Social Network Analysis using Knowledge-Graph Embeddings and Convolution Operations

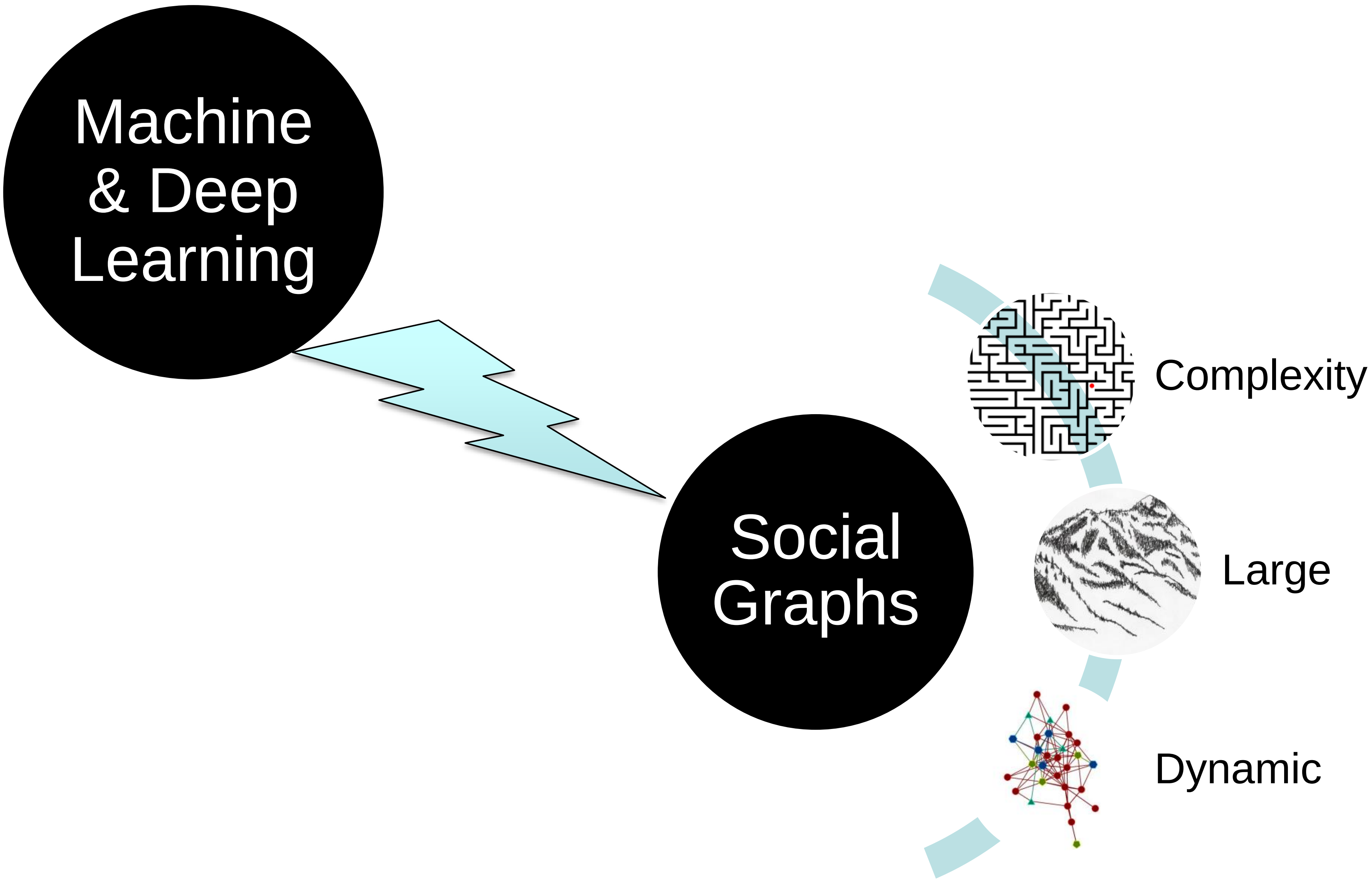
at

25th International Conference on Pattern Recognition

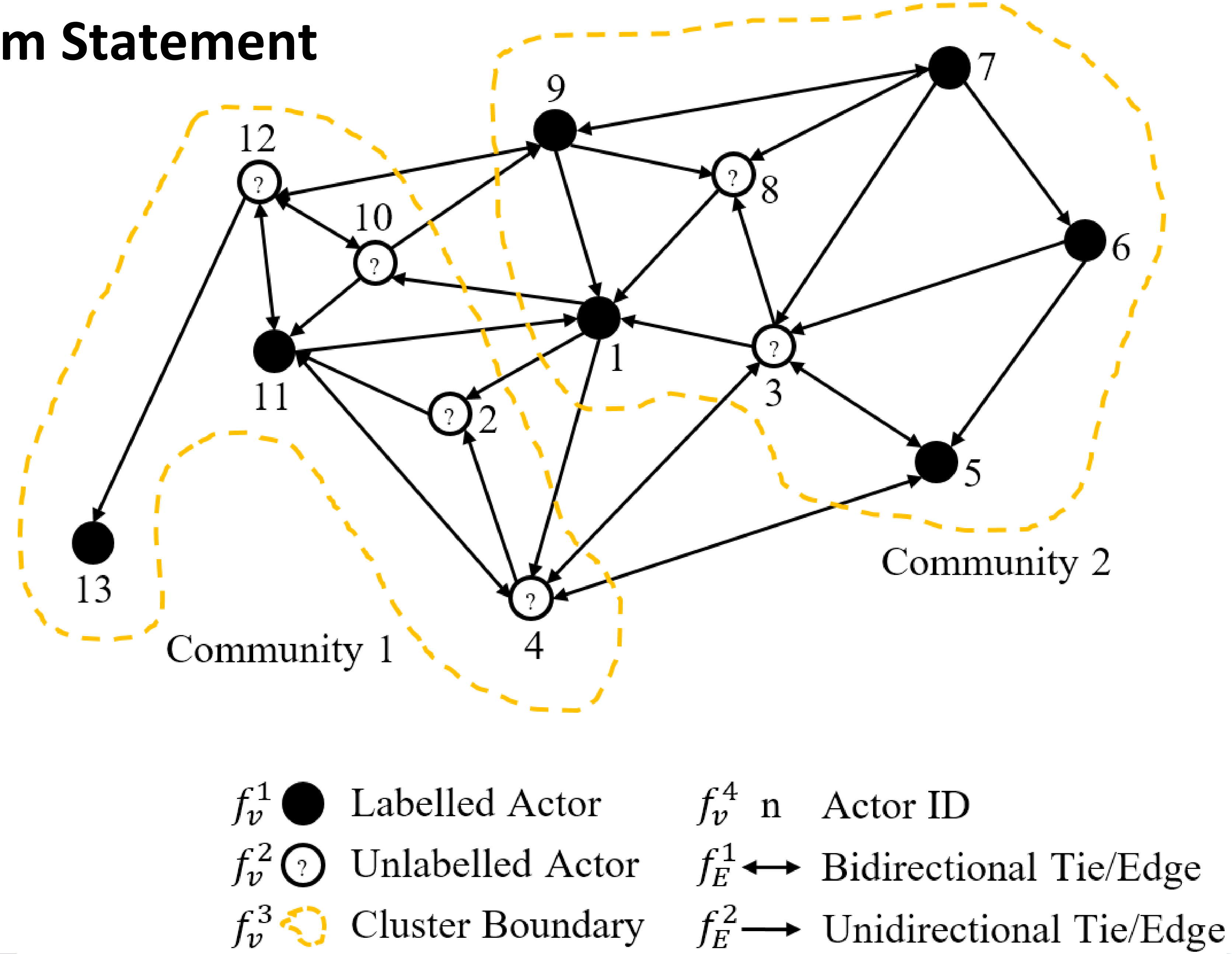
Bonaventure C. Molokwu
molokwub@uwindsor.ca
School of Computer Science - University of Windsor

University of Windsor

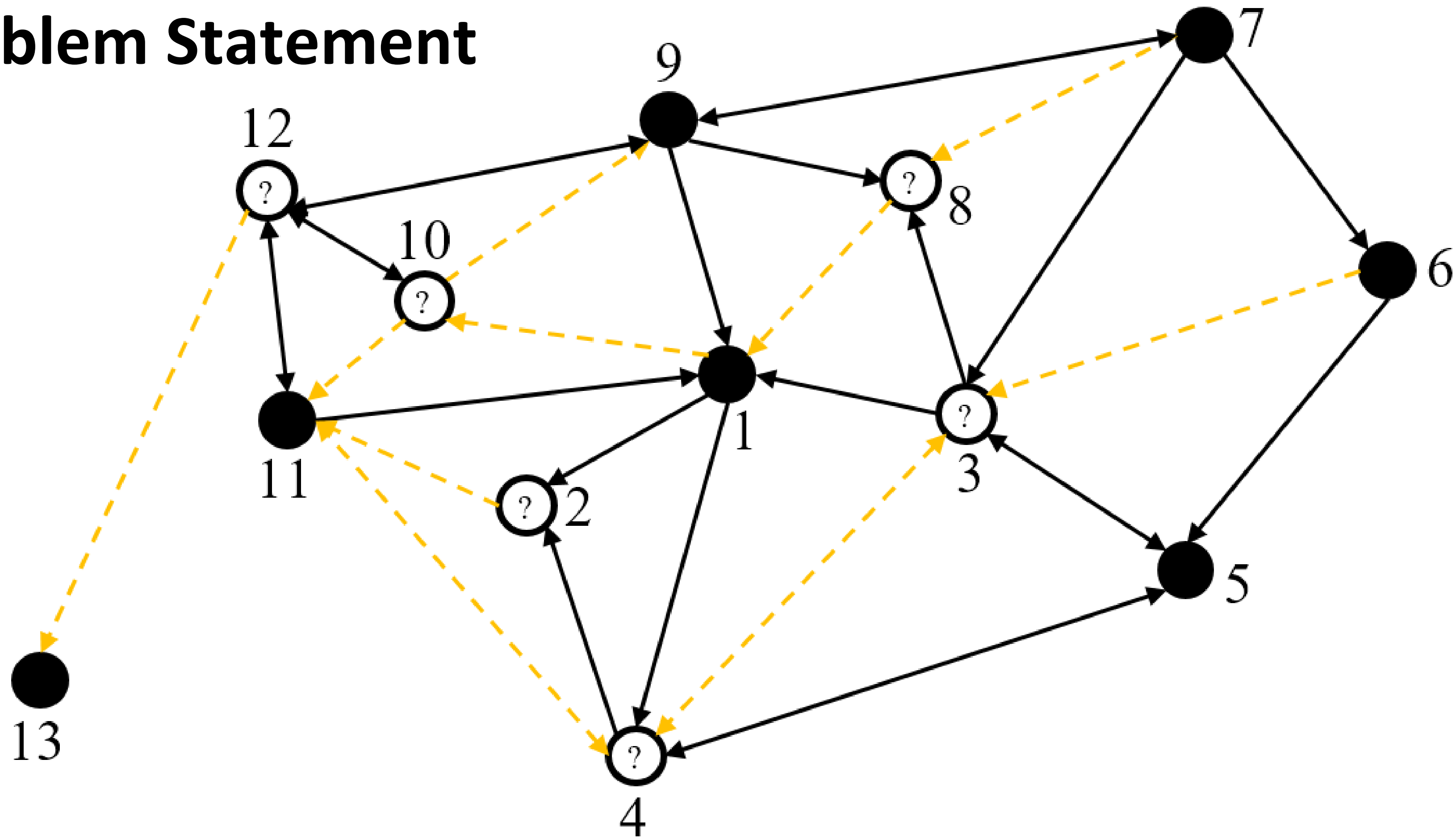
Introduction



Problem Statement



Problem Statement



f_v^1	●	Labelled Actor	f_E^1	- - -	Missing Tie
f_v^2	?	Unlabelled Actor	f_E^2	⟷	Bidirectional Tie/Edge
f_v^3	n	Actor ID	f_E^3	⟶	Unidirectional Tie/Edge

Problem Statement

Social Network

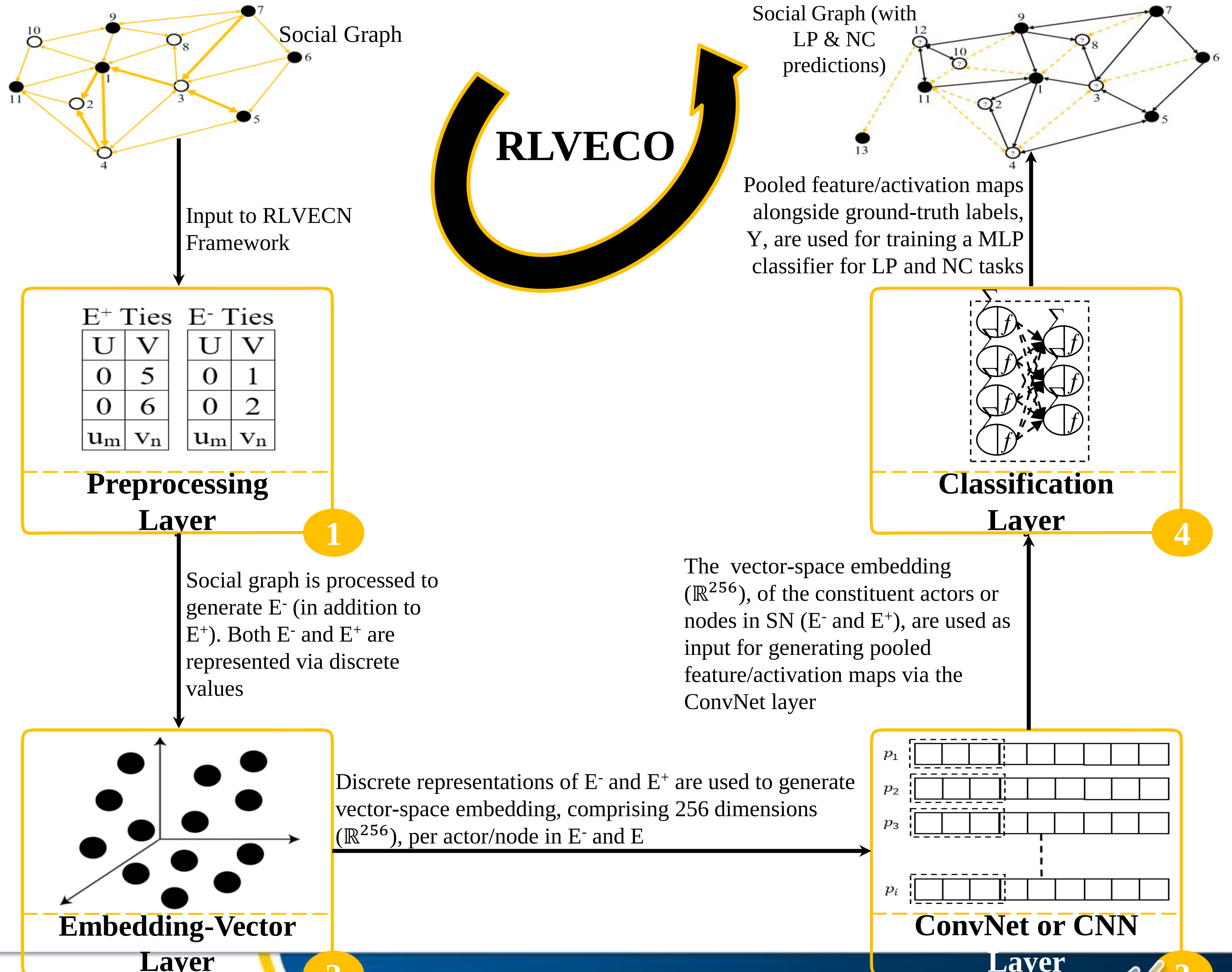
$$
\begin{aligned}
SN = (V, E, f_V, f_E) &\equiv (G, f_V, f_E) \\
V : |\{V\}| = M & \qquad \text{set of actors/vertices with size, M} \\
E : E \subset \{U \times V\} \subset \{V \times V\} & \qquad \text{set of ties/edges between V} \\
f_V : V \rightarrow V' & \qquad \text{vertices' metadata function} \\
f_E : E \rightarrow E' & \qquad \text{edges' metadata function}
\end{aligned} \tag{1}
$$

Node Classification

$$
\begin{aligned}
V_{lbl} \subset V : V_{lbl} \rightarrow Y_{lbl} & \qquad \text{partially labelled actors (or vertices)} \\
V_{ulb} = V - V_{lbl} & \qquad \text{unlabelled actors (or vertices)} \\
f : V \rightarrow Y &
\end{aligned} \tag{2}
$$

Link Prediction

$$
\begin{aligned}
U \subset V : \{U | u_0, u_1, ..., u_m\} \subset \{V | v_0, v_1, ..., v_m\} & \qquad \text{set of actors/vertices} \\
E : (u_i, v_j) \in \{U \times V\} & \qquad \text{set of ties/edges} \\
f : similarity(U, V) &
\end{aligned} \tag{3}
$$



Proposed System Architecture

Proposed (Formal) Methodology

Knowledge-Graph (Vector) Embeddings RL-Layer:

$$\sum_{(u_i,v_j)\in E} logPr(u_i|v_j) = \sum_{(u_i,v_j)\in E} log \frac{exp(u_i \cdot v_j)}{\sum_{m=1}^{M} exp(u_m \cdot v_j)} \tag{1}$$

ConvNet Layer RL-Layer:

$$\begin{aligned} FeatureMap(F) &= 1D\_InputMatrix(X) * Kernel(K) \\ f_i = (X * K)_i = (K * X)_i &= \sum_{j=0}^{J-1} x_j \cdot k_{i-j} = \sum_{j=0}^{J-1} k_j \cdot x_{i-j} \\ r_i \in R = g(f_i \in F) &= max(0, F) \\ p_i \in P = h(r_i \in R) &= maxPool(R) \end{aligned} \tag{2}$$

Multi-Layer Perceptron (MLP) Classifier Layer:

$$Y = f_c(P, \Theta) \tag{3}$$

Proposed Node-Classification Algorithm

Input: $\{V, E, Y_{lbl}\} \equiv \{\text{Actors, Ties, Ground-Truth Labels}\}$
Output: $\{Y_{ulb}\} \equiv \{\text{Predicted Labels}\}$

Preprocessing:

$V_{lbl}, V_{ulb} \subset V = V_{lbl} \cup V_{ulb}$ // V_{lbl} : Labelled actors, V_{ulb} : Unlabelled actors
$E : (u_i, v_j) \in \{U \times V\}$ // $(u_i, v_j) \equiv$ (source, target)
$E_{train} = E_t : u_i, v_j \in V_{lbl}$ // $|E_{train}| = \sum indegree(V_{lbl}) + \sum outdegree(V_{lbl})$
$E_{pred} = E_p : u_i, v_j \in V_{ulb}$

$f_c \leftarrow$ Initialize // Construct classifier model

Training:

for $t \leftarrow 0$ **to** $|E_{train}|$ **do**
  $f : E_t \rightarrow [X_t \in \mathbb{R}^q]$ // Embedding operation
  $f_t \in F = X_t \odot K_t$ // Convolution operation
  $r_t \in R = g(F) = max(0, f_t)$
  $p_t \in P = h(R) = maxPool(r_t)$
  $f_c|\Theta : p_t \rightarrow Y_{lbl}$ // MLP classification operation
end for

return $Y_{ulb} = f_c(E_{pred}, \Theta)$

Proposed Link-Prediction Algorithm

Input: $\{V, E, \mathbb{B}_{gTruth}\} \equiv \{\text{Actors, Ties, Ground-Truth Entities}\}$
Output: $\{\mathbb{B}_{pred}\} \equiv \{\text{Predicted Entities}\}$

Preprocessing:
$\mathbb{B}_{gTruth} : \{0, 1\} \equiv \{\text{C0 : -ve/False tie, C1 : +ve/True tie}\}$
$E = E_{+ves} \cup E_{-ves}$
$E : (u_i, v_j) \in \{U \times V\} \subset \{V \times V\}$ // $\{V \times V\} \equiv$ Set of all possible ties
$E_{train} = E_t : E \to \mathbb{B}_{gTruth}$ // $|E_{train}| = E - E_{pred} =$ Ground-Truth edgelist
$E_{pred} = E - E_{train}$ // $E_{pred} : E'_{train} =$ Complement of E_{train}

$f_c \leftarrow$ Initialize // Construct prediction model

Training:
while $E_{train} \neq NULL$ **do**
$f : E_t \to [X_t \in \mathbb{R}^q]$ // Embedding operation
$f_t \in F = X_t \odot K_t$ // Convolution operation
$r_t \in R = g(F) = max(0, f_t)$
$p_t \in P = h(R) = maxPool(r_t)$
$f_c|\Theta : p_t \to \mathbb{B}_{gTruth}$ // MLP : $\Theta = similarity()$
end while

return $\mathbb{B}_{pred} = f_c(E_{pred}, \Theta)$

Experiments and Results (Node Classification)

	Cora		CiteSeer		Facebook	
	F1	RO	F1	RO	F1	RO
RLVECN	**0.83**	**0.91**	**0.73**	**0.85**	**0.87**	**0.98**
GCN	0.81	0.84	0.65	0.76	Vector features absent	
Node2Vec	0.69	0.80	0.45	0.68	0.81	0.87
DeepWalk	0.58	0.74	0.40	0.65	0.76	0.84
LINE	0.34	0.60	0.22	0.54	0.59	0.72
SDNE	0.33	0.60	0.23	0.55	0.55	0.71

	PubMed		Internet-Industry		Terrorists-Relation	
	F1	RO	F1	RO	F1	RO
RLVECN	**0.80**	**0.94**	**0.57**	**0.80**	0.83	0.96
GCN	Vector features absent				**0.84**	**0.97**
DeepWalk	0.50	0.64	0.49	0.65	0.80	0.87
Node2Vec	0.37	0.56	0.42	0.59	0.79	0.86
LINE	0.34	0.53	0.25	0.50	0.76	0.83
SDNE	0.32	0.55	0.25	0.50	0.76	0.82

Experiments and Results (Link Prediction)

$C0 : \mathbb{B} = 0$ **(-ve/False tie)**												
	Cora		CiteSeer		Internet-Industry		Facebook		PubMed		Terrorists-Relation	
	F1	RO	F1	RO	F1	RO	F1	RO	F1	RO	F1	RO
RLVECN	**0.98**	**0.98**	**0.99**	**0.99**	**0.85**	0.89	0.89	**0.99**	**0.99**	**1.00**	0.90	**0.99**
DistMult	0.91	0.93	0.94	0.94	0.80	**0.90**	0.89	0.98	0.92	0.96	0.84	0.97
ComplEx	0.90	0.92	0.93	0.94	0.79	**0.90**	0.86	0.97	0.91	0.95	0.77	0.95
ConvKB	0.89	0.91	0.93	0.93	0.79	**0.90**	0.82	0.93	0.87	0.91	0.71	0.93
HolE	0.81	0.84	0.82	0.85	0.79	0.85	**0.92**	0.98	0.94	0.95	**0.95**	**0.99**

$C1 : \mathbb{B} = 1$ **(+ve/True tie)**												
	Cora		CiteSeer		Internet-Industry		Facebook		PubMed		Terrorists-Relation	
	F1	RO	F1	RO	F1	RO	F1	RO	F1	RO	F1	RO
RLVECN	**1.00**	**1.00**	**1.00**	**1.00**	**0.95**	**0.90**	**0.98**	**1.00**	**0.99**	**1.00**	**0.99**	**1.00**
DistMult	0.92	0.93	0.94	0.94	0.89	**0.90**	0.97	0.98	0.95	0.96	0.97	0.97
ComplEx	0.91	0.92	0.93	0.94	0.88	**0.90**	0.97	0.97	0.94	0.95	0.95	0.95
ConvKB	0.90	0.91	0.92	0.93	0.88	**0.90**	0.95	0.95	0.92	0.93	0.93	0.93
HolE	0.89	0.84	0.87	0.85	0.91	0.85	**0.98**	0.98	0.97	0.95	**0.99**	0.99