HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Knowledge Graphs

Concept Learning is Reinforcement Learning

Axel Ngonga



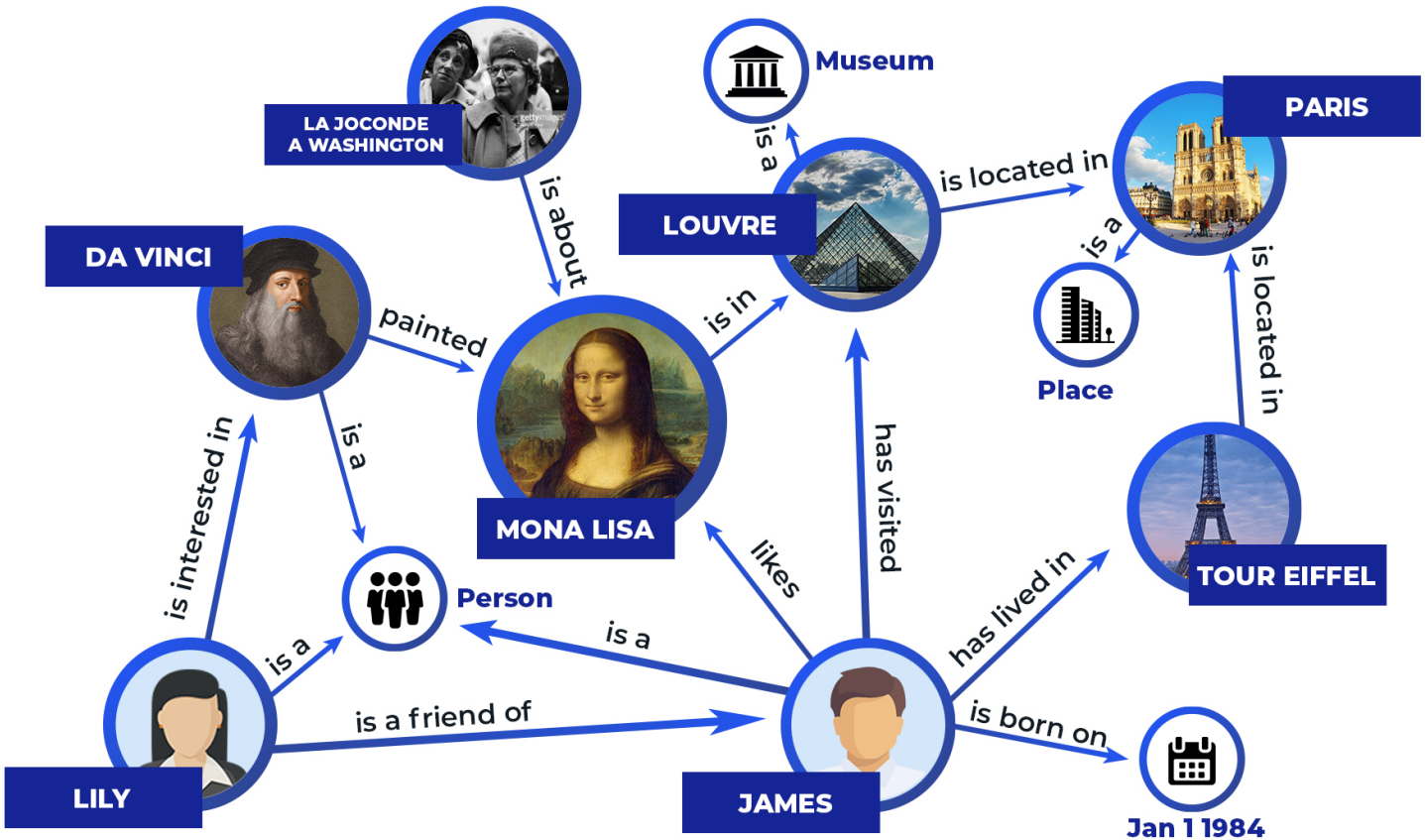
Core AI Thematic School
March 25th, 2026

Section 1

Basics

Basics

Example



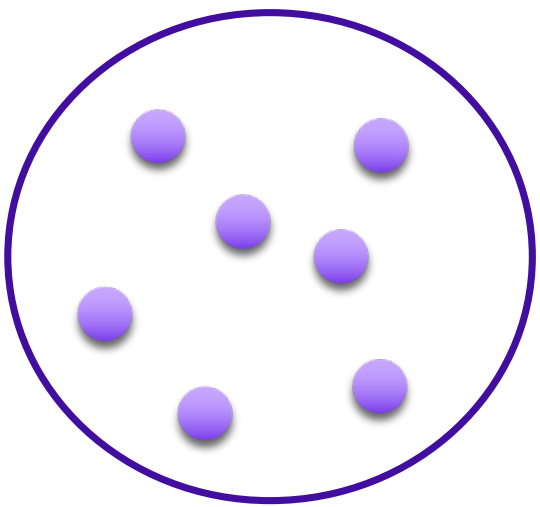
[1]

[1] Source: https://bit.ly/3sxCj6e

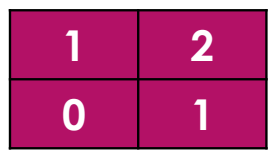
Basics

Why knowledge graphs?

- Graphs are a generic data model
- Can be used to model
  - Sets . . . are graphs without edges
  - Lists . . . are equivalent to paths
  - Matrices . . . are weighted graphs
  - Tensors . . . are weighted hypergraphs



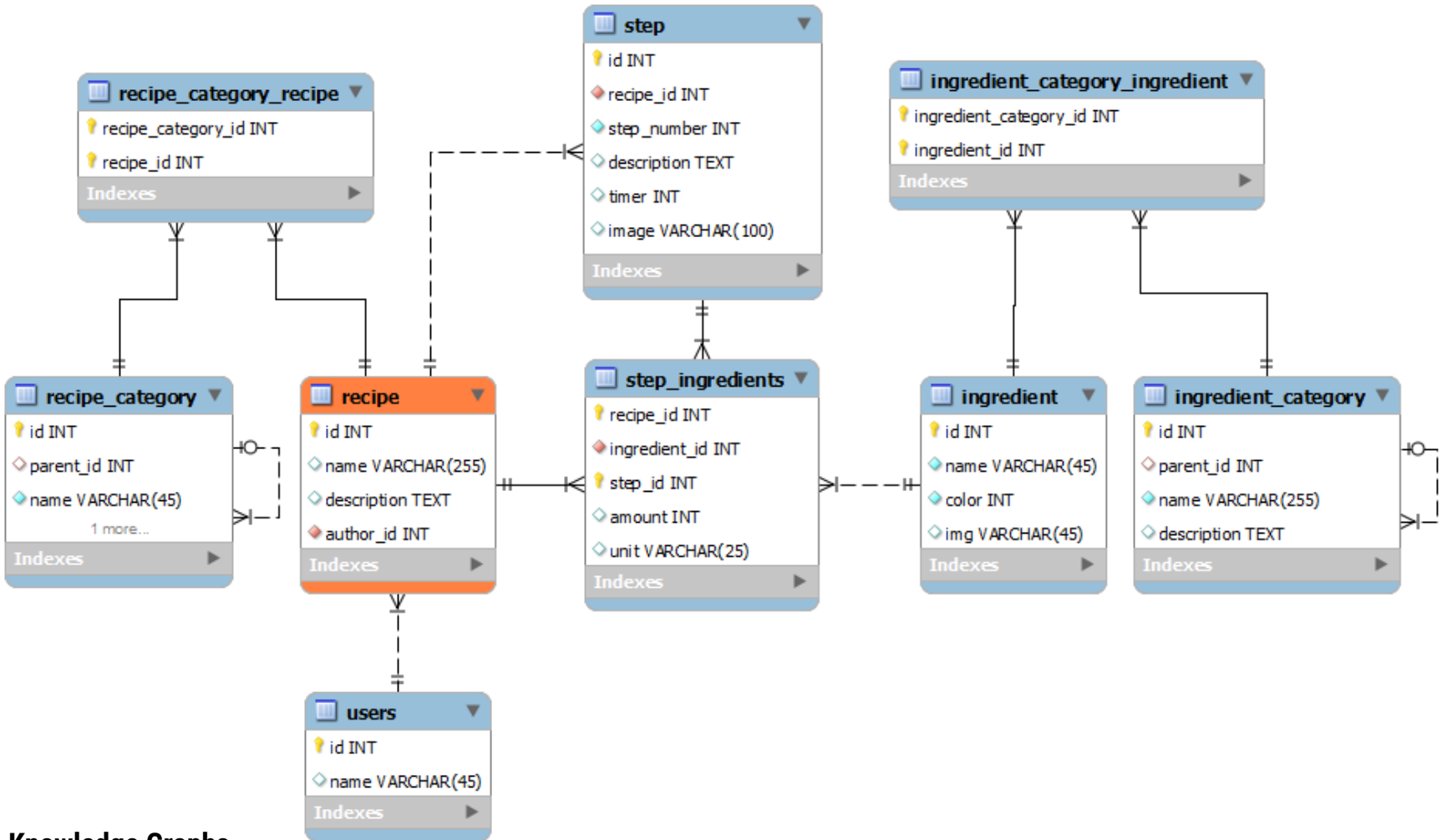




Basics

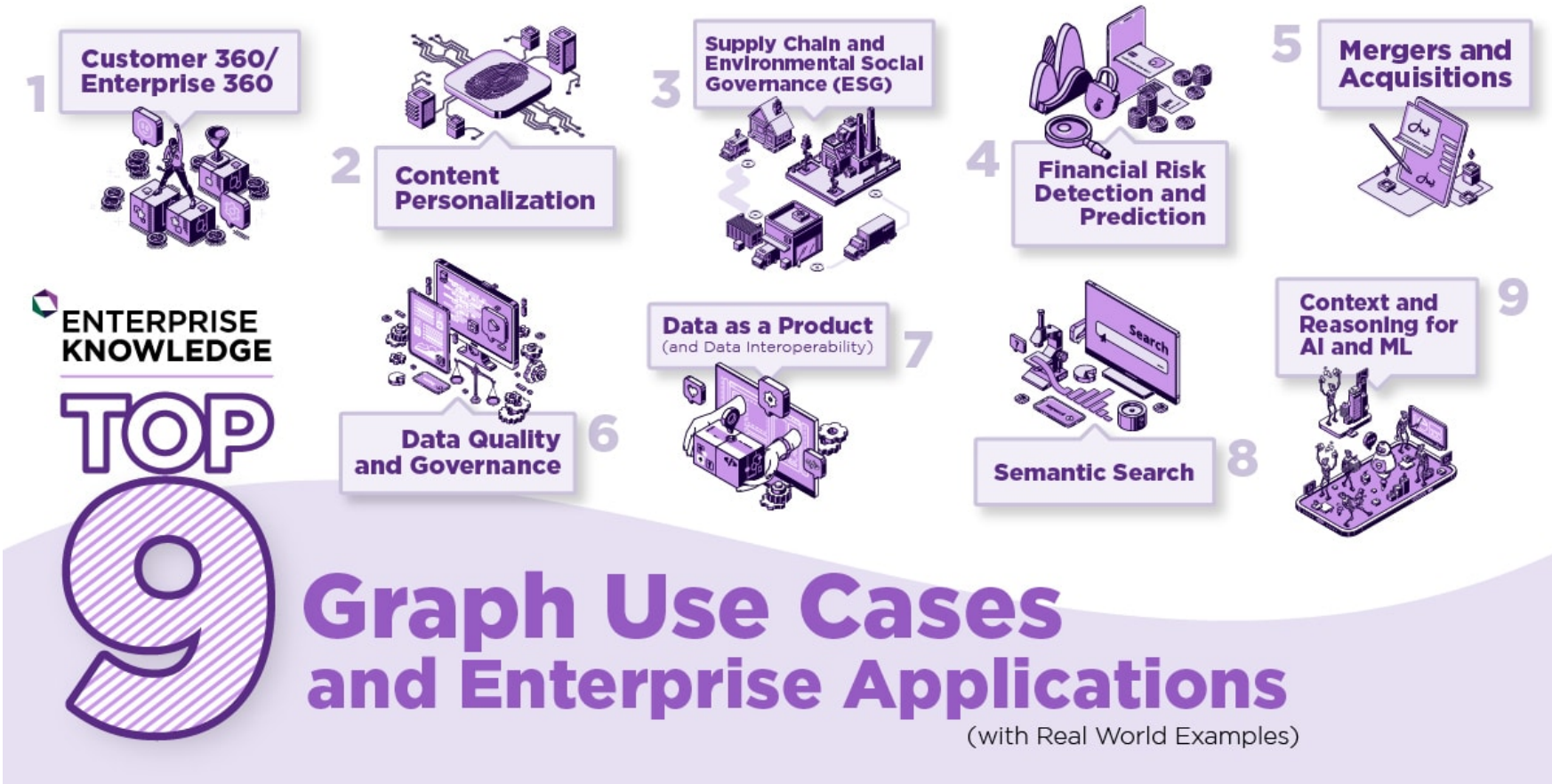
Why knowledge graphs?

- Generalize over relational database tables
- Can incorporate formal logics
- Allow for separate treatment of data and schema



Basics

Applications



- 250+ billion triples in open knowledge bases
- Regarded as key for XAI (see Data and AI Acts)
- Must hence be made amenable to machine learning

Section 2

Concept Learning

Concept Learning

Formal model

Definition

A knowledge graph is a set of triples $\mathcal{K} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where

- $\mathcal{E}$ is an infinite countable set of entities
- $\mathcal{R}$ is an infinite countable set of relations

Concept Learning

Formal model

Definition

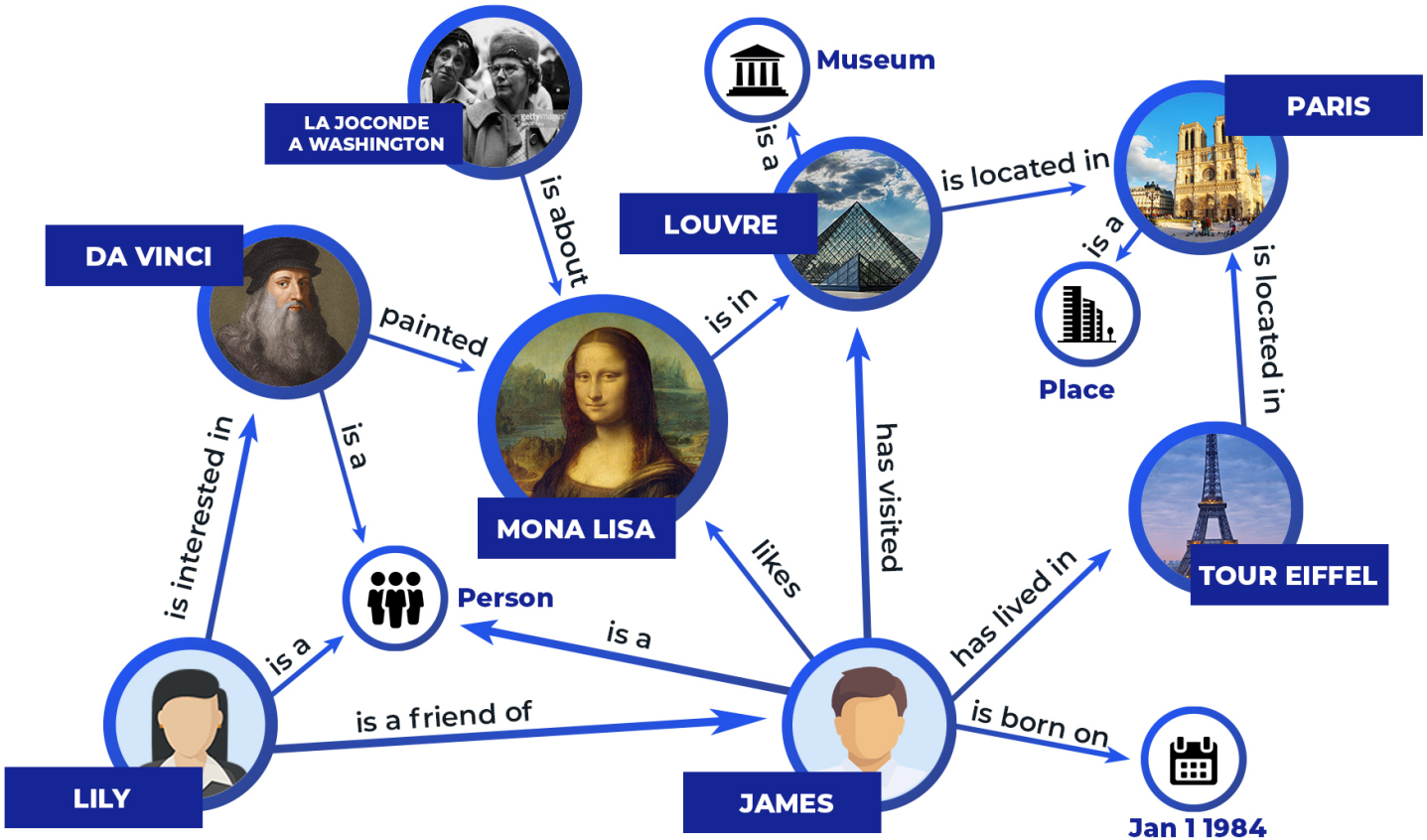
A knowledge graph is a set of triples $\mathcal{K} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where

- $\mathcal{E}$ is an infinite countable set of entities
- $\mathcal{R}$ is an infinite countable set of relations

- Often results from partial/full inference applied to a knowledge base $\mathcal{B} = \langle \mathcal{O}, \mathcal{D} \rangle$, where
  - $\mathcal{O}$ is a set of subsumption axioms $C \sqsubseteq D$ that describes the domain terminology in a description logic, e.g., $\mathcal{ALC}$
  - $\mathcal{D}$ is a set of assertions $r(a, b)$ or $C(a)$
- Here, we use $\mathcal{K} = \langle \mathcal{O}, \mathcal{D} \rangle$

Concept Learning

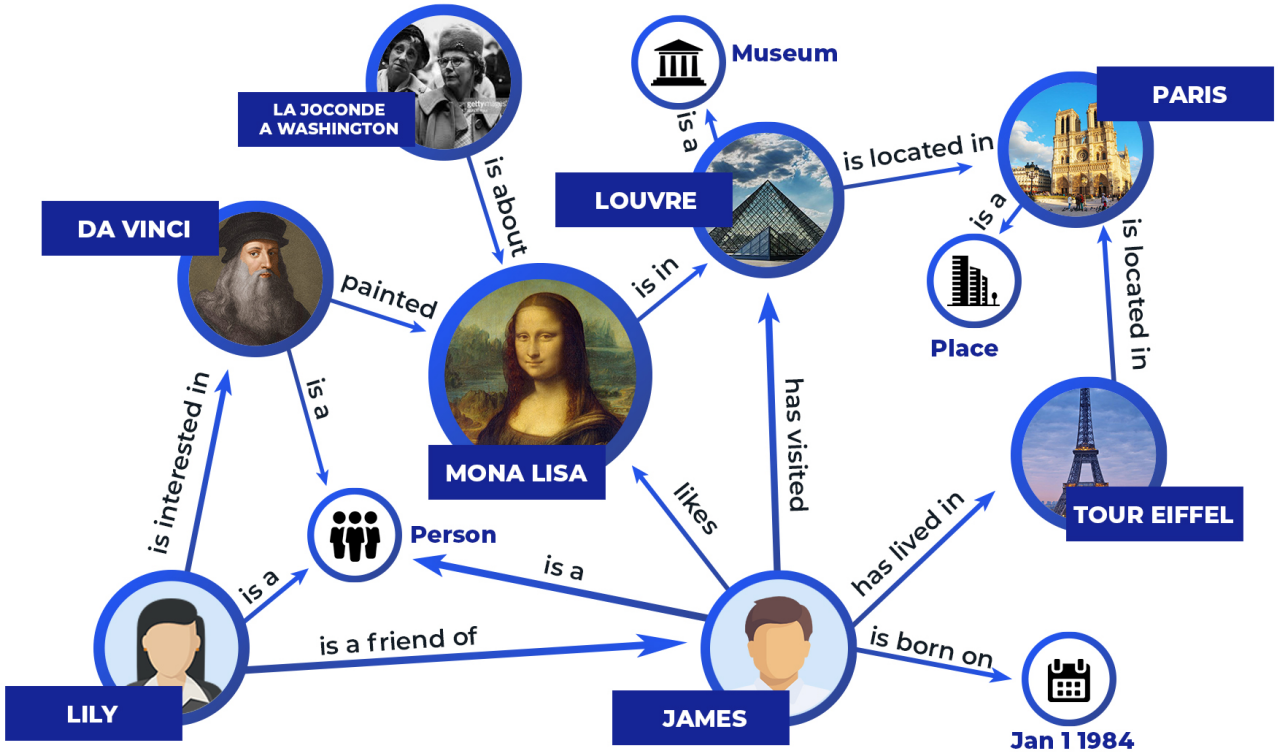
Example



- $\mathcal{D} = \{\texttt{isA}(\texttt{daVinci}, \texttt{Person}), \texttt{isIn}(\texttt{MonaLisa}, \texttt{Louvre}) \ldots\}$
- $\mathcal{O} = \{\texttt{Museum} \sqsubseteq \texttt{Place}, \texttt{Place} \sqcap \texttt{Person} \sqsubseteq \bot, \ldots\}$

Concept Learning

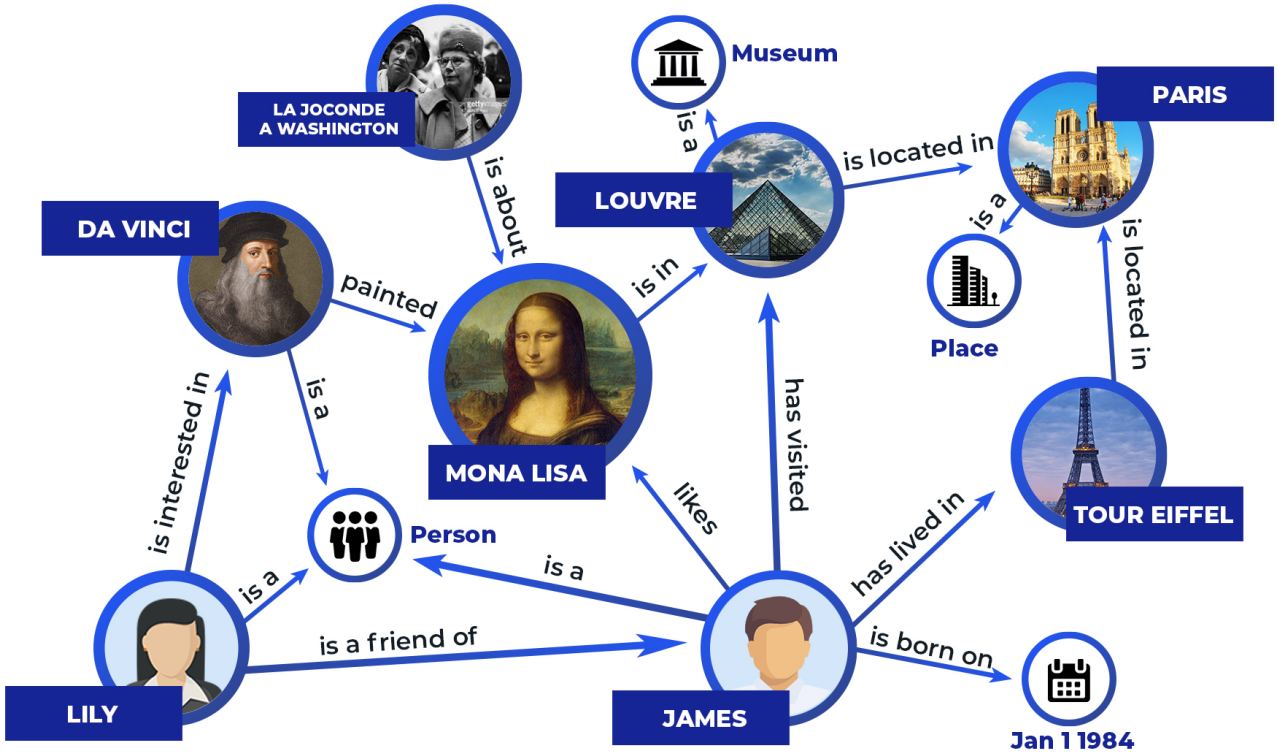
Example



- $E^+ = \{\texttt{Louvre}, \texttt{TourEiffel}\}, E^- = \{\texttt{Lily}, \texttt{James}\}$

Concept Learning

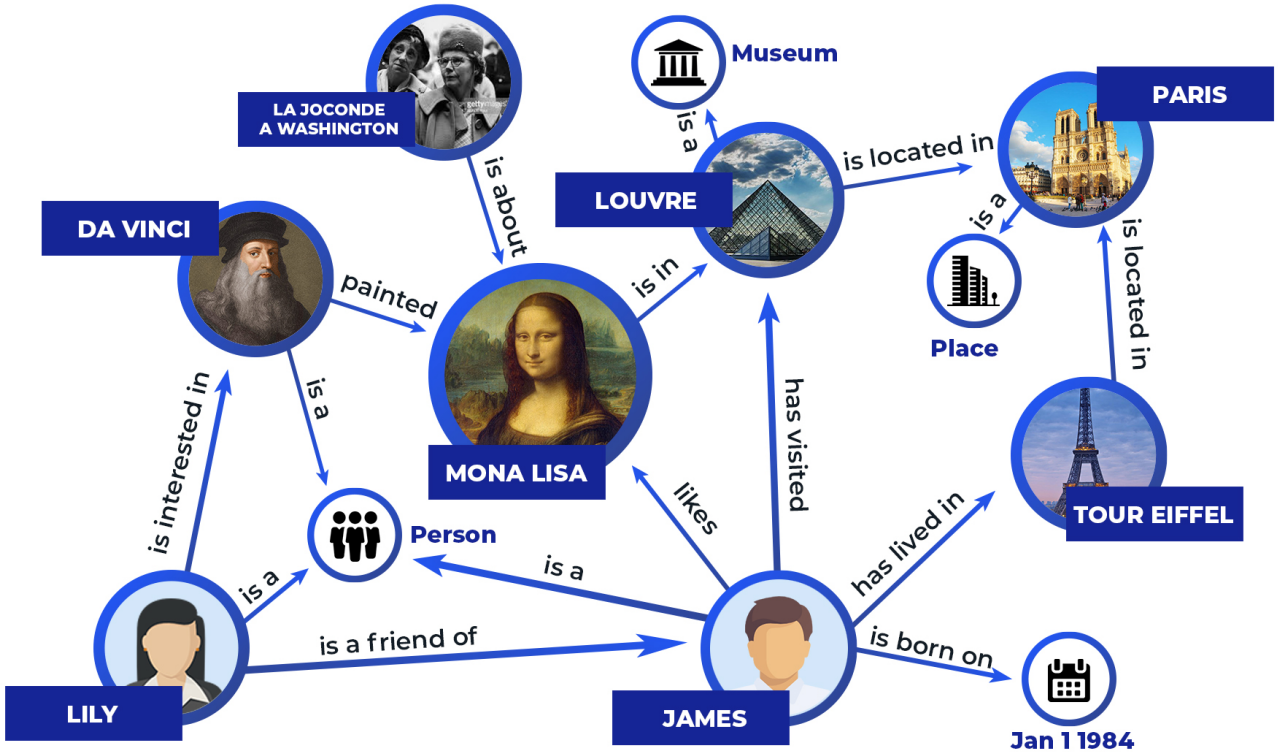
Example



- $E^+ = \{\texttt{Louvre}, \texttt{TourEiffel}\}, E^- = \{\texttt{Lily}, \texttt{James}\}$
- $H = \{\exists \texttt{locatedIn}.\{\texttt{Paris}\}, \exists \texttt{locatedIn}.\texttt{Place}\}$

Concept Learning

Example



- $E^+ = \{\texttt{Louvre}, \texttt{TourEiffel}\}, E^- = \{\texttt{Lily}, \texttt{James}\}$
- $H = \{\exists \texttt{locatedIn}.\{\texttt{Paris}\}, \exists \texttt{locatedIn}.\texttt{Place}\}$
- $H = \{\text{Things located in Paris, things located in a place}\} \Rightarrow$ XAI

Concept Learning

Formal definition

- Supervised learning with background knowledge (adapted from [Lehmann and Hitzler, 2010])
- Given
  - Formal logic $\mathcal{L}$, e.g., $\mathcal{ALC}$
  - Background knowledge in form of knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{D} \rangle$
  - Non-empty set of positive examples $E^+ \subseteq N_I$
  - Set of negative examples $E^- \subseteq N_I$

Concept Learning

Formal definition

- Supervised learning with background knowledge (adapted from [Lehmann and Hitzler, 2010])
- Given
  - Formal logic $\mathcal{L}$, e.g., $\mathcal{ALC}$
  - Background knowledge in form of knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{D} \rangle$
  - Non-empty set of positive examples $E^+ \subseteq N_I$
  - Set of negative examples $E^- \subseteq N_I$
- Goal: Find at least one hypothesis $H \in \mathcal{H}$ with
  1. H is a class expression in $\mathcal{L}$, and (ideally)
  2. $\forall e^+ \in E^+ : \mathcal{K} \models H(e^+)$
  3. $\forall e^- \in E^- : \mathcal{K} \not\models H(e^-)$

Concept Learning

Formal definition

- Supervised learning with background knowledge (adapted from [Lehmann and Hitzler, 2010])
- Given
  - Formal logic $\mathcal{L}$, e.g., $\mathcal{ALC}$
  - Background knowledge in form of knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{D} \rangle$
  - Non-empty set of positive examples $E^+ \subseteq N_I$
  - Set of negative examples $E^- \subseteq N_I$
- Goal: Find at least one hypothesis $H \in \mathcal{H}$ with
  1. H is a class expression in $\mathcal{L}$, and (ideally)
  2. $\forall e^+ \in E^+ : \mathcal{K} \models H(e^+)$
  3. $\forall e^- \in E^- : \mathcal{K} \not\models H(e^-)$

Implementations

Practically, find $H \in \underset{C \in \mathcal{L}}{argmax}\, Q(C)$ [Demir and Ngonga Ngomo, 2023]

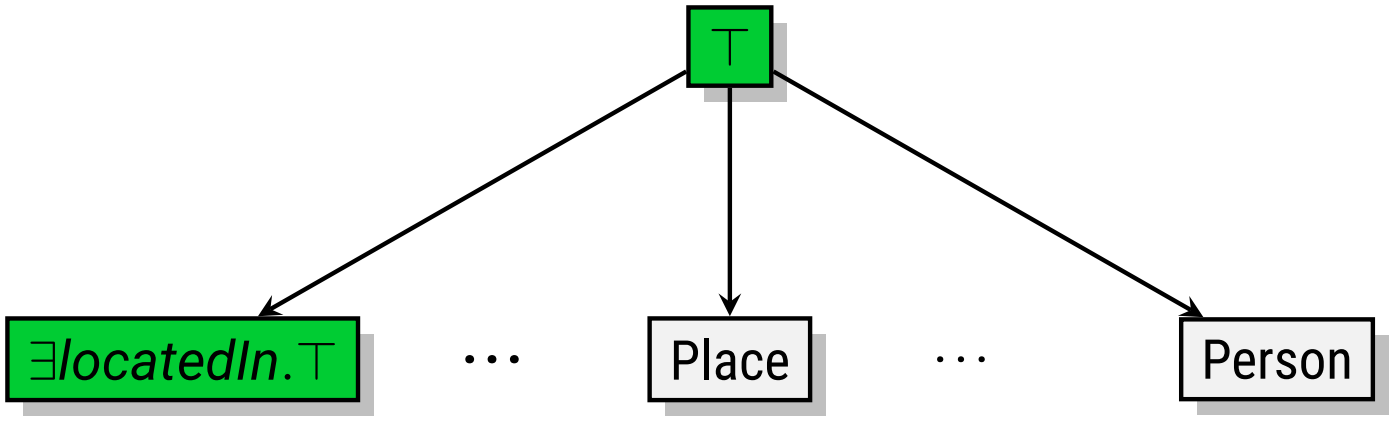
Concept Learning

Example



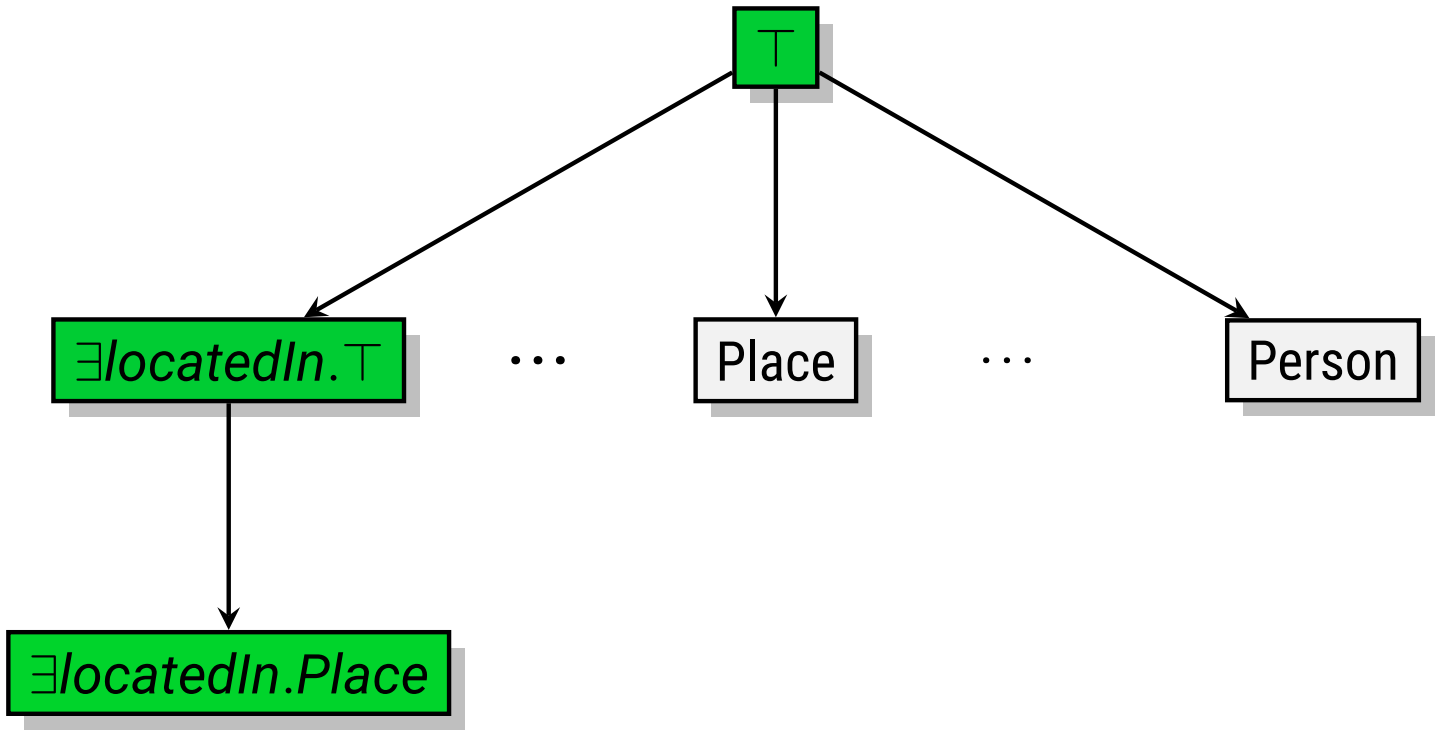
Concept Learning

Example



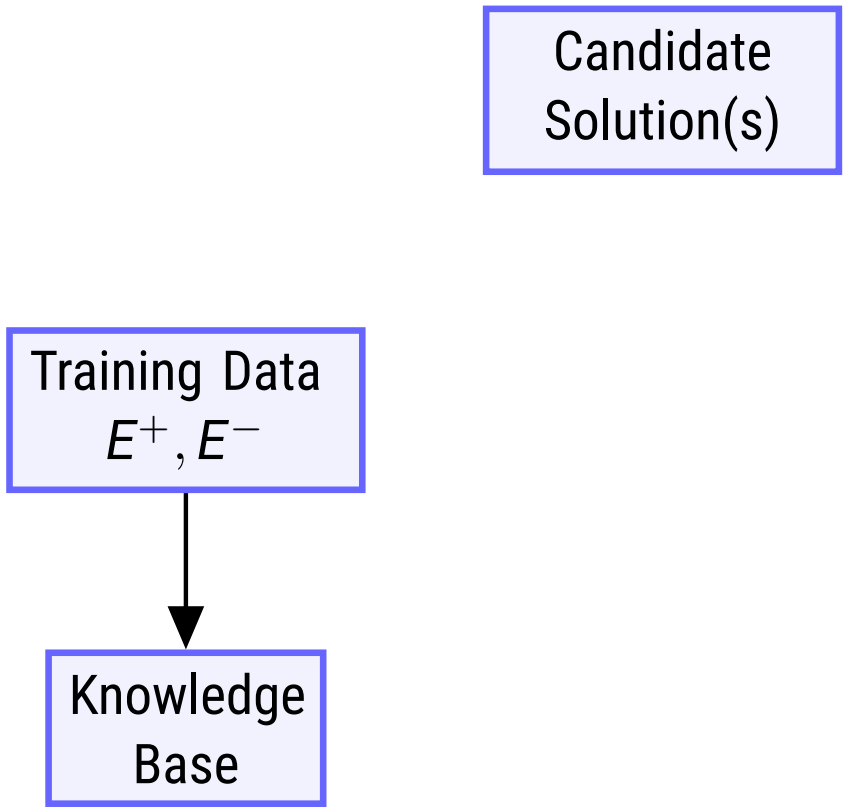
Concept Learning

Example



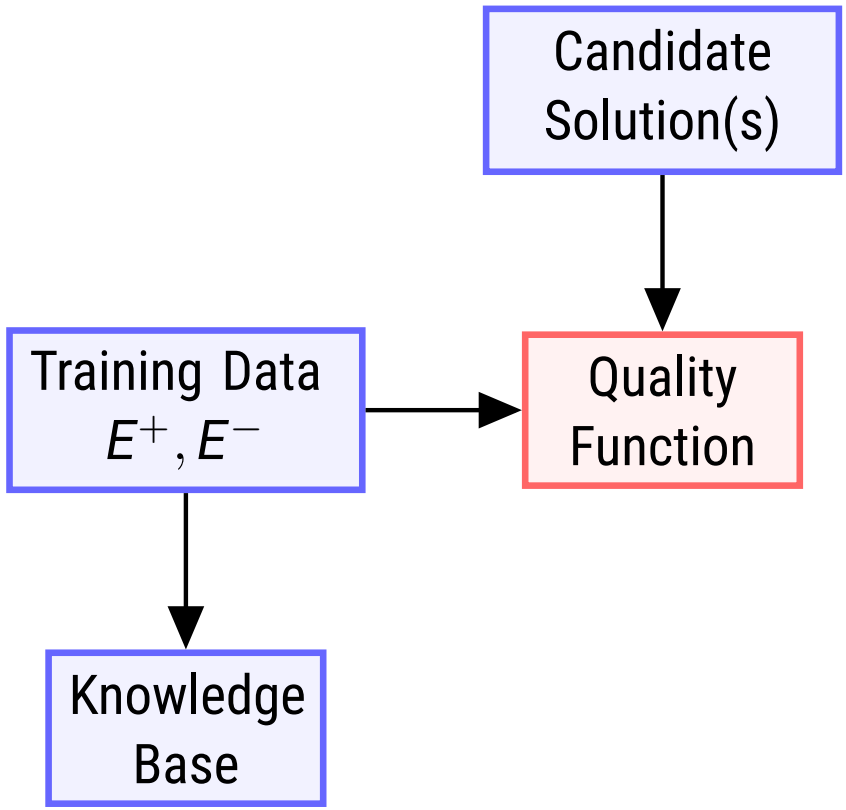
Concept Learning

Traditional Approach



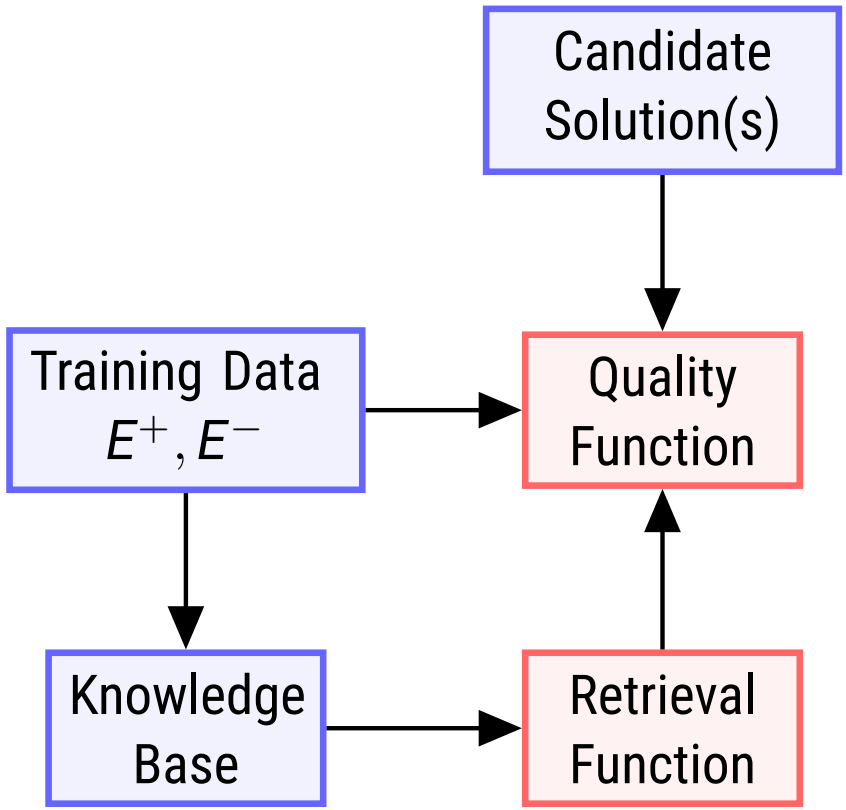
Concept Learning

Traditional Approach



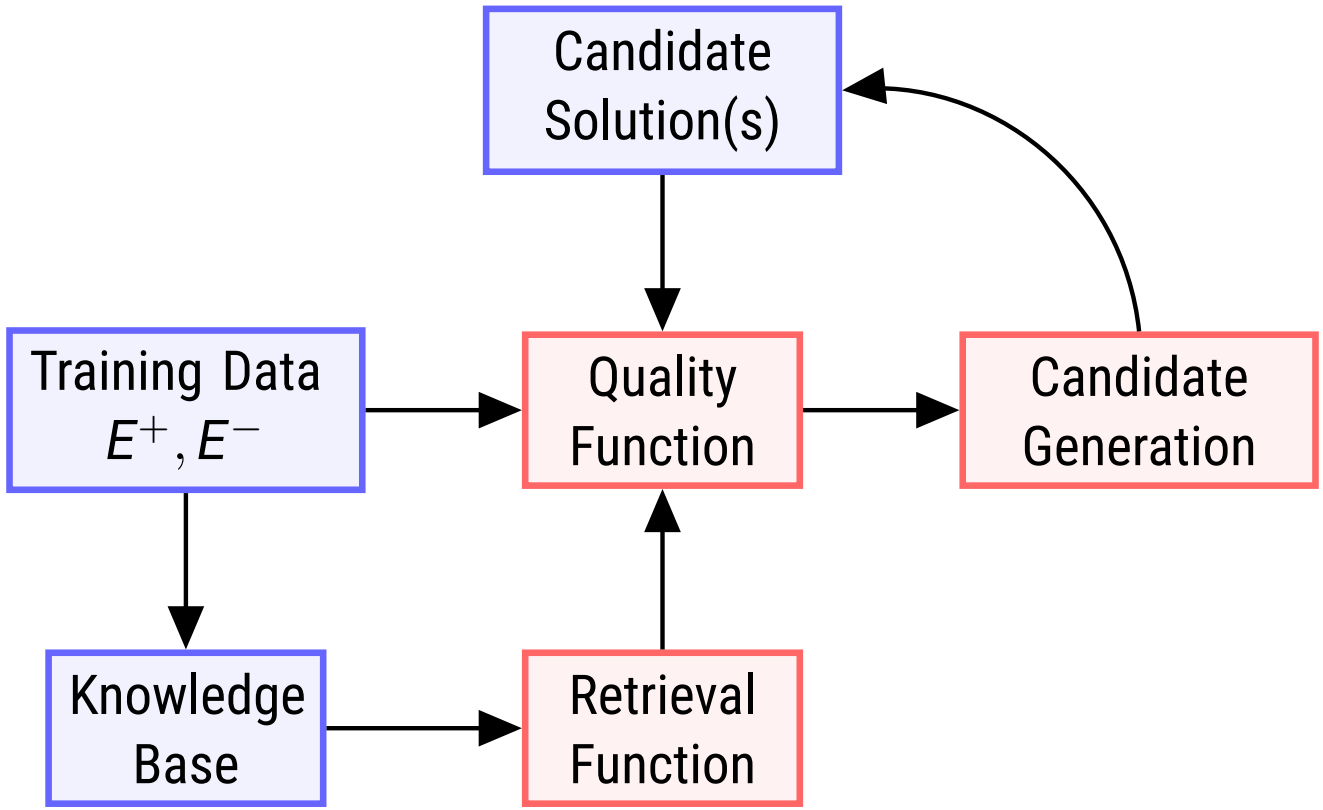
Concept Learning

Traditional Approach



Concept Learning

Traditional Approach



Concept Learning

CELOE

- Accuracy metric acc_c:

$$\mathrm{acc}_c(C,t) = \frac{1}{t+1} \cdot \left(t \cdot \frac{|E^+ \cap R(C)|}{|E^+|} + \sqrt{\frac{|E^+ \cap R(C)|}{|R(C)|}} \right)$$

$$\mathrm{acc_gain}_c(C) = \mathrm{acc}_c(C,t) - \mathrm{acc}_c(C',t)$$

Concept Learning

CELOE

- Accuracy metric acc_c:

$$\mathrm{acc}_c(C,t) = \frac{1}{t+1} \cdot \left(t \cdot \frac{|E^+ \cap R(C)|}{|E^+|} + \sqrt{\frac{|E^+ \cap R(C)|}{|R(C)|}} \right)$$

$$\mathrm{acc_gain}_c(C) = \mathrm{acc}_c(C,t) - \mathrm{acc}_c(C',t)$$

- $\mathrm{score}(C) = \mathrm{acc}_c(C,t) + \alpha \cdot \mathrm{acc_gain}_c(C) - \beta \cdot |C| \quad (\alpha, \beta \geq 0)$
  where typical values are $\alpha = 0.3$ and $\beta = 0.05$.

Concept Learning

CELOE

- Accuracy metric acc_c:

$$\mathrm{acc}_c(C,t) = \frac{1}{t+1} \cdot \left(t \cdot \frac{|E^+ \cap R(C)|}{|E^+|} + \sqrt{\frac{|E^+ \cap R(C)|}{|R(C)|}} \right)$$

$$\mathrm{acc_gain}_c(C) = \mathrm{acc}_c(C,t) - \mathrm{acc}_c(C',t)$$

- $\mathrm{score}(C) = \mathrm{acc}_c(C,t) + \alpha \cdot \mathrm{acc_gain}_c(C) - \beta \cdot |C| \quad (\alpha, \beta \geq 0)$
  where typical values are $\alpha = 0.3$ and $\beta = 0.05$.

Problem: Myopia

- Current metrics do not consider accuracy of future concepts

Concept Learning

CELOE

- Accuracy metric acc_c:

$$\mathrm{acc}_c(C,t) = \frac{1}{t+1} \cdot \left(t \cdot \frac{|E^+ \cap R(C)|}{|E^+|} + \sqrt{\frac{|E^+ \cap R(C)|}{|R(C)|}} \right)$$

$$\mathrm{acc_gain}_c(C) = \mathrm{acc}_c(C,t) - \mathrm{acc}_c(C',t)$$

- $\mathrm{score}(C) = \mathrm{acc}_c(C,t) + \alpha \cdot \mathrm{acc_gain}_c(C) - \beta \cdot |C| \quad (\alpha, \beta \geq 0)$
  where typical values are $\alpha = 0.3$ and $\beta = 0.05$.

Problem: Myopia

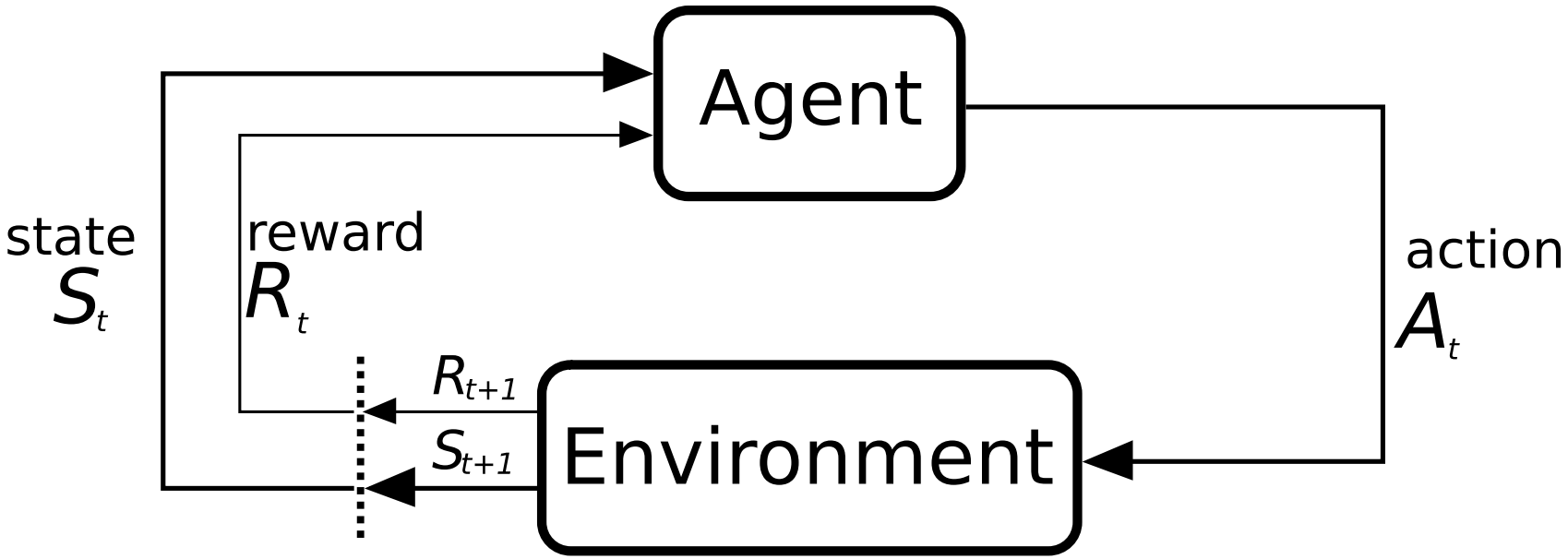
- Current metrics do not consider accuracy of future concepts
- Optimize for cumulative discounted future rewards
  [Demir and Ngonga Ngomo, 2023]

Section 3

Deep Reinforcement Learning

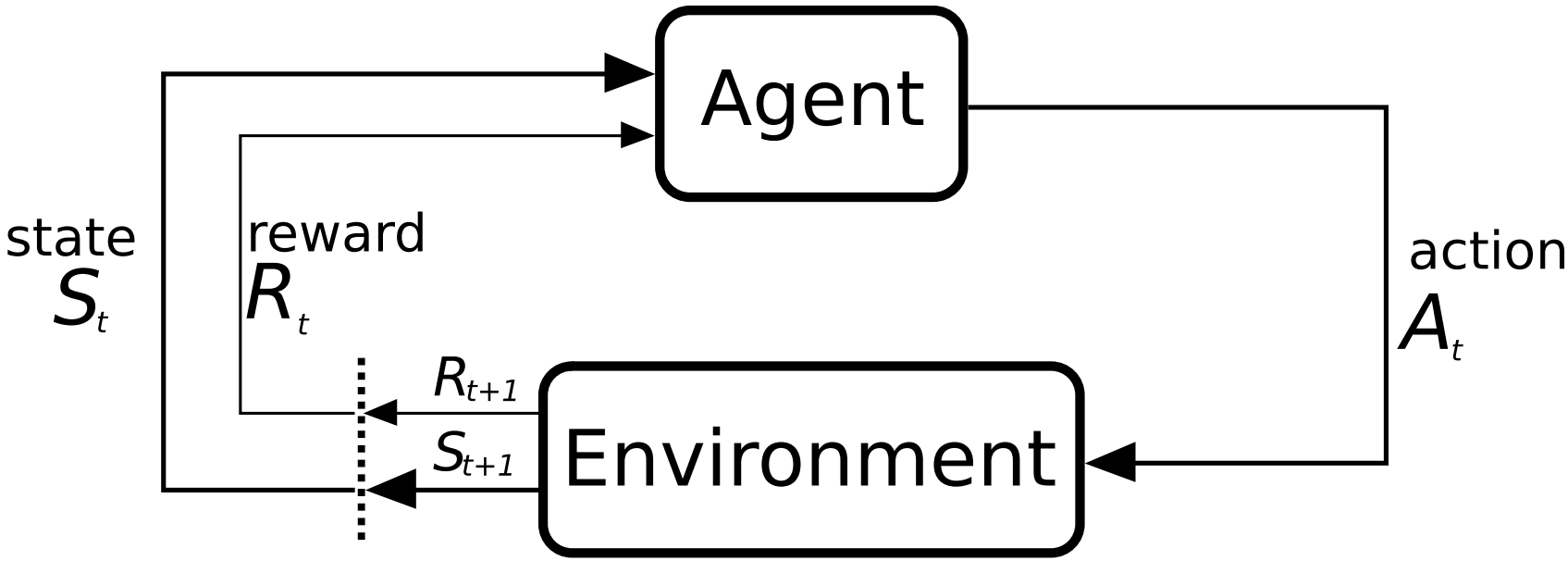
Deep Reinforcement Learning

Overview



Deep Reinforcement Learning

Overview

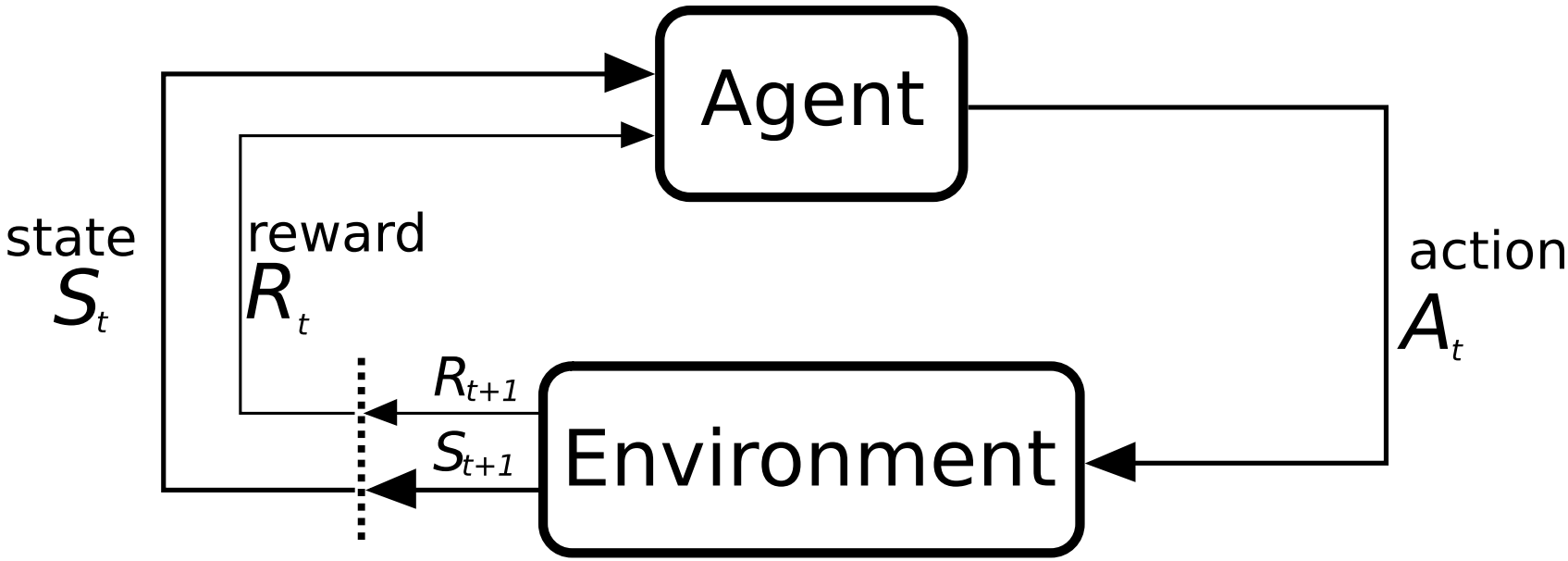


- Goal: Find policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ which maximizes accumulated reward

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i} \text{ with } \gamma \in (0, 1]$$

Deep Reinforcement Learning

Overview



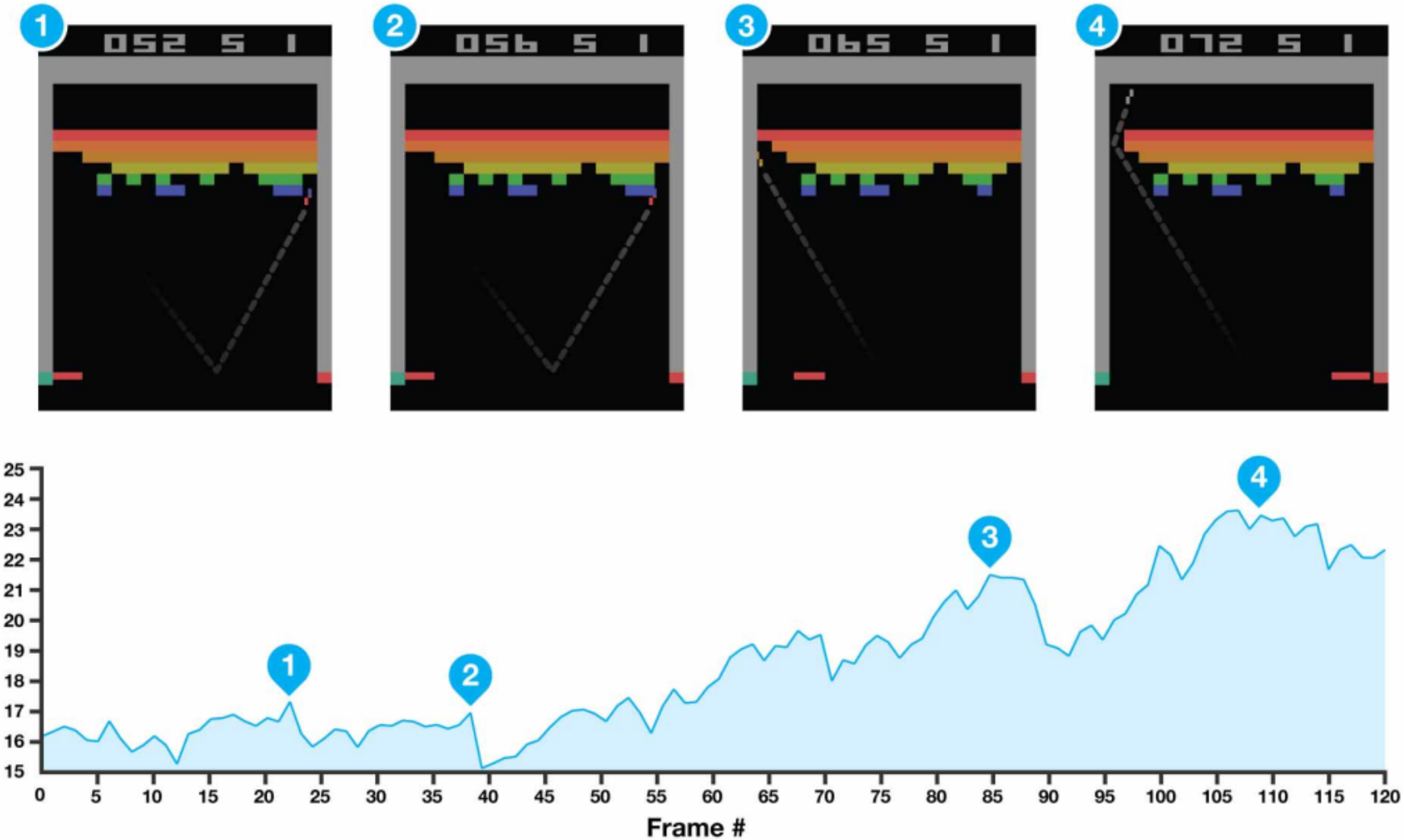
- Goal: Find policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ which maximizes accumulated reward

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i} \text{ with } \gamma \in (0, 1]$$

- Example: Atari, Chess, Go, water plants

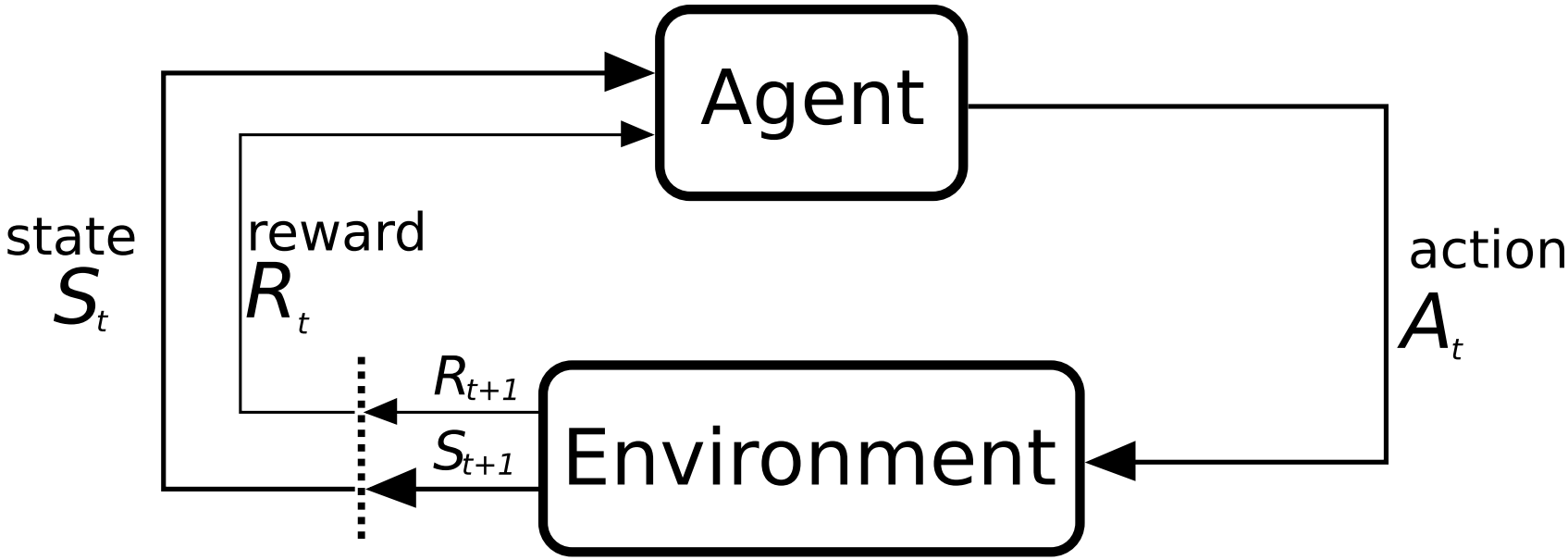
Deep Reinforcement Learning

Example



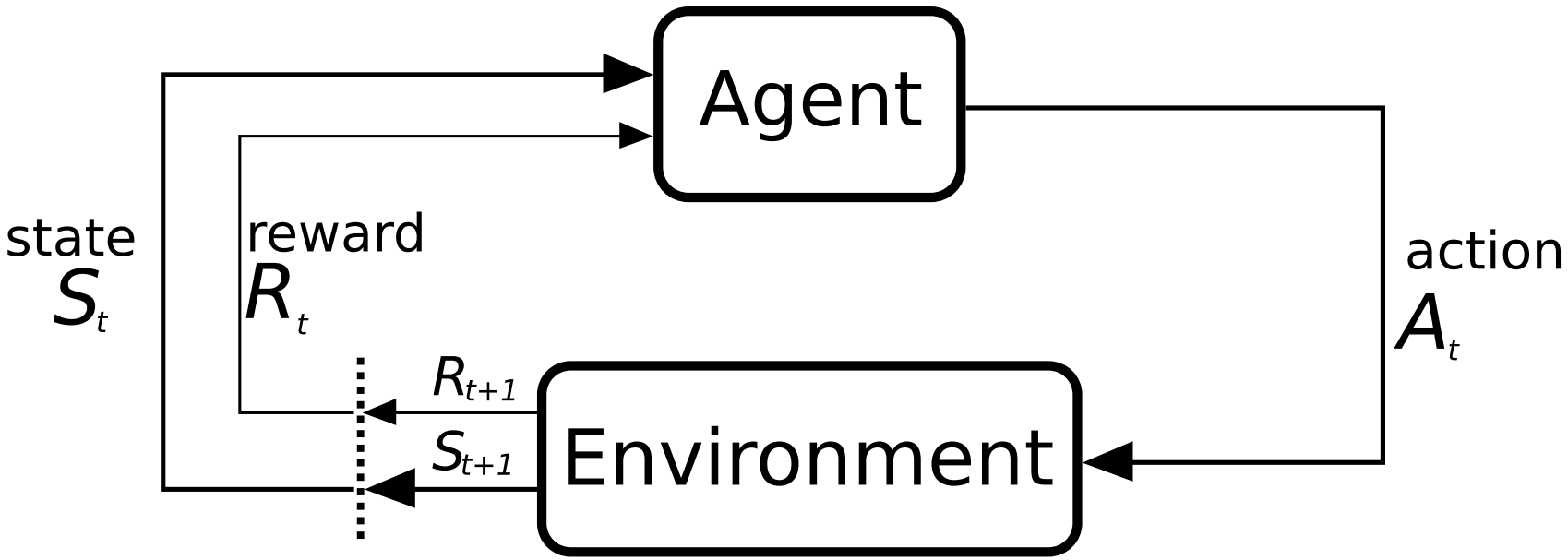
Deep Reinforcement Learning

Overview



Deep Reinforcement Learning

Overview



- S_t = Concept C
- $R_t = \begin{cases} 1 & \text{if } acc(C) = 1 \\ 0 & \text{else} \end{cases}$
- A_t = Transition from concept C to some concept D

Deep Reinforcement Learning

Action Value Function

- Maximize

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i} \text{ with } \gamma \in (0, 1]$$

Deep Reinforcement Learning

Action Value Function

- Maximize

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i} \text{ with } \gamma \in (0, 1]$$

- Optimize state-action value function $Q_\pi : S \times A \rightarrow \mathbb{R}$ with

$$Q_\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_\pi \left[G_t \mid S_t = \mathbf{s}, A_t = \mathbf{a} \right]$$

Deep Reinforcement Learning

Action Value Function

- Maximize

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i} \text{ with } \gamma \in (0, 1]$$

- Optimize state-action value function $Q_\pi : S \times A \rightarrow \mathbb{R}$ with

$$Q_\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_\pi \left[G_t \mid S_t = \mathbf{s}, A_t = \mathbf{a} \right]$$

- Observation: Infinite number of states as search space is infinite

Deep Reinforcement Learning

Action Value Function

- Maximize

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i} \text{ with } \gamma \in (0, 1]$$

- Optimize state-action value function $Q_\pi : S \times A \rightarrow \mathbb{R}$ with

$$Q_\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_\pi \left[G_t \mid S_t = \mathbf{s}, A_t = \mathbf{a} \right]$$

- Observation: Infinite number of states as search space is infinite
- Apply deep Q learning with target network [Mnih et al., 2015]

$$\mathcal{L}(\Theta_i) = \mathbb{E}_{(s,a,R,s') \sim U(\mathcal{D})} \left[\left(R + \gamma \max_{\mathbf{a}' \in A(\mathbf{s}')} Q(s', a'; \Theta_i^-) - Q(s, a; \Theta_i) \right)^2 \right]$$

Deep Reinforcement Learning

State Value Function

- Maximize

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i}$$

Deep Reinforcement Learning

State Value Function

- Maximize

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i}$$

- Optimize state value function $V_\pi : S \to \mathbb{R}$ with

$$V_\pi(\mathbf{s}) = \mathbb{E}_\pi \left[G_t \mid S_t = \mathbf{s} \right]$$

- Observation: Actions do not matter, only states do

Deep Reinforcement Learning

State Value Function

- Maximize

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i}$$

- Optimize state value function $V_\pi : S \to \mathbb{R}$ with

$$V_\pi(\mathbf{s}) = \mathbb{E}_\pi \left[G_t \mid S_t = \mathbf{s} \right]$$

- Observation: Actions do not matter, only states do
- Apply deep V learning with target network

$$\mathcal{L}(\Theta_i) = \mathbb{E}_{(s,R,s') \sim U(\mathcal{D})} \left[\left(R + \gamma \max_{\mathbf{s}' \in \rho(\mathbf{s})} V(s'; \Theta_i^-) - V(s; \Theta_i) \right)^2 \right]$$

Deep Reinforcement Learning

Self-Supervised Learning

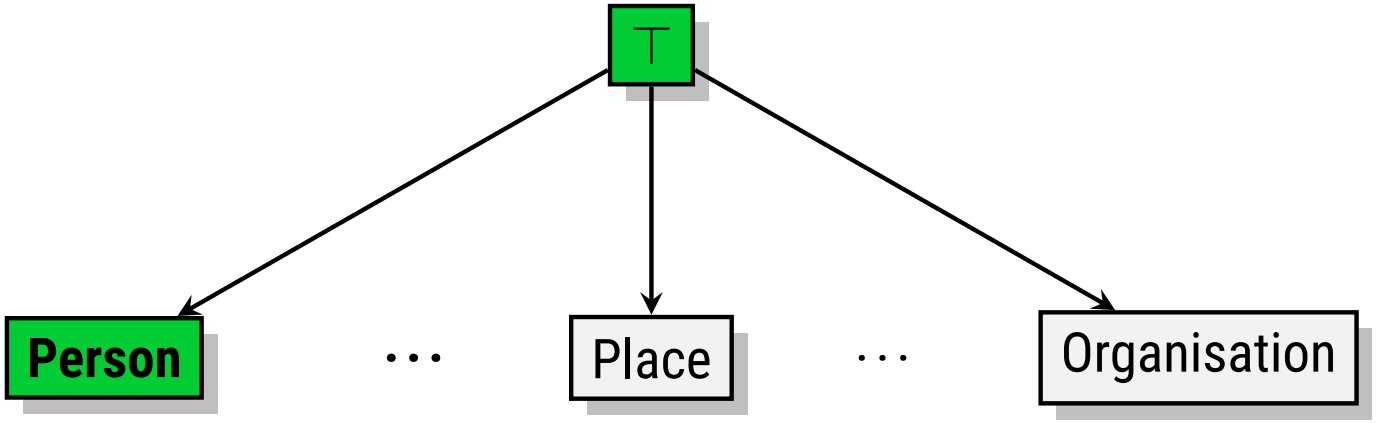
- Follow refinement path at random
- Select concept C
- Set $E^+ \subseteq R(C)$ and $E^- \cap R(C) = \emptyset$



Deep Reinforcement Learning

Self-Supervised Learning

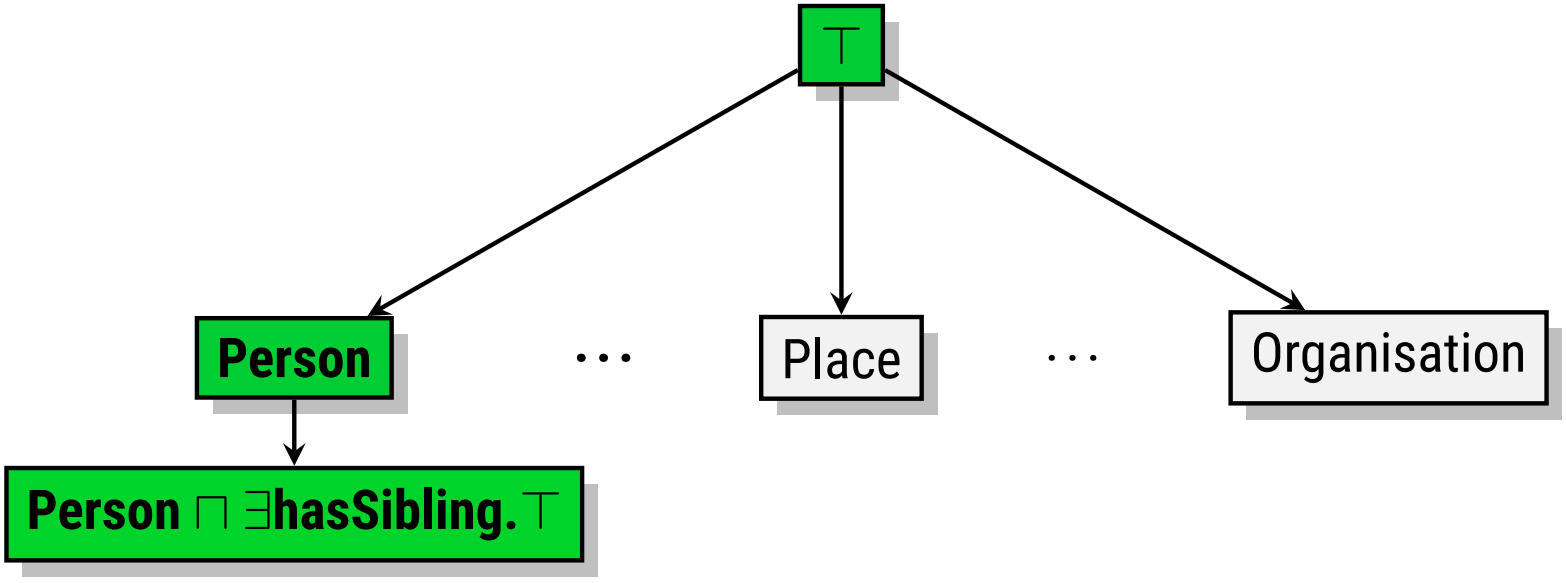
- Follow refinement path at random
- Select concept C
- Set $E^+ \subseteq R(C)$ and $E^- \cap R(C) = \emptyset$



Deep Reinforcement Learning

Self-Supervised Learning

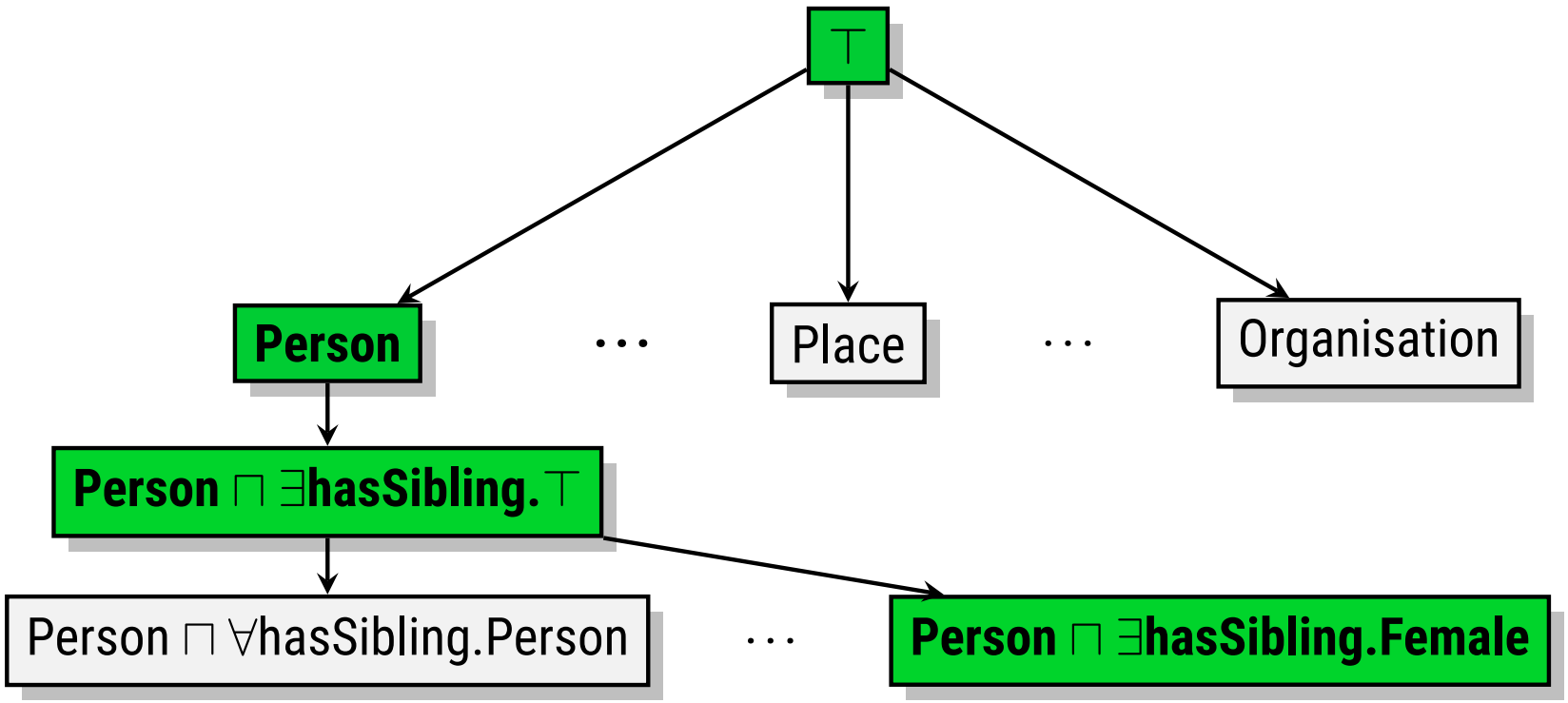
- Follow refinement path at random
- Select concept C
- Set $E^+ \subseteq R(C)$ and $E^- \cap R(C) = \emptyset$



Deep Reinforcement Learning

Self-Supervised Learning

- Follow refinement path at random
- Select concept C
- Set $E^+ \subseteq R(C)$ and $E^- \cap R(C) = \emptyset$



Deep Reinforcement Learning

Evaluation

- Used Family und BioPax datasets
- Evaluation on 114 learning problems

Approaches	F1	Acc	Runtime	# Exp.
CELOE	$.995 \pm 0.03$	$.993 \pm 0.04$	7.5 ± 1.1	33.5 ± 129.3
OCEL	*	1.00 ± 0.00	11.0 ± 1.4	2271.6 ± 1269.2
ELTL	$.990 \pm 0.06$	$.984 \pm 0.09$	8.1 ± 1.6	*
Q-Learning	1.00 ± 0.00	1.00 ± 0.00	1.1 ± 0.5	9.88 ± 38.5

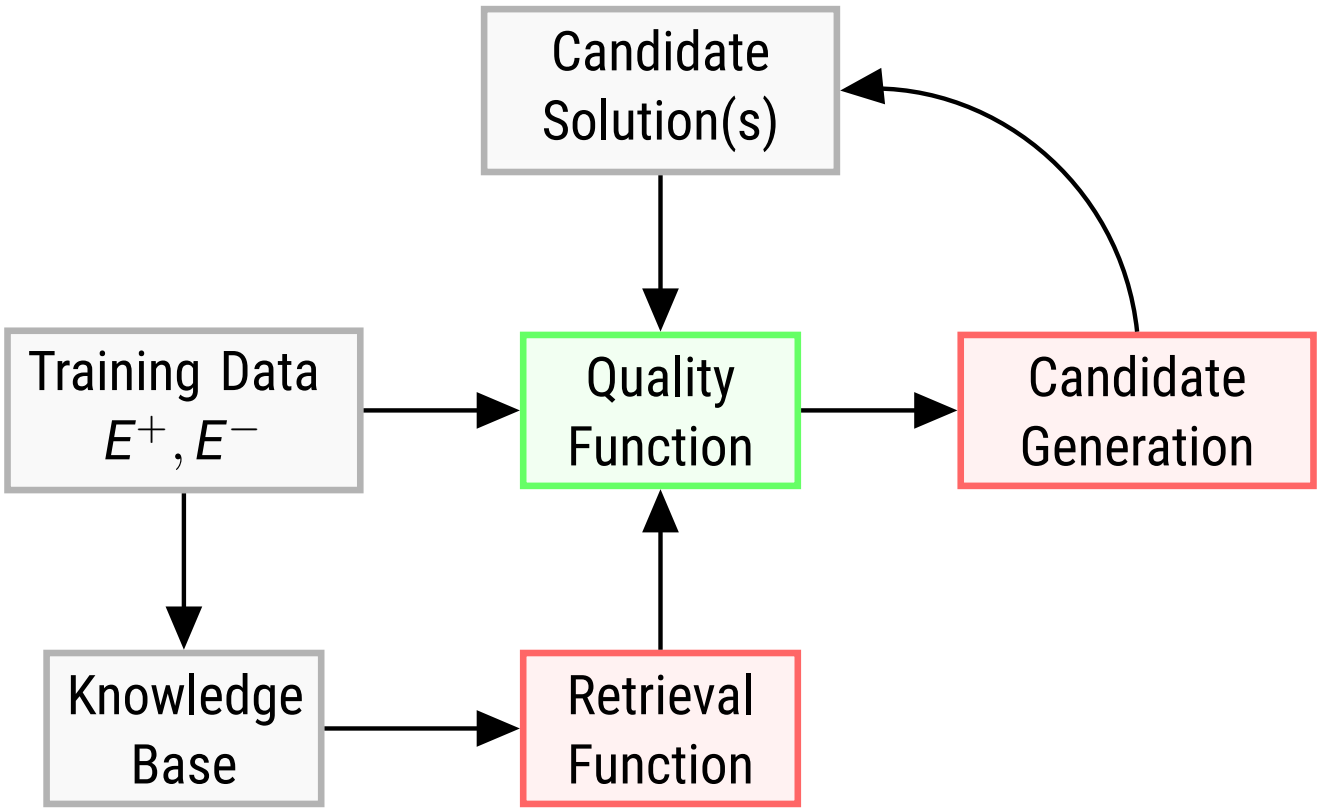
Deep Reinforcement Learning

Evaluation (ongoing)

	Carcinogenesis	Mutagenesis	Vicodi	Semantic Bible
Test F1 (↑)				
Q Learning	1.00 ± 0.00	0.95 ± 0.14	0.74 ± 0.21	0.65 ± 0.27
V Learning	1.00 ± 0.00	**0.97 ± 0.10**	**0.81 ± 0.21**	**0.78 ± 0.25**
Train F1 (↑)				
Q Learning	1.00 ± 0.00	0.97 ± 0.06	0.76 ± 0.19	0.73 ± 0.19
V Learning	1.00 ± 0.00	**0.99 ± 0.05**	**0.82 ± 0.19**	**0.85 ± 0.15**
Runtimes (↓)				
Q Learning	**0.37 ± 0.07**	14.18 ± 29.84	78.07 ± 19.12	62.92 ± 4.23
V Learning	0.44 ± 0.12	**8.53 ± 22.17**	**71.03 ± 16.88**	**56.16 ± 13.83**
Number of tested Concepts (↓)				
Q Learning	4.40 ± 6.5	140.9 ± 295.3	253.0 ± 60.5	13271 ± 1605
V Learning	**3.6 ± 0.5**	**21.9 ± 51.0**	**69.5 ± 17.3**	**2265 ± 684**

Learning problem

Challenges

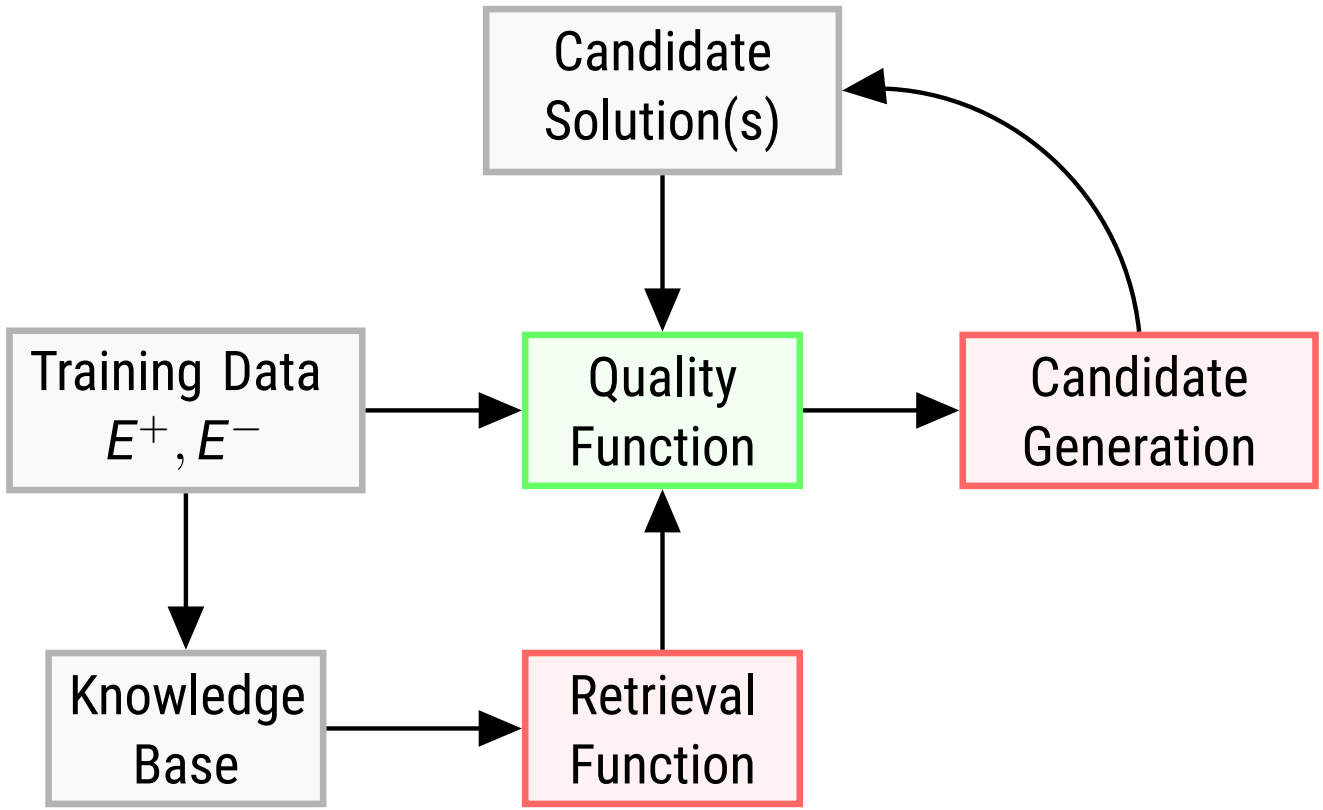


✓ Quality functions are often myopic ⇒ Deep Reinforcement Learning

- But . . .

Learning problem

Challenges



- ✓ Quality functions are often myopic ⇒ Deep Reinforcement Learning
- ► But . . . where do the vectors come from?

Section 4

Embeddings

Embeddings

Overview

- Goal
  - Provide faithful vectorial representation of semantics in $\mathcal{K}$
  - Often modelled via optimisation of a score function
- DistMult [Yang et al., 2015]
  - For all $(h, r, t) \in \mathcal{K}$ set $\varphi(h), \varphi(r), \varphi(t) \in \mathbb{R}^d$
  - Sample $(h', r', t') \notin \mathcal{K}$ for each $(h, r, t) \in \mathcal{K}$
  - $\varphi(s, p, o) = \sum_{i=1}^{d} s_i p_i o_i$
  - Loss function over training data Γ with $Y_{spo} \in \{-1, +1\}$ is given by $\min \sum_{(s,p,o)\in\Gamma} \log(1 + exp(-Y_{spo}\varphi(s, p, o)))$

Embeddings

Overview

- Goal
  - Provide faithful vectorial representation of semantics in $\mathcal{K}$
  - Often modelled via optimisation of a score function
- DistMult [Yang et al., 2015]
  - For all $(h, r, t) \in \mathcal{K}$ set $\varphi(h), \varphi(r), \varphi(t) \in \mathbb{R}^d$
  - Sample $(h', r', t') \notin \mathcal{K}$ for each $(h, r, t) \in \mathcal{K}$
  - $\varphi(s, p, o) = \sum_{i=1}^{d} s_i p_i o_i$
  - Loss function over training data Γ with $Y_{spo} \in \{-1, +1\}$ is given by $\min \sum_{(s,p,o)\in\Gamma} \log(1 + exp(-Y_{spo}\varphi(s, p, o)))$

Embeddings

Overview

- Goal
  - Provide faithful vectorial representation of semantics in $\mathcal{K}$
  - Often modelled via optimisation of a score function
- DistMult [Yang et al., 2015]
  - For all $(h, r, t) \in \mathcal{K}$ set $\varphi(h), \varphi(r), \varphi(t) \in \mathbb{R}^d$
  - Sample $(h', r', t') \notin \mathcal{K}$ for each $(h, r, t) \in \mathcal{K}$
  - $\varphi(s, p, o) = \sum_{i=1}^{d} s_i p_i o_i$
  - Loss function over training data Γ with $Y_{spo} \in \{-1, +1\}$ is given by $\min \sum_{(s,p,o)\in\Gamma} \log(1 + exp(-Y_{spo}\varphi(s, p, o)))$
- ComplEx [Trouillon et al., 2016]
  - Extension to complex numbers $\mathbb{C}$
  - $\varphi(s, p, o) = \sum_{i=1}^{d} \mathfrak{Re}(s_i p_i \bar{o}_i)$
  - Similar construction for Hamiltonians, and octonions

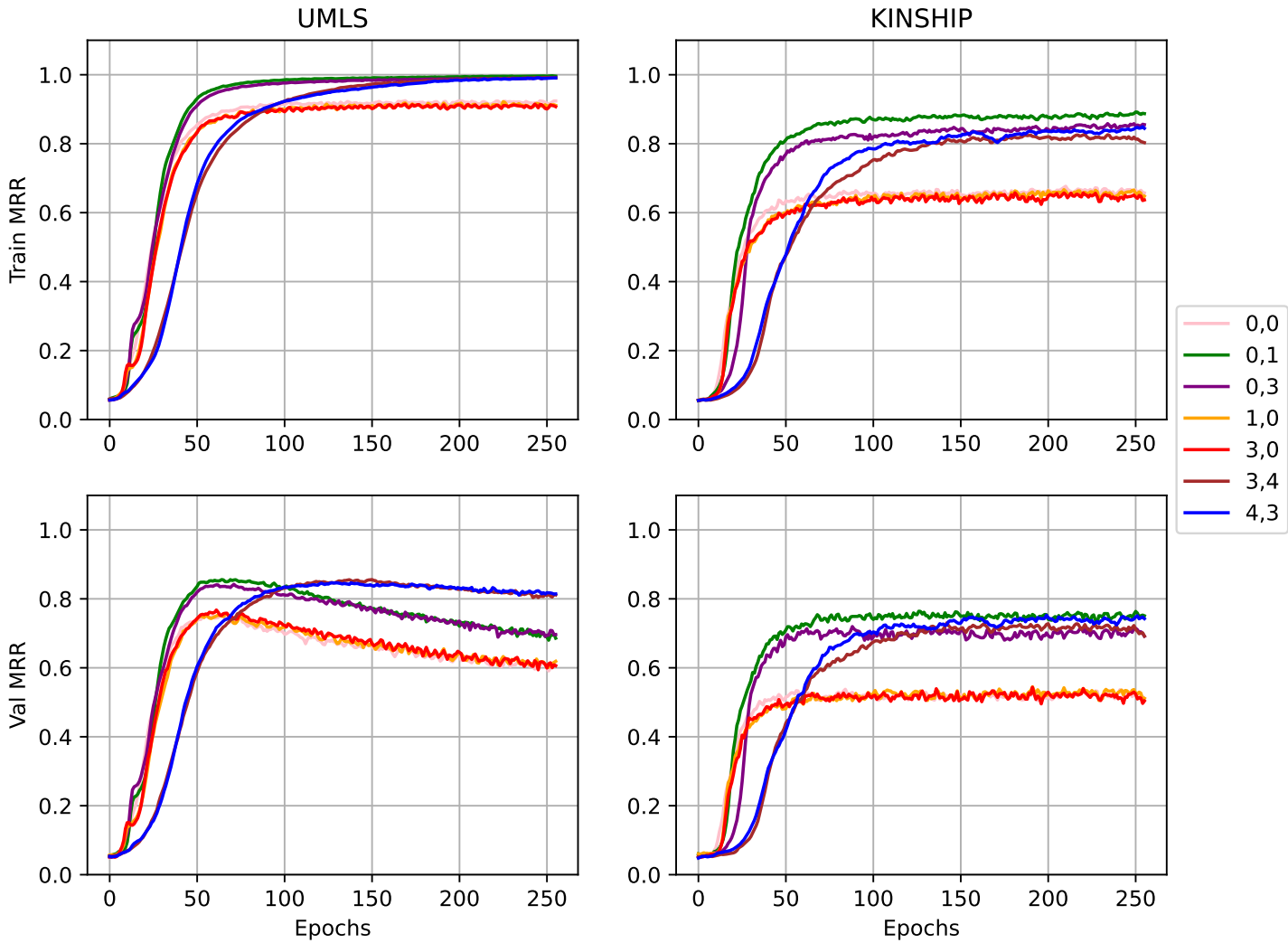
Embeddings

Generalizations

- Generalization to Clifford Algebras $Cl_{p,q,r}$ [Teyou et al., 2024] with
  - $e_1^2 = \ldots = e_p^2 = 1$
  - $e_{p+1}^2 = \ldots = e_{p+q}^2 = -1$
  - $e_{p+q+1}^2 = \ldots = e_{p+q+r}^2 = 0$
- $\vec{s}_m = \left(a + \sum_{i=1}^{p} b_i e_i + \sum_{j=p+1}^{p+q} c_j e_j + \sum_{j=p+q+1}^{p+q+r} d_k e_k \right)$
- $\varphi(s, p, o) = \sum s_m p_m o_m$
- Parameterised dimensions allows for embedding space search

Embeddings

Grid search for $0 \leq p, q \leq 4, r = 0$



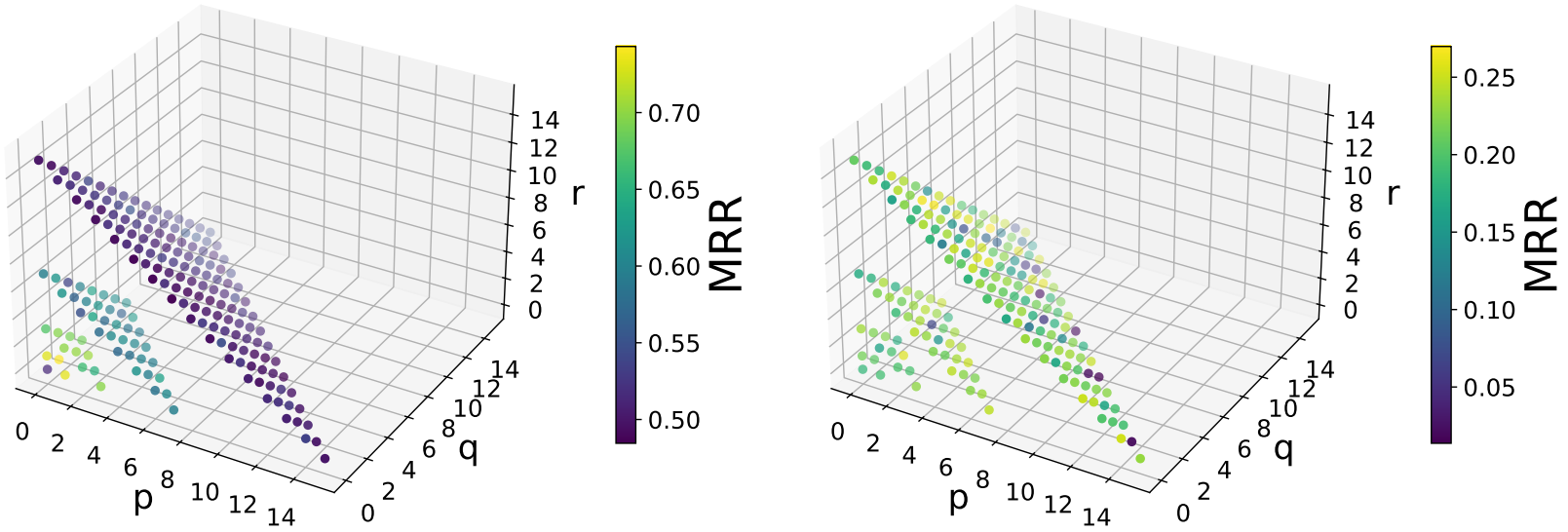
Embeddings

Evaluation of KECI on link prediction

Models	WN18-RR			
	MRR	H@1	H@3	H@10
DistMult	0.231	0.191	0.249	0.305
ComplEx	0.274	0.233	0.295	0.349
QMult	0.242	0.199	0.262	0.318
OMult	0.121	0.080	0.135	0.203
MuRE	0.259	0.202	0.280	0.366
MuRP	0.212	0.160	0.234	0.309
RotH	0.230	0.177	0.247	0.332
RotE	0.258	0.203	0.280	0.357
DualE	0.157	0.118	0.174	0.228
KECI	0.285	0.245	0.302	0.357
DECAL + LES	0.291	0.249	0.313	0.364
DECAL + GSDC	**0.296**	0.250	**0.323**	**0.381**
DECAL + GS	0.295	**0.252**	0.317	0.372
DECAL + VSP	0.285	0.245	0.302	0.357

Embeddings

Paramater search for $d \equiv 0\ [p + q + r + 1]$



- Small dimensionality often sufficient for simple datasets (Kinship)
- No final verdict on optimization for (p, q, r)

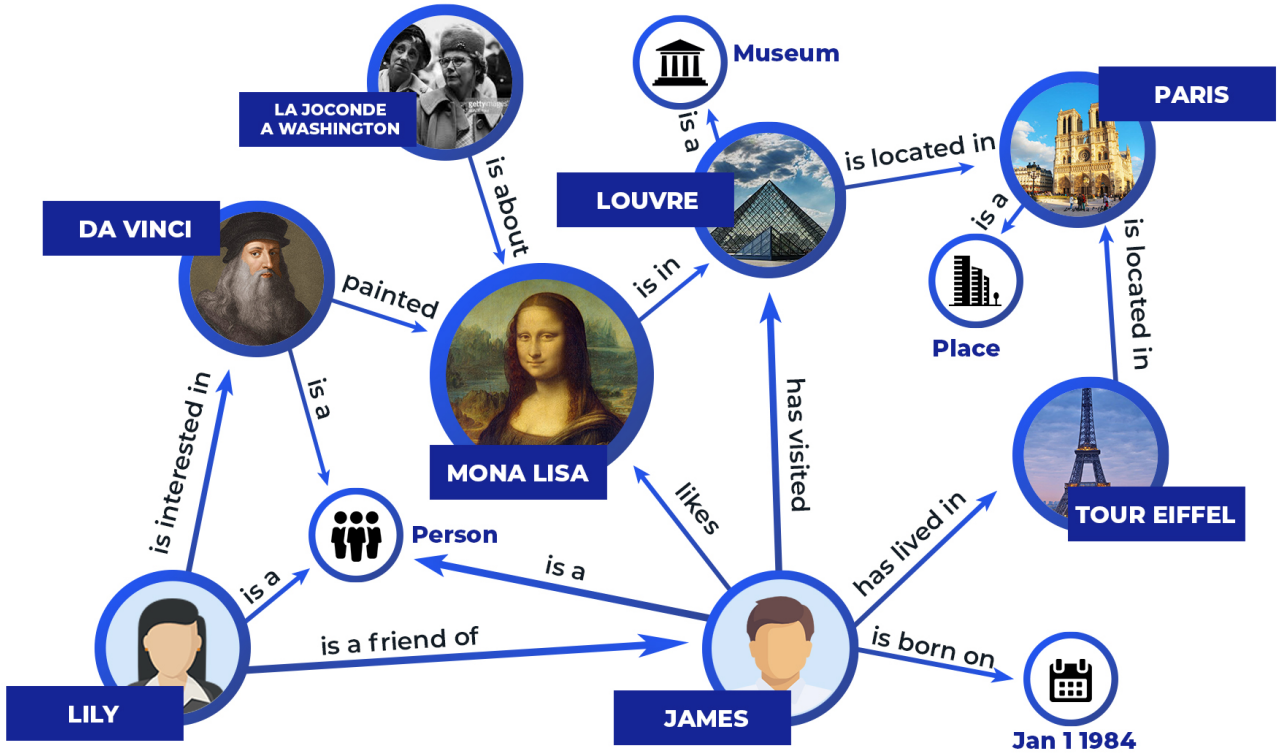
Section 5

Retrieval

Goal

Goals

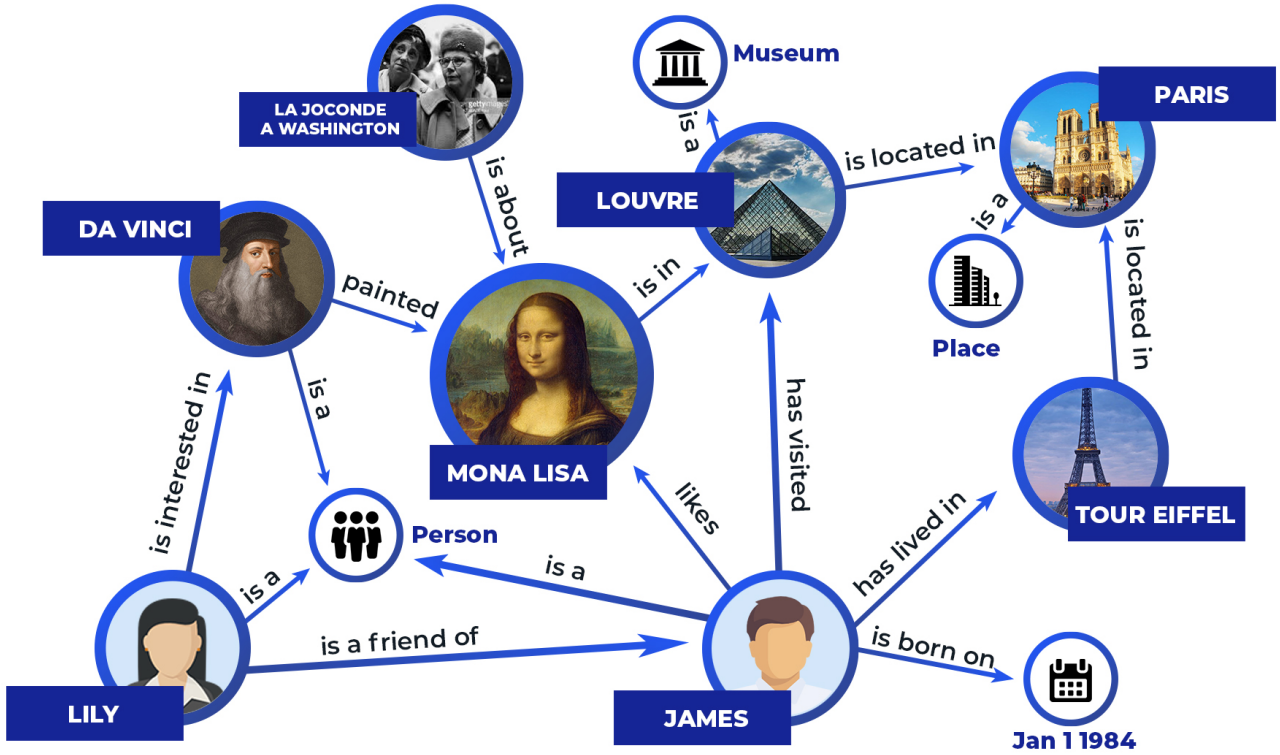
- Given concept *C*, find all instances of *C* in input data



Goal

Goals

- Given concept *C*, find all instances of *C* in input data
- Operation is EXPTIME-complete in $\mathcal{ALC}$

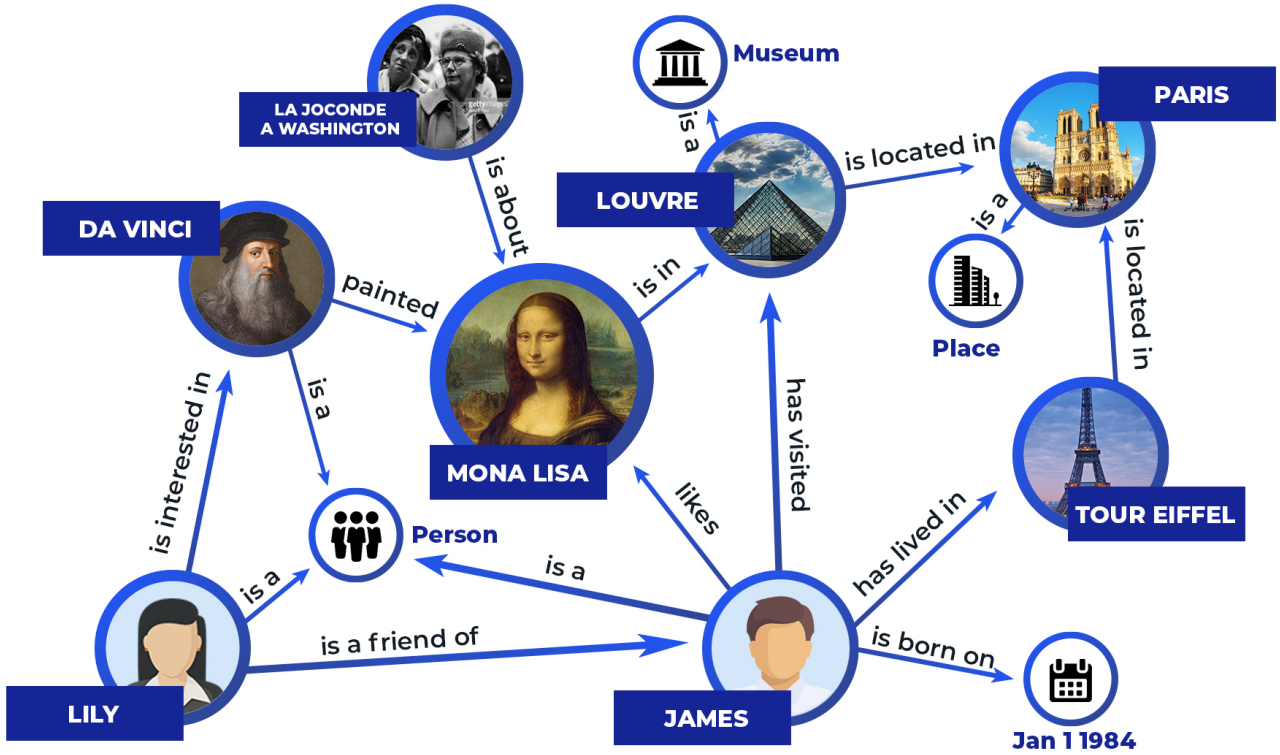


2

Goal

Goals

- Given concept C, find all instances of C in input data
- Operation is EXPTIME-complete in $\mathcal{ALC}$
- Accounts for ca. 90% of runtime of standard learners



Retrieval

From DLs to SPARQL

- Assume closed world and fully materialized knowledge graph
- Retrieval can be realized via SPARQL queries
  [Bin et al., 2016, Karalis et al., 2024]

Retrieval

From DLs to SPARQL

- Assume closed world and fully materialized knowledge graph
- Retrieval can be realized via SPARQL queries
  [Bin et al., 2016, Karalis et al., 2024]

Class Expression	Graph Pattern $\mathfrak{p} = \tau(C_i, \texttt{?var})$
$A \in N_C$	`?var rdf:type A.`

Retrieval

From DLs to SPARQL

- Assume closed world and fully materialized knowledge graph
- Retrieval can be realized via SPARQL queries
  [Bin et al., 2016, Karalis et al., 2024]

Class Expression	Graph Pattern $\mathfrak{p} = \tau(C_i, \texttt{?var})$
$A \in N_C$	`?var rdf:type A.`
$\neg C$	`{?var ?p ?o} UNION {?s ?p ?var}.` `FILTER NOT EXISTS` $\{\tau(C, \texttt{?var})\}$

Retrieval

From DLs to SPARQL

- Assume closed world and fully materialized knowledge graph
- Retrieval can be realized via SPARQL queries
  [Bin et al., 2016, Karalis et al., 2024]

Class Expression	Graph Pattern $\mathfrak{p} = \tau(C_i, \texttt{?var})$
$A \in N_C$	`?var rdf:type A.`
$\neg C$	`{?var ?p ?o} UNION {?s ?p ?var}.` `FILTER NOT EXISTS` $\{\tau(C, \texttt{?var})\}$
$C_1 \sqcap \ldots \sqcap C_n$	$\{\tau(C_1, \texttt{?var}) \ldots \tau(C_n, \texttt{?var})\}$

Retrieval

From DLs to SPARQL

- Assume closed world and fully materialized knowledge graph
- Retrieval can be realized via SPARQL queries
  [Bin et al., 2016, Karalis et al., 2024]

Class Expression	Graph Pattern $\mathfrak{p} = \tau(C_i, \texttt{?var})$
$A \in N_C$	`?var rdf:type A.`
$\neg C$	`{?var ?p ?o} UNION {?s ?p ?var}.` `FILTER NOT EXISTS` $\{\tau(C, \texttt{?var})\}$
$C_1 \sqcap \ldots \sqcap C_n$	$\{\tau(C_1, \texttt{?var}) \ldots \tau(C_n, \texttt{?var})\}$
$C_1 \sqcup \ldots \sqcup C_n$	$\{\tau(C_1, \texttt{?var})\}$ `UNION` $\ldots$ `UNION` $\{\tau(C_n, \texttt{?var})\}$

Retrieval

From DLs to SPARQL

- Assume closed world and fully materialized knowledge graph
- Retrieval can be realized via SPARQL queries
  [Bin et al., 2016, Karalis et al., 2024]

Class Expression	Graph Pattern $\mathfrak{p} = \tau(C_i, \texttt{?var})$
$A \in N_C$	`?var rdf:type A.`
$\neg C$	`{?var ?p ?o} UNION {?s ?p ?var}.` `FILTER NOT EXISTS` $\{\tau(C, \texttt{?var})\}$
$C_1 \sqcap \ldots \sqcap C_n$	$\{\tau(C_1, \texttt{?var}) \ldots \tau(C_n, \texttt{?var})\}$
$C_1 \sqcup \ldots \sqcup C_n$	$\{\tau(C_1, \texttt{?var})\}$ `UNION` $\ldots$ `UNION` $\{\tau(C_n, \texttt{?var})\}$
$\exists\, r.C$	{`?var r ?s.` $\tau(C, \texttt{?s})$}

Retrieval

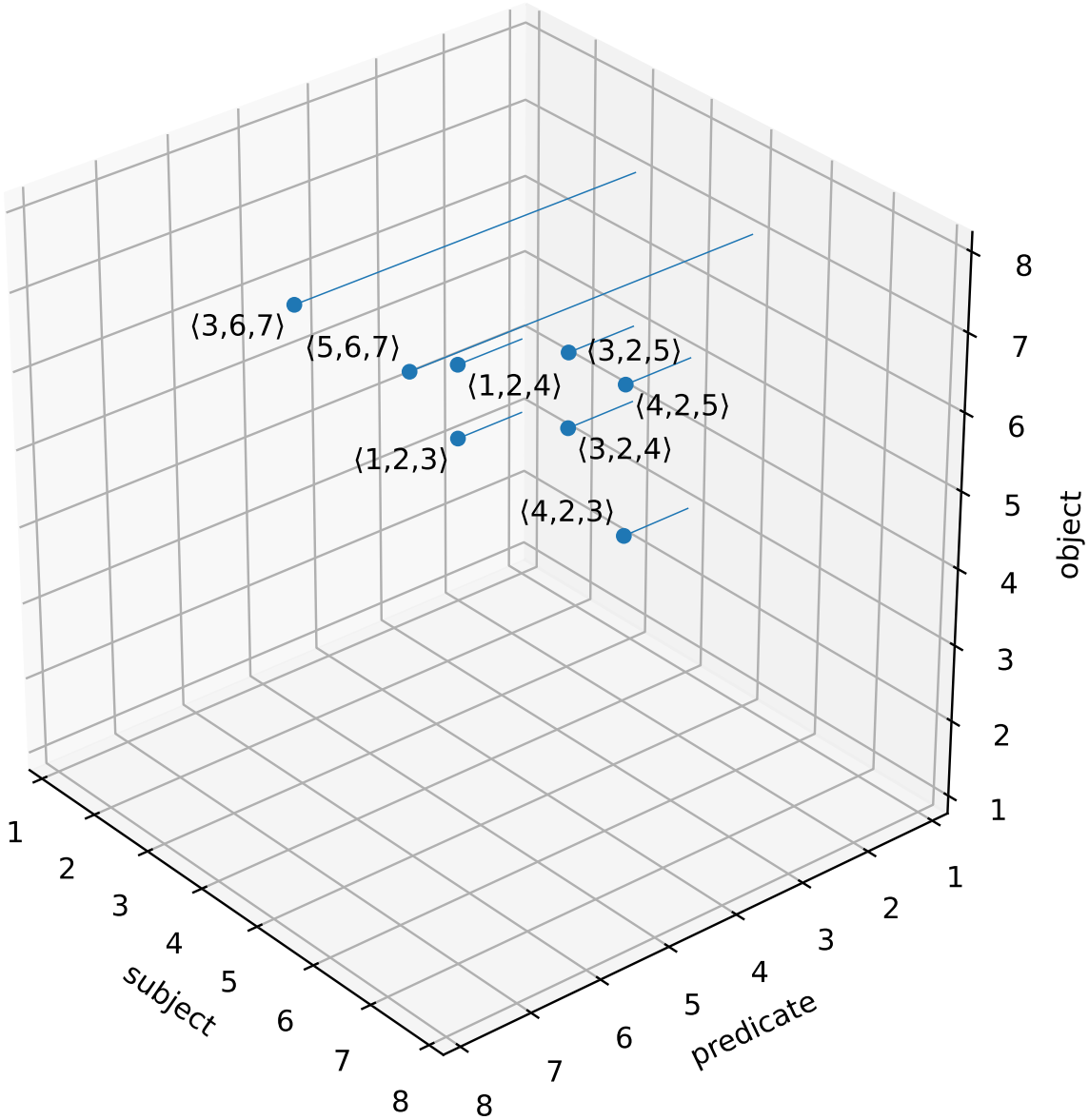
From DLs to SPARQL

- Assume closed world and fully materialized knowledge graph
- Retrieval can be realized via SPARQL queries
  [Bin et al., 2016, Karalis et al., 2024]

Class Expression	Graph Pattern $\mathfrak{p} = \tau(C_i, \texttt{?var})$
$A \in N_C$	`?var rdf:type A.`
$\neg C$	`{?var ?p ?o} UNION {?s ?p ?var}.` `FILTER NOT EXISTS` $\{\tau(C, \texttt{?var})\}$
$C_1 \sqcap \ldots \sqcap C_n$	$\{\tau(C_1, \texttt{?var}) \ldots \tau(C_n, \texttt{?var})\}$
$C_1 \sqcup \ldots \sqcup C_n$	$\{\tau(C_1, \texttt{?var})\}$ `UNION` $\ldots$ `UNION` $\{\tau(C_n, \texttt{?var})\}$
$\exists\, r.C$	$\{$`?var r ?s.` $\tau(C, \texttt{?s})\}$
$\forall\, r.C$	$\{$ *?var ?p ?o* . `FILTER NOT EXISTS` $\{$ *?var r ?s* . `FILTER NOT EXISTS` $\{\tau(C, \texttt{?s})\}\}\}$

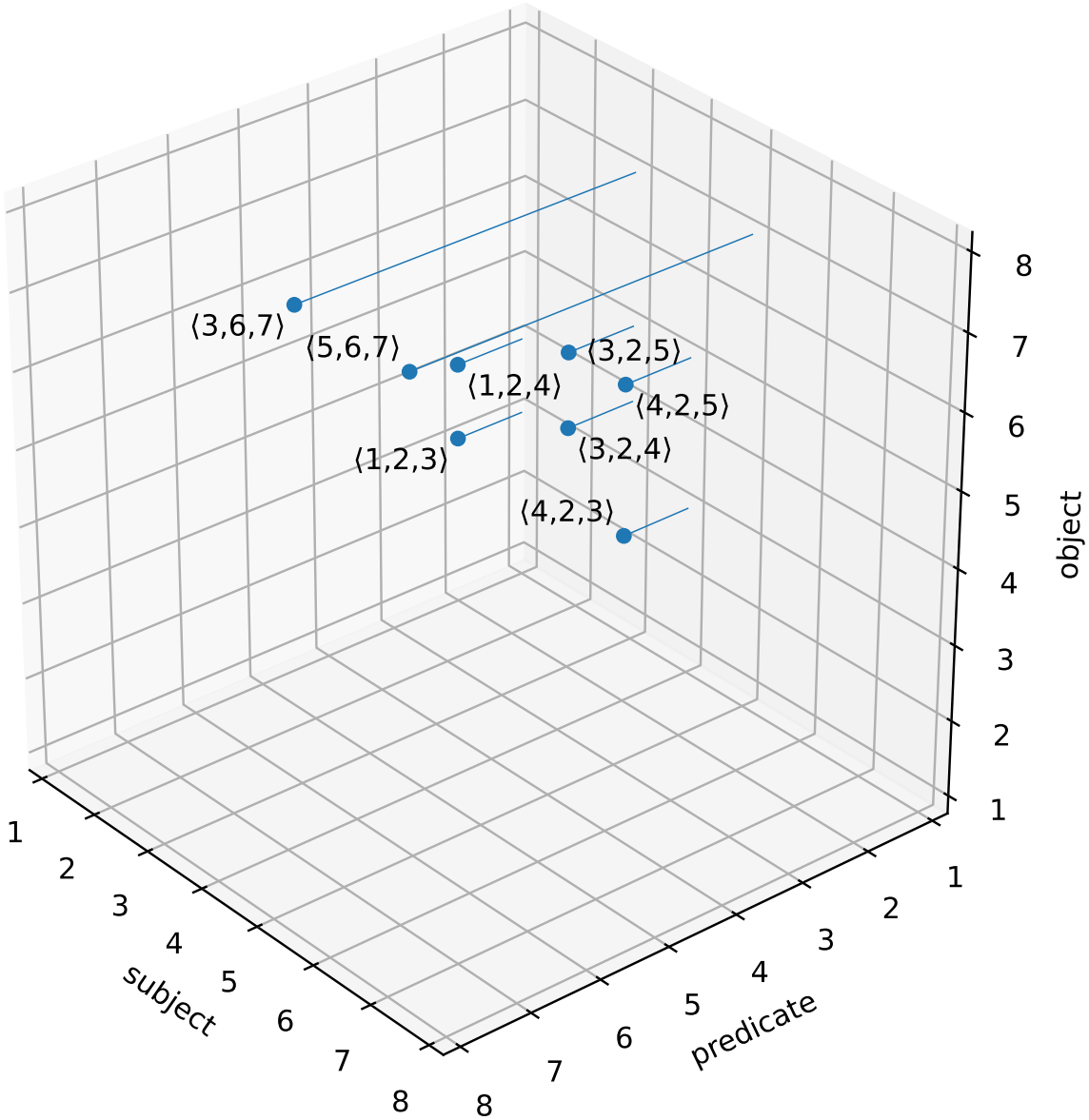
Retrieval

TENTRIS – Data Model



Retrieval

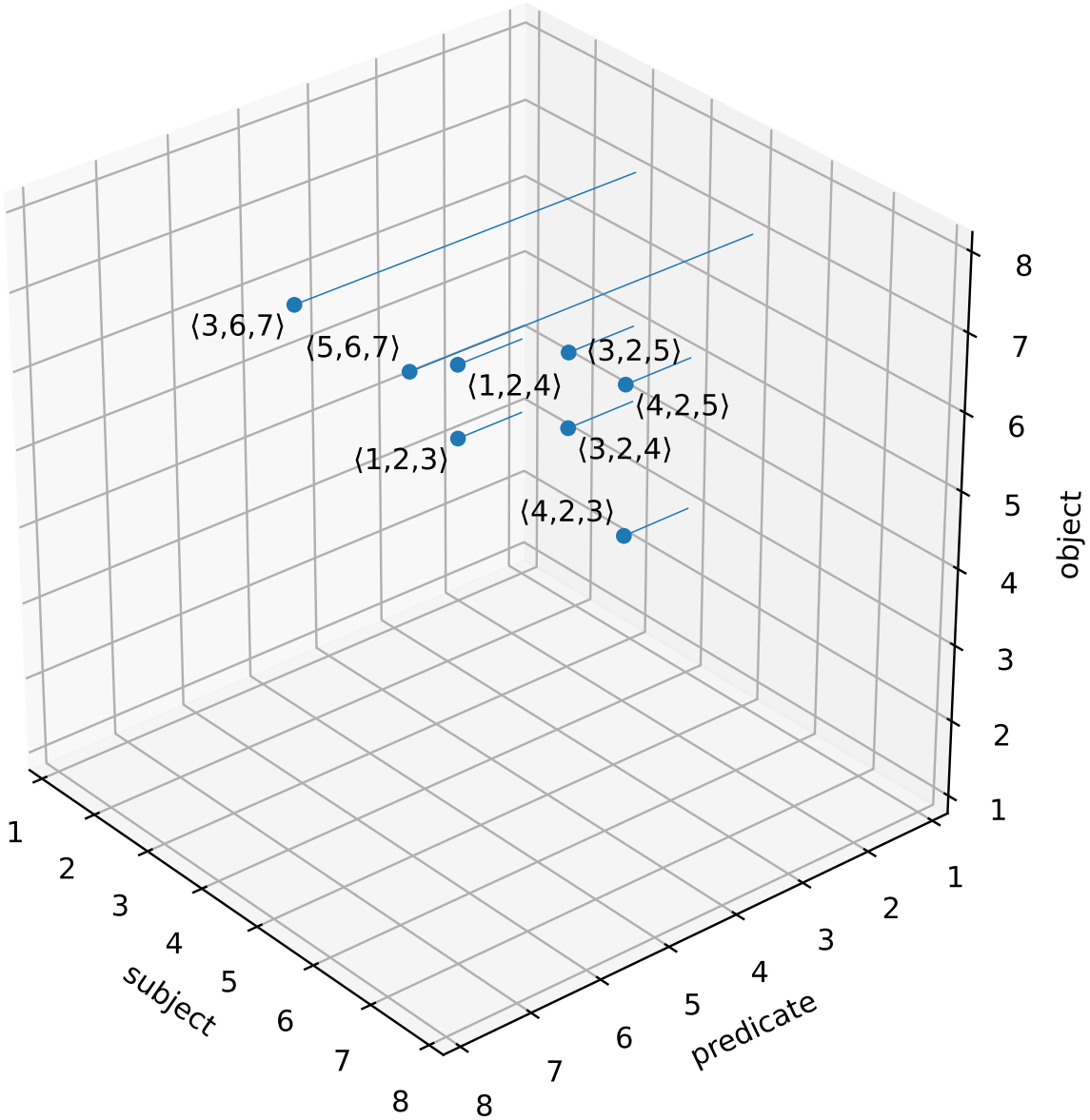
TENTRIS – Data Model



- $\mathbf{K} = \mathbb{N}^3$
- $V = \mathbb{B}$

Retrieval

TENTRIS – Data Model

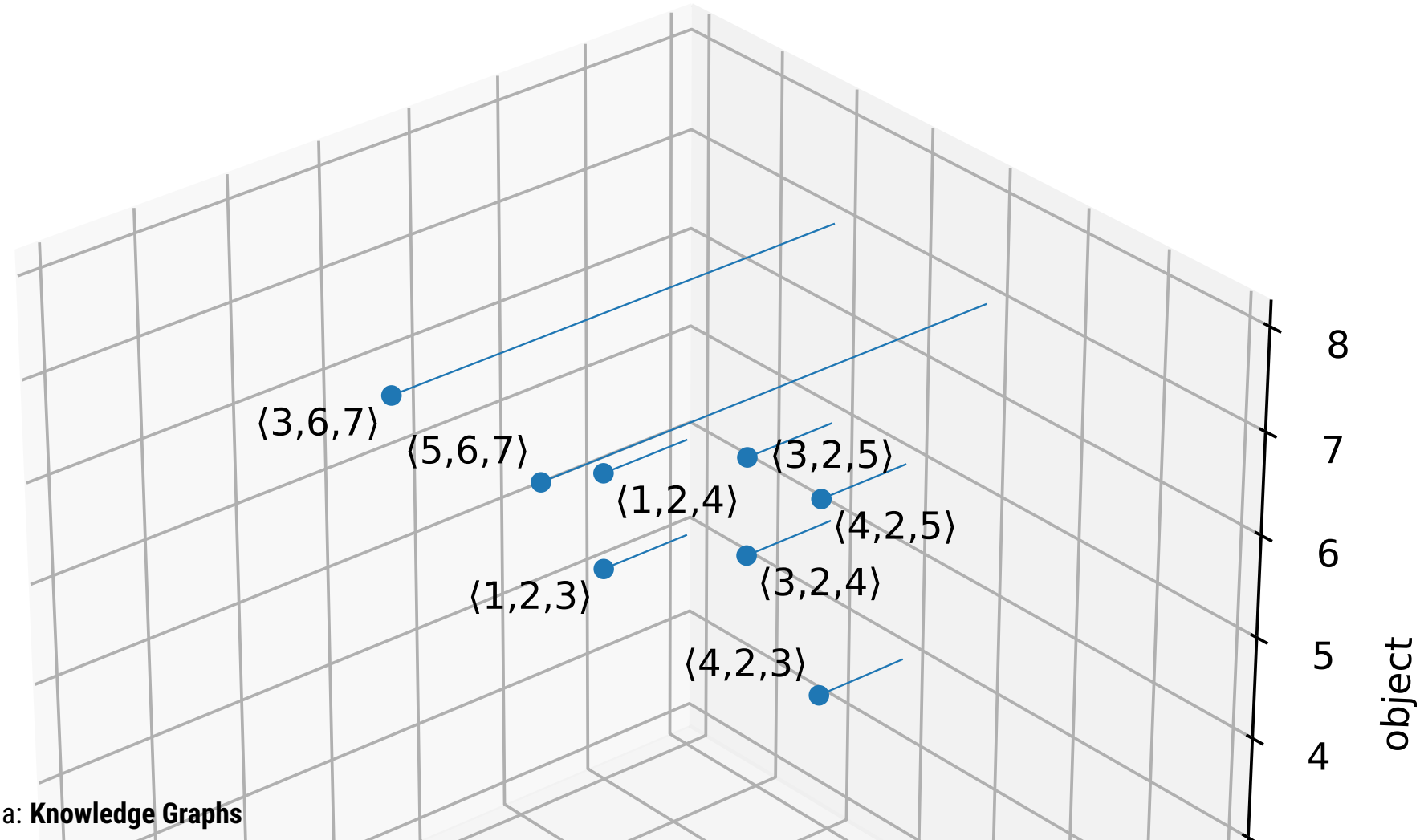


- $\mathbf{K} = \mathbb{N}^3$
- $V = \mathbb{B}$
- $T[\langle 3, 6, 7\rangle] = 1$
- $T[\langle 3, 6, 3\rangle] = 0$

Retrieval

TENTRIS– Data Model

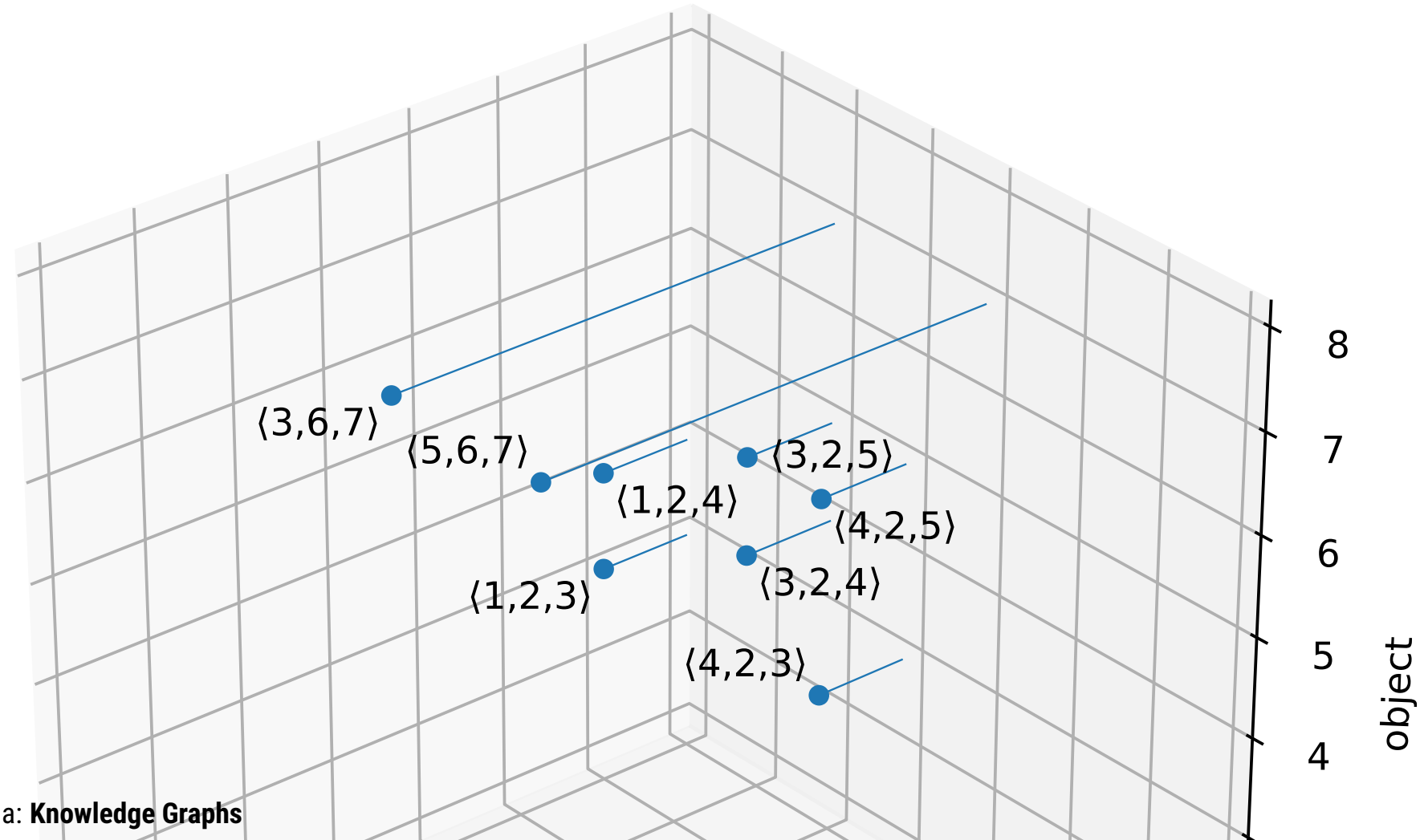
- Slicing selects portion of T, e.g., $T^{(1)} := T[1, 2, :]$ is order-1 tensor



Retrieval

TENTRIS– Data Model

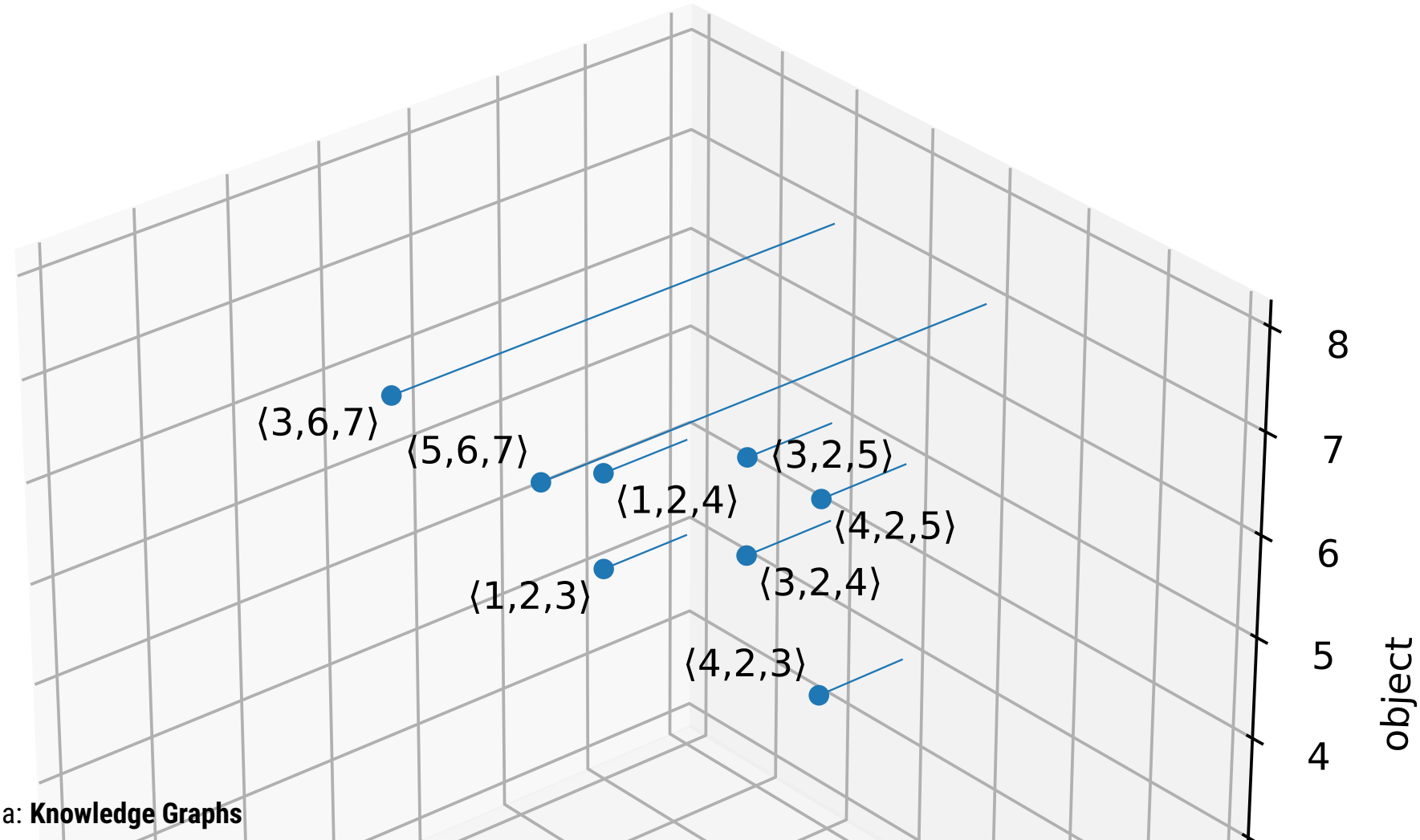
- Slicing selects portion of T, e.g., $T^{(1)} := T[1, 2, :]$ is order-1 tensor
- For our example, $T[1, 2, :] = [0, 0, 1, 1, 0, 0, 0, 0]$



Retrieval

TENTRIS– Data Model

- Slicing selects portion of T, e.g., $T^{(1)} := T[1, 2, :]$ is order-1 tensor
- For our example, $T[1, 2, :] = [0, 0, 1, 1, 0, 0, 0, 0]$
- Slices can be joined via Einstein summation [Barr, 1989]



Retrieval

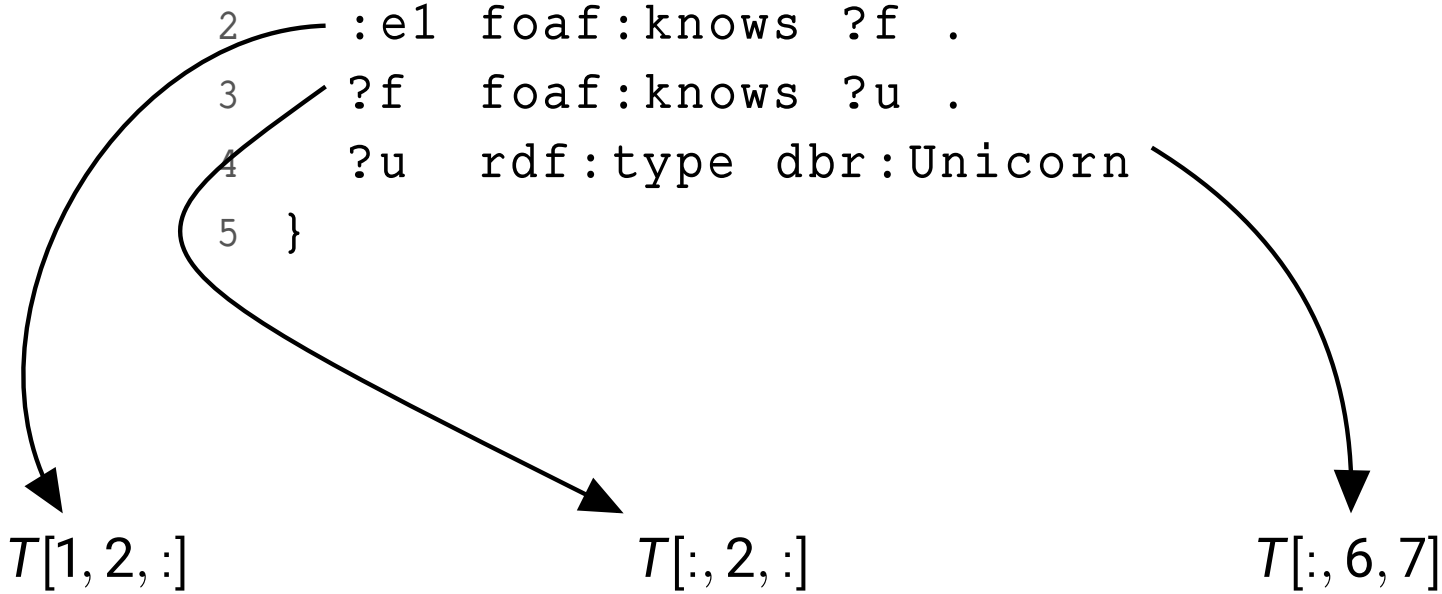
TENTRIS – Einstein Summation

```
1 SELECT ?f WHERE {
2   :e1 foaf:knows ?f .
3   ?f  foaf:knows ?u .
4   ?u  rdf:type dbr:Unicorn
5 }
```

Retrieval

TENTRIS – Einstein Summation

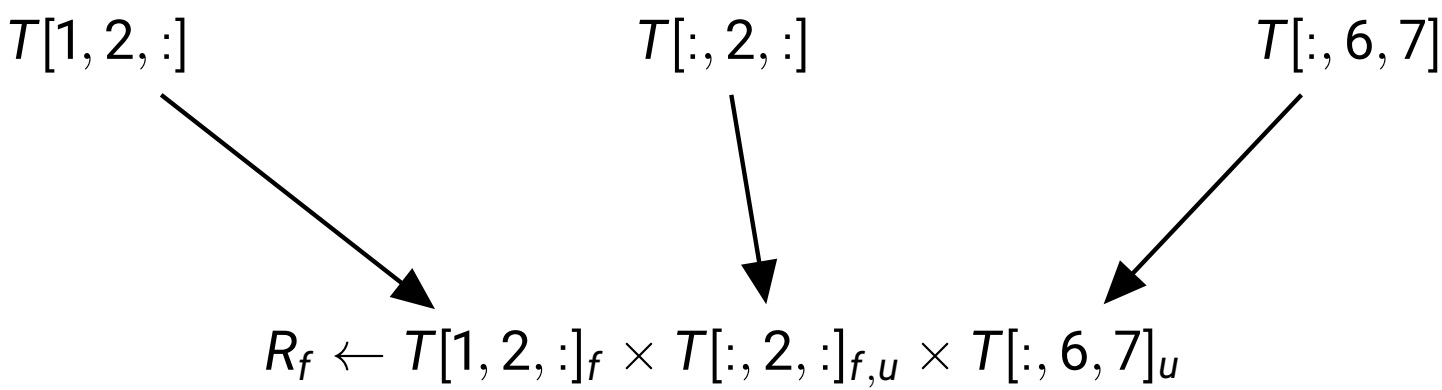
```
1 SELECT ?f WHERE {
2   :e1 foaf:knows ?f .
3   ?f  foaf:knows ?u .
4   ?u  rdf:type dbr:Unicorn
5 }
```



Retrieval

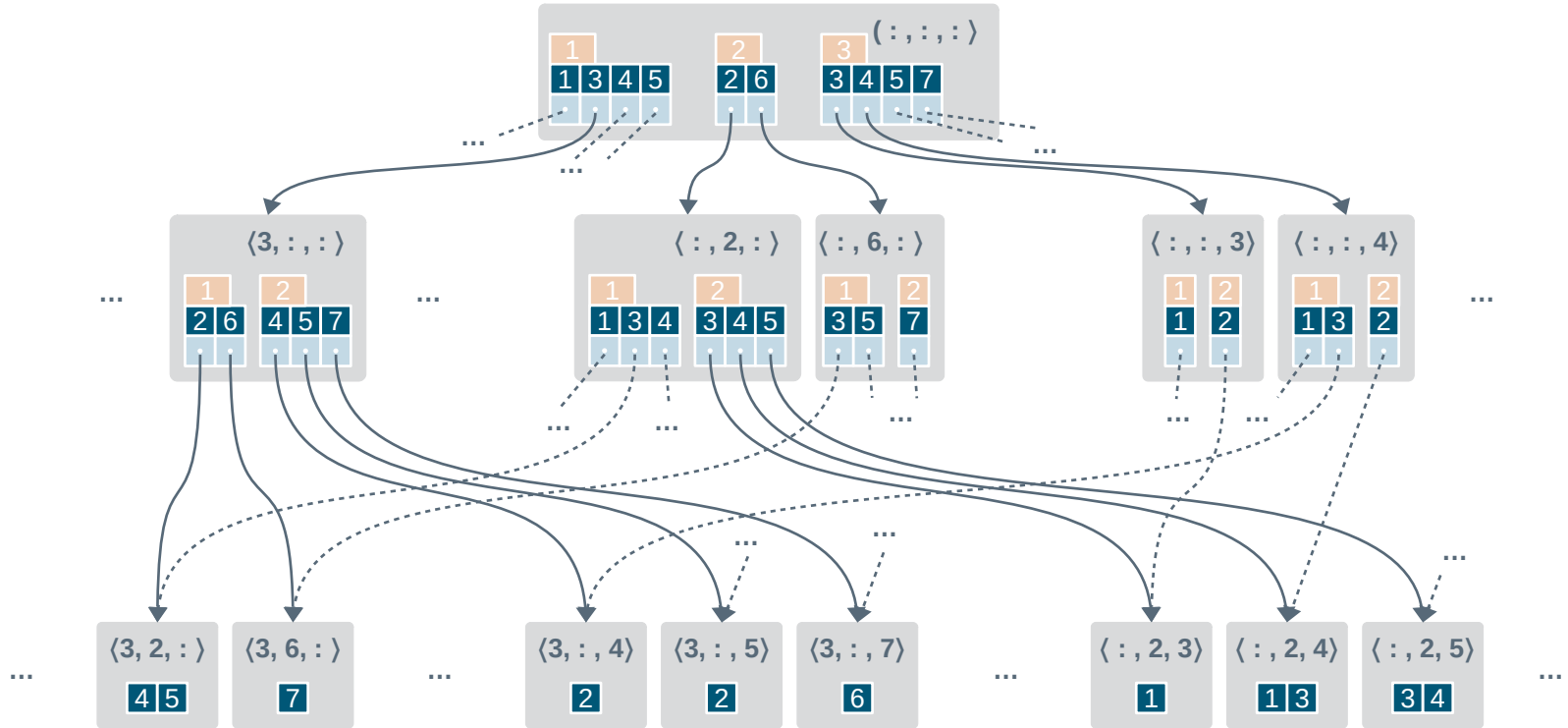
TENTRIS – Einstein Summation

```
1 SELECT ?f WHERE {
2    :e1  foaf:knows  ?f .
3    ?f   foaf:knows  ?u .
4    ?u   rdf:type dbr:Unicorn
5 }
```



Retrieval

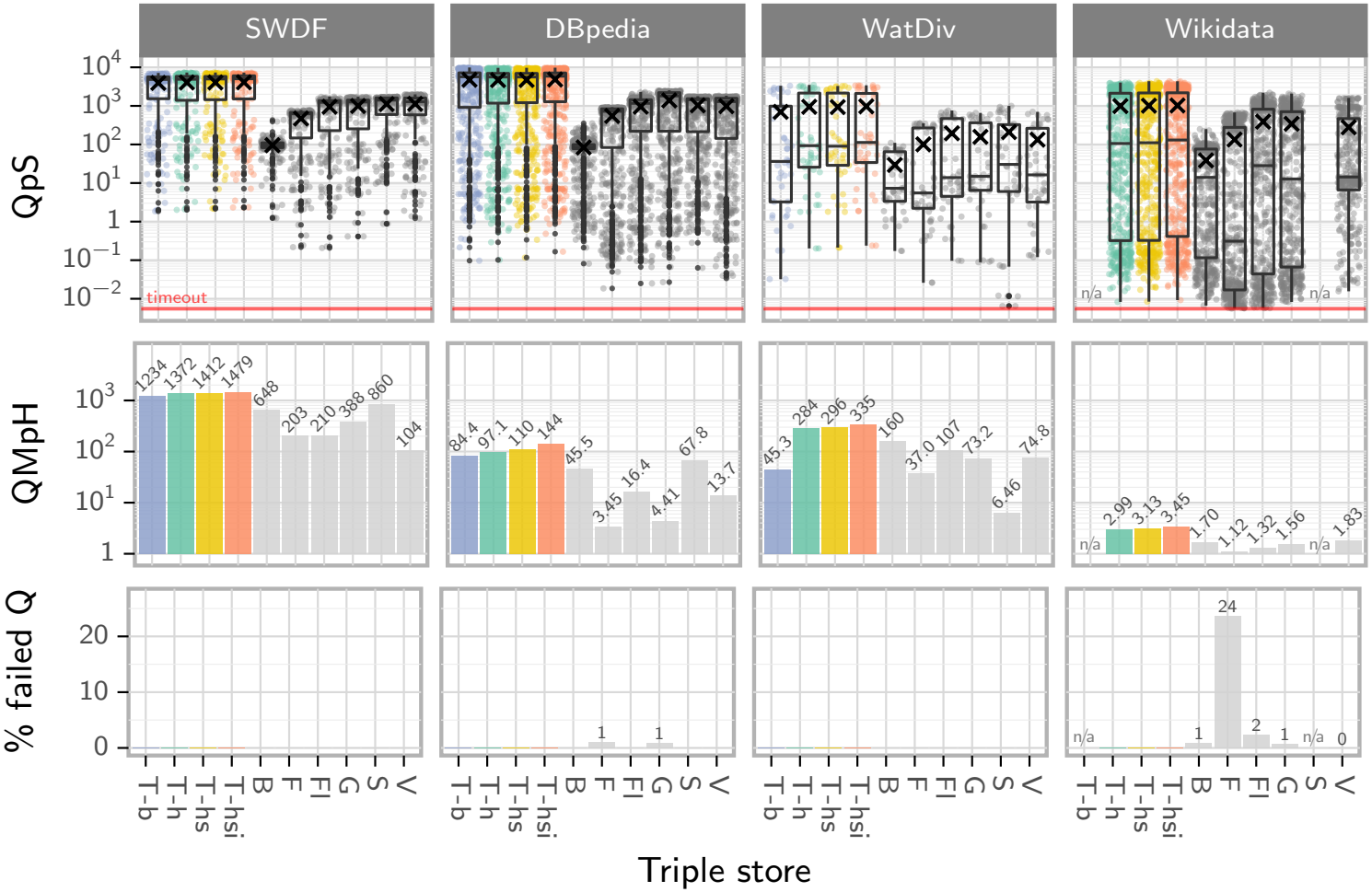
TENTRIS – Hypertrie



- Query for any tensor slice efficiently
- Storage bound is reduced from $\mathcal{O}(d! \cdot d \cdot z(h))$ for all collation orders to $\mathcal{O}(2^{d-1} \cdot d \cdot z(h))$
- Compression schemes on hypertries [Bigerl et al., 2022]

Retrieval

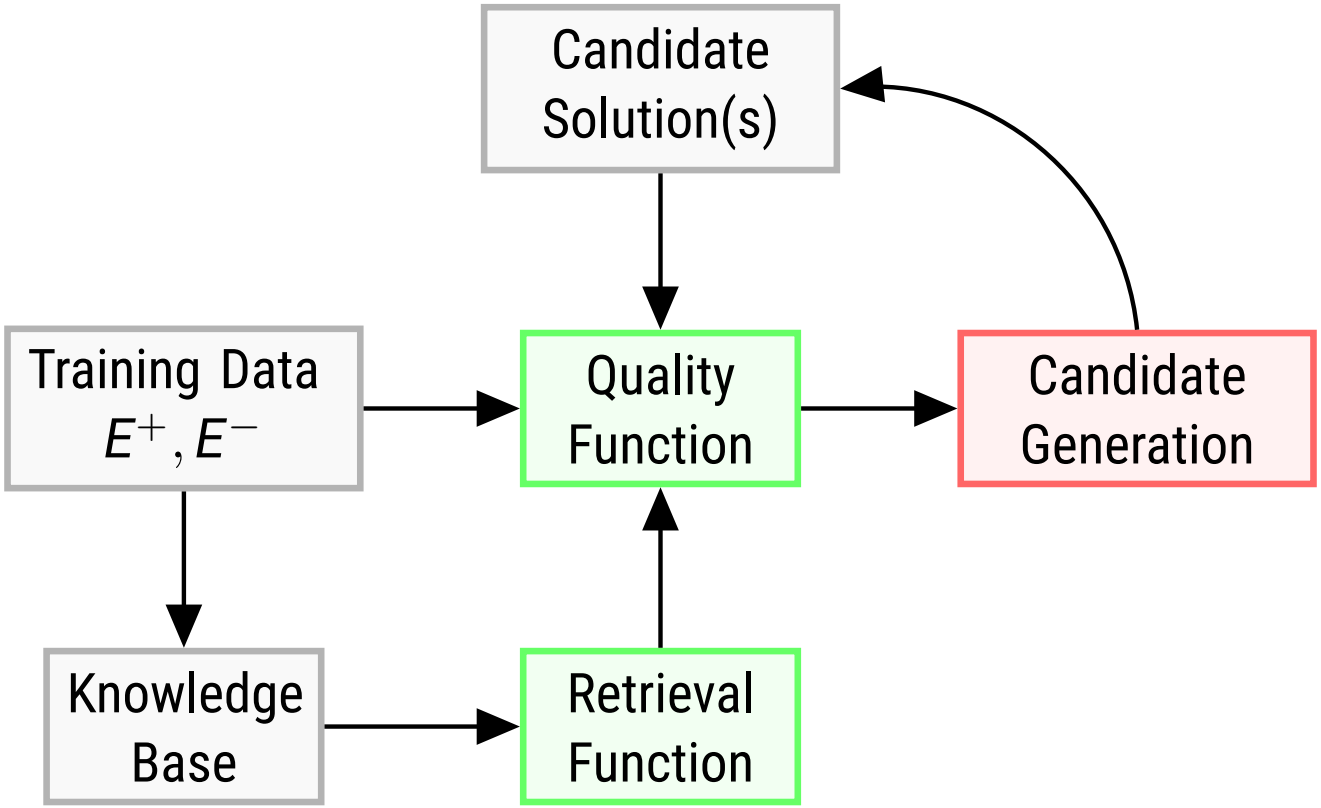
TENTRIS – Compressed Hypertrie



- Better runtimes on all datasets [Bigerl et al., 2022]
- Support updates on very large datasets [Bigerl et al., 2025]
- Native reasoner support upcoming

Learning problem

Challenges



✓ Retrieval is expensive ⇒ Exploit SPARQL

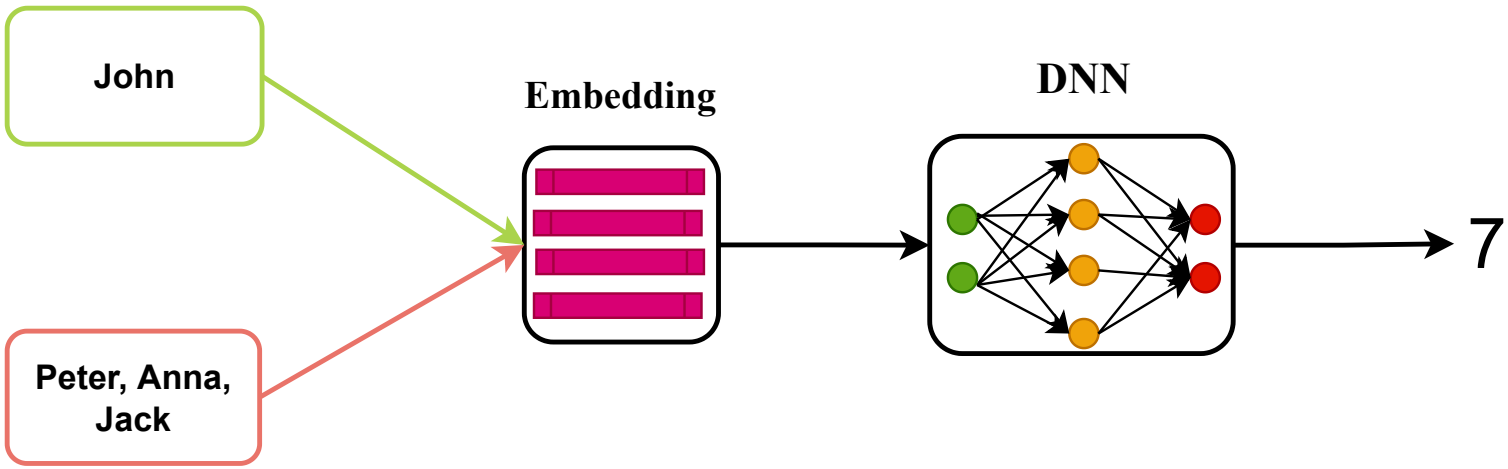
✓ Quality functions are often myopic ⇒ Exploit embeddings

Section 6

Concept Synthesis

Concept Synthesis

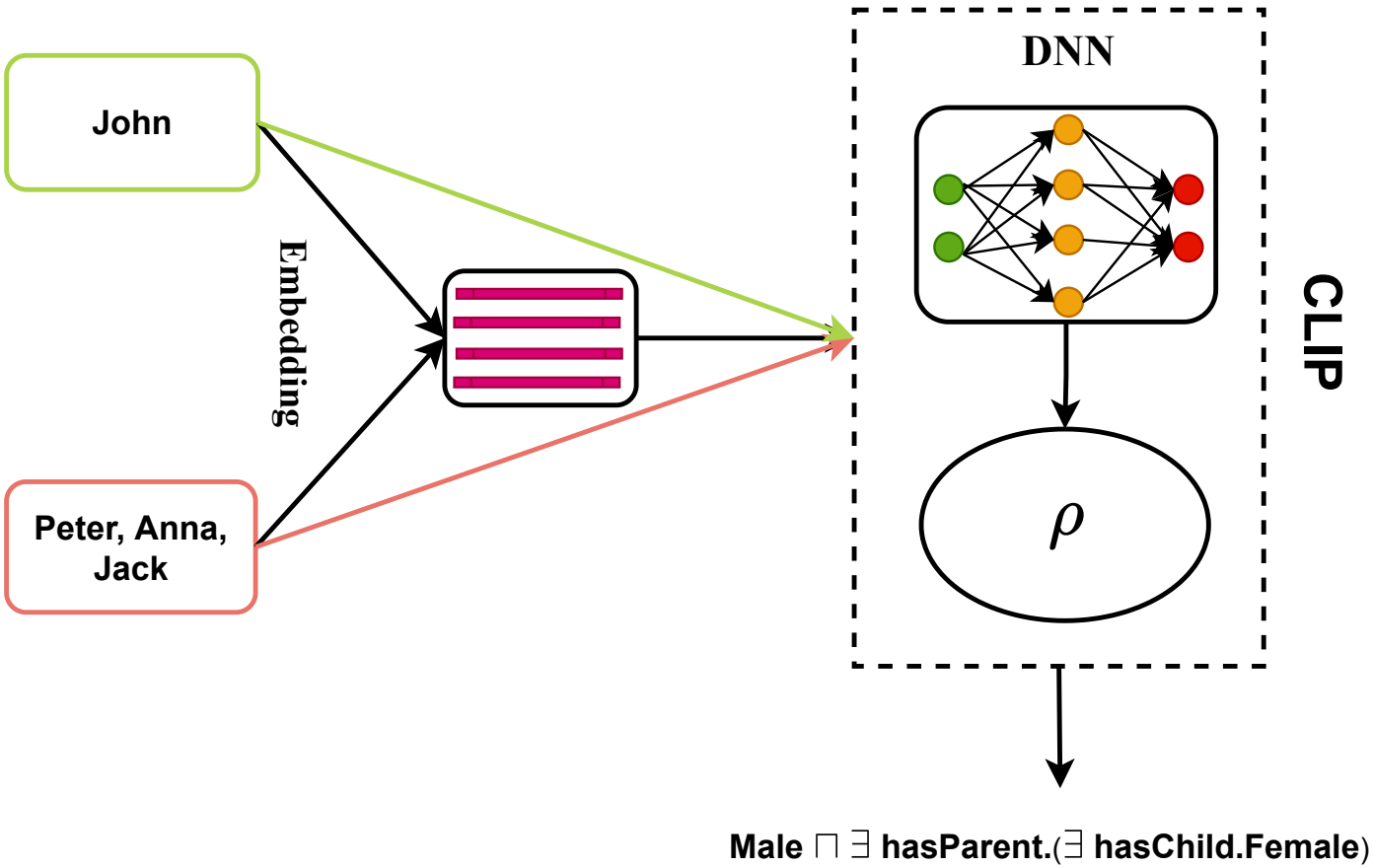
Concept Length Prediction



- Input: positive and negative examples
- Output: length of the target concept
- Implemented as classification problem

Concept Synthesis

Concept Learning



Concept Synthesis

Network Architecture

	Carcinogenesis					Mutagenesis				
Metric	LSTM	GRU	CNN	MLP	RM	LSTM	GRU	CNN	MLP	RM
Train. Acc.	0.89	0.96	0.97	0.80	0.48	0.83	0.97	0.98	0.68	0.33
Val. Acc.	0.76	0.93	0.82	0.77	0.48	0.70	0.82	0.71	0.65	0.35
Test Acc.	0.92	**0.95**	0.84	0.80	0.49	0.78	**0.85**	0.70	0.68	0.33
Test F1	0.88	**0.92**	0.71	0.59	0.33	0.76	**0.85**	0.70	0.67	0.32

	Semantic Bible					Vicodi				
Metric	LSTM	GRU	CNN	MLP	RM	LSTM	GRU	CNN	MLP	RM
Train. Acc.	0.85	0.93	0.99	0.68	0.33	0.73	0.81	0.83	0.66	0.28
Val. Acc.	0.49	0.58	0.44	0.46	0.26	0.55	0.77	0.70	0.64	0.30
Test Acc.	0.52	**0.53**	0.37	0.40	0.25	0.66	**0.80**	0.69	0.66	0.29
Test F1	0.27	**0.38**	0.20	0.22	0.16	0.45	**0.50**	0.45	0.38	0.20

Concept Synthesis

Comparison with SOTA

	Carcinogenesis			
Metric	**CELOE**	**OCEL**	**ELTL**	**CLIP**
Acc. ↑	0.78 ± 0.27	0.89 ± 0.31	0.58 ± 0.46	0.99 ± 0.00
F1 ↑	0.62 ± 0.46	–	0.51 ± 0.47	$0.96* \pm 0.10$
Runtime (min) ↓	0.93 ± 0.94	3.01 ± 0.72	0.75 ± 0.07	$0.10* \pm 0.09$
Length ↓	1.69 ± 0.89	7.81 ± 6.88	1.04 ± 0.39	2.00 ± 1.28

	Mutagenesis			
Metric	**CELOE**	**OCEL**	**ELTL**	**CLIP**
Acc. ↑	0.99 ± 0.00	0.71 ± 0.45	0.37 ± 0.43	0.99 ± 0.00
F1 ↑	0.81 ± 0.35	–	0.29 ± 0.40	$0.93* \pm 0.18$
Runtime (min) ↓	0.70 ± 0.77	2.39 ± 0.18	0.29 ± 0.16	$0.07* \pm 0.05$
Length ↓	2.79 ± 1.17	12.63 ± 7.03	1.10 ± 0.81	2.20 ± 1.16

Concept Synthesis

Comparison with SOTA

	Semantic Bible			
Metric	**CELOE**	**OCEL**	**ELTL**	**CLIP**
Acc. ↑	0.99 ± 0.02	0.66 ± 0.47	0.59 ± 0.37	0.99 ± 0.00
F1 ↑	0.97 ± 0.10	–	0.57 ± 0.38	0.98 ± 0.05
Runtime (min) ↓	0.47 ± 0.80	22.15 ± 96.55	0.09 ± 0.07	$0.06* \pm 0.05$
Length ↓	3.85 ± 2.44	9.54 ± 5.73	1.38 ± 1.76	$2.52* \pm 1.26$

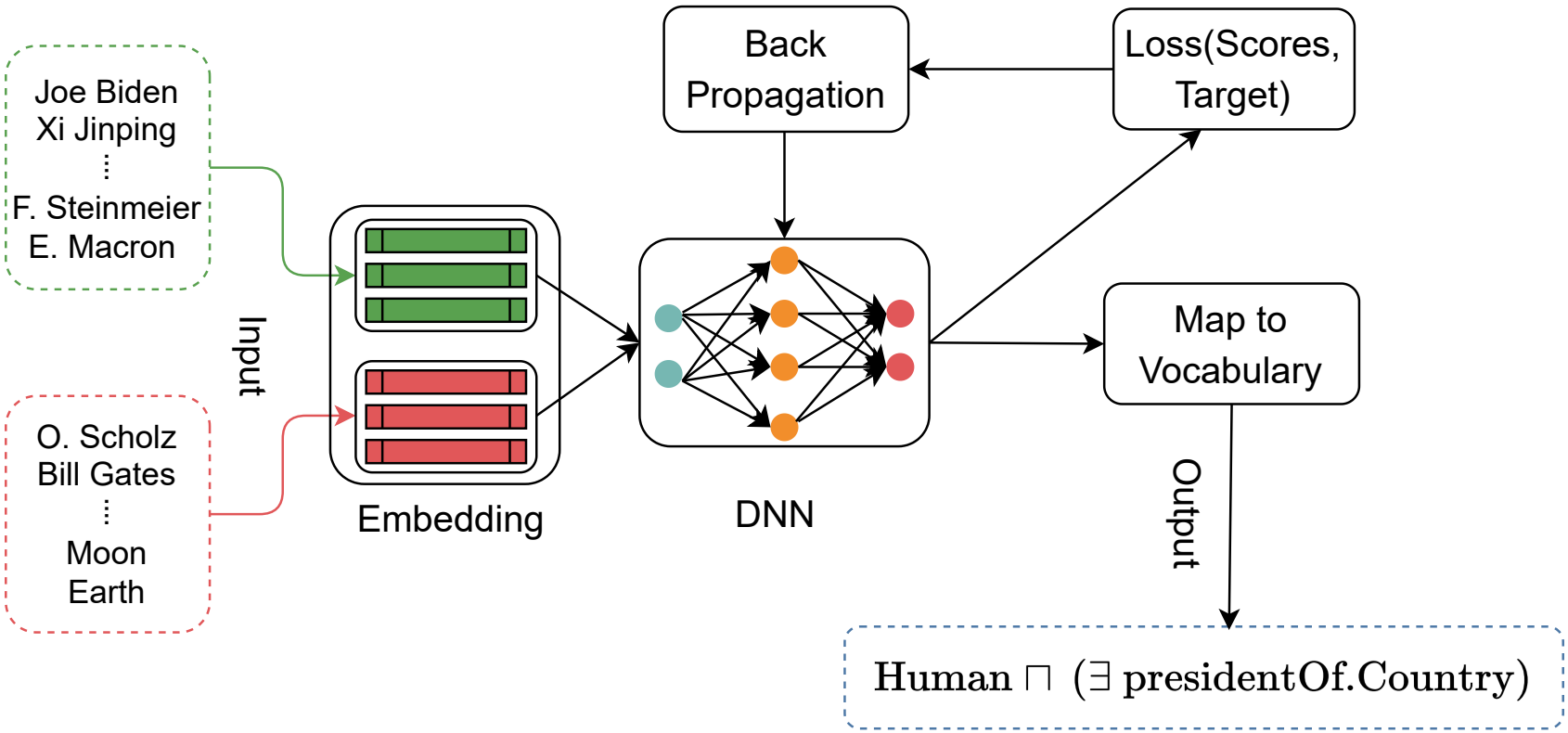
	Vicodi			
Metric	**CELOE**	**OCEL**	**ELTL**	**CLIP**
Acc. ↑	0.29 ± 0.44	0.25 ± 0.43	0.28 ± 0.44	$0.99* \pm 0.00$
F1 ↑	0.25 ± 0.44	–	0.25 ± 0.44	$0.97* \pm 0.09$
Runtime (min) ↓	1.30 ± 0.71	4.78 ± 1.12	1.81 ± 0.46	$0.16* \pm 0.12$
Length ↓	10.79 ± 6.30	11.54 ± 6.00	11.14 ± 6.11	$1.68* \pm 0.98$

- Same or better accuracy and F-measure
- 1 order of magnitude faster
- Mostly shorter concepts

Concept Synthesis

Neural Concept Synthesis

- Rethink class expression learning as machine translation [Kouagou et al., 2023a, Kouagou et al., 2023b]
- Predict class expression directly



Concept Synthesis

Evaluation

	F_1 (%) ↑			
	Carcinogenesis	Mutagenesis	Semantic Bible	Vicodi
CELOE	37.92±44.25	82.95±33.48	**93.18±17.52***	35.66±42.06
ELTL	13.35±25.84	28.81±34.44	42.77±38.46	16.70±33.31
ECII	15.74±27.82	27.14±31.64	33.97±37.71	43.66±36.13
EvoLearner	91.48±14.30	**93.27±12.95**	91.88±10.14	92.74±10.28
$NCES_{LSTM}$	82.21±29.29	81.47±27.77	72.32±34.37	72.35±35.34
$NCES_{GRU}$	89.51±25.24	78.24±30.99	52.37±39.86	86.60±26.84
$NCES_{ST}$	90.32±25.12	80.55±34.33	72.95±38.68	77.75±37.16
$NCES_{ST+GRU}$	96.77±12.72	89.50±26.09	84.32±26.82	92.91±20.14
$NCES_{ST+LSTM}$	96.72±12.96	89.19±26.16	80.78±28.30	90.91±21.57
$NCES_{GRU+LSTM}$	94.51±15.39	81.49±30.71	75.91±32.48	88.28±24.17
$NCES_{ST+GRU+LSTM}$	**97.06±13.06***	91.39±22.91	87.11±24.05	**95.51±12.14***

Concept Synthesis

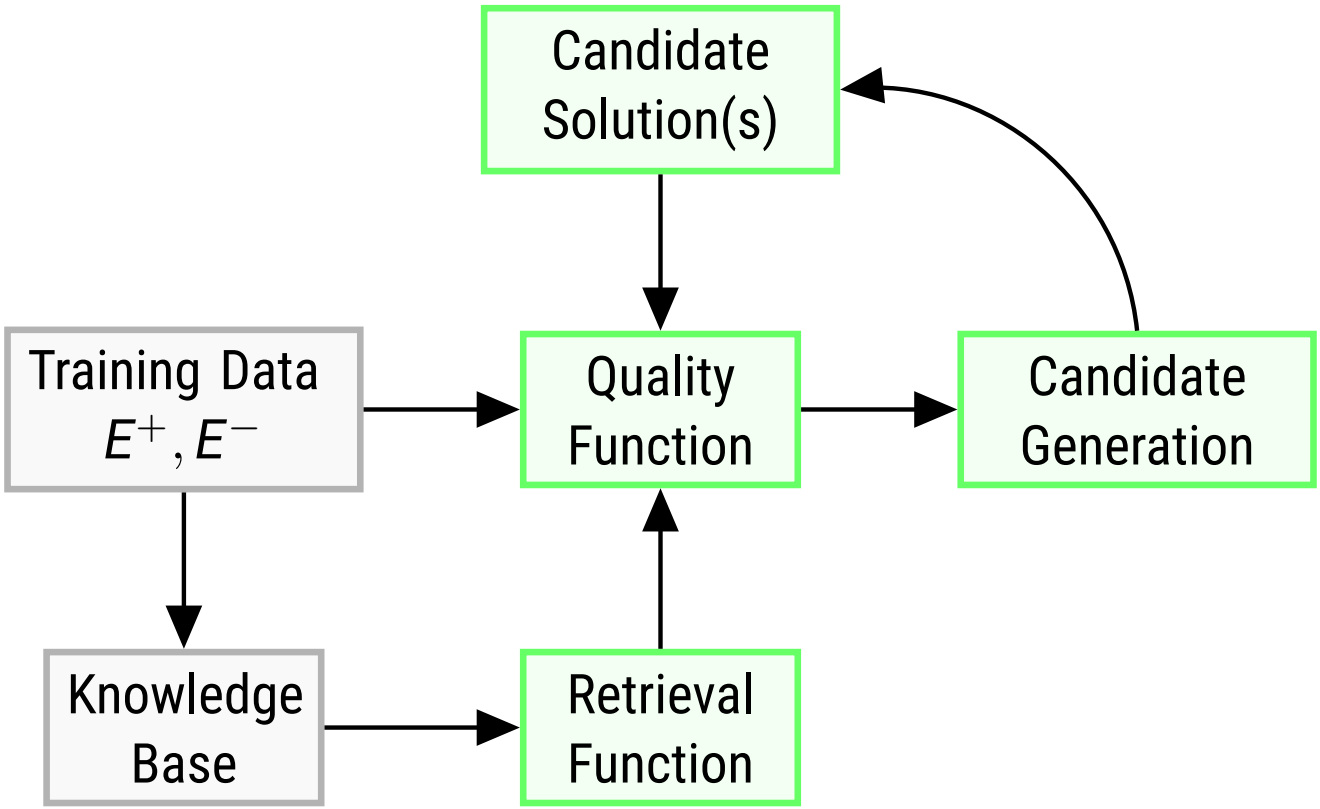
Runtime

- Runtime improvement by 2 orders of magnitude
- Better performance with large T-boxes
- Web-ready

	Runtime (sec.) ↓			
	Carcinogenesis	Mutagenesis	Semantic Bible	Vicodi
CELOE	239.58±132.59	92.46±125.69	135.30±139.95	289.95±103.63
ELTL	23.81±1.47	15.19±12.50	4.12±0.11	299.14±202.21
ECII	22.93±2.63	18.11±4.93	6.45±1.42	37.94±28.25
EvoLearner	54.73±25.86	48.00±31.38	17.16±9.20	213.78±81.03
$NCES_{LSTM}$	0.16±0.00	0.19±0.00	0.08±0.00	0.13±0.00
$NCES_{GRU}$	0.15±0.00	0.18±0.00	0.08±0.00	0.06±0.00
$NCES_{ST}$	**0.08±0.00***	**0.11±0.00***	**0.07±0.00***	**0.04±0.00***
$NCES_{ST+GRU}$	0.16±0.00	0.25±0.00	0.11±0.00	0.09±0.00
$NCES_{ST+LSTM}$	0.23±0.00	0.23±0.00	0.11±0.00	0.11±0.00
$NCES_{GRU+LSTM}$	0.24±0.00	0.32±0.00	0.13±0.00	0.17±0.00
$NCES_{ST+GRU+LSTM}$	0.27±0.00	0.31±0.00	0.15±0.00	0.15±0.00

Learning problem

Challenges



- ✓ Retrieval is expensive ⇒ Exploit SPARQL
- ✓ Quality functions are often myopic ⇒ Exploit embeddings

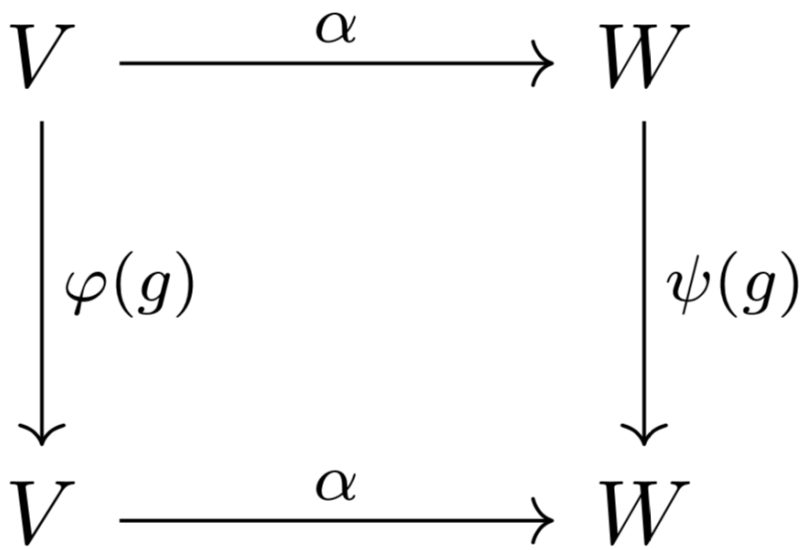
Section 7

Summary

Summary

Open Questions

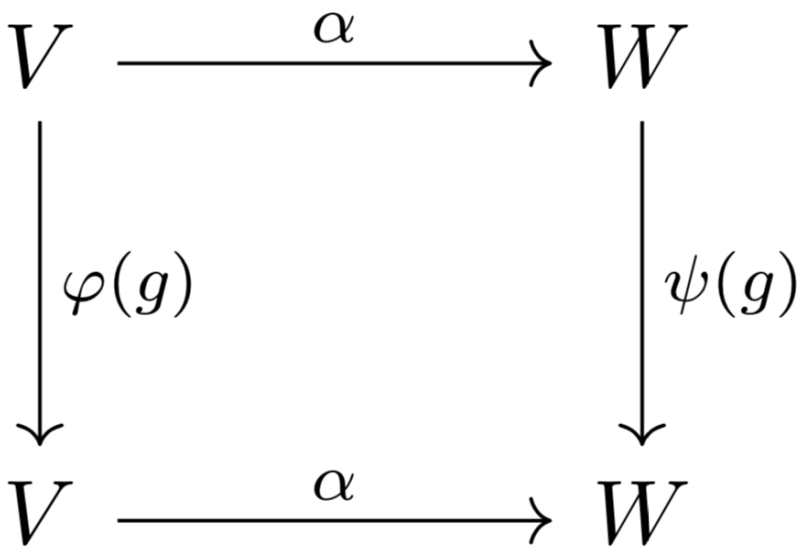
- Switching representation is an old trick that still works



Summary

Open Questions

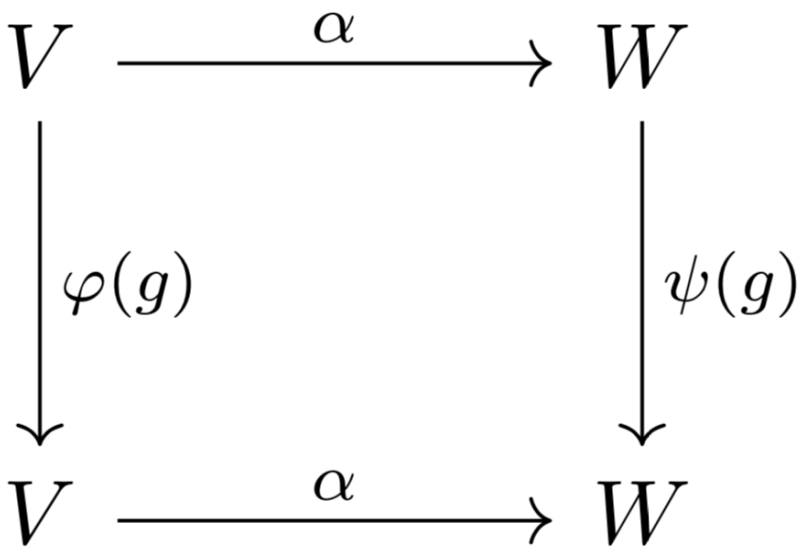
- Switching representation is an old trick that still works
- Selecting the right representation leads to improvement in runtimes and F-measure



Summary

Open Questions

- Switching representation is an old trick that still works
- Selecting the right representation leads to improvement in runtimes and F-measure
- How can we detect the right representation at the right time?



Summary

Thank You!

Joint works with Alexander Bigerl, Caglar Demir, Hamada Zahera, Louis Kamdem, N'Dah Jean Kouagou, Nikoloas Karalis, Stefan Heindorf, Mohamed Sherif, Muhammed Saleem, and many more

Thank You!
Questions?

- https://dice-research.org
- https://www.linkedin.com/company/dice-research
- https://www.linkedin.com/in/axel-ngonga

SAIL
SUSTAINABLE LIFE-CYCLE OF INTELLIGENT SOCIO-TECHNICAL SYSTEMS

LEMUR

KNOW GRAPHS

enexa

TRR 318 CONSTRUCTING EXPLAINABILITY

References I

[Barr, 1989] Barr, A. H. (1989).
The einstein summation notation: Introduction and extensions.
SIGGRAPH 89 Course notes# 30 on Topics in Physically-Based Modeling, pages J1–J12.

[Bigerl et al., 2022] Bigerl, A., Conrads, L., Behning, C., Saleem, M., and Ngonga Ngomo, A.-C. (2022).
Hashing the hypertrie: Space-and time-efficient indexing for sparql in tensors.
In *International Semantic Web Conference*, pages 57–73. Springer.

References II

[Bigerl et al., 2025] Bigerl, A., Karalis, N., Heidrich, L., and Ngomo, A. N. (2025).
Efficient updates for worst-case optimal join triple stores.
In Garijo, D., Kirrane, S., Salatino, A. A., Shimizu, C., Acosta, M., Nuzzolese, A. G., Ferrada, S., Soulard, T., Kozaki, K., Takeda, H., and Gentile, A. L., editors, *The Semantic Web - ISWC 2025 - 24th International Semantic Web Conference, Nara, Japan, November 2-6, 2025, Proceedings, Part I*, volume 16140 of *Lecture Notes in Computer Science*, pages 539–555. Springer.

[Bin et al., 2016] Bin, S., Bühmann, L., Lehmann, J., and Ngonga Ngomo, A.-C. (2016).
Towards sparql-based induction for large-scale rdf data sets.
In *ECAI 2016*, pages 1551–1552. IOS Press.

References III

[Demir and Ngonga Ngomo, 2023] Demir, C. and Ngonga Ngomo, A.-C. (2023).
Neuro-symbolic class expression learning.
In *Proceedings of the International Joint Conference on Artificial Intelligence*.

[Karalis et al., 2024] Karalis, N., Bigerl, A., Demir, C., Heidrich, L., and Ngomo, A. N. (2024).
Evaluating negation with multi-way joins accelerates class expression learning.
In Bifet, A., Davis, J., Krilavicius, T., Kull, M., Ntoutsi, E., and Zliobaite, I., editors, *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2024, Vilnius, Lithuania, September 9-13, 2024, Proceedings, Part VI*, volume 14946 of *Lecture Notes in Computer Science*, pages 199–216. Springer.

References IV

[Kouagou et al., 2023a] Kouagou, N. J., Heindorf, S., Demir, C., and Ngomo, A. N. (2023a).
Neural class expression synthesis.
In Pesquita, C., Jiménez-Ruiz, E., McCusker, J. P., Faria, D., Dragoni, M., Dimou, A., Troncy, R., and Hertling, S., editors, *The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings*, volume 13870 of *Lecture Notes in Computer Science*, pages 209–226. Springer.

[Kouagou et al., 2023b] Kouagou, N. J., Heindorf, S., Demir, C., and Ngomo, A. N. (2023b).
Neural Class Expression Synthesis in ALCHIQ(D).
In *European Conference on Machine Learning*.

References V

[Lehmann and Hitzler, 2010] Lehmann, J. and Hitzler, P. (2010).
Concept learning in description logics using refinement operators.
Machine Learning, 78(1):203–250.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015).
Human-level control through deep reinforcement learning.
nature, 518(7540):529–533.

[Teyou et al., 2024] Teyou, L. M. K., Demir, C., and Ngomo, A. N. (2024).
Embedding knowledge graphs in degenerate clifford algebras.
In Endriss, U., Melo, F. S., Bach, K., Diz, A. J. B., Alonso-Moral, J. M., Barro, S., and Heintz, F., editors, *ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of*

References VI

Intelligent Systems (PAIS 2024), volume 392 of *Frontiers in Artificial Intelligence and Applications*, pages 1293–1300. IOS Press.

[Trouillon et al., 2016] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016).
Complex embeddings for simple link prediction.
In *International conference on machine learning*, pages 2071–2080. PMLR.

[Yang et al., 2015] Yang, B., tau Yih, W., He, X., Gao, J., and Deng, L. (2015).
Embedding entities and relations for learning and inference in knowledge bases.