Relational Learning on Linked Building Data for Reinforcement Learning-based building control

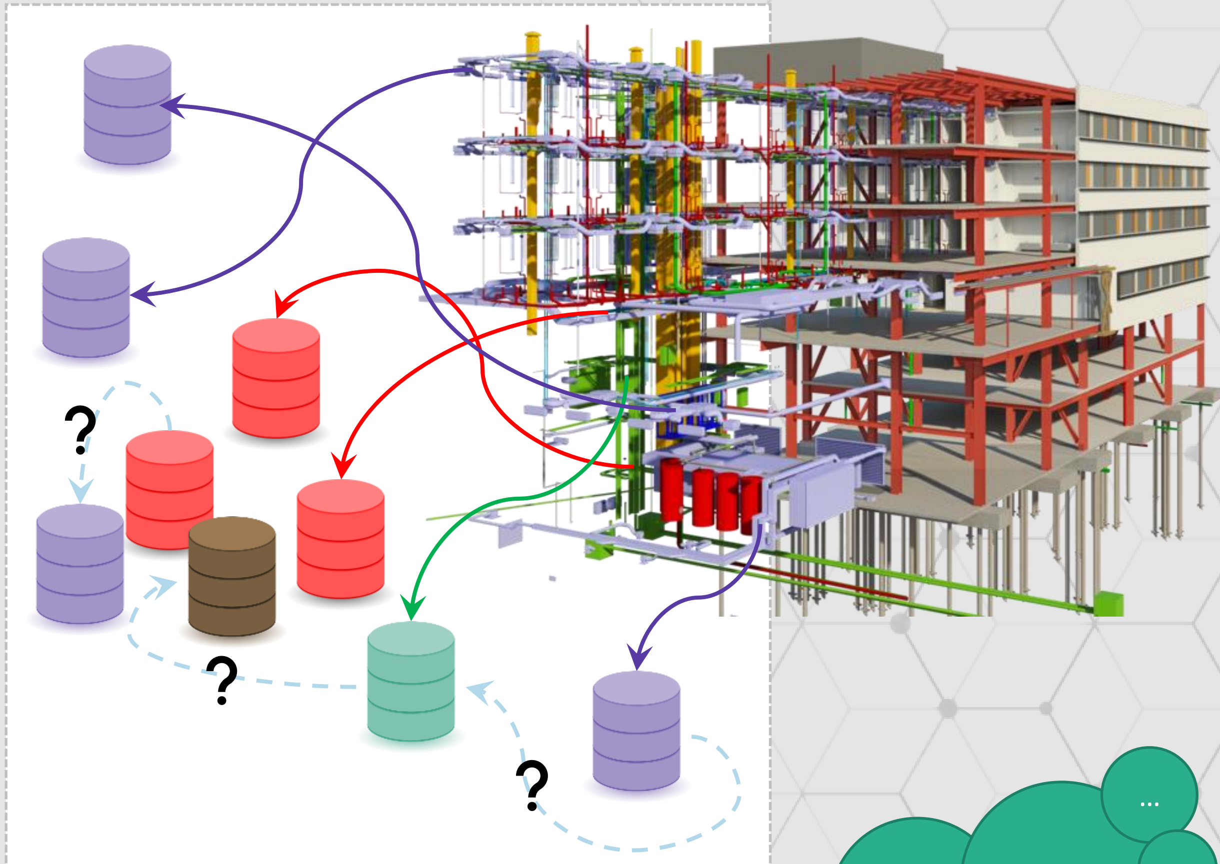
Kevin Luwemba Mugumya,
PhD Student, University of Nottingham Malaysia
keey4kly@nottingham.edu.my

Let’s break this down

Linked Building Data

Problem faced by today's buildings

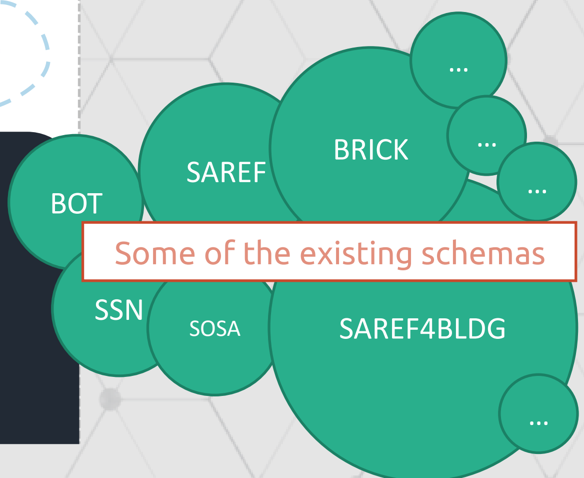
1. Buildings are becoming very complex.
2. Multiple operational building systems that struggle to talk to each other.
3. Hundreds of connected embedded devices (sensors, actuators).
4. Various metadata and labelling schemes for things.
5. Departmentalized data processing for downstream building automation tasks.



A recently promising solution

Utilizing standardized metadata schemas to

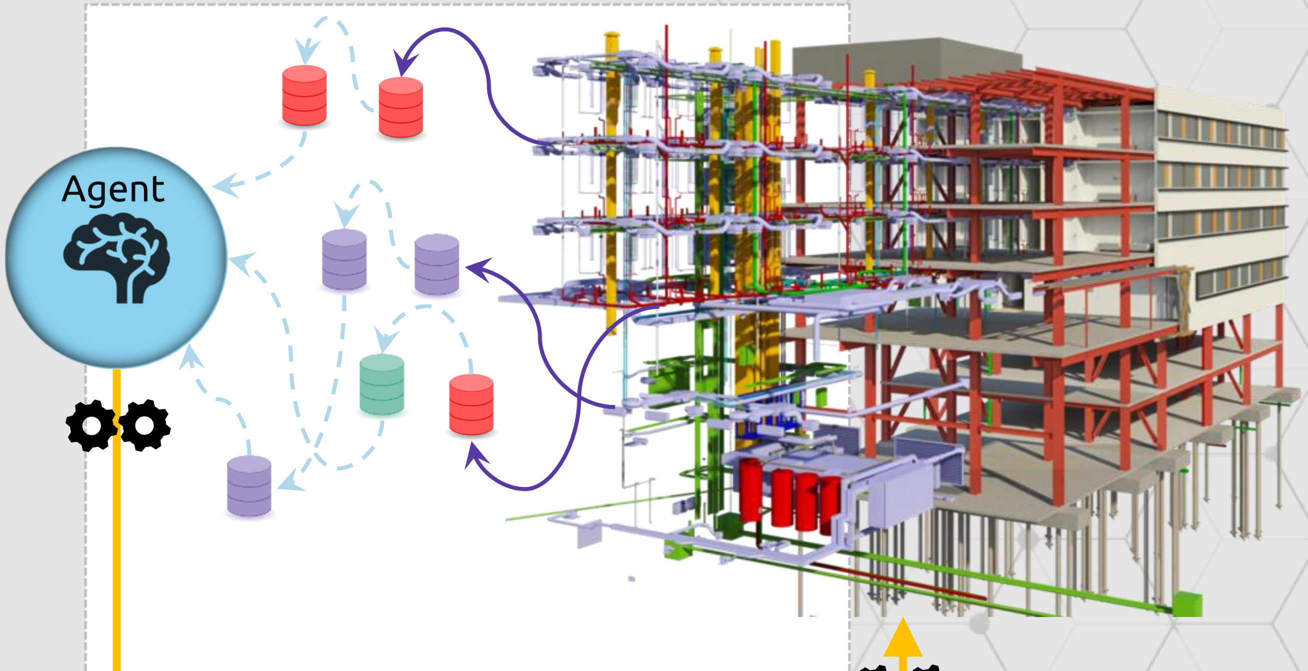
Link Building Data



Reinforcement Learning

An even BIGGER problem is faced by many current building control agents

1. They operate using siloed building information.
2. This inhibits their collective-reasoning capabilities.
3. Multi-agent collaborative building control is still inefficient.
4. They are heavily reliant on data and not so much on how the data inter-links.
5. They are not very adaptive



- Underutilized building control information
- Less performant control actions.

An existing PARTIAL solution

Utilizing adaptive and experience driven control agents powered by

Reinforcement Learning

Model Based RL

Existing RL architectures

Model free RL

Reinforcement Learning (in a nutshell)

University of Nottingham
UK | CHINA | MALAYSIA

(Experience-Driven)
Reinforcement Learning

Mathematical Modelling that mimics **sequential-decision making**.

In a nutshell;

The **Control Problem** is modelled as a Markov Decision Process;

5-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ of

$states \in \mathcal{S}$ $actions \in \mathcal{A}$ $dynamics \in \mathcal{P}$ $reward \in \mathcal{R}$ $discount\,factor \in \gamma$

Reinforcement Learning (MDP)

Markov Decision Process of an RL agent

5-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

$states \in \mathcal{S}$

- Indoor Temp
- Humidity level
- Occupancy
- Heat index
- Comfort Index
- PV storage level
- Outdoor Temp
- Grid Demand cycles

OLD STATE

$dynamics \in \mathcal{P}$

NEW STATE

POLICY

$\pi(a|s)$

Agent chooses actions by learning this policy distribution which maps states to actions

$reward \in \mathcal{R}$

Function (state, action)

$$r_t^{AC} = \begin{cases} -\left[\beta_t E_t^{AC} + \omega\left(T^{\min} - T_t^{\text{comfy}}\right)\right], & \text{if } T_t^{\text{comfy}} < T^{\min} \\ -\left[\beta_t E_t^{AC} + \omega\left(T_t^{\text{comfy}} - T^{\max}\right)\right], & \text{if } T_t^{\text{comfy}} > T^{\max} \\ -\beta_t E_t^{AC}, & \text{otherwise} \end{cases}$$

Returns a scaler signal (how good/ bad the action was)

$actions \in \mathcal{A}$

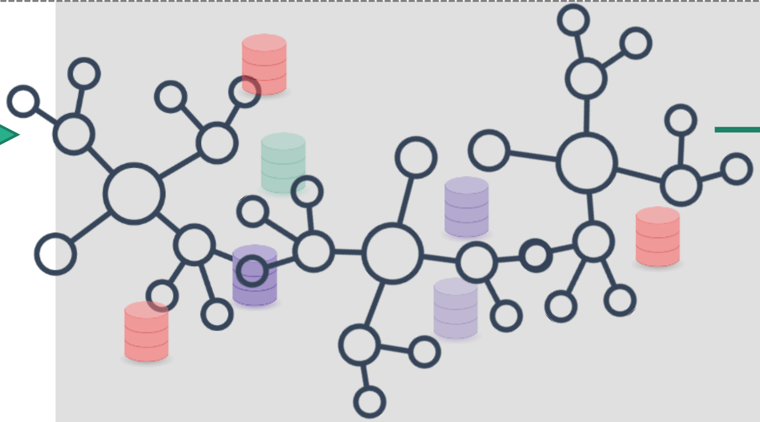
- [$16^0 - 28^0$]
- Charge/ Stop Charging PV,
-
- Release/Hold PV Battery Charge

Future rewards can always be discounted to account for uncertainty of the agent's future

$discount\,factor \in \gamma$

Relational Learning (the big picture)

Knowledge graph augmentation of the building control information before it reaches the agent



RL Agent

Learning from raw data is somewhat straight forward.
But dealing with relational data is quite tricky.

Linked Building Data

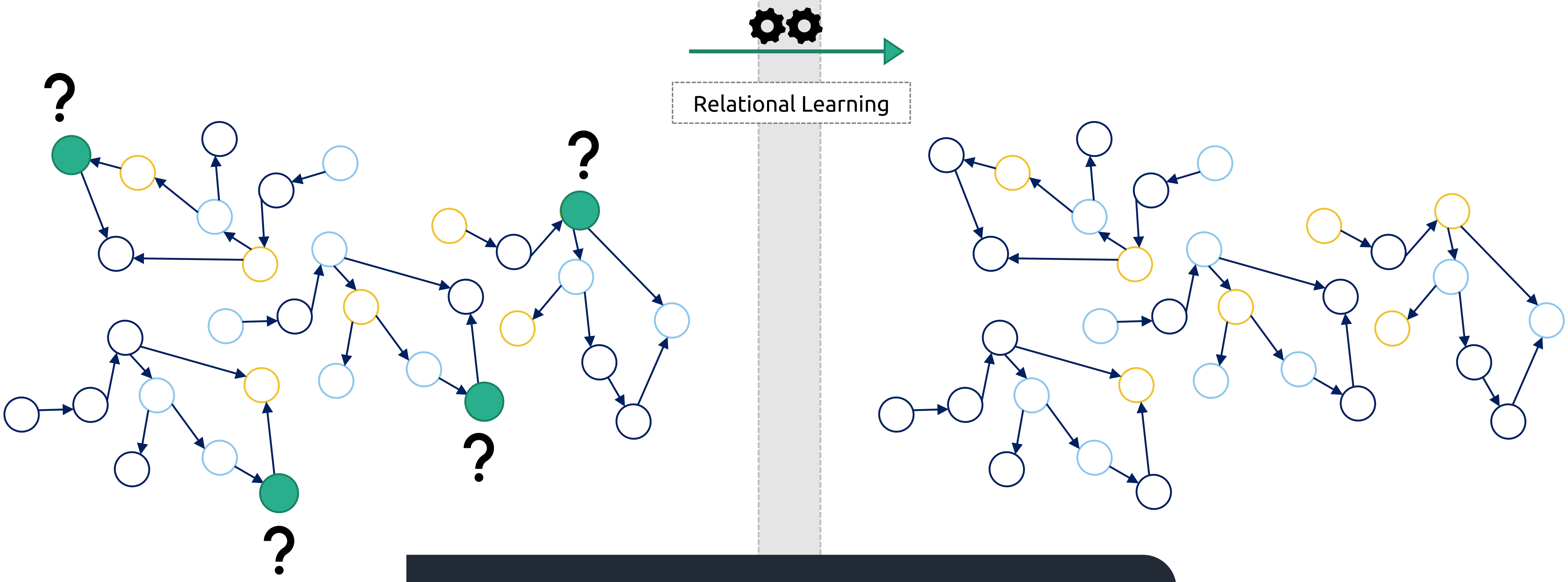
Relational Learning
To the rescue

Reinforcement Learning

Learning from relational building information

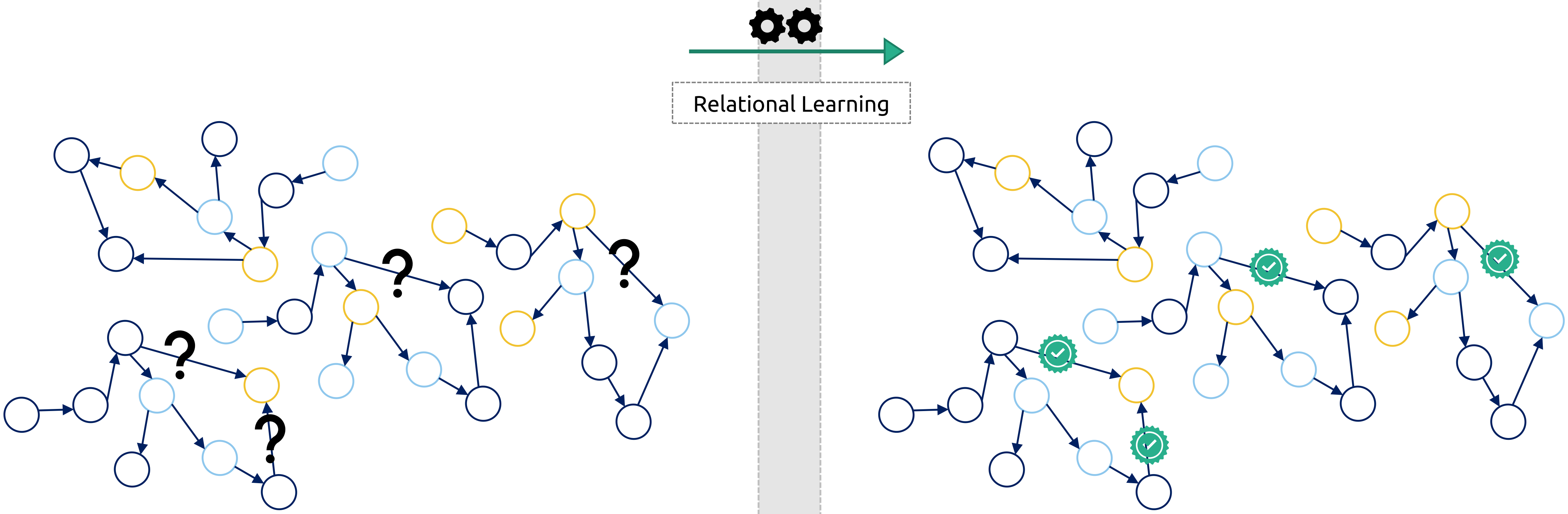
What does it take ?

Relational Learning on LBD



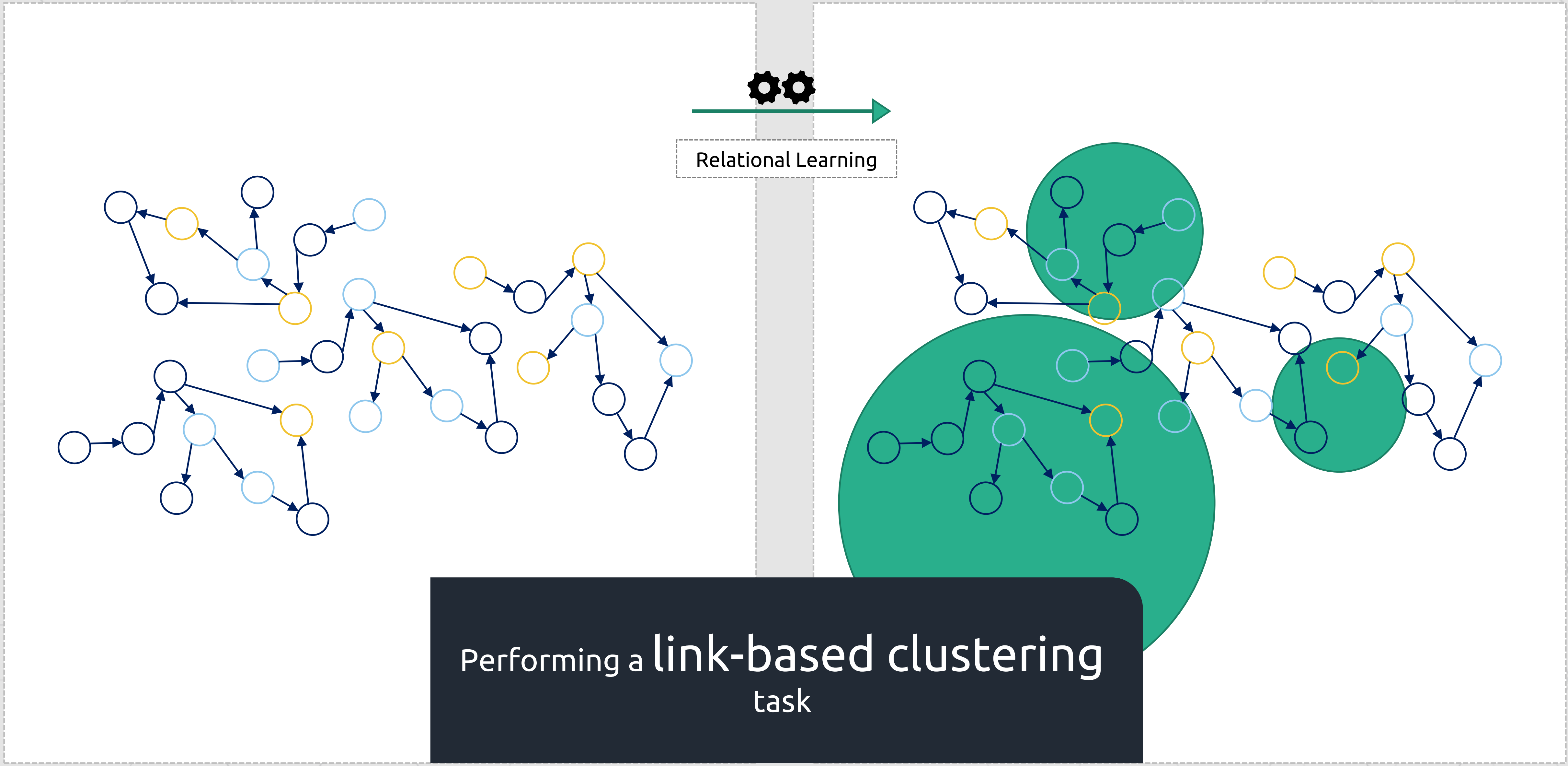
Performing a **node classification** task

Relational Learning on LBD



Performing a **link prediction** task

Relational Learning on LBD



Performing a **link-based clustering** task

Relational Learning on LBD

Typical Machine learning Lifecycles always require a **manual feature engineering** step

Raw Data → Structured Data → Learning Algorithm → Model

Feature Engineering

Very complicated for graph data.

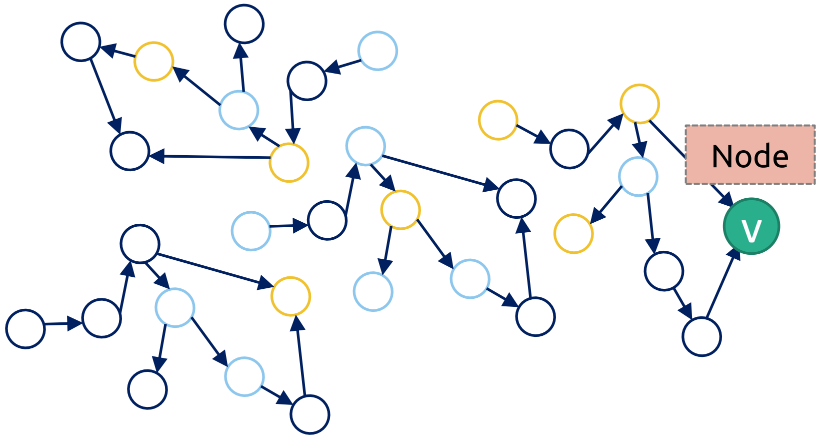
!

Can we skip this step and automatically learn the features

Down stream tasks

Relational Learning on LBD

Efficient **task-independent feature learning** on Graph Networks



$$f: u \rightarrow \mathbb{R}^d$$

Vector

$\mathbb{R}^d$

Feature Representation of the node

- Anomaly detection
- Clustering
- Link prediction and classification

Relational Learning on LBD

An example 2D embedding of nodes for a network

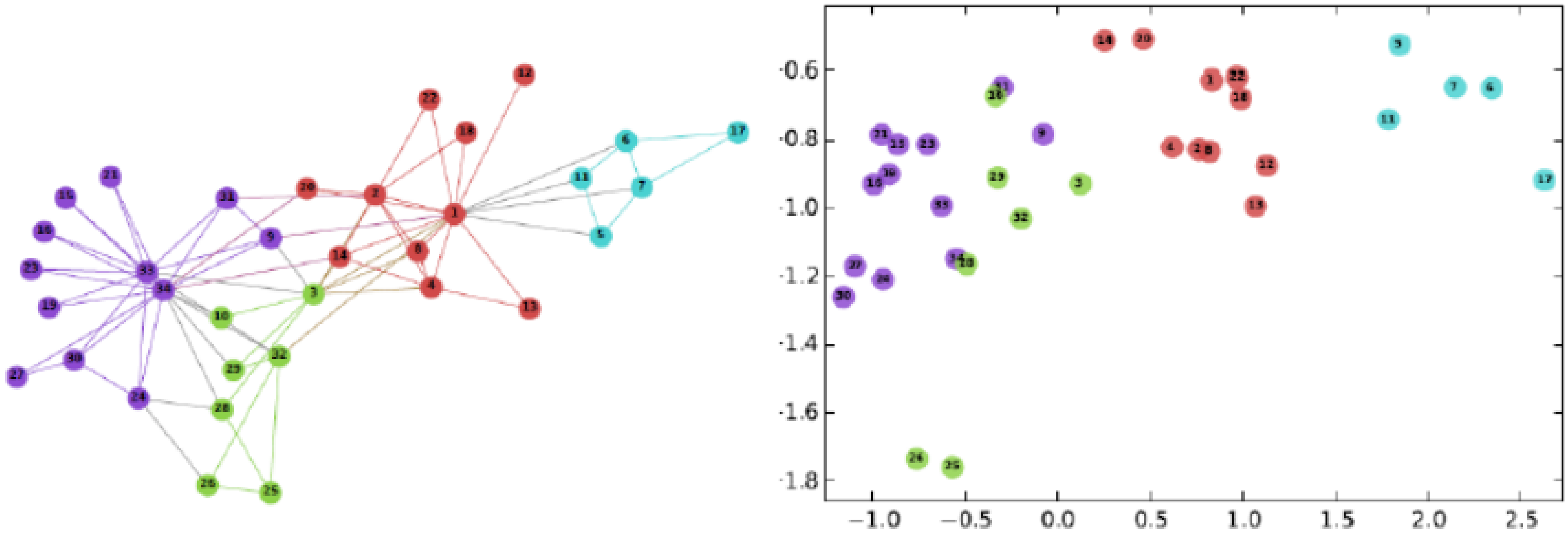
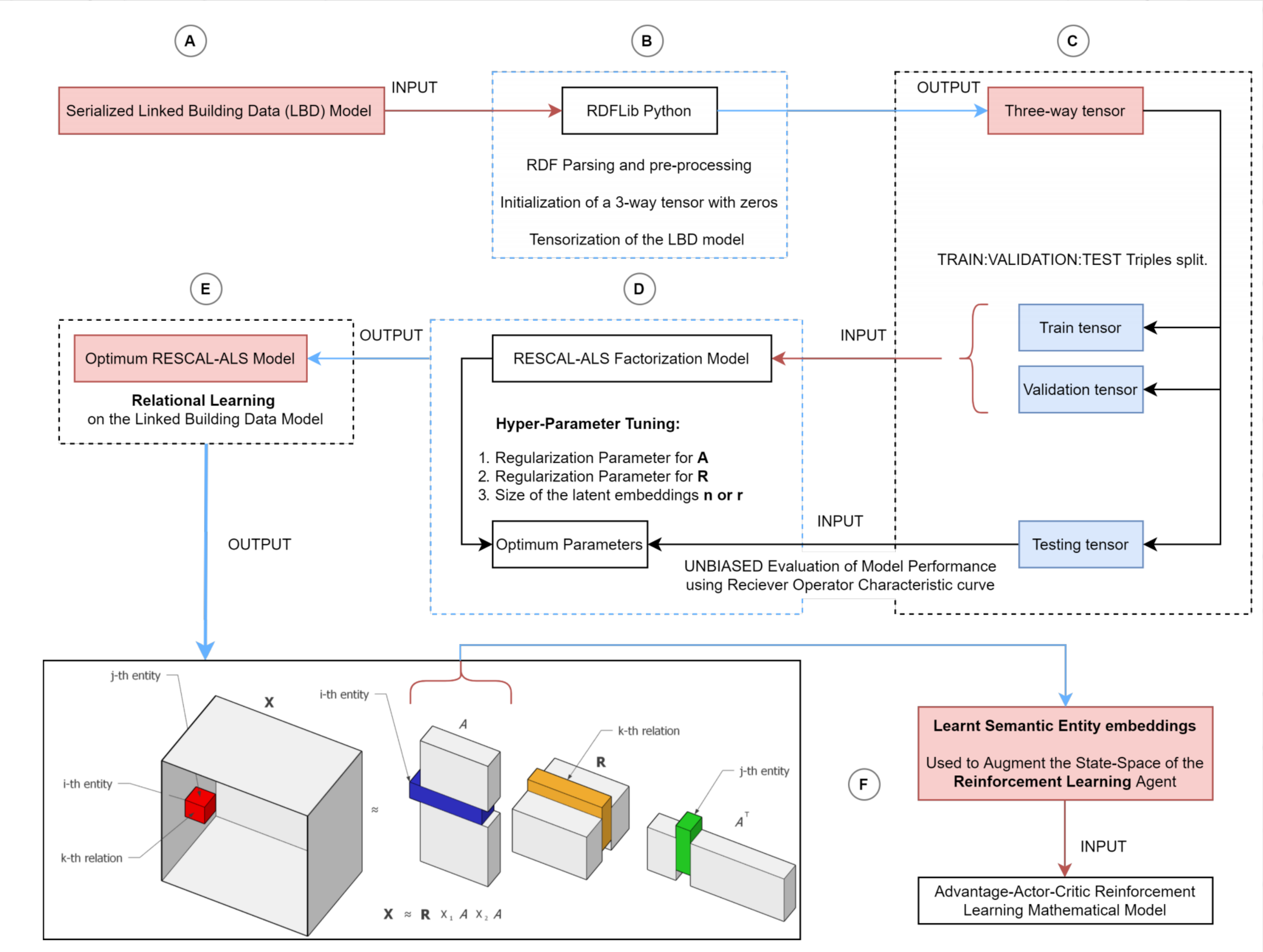


Image from: Perozzi et al. DeepWalk: Online Learning of Social Representations. KDD 2014

Whole Pipeline



Existing Relational Learning models

	Model	Score function $f(h, r, t)$
Translation	Unstructured	$-\lVert \boldsymbol{v}_h - \boldsymbol{v}_t \rVert_{\ell_{1/2}}$
	SE	$-\lVert \mathbf{W}_{r,1}\boldsymbol{v}_h - \mathbf{W}_{r,2}\boldsymbol{v}_t \rVert_{\ell_{1/2}}$ where $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{k\times k}$
	TransE	$-\lVert \boldsymbol{v}_h + \boldsymbol{v}_r - \boldsymbol{v}_t \rVert_{\ell_{1/2}}$ where $\boldsymbol{v}_r \in \mathbb{R}^k$
	TransH	$-\lVert (\mathbf{I} - \boldsymbol{r}_p\boldsymbol{r}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} - \boldsymbol{r}_p\boldsymbol{r}_p^\top)\boldsymbol{v}_t \rVert_{\ell_{1/2}}$ where $\boldsymbol{r}_p, \boldsymbol{v}_r \in \mathbb{R}^k$, $\mathbf{I}$ denotes an identity matrix size $k \times k$
	TransR	$-\lVert \mathbf{W}_r\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_r\boldsymbol{v}_t \rVert_{\ell_{1/2}}$ where $\mathbf{W}_r \in \mathbb{R}^{n\times k}$, $\boldsymbol{v}_r \in \mathbb{R}^n$
	STransE	$-\lVert \mathbf{W}_{r,1}\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_{r,2}\boldsymbol{v}_t \rVert_{\ell_{1/2}}$ where $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{k\times k}$, $\boldsymbol{v}_r \in \mathbb{R}^k$
	TranSparse	$-\lVert \mathbf{W}_{r,1}(\theta_{r,1})\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_{r,2}(\theta_{r,2})\boldsymbol{v}_t \rVert_{\ell_{1/2}}$ where $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{n\times k}$; $\theta_{r,1}, \theta_{r,2} \in \mathbb{R}$; $\boldsymbol{v}_r \in \mathbb{R}^n$
	TransD	$-\lVert (\mathbf{I} + \boldsymbol{r}_p\boldsymbol{h}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} + \boldsymbol{r}_p\boldsymbol{t}_p^\top)\boldsymbol{v}_t \rVert_{\ell_{1/2}}$ where $\boldsymbol{r}_p, \boldsymbol{v}_r, \boldsymbol{h}_p, \boldsymbol{t}_p \in \mathbb{R}^k$
	lppTransD	$-\lVert (\mathbf{I} + \boldsymbol{r}_{p,1}\boldsymbol{h}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} + \boldsymbol{r}_{p,2}\boldsymbol{t}_p^\top)\boldsymbol{v}_t \rVert_{\ell_{1/2}}$ where $\boldsymbol{r}_{p,1}, \boldsymbol{r}_{p,2}, \boldsymbol{v}_r, \boldsymbol{h}_p, \boldsymbol{t}_p \in \mathbb{R}^k$
Bilinear & Tensor	Bilinear	$\boldsymbol{v}_h^\top \mathbf{W}_r \boldsymbol{v}_t$ where $\mathbf{W}_r \in \mathbb{R}^{k\times k}$
	DISTMULT	$\boldsymbol{v}_h^\top \mathbf{W}_r \boldsymbol{v}_t$ where $\mathbf{W}_r$ is a diagonal matrix $\in \mathbb{R}^{k\times k}$
	SimplE	$\frac{1}{2}\left(\boldsymbol{v}_{h,1}^\top \mathbf{W}_r \boldsymbol{v}_{t,2} + \boldsymbol{v}_{t,1}^\top \mathbf{W}_{r^{-1}} \boldsymbol{v}_{h,2}\right)$ where $\boldsymbol{v}_{h,1}, \boldsymbol{v}_{h,2}, \boldsymbol{v}_{t,1}, \boldsymbol{v}_{t,2} \in \mathbb{R}^k$; $\mathbf{W}_r$ and $\mathbf{W}_{r^{-1}}$ are diagonal matrices $\in \mathbb{R}^{k\times k}$
	SME(bilinear)	$\boldsymbol{v}_h^\top (\mathbf{M}_1 \times_3 \boldsymbol{v}_r)^\top (\mathbf{M}_2 \times_3 \boldsymbol{v}_r)\boldsymbol{v}_t$ where $\boldsymbol{v}_r \in \mathbb{R}^k$; $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{n\times k\times k}$
	TuckER	$\mathbf{M} \times_1 \boldsymbol{v}_h \times_2 \boldsymbol{v}_r \times_3 \boldsymbol{v}_t$ where $\boldsymbol{v}_r \in \mathbb{R}^n$, $\mathbf{M} \in \mathbb{R}^{k\times n\times k}$; $\times_d$ denotes the tensor product along the d-th mode
	HolE	$\mathsf{sigmoid}(\boldsymbol{v}_t^\top(\boldsymbol{v}_h \star \boldsymbol{v}_r))$ where $\star$ denotes circular correlation
Neural network	NTN	$\boldsymbol{v}_r^\top \mathsf{tanh}(\boldsymbol{v}_h^\top \mathbf{M}_r \boldsymbol{v}_t + \mathbf{W}_{r,1}\boldsymbol{v}_h + \mathbf{W}_{r,2}\boldsymbol{v}_t + \mathbf{b}_r)$ where $\boldsymbol{v}_r, \mathbf{b}_r \in \mathbb{R}^n$; $\mathbf{M}_r \in \mathbb{R}^{k\times k\times n}$; $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{n\times k}$
	ER-MLP	$\mathsf{sigmoid}(\mathbf{w}^\top \mathsf{tanh}(\boldsymbol{W} \mathsf{concat}(\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t)))$
	ConvE	$\boldsymbol{v}_t^\top \mathsf{ReLU}\,(\boldsymbol{W} \mathsf{vec}\,(\mathsf{ReLU}\,(\mathsf{concat}(\overline{\boldsymbol{v}}_h, \overline{\boldsymbol{v}}_r) * \mathbf{\Omega})))$
	ConvKB	$\mathbf{w}^\top \mathsf{concat}\,(\mathsf{ReLU}\,([\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t] * \mathbf{\Omega}))$
Complex vector	ComplEx	$\mathrm{Re}\left(\boldsymbol{c}_h^\top \mathbf{C}_r \hat{\boldsymbol{c}}_t\right)$ where $\mathrm{Re}(c)$ denotes the real part of the complex value $c \in \mathbb{C}$ $\boldsymbol{c}_h, \boldsymbol{c}_t \in \mathbb{C}^k$; $\mathbf{C}_r \in \mathbb{C}^{k\times k}$ is a diagonal matrix ; $\hat{\boldsymbol{c}}_t$ is the conjugate of $\boldsymbol{c}_t$
	RotatE	$-\lVert \boldsymbol{c}_h \circ \boldsymbol{c}_r - \boldsymbol{c}_t \rVert_{\ell_{1/2}}$ where $\boldsymbol{c}_h, \boldsymbol{c}_r, \boldsymbol{c}_t \in \mathbb{C}^k$; $\circ$ denotes the element-wise product
	QuatE	$\boldsymbol{q}_h \otimes \frac{\boldsymbol{q}_r}{\lvert \boldsymbol{q}_r \rvert} \bullet \boldsymbol{q}_t$ where $\boldsymbol{q}_h, \boldsymbol{q}_r, \boldsymbol{q}_t \in \mathbb{H}^k$; $\otimes$ and $\bullet$ denote Hamilton and quaternion inner products, respectively
Path	TransE-COMP	$-\lVert \boldsymbol{v}_h + \boldsymbol{v}_{r_1} + \boldsymbol{v}_{r_2} + \ldots + \boldsymbol{v}_{r_m} - \boldsymbol{v}_t \rVert_{\ell_{1/2}}$ where $\boldsymbol{v}_{r_1}, \boldsymbol{v}_{r_2}, \ldots, \boldsymbol{v}_{r_m} \in \mathbb{R}^k$
	Bilinear-COMP	$\boldsymbol{v}_h^\top \mathbf{W}_{r_1}\mathbf{W}_{r_2}\ldots\mathbf{W}_{r_m}\boldsymbol{v}_t$ where $\mathbf{W}_{r_1}, \mathbf{W}_{r_2}, \ldots, \mathbf{W}_{r_m} \in \mathbb{R}^{k\times k}$