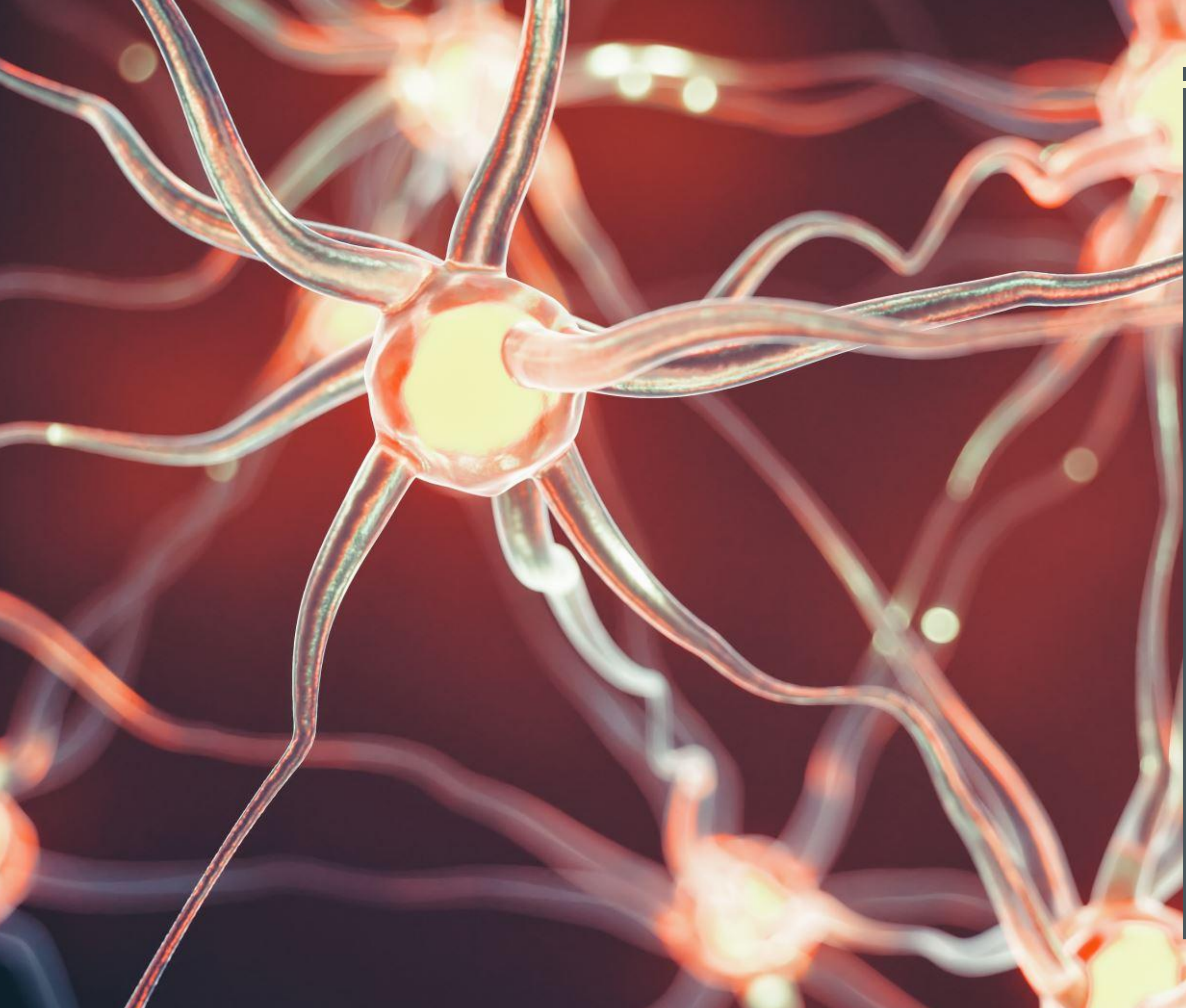


# GRAPH NEURAL NETWORKS

MICHAŁ FILIPIUK

CORE #2
15.04.2020

# RELATIONAL INDUCTIVE BIAS

# WHY DO WE NEED A NEW KIND OF NEURAL NETWORKS?

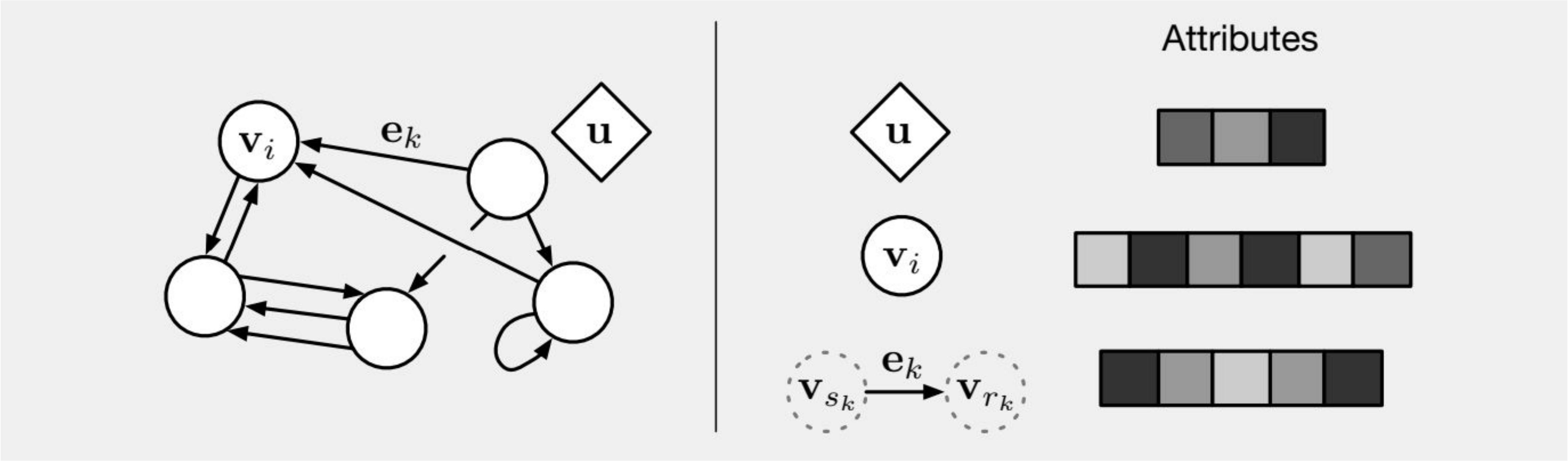
- *We argue that combinatorial generalization must be a top priority for AI to achieve human-like abilities, and that structured representation and computations are key to realizing this objective… We explore how using **relational inductive biases** within deep learning architectures can facilitate learning about entities, relations, and the rules for composing them.*
- *… the world is compositional, or at least, we understand it in compositional terms. When learning, we either fit new knowledge into our existing structured representations, or adjust the structure itself to better accommodate (and make use of ) the new and the old.*

# WHY DO WE NEED A NEW KIND OF NEURAL NETWORKS?

| Component | Entities | Relations | Rel. inductive bias | Invariance |
|---|---|---|---|---|
| Fully connected | Units | All-to-all | Weak | - |
| Convolutional | Grid elements | Local | Locality | Spatial translation |
| Recurrent | Timesteps | Sequential | Sequentiality | Time translation |
| Graph network | Nodes | Edges | Arbitrary | Node, edge permutations |

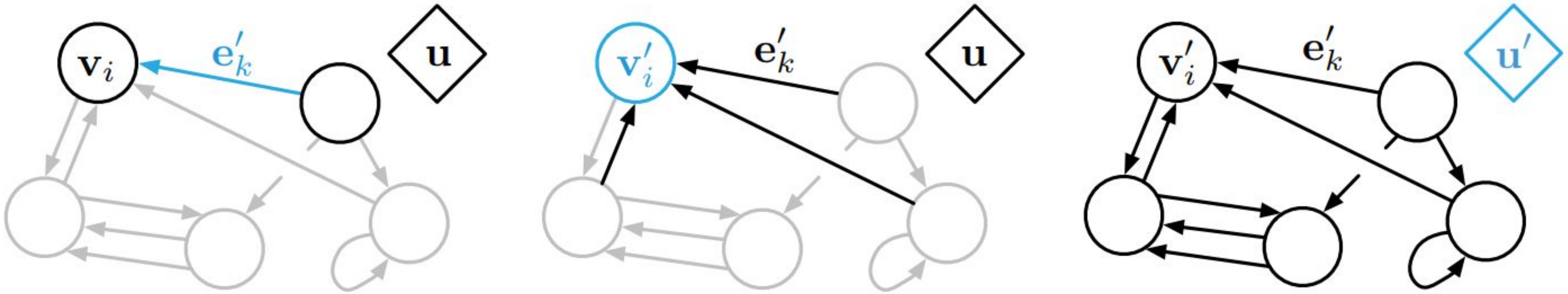
Table 1: Various relational inductive biases in standard deep learning components. See also Section 2.

# WHAT'S GRAPH?

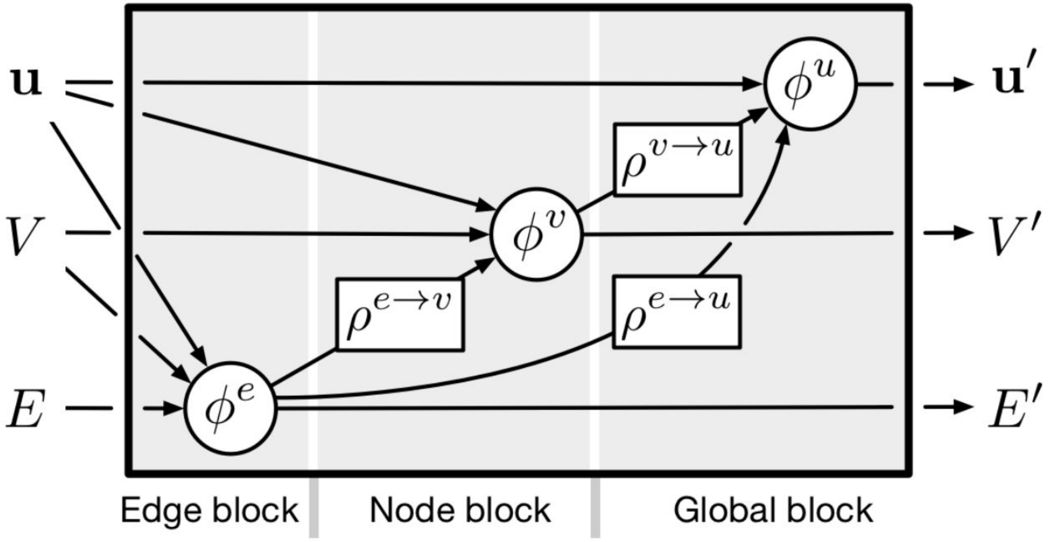


Attributes
$\mathbf{v}_i$
$\mathbf{e}_k$
$\mathbf{u}$
$\mathbf{u}$
$\mathbf{v}_i$
$\mathbf{v}_{s_k}$
$\mathbf{e}_k$
$\mathbf{v}_{r_k}$

# GNN ALGORITHM




(a) Edge update (b) Node update (c) Global update




(a) Full GN block

**Algorithm 1** Steps of computation in a full GN block.

**function** GraphNetwork($E$, $V$, $\mathbf{u}$)
  **for** $k \in \{1 \ldots N^e\}$ **do**
    $\mathbf{e}'_k \leftarrow \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u})$ ▷ 1. Compute updated edge attributes
  **end for**
  **for** $i \in \{1 \ldots N^n\}$ **do**
    **let** $E'_i = \{(\mathbf{e}'_k, r_k, s_k)\}_{r_k=i,\ k=1:N^e}$
    $\bar{\mathbf{e}}'_i \leftarrow \rho^{e\to v}(E'_i)$ ▷ 2. Aggregate edge attributes per node
    $\mathbf{v}'_i \leftarrow \phi^v(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u})$ ▷ 3. Compute updated node attributes
  **end for**
  **let** $V' = \{\mathbf{v}'\}_{i=1:N^v}$
  **let** $E' = \{(\mathbf{e}'_k, r_k, s_k)\}_{k=1:N^e}$
  $\bar{\mathbf{e}}' \leftarrow \rho^{e\to u}(E')$ ▷ 4. Aggregate edge attributes globally
  $\bar{\mathbf{v}}' \leftarrow \rho^{v\to u}(V')$ ▷ 5. Aggregate node attributes globally
  $\mathbf{u}' \leftarrow \phi^u(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u})$ ▷ 6. Compute updated global attribute
  **return** $(E', V', \mathbf{u}')$
**end function**

# GNN ALGORITHM

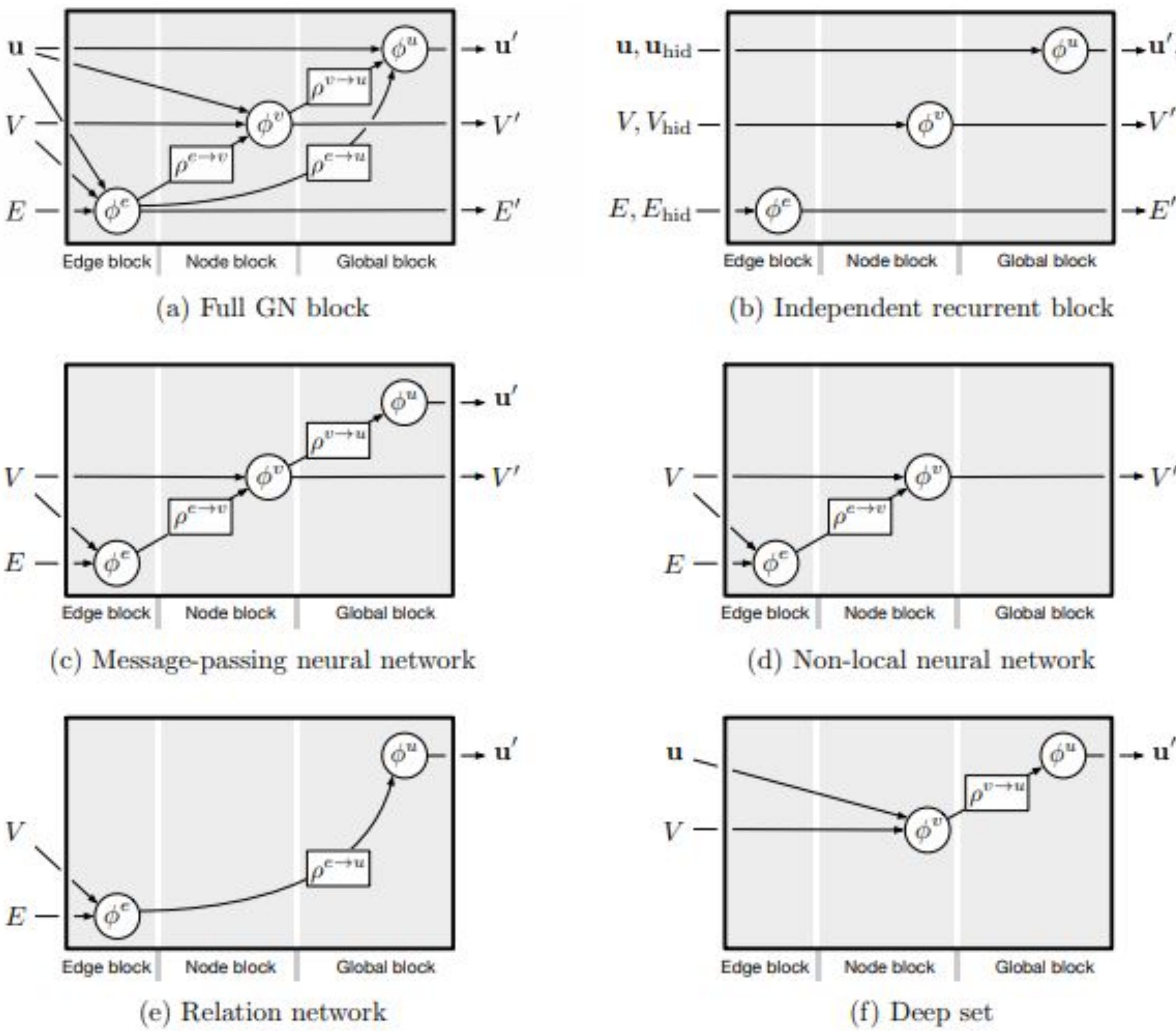
# GNN ALGORITHM

A GN block contains three "update" functions, $\phi$, and three "aggregation" functions, $\rho$,

$$
\begin{aligned}
\mathbf{e}'_k &= \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right) & \qquad \bar{\mathbf{e}}'_i &= \rho^{e\rightarrow v}\left(E'_i\right) \\
\mathbf{v}'_i &= \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right) & \qquad \bar{\mathbf{e}}' &= \rho^{e\rightarrow u}\left(E'\right) \\
\mathbf{u}' &= \phi^u\left(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}\right) & \qquad \bar{\mathbf{v}}' &= \rho^{v\rightarrow u}\left(V'\right)
\end{aligned}
\tag{1}
$$

where $E'_i = \left\{\left(\mathbf{e}'_k, r_k, s_k\right)\right\}_{r_k=i,\; k=1:N^e}$, $V' = \left\{\mathbf{v}'_i\right\}_{i=1:N^v}$, and $E' = \bigcup_i E'_i = \left\{\left(\mathbf{e}'_k, r_k, s_k\right)\right\}_{k=1:N^e}$.

# POSSIBLE VARIATIONS ON GNN




(a) Full GN block

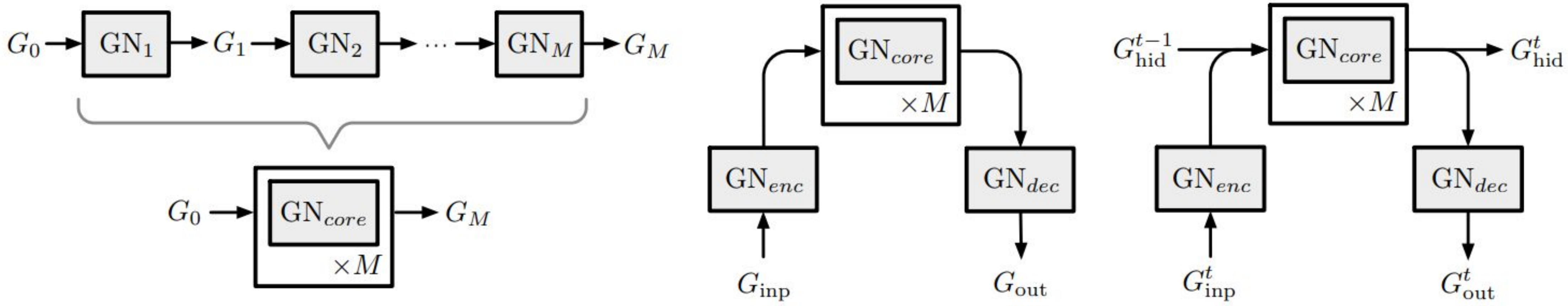
(b) Independent recurrent block

(c) Message-passing neural network

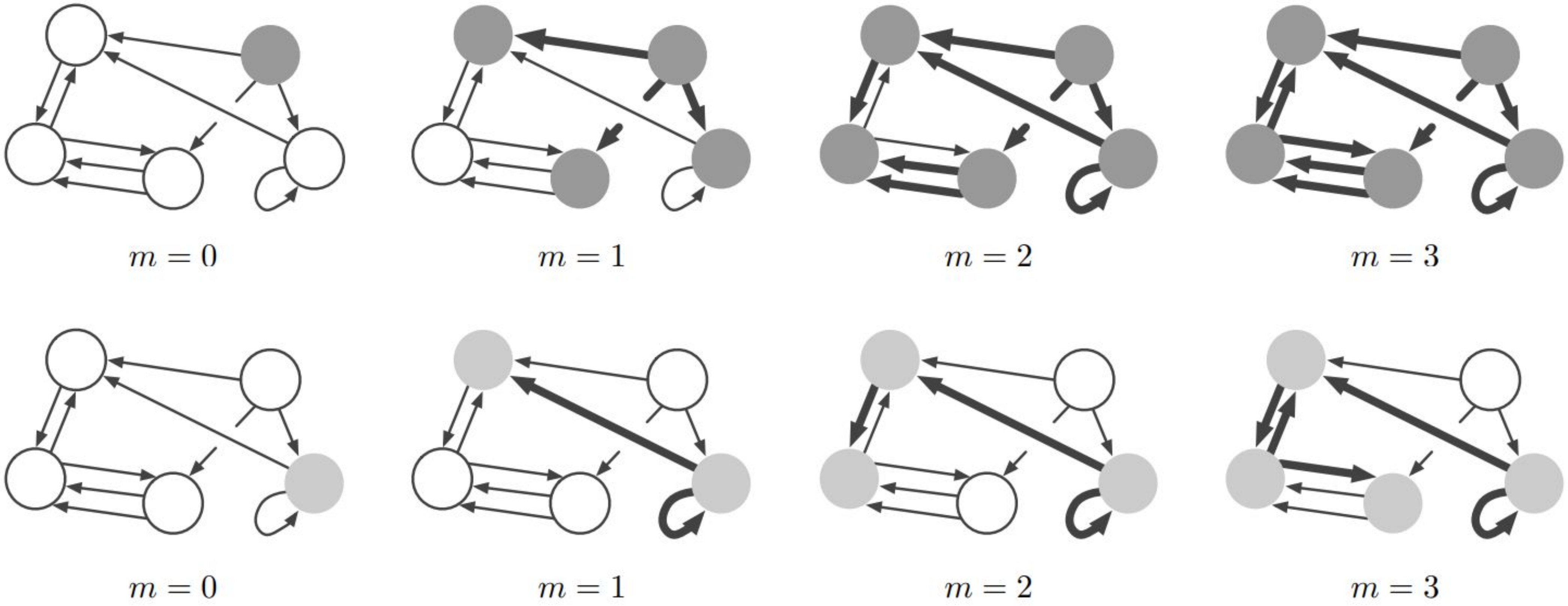
(d) Non-local neural network

(e) Relation network

(f) Deep set



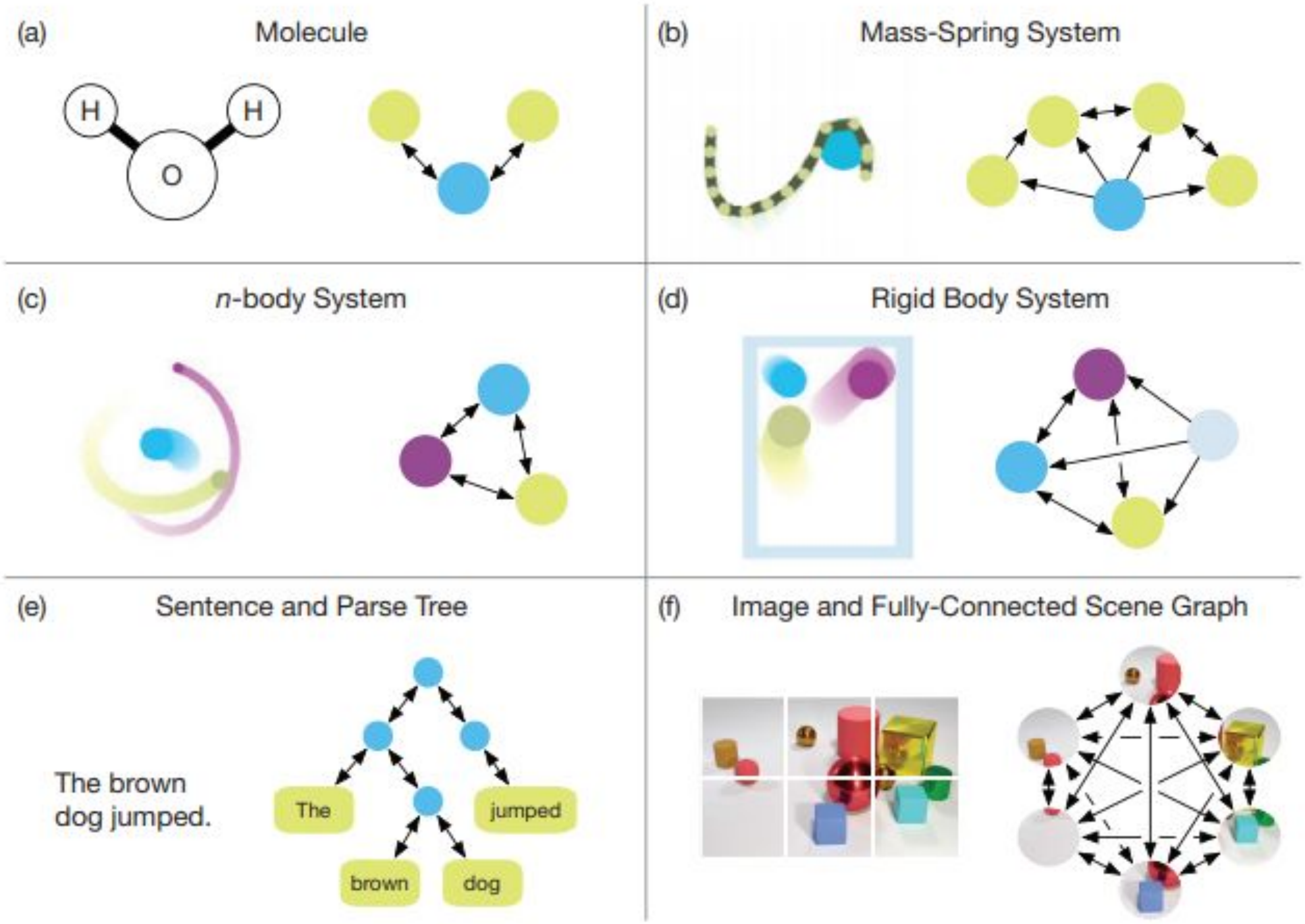

(a) Composition of GN blocks (b) Encode-process-decode (c) Recurrent GN architecture


$m = 0$
$m = 1$
$m = 2$
$m = 3$
$m = 0$
$m = 1$
$m = 2$
$m = 3$

# APPLICATIONS OF GNN

- Physics
- Chemistry and Biology
- Knowledge Graph
- Recommende r Systems
- Computer Vision
- Natural Language Processing
- Generation
- Combinatorial Optimization
- Adversarial Attack
- Graph Clustering
- Graph Classification
- Reinforcemen t Learning
- Traffic Network
- Few-shot and Zero-shot Learning
- Program Representatio n
- Social Network
- Graph Matching

# POSSIBLE DATA FOR GNN



# MULTIPLE EXISTING FRAMEWORKS

- https://github.com/deepmind/graph_nets (TensorFlow)
- https://pytorch-geometric.readthedocs.io/en/latest/ (Pytorch)
- https://github.com/dmlc/dgl (TensorFlow, Pytorch, MXNet)

# DEMO

- tinyurl.com/gn-sort-demo
- tinyurl.com/gn-shortest-path-demo
- tinyurl.com/gn-physics-demo

# BIBLIOGRAPHY AND FURTHER READING


- **Relational inductive biases, deep learning, and graph networks (Battaglia et al., 2018)**
- Neural Message Passing for Quantum Chemistry (Gilmer et al., 2017)
- https://arxiv.org/pdf/1609.02907.pdf
- **https://github.com/thunlp/GNNPapers**
- **https://tkipf.github.io/graph-convolutional-networks/**
- https://www.inference.vc/how-powerful-are-graph-convolutions-review-of-kipf-welling-2016-2/
- **https://deepmind.com/blog/article/Towards-understanding-glasses-with-graph-neural-networks**
- **https://www.youtube.com/watch?v=sTGKOUzIpaQ**

# MNIST OF GNN

[2 1]
[3 1] [3 2]
[4 1] [4 2] [4 3]
[5 1]
[6 1]
[7 1] [7 5] [7 6]
[8 1] [8 2] [8 3] [8 4]
[9 1] [9 3]
[10 3]
[11 1] [11 5] [11 6]
[12 1]
[13 1] [13 4]
[14 1] [14 2] [14 3] [14 4]
[17 6] [17 7]
[18 1] [18 2]
[20 1] [20 2]
[22 1] [22 2]
[26 24] [26 25]
[28 3] [28 24] [28 25]
[29 3]
[30 24] [30 27]
[31 2] [31 9]
[32 1] [32 25] [32 26] [32 29]
[33 3] [33 9] [33 15] [33 16] [33 19] [33 21] [33 23] [33 24] [33 30] [33 31] [33 32]
[34 9] [34 10] [34 14] [34 15] [34 16] [34 19] [34 20] [34 21] [34 23] [34 24] [34 27] [34 28] [34 29] [34 30] [34 31] [34 32] [34 33]