Spatial Relation Categorization in Infants and Deep Neural Networks

Guy Davidson[1] and **A. Emin Orhan**[1] and **Brenden M. Lake**[1,2]
[1]Center for Data Science, New York University; [2]Department of Psychology, New York University

Abstract

Spatial relations, such as above, below, between, and containment, are important mediators in children's understanding of the world (Piaget, 1954). The development of these relational categories in infancy has been extensively studied (Quinn, 2003) yet little is known about their computational underpinnings. Using developmental tests, we examine the extent to which deep neural networks, pretrained on a standard vision benchmark or egocentric video captured from one baby's perspective, form categorical representations for visual stimuli depicting relations. Notably, the networks did not receive any explicit training on relations. We then analyze whether these networks recover similar patterns to ones identified in the development, such as reproducing the relative difficulty of categorizing different spatial relations and different stimulus abstractions. We find that our models tend to recover many of the patterns observed with the simpler relations of "above versus below" or "between versus outside", but struggle to match developmental findings related to the "containment" relation. We identify factors in the choice of model architecture, pretraining data, and experimental design that contribute to the extent our models match developmental patterns, and highlight experimental predictions made by our models. Our results open the door to modeling infants' earliest categorization abilities with modern machine learning tools and demonstrate the utility and productivity of this approach.

Introduction

Our understanding of the visual world around us is mediated by spatial relations, as they help distinguish individual objects and combine them in order to understand visual scenes (Piaget, 1954; Johnson, 2010). A breadth of experimental work explored how infants form categories and make categorical judgments (Bomba and Siqueland, 1983; Younger and Cohen, 1985; Eimas and Quinn, 1994) and specifically how infants form category representations for spatial relations (Quinn, 1994; Quinn et al., 1999; Casasola and Cohen, 2002; Casasola et al., 2003). Despite the importance of relations, little computational work has examined how infants could learn to categorize spatial relations, and why some categories are acquired before others over the course of development.

Our goal in this article is not to build a bespoke model of spatial relation categorization, for example by fitting models to developmental data, or by training models to categorize between different relations. Instead, we identify several key findings in the development of relation learning, translate their experimental paradigms to tasks suited for modern deep neural networks, and investigate whether *absent any explicit relational training*, models can categorize between spatial relations such as "above versus below" or "between versus outside." Figure 1 summarizes our approach. We find that our models are capable of making such categorizations, albeit with substantial variation by the relation examined, the data on which models were trained, and other experimental factors. Given this success, we then evaluate whether the performance of models tracks with the developmental findings that motivated this work—that is, to what extent do relations infants acquire later in development also challenge models more. Our hope is that exploring correspondences between development and model performance could highlight potential computational mechanisms underlying the developmental findings while simultaneously offering insight into how modern neural networks can be utilized and further developed as models of developmental cognition.

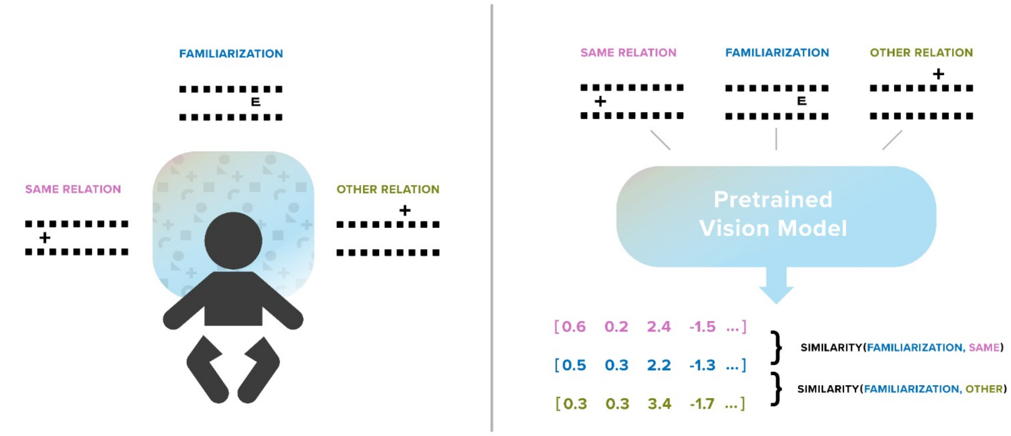

Figure 1: Spatial Relation Categorization in Infants and Deep Neural Networks. Left: after being familiarized with stimuli depicting a particular relation ("familiarization", infants find novel stimuli depicting the same relation ("same relation") less surprising than stimuli depicting a different relation ("other relation") as measured by looking times (Quinn, 2003). Right: to evaluate neural networks using a similar paradigm, we present three stimuli to a model, extract a vector embedding for each stimulus, and examine whether the "familiarization" stimulus embedding is more similar to the "same relation" stimulus embedding or to the "other relation" stimulus embedding.

We build on an important tradition of connectionist models at the intersection of cognitive science and machine learning (Donahoe and Dorsel, 1997; Rogers and Mcclelland, 2014). In particular, prior computational work used connectionist networks to model aspects of infant categorization (Mareschal et al., 2000; French et al., 2004) and spatial language (Regier, 1995). A different approach described by Ullman et al. (2019) proposed a non-neural computer vision model capable of learning to identify relations from videos in a self-supervised fashion and demonstrated it recovers various developmental patterns. As outlined above, we pursue a

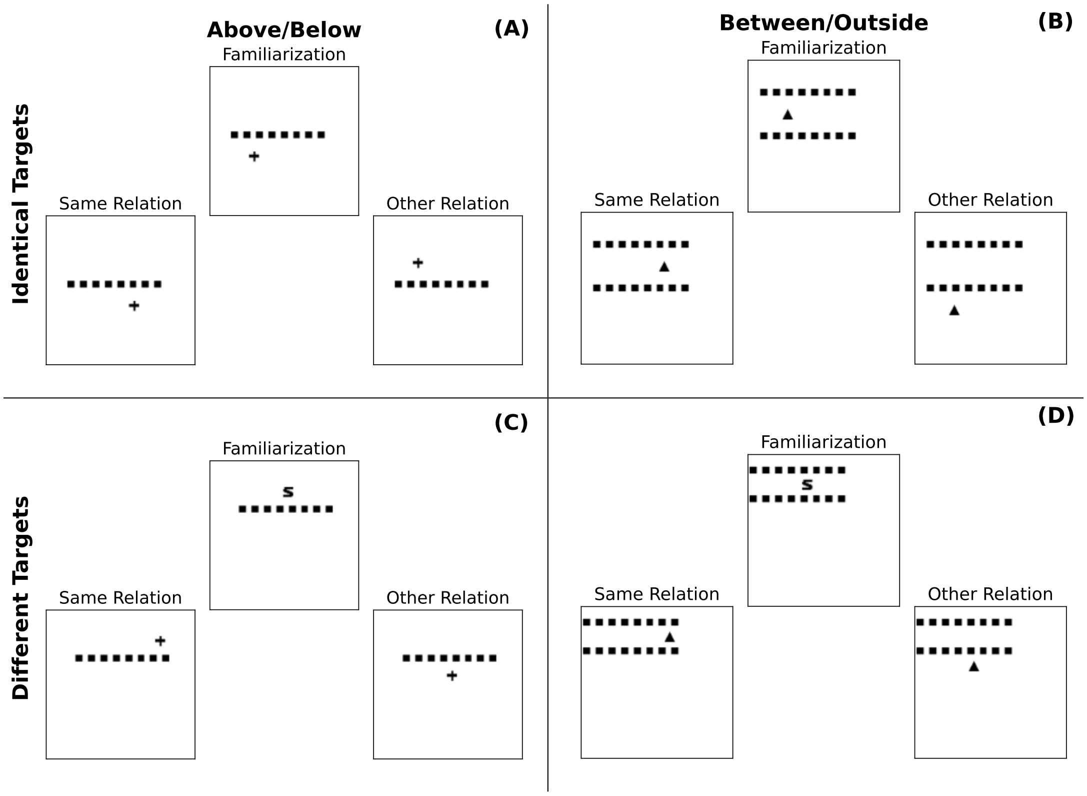

Figure 2: Example Stimuli. In each triplet, the central stimulus is the familiarization example, the left one is the same relation test, and the right one is the other relation test. (A): *above/below* with identical target objects. (B): *between* with identical target objects. (C): *above/below* with identical target objects. (D): *between* different target objects.

different approach: we identify several phenomena in the development of spatial relations and study the extent to which powerful, general-purpose computer vision neural networks replicate these phenomena without being trained to do so. We are motivated by the vast recent progress in using neural networks for computer vision. Starting with Krizhevsky et al. (2012), deep neural networks have risen to prominence as highly capable models for most computer vision tasks. We focus our attention on convolutional neural networks, a class of computer vision architectures that have proven to be useful models in the cognitive sciences. For instance, Lindsay (2021) reviews their use as models of the visual system, and Battleday et al. (2021) study the extension of these models from the visual system toward higher-level cognitive capacities such as judgments of similarity and categorization. Work on relation learning using deep neural networks tends to focus on bespoke architectures, such as Relation Networks (Santoro et al., 2017) or PrediNet (Shanahan et al., 2019), and see Battaglia et al. (2018) for a review. Other recent work focuses on graph-based networks (Baldassarre et al., 2020) or on learning to generate images with particular relations (Liu et al., 2021). In comparison, our contribution is to evaluate the latest generation of neural network architectures, pretrained on two sources of realistic image data, on their representation of simple spatial relations without explicit training or architectural modifications.

We examine the extent to which models replicate several findings on the development of infant relation categorization, focusing on the relations “above versus below” and “between versus outside” (see Quinn (2003) for a review) and the “containment” relation (see review by Casasola, 2008). In a series of studies (Quinn, 1994; Quinn et al., 1996, 1999; Quinn, 2002; Quinn et al., 2003; Quinn, 2004), Quinn and colleagues use similar methodologies to establish several patterns regarding the development of relational categories. Using stimuli similar to the one in Figure 2, babies were familiarized with several stimuli of the type appearing in the middle of each triplet. The infants were then shown the two test stimuli, one depicting the same relation and one depicting the opposite relation. To establish the existence of a category representation, the studies measured the amount of time spent looking at the stimulus depicting the opposite relation, divided by the total looking time at both test stimuli. The higher this percentage is, the stronger a novelty preference (Fantz, 1964) the infant displays, and the more evidence it provides for a categorical representation of the familiarized relation.

Quinn (2003) surveys two primary findings. The first finding is that, by 3-4 months of age, infants can categorize “above versus below” (or “left versus right”, Quinn, 2004), although they fail to categorize “between” (Figure 2; (A) and (B)). By 6-7 months, infants can also categorize “between.” In a representative experiment, Quinn (1994) familiarized infants with several stimuli, all containing a dot either above or below a horizontal bar (Figure 2; Familiarization). After familiarization, infants were presented with a novel category preference test, finding that infants look longer at a stimulus with the dot on the other side of the bar (Figure 2; Other relation) compared to a new location on the same side (Figure 2; Same relation).

The second finding is that infants categorize spatial relations depicting specific objects before categorizing the same relations composed of varying objects. Quinn et al. (1996; 2003) replicate the previous experiments except that the target object varies between familiarization and test (Figure 2; (C) and (D)). In both cases, changing the target object requires the infants to be older to show the same novelty preference—from 3-4 months to 6-7 months for above versus below, and from 6-7 months to 9-10 months for between versus outside.

A second line of work by Casasola and Cohen (2002) and Casasola et al. (2003) studies the emergence of the containment relation. In a representative experiment, Casasola et al. (2003, Experiment 2) examine infants’ category for the *containment* relation (one object placed inside another object). The authors familiarized the infants with a video clip depicting the *containment* relation—one object being picked up and placed inside another one (whose final frame is represented in Figure 3(A/B), “Familiarization”). Casasola et al. (2003) then tested the infants using three different test probes, all filmed from a different camera angle. The first probe also depicted a *containment* relation (Figure 3(A/B), “Same Relation”). The second probe showed an object being picked up and placed *behind* another object (Figure 3(A), “Other Relation” under “Containment vs. Behind”). The third and final test probe presented a *support* relation, with the object being picked up and placed on top of another one (Figure 3(B), “Other Relation”). They find that even when controlling for the degree of object occlusion in their stimuli, infants reliably find the test probes depicting the *containment* relation as most similar to the familiarization probes, as measured by looking times. This is taken as evidence that the infants constructed a category representation of the *containment* relation.

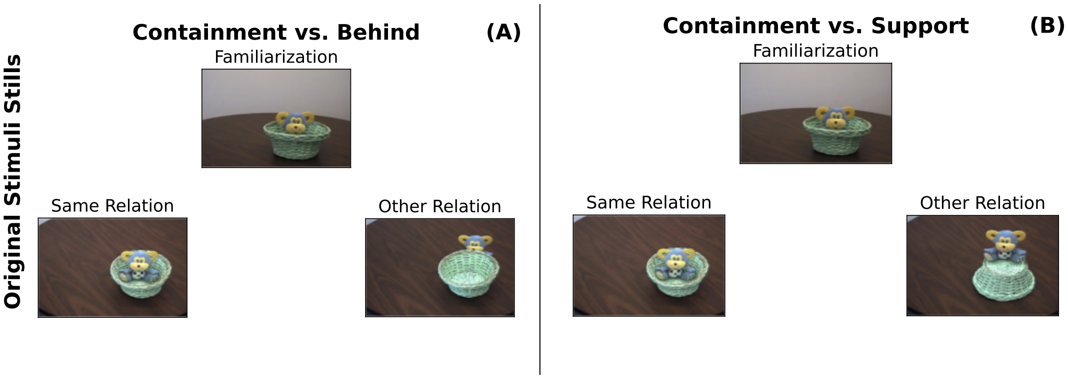

Figure 3: Example stimuli from Casasola et al. (2003). We present the stimuli in a similar triplet form to the one used in Figure 2. In each triplet, the central stimulus is the familiarization example, the left one is the same relation test, and the right one is the other relation test. We depict the final frames of the stimuli videos presented to infants in Casasola et al. (2003), reproduced from Casasola (2008, Figure 1). (A): comparing a test probe depicting the *containment* relation to a test probe depicting the *behind* relation. Rendered using the wooden basket container and Lego target object. (B): comparing a test probe depicting the *containment* relation to a test probe depicting the *support* relation.

In Experiments 1-4, we evaluate a collection of pretrained, large-scale computer vision neural network models on tasks inspired by the various developmental experiments and findings surveyed. We view the pretraining as a proxy for prior visual experience, and compare the experience gained from egocentric video capture based on one baby's experiences (SAYCam, Sullivan et al., 2020) to experience from a popular computer vision benchmark (ImageNet, Russakovsky et al., 2015), neither of which explicitly requires relational categorization:

1. In Experiment 1, we find that the models succeed in capturing developmental findings surveyed by Quinn (2003) using the *above/below* and *between* relations. We also compare the different model architectures and pretraining approaches and find that representations from the models trained on developmentally-realistic data appear to promote relational information more than the alternatives we evaluate.

2. In Experiment 2, we flip the relations on their side and evaluate the models on the relations *left/right* and *sideways between*. We find that our initial set of models fails to replicate the developmental findings of interest, and identify model training choices that explain the deviation and enable recovering the initial findings.

3. In Experiment 3, we examine the extent to which the networks' representations of these relations are sufficiently abstract to handle different types of stimuli. We do so by generating more complex three-dimensional scenes that more closely resemble real relation scenarios. We find success in replicating all findings of interest from Experiment 1, demonstrating that the relational representations are abstract enough to generalize to a substantially different class of visual stimuli.

4. In Experiment 4, we find that our models struggle to recover the relevant empirical patterns with the *containment*, *behind*, and *support* relations as described by Casasola et al. (2003). We explore these results to examine the extent to which the models we evaluate embed information about these more complex relations and discover that the information is still present and linearly decodable even when the models struggle on the task using a generic similarity metric.

We find that the pretrained visual representations are sufficient for the categorical perception of simple relations (*above/below* and *between*). Moreover, these representations are sufficiently abstract for handling either 2D and 3D stimuli. In the case of the more complex relations of *containment*, *behind*, and *support*, the embeddings contained sufficient information to linearly decode the relation with very high accuracies, even when representational similarity was not driven by the spatial relation. We conclude by attempting to identify useful methodological aspects to support future work and highlight current gaps and open questions.

Experiment 1: Classifying *above/below* and *between/outside* from 2D stimuli

We begin by studying the extent to which large-scale, pretrained computer vision models recover the two developmental findings reviewed by Quinn (2003). We use pretrained models to study whether an infant's ability to categorize different relations in a lab study could emerge from high-level visual representations developed independently, without training models explicitly for relation categorization. As a first step, we evaluate to what extent these spatial relations are perceived categorically (see Goldstone and Hendrickson (2010)); we do so by examining the similarity of stimuli encoding the same relation compared to stimuli encoding different relations. As a second step, assuming there is a categorical response, we examine whether capacities demonstrated by infants earlier in development are also easier for our models, which is an assumption we make in order to compare the developmental phenomena to model performance. For instance, given that Quinn (1994) demonstrated that infants acquire category representations for "above or below" earlier in development than for "between or outside," then we would examine whether the model is more accurate in the *above/below* condition compared to the *between/outside* condition.

We evaluate our models using representational similarity, without any training or explicit prediction of relations. In each triplet (Figure 2), the central image corresponds to a familiarization stimulus, and the other two images represent test stimuli, one depicting the same spatial relation and one its opposite (see, e.g., Colunga and Smith, 2005 and Kim et al., 2021 for other triplet-based similarity response approaches). We pass each image independently through a model, extracting an embedding (latent representation) of each one, and test whether stimuli representing the same relation are represented more similarly, as an emergent consequence of training a model on broad visual experience (in one case, of the sort a single baby would actually experience). We implement this similarity test using the cosine similarity between the embeddings of the familiarization stimulus and the two test stimuli. Given two embeddings vectors $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{R}^D$ in a D-dimensional embedding space, their cosine similarity is defined as $S_{cos}(\mathbf{e}_1, \mathbf{e}_2) = \frac{\mathbf{e}_1 \cdot \mathbf{e}_2}{||\mathbf{e}_1|| ||\mathbf{e}_2||}$, that is, the angle between the vectors. We view this metric as appropriate as it represents an unstructured similarity comparison between the embeddings, potentially analogous to the implicit judgment infants make when evaluating how novel a test stimulus is compared to a previous habituation stimulus. We consider a triplet to be accurately classified when the model embeds the two congruent images (depicting the same relation) more similarly than the two incongruent images (depicting different relations), where the incongruent pair acts as a perceptual lure that matches in another dimension.

Experiment 1a: Initial findings

In our first experiment, we use pretrained models to examine (a) whether the representations produced by these models capture the spatial relations, and (b) to what extent they recover the developmental findings of interest. The experiment varies several factors: computer vision architecture, pretraining dataset, and stimulus rendering details.

Methods

Model Architectures. We evaluate two computer vision architectures, to validate any findings we discover are not unique to a specific model and examine whether more performant architectures also fare better on our developmental comparison:

- MobileNetV2: this model aims to offer competitive performance with fairly limited computational resources, offering an efficient trade-off between compute resources required and performance attained (Sandler et al., 2018).

- ResNeXt: this model is considered a highly capable computer vision backbone for various tasks (Xie et al., 2017). We use the ResNeXt-50 variety of this architecture.

We visualize the ResNeXt architecture and where we extract our vector embeddings from in Figure 4, and see subsection A.1 for additional details.

Pretraining. We test the embeddings created by randomly initialized models and compare them to models trained on two other datasets. One dataset and training approach reflects common practice in computer vision, while the other offers a closer comparison to a developing child:

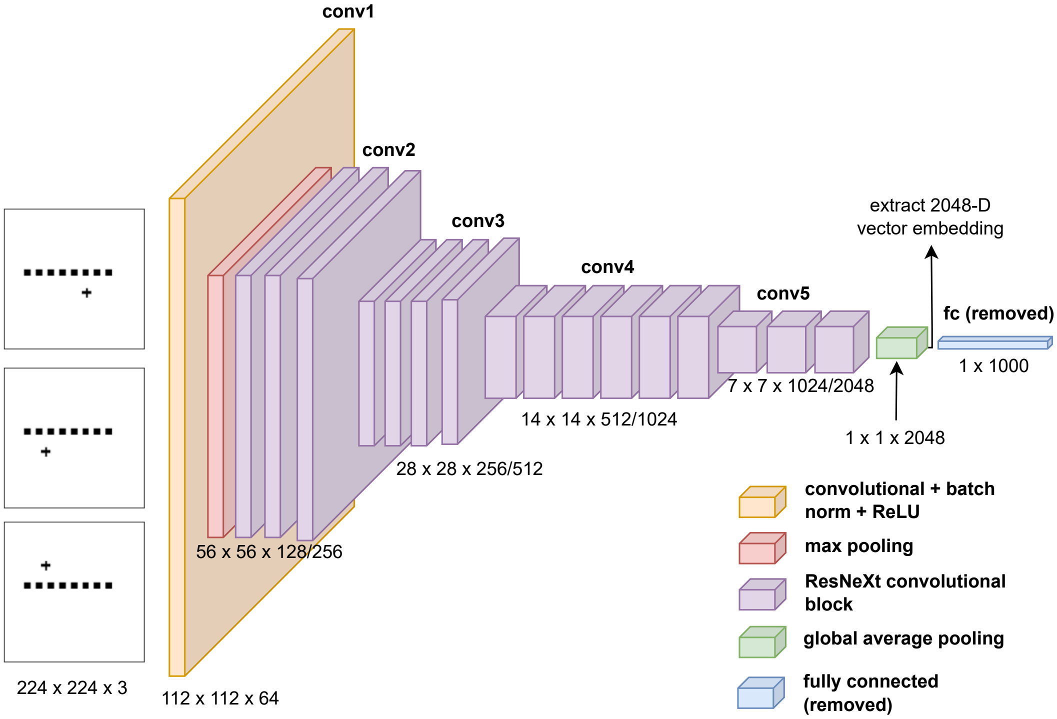

Figure 4: ResNeXt model diagram. We pass 224 x 224 images into the model, which begins with a convolutional block (orange) and a pooling layer (red), and then proceeds to ResNeXt convolutional blocks (purple) operating on increasingly smaller representations of the input images (see subsection A.1 for further details). We extract our vector embedding, which with this architecture has 2048 entries, after the global average pooling layer (in green). In a standard classification setting this embedding would be classified using a fully connected layer (blue), which we remove from the models we evaluate.

- Randomly initialized models: we examine untrained models whose weights have been randomly initialized to observe whether or not the inductive biases conveyed by the architecture alone are sufficient to embed objects in the same relation more similarly.

- ImageNet: a landmark computer vision dataset, offering 1.2M images in 1000 object classes (Russakovsky et al., 2015). ImageNet does not correspond to an infant's natural experience but it is commonly used for general computer vision pretraining, offering a useful comparison. The ImageNet models were pretrained using the standard classification task as described in the torchvision documentation[1].

- SAYCam: this dataset consists of longitudinal headcam videos from a small number of babies (Sullivan et al., 2020). This offers the opportunity to train vision models on a subset of the experience a child receives in development, albeit ranging to older ages than the infants studied in the experiments modeled. We utilize a pretrained network from Orhan et al. (2020) trained with temporal classification, a self-supervised learning algorithm inspired by psychologically plausible mechanisms. Temporal classification only makes use of the temporal ordering of data to supervise the learning process. We use models trained on a single child's footage (child S), approximately two hours per week while the child was between 6-30 months old, a total of 221 hours.

Stimulus Generation. We synthesize custom stimuli to probe the model in this task (Figure 2). We sample location(s) for the reference object(s) and then place the target objects relative to them. Similarly to Quinn (1994; 1996; 1999), we place the target object in one relation relative to the reference object in the familiarization example, and then place it in a different location in the same relation (first test probe) or in the other relation (second test probe). The target objects in the test probes are both equidistant from the target object

[1] `https://pytorch.org/vision/stable/models.html`

in the familiarization probe, controlling for any effect of distance on the representational similarity. We examine triplets where the target object matches between the familiarization and probe stimuli ("identical targets"; Quinn (1994); Figure 2; (A)) and (B) and triplets where the probe stimuli use a different target object ("different targets"; Quinn et al. (1996); Figure 2; (C) and (D)). We explore a few ways to render the reference and target objects, detailed in subsection A.1. We render these stimuli to 224x224 pixel images.

Methods Summary. We evaluate models from two architectures, either randomly initialized or pretrained on one of two visual datasets, on two relations (*above/below* and *between*), using stimuli rendered with three different approaches. For each relation and rendering method, we sample 1024 triplets (identical for all models) and report the average accuracy for each model and pretraining setting—how often are the embeddings for the congruent pair of stimuli more similar (using cosine similarity) than the embeddings for the incongruent pair. For every set of results, we compute a mean accuracy and standard error of the mean (SEM) over the 1024 triplets, and below we report different aggregations of these mean accuracy measurements across experimental conditions of interest. We omit drawing error bars as the averaged SEMs all fall below 2% accuracy.

Results

A summary of the results is shown in Figure 5 and Table 1. Without pretraining, the models performed near chance, with levels of accuracy ranging from 0.47 to 0.58. This suggests that inductive biases conferred by the architecture alone are insufficient for representing relations (see the results marked by an 'X' in Figure 5). Therefore, we focus our analysis on the trained models. We aggregate across the different stimulus generation approaches (subsection A.1) as qualitative results are consistent between them (Figure B.3). Across both pretraining datasets (Figure 5, circles for SAYCam and squares for ImageNet) and model architectures (green for MobileNetV2, orange for ResNeXt), models tended to represent the same relation test probes more similarly to the habituation stimuli than the different relation probes. This is seen in the consistent above-chance levels of accuracy, which vary by model and experimental condition, but range between roughly 60% and almost 100%. Given that we find that our models appear to represent these stimuli in a manner reflecting relational categories, we can examine to what extent the models reflect the findings reviewed by Quinn (2003).

Using both architectures and training datasets, we recover both developmental phenomena of interest. Analogously to infants acquiring the *above/below* relation earlier in development, we found consistently higher levels of accuracy for each model and dataset in the *above/below* relation compared to the *between* relation (compare left-side results to right-side results in Figure 5(A), or examine the 'By relation' column in Table 1). We also observed slightly higher levels of accuracy in the conditions using the same target objects across all three stimuli than the conditions using different targets in the test stimuli, corresponding to infants acquiring category representations with identical target objects before acquiring them with varying targets (compare left-side results to right-side results in Figure 5(B), or examine the 'By targets' column in Table 1). We ran several additional controls to more closely match the above/below and between/outside conditions (e.g., such that each condition uses two horizontal bars), and to vary the number of habituation stimuli. The results were remarkably consistent across these factors (see subsection B.3 for details).

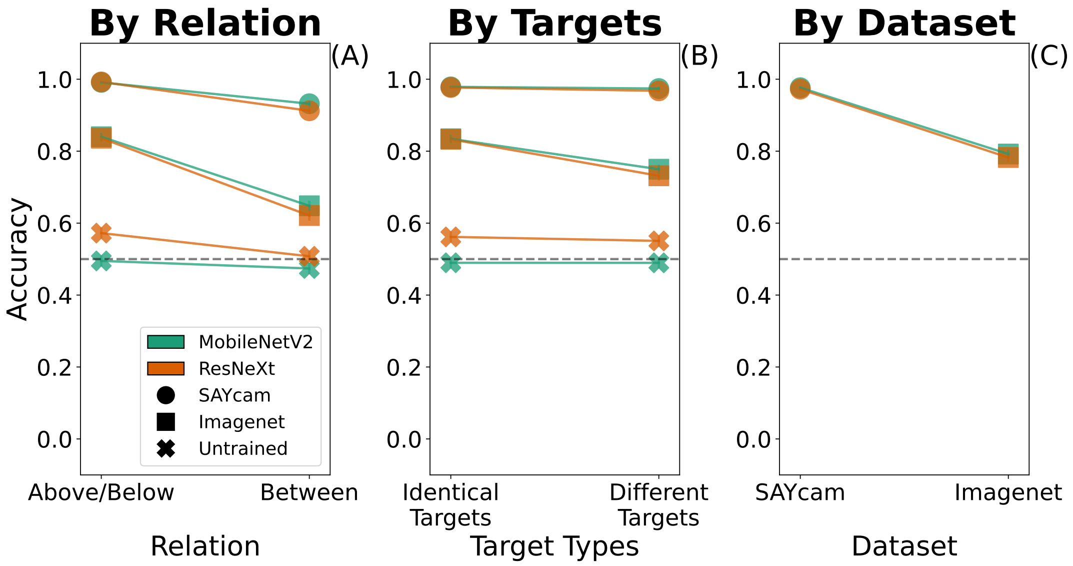

Figure 5: Our models represent relational categories and recover developmental phenomena. All three figures reflect the same set of experimental results, aggregated by different conditions of interest: (A): in a comparison between relations—*above/below* (left) versus *between* (right)—accuracy is higher in the *above/below* relation (B): in a comparison between target types—identical target objects (left) versus different target objects (right)—accuracy is higher when using identical target objects. (C): in a comparison between pre-training datasets—self-supervised SAYCam (left) versus supervised ImageNet (right)—accuracy is higher when using the SAYCam dataset. The color reflects model architecture, and the marker the training method. The dashed line indicates chance accuracy (50%).

	Relation		*Above/Below*		*Between/Outside*		Mean Change in Accuracy	
Experiment	Training	Model	Identical Targets	Different	Identical Targets	Different	By Relation	By Targets
1a	Untrained	MobileNetV2	0.50 ± 0.02	0.49 ± 0.02	0.47 ± 0.02	0.47 ± 0.02	0.02	0
		ResNeXt	0.58 ± 0.02	0.56 ± 0.02	0.51 ± 0.02	0.51 ± 0.02	0.06	0.01
1a	ImageNet	MobileNetV2	0.88 ± 0.01	0.80 ± 0.01	0.68 ± 0.01	0.61 ± 0.01	0.19	0.08
		ResNeXt	0.89 ± 0.01	0.78 ± 0.01	0.66 ± 0.01	0.58 ± 0.02	0.22	0.09
1a	SAYCam(S)	MobileNetV2	0.99 ± 0.00	0.99 ± 0.00	0.93 ± 0.01	0.93 ± 0.01	0.06	0
		ResNeXt	1.00 ± 0.00	0.99 ± 0.00	0.92 ± 0.01	0.91 ± 0.01	0.08	0.01
1b	DINO-ImageNet	ResNeXt	0.93 ± 0.01	0.82 ± 0.01	0.64 ± 0.01	0.57 ± 0.02	0.27	0.09
		ViT-B/14	0.92 ± 0.01	0.77 ± 0.01	0.76 ± 0.01	0.59 ± 0.02	0.17	0.16
1b	DINO-SAYCam(S)	ResNeXt	0.99 ± 0.00	0.98 ± 0.00	0.78 ± 0.01	0.76 ± 0.01	0.22	0.02
		ViT-B/14	0.97 ± 0.00	0.89 ± 0.01	0.91 ± 0.01	0.73 ± 0.01	0.11	0.13
	Mean difference						**0.14**	**0.06**

Table 1: Summary of Experiment 1a/b Results. We report the mean levels of accuracy for each combination of training data, model architecture, relation, and target type variation. The right-most columns report the mean difference across the two manipulations corresponding to the developmental phenomena of interest: the "By relation" column offers the mean drop in accuracy from the *above/below* condition to the *between* one, and the "By targets" column offers the mean drop in accuracy from the "identical targets" condition to the "different targets" one. In both cases, the mean change found is congruent with the developmental phenomena examined—we observe higher accuracies in the conditions infants acquire earlier in development. Margins represent the standard errors of the mean.

To better understand how the embeddings learned by these models solve this task, we synthesize a set of test stimuli by tiling a target object across a canvas with respect to a fixed reference object (Figure 6, right). We produce such sets of stimuli with the Quinn-like stimulus generation approach and embed these with the models we evaluated in experiment 1a. To visualize, we perform unsupervised dimensionality reduction using PCA (using $n = 32$ principal components), and further reduce dimensionality to 2D (X and Y coordinates) using t-SNE (van der Maaten and Hinton, 2008) with the cosine distance metric. We color each marker (representing a single stimulus, Figure 6, right) by the vertical position of the target object in the stimulus (Figure 6, left). Unsurprisingly, we find no structure in the 2-D representations of the untrained model embeddings. Models trained on ImageNet show a separation between stimuli whose target object was above the bar (shades of red) and stimuli whose target object was below the bar (shades of blue). Models trained on SAYCam show a much stronger separation between these categories. Stimuli rendered with our other two stimulus generation approaches replicate these results (Figure B.7, Figure B.8).

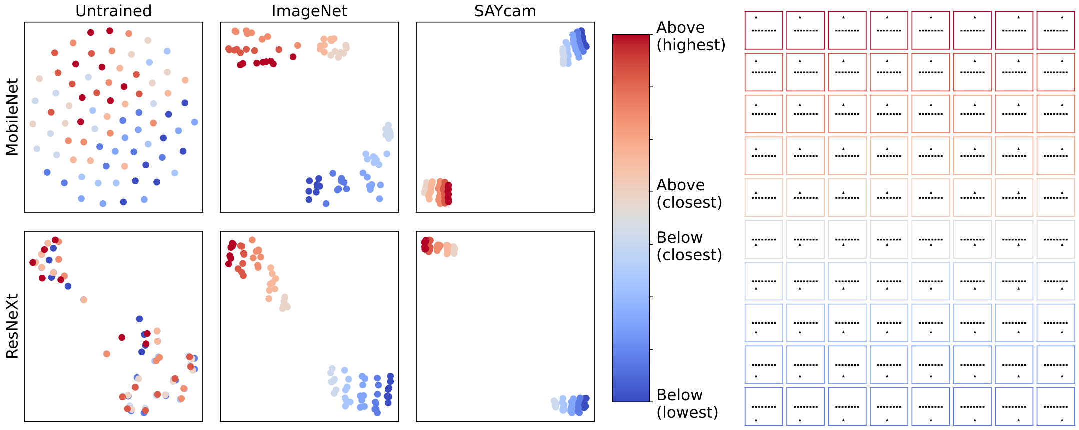

Figure 6: Categorical perception of *above/below* in our model embeddings. We synthesize a set of controlled stimuli (right) by varying the position of a target object in relation to a fixed reference object. We embed these stimuli with our models and reduce dimensionality to 2-D (see text for details). Each stimulus is colored by the vertical position of the target object (see the color bar). We find that while there is little structure in the untrained model embeddings, both the ImageNet-trained models and the SAYCam-trained ones produce embeddings preserving the relational structure. Rows: model architectures (top: MobileNetV2, bottom: ResNeXt). Columns: model training methods (left: untrained, middle: supervised training on ImageNet, right: self-supervised training on SAYCam).

We repeat this embedding visualization procedure with synthesized stimuli with two reference objects that match our "between" relation stimuli (Figure 7, right). We once again observe no structure in the embeddings produced by the untrained models (Figure 7, left). The models trained on ImageNet show three separate clusters: above both reference objects (red and orange), between the two reference objects (light green and light orange), and below both reference objects (blue and green). The SAYCam-trained models show even tighter clustering, indicating stronger similarities within each group and more pronounced differences between the groups. Alternative stimuli renderings replicate these results as well (Figure B.9, Figure B.10).

We observe that our SAYCam-trained models, which acquire their perceptual features from the visual experience of young children, outperformed the ImageNet-trained models, which acquire their perceptual features from categorizing objects curated using a web search. We see this effect both quantitatively, in the higher accuracy reached by these models (Figure 5), and qualitatively, in the tightness of the embedding clusters visualized (Figure 6, Figure 7). Although these results suggest an exciting intuitive conclusion ("models trained on infants' visual experience develop stronger relational features"), our results are confounded by the fact our models were trained using different approaches. The ImageNet model was trained using supervised learning to label objects according to their category, while our SAYCam models were trained in a self-supervised fashion using a temporal classification approach that does not require object labels. We deconfound these results in the next experiment.

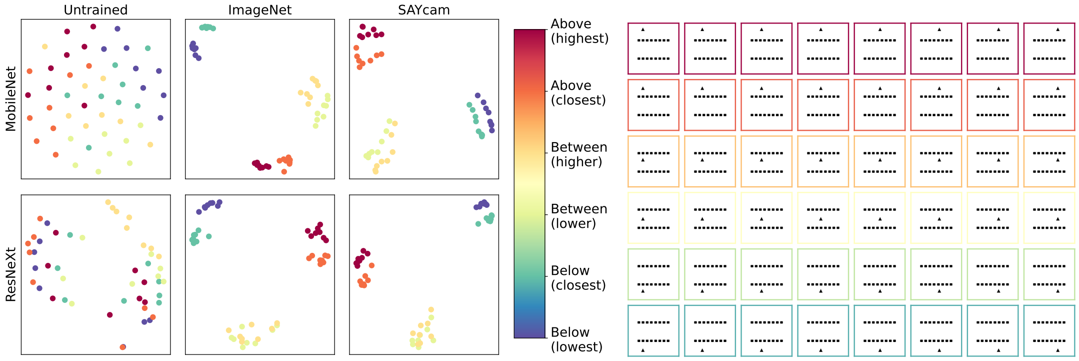

Figure 7: Categorical perception of *between/outside* in our model embeddings. We synthesize a set of controlled stimuli (right) by varying the position of a target object in relation to two fixed reference objects. We embed these stimuli with our ResNeXt models and reduce dimensionality to 2-D (see text for details). Each stimulus is colored by the vertical position of the target object (see the color bar). As in Figure 6, we find clustering preserving the relational information in the trained model embeddings. Rows: model architectures (top: MobileNet, bottom: ResNeXt). Columns: model training methods (left: untrained, middle: supervised training on ImageNet, right: self-supervised training on SAYCam).

Experiment 1b: Improved model and dataset controls

In this experiment, we introduce a third model architecture and train two architectures on the previous datasets using the same training method. Fixing the training algorithm allows us a controlled comparison of the effect of the naturalistic, infant's perspective data. We leave all other aspects of Experiment 1a unchanged.

Methods

Model Architectures. We evaluate one of the models from the previous experiment and add another prevalent computer vision architecture:

- ResNeXt: identical to the architecture we evaluated in Experiment 1a (Xie et al., 2017).
- ViT-B/14: We add the Vision Transformer model (Dosovitskiy et al., 2021) as another model architecture. This category of models applies the Transformer architecture (Vaswani et al., 2017) to images by extracting individual image patches, flattening each patch to a vector, embedding each vector independently using a small linear model, and passing a sequence of the vector embeddings representing the image into a series of Transformer blocks. We use the ViT-B/14 variation of the model, which uses the "Base" model size offered by Dosovitskiy et al. (2021) with a 14 x 14 patch size.

Beyond its overall recent success in a variety of computer vision tasks, we add this architecture as the Transformer self-attention architecture might offer a stronger inductive bias to relational representation than the convolutional neural networks we compared in Experiment 1a.

Pretraining. In this experiment, we study models trained using the DINO algorithm (Caron et al., 2021). DINO is a self-supervised learning algorithm that does not rely on labels, allowing us to use it with both of our datasets (while ImageNet contains a label for every image, SAYCam does not). DINO relies on generating multiple views of each input image through data augmentations, and learning representations that are similar between different views of the same image, but different for views of different images. We direct the reader to Caron et al. (2021) for further details.

Results

A summary of the results is presented in Figure 8 and Table 1. We continue to successfully recover the two developmental phenomena of interest. Accuracy in the *above/below* relation is consistently higher than accuracy in the *between* relation, and accuracy when using the same target objects is consistently higher than accuracy when using different target objects. We also replicate the training dataset pattern from Experiment 1a—the models trained on SAYCam reliably reach higher levels of accuracy than the models trained on ImageNet (matched-pairs T-test, $t = 12.797, P < 1e-16$). As this experiment properly controls for the training algorithm

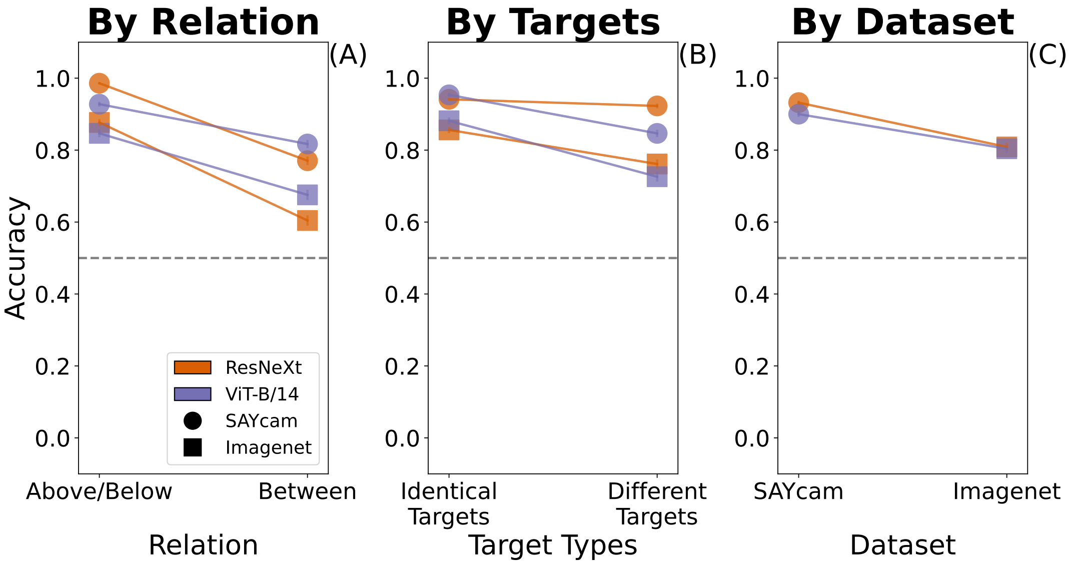

Figure 8: DINO-trained models continue to recover developmental phenomena of interest. All three figures reflect the same set of experimental results, aggregated by different conditions of interest: (A): in comparison between relations—*above/below* (left) versus *between* (right)—accuracy is higher in the *above/below* condition. (B): in comparison between target types—identical target objects (left) versus different target objects (right)—accuracy is higher in the identical target objects condition. (C): in comparison between pre-training datasets—SAYCam (left) versus ImageNet (right)—accuracy is higher for the models trained with DINO on the SAYCam dataset. Color indicates model architecture and the marker type indicates the training dataset. The dashed line indicates chance accuracy (50%).

used, we see converging evidence that training on a child's egocentric visual experience yields a representation with more pronounced relation-based similarity than training on assorted object images.

Experiment 1 Discussion

We find that large-scale, pretrained computer vision models successfully replicate a variety of developmental patterns in infant relation categorization. Across a variety of model architectures, training approaches, and control conditions, we observe that:

1. Absent any explicit relational training, embeddings extracted from our models consistently display higher similarity for stimuli representing the same spatial relation, suggesting that broad visual expertise is sufficient to induce sensitivity to some relational categories.
2. Consistent with Quinn (1994) and Quinn et al. (1999), who found that infants acquire category representations for *above/below* earlier in development than *between*, our pretrained models display higher levels of accuracy on the *above/below* relation than on the *between* relation.
3. Consistent with Quinn et al. (1996) and Quinn et al. (2003), who found that infants acquire category representations for consistent target objects earlier in development than for varying objects, our pretrained models display higher levels of accuracy when target objects remain identical ("identical targets") than when target objects vary ("different targets").

We also observe that our models trained on the developmentally-relevant visual experience of SAYCam outperform models trained on the generic object recognition data in ImageNet. We find this to be true both when models were trained using different approaches that match each dataset (Experiment 1a) and when trained using an identical approach that could be applied to both datasets (Experiment 1b). Although it's plausible that training models on naturalistic visual experience could increase their utility as cognitive models, we view our evidence as preliminary. One potential piece of supporting evidence: in concurrent work, Orhan & Lake (in prep.) find that models trained with visual data from child S in SAYCam perform at around 70% of ImageNet-trained models across a diverse range of downstream evaluations with real-world stimuli. In our evaluations, the SAYCam-trained models outperform models trained on the entirety of ImageNet, suggesting that something about the SAYCam training data facilitates embedding relations in the context of our stimuli and task. With these findings in hand, we proceed to study another set of relations examined by Quinn and colleagues.

Experiment 2: Classifying *left/right* and *between/outside (sideways)* from 2D stimuli

Quinn (2004) followed up on the "above or below" experiments of Quinn (1994), and demonstrated two distinct phenomena. The first is that if the "above or below" stimuli are rotated by 90 degrees to become a "left or right" category distinction, 3-4 month-old infants continue to demonstrate a categorical preference, preferring test stimuli with the target on a novel side of the bar. Conversely, when the reference object was rotated at an angle of 45°, 3-4 month-old infants show no preference to objects placed on a novel side of this diagonal reference object, unlike both previous examples. Figure 9 shows example stimuli with the reference objects rotated by 90 degrees, where *left/right* replaces *above/below* and the sideways between relation replaces the previous between one. Other than the angle at which the stimuli are rendered, all other experimental details remain identical to experiments 1a and 1b.

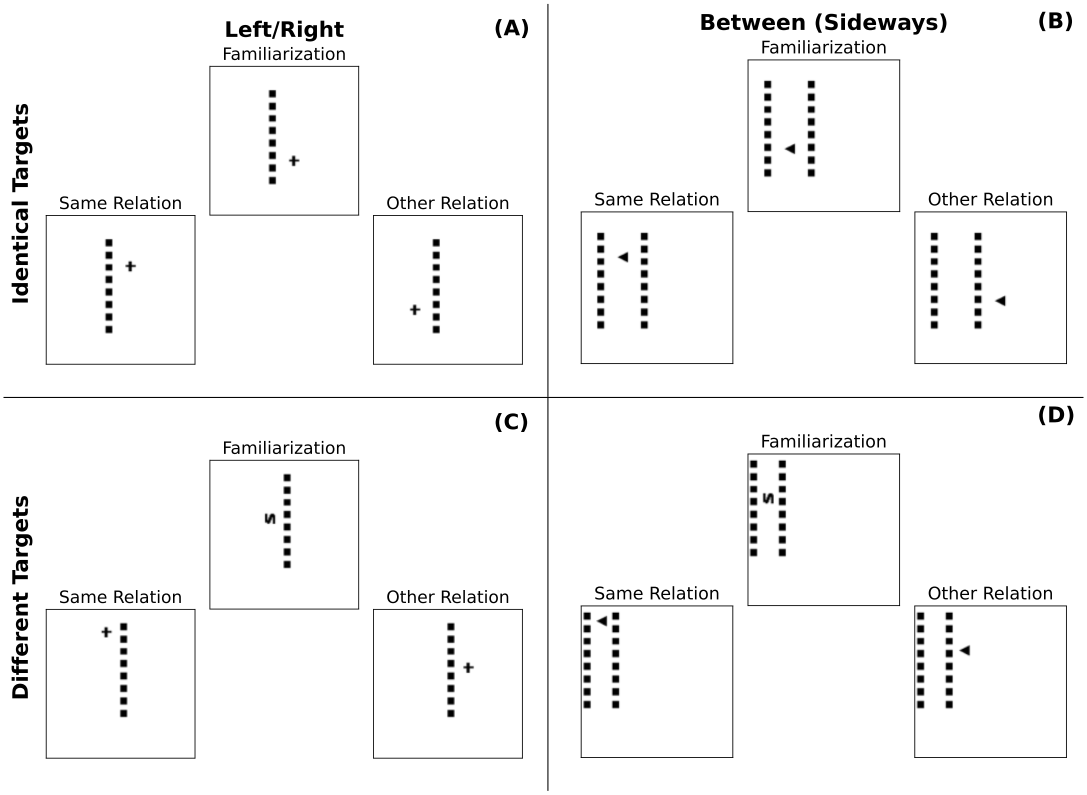

Figure 9: Example stimuli rotated 90°**.** In each triplet, the central stimulus is the familiarization example, the left one is the same relation test, and the right one is the other relation test. (A): *left/right* with identical target objects. (B): *between (sideways)* with identical target objects. (C): *left/right* with identical target objects. (D): *between (sideways)* different target objects.

Experiment 2a: Evaluating our models on the flipped relations

Methods

Model Architectures. We use the architectures from Experiments 1a and 1b: MobiletNetV2, ResNeXt, and ViT-B/14.

Pretraining. We use the pretraining approaches explored in Experiments 1a and 1b: randomly initialized models, supervised pretraining on ImageNet, self-supervised temporal classification on SAYCam, and self-supervised training with DINO on both ImgeNet and SAYCam.

Stimulus Generation. We generate stimuli identically to Experiments 1a and 1b and rotate the images by 90 degrees.

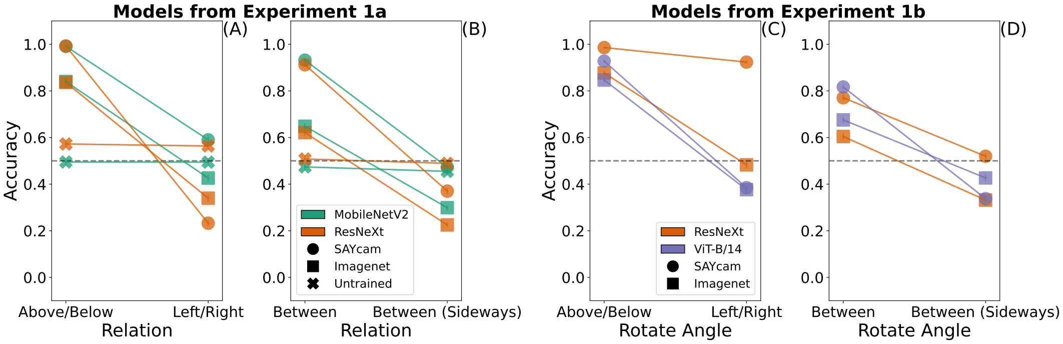

Figure 10: Most models evaluated perform below chance on the sideways-presented relations. Models evaluated in Experiment 1a show a substantial accuracy drop from the *above/below* to the *left/right* relation (panel A) and from the *between* to the *between (sideways)* relation (panel B), other than untrained models which are unaffected. Models evaluated in Experiment 1b show similar patterns (panels C and D), other than the ResNeXt models trained with DINO on the SAYCam dataset (we offer no explanation for this aberration). Colors indicate model architecture, and marker types indicate the training dataset. The dashed lines indicate chance accuracy (50%).

Results

Figure 10 depicts results on the “left/right” and “between (sideways)” relations with our models from Experiment 1a (panels (A) and (B)) and with DINO models from Experiment 1b (panels (C) and (D)). One configuration of models, ResNeXt models trained with DINO on SAYCam, performs well on the “left/right” relation (an abnormality we currently have no explanation for). The remaining models perform at chance or below on both relations, a substantial accuracy drop from the initial relations we examined. This represents a drastic qualitative deviation from the developmental results we model, where infants showed no meaningful change in the degree to which they construct a category representation for a relation contingent on whether it was presented vertically or horizontally. To attempt to isolate the cause of this effect, and identify potential conditions under which our models recover the developmental findings on these relations, we train several additional models in the next experiment.

Experiment 2b: Evaluating the effect of flipping data augmentations

Data augmentation refers to a set of techniques to modify the input data to a deep neural network as it is being trained, in an attempt to enable the network to learn representations that generalize and transfer better from limited amounts of data (Shorten and Khoshgoftaar, 2019). Horizontal axis flipping (across the vertical axis) is among the most common data augmentations for naturalistic image data. It is predicated on the natural symmetry across this axis (the mirror images of most objects are semantically similar or equivalent to their originals), and is trivial to implement. We hypothesize that it is this data augmentation that causes the effect we observe. By training models with horizontal flipping, we encourage our models to represent images with a target object to the left of a reference and images with a target object to the right of a reference similarly to each other, and perhaps more similarly than to other images depicting the same relation.

Methods

Model Architecture. We perform this experiment with the ResNeXt architecture used in Experiments 1a and 1b[2].

Pretraining. We train the models for this experiment on the SAYCam dataset (Sullivan et al., 2020) using temporal classification (Orhan et al., 2020), as in Experiment 1a. We train three variants on this, manipulating only the types of flips performed in data augmentation:

[2] As this experiment required training models in unique conditions, and produced clear results, we opted not to repeat it with other architectures from Experiment 1

Neither: a model trained without any flipping as part of its data augmentation suite.
Horizontal: a model trained with horizontal flipping as part of its data augmentation suite (identical to the baseline ResNeXt-SAYCam model in Experiment 1a).
Vertical: a model trained with vertical flipping as part of its data augmentation suite.
All other data augmentations (such as color jittering, random blurring, or random cropping) were identical between these models. We also note that these augmentations were only active during model pretraining. They were not active during any of the evaluations we report.

Stimulus Generation. We use the same approach to generating stimuli as is detailed in Experiment 1a, with the exception of our rotation procedure, which is detailed in subsection A.2 We rotate stimuli at angles of $30°, 45°, 60°, 90°, 120°, 150°$, and 180° counter-clockwise from the horizontal. When plotting the results, we group by the effective angle from the horizontal, e.g. as rotating at an angle of 120° is equivalent to 60° above the horizontal, we group the results for 60° and 120° under 60°. We render a collection of these rotated stimuli and the effect that each type of flipping would have on them, with one reference object (Figure B.11) and with two reference objects (Figure B.12).

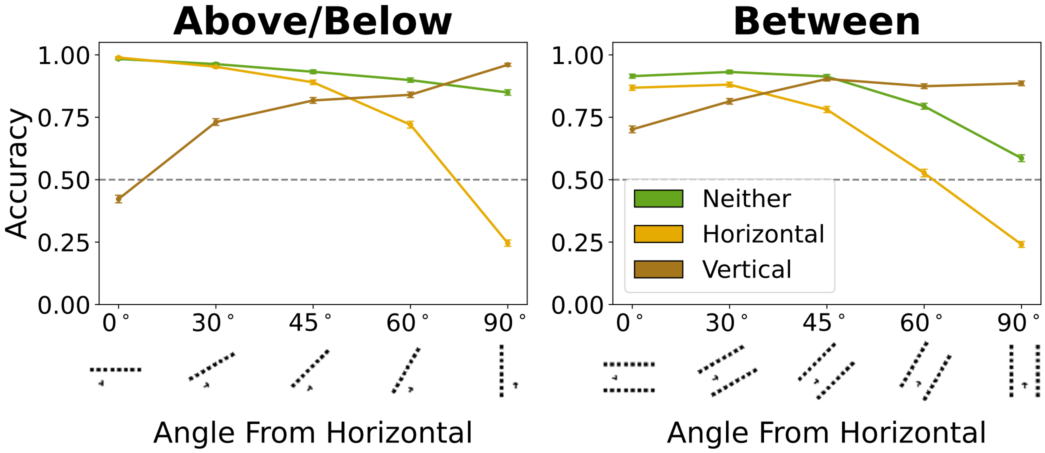

Figure 11: The presence and type of augmentation explains (most of) the change by stimulus angle. Models trained with neither flipping augmentations (green) show a slight degradation from 0° to 90°. Models trained with the standard horizontal flipping augmentation (yellow) show a drastic degradation, matching the results shown in Figure 10. Models trained with a non-standard vertical flipping augmentation show the opposite trend, improving gradually in accuracy from 0° to 90°. These patterns hold to varying extents in both relations.

Results

A summary of the results for the three flipping model variants, evaluated across the various stimulus rotation angles, is shown in Figure 11. We found that the model with horizontal augmentations only (plotted in yellow) recovered the results from previous experiments, with high levels of accuracy at 0° (compare to Figure 5), low ones at 90° (compare to Figure 10), and gradual degradation in the intermediate angles. The other two flipping models provide comparison cases to demonstrate the causal effect of the standard horizontal flipping. The model without any augmentations (plotted in green) showed a much more mild yet consistent degradation in accuracy from 0° to 90°. We take this gradation to be the extent to which the model learns to favor horizontal symmetry absent any data augmentation, only from the symmetries that naturally present themselves in the training data. Conversely, the model trained with vertical flipping (plotted in brown) depicted the opposite effect. Its accuracy was lowest at 0°, and as the stimuli are rendered closer to vertical, its accuracy gradually improved, peaking at 90°. Qualitatively, we find that the model trained with neither flipping directions recovers the developmental phenomenon from Quinn (2004), where discriminating *above/below* is equally easy as discriminating *left/right.* However, none of our models recover the other finding from Quinn (2004), that infants show an inability to discriminate between objects with respect to a diagonal reference object. Regardless of what sort of flipping was applied, the three models evaluated in Experiment 2b all reached fairly high levels of accuracy at 45°.

Experiment 2 Discussion

We evaluate the existing models from Experiment 1 and specially-trained models with custom data augmentations on stimuli rotated to various angles, and find that:

1. Unlike Quinn (2004, experiment 1), who found that infants distinguish "left or right" at a similar age to "above or below," many of our models have a strong preference for stimuli depicting vertical relations. We discover this is an artifact of the data augmentation often used to train these models, and show that models without data augmentation display a weaker preference.

2. Unlike Quinn (2004, experiment 3), who found that infants fail to distinguish between objects on opposite sides of a diagonal reference object (presented at an angle of 45 degrees), our models consistently categorize relations presented at this angle. This holds both relative to one reference object (*above/below* at 45 degrees) as well as relative to two reference objects (*between/outside* at 45 degrees).

Although our results from Experiment 1 broadly suggest that pre-trained computer vision neural networks can model important aspects of infant relation categorization, our findings suggest that some care and caution are required in the choice of model and training setup. Models trained with horizontal flipping, a data augmentation approach designed to mimic the horizontal reflection invariance many real-world objects display, struggled once the task evaluated was in direct contrast to the augmentation. Although we did not examine this in other contexts, it is not out of the question that other augmentations, such as color jittering, image solarization, or manipulations of image brightness or contrast might be in conflict with evaluating the development of visual perception.

Item (2) above summarizes a discrepancy from the developmental results with stimuli presented at an angle of 45°. The infants evaluated in Experiment 3 of Quinn (2004) were 3-4 month-olds, the youngest age bracket evaluated across the experiments surveyed. While infants at that age successfully appeared to develop a categorical response in the "above or below" condition, they failed to do so in the "between" condition, while 6-7 month-old infants were able to. Our models show comparable levels of accuracy for *above/below* at an angle of 45° and "between" at an angle of 0° (compare the accuracies for these angles in Figure 11). This suggests that to the extent the levels of accuracy displayed by our models track the developmental difficulty of these relations, we would predict that 6-7 month-old infants should be able to form category representations for "object on either side of a diagonal bar." We leave it to future work to experimentally examine this prediction.

Experiment 3: Classifying *above/below* and *between/outside* from 3D-rendered stimuli

In Experiment 1, we found that pretrained computer vision models appear to categorically represent spatial relations when evaluated with abstract stimuli that resemble developmental experiments. To study how generalizable our findings are, in this experiment we follow a similar methodology to Experiment 1 although with a different, more complex approach to stimulus rendering. Experiment 1 employed simple 2D renderings, either closely matching the stimuli Quinn showed infants (Figure 2) or in alternative control conditions (Figure B.1, Figure B.2). In this experiment, we evaluate models on 3D renders of scenes instantiating the same spatial relations Figure 12. These stimuli are more similar to the images used to train our models, and therefore allow evaluation of the extent to which model embeddings organize by categorical representations in more realistic data.

We begin our examination of the more realistic stimuli by reproducing our results from Experiments 1a and 1b, using the same models in similar conditions:

Methods

Model Architectures. We evaluate the three model architectures evaluated in Experiment 1: MobileNetV2, ResNeXt, and ViT-B/14.

Pretraining. We compare the same pretraining approaches from Experiments 1a and 1b. Experiment 1a: randomly initialized and untrained models, supervised classification pretraining on ImageNet, and self-supervised temporal classification on SAYCam. Experiment 1b: self-supervised pertaining using the DINO algorithm on both the ImageNet and SAYCam datasets.

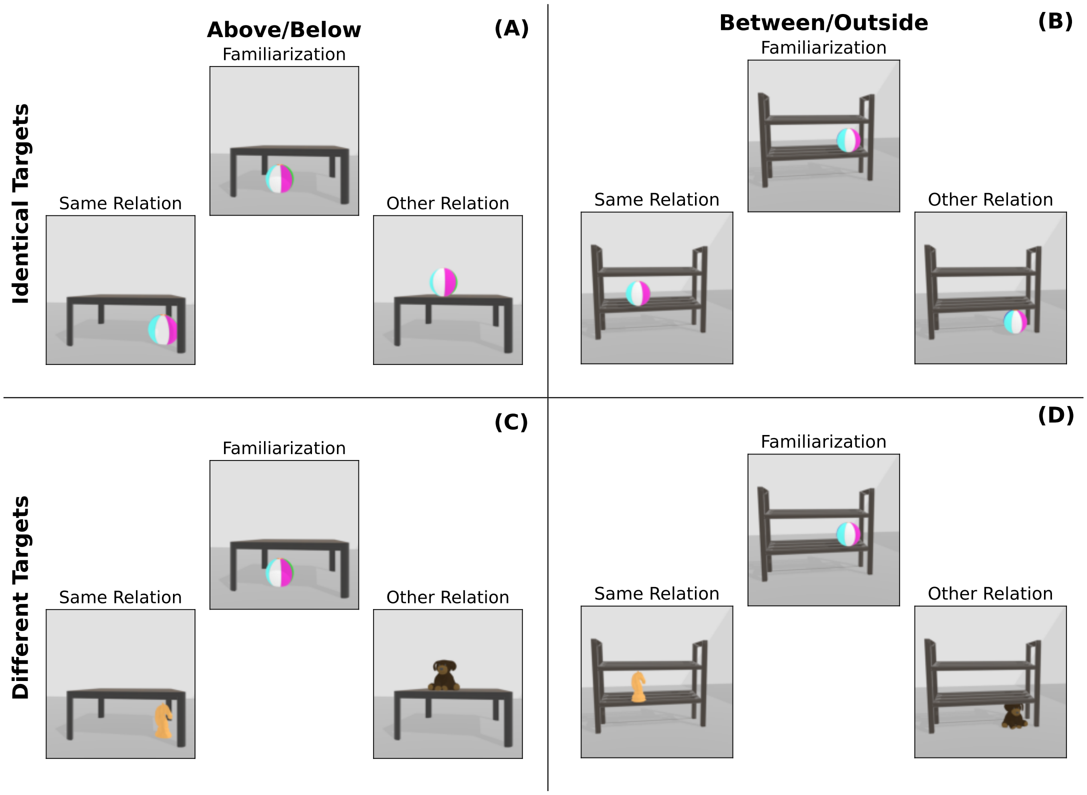

Figure 12: Experiment 2 Example Stimuli. In each triplet, the central stimulus is the familiarization example, the left one is the same relation test, and the right one is the other relation test. (A): *above/below* with identical target objects. (B): *between* with identical target objects. (C): *above/below* with identical target objects. (D): *between* different target objects.

Stimulus Generation. We render stimuli using Blender (Blender Online Community, 2018) (see Figure 12) following a similar procedure to the one described in Experiment 1a. We refer the reader to subsection A.3 for complete details. As in Experiment 1, we examine triplets where the target object matches between the familiarization and test stimuli ("identical targets"), and ones where the target object varies in the two test stimuli ("different targets"). We render scenes of both relations (*above/below* and *between/outside*) using eight different target objects: a beach ball, a chess knight, a Lego piece, a toy pineapple, a ping-pong paddle, a toy robot, a rubber duck, and a stuffed animal (see Figure B.13 for examples). We generate 256 unique scenes, each with all eight target objects, resulting in 2048 total triplets for each of the two relations.

Methods Summary. We evaluate the same set of models and pretraining approaches we used in Experiment 1, on the same two relations (*above/below* and *between*), We render stimuli that are richer than those used in Experiment 1 to examine how sensitive the models are to relational information in more visually complex stimuli. Our stimuli in Experiment 2 use one of eight different target objects. For each relation, we sample 2048 triplets and report the average accuracy for each model and pretraining approach—how often the embeddings for the congruent pair of stimuli are more similar (using cosine similarity) than the embeddings for the incongruent pair.

Results

A summary of our findings using 3D stimuli is shown in Figure 13, which parallels our previous findings using abstract stimuli (Figure 5). We again find the randomly initialized and untrained models around chance accuracy (results marked with an 'X'), and so we omit them from further discussion. We find that these stimuli tend to make the task harder for our models—most trained models have lower accuracies in this experiment than in the previous one. For instance, of the models evaluated in Experiment 1a, the MobileNetV2 models reached 3-10% lower accuracy levels in this experiment, and the ResNeXt models were 9-12% lower. The

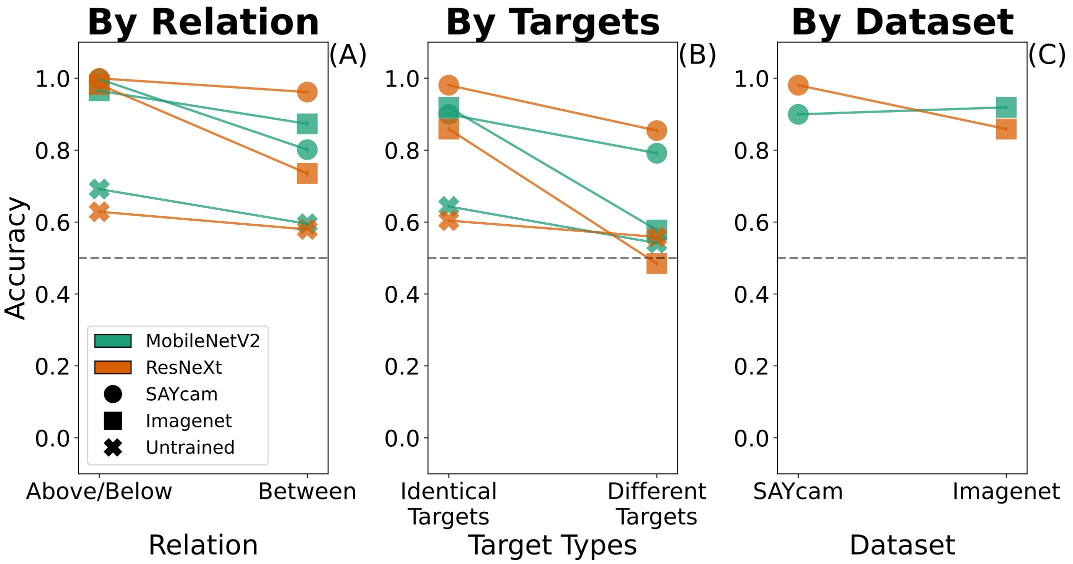

Figure 13: Models from experiment 1a recover the same patterns with 3D-rendered stimuli. Both in a comparison between relations (panel (A)) and identical or different targets (panel (B)), our models continue to recover the same developmental patterns with the more complex stimuli (compare this to Figure 5). The effect of the choice of the training dataset is not evident with these stimuli and is inconsistent across models. Color indicates the model architecture and marker type indicates the training method. The dashed line indicates chance accuracy (50%).

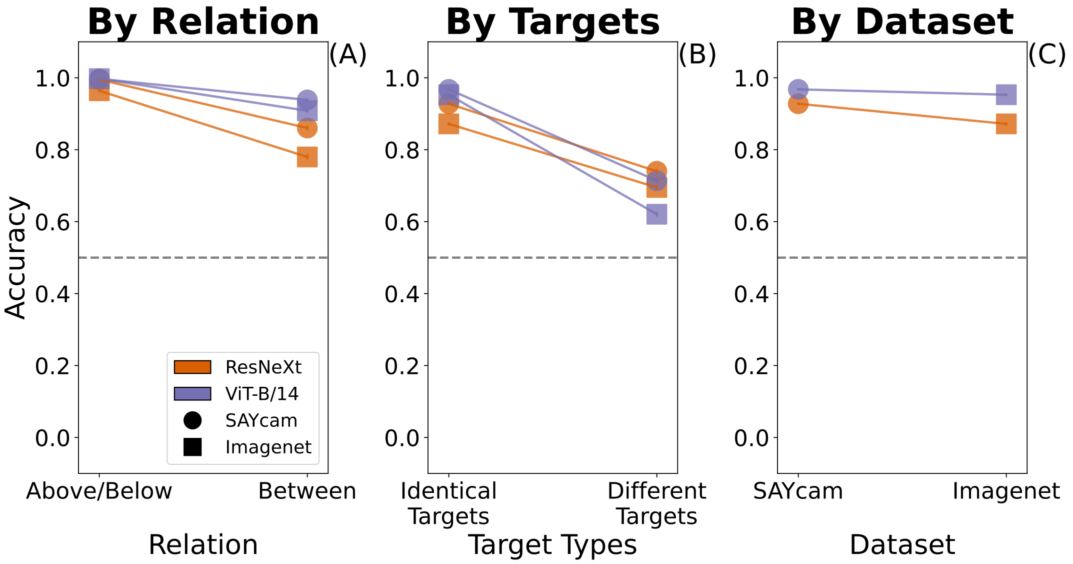

Figure 14: DINO-trained models continue to recover developmental phenomena of interest. As in Figure 13, our models continue to recover the phenomena studied in Experiment 1, both when comparing by relation (panel (A)) and when comparing by identical or different target objects (panel (B)). Color indicates the model architecture and the marker type indicates the training dataset. The dashed line indicates chance accuracy (50%).

DINO-trained models show a deviation in this pattern—the DINO ResNeXt models had an accuracy roughly 20% lower in this experiment, while the ViT-B/14 models had an accuracy that was 2-5% higher in this experiment compared to the previous one. However, even with the relative difficulty, we find the same pattern of results seen in the developmental experiments. The neural networks attain higher accuracy on the *above/below* task than on the *between* task, analogously to infants acquiring category representations for this relation earlier in development.

Our models consistently reach higher accuracy on the "identical targets" condition compared to the "different targets" one, reflecting the same pattern in infants. We also continue to observe a higher accuracy in models trained on the SAYCam dataset, although this effect is abated.

Figure 14 mirrors Figure 8, depicting results from our DINO-trained models. The same developmental phenomena previously outlined continue to present themselves: *above/below* is easier than *between*, identical targets are easier than different ones, and SAYCam-trained models (slightly, yet consistently) outperform ImageNet-trained ones (matched-pairs T-test, $t = 4.109, P < 0.005$).

Discussion

We reproduce the developmental phenomena explored in Experiment 1 with 3D-rendered stimuli. The use of richer, 3D-rendered stimuli allows us to conclude that our models' ability to represent relational categories, and their ability to mirror findings from the developmental literature, is not an artifact of using the simplistic stimuli of Experiment 1. Given the discovery that we can reproduce findings on spatially simple relations such as "above or below" or "between or outside" with more realistic stimuli, we ask: can we reproduce developmental patterns with more complex relations as well?

Experiment 4: Classifying *containment*, *behind*, and *support* from 3D-rendered stimuli

Following the success of replicating Experiment 1's results with 3D rendered stimuli in Experiment 3, in this experiment we use similarly rendered stimuli to examine more spatially complex relations. Casasola et al. (2003, Experiment 2[3]) studied whether 6 months old infants can categorize scenes based on whether or not they show a containment relation. Infants were habituated to a short video depicting a hand placing an object inside a container, that is, in a containment relation (Figure 15(A/B), "Familiarization"). Infants were then tested with the familiarization video and with three novel probes. The test probes varied along two key dimensions: relation (whether or not they also depicted a containment scene) and occlusion (what fraction of the target object is visible at the end of the video). The first test probe (Figure 15(A/B), "Same relation") showed the same event filmed from a higher camera angle: this produces the same relation but with novel occlusion—the higher camera angle causes much more of the object to be visible. The second test probe (Figure 15(A), "Other relation") uses the same high angle but places the object behind the container. This results in similar occlusion to the familiarization video but with the object placed in a novel relation with respect to the container. Finally, the third probe (Figure 15(B), "Other relation") offered both a novel relation and occlusion: filmed from the same angle, this test showed an object being placed on top of an upside-down container, in a support relation. This final test probe serves as a control condition with mismatches on both relation and occlusion. As expected, Casasola et al. (2003, Figure 2) found the lowest test-time looking times when showing the familiarization stimulus a second time. The 'containment' test probes (with novel degrees of occlusion) were found to elicit significantly shorter looking times than the 'behind' stimuli, which depict a novel relation with similar degrees of occlusion to the familiarization stimulus. On this basis, Casasola et al. (2003) conclude (and see also Casasola (2008) for a review) that 6-month-old infants successfully form a category representation for the containment relation. In this experiment, we will evaluate whether the models we tested in Experiments 1 and 2 can replicate these patterns.

A key methodological difference between our experimental setup and the one used by Casasola et al. (2003) is our use of still images, rather than videos. We motivate this decision from two perspectives. First, to be able to compare to our previous results in Experiments 1 and 2, we wished to use the same models, and as these models are trained on single images[4], rather than videos, we opted to adapt the task. Second, models trained for image classification or self-supervised image-level tasks are more widely available than models trained for video classification. To the extent we hope this work can serve as methodological inspiration for studying other developmental phenomena with pretrained models, we wished to examine whether translating video stimuli to representative still images is sufficient to recover

[3] We skip the perceptually mismatched stimuli examined in Experiment 1 by Casasola et al. (2003) and proceed directly to the better-controlled stimuli the authors used in Experiment 2.

[4] With the exception of the models trained using Temporal Classification with the SAYCam dataset, which are trained using the temporal ordering of short video clips.

developmental findings. We generate our stimuli to match the terminal frames of the videos Casasola et al. (2003) used (Figure 15, top). As in Experiments 1 and 2, we generate stimuli triplets to compare the similarity between an embedding of a single familiarization and the embeddings of two test probes. We visualize our two comparison cases in the bottom half of Figure 15. In both cases, we use a familiarization stimulus showing a containment event from a low angle, similar to the familiarization event used by Casasola et al. (2003). We also use the same type of same relation test stimulus, depicting a containment event from a higher angle. In one condition, "Containment vs. Behind," we compare to a stimulus depicting the target object behind the container, rendered from the same higher angle (Figure 15, left). In the other condition, "Containment vs. Support," we compare to a stimulus rendering the target object supported by the container, with the container flipped upside-down, also rendered from the same higher angle (Figure 15, right).

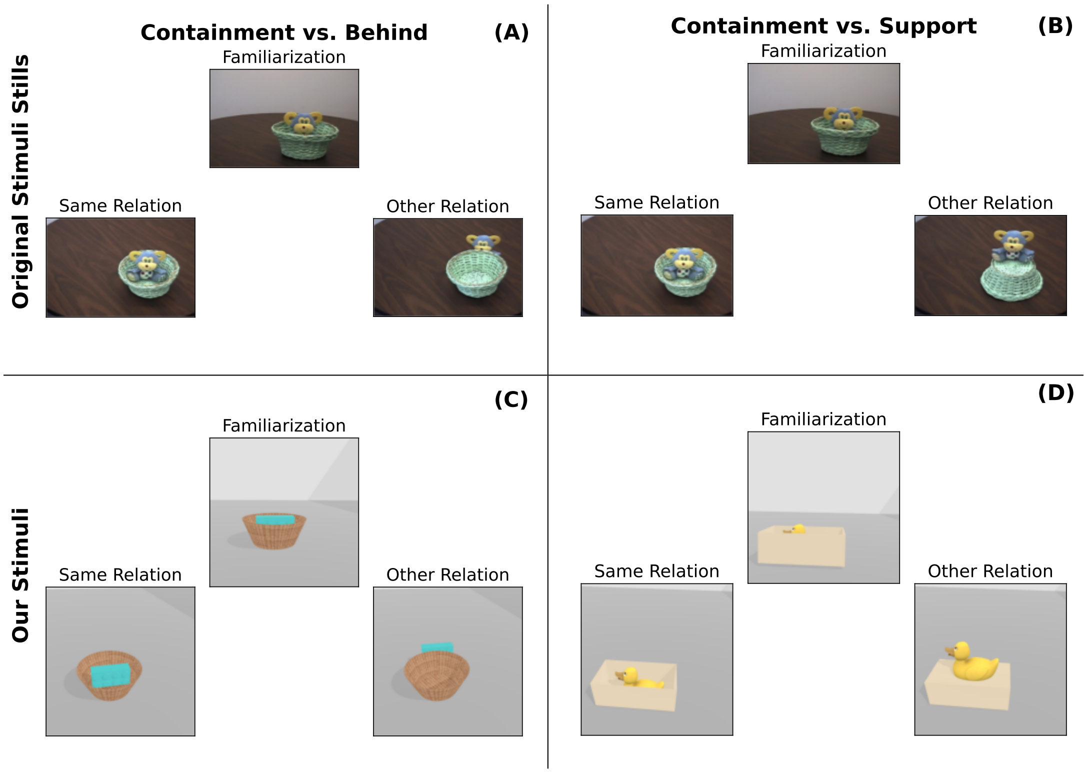

Figure 15: Experiment 4 Example Stimuli. In each triplet, the central stimulus is the familiarization example, the left one is the same relation test, and the right one is the other relation test. Top: the final frames of the stimuli video presented to infants, reproduced from Casasola (2008, Figure 1). Bottom: our rendering of matching stimuli. Left: comparing a test probe depicting the *containment* relation to a test probe depicting the *behind* relation. Rendered using the wooden basket container and Lego target object. Right: comparing a test probe depicting the *containment* relation to a test probe depicting the *support* relation. Rendered using the shorter cardboard box container and rubber duck target object.

Experiment 4a: Evaluating the containment relation

Methods

Model Architectures. We evaluate the same three architectures from Experiments 1 and 2: MobileNetV2, ResNeXt, and ViT-B/14.

Pretraining. We use the same pretraining approaches from the previous experiments: randomly initialized models serving as a control, supervised pretraining on ImageNet, self-supervised pretraining on SAYCam, and models trained using self-supervised DINO on both the SAYCam and ImageNet datasets.

Stimulus Generation. As in Experiment 3, we use Blender (Blender Online Community, 2018) to render stimuli. Our stimuli use four different containers, and eight different target objects, identical to the ones used in Experiment 3. For additional details, see subsection A.4. We render four images for each stimulus (see Figure 15(C/D) for examples split into triplets). The first is the familiarization stimulus, with the target object in the containment relation and a lower camera angle. We then raise the camera to a higher angle and render three test stimuli. The first is the *containment* test stimulus, where we render the same scene as in the familiarization stimulus from the new camera angle, matching the familiarization relation but differing in occlusion. The second is the *behind* test stimulus, where we move the target object behind the container, creating similar occlusion between the container and test object to the familiarization stimulus, but in a different spatial relation. The third is the *support* test stimulus, where we flip the container upside-down, and place the target object on top of it. This offers a stimulus mismatched in both dimensions (relation and occlusion) to the familiarization stimulus. We render 128 unique scenes by sampling different camera parameters (see subsection A.4 for full details), each with all 32 combinations of the four containers and eight target objects, resulting in a total of 4096 unique sets of stimuli. See Figures B.14 to B.17 for visualizations of scenes with all objects, and Figure B.21 for visualization of camera parameter variations for a single object combination.

Methods Summary. We evaluate the same collection of models pre-trained on the same datasets as in Experiments 1 and 2. We compare accuracies in two primary conditions. In the first, "Containment vs. Behind," we use a low-angle *containment* scene as our familiarization stimulus, and use test probes depicting a *containment* scene rendered from a higher angle, and a *behind* scene rendered from the same higher angle. In the second, "Containment vs. Support,", we use the same familiarization stimulus and first test probe and replace the *behind* test probe with a *support* scene rendered from the same higher angle. In both conditions, we report the average accuracy over the 4096 sets of stimuli—how often the embeddings for the pair of stimuli depicting a containment relation are more similar (using cosine similarity) than the embeddings for the familiarization stimulus and the incongruent test probe.

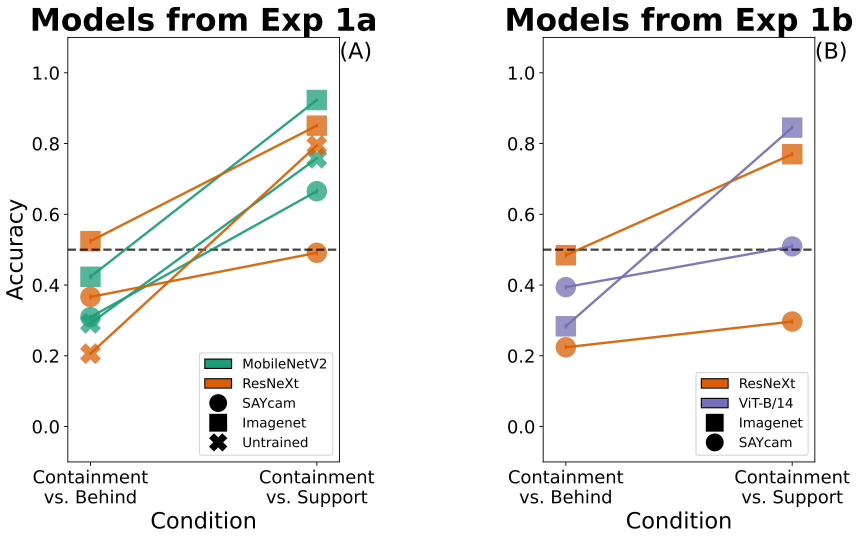

Figure 16: Models prefer the *containment* test probe over the *support* one, but not over the *behind* one. We compare the average accuracy over triplets where the foil test probe depicts the *behind* relation ("Containment vs. Behind", left data points) to the average accuracy over triplets where the foil test probe depicts the *support* relation ("Containment vs. Support", right data points). Our models, including the untrained ones, are fairly consistent with their preferences. The *containment* test probe is often judged as more similar to the familiarization stimulus when paired with a *support* foil, but not when paired with a *behind* foil. This holds with both the models discussed in Experiment 1a (panel (A)) and the DINO-trained models introduced in Experiment 1b (panel (B)). The color indicates the model architecture and the marker type indicates the training method. The dashed line indicates chance accuracy (50%).

Results

We compare our models' levels of accuracy between the "Containment vs. Behind" condition and the "Containment vs. Support" condition in Figure 16. We find that across various training datasets and model architectures, all of our models reached higher levels of accuracy when comparing two *containment* scenes to a *support* scene. This by itself did not surprise us, as this is the easier foil relation, which does not match the degree of occlusion in the familiarization stimuli (compare the right-hand triplets in Figure 15 to the left-hand ones). In the better-matched condition of "Containment vs. Behind," our models peaked around chance accuracy. That is, most models we evaluated systematically found the *behind* test probes (which match in occlusion, but not in relation) more similar to the familiarization containment stimuli than the *containment* test probes. This is in direct opposition to the patterns infants demonstrated in Casasola et al. (2003), who found the *containment* test stimuli to be the least surprising ones. In another reversal from Experiment 1b, we found that the models trained on ImageNet (both using DINO and in a supervised fashion) outperformed the models trained on SAYCam (compare the circle markers to the squares in Figure 16). A final unexpected result was a consistent preference present in the randomly initialized model, reaching substantially lower accuracies in the "Containment vs. Behind" condition than in the "Containment vs. Support" one. To verify it is not the result of random noise, we replicated our initial randomly initialized model with nine additional ones. We demonstrate in Figure B.18 that while the degree of this preference varies, all of our randomly initialized models (across both the MobileNetV2 and ResNeXt architectures) replicated this pattern.

We hypothesized that one potential culprit in the models' failure in the "Containment vs. Behind" condition might be the pooling operations that precede our embedding extraction. In the MobileNetV2 and ResNeXt architectures (but not in the ViT-B/14 one), the final two-dimensional representation of each input image is pooled in order to create an embedding vector. The pooling mechanism struck us as potentially related to the failure as it collapses much of the spatial information, which might leave more remaining information in the degree of occlusion (which roughly corresponds to how many pixels of the target object are visible) than in the spatial relation. To examine this hypothesis, we repeat our similarity judgments, using embeddings extracted before the pooling operation. We find that our models consistently reached higher accuracy when similarity was compared using embeddings extracted before pooling (see subsection B.9 for the complete details, Figure B.20 for a summary of the results, and Table 2 for the complete results). Excluding the untrained models, our models reach a mean accuracy of 0.798 pre-pooling in the "Containment vs. Behind" condition compared to a mean accuracy of 0.389 post-pooling. Similarly, in the "Containment vs. Support" condition, our models reach a mean accuracy of 0.933 pre-pooling compared to a mean accuracy of 0.666 post-pooling. This supports a hypothesis that the relational information is more prominent in the pre-pooling embeddings, compared to the post-pooling embeddings, at least as measured by the cosine similarity. It appears sensible that the pooling operation, which collapses across spatial locations in an attempt to extract a location-invariant representation of the stimulus, reduces the degree of spatial and relational information preserved. We examine to what extent the relational information remains present in the final embeddings in the next experiment.

Experiment 4b: Linearly decoding the containment relation

All results presented so far examine relational information through the lens of embedding similarity between a familiarization example depicting a particular relation and two test probes, one depicting the same relation and one a foil depicting a different relation. In this experiment we take a more direct approach using supervised learning, and ask the following question: can relational information be decoded, using a single linear layer, from the embeddings created by our models? As we previously discovered that pooling results in lower similarity by relation, we examine the pooled embeddings, assuming that any information present after pooling would also be present before the pooling operation.

Methods

Model Architectures and Pretraining. We evaluate the same models, using the same pretraining approaches, as Experiment 4a.

Stimulus Generation. We use the same dataset of stimuli generated for Experiment 4a.

Linear Decoding Datasets. We evaluate each model and pretraining method across many systematic partitions of our dataset. In each partition, we split the full set of 4096 sets of stimuli into a training set and a test set. Each partition is defined by a held-out container, a held-out target object, and a held-out fraction (12.5%) of the camera configurations. See Figure 17 for an example and subsection A.4 for the full details.

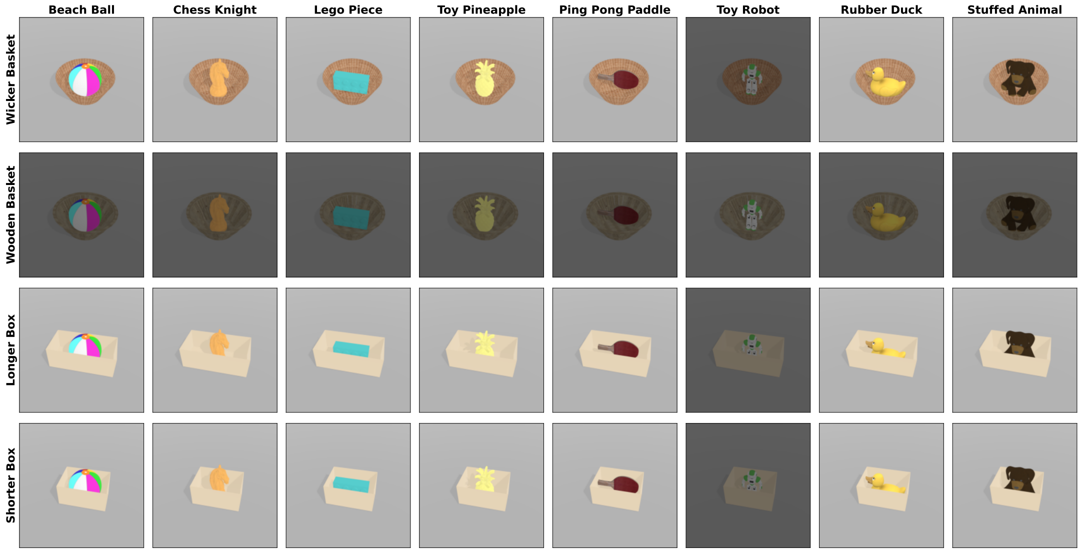

Figure 17: Linear Decoding Dataset Visualization. We visualize a partition of the dataset for a linear decoding experiment, for a single camera configuration (see Figure B.21 for a visualization of the between-scenes variations induced by the camera configurations). In each partition, all stimuli using a particular reference object (here the Wooden Basket) and a single target object (here the Toy Robot) were held out as a test set. We train a linear decoder to classify the relation in a scene from the embeddings produced by a model over the training set and then evaluate the decoder on the test set.

Task Setup. We treat the model evaluated as an embedding extractor, and the model itself is not modified or updated during this task. We train a single linear layer, which takes in an embedding of an image as input and outputs the logit (unnormalized log-probability) of the image depicting each relation (containment, support, or behind) as an output. We then update the weights of this linear layer based on the cross-entropy loss between the layer's prediction and the true relation in the image. We continue to update this linear decoder until it stops improving on the validation set, at which point we also evaluate it on the test set.

Results

We summarize the average accuracy reached by decoders trained on embeddings from different models in Figure 18. We compute the accuracy only over the test-set examples, that is ones that have either a held-out container, or a held-out target object, or a held-out camera configuration (or multiple held-out items). Other than the decoders trained on the randomly-initialized models, we find almost perfect accuracies across the various training methods, datasets, and model architectures. To explore the extent to which decoding difficulty varies by held-out container and object, Table 3 provides a breakdown of decoding results. In the table, we focus only on examples that depicted the held-out container and held-out target object in each dataset, and only from the held-out configurations. These are examples in which the decoders received no training on either the target object, the container, or the exact camera configuration used in the scene, and as such offer the strongest measure of generalization. While there is some variability by the particular objects, the lowest held-out accuracy reached is 0.927 on the combination of the wicker basket container and stuffed animal target object. We take this as evidence that the post-pooling embeddings contain sufficient relational information to decode the relations with almost perfect accuracy, even if embedding similarity is not primarily driven by the relation depicted.

Discussion

In this experiment, we examine the extent to which our success in replicating phenomena in categorizing simpler visual relations (above or below, between or outside) transfers to more complex relations (containment, behind, support). Unlike in previous experiments, where models, for the most part, mirrored developmental phenomena, here we fail to recover the main phenomenon of interest. We use containment stimuli rendered from a low angle as our

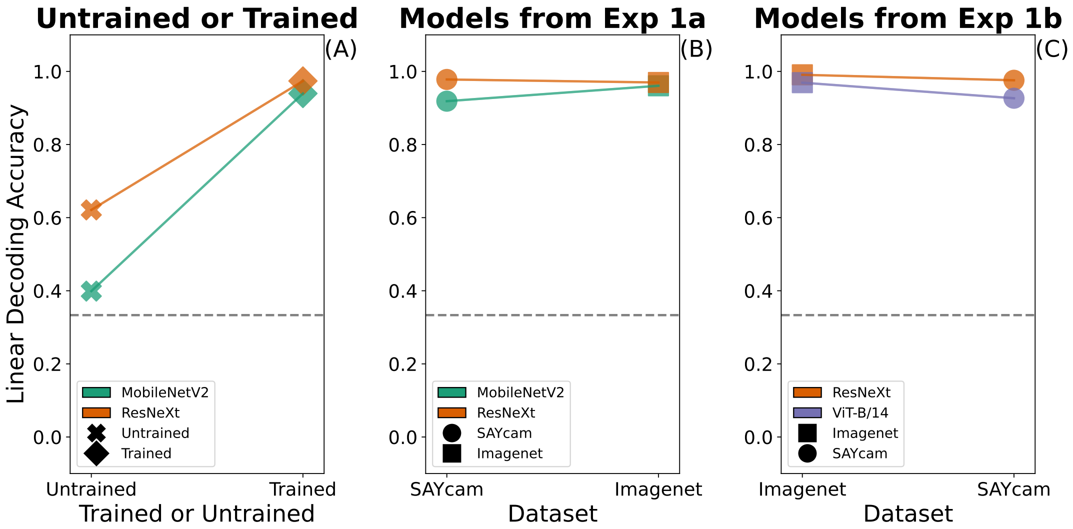

Figure 18: Linear decoders successfully recover relation from trained model embeddings. Other than the untrained models (panel (A), left), decoders trained on embeddings from our various models all classify relations at high accuracies. We depict the average test-set accuracy of decoders trained on the embeddings of our different models, averaging over the various partitions of our dataset into train and test sets. The color indicates the model architecture and the marker type indicates the training method. The dashed line indicates chance accuracy (33%, as there are three categories).

familiarization examples, and three types of test probes: *containment* stimuli rendered from a higher angle (matching on the relation, but not object occlusion), *behind* stimuli (matching on occlusion, but not on the relation), and *support* stimuli (matching on neither relation nor occlusion). In Experiment 4a, we discover that our models repeatedly embed the *behind* test probes more similarly to the familiarization stimuli than the *containment* test probes. This is inconsistent with the findings outlined in Casasola et al. (2003), where infants measured lower looking times to the *containment* test over the *behind* one. When tasking our models to compare the *containment* test probe to the easier *support* one, our models do substantially better. Therefore, we hypothesize that similarity in the model embeddings is driven first by lower-level perceptual features, and second by higher-level relational ones[5]. In Experiment 4b, we validate that relational information is maintained in the final embeddings, as we successfully decode with very high levels of accuracy.

The model-to-infant comparison in this experiment is less faithful than in Experiments 1 and 2, as the infants in Casasola et al. (2003) watched short video clips depicting the object being placed in the specified relation to the container, rather than making judgments based on still images. We are not aware of any work investigating the extent to which infants make similar relational judgments of the containment relation from still images. To the extent Experiments 1 and 2 demonstrated that pretrained neural network models can successfully recover patterns in infant relation categorization, we would hypothesize that infants would be less consistent in judging stimuli by relational similarity if only offered still images. We note that while the models developed by Ullman et al. (2019) successfully categorize still images of relations, they do so after being trained on video stimuli. Their findings imply that deep neural network models trained on videos could potentially offer a closer match to the findings outlined by Casasola et al. (2003), and we leave that for future work to examine.

General Discussion

We investigate the capacity of various large-scale pretrained computer vision neural network models to replicate findings regarding the development of relation categorization. We first find that without explicit relational training, the trained models we evaluate learn embeddings that

[5] Casasola and Cohen (2002) and Casasola et al. (2003, Experiment 1) raised the concern that perhaps infants also make similarity judgments according to such lower-level features, and assuaged this concern in Casasola et al. (2003, Experiment 2).

tend to represent stimuli depicting the same relation more similarly to each other than stimuli representing different relations. We then successfully recover most patterns of interest relating to how infants process relations such as “above or below” and “between or outside” (Quinn, 2003). We observe that our models show similar difficulty gradations to the infants: our models reach higher accuracy levels on the *above/below* relation than on the *between/outside* one, matching infants’ ability to form categorical relations for the former earlier in development than for the latter. Infants also respond categorically to stimuli depicting identical target objects earlier in development than to stimuli using different target objects; likewise, the models we evaluate have higher accuracy levels when using identical target objects than when using different ones. We encounter these patterns both with 2D stimuli closely resembling the developmental ones (Experiment 1) and with rendered 3D stimuli that more closely resemble the data our models were pretrained on (Experiment 3). However, when evaluating the same pretrained neural networks on the *containment* relation (Casasola et al., 2003), we find (in Experiment 4) that the models appear to organize their embedding primarily by object visibility and secondarily by relation, even when relational information is present. This is evident in the models’ consistently lower levels of accuracy when probed with a foil that is matched on occlusion (depicting the *behind* relation), compared to when probed with a foil that is mismatched on occlusion (with the *support* relation).

We find that shortcomings in our models’ abilities to replicate developmental patterns, and the variation between models, can help highlight methodological nuances meaningful for future work. In Experiment 2a, we find that many models are consistent with infants’ patterns when a relation is presented vertically (e.g. “above or below”), but drastically inconsistent with the same patterns when a relation is presented horizontally (e.g. “left or right”). To explain this inconsistency, we evaluate (in Experiment 2b) the effect of image flipping, a particular form of data augmentation often used in pretraining computer vision models, and discover that the use of image flipping explains the observed deviation. We also find that on the visually simpler relations of “above or below” and “between”, models trained on the egocentric, developmentally realistic SAYCam dataset (Sullivan et al., 2020) outperform models trained on ImageNet, using both simpler stimuli (Experiment 1b) and more complex, rendered stimuli (Experiment 3). We find no such pattern on the *containment* relation stimuli evaluated in Experiment 4. In a supplemental experiment (Appendix C), we identify that for neural networks to recover similar patterns from symbolic inputs, they should flexibly allow comparing between multiple objects, as only the architectures that cannot (the MLP and RelationNet) struggled to learn a relation entirely.

One novel contribution we make is to evaluate models trained on SAYCam (Sullivan et al., 2020), the best available proxy for a child’s visual experience, using developmental behavioral paradigms. Prior work has used this dataset to train models (such as some of the pretrained models from Orhan et al., 2020 we use in this work), or to evaluate such models on cognitive biases (Tartaglini et al., 2022). Ongoing work by Vong et al. (in prep) uses this dataset to evaluate grounded language acquisition through cross-situation word learning. We hope that this line of work can serve as inspiration for future work in computational developmental psychology and computer vision. From the perspective of developmental psychology, we are excited about the ability to evaluate suggested computational mechanisms in the context of rich, large-scale data—beyond, for instance, fitting models to choice data in an attempt to compare them, we can now train models on proxies of perceptual inputs and examine emergent phenomena of these models. We can engineer these models to be amenable to evaluation using closely modified versions of developmental paradigms, facilitating closer comparison between models and infants. From the perspective of computer vision, these rich comparisons can also help elucidate whether proposed models of human perception can achieve the abilities of a developing child. Although there will always be implementation-level (Marr, 1982) differences between artificial approaches and human biology, these advancements allow us to compare computational- and algorithmic-level approaches to vision, and study whether given developmentally realistic experience, they offer infant-like results. Our approach also differs from most work on learning relations with deep neural networks. Prior work (Santoro et al., 2017; Shanahan et al., 2019) focused on developing and evaluating custom architectures for relation learning. We show that the embeddings learned by pretrained models with no explicit relational bias allow judging similarity based on relation, and we leave it to future work to further study how much relational information is decodable from these embeddings.

When infants discriminate between categories in a laboratory study, it is often unclear whether these abilities reflect top-down processing of categories acquired outside the lab, or bottom-up processing of categories developed during the familiarization phase (Thelen and Smith, 1994; Murphy, 2002, ch. 9; French et al., 2004, Newcombe et al., 2005). Our findings are consistent with both possibilities. Our supplemental experiment in Appendix C directly examines the learnability of relational categories using a supervised learning paradigm on datasets ranging from as few as 8 examples to a few thousand data points. We view this as analogous to the first possibility, of learning relation concepts from numerous varied examples, as infants might acquire these categories over an extended period outside the lab. The pretrained computer vision models used in Experiments 1-4 do not separate between the top-down and bottom-up hypotheses. The pretraining process guides the model in acquiring useful perceptual features to represent its inputs, which may also serve in promoting relational similarity in the models' embeddings. These models may also acquire a more abstract latent concept of the different relations—as we cannot rule this possibility out, we cannot adjudicate between top-down processing of prior categories and bottom-up processing of categories developed in familiarization.

Finally, our work allows us to make an experimental prediction and raise a source of uncertainty. In Experiment 2b, we discovered that the pretrained models we evaluated reach high levels of accuracy when stimuli are presented at a 45° angle, unlike the infants evaluated by Quinn (2004, Experiment 3). The high levels of accuracy reached by the models make the prediction that slightly older infants (e.g. 6-7-months-old) than those evaluated by Quinn (2004) would demonstrate evidence for a category representation for an object on either side of a diagonal line. We also note a lack of experimental evidence (to the best of our awareness) for whether or not infants construct category representations for the containment relation from static stimuli. Both experimental work (Casasola and Cohen, 2002; Casasola et al., 2003) and computational models Ullman et al. (2019) rely on dynamic video stimuli. Further experimental work could demonstrate at what stage of development a categorical response to still image stimuli depicting the containment relation is acquired, which would shed light on the discrepancy between our findings and existing experimental results.

Acknowledgments

The authors would like to thank Wai Keen Vong for helpful ideas and thoughtful discussions at several points in time over this project's various stages. We appreciate the feedback we received from The Human and Machine Learning Lab at NYU during lab meetings. We thank Wai Keen Vong, Reuben Feinman, Pat Little, and Alexa Tartaglini for their feedback on the draft. Finally, we would like to thank Kenna Heller for her work on Figure 1.

This work was supported by the DARPA Machine Common Sense program and NSF Award 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science.

References

Baldassarre, F., Smith, K., Sullivan, J., and Azizpour, H. (2020). Explanation-based weakly-supervised learning of visual relations with graph networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12373 LNCS:612–630.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks.

Battleday, R. M., Peterson, J. C., and Griffiths, T. L. (2021). From convolutional neural networks to models of higher-level cognition (and back again). *Annals of the New York Academy of Sciences*, 1505:55–78.

Blender Online Community (2018). *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

Bomba, P. C. and Siqueland, E. R. (1983). The nature and structure of infant form categories. *Journal of Experimental Child Psychology*, 35:294–328.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660.

Casasola, M. (2008). The Development of Infants' Spatial Categories. *Current Directions in Psychological Science*, 17(1):21–25.

Casasola, M. and Cohen, L. B. (2002). Infant categorization of containment, support and tight-fit spatial relationships. *Developmental Science*, 5(2):247–264.

Casasola, M., Cohen, L. B., and Chiarello, E. (2003). Six-month-old infants' categorization of containment spatial relations. *Child Development*, 74(3):679–693.

Colunga, E. and Smith, L. B. (2005). From the lexicon to expectations about kinds: A role for associative learning. *Psychological Review*, 112:347–382.

Donahoe, J. W. and Dorsel, V. P. (1997). *Neural-network models of cognition : biobehavioral foundations.* Elsevier.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations.*

Eimas, P. D. and Quinn, P. C. (1994). Studies on the formation of perceptually based basic-level categories in young infants. *Child Development*, 65:903–917.

Falcon, W. (2019). Pytorch lightning. *GitHub. Note: https://github.com/williamFalcon/pytorch-lightning Cited by*, 3.

Fantz, R. L. (1964). Visual experience in infants: Decreased attention to familiar patterns relative to novel ones. *Science*, 146(3644):668–670.

French, R. M., Mermillod, M., Mareschal, D., and Quinn, P. C. (2004). The role of bottom-up processing in perceptual categorization by 3- to 4-month-old infants: Simulations and data. *Journal of Experimental Psychology: General*, 133(3):382–397.

Glasbey, C., van der Heijden, G., Toh, V. F. K., and Gray, A. (2007). Colour displays for categorical images. *Color Research & Application*, 32(4):304–309.

Goldstone, R. L. and Hendrickson, A. T. (2010). Categorical perception. *WIREs Cognitive Science*, 1(1):69–78.

Huttenlocher, J. and Newcombe, N. S. (1984). The child's representation of information about location. In Sophian, C., editor, *Origin of cognitive skills*, pages 81–111. Erlbaum, Hillsdale, NJ.

Johnson, S. P. (2010). How Infants Learn About the Visual World. *Cognitive Science*, 34(7):1158–1184.

Kim, B., Reif, E., Wattenberg, M., Bengio, S., and Mozer, M. C. (2021). Neural networks trained on natural scenes exhibit gestalt closure. *Computational Brain and Behavior*, 4:251–263.

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *ICLR 2015.*

Kovesi, P. (2015). Good Colour Maps: How to Design Them.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *The Twenty-sixth Annual Conference on Neural Information Processing Systems (NIPS).*

Lindsay, G. W. (2021). Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of Cognitive Neuroscience*, 33:2017–2031.

Liu, N., Li, S., Du, Y., Tenenbaum, J. B., and Torralba, A. (2021). Learning to compose visual relations. *Advances in Neural Information Processing Systems*, 28:23166–23178.

Mareschal, D., French, R. M., and Quinn, P. C. (2000). A connectionist account of asymmetric category learning in early infancy. *Developmental psychology*, 36(5):635–645.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information.* MIT press.

Murphy, G. (2002). *The Big Book of Concepts.* MIT Press, Cambridge, MA.

Newcombe, N. S., Sluzenski, J., and Huttenlocher, J. (2005). Preexisting knowledge versus on-line learning: What do young infants really know about spatial location? *Psychological Science*, 16(3):222–227.

Orhan, A. E., Gupta, V. V., and Lake, B. M. (2020). Self-supervised learning through the eyes of a child. In *NeurIPS 2020.* arXiv.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NeurIPS 2017.*

Piaget, J. (1954). *The construction of reality in the child.* Basic Books, New York, NY.

Quinn, P. C. (1994). The Categorization of Above and Below Spatial Relations by Young Infants. *Chil Dev*, 65(1):58–69.

Quinn, P. C. (2002). Category Representation in Young Infants. *Current Directions in Psychological Science*, 11(2):66–70.

Quinn, P. C. (2003). Concepts are not just for objects: Categorization of spatial relation information by infants. In Rakison, D. H. and Oakes, L. M., editors, *Early category and concept development: Making sense of the blooming, buzzing confusion*. Oxford University Press.

Quinn, P. C. (2004). Spatial representation by young infants: Categorization of spatial relations or sensitivity to a crossing primitive? *Memory and Cognition*, 32(5):852–861.

Quinn, P. C., Adams, A., Kennedy, E., Shettler, L., and Wasnik, A. (2003). Development of an abstract category representation for the spatial relation between in 6- to 10-month-old infants. *Developmental Psychology*, 39(1):151–163.

Quinn, P. C., Cummins, M., Kase, J., Erin, M., and Weissman, S. (1996). Development of categorical representations for above and below spatial relations in 3- to 7-month-old infants. *Developmental Psychology*, 32(5):942–950.

Quinn, P. C., Norris, C. M., Pasko, R. N., Schmader, T. M., and Mash, C. (1999). Formation of a categorical representation for the spatial relation between by 6- to 7-month-old infants. *Visual Cognition*, 6(5):569–585.

Regier, T. (1995). A model of the human capacity for categorizing spatial relations. *Cognitive Linguistics*, 6(1):63–88.

Rogers, T. T. and Mcclelland, J. L. (2014). Parallel distributed processing at 25: further explorations in the microstructure of cognition. *Cognitive science*, 38:1024–1077.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. In *IJCV*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *CVPR 2018*.

Santoro, A., Raposo, D., Barrett, D. G. T., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T., and London, D. (2017). A simple neural network module for relational reasoning. In *NeurIPS 2017*.

Shanahan, M., Nikiforou, K., Creswell, A., Kaplanis, C., Barrett, D., and Garnelo, M. (2019). An Explicitly Relational Neural Network Architecture.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48.

Sullivan, J., Mei, M., Perfors, A., Wojcik, E., and Frank, M. (2020). SAYCam: A large, longitudinal audiovisual dataset recorded from the infant's perspective.

Tartaglini, A. R., Vong, W. K., and Lake, B. M. (2022). A developmentally-inspired examination of shape versus texture bias in machines. *Proceedings of the 44th Annual Meeting of the Cognitive Science Society: Cognitive Diversity, CogSci 2022*, pages 1284–1290.

Thelen, E. and Smith, L. B. (1994). *A dynamic systems approach to the development of cognition and action*. MIT Press, Cambridge, MA.

Ullman, S., Dorfman, N., and Harari, D. (2019). A model for discovering 'containment' relations. *Cognition*, 183:67.

van der Maaten, L. and Hinton, G. (2008). Visualizing high-dimensional data using t-sne.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention Is All You Need. In *NeurIPS 2017*, Long Beach, CA, USA.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *CVPR 2017*, pages 5987–5995.

Younger, B. A. and Cohen, L. B. (1985). How infants form categories. *Psychology of Learning and Motivation - Advances in Research and Theory*, 19:211–247.

A Additional Experimental Details

A.1 Experiment 1 Additional Details

Model Architectures. We provide additional details on the model architectures we evaluate:

- MobileNetV2: We use the MobileNetv2 model as described by Sandler et al. (2018) and implemented in the the PyTorch torchvision package[6]. The model consists of a convolutional block, followed by a series of bottleneck blocks (a particular sequence of convolution and linear layer defined by Sandler et al. (2018)), followed by a final standard convolutional layer and an average pooling operator. We extract our embeddings from the output of this average pooling.

- ResNeXt: We use the ResNeXt-50 (32x4d) model as described by Xie et al. (2017), and implemented in the PyTorch torchvision package[7]. The model consists of five convolutional blocks (Figure 4), with a max pooling operation after the first block, and global average pooling after the last block. We omit the 1000-d fully-connected layer used for ImageNet classification and extract the embedding from the input to this layer.

- ViT-B/14: We use the ViT-Base architecture with a 14x14 patch size, termed ViT-B/14 by the authors who introduced the model (Dosovitskiy et al., 2021). We use an implementation by one of the authors based on the code in huggingface's pytorch-image-models repository[8]. The model tokenizes an input image by extracting fixed-size patches (in our case, 14x14 pixels) embedding them using a linear layer, and adding position embeddings. The sequence of tokens is then processed in a sequence of Transformer encoder blocks (Vaswani et al., 2017). We extract our embeddings from the output of the last Transformer block.

Stimulus Generation. We explore the following rendering approaches for our stimuli:

- Quinn-like: Most similar to Quinn et al. (1996), we render the reference object as a sequence of squares and the target object as one of the symbols used in that paper (a triangle, ‘s’, ‘E’, +, and →), all colored black (Figure 2).

- Geometric shapes: we render the reference as an elongated ellipse and the target as either a square, a circle, or a triangle, all colored black (Figure B.1).

- Random colors: again we render the reference as an elongated ellipse and the target as a circle, sampling perceptually distinct colors for both using the glasbey method (Glasbey et al., 2007; Kovesi, 2015, Figure B.2; bottom row).
  The latter deviates most from the original formulation but allows programmatically sampling a larger variety of stimuli to verify result robustness.

We also experimented with slightly blurring the stimuli to make them less perceptually perfect; this did not substantially impact any results.

A.2 Experiment 2 Additional Details

Stimulus Rotation. To rotate stimuli, we follow a render-rotate-crop approach:

- Render: render the stimuli as in Experiment 1a, noting the randomly sampled centroid position of the stimulus.

- Rotate: we place the rendered stimulus in the center of a much larger canvas, and rotate the entire canvas by the appropriate angle. This step is necessary as rotating a stimulus around its centroid could cause some of its pixels to end up outside of its original 224x224 canvas.

- Crop: we crop the stimulus back to 224x224 such that its centroid in the newly cropped version is identical to its initial centroid before the rotation and crop.

[6] https://pytorch.org/hub/pytorch_vision_mobilenet_v2/

[7] https://pytorch.org/hub/pytorch_vision_resnext/

[8] https://github.com/huggingface/pytorch-image-models/blob/main/timm/models/vision_transformer.py

A.3 Experiment 3 Additional Details

Stimulus Generation. We begin generating each stimulus by placing a reference object (a table in the “above/below” scenes, and a storage rack in the “between” scenes) at the origin of a scene. To create the familiarization stimuli, we then place a target object in one relation to this reference object (e.g., below it, in the case of “above/below”). We then create two test stimuli, one with the target object in the same relation (below), and the other with the target object in the other relation (above). As in Experiment 1a, the target objects in the test stimuli are approximately equidistant from the target object in the habituation stimulus. We then randomly sample camera parameters (location, angle, focus height) to create visual variability from ranges deemed to create acceptable stimuli and render the scene from the perspective of the camera. See Figure B.13 for examples of the various objects and variations in scene rendering.

A.4 Experiment 4 Additional Details

Stimulus Generation. We generate stimuli using one of four containers: a wicker basket, a wooden basket, a short cardboard box, and a longer cardboard box. Our stimuli also use one of the same eight target objects used in Experiment 3: a beach ball, a chess knight, a Lego piece, a toy pineapple, a ping-pong paddle, a toy robot, a rubber duck, and a stuffed animal (see Figures B.14 to B.17 for examples). We begin generating each stimulus by sampling a location for the camera and its focus from a distribution of values providing minor variation in the rendered stimuli. For each sampled set of camera parameters, we render a set of stimuli for each container and target object.

Linear Decoding Datasets. In each partition of the dataset, we assign to the test set all images from stimuli that use either the held-out container, or the held-out target object, or one of the held-out camera configurations, which results in assigning 42.578% of stimuli to the test set. We repeat this procedure five times (using different random seeds) for each held-out container and target object, for a total of 160 unique partitions with which we evaluate every model. We further split each training set into a training set and a validation set, by randomly assigning all stimuli from 10% of camera configurations in the training set to the validation set.

B Appendix Figures

B.1 Experiment 1 Alternative Stimuli Rendering Visualizations

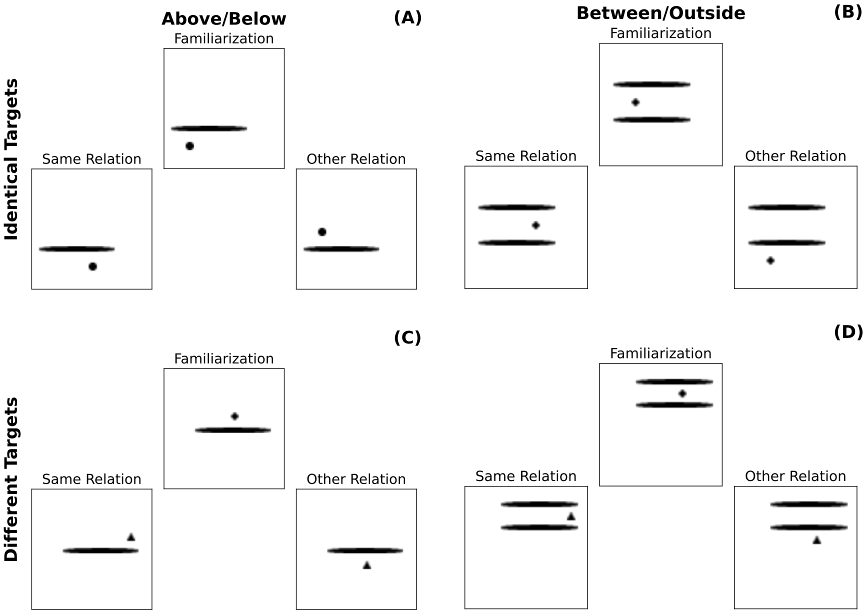

Figure B.1: Example stimuli rendered with geometric shapes. Identical to Figure 2, but with our geometric shape-based generator.

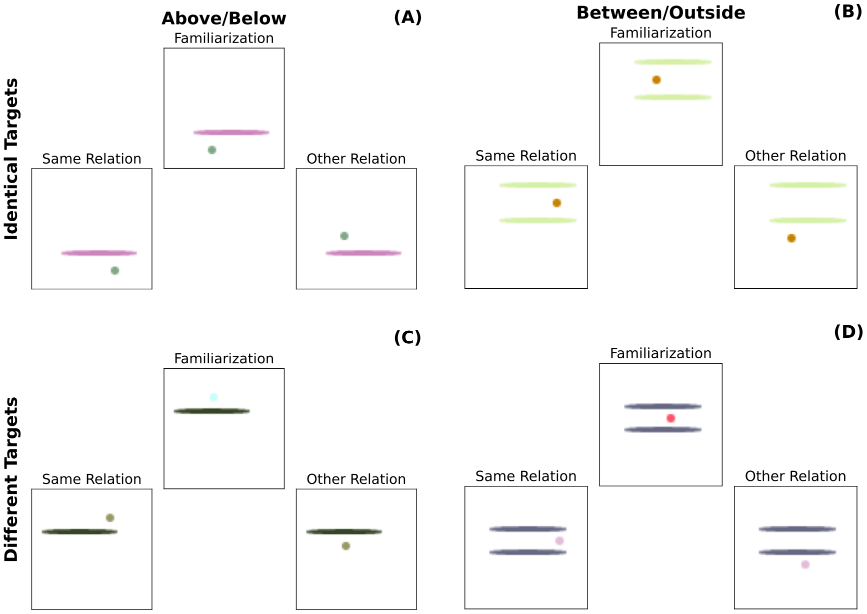

Figure B.2: Example stimuli rendered with random colors. Identical to Figure 2, but with our random color-based generator.

B.2 Additional Experiment 1 Results

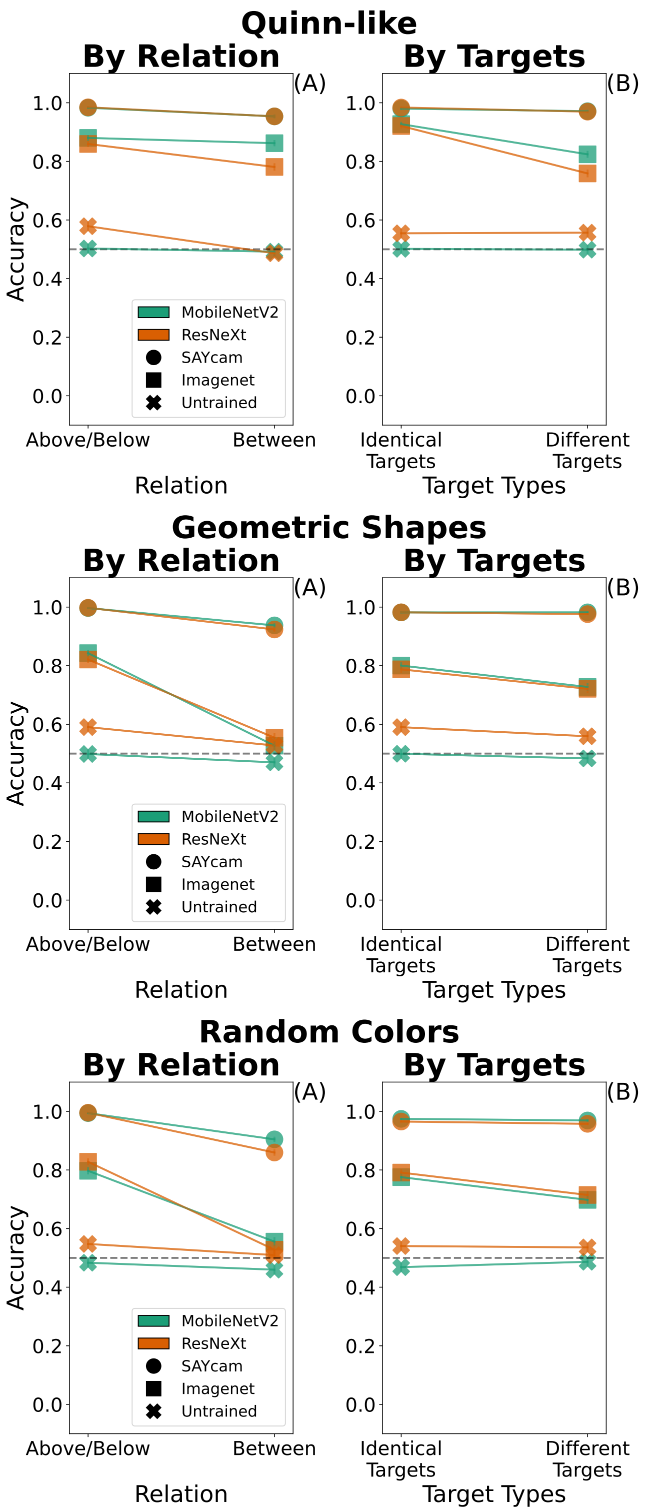

Figure B.3: Experiment 1a results by stimulus generation approach. Top: *Quinn-like* stimuli (Figure 2). Middle: *Geometric shapes* stimuli (Figure B.1). Bottom: *Random colors stimuli* (Figure B.2). Accuracy on the Quinn-like condition appears consistently highest, however the qualitative results replicate across all methods: *above/below* accuracy higher than *between*, same targets accuracy higher than different targets.

B.3 Experiment 1 control conditions

To verify the resilience of our results to experimental manipulations, we performed several additional experimental controls on the basic approach outlined in Experiment 1.

Above/below variations. To control for the relative visual complexity of the *between* stimuli compared to the *above/below* stimuli, we introduce two additional variations of *above/below* stimuli with additional reference objects (Figure B.4):
Adjacent references: we introduce a second reference object but place it adjacent to the original reference object. This variation maintains the same relational complexity as our original *above/below* stimuli, but the addition of a second reference object leads to more foreground pixels, akin to our *between* condition stimuli.
Gapped references: we introduce a second reference object and maintain a gap between reference objects, like in the *between* condition stimuli. This results in similar reference objects placements to the *between* scenes, but with the target objects placed either above or below both.

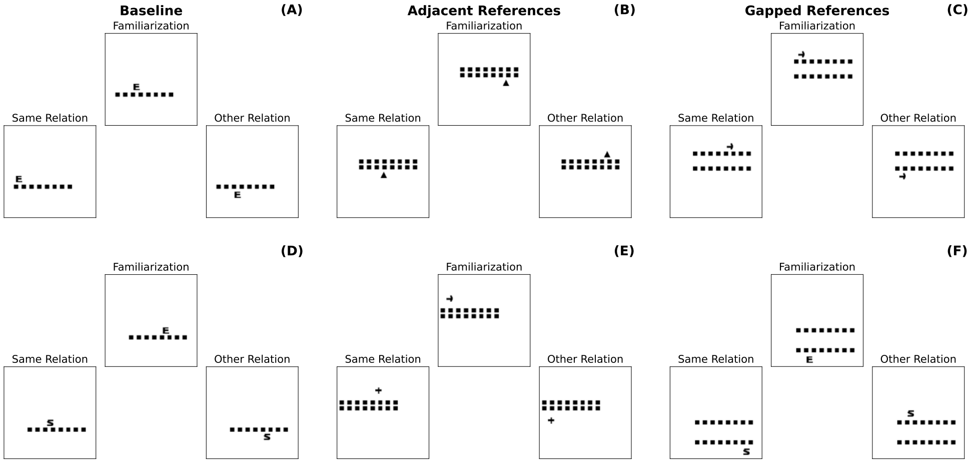

Figure B.4: Example stimuli rendered with above/below variations. Similar to Figure 2, but with the basic *above/below* stimuli and our two variations.

Results are remarkably consistent across the variants (Figure B.5), suggesting the increased difficulty of the *between* relation is in representing the relation itself, rather than the increased visual complexity of the scenes.

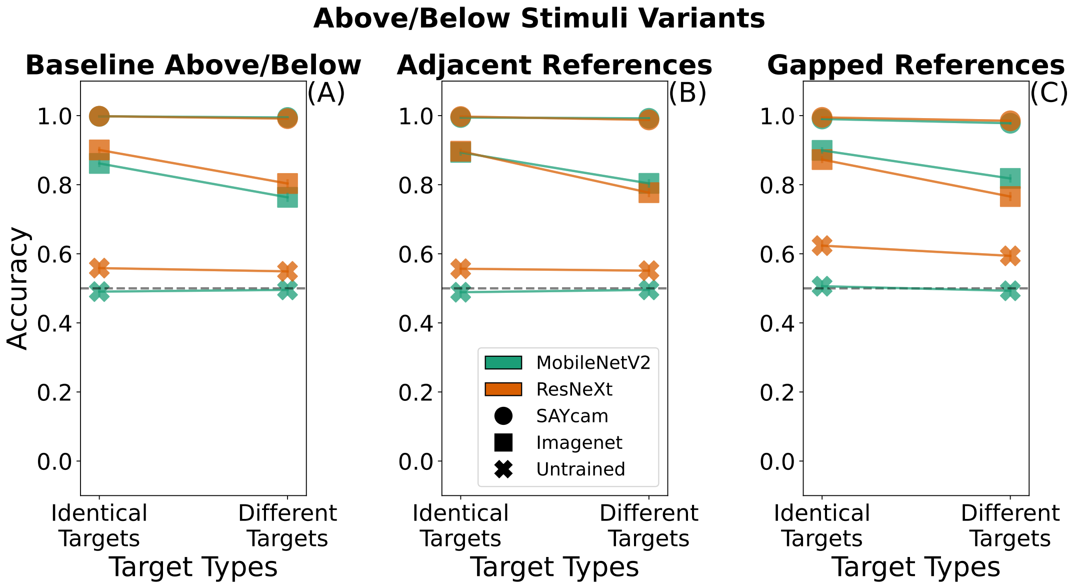

Figure B.5: Experiment 1 above/below variant results.

Using additional habituation stimuli. In their experiments, Quinn et al. present infants with four habituation stimuli before presenting them with a single test stimulus. We replicated a similar condition with our models. Instead of sampling a single habituation stimuli, we sampled four habituation stimuli whose target objects were placed in a small radius around a habituation centroid location. The target objects in the test stimuli were placed equidistant to this habituation centroid. To evaluate models, we extracted vector embeddings for each of the habituation stimuli independently, and averaged them to create an overall habituation embedding, which we again compared to the test stimuli embedding using the cosine similarity metric.

In Figure B.6, we change the hatching to reflect the use of one or four habituation stimuli. The results are again remarkably consistent, with some models showing a slight improvement with additional habituation examples, and other models showing a slight degradation — but broadly performance is unaffected by this manipulation.

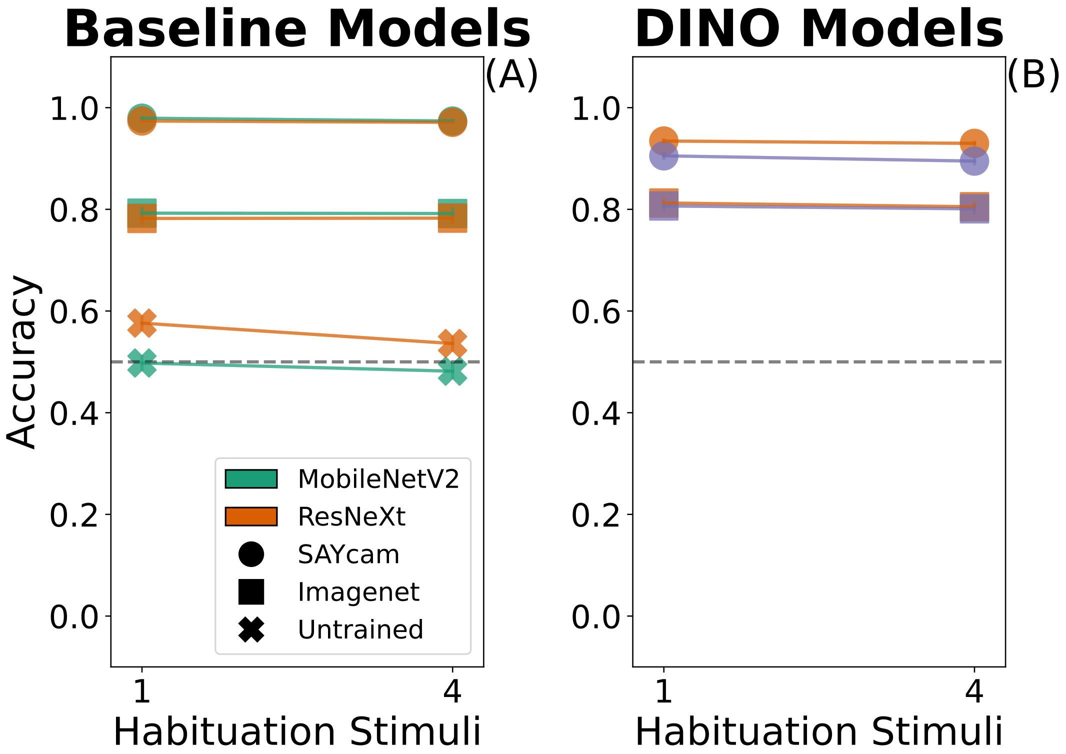

Figure B.6: Experiment 1a/b with 4 habituation stimuli. Left two panels: baseline results from experiment 1a (compare to Figure 5. Right two panels: DINO models results from experiment 1b (compare to Figure 8.

B.4 Experiment 1a additional t-SNE visualizations

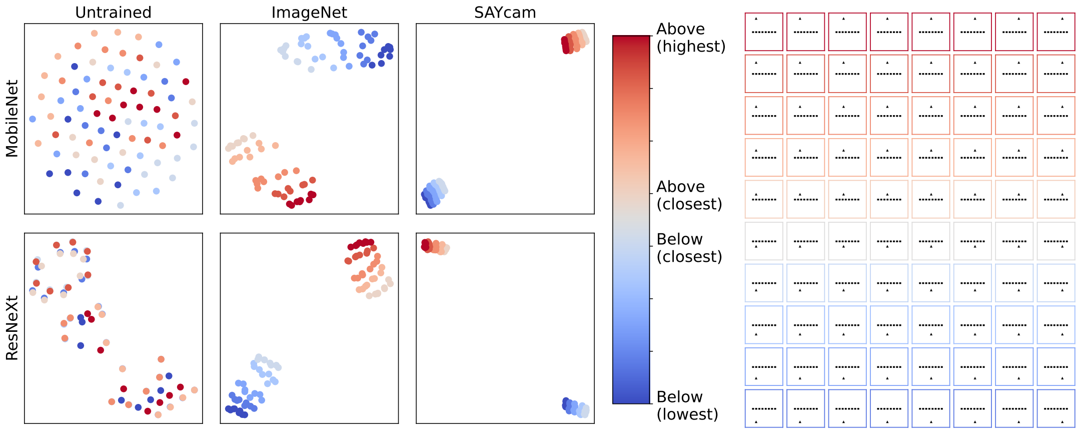

Figure B.7: Replication of Figure 6. Using our geometric shape-based generator instead of our Quinn-like generator reported in Figure 6.

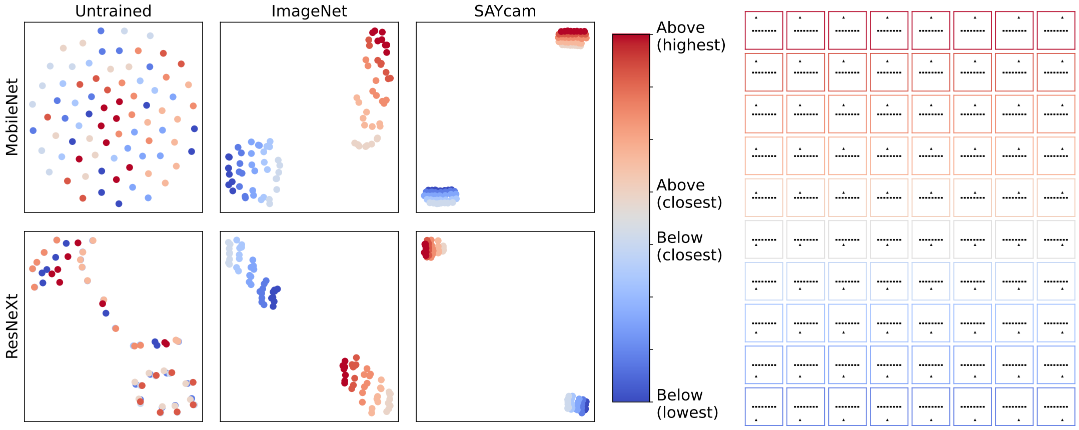

Figure B.8: Replication of Figure 6. Using our random color-based generator instead of our Quinn-like generator reported in Figure 6.

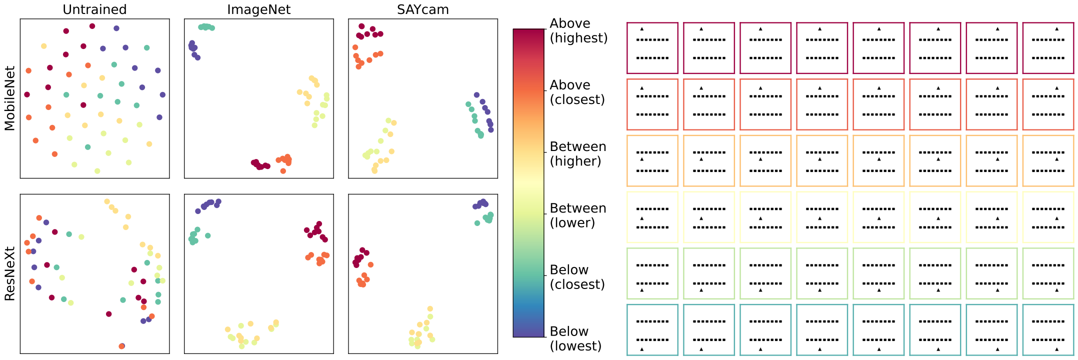

Figure B.9: Replication of Figure 7. Using our geometric shape-based generator instead of our Quinn-like generator reported in Figure 7.

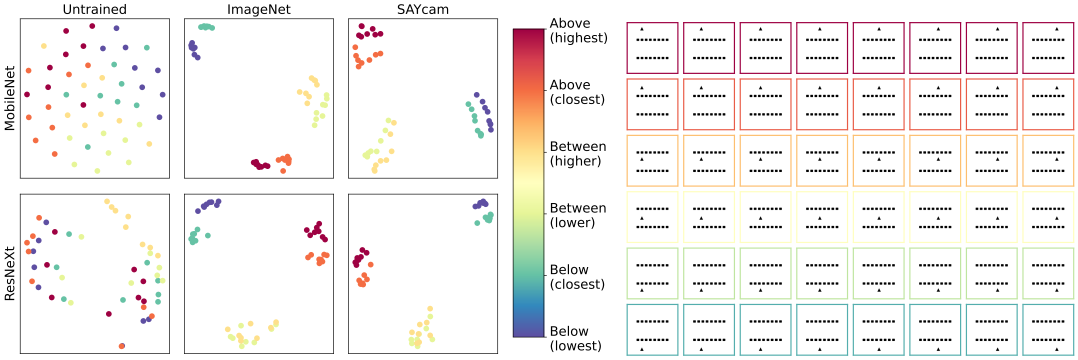

Figure B.10: Replication of Figure 7. Using our random color-based generator instead of our Quinn-like generator reported in Figure 7.

B.5 Experiment 2b rotated and flipped stimuli visualizations

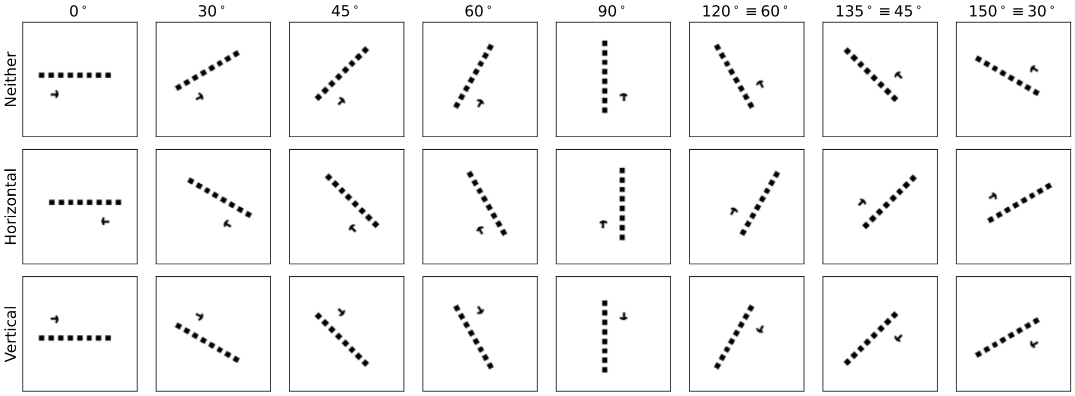

Figure B.11: Rotated and flipped stimuli with one reference. A set of stimuli with one reference object, matching our "above/below" (0°) and "left/right" (90°). Columns represent different rotation angles, noting the equivalency for stimuli rotated more than 90° from the horizontal. Rows represent the different symmetries learned by the various flipping models. The model with horizontal flipping would be trained to find the stimulus in the second row similar to the stimulus in the first row. Similarly, the model with vertical flipping would be trained to find the stimulus in the third row similar to the stimulus in the first row.

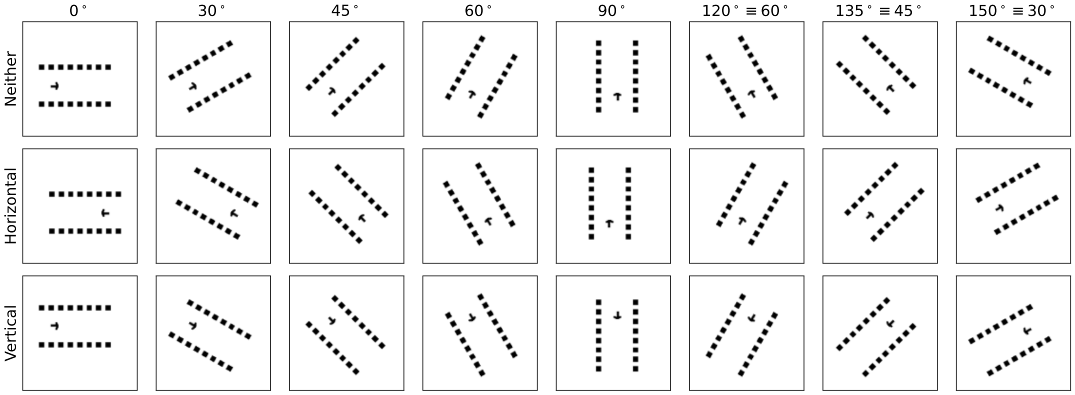

Figure B.12: Rotated and flipped stimuli with two references. A set of stimuli with two reference objects, matching our "between" (0°) and "sideways between" (90°). Columns represent different rotation angles, noting the equivalency for stimuli rotated more than 90° from the horizontal. Rows represent the different symmetries learned by the various flipping models. The model with horizontal flipping would be trained to find the stimulus in the second row similar to the stimulus in the first row. Similarly, the model with vertical flipping would be trained to find the stimulus in the third row similar to the stimulus in the first row.

B.6 Experiment 3 Target Object Visualizations

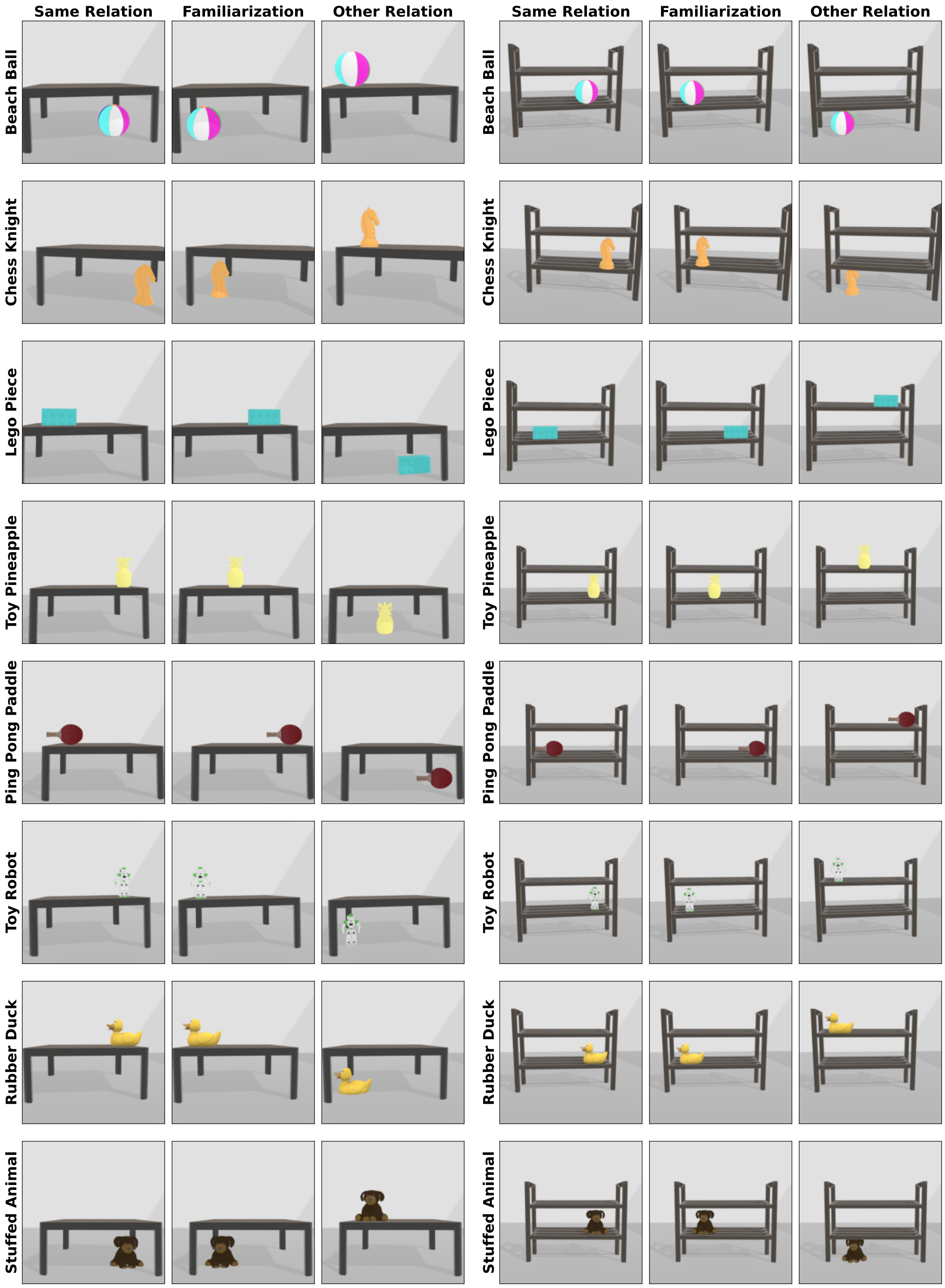

Figure B.13: All Target Objects used in Experiment 3. Left: example renderings in "above/below" scenes. Right: example renderings in "between" scenes. Each row represents a different target object, from top to bottom: a beach ball, a chess knight, a Lego piece, a toy pineapple, a ping-pong paddle, a toy robot, a rubber duck, and a stuffed animal.

B.7 Experiment 4 Stimuli Visualizations

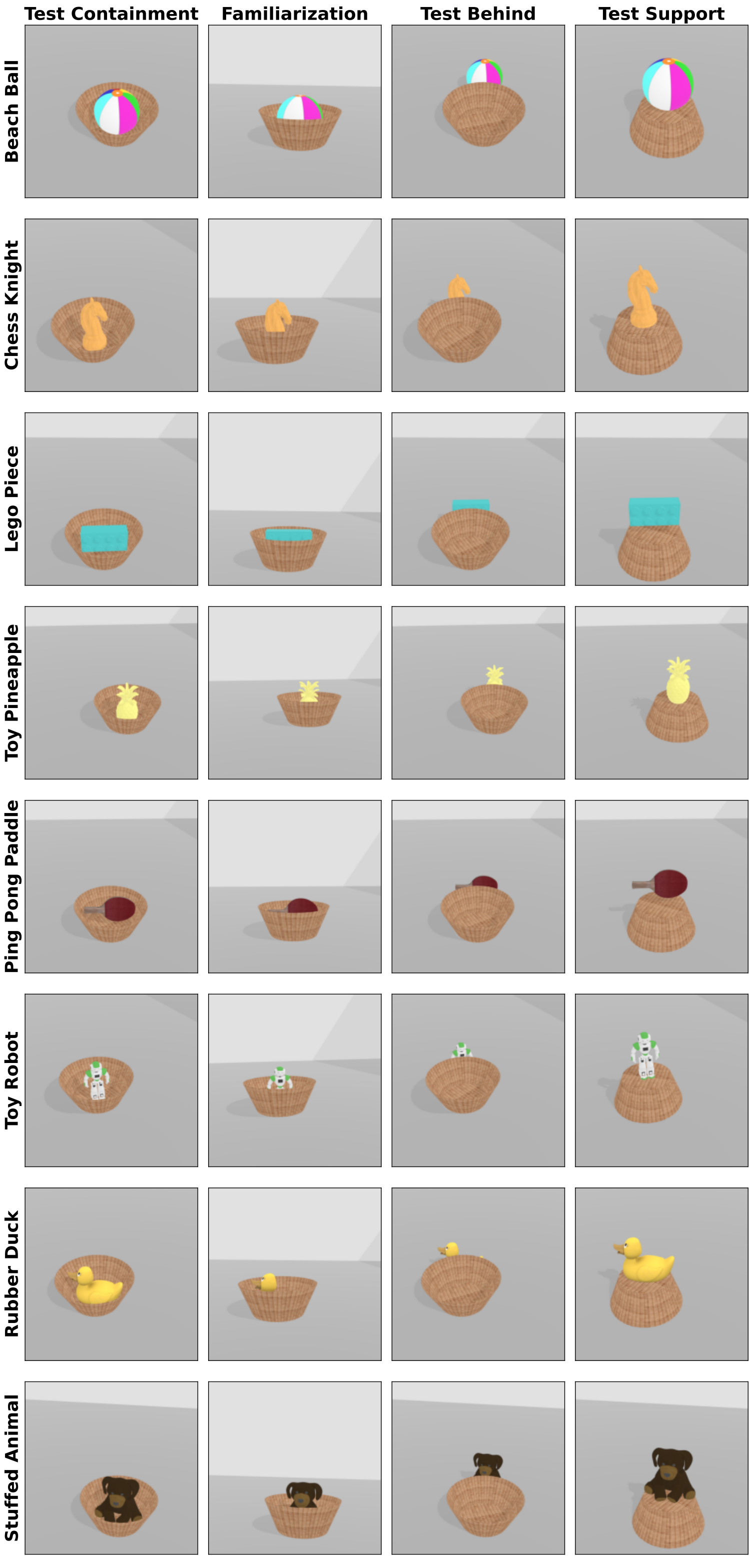

Figure B.14: Experiment 3 Example Stimuli With Wicker Basket container. Each row represents a different target object, from top to bottom: a beach ball, a chess knight, a Lego piece, a toy pineapple, a ping-pong paddle, a toy robot, a rubber duck, and a stuffed animal.

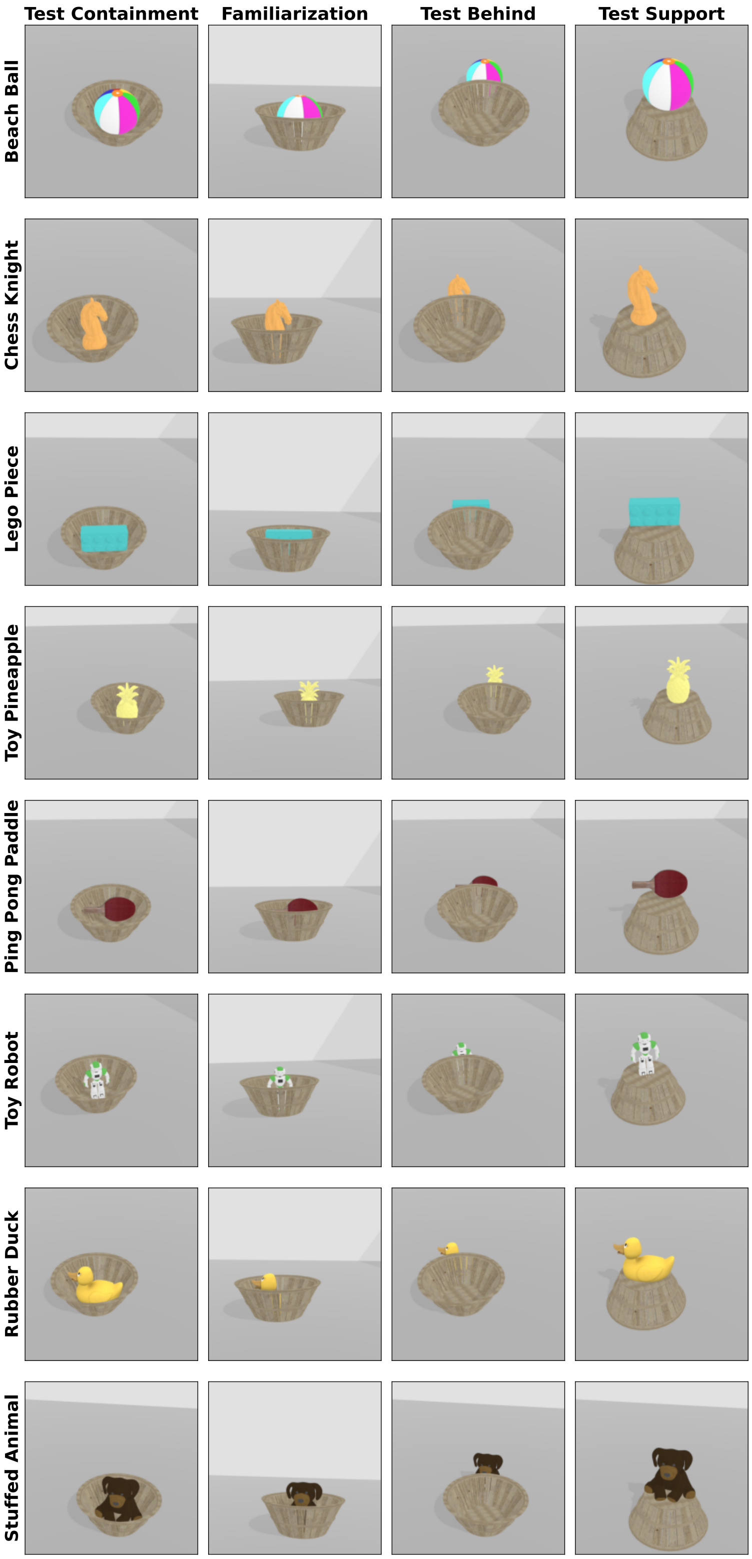

Figure B.15: Experiment 3 Example Stimuli With Wooden Basket container. Each row represents a different target object, from top to bottom: a beach ball, a chess knight, a Lego piece, a toy pineapple, a ping-pong paddle, a toy robot, a rubber duck, and a stuffed animal.

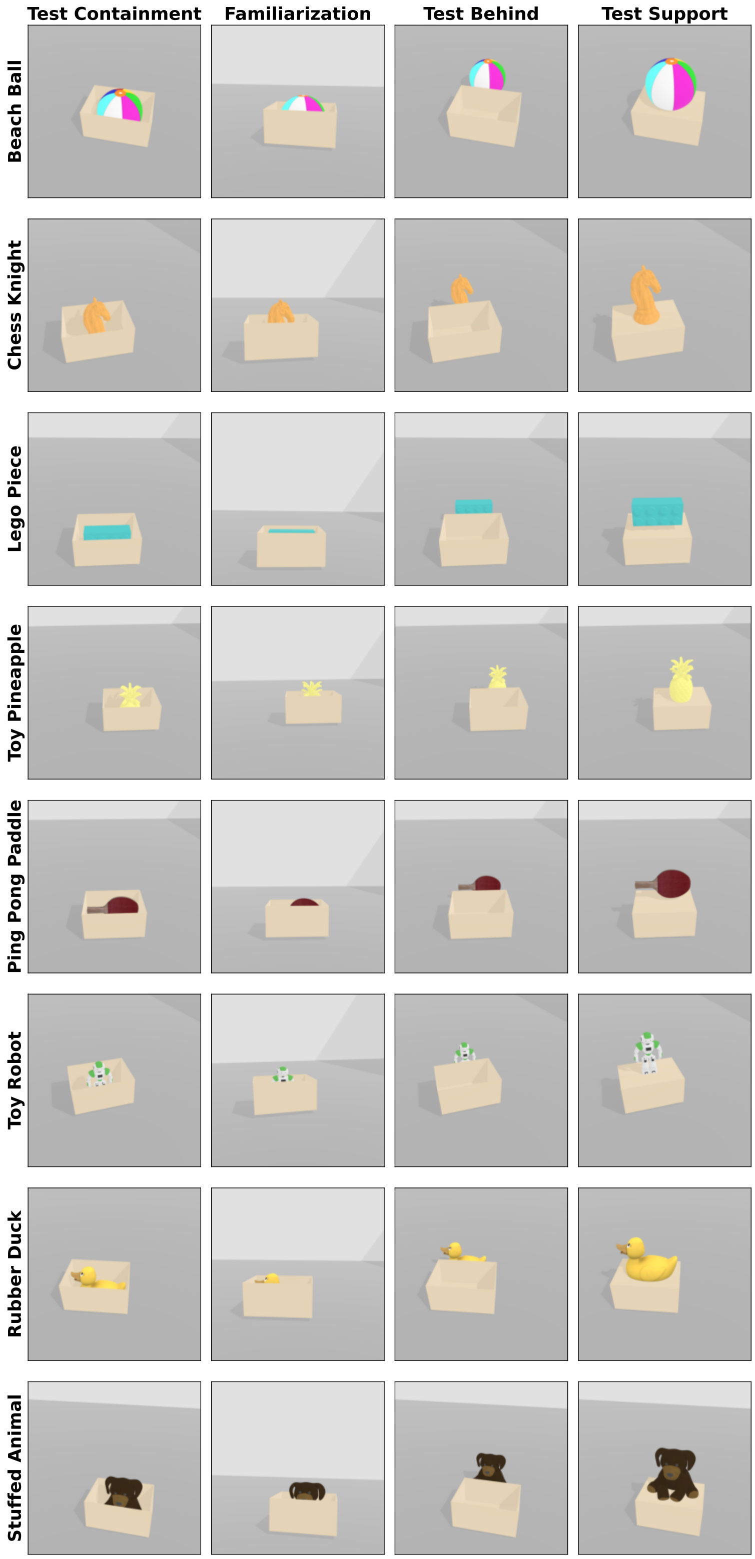

Figure B.16: Experiment 3 Example Stimuli With Shorter Box container. Each row represents a different target object, from top to bottom: a beach ball, a chess knight, a Lego piece, a toy pineapple, a ping-pong paddle, a toy robot, a rubber duck, and a stuffed animal.

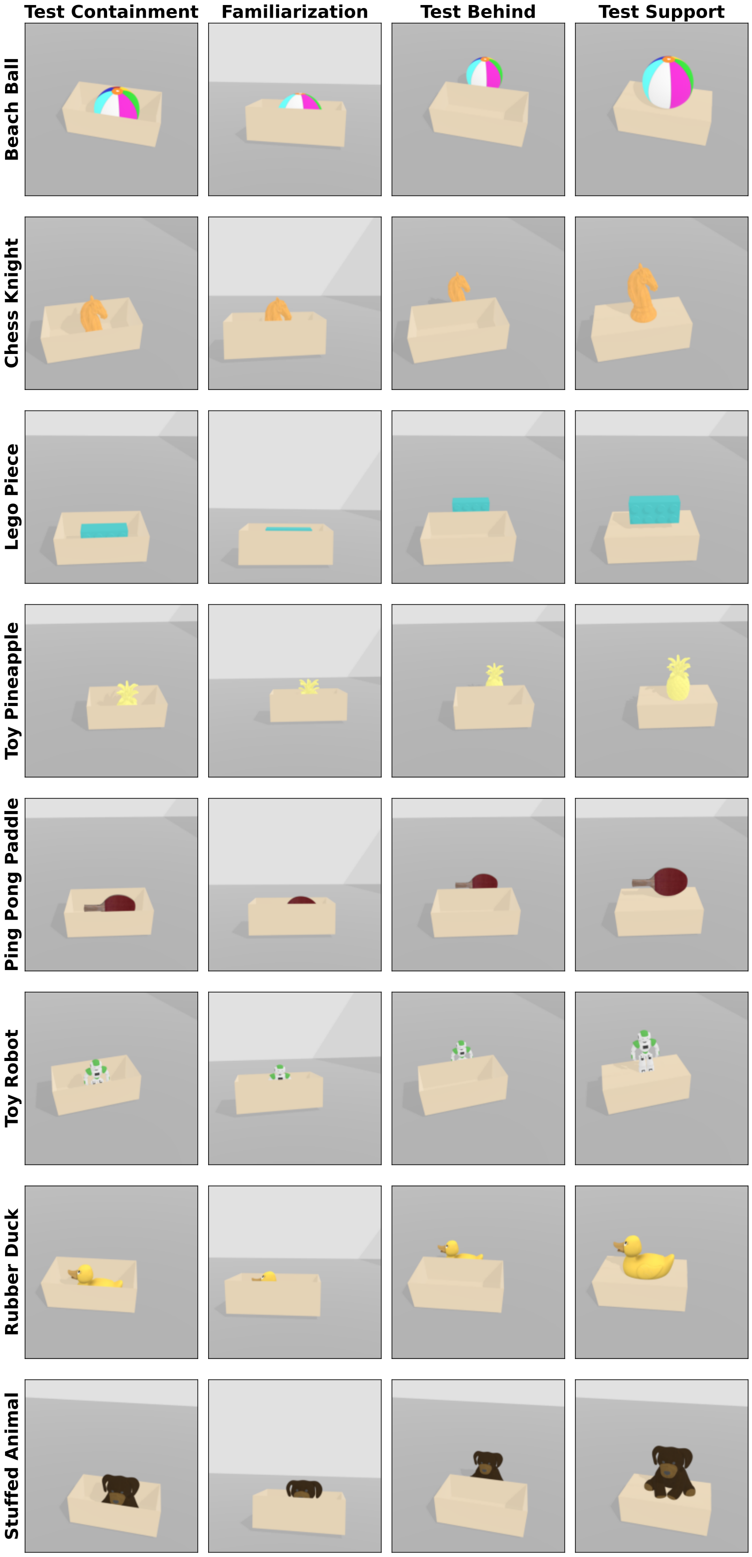

Figure B.17: Experiment 3 Example Stimuli With Longer Box container. Each row represents a different target object, from top to bottom: a beach ball, a chess knight, a Lego piece, a toy pineapple, a ping-pong paddle, a toy robot, a rubber duck, and a stuffed animal.

B.8 Experiment 4a Untrained Model Replications

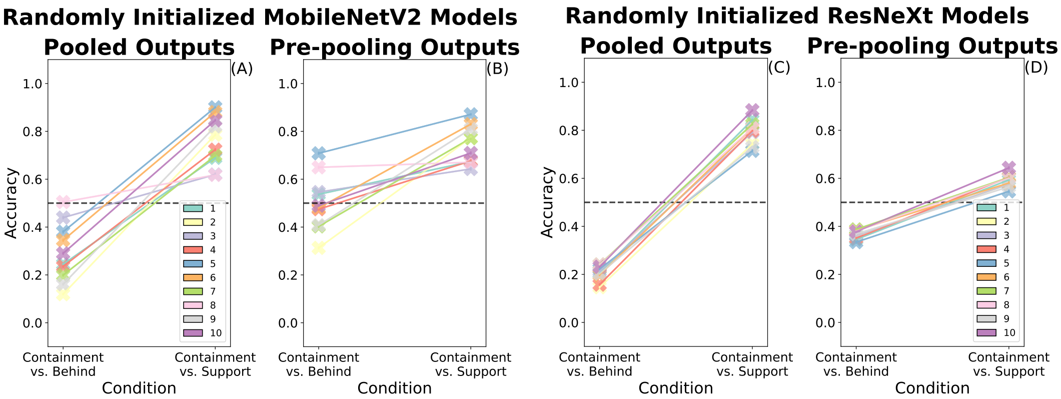

Figure B.18: Experiment 4a Untrained Model Replications. We replicate the results with randomly initialized and untrained models across ten random seeds, whose average performance is reported in Figure 16. (A) and (B): MobileNetV2 models. (A) comparing embedding similarity after the pooling operation. (B) comparing embedding similarity before the pooling operation. (C) and (D): ResNeXt models. (C) comparing embedding similarity after the pooling operation. (D) comparing embedding similarity before the pooling operation. Across all random seeds, we find that the randomly initialized models find the *behind* test probes more similar to the familiarization examples than the *containment* test probes.

B.9 Experiment 4a Pre-Pooling results

We hypothesized that one potential culprit in the models' failure in the "Containment vs. Behind" condition might be the pooling operations that precede our embedding extraction. In the MobileNetV2 and ResNeXt architectures (but not in the ViT-B/14 one), the final two-dimensional representation of each input image is pooled in order to create an embedding vector. This last two-dimensional representation is a tensor T of shape $C \times H \times W$, where C is the number of channels, H is the height, and W is the width. Both methods apply average pooling: the entry for the cth channel in the embedding vector v is defined by $v_c = \frac{1}{H}\frac{1}{W}\sum_{h=1}^{H}\sum_{w=1}^{W} T_{c,h,w}$, the average value of this channel across the input. The pooling mechanism struck us as potentially related to the failure as it collapses much of the spatial information, which might leave more remaining information in the degree of occlusion (which roughly corresponds to how many pixels of the target object are visible) than in the spatial relation. To examine this hypothesis, we repeat our similarity judgments, using embeddings extracted before the pooling operation (see Figure B.19). To extract these embeddings, we flatten the tensor T (with shape $C \times H \times W$) to a one-dimensional vector with CHW entries. We omit the ViT-B/14 model as it does not apply a pooling operation.

We visualize the results of this modification to our process in Figure B.20. Our models, including the randomly initialized and untrained ones, all reached substantially higher accuracies when similarity was compared using embeddings extracted before pooling (compare accuracies in Figure 16 to accuracies in Figure B.20).

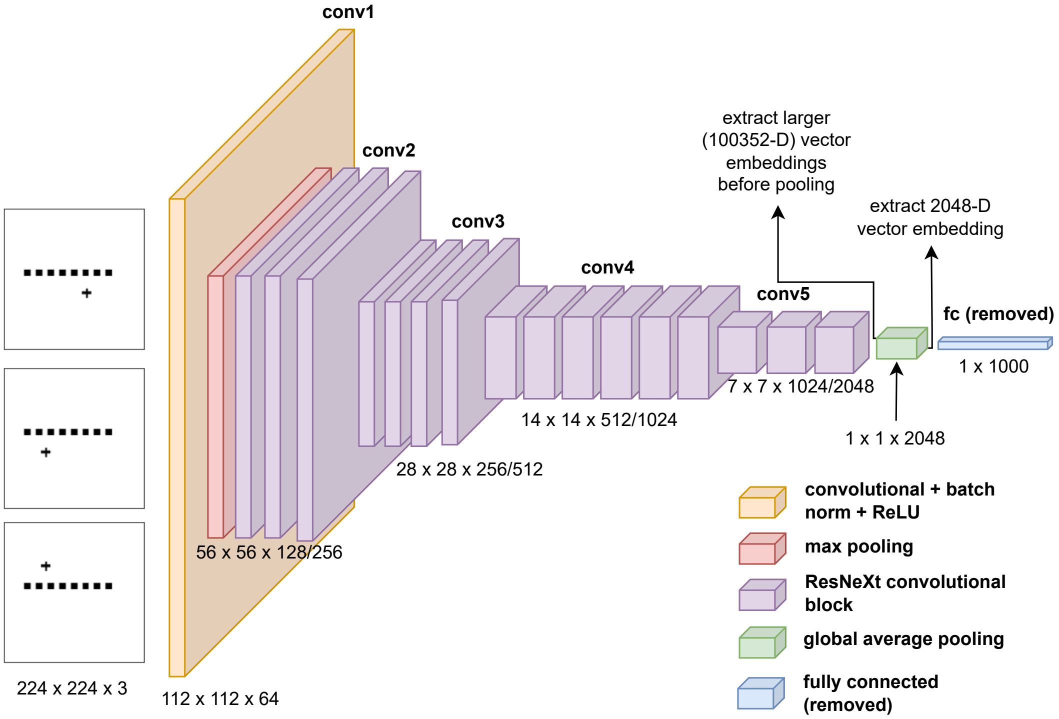

Figure B.19: ResNeXt model diagram marked for pre-pooling embeddings. Identical to Figure 4, but depicting where in the model we extract pre-pooling embeddings from.

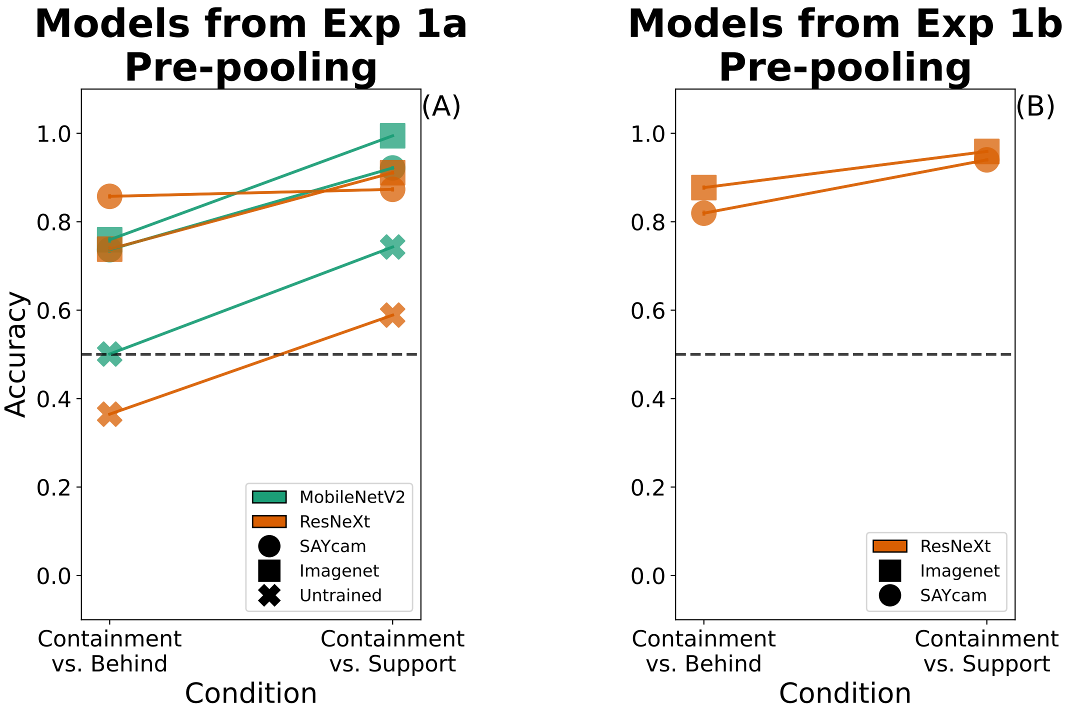

Figure B.20: Experiment 4a Pre-pooling. Identical to Figure 16, but with embeddings extracted prior to the pooling operation. (A): using the baseline set of models discussed in Experiment 1a. (B): using the DINO-trained models introduced in Experiment 1b. We omit the ViT-B/14 model as it does not use a pooling operation. Color: model architecture. Marker type: training method. The dashed line indicates chance accuracy (50%).

	Condition		Containment vs. Behind			Containment vs. Support			
Experiment	Training	Model	Pre-Pooling	Post-Pooling	Difference	Pre-Pooling	Post-Pooling	Difference	Mean Difference
1a	Untrained	MobileNetV2	0.501	0.292	0.209	0.743	0.758	-0.015	0.097
		ResNeXt	0.365	0.206	0.159	0.589	0.795	-0.206	-0.024
1a	ImageNet	MobileNetV2	0.759	0.423	0.336	0.994	0.923	0.071	0.204
		ResNeXt	0.738	0.524	0.214	0.911	0.851	0.060	0.137
1a	SAYCam(S)	MobileNetV2	0.737	0.309	0.427	0.922	0.665	0.256	0.342
		ResNeXt	0.857	0.366	0.491	0.873	0.491	0.382	0.437
1b	DINO-ImageNet	ResNeXt	0.877	0.484	0.393	0.959	0.770	0.189	0.291
1b	DINO-SAYCam(s)	ResNeXt	0.819	0.224	0.595	0.940	0.297	0.643	0.619
	Mean differences				**0.353**			**0.173**	**0.263**

Table 2: Summary of Pre-Pooling vs. Post-Pooling results. We report the mean levels of accuracy for each combination of training data, model architecture, and condition (either “Containment vs. Behind” or “Containment vs. Support”). We omit the ViT-B/14 models as they do not use a pooling operation. The right-most column reports the mean difference across both conditions for each model and training process. In both conditions, we find that the model reaches much higher accuracy when using the pre-pooling embeddings, particularly in the “Containment vs. Behind” condition. That is, when using embeddings that precede the pooling operation, the models represent the familiarization scene depicting the containment relation more similarly to the test scene depicting the same relation than it does to the test probes depicting other relations.

B.10 Experiment 4b Additional details and results

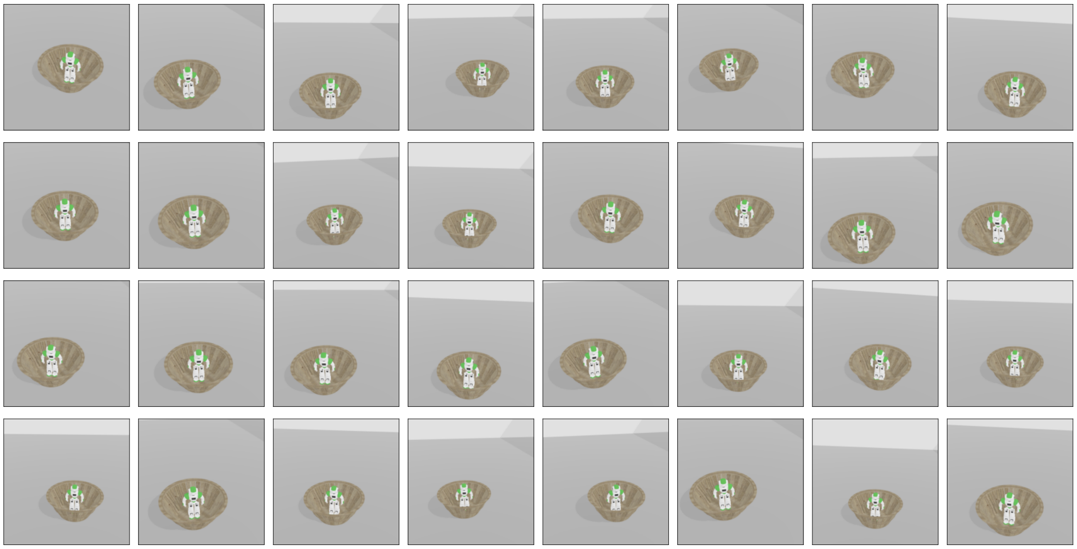

Figure B.21: Stimuli variations for one target and reference object. This figure highlights the variability in our dataset induced by the changes in camera location and angle between stimuli. We visualize 32 of the 128 scenes for the Toy Robot and Wooden Basket. In the linear decoding experiments, 16 (1/8th) of the scenes would be assigned to the held-out test-sets across all object combinations.

Container	Beach Ball	Chess Knight	Lego Piece	Toy Pineapple
Wicker Basket	0.929 ± 0.002	0.939 ± 0.002	0.938 ± 0.002	0.938 ± 0.002
Wooden Basket	0.952 ± 0.001	0.974 ± 0.001	0.940 ± 0.002	0.965 ± 0.001
Shorter Box	0.981 ± 0.001	0.990 ± 0.001	0.966 ± 0.001	0.991 ± 0.001
Longer Box	0.978 ± 0.001	0.990 ± 0.001	0.966 ± 0.001	0.988 ± 0.001
Container	Ping-Pong Paddle	Toy Robot	Rubber Duck	Stuffed Animal
Wicker Basket	0.948 ± 0.002	0.929 ± 0.002	0.945 ± 0.002	0.927 ± 0.002
Wooden Basket	0.954 ± 0.001	0.951 ± 0.002	0.964 ± 0.001	0.953 ± 0.001
Shorter Box	0.979 ± 0.001	0.974 ± 0.001	0.991 ± 0.001	0.972 ± 0.001
Longer Box	0.973 ± 0.001	0.972 ± 0.001	0.992 ± 0.001	0.984 ± 0.001

Table 3: Linear decoding accuracy is high over fully held-out stimuli. We report accuracy over the hardest examples of each test-set partition. These are the examples that used both the held-out container and the held-out target object in each configuration (e.g. examples using the Toy Robot and Wooden Basket in the partition visualized in Figure 17). The accuracies in the table are using linear decoders from all trained models (omitting the randomly initialized and untrained ones). All margins reported represent the standard errors of the mean.

C Appendix Experiment: learning simple relations from symbolic stimuli

In this experiment, we revisit the first finding discussed in Experiments 1 and 3, that infants acquire the capacity to represent “above or below” (a target object relative to a single reference object) before they develop the ability to represent “between” (a target relative to two references). In two studies (Quinn, 1994; Quinn et al., 1996), 3-4 months old infants familiarized with stimuli depicting a single relation (either above or below) exhibit a looking-time preference to a stimulus showing the opposite relation, compared to a new stimulus showing the familiarized relation. Quinn et al. (1999) followed up on those experiments, using examples of a target object between two reference objects, using both horizontal and vertical reference objects.

3-4 months old infants did not display a preference towards test stimuli containing an object outside the references, but infants 6-7 months old did. The ability to reason relative to two reference objects develops after the ability to reason relative to a single reference, consistent with the notion that infants first encode with respect to a single landmark, and later encode in a "local spatial framework" (Huttenlocher and Newcombe, 1984). Experiments 1 and 2 evaluated the extent to which models pretrained on broad visual experience represent such relations. In this experiment, we train small models on symbolic (rather than image-rendered) versions of this question, and evaluate how well (in accuracy) and how quickly (in the number of training steps required to reach an accuracy criterion).

Methods

Our simulations in this experiment evaluate the relative ease of learning two different classes of relations, both cast as binary classification problems: *above/below* (learning to classify above vs. below), and *between* (learning to classify between versus outside).

Objects. To model relation learning independently from learning to represent objects, we provide the models with minimal object representations as inputs. Each object is represented as a vector of length 4, with integer x and y positions, and a one-hot encoding marking the object as target or reference (Figure C.1 bottom). The objects are implicitly understood to be occupying a 1x1 unit square. The reference objects, which we take to be 9 units long, are represented as a collection of 9 adjacent identically-sized objects. We also explored an alternative representation that treats the reference bar as a single object, where each object vector has an additional integer dimension specifying its length (as all objects we use have a height of 1 unit, we omit a height dimension). Results with the alternative representation were qualitatively similar, even though the task is easier (as the models receive fewer object vectors as their input), so we focus on describing the results with the first representation (without the length dimension).

Dataset Generation. Figure C.1 visualizes stimuli from the different relation categories. To create stimuli, we sample locations for the reference object (series of blue cells) and then sample the target object's location uniformly from the 'target grid' above and below the reference object(s). In the *above/below* condition, we split the eight rows of the target grid evenly between above and below. In the *between* condition, we split the locations evenly between the between and outside relations. We only consider cases where the target object occupies the same horizontal space as the reference object(s), avoiding having the target object off to the side. To create training and test sets, we randomly split the reference object locations (in the large canvas, 90% training, 10% test) and the target object locations (relative to the reference object, 80% training, 20% test). We then set aside 10% of the training set as a validation set. This process creates a maximal training set of 3628 examples, a validation set of 404 examples, and a test set of 1800 examples. We also evaluate models trained on randomly sampled subsets of the training sets, using 8, 32, 128, 512, 1024, or 2048 items.

Architectures. We evaluate five different neural networks, each incorporating a distinct inductive bias. To the extent possible, the architectures were chosen to gracefully handle varying numbers of objects present in a scene. Other than the convolutional neural network, all models begin with an object-wise embedding function, a single layer with ReLU activations. We denote the input collection of objects as $O = \{o_1, ..., o_N\}$, the embedding function as e_ω, and the embedded objects as $E = e_\omega(O) : \{e_i = e_\omega(o_i)\}$. All models have two softmax output units (the two classes learned), and are trained using the cross-entropy loss to maximize the probability of the correct class.

'Bag of objects' MLP: this architecture is the simplest we could conceive of that would be invariant to the number of objects present in a stimulus. It treats the embedded vectors as a single vector by taking their mean and passes it into a standard feedforward network with ReLU activations. Denoting the MLP as f_ϕ:

$$MLP(O) = f_\phi\Big(\frac{1}{N}\sum_{i=1}^{N} e_\omega(o_i)\Big)$$

Convolutional Neural Network (CNN): this model encodes a translation invariance bias, receiving the objects as a 2D grid S rather than as an unordered list of vectors. As the objects' positions are represented by their placement in the grid, we use two channels in the spatial input, one marking the target object's location and another marking the locations of all reference objects. We use a standard convolutional architecture (conv) followed by global

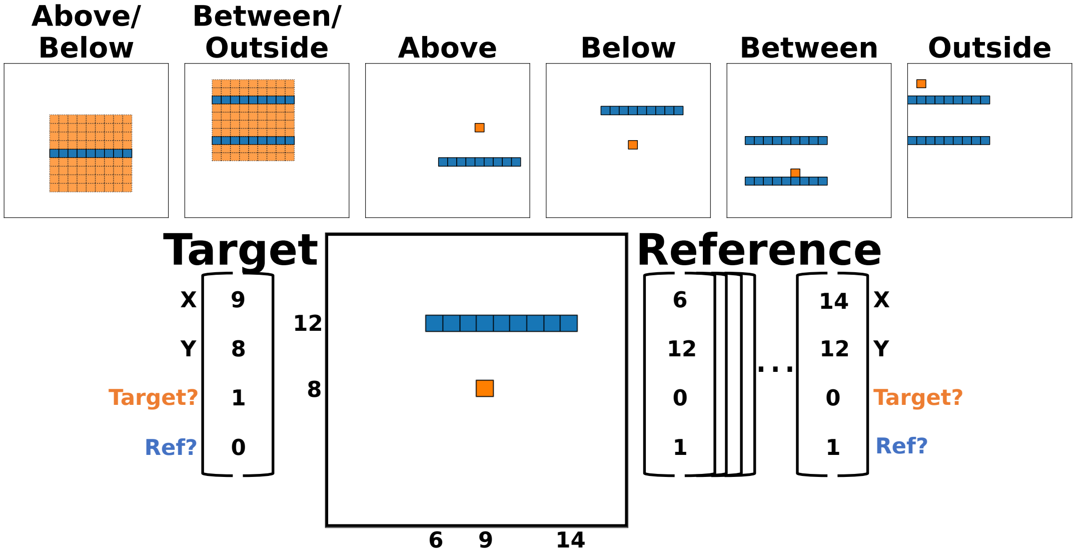

Figure C.1: Experiment 4 Stimuli . Top: Left two panels: a sample location of the reference object(s) (in blue), with the entire grid of possible target object locations visualized (in orange). Middle two panels: example *above/below* stimuli. Right two panels: example *between* stimuli. **Bottom:** the vector object representations associated with the Below example—as the models receive only these vectors, the choice colors and shapes here is arbitrary. We do not mark which coordinate is X and which is Y, so the models are agnostic to this fact (and above/below is identical to left/right), other than the CNN model, which receives a spatial input. The borders signify each object vector, so the blue reference object is comprised of nine vectors.

average pooling and an MLP:

$$CNN(S) = f_\phi(\text{average\_pool}(\text{conv}(S)))$$

Relation Net: Santoro et al. (2017) offer a compact way of modeling relations between pairs of objects, using two functions: a function g_θ that acts on object pairs and a global MLP f_ϕ acting on their combined representation:

$$RN(O) = f_\phi\left(\sum_{i=1}^{N}\sum_{j=1}^{N} g_\theta\left(e_\omega(o_i), e_\omega(o_j)\right)\right)$$

Transformer: a simplification of the Transformer (see Vaswani et al. (2017) for details), this network reasons about all objects jointly rather than through object pairs. The self-attention ('SelfAttn' below) operator acts on the entire set of objects simultaneously to capture their interactions. We pass the input through one or more such Encoders, and the transformed representations are averaged and passed through a MLP:

$$\text{Encoder}(E) = E + \text{SelfAttn}(E) + f_\phi\left(E + \text{SelfAttn}(E)\right)$$

$$T(O) = f_\phi\Big(\frac{1}{N}\sum_{i=1}^{N}\overbrace{\text{Encoder}(e_\omega(O))}^{\text{One or more times}}\Big)$$

PrediNet: this model is explicitly designed to learn different relations between objects, making for a task-optimized comparison architecture. It uses a modified form of self-attention, combining global information over the entire set of objects with information from each individual object, and treats the difference between object representations in a latent space as capturing different relations between them. See Shanahan et al. (2019) for the full details.

Implementation and Training. To test the effect of model size, we created two configurations of each model, a smaller one (using around 2000 parameters) and a larger one (using around 8000 parameters). We report results from ten random seeds for each simulation, varying three factors: relation (*above/below* or *between*), model size (smaller or larger models), and the number of training examples (8, 32, 128, 512, 1024, 2048, or the full size of the dataset created for each task, around 4000). We terminated each run when performance on a validation set plateaued. All models were optimized using Adam Kingma and Ba (2015) with a learning rate of 1e-3 and a batch size of 256. All models were implemented in PyTorch (Paszke et al., 2017) using PyTorch Lightning (Falcon, 2019).

C.1 Results

We focus our analysis on two measures of learning difficulty: *Sample complexity:* how many examples does it take to learn each concept? *Number of epochs:* how many passes through the training set does it take to learn each concept? We evaluate all five models on their ability to capture the developmental phenomena described above, including which architectures may be too powerful (learning both conditions trivially) or too weak (failing to learn either condition) when compared with competencies in infancy.

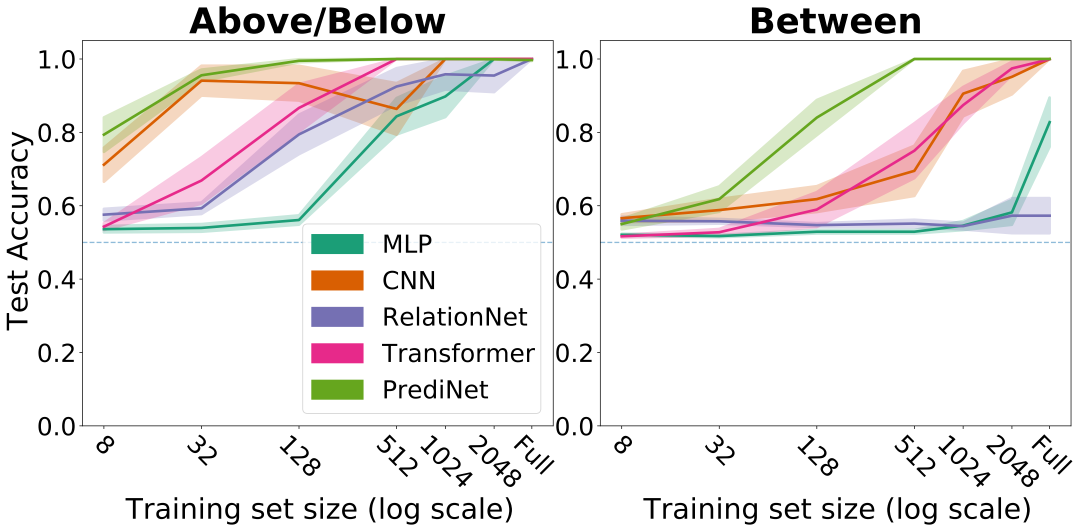

Figure C.2: Smaller model test set accuracy by training set size. Left: *above/below*. Right: *between*. Average over ten random seeds, shaded regions mark the SEM. Dashed line indicates chance (50%).

To evaluate the sample complexity, Figure C.2 depicts the test set accuracy attained by each architecture as a function of the size of the training set used, using the smaller (2000 parameter) configurations. We plot only the test set accuracies, as the networks generalize well above a reasonable sample size: the maximal difference between the training and test accuracy, averaged over the replications of each network, is 12.7% with 128 samples, 2.4% with 512 samples, and $< 0.1\%$ with the full training set. At all training set sizes, the networks perform better in the *above/below* condition than they do in the *between* condition, unless they fail to learn both. This is true from the most successful network (PrediNet) to the simplest (MLP) one, using both the smaller and larger network configurations. The RelationNet is the only network that fails to learn a relation, never reaching much above chance accuracy in the *between* condition; the MLP also struggles with *between*, rising above chance only with the full dataset. Results using the larger model configurations showed the same qualitative patterns.

To explore how long it takes the networks to acquire the concepts, Figure C.3 illustrates the learning curves using a 1024-item training set. Unsurprisingly, the models that reach a higher test accuracy (Figure C.2) also tend to require fewer training epochs to reach high performance. All architectures reach peak accuracy faster in the *above/below* condition than in the *between* condition. At this dataset size, both the RelationNet and the MLP networks fail to learn in the *between* condition.

C.2 Discussion

Most of the architectures examined are consistent with the basic developmental phenomenon: learning to spatially categorize above versus below is easier than between versus outside. This holds both when we take the sample complexity as a proxy for experience, and when we take the number of training epochs as the measure of experience. The RelationNet model struggled

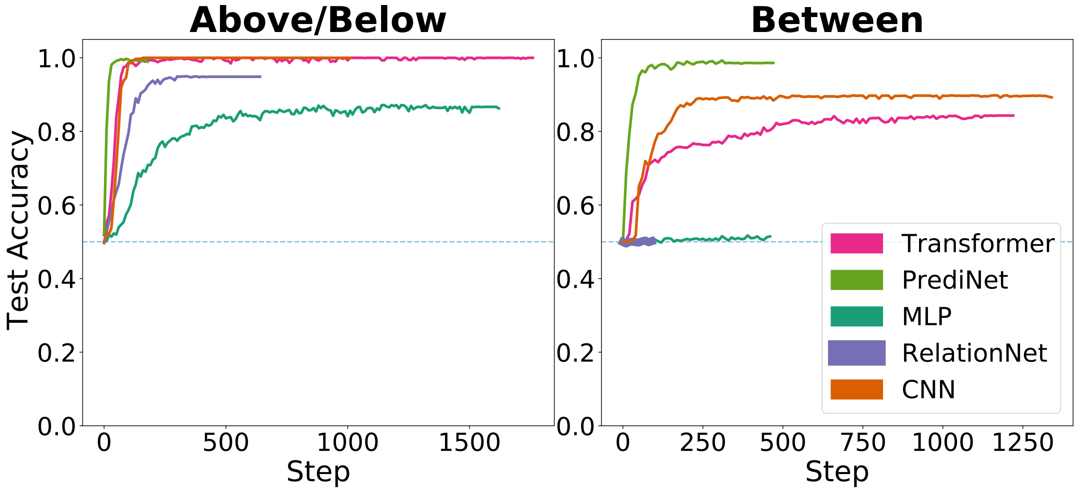

Figure C.3: Smaller model learning curves using 1024 training items. Left: *above/below*. Right: *between*. Average over ten random seeds for each model. Dashed line indicates chance accuracy (50%).

with learning the *between* relation, suggesting it may be an inadequate model of infant relation learning. In our alternative object formulation (see "Objects" in the Methodology subsection), which adds a length entry and hence reduces the number of input vectors, the RelationNet succeeded to learn this relation, performing closer to the Transformer. We attribute this failure to the fact that learning to reason using a pairwise function over the objects is harder to scale to higher numbers of entities. Models that natively reason over the entire collection struggle less with the *between* relation, which requires comparing three objects, the "local spatial framework" discussed by Huttenlocher and Newcombe (1984). The CNN and the Transformer both recover patterns qualitatively resembling the developmental findings, as does the PrediNet, even though it requires substantially less data than the other architectures to reach perfect accuracy. Conversely, the MLP might be overly generic, as it struggles with the *between* condition, only reaching above-chance performance with the full training set. We take these results to imply that any compelling computational model of infant reasoning should flexibly allow for variation in the number of objects reasoned over, being neither entirely generic (the MLP) nor restricted to pairwise interactions (the RelationNet). Beyond these constraints and considerations, the data does not help us distinguish the other architectures as potential cognitive models. The finding that learning above/below is easier than between/outside appears to be a fairly general property of the neural architectures we evaluated.