A Compositional Object-Based Approach to Learning Physical Dynamics

Michael Chang[1,2,4], Tomer Ullman[1,3], Antonio Torralba[1,4], Joshua B. Tenenbaum[1,3]

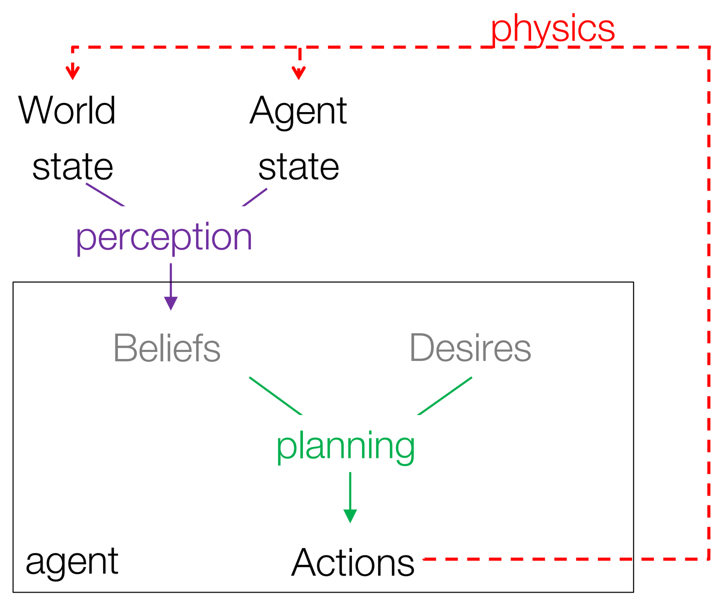
1 MIT 2 SuperUROP 3 brain+cognitive sciences 4 MIT EECS Electrical Engineering and Computer Science

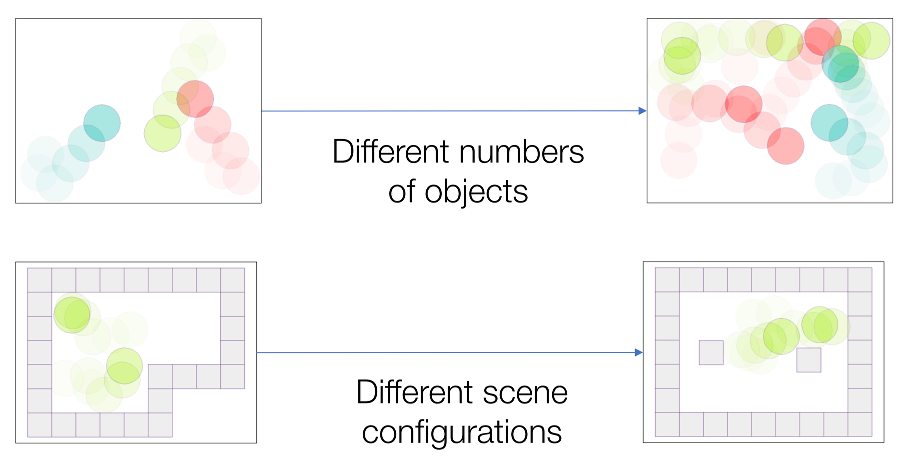
Project Webpage: http://mbchang.github.io/npe

Code: http://github.com/mbchang/dynamics

1 Vision

Vision: In order to rapidly learn new tasks and flexibly adapt to changes in inputs and goals, an agent needs a prior on physics that allows it to naturally generalize reasoning to novel scenes.





Approach: Endow the agent with the prior as a learned physics simulator.

This work: We present the Neural Physics Engine, a framework for learning simulators of intuitive physics that naturally generalizes across variable object count and different scene configurations.

2 Steps

Combining the advantages of symbolic and neural models

How to incorporate inductive biases that are not only strong enough to generalize across scenes, but also flexible enough to learn from and adapt to different inputs?

Symbolic Physics Engines

- Expressive
- Knowledge encoded in structure
- Difficult to adapt to scenarios outside description language

Neural Networks

- Adaptive
- Knowledge emerges through training
- Requires retraining for new scenarios

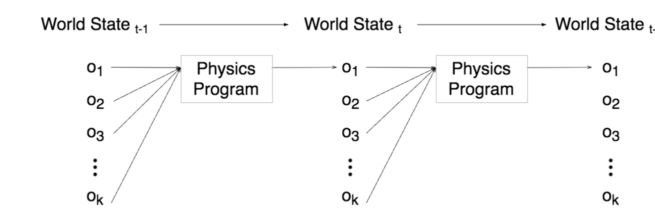
Neural Physics Engine (NPE)

- *Generic* notions of objects and their interactions
- Trained to model *specific* object properties and dynamics of different worlds
- Transfers knowledge across different object counts and scene configurations

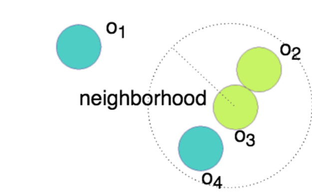
Ingredients useful for generalization

What are the primitives, means of combination, and means of abstraction that underlie the training and testing distributions, such that generalization comes for free?

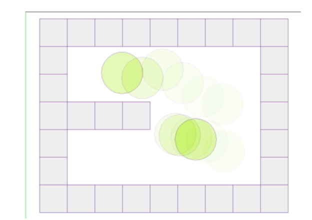
Object-based representations



Context Selection Mechanism

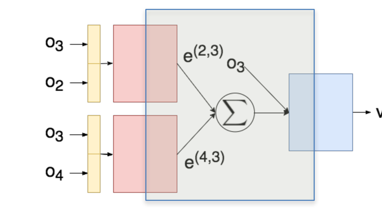


Factorization



Factorize larger objects into smaller building blocks

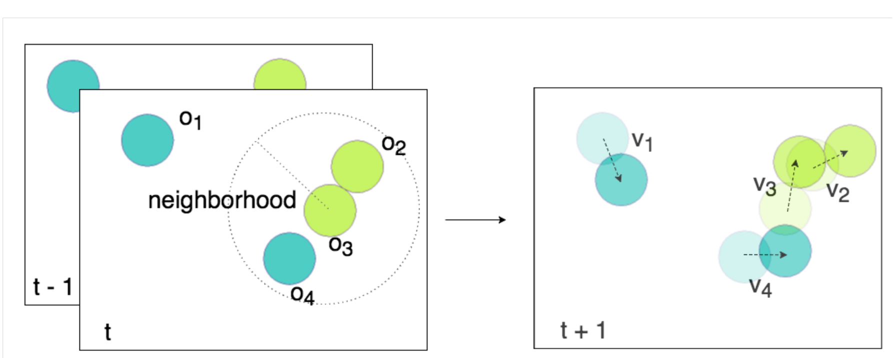
Compositionality



Compute prediction as a composition of pairwise interactions

3 Neural Physics Engine (NPE)

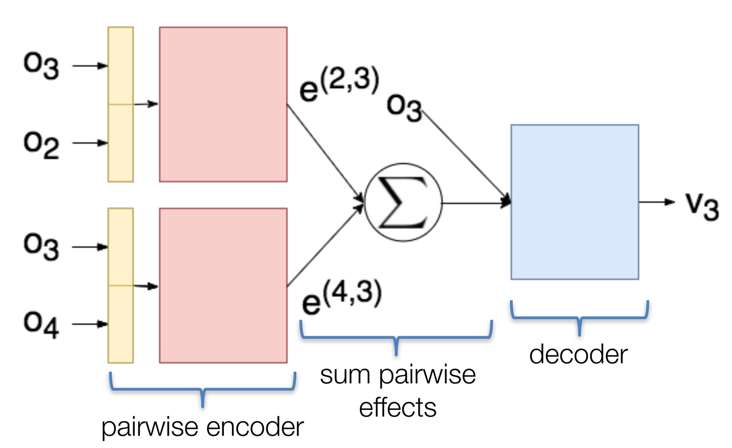
Scenario

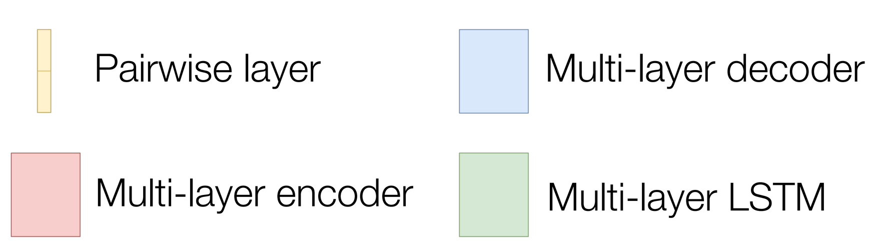


Predict the velocity of each object in turn, given the pairwise interactions with its neighborhood context objects.

Model Architecture

NPE applied on object 3

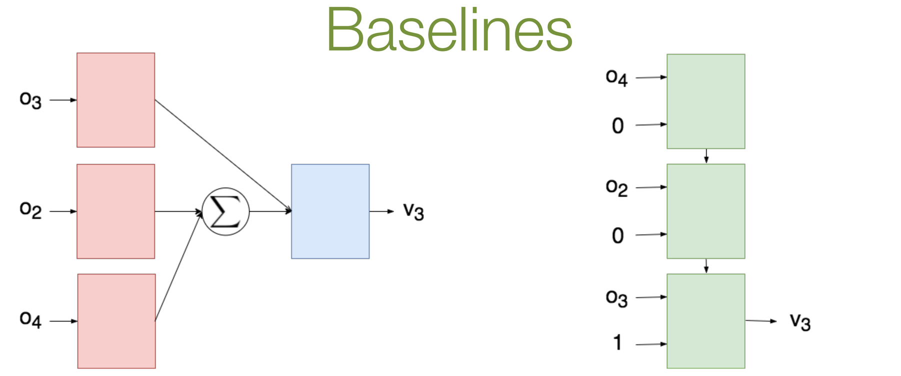




No-Pairwise (NP) Baseline: no pairwise factorization

LSTM Baseline: no composition of independent effects

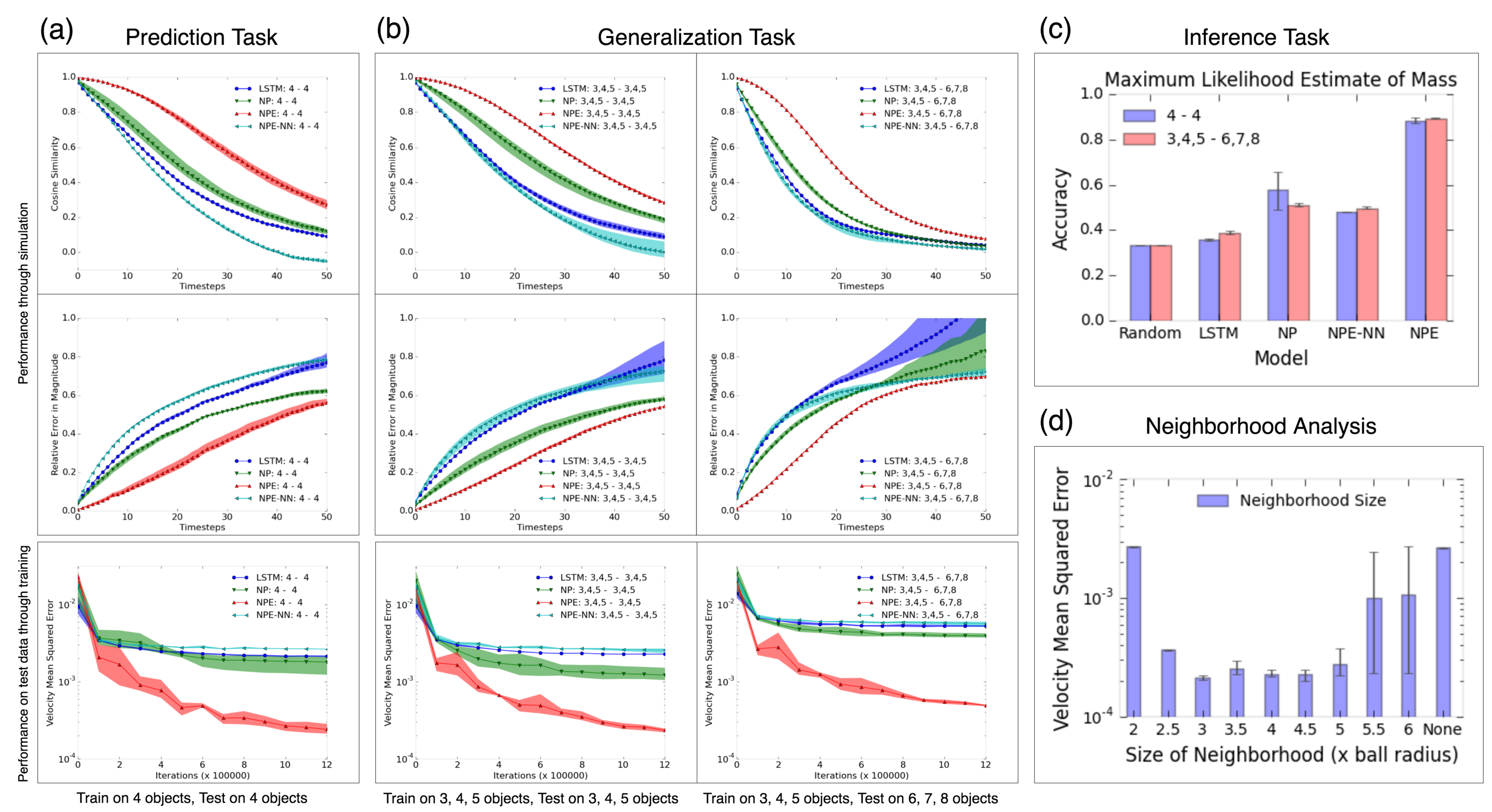
Baselines



NP applied on object 3

LSTM applied on object 3

4a Generalization: different numbers of objects



3, 4, 5: *worlds with fewer objects*

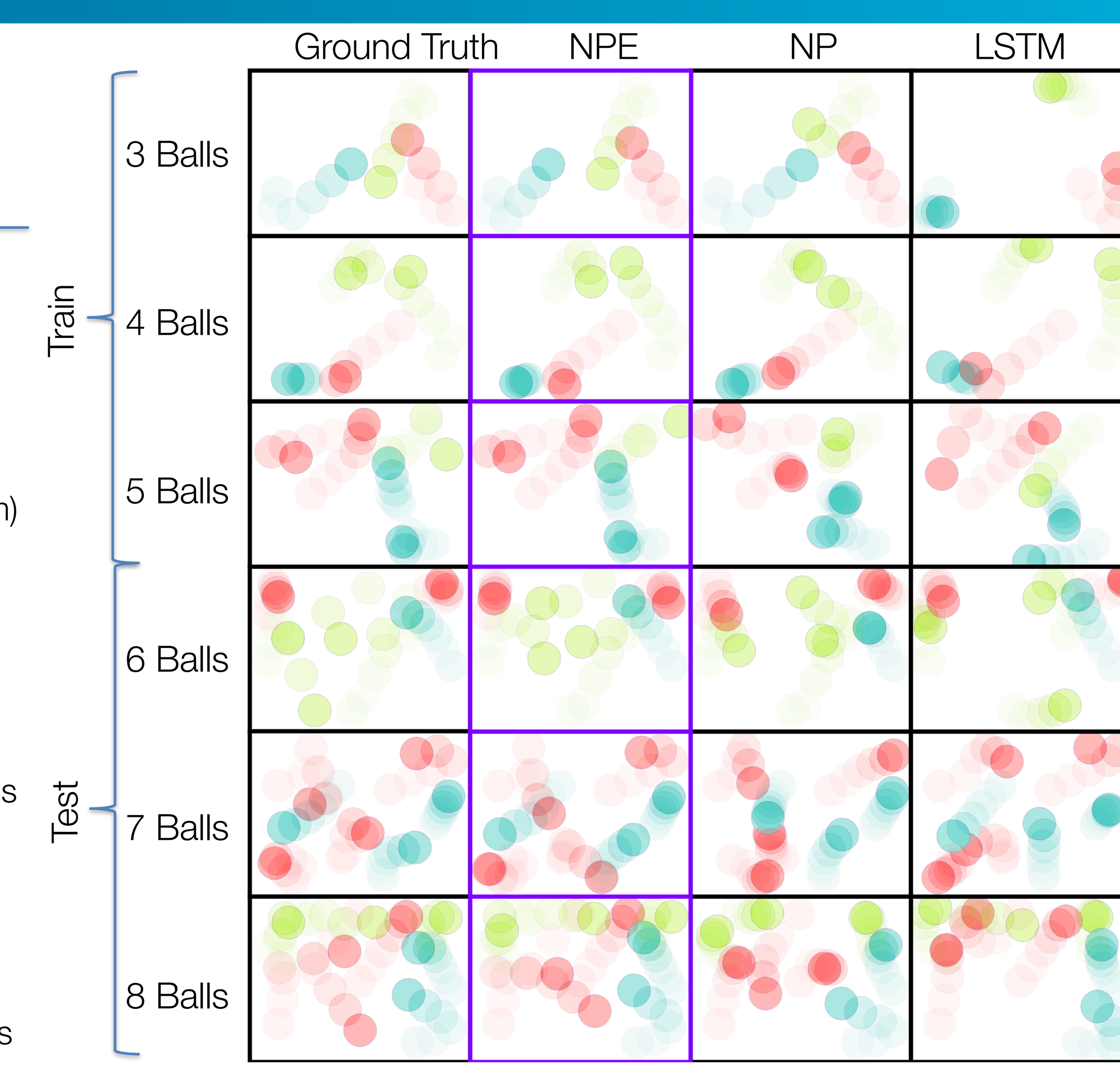
6, 7, 8: worlds with *more* objects

a) *Prediction Task:* Cosine similarity (top) and relative magnitude (middle) of prediction vs ground truth over 50 steps of simulation. Log MSE on velocity (bottom) over the course of training

b) *Generalization Task:* Same metrics as prediction, but train on 3, 4, 5; test on 6, 7, 8.

c) *Inference Task:* NPE gets about 90% accuracy in prediction and generalization setting.

d) *Neighborhood:* constrains the search space of context objects



5 Contributions

By combining the *expressiveness* of physics engines and the *adaptability* of neural networks in a *compositional* architecture that *naturally supports generalization* in fundamental aspects of physical reasoning, the Neural Physics Engine is an important step towards lifting an agent's *ability to think at a level of abstraction* where the concept of physics is *primitive*.

Contributions

- Presented a framework for learning a physics simulator
- Proposed ingredients useful for generalization
- Combined the strengths of symbolic and neural models in an object-based neural network
- Demonstrated an instantiation of the NPE framework for prediction, generalization, and inference tasks in worlds of balls and obstacles

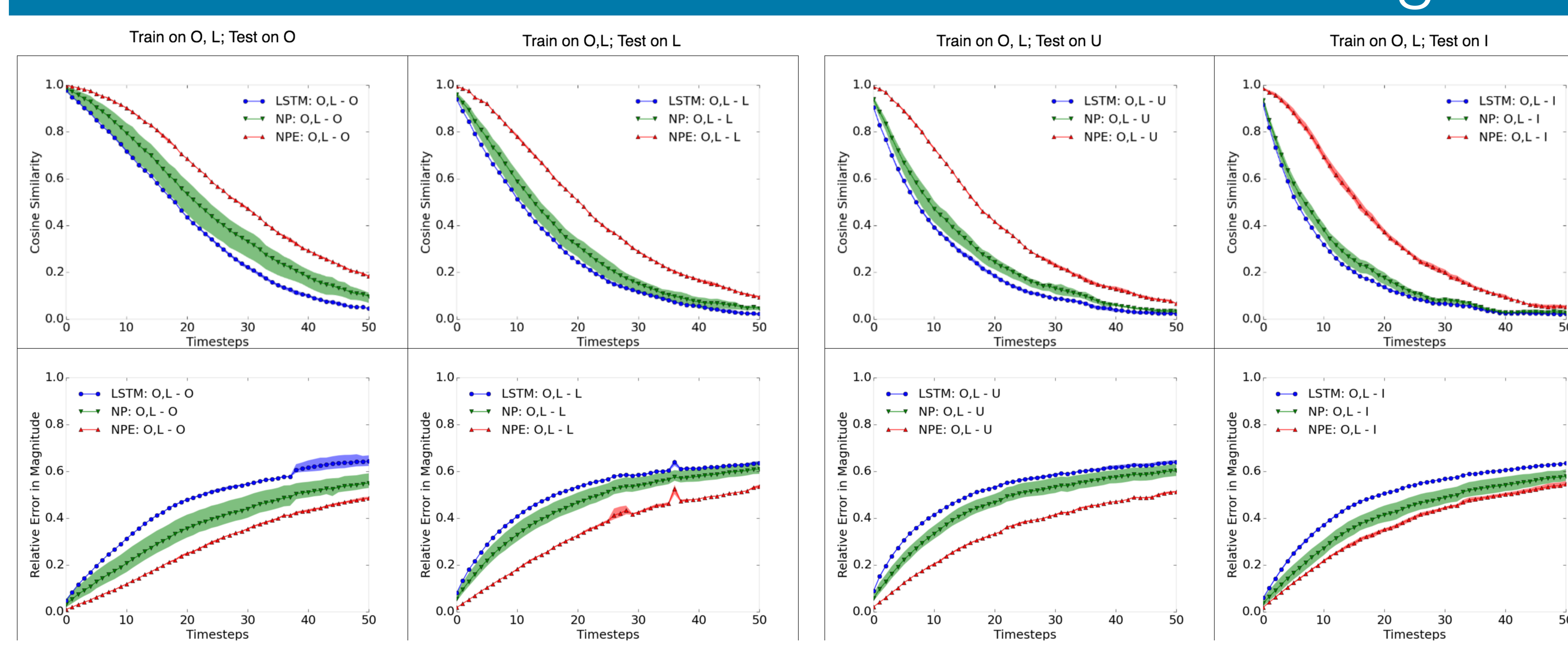
Ingredients useful for generalization

- Object-based representations
- Context-selection mechanism
- Factorization
- Compositionality

Simulation videos

https://goo.gl/BWYuOF

4b Generalization: different scene configurations



"O", "L": worlds *without* internal obstacles

"U", "I": worlds *with* internal obstacles

NPE does not overlap with internal obstacles, while the NP and LSTM do. This shows the NPE is invariant to position and scene configuration, while NP and LSTM memorize the training wall configuration.