Learning Physical Laws with Deep Learning [quickly]

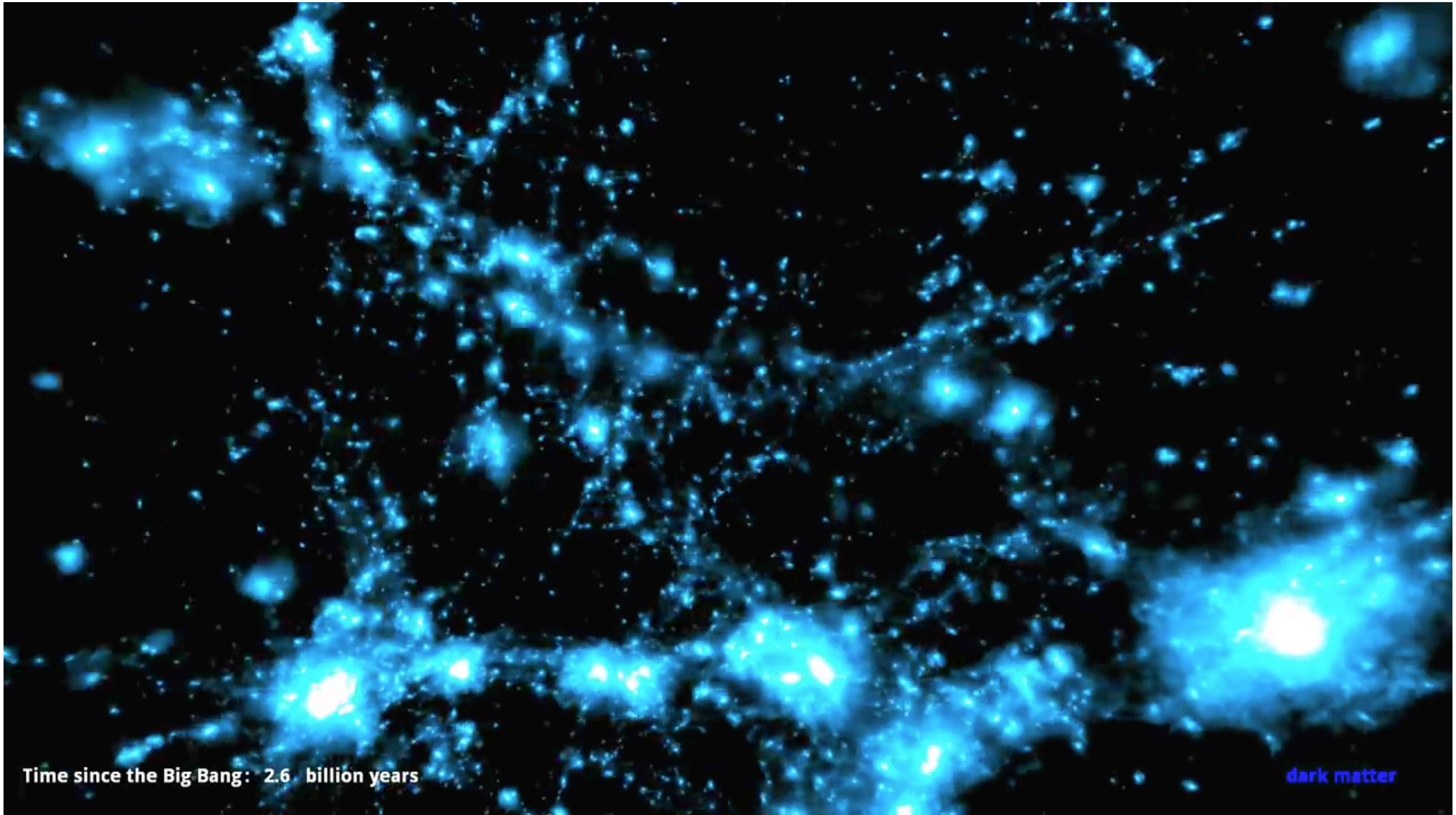
Shirley Ho
Flatiron Institute/ Princeton University/ Carnegie Mellon University

work with Miles Cranmer (Princeton), Rui Xu (Princeton),
Peter Battaglia (Deepmind)

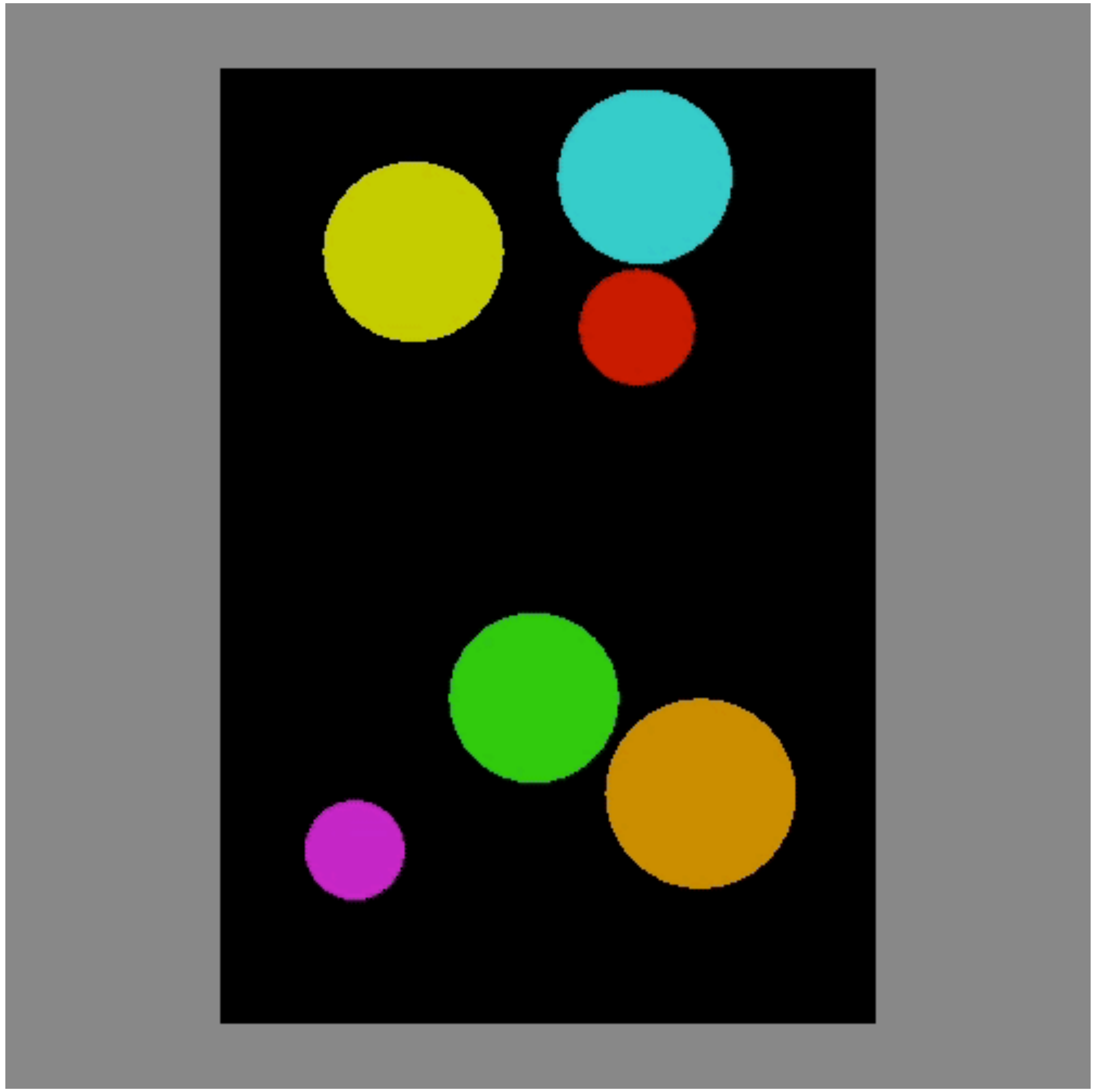
Wait… are you talking about Learning New Physical Laws?

No, this is not yet Artificial General Intelligence talk.

Can we derive the physical law that governs the Universe?

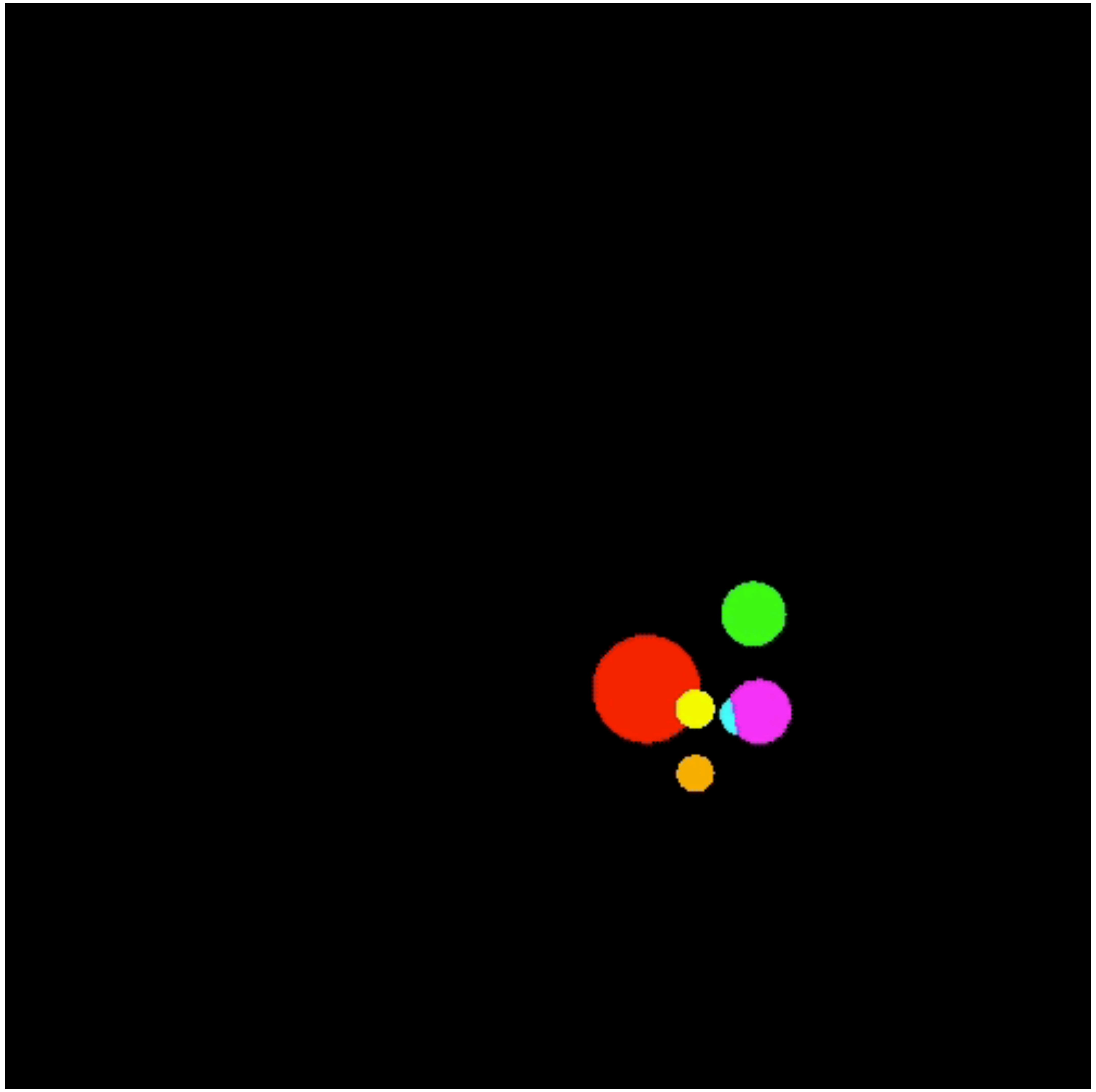


How about this one?



Battaglia et al., 2016, NeurIPS

What is the physical law that governs this system?



Battaglia et al., 2016, NeurIPS

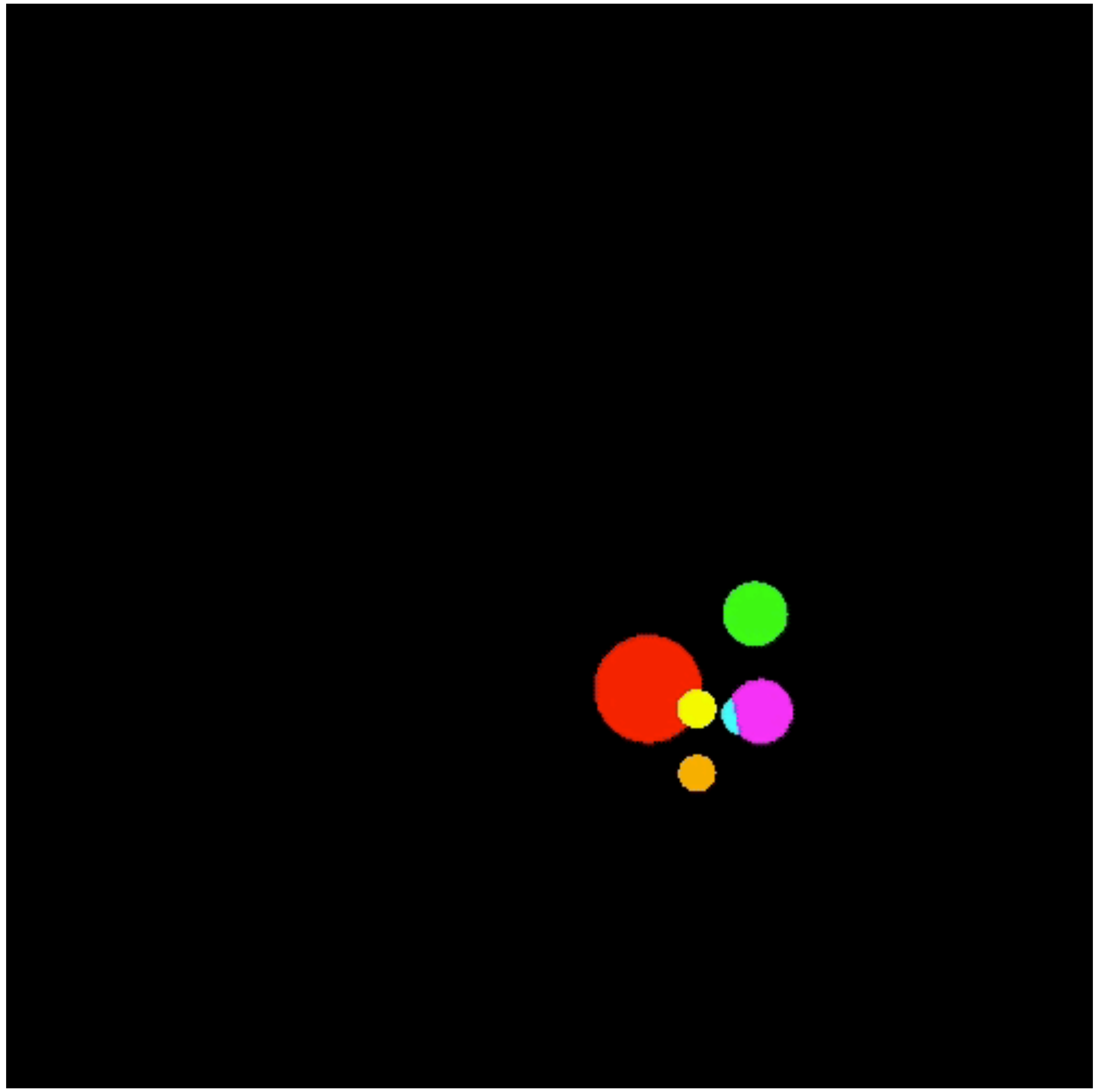
Convolutional Neural Net? Um… not really.

- We know that we can deal with images, cubes of images that you can convolve over and send them along layers of NN.

- But for the problems we talked about earlier, there are no obvious convolution to do that conserves information.

- We cannot simply convolve over these balls bouncing within 4 walls and expect that we will be able to retain all information.

- So what do we do?



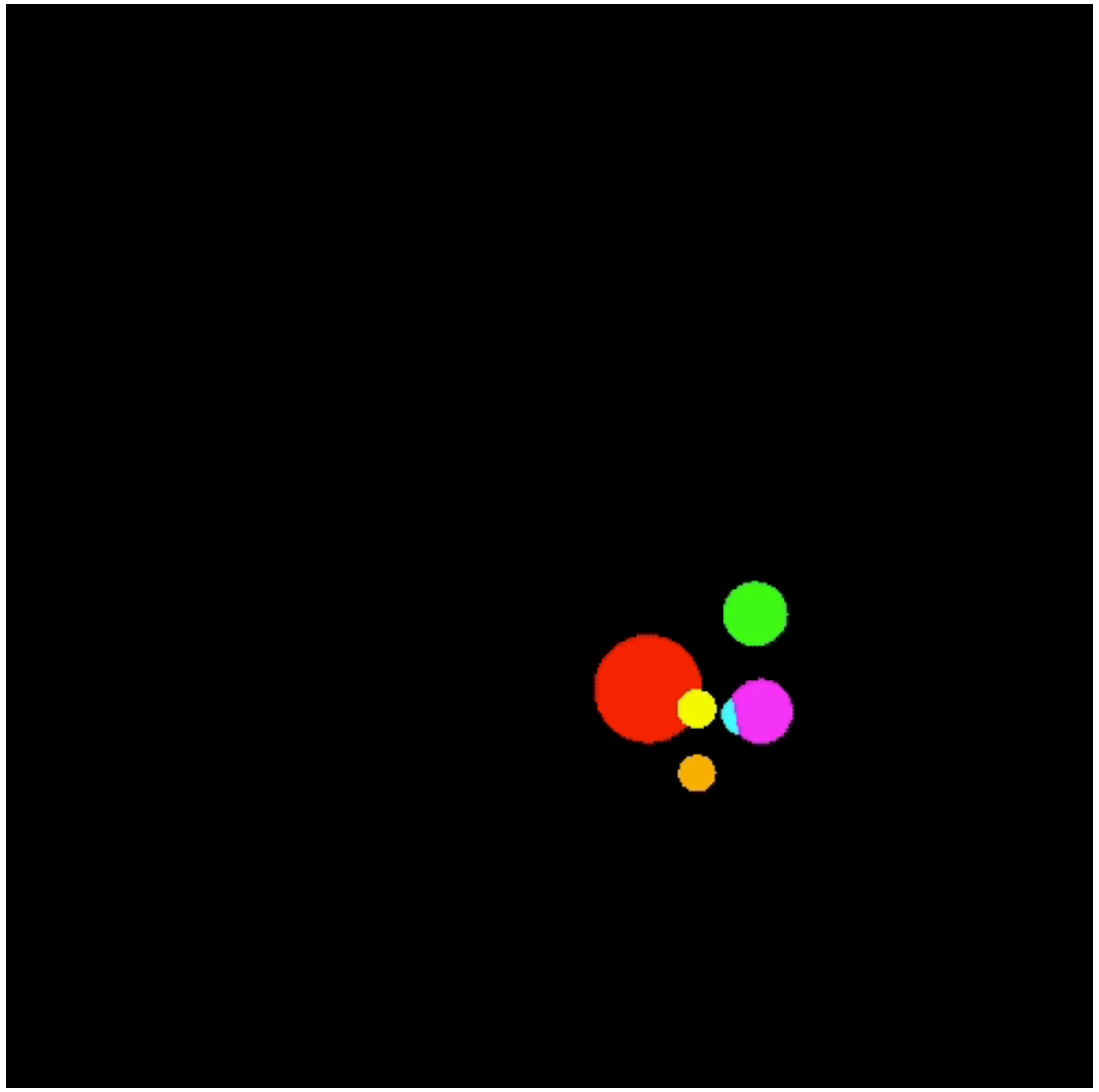
How about something different?

- We know that we can deal with images, cubes of images that you can convolve over and send them along layers of NN.

- But for the problems we talked about earlier, there are no obvious convolution to do that conserves information.

- We cannot simply convolve over these balls bouncing within 4 walls and expect that we will be able to retain all information.

- So what do we do?



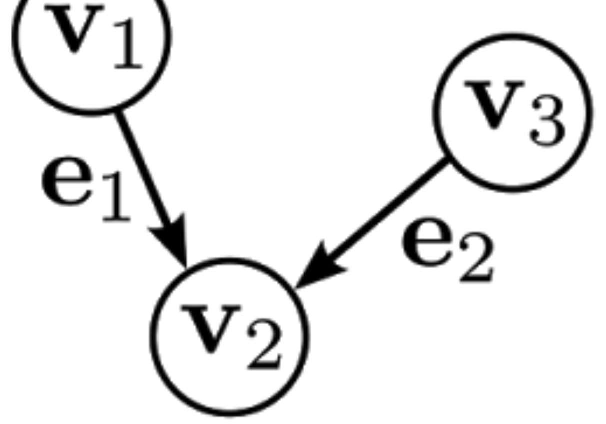
A graph?

- We know that we can deal with images, cubes of images that you can convolve over and send them along layers of NN.

- But for the problems we talked about earlier, there are no obvious convolution to do that conserves information.

- We cannot simply convolve over these balls bouncing within 4 walls and expect that we will be able to retain all information.

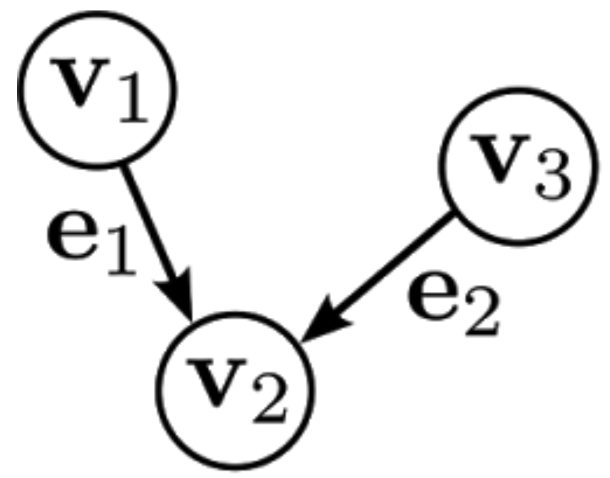
- So what do we do?



A **graph** is a natural way to represent entities and their relations!

A lot of physical laws involve n-body where n>=2

A **graph** is a natural way to represent entities and their relations**:**

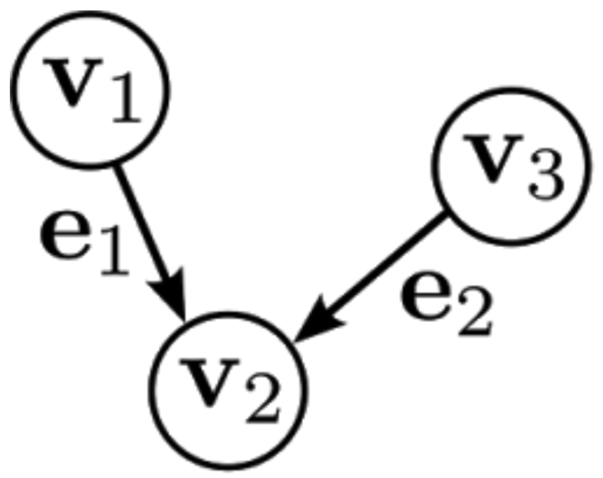


- “Nodes" correspond to entities, objects, events, etc.
- "Edges" correspond to their relations, interactions, transitions, etc.
- Inferences about entities and relations respect the graphical structure.

A lot of physical laws involve n-body where n>=2

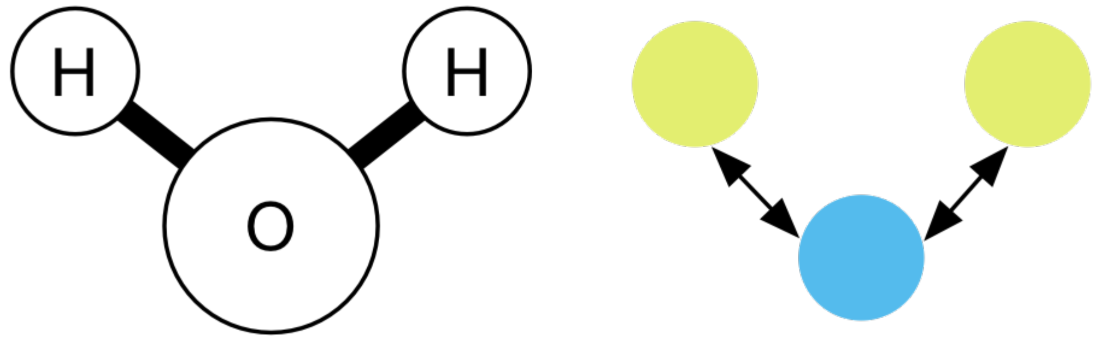
A **graph** is a natural way to represent entities and their relations**:**

- “Nodes" correspond to entities, objects, events, etc.
- "Edges" correspond to their relations, interactions, transitions, etc.
- Inferences about entities and relations respect the graphical structure.

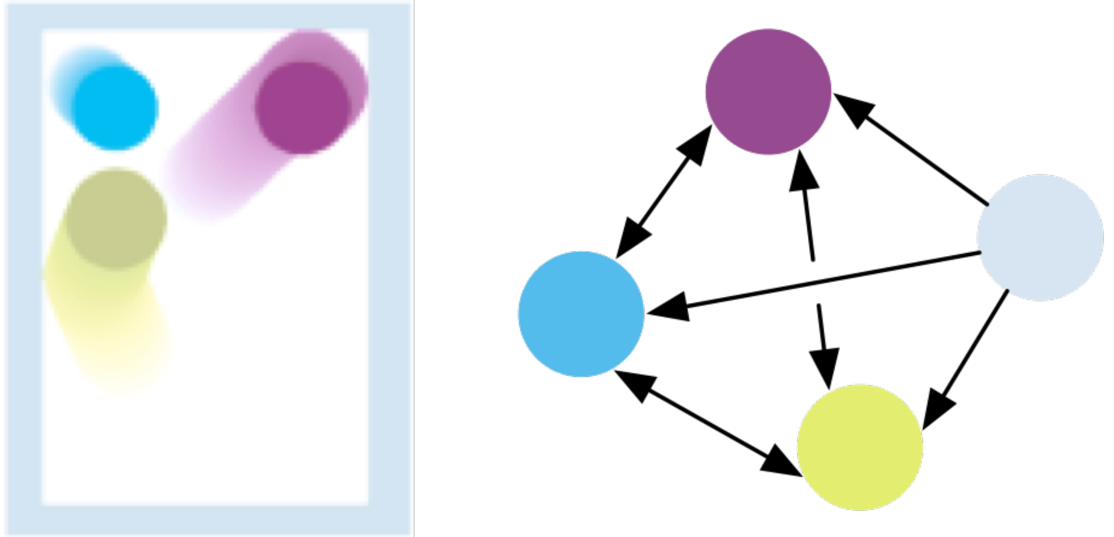


Graphs can capture many complex object/relation systems:

Molecule



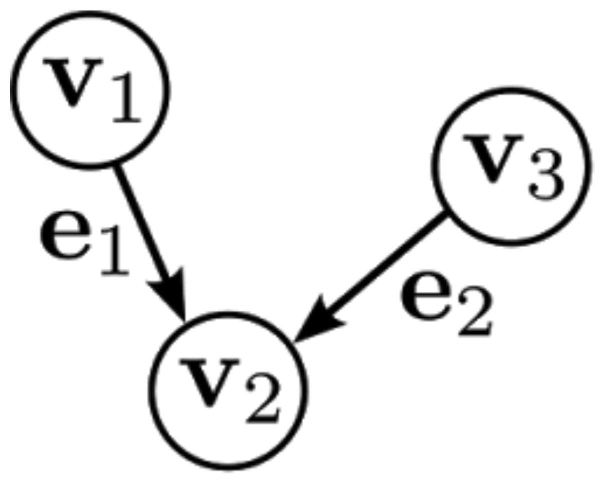
Rigid Body System



A lot of physical laws involve n-body where n>=2

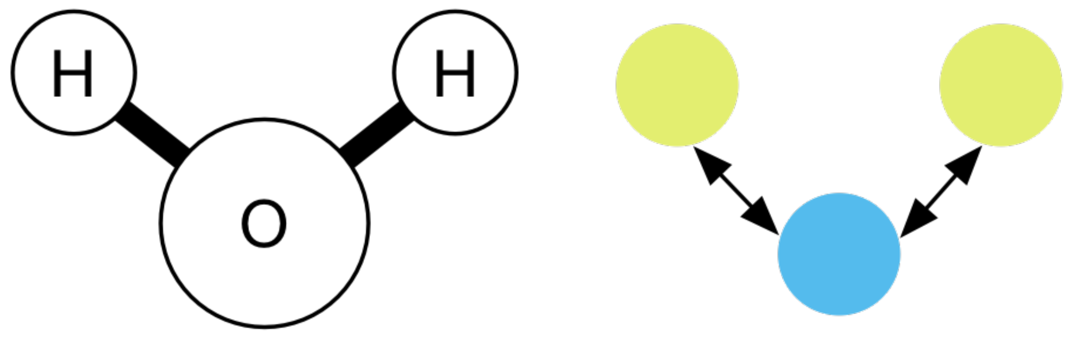
A **graph** is a natural way to represent entities and their relations:

- “Nodes" correspond to entities, objects, events, etc.
- "Edges" correspond to their relations, interactions, transitions, etc.
- Inferences about entities and relations respect the graphical structure.

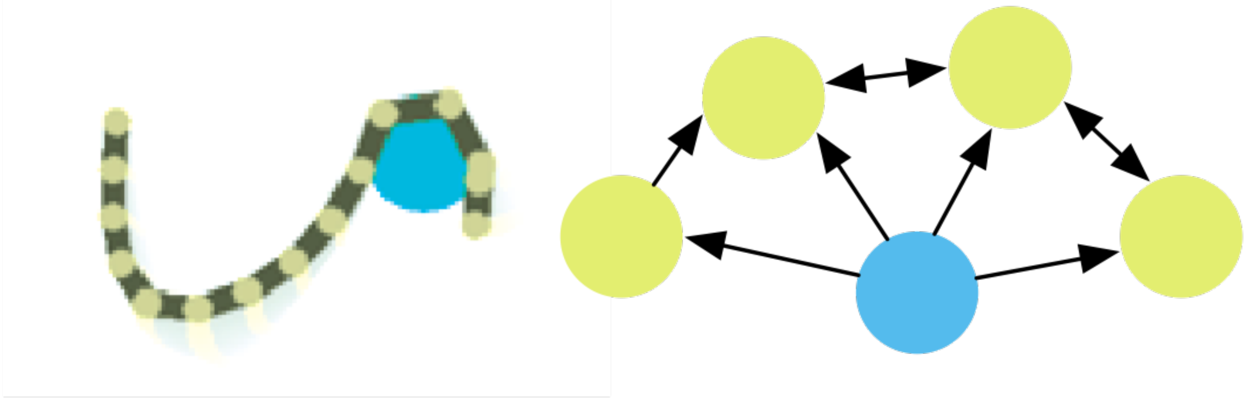


Graphs can capture many complex object/relation systems:

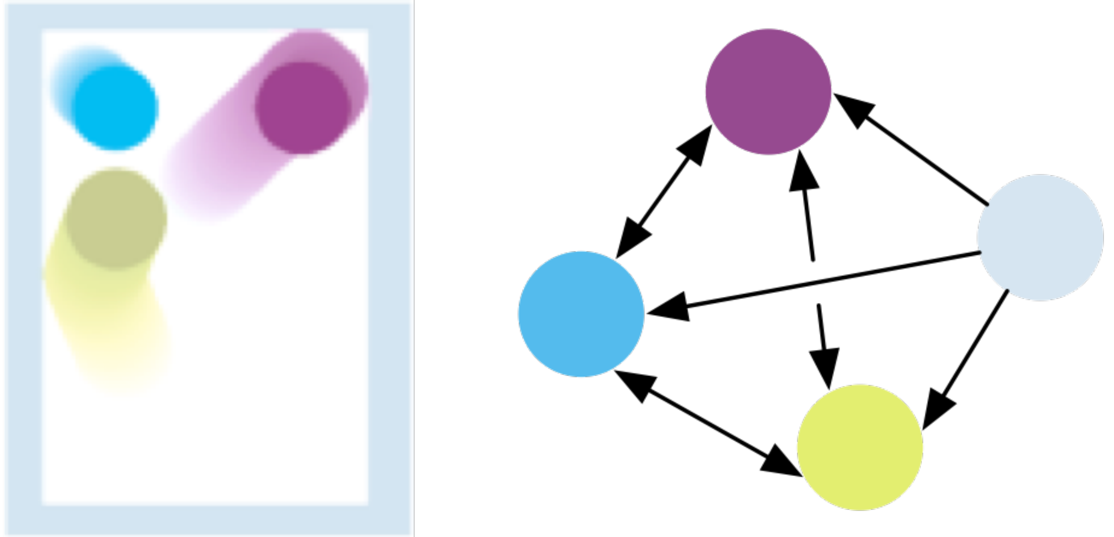
Molecule



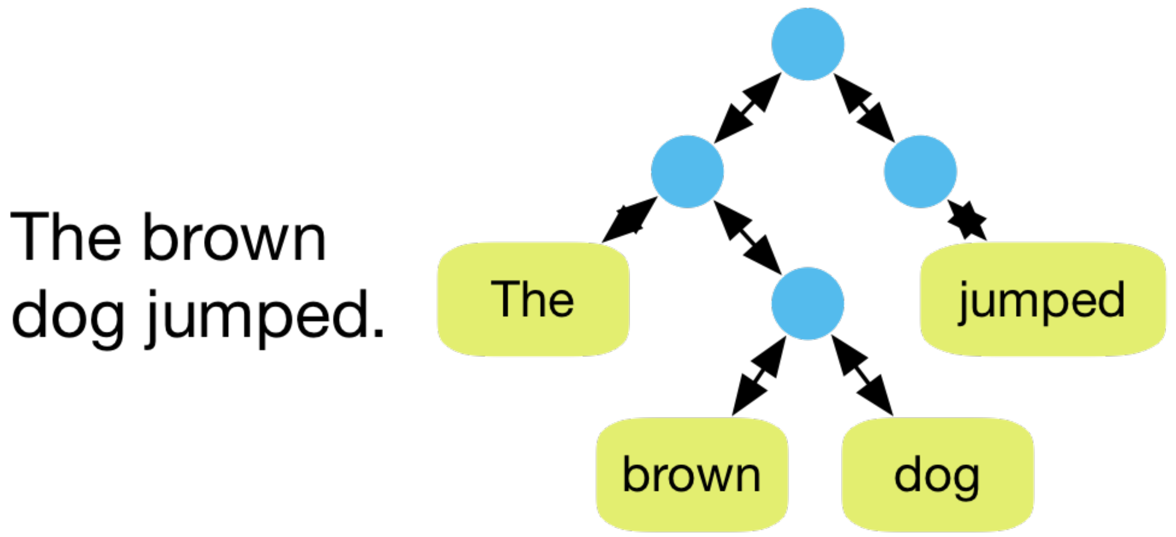
Mass-Spring System



Rigid Body System



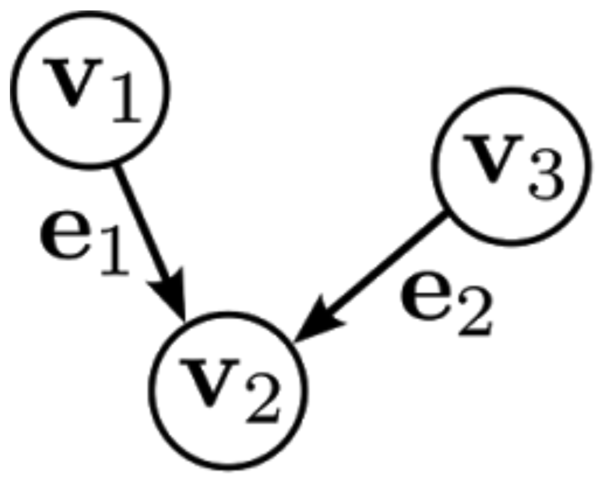
Sentence and Parse Tree



A lot of physical laws involve n-body where n>=2

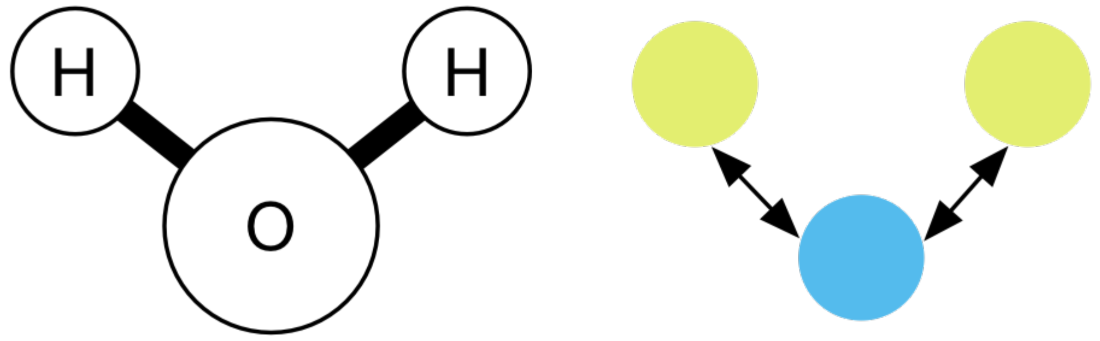
A **graph** is a natural way to represent entities and their relations:

- “Nodes" correspond to entities, objects, events, etc.
- "Edges" correspond to their relations, interactions, transitions, etc.
- Inferences about entities and relations respect the graphical structure.

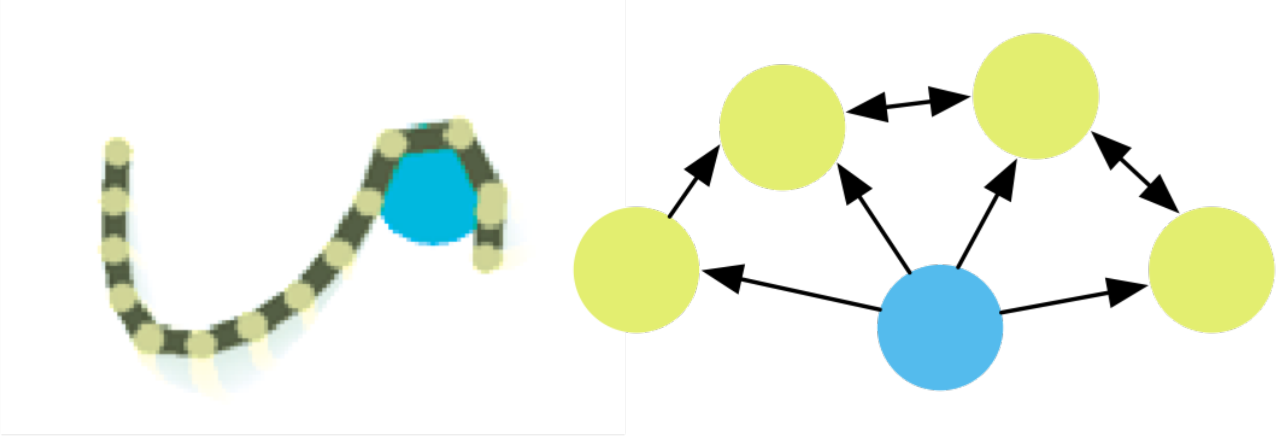


Graphs can capture many complex object/relation systems:

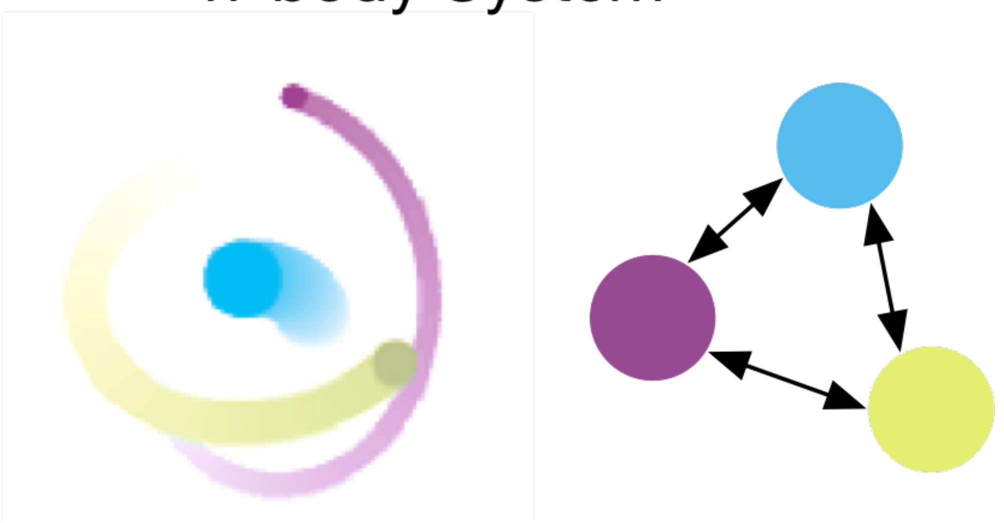
Molecule



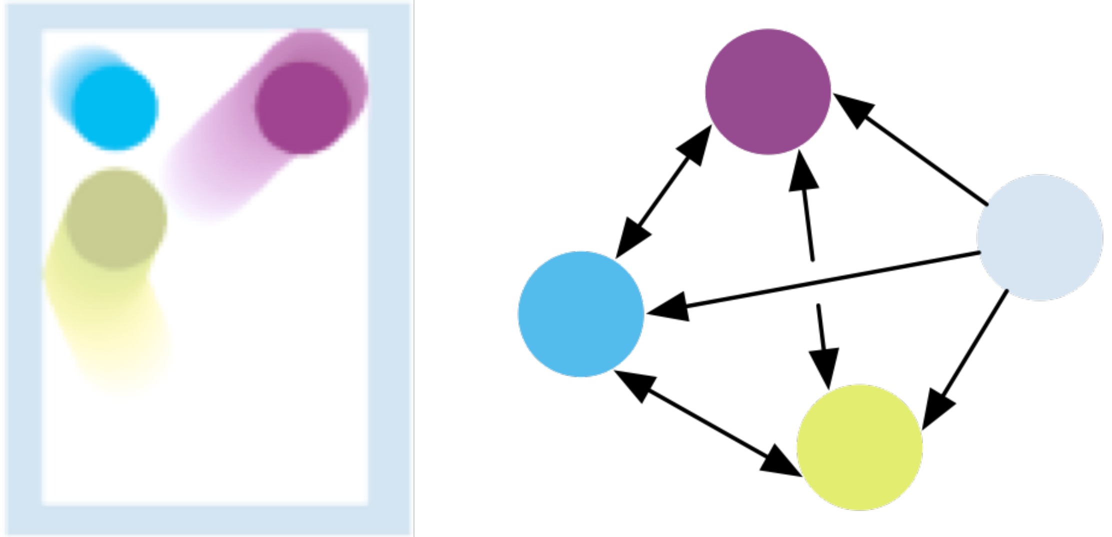
Mass-Spring System



n-body System



Rigid Body System



Sentence and Parse Tree

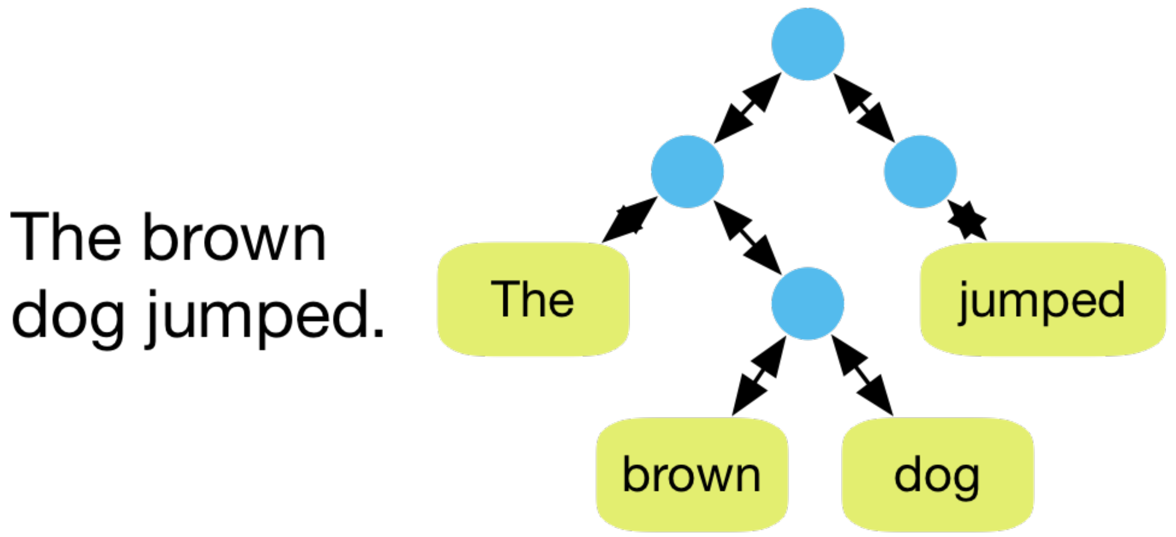
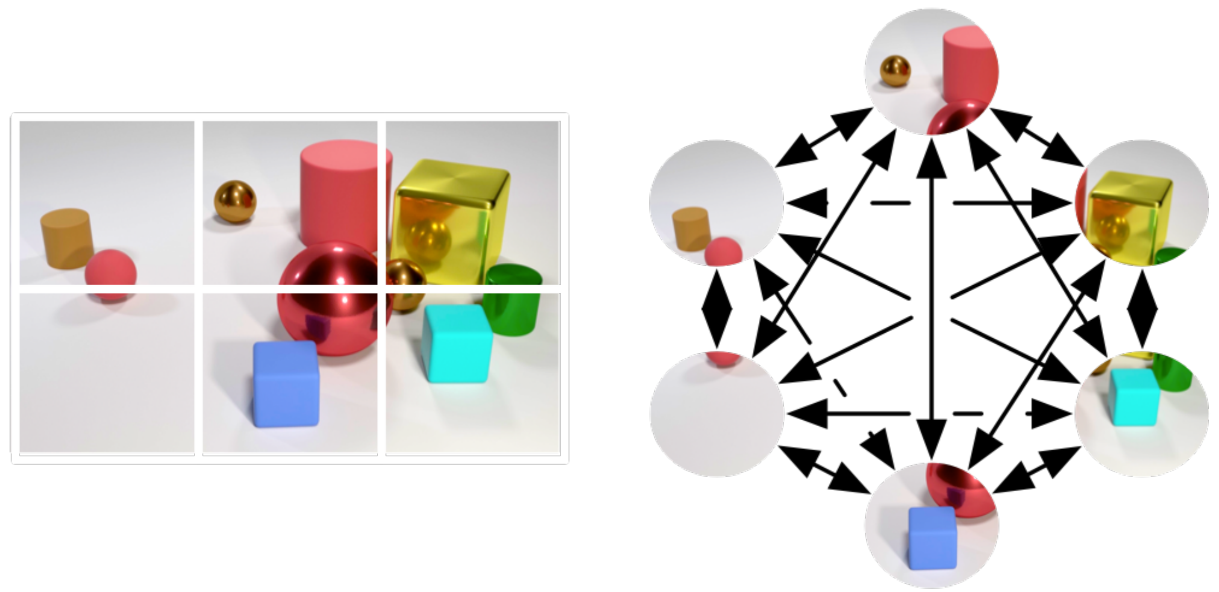
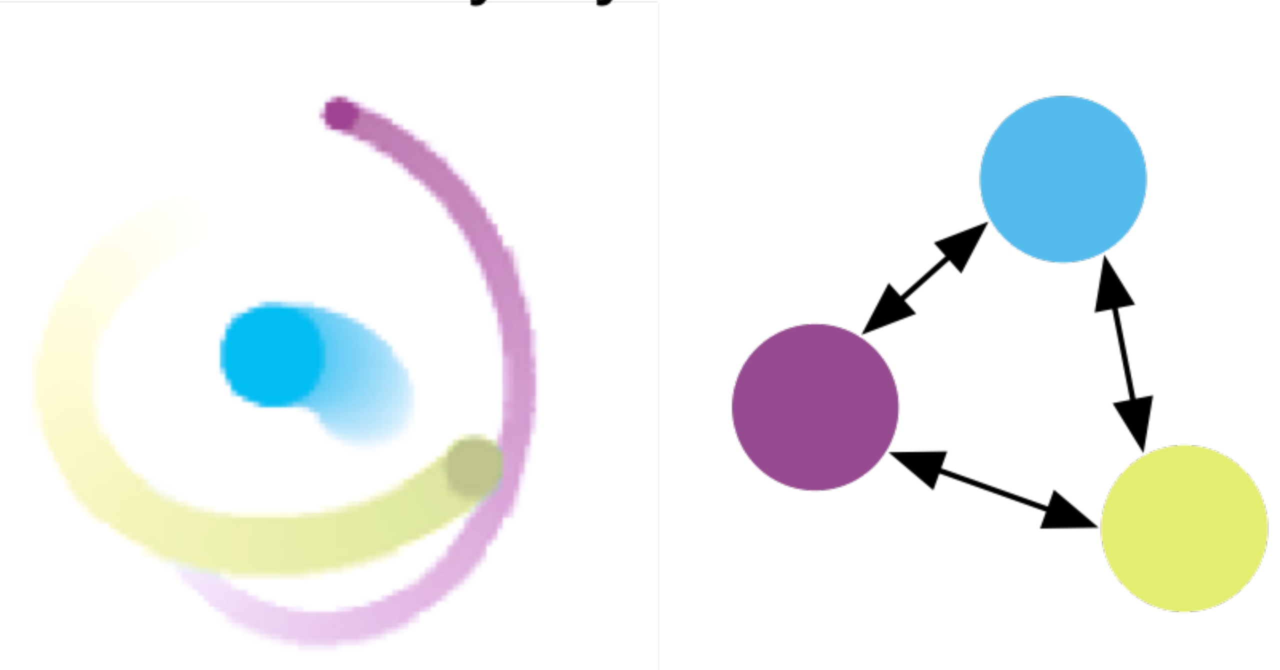


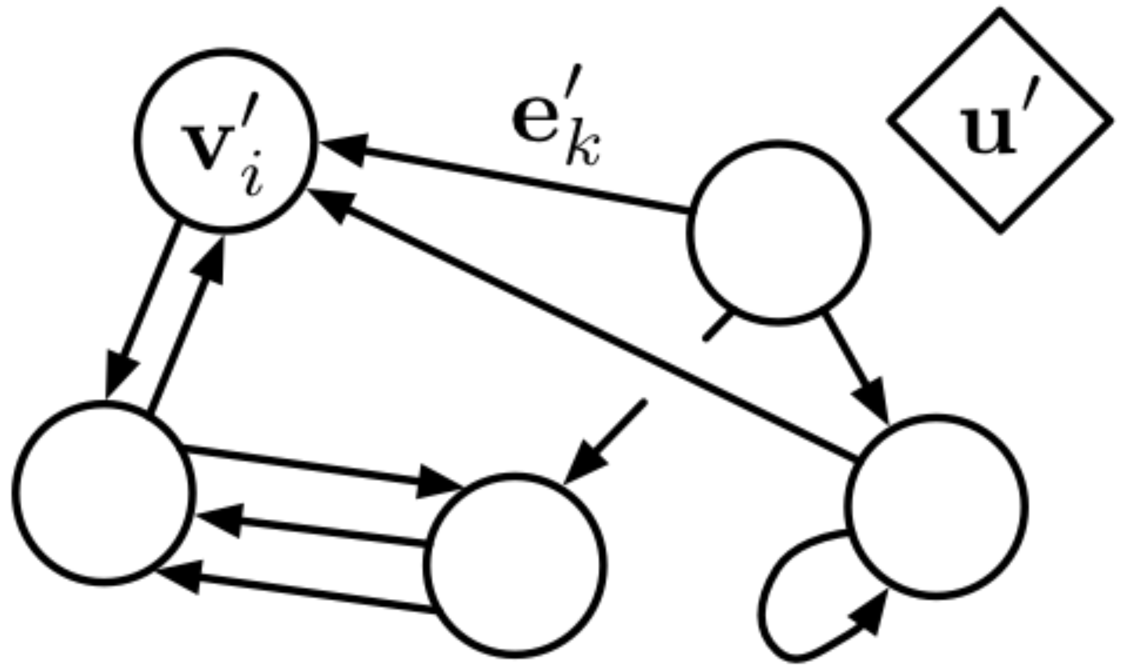
Image and Fully-Connected Scene Graph

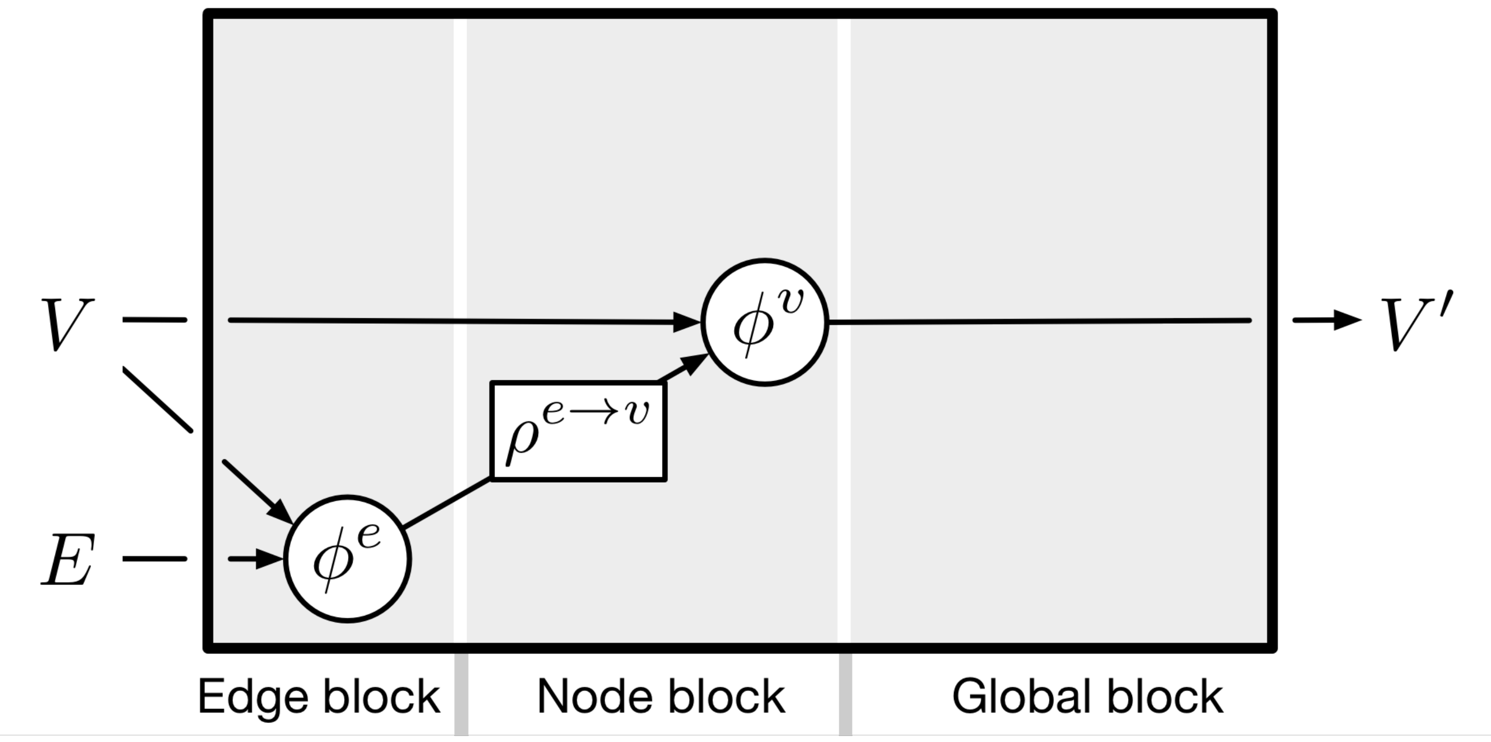


Introducing Graphical Network

n-body System





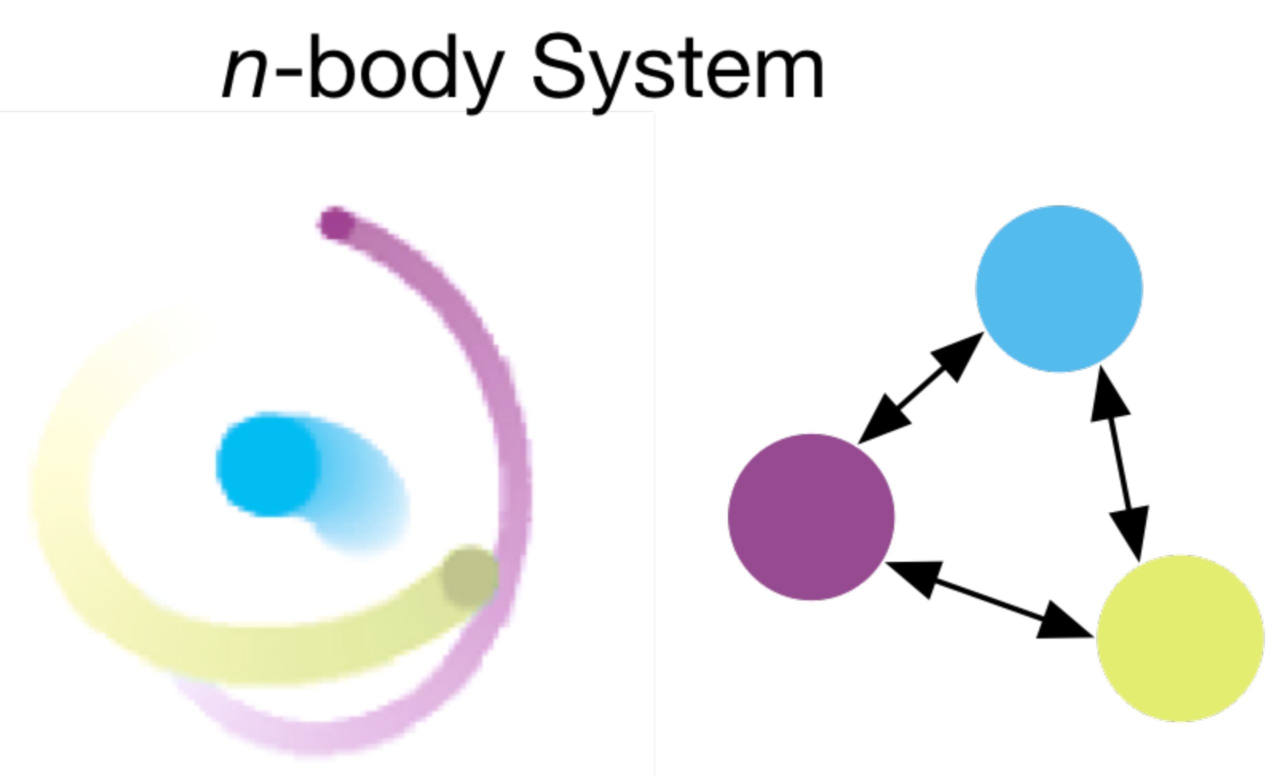


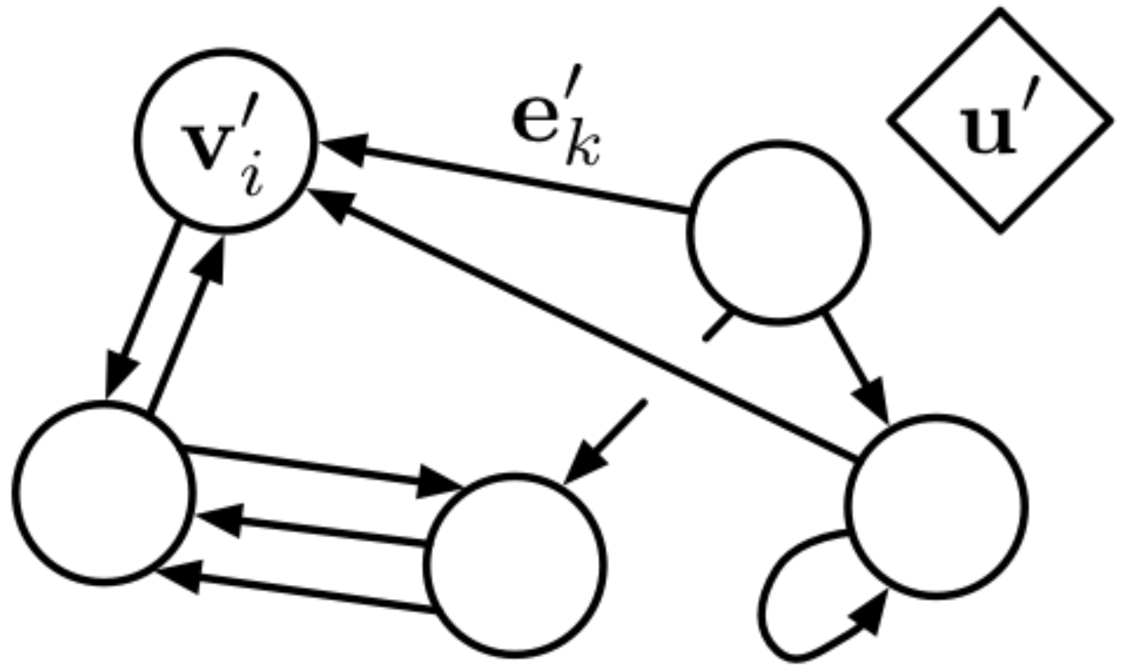
$\mathbf{u} \in \mathbb{R}^{L^u}$ is a global attribute vector of length L^u,

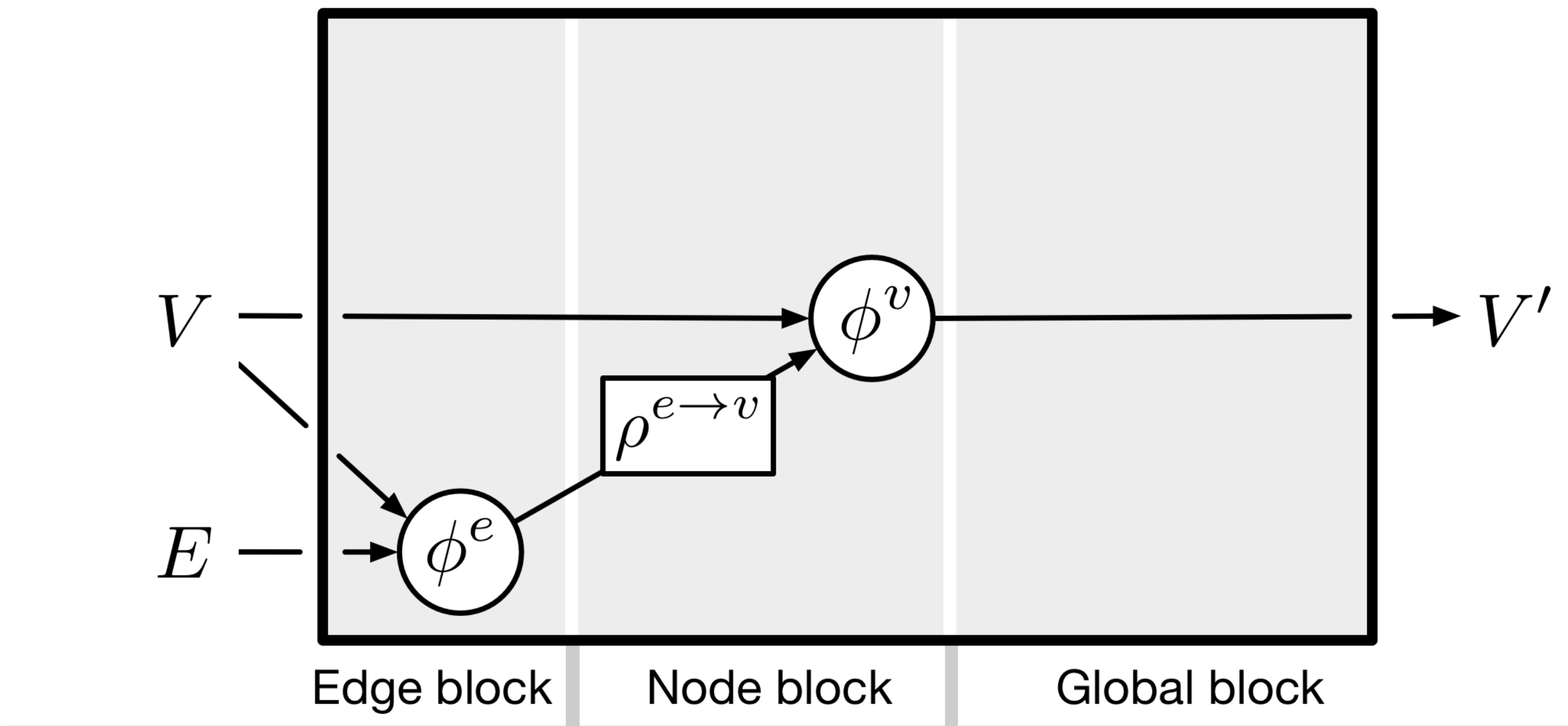
$V = \{\mathbf{v}_i\}_{i=1:N^v}$ is a set of node attribute vectors, $\mathbf{v}_i \in \mathbb{R}^{L^v}$ of length L^v, and

$E = \{(\mathbf{e}_k, r_k, s_k)\}_{k=1:N^e}$ is a set of edge attribute vectors, $\mathbf{e}_k \in \mathbb{R}^{L^e}$ of length L^e, and indices $r_k, s_k \in \{1{:}N^v\}$ of the "receiver" and "sender" nodes connected by the k-th edge.

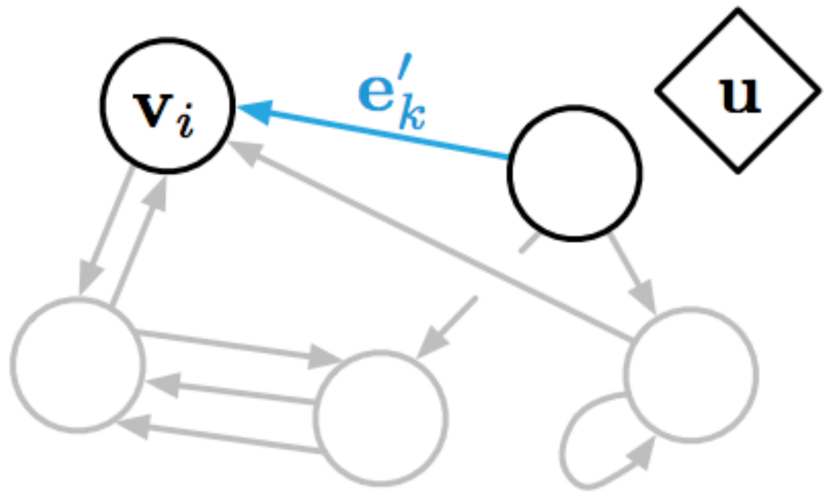
A Variant of Graphical Network: Interaction Network (Battaglia et al., 2016, NeurIPS)







$$\mathbf{e}_k' \leftarrow \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$$



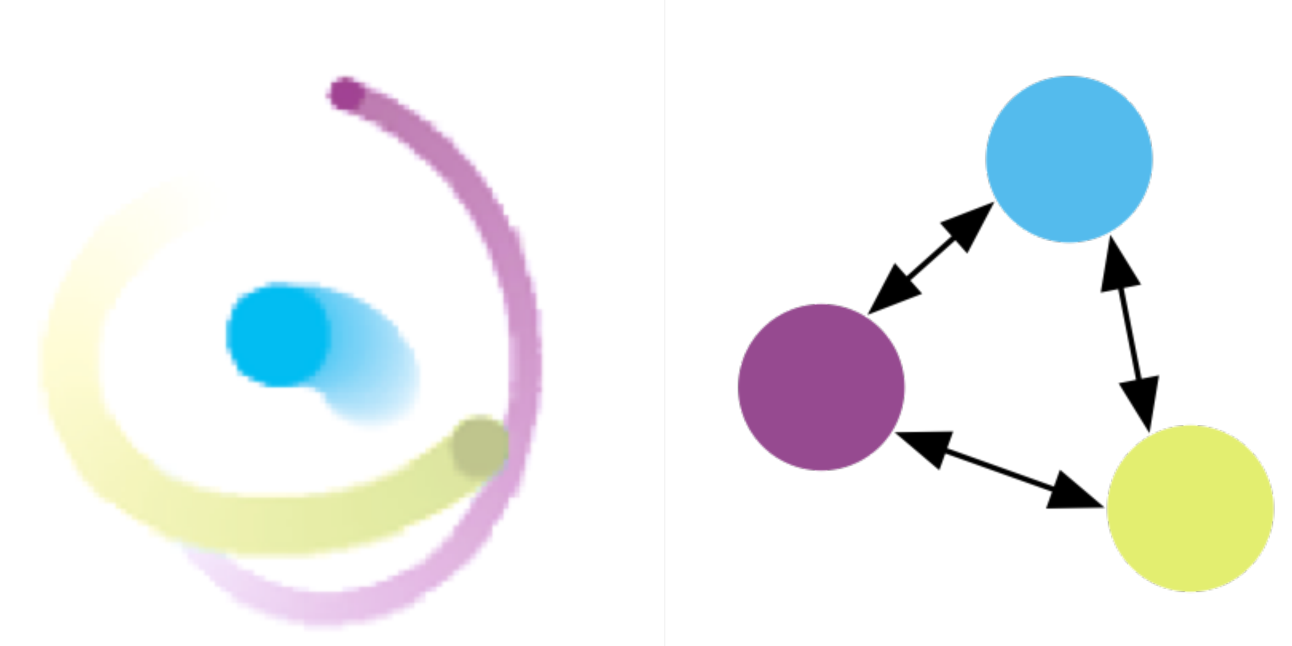
Edge function

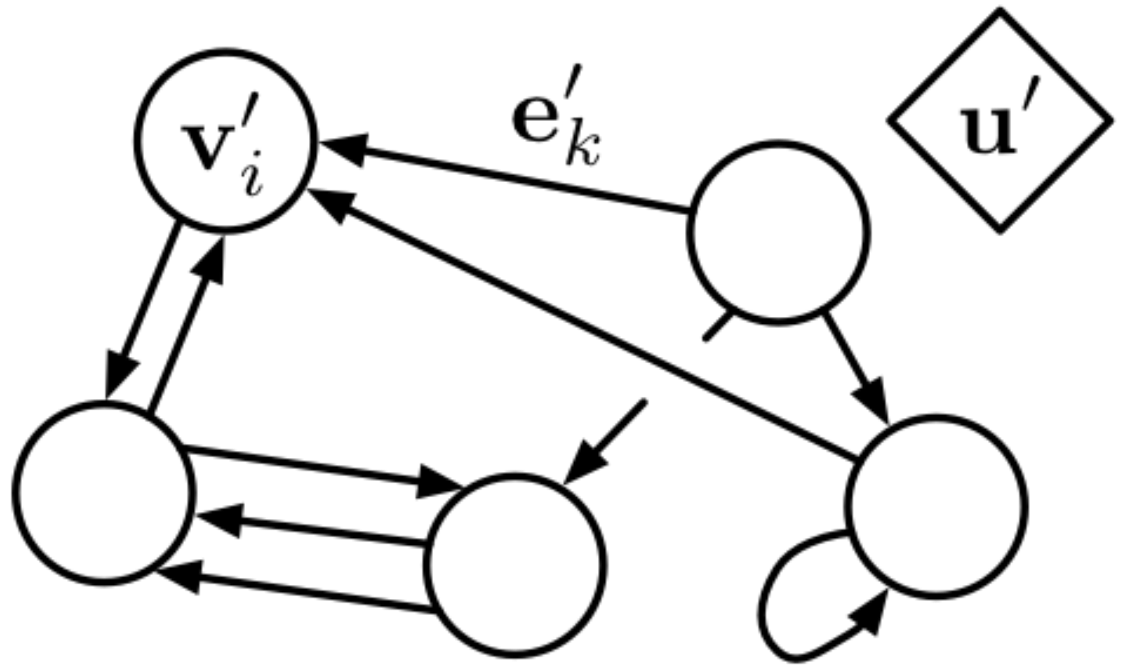
- Compute "message" from node and edge attributes associated with an edge

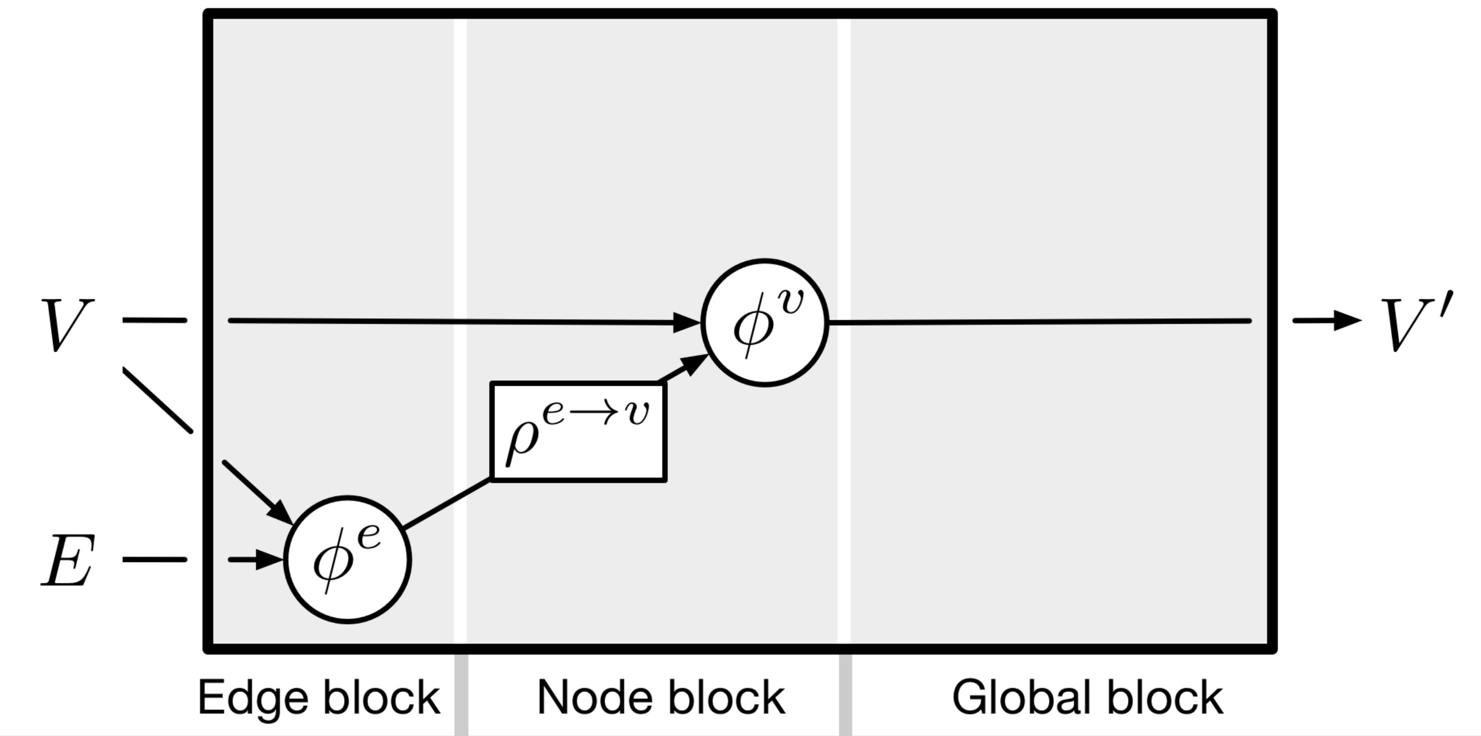
$\mathbf{u} \in \mathbb{R}^{L^u}$ is a global attribute vector of length L^u,
$V = \{\mathbf{v}_i\}_{i=1:N^v}$ is a set of node attribute vectors, $\mathbf{v}_i \in \mathbb{R}^{L^v}$ of length L^v, and
$E = \{(\mathbf{e}_k, r_k, s_k)\}_{k=1:N^e}$ is a set of edge attribute vectors, $\mathbf{e}_k \in \mathbb{R}^{L^e}$ of length L^e, and indices $r_k, s_k \in \{1{:}N^v\}$ of the "receiver" and "sender" nodes connected by the k-th edge.

A Variant of Graphical Network: Interaction Network (Battaglia et al., 2016, NeurIPS)

n-body System



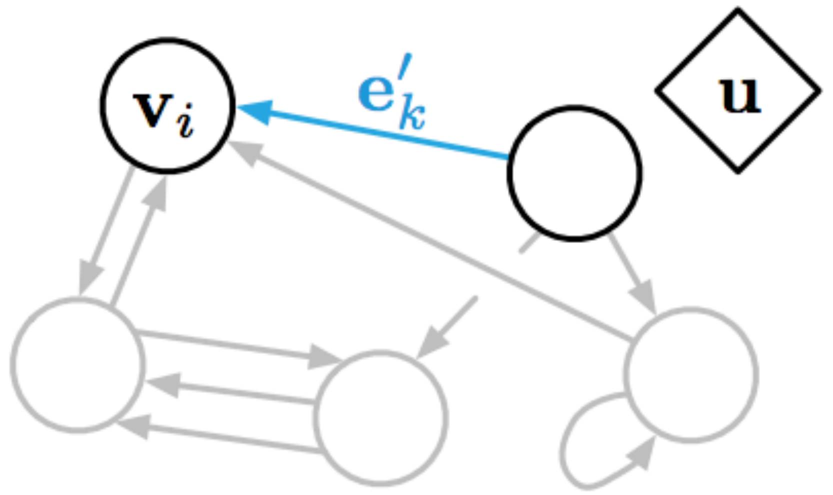


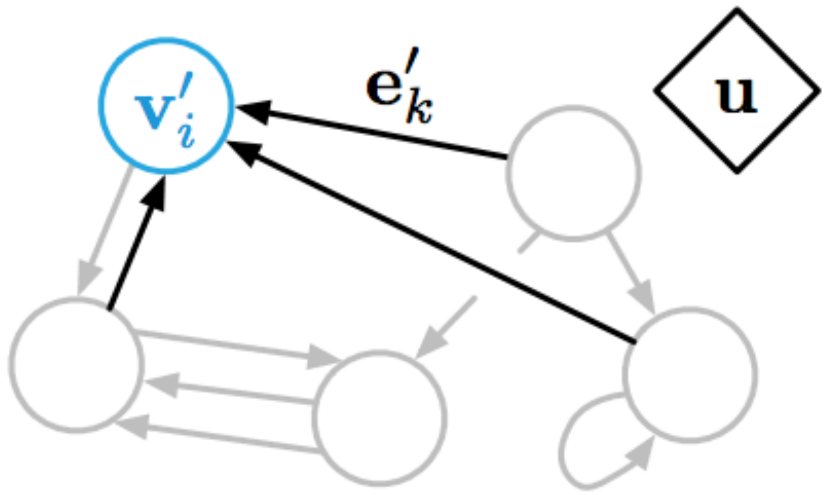


$\mathbf{e}'_k \leftarrow \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$

$\bar{\mathbf{e}}'_i \leftarrow \rho^{e\rightarrow v}\left(E'_i\right)$

$\mathbf{v}'_i \leftarrow \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right)$





Edge function

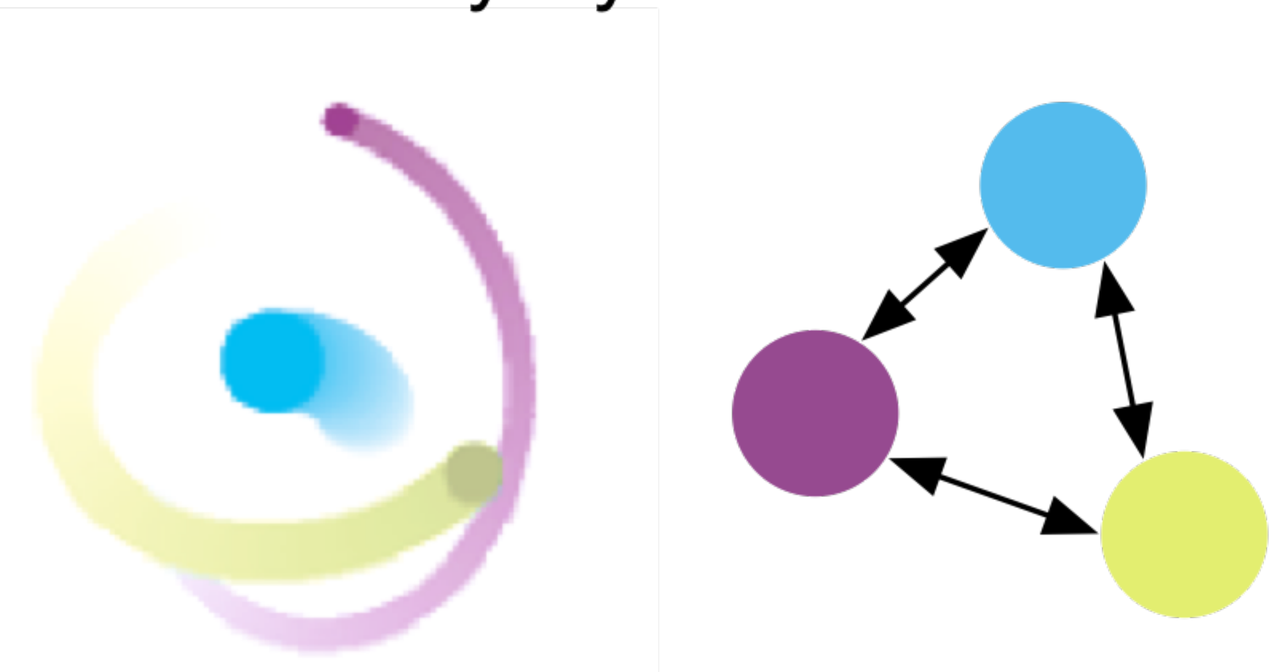
- Compute "message" from node and edge attributes associated with an edge

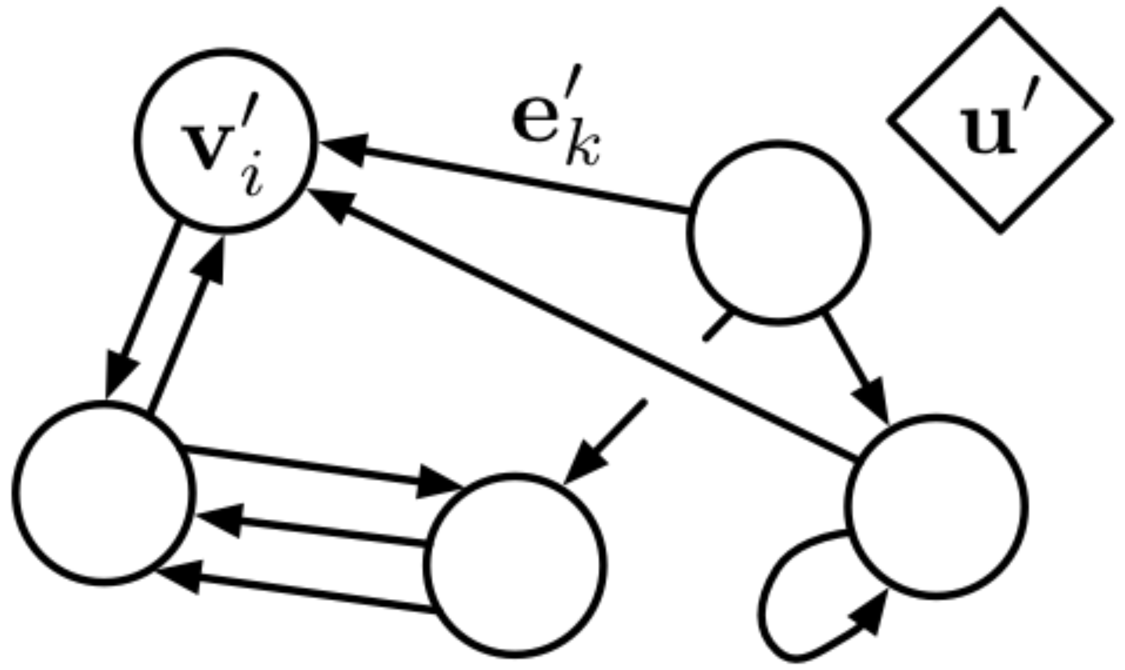
Node function

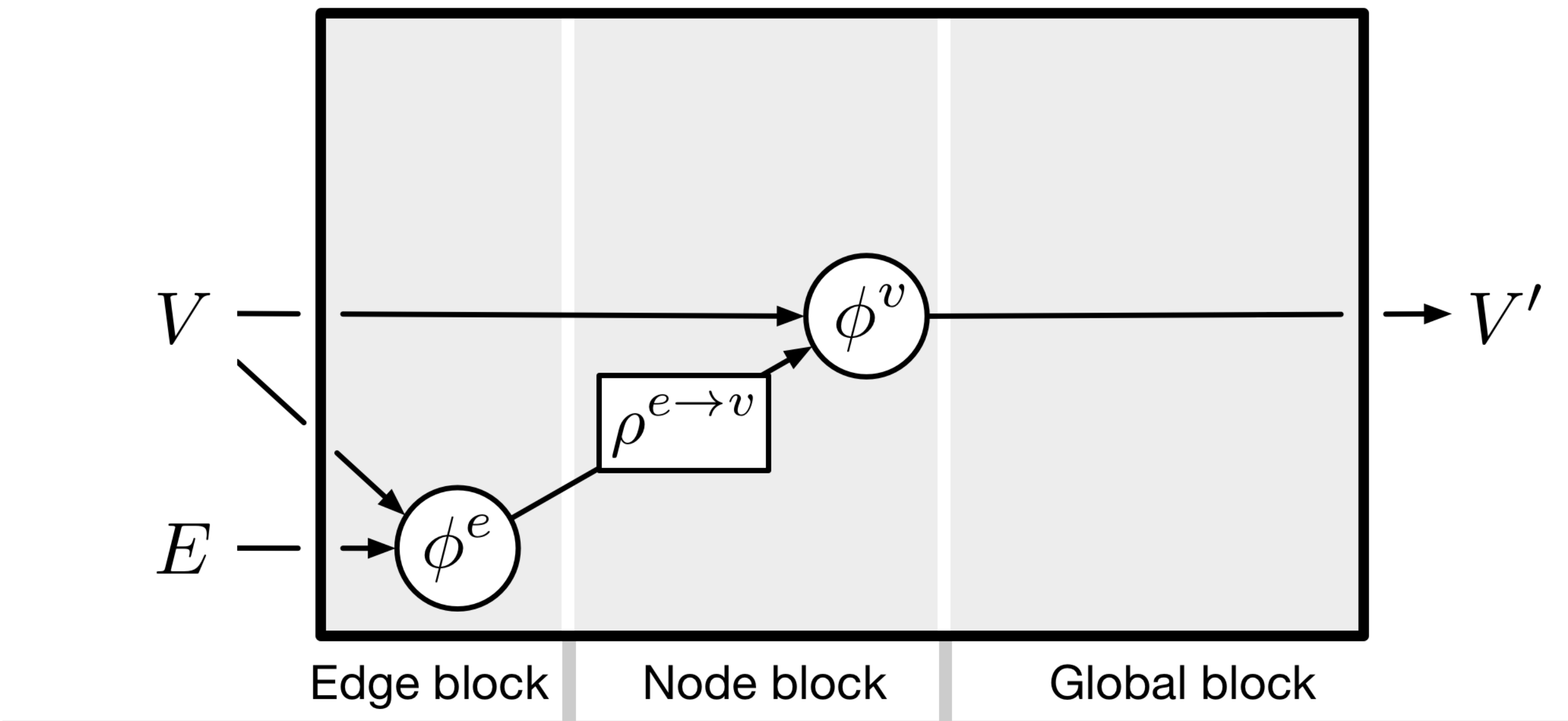
- Update node info from previous node state and aggregated "messages"

A Variant of Graphical Network: Interaction Network (Battaglia et al., 2016, NeurIPS)

n-body System



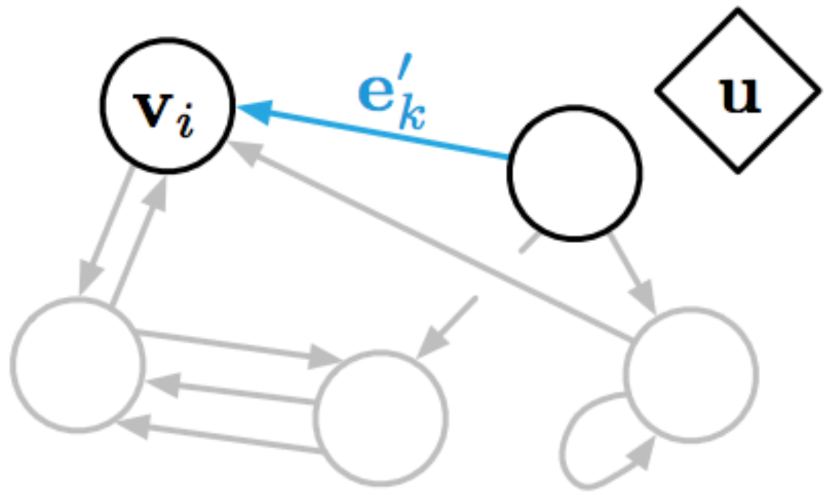


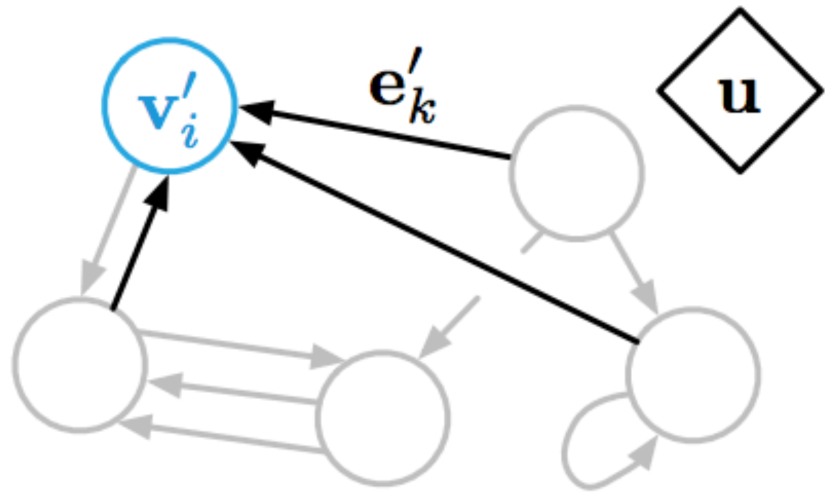


$\mathbf{e}'_k \leftarrow \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$

$\bar{\mathbf{e}}'_i \leftarrow \rho^{e\to v}\left(E'_i\right)$

$\mathbf{v}'_i \leftarrow \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right)$





Edge function

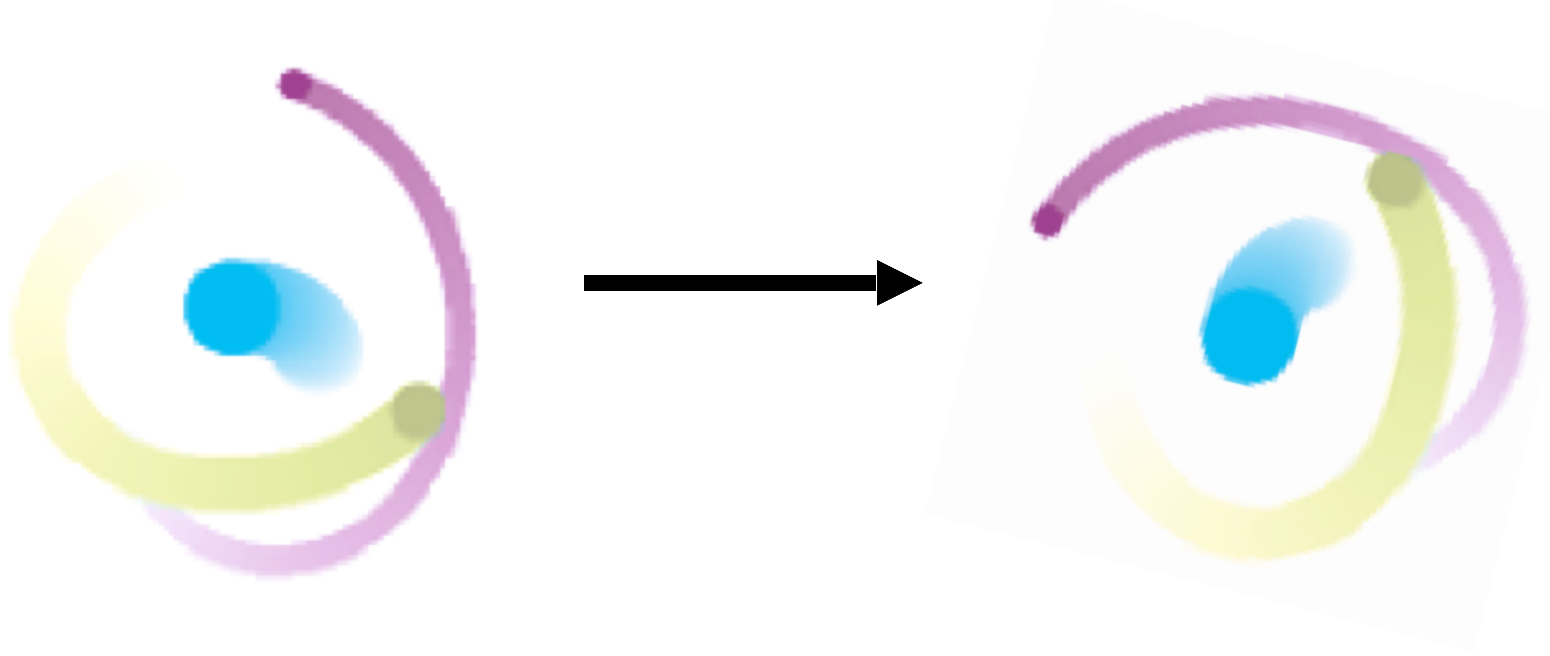
- Compute "message" from node and edge attributes associated with an edge

Node function

- Update node info from previous node state and aggregated "messages"

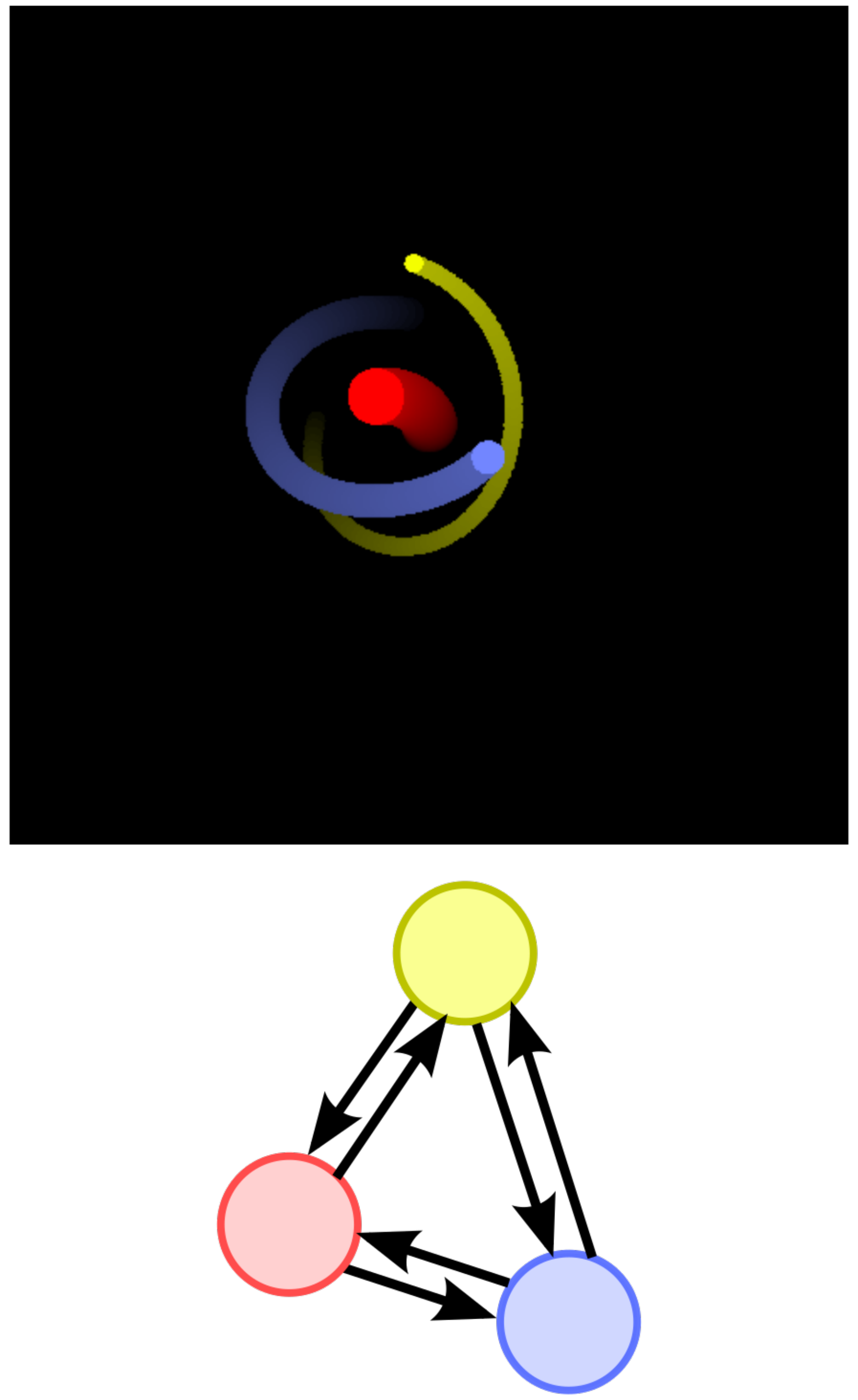
Trained to predict node states at t_1 from states at t_0

t_0 t_1



What we are doing today? Learn to simulate and find the force laws of the following systems

n-body

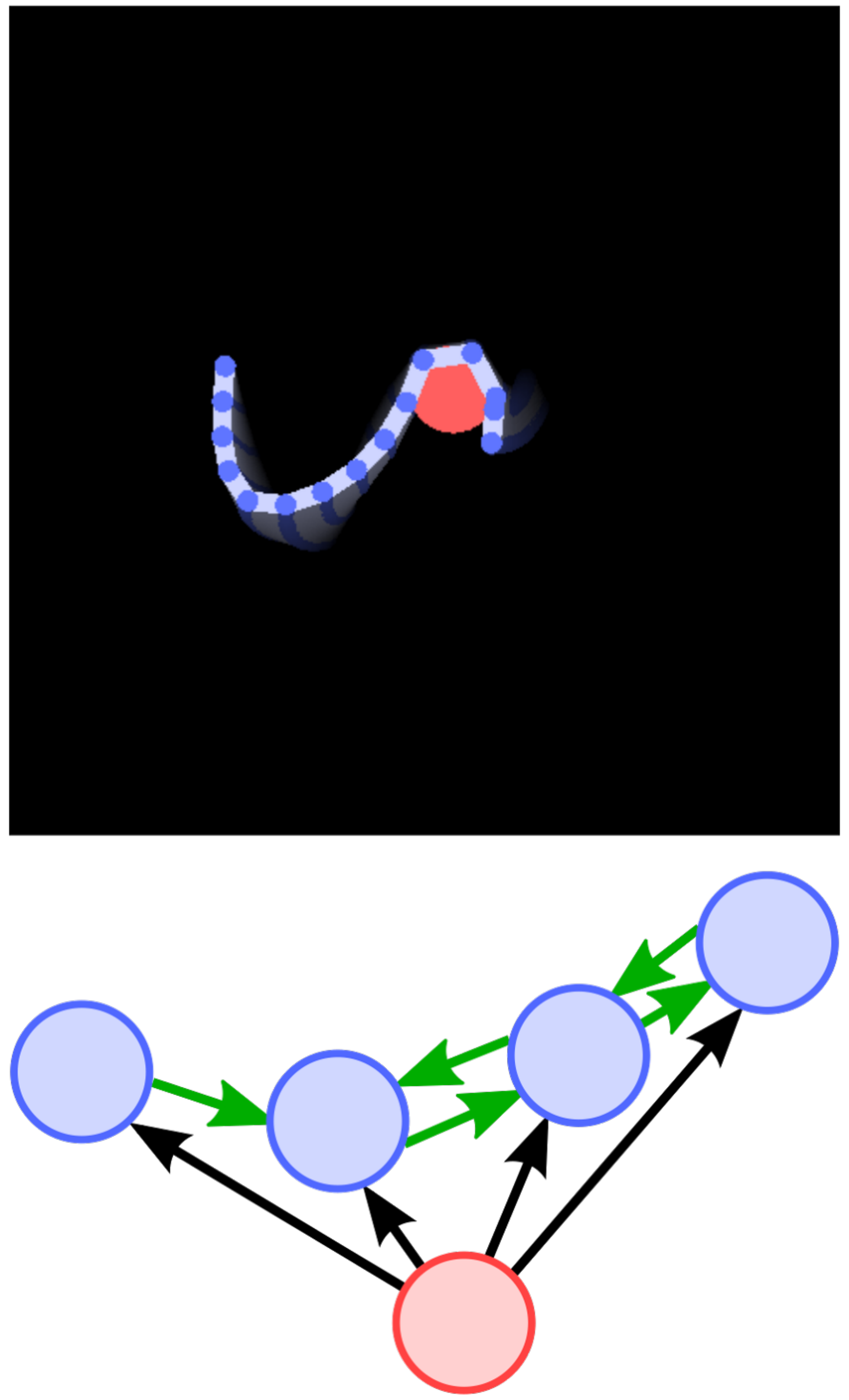


Edges: gravitational forces

Setup:

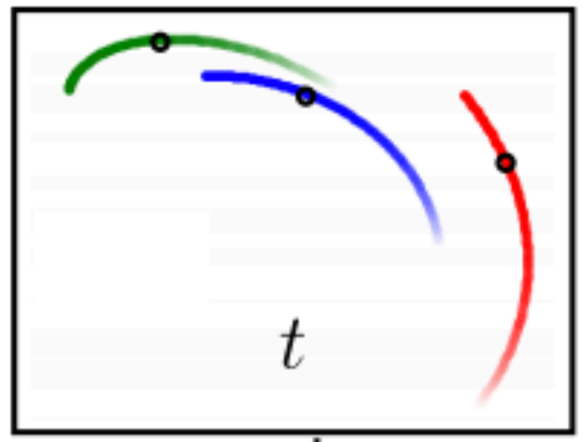
- 1/r, 1/(r^2) force in 2D for 3-body
- 1/(r^2) force in 3D for 3-body
- string with 1/(r^2) force in 2D
- 100,000 simulations each
- 1000 time-steps each

String



Edges: springs and rigid collisions

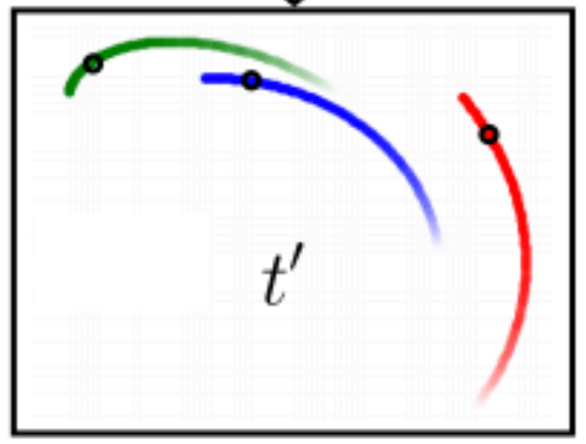
Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS



Input : **A single time-step of 3 planets interacting with each other at t**

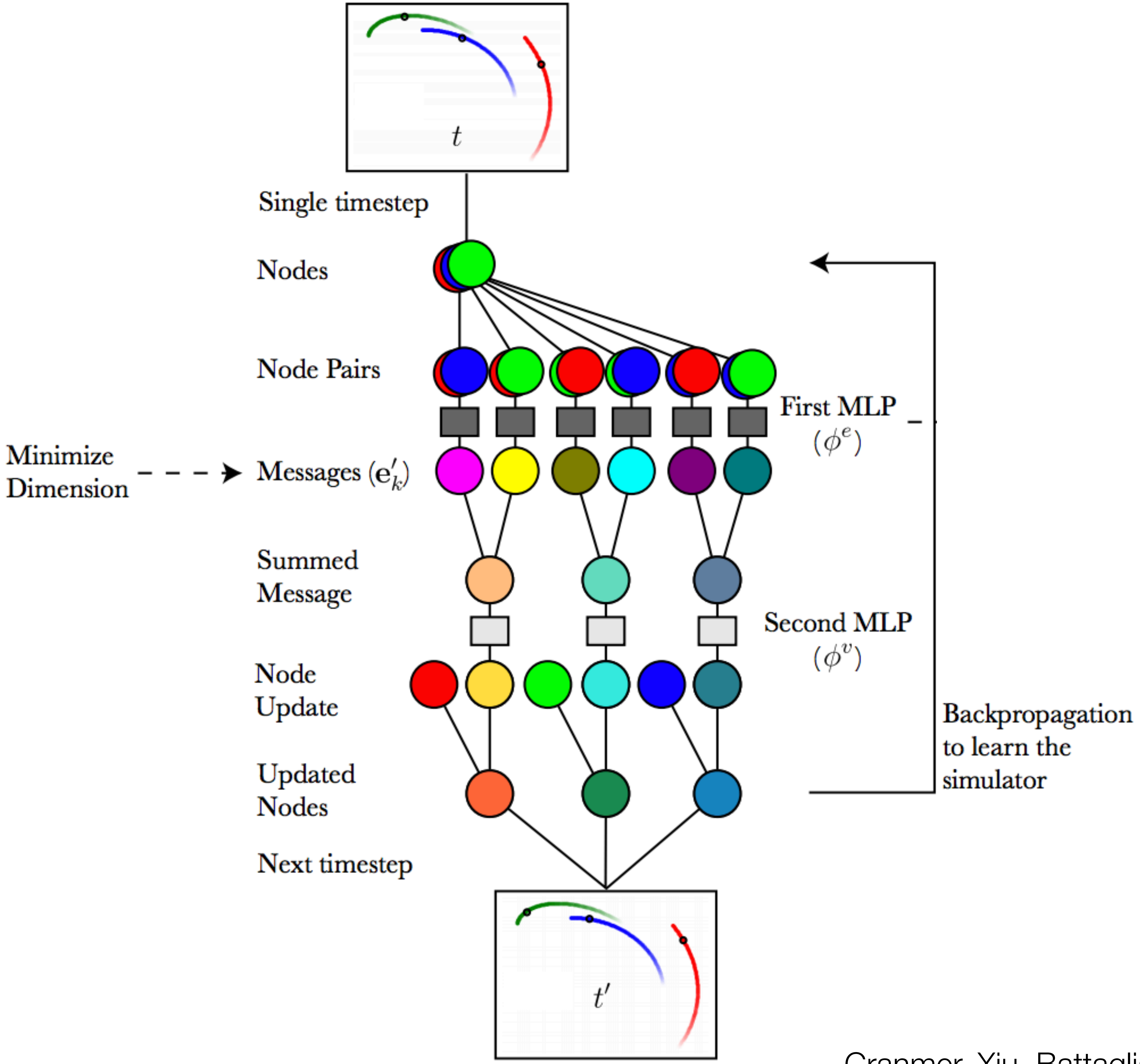
Single timestep

Machine Learning Model

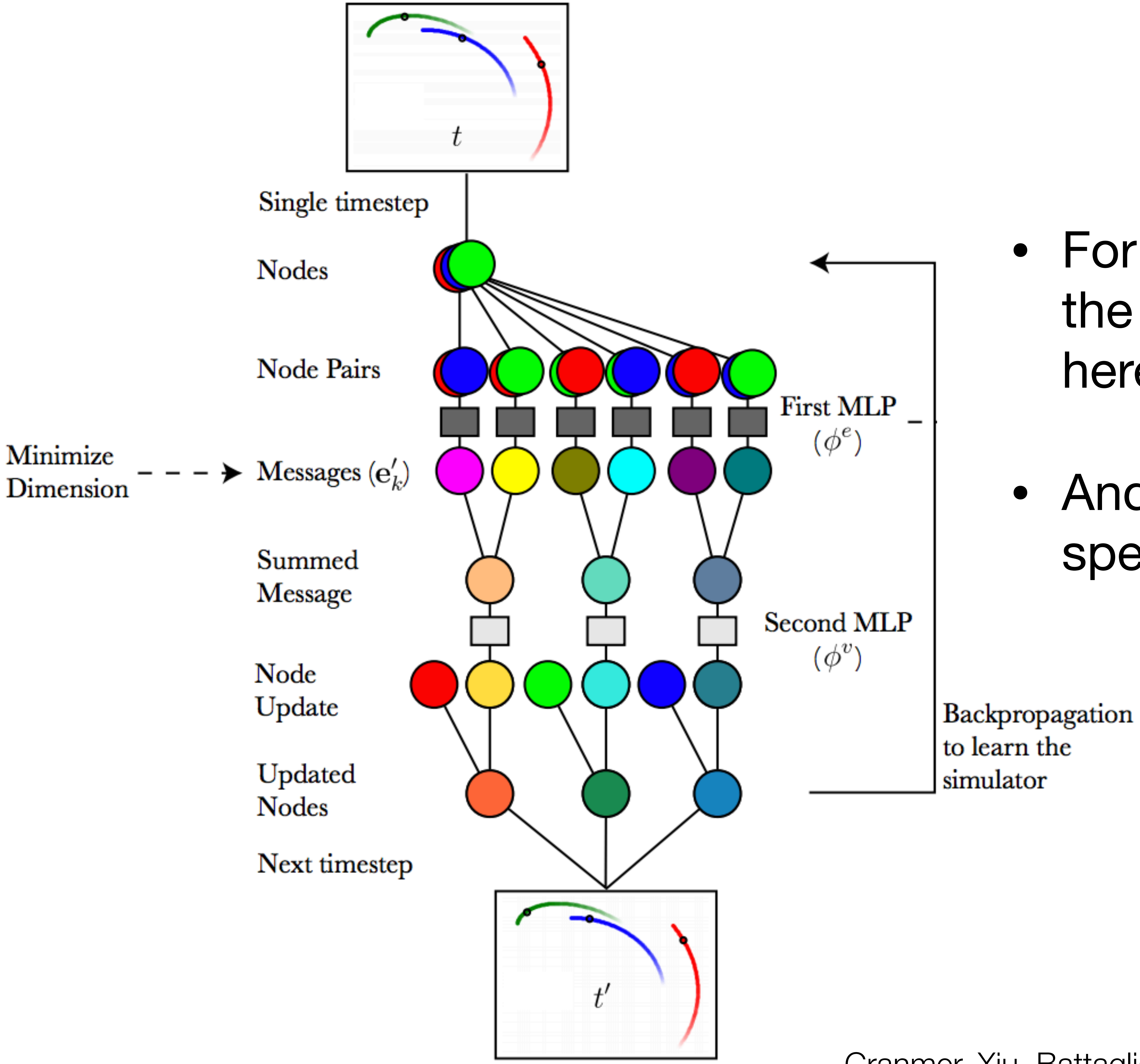


Output: **A single time-step of 3 planets interacting with each other at t'**

Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS



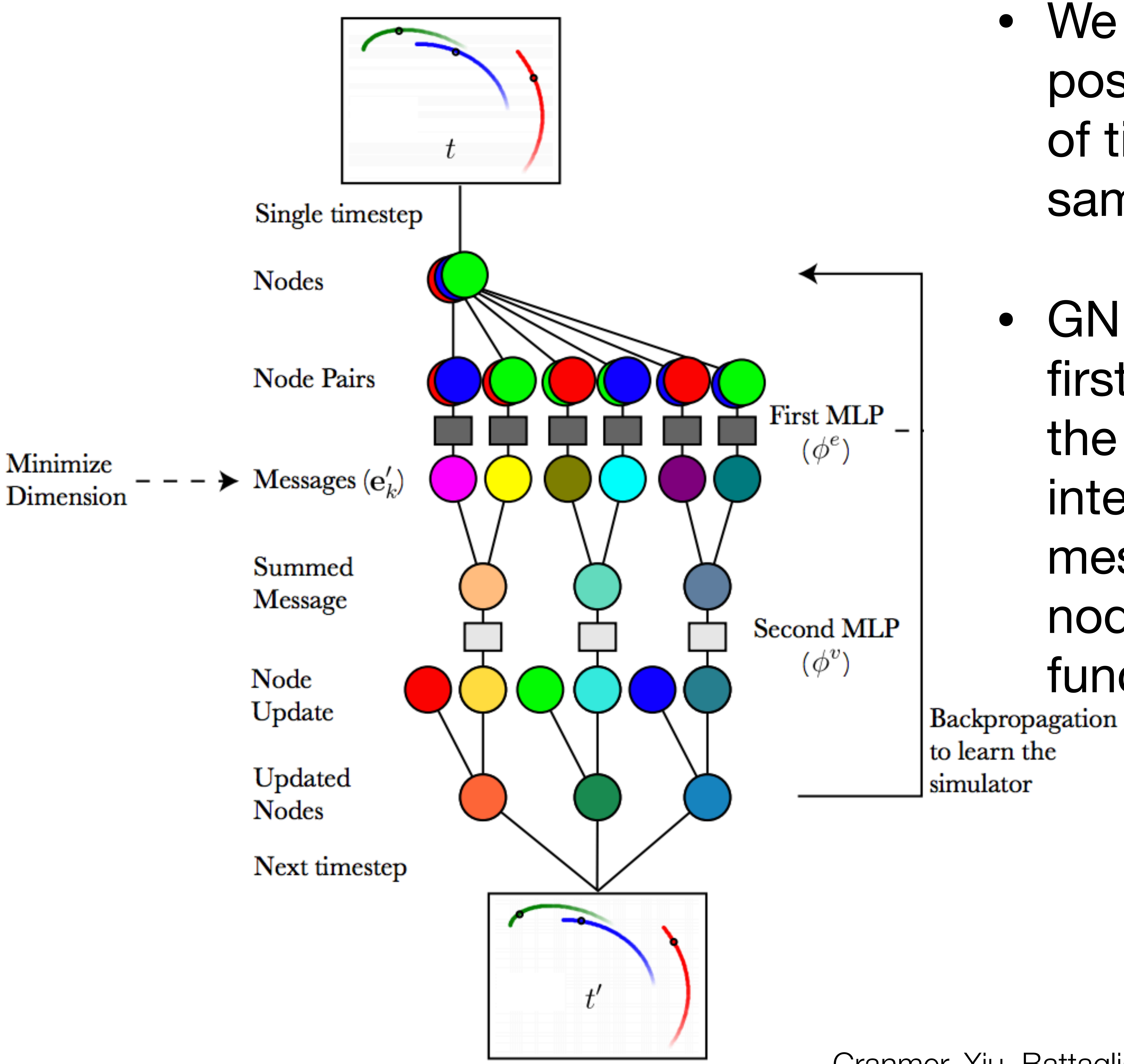
Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS



- For simplicity, we skip the global property here

- And the edges have no special attributes

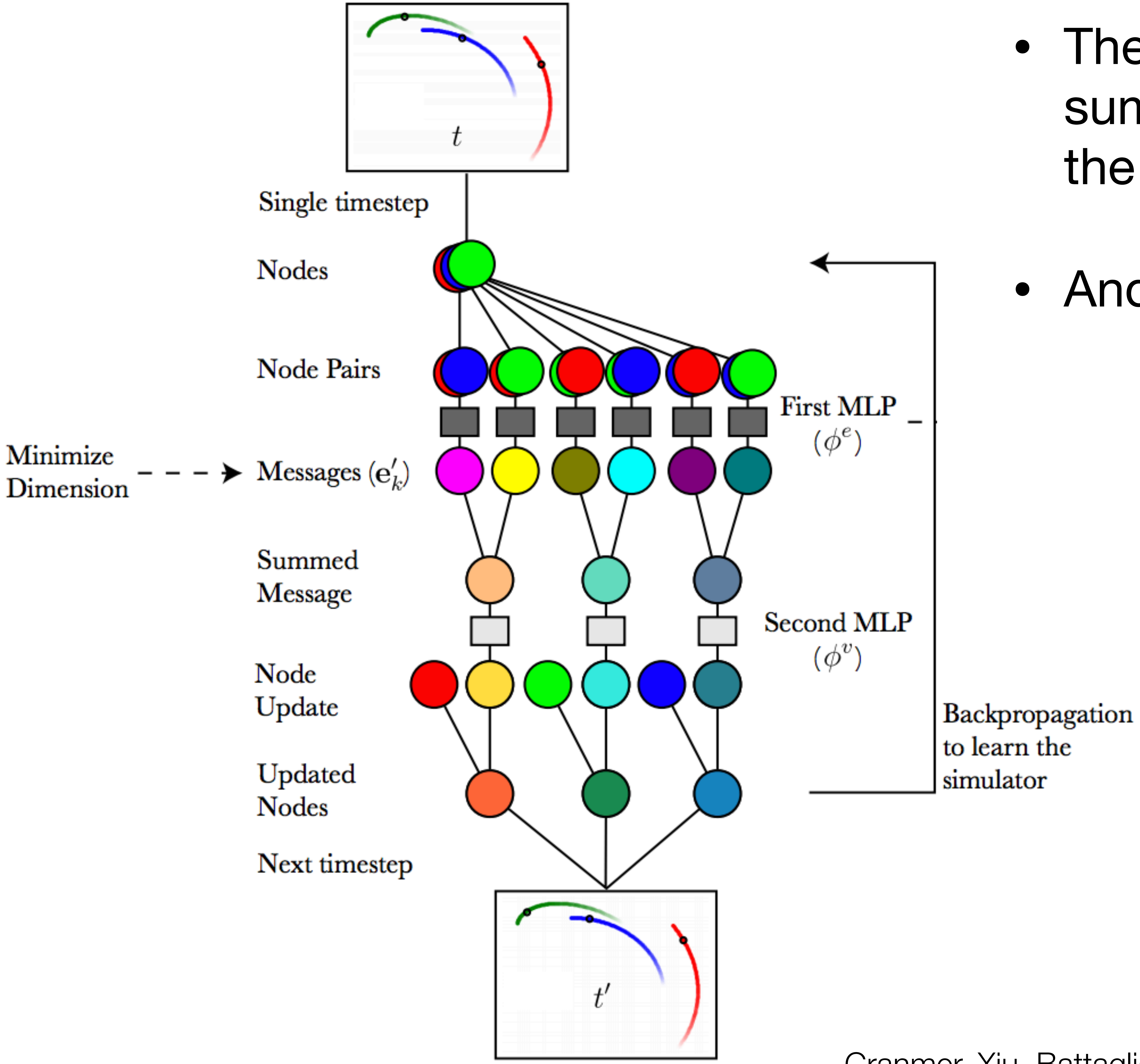
Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS



- We have nodes (with position as a function of time, they are all of same mass)

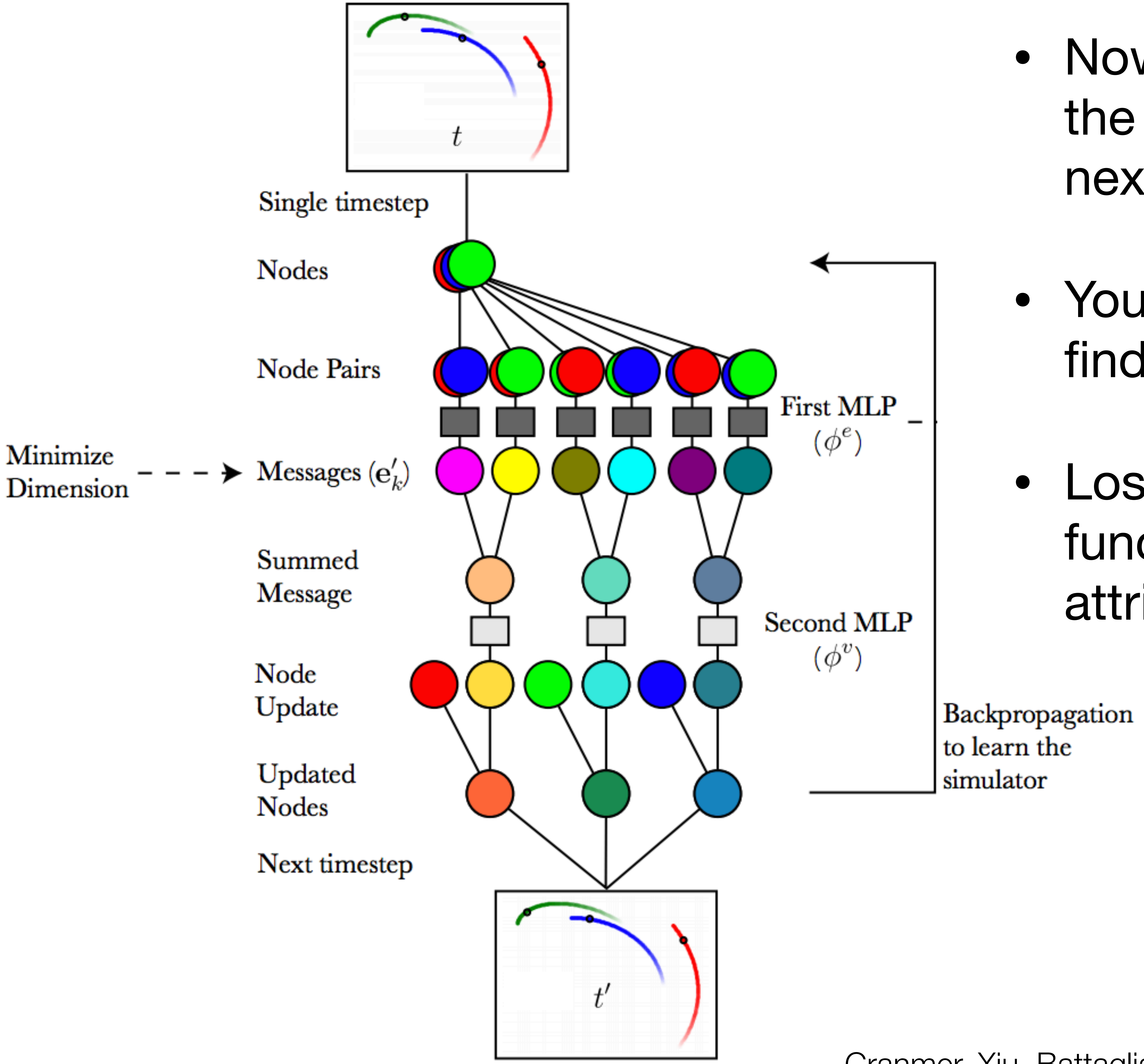
- GN process the graphs first by computing all the pair-wise interactions (aka. messages) between nodes, with a message function.

Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS



- Then we calculate the summed messages on the incident node
- And update the node

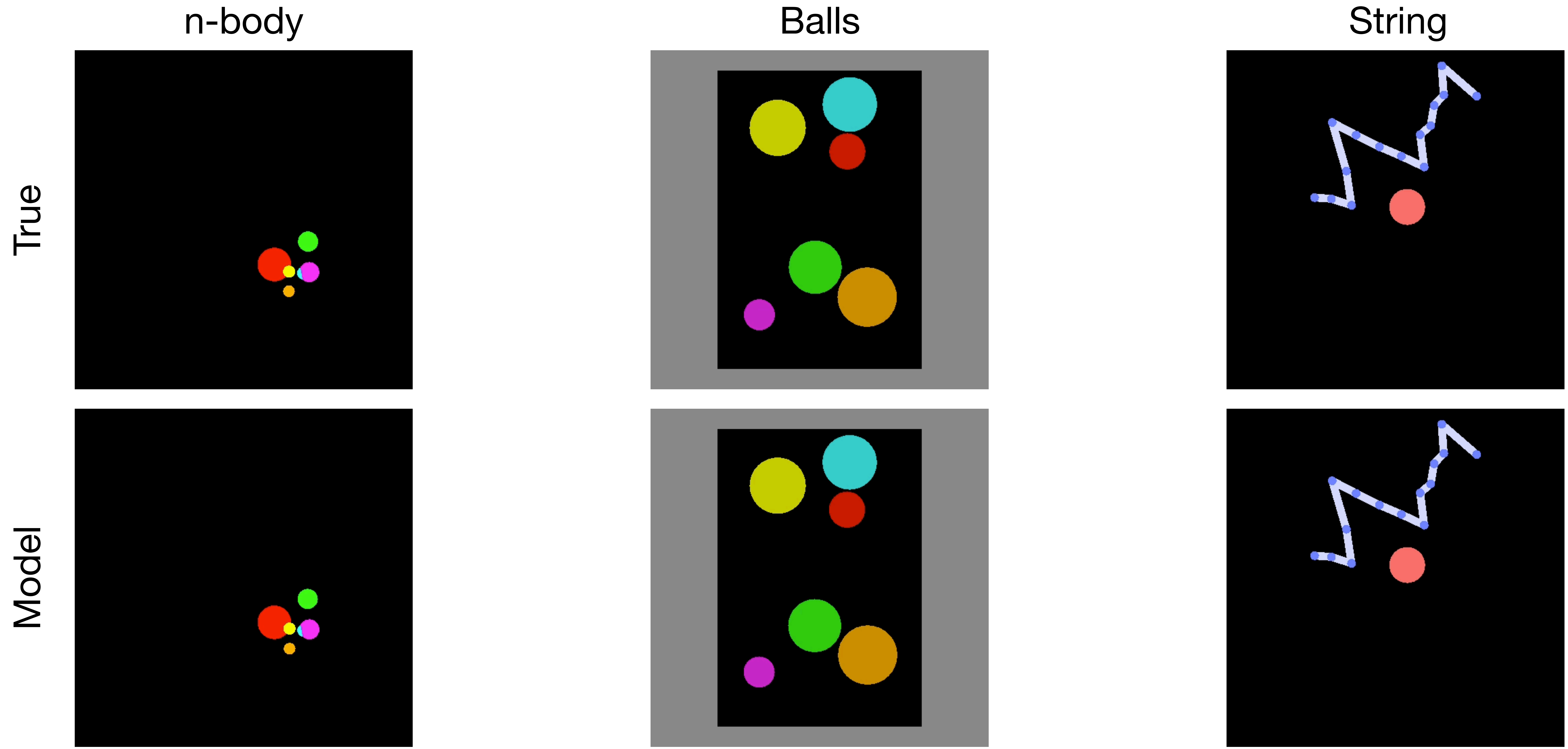
Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS



- Now you can predict the node attribute of next time-step
- You backpropagate to find the best weights
- Loss function: a function of the node attributes

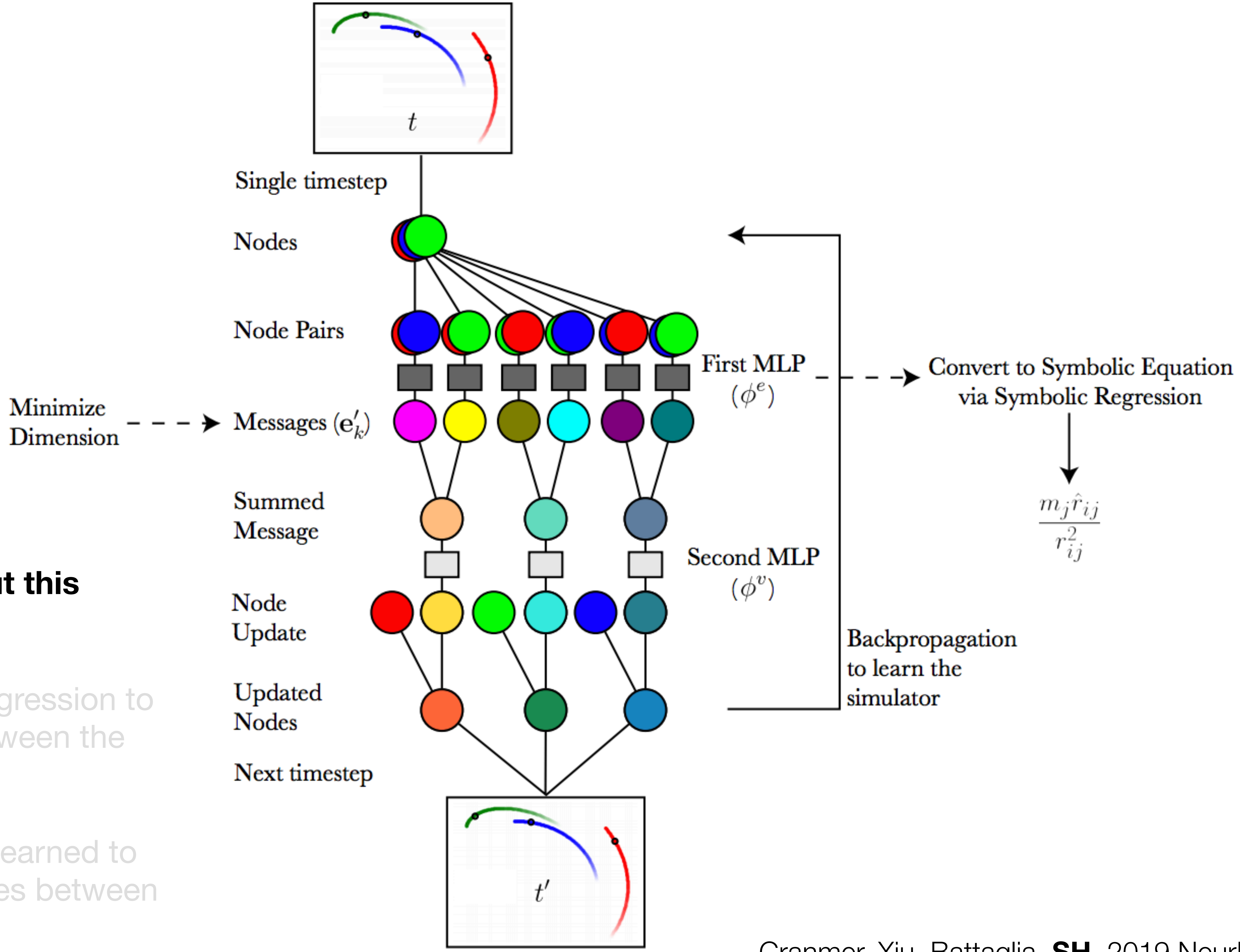
Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS

Outcome? We are able to predict the next steps!



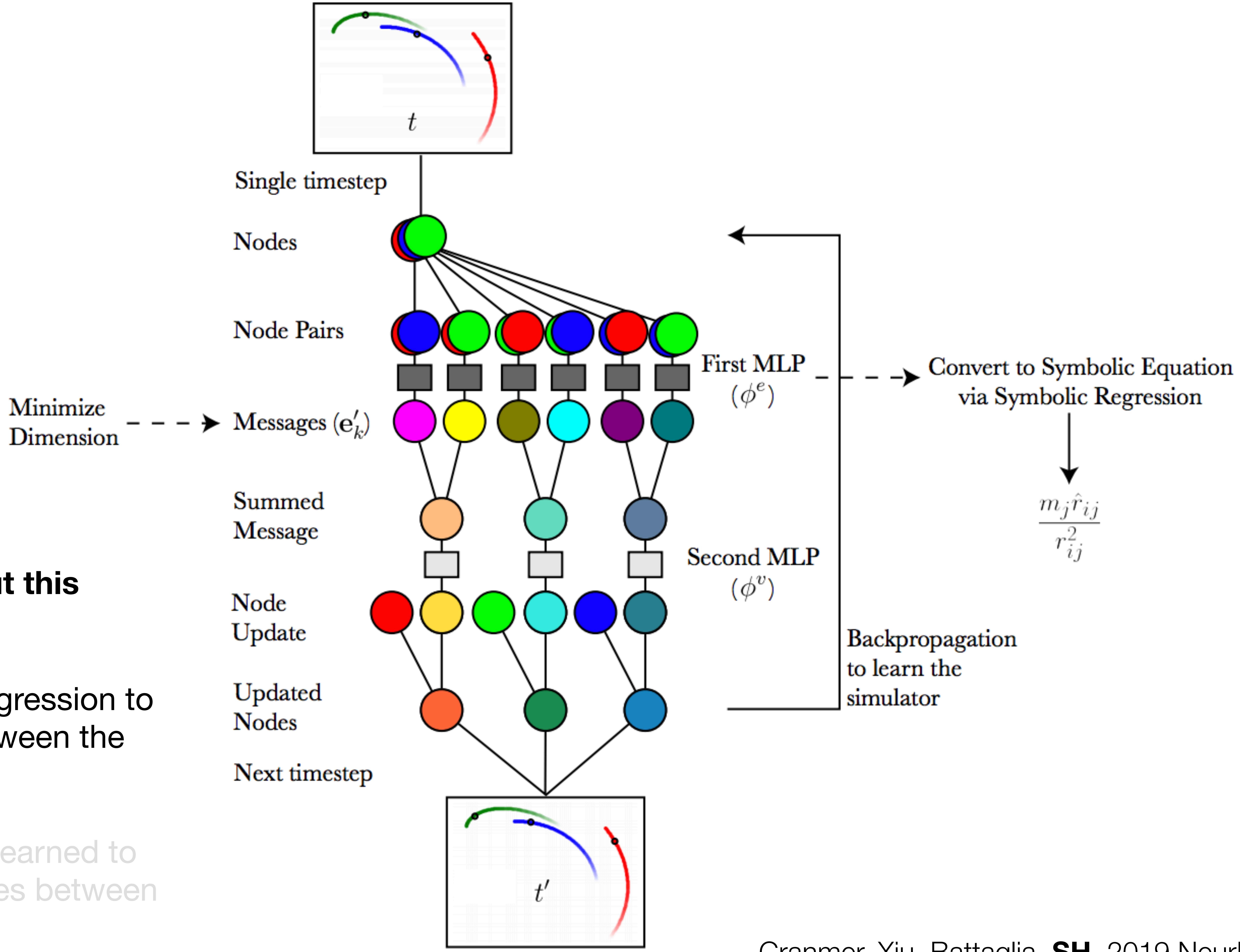
Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS
Battaglia et al., 2016, NeurIPS

- **What is more about this paper?**
- We use symbolic regression to learn the forces between the nodes.
- Namely, e_k' can be learned to be the relevant forces between the nodes!

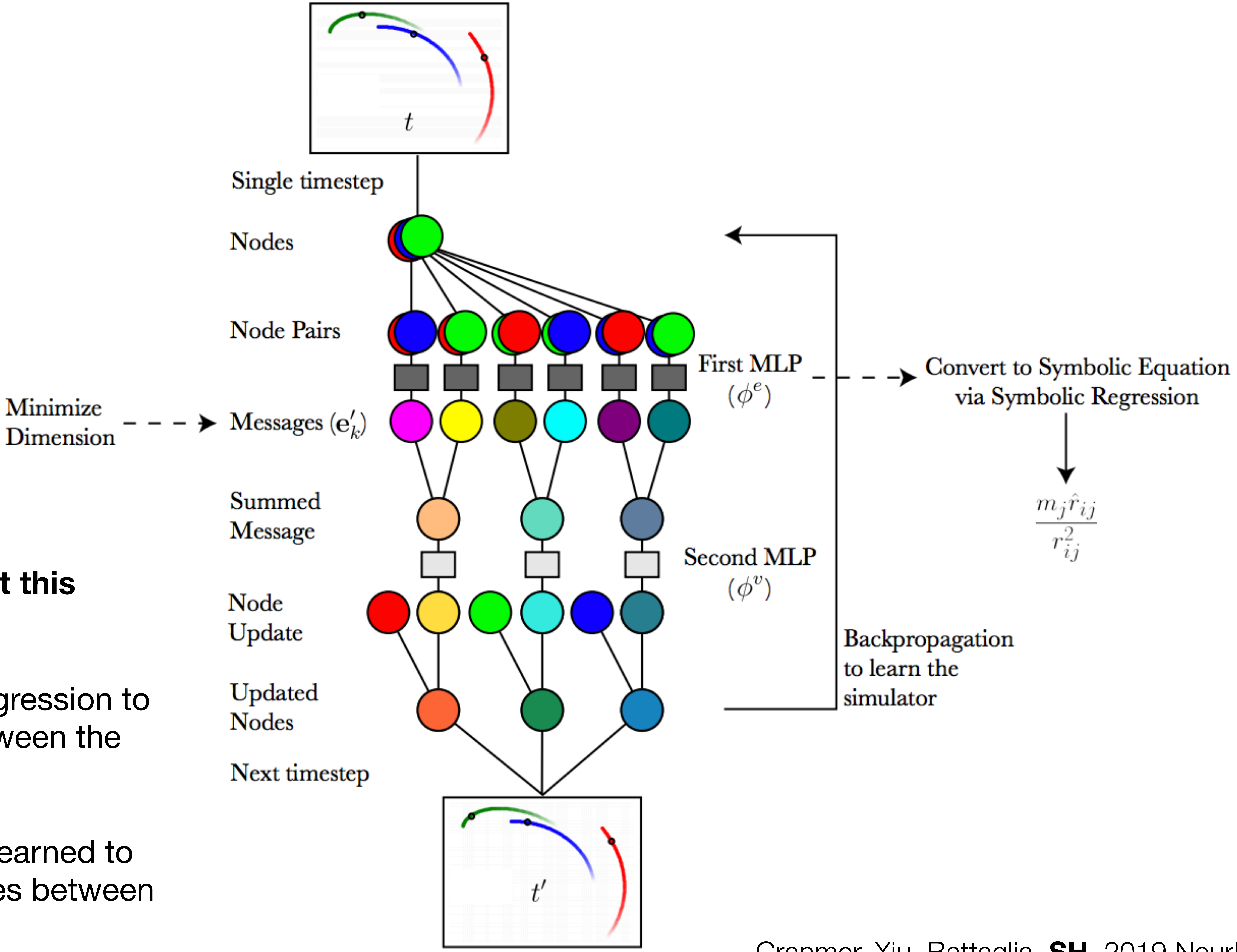


Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS

- **What is more about this paper?**
- We use symbolic regression to learn the forces between the nodes.
- Namely, e_k' can be learned to be the relevant forces between the nodes!



Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS



- **What is more about this paper?**

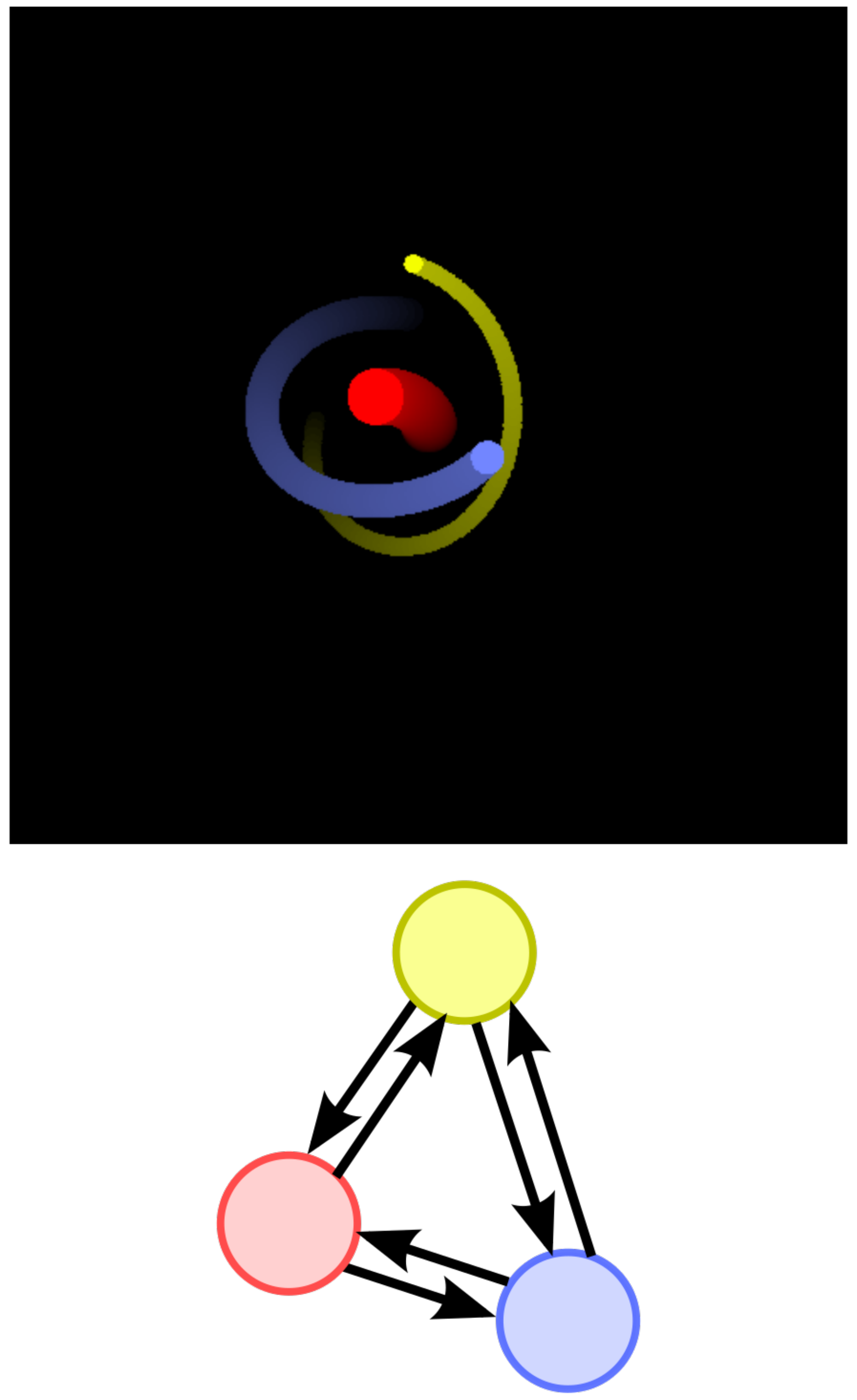
- We use symbolic regression to learn the forces between the nodes.

- Namely, e_k' can be learned to be the relevant forces between the nodes!

Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS

Did we learn force laws of the following systems?

n-body

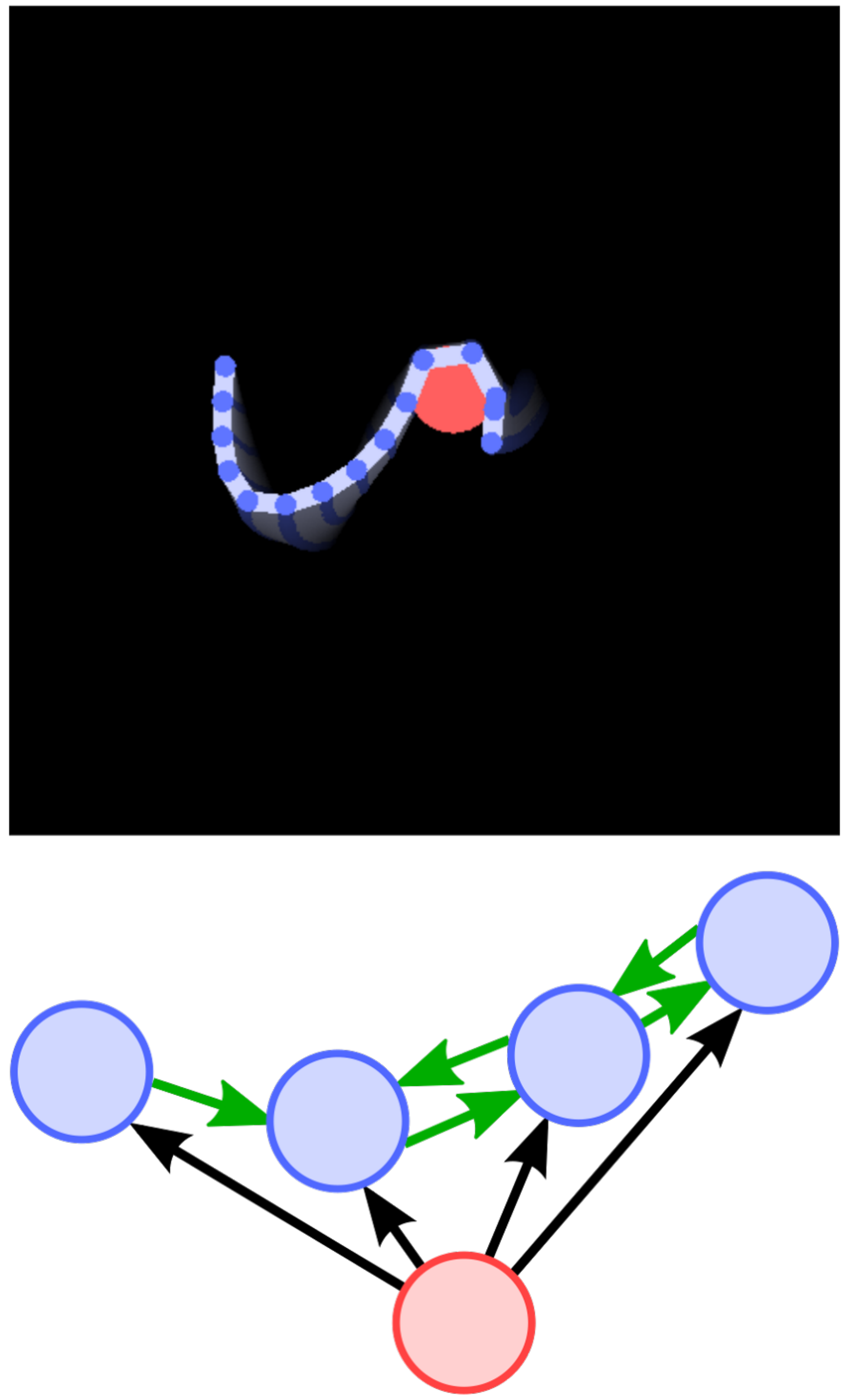


Edges: gravitational forces

Setup:

- 1/r, 1/(r^2) force in 2D for 3-body
- 1/(r^2) force in 3D for 3-body
- string with 1/(r^2) force in 2D
- 100,000 simulations each
- 1000 time-steps each

String



Edges: springs and rigid collisions

Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS

Yes we can!

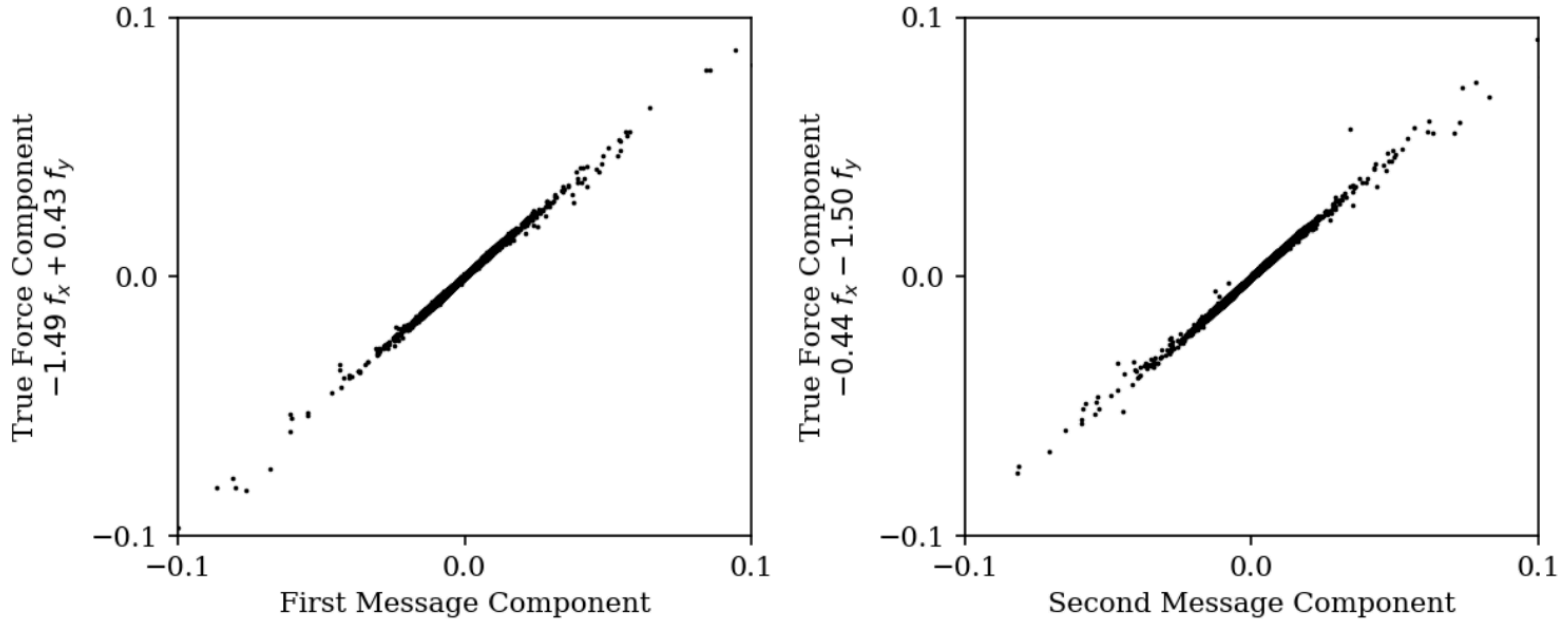
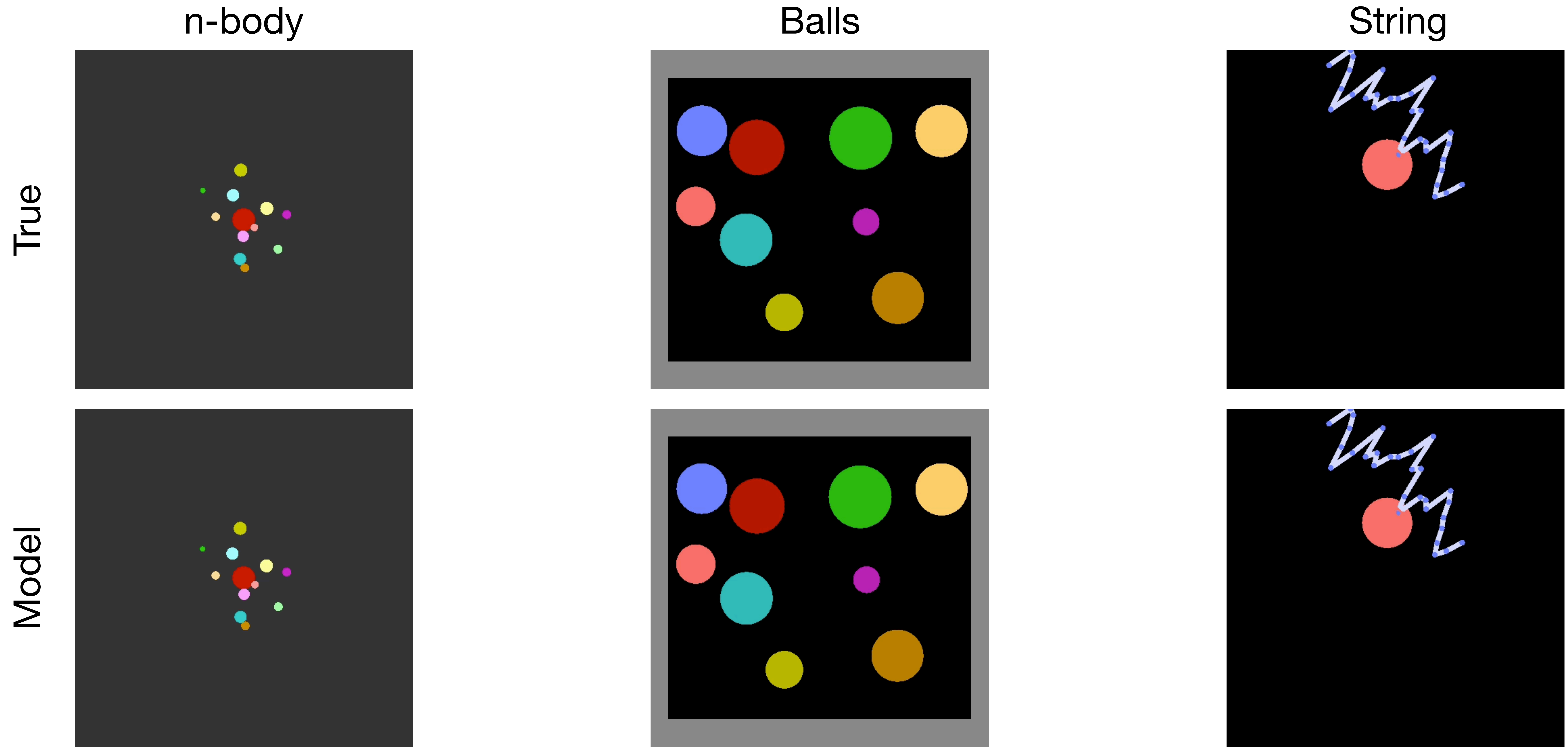


Figure 2: These plots demonstrate that the graph network's messages have learned to be linear transformations of the two vector components of the true force: f_x and f_y, for the $1/r$ law in 2D.

Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS

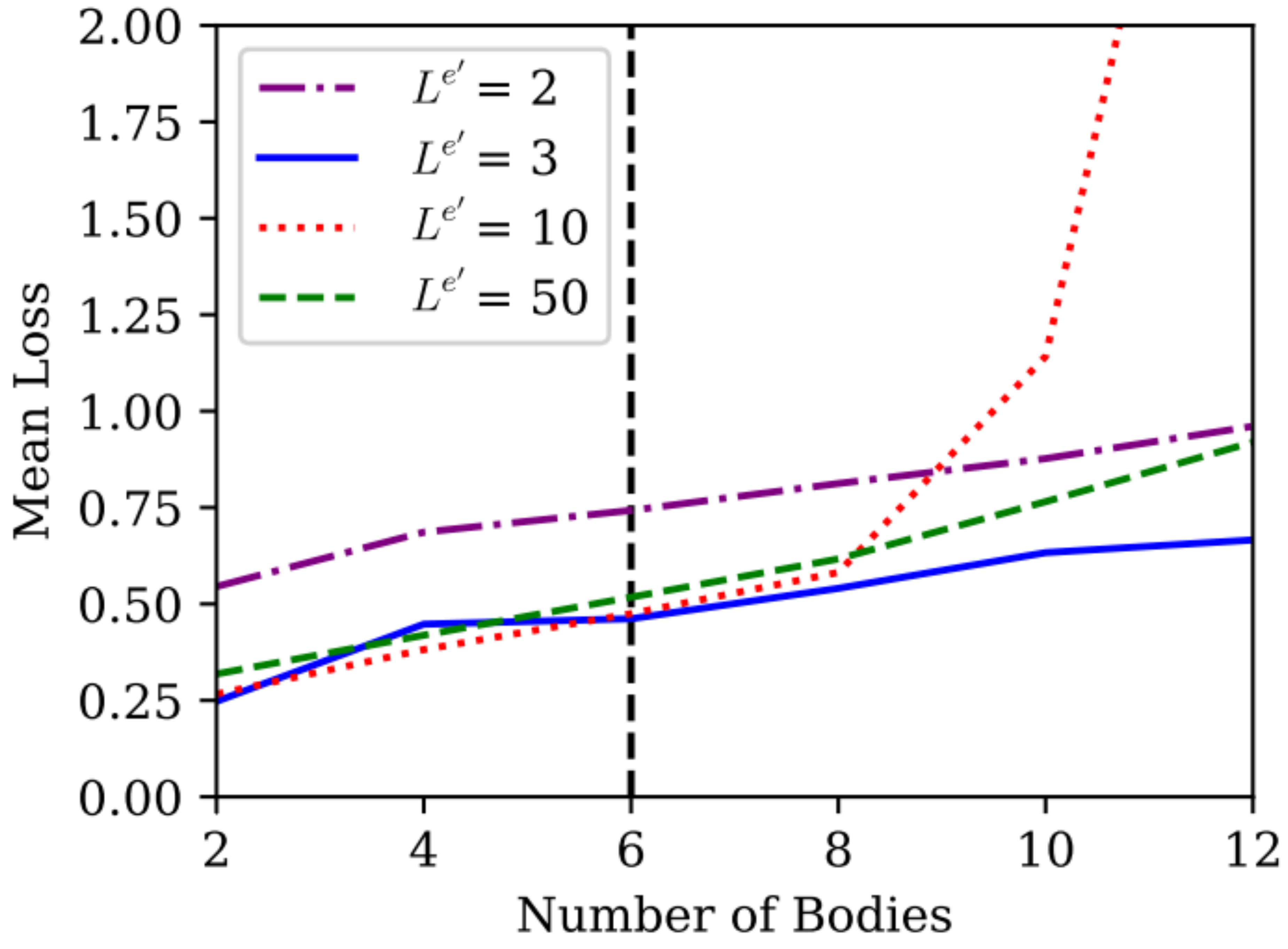
Even better: “Zero shot” generalization to larger systems



Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS

Battaglia et al., 2016, NeurIPS

And the generalization works better if you limit the dimension of the message passing



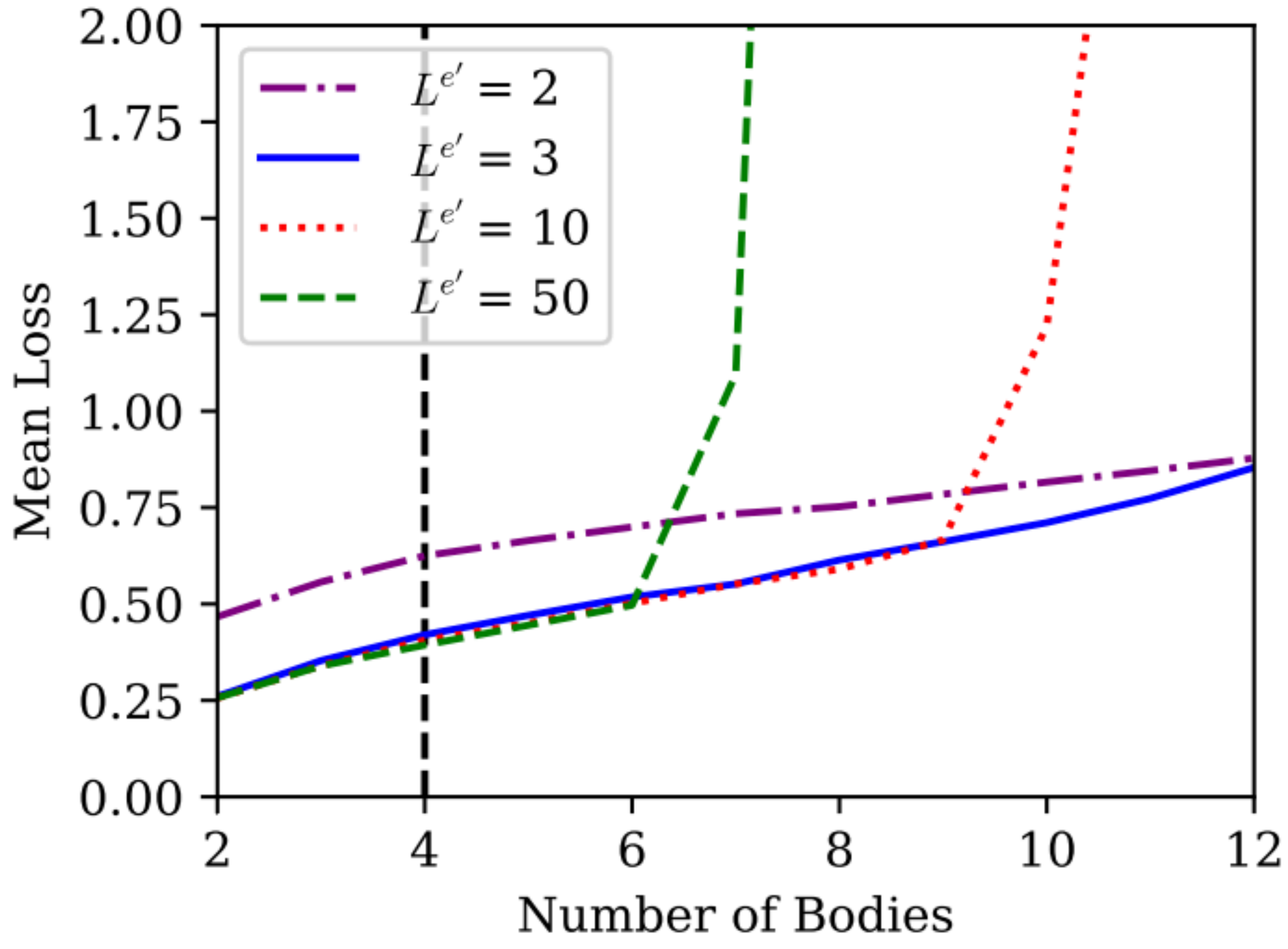


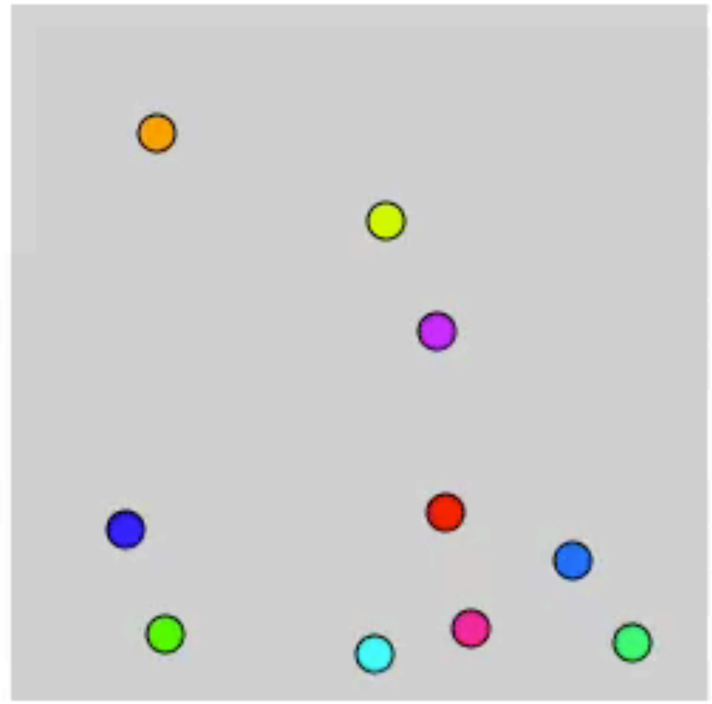
Figure 3: These plots demonstrate the improvement in generalization from minimizing the message passing space. The loss of GNs with different message-passing space dimension ($L^{e'}$), trained on a 6-body and 4-body system, in the left and right plots, respectively (indicated by the vertical line), are tested on a variable number of bodies in a $1/r^2$ simulation in 3D.

Cranmer, Xiu, Battaglia, **SH**, 2019 NeurIPS ML4PS

Other examples of what GN can do: Predicting the invisible element

Predict invisible springs in a mass-spring system

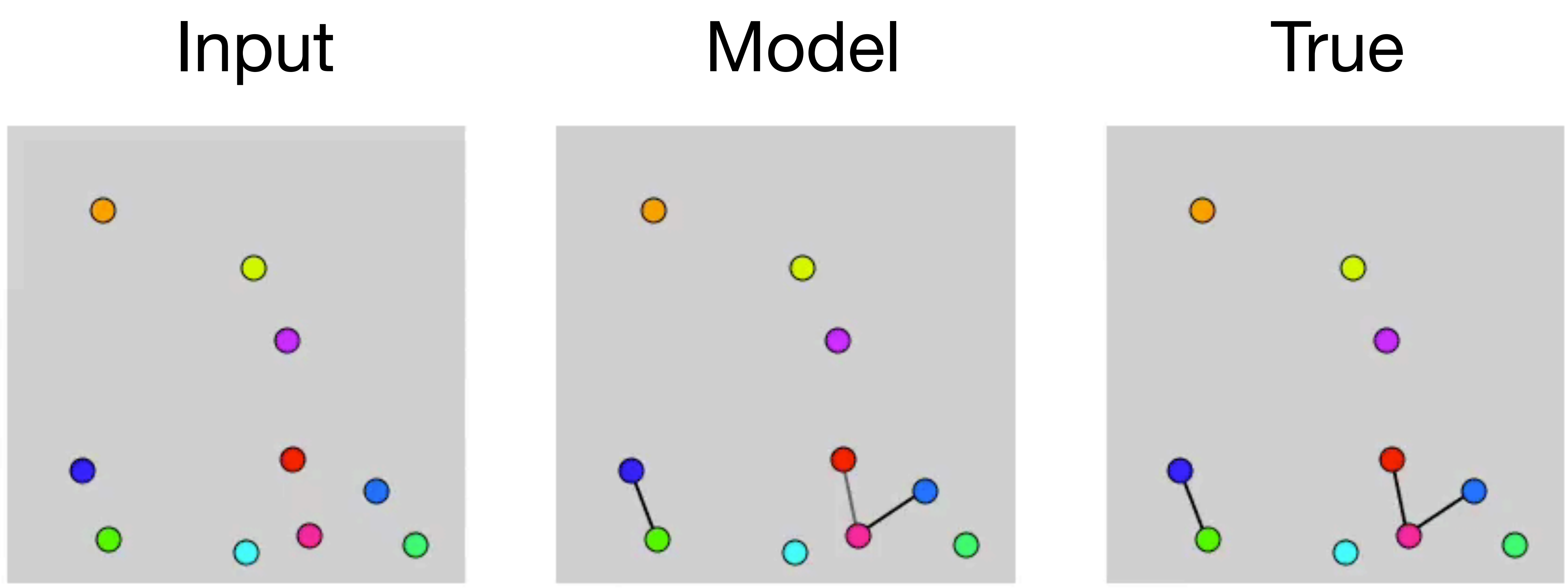
Input



Santoro et al., 2017, NeurIPS

Other examples of what GN can do: Predicting the invisible element

Predict invisible springs in a mass-spring system



Santoro et al., 2017, NeurIPS

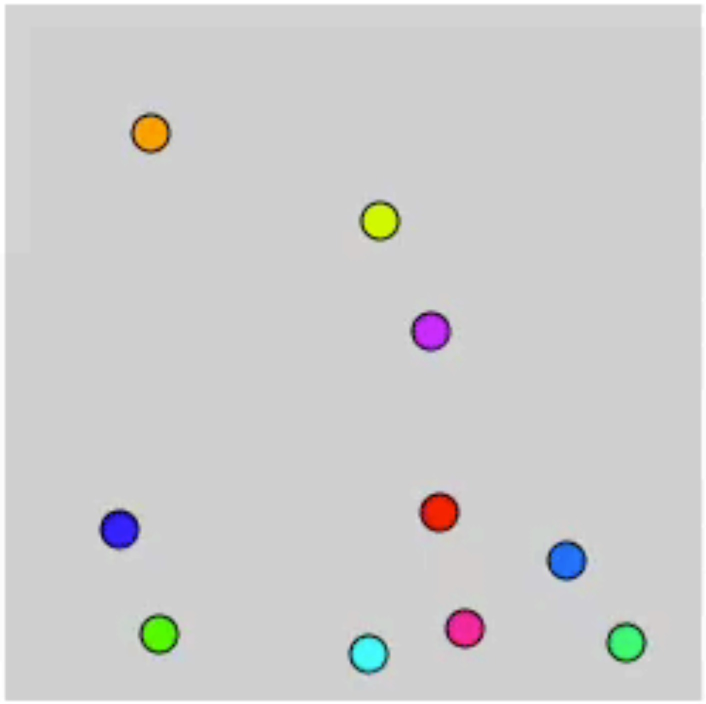
Conclusion

- It seems like it can learn from a set of simulations and generate more of the same without running the simulations again.
- The model seems to generalize well to larger N systems. Why?
- We have found ways to combine this with symbolic regression to find the physical rules that govern the forces between the nodes. Neural Programming Synthesis maybe even cooler?
- We includes an inductive bias in the message passing and this helps find the physical laws a lot faster than before.
- The generalization works for even larger N when this inductive bias is included!
- Graph Networks Rocks! Talk to Danilo who is here *who knows way more about GNN than me*!

Other examples of what GN can do: Predicting the invisible element

Predict invisible springs in a mass-spring system

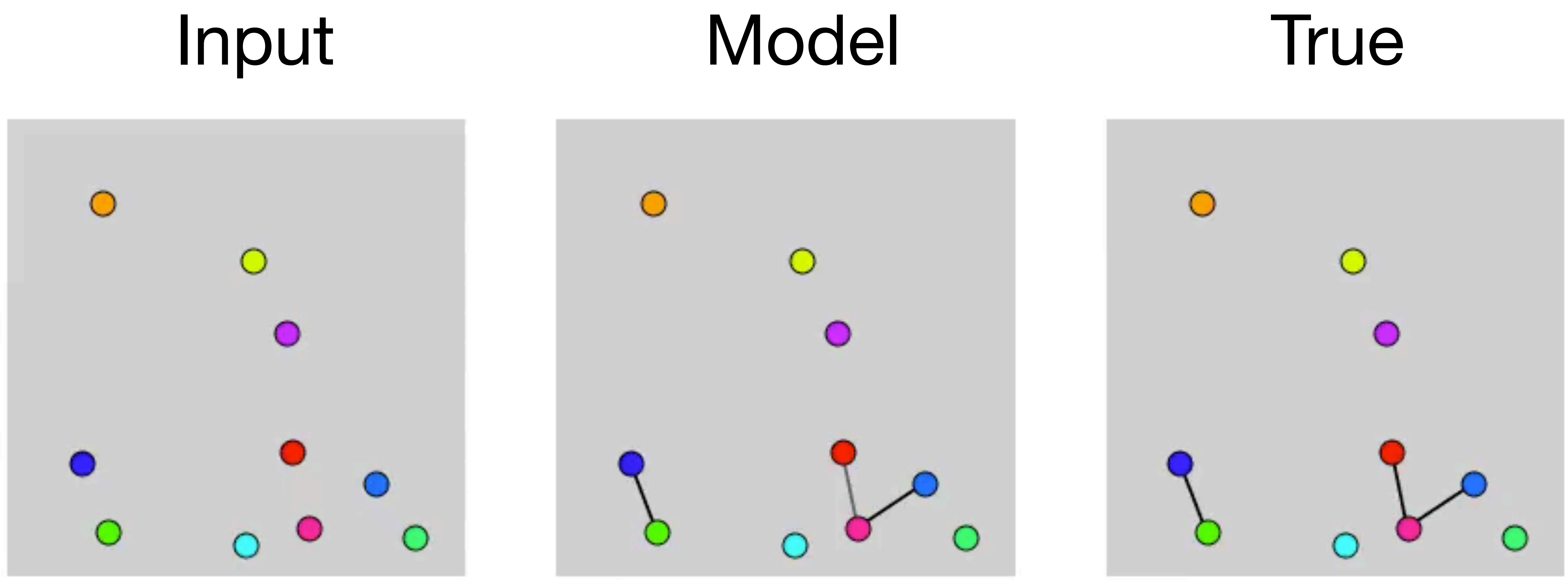
Input



Santoro et al., 2017, NeurIPS

Other examples of what GN can do: Predicting the invisible element

Predict invisible springs in a mass-spring system

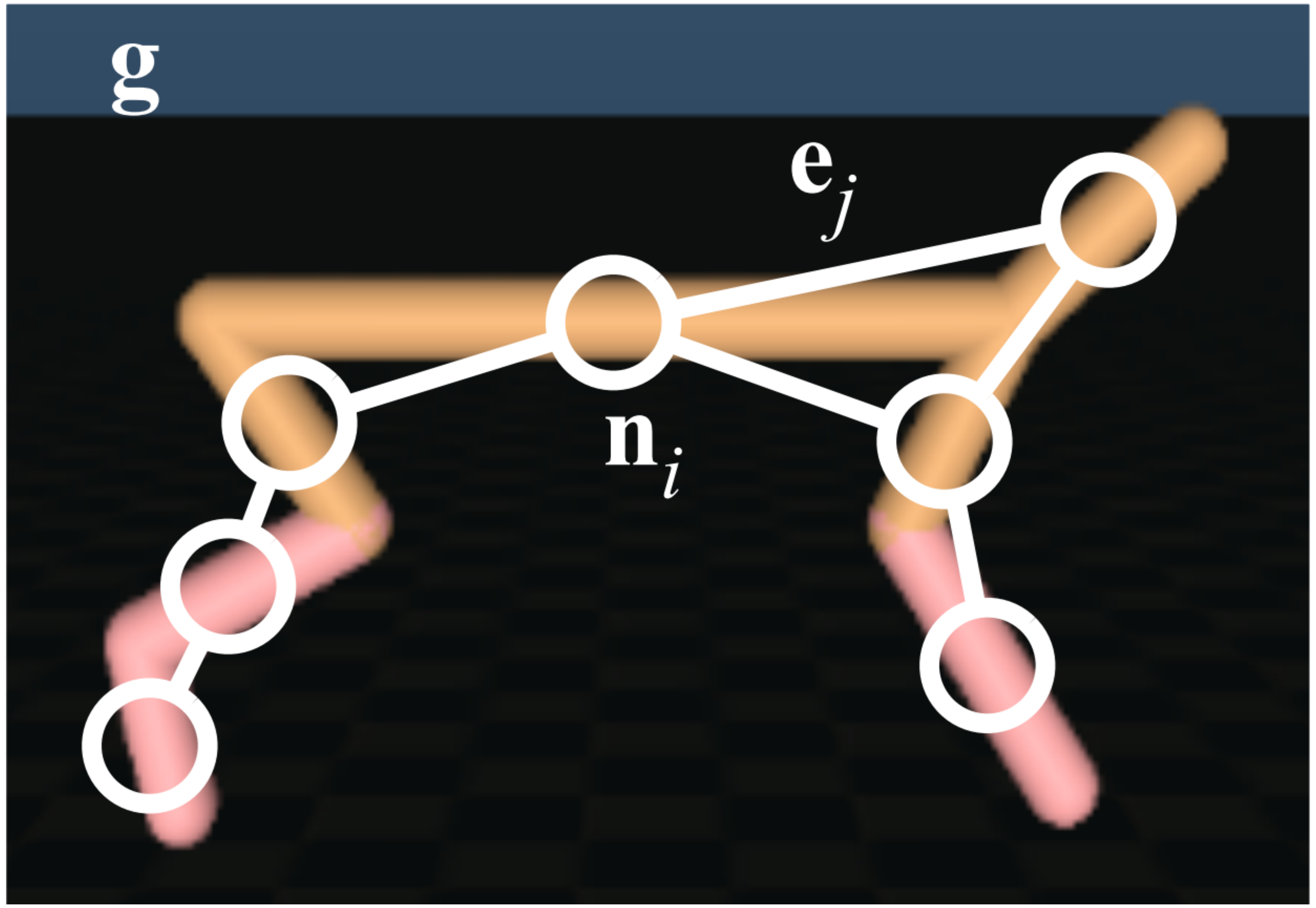


Santoro et al., 2017, NeurIPS

Other examples of what GN can do
Representing the actuated system as a graph

Representing physical system as a graph:

- Nodes ~ Bodies
- Edges ~ Joints
- Global properties



Sanchez-Gonzalez et al., 2018, ICML