Knowledge graph embeddings
a neuro-symbolic perspective

antonio vergari (he/him)

@tetraduzione

13th Mar 2024 - **Advanced Probabilistic Modeling** - *University of Trento*

How to Turn Your Knowledge Graph Embeddings into Generative Models via Probabilistic Circuits

Lorenzo Loconte
University of Edinburgh, UK
l.loconte@sms.ed.ac.uk

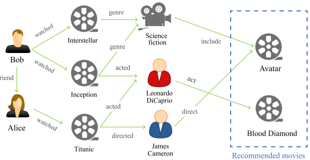
Nicola Di Mauro
University of Bari, Italy
nicola.dimauro@uniba.it

Robert Peharz
TU Graz, Austria
robert.peharz@tugraz.at

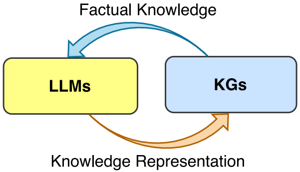
Antonio Vergari
University of Edinburgh, UK
avergari@ed.ac.uk

oral presentation at NeurIPS (top 0.6% papers)!

Knowledge graphs



Item recommendation



Augment LLMs

ChEMBL
HGNC
Ensembl
GOSTAR
Global Online Structure Activity Relationship Database
BioGrid
EBI Expression Atlas
DisGeNET
OMIM
Human Genetics Knowledge for the World

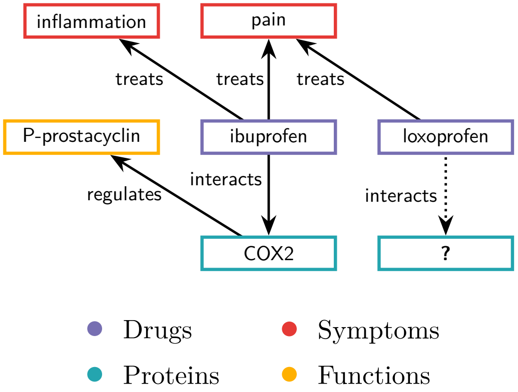
Drug discovery

Guo et al., "A Survey on Knowledge Graph-Based Recommender Systems", IEEE Transactions on Knowledge and Data Engineering, 2020

Pan et al., "Unifying Large Language Models and Knowledge Graphs: A Roadmap", ArXiv, 2023

Gogleva et al., "Knowledge Graph-based Recommendation Framework Identifies [...] Resistance in [...] Cell Lung Cancer", bioRxiv, 2021

Knowledge Graphs



$\langle$loxoprofen, treats, pain$\rangle$

$\langle$ibuprofen, treats, pain$\rangle$

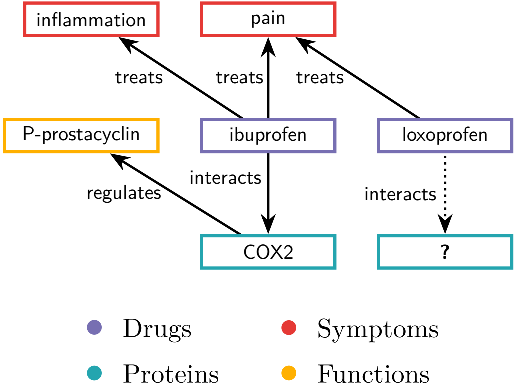
$\vdots$

$\langle$COX2, regulates, P-prostacyclin$\rangle$

$\langle$ibuprofen, interacts, COX2$\rangle$

$\mathcal{Q}$: $\langle$loxoprofen, interacts, ?$\rangle$

Knowledge Graphs



$\langle$loxoprofen, treats, pain$\rangle$

$\langle$ibuprofen, treats, pain$\rangle$

$\vdots$

$\langle$COX2, regulates, P-prostacyclin$\rangle$

$\langle$ibuprofen, interacts, COX2$\rangle$

$\mathcal{Q}$: $\langle$loxoprofen, interacts, **?**$\rangle$

KGE Models

SOTA ***knowledge graph embeddings*** (**KGE**) models

CP, RESCAL, TuckER, ComplEx

define a ***score function*** $\phi(s, r, o) \in \mathbb{R}$

E.g., for ComplEx:

$$\phi_{\text{ComplEx}}(s, r, o) = \Re(\langle \mathbf{e}_s, \mathbf{w}_r, \overline{\mathbf{e}_o} \rangle)$$

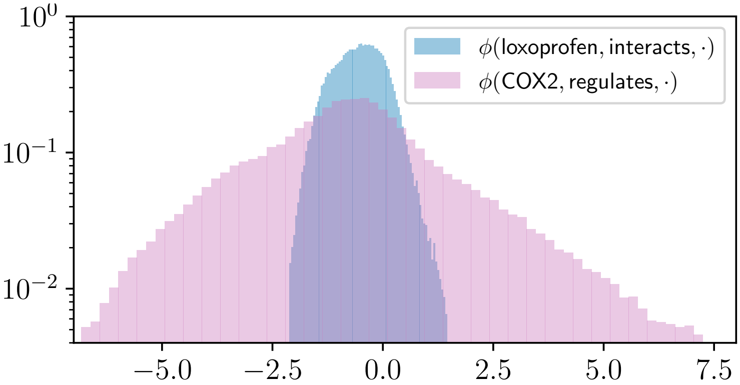
issues?

I

How to measure confidence of predictions?

and compare scores across models

Scores...



they are hard to *interpret and compare* instead!

Scores...

$$
\begin{aligned}
\phi(\text{loxprofen, interacts, phosp-acid}) &= 2.3 \\
\phi(\text{loxoprofen, interacts, COX2}) &= 1.3 \\
\ldots & \\
\phi(\text{paracetamol, treats, fever}) &= 42.1 \\
\phi(\text{paracetamol, treats, cancer}) &= -0.3
\end{aligned}
$$

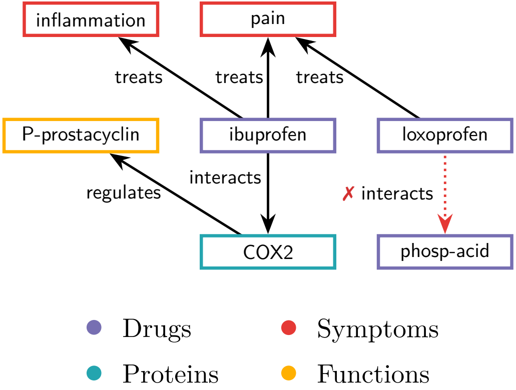
but we want *calibrated probabilities* instead!

issues?

I ***How to measure confidence of predictions?***
and compare scores across models

II ***How to guarantee satisfaction of constraints?***
and other background knowledge

e.g., ComplEx



$\boldsymbol{K}$: only drugs and proteins interact

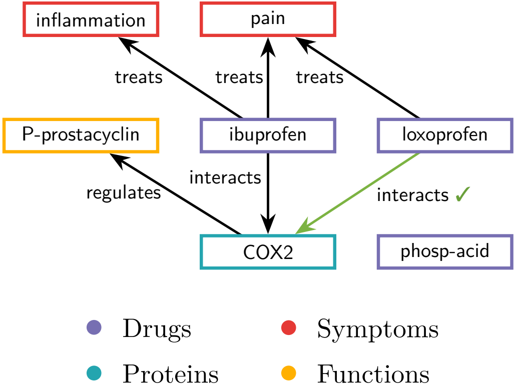
$\mathcal{A}$: $\langle$loxoprofen, interacts, **phosp-acid**$\rangle$

✘

$\mathcal{A}$: $\langle$loxoprofen, interacts, **COX2**$\rangle$

✔

e.g., ComplEx



$\boldsymbol{K}$: only drugs and proteins interact

$\mathcal{A}$: $\langle$loxoprofen, interacts, **phosp-acid**$\rangle$

✘

$\mathcal{A}$: $\langle$loxoprofen, interacts, **COX2**$\rangle$

✔

issues?

I **How to measure confidence of predictions?**
and compare scores across models

II **How to guarantee satisfaction of constraints?**
and other background knowledge

III **How to scale to KGs with millions of entities?**
by consuming less memory

solution!

I

***Generative models for KGs* (GeKCs)**

calibrated probabilistic predictions,   sampling of new triples

II

How to guarantee satisfaction of constraints?

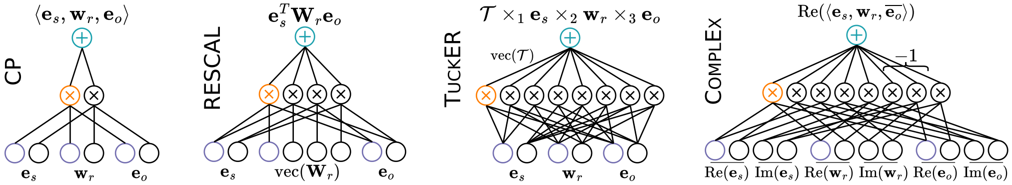
and other background knowledge

III

How to scale to KGs with millions of entities?

by consuming less memory

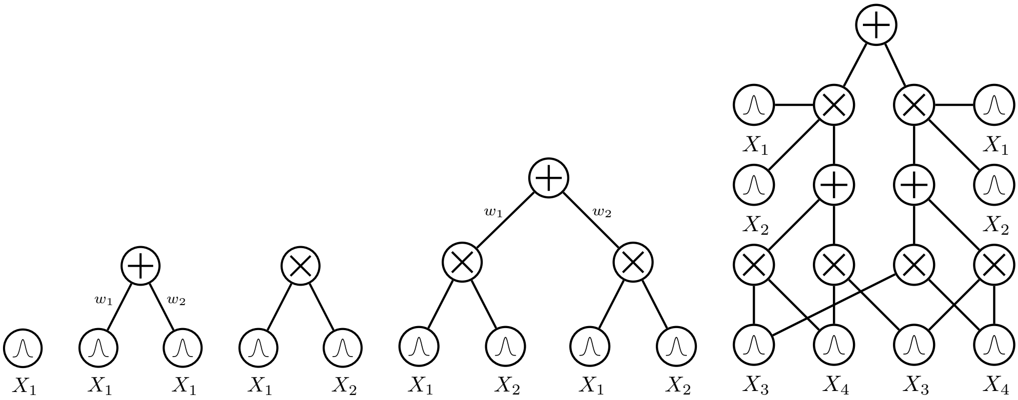
From KGE Models ...



ϕ_{CP} ϕ_{RESCAL} ϕ_{TuckER} ϕ_{ComplEx}

to probabilistic circuits (PCs)

A grammar for tractable computational graphs



...why PCs?

1. A grammar for tractable models

One formalism to represent many models. *#HMMs #Trees #XGBoost*, ...

2. Expressiveness

Competitive with intractable models, VAEs, Flow...*#hierachical #mixtures #polynomials*

From KGE Models ...

$$p(S, R, O) = \underbrace{\frac{1}{Z}}_{\text{Normalisation constant}} \cdot \exp \underbrace{\phi(S, R, O)}_{\text{Negated energy}}$$

$$Z = \sum_{s \in \mathcal{E},\, r \in \mathcal{R},\, o \in \mathcal{E}} \exp \phi(s, r, o)$$

Computing Z quickly becomes infeasible

---

bordes2013transe, **bordes2013transe**, **bordes2013transe**, **bordes2013transe**
minervini2016efficient, **minervini2016efficient**, **minervini2016efficient**, **minervini2016efficient**

From KGE Models ...

$$p(S, R, O) = \underbrace{\frac{1}{Z}}_{\text{Normalisation constant}} \cdot \exp \underbrace{\phi(S, R, O)}_{\text{Negated energy}}$$

$$Z = \sum_{s \in \mathcal{E},\, r \in \mathcal{R},\, o \in \mathcal{E}} \exp \phi(s, r, o)$$

Computing Z quickly becomes infeasible

---

bordes2013transe, bordes2013transe, bordes2013transe, bordes2013transe
minervini2016efficient, minervini2016efficient, minervini2016efficient, minervini2016efficient

From EBM Models ...

$$p(S,R,O) = \frac{1}{Z} \cdot \exp \phi(S,R,O)$$

solution remove $\exp$ and ensure $\phi(S,R,O) \geq 0$

From EBM Models ...

$$p(S, R, O) = \frac{1}{Z} \cdot \exp \phi(S, R, O)$$

solution remove $\exp$ and ensure $\phi(S, R, O) \geq 0$

... to probabilistic models

Choi, Vergari, and Van den Broeck, "Probabilistic Circuits: A Unifying Framework for Tractable Probabilistic Modeling", Technical Report, 2021

... to Probabilistic Circuits

in two ways

$$p(S, R, O) = \frac{1}{Z} \cdot \phi^{+}(S, R, O)$$

I) ***Non-negative restriction*** of computational units,

$\Longrightarrow$ *enforce embeddings to be non-negative*

CP$^+$ RESCAL$^+$ TuckER$^+$ ComplEx$^+$

... to Probabilistic Circuits

in two ways

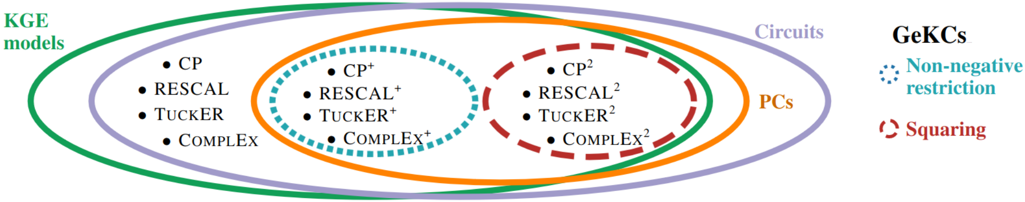
$$p(S, R, O) = \frac{1}{Z} \cdot \phi^{\mathbf{2}}(S, R, O)$$

II) ***Squaring*** the computational graph output

$\Longrightarrow$ *more expressive, no constraining parameters*

CP^2 RESCAL^2 TuckER^2 ComplEx^2

solution!



KGE models
Circuits
PCs
• CP
• RESCAL
• TuckER
• ComplEx
• CP⁺
• RESCAL⁺
• TuckER⁺
• ComplEx⁺
• CP²
• RESCAL²
• TuckER²
• ComplEx²
GeKCs
Non-negative restriction
Squaring

...why PCs?

1. A grammar for tractable models

One formalism to represent many models. *#HMMs #Trees #XGBoost*, ...

2. Expressiveness

Competitive with intractable models, VAEs, Flow...*#hierachical #mixtures #polynomials*

...why PCs?

1. A grammar for tractable models

One formalism to represent many models. *#HMMs #Trees #XGBoost*, ...

2. Expressiveness

Competitive with intractable models, VAEs, Flow...*#hierachical #mixtures #polynomials*

3. Tractability == Structural Properties!!!

Exact computations of reasoning tasks are certified by guaranteeing certain structural properties. *#marginals #expectations #MAP*, *#product* ...

Structural properties

smoothness

decomposability

determinism

compatibility

Vergari *et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

Structural properties

property A

property B

property C

property D

Vergari *et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

Structural properties

property A

property B

property C

property D

tractable computation of ***arbitrary integrals***

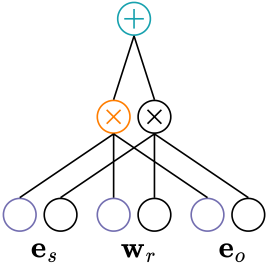
$$p(\mathbf{y}) = \int_{\mathsf{val}(\mathbf{Z})} p(\mathbf{z}, \mathbf{y})\, d\mathbf{Z}, \quad \forall \mathbf{Y} \subseteq \mathbf{X}, \quad \mathbf{Z} = \mathbf{X} \setminus \mathbf{Y}$$

$\Longrightarrow$ ***sufficient*** *and* ***necessary*** *conditions for a single feedforward evaluation*

$\Longrightarrow$ *tractable partition function*

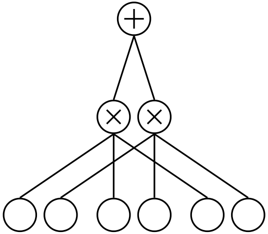
Vergari *et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

$$\phi_{\mathrm{CP}}(s, r, o) = \sum_{i=1}^{d} \mathbf{e}_{si} \mathbf{w}_{ri} \mathbf{e}_{oi}$$



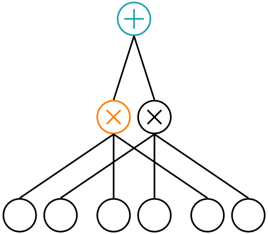
CP$^+$ score function

$$\sum_{s \in \mathcal{E},\, r \in \mathcal{R},\, o \in \mathcal{E}} \sum_{i=1}^{d} \mathbf{e}_{si} \mathbf{w}_{ri} \mathbf{e}_{oi}$$



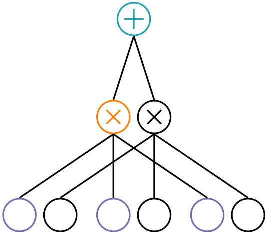
Summations over triples ...

$$\sum_{i=1}^{d} \left(\sum_{s \in \mathcal{E}} \mathbf{e}_{si}\right) \left(\sum_{r \in \mathcal{R}} \mathbf{w}_{ri}\right) \left(\sum_{o \in \mathcal{E}} \mathbf{e}_{oi}\right)$$



... can be broke down ...

$$\sum_{s\in\mathcal{E},\, r\in\mathcal{R},\, o\in\mathcal{E}} \phi_{\mathrm{CP}}(s, r, o)$$



$$\sum_{s\in\mathcal{E}} \mathbf{e}_{si} \qquad \sum_{r\in\mathcal{R}} \mathbf{w}_{ri} \qquad \sum_{o\in\mathcal{E}} \mathbf{e}_{oi}$$

... thus can be done linearly in the number of KG entities !

Learning ...

...by ***discriminative objectives*** unified as a ***weighted pseudo-loglikelihood***

$$\mathcal{L}_{\text{PLL}} := \sum_{(s,r,o)\in\mathcal{D}} w_s \log p(s \mid r, o) + w_r \log p(r \mid s, o) + w_o \log p(o \mid s, r)$$

...by ***generative objectives*** as ***maximum likelihood***

$$\mathcal{L}_{\text{MLE}} := \sum_{(s,r,o)\in\mathcal{D}} \log p(S = s, R = r, O = o)$$

Ruffinelli, Broscheit, and Gemulla, “You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings”, , 2020

Learning ...

...by ***discriminative objectives*** unified as a ***weighted pseudo-loglikelihood***

$$\mathcal{L}_{\text{PLL}} := \sum_{(s,r,o)\in\mathcal{D}} w_s \log p(s \mid r, o) + w_r \log p(r \mid s, o) + w_o \log p(o \mid s, r)$$

...by ***generative objectives*** as ***maximum likelihood***

$$\mathcal{L}_{\text{MLE}} := \sum_{(s,r,o)\in\mathcal{D}} \log p(S = s, R = r, O = o)$$

Ruffinelli, Broscheit, and Gemulla, "You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings", , 2020

Ok but … how do they perform ?!

Model	FB15k-237		WN18RR		ogbl-biokg	
	PLL	MLE	PLL	MLE	PLL	MLE
CP	0.310	—	**0.105**	—	0.831	—
CP^+	0.237	0.230	0.027	0.026	0.496	0.501
CP^2	**0.315**	0.282	**0.104**	0.091	**0.848**	0.829
ComplEx	**0.342**	—	**0.471**	—	0.829	—
ComplEx^+	0.214	0.205	0.030	0.029	0.503	0.516
ComplEx^2	0.334	0.300	0.420	0.391	**0.858**	0.840

Sampling Triples $(s, r, o) \sim p(S, R, O)$

A **kernel triple distance** to measure their quality

Model	FB15k-237		WN18RR		ogbl-biokg	
Uniform	0.589		0.766		1.822	
	PLL	MLE	PLL	MLE	PLL	MLE
ComplEx^{+}	0.336	0.323	0.456	0.478	0.175	0.097
ComplEx^{2}	0.326	**0.102**	0.338	**0.278**	0.104	**0.034**

solution!

I

***Generative models for KGs* (GeKCs)**

calibrated probabilistic predictions,   sampling of new triples

II

How to guarantee satisfaction of constraints?

and other background knowledge

III

How to scale to KGs with millions of entities?

by consuming less memory

solution!

I

***Generative models for KGs* (GeKCs)**
calibrated probabilistic predictions,   sampling of new triples

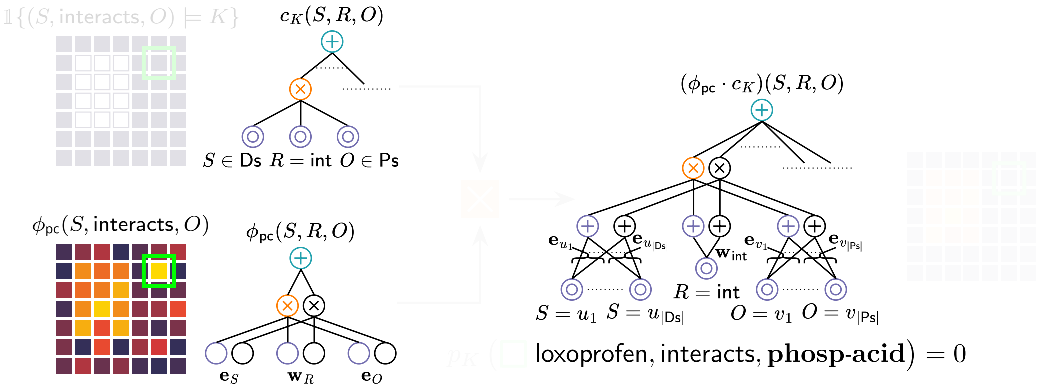
II

Provably reliable integration of constraints
e.g., domain constraints for relations

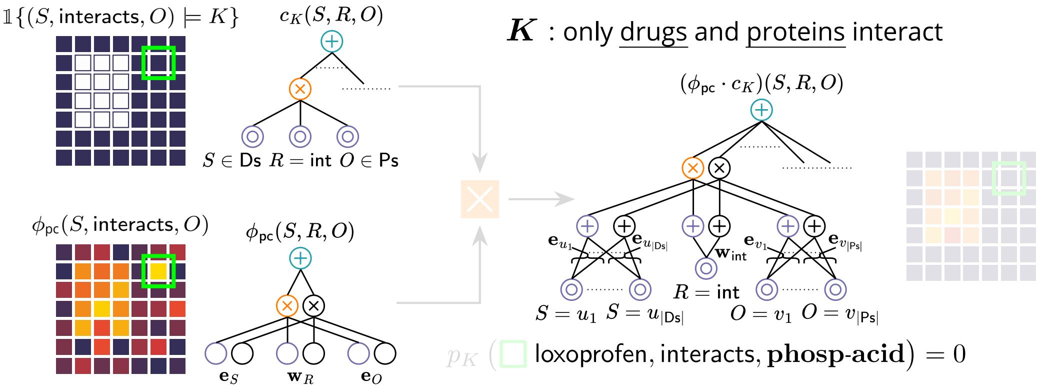
III

How to scale to KGs with millions of entities?
by consuming less memory

Guaranteed satisfaction of constraints

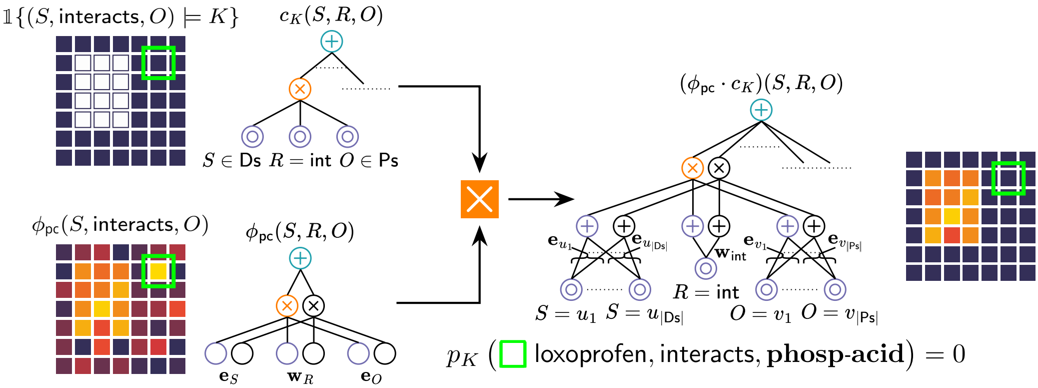


Guaranteed satisfaction of constraints

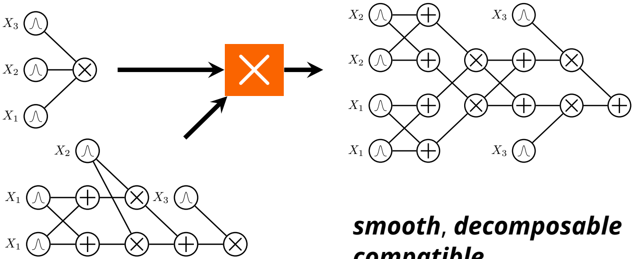


Ahmed et al., "Semantic probabilistic layers for neuro-symbolic learning", , 2022

Guaranteed satisfaction of constraints



Tractable products



smooth, ***decomposable***
compatible

exactly compute $\mathcal{Z}$ in time $O(|\boldsymbol{q}||\boldsymbol{c}|)$

Vergari *et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

solution!

I

Generative models for KGs **(GeKCs)**
calibrated probabilistic predictions,   sampling of new triples

II

Provably reliable integration of constraints
e.g., domain constraints for relations

III

How to scale to KGs with millions of entities?
by consuming less memory

solution!

I — ***Generative models for KGs* (GeKCs)**
calibrated probabilistic predictions, sampling of new triples

II — ***Provably reliable integration of constraints***
e.g., domain constraints for relations

III — ***Scaling to very large batch sizes***
greatly speed-up training

Scaling training

on KGs with millions of entities