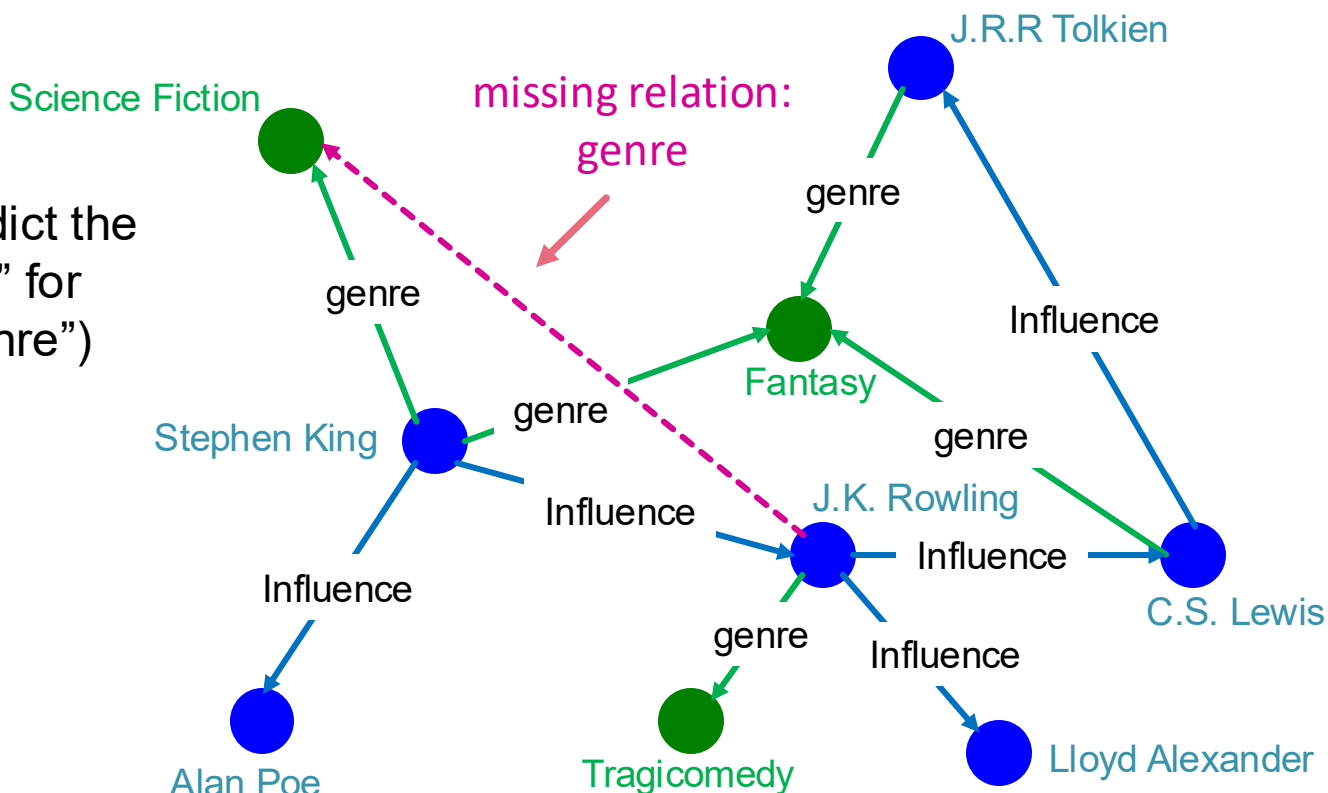
- What if I want to transfer knowledge from one KG to another?
- The KGs have different entities and relations

Recap: KG Completion Task

Given a KG, we train a model such that:

- For a given (**head**, **relation**), we predict missing **tails**.

Example task: predict the tail “Science Fiction” for (“J.K. Rowling”, “genre”)

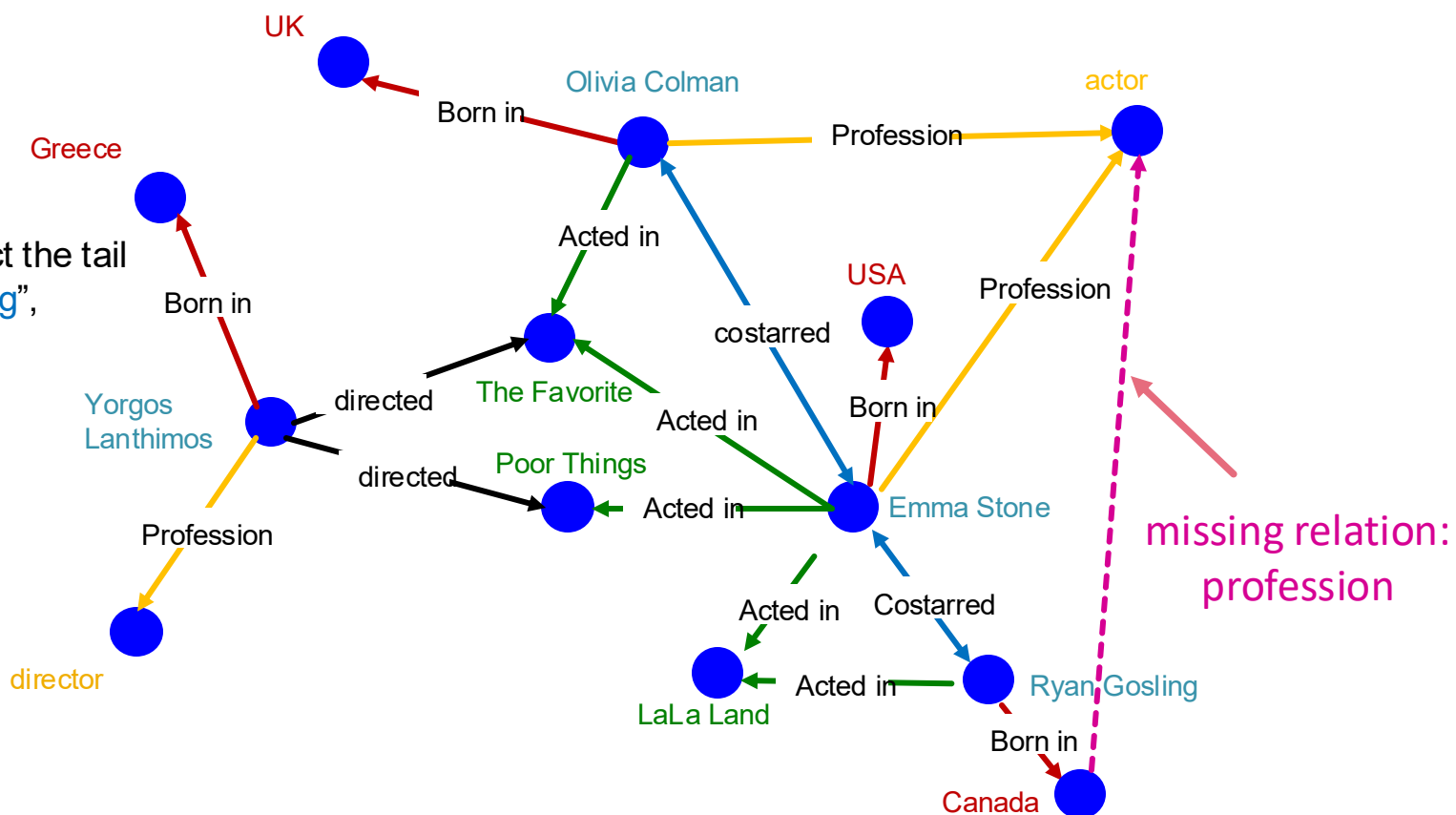


Recap: KG Completion Task

Given a trained model we want:

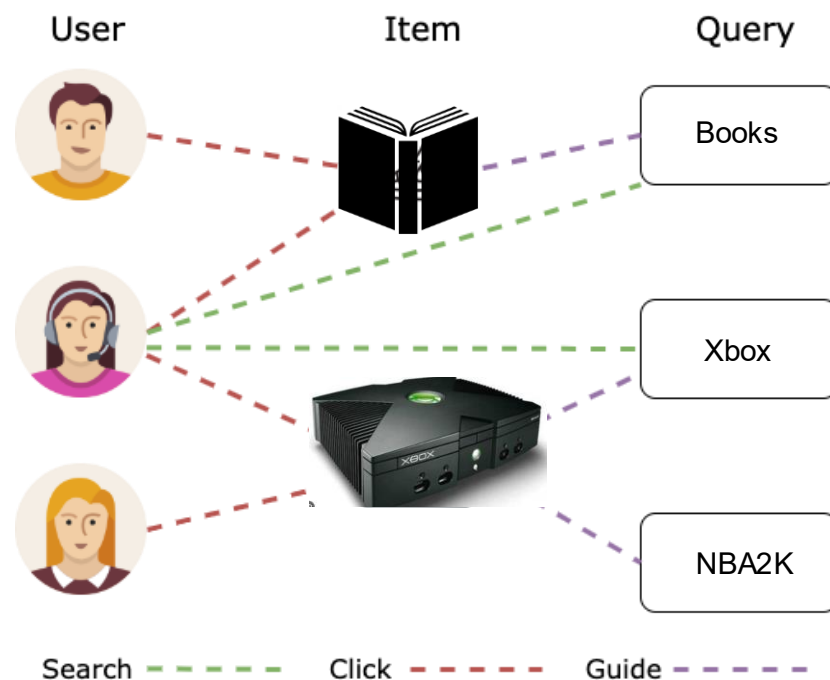
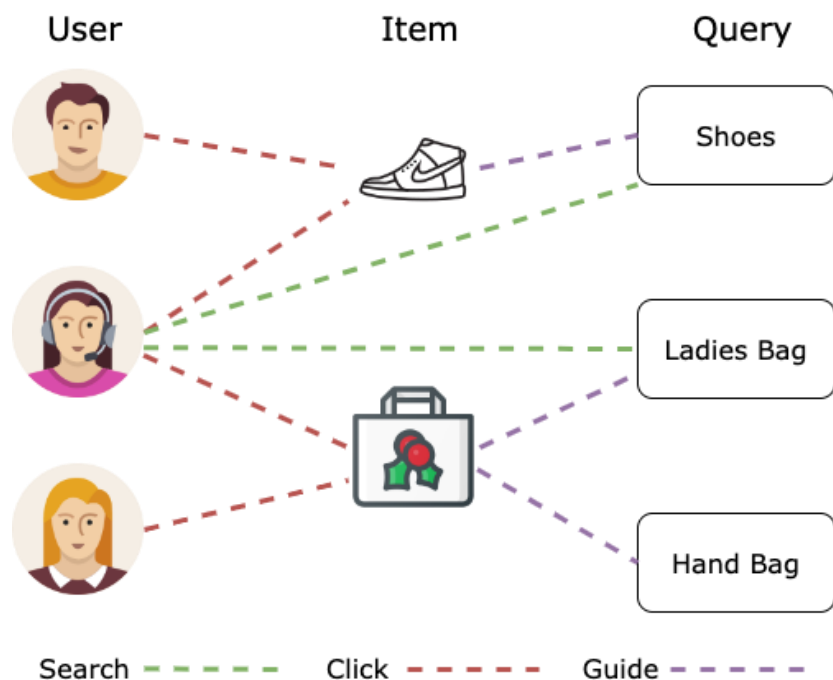
- For a given (**head**, **relation**), we predict missing **tails**.

Example task: predict the tail
“Actor” for (“R. Gosling”,
“profession”)



Generalize across similar KGs

- What if I want to transfer knowledge from one KG to another?
- The KGs have the same relations



Stanford CS224W: InGram: Inductive KG Embedding via Relation Graphs

CS224W: Machine Learning with Graphs
Jure Leskovec, Stanford University
<http://cs224w.stanford.edu>



Generalize across different KGs

- What if I want to transfer knowledge from one KG to another?
- The KGs have different entities relations

Solution:

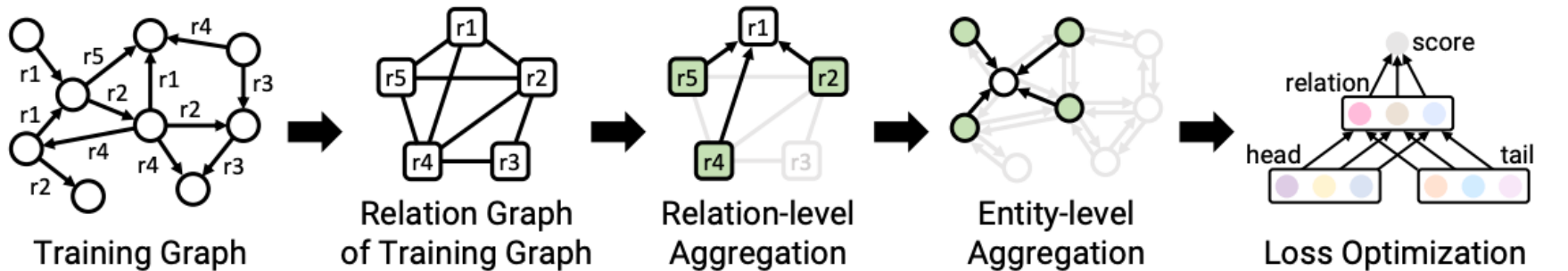
- Build double equivariant models
 - E.g., InGram

KG embedding methods

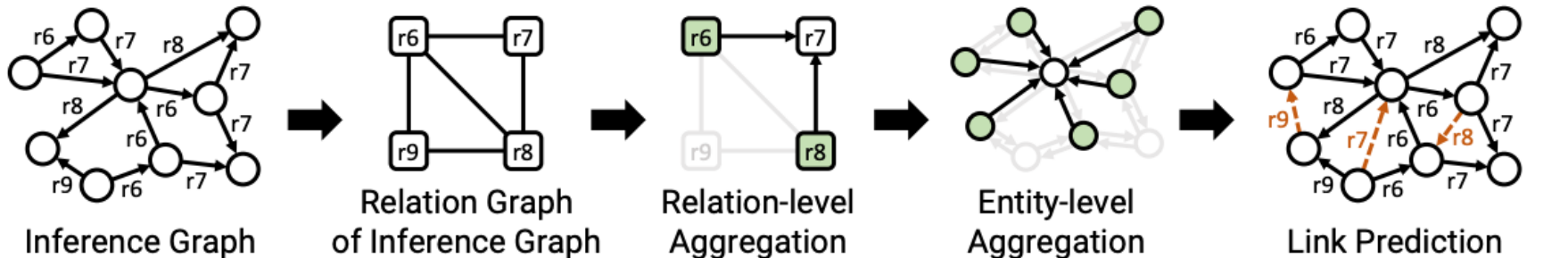
- We achieve inductive link prediction on new entities by:
 - Structure agnostic shallow encoders to structure aware GNN encoders to **learn embeddings** for **entities**
 - The **structure of the KG** was moved from the loss function to the loss function and the model
- Let's do the same for **relations**
 - Utilize the **structure of relations** to generate relation **embeddings**

InGram Model

Training Time:



Inference Time:



Source: https://bdi-lab.kaist.ac.kr/download/ingram_icml2023_slides.pdf

■ New component: Relation Graph

Relation graph

- We define **tree** graph matrices
- E_h : **head entities x relations** adjacency
 - $E_h[i, j] = 1$ if entity i is head in relation j
- E_t : **tail entities x relations** adjacency
 - $E_t[i, j] = 1$ if entity i is tail in relation j
- D_h, D_t : diagonal **entity x entity** matrices with entity degrees as heads and tails

Relation graph

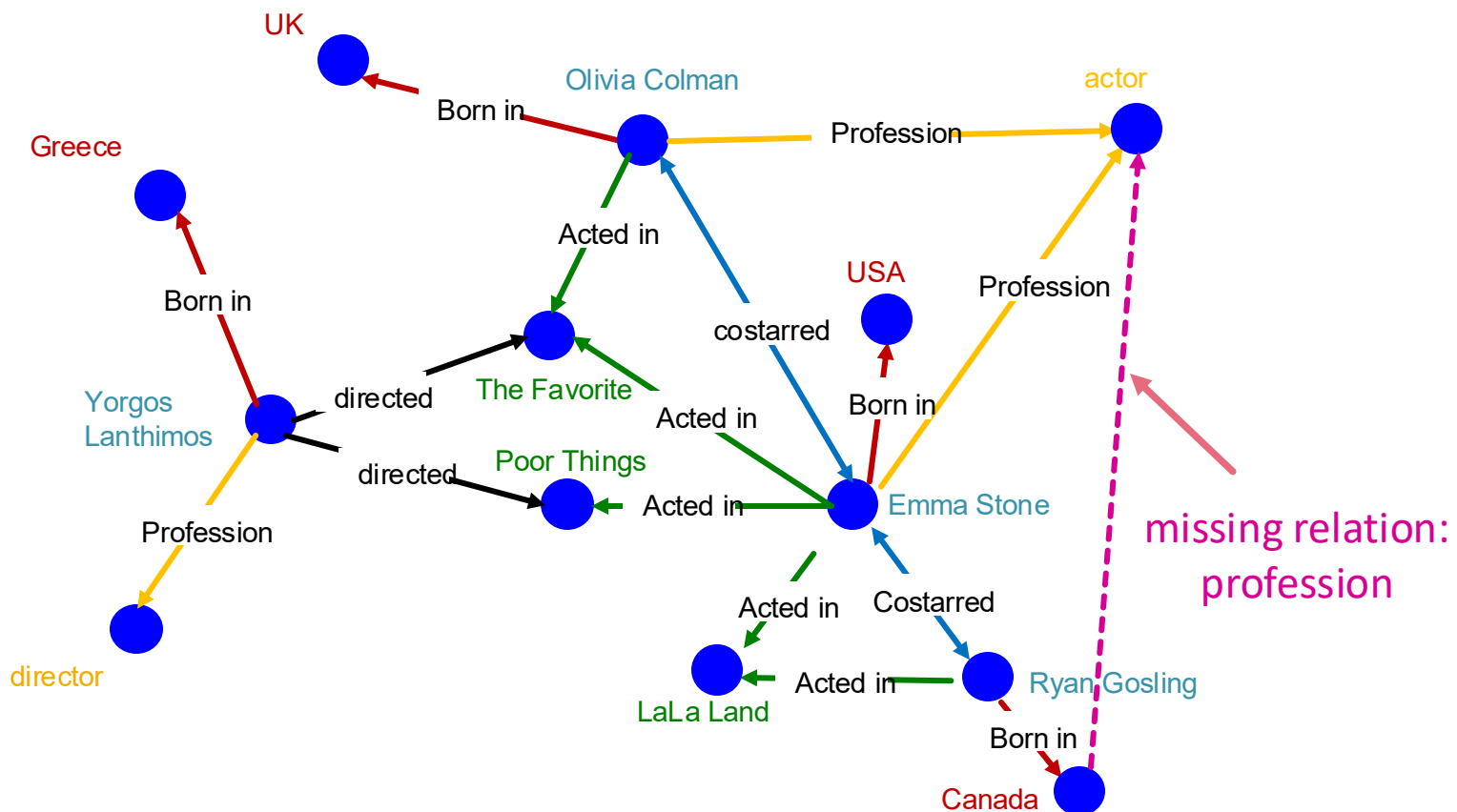
- Then the **adjacency matrix of relations** is:

$$\mathbf{A} = \mathbf{E}_h^\top \mathbf{D}_h^{-2} \mathbf{E}_h + \mathbf{E}_t^\top \mathbf{D}_t^{-2} \mathbf{E}_t$$

- \mathbf{A} is a **weighted adjacency matrix**
 - Every edge has a weight that determines the **importance of the edge**
- In practice we are looking for entities that serve as heads and/or tails for relation pairs

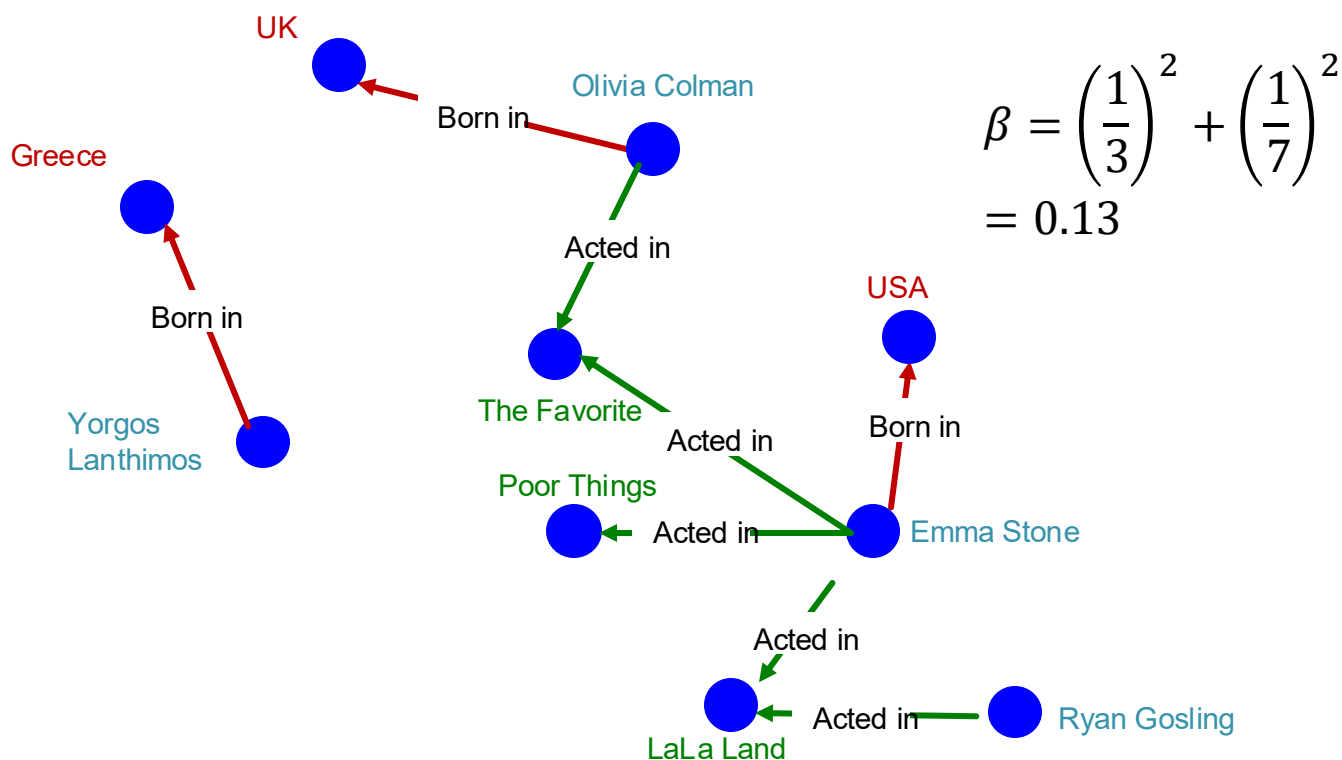
Relation Graph Construction

- Consider relation pairs
 - Find entities that serve as heads and/or tails



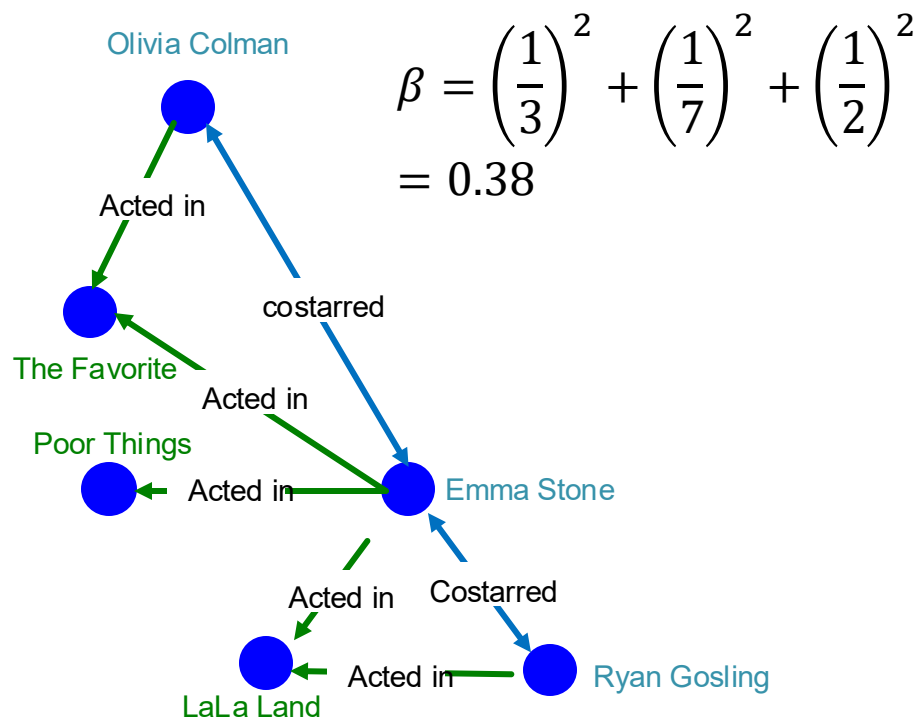
Recap: KG Completion Task

- Consider relation pairs
 - Find entities that serve as heads and/or tails



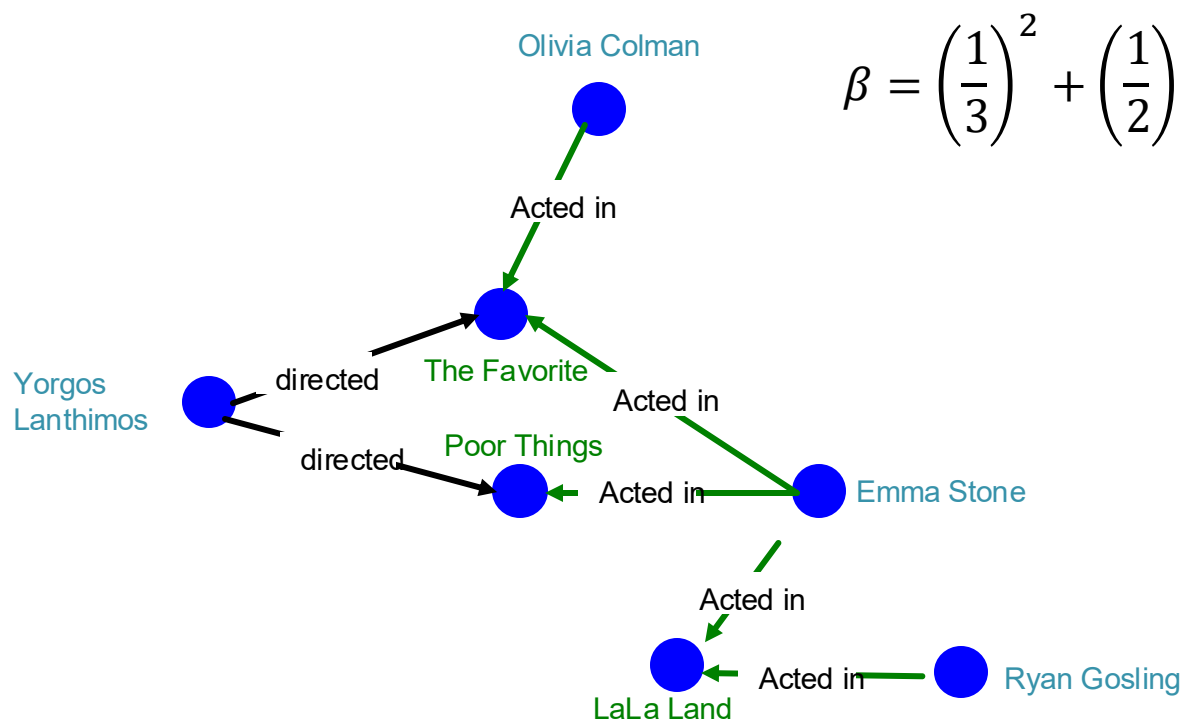
Recap: KG Completion Task

- Consider relation pairs
 - Find entities that serve as heads and/or tails



Recap: KG Completion Task

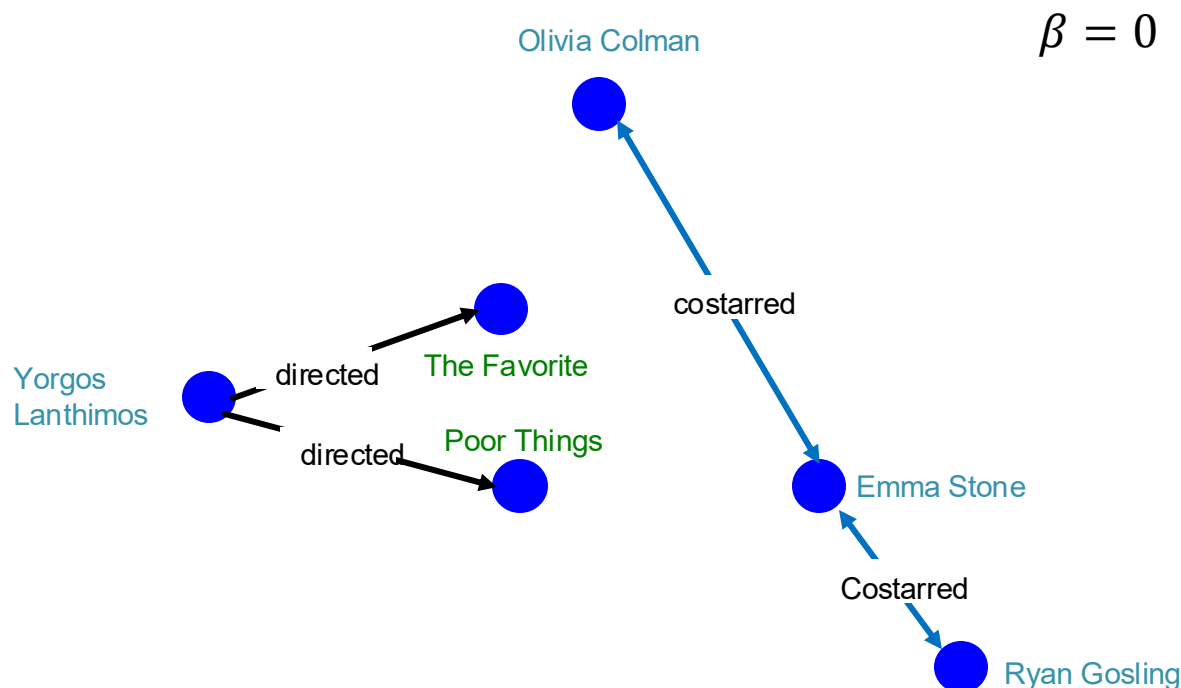
- Consider relation pairs
 - Find entities that serve as heads and/or tails



$$\beta = \left(\frac{1}{3}\right)^2 + \left(\frac{1}{2}\right)^2 = 0.36$$

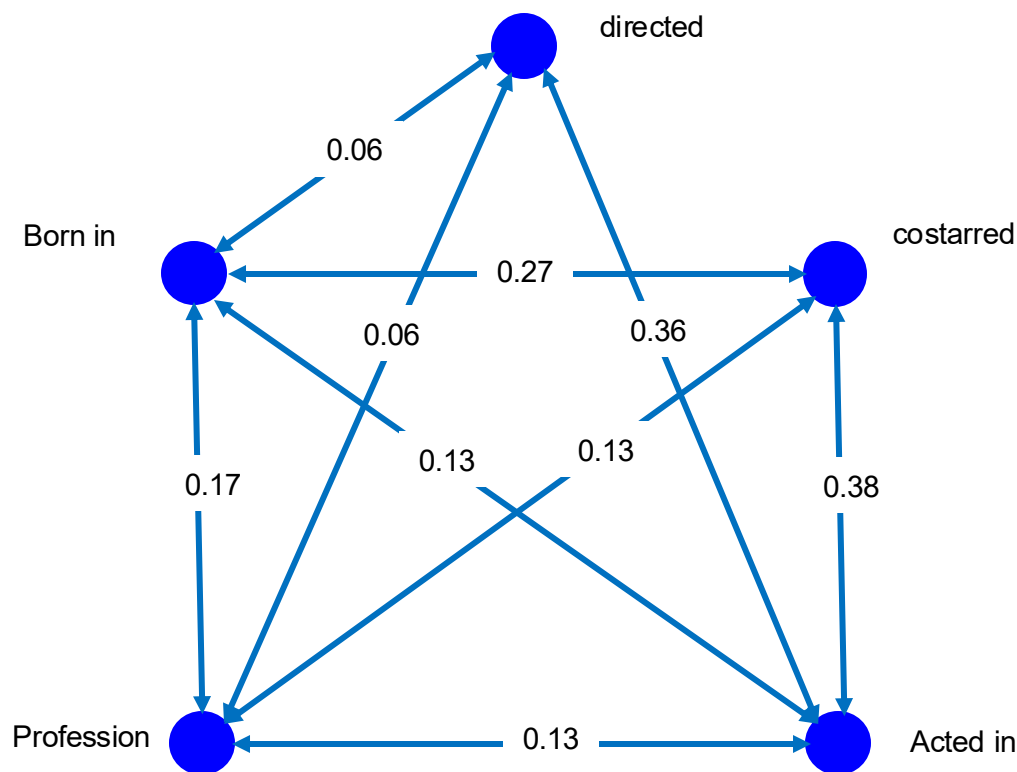
Recap: KG Completion Task

- Consider relation pairs
 - Find entities that serve as heads and/or tails



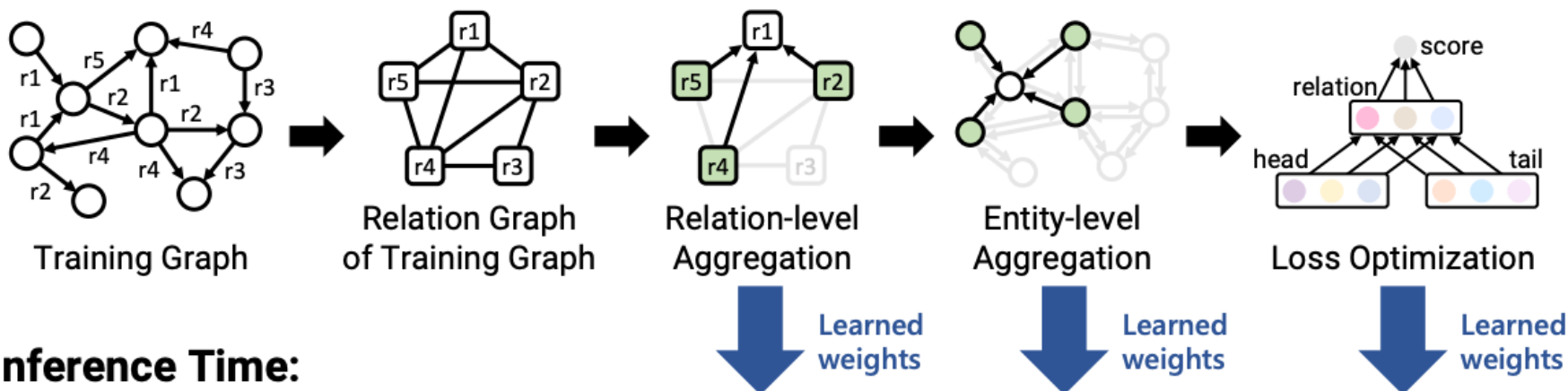
Relation Graph

- We can construct a graph **between relations**
 - Encodes the **proximity** between different relations

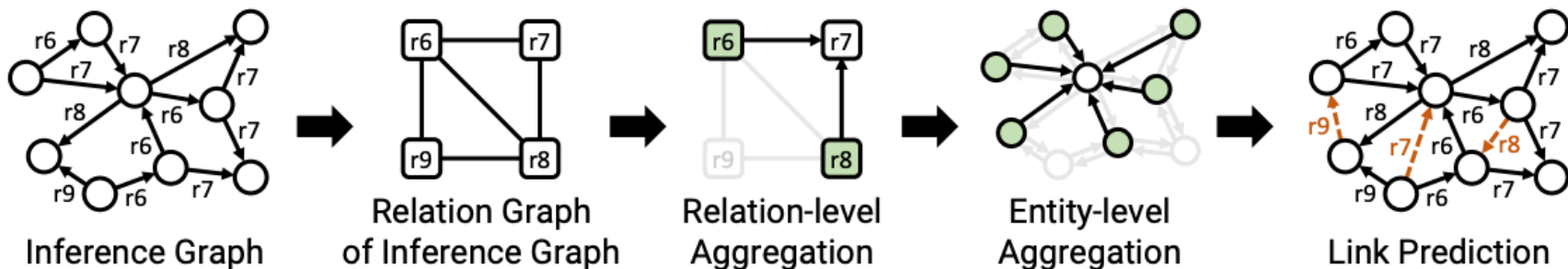


InGram Model

Training Time:



Inference Time:



Source: https://bdi-lab.kaist.ac.kr/download/ingram_icml2023_slides.pdf

■ Next: Relation-level message-passing

Recap: Graph Attention Network

■ (3) Graph Attention Networks

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \alpha_{vu} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Attention weights

Not all node's neighbors are equally important

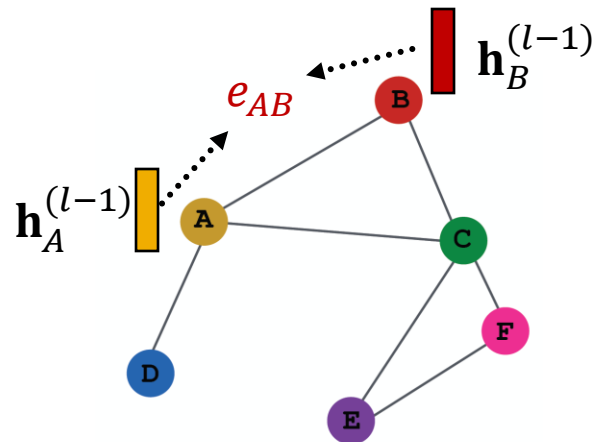
- **Attention** is inspired by cognitive attention.
- The **attention** α_{vu} focuses on the important parts of the input data and fades out the rest.
 - **Idea:** the NN should devote more computing power on that small but important part of the data.
 - Which part of the data is more important depends on the context and is learned through training.

Recap: Attention Mechanism (1)

- Let α_{vu} be computed as a byproduct of an **attention mechanism a** :
 - (1) Let a compute **attention coefficients e_{vu}** across pairs of nodes u, v based on their messages:

$$e_{vu} = a(\mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}, \mathbf{W}^{(l)} \mathbf{h}_v^{(l-1)})$$

- e_{vu} indicates the importance of u 's message to node v

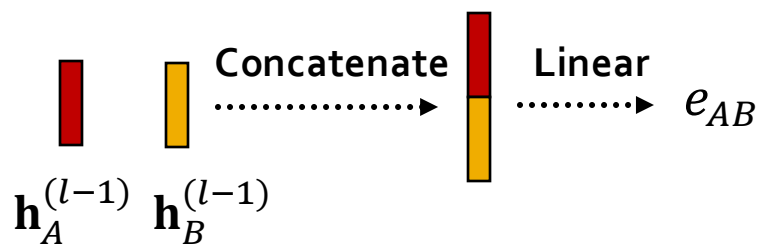


$$e_{AB} = a(\mathbf{W}^{(l)} \mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)} \mathbf{h}_B^{(l-1)})$$

Recap: Attention Mechanism (2)

■ What is the form of attention mechanism a ?

- The approach is agnostic to the choice of a
 - E.g., use a simple single-layer neural network
 - a have trainable parameters (weights in the Linear layer)



$$\begin{aligned} e_{AB} &= a\left(\mathbf{W}^{(l)}\mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)}\mathbf{h}_B^{(l-1)}\right) \\ &= \text{Linear}\left(\text{Concat}\left(\mathbf{W}^{(l)}\mathbf{h}_A^{(l-1)}, \mathbf{W}^{(l)}\mathbf{h}_B^{(l-1)}\right)\right) \end{aligned}$$

- Parameters of a are trained jointly:
 - Learn the parameters together with weight matrices (i.e., other parameter of the neural net $\mathbf{W}^{(l)}$) in an end-to-end fashion

Recap: Attention Mechanism (3)

- **Normalize** e_{vu} into the **final attention weight** α_{vu}

- Use the **softmax** function, so that $\sum_{u \in N(v)} \alpha_{vu} = 1$:

$$\alpha_{vu} = \frac{\exp(e_{vu})}{\sum_{k \in N(v)} \exp(e_{vk})}$$

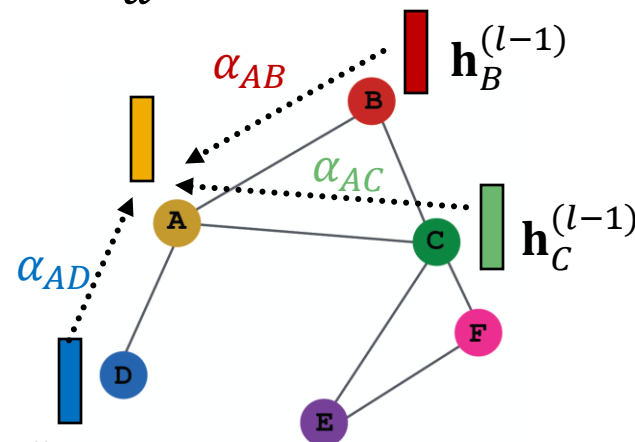
- **Weighted sum** based on the **final attention weight**

α_{vu} :

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \alpha_{vu} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}\right)$$

Weighted sum using α_{AB} , α_{AC} , α_{AD} :

$$\mathbf{h}_A^{(l)} = \sigma(\alpha_{AB} \mathbf{W}^{(l)} \mathbf{h}_B^{(l-1)} + \alpha_{AC} \mathbf{W}^{(l)} \mathbf{h}_C^{(l-1)} + \alpha_{AD} \mathbf{W}^{(l)} \mathbf{h}_D^{(l-1)})$$



Relation Attention Mechanism

- **Normalize** e_{vu} into the **final attention weight** α_{vu}

- Use the **softmax** function, so that $\sum_{u \in N(v)} \alpha_{vu} = 1$:

$$\alpha_{vu} = \frac{\exp(e_{vu} + \beta_{vu})}{\sum_{k \in N(v)} \exp(e_{vk} + \beta_{vk})}$$

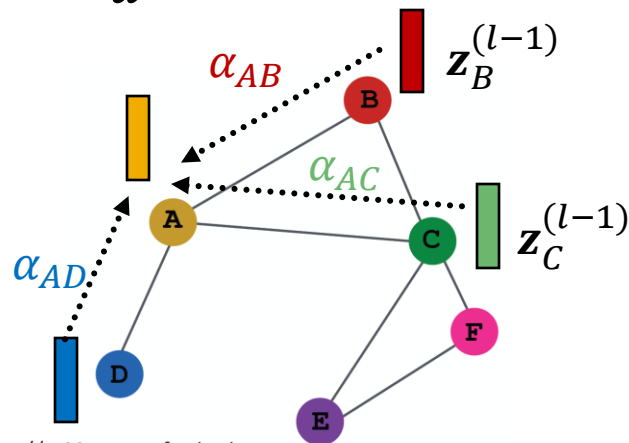
- **Weighted sum** based on the **final attention weight**

α_{vu} :

$$\mathbf{z}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \alpha_{vu} \mathbf{W}^{(l)} \mathbf{z}_u^{(l-1)}\right)$$

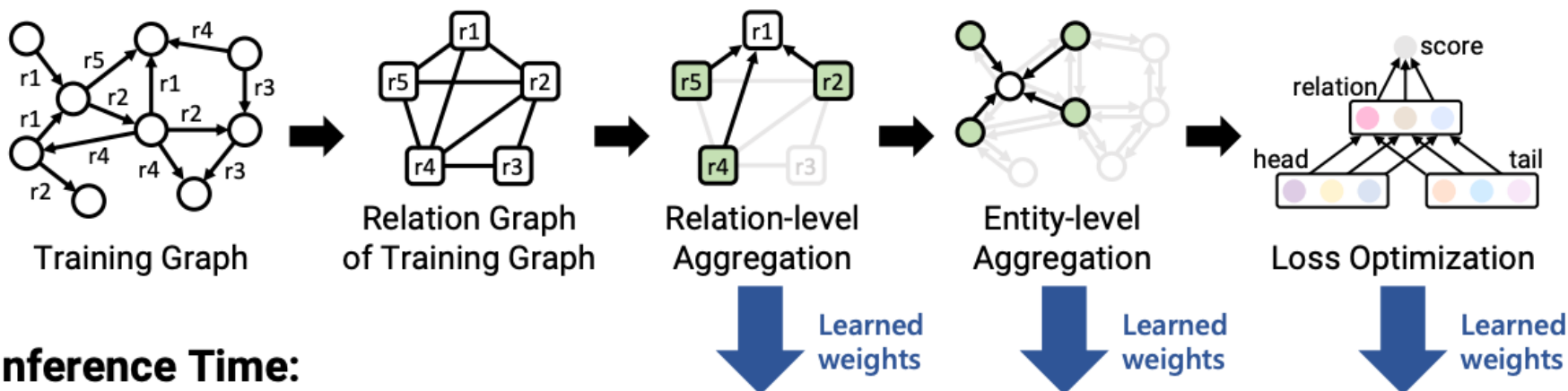
Weighted sum using α_{AB} , α_{AC} , α_{AD} :

$$\mathbf{z}_A^{(l)} = \sigma(\alpha_{AB} \mathbf{W}^{(l)} \mathbf{z}_B^{(l-1)} + \alpha_{AC} \mathbf{W}^{(l)} \mathbf{z}_C^{(l-1)} + \alpha_{AD} \mathbf{W}^{(l)} \mathbf{z}_D^{(l-1)})$$

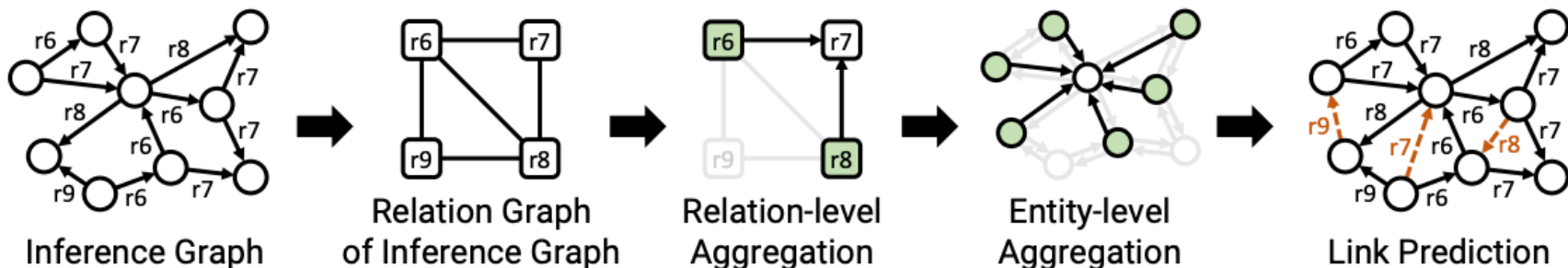


InGram Model

Training Time:



Inference Time:



Source: https://bdi-lab.kaist.ac.kr/download/ingram_icml2023_slides.pdf

■ **Next: Entity-level message-passing**

Entity Attention Mechanism

- Entity-level message-passing

$$\mathbf{h}_v^{\dagger(l+1)} = \sigma \left(\sum_{\mathbf{r} \in R} \sum_{u \in N_v^{\mathbf{r}}} \mathbf{W}_1^{(l)} \mathbf{h}_u^{(l)} + \mathbf{W}_0^{(l)} \mathbf{h}_v^{(l)} \right)$$

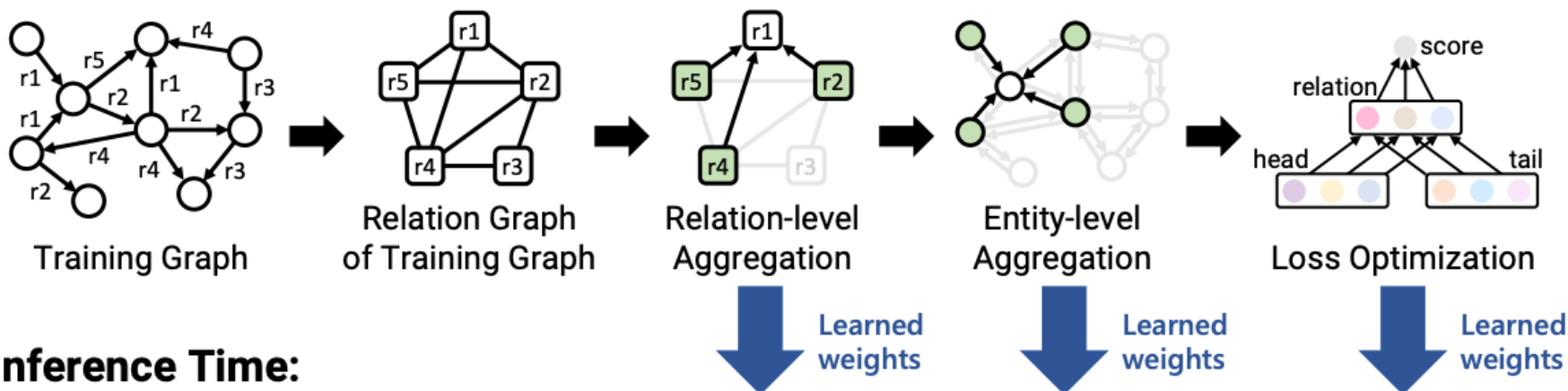
$$\mathbf{h}_v^{*(l+1)} = \sigma \left(\sum_{\mathbf{r} \in R} \sum_{u \in N_v^{\mathbf{r}}} \alpha_{urv} \left(\mathbf{W}_3^{(l)} \mathbf{z}_r^{(l)} + \mathbf{W}_2^{(l)} \mathbf{h}_u^{(l)} \right) \right)$$

$$\mathbf{h}_v^{(l+1)} = \mathbf{h}_v^{\dagger(l+1)} + \mathbf{h}_v^{*(l+1)}$$

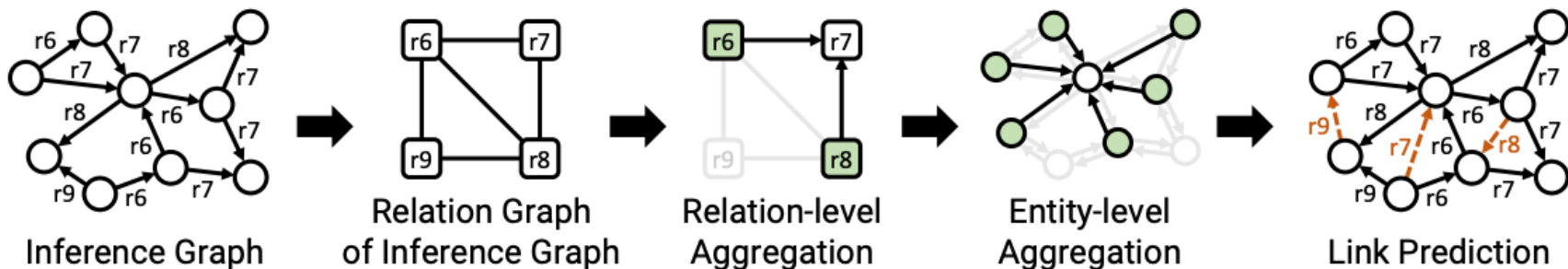
- The entity update depends on the relation update

InGram Model

Training Time:



Inference Time:



Source: https://bdi-lab.kaist.ac.kr/download/ingram_icml2023_slides.pdf

■ Next: Link-prediction head

Link-Prediction Head

- Final embedding vectors computation

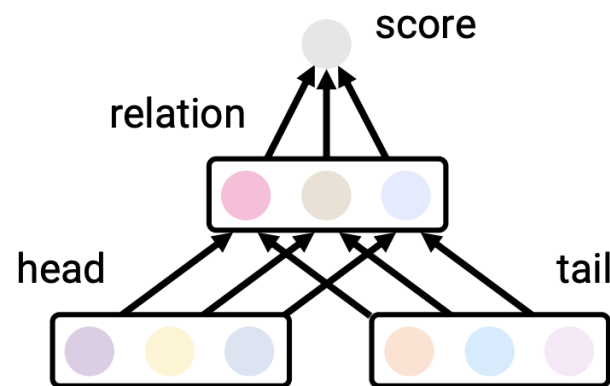
$$\mathbf{z}_k = \mathbf{M}\mathbf{z}_k^{(L)} \text{ and } \mathbf{h}_i = \widehat{\mathbf{M}}\mathbf{h}_i^{(\widehat{L})}$$

- Scoring function

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

- Loss

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



Source: https://bdi-lab.kaist.ac.kr/down/ingram_icml2023_slides.pdf

Link-Prediction Head

- Final embedding vectors computation

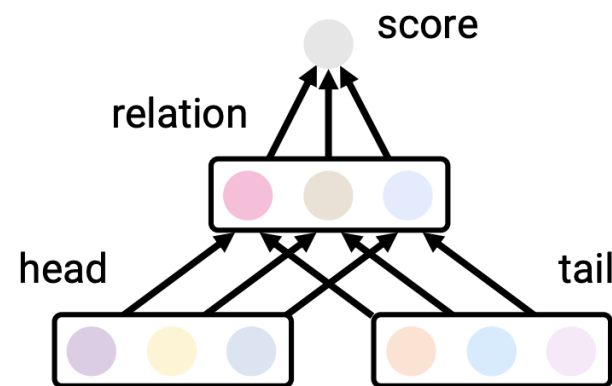
$$\mathbf{z}_k = \mathbf{M}\mathbf{z}_k^{(L)} \text{ and } \mathbf{h}_i = \widehat{\mathbf{M}}\mathbf{h}_i^{(\widehat{L})}$$

- Scoring function

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

- Loss

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



Source: https://bdi-lab.kaist.ac.kr/down/ingram_icml2023_slides.pdf

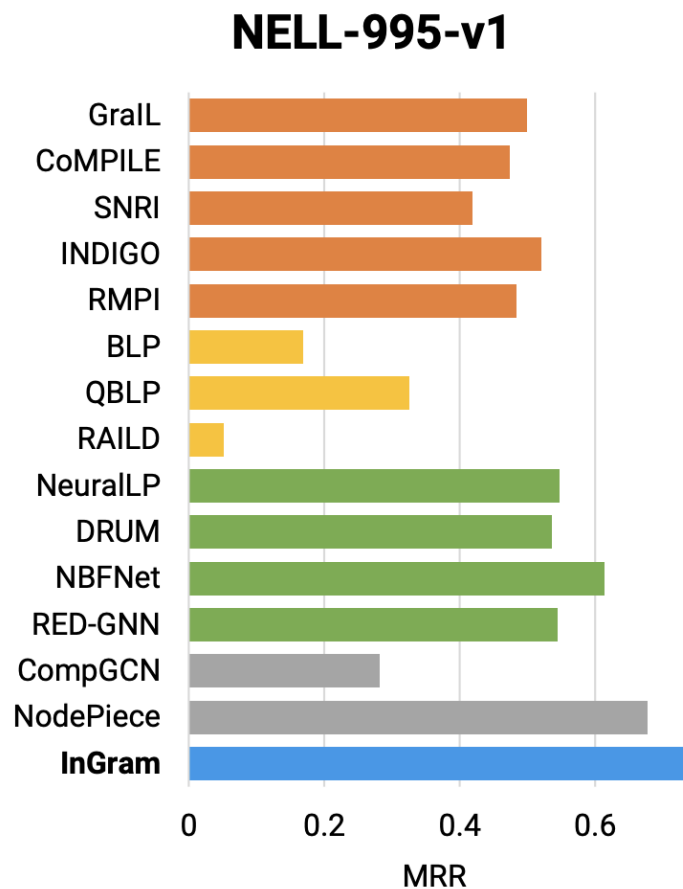
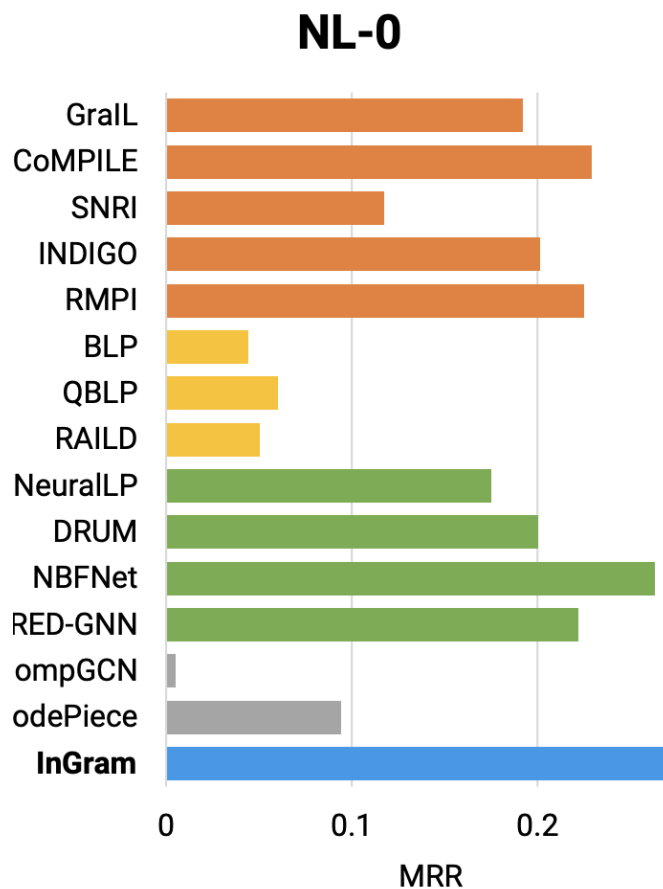
InGram

- Structure aware GNN encoders **learn embeddings** for **entities**
- Structure aware GNN encoders **learn embeddings** for **relations**
- The structure of the KG is also utilized in the loss function
- InGram can be used for inductive link prediction across new entities and new relation
 - **It is a model we can use to train a foundation model**

Experiments

- **Create different datasets from NELL, Wikidata, and Freebase**
- **Test the performance of InGram in Transductive and Inductive link prediction**

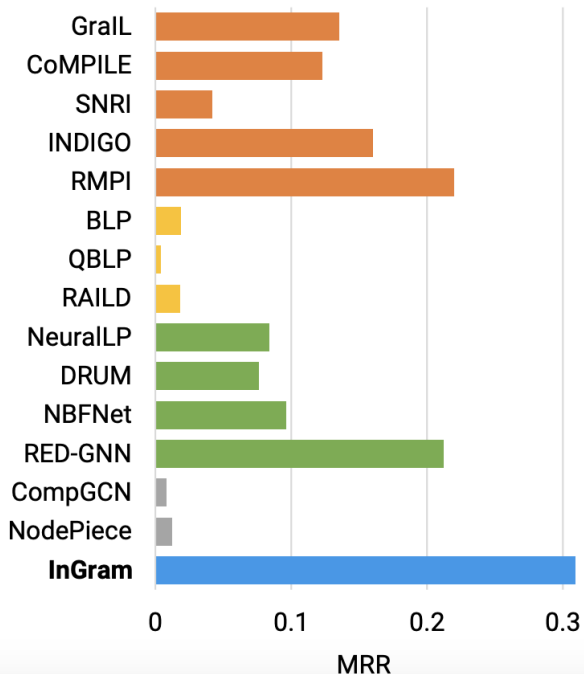
Transductive Link Prediction



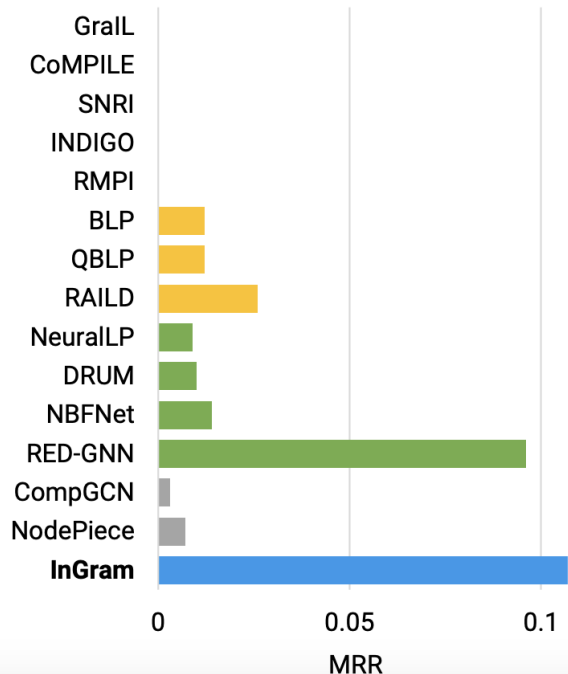
Source: https://bdi-lab.kaist.ac.kr/down/ingram_icml2023_slides.pdf

Inductive Link Prediction

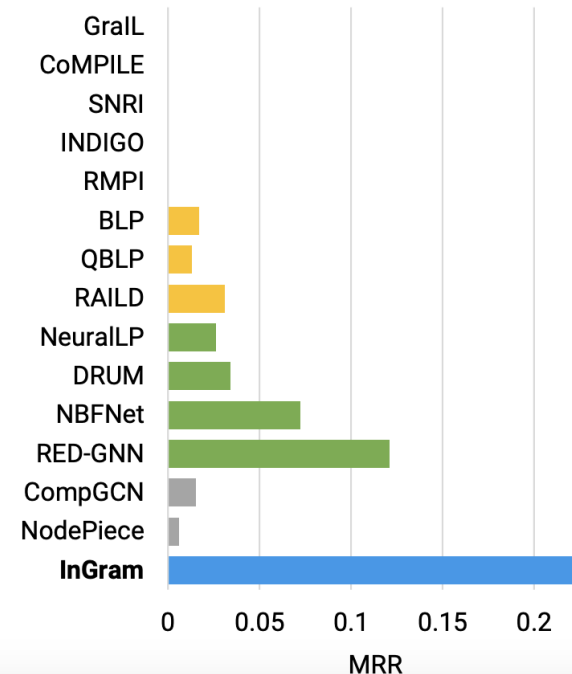
NL-100



WK-100



FB-100



Source: https://bdi-lab.kaist.ac.kr/down/ingram_icml2023_slides.pdf

Summary of the Lecture

- Introduced a model that **learns embeddings** for **entities** and **relations** of **any KG**
- The model (InGram) is trained in an **unsupervised fashion** for **KG completion**
- The model can **transfer knowledge** between different KGs
- This is a step towards **foundation models for KGs**