Learning Compositional Koopman Operators for Model-Based Control

Yunzhu Li
MIT CSAIL

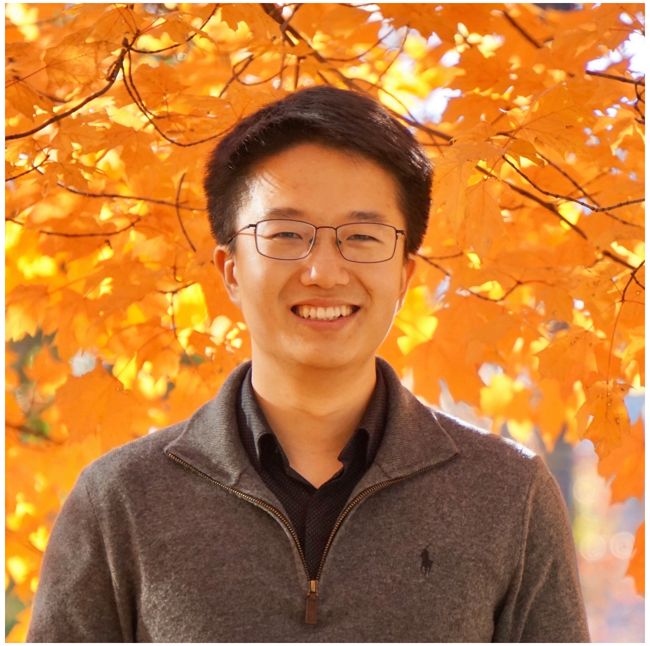
Talk at UToronto
Robotics Reading Group
2020/09/04

MIT

CSAIL

About Me

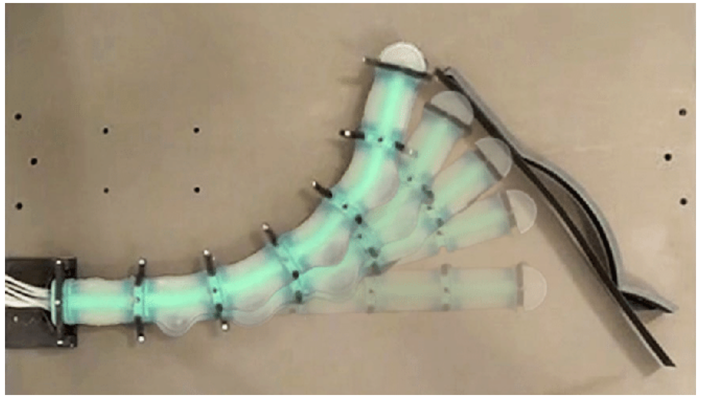
- Yunzhu Li
- Starting my fourth year PhD at MIT

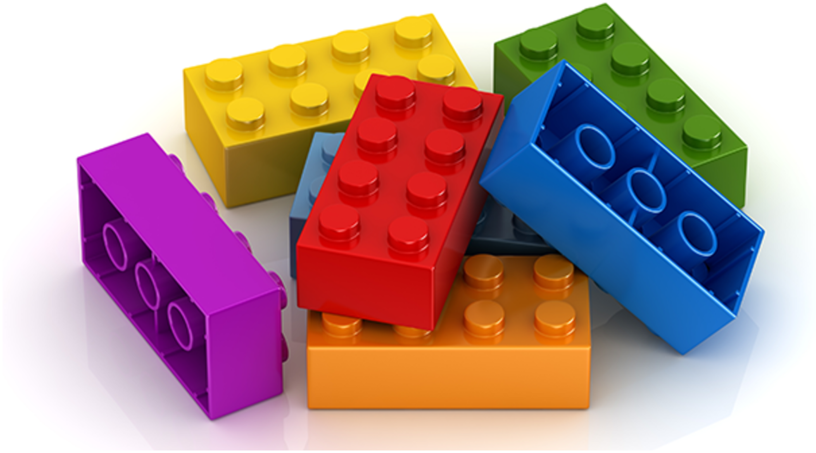


Advisors: Antonio Torralba and Russ Tedrake

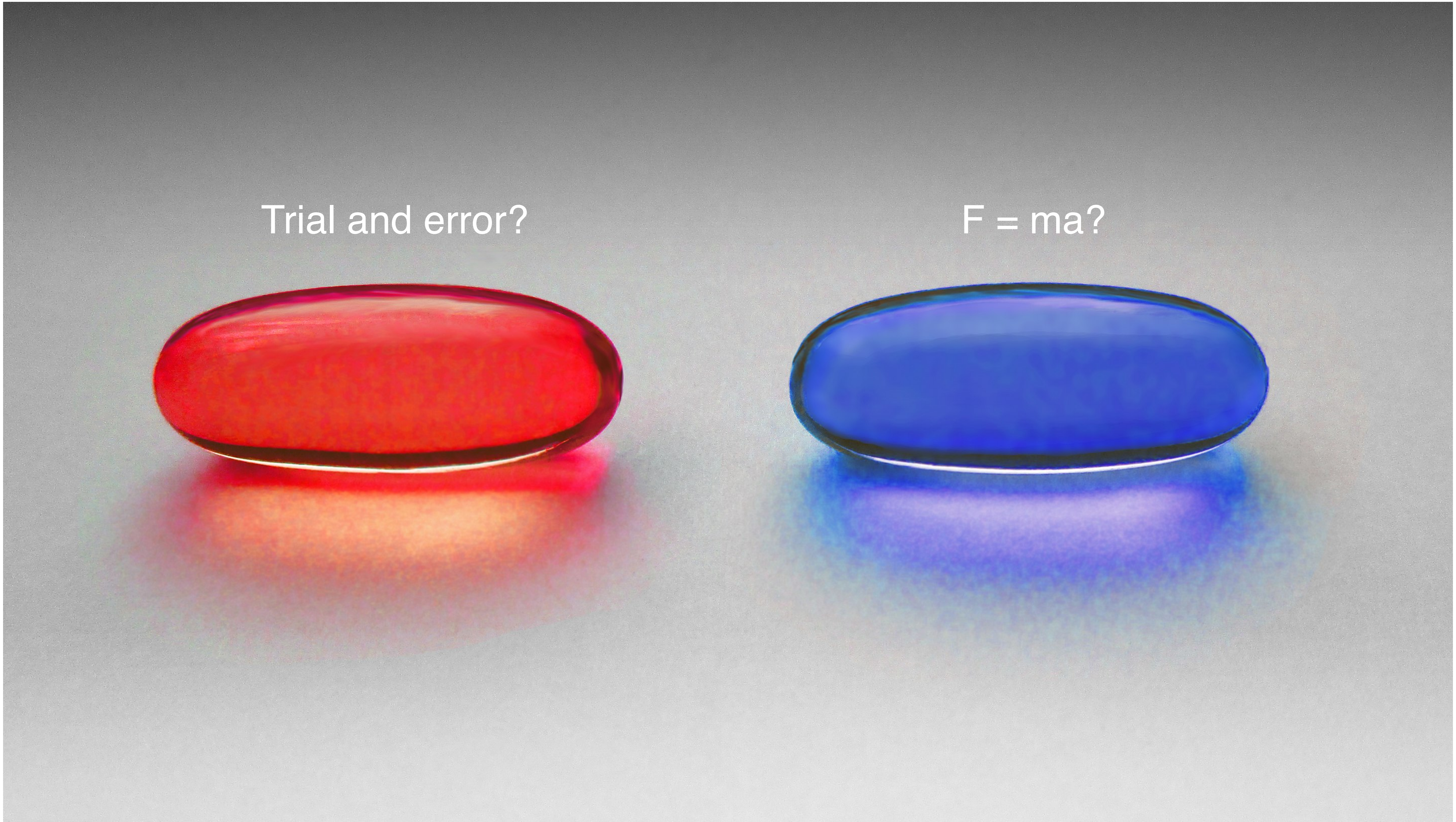
Learning-based dynamics modeling
Multi-modal perception

Compositionality in daily life







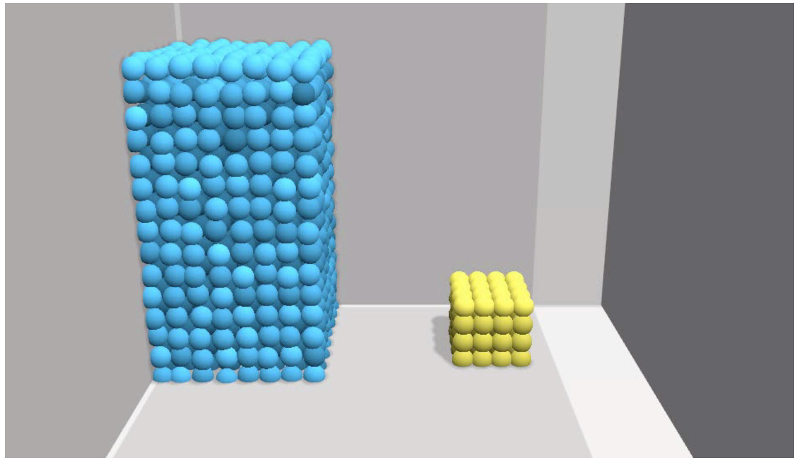


Trial and error?
F = ma?

Intuitive Physics

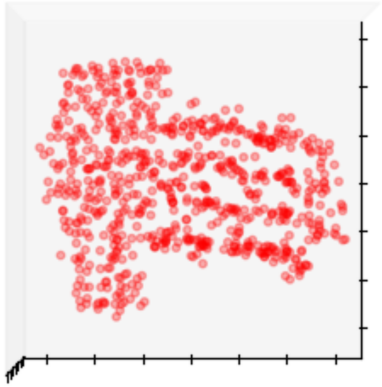
State Representation? Model Class?

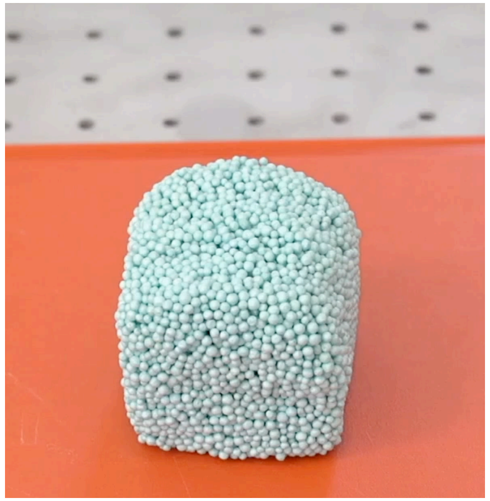
State Representation? Model Class?

Particle + Graph Neural Networks

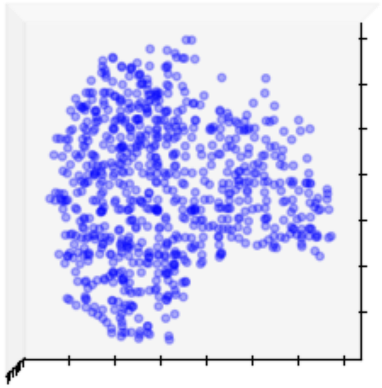


Goal





Result



Li, Wu, Tedrake, Tenenbaum, Torralba
Learning Particle Dynamics for Manipulating Rigid Bodies, Deformable Objects, and Fluids
ICLR 2019

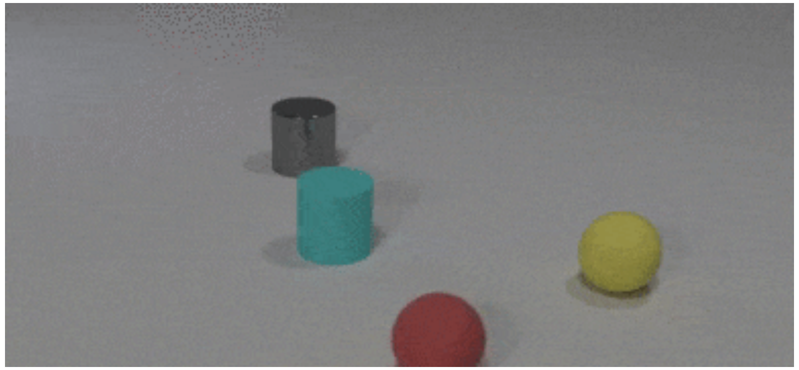
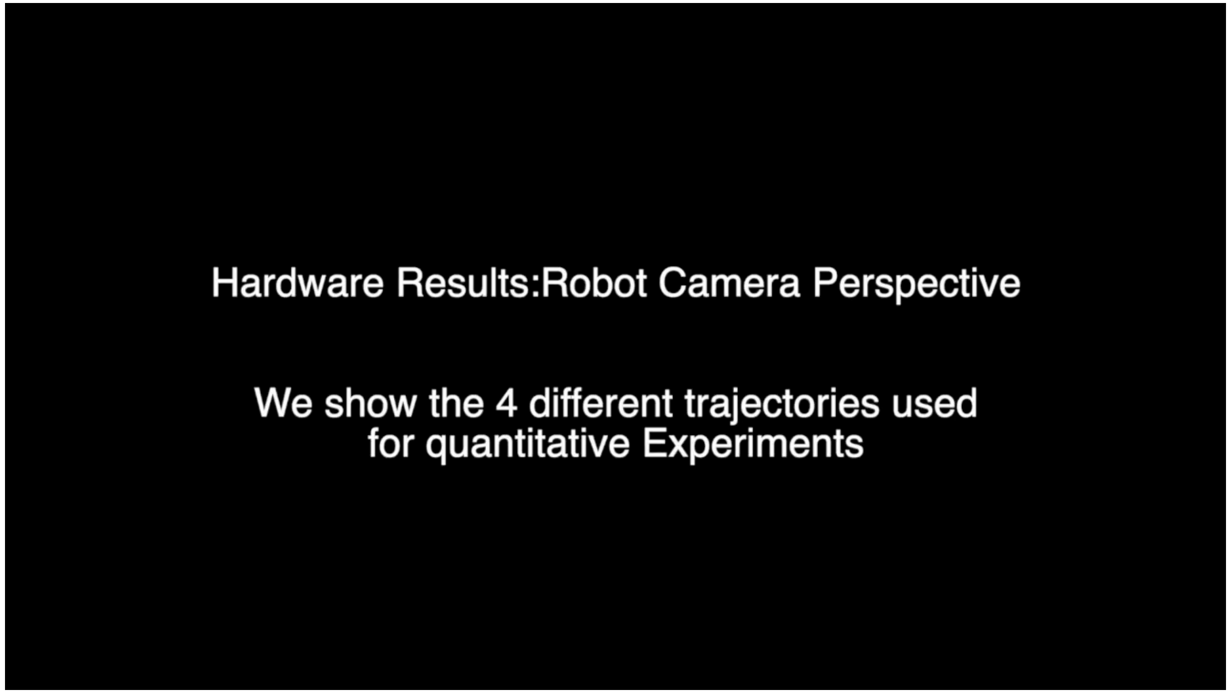


Image Patch + Graph Neural Networks

Yi*, Gan*, Li, Kohli, Wu, Torralba, Tenenbaum
CLEVRER: Collision Events for Video Representation and Reasoning
ICLR 2020

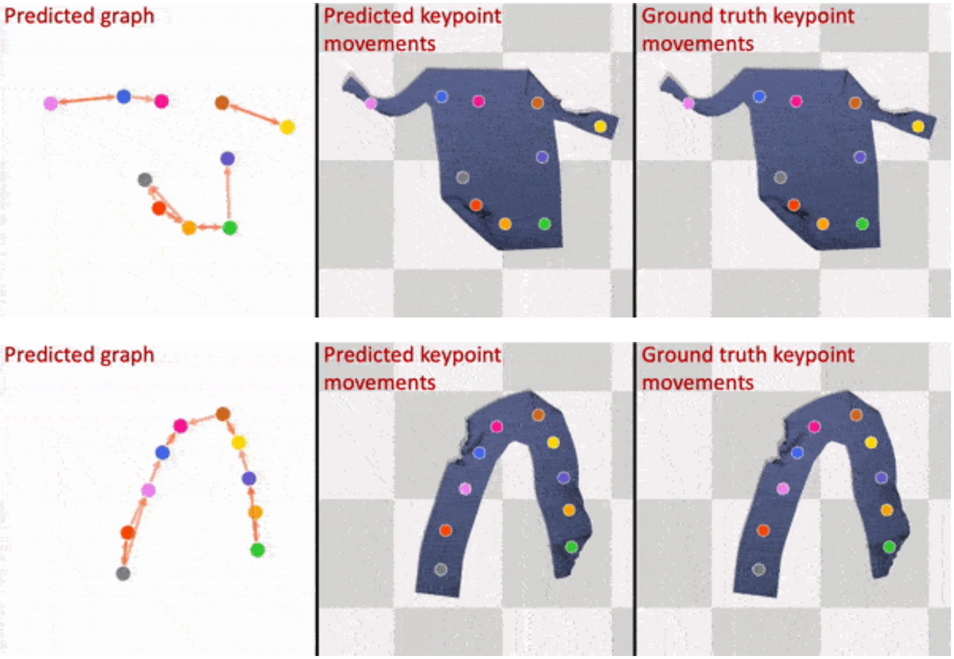
State Representation? Model Class?

Keypoints + MLP



Manuelli, Li, Florence, Tedrake.
In submission

Keypoints + Graph Neural Networks



Li, Torralba, Anandkumar, Fox, Garg
Causal Discovery in Physical Systems from Videos
In submission.

State Representation? Model Class?

- Different representations and model classes are suitable for different scenarios / tasks.
- There may not need a “universal” choice that works for all use cases.
- It is essential to understand the advantages and limitations.

State Representation? Model Class?

- Different representations and model classes are suitable for different scenarios / tasks.
- There may not need a “universal” choice that works for all use cases.
- It is essential to understand the advantages and limitations.

- ***Compositional Koopman Operators*** lies in the category of
- Object-centric latent vectors
- Graph Neural Networks + Linear Dynamics

Problem

- Given observations from a system of unknown dynamics

$$\boldsymbol{x}^{t+1} = \mathbf{F}(\boldsymbol{x}^t, \boldsymbol{u}^t)$$

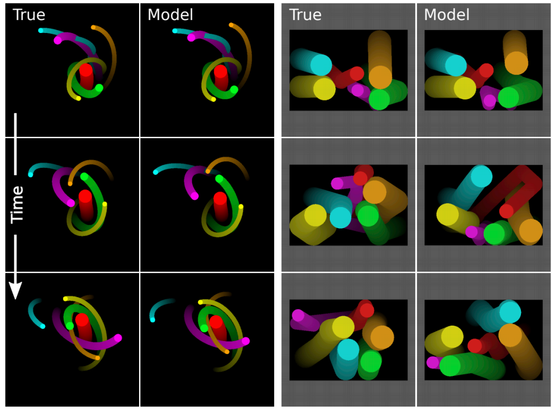
system state $\boldsymbol{x}^t$ control signal $\boldsymbol{u}^t$ dynamics $\mathbf{F}$

- Task 1: system identification

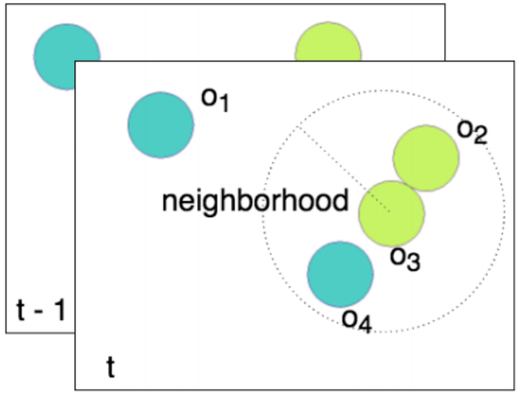
- Task 2: control synthesis

Previous Methods

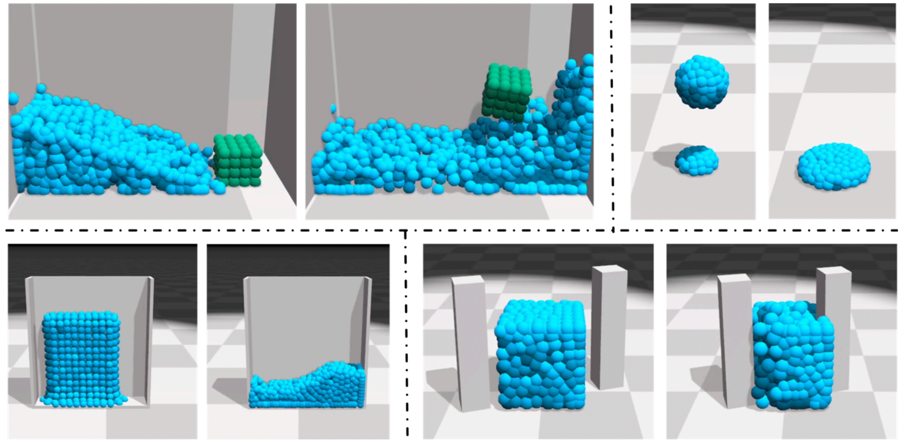
- Graph Neural Networks



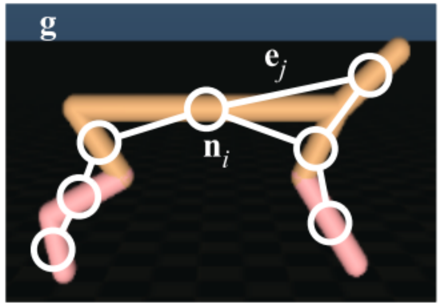
Battaglia, Pascanu, Lai, Rezende, Kavukcuoglu. NeurIPS’16



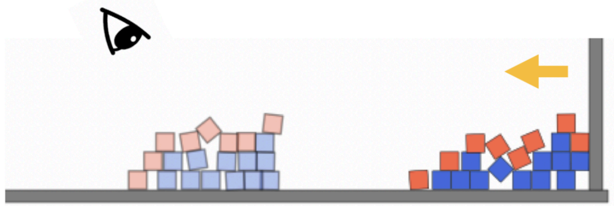
Chang, Ullman, Torralba, Tenenbaum. ICLR’17



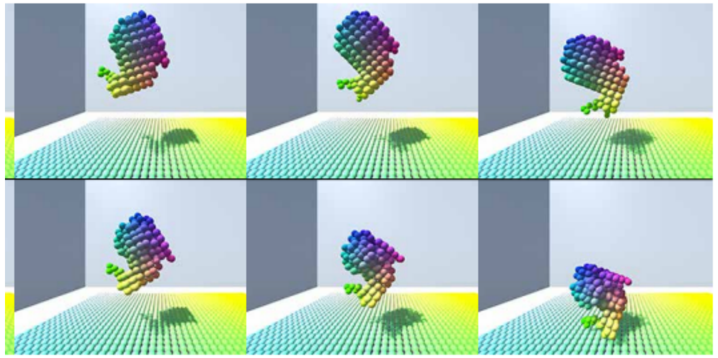
Li, Wu, Tedrake, Tenenbaum, Torralba. ICLR’19



Sanchez-Gonzalez, Heess, Springenberg, Merel, Riedmiller, Hadsell, Battaglia. ICML’18



Li, Wu, Zhu, Tenenbaum, Torralba, Tedrake. ICRA’19



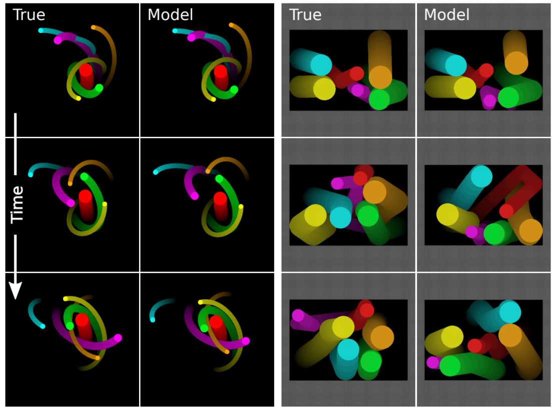
Mrowca, Zhuang, Wang, Haber, Fei-Fei, Tenenbaum, Yamins. NeurIPS’18

Previous Methods

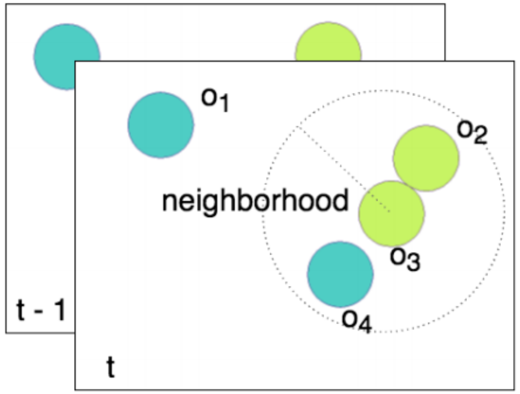
- Graph Neural Networks

Learned dynamics is highly nonlinear

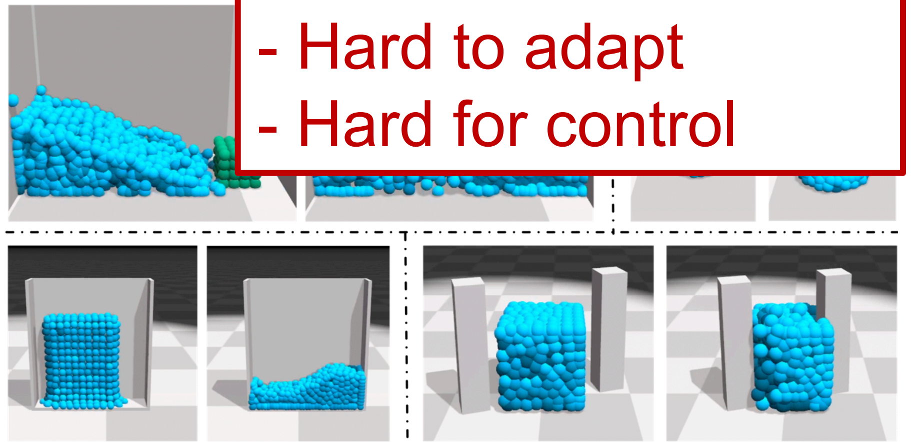
- Hard to adapt
- Hard for control



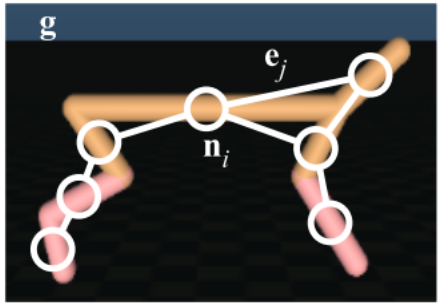
Battaglia, Pascanu, Lai, Rezende, Kavukcuoglu. NeurIPS'16



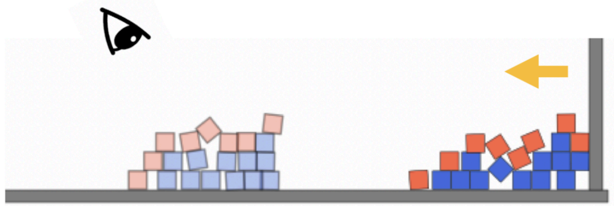
Chang, Ullman, Torralba, Tenenbaum. ICLR'17



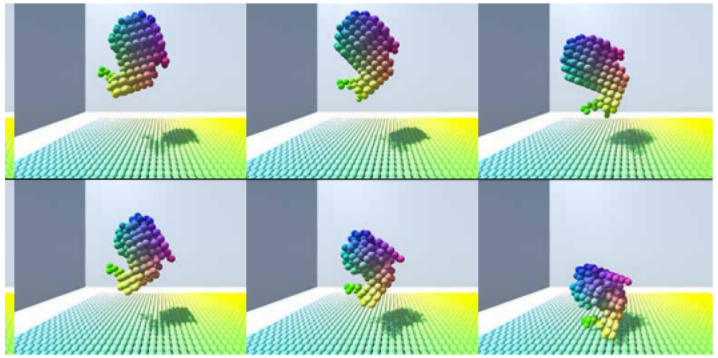
Li, Wu, Tedrake, Tenenbaum, Torralba. ICLR'19



Sanchez-Gonzalez, Heess, Springenberg, Merel, Riedmiller, Hadsell, Battaglia. ICML'18



Li, Wu, Zhu, Tenenbaum, Torralba, Tedrake. ICRA'19

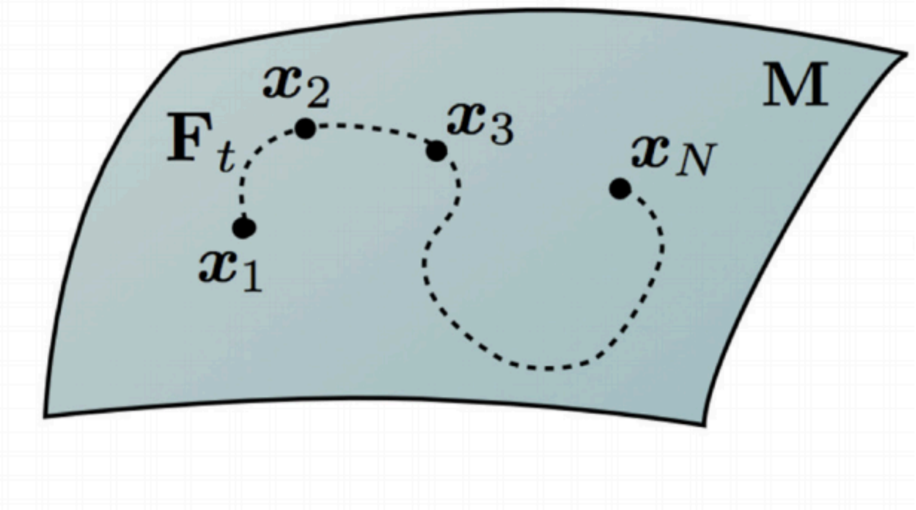


Mrowca, Zhuang, Wang, Haber, Fei-Fei, Tenenbaum, Yamins. NeurIPS'18

Previous Methods

- The Koopman Operator Theory

$$\boldsymbol{x}_{t+1} = F(\boldsymbol{x}_t)$$

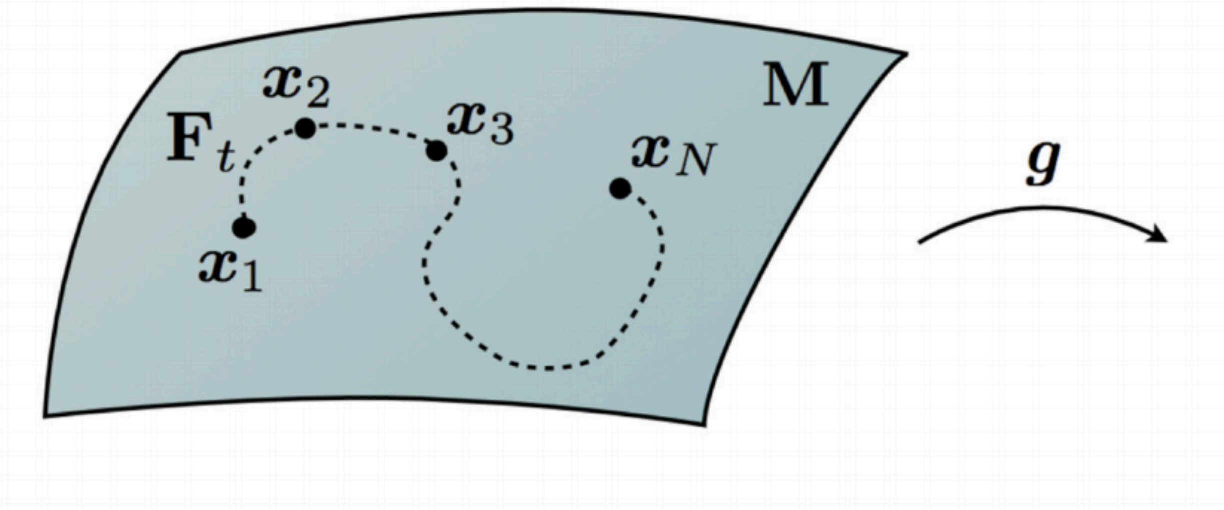


Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz
Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control
PloS one 11.2 (2016).

Previous Methods

- The Koopman Operator Theory

$$\boldsymbol{x}_{t+1} = F(\boldsymbol{x}_t) \qquad \boldsymbol{y}_t = g(\boldsymbol{x}_t)$$

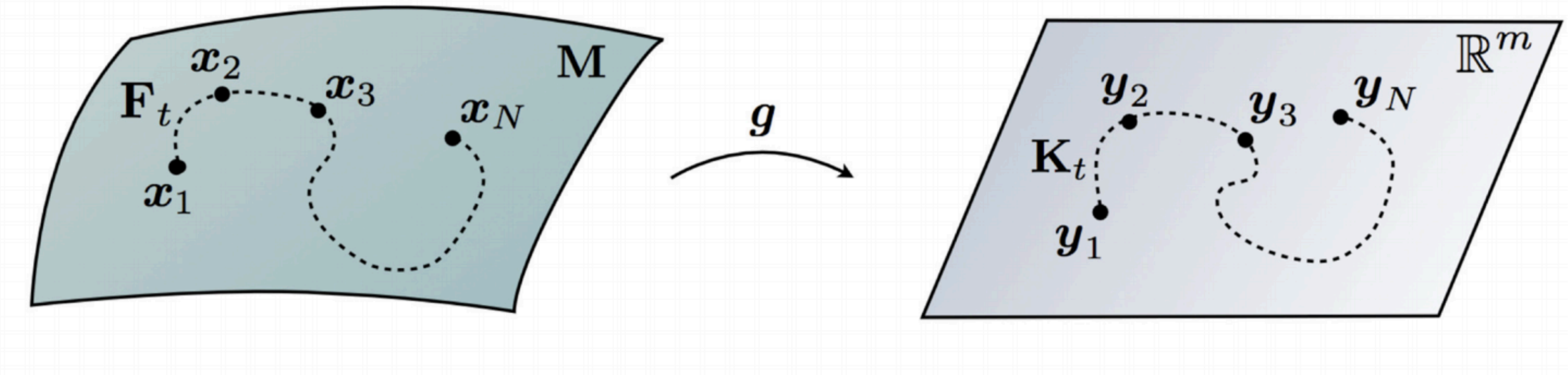


Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz
Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control
PloS one 11.2 (2016).

Previous Methods

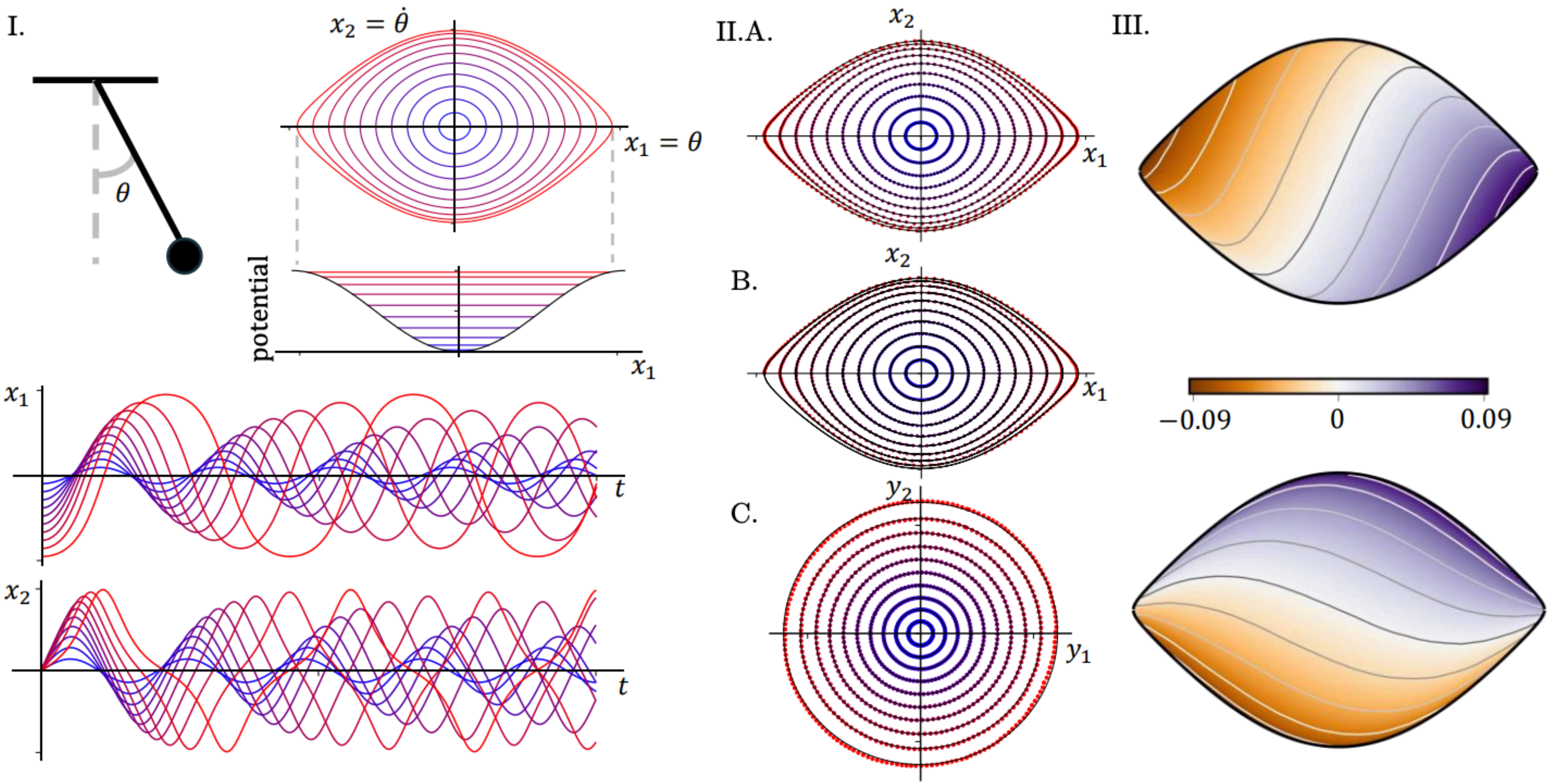
- The Koopman Operator Theory

$$\boldsymbol{x}_{t+1} = F(\boldsymbol{x}_t) \qquad \boldsymbol{y}_t = g(\boldsymbol{x}_t) \qquad \boldsymbol{y}_{t+1} = K\boldsymbol{y}_t$$



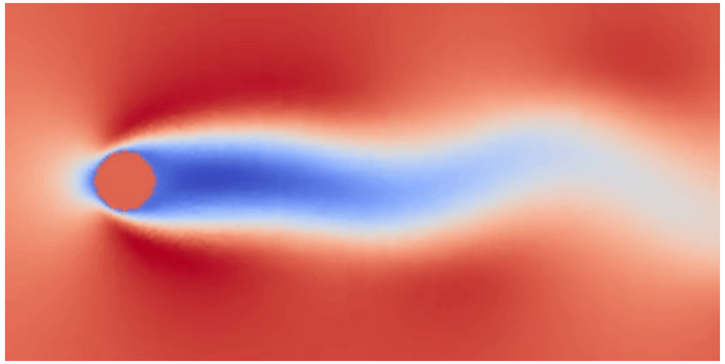
Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz
Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control
PloS one 11.2 (2016).

Previous Methods



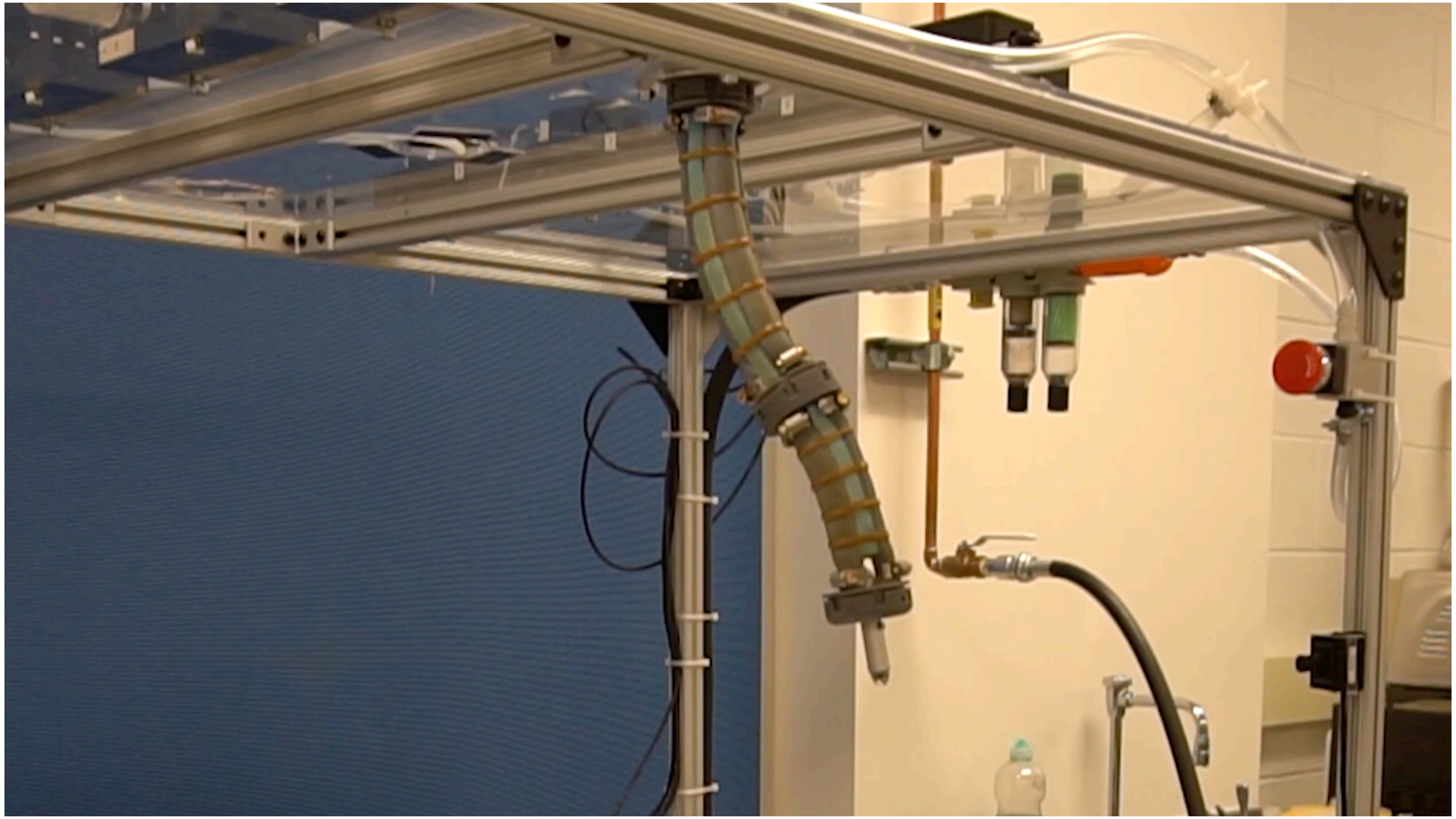
Lusch, Bethany, J. Nathan Kutz, and Steven L. Brunton
Deep learning for universal linear embeddings of nonlinear dynamics
Nature communications 9.1 (2018): 4950.

Previous Methods



Morton, Jeremy, et al.
Deep dynamical modeling and control of unsteady fluid flows.
Advances in Neural Information Processing Systems. 2018.

Previous Methods

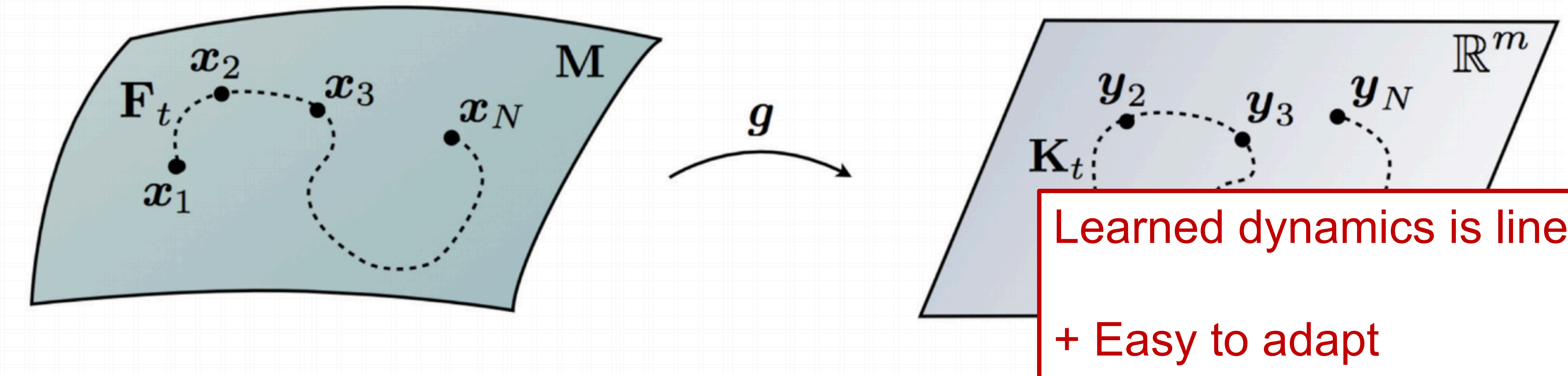


Bruder, Daniel, Brent Gillespie, C. David Remy, and Ram Vasudevan
Modeling and Control of Soft Robots Using the Koopman Operator and Model Predictive Control
RSS 2019.

Previous Methods

- The Koopman Operator Theory

$$x_{t+1} = F(x_t) \qquad y_t = g(x_t) \qquad y_{t+1} = K y_t$$



Learned dynamics is linear

+ Easy to adapt
+ Easy for control

- Unable to handle compositional system

Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz
Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynam
PloS one 11.2 (2016).

Graph Neural Networks

The Koopman Operator Theory

+ Capture the compositionality

- Hard to adapt
- Hard for control

- Unable to handle compositional systems

+ Easy to adapt
+ Easy for control

Compositional Koopman Operators

+ Generalize to compositional systems

+ Easy to adapt
+ Easy for control

Motivating example

Consider a system with N balls connected by linear spring.

$$\boldsymbol{x}_i \triangleq [x_i, y_i, \dot{x}_i, \dot{y}_i]^T$$

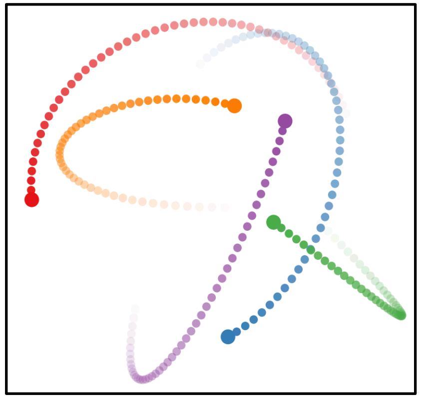


Image from Kipf et al. ICML 2018.

Motivating example

Consider a system with N balls connected by linear spring.

$$\boldsymbol{x}_i \triangleq [x_i, y_i, \dot{x}_i, \dot{y}_i]^T$$

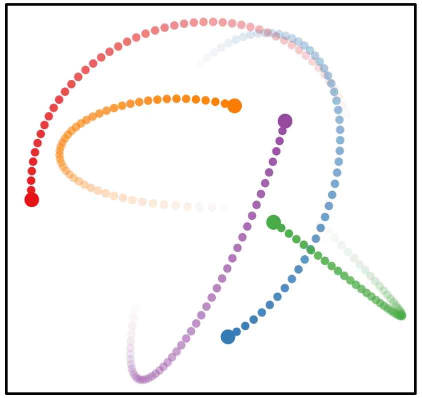


Image from Kipf et al. ICML 2018.

$$\dot{\boldsymbol{x}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \ddot{x}_i \\ \ddot{y}_i \end{bmatrix} = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \sum_{j=1}^{N} k(x_j - x_i) \\ \sum_{j=1}^{N} k(y_j - y_i) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ k - Nk & 0 & 0 & 0 \\ 0 & k - Nk & 0 & 0 \end{bmatrix}}_{\triangleq A} \begin{bmatrix} x_i \\ y_i \\ \dot{x}_i \\ \dot{y}_i \end{bmatrix} + \sum_{j \neq i} \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \end{bmatrix}}_{\triangleq B} \begin{bmatrix} x_i \\ y_i \\ \dot{x}_i \\ \dot{y}_i \end{bmatrix}$$

Motivating example

Consider a system with N balls connected by linear spring.

$$\boldsymbol{x}_i \triangleq [x_i, y_i, \dot{x}_i, \dot{y}_i]^T$$

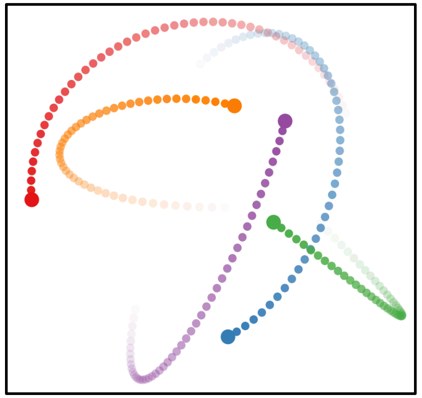


Image from Kipf et al. ICML 2018.

$$\dot{\boldsymbol{x}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \ddot{x}_i \\ \ddot{y}_i \end{bmatrix} = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \sum_{j=1}^{N} k(x_j - x_i) \\ \sum_{j=1}^{N} k(y_j - y_i) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ k - Nk & 0 & 0 & 0 \\ 0 & k - Nk & 0 & 0 \end{bmatrix}}_{\triangleq A} \begin{bmatrix} x_i \\ y_i \\ \dot{x}_i \\ \dot{y}_i \end{bmatrix} + \sum_{j \neq i} \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \end{bmatrix}}_{\triangleq B} \begin{bmatrix} x_i \\ y_i \\ \dot{x}_i \\ \dot{y}_i \end{bmatrix}$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \vdots \\ \dot{\boldsymbol{x}}_N \end{bmatrix} = \begin{bmatrix} A & B & \cdots & B \\ B & A & \cdots & B \\ \vdots & \vdots & \ddots & \vdots \\ B & B & \cdots & A \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix}$$

Motivating example

Consider a system with N balls connected by linear spring.

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \vdots \\ \dot{\boldsymbol{x}}_N \end{bmatrix} = \begin{bmatrix} A & B & \cdots & B \\ B & A & \cdots & B \\ \vdots & \vdots & \ddots & \vdots \\ B & B & \cdots & A \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix}$$

Three observations:

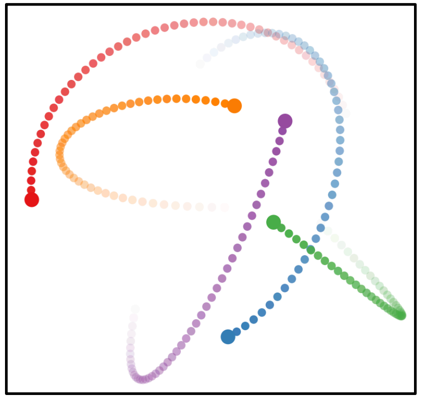


Image from Kipf et al. ICML 2018.

Motivating example

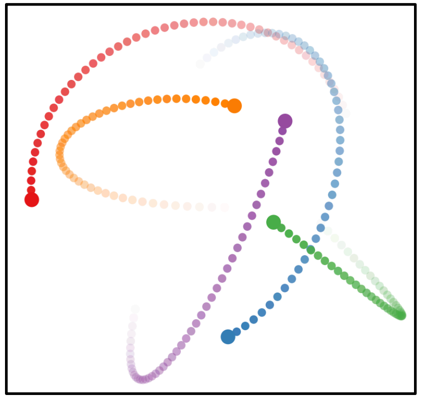


Image from Kipf et al. ICML 2018.

Consider a system with N balls connected by linear spring.

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \vdots \\ \dot{\boldsymbol{x}}_N \end{bmatrix} = \begin{bmatrix} A & B & \cdots & B \\ B & A & \cdots & B \\ \vdots & \vdots & \ddots & \vdots \\ B & B & \cdots & A \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix}$$

Three observations:

(1) The system state is composed of the state of each individual object.

Motivating example

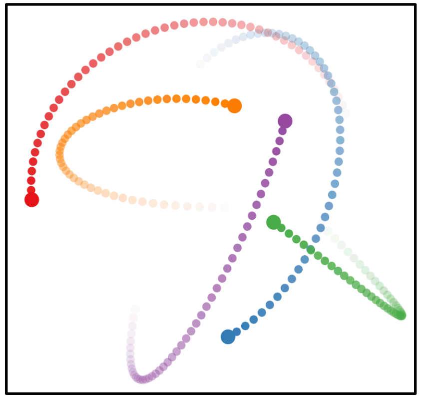


Image from Kipf et al. ICML 2018.

Consider a system with N balls connected by linear spring.

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \vdots \\ \dot{\boldsymbol{x}}_N \end{bmatrix} = \begin{bmatrix} A & B & \cdots & B \\ B & A & \cdots & B \\ \vdots & \vdots & \ddots & \vdots \\ B & B & \cdots & A \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix}$$

Three observations:

(1) The system state is composed of the state of each individual object.
(2) The transition matrix has a block-wise substructure.

Motivating example

Consider a system with N balls connected by linear spring.

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \vdots \\ \dot{\boldsymbol{x}}_N \end{bmatrix} = \begin{bmatrix} A & B & \cdots & B \\ B & A & \cdots & B \\ \vdots & \vdots & \ddots & \vdots \\ B & B & \cdots & A \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix}$$

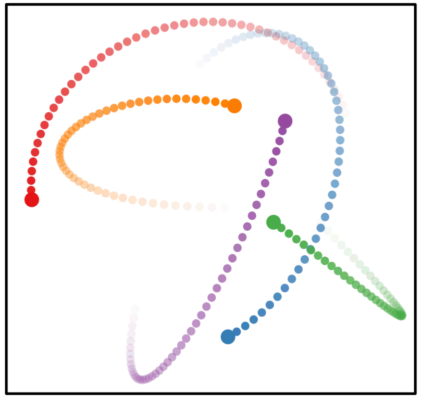


Image from Kipf et al. ICML 2018.

Three observations:

(1) The system state is composed of the state of each individual object.
(2) The transition matrix has a block-wise substructure.
(3) The same physical interactions share the same transition block.

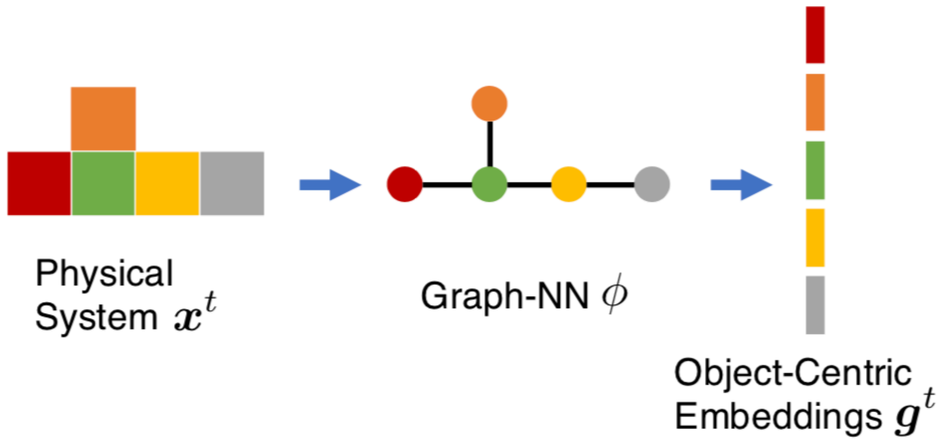
Compositional Koopman Operators

Three observations from the spring system:

(1) The system state is composed of the state of each individual object.
(2) The transition matrix has a block-wise substructure.
(3) The same physical interactions share the same transition block.

Compositional Koopman Operators

Three observations from the spring system:
(1) The system state is composed of the state of each individual object.
(2) The transition matrix has a block-wise substructure.
(3) The same physical interactions share the same transition block.

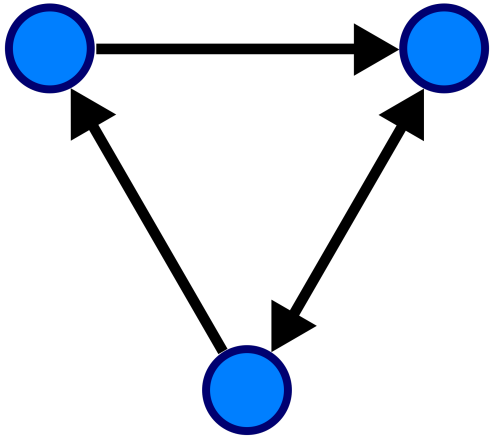


(1) The Koopman embedding of the system is composed of the Koopman embedding of every objects.

$\boldsymbol{g}^t \in \mathbb{R}^{Nm}$ is the concatenation of $\boldsymbol{g}_1^t, \cdots, \boldsymbol{g}_N^t$

Graph Neural Networks

- Represent the state as a graph, where each component is a node
- Model the interactions between components using neural networks



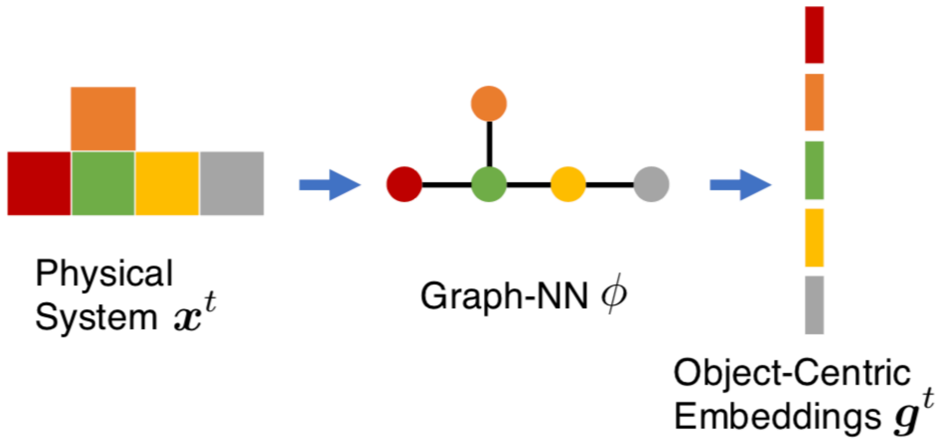
$$G = \langle O, R \rangle$$

$$e_k = f_R(o_i, o_j), r_k = \langle o_i, o_j \rangle$$

$$h_i = f_O(o_i, \textstyle\sum_{k \in \mathcal{N}_i} e_k)$$

Compositional Koopman Operators

Three observations from the spring system:
(1) The system state is composed of the state of each individual object.
(2) The transition matrix has a block-wise substructure.
(3) The same physical interactions share the same transition block.



(1) The Koopman embedding of the system is composed of the Koopman embedding of every objects.

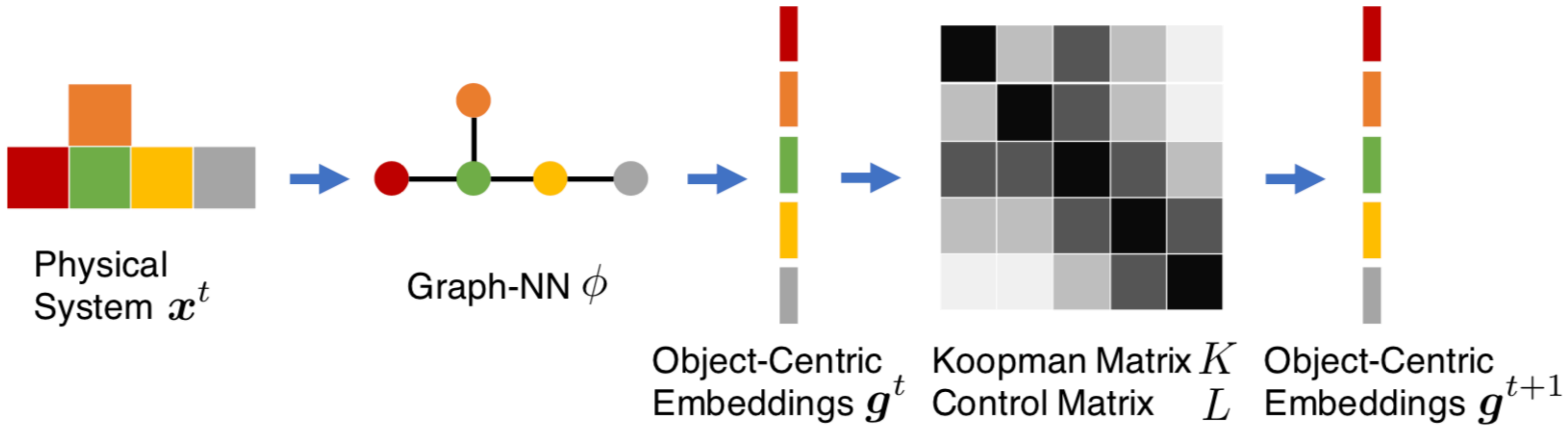
$\boldsymbol{g}^t \in \mathbb{R}^{Nm}$ is the concatenation of $\boldsymbol{g}_1^t, \cdots, \boldsymbol{g}_N^t$

Compositional Koopman Operators

Three observations from the spring system:
(1) The system state is composed of the state of each individual object.
(2) The transition matrix has a block-wise substructure.
(3) The same physical interactions share the same transition block.

Assuming

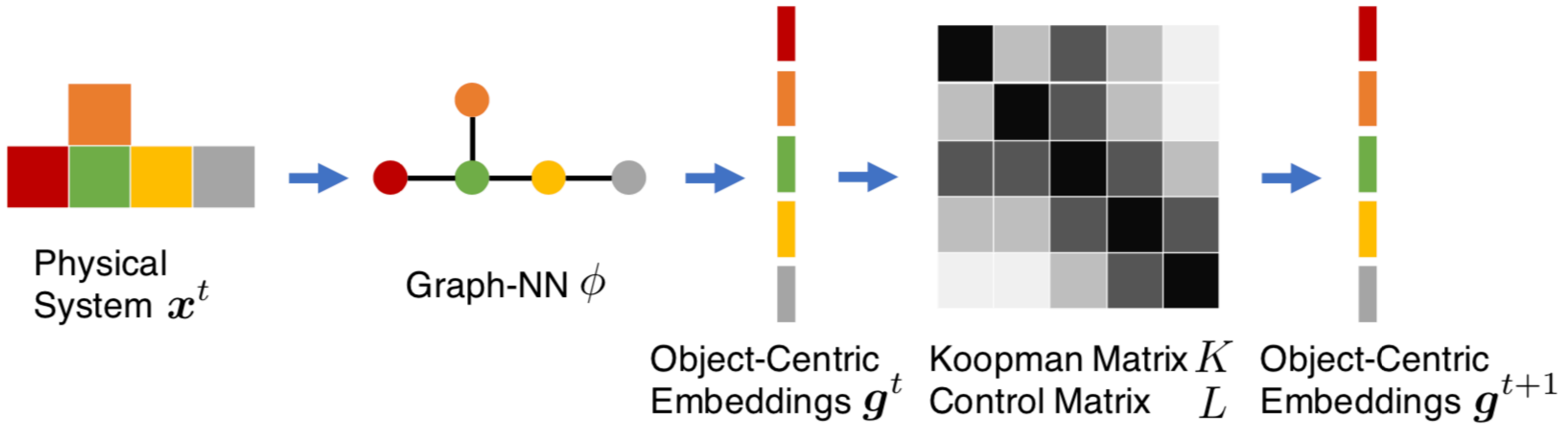
$$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$$



Compositional Koopman Operators

Three observations from the spring system:
(1) The system state is composed of the state of each individual object.
(2) The transition matrix has a block-wise substructure.
(3) The same physical interactions share the same transition block.

Assuming
$$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$$



(2) The Koopman matrix has a block-wise structure.
(3) The same physical interactions shall share the same transition block.

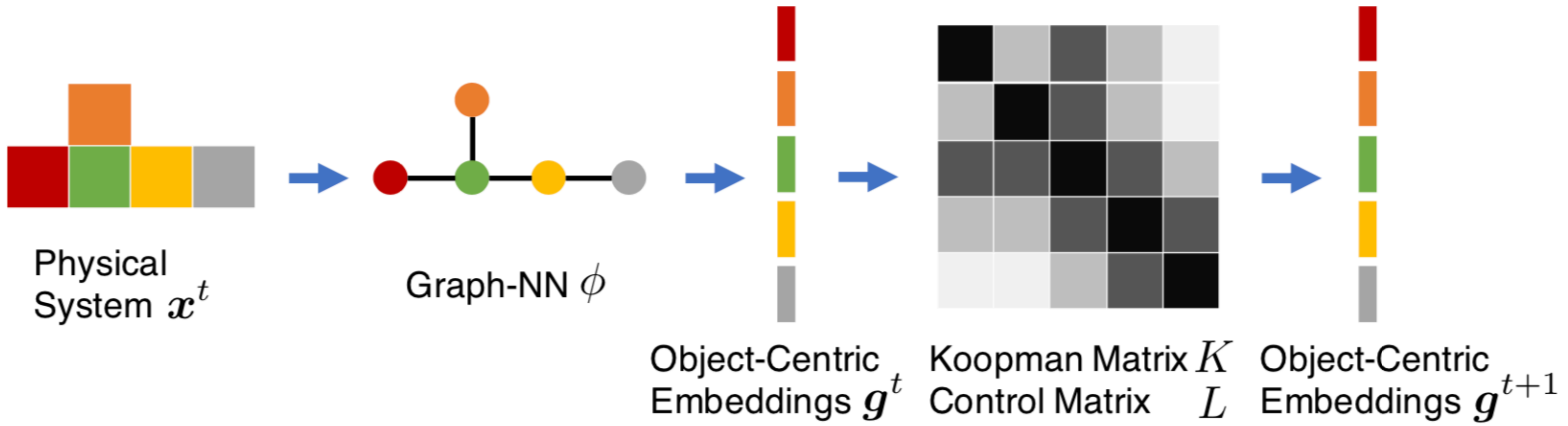
$$\begin{bmatrix} \boldsymbol{g}_1^{t+1} \\ \vdots \\ \boldsymbol{g}_N^{t+1} \end{bmatrix} = \begin{bmatrix} K_{11} & \cdots & K_{1N} \\ \vdots & \ddots & \vdots \\ K_{N1} & \cdots & K_{NN} \end{bmatrix} \begin{bmatrix} \boldsymbol{g}_1^{t} \\ \vdots \\ \boldsymbol{g}_N^{t} \end{bmatrix} + \begin{bmatrix} L_{11} & \cdots & L_{1N} \\ \vdots & \ddots & \vdots \\ L_{N1} & \cdots & L_{NN} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^{t} \\ \vdots \\ \boldsymbol{u}_N^{t} \end{bmatrix}$$

Compositional Koopman Operators

Three observations from the spring system:
(1) The system state is composed of the state of each individual object.
(2) The transition matrix has a block-wise substructure.
(3) The same physical interactions share the same transition block.

Assuming

$$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$$



Block-wise structure of the Koopman matrix:
1. Each block encodes an interaction.
2. Block can share parameters which significantly reduce its parameters.

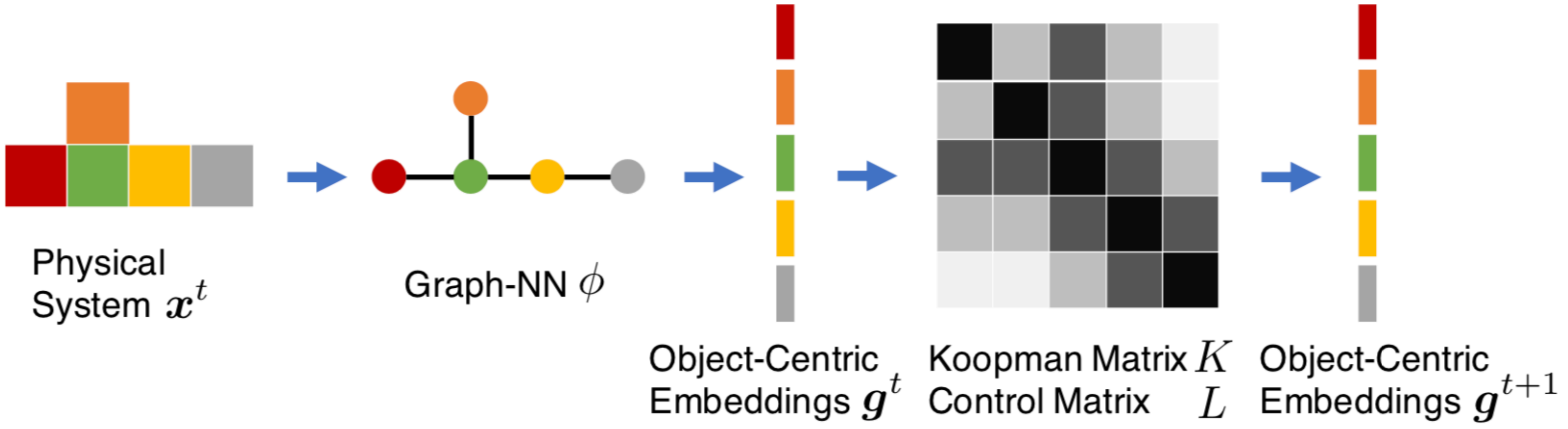
(2) The Koopman matrix has a block-wise structure.
(3) The same physical interactions shall share the same transition block.

$$\begin{bmatrix} \boldsymbol{g}_1^{t+1} \\ \vdots \\ \boldsymbol{g}_N^{t+1} \end{bmatrix} = \begin{bmatrix} K_{11} & \cdots & K_{1N} \\ \vdots & \ddots & \vdots \\ K_{N1} & \cdots & K_{NN} \end{bmatrix} \begin{bmatrix} \boldsymbol{g}_1^{t} \\ \vdots \\ \boldsymbol{g}_N^{t} \end{bmatrix} + \begin{bmatrix} L_{11} & \cdots & L_{1N} \\ \vdots & \ddots & \vdots \\ L_{N1} & \cdots & L_{NN} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^{t} \\ \vdots \\ \boldsymbol{u}_N^{t} \end{bmatrix}$$

Compositional Koopman Operators

Three observations from the spring system:
(1) The system state is composed of the state of each individual object.
(2) The transition matrix has a block-wise substructure.
(3) The same physical interactions share the same transition block.

Assuming
$$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$$



(2) The Koopman matrix has a block-wise structure.
(3) The same physical interactions shall share the same transition block.

$$\begin{bmatrix} \boldsymbol{g}_1^{t+1} \\ \vdots \\ \boldsymbol{g}_N^{t+1} \end{bmatrix} = \begin{bmatrix} K_{11} & \cdots & K_{1N} \\ \vdots & \ddots & \vdots \\ K_{N1} & \cdots & K_{NN} \end{bmatrix} \begin{bmatrix} \boldsymbol{g}_1^{t} \\ \vdots \\ \boldsymbol{g}_N^{t} \end{bmatrix} + \begin{bmatrix} L_{11} & \cdots & L_{1N} \\ \vdots & \ddots & \vdots \\ L_{N1} & \cdots & L_{NN} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^{t} \\ \vdots \\ \boldsymbol{u}_N^{t} \end{bmatrix}$$

Compositional Koopman Operators

Three observations from the spring system:

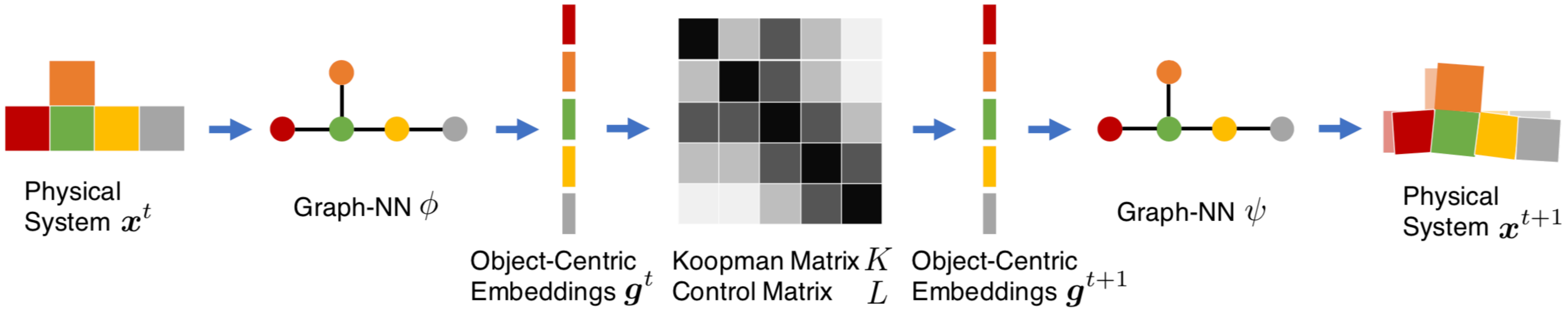
(1) The system state is composed of the state of each individual object.

(2) The transition matrix has a block-wise substructure.

(3) The same physical interactions share the same transition block.

Assuming

$$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$$

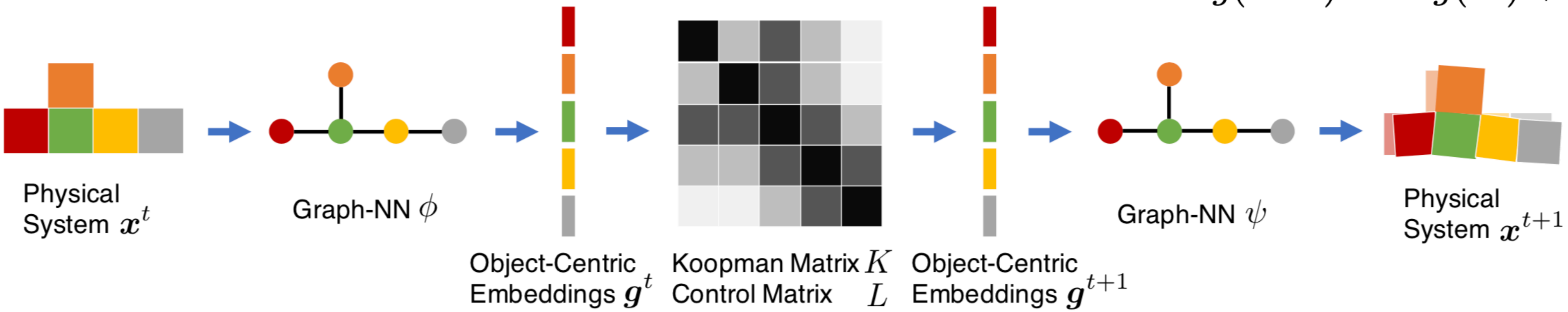


Graph neural network to decode the new state.

Compositional Koopman Operators

Assuming

$$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$$

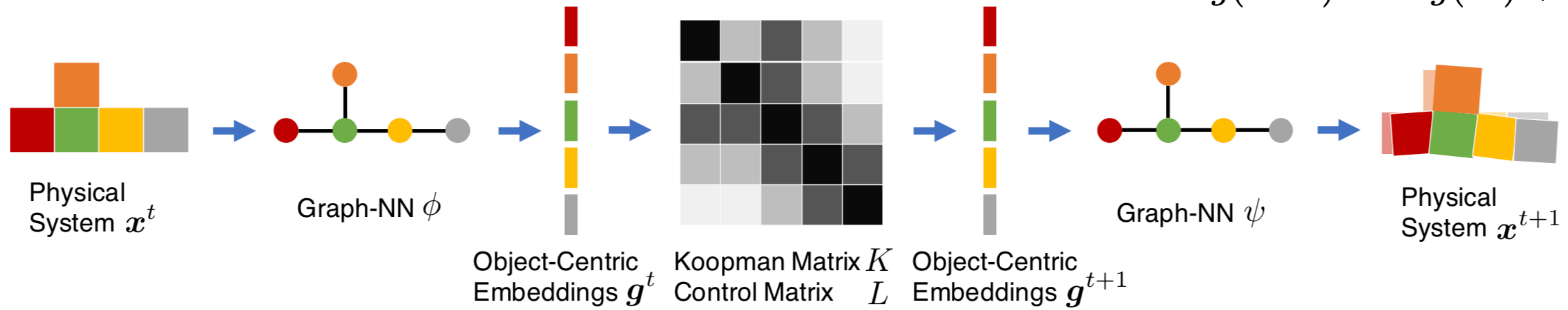


Training

Compositional Koopman Operators

Assuming

$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$



Training

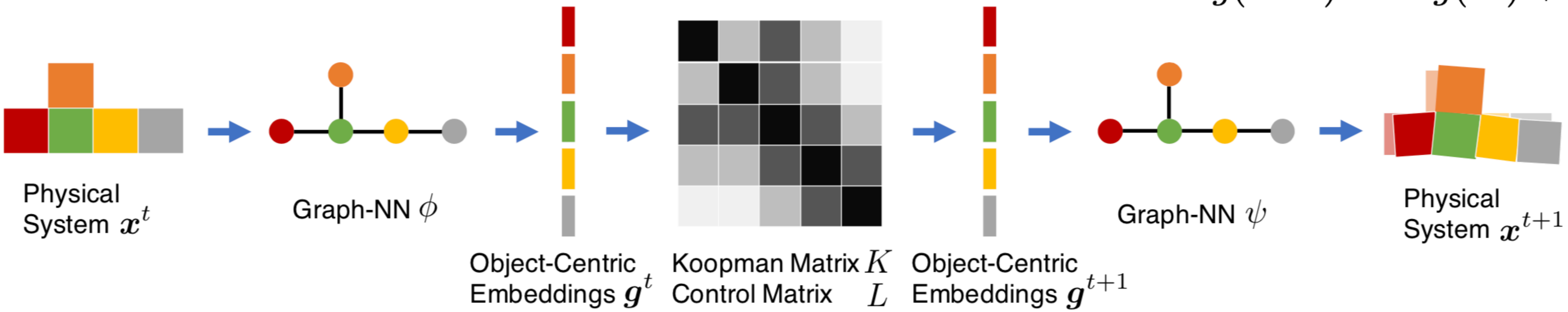
- System Identification
  - Given $g(\boldsymbol{x}^t)$ and action $\boldsymbol{u}^t, t = 0, \ldots, T$
  - Solve for K and L.
  - Least-square fitting.

$$\min_{K,L} \|K\boldsymbol{g}^{1:T-1} + L\widetilde{\boldsymbol{u}} - \boldsymbol{g}^{2:T}\|_2$$

Compositional Koopman Operators

Assuming

$$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$$



Training

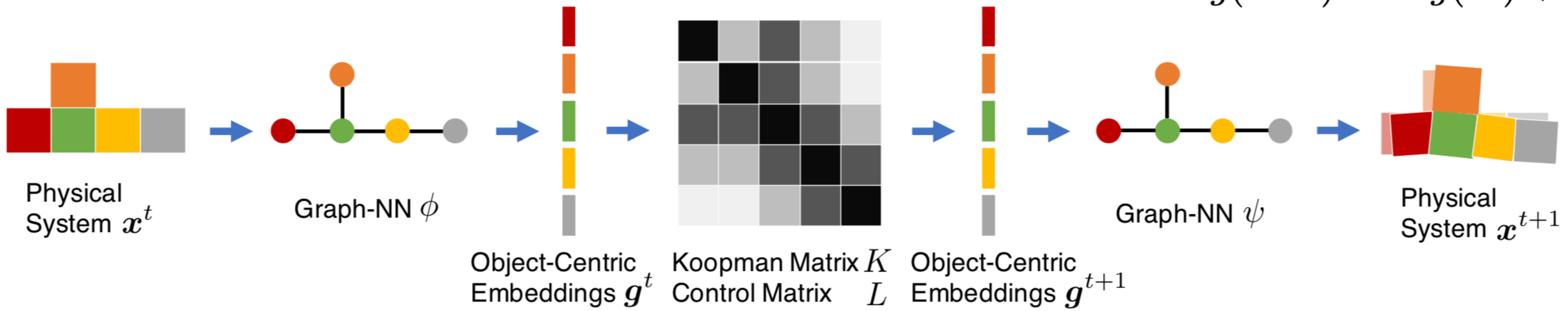
Auto-encoding $\mathcal{L}_{\text{ae}} = \frac{1}{T}\sum_{i}^{T} \|\psi(\phi(\boldsymbol{x}^i)) - \boldsymbol{x}^i\|$

- System Identification
  - Given $g(\boldsymbol{x}^t)$ and action $\boldsymbol{u}^t, t = 0, \ldots, T$
  - Solve for K and L.
  - Least-square fitting.

$$\min_{K,L} \|K\boldsymbol{g}^{1:T-1} + L\tilde{\boldsymbol{u}} - \boldsymbol{g}^{2:T}\|_2$$

Compositional Koopman Operators

Assuming
$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$



Training

- System Identification
  - Given $g(\boldsymbol{x}^t)$ and action $\boldsymbol{u}^t, t = 0, \ldots, T$
  - Solve for K and L.
  - Least-square fitting.

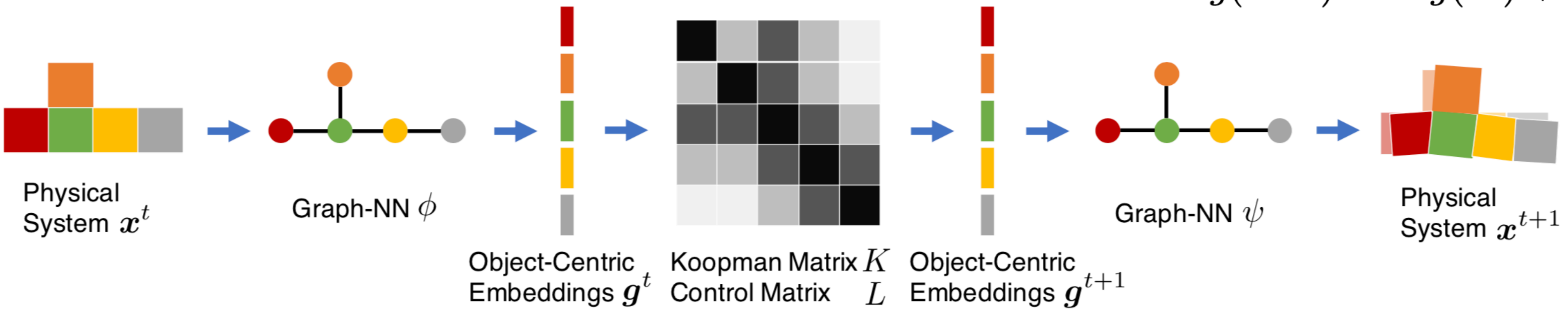
$$\min_{K,L} \|K\boldsymbol{g}^{1:T-1} + L\tilde{\boldsymbol{u}} - \boldsymbol{g}^{2:T}\|_2$$

Auto-encoding $\mathcal{L}_{\text{ae}} = \frac{1}{T}\sum_{i}^{T} \|\psi(\phi(\boldsymbol{x}^i)) - \boldsymbol{x}^i\|$

Prediction loss $\mathcal{L}_{\text{pred}} = \frac{1}{T}\sum_{i=1}^{T} \|\psi(\hat{\boldsymbol{g}}^i) - \boldsymbol{x}^i\|$

Compositional Koopman Operators

Assuming
$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$



Training

- System Identification
  - Given $g(\boldsymbol{x}^t)$ and action $\boldsymbol{u}^t$, $t = 0, \ldots, T$
  - Solve for K and L.
  - Least-square fitting.

$$\min_{K,L} \|K\boldsymbol{g}^{1:T-1} + L\tilde{\boldsymbol{u}} - \boldsymbol{g}^{2:T}\|_2$$

Auto-encoding $\mathcal{L}_{\text{ae}} = \frac{1}{T}\sum_i^T \|\psi(\phi(\boldsymbol{x}^i)) - \boldsymbol{x}^i\|$

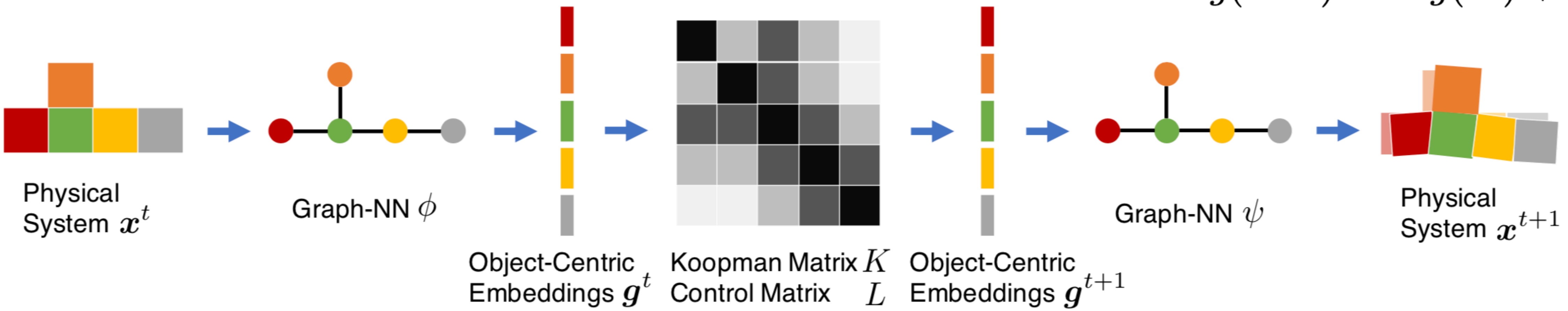
Prediction loss $\mathcal{L}_{\text{pred}} = \frac{1}{T}\sum_{i=1}^T \|\psi(\hat{\boldsymbol{g}}^i) - \boldsymbol{x}^i\|$

Metric loss $\mathcal{L}_{\text{metric}} = \sum_{ij} \left| \|\boldsymbol{g}^i - \boldsymbol{g}^j\| - \|\boldsymbol{x}^i - \boldsymbol{x}^j\| \right|$

Compositional Koopman Operators

Assuming

$$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^{t}) + L\boldsymbol{u}^{t}$$



Training

- System Identification
  - Given $g(\boldsymbol{x}^t)$ and action $\boldsymbol{u}^t, t = 0, \ldots, T$
  - Solve for K and L.
  - Least-square fitting.

$$\min_{K,L} \|K\boldsymbol{g}^{1:T-1} + L\tilde{\boldsymbol{u}} - \boldsymbol{g}^{2:T}\|_2$$

Auto-encoding $\mathcal{L}_{\text{ae}} = \frac{1}{T}\sum_{i}^{T} \|\psi(\phi(\boldsymbol{x}^i)) - \boldsymbol{x}^i\|$

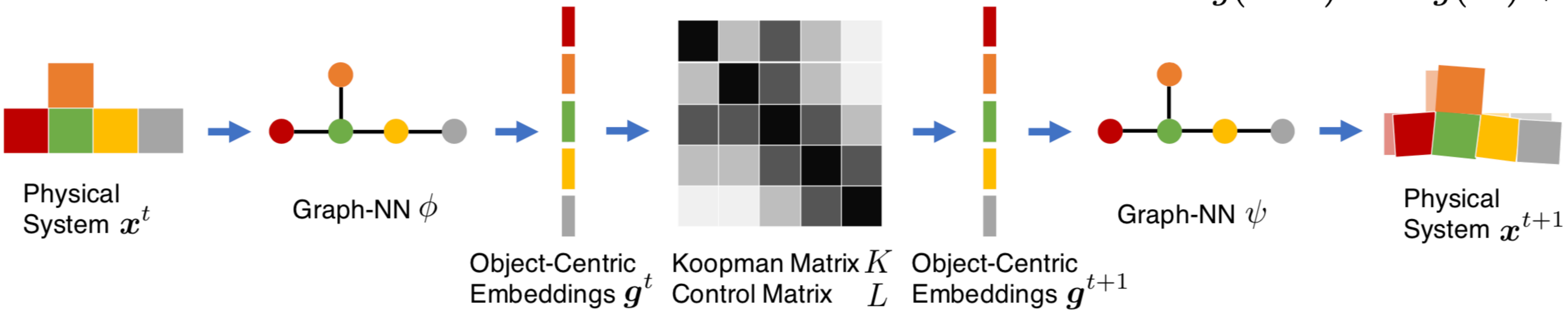
Prediction loss $\mathcal{L}_{\text{pred}} = \frac{1}{T}\sum_{i=1}^{T} \|\psi(\hat{\boldsymbol{g}}^i) - \boldsymbol{x}^i\|$

Metric loss $\mathcal{L}_{\text{metric}} = \sum_{ij} \left| \|\boldsymbol{g}^i - \boldsymbol{g}^j\| - \|\boldsymbol{x}^i - \boldsymbol{x}^j\| \right|$

$$\mathcal{L} = \mathcal{L}_{\text{ae}} + \lambda_1 \mathcal{L}_{\text{pred}} + \lambda_2 \mathcal{L}_{\text{metric}}$$

Compositional Koopman Operators

Assuming
$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$



Test time

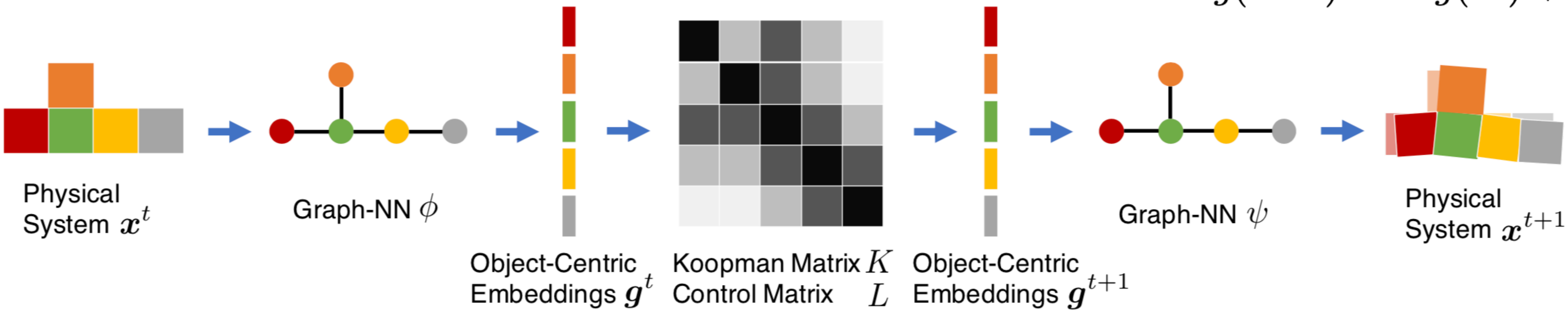
- System Identification / Online adaptation
  - Given $g(\boldsymbol{x}^t)$ and action $\boldsymbol{u}^t$, $t = 0, \ldots, T$
  - Solve for K and L.
  - Least-square fitting.

$$\min_{K,L} \|K\boldsymbol{g}^{1:T-1} + L\tilde{\boldsymbol{u}} - \boldsymbol{g}^{2:T}\|_2$$

Compositional Koopman Operators

Assuming

$g(\boldsymbol{x}^{t+1}) = Kg(\boldsymbol{x}^t) + L\boldsymbol{u}^t$



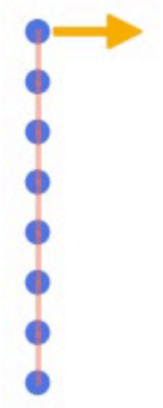
Test time

- System Identification / Online adaptation
  - Given $g(\boldsymbol{x}^t)$ and action $\boldsymbol{u}^t$, $t = 0, \ldots, T$
  - Solve for K and L.
  - Least-square fitting.

$$\min_{K,L} \|K\boldsymbol{g}^{1:T-1} + L\tilde{\boldsymbol{u}} - \boldsymbol{g}^{2:T}\|_2$$

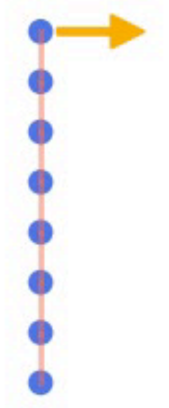
- Control Synthesis,
  - Given $g(\boldsymbol{x}^0)$, $g(\boldsymbol{x}^T)$, K and L.
  - Solve for $\boldsymbol{u}^t, t = 0, \ldots, T$
  - Quadratic programing (QP).

Experiments

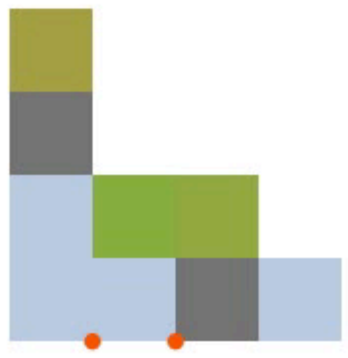


1) Manipulating
a Rope

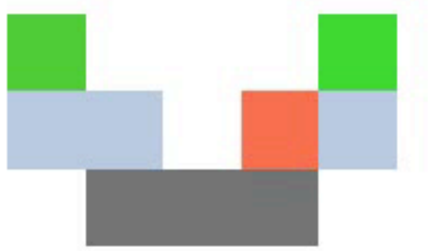
Experiments



1) Manipulating a Rope



2) Controlling a soft robot to swing



3) Controlling a soft robot to swim in fluids

Soft Tissue

Rigid Tissue

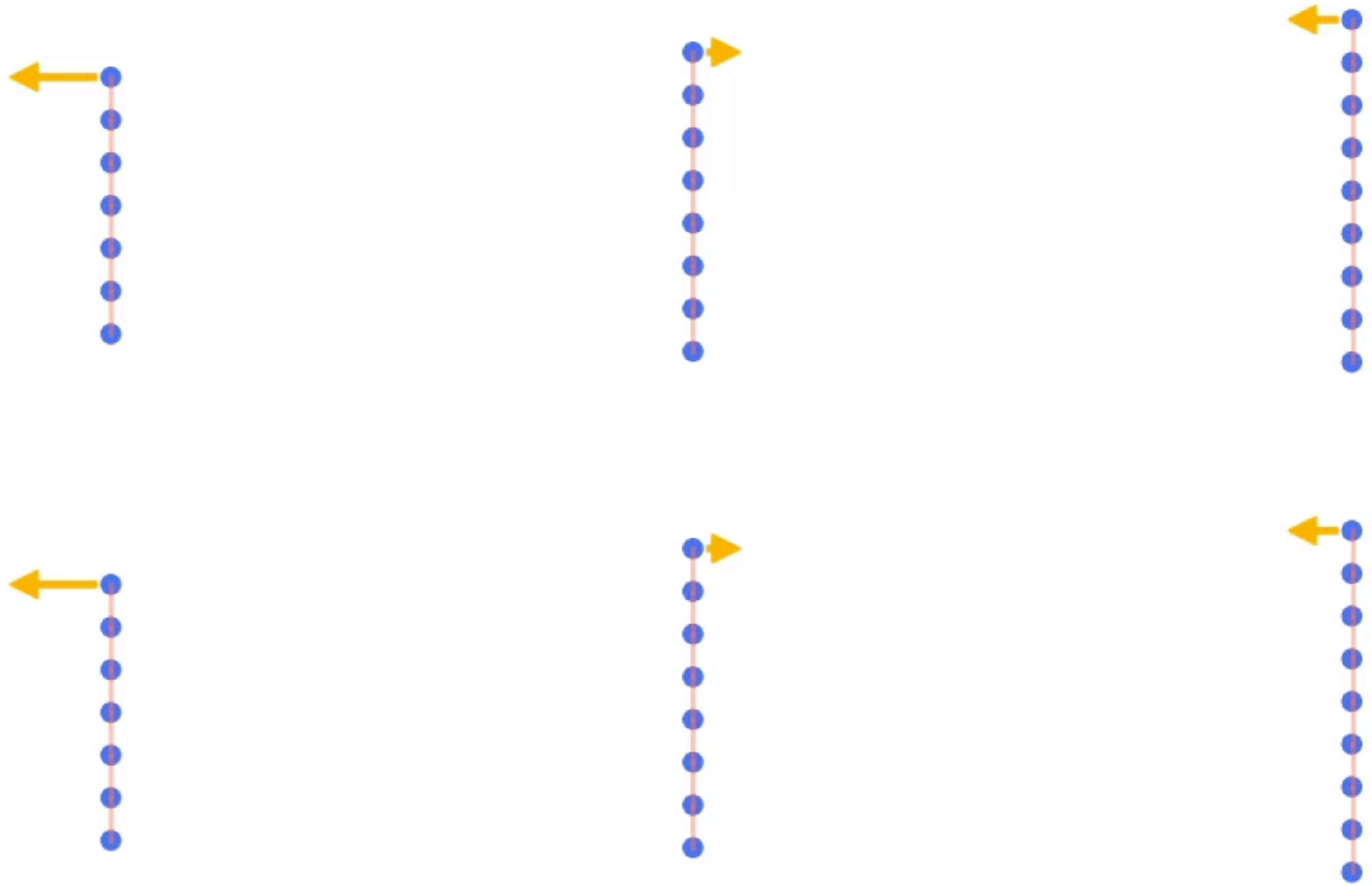
Contracting Actuator

Expanding Actuator

Rope Manipulation (Simulation)

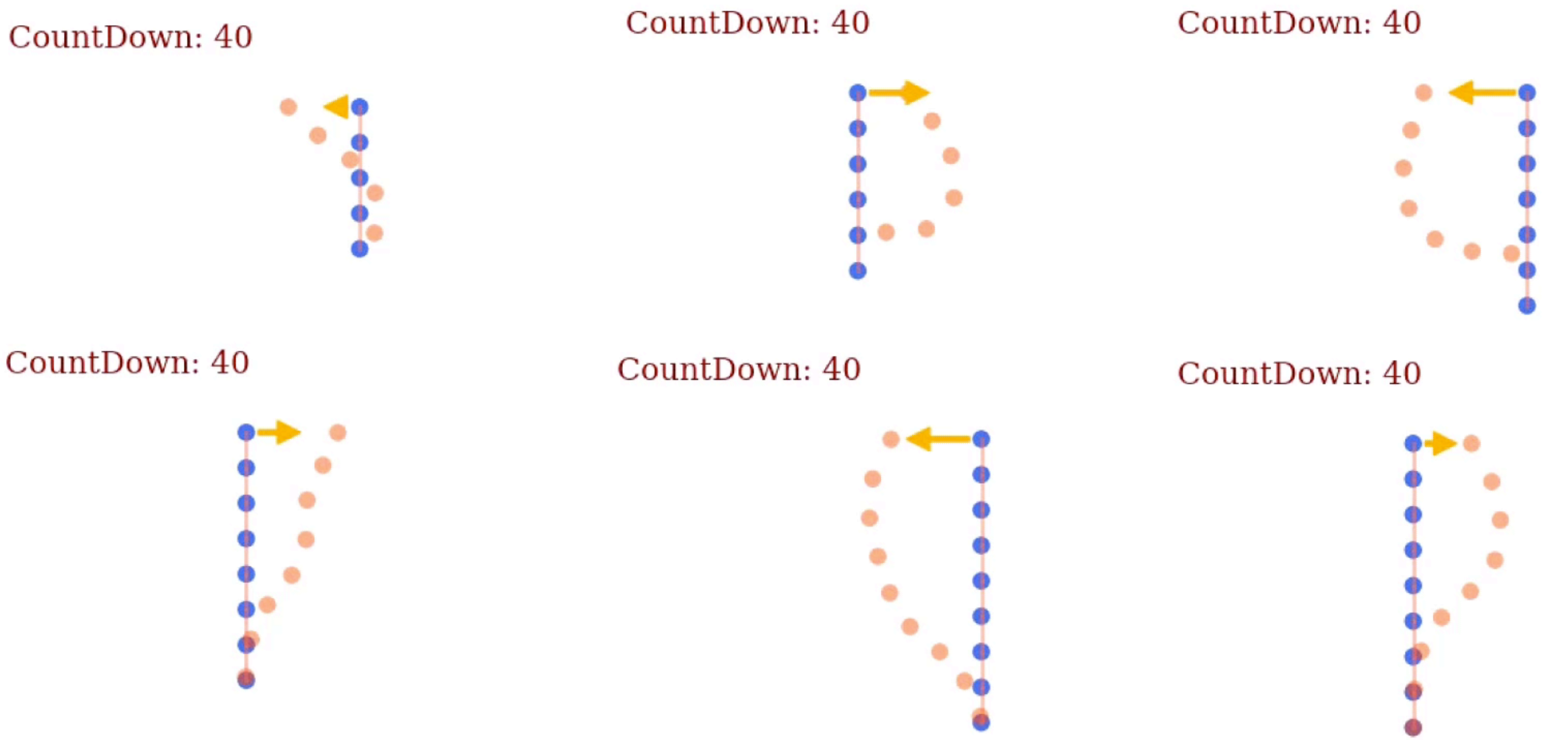
Our Prediction

Ground Truth

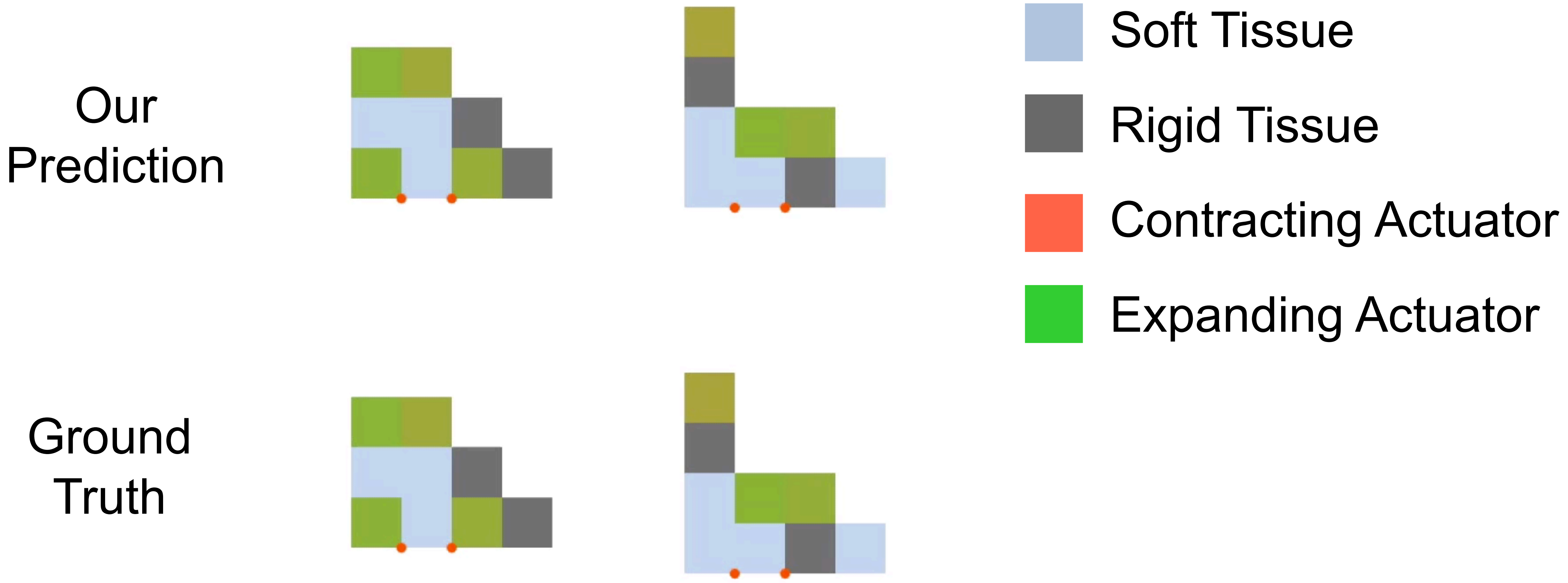


Rope Manipulation (Control)

Target state is shown as red dots.

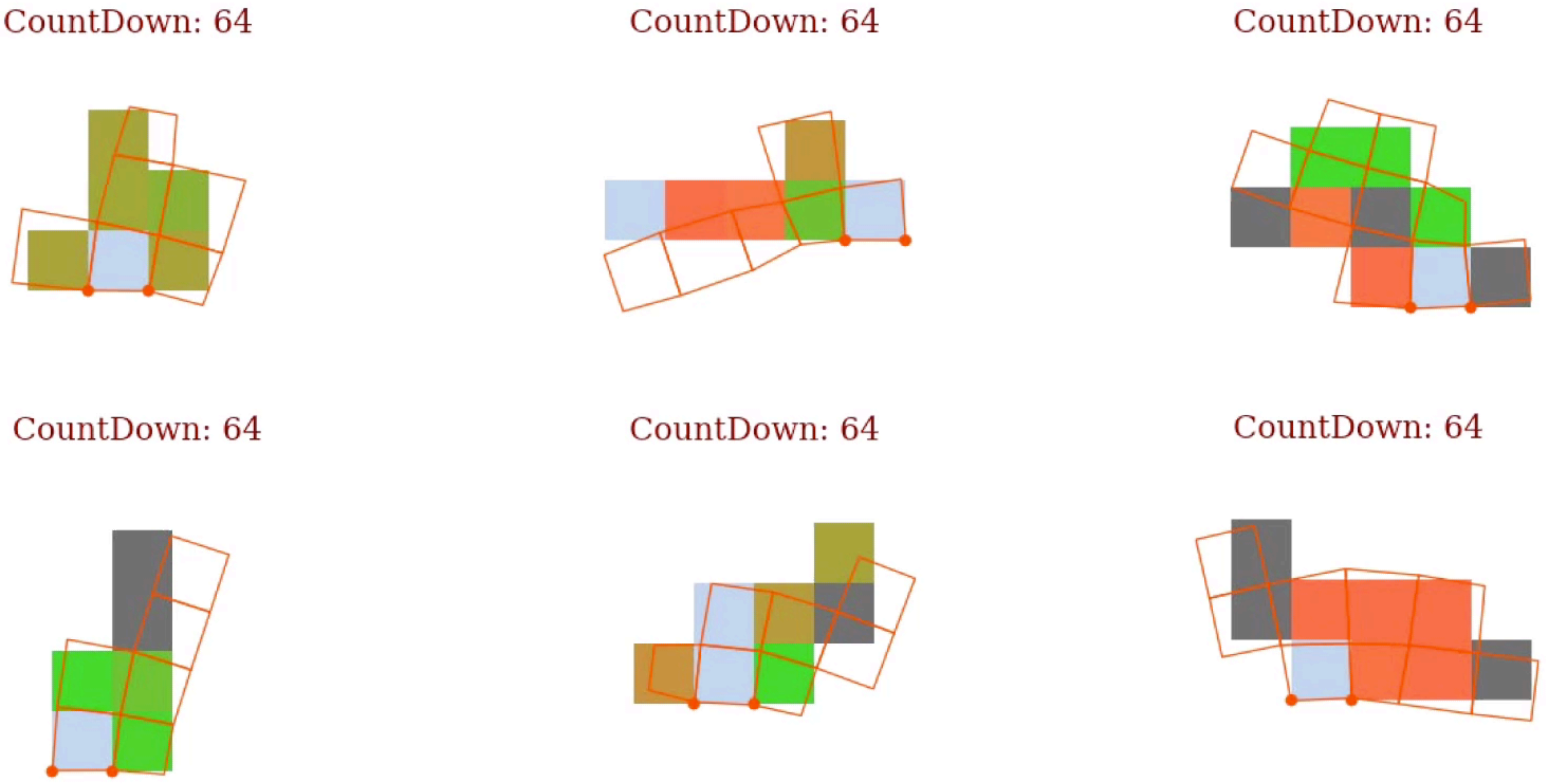


Soft Robot Swing (Simulation)

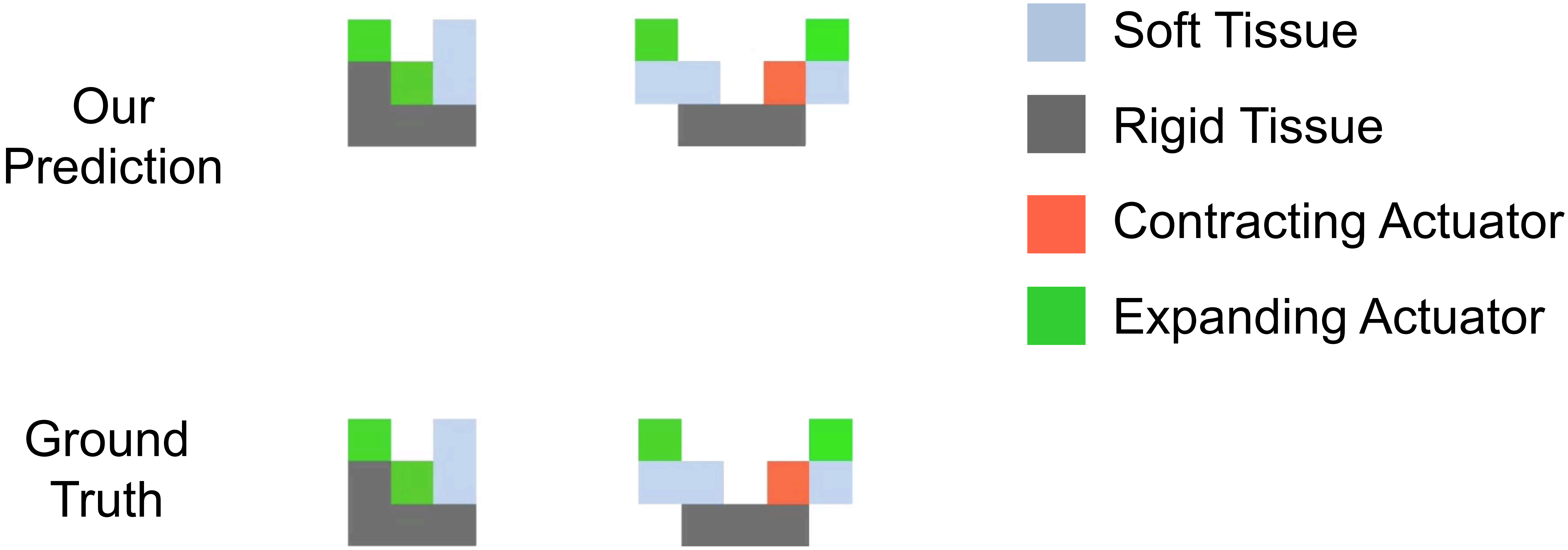


Soft Robot Swing (Control)

Target state is shown as red grids.

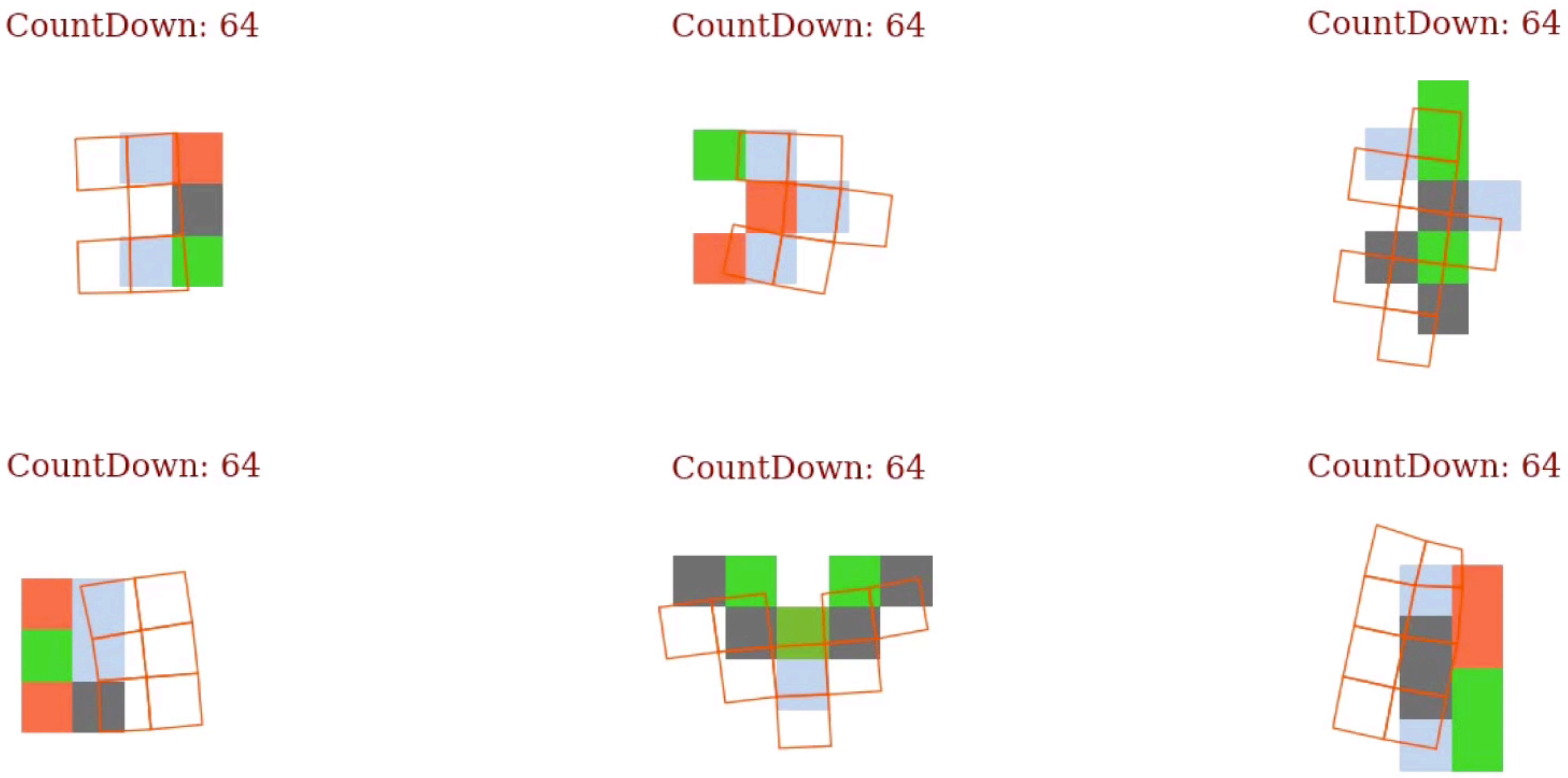


Soft Robot Swim (Simulation)



Soft Robot Swim (Control)

Target state is shown as red grids.



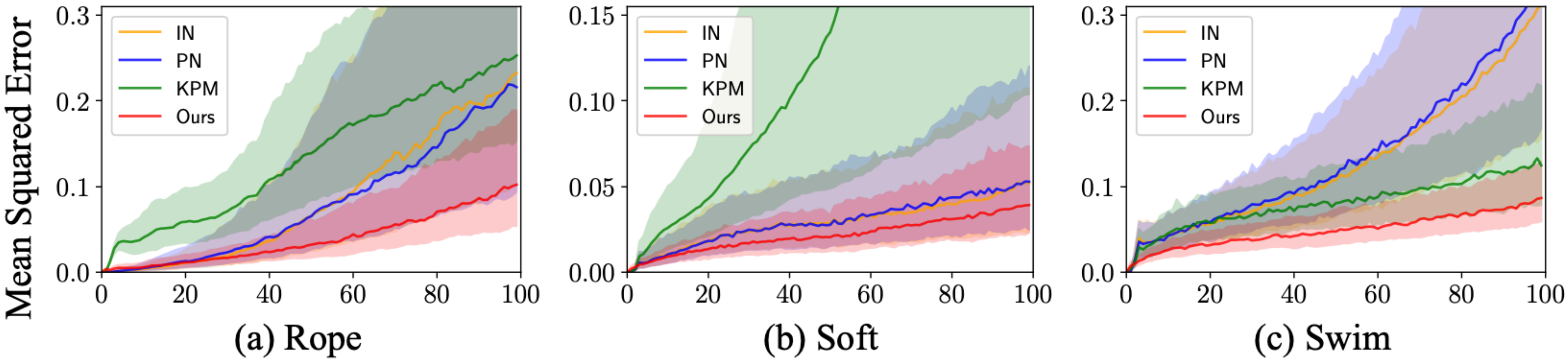


Figure 3: **Quantitative results on simulation.** The x axis shows time steps. The solid lines indicate medians and the transparent regions are the interquartile ranges of simulation errors. Our method significantly outperforms the baselines in all testing environments.

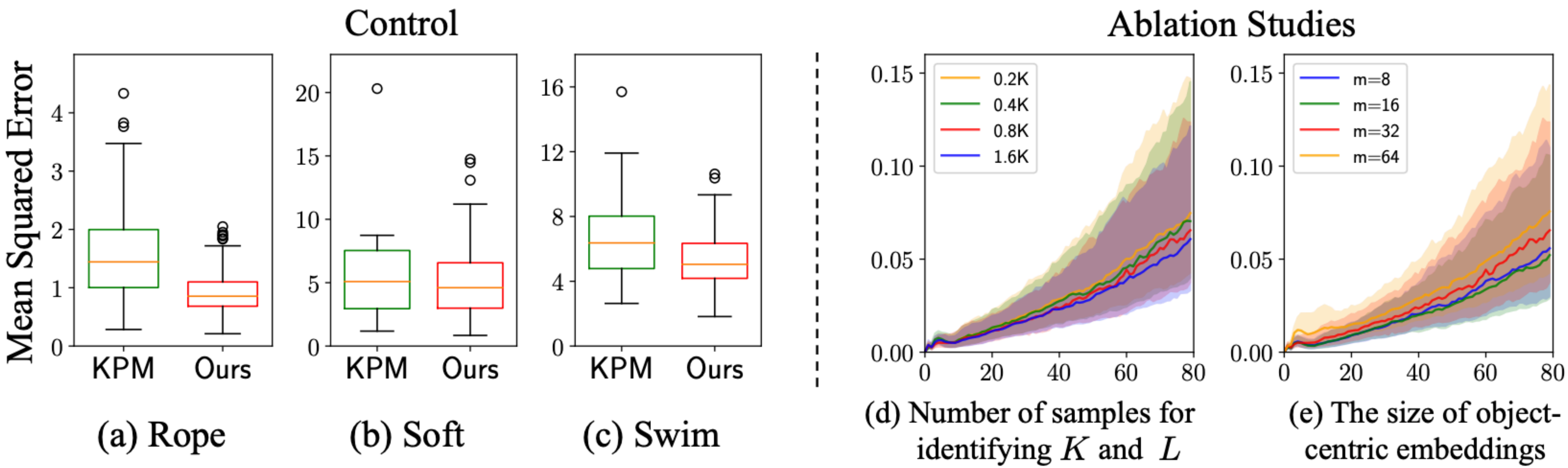


Figure 4: **Quantitative results on control and ablation studies on model hyperparameters.** Left: box-plots show the distributions of control errors. The yellow line in the box indicates the median. Our model consistently achieves smaller errors in all environments against KPM. Right: our model's simulation errors with different amount of data for system identification (d) and different dimensions of the Koopman space (e).

Table 1: Ablation study results on the Koopman matrix structure (Rope environment). For simulation, we show the Mean Squared Error between the prediction and the ground truth at $T = 100$, whereas for control, we show the performance with a horizon of length 40. The numbers in parentheses show the performance on extrapolation.

	Simulation	Control
Diag	0.133 (0.174)	2.337 (2.809)
None	0.117 (0.083)	1.522 (1.288)
Block	**0.105 (0.075)**	**0.854 (1.101)**

Summary

- We propose to combine graph neural networks and Koopman Operator Theory

Our formulation

- Captures the compositional structures of the underlying system
- Generalizes to systems with variable numbers of components
- Generalizes to systems with different configurations

Summary

- We propose to combine graph neural networks and Koopman Operator Theory

Our formulation

- Captures the compositional structures of the underlying system
- Generalizes to systems with variable numbers of components
- Generalizes to systems with different configurations

The internal linear structure allows

- Quick adaptation to system of unknown physical parameters
    - via Least Squares Regression
- Efficient control synthesis
    - via Quadratic Programming (QP)

Limitation and Future Studies

- Assuming the underlying dynamics is smooth or a few times differentiable.
- Did not succeed for modeling hard contact.

Limitation and Future Studies

- Assuming the underlying dynamics is smooth or a few times differentiable.
- Did not succeed for modeling hard contact.

- Adapt to more complicated/realistic scenarios
  - Deformable objects / Soft robots
  - Fluid / Granular materials
  - Cloth / Rope manipulation

Limitation and Future Studies

- Assuming the underlying dynamics is smooth or a few times differentiable.
- Did not succeed for modeling hard contact.

- Adapt to more complicated/realistic scenarios
  - Deformable objects / Soft robots
  - Fluid / Granular materials
  - Cloth / Rope manipulation
- How well can it cope with different state representations?

Limitation and Future Studies

- Assuming the underlying dynamics is smooth or a few times differentiable.
- Did not succeed for modeling hard contact.

- Adapt to more complicated/realistic scenarios
  - Deformable objects / Soft robots
  - Fluid / Granular materials
  - Cloth / Rope manipulation
- How well can it cope with different state representations?
- Extend to piecewise affine model
  - Fewer pieces to cover the state space

Limitation and Future Studies

- Assuming the underlying dynamics is smooth or a few times differentiable.
- Did not succeed for modeling hard contact.

- Adapt to more complicated/realistic scenarios
  - Deformable objects / Soft robots
  - Fluid / Granular materials
  - Cloth / Rope manipulation
- How well can it cope with different state representations?
- Extend to piecewise affine model
  - Fewer pieces to cover the state space
- Augment with policy function and/or value function

Limitation and Future Studies

- Assuming the underlying dynamics is smooth or a few times differentiable.
- Did not succeed for modeling hard contact.

- Adapt to more complicated/realistic scenarios
  - Deformable objects / Soft robots
  - Fluid / Granular materials
  - Cloth / Rope manipulation
- How well can it cope with different state representations?
- Extend to piecewise affine model
  - Fewer pieces to cover the state space
- Augment with policy function and/or value function
- More theoretical probe on the discrepancy between the Koopman and the state space


Cagen
EATER

Collaborators


Hao He


Jiajun Wu


Dina Katabi


Antonio Torralba


Russ Tedrake

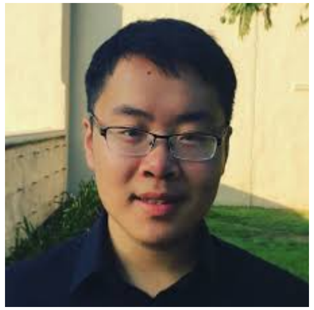

Joshua B. Tenenbaum

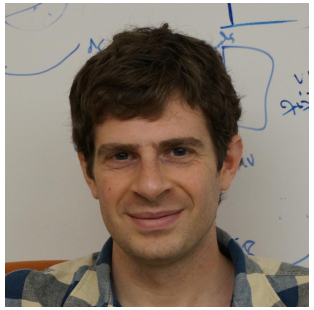

Animesh Garg


Dieter Fox

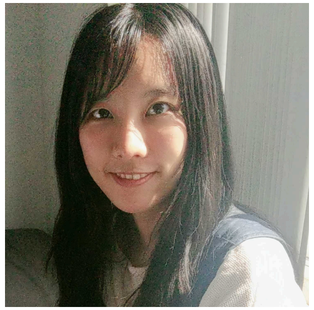

Animashree Anandkumar


Daniel L.K. Yamins


Kexin Yi


Daniel M. Bear


Chuang Gan


Toru Lin


Jun-Yan Zhu