KGBench: Towards Understanding and Benchmarking Model Search for Knowledge Graph Embedding

Presenter: Zhanke Zhou

Advisors: Yongqi Zhang and Quanming Yao

2021. 07. 09

Outline

- Background
- Motivation
- Understanding of KGE components
- Searching experiments
- Key takeaway

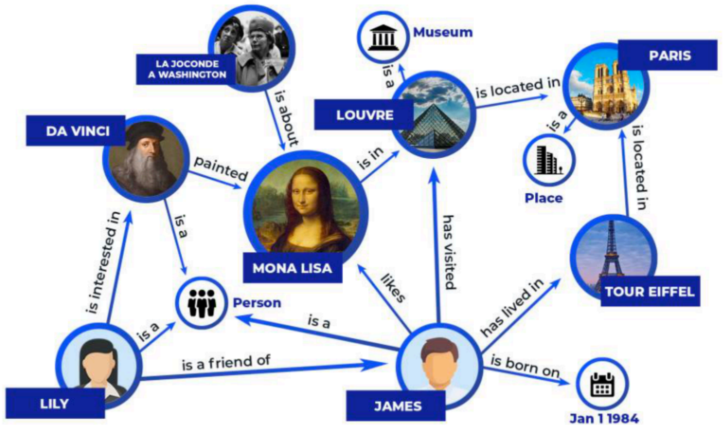
Background – Knowledge Graph (KG)

A knowledge graph

- Mainly describe real world entities and relations, organized in a graph
- Allows potentially interacting entities with each other

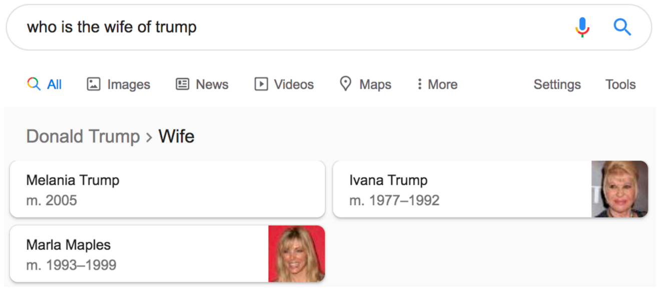
Preliminaries

- Graph representation: $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{F})$
- Entities $\mathcal{E}$
  - real world objects or concepts
- Relations $\mathcal{R}$
  - interactions between entities
- Facts $\mathcal{F}$
  - the basic unit in form of (h, r, t)
  - (head entity, relation, tail entity)

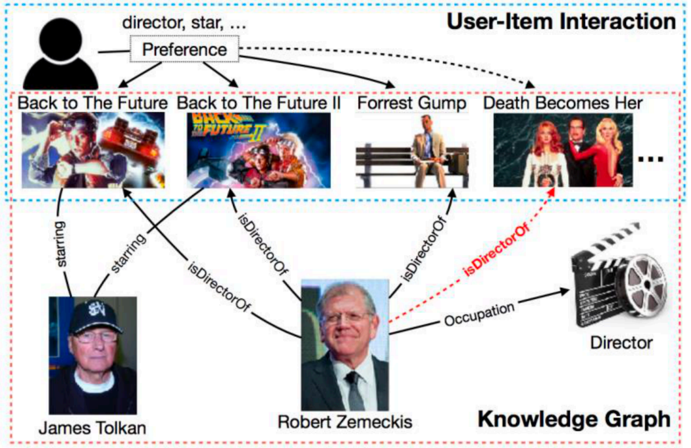


Applications

KGQA:

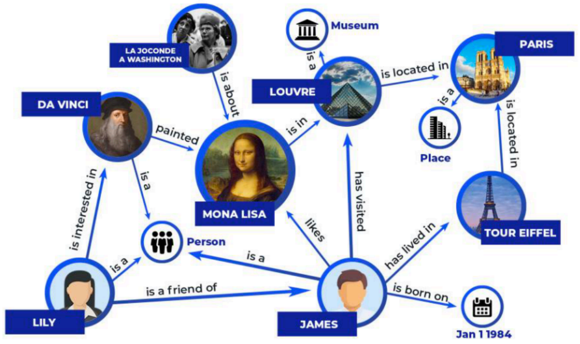


Recommendation:



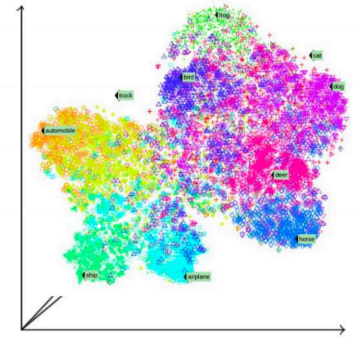
Background – Knowledge Graph Embedding (KGE)

- Knowledge Graph Embedding (KGE)
  - Encode entities and relations in KG into low-dimensional vectors space
  - while capturing nodes' and edges' connection properties



original KG

embedding



continuous space

- Most KGE models define a scoring function f to estimate the plausibility of any fact (h, r, t) using their embeddings: $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$

Background – Knowledge Graph Embedding (KGE)

- Training
  - S^+: positive samples S^-: negative samples
  - Objectives: $\max f(S^+)$ and $\min f(S^-)$
- Inference
  - head/tail prediction $(?, r, t)/(h, r, ?)$
  - the missing tail is inferred as the entity that results in the highest score:

$$t = \operatorname*{argmax}_{e \in \mathcal{E}} f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{e})$$

- Evaluation metrics
  - q: the **rank** of correct entity

$$MR = \frac{1}{|Q|} \sum_{q \in Q} q \qquad MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{q} \qquad H@K = \frac{|\{q \in Q : q \leq K\}|}{|Q|}$$

Machine learning on Knowledge Graph

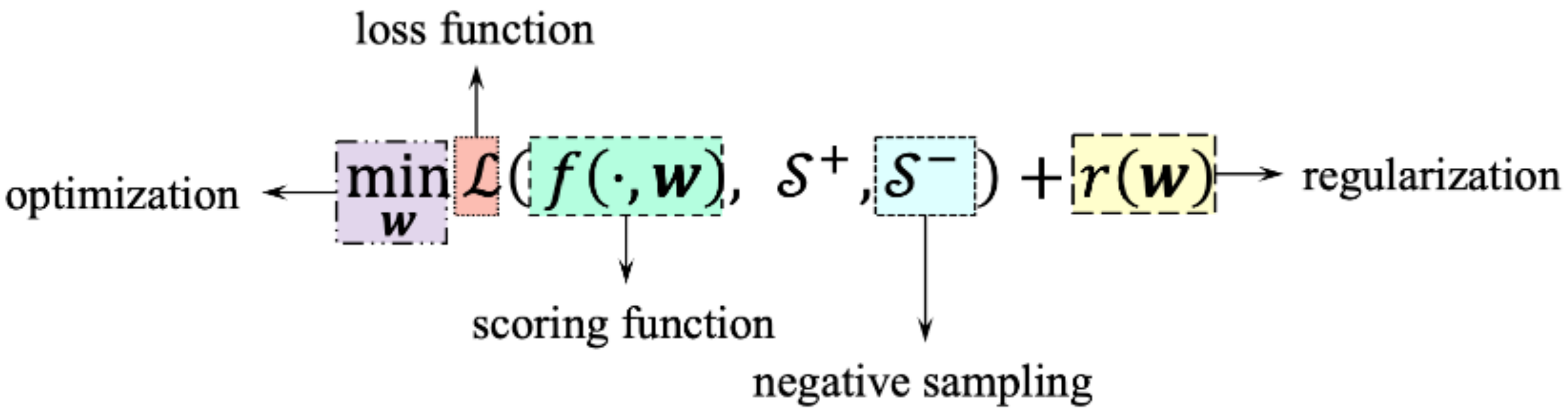
For obtaining embeddings of entities and relations, and finishing KGE task (e.g., predict potential facts)

- **the scoring function f** is expected to discriminate positive/negative factual triples
- a sampling scheme is needed to generate **negative samples S^-**
- a **loss function L** and **regularization r** are required for defining learning problem
- a **optimization** strategy is needed for convergence procedure

Considering the above 5 factors,
we can formulate the KGE learning framework as:

$$\underbrace{\min_{\boldsymbol{w}}}_{\text{optimization}} \overbrace{\mathcal{L}}^{\text{loss function}}\Big(\underbrace{f(\cdot, \boldsymbol{w})}_{\text{scoring function}}, \mathcal{S}^+, \underbrace{\mathcal{S}^-}_{\text{negative sampling}}\Big) + \underbrace{r(\boldsymbol{w})}_{\text{regularization}}$$

Learning Framework of KGE



5 KGE components:

Component	Role	Inputs	Outputs	Example
Scoring function $f()$	Plausibility estimation	Factual triples	Scores for each triple	TransE / CP
Loss function $L()$	Learning problem definition	Scores and labels	Loss value	BCE / CE
Negative sampling	Budget tradeoff	Positive samples S^+	Negative samples S^-	Uniform
Regularization $r()$	Avoid overfitting	Learnable parameters	Regularization value	L2 / N3
Optimization	Convergence control	-	-	-

Learning Framework of KGE

Learning objective:

$$\underbrace{\min_{\mathbf{w}}}_{\text{optimization}} \overbrace{\mathcal{L}}^{\text{loss function}}\Big(\underbrace{f(\cdot,\mathbf{w})}_{\text{scoring function}},\ \mathcal{S}^+,\ \underbrace{\mathcal{S}^-}_{\text{negative sampling}}\Big) + \underbrace{r(\mathbf{w})}_{\text{regularization}}$$

Training procedure of knowledge graph embedding

Input data: training triples S_{tra}

repeat
mini-batch training
until convergence

- step1: initialize learnable parameters w (embeddings / model weights)
- step2: sample negative triples $\tilde{S}_{(h,r,t)}$ (S^-) for each positive triple $(h,r,t) \in S_{tra}$ (S^+)
- step3: $f(\ )$ forward inference to obtain $Scores$ for triples in $\{(h,r,t)\} \cup \tilde{S}_{(h,r,t)}$
- step4: compute loss and regularization term w.r.t. $L(\ )$ and $r(\ )$
- step5: backward propagation, and update w

Output: w

Review of Current KGE Models

		FB15k				WN18				FB15k-237				WN18RR				YAGO3-10			
		H@1	H@10	MR	MRR	H@1	H@10	MR	MRR	H@1	H@10	MR	MRR	H@1	H@10	MR	MRR	H@1	H@10	MR	MRR
Tensor Decomposition Models	DistMult	73.61	86.32	173	0.784	72.60	94.61	675	0.824	22.44	49.01	199	0.313	39.68	50.22	5913	0.433	41.26	66.12	1107	0.501
	ComplEx	81.56	90.53	34	0.848	94.53	95.50	3623	0.949	25.72	52.97	202	0.349	42.55	52.12	4907	0.458	50.48	70.35	1112	0.576
	ANALOGY	65.59	83.74	126	0.726	92.61	94.42	808	0.934	12.59	35.38	476	0.202	35.82	38.00	9266	0.366	19.21	45.65	2423	0.283
	SimplE	66.13	83.63	138	0.726	93.25	94.58	759	0.938	10.03	34.35	651	0.179	38.27	42.65	8764	0.398	35.76	63.16	2849	0.453
	HolE	75.85	86.78	211	0.800	93.11	94.94	650	0.938	21.37	47.64	186	0.303	40.28	48.79	8401	0.432	41.84	65.19	6489	0.502
	TuckER	72.89	88.88	39	0.788	94.64	95.80	510	0.951	25.90	53.61	162	0.352	42.95	51.40	6239	0.459	46.56	68.09	2417	0.544
Geometric Models	TransE	49.36	84.73	45	0.628	40.56	94.87	279	0.646	21.72	49.65	209	0.31	2.79	49.52	3936	0.206	40.57	67.39	1187	0.501
	STransE	39.77	79.60	69	0.543	43.12	93.45	208	0.656	22.48	49.56	357	0.315	10.13	42.21	5172	0.226	3.28	7.35	5797	0.049
	CrossE	60.08	86.23	136	0.702	73.28	95.03	441	0.834	21.21	47.05	227	0.298	38.07	44.99	5212	0.405	33.09	65.45	3839	0.446
	TorusE	68.85	83.98	143	0.746	94.33	95.44	525	0.947	19.62	44.71	211	0.281	42.68	53.35	4873	0.463	27.43	47.44	19455	0.342
	RotatE	73.93	88.10	42	0.791	94.30	96.02	274	0.949	23.83	53.06	178	0.336	42.60	57.35	3318	0.475	40.52	67.07	1827	0.498
Deep Learning Models	ConvE	59.46	84.94	51	0.688	93.89	95.68	413	0.945	21.90	47.62	281	0.305	38.99	50.75	4944	0.427	39.93	65.75	2429	0.488
	ConvKB	11.44	40.83	324	0.211	52.89	94.89	202	0.709	13.98	41.46	309	0.230	5.63	52.50	3429	0.249	32.16	60.47	1683	0.420
	ConvR	70.57	88.55	70	0.773	94.56	95.85	471	0.950	25.56	52.63	251	0.346	43.73	52.68	5646	0.467	44.62	67.33	2582	0.527
	CapsE	1.93	21.78	610	0.087	84.55	95.08	233	0.890	7.34	35.60	405	0.160	33.69	55.98	720	0.415	0.00	0.00	60676	0.000
	RSN	72.34	87.01	51	0.777	91.23	95.10	346	0.928	19.84	44.44	248	0.280	34.59	48.34	4210	0.395	42.65	66.43	1339	0.511
	AnyBURL	81.09	87.86	288	0.835	94.63	95.96	233	0.951	24.03	48.93	480	0.324	44.93	55.97	2530	0.485	45.83	66.07	815	0.528

[1]

		RESCAL	TransE	DistMult	ComplEx	ConvE
FB15K-237	Valid. MRR	*36.1*	*31.5*	*35.0*	*35.3*	*34.3*
	Emb. size	128 *(-0.5)*	512 *(-3.7)*	256 *(-0.2)*	256 *(-0.3)*	256 *(-0.4)*
	Batch size	512 *(-0.5)*	128 *(-7.1)*	1024 *(-0.2)*	1024 *(-0.3)*	1024 *(-0.4)*
	Train type	1vsAll *(-0.8)*	NegSamp –	NegSamp *(-0.2)*	NegSamp *(-0.3)*	1vsAll *(-0.4)*
	Loss	CE *(-0.9)*	CE *(-7.1)*	CE *(-3.1)*	CE *(-3.8)*	CE *(-0.4)*
	Optimizer	Adam *(-0.5)*	Adagrad *(-3.7)*	Adagrad *(-0.2)*	Adagrad *(-0.5)*	Adagrad *(-1.5)*
	Initializer	Normal *(-0.8)*	XvNorm *(-3.7)*	Unif. *(-0.2)*	Unif. *(-0.5)*	XvNorm *(-0.4)*
	Regularizer	None *(-0.5)*	L2 *(-3.7)*	L3 *(-0.2)*	L3 *(-0.3)*	L3 *(-0.4)*
	Reciprocal	No *(-0.5)*	Yes *(-9.5)*	Yes *(-0.3)*	Yes *(-0.3)*	Yes –
WNRR	Valid. MRR	*46.8*	*22.6*	*45.4*	*47.6*	*44.3*
	Emb. size	128 *(-1.0)*	512 *(-5.1)*	512 *(-1.1)*	128 *(-1.0)*	512 *(-1.2)*
	Batch size	128 *(-1.0)*	128 *(-5.1)*	1024 *(-1.1)*	512 *(-1.0)*	1024 *(-1.3)*
	Train type	KvsAll *(-1.0)*	NegSamp –	KvsAll *(-1.1)*	1vsAll *(-1.0)*	KvsAll *(-1.2)*
	Loss	CE *(-2.0)*	CE *(-5.1)*	CE *(-2.4)*	CE *(-3.5)*	CE *(-1.4)*
	Optimizer	Adam *(-1.2)*	Adagrad *(-5.8)*	Adagrad *(-1.5)*	Adagrad *(-1.5)*	Adam *(-1.4)*
	Initializer	Unif. *(-1.0)*	XvNorm *(-5.1)*	Unif. *(-1.3)*	Unif. *(-1.5)*	XvNorm *(-1.4)*
	Regularizer	L3 *(-1.2)*	L2 *(-5.1)*	L3 *(-1.1)*	L2 *(-1.0)*	L1 *(-1.2)*
	Reciprocal	Yes *(-1.0)*	Yes *(-5.9)*	Yes *(-1.1)*	No *(-1.0)*	Yes –

[2]

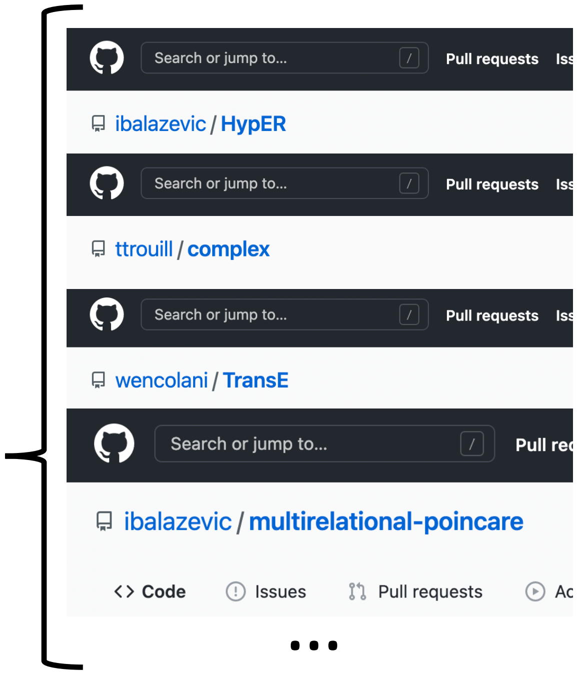
No best models 🤔

No best configurations 🤔

Motivation and Objective

Difficulties and Challenges

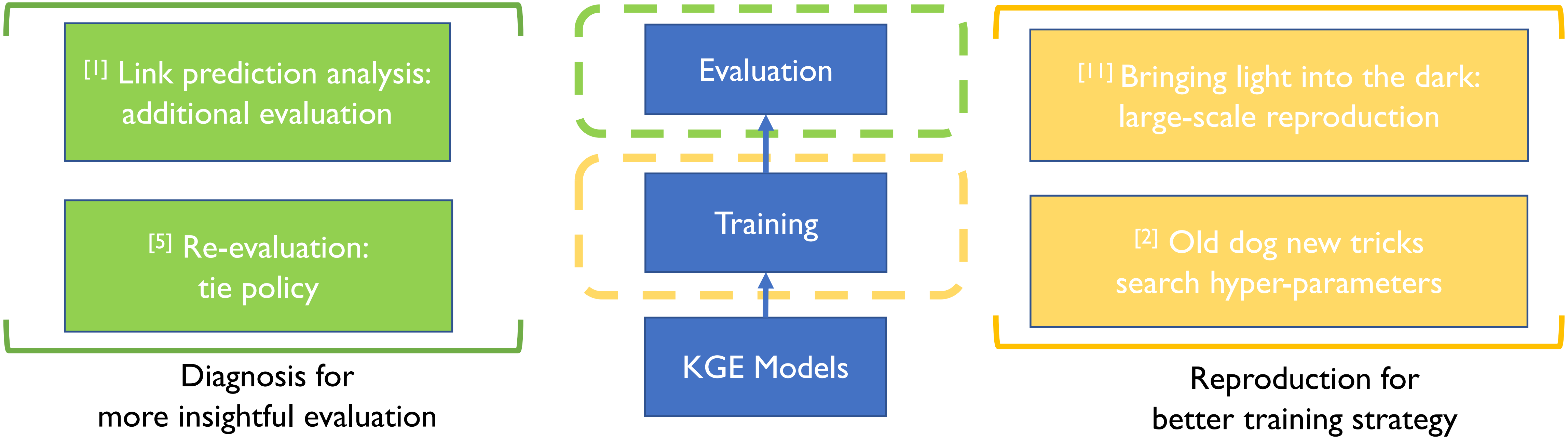
1. The choice of KGE model and configuration
   - usually in a time-consuming trial-and-error way
2. A fair comparison of model or strategy
   - due to the heterogeneity in implementation, training, and evaluation
3. Lacking understanding of KGE components
   - interaction, importance, and tunability are unclear



Ultimate objective of KGbench:

- *Design an AutoML approach,*
- *for any given dataset,*
- *with requirements and limited budget,*
- *to search for the optimal KGE model and configuration*

Comparing with related works



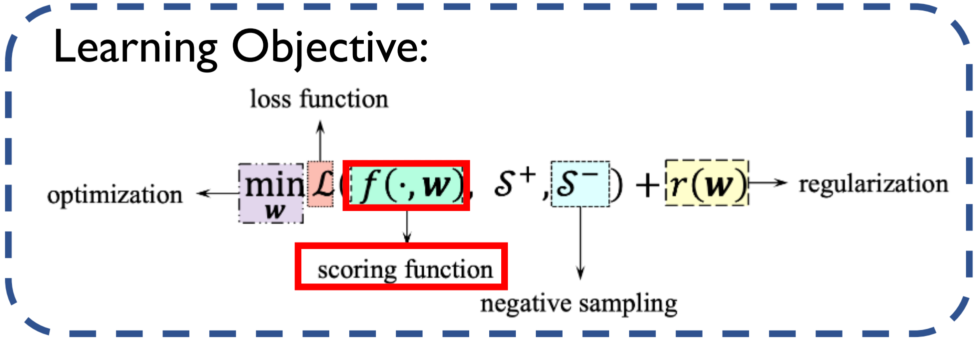
KGbench

- Design space(s) for KGE: model (configuration) / dataset / task
- Deep insights and theoretical analysis of KGE components
- Efficient automatic search for optimal model and configuration

Outline

- Background
- Motivation
- Understanding of KGE components
  - Part1: Scoring Function $f(\,)$
  - Part2: Loss Function $L(\,)$
  - Part3: Negative Sampling S^-
  - …
- Searching experiments
- Key takeaway

Part1: Scoring Function $f(\,)$

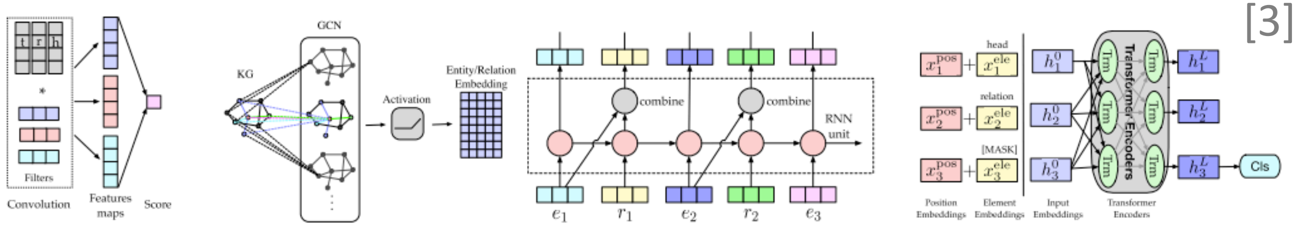


Category

- Triple-based (focus point)
  - geometric models → need additional constraints
  - tensor decomposition models → expressive
  - neural network models → more prone to overfitting
- Path / (Sub) Graph-based
  - utilize observable topological features
- Rule-based
  - logical rule mining

[3]

Model	Ent. & Rel. embed.	Scoring Function $f_r(h,t)$
RotatE [24]	$\mathbf{h},\mathbf{t} \in \mathbb{C}^d$, $\mathbf{r} \in \mathbb{C}^d$	$\Vert \mathbf{h} \circ \mathbf{r} - \mathbf{t} \Vert$
TorusE [15]	$[\mathbf{h}],[\mathbf{t}] \in \mathbb{T}^n$, $[\mathbf{r}] \in \mathbb{T}^n$	$\min_{(x,y)\in([h]+[r])\times[t]} \Vert x-y \Vert_i$
SimplE [48]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r},\mathbf{r}' \in \mathbb{R}^d$	$\frac{1}{2}\left(\mathbf{h} \circ \mathbf{r}\mathbf{t} + \mathbf{t} \circ \mathbf{r}'\mathbf{t}\right)$
TuckER [52]	$\mathbf{h},\mathbf{t} \in \mathbb{R}_e^d$, $\mathbf{r} \in \mathbb{R}_r^d$	$\mathcal{W} \times_1 \mathbf{h} \times_2 \mathbf{r} \times_3 \mathbf{t}$
ITransF [36]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\left\Vert \boldsymbol{\alpha}_r^H \cdot \mathbf{D} \cdot \mathbf{h} + \mathbf{r} - \boldsymbol{\alpha}_r^T \cdot \mathbf{D} \cdot \mathbf{t} \right\Vert_\ell$
HolEx [40]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\sum_{j=0}^{l} p\left(\mathbf{h}, \boldsymbol{r}; \mathbf{c}_j\right) \cdot \boldsymbol{t}$
CrossE [42]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\sigma\left(\sigma\left(\mathbf{c}_r \circ \mathbf{h} + \mathbf{c}_r \circ \mathbf{h} \circ \mathbf{r} + \mathbf{b}\right)\mathbf{t}^\top\right)$
QuatE [25]	$\mathbf{h},\mathbf{t} \in \mathbb{H}^d$, $\mathbf{r} \in \mathbb{H}^d$	$\mathbf{h} \otimes \frac{\mathbf{r}}{\vert\mathbf{r}\vert} \cdot \mathbf{t}$
SACN [44]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$g\left(\operatorname{vec}\left(\mathbf{M}\left(\mathbf{h},\mathbf{r}\right)\right) W\right)\mathbf{t}$
ConvKB [43]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$	$\operatorname{concat}\left(g\left([\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}] * \omega\right)\right)\mathbf{w}$
ConvE [55]	$\mathbf{M}_h \in \mathbb{R}^{d_w \times d_h}$, $\mathbf{t} \in \mathbb{R}^d$ $\mathbf{M}_r \in \mathbb{R}^{d_w \times d_h}$	$\sigma\left(\operatorname{vec}\left(\sigma\left([\mathbf{M}_h; \mathbf{M}_r] * \boldsymbol{\omega}\right)\right)\mathbf{W}\right)\mathbf{t}$
DihEdral [31]	$\mathbf{h}^{(l)}, \mathbf{t}^{(l)} \in \mathbb{R}^2$ $\mathbf{R}^{(l)} \in \mathbb{D}_K$	$\sum_{l=1}^{L} \mathbf{h}^{(l)\top}\mathbf{R}^{(l)}\mathbf{t}^{(l)}$
HAKE [19]	$\mathbf{h}_m, \mathbf{t}_m \in \mathbb{R}^d, \mathbf{r}_m \in \mathbb{R}_+^d$ $\mathbf{h}_p, \mathbf{r}_p, \mathbf{t}_p \in [0, 2\pi)^d$	$-\Vert \mathbf{h}_m \circ \mathbf{r}_m - \mathbf{t}_m \Vert_2 -$ $\lambda \Vert \sin\left((\mathbf{h}_p + \mathbf{r}_p - \mathbf{t}_p)/2\right) \Vert_1$
MuRP [29]	$\mathbf{h},\mathbf{t},\mathbf{r} \in \mathbb{B}_c^d, b_h, b_t \in \mathbb{R}$	$-d_{\mathbb{B}}\left(\mathbf{h}^{(r)}, \mathbf{t}^{(r)}\right)^2 + b_s + b_o$
AttH [30]	$\mathbf{h},\mathbf{t},\mathbf{r} \in \mathbb{B}_c^d, b_h, b_t \in \mathbb{R}$	$-d_{\mathbb{B}}^{c_r}\left(Q(h,r), \mathbf{e}_t^H\right)^2 + b_h + b_t$
LowFER [53]	$\mathbf{h},\mathbf{t} \in \mathbb{R}^d, \mathbf{r} \in \mathbb{R}^d$	$\left(\mathbf{S}^k \operatorname{diag}\left(\mathbf{U}^T\mathbf{h}\right)\mathbf{V}^T\mathbf{r}\right)^T\mathbf{t}$



(a) CNN. (b) GCN. (c) RSN. (d) Transformer.

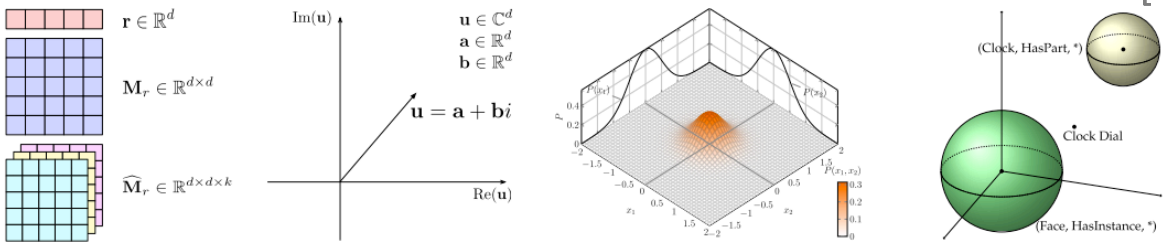
[3]

Part1: Scoring Function $f(\,)$

Questions to answer for developing a novel f

1) which representation space to choose
2) which encoding model to use for modeling relational interactions (encoder)
3) how to measure the plausibility of triplets in a specific space (decoder)
4) whether to utilize auxiliary information

[3]



(a) Point-wise space. (b) Complex vector space. (c) Gaussian distribution. (d) Manifold space.

KGbench: $f(h, r, t) = \delta(\phi(\boldsymbol{h}, \boldsymbol{r}), \boldsymbol{t})$

decoupling and re-combination of existing f

- Real/complex vector space
- f is the combination of candidate ϕ and δ
- Not requiring additional information

model	$\phi(\boldsymbol{h}, \boldsymbol{r})$	shared $\boldsymbol{r}^{-1}$	$\delta(\boldsymbol{v}, \boldsymbol{t})$
TransE [2]	$\boldsymbol{v} = \boldsymbol{h} + \boldsymbol{r}$	$\boldsymbol{r}^{-1} = -\boldsymbol{r}$	$-\Vert\boldsymbol{v} - \boldsymbol{t}\Vert_1$
RotatE [13]	$\boldsymbol{v} = \boldsymbol{h} \circ \boldsymbol{r} = [\boldsymbol{h}_{re}\boldsymbol{r}'_{re} - \boldsymbol{h}_{im} \cdot \boldsymbol{r}'_{im}, \boldsymbol{h}_{re}\boldsymbol{r}'_{im} + \boldsymbol{h}_{im}\boldsymbol{r}'_{re}]$ with $[\boldsymbol{h}_{re}, \boldsymbol{h}_{im}] = \boldsymbol{h}$, $[\boldsymbol{r}'_{re}, \boldsymbol{r}'_{im}] = \frac{[\boldsymbol{r}_{re}, \boldsymbol{r}_{im}]}{\sqrt{\boldsymbol{r}_{re}^2 + \boldsymbol{r}_{im}^2}}$ and $[\boldsymbol{r}_{re}, \boldsymbol{r}_{im}] = \boldsymbol{r}$	$\boldsymbol{r}^{-1} = \mathrm{conj}(\boldsymbol{r})$	$-\Vert\boldsymbol{v} - \boldsymbol{t}\Vert_{c1}$
DistMult [17]	$\boldsymbol{v} = \boldsymbol{h} \cdot \boldsymbol{r}$	$\boldsymbol{r}^{-1} = \boldsymbol{r}$	$\langle \boldsymbol{v}, \boldsymbol{t} \rangle$
ComplEx [14]	$\boldsymbol{v} = [\boldsymbol{h}_{re}\boldsymbol{r}_{re} - \boldsymbol{h}_{im}\boldsymbol{r}_{im}, \boldsymbol{h}_{re}\boldsymbol{r}_{im} + \boldsymbol{h}_{im}\boldsymbol{r}_{re}]$ with $[\boldsymbol{h}_{re}, \boldsymbol{h}_{im}] = \boldsymbol{h}$ and $[\boldsymbol{r}_{re}, \boldsymbol{r}_{im}] = \boldsymbol{r}$	$\boldsymbol{r}^{-1} = \mathrm{conj}(\boldsymbol{r})$	$\langle \boldsymbol{v}, \boldsymbol{t} \rangle$
BLM [20]	$\boldsymbol{v} = \boldsymbol{h}\boldsymbol{R}_{\boldsymbol{r}}$	$\boldsymbol{R}_{\boldsymbol{r}^{-1}} = \boldsymbol{R}_{\boldsymbol{r}}^{\top}$	$\langle \boldsymbol{v}, \boldsymbol{t} \rangle$
ConvE [4]	$\boldsymbol{v} = \sigma\big(\mathrm{vec}(\sigma([\bar{\boldsymbol{h}}, \bar{\boldsymbol{r}}] * \boldsymbol{\omega}))\boldsymbol{W}\big)$	$\times$	$\langle \boldsymbol{v}, \boldsymbol{t} \rangle$

In progress

Part2: Loss Function $L(\ )$

Learning Objective:

$$\underbrace{\min_{\mathbf{w}}}_{\text{optimization}} \overbrace{\boxed{\mathcal{L}}}^{\text{loss function}}\big(\underbrace{f(\cdot,\mathbf{w})}_{\text{scoring function}},\ \mathcal{S}^+, \underbrace{\mathcal{S}^-}_{\text{negative sampling}}\big) + \underbrace{r(\mathbf{w})}_{\text{regularization}}$$

Category

- Point-wise $$\min_{\omega} \sum_{(h,r,t)\in\mathcal{S}^+\cup\mathcal{S}^-} \mathcal{L}(f(h,r,t;\omega))$$
- Pair-wise $$\min_{\omega} \sum_{(h,r,t)\in\mathcal{S}^+} \sum_{(\tilde{h},r,\tilde{t})\in\mathcal{S}^-} \mathcal{L}(f(h,r,t;\omega), f(\tilde{h},r,\tilde{t};\omega))$$
- Set-wise $$\min_{\omega} \mathcal{L}\left\{f(h,r,t;\omega) | (h,r,t) \in \mathcal{S}^+ \cup \mathcal{S}^-\right\}$$

Comparison scope	#positive samples	#negative samples	Representatives
Point-wise	1 (0)	0 (1)	BCE / MSE
Pair-wise	1	1	MR
Set-wise	All (including peers)	All	CE / NLL / Adversarial

Part3: Negative Sampling S^-

Learning Objective:

$$\underbrace{\min_{\boldsymbol{w}}}_{\text{optimization}} \underbrace{\mathcal{L}}_{\text{loss function}}\big(\underbrace{f(\cdot,\boldsymbol{w})}_{\text{scoring function}},\ \mathcal{S}^+\ \underbrace{\boxed{\mathcal{S}^-}}_{\text{negative sampling}}\big) + \underbrace{r(\boldsymbol{w})}_{\text{regularization}}$$

positive $(h,r,t) \to$ negative $(\tilde{h},r,t)$ or $(h,r,\tilde{t})$

Methods

- Uniform / Bernoulli sampling — Efficient
- GAN-based (with additional parameters to learn)
- Cache(score)-based (NSCaching ICDE 2019)
- Bias/variance-based (SRNS NeurIPS 2020)

Effective

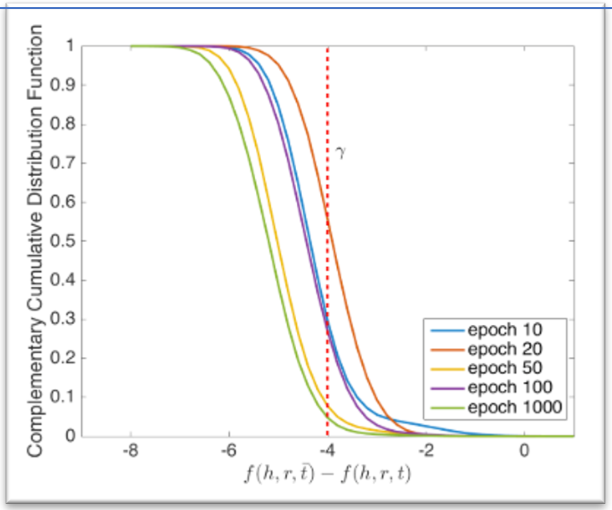
Importance weighting: [4]

$$p(\bar{h}|(t,r)) = \frac{\exp(f(\bar{h},r,t))}{\sum_{h_i \in \hat{\mathcal{H}}_{(r,t)}} \exp(f(\bar{h}_i,r,t))}$$

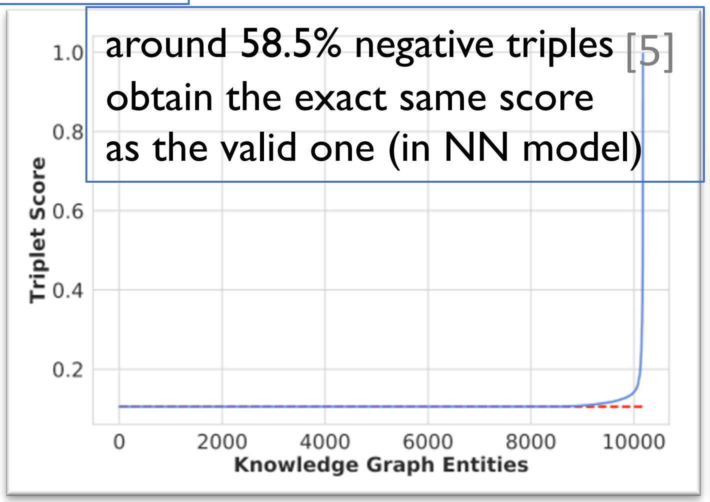
Self-contrast approximation: [7]

$$p_n(u|v) \propto p_d(u|v)^{\alpha} \approx \frac{(E_\theta(u)\cdot E_\theta(v))^{\alpha}}{\sum_{u'\in U}(E_\theta(u')\cdot E_\theta(v))^{\alpha}}$$

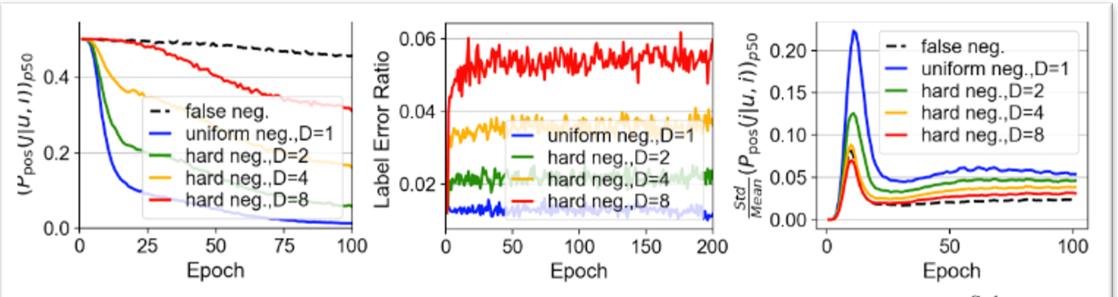
negative triples with large scores are rare. [4]



around 58.5% negative triples obtain the exact same score as the valid one (in NN model) [5]



- Both false and hard negative instances have large scores,
- false negative instances have lower prediction variance [6]



(b) Comparing P_{pos} of neg. (c) Comparing LER. (d) Comparing $\frac{\text{Std}}{\text{Mean}} P_{\text{pos}}$.

Outline

- Background
- Motivation
- Understanding of KGE components
  - …
  - Part4: Regularization $r(\ )$ [NeurIPS 2020]
  - Part5: Optimization
- Searching experiments
- Key takeaway

Part4: Regularization $r(\,)$

- $\boldsymbol{r}(\,)$: to avoid overfitting in KGE
  - trade off between expressiveness and complexity
- No general & promising regularization schemes
  - squared frobenius norm (L2 norm)
  - tensor nuclear 3-norm (N3 norm)
    - designed for CP-like tensor decomposition models

Learning Objective:

loss function

optimization $\leftarrow$ $\min_{\boldsymbol{w}} \mathcal{L}(f(\cdot, \boldsymbol{w}), \mathcal{S}^+, \mathcal{S}^-) + \boxed{r(\boldsymbol{w})}$ $\rightarrow$ regularization

scoring function

negative sampling

$$r_{FRO} = \|\boldsymbol{H}\|_F^2 + \|\boldsymbol{T}\|_F^2 + \sum_{j=1}^{|R|} \left\|\boldsymbol{R}_j\right\|_F^2$$

$$r_{N3} = \sum_{d=1}^{|D|} (\|\boldsymbol{h}_{:d}\|_3^3 + \|\boldsymbol{r}_{:d}\|_3^3 + \|\boldsymbol{t}_{:d}\|_3^3)$$

Method	Scoring function $f_r(h,t)$	Constraints/Regularization
TransE [14]	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
TransH [15]	$-\|(\mathbf{h}-\mathbf{w}_r^\top \mathbf{h}\mathbf{w}_r)+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top \mathbf{t}\mathbf{w}_r)\|_2^2$	$\|\mathbf{h}\|_2 \le 1, \|\mathbf{t}\|_2 \le 1$ $\|\mathbf{w}_r^\top \mathbf{r}\|/\|\mathbf{r}\|_2 \le \epsilon, \|\mathbf{w}_r\|_2 = 1$
TransR [16]	$-\|\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\|_2^2$	$\|\mathbf{h}\|_2 \le 1, \|\mathbf{t}\|_2 \le 1, \|\mathbf{r}\|_2 \le 1$ $\|\mathbf{M}_r\mathbf{h}\|_2 \le 1, \|\mathbf{M}_r\mathbf{t}\|_2 \le 1$
TransD [50]	$-\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2^2$	$\|\mathbf{h}\|_2 \le 1, \|\mathbf{t}\|_2 \le 1, \|\mathbf{r}\|_2 \le 1$ $\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}\|_2 \le 1$ $\|(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2 \le 1$
TranSparse [51]	$-\|\mathbf{M}_r(\theta_r)\mathbf{h}+\mathbf{r}-\mathbf{M}_r(\theta_r)\mathbf{t}\|_{1/2}^2$ $-\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}+\mathbf{r}-\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\|_{1/2}^2$	$\|\mathbf{h}\|_2 \le 1, \|\mathbf{t}\|_2 \le 1, \|\mathbf{r}\|_2 \le 1$ $\|\mathbf{M}_r(\theta_r)\mathbf{h}\|_2 \le 1, \|\mathbf{M}_r(\theta_r)\mathbf{t}\|_2 \le 1$ $\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}\|_2 \le 1, \|\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\|_2 \le 1$
TransM [52]	$-\theta_r\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
ManifoldE [53]	$-(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2^2-\theta_r^2)^2$	$\|\mathbf{h}\|_2 \le 1, \|\mathbf{t}\|_2 \le 1, \|\mathbf{r}\|_2 \le 1$
TransF [54]	$(\mathbf{h}+\mathbf{r})^\top\mathbf{t}+(\mathbf{t}-\mathbf{r})^\top\mathbf{h}$	$\|\mathbf{h}\|_2 \le 1, \|\mathbf{t}\|_2 \le 1, \|\mathbf{r}\|_2 \le 1$
TransA [55]	$-(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)^\top\mathbf{M}_r(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)$	$\|\mathbf{h}\|_2 \le 1, \|\mathbf{t}\|_2 \le 1, \|\mathbf{r}\|_2 \le 1$ $\|\mathbf{M}_r\|_F \le 1, [\mathbf{M}_r]_{ij} = [\mathbf{M}_r]_{ji} \ge 0$

[8]

	WN18RR			FB15k-237			YAGO3-10		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
RotatE	.476	.428	.571	.338	.241	.533	.495	.402	.670
MuRP	.481	.440	.566	.335	.243	.518	-	-	-
HAKE	.497	.452	**.582**	.346	.250	.542	.546	.462	.694
TuckER	.470	.443	.526	.358	.266	.544	-	-	-
CP	.438	.414	.485	.333	.247	.508	.567	.494	.698
RESCAL	.455	.419	.493	.353	.264	.528	.566	.490	.701
ComplEx	.460	.428	.522	.346	.256	.525	.573	.500	.703
CP-DURA	.478	.441	.552	.367	.272	.555	.579	.506	.709
RESCAL-DURA	**.498**	**.455**	.577	.368	**.276**	.550	.579	.505	.712
RESCAL-FRO	.397	.363	.452	.323	.235	.501	.474	.392	.628

[9]

Even worse performance when equipped with FRO regularizer

Part4: Regularization $r(\,)$

Duality-induced regularizer (DURA[9], NeurIPS 2020)

- for an existing tensor factorization based model (primal),
- there is often another distance based model (dual) closely associated with it.

Tensor factorization based (TFB): $f_{TFB}(h_i, r_j, t_k) = Re(\bar{\boldsymbol{h}}_i \boldsymbol{R}_j \boldsymbol{t}_k) = Re(\langle \boldsymbol{h}_i \bar{\boldsymbol{R}}_j, \boldsymbol{t}_k \rangle)$

Distance based (DB): $f_{DB}(h_i, r_j, t_k) = -\|\boldsymbol{h}_i \bar{\boldsymbol{R}}_j - \boldsymbol{t}_k\|_2^2$

Notice that $f_{DB}(h_i, r_j, t_k) = 2Re(\boldsymbol{h}_i \bar{\boldsymbol{R}}_j \boldsymbol{t}_k) - \|\boldsymbol{h}_i \bar{\boldsymbol{R}}_j\|_2^2 - \|\boldsymbol{t}_k\|_2^2$

$$= 2f_{TFB} - \|\boldsymbol{h}_i \bar{\boldsymbol{R}}_j\|_2^2 - \|\boldsymbol{t}_k\|_2^2$$

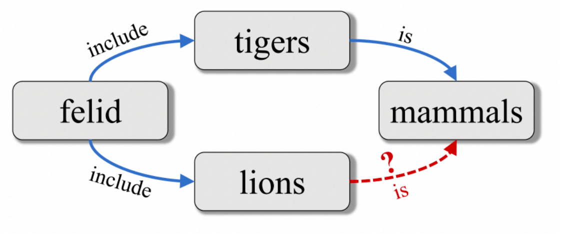
Such that $\mathbf{max}\, \boldsymbol{f_{DB}} = \mathbf{min} - \boldsymbol{f_{DB}} = \mathbf{min}(-2\boldsymbol{f_{TFB}} + \|\boldsymbol{h_i \bar{R}_j}\|_2^2 + \|\boldsymbol{t_k}\|_2^2)$

Derive the Basic DURA: $r_{B\_DURA} = \sum_{(h_i, r_j, t_k) \in S} (\|\boldsymbol{h}_i \bar{\boldsymbol{R}}_j\|_2^2 + \|\boldsymbol{t}_k\|_2^2)$

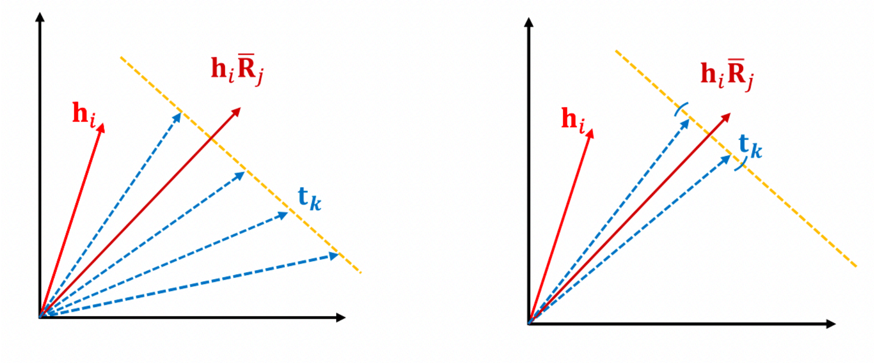
Part4: Regularization $r(\,)$

- Explanation of basic DURA
  - (felid, include, tigers)
  - (felid, include, lions)

→ representation of tigers and lions should be similar



- (tigers, is, mammals)
- to predict (lions, is, mammals)?



(b) Without regularization. (c) With DURA.

- Basic DURA → DURA
  - act on tails → heads and tails

$$f_{TFB}(h_i, r_j, t_k) = Re(\bar{\boldsymbol{h}}_i \boldsymbol{R}_j \boldsymbol{t}_k^T) \qquad f_{TFB}(h_i, r_j, t_k) = Re(\bar{\boldsymbol{t}}_k \boldsymbol{R}_j^T \boldsymbol{h}_i^T)$$

$$f_{DB}(h_i, r_j, t_k) = -\left\|\boldsymbol{h}_i \bar{\boldsymbol{R}}_j - \boldsymbol{t}_k\right\|_2^2 \qquad f_{DB}(h_i, r_j, t_k) = -\left\|\boldsymbol{t}_k \boldsymbol{R}_j^T - \boldsymbol{h}_i\right\|_2^2$$

$$r = \sum_{(h_i, r_j, t_k)\in S} \left(\left\|\boldsymbol{h}_i \bar{\boldsymbol{R}}_j\right\|_2^2 + \|\boldsymbol{t}_k\|_2^2\right) \qquad r = \sum_{(h_i, r_j, t_k)\in S} \left(\left\|\boldsymbol{t}_k \boldsymbol{R}_j^T\right\|_2^2 + \|\boldsymbol{h}_i\|_2^2\right)$$

Basic DURA:

$$r_{B\_DURA} = \sum_{(h_i, r_j, t_k)\in S} \left(\left\|\boldsymbol{h}_i \bar{\boldsymbol{R}}_j\right\|_2^2 + \|\boldsymbol{t}_k\|_2^2\right)$$

DURA:

$$r_{DURA} = \sum_{(h_i, r_j, t_k)\in S} \left(\left\|\boldsymbol{h}_i \bar{\boldsymbol{R}}_j\right\|_2^2 + \|\boldsymbol{t}_k\|_2^2 + \left\|\boldsymbol{t}_k \boldsymbol{R}_j^T\right\|_2^2 + \|\boldsymbol{h}_i\|_2^2\right)$$

Part4: Regularization $r(\,)$

- Practical usage (in a weighted form)

$$\min \sum_{(e_i,r_j,e_k)\in\mathcal{S}} [\ell_{ijk}(\mathbf{H},\mathbf{R}_1,\ldots,\mathbf{R}_J,\mathbf{T}) + \boxed{\lambda}(\lambda_1(\|\mathbf{h}_i\|_2^2+\|\mathbf{t}_k\|_2^2)+\lambda_2(\|\mathbf{h}_i\overline{\mathbf{R}}_j\|_2^2+\|\mathbf{t}_k\mathbf{R}_j^\top\|_2^2))],$$

- Smaller dataset scale, larger improvement

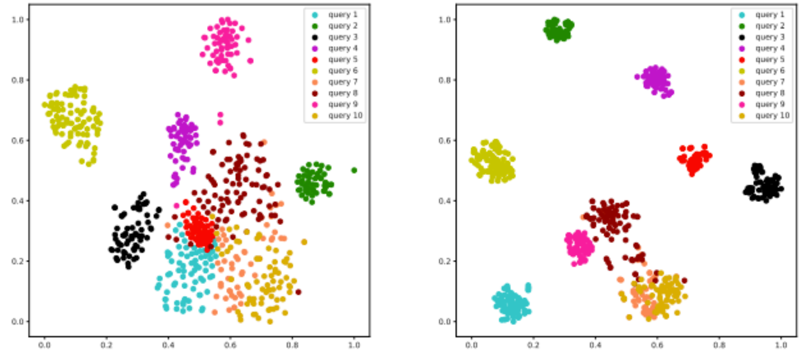
	WN18RR					FB15k-237					YAGO3-10				
	k	b	λ	λ_1	λ_2	k	b	λ	λ_1	λ_2	k	b	λ	λ_1	λ_2
CP	2000	100	1e-1	0.5	1.5	2000	100	5e-2	0.5	1.5	1000	1000	5e-3	0.5	1.5
ComplEx	2000	100	1e-1	0.5	1.5	2000	100	5e-2	0.5	1.5	1000	1000	5e-2	0.5	1.5
RESCAL	512	1024	1e-1	1.0	1.0	512	512	1e-1	2.0	1.5	512	1024	5e-2	1.0	1.0

- KG $\rightarrow$ TKG (ICLR 2020)[10]

$$\Omega^3(U,V,T;(i,j,k,l)) = \frac{1}{3}\left(\|u_i\|_3^3+\|u_k\|_3^3+\|v_k\odot t_l\|_3^3\right)$$

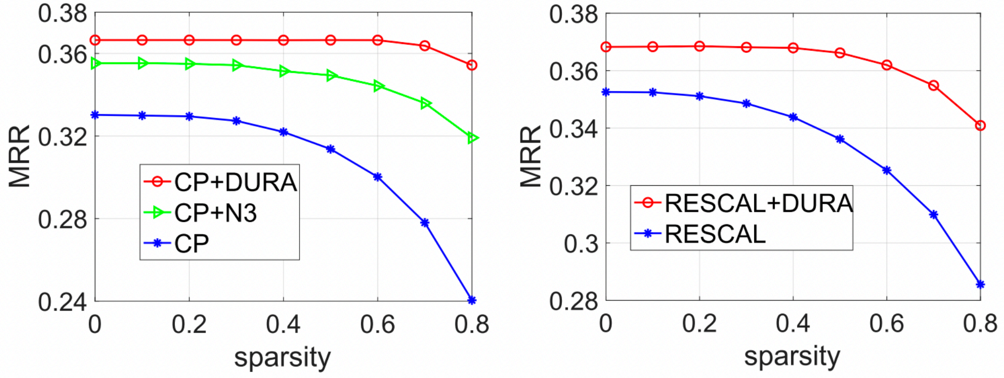
$$\Omega^3(U,V^t,V,T;(i,j,k,l)) = \frac{1}{3}\left(2\|u_i\|_3^3+2\|u_k\|_3^3+\|v_j^t\odot t_l\|_3^3+\|v_j\|_3^3\right)$$

T-SNE visualization:
the same query $\phi(\boldsymbol{h},\boldsymbol{r})$ are assigned
more similar representation



(a) CP (b) CP-DURA

Sparsity analysis:
DURA can reduce the storage usage



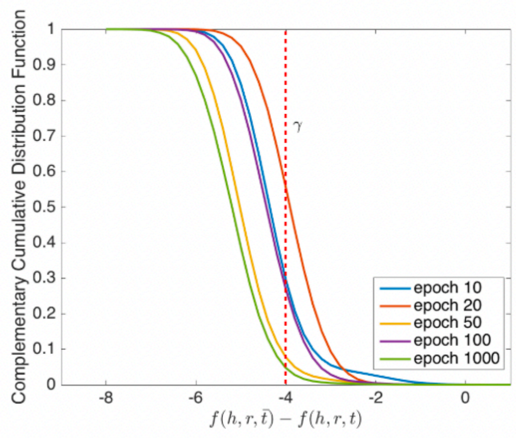
(a) CP (b) RESCAL

$$s_\lambda = \frac{\sum_{i=1}^{I}\sum_{d=1}^{D}\mathbb{1}_{\{|x|<\lambda\}}(\mathbf{E}_{id})}{I\times D}$$

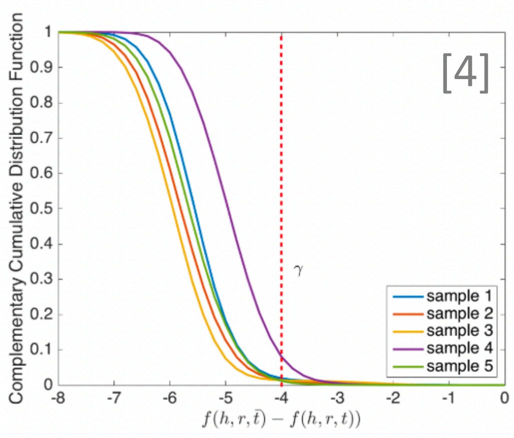
Part5: Optimization

Monitoring and control of convergence procedure

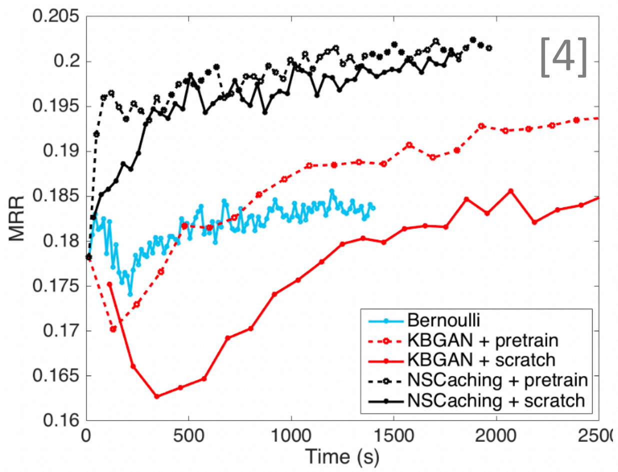
- interact with other 4 KGE components
- with plenty of hyper-parameters to tune
  - e.g., optimizer / initializer / learning rate / batch size

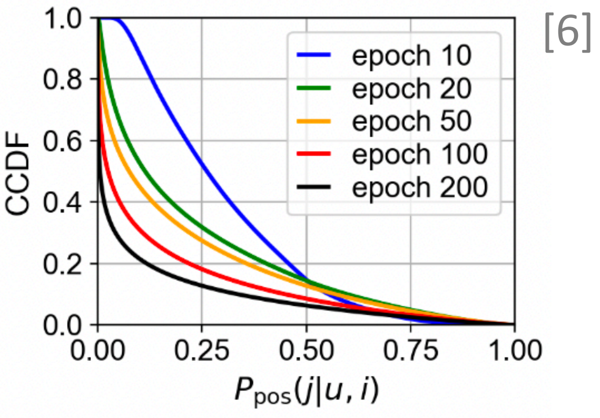


(a) Different epochs.



(b) Different triplets.





(a) Distribution of P_{pos}.

Review the Learning Objective

$$\underbrace{\min_{\boldsymbol{w}}}_{\text{optimization}} \overbrace{\mathcal{L}}^{\text{loss function}}\big(\underbrace{f(\cdot,\boldsymbol{w})}_{\text{scoring function}}, \mathcal{S}^+, \underbrace{\mathcal{S}^-}_{\text{negative sampling}}\big) + \underbrace{r(\boldsymbol{w})}_{\text{regularization}}$$

Five core components

- Scoring function $f(\,)$
- Negative sampling S^-
- Loss function $L(\,)$
- Regularization $r(\,)$
- Optimization

What can be conducted with AutoML? 🤔

Summary

Scoring function $f(\,)$
- simple bi-linear models reach SOTA performance
- complex models are not often promising but more likely to overfitting
- trend: pure KGE model → GNN-based / Path-based model

Negative sampling S^-
- tradeoff between efficiency and effectiveness
- false negative and hard samples play essential roles

Loss function $L(\,)$
- likelihood losses are empirically better than ranking losses
- lacking theoretical analysis and deep insights

Regularization $r(\,)$
- can be derived from associating scoring functions
- queries ($\phi(\boldsymbol{h},\boldsymbol{r})$ / $\phi(\boldsymbol{t},\boldsymbol{r})$) and targets ($\boldsymbol{t}/\boldsymbol{h}$) can be closer

Optimization
- closely interact with other components
- with plenty of hyper-parameters to tune

Outline

- Background
- Motivation
- Understanding of KGE components
- Searching experiments
    - Configuration Space of KGE
    - Searching on original KG
    - Searching on sampled KG
- Key takeaway

Configuration Space of KGE

Training procedure

step1: 3 HP
step2: 3 HP
step3: 1 HP
step4: 6 HP
step5: 2 HP

Input data: training triples S_{tra}

- step1: initialize learnable parameters w (embeddings / model weights)
- step2: sample negative triples $\tilde{S}_{(h,r,t)}$ (S^-) for each positive triple $(h, r, t) \in S_{tra}$ (S^+)
- step3: $f(\,)$ forward inference to obtain $Scores$ for triples in $\{(h, r, t)\} \cup \tilde{S}_{(h,r,t)}$
- step4: compute loss and regularization term w.r.t. $L(\,)$ and $r(\,)$
- step5: backward propagation, and update w & optimizer

Output: w

Any patterns in searching configuration?

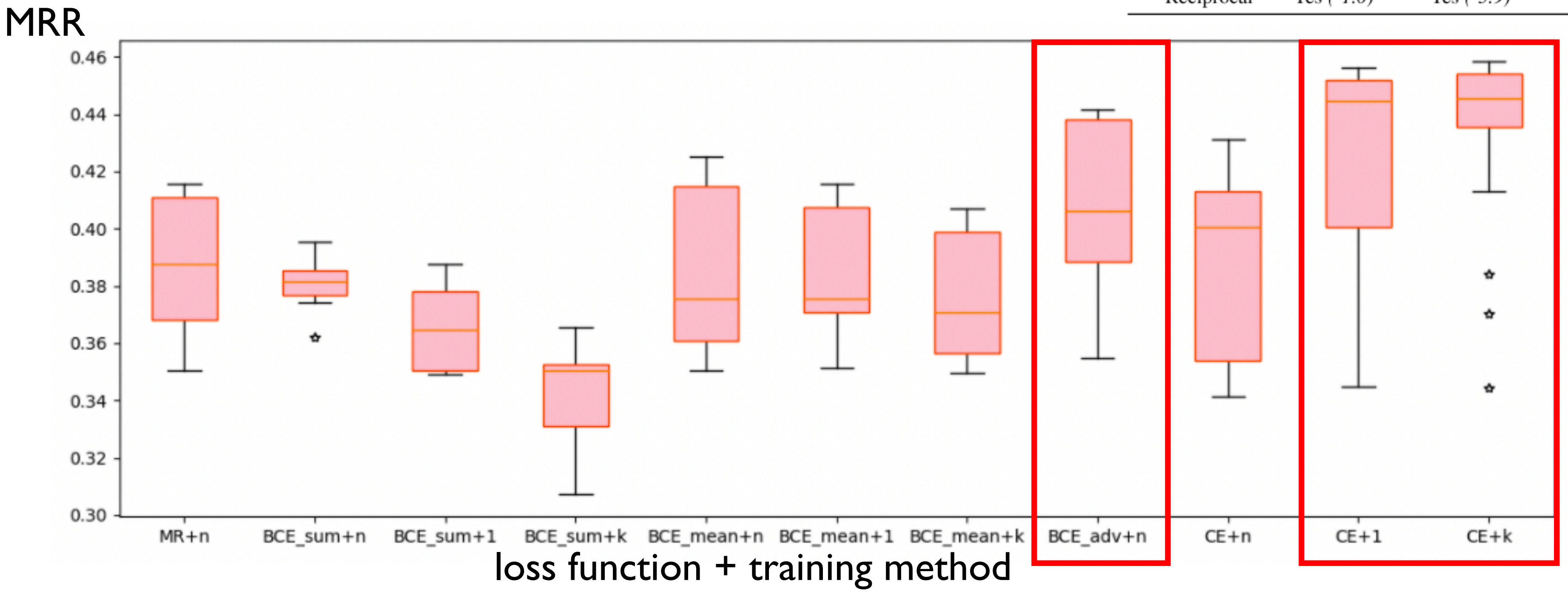
No.	Hyper-parameter	Range
1	L2 norm regularization	0, 10^{-8}, 10^{-6}, 10^{-4}
2	L3 norm regularization	0, 10^{-8}, 10^{-6}, 10^{-4}
3	Gamma	10, 50, 200
4	Embedding dimension	100, 200, 500, 1000, 2000
5	#negative samples	1, 8, 32, 128, 512, 2048
6	Self-adversarial rate	0.5, 1.0, 2.0
7	Training mode	negSamp, 1vs All, k vs All
8	Filtering false negative samples	True, False
9	Initialization mode	Uniform, xavier_norm
10	Loss function	MR, BCE, BCE_adv, CE
11	Learning rate	10^{-2}, 10^{-3}, 10^{-4}
12	Batch size	128, 512, 2048

Experiments: searching on original KG

- Experiment settings
  - Dataset: WN18RR
  - Model: ComplEx
  - Searching by {*loss function* + *training method*}
- Observations
  - CE + 1/ CE + k are generally better
  - BCE_adv performs best with negative sampling

Old dog new tricks [ICLR 2020][2]

		RESCAL	TransE	DistMult	ComplEx	ConvE
FB15K-237	Valid. MRR	*36.1*	*31.5*	*35.0*	*35.3*	*34.3*
	Emb. size	128 (*-0.5*)	512 (*-3.7*)	256 (*-0.2*)	256 (*-0.3*)	256 (*-0.4*)
	Batch size	512 (*-0.5*)	128 (*-7.1*)	1024 (*-0.2*)	1024 (*-0.3*)	1024 (*-0.4*)
	Train type	1vsAll (*-0.8*)	NegSamp –	NegSamp (*-0.2*)	NegSamp (*-0.3*)	1vsAll (*-0.4*)
	Loss	CE (*-0.9*)	CE (*-7.1*)	CE (*-3.1*)	CE (*-3.8*)	CE (*-0.4*)
	Optimizer	Adam (*-0.5*)	Adagrad (*-3.7*)	Adagrad (*-0.2*)	Adagrad (*-0.5*)	Adagrad (*-1.5*)
	Initializer	Normal (*-0.8*)	XvNorm (*-3.7*)	Unif. (*-0.2*)	Unif. (*-0.5*)	XvNorm (*-0.4*)
	Regularizer	None (*-0.5*)	L2 (*-3.7*)	L3 (*-0.2*)	L3 (*-0.3*)	L3 (*-0.4*)
	Reciprocal	No (*-0.5*)	Yes (*-9.5*)	Yes (*-0.3*)	Yes (*-0.3*)	Yes –
WNRR	Valid. MRR	*46.8*	*22.6*	*45.4*	*47.6*	*44.3*
	Emb. size	128 (*-1.0*)	512 (*-5.1*)	512 (*-1.1*)	128 (*-1.0*)	512 (*-1.2*)
	Batch size	128 (*-1.0*)	128 (*-5.1*)	1024 (*-1.1*)	512 (*-1.0*)	1024 (*-1.3*)
	Train type	KvsAll (*-1.0*)	NegSamp –	KvsAll (*-1.1*)	1vsAll (*-1.0*)	KvsAll (*-1.2*)
	Loss	CE (*-2.0*)	CE (*-5.1*)	CE (*-2.4*)	CE (*-3.5*)	CE (*-1.4*)
	Optimizer	Adam (*-1.2*)	Adagrad (*-5.8*)	Adagrad (*-1.5*)	Adagrad (*-1.5*)	Adam (*-1.4*)
	Initializer	Unif. (*-1.0*)	XvNorm (*-5.1*)	Unif. (*-1.3*)	Unif. (*-1.5*)	XvNorm (*-1.4*)
	Regularizer	L3 (*-1.2*)	L2 (*-5.1*)	L3 (*-1.1*)	L2 (*-1.0*)	L1 (*-1.2*)
	Reciprocal	Yes (*-1.0*)	Yes (*-5.9*)	Yes (*-1.1*)	No (*-1.0*)	Yes –

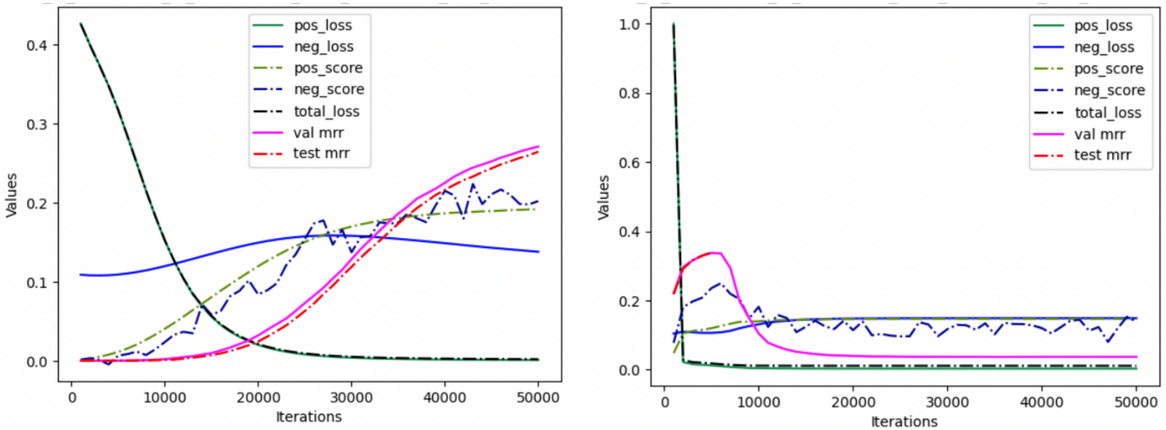


training methods
- n: negative sampling
- 1: 1 vs all
- k: k vs all

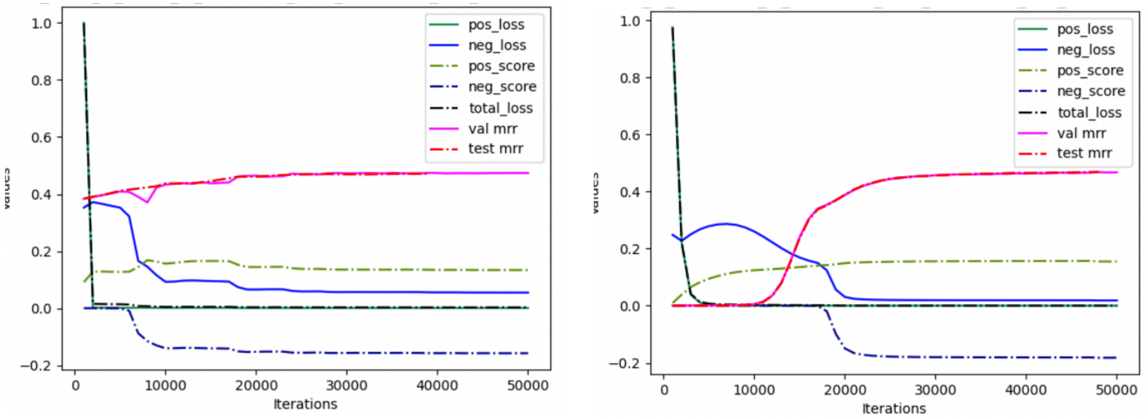
Experiments: searching on original KG

- Visualization of training process
  - No obvious patterns found

Configurations with poor performances:



Configurations with good performances

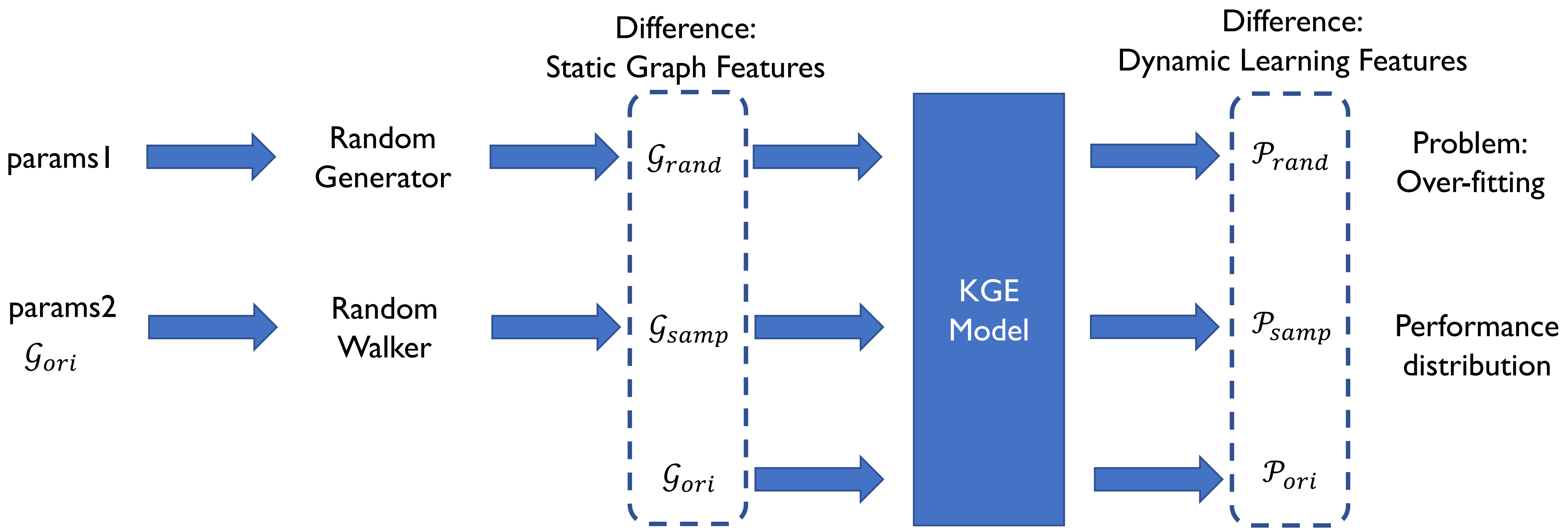


Model	dataset	original MRR	KGbench MRR	Promotion
ComplEx	WN18RR	0.440	0.474	+0.034
ComplEx	FB15K237	0.247	0.339	+0.092
DistMult	WN18RR	0.430	0.444	+0.014
DistMult	FB15K237	0.241	0.337	+0.096
RESCAL	WN18RR	0.420	0.462	+0.042
RESCAL	FB15K237	0.270	0.338	+0.068

Pipeline for KG sampling analysis

Searching on original KG is too time-consuming

- How can boost the searching speed?
- What about searching on sampled KGs?



Experiments: searching via KG sampling

Data statistics

- smaller subgraph $\longrightarrow$ denser
- obtain multi-scale KGs via sampling

$$\text{sparsity} = \frac{\#\text{triple}}{(\#\text{entity} * \#\text{entity} * \#\text{relation})}$$

dataset	wn18rr	wn18rr	wn18rr	wn18rr
sample ratio	0.01	0.1	0.3	Full
min sparsity	2.22E-04	2.22E-05	7.40E-06	
max sparsity	4.78E-04	4.08E-05	1.60E-05	5.00E-06
dataset	FB15k-237	FB15k-237	FB15k-237	FB15k-237
sample ratio	0.01	0.1	0.3	Full
min sparsity	2.90E-05	2.90E-06	9.67E-07	
max sparsity	3.81E-04	5.11E-05	2.24E-05	6.20E-06

dataset	FB15k-237	FB15k-237	FB15k-237	FB15k-237
sample ratio	0.2	0.5	0.8	Full
#entity	2908	7270	11632	14541
#relation	236	237	237	237
#triples	57.9k	182.4k	283.0k	310.1k
sparsity	5.81e-05	2.91e-05	1.77e-05	1.24e-05
validate depth distribution (1-7)	28.9 - 51.5 - 19.4 - 0.0 - 0.0 - 0.0 - 0.03	31.1 - 52.8 - 15.9 - 0.03 - 0.0 - 0.0 - 0.01	31.9 - 52.7 - 15.2 - 0.02 - 0.0 - 0.0 - 0.00	0.51 - 73.2 - 26.0 - 0.09 - 0.00 - 0.0 - 0.05
mean validate depth	1.90	1.84	1.83	2.26
test depth distribution (1-7)	27.3 - 51.3 - 21.2 - 0.0 - 0.0 - 0.03 - 0.03	31.1 - 52.5 - 16.1 - 0.03 - 0.0 - 0.0 - 0.02	32.0 - 52.5 - 15.3 - 0.02 - 0.0 - 0.0 - 0.00	0.44 - 73.4 - 25.8 - 0.17 - 0.00 - 0.0 - 0.13
mean test depth	1.94	1.85	1.83	2.26
mean inDegree	19.9	25.0	24.3	21.3
inDegree distribution (0/1/2/3/3+)	5.74 - 5.15 - 5.57 - 5.43 - 78.0	4.64 - 3.45 - 3.64 - 4.41 - 83.8	4.61 - 4.10 - 4.01 - 4.74 - 82.5	7.13 - 5.59 - 5.22 - 5.79 - 76.2
mean outDegree	19.9	25.0	24.3	21.3
outDegree distribution (0/1/2/3/3+)	3.85 - 2.37 - 1.92 - 2.33 - 89.5	3.86 - 2.11 - 1.40 - 1.63 - 90.9	2.77 - 2.32 - 1.97 - 2.07 - 90.8	4.47 - 4.59 - 3.23 - 3.12 - 84.5

Outline

- Background
- Motivation
- Understanding of KGE components
- Searching experiments
  - Searching on original KG
  - Searching on sampled KG
    - Correlation across sampling ratios (scales)
    - Correlation across computing budgets
    - Efficiency analysis
    - Broader correlation
- Key takeaway

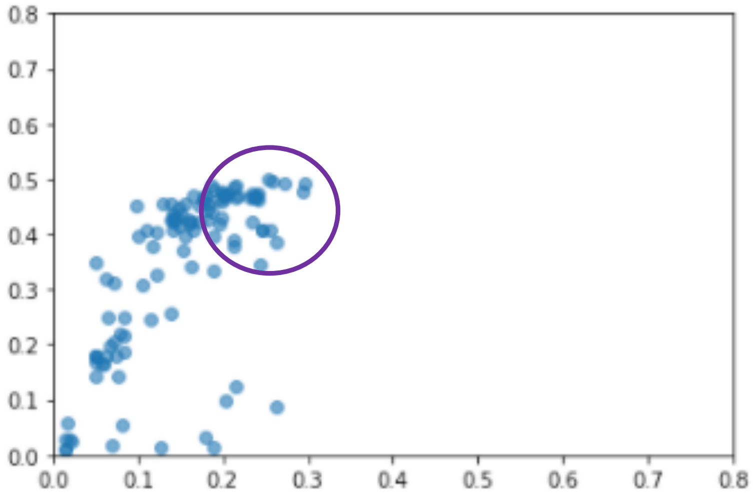
Experiments: searching via KG sampling

- Correlation across sampling ratios (scales)
  - 0.01 → sample ratio = 0.01 (of keeping nodes)

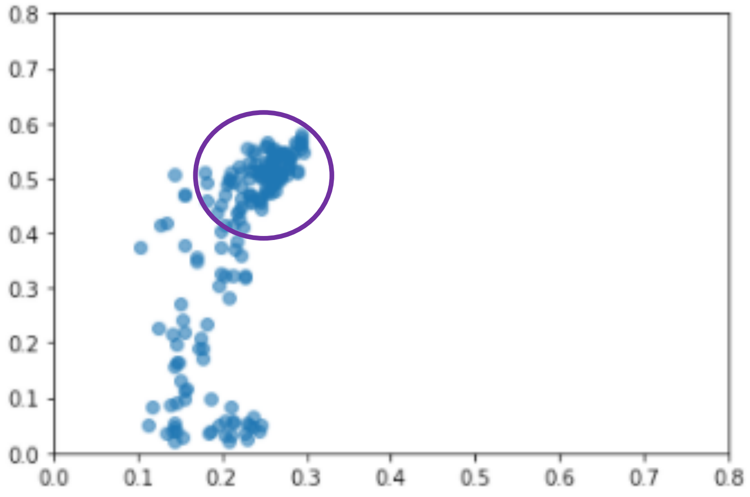
No.	X axis	Y axis	Spearman	Pearson
1	NELL-995	NELL-995 0.01	0.6738	0.6680
2	FB15k-237	FB15k-237 0.01	0.7674	0.6624

Observation:
- Acceptable correlation between original KG and sampled KG
- Well-performed configurations can be selected

x: NELL-995 (20w iterations)
y: NELL-995 0.01 (1w iterations)



x: FB15k-237
y: FB15k-237 0.01



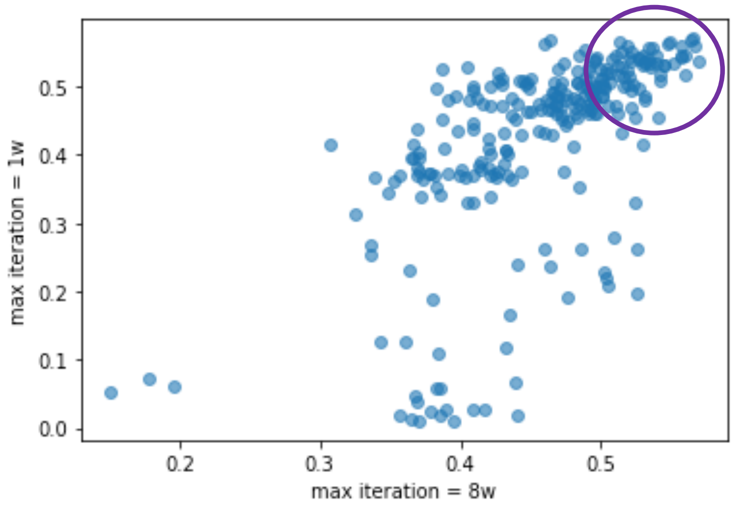
Experiments: searching via KG sampling

- Correlation across computing budgets
  - Dataset: WN18RR 0.01
  - Searching by max #iterations: 8w / 1w / 5k / 2k

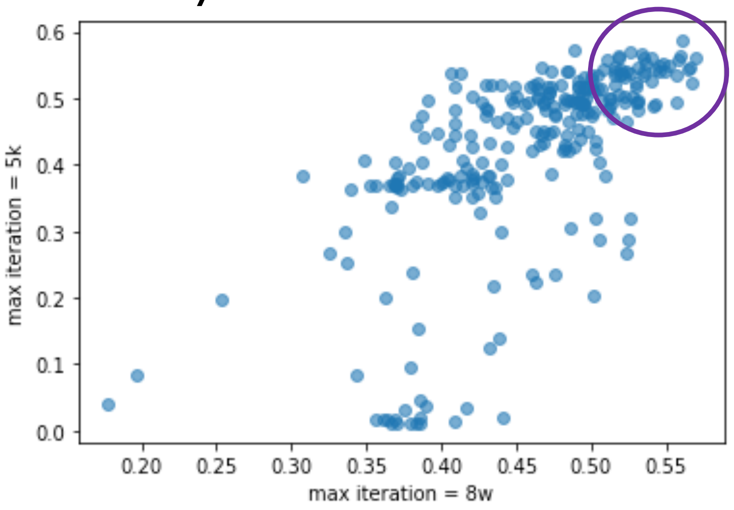
No.	X #iteration	Y #iteration	Spearman	Pearson
1	8w	1w	0.6863	0.6140
2	8w	5k	0.7076	0.6467
3	8w	2k	0.5846	0.5714

Observation:
- Less training time, weaker correlation

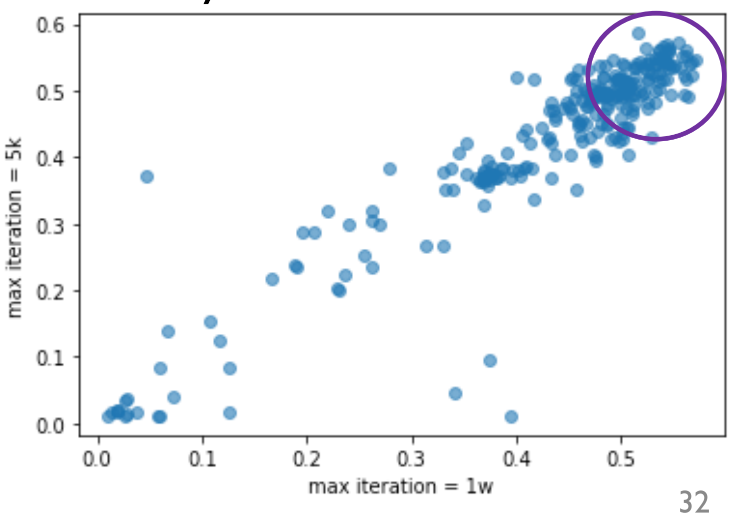
x: MRR of 8w iters
y: MRR of 1w iters



x: MRR of 8w iters
y: MRR of 5k iters



x: MRR of 8w iters
y: MRR of 2k iters



Experiments: searching via KG sampling

- Efficiency Analysis
  - Full KG: $iter_{full} = \#C \times iter_{max1}$
  - Sampled KG: $iter_{sample} = \#C \times iter_{max2} + K \times iter_{max1}$
  - Two-stage speed-up ratio: $R = \frac{iter_{full}}{iter_{sample}}$

Observation:
- 6-10X acceleration for the whole two-stage pipeline

- First stage: comparison of convergence speed

Mean iterations	Original KG	Sampled KG	$\text{Ratio}_{\text{stage1}}$
NELL-995 v.s. sample 0.01	85.8k	5.1k	16.6 X
FB15k-237 v.s. sample 0.01	85.8k	7.4k	11.5 X
FB15k-237 v.s. sample 0.05	76.5k	7.1k	10.7 X

To fully cover top-k configurations of original KG

FB15k-237	FB15k-237 0.01
Top5	Top20
Top10	Top70
Top20	Top70
Top30	Top100
Top50	Top100

Experiments: searching via KG sampling

- Correlation across models
  - with the same configuration

No.	X axis	Y axis	Spearman	Pearson
1	ComplEx	RotatE	0.2096	0.5842
2	ComplEx	DistMult	0.7097	0.6818
3	RotatE	DistMult	0.3153	0.5343

- Correlation across datasets
  - for certain model with the same configuration

No.	X axis	Y axis	Spearman	Pearson
1	WN18RR 0.01	NELL-995 0.01	0.7597	0.7716
2	WN18RR 0.01	FB15k-237 0.01	0.7106	0.7726
3	NELL-995 0.01	FB15k-237 0.01	0.8022	0.9297

Observation:

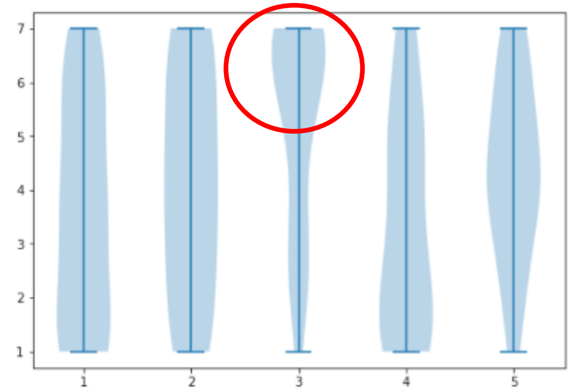
- Stronger correlation between models of the same type
- Good correlation across sampled KGs

Experiments: searching via KG sampling

No.	Hyper-parameter
1	L2 norm regularization
2	L3 norm regularization
3	Gamma
4	Embedding dimension
5	#negative samples
6	Self-adversarial rate
10	Training mode
11	Filtering false negative samples
12	Initialization mode
13	Loss function
14	Learning rate
15	Batch size

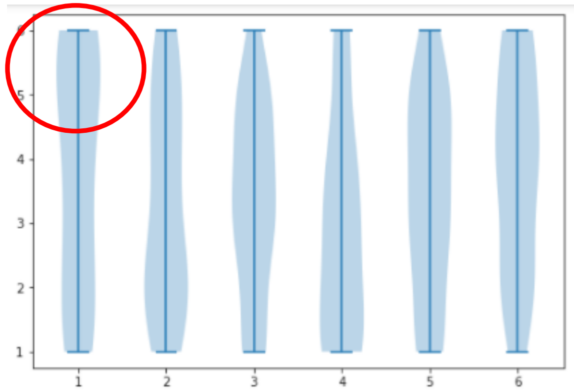
loss function

- MR, BCE_mean, BCE_sum, CE, BCE_adv



#negative samples

- 1, 8, 32, 128, 512, 2048

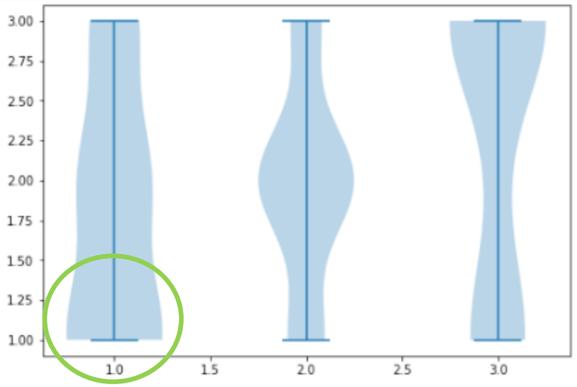


✗ lower rankings

✓ higher rankings

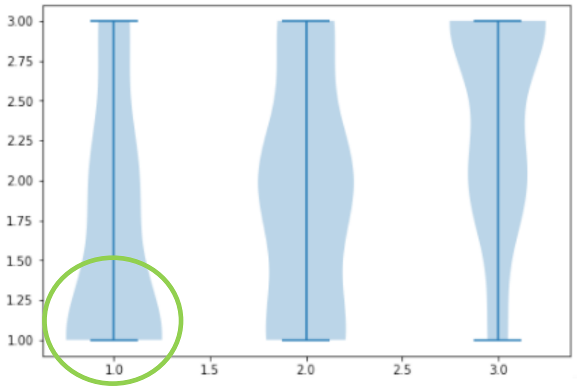
Learning rate

- 10^{-4}, 10^{-3}, 10^{-2},



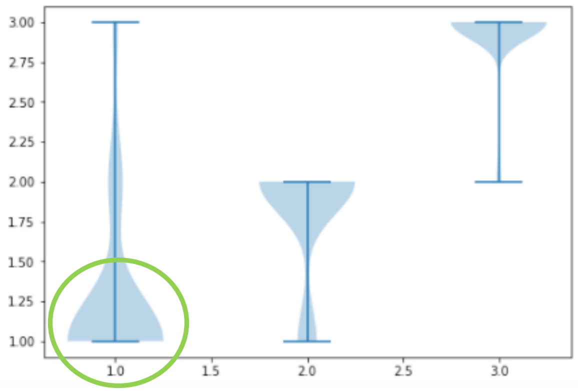
Batchsize

- 128, 512, 2048



Gamma

- 10, 50, 200



Experiments

Summary

- directly search on full data is quite slow
- good correlation across scale/model/dataset
- two-step searching might be more practical
  - sample subgraph and proceed searching
  - transfer to full data and finetune

TODO experiments

- Importance/sensitivity estimation
- General model search
- Transfer to original KG
- Transfer to other datasets
- Transfer to other KGE tasks

Potential two-step configuration searching for knowledge graph embedding

Inputs: KG $\mathcal{G}$, KGE model M

- step1: sample configurations Θ and train on $\mathcal{G}_{samp}$, $\mathcal{M}_{\Theta} \leftarrow \{\mathcal{M}(\theta), \ \forall\theta \in \Theta\}$
- step2: get top-k1 configurations Θ_{k1} w.r.t. $\mathcal{M}_{\Theta}$
- step3: compute dataset similarity by comparing $\mathcal{M}_{\Theta}$ and $\mathcal{M}'_{\Theta}$, and recommend configurations Θ_{k2}
- step4: finetune $\Theta' \leftarrow \Theta_{k1} \cup \Theta_{k2}$ on $\mathcal{G}$, get optimal $\theta^* \leftarrow \operatorname{argmax} \mathcal{M}_{\Theta'}$

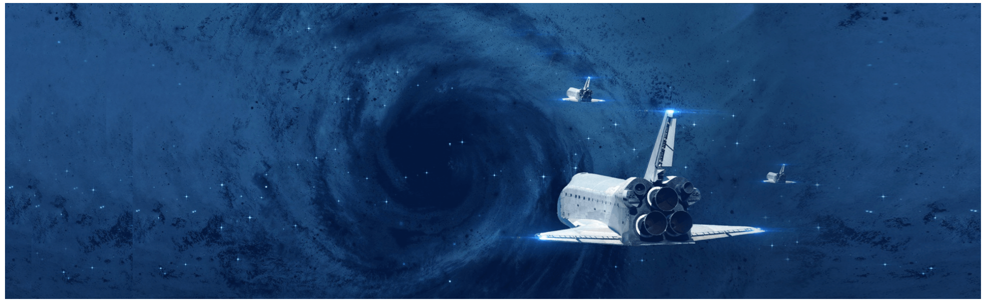
Output: θ^*

Outline

- Background
- Motivation
- Understanding of KGE components
- Searching experiments
- Key takeaway

Key takeaways

Recall the difficulties	KGbench
The choice of KGE model and configuration	Automated configuration search
A fair comparison of model and strategy	Benchmarking for fair comparison
Lacking understanding of KGE components	Study the principle and interaction of KGE components



TODO List

- Experiment-driven → comprehensive experiments
- Deep insights + theoretical analysis
- Summary and refine key novelty

Reference

[1] Rossi, Andrea, et al. "Knowledge graph embedding for link prediction: A comparative analysis." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15.2 (2021): 1-49.

[2] Ruffinelli, Daniel, Samuel Broscheit, and Rainer Gemulla. "You can teach an old dog new tricks! on training knowledge graph embeddings." *International Conference on Learning Representations*. 2019.

[3] Ji, Shaoxiong, et al. "A survey on knowledge graphs: Representation, acquisition, and applications." *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[4] Zhang, Yongqi, et al. "NSCaching: simple and efficient negative sampling for knowledge graph embedding." *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019.

[5] Sun, Zhiqing, et al. "A Re-evaluation of Knowledge Graph Completion Methods." *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020.

[6] Ding, Jingtao, et al. "Simplify and Robustify Negative Sampling for Implicit Collaborative Filtering." *arXiv preprint arXiv:2009.03376* (2020).

[7] Yang, Zhen, et al. "Understanding negative sampling in graph representation learning." *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020.

[8] Wang, Quan, et al. "Knowledge graph embedding: A survey of approaches and applications." *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017): 2724-2743.

[9] Zhang, Zhanqiu, Jianyu Cai, and Jie Wang. "Duality-Induced Regularizer for Tensor Factorization Based Knowledge Graph Completion." *Advances in Neural Information Processing Systems* 33 (2020).

[10] Lacroix, Timothée, Guillaume Obozinski, and Nicolas Usunier. "Tensor decompositions for temporal knowledge base completion." *arXiv preprint arXiv:2004.04926* (2020).

[11] Ali, Mehdi, et al. "Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework." *arXiv preprint arXiv:2006.13365* (2020).

Q&A

Thanks for your attention!

model	training approach	loss	scope	regularizer
TransE [2]	uniform negative sampling	MR	pair	L2
TransH [16]	bernoulli negative sampling	MR	pair	L2
ComplEx [14]	uniform negative sampling	BCE	point	L2
SimplE [6]	uniform negative sampling	BCE	point	L2
RotatE [13]	uniform negative sampling	self-adv BCE	set	-
QuatE [18]	uniform negative sampling	BCE	point	L2
DistMult [17]	uniform negative sampling	MR	pair	L2
HolE [10]	uniform negative sampling	BCE	point	L2
RESCAL [11]	-	MSE	point	L2
TuckER [1]	1 vs all	BCE	point	-
CP [7]	1 vs all	CE	set	L3
ConvE [4]	k vs all	BCE	point	L2 + dropout
ConvKB [9]	bernoulli negative sampling	BCE	point	L2