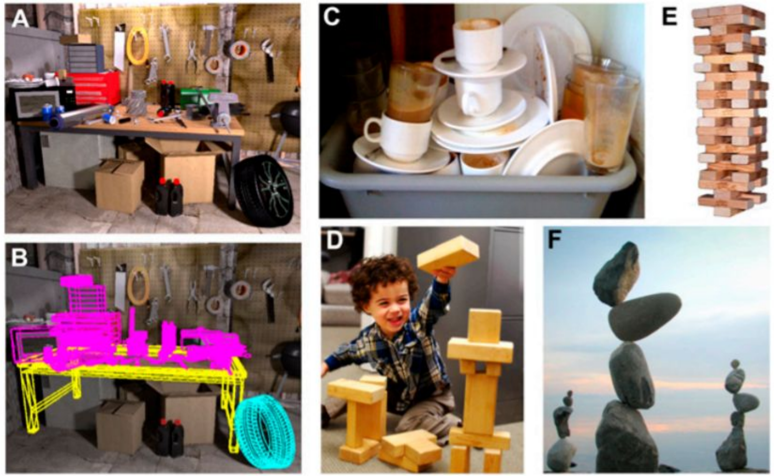
Relational inductive biases, deep learning, and graph networks

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst,
Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski,
Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner,
Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer,
George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash,
Victoria Langston, Chris Dyer, Nicolas Heess,
Daan Wierstra, Pushmeet Kohli, Matt Botvinick,
Oriol Vinyals, Yujia Li, Razvan Pascanu

Presentation: Sebastian Värv

Human vs computer

- Ability to make “infinite use of infinite means"
- Combinatorial generalization
- Representation of complex systems
- Estimating if stack of objects is stable.

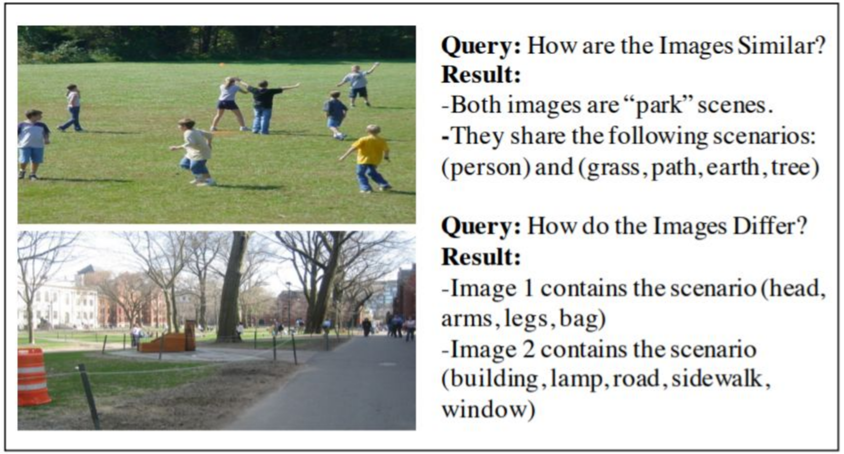


Battaglia et al. 0

© Can Stock Photo - csp33644203

Current challenges

- Complex language and scene understanding
- Reasoning about structured data
- Transferring learning beyond training conditions
- Learning from small amounts of experience



A.Daniels et al. 08

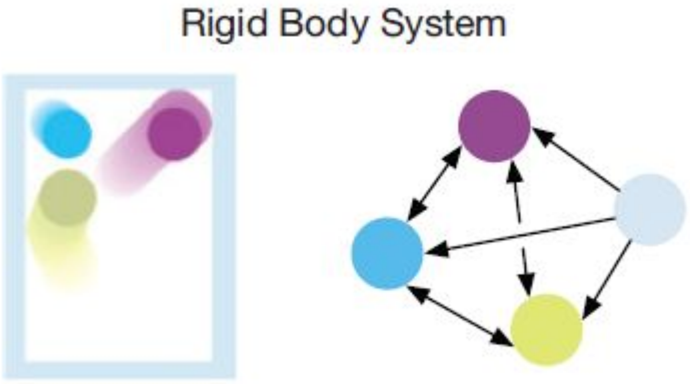
Relational reasoning and inductive bias

Relational reasoning and inductive bias

- Combining relational reasoning and inductive bias
- Hand engineered models do not generalize
- End-to-end learning lacks control and does not have clear structure to represent relations
- Maybe combining these methods works

Relational reasoning

- Defining the structure - composing set of known building blocks.
- Entity - element with attributes.
- Relation - property between entities.
- Rule - function which maps entities and relations to other entities and relations.



Inductive biases

- Multiple solutions to the problem, both equally good
- We use inductive bias to choose one solution

In neural nets context:

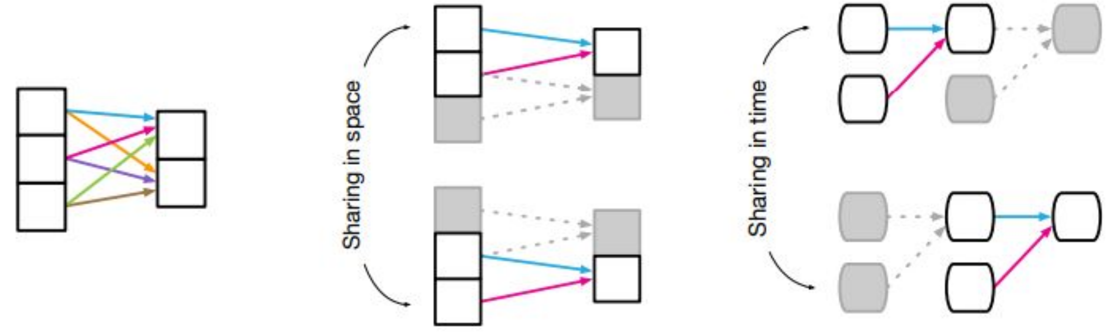
- Regularization term
    - Dropout
    - L2 regularization etc.

Relational inductive biases

- Inductive biases which impose constraints on relationships and interactions among entities in a learning process.

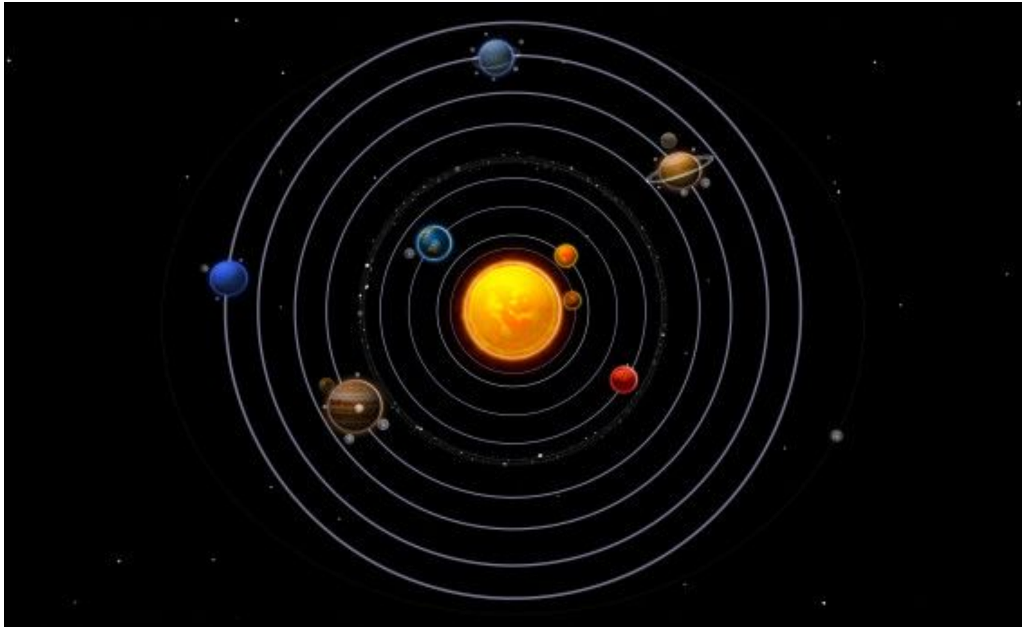
Relational inductive bias in DL building blocks

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



(a) Fully connected (b) Convolutional (c) Recurrent

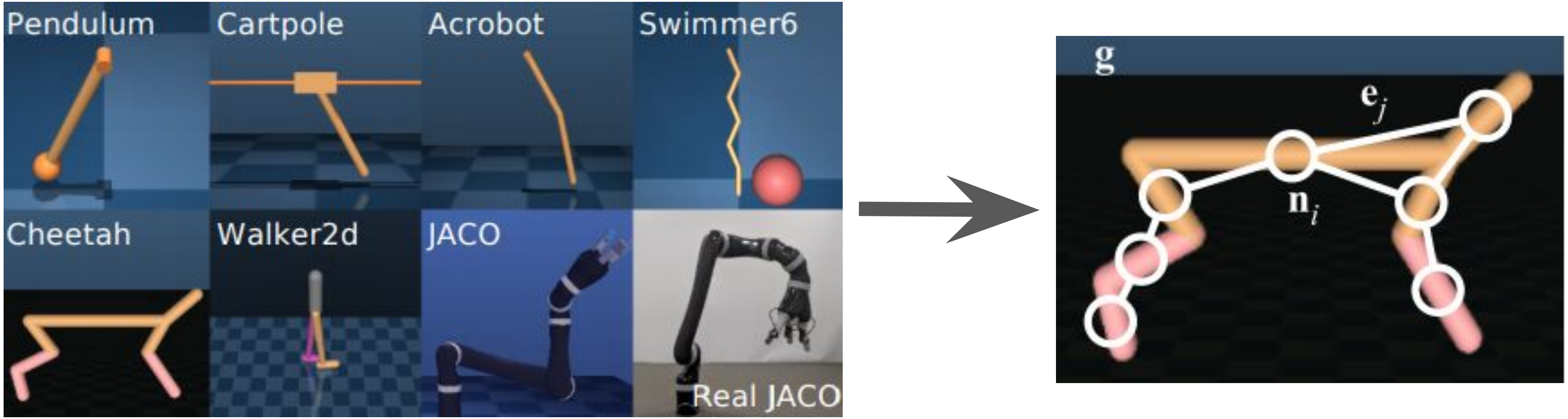
Graph networks



Background

- (Neural networks) that operate on graphs
- Structure the computations accordingly
- First discussed by Gori et al. in 2005.
- Became popular in recent years, still alive (2016- …)
- Effective at tasks thought to have rich relational structure (scene understanding or one-shot learning)

Example use case

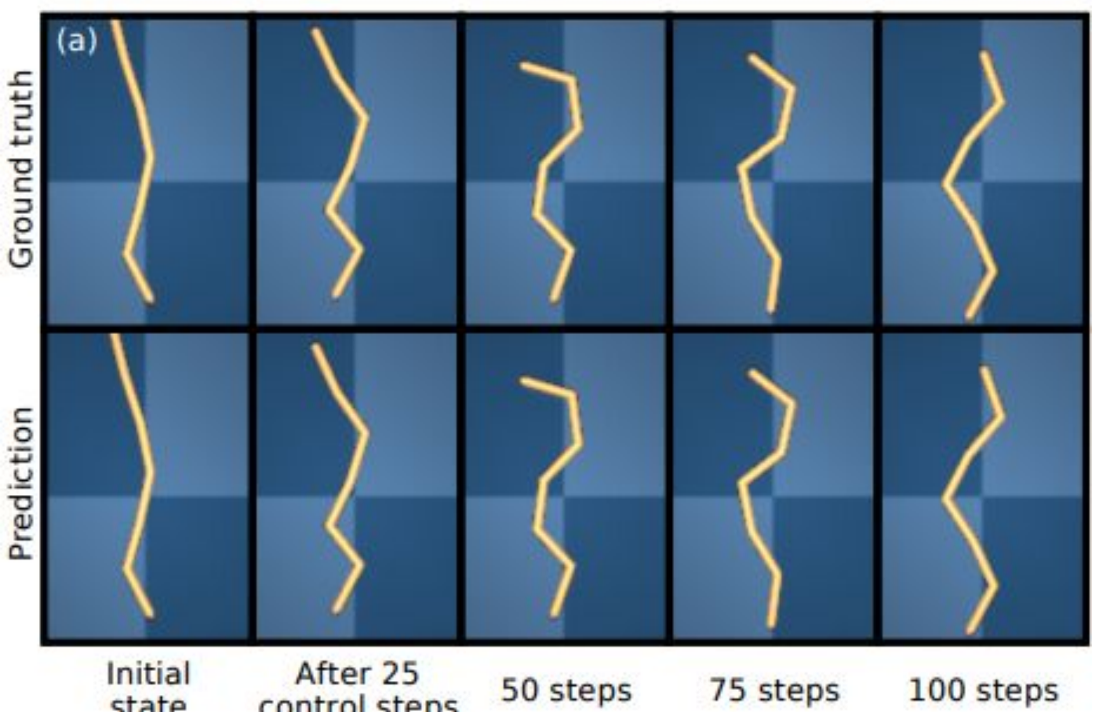


Graph Networks as Learnable Physics Engines for Inference and Control by A.Sanchez-Gonzales et al.

https://drive.google.com/file/d/1xZme1bxvUWQeb9fWFeIECR7YKUXur4XU/view

Task

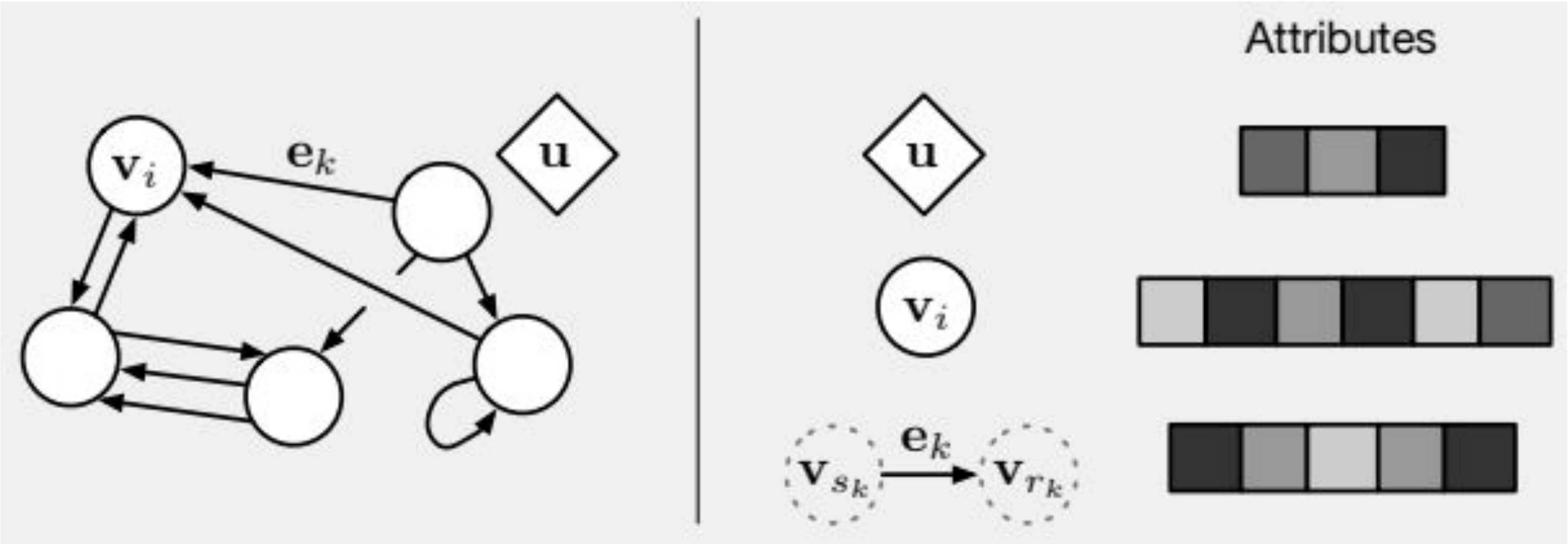
- Predicting the state after n steps (understanding environment)



Results

https://drive.google.com/file/d/15dEUgf5T4ddehMgZiVQ2FtXGqJi9JIDn/view

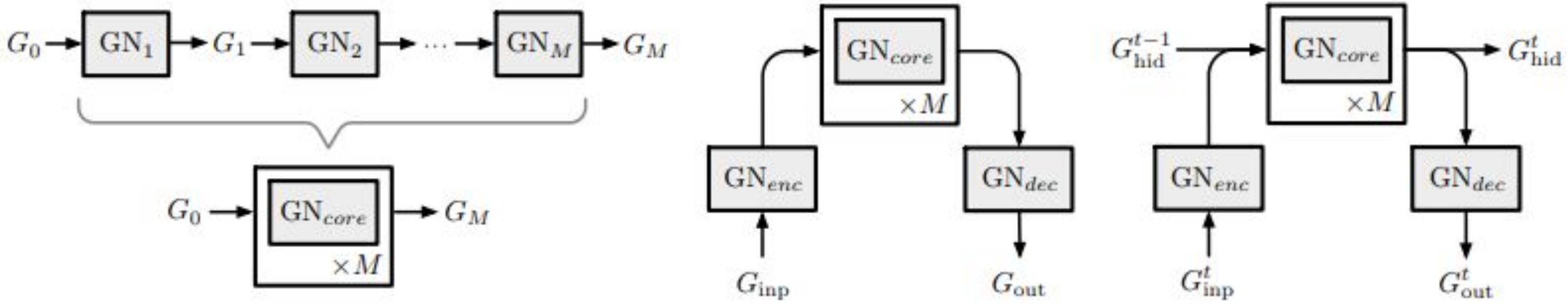
Graph



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

Graph network (GN)

- Authors propose generalised GN framework which should satisfy all existing models.
- GN Block - “graph-to-graph” module



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

Sample task

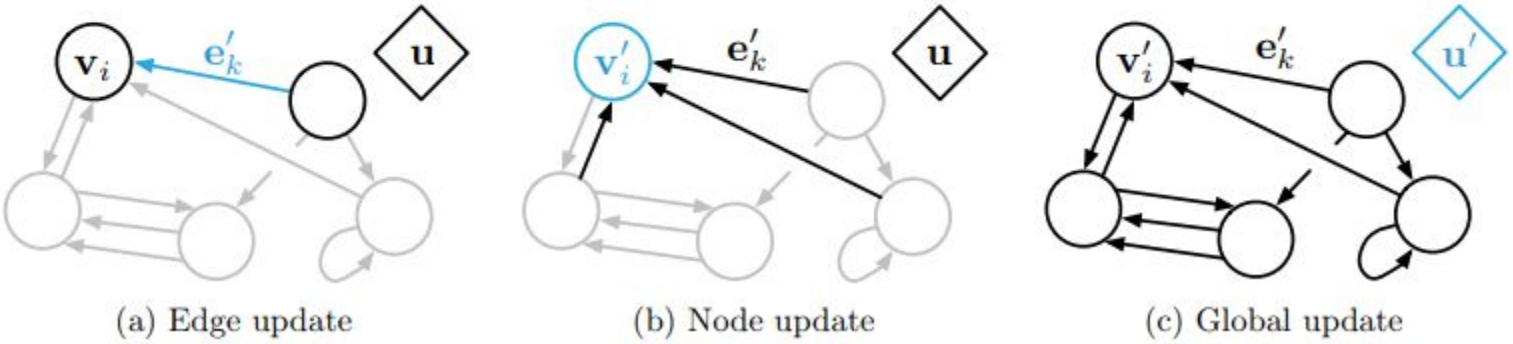
- Predict movements a set of rubber balls in arbitrary gravitational field
- Instead of bounding against each other they have one or more springs to connect with some others.
- Task to predict the position of each rubber ball after certain time.

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

Update sequence in GN block



Algorithm 1 Steps of computation in a full GN block.

```
function GraphNetwork(E, V, u)
    for k ∈ {1 ... N^e} do
        e'_k ← φ^e (e_k, v_{r_k}, v_{s_k}, u)          ▷ 1. Compute updated edge attributes
    end for
    for i ∈ {1 ... N^n} do
        let E'_i = {(e'_k, r_k, s_k)}_{r_k=i, k=1:N^e}
        ē'_i ← ρ^{e→v} (E'_i)                          ▷ 2. Aggregate edge attributes per node
        v'_i ← φ^v (ē'_i, v_i, u)                       ▷ 3. Compute updated node attributes
    end for
    let V' = {v'}_{i=1:N^v}
    let E' = {(e'_k, r_k, s_k)}_{k=1:N^e}
    ē' ← ρ^{e→u} (E')                                  ▷ 4. Aggregate edge attributes globally
    v̄' ← ρ^{v→u} (V')                                  ▷ 5. Aggregate node attributes globally
    u' ← φ^u (ē', v̄', u)                                ▷ 6. Compute updated global attribute
    return (E', V', u')
end function
```

function $\textsc{GraphNetwork}(E, V, \mathbf{u})$

  for $k \in \{1 \ldots N^e\}$ **do**

    $\mathbf{e}'_k \leftarrow \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$ ▷ 1. Compute updated edge attributes

  end for

  for $i \in \{1 \ldots N^n\}$ **do**

    let $E'_i = \left\{\left(\mathbf{e}'_k, r_k, s_k\right)\right\}_{r_k=i,\; k=1:N^e}$

    $\bar{\mathbf{e}}'_i \leftarrow \rho^{e\rightarrow v}\left(E'_i\right)$ ▷ 2. Aggregate edge attributes per node

    $\mathbf{v}'_i \leftarrow \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right)$ ▷ 3. Compute updated node attributes

  end for

  let $V' = \left\{\mathbf{v}'\right\}_{i=1:N^v}$

  let $E' = \left\{\left(\mathbf{e}'_k, r_k, s_k\right)\right\}_{k=1:N^e}$

  $\bar{\mathbf{e}}' \leftarrow \rho^{e\rightarrow u}\left(E'\right)$ ▷ 4. Aggregate edge attributes globally

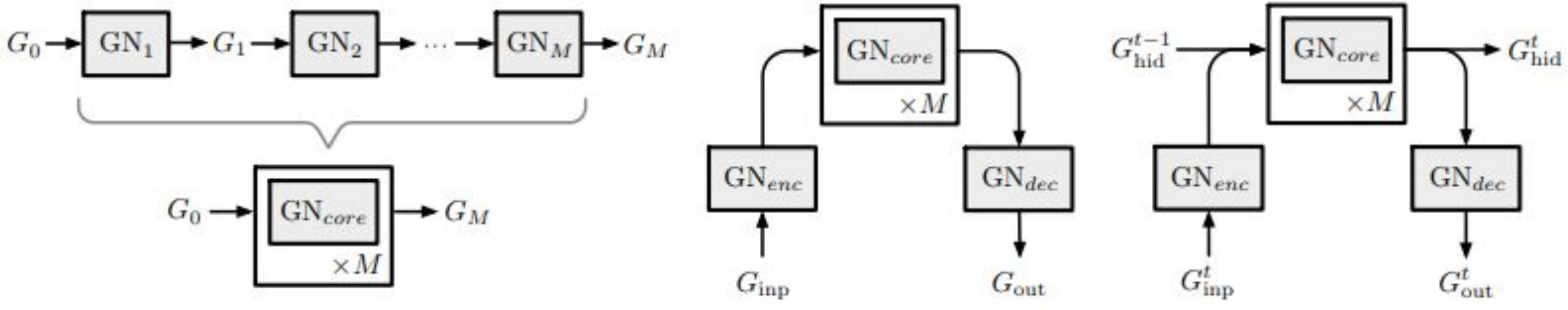
  $\bar{\mathbf{v}}' \leftarrow \rho^{v\rightarrow u}\left(V'\right)$ ▷ 5. Aggregate node attributes globally

  $\mathbf{u}' \leftarrow \phi^u\left(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}\right)$ ▷ 6. Compute updated global attribute

  return $(E', V', \mathbf{u}')$

end function

Configurable structure



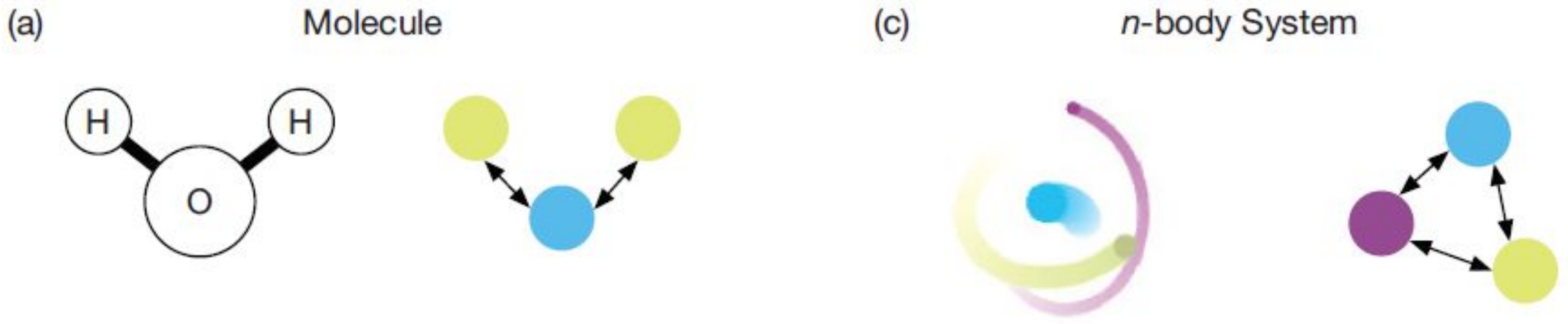
(a) Composition of GN blocks (b) Encode-process-decode (c) Recurrent GN architecture

Full graph network structure

$$
\begin{aligned}
\phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right) &:= f^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right) &&= \mathrm{NN}_e\left(\left[\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right]\right) \\
\phi^v\left(\overline{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right) &:= f^v\left(\overline{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right) &&= \mathrm{NN}_v\left(\left[\overline{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right]\right) \\
\phi^u\left(\overline{\mathbf{e}}', \overline{\mathbf{v}}', \mathbf{u}\right) &:= f^u\left(\overline{\mathbf{e}}', \overline{\mathbf{v}}', \mathbf{u}\right) &&= \mathrm{NN}_u\left(\left[\overline{\mathbf{e}}', \overline{\mathbf{v}}', \mathbf{u}\right]\right) \\
\rho^{e \rightarrow v}\left(E'_i\right) &:= &&= \sum_{\{k:\, r_k=i\}} \mathbf{e}'_k \\
\rho^{v \rightarrow u}\left(V'\right) &:= &&= \sum_i \mathbf{v}'_i \\
\rho^{e \rightarrow u}\left(E'\right) &:= &&= \sum_k \mathbf{e}'_k
\end{aligned}
$$

Obtaining graph structure

- Defining how the input data will be represented as graph manually
- Input may explicitly specify relational structure
- This structure may be inferred or assumed.



Limitations of graph networks

- Limitations inherited directly from graph structure
- Implementation difficulty

Conclusion

- Blending powerful deep learning approaches with structured representations
- Moving a bit closer to combinatorial generalization
- Proposed framework for representing all kinds of relational data using GN

Thank you for listening!