Pretraining on Interactions for Learning Grounded Affordance Representations

Jack Merullo
jack_merullo@brown.edu

Dylan Ebert
dylan_ebert@brown.edu

Carsten Eickhoff
carsten@brown.edu

Ellie Pavlick
ellie_pavlick@brown.edu

Abstract

Lexical semantics and cognitive science point to *affordances* (i.e. the actions that objects support) as critical for understanding and representing nouns and verbs. However, study of these semantic features has not yet been integrated with the "foundation" models that currently dominate language representation research. We hypothesize that predictive modeling of object state over time will result in representations that encode object affordance information "for free". We train a neural network to predict objects' trajectories in a simulated interaction and show that our network's latent representations differentiate between both observed and unobserved affordances. We find that models trained using 3D simulations from our SPATIAL dataset outperform conventional 2D computer vision models trained on a similar task, and, on initial inspection, that differences between concepts correspond to expected features (e.g., *roll* entails *rotation*). Our results suggest a way in which modern deep learning approaches to grounded language learning can be integrated with traditional formal semantic notions of lexical representations.

1 Introduction

Much of natural language semantics concerns events and their participants–i.e., verbs and the nouns with which they compose. Evidence from cognitive science (Borghi and Riggio, 2009; Mazzuca et al., 2021) and neuroscience (Sakreida et al., 2013) suggests that grounding such words in perception is an essential part of linguistic processing, in particular suggesting that humans represent nouns in terms of their *affordances* (Gibson, 1977), i.e., the interactions which they support. Affordance-based representations have been argued to form the basis of formal accounts of compositional syntax and semantics (Steedman, 2002). As such, prior work in formal semantics has sought to build grounded lexical semantic representations in terms of objects and their interactions in 3D space. For example, Pustejovsky and Krishnaswamy (2014) and Siskind (2001) represent verbs like *roll* as a set of entailed positional and rotational changes specified in formal logic, and Pustejovsky and Krishnaswamy (2018) argue that nouns imply (latent) events–e.g., that *cups* generally *hold* things–which should be encoded as TELIC values within the noun's formal structure.

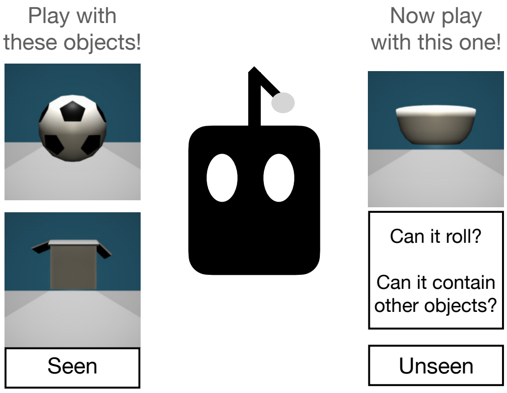


Figure 1: We investigate whether observing interactions with an object in a 3D environment encodes information about their affordances and whether this generalizes in the zero shot setting to unseen object types

Such work provides a compelling story of grounded semantics, but has not yet been connected to the types of large scale neural network models that currently dominate NLP. Thus, in this work, we ask whether such semantic representations emerge naturally as a consequence of self-supervised predictive modeling. Our motivation stems from the success of predictive language modeling at encoding syntactic structure. That is, if neural language models trained to predict text sequences learn to encode desirable grammatical structures (Kim and Smolensky, 2021; Tenney et al., 2018), perhaps similar models trained to predict event sequences will learn to encode desirable semantic structures. To test this intuition, we investigate whether a transformer (Vaswani et al., 2017) trained to predict the

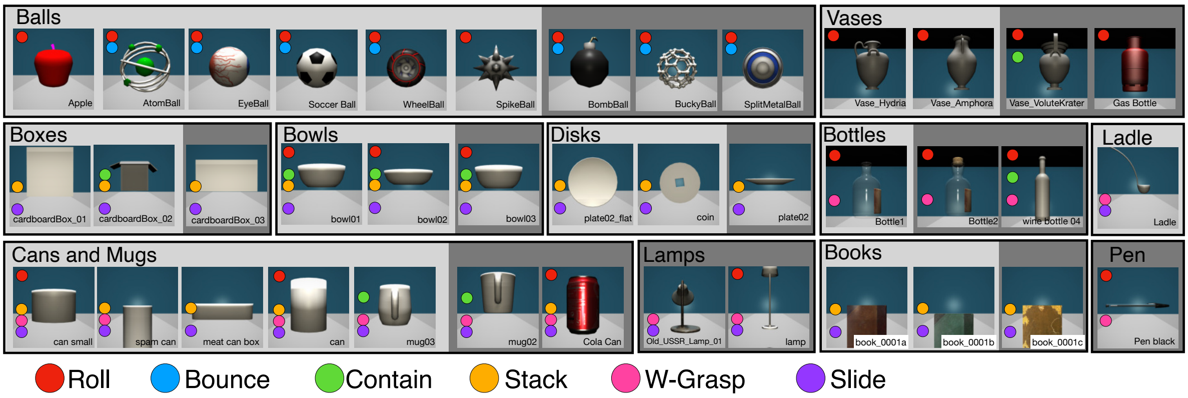


Figure 2: Objects uses for train (light gray) and test (dark gray). Colored dots indicate which affordances each object has.

future state of an object given perceptual information about its appearance and interactions will latently encode affordance information of the type thought to be central to lexical semantic representations. In sum:

- We present a first proof-of-concept neural model that learns to encode the concept of affordance without any explicit supervision.
- We demonstrate empirically that 3D spatial representations (simulations) substantially outperform 2D pixel representations in learning the desired semantic features.
- We release the `SPATIAL` dataset of 9.5K simulated object interactions and accompanying videos, and an additional 200K simulations without videos to support further research in this area.[1]

Overall, our findings suggest a process by which grounded lexical representations–of the type discussed by Pustejovsky and Krishnaswamy (2014) and Siskind (2001)–could potentially arise organically. That is, grounded interactions and observations, without explicit language supervision, can give rise to the types of conceptual representations to which nouns and verbs are assumed to ground. We interpret this as corroborative of traditional feature-based lexical semantic analyses and as a promising mechanism of which modern "foundation" model (Bommasani et al., 2021) approaches to language and concept learning can take advantage.

2 Experimental Setup

2.1 Objects and Affordances in `SPATIAL`

To collect a set of affordances to use in our study, we begin with lists of affordances and associated objects that have been compiled by previous work on affordance learning: Aroca-Ouellette et al. (2021) provides on a small list of concrete actions for evaluating physical reasoning in large language models; Myers et al. (2015) provides a small list for training computer vision models to recognize which parts of objects afford certain actions; Chao et al. (2015) use crowdworkers[2] to judge noun-verb pairs and includes over 900 verbs that are both abstract and concrete in nature. We then filter this list down to only a subset of concrete actions that include objects which exist in the Unity asset store, since we use Unity simulated environments to build our training and evaluation data (§2.4). This results in a list of six affordances (`roll`, `slide`, `stack`, `contain`, `wrap-grasp`, `bounce`) which are used to assign binary labels to each of 39 objects from 11 object categories (Figure 2; see also Appendix A).

2.2 Representation Learning

We hypothesize that predictive modeling of object state will result in implicit representations of affordance and event concepts, similarly to how predictive language modeling results in implicit representations of syntactic and semantic types. Thus, for representation learning, we use a sequence-modeling task defined in the visual and/or spatial world. Specifically, given a sequence of frames

[1] `https://github.com/jmerullo/affordances`

[2] In the case of Chao et al. (2015), we use a score ≥ 4 as positive label, as they do in their paper.

depicting an object's trajectory, our models' objective is to predict the next several timesteps of the object's trajectory as closely as possible. We consider several variants of this objective, primarily differing in how the represent they frames (e.g., as 2D visual vs. 3D spatial features). These models are described in detail in Section 3.

2.3 Evaluation Task

We are interested in evaluating which variants of the above representation learning objective result in readily-accessible representations of affordance and event concepts. To do this, we train probing classifiers (Belinkov and Glass, 2019) on top of the latent representations that result from the representation learning phase. That is, we freeze the weights of our pretrained models and feed the intermediate representation for a given input from the encoder into a single linear-layer trained to classify whether the observed object has the affordance. We train a separate classifier probe for each affordance.

We construct train and test splits by holding out a fraction of the objects from each category. In some cases, the held-out objects are very similar to what has been seen in training (e.g., slightly different dimensions of boxes) and in other cases, the objects are visually very distinct (e.g., a wine bottle vs. a gas tank as instances of objects which afford both `roll` and `contain`). Figure 2 shows our objects, affordances, and train-test splits.

2.4 `SPATIAL` Environment and Data Collection

The `SPATIAL` dataset consists of simulations of interactions with a variety of 3D objects in the Unity game engine[3]. Our data is collected in a flat empty room using the Unity physics engine on the above-described 39 objects. For each sequence, an object is instantiated at rest on the ground. A random impulse force–either a 'push' flat along the ground, or a 'throw' into the air–is exerted on the object. We only exert a single impulse on an object per sequence. The sequence ends when the object stops moving or after 4 seconds elapse.

We record the coordinates of the object in 3D space at a rate of 60 frames per second. Specifically, each sequence is defined by the coordinates describing the object's 3D position in space $P = \{p_1, ..., p_t\}$ for t timesteps. Since we care about capturing the manner in which the object travels and rotates through space, p_i contains 9 distinct 3D points around the object: 8 corners around an imaginary bounding box and the center point of that bounding box (see Appendix A for a visual aid). Simultaneously we collect videos of each interaction from a camera looking down at a 60 degree angle towards the object that we will use to train our 2D vision based model. Each image in the videos is collected at a resolution of 384x216 pixels. We filter videos where the object leaves the frame. Overall, this process results in 2,376 training sequences and 9,283 evaluation sequences. Due to computational constraints, we decided to focus on collecting as many evaluation examples as possible to make comparison to spatial models easier and more accurate. It may be the case that adding more data creates stronger representations, but even with this smaller training set, we see high test time performance on the visual dynamics task. All our data are publicly available at `https://github.com/jmerullo/affordances`.

2.5 Assumptions and Limitations

This work serves as initial investigation of our hypothesis about representation learning for affordances (§2.2). We use simple simulations which involve only a single object. Thus, we expect that our setup makes some affordances (`roll, slide, bounce`) more readily available than others (`contain, stack, w-grasp`). For example, our models likely will observe objects rolling during pretraining, but will never observe objects being stacked on top of one another. However, during evaluation, we will assess how well the model's internal representations encode both types of affordances. This is intended. Our hypothesis is that, to a large extent, these affordances are a function of the relationship between the shape[4] of the objects and the physics of how those objects behave in our simulation. For example, we expect that long, thin `grasp`-able objects will display different trajectories than will wide, round objects

[3] `https://unity.com/`

[4] In fact, Gibson (1977)'s original theory of affordances defined them to be purely-perceptual, without even depending on internal processing and representation. We do not endorse this view in general; we are enthusiastic about future work which involves richer internal processing (e.g., interaction and planning) during pretraining. See Şahin et al. (2007) for a review of the various definitions and interpretations of the term that have been used in different fields and Mota (2021) for an argument that affordances are not solely perceptual. That said, this basic-perception approach is helpful starting point for understanding the relationship between pretraining and affordance representations.

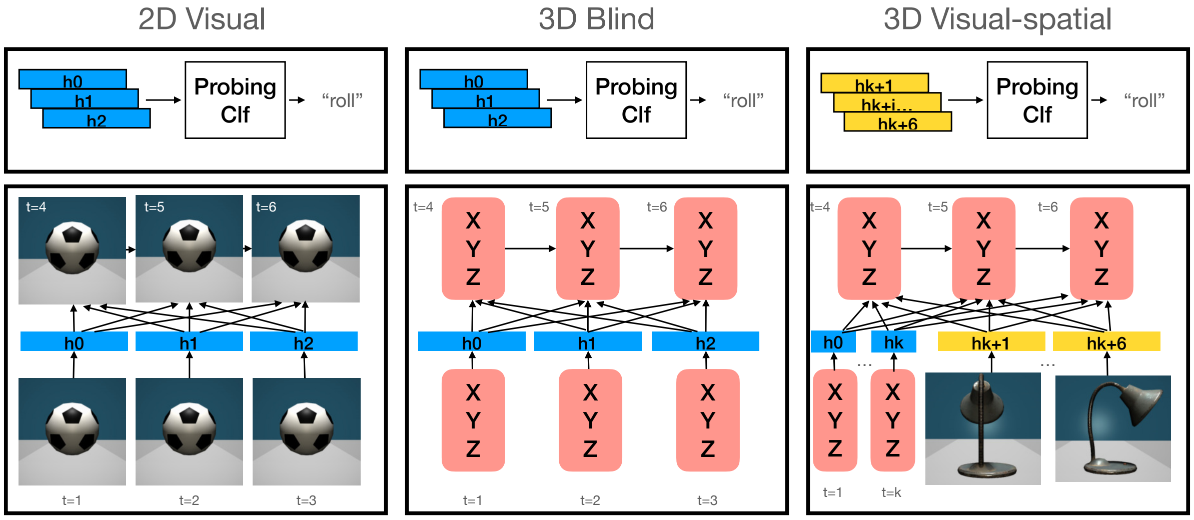


Figure 3: Model architectures. The model receives either images, 3D coordinates or both to make predictions. For the 3D models, the transformer encoder encodes the input sequence (and multiview images, if applicable) and the decoder predicts the rest of the sequence. For the 2D model (RPIN), a convolutional network extracts object-centric features (h_i) and interaction reasoning is performed over each to predict the next time steps.

that cannot be grasped. Thus, we expect that a model trained to predict object trajectories can encode differences that map onto affordances such as `grasp` or `contain`, even without observing those actions *per se*. Given initial promising results (§4), we are excited about future work which extends the simulation to include richer multi-object and agent-object interactions, which likely would enable learning of more complex semantic concepts.

3 Models

We consider two primary variants of the representation learning task described in Section 2.2 which differ in how they represent the world state–i.e., using 2D visual data (§3.1) vs. using 3D visual-spatial data (§3.2). To provide additional insights into performance differences, we also consider two ablations in the 3D model (§3.2.2), one that removes visual information and one that further removes pretraining altogether. These models are summarized in Figure 3.

3.1 2D Visual Model

We first consider a standard computer-vision (CV) approach for our defined representation learning objective. For this, we use a Region Proposal Interaction Network (RPIN) proposed in Qi et al. (2021). We choose to use RPIN because it was designed to solve a task very similar to ours–i.e., object tracking over time–and has access to object representations via bounding boxes provided as supervision during training. Using a model with access to explicit object representations ensures that we are not unfairly handicapping the CV approach (by requiring it to learn the concept of objects from scratch) but rather are analyzing the relative benefits of a 2D CV approach vs. a 3D spatial data approach for latently encoding semantic event and affordance concepts.

We train the model with similar settings to those Qi et al. (2021) used to train on the Simulated Billiards dataset, but with some small differences. For example, we subsample our frames to be coarser-grained to encourage learning of longer-range dependencies. Exact details and explanations of other parameter differences can be found in Appendix B.

To probe object representations for affordance properties, we take the average of the hidden representations–i.e., the model's representations just prior to predicting explicit bounding box coordinates on the screen.

3.2 3D Visual-spatial Model

3.2.1 Full Model

Recent work has argued that models based directly on 3D game engine data are more cognitively plausible for modeling verb semantics (Ebert and Pavlick, 2020). In this spirit, we consider a model that learns to encode the objects visual appearance jointly with predicting the objects' behavior in 3D space. Specifically, our model is trained with both an object loss and a trajectory loss as follows.

To model the 3D trajectory, the model encodes a sequence P containing positions $\{p_1, p_2, ..., p_t\}$. As described in Section 2, each position p_i contains 9 distinct points corresponding to the object center and the 8 corners of the rectangular bounding box encapsulating the object. We use a single linear layer to project the 27D (9 3D points) input coordinate vectors to the embedding dimension of a transformer (Vaswani et al., 2017). The transformer is then fed the first $t - k$ timesteps where $k \geq 1$. We treat k as a hyperparameter, and find that a value of $k = 8$ or $k = 16$ tends to work the best. Our model is trained to minimize the Mean Squared Error (MSE) computed against the true future location of the object, summed over all of the predicted points.

To model the object appearance, we give the model access to a static view of the object at rest. We use ResNet-34 (He et al., 2016) to encode the object's *multiview*–i.e., images of the object's six faces, one from each side of the object–denoted as I, and pass these as additional inputs to the model, separated by a SEP token. The transformer encoder encodes the sequence P and I together, and the transformer decoder predicts the object's next several positions in space. To encourage the model to connect the sequence and image representations, we randomly (50% of the time) replace the object in I with an object with different affordances and add an auxiliary loss in which the model must classify whether the object was perturbed. We add a linear binary classification layer on top of a CLS token to perform this task, and add the cross entropy loss of this objective to our MSE loss for the trajectory objective.

The hidden representation we use for probing experiments is the average pooled transformer encoder output of the multiview tokens only.

3.2.2 Ablation Models

To better understand which aspects of the above model matter most, we also train and evaluate two ablated variants.

Without Visual Information (3D Blind). Our 3D Blind model is like the above, but contains no multiview tokens or associated loss. That is, the model is trained only on the 3D positional data, using an MSE loss to predict the future location of the object. For probing, we average the transformer encoder outputs across all timesteps and feed the single averaged emebedding into the probing classifier. This model provides insight into how well the physical behavior alone, with *no visual inputs*, encodes key features for determining affordances, such as shape and material.

Without Pretraining (No-Training). Gibson believed that understanding affordances only required raw perception, without need for mental processing. Given how saliently actions like rolling and sliding are encoded in 3D coordinates (Figure 6), it is reasonable to ask how much benefit our pretraining objective provides for encoding affordance information. To test this, we evaluate a model that is identical to the 3D Blind model, but contains only randomly initialized encoder weights (i.e., which are never set via pretraining). If the pretraining task encodes affordance structure the way we hypothesize, the randomly initialized model should perform much worse than the trained 3D Blind variant. We refer to this model simply as the No-Training model.

4 Results

Figure 4 shows our primary results. Overall, the 3D Visual-spatial model substantially outperforms the 2D Vision-only model across all affordances, often by a large margin (4–11 percentage points). We also see, perhaps unexpectedly, that the 3D pretrained representations encode information about affordances even when the associated actions are not explicitly observed. For example, the model differentiates objects that can `stack` and objects that can `contain` other objects from those that cannot, even though the model has not directly observed objects being stacked or serving as containers during training. This result points to the richness of the physical information that is required to perform the pretraining task of next-state prediction.[5]

Looking more closely at the ablated variants of the 3D model, we see that most of the gains are from the 3D input representation itself. That

[5] We note that, unintuitively, `stack` and `contain` probes generally outperform `slide` probes. One reason may be because our data are labeled by object rather than by individual interaction. For example, although an object *typically* slides, it's not hard to imagine scenarios where a cardboard box might roll over. This is not the case for affordances like `stack` and `contain`. In the rolling cardboard box example, the sharp edges of the box and the distinct way it rolls is still indicative of the object being stackable.

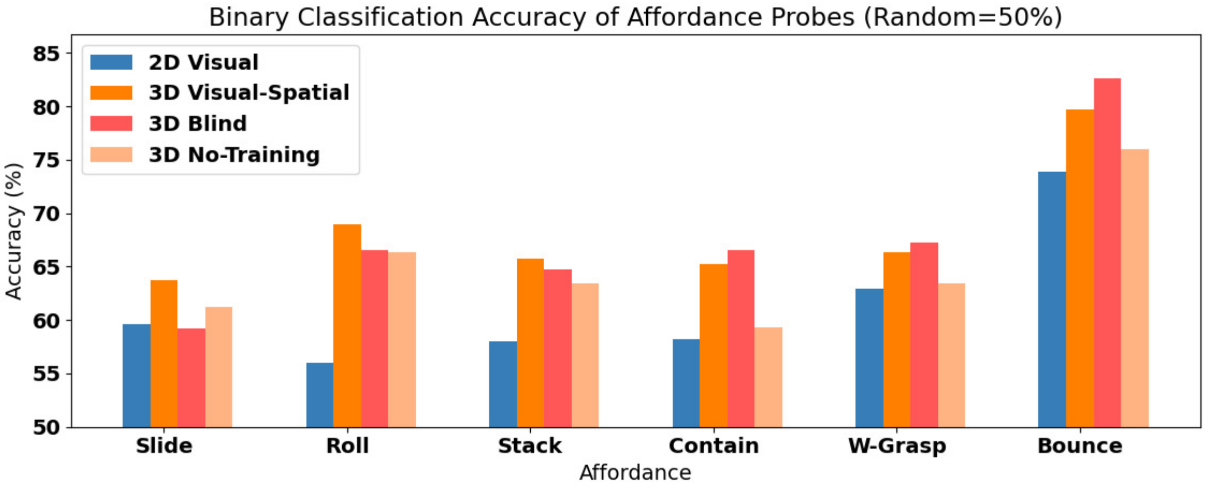
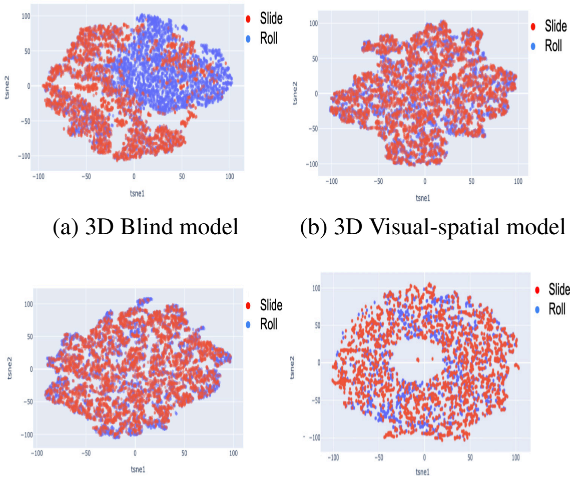


Figure 4: Results for predicting affordances of objects given frozen hidden states of 2D and 3D sequence prediction models. Test sets are balance so that random guess achieves 50%. 3D models (even ablated variants) outperform the 2D computer vision models across the board.



(a) 3D Blind model

(b) 3D Visual-spatial model

(c) 3D No-Training

(d) 2D Computer-Vision model (RPIN).

Figure 5: t-SNE projections of model representations of sliding (red) vs. rolling (blue) objects.

is, the 3D No-Training model–which does not include visual information and does not even include pretraining–outperforms the CV baseline in all cases, and often substantially. Pretraining on top of the 3D inputs often (but not always) yields performance gains. Pretraining with visual information does not provide a clear benefit over pretraining on the spatial data alone–i.e., visual information leads to performance gains on three affordances (`slide`, `roll`, and `stack`) and losses on the other three (`contain`, `w-grasp`, and `bounce`).

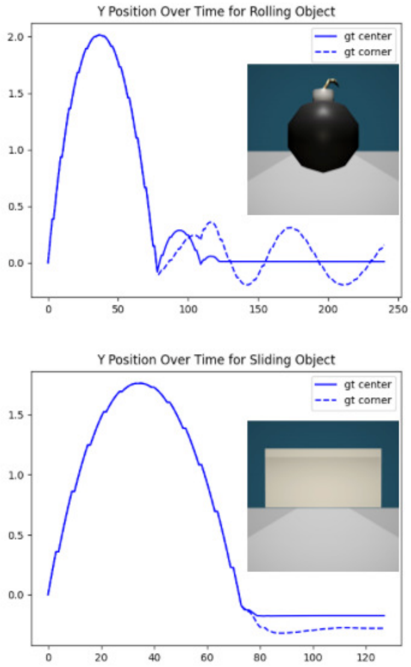


Figure 6: Visualization showing how 3D coordinate data clearly distinguishes a rolling object from a sliding one, making it easier for a model to learn the difference between the two.

5 Qualitative Analysis

In order to better understand the nature of the affordance-learning problem, we run a series of qualitative analyses on the trained models. We focus our analysis on the pair of affordances `roll` vs. `slide`. These are verbs have received significant attention in prior literature (Pustejovsky and Krishnaswamy, 2014; Levin, 1993) since they exemplify the types of manner distinctions that we would like lexical semantic representations to encode.

We first compare the 2D video vs. 3D simulation variants of our pretraining objective. Figure 5 shows a t-SNE projection of the sequence representations from all four models, labeled based on if the object affords rolling or sliding. We find that object representations from the 3D Blind model cluster strongly according to the distinction between these two concepts. The trend is notably not apparent in the No-Training model. Figure 6 demonstrates why spatial data pretraining may encourage this split. In the example shown, we take two thrown objects from our dataset–one round and one not round–and track the height of the center point of the object and one of the corners of the object bounding box. When they hit the ground the center point stays relatively constant as it moves across the floor in both, but in the rolling action, the corner point moves up and down as it rotates around the center point. Since this is so distinguishable given the input representation, the model is better able to differentiate these concepts.

It may be that the next state prediction task facilitates learning the slide vs. roll distinction in the 3D Blind setting. However, the same pattern is not present in the 3D Visual-spatial model (which also predicts next state). One possibility is that the presence of visual information competes with the 3D information, and as a result the joint space does not encode this distinction as well as the 3D space alone. Designing more sophisticated models that incorporate visual and spatial information and preserve the desirable features of both is an interesting area for future work.

5.1 Counterfactual Perturbations

An important aspect of lexical semantics is determining the *entailments* of a word–e.g., what about an observation allows it to be described truthfully as `roll`? Thus, in asking whether affordances are learned from next-state-prediction pretraining, it is important to understand not just whether the model can differentiate the concepts, but whether it differentiates them for the "right" reasons.

We investigate this using counterfactual perturbations of the inputs as a way of doing feature attribution, similar in spirit to prior work in NLP (Huang et al., 2020) and CV (Goyal et al., 2019). Specifically, we create a controlled dataset in which, for each of 10 interactions, we generate 20 minimal-pairs which differ from their originals by a single parameter of the underlying physics simulation. The parameters we perturb are {`mass, force velocity, starting x position, starting z position, shape, angular rotation`}. For example, given an instance of a lamp rolling across the floor, we would generate one minimally-paired example in which we only change the mass of the lamp, and another the same as the original except it does not exhibit any angular rotation, and so forth for each of the parameters of interest. More implementation details are given in Appendix C.

We use our pretrained `slide` probe to classify the representations from each sequence as either rolling or sliding, and compare the effect of each perturbation on the model's belief about the affordance label. Figure 7 shows the resulting belief changes for several of the perturbed parameters. We see that changing the angular rotation of an otherwise identical sequence has the greatest effect on whether an instance is deemed to afford `roll`ing. This is an encouraging result, as it aligns well with standard lexical semantic analyses: i.e., generally, `roll` is assumed to entail rotation in the direction of the translation.

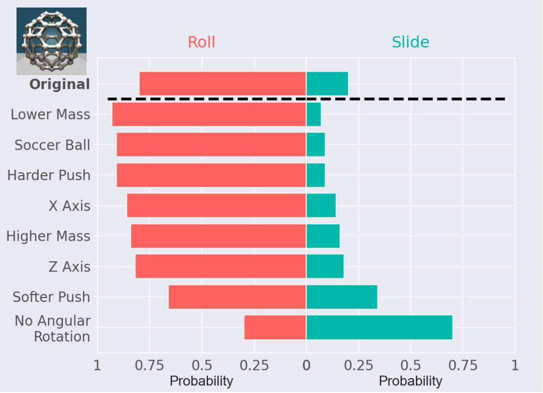


Figure 7: Change in predicted probability of the encoding of a round object affording `slide` after generating the interaction again with one feature changed (See Appendix C for a visualization)

However, our analysis also reveals that the models rely on some spurious features which, ideally, would not be part of the lexical semantic representation. For example, the 3D blind model is affected by the travel distance the object. If we increased the mass or decreased the force applied to a rolling object, such that it only moved a small distance or rotated a small number of times, the model was less inclined to label the instance as rolling; though this was usually not by enough to have the undesirable effect of flipping the prediction. Intuitively this makes sense given the model's training data: rolling objects tend to travel a greater distance than sliding objects. An interesting direction for future work is to investigate how changes in pretraining or data distribution influence which features are encoded as "entailments", i.e., key distinguishing features of a concept's representation.

6 Related Work

6.1 Lexical Semantics and Cognitive Science

In formal semantics, there has been significant support for the idea that motor trajectories and affordances should form the basis of lexical semantic representations (Pustejovsky and Krishnaswamy, 2014; Siskind, 2001; Steedman, 2002). Such work builds on the idea in cognitive science that simulation lies at the heart of cognitive and linguistic processing (Feldman, 2008; Bergen et al., 2007; Bergen, 2012). For example, Borghi and Riggio (2009) argue that language comprehension involves mental simulation resulting in a "motor prototype" which encodes stable affordances and affects processing speed for identifying objects. Cosentino et al. (2017) point to such simulation as a factor in determining surprisal of affordances depending on linguistic context. Similar arguments have been made based on evidence from fMRI data (Sakreida et al., 2013) as well as processing in patients with brain lesions (Taylor et al., 2017). It is worth noting that there is debate on the general nature of affordances in humans. For instance, Mota (2021) argues that affordances are not solely perceptual. We view our work as being compatible with this more general view of affordances, in which direct perception plays a role, but not the only role, in concept formation.

6.2 Affordances in Language Technology

The idea of affordances has been incorporated into work on robotics (Şahin et al., 2007; Zech et al., 2017). Kalkan et al. (2014); Ugur et al. (2009) build a model of affordances based on (object, action, effect) tuples, but focus only on start and end state, and do not encode anything about manner. Relatedly, Nguyen et al. (2020) connects images of objects to language queries describing their uses.

Affordances are also well studied for text understanding tasks. McGregor and Jezek (2019) discuss the importance of affordances in disambiguating meaning of sentences such as "we finished the wine". Other neural net based approaches for affordance learning have relied on curated datasets with explicit affordance labels for each object (Chao et al., 2015; Do et al., 2018). Sometimes, affordance datasets leverage multimodal settings such as images (Myers et al., 2015), or 3D models and environments (Suglia et al., 2021; Mandikal and Grauman, 2021; Nagarajan and Grauman, 2020), but require annotations for every object. In contrast, our model learns affordances in an unsupervised manner, and unlike Fulda et al. (2017), Loureiro and Jorge (2018), McGregor and Lim (2018), and Persiani and Hellström (2019) which extract affordance structure automatically from word embeddings alone, our model learns from interacting with objects in a 3D space, grounding its representations to cause-and-effect pairs of physical forces and object motion.

6.3 Physical Commonsense Reasoning

There has been success in building deep learning networks that reason about object physics by learning to predict their trajectories. These can be broken up into either predicting points in 3D space given object locations (like our approach, e.g. Mrowca et al. (2018), Byravan and Fox (2017), Battaglia et al. (2016), Fragkiadaki et al. (2016), Ye et al. (2018), Rempe et al. (2020)) or inferring future bounding box locations of objects in videos (Weng et al., 2006; Do et al., 2018; Qi et al., 2021; Ding et al., 2021). Both approaches have been successful in encoding complex visual and physical features of objects. We focus on training with 3D simulations, but also test a visual dynamics model (Qi et al., 2021) to compare the affordance information that is encoded from spatial vs. visual data.

More broadly, we contribute to a line of work on building non-linguistic representations of lexical concepts (Bisk et al., 2019). Explicit attempts at grounding to the physical world ground language in 2D images or videos (i.e., pixels) (Hahn et al., 2019; Groth et al., 2018), despite the fact that recent work suggests that text and video pretraining offers no boost to lexical semantic understanding (Yun et al., 2021). Such efforts motivate the creation of large datasets such as Krishna et al. (2016), Yatskar et al. (2016), and Gupta and Malik (2015), which require in-depth human provided annotations that provide a limited list of semantic roles of objects.

Our approach is most directly related to prior work that learns in interactive, 3D settings (Thomason et al., 2016; Ebert and Pavlick, 2020). Especially related are Nagarajan and Grauman (2020) and Zellers et al. (2021). However, their models do not directly ground to the physical phenomena (e.g., entailed positional changes). Instead, they use a symbolic vocabulary of object state changes, whereas our model learns from unlabeled interactions.

7 Conclusion

We propose an unsupervised pretraining method for learning representations of object affordances from observations of interactions in a 3D environment. We show that 3D trajectory data is a strong signal for grounding such concepts and performs better than a standard computer vision approach for learning the desired concepts. Moreover, we show through counterfactual analyses that the learned representations can encode the desired entailments–e.g., that `roll` entails axial rotation.

Our work contributes to an existing line of work that seeks to develop lexical semantic representations of nouns and verbs that are grounded in physical simulations. We advance this agenda by offering a way in which modern "foundation model" approaches to visual and linguistic processing can in fact be corroborative of traditional feature-based approaches to formal lexical semantics. Our results suggest a promising direction for future work, in which pretraining objectives can be augmented to include richer notions of embodiment (e.g., planning, agent-agent interaction) and consequently encoder richer lexical semantic structure (e.g., presuppositions, transitivity).

Acknowledgments

This research is supported in part by DARPA via the GAILA program (HR00111990064). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA or the U.S. Government. Thanks to Chen Sun, Roman Feiman, members of the Language Understanding and Representation (LUNAR) Lab and AI Lab at Brown, and the reviewers for their help and feedback on this work.

References

Stéphane Aroca-Ouellette, Cory Paik, Alessandro Roncone, and Katharina Kann. 2021. PROST: Physical reasoning about objects through space and time. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4597–4608, Online. Association for Computational Linguistics.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and koray kavukcuoglu. 2016. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Benjamin K Bergen. 2012. *Louder than words: The new science of how the mind makes meaning*. Basic Books.

Benjamin K Bergen, Shane Lindsay, Teenie Matlock, and Srini Narayanan. 2007. Spatial and linguistic aspects of visual imagery in sentence comprehension. *Cognitive science*, 31(5):733–764.

Yonatan Bisk, Jan Buys, Karl Pichotta, and Yejin Choi. 2019. Benchmarking hierarchical script knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4077–4085, Minneapolis, Minnesota. Association for Computational Linguistics.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

Anna M Borghi and Lucia Riggio. 2009. Sentence comprehension and simulation of object temporary, canonical and stable affordances. *Brain Research*, 1253:117–128.

Arunkumar Byravan and Dieter Fox. 2017. SE3-nets: Learning rigid body motion using deep neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 173–180.

Yu-Wei Chao, Zhan Wang, Rada Mihalcea, and Jia Deng. 2015. Mining semantic affordances of visual object categories. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4259–4267.

Erica Cosentino, Giosuè Baggio, Jarmo Kontinen, and Markus Werning. 2017. The time-course of sentence meaning composition. n400 effects of the interaction between context-induced and lexically stored affordances. *Frontiers in Psychology*, 8.

David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matt Botvinick. 2021. Attention over learned object embeddings enables complex visual reasoning. In *Advances in Neural Information Processing Systems*, volume 34, pages 9112–9124. Curran Associates, Inc.

Thanh-Toan Do, Anh Nguyen, and Ian Reid. 2018. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5882–5889.

Dylan Ebert and Ellie Pavlick. 2020. A visuospatial dataset for naturalistic verb learning. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 143–153.

Jerome Feldman. 2008. *From molecule to metaphor: A neural theory of language*. MIT press.

Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. 2016. Learning Visual Predictive Models of Physics for Playing Billiards. *arXiv:1511.07404 [cs]*. ArXiv: 1511.07404.

Nancy Fulda, Daniel Ricks, Ben Murdoch, and David Wingate. 2017. What can you do with a rock? affordance extraction via word embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1039–1045.

James J Gibson. 1977. The theory of affordances. *Hilldale, USA*, 1(2):67–82.

Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR.

Oliver Groth, Fabian B. Fuchs, Ingmar Posner, and Andrea Vedaldi. 2018. Shapestacks: Learning vision-based physical intuition for generalised object stacking. In *The European Conference on Computer Vision (ECCV)*.

Saurabh Gupta and Jitendra Malik. 2015. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*.

Meera Hahn, Andrew Silva, and James M. Rehg. 2019. Action2Vec: A Crossmodal Embedding Approach to Action Learning. *arXiv:1901.00484 [cs]*. ArXiv: 1901.00484 version: 1.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2020. Reducing sentiment bias in language models via counterfactual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 65–83, Online. Association for Computational Linguistics.

Sinan Kalkan, Nilgün Dag, Onur Yürüten, Anna M Borghi, and Erol Şahin. 2014. Verb concepts from affordances. *Interaction Studies*, 15(1):1–37.

Najoung Kim and Paul Smolensky. 2021. Testing for grammatical category abstraction in neural language models. *Proceedings of the Society for Computation in Linguistics*, 4(1):467–470.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

Daniel Loureiro and Alípio Jorge. 2018. Affordance extraction and inference based on semantic role labeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 91–96, Brussels, Belgium. Association for Computational Linguistics.

Priyanka Mandikal and Kristen Grauman. 2021. Learning dexterous grasping with object-centric visual affordances. In *ICRA*.

Claudia Mazzuca, Chiara Fini, Arthur Henri Michalland, Ilenia Falcinelli, Federico Da Rold, Luca Tummolini, and Anna M. Borghi. 2021. From affordances to abstract words: The flexibility of sensorimotor grounding. *Brain Sciences*, 11(10).

Stephen McGregor and Elisabetta Jezek. 2019. A distributional model of affordances in semantic type coercion. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, pages 1–7, Gothenburg, Sweden. Association for Computational Linguistics.

Stephen McGregor and KyungTae Lim. 2018. Affordances in grounded language learning. In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 41–46, Melbourne. Association for Computational Linguistics.

Sergio Mota. 2021. Dispensing with the theory (and philosophy) of affordances. *Theory & Psychology*, 31(4):533–551.

Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li Fei-Fei, Joshua B. Tenenbaum, and Daniel Yamins. 2018. Flexible neural representation for physics prediction. In *NeurIPS*.

Austin Myers, Ching L. Teo, Cornelia Fermüller, and Yiannis Aloimonos. 2015. Affordance detection of tool parts from geometric features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1374–1381.

Tushar Nagarajan and Kristen Grauman. 2020. Learning affordance landscapes for interaction exploration in 3d environments. In *NeurIPS*.

Thao Nguyen, Nakul Gopalan, Roma Patel, Matthew Corsaro, Ellie Pavlick, and Stefanie Tellex. 2020. Robot Object Retrieval with Contextual Natural Language Queries. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA.

Michele Persiani and Thomas Hellström. 2019. Unsupervised inference of object affordance from text corpora. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 115–120, Turku, Finland. Linköping University Electronic Press.

James Pustejovsky and Nikhil Krishnaswamy. 2014. Generating simulations of motion events from verbal descriptions. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 99–109, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

James Pustejovsky and Nikhil Krishnaswamy. 2018. Every object tells a story. In *Proceedings of the Workshop Events and Stories in the News 2018*, pages 1–6, Santa Fe, New Mexico, U.S.A. Association for Computational Linguistics.

Haozhi Qi, Xiaolong Wang, Deepak Pathak, Yi Ma, and Jitendra Malik. 2021. Learning long-term visual dynamics with region proposal interaction networks. In *ICLR*.

Davis Rempe, Srinath Sridhar, He Wang, and Leonidas J. Guibas. 2020. Predicting the physical dynamics of unseen 3d objects. *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*.

Katrin Sakreida, Claudia Scorolli, Mareike M Menz, Stefan Heim, Anna M Borghi, and Ferdinand Binkofski. 2013. Are abstract action words embodied? an fmri investigation at the interface between language and motor cognition. *Frontiers in human neuroscience*, 7:125.

Jeffrey Mark Siskind. 2001. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of artificial intelligence research*, 15:31–90.

Mark Steedman. 2002. Plans, affordances, and combinatory grammar. *Linguistics and Philosophy*, 25(5):723–753.

Alessandro Suglia, Qiaozi Gao, Jesse Thomason, Govind Thattai, and Gaurav S. Sukhatme. 2021. Embodied BERT: A transformer model for embodied, language-guided visual task completion. *CoRR*, abs/2108.04927.

Lawrence J. Taylor, Carys Evans, Joanna Greer, Carl Senior, Kenny R. Coventry, and Magdalena Ietswaart. 2017. Dissociation between semantic representations for motion and action verbs: Evidence from patients with left hemisphere lesions. *Frontiers in Human Neuroscience*, 11.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2018. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Jesse Thomason, Jivko Sinapov, Maxwell Svetlik, Peter Stone, and Raymond J. Mooney. 2016. Learning multi-modal grounded linguistic semantics by playing "i spy". In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 3477–3483, New York City.

Emre Ugur, Erol Sahin, and Erhan Oztop. 2009. Predicting future object states using learned affordances. In *2009 24th International Symposium on Computer and Information Sciences*, pages 415–419. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shiuh-Ku Weng, Chung-Ming Kuo, and Shu-Kang Tu. 2006. Video object tracking using adaptive Kalman filter. *Journal of Visual Communication and Image Representation*, 17(6):1190–1208.

Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual semantic role labeling for image understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5534–5542.

Tian Ye, Xiaolong Wang, James Davidson, and Abhinav Gupta. 2018. Interpretable intuitive physics model. In *ECCV*.

Tian Yun, Chen Sun, and Ellie Pavlick. 2021. Does vision-and-language pretraining improve lexical grounding? In *Findings of EMNLP*.

Philipp Zech, Simon Haller, Safoura Rezapour Lakani, Barry Ridge, Emre Ugur, and Justus Piater. 2017. Computational models of affordance in robotics: a taxonomy and systematic classification. *Adaptive Behavior*, 25(5):235–271.

Rowan Zellers, Ari Holtzman, Matthew E. Peters, R. Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. 2021. PIGLeT: Language Grounding Through Neuro-Symbolic Interaction in a 3D World. In *ACL/IJCNLP*.

Erol Şahin, Maya Çakmak, Mehmet R. Doğar, Emre Uğur, and Göktürk Üçoluk. 2007. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4):447–472.

A Appendix A

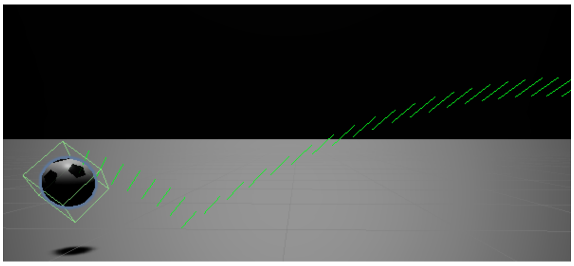


Figure 8: Example of an interaction from SPATIAL. The model predicts the position of the soccer ball at future timesteps. To do that, it must encode some knowledge that soccer balls bounce and roll. As input, our model takes 9 3D points: the eight corners of the box surrounding the ball, plus the center point.

A.1 Spatial Model Training Details

For both of the 3D spatial data models, we train with an encoder-decoder transformer with one encoder and one decoder layer with one attention head. We found that changing the number of attention heads did not affect performance noticeably in either direction. We use a batch size of 64 and a transformer embedding dimension of 100. We use a feed-forward dimension of 200. We initialize with a learning rate of 1e-4. The models were trained to predict the next $k = 8 - 16$ frames and we did not see a large benefit in training to predict longer sequences. We trained the models for 400 epochs although we notcied the ablated 3D Blind model tended to converge at or before 100 epochs across our experiments.

The beginning of the sequence, which was up to four seconds minus the k prediction frames, was fed into the transformer encoder which encoded representations of dimension e. We averaged these output embeddings as input in our probing experiments. The e embeddings were fed into the decoder network, which then predicts the next k frames. We believe that training with longer sequences would be more beneficial for training a decoder-only model, which we would like to explore in future work. In preliminary experiments, we tested whether masking a proportion of frames in the encoder would be beneficial for the representation learning task. We saw a slight decrease in performance, and so did not perform a thorough analysis on the effect of masking.

A.2 t-SNE Configuration

We report a t-SNE of representations derived from our 3D Blind model and the 2D Visual model. The parameters for creating each t-SNE was similar but varied in a few ways: **Common Hyperparameters:** learning rate: 200, iterations: 1000, stopping threshold of gradient norm: 1e-7 **3D Blind t-SNE specifics:** perplexity: 30, initialized randomly **2D Visual specifics:** perplexity: 5, initialized with PCA. We found that random initialization was inconsistent in that it would sometimes cause small clouds of dense points to appear as their own clusters.

B Appendix B

B.1 RPIN Training details

We use a learning rate of 1e-3 with a batch size of 50 and train for a maximum of 20M iterations with 40,000 warmup iterations. Training data is augmented with random horizontal flips. Unlike in Qi et al. (2021) we don't use vertical flips because our videos contain objects falling due to gravity. One important difference is that at training time the model predicts 10 frames in the future, and at test time predicts 20 (as opposed to 20 and 40 respectively in Simulated Billiards). Within one video, our interactions seem more complex than one sequence in the Simulated Billiards dataset, so we introduced this difference to create more training examples.

C Appendix C

C.1 Counterfactual Perturbations Setup

We start with a base set of 10 sequences: 5 with a sliding object (cardboardBox_03) and 5 with a rolling object (BuckyBall). We then create 20 minimal-edit perturbations to create a final set of 200 sequences. We perturb the following features one at a time: {mass, force velocity, starting x position, starting z position, shape, angular rotation}. For most features, we generate 4 perturbations. For example, the x and z positions are altered by {-2m, -1m, +1m, +2m} where 'm' is the Unity meter. All objects start with 1.14 units of mass and similar to the starting position variable, is altered by $1.14 + (i \times .1)$ where i is in the set {-2, -1, 1, 2}. For the shape parameter, we only change the 3D model used to generate the base sequence.

For the sliding videos, we use plate02 and book_0001c. For the rolling videos we use BombBall and modified Soccer Ball. Note that we modify the Soccer Ball model that is in the train set, but modify the mass (1.14) and size of the model so that it is technically an unseen object. We chose to do this because we wanted to use a more plain spherical object, which was not an option for the remaining test objects. Angular rotation either perturbs the sequence by freezing the rotation along all axes (in the case of objects that normally roll) or replacing the physics collider with a sphere (causing the object to roll – in the case of objects that tend to slide instead of roll). Figure 11 shows additional perturbations and a sliding object example of the counterfactual analysis.

Object	**Test set?**	**Slide**	**Roll**	**Stack**	**Contain**	**W-Grasp**	**Bounce**
BombBall	✓		✓				✓
EyeBall			✓				✓
SpikeBall			✓				
Vase_Amphora			✓				
Vase_Hydria			✓				
Vase_VoluteKrater	✓		✓		✓		
book_0001a		✓		✓			
book_0001b		✓		✓			
book_0001c	✓	✓		✓			
bowl01		✓	✓	✓	✓		
cardboardBox_01		✓		✓			
cardboardBox_02		✓		✓	✓		
cardboardBox_03	✓	✓		✓			
Cola Can		✓	✓	✓		✓	
Pen black	✓		✓			✓	
Gas Bottle	✓		✓				
Soccer Ball			✓				✓
can small		✓	✓	✓		✓	
can	✓	✓	✓	✓		✓	
meat can box		✓		✓			
spam can		✓		✓		✓	
AtomBall			✓				✓
Bottle2	✓		✓			✓	
plate02	✓	✓		✓			
plate02_flat		✓		✓			
Bottle1			✓			✓	
WheelBall			✓				✓
wine bottle 04	✓		✓		✓	✓	
coin	✓	✓		✓			
BuckyBall	✓		✓				✓
SplitMetalBall	✓		✓				✓
bowl02		✓	✓	✓	✓		
bowl03	✓	✓	✓	✓	✓		
mug02		✓			✓	✓	
mug03	✓	✓			✓	✓	
Old_USSR_Lamp_01	✓	✓				✓	
lamp	✓	✓	✓			✓	
Ladle		✓				✓	
Apple			✓				

Table 1: All objects in the dataset and their associated affordances

Binary Classification Accuracy of Affordance Probes (Random=50%)

	N examples	3D Blind		3D Visual-Spatial		2D Visual		No-Training	
Affordance		Acc. (%)	F1	Acc. (%)	F1	Acc. (%)	F1	Acc. (%)	F1
Slide	1715	59.2	62	**63.7**	67	59.6	62	61.2	64
Roll	1258	66.5	66	**69.0**	70	56.0	58	66.3	66
Stack	1307	64.7	65	**65.7**	63	58.0	58	63.4	63
Contain	1510	**66.5**	68	65.2	67	58.2	63	59.3	64
W-Grasp	1652	**67.2**	68	66.3	0.68	62.9	61	63.4	64
Bounce	276	**82.6**	83	79.7	79	73.9	76.	76.0	75

Table 2: Results from probing experiments on RPIN compared to the unity models trained on the same amount of data. Because data was limited, we partition the data so that there is an even number of positive and negative examples in the test set for each affordance. Interaction based pretraining outperforms visual dynamics in all categories

Affordance	Number of Objects	Percentage of objects (%)
Slide	22	56.41
Roll	23	58.97
Stack	17	43.59
Contain	8	20.51
Wrap-grasp	13	33.33
Bounce	7	17.95

Table 3: Each affordance we are interested in learning and the number and percentage of objects out of the 39 have a positive label for that affordance.

RPIN Model validation loss in the $t \in [T_{train}, 2 \times T_{train}]$ setting

Model	Loss (MSE)
SimB (Qi et al., 2021)	25.77
SimB (our results)	15.53
Unity Videos	20.98

Table 4: Each affordance we are interested in learning and the number and percentage of objects out of the 39 have a positive label for that affordance.

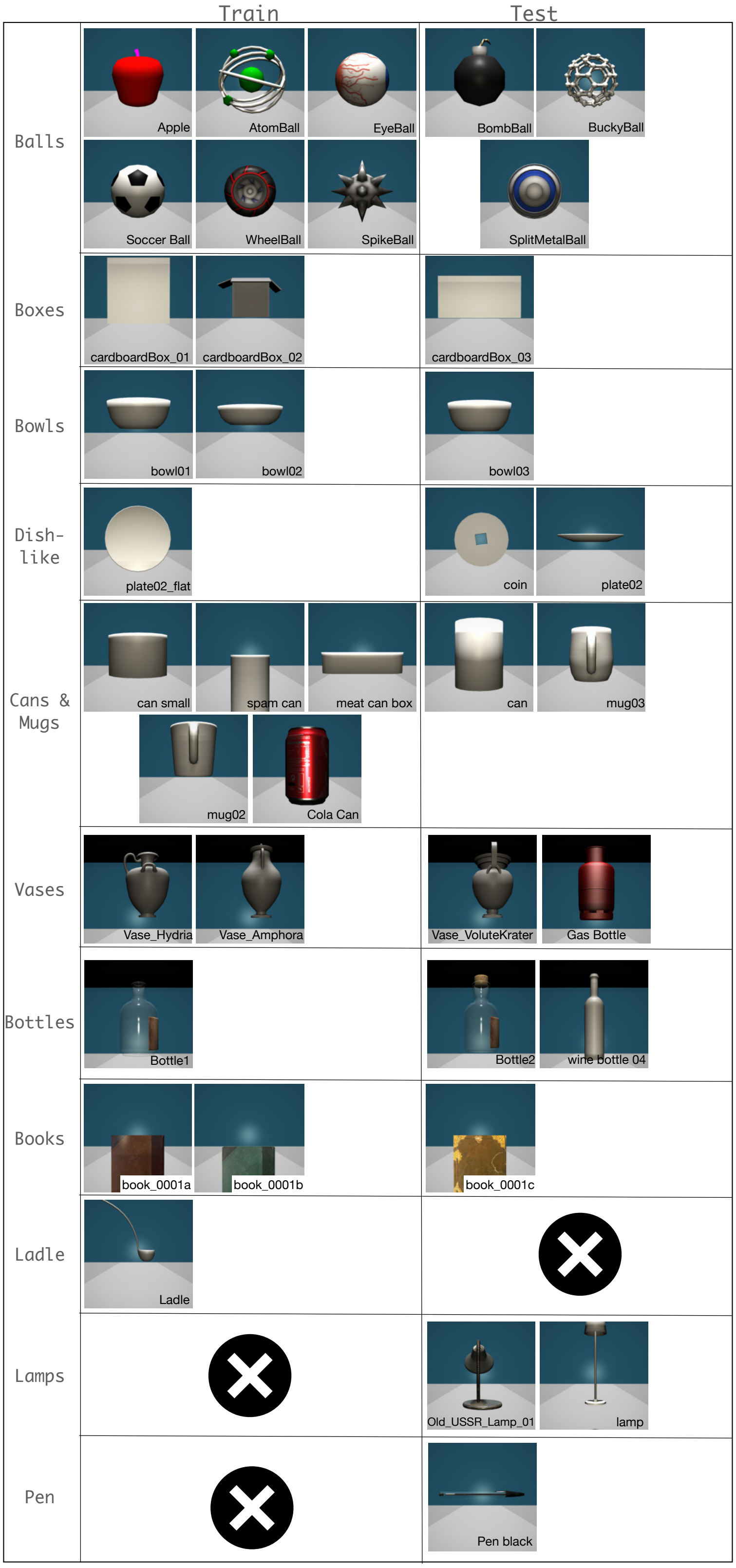


Figure 9: All objects that were used in training and testing. Some objects in the test set are visually similar to their training analogues, but differ in size and mass.

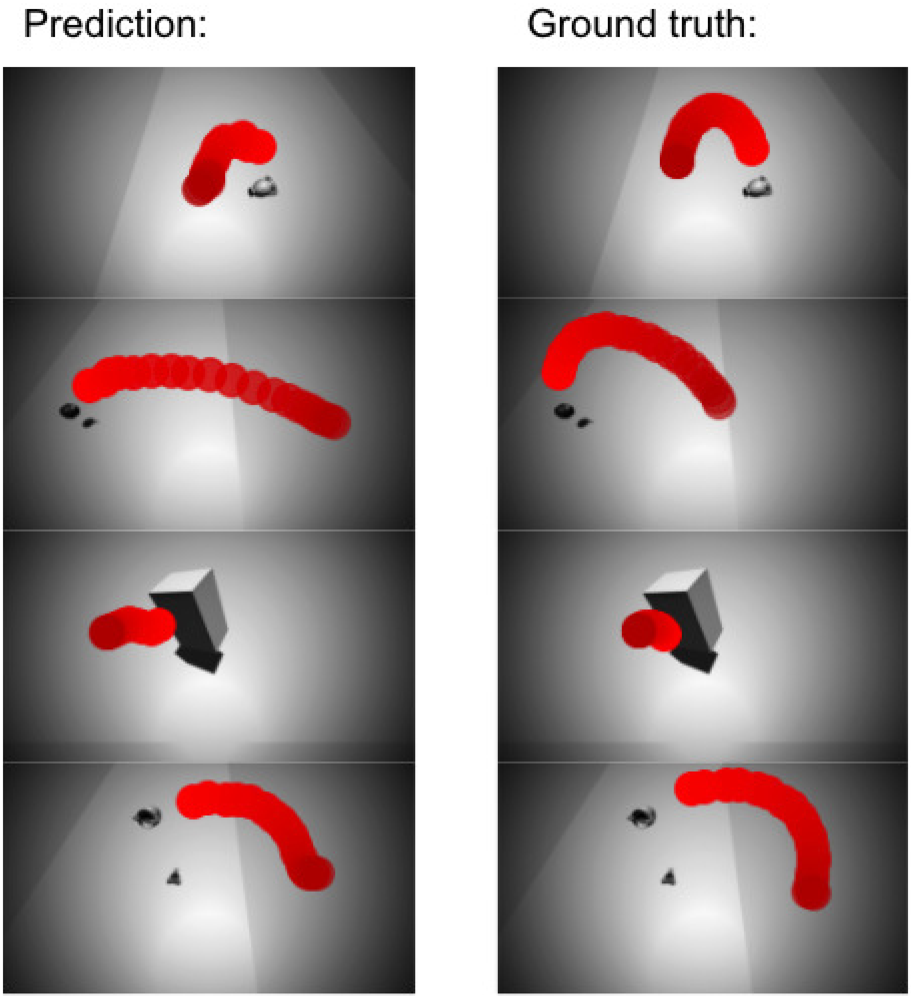


Figure 10: Results from the training of the RPIN visual dynamics model on videos of our Unity dataset interactions. Red circles show the predictions of the following center points of the bounding boxes of the object given the start of the interaction

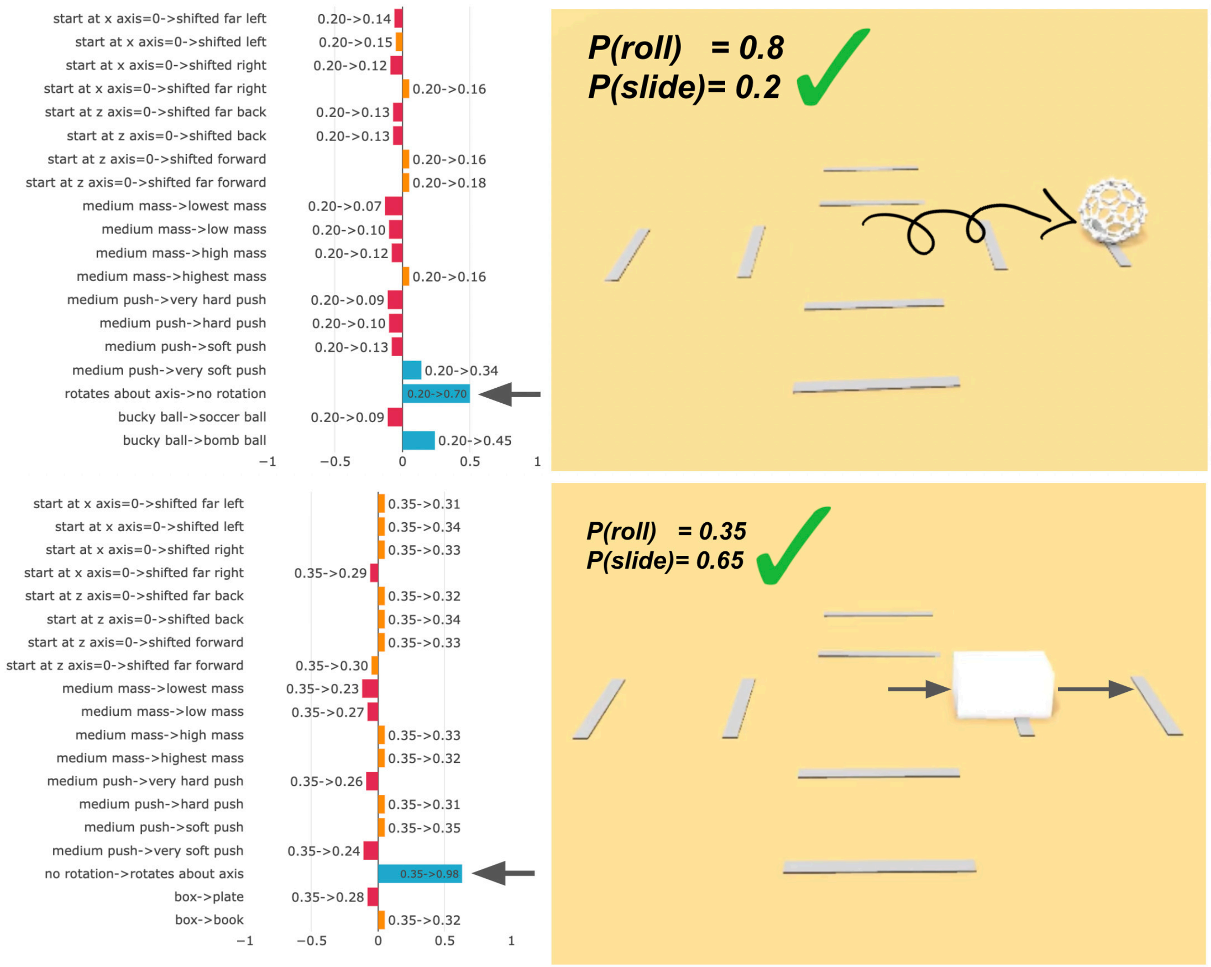


Figure 11: Two examples from the counterfactual analysis that show robustness to changing spurious features. The table in the top example displays the changes in probability in predicting the object as sliding. Conversely, the bottom example table shows the change in probability of predicting the object as rolling. Arrows in the left table indicate where the perturbation *does* affect the label of the action (either by making the object able or unable to roll). In both cases, the probe correctly flips its prediction on the encoding. The sequence prediction model appears to be sensitive to certain features such as distance traveled. For example, changing the object from the "bucky ball" to the "bomb ball" decreases the model's confidence that the object rolling (though, the probe still correctly assigns a majority of the probability to `roll`). However, in this perturbation, the bomb ball gets stuck on its 'cap' (Figure 9) and only completes one rotation.

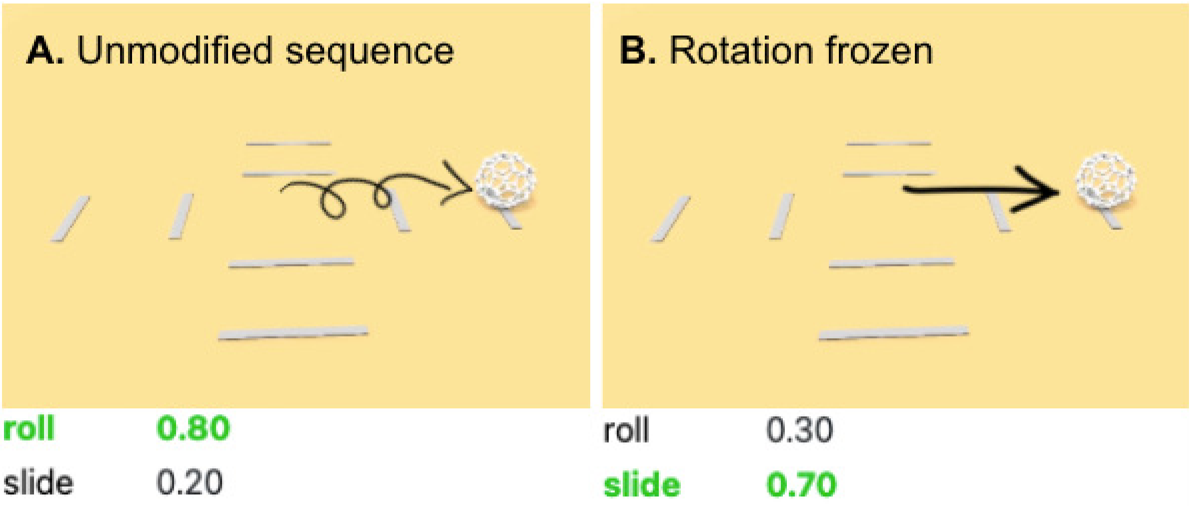


Figure 12: Left: a sequence generated with normal physics. Right: rotation locked, with all other physical properties of the interaction the same. Freezing the rotation such that the object slides causes the model to encode the action as a `slide` rather than a `roll`