Detecting and Accommodating Novel Types and Concepts in an Embodied Simulation Environment

Sadaf Ghaffari and Nikhil Krishnaswamy

ACS 2022, November 19, 2022, Arlington, VA

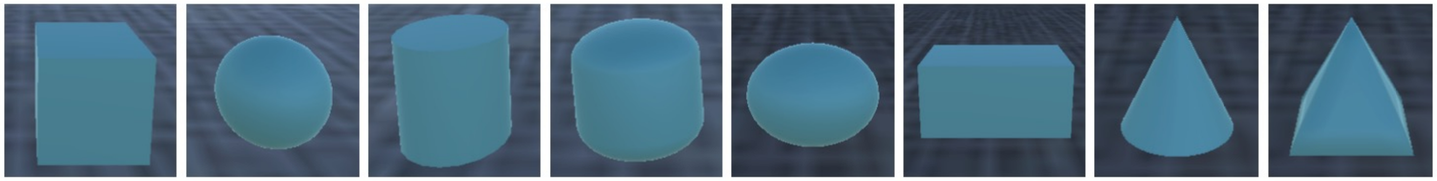
Colorado State University

Outline

- Introduction
- Environment and Data
- Object Similarity Analysis
- Transfer Learning to Accommodate New Classes
- Inferring Abstract Concepts
- Detecting Novel Classes
- Conclusion and Future Work

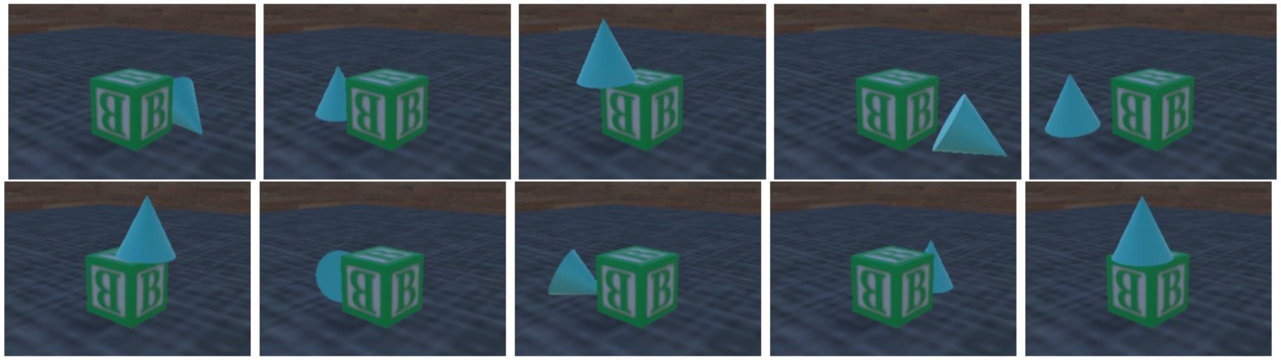
Introduction

- Humans efficiently seek out informative experience, learning from few samples through previous examples (Clark, 2006)
- Artificial neural networks require large numbers of samples to train (5-8 layers of artificial neurons ~ 1 cortical neuron) (Beniaguev et al., 2021)
- **They do not easily expand to accommodate new concepts given a few unseen samples**
- We investigate the ability of machine learning systems to detect and acquire new concepts through interaction
- These “metacognitive” processes require the system to be aware of what it does and doesn’t know
- For tractability, we focus on a domain of object interaction, inspired by geometric children’s toys



Environment and Data

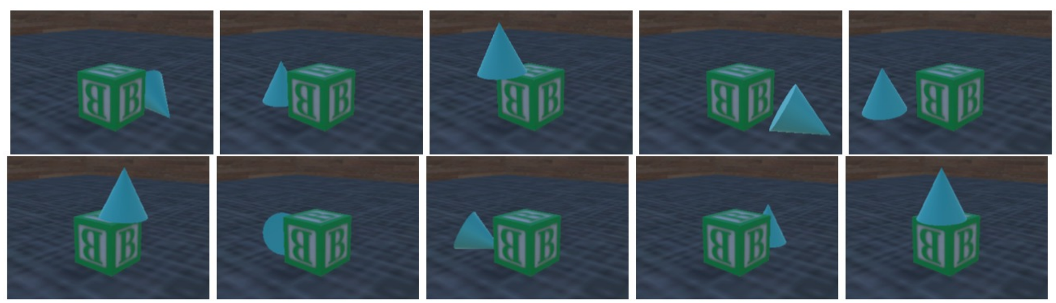
- We create environments with the VoxWorld platform for interactive agents (Krishnaswamy et al., 2022)
- Agent is presented with *cube* and one instance of another object (*theme object*)
- Pairs of objects show minimal pair distinctions (flat vs. round sides, length along an axis, etc.)
- Agent samples from environment by stochastically placing theme object on top of destination cube
- If resulting configuration is stable, theme object will stay still. If not, it will fall off



Environment and Data

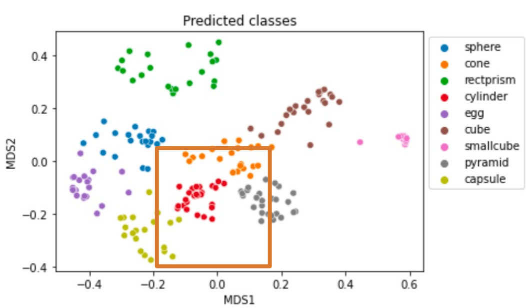
- To simulate realistic environment, we perturb object placement with a “jitter” derived from object semantics in VoxML (Pustejovsky and Krishnaswamy, 2016)
- Distinctions in object behavior correspond to *habitats* (Pustejovsky, 2013) and *affordances* (Gibson, 1977) pertaining to object’s “stackability”
- Gather **10,000** samples of each object instance
- Record geometric features of object interaction and configuration
- Try to predict object type from its behavior under interaction

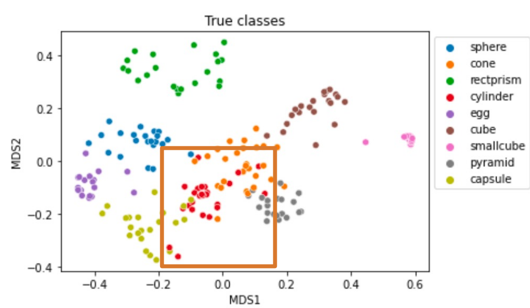
$$\begin{bmatrix} \textbf{cylinder} & \\ \text{TYPE} = & \begin{bmatrix} \text{HEAD} = \textbf{cylindroid} \\ \text{COMPONENTS} = \textbf{nil} \\ \text{ROTATSYM} = \{Y\} \\ \text{REFLSYM} = \{XY, YZ\} \end{bmatrix} \end{bmatrix}$$



Object Similarity Analysis

- **Try to predict object type from its behavior under interaction**
- 4-layer (200, 100, 50, 25) feed-forward neural network, Leaky ReLU activation function, weight decay 0.01, Adam optimization, trained for 200 epochs
- Perform Multi-Dimensional Scaling (MDS) on final hidden layer activation to capture object similarity

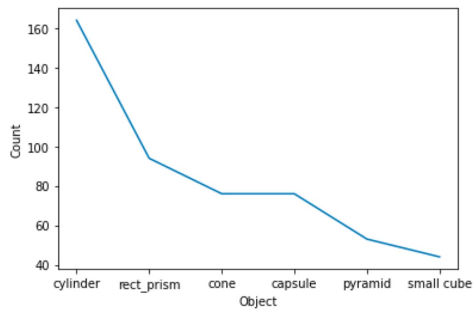




True label \ Predicted label	cube	sphere	cylinder	capsule	smallcube	egg	rectprism	cone	pyramid
cube	21	0	0	0	0	0	0	0	0
sphere	0	21	0	0	0	0	0	1	0
cylinder	0	0	19	2	0	0	0	4	1
capsule	0	0	2	18	0	2	0	0	0
smallcube	0	0	0	0	20	0	0	0	0
egg	0	0	0	0	0	19	0	0	0
rectprism	0	0	0	0	0	0	21	0	0
cone	1	0	4	0	0	0	0	18	6
pyramid	0	0	0	0	0	0	0	0	20

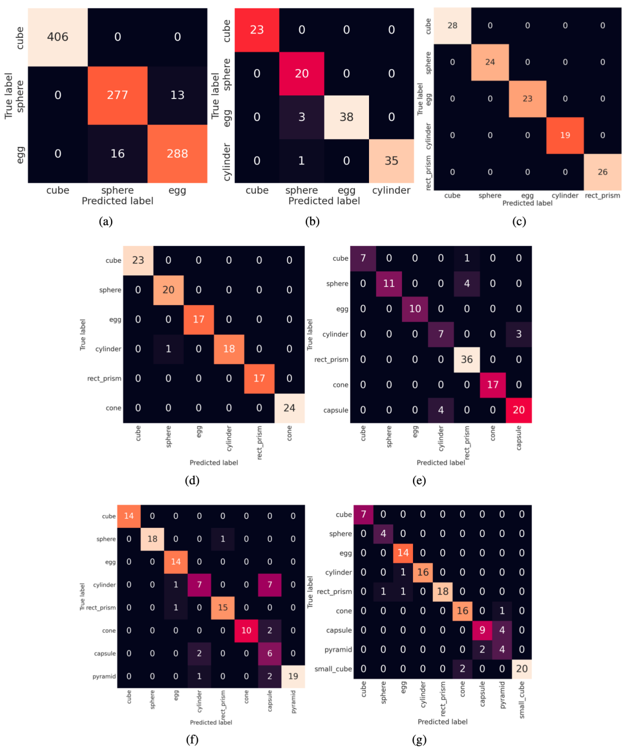
Transfer Learning to Accommodate New Classes

- **What features are most important? What concepts has the network modeled to make type distinctions?**
- Begin by training a deep feedforward architecture on *cube*, *sphere*, and *egg* only, using 5000 samples
  - These objects capture distinguishing abstract properties: flatness, roundness, and axis of rotational symmetry
- Objects added one at a time to object vocabulary
- First two hidden layers of source model are frozen, a new hidden layer is added
- Source model trained on *k-1* objects is fine tuned to target model for *k* objects

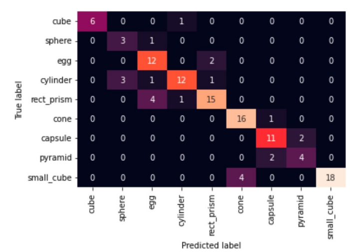


Fine tuning samples per object

Transfer Learning to Accommodate New Classes

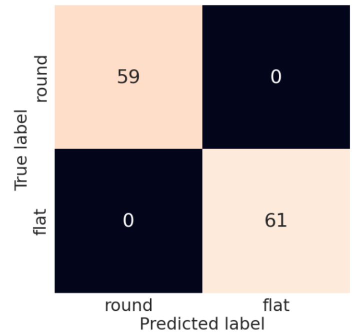


- Confusion matrices of transfer-learned model: (a) base, (b) +cylinder, (c) +rect. prism, (d) +cone, (e) +capsule, (f) +pyramid, (g) +small cube
- Able to maintain high classification accuracy by incrementally introducing one novel object and fine tuning source model
- As new objects are added, the number of samples *per object* needed for fine tuning goes down
- Dynamically growing model accuracy: **90%**
- Equal-sized static model accuracy: **80.83%**



Inferring Abstract Concepts

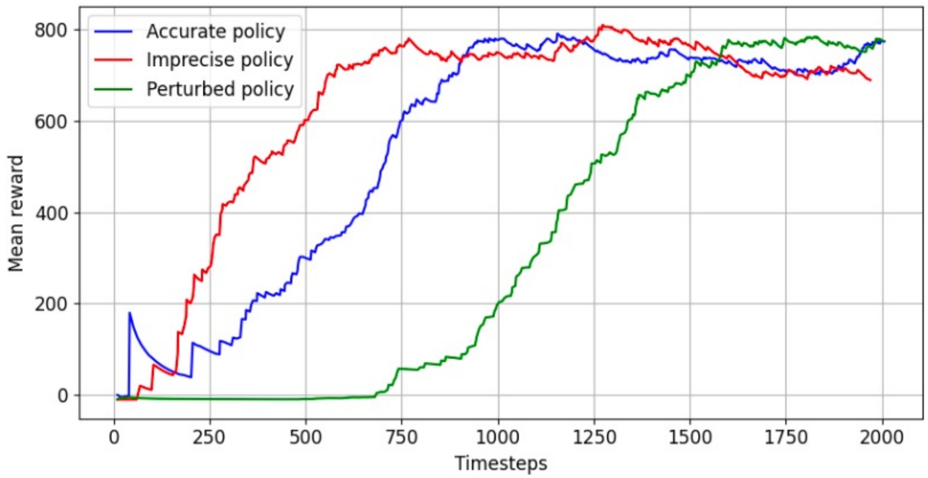
- Objects are not just instances of multiple classes
- Properties and contrasts also inhere across multiple object classes
- We have both round objects and flat object, and objects with both properties
- Data contains rotation of objects after action, which correlates to round or flat edge of objects with both
- Split cone and cylinder stacking data according to “resting on round edge” versus “resting on flat edge”
- Apply same fine-tuning procedure to previous 10-layer model to test if model can infer these abstract contrasts independent of type



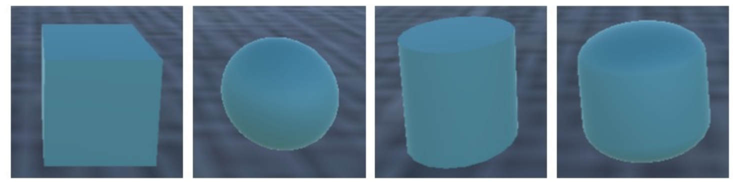
100% accuracy!

Detecting Novel Concepts

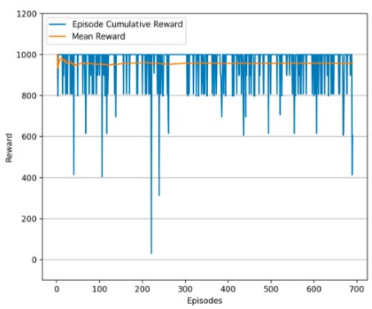
- If an agent has a fixed concept inventory, how can it detect when a novel type of object is introduced?
- Train a Twin Delayed DDPG (TD3) policy to stack blocks



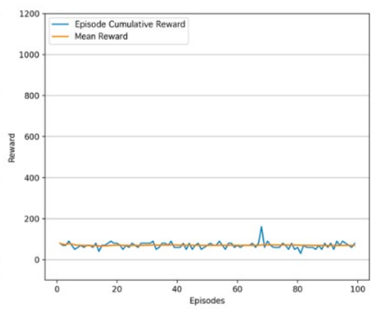
Detecting Novel Concepts



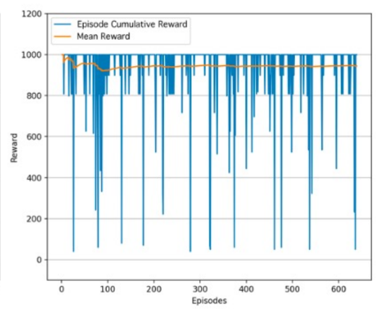
- Then, use that policy to attempt to stack a variety of objects
- Agent attempts to stack all objects *as if they were cubes*
- Store information about each attempt, including rewards



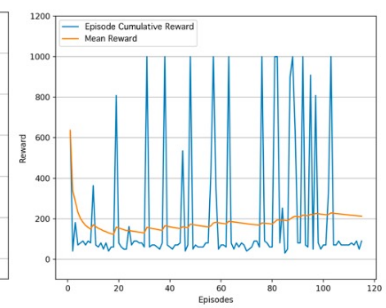
Cube



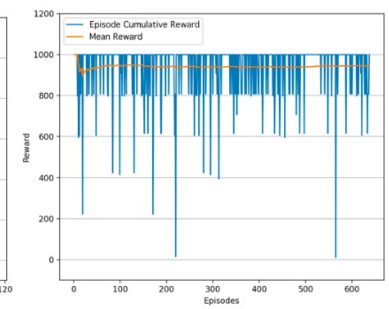
Sphere



Cylinder



Capsule

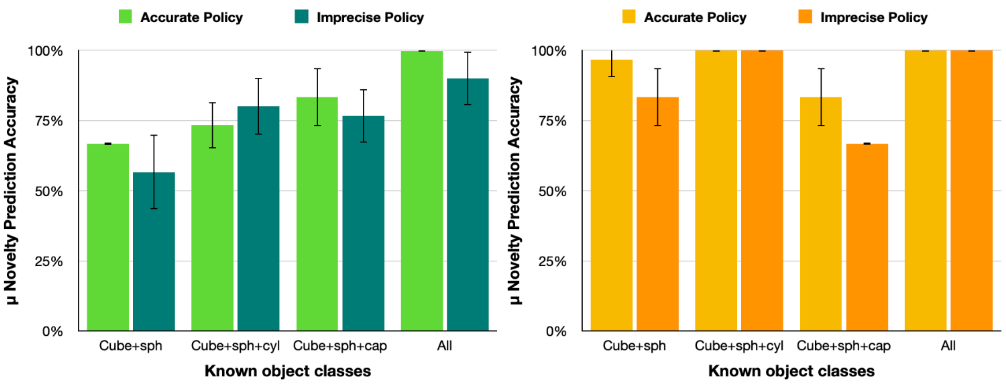


Small Cube

Accurate policy reward plots

Detecting Novel Concepts

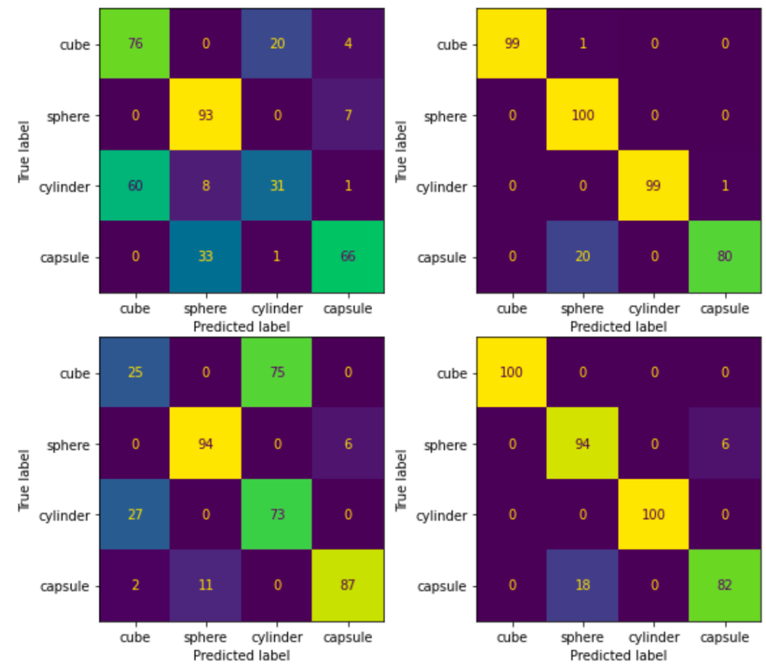
- Data is now time-sensitive; train 1D CNN classifier on subset of objects (e.g. cubes and spheres **only**)
- Retrieve embeddings for objects and compare similarities of known objects to new samples
    - Now an outlier detection problem
- If a set of vectors fall substantially outside the subspace defined by samples of known object, these vectors likely represent a new type of object



Accuracy in detecting new types of objects vs. object known to classifier (L: without jitter, R: with jitter)

Detecting Novel Concepts

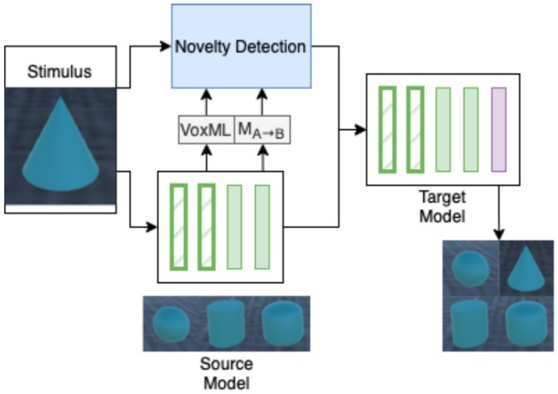
- VoxML jitter information results in impressive performance boost!
- Can correctly identify capsules and cylinders as novel, while small cube is not a novel type
- Without implicit encoding of habitats, cubes confused with cylinders, spheres confused with capsules



Aggregated CNN outputs over dev-test set

Conclusion and Future Work

- A model must be able to detect when it is inadequate to the environment
- No individual component (neural network, environment model, statistical metrics) bears sole responsibility for this capability
  - Hybrid approach or combination
- Key concepts of flatness or roundness can be exposed through stacking task
- Other concepts may require other tasks to expose
- Future work: combining two suites of experiments
  - Detecting that an object type is novel, and automatically expanding or fine-tuning model to accommodate it
  - Representations from different classifiers need to be aligned for direct comparison
  - Outputs can flow backward into RL task, where policy failure is detected and adapted for



Proposed integration architecture

Thank you!

{sadafgh,nkrishna}@colostate.edu

Colorado State University

References

- Beniaguev, D., Segev, I., & London, M. (2021). Single cortical neurons as deep artificial neural networks. *Neuron*, *109*, 2727–2739.
- Clark, A. (2006). Language, embodiment, and the cognitive niche. *Trends in cognitive sciences*, *10*, 370–374.
- Gibson, J. J. (1977). The theory of affordances. *Hilldale, USA*, *1*, 67–82.
- Krishnaswamy, N., Pickard, W., Cates, B., Blanchard, N., & Pustejovsky, J. (2022). The VoxWorld Platform for Multimodal Embodied Agents. *Proceedings of the Language Resources and Evalua- tion Conference* (pp. 1529–1541). Marseille, France: European Language Resources Association.
- Pustejovsky, J. (2013). Dynamic event structure and habitat theory. Proceedings of the 6th Interna- tional Conference on Generative Approaches to the Lexicon (GL2013) (pp. 1–10).
- Pustejovsky, J., & Krishnaswamy, N. (2016). VoxML: A Visualization Modeling Language. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16) (pp. 4606–4613).