TECHNISCHE UNIVERSITÄT DRESDEN

SDA

Fraunhofer Dresden IAIS

The 2020 Conference on Empirical Methods in Natural Language Processing

Message Passing for Hyper-Relational Knowledge Graphs

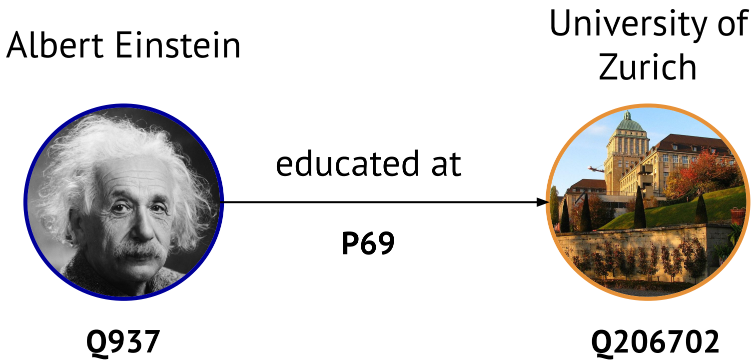




Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, Jens Lehmann

TU Dresden & Fraunhofer IAIS & SDA Research
Dresden, Germany

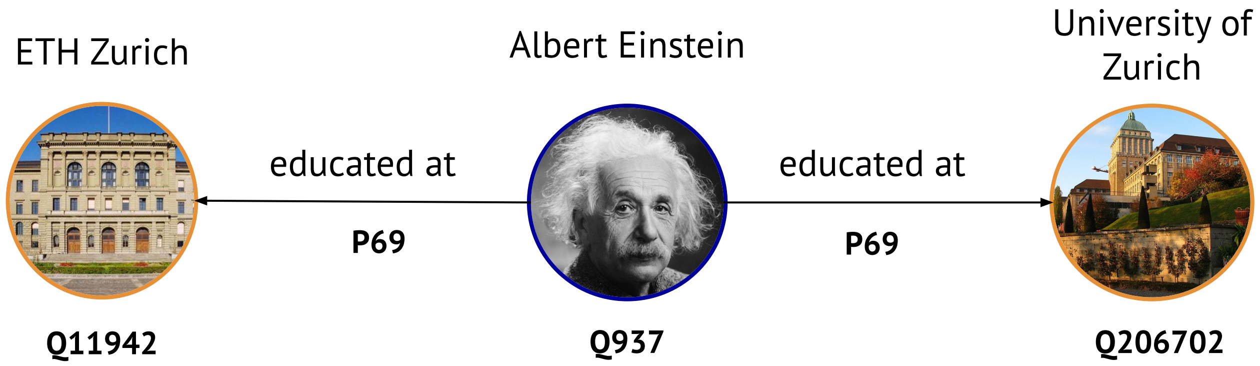
Triple-based Knowledge Graphs



RDF `Albert Einstein` `educatedAt` `University of Zurich .`

Source: [Albert Einstein](#) on Wikidata

Triple-based Knowledge Graphs



RDF

```
Albert Einstein    educatedAt    University of Zurich .
Albert Einstein    educatedAt    ETH Zurich .
```

Source: [Albert Einstein](#) on Wikidata

Triple-based Knowledge Graphs

Wait a sec

Studied the same subject?

Got the same degree?

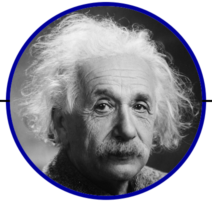
ETH Zurich

Q11942

educated at

P69

Albert Einstein



Q937

educated at

P69

University of Zurich

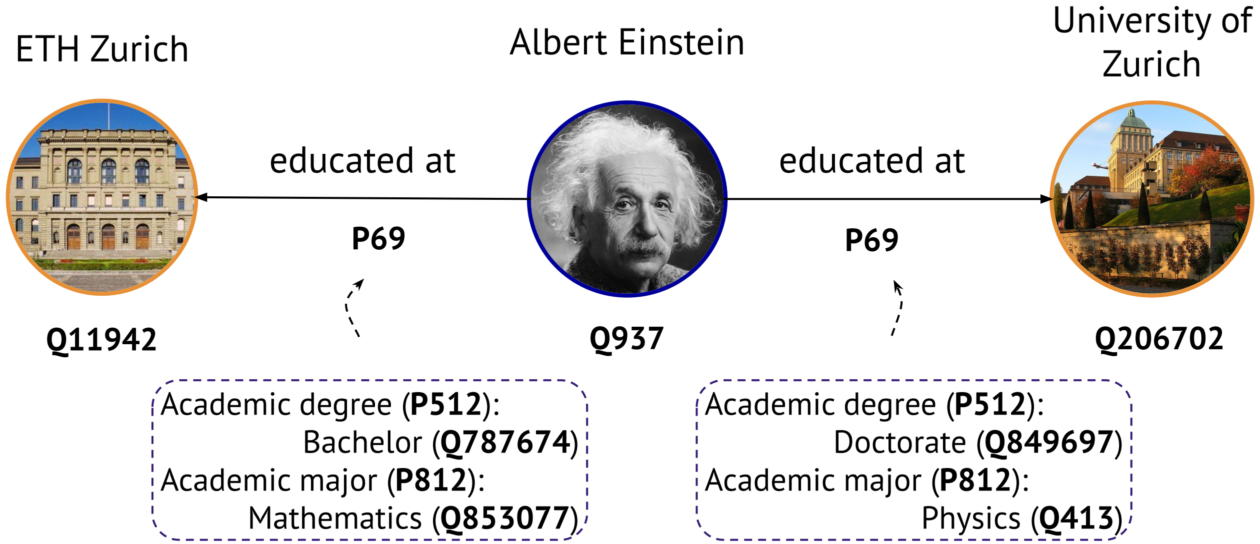
Q206702

RDF

```
Albert Einstein    educatedAt    University of Zurich .
Albert Einstein    educatedAt    ETH Zurich .
```

Source: Albert Einstein on Wikidata

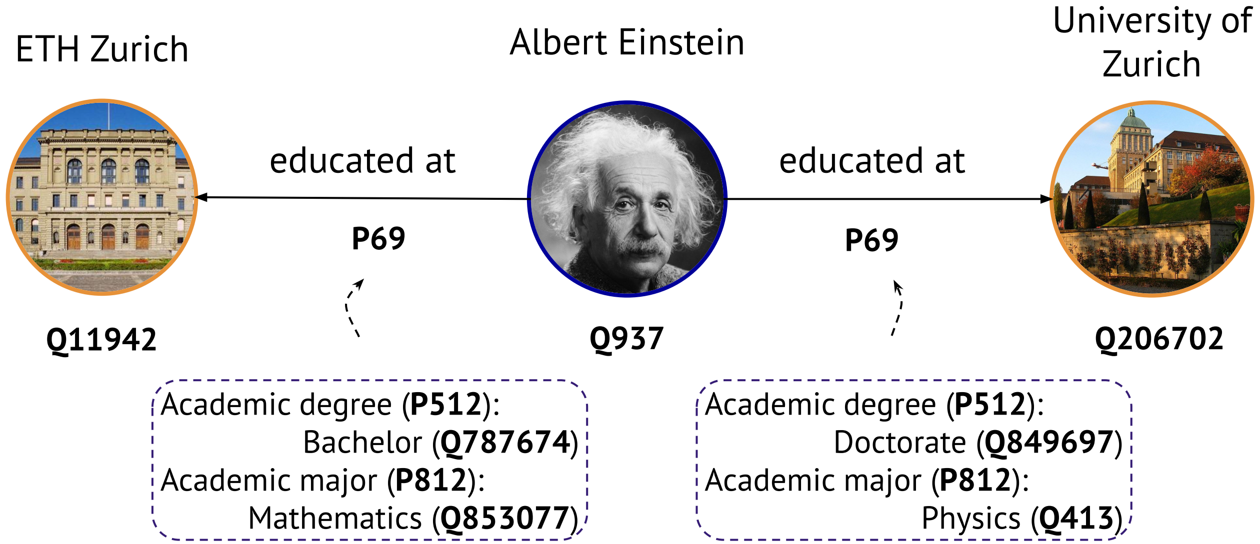
Hyper-Relational Knowledge Graphs



Statements with Qualifiers in Wikidata
Entity-relation Edge Attributes
Edge Instances

[Rosso et al] Beyond Triplets: Hyper-Relational Knowledge Graph Embedding for Link Prediction. WWW 2020
[Hartig et al] Foundations of rdf* and sparql* (an alternative approach to statement-level metadata in RDF)

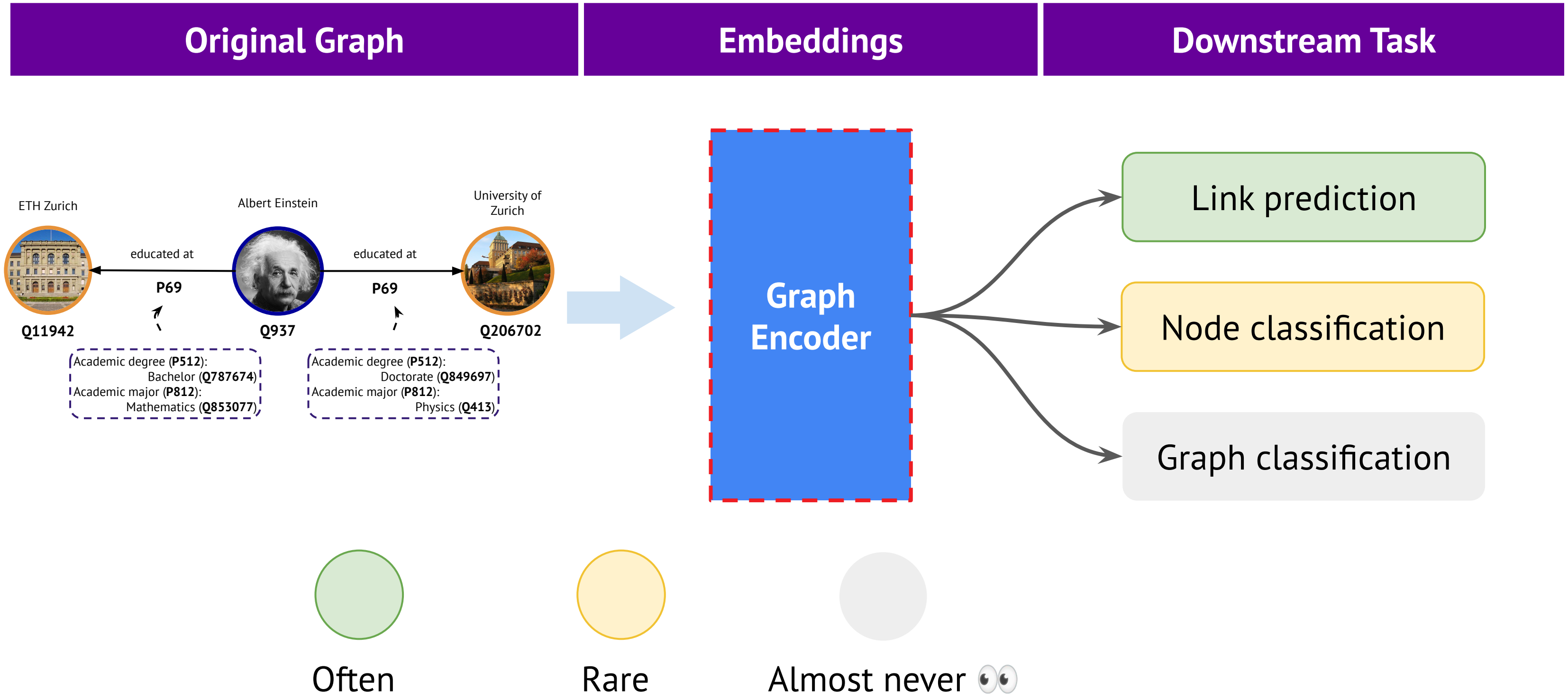
Hyper-Relational KG: RDF and SPARQL*



RDF*

```
<< Albert Einstein educatedAt University of Zurich >>
    academic degree Doctorate;
    academic major  Physics .
<< Albert Einstein educatedAt ETH Zurich >>
    Academic degree Bachelor ;
    Academic major  Mathematics .
```

Graph Representation Learning Pipeline



Multirelational GNN Encoders for KGs

$$\mathbf{h}_v^{(k)} = f\left(\sum_{u \in \mathcal{N}(v)} \mathbf{W}^{(k)} \mathbf{h}_u^{(k-1)}\right)$$

Vanilla GCN [1]: no relations

$$\mathbf{h}_v^{(k)} = f\left(\sum_{(u,r) \in \mathcal{N}(v)} \mathbf{W}_r^{(k)} \mathbf{h}_u^{(k-1)}\right)$$

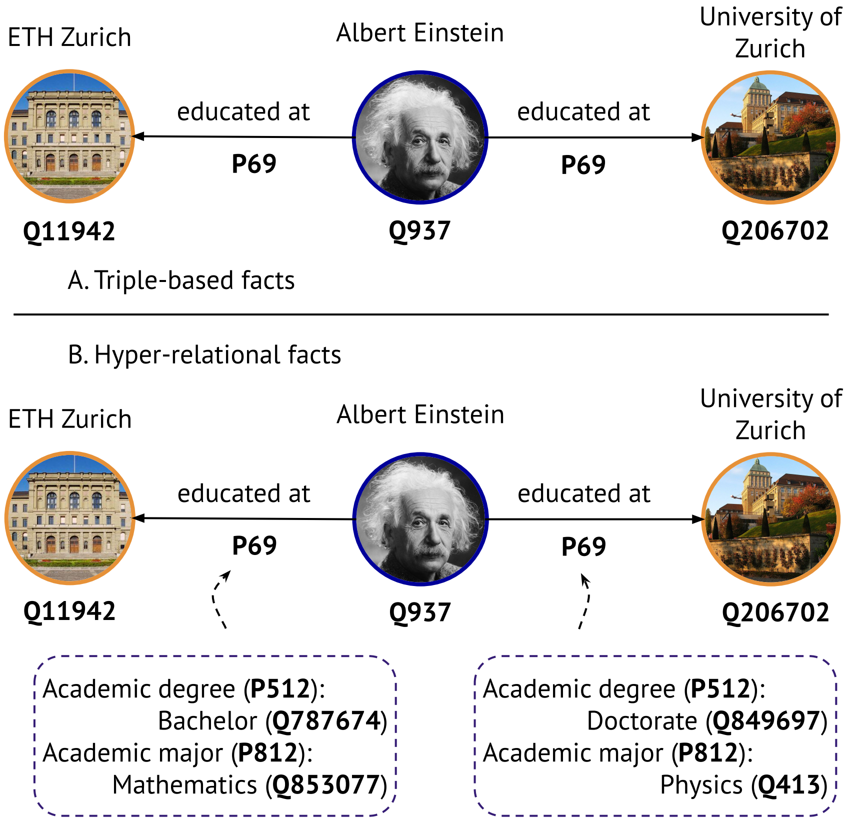
R-GCN [2]: a whole matrix W per relation

$$\boldsymbol{h}_v = f\left(\sum_{(u,r) \in \mathcal{N}(v)} \boldsymbol{W}_{\lambda(r)} \phi(\boldsymbol{x}_u, \boldsymbol{z}_r)\right)$$

CompGCN [3]: a vector z_r per relation +
composition of (h,r) +
only 3 different W: input/output/self-loop

[1] Kipf et al. Semi-supervised Classification with Graph Convolutional Networks. ICLR 2017
[2] Schlichtkrull et al. Modeling Relational Data with Graph Convolutional Networks. ESWC 2018
[3] Vashishth et al. Composition-Based Multi-Relational Graph Convolutional Networks. ICLR 2020

Embedding Hyper-Relational KGs

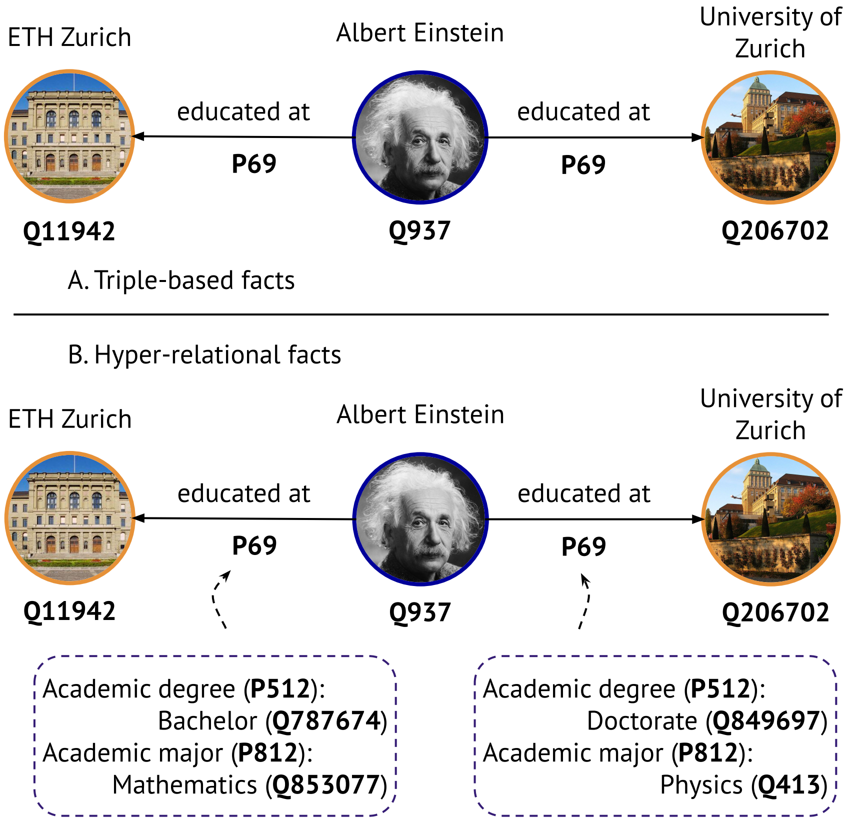


$$\boldsymbol{h}_v = f\left(\sum_{(u,r)\in\mathcal{N}(v)} \boldsymbol{W}_{\lambda(r)} \phi(\boldsymbol{x}_u, \boldsymbol{z}_r) \right)$$

?

- Qualifying relations and entities can be used as main terms in other facts
- Not all facts might have qualifiers

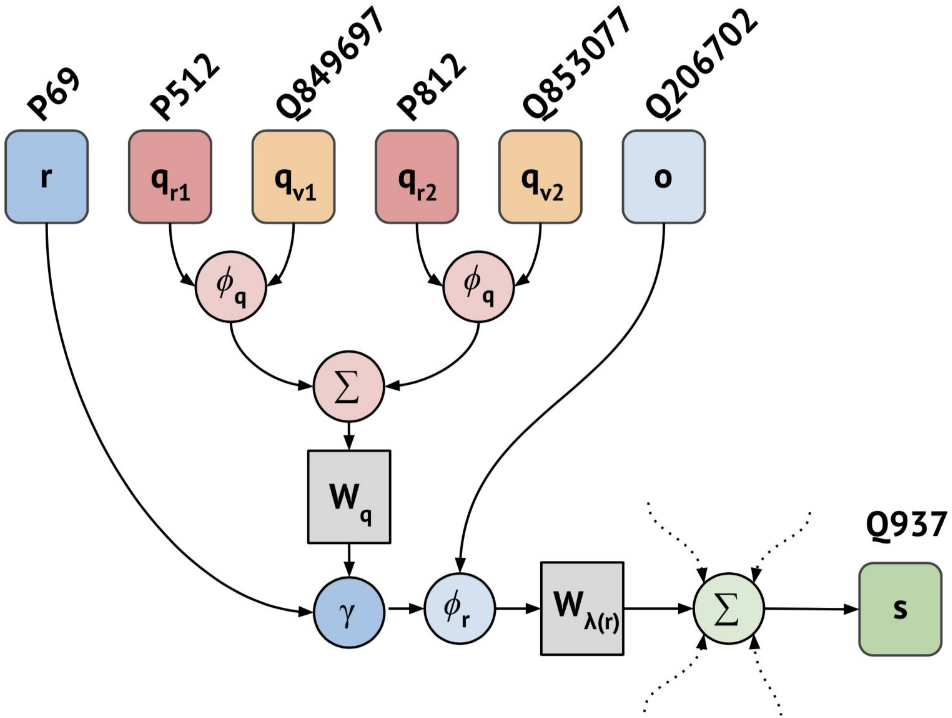
Embedding Hyper-Relational KGs

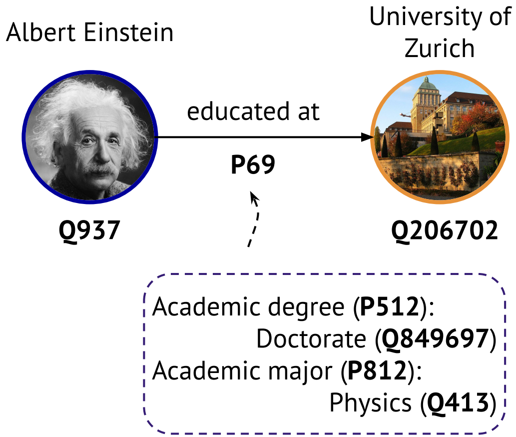


$$\boldsymbol{h}_v = f\left(\sum_{(u,r)\in\mathcal{N}(v)} \boldsymbol{W}_{\lambda(r)}\phi(\boldsymbol{x}_u, \boldsymbol{z}_r)\right)$$

$$\mathbf{h}_v = f\left(\sum_{(u,r)\in\mathcal{N}(v)} \mathbf{W}_{\lambda(r)}\phi_r(\mathbf{h}_u, \gamma(\mathbf{h}_r, \mathbf{h}_q)_{vu})\right)$$

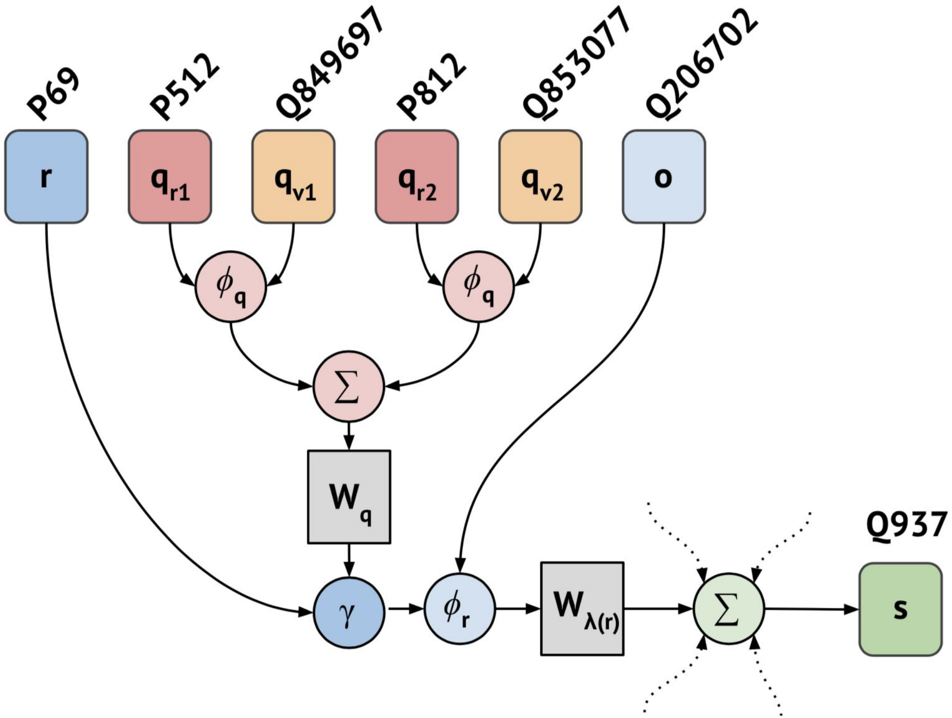
★ StarE: Embedding Hyper-Relational KGs

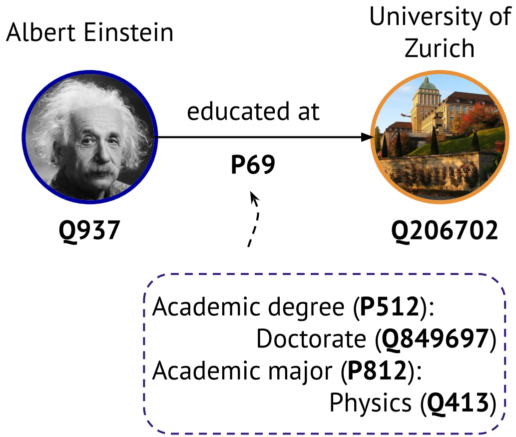




$$\mathbf{h}_v = f\left(\sum_{(u,r)\in\mathcal{N}(v)} \mathbf{W}_{\lambda(r)} \phi_r(\mathbf{h}_u, \gamma(\mathbf{h}_r, \mathbf{h}_q)_{vu}) \right)$$

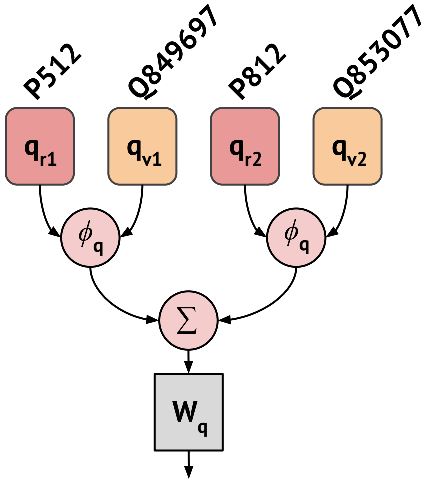
★ StarE: Embedding Hyper-Relational KGs





$$\mathbf{h}_v = f\left(\sum_{(u,r)\in\mathcal{N}(v)} \mathbf{W}_{\lambda(r)}\phi_r(\mathbf{h}_u, \gamma(\mathbf{h}_r, \boxed{\mathbf{h}_q})_{vu})\right)$$

★ StarE: Qualifiers Aggregation



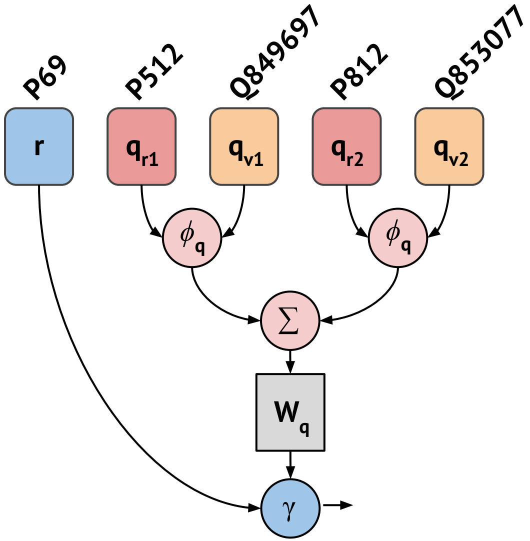
$$\mathbf{h}_q = \mathbf{W}_q \sum_{(qr,qv) \in Q_{jrvu}} \phi_q(\mathbf{h}_{qr}, \mathbf{h}_{qv})$$

ϕ_q is any entity-relation composition function like

- TransE
- HolE
- RotatE

etc

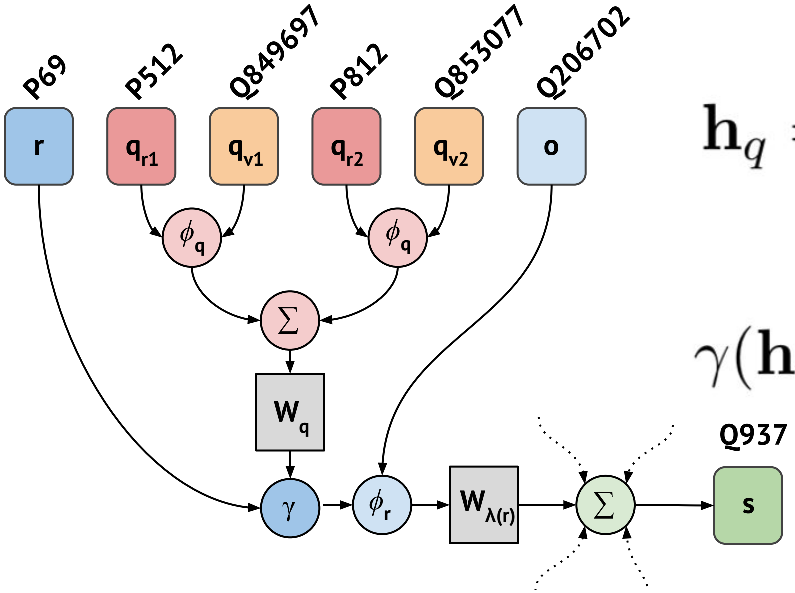
Combining Main Relation w/ Qualifiers



$$\mathbf{h}_q = \mathbf{W}_q \sum_{(qr,qv)\in Q_{jrvu}} \phi_q(\mathbf{h}_{qr}, \mathbf{h}_{qv})$$

$$\gamma(\mathbf{h}_r, \mathbf{h}_q) = \alpha \odot \mathbf{h}_r + (1-\alpha) \odot \mathbf{h}_q$$

Composing a Full Message



$$\mathbf{h}_q = \mathbf{W}_q \sum_{(qr,qv)\in Q_{jrvu}} \phi_q(\mathbf{h}_{qr}, \mathbf{h}_{qv})$$

$$\gamma(\mathbf{h}_r, \mathbf{h}_q) = \alpha \odot \mathbf{h}_r + (1-\alpha) \odot \mathbf{h}_q$$

$$\boldsymbol{W}_{\mathrm{dir}(r)} = \begin{cases} \boldsymbol{W}_O, & r \in \mathcal{R} \\ \boldsymbol{W}_I, & r \in \mathcal{R}_{inv} \\ \boldsymbol{W}_S, & r = \top \textit{ (self-loop)} \end{cases}$$

$$\mathbf{h}_v = f\left(\sum_{(u,r)\in\mathcal{N}(v)} \mathbf{W}_{\lambda(r)} \phi_r(\mathbf{h}_u, \gamma(\mathbf{h}_r, \mathbf{h}_q)_{vu}) \right)$$

Encoding Hyper-Relational KGs

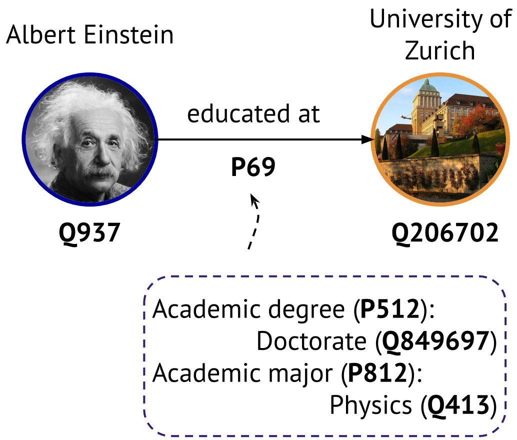
Sparse Triple Representation

s	Q937	…	…
o	Q206702	…	…
r	P69	…	…
index	*k*	*k+1*	*k+2*

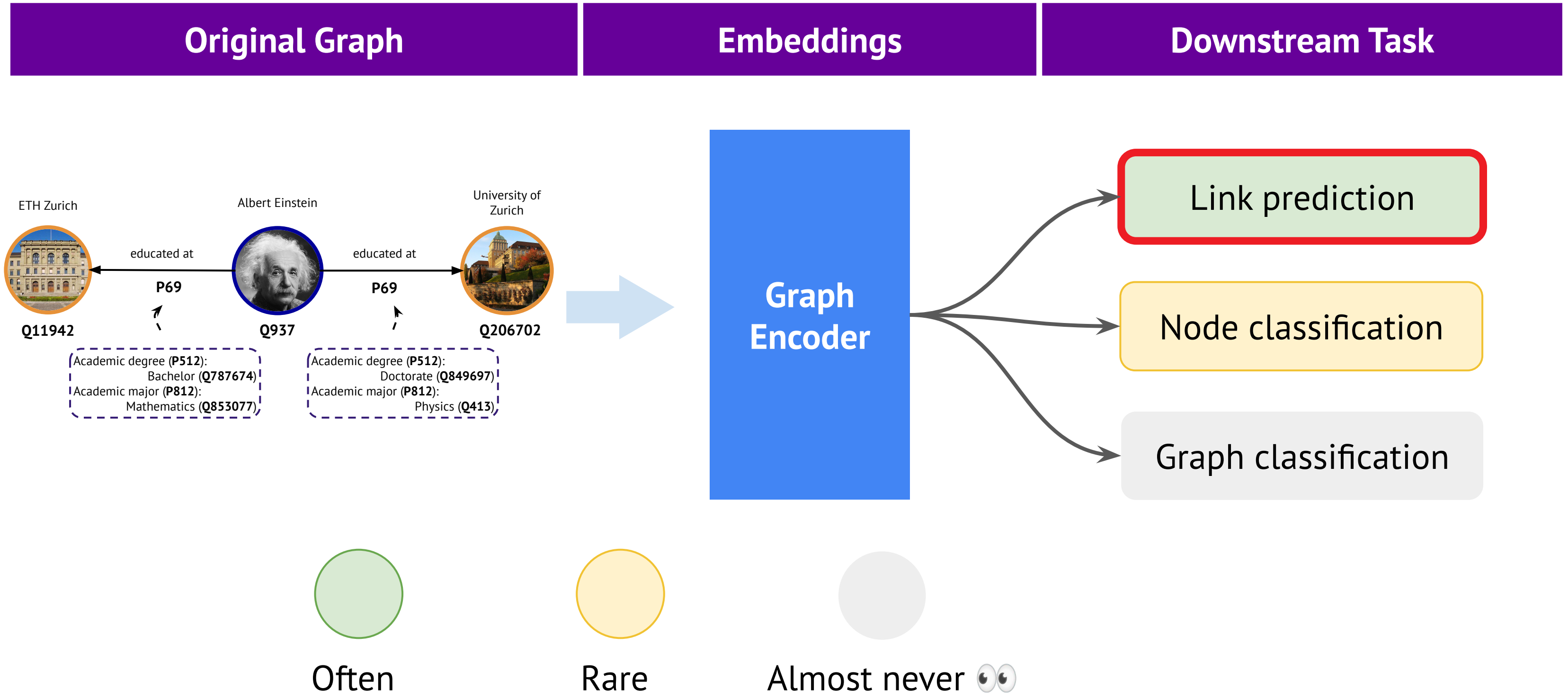
Sparse Qualifier Representation

index	*k*	*k*	…
qr	P812	P512	…
qv	Q413	Q849697	…

$$O(|\mathcal{E}| + |\mathcal{Q}|)$$ Space complexity



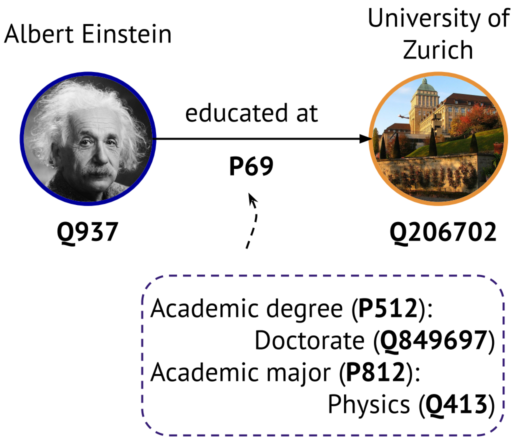
Graph Representation Learning Pipeline



Research Questions

1. Do qualifiers improve the accuracy of subject / object prediction of a fact?

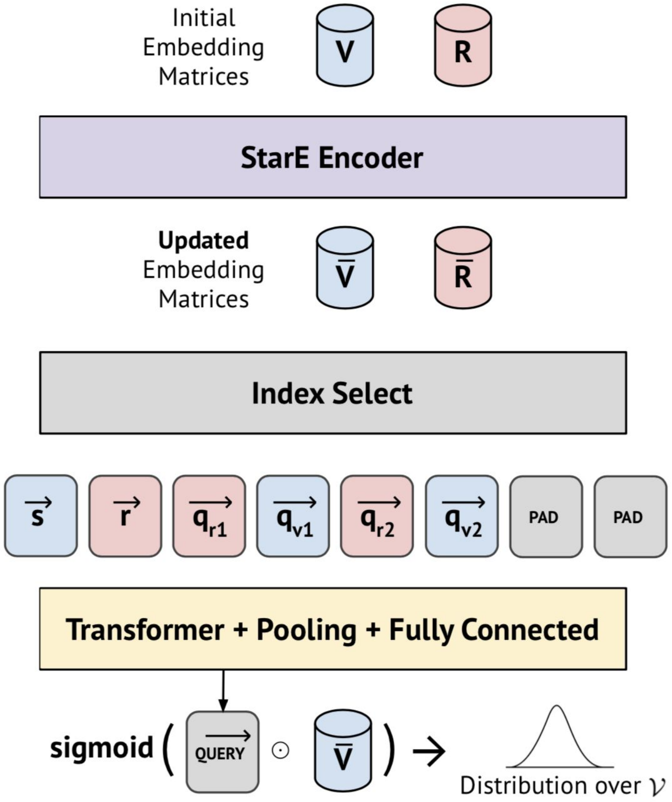
```
Q937, P69, ?
Q937, P69, ?, P512, Q849697, P812, Q413

?, P69, Q206702
?, P69, Q206702, P512, Q849697, P812, Q413
```



2. How many qualifiers do you need?

- In the whole KG
- Per fact

Decoders for Downstream Tasks



← A decoder for link prediction

- Datasets issue:
  - No such graphs in the Open Graph Benchmark
  - Very few LP datasets

Datasets Issue

Dataset	Entities	Statements	w/ Quals (%)	Relations	E in quals	R in quals	Train	Valid	Test
WikiPeople	34,839	369,866	9,482 (2.6%)	375	416	35	294,439	37,715	37,712
JF17K	28,645	100,947	46,320 (45.9%)	322	3652	180	76,379	-	24,568

WikiPeople

- Most statements contain datetime literals (usually dropped in LP)
- After cleaning: only ~3% of facts have qualifiers
    - 80% of which have 1 qualifier

JF17K

- Obtained from discontinued Freebase
- Test set leakages: ~45% of test facts have the same *(h, r, t)* in train
- Triple-only models can exploit the leak

WD50K: a Family of Hyper-Relational Datasets

Dataset	Entities	Statements	w/ Quals (%)	Relations	E in quals	R in quals	Train	Valid	Test
WD50K	47,156	236,507	32,167 (13.6%)	532	5460	45	166,435	23,913	46,159
WD50K (33)	38,124	102,107	31,866 (31.2%)	475	6463	47	73,406	10,568	18,133
WD50K (66)	27,347	49,167	31,696 (64.5%)	494	7167	53	35,968	5,154	8,045
WD50K (100)	18,792	31,314	31,314 (100%)	279	7862	75	22,738	3,279	5,297

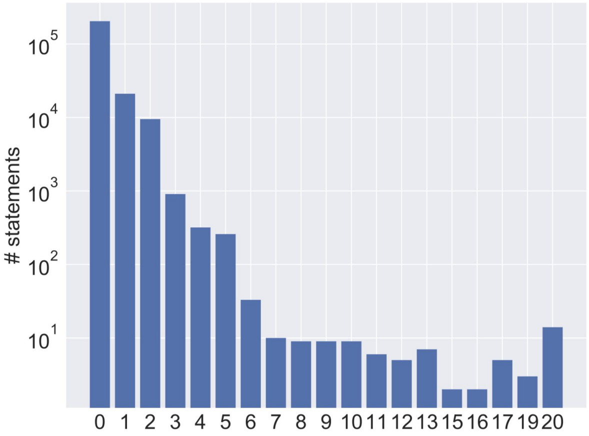


Figure 5: Number of qualifiers per statement

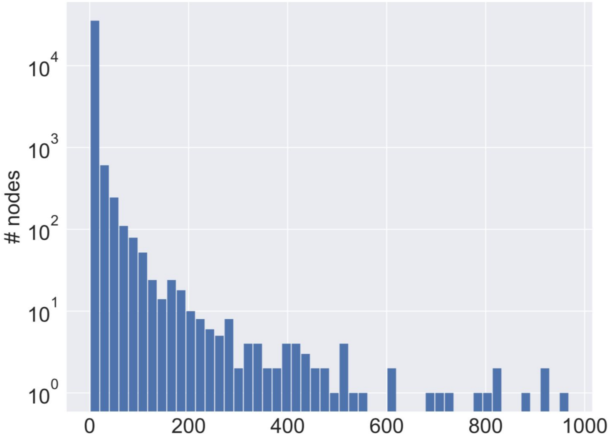


Figure 6: In-degree distribution

Hyper-Relational KGs: Link Prediction

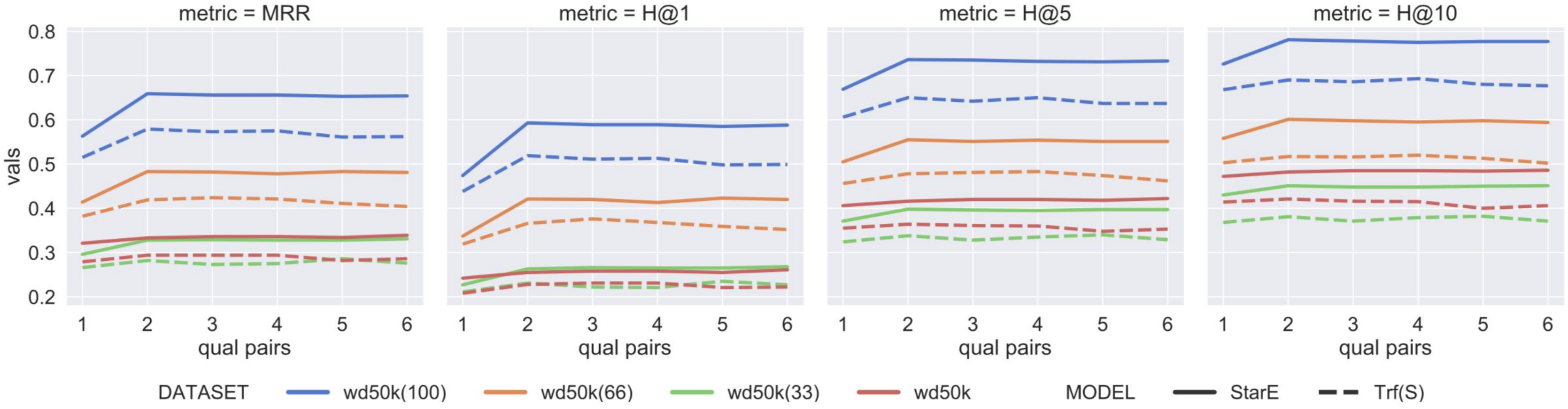
Exp #	Method	WikiPeople				JF17K			
		MRR	H@1	H@5	H@10	MRR	H@1	H@5	H@10
1	m-TransH	0.063	0.063	-	0.300	0.206	0.206	-	0.463
1	RAE	0.059	0.059	-	0.306	0.215	0.215	-	0.469
1	NaLP-Fix	0.420	0.343	-	0.556	0.245	0.185	-	0.358
1	HINGE	0.476	**0.415**	-	0.585	0.449	0.361	-	0.624
1,4	Transformer (H)	0.469	0.403	0.538	0.586	0.512	0.434	0.593	0.665
1,4	StarE (H) + Transformer(H)	**0.491**	0.398	**0.592**	**0.648**	**0.574**	**0.496**	**0.658**	**0.725**
4	Transformer (T)	0.474	0.419	0.532	0.575	0.537	0.473	0.606	0.663
4	StarE (T) + Transformer (T)	0.493	0.400	0.592	0.648	0.562	0.493	0.637	0.702

Hyper-Relational KGs: Link Prediction

Dataset →	WD50K			WD50K (33)			WD50K (66)			WD50K (100)		
Method ↓	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
Baseline (Transformer (T))	0.275	0.207	0.404	0.218	0.158	0.334	0.270	0.197	0.417	0.351	0.261	0.530
StarE (T) + Transformer(T)	0.308	0.228	0.465	0.246	0.173	0.388	0.297	0.212	0.470	0.380	0.276	0.584
NaLP-Fix	0.177	0.131	0.264	0.204	0.164	0.277	0.334	0.284	0.423	0.458	0.398	0.563
HINGE	0.243	0.176	0.377	0.253	0.190	0.372	0.378	0.307	0.512	0.492	0.417	0.636
Baseline (Transformer (H))	0.286	0.222	0.406	0.276	0.227	0.371	0.404	0.352	0.502	0.562	0.499	0.677
StarE (H) + Transformer(H)	**0.349**	**0.271**	**0.496**	**0.331**	**0.268**	**0.451**	**0.481**	**0.420**	**0.594**	**0.654**	**0.588**	**0.777**

- ★ Hyper-relational models **effectively** leverage qualifiers to improve predictions
- ★ The **more** hyper-relational facts - the **better** are predictions
- ★ The improvement upon triple-only models **grows with the ratio** of hyper-relational edges in the KG

Hyper-Relational KGs: Link Prediction



- ★ As few as **two** qualifiers per fact yield an observable improvement
- ★ StarE + decoder ensure the performance slowly **grows or remains stable** while decoder-only model shows degradation in performance

TL;DR

TECHNISCHE UNIVERSITÄT DRESDEN

SDA

Fraunhofer Dresden IAIS

1. **Hyper-relational** KGs are much more informative than vanilla KGs
2. Contribution: **StarE** - a GNN encoder for hyper-relational KGs (RDF* and LPG)
3. Contribution: **WD50K** - a new family of hyper-relational datasets for LP
4. Experiments: hyper-relational models **drastically** outperform triple baselines

Future work: design datasets for node and graph classification, and probe StarE

Code & Data https://github.com/migalkin/StarE

Contact mikhail.galkin@tu-dresden.de

Socials @michael_galkin

Multirelational GNN Encoders for KGs

GCN

$$\mathbf{h}_v^{(k)} = f\left(\sum_{u \in \mathcal{N}(v)} \mathbf{W}^{(k)} \mathbf{h}_u^{(k-1)}\right)$$

R-GCN

$$\mathbf{h}_v^{(k)} = f\left(\sum_{(u,r) \in \mathcal{N}(v)} \mathbf{W}_r^{(k)} \mathbf{h}_u^{(k-1)}\right)$$

CompGCN

$$\boldsymbol{h}_v = f\left(\sum_{(u,r) \in \mathcal{N}(v)} \boldsymbol{W}_{\lambda(r)} \phi(\boldsymbol{x}_u, \boldsymbol{z}_r)\right)$$

StarE

$$\mathbf{h}_v = f\left(\sum_{(u,r) \in \mathcal{N}(v)} \mathbf{W}_{\lambda(r)} \phi_r(\mathbf{h}_u, \gamma(\mathbf{h}_r, \mathbf{h}_q)_{vu})\right)$$