Pretraining over Interactions for Learning Grounded Object Representations

Anonymous ACL submission

Abstract

Large language models have been criticized for their limited ability to reason about *affordances* - the actions that can be performed on an object. It has been argued that to accomplish this, models need some form of grounding, i.e., connection, to objects and how they interact in the physical world. Inspired by the way humans learn about the world through interaction, we develop an approach to learning physical properties directly. We introduce a dataset of 200k object interactions in a 3D virtual environment and a self-supervised pretraining objective for learning representations of these objects. We show with probing and clustering experiments that even in the zero-shot setting, derived models learn robust representations of objects and their affordances in an unsupervised manner. Our model outperforms pretrained language and vision models on an affordance prediction baseline, suggesting that pretraining on observed interactions encodes grounded information that is not readily learned in conventional text or vision models.

1 Introduction

Although representations learned from large language models have proven useful on many language understanding evaluations (Raffel et al., 2019), (Brown et al., 2020), it is unclear how much, even basic, physical commonsense is captured by language model pretraining. Humans can rely on rich background information on how the world works when reasoning with language. In part, this background knowledge is supplied in the form of *affordances*, representations of the actions that are applicable to objects. An understanding of affordances endows humans with the ability to reason about novel situations and objects using language. Though language models learn from text statistics and can learn to associate high-frequency noun-verb pairs with objects and their affordances (Fulda et al., 2017), they struggle with basic relationships if they are not explicitly manifested in written language. Further, they struggle to infer the properties that explain *why* objects afford those actions (Forbes et al., 2019), thus limiting their ability to generalize to novel words and situations. For example, when tested in a zero-shot setting, large language models can not accurately predict if a common object can afford simple actions like rolling or bouncing (Aroca-Ouellette et al., 2021), a task that is trivially solved by humans. A proposed solution to contend with this gap is through grounded language learning (Bender and Koller, 2020; Merrill et al., 2021; Bisk et al., 2020), but there is little evidence that text and vision pretraining, a common approach, improves language representations in general (Yun et al., 2021).

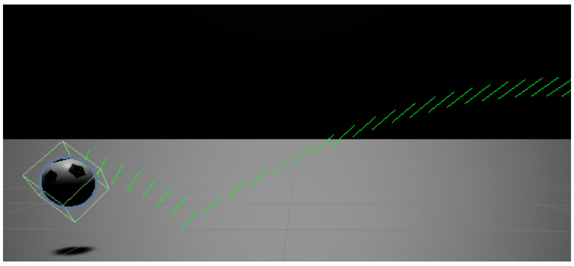


Figure 1: Example of an interaction in our dataset. The model predicts the position of the soccer ball at future timesteps. To do that, it must encode some knowledge that soccer balls `bounce` and `roll`. As input, our model takes the eight 3D points illustrated as the corners of the box surrounding the ball, as well as the center point.

In this work, we address this gap. Inspired by the way in which infants learn about objects by touching, lifting, dropping or throwing them and observing their subsequent behavior, we attempt to teach a model via interaction with objects in a 3D virtual environment. This way, our model directly learns what aspects of an object imply certain affordances, such as the roundness of an object might imply its roll-ability. We use a set of objects and their affordances—that is, the actions that can be performed on an object (e.g., a ball affords rolling)—to evaluate how well our model captures physical concepts.

In order to do this, we train a transformer to predict an object's motion after applying physical forces to it. We demonstrate through probing and clustering experiments that the intermediate representations from our model encode human-interpretable affordance categories, even in a zero-shot setting on unseen object types. Importantly, our model is able to learn these concepts simply from learning to predict how objects will behave in a 3D physics simulation. We finally show that the learned object representations contain critically richer information than what is encoded in large pretrained text and vision models, outperforming both on an affordance prediction task. This paper provides evidence that interaction-based pretraining improves upon language and image representations' ability to encode physical commonsense knowledge and lays the groundwork for incorporating this knowledge into pretrained text embeddings. In summary, the main contributions of this paper are as follows:

1. We introduce a novel pretraining objective for learning affordance concepts in an unsupervised manner.
2. We release a dataset of 200k simulated object interactions and their motions through space.[1]
3. We demonstrate shortcomings of text and vision representations in encoding affordance information and show that interaction pretraining outperforms both in an affordance prediction task

2 Related Work

2.1 Affordances

Past research on affordances generally frames the objective as learning which actions an object allows in a specified situational context. However, these works rely 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 (Mandikal and Grauman, 2021; Nagarajan and Grauman, 2020), but require annotations for every object. Different from this line of work, our approach learns affordances in an unsupervised manner, and unlike Fulda et al. (2017), which extracts an affordance structure 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.

2.2 Intuitive Physics

There has been success in building neural models that are able to predict object motion and dynamics in virtual worlds. Many recent works train networks that model complex collisions between multiple objects (Byravan and Fox, 2017), Weng et al. (2006), as well as deformable object collisions (Mrowca et al., 2018). Oftentimes, these approaches involve interaction with a 2D or 3D world (Fragkiadaki et al., 2016), (Battaglia et al., 2016). These works inform our approach, however, our goal differs in that we examine how to optimally learn concepts of object affordances to support language learning.

2.3 Physical Commonsense Reasoning

There is doubt that language representations encode robust commonsense reasoning knowledge (Bisk et al., 2019). An analysis of the representation contents of these models shows that only explicitly manifested and documented associations are reliably learned (Forbes et al., 2019). Explicit attempts at grounding to the physical world use multimodal models (Hahn et al., 2019), connecting language to videos. Such spatio-temporal features are limited to the 2D information provided by pixels, and do not offer the same degree of granularity attained by our approach. Nguyen et al. (2020) relate images of objects to language queries describing their uses. Large datasets such as Krishna et al. (2016), Yatskar et al. (2016), and Gupta and Malik (2015) require in-depth human provided annotations that provides a limited list of semantic roles of objects. Additionally, Yun et al. (2021) finds no evidence that text and vision pretraining offers any boost to lexical semantic understanding, and it has been shown that learning through interaction can outperform purely visually grounded models (Thomason et al., 2016).

In an effort similar to ours, Zellers et al. (2021), aim to ground language to interactions with objects based on events in a simulated environment using a

[1] https://drive.google.com/drive/folders/<anonymized>

neurosymbolic architecture. Unlike our work, the model requires a symbolic vocabulary of object properties and actions that occur in the interactions. Our approach requires no such knowledge injection and instead directly learns from the raw interactions themselves. Like Nagarajan and Grauman (2020), which uses the same simulation environment, the model trained is not directly grounded to the physical phenomena that occur during these actions. In this paper, we focus on unsupervised interaction with objects in the simulated world and show that our model coincidentally learns physical commonsense concepts important for understanding language.

3 Dataset

Instead of modeling objects using visual data, we take inspiration from the way children explore objects by interacting with them. The intuition is that from frequent observation of an object's reaction to physical impulses (the analagous equivalent of pushing, throwing, or dropping the object) we can infer which actions the object affords, and thereby learn robust representations of those objects. Learning affordances is a difficult task that requires knowledge of highly nuanced properties that are often independent of the visual appearance of an object. A can of soup affords rolling, but only when it lies on its side. Rarely, if ever, will we use language to express this concept; humans supply it from experience. Collecting a video dataset to distinguish these differences clearly is too costly and noisy, and only provides 2D pixel data. To overcome these limitations, we gather a dataset of 200k simulations of forces exerted on objects in the Unity 3D game engine[2]. Unity uses a realistic physics simulation, which allows us to emulate interactions with models of objects that a human could have in the real world.

3.1 Environment and Data Collection

Data is collected in a flat empty room using the Unity physics engine on a collection of 39 realistic objects collected from the Unity Asset Store (See Appendix A). These objects were chosen based on the availability of affordance labels (see Section 3.2). For each sequence, an object is instantiated at rest on the ground. A random impulse force determined as either a 'push' flat along the ground, or a 'throw' into the air is exerted on the object. Instead of collecting flat pixel-level data from the point of view of a camera, we record the coordinates of the object in 3D space at a rate of 60 frames per second. The sequence ends when the object stops moving or after 4 seconds elapses. We only exert a single impulse on an object per sequence. 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, as shown in Figure 1.

3.2 Affordances

If we are told that a *dax* can roll, is that object more likely to be round or flat? Such questions require the understanding of an object's *affordances*. To evaluate the model's ability to learn affordance concepts, we use previous work (Aroca-Ouellette et al., 2021; Chao et al., 2015; Myers et al., 2015) to assign affordance labels to object classes. The 39 objects we use in our experiments were selected based on their availability on the Asset Store and their overlap with the object classes used in these datasets. Our objects have binary labels for the set of affordances: $A = \{$roll, slide, stack, contain, wrap-grasp, bounce$\}$. These affordances were chosen because of their use in physical reasoning benchmarks (e.g., Aroca-Ouellette et al. (2021)), and they can be observed from single-object interactions alone. Note that our model never sees these labels during training; we only use them for evaluation. Wrap-grasp (wgrasp) refers to the act of wrapping a hand around an object, and holding it with the palm and fingers (e.g., afforded by mugs, or long slender objects). See Appendix A for a breakdown of object types and their affordances. This set of actions is well represented in our dataset. We acknowledge that contain and wgrasp are likely to be particularly challenging for a model because grasping presupposes having a hand (which our disembodied agent cannot simulate), and containing can only be learned indirectly by learning about concavity.

4 Models

We develop a self-supervised pretraining task under which a model is trained to predict the motion of an object given a starting sequence of object positions.

[2] https://unity.com/

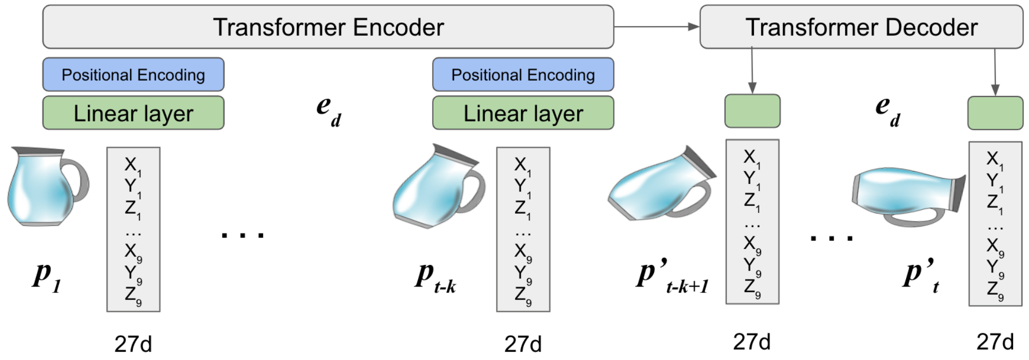


Figure 2: Basic architecture. The model receives coordinates describing an object's trajectory and predicts the remainder of the sequence.

Each input to the model consists of one interaction with a single object, with the goal of predicting its trajectory. Consider the example in Figure 1, where a soccer ball that is thrown, moves in an arc, and is about to hit the ground. When the ball lands, how high will it bounce? As it settles, will it roll or slide across the ground? If a model can connect the visual appearance and movement of objects to object affordance concepts, it should be able to infer the positions of the ball in some future states. We use a transformer architecture (Vaswani et al., 2017) with embedding size e_d (Figure 2). First, a single linear layer is used to encode our input coordinates to the dimension of the transformer. The transformer is then fed the first $t-k$ timesteps where $k \geq 1$. Given some ground truth coordinates p_i our model is trained using a Mean Squared Error (MSE) loss summed for each of the predicted point $p_i\prime$:

$$MSE(P\prime, P) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=t-k}^{t}(p_{ij}\prime - p_{ij})^2 \quad (1)$$

The model outputs predictions for each timestep up to t. We also train a variant of the model with an additional set of input vectors encoding visual information about the object at rest. We describe these models in Sections 4.1 and 4.2. In both cases, the final object representations are obtained from the transformer encoder output.

4.1 Base Model

Our base model encodes the input sequence P containing the object's 3D coordinates in virtual space. Each input token p_i is a vector of the position of the object in 3D space at time step i. As described in Section 3, each position p_i contains 9 distinct points corresponding to the object center and the eight corners of the rectangular bounding box encapsulating the object. Before processing, each p_i is fed into a single feed-forward layer to project it to the input size of the transformer.

4.2 Multiview Images

The input sequence P contains the object's position over time, but does not include any explicit clues regarding object shape. We therefore also consider a model in which the agent has information about what the object looks like and can use these features as clues to how an object will behave. We include this visual information as additional inputs to the model. We implement this by inputting coordinates P followed by a SEP token and a sequence of image encodings I of the object's six faces (an image taken from each side of the object). We refer to these impressions as the object's multiview. Image encodings are bottleneck embeddings from a pretrained ResNet-34 model (He et al., 2015) and are frozen for training.

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 train the model to classify if the sample was perturbed. We add a linear binary classification layer on top of a CLS token to predict c and add the cross entropy loss to our main objective:

$$L(P\prime, P, y) = MSE(P\prime, P) + CE(c, y) \quad (2)$$

4.3 Intermediate Representations

The goal of our modeling effort is to obtain intermediate object representations that carry distinguishing information for affordance prediction. To create a single representation of a sequence, we average the encoder outputs. In the base model

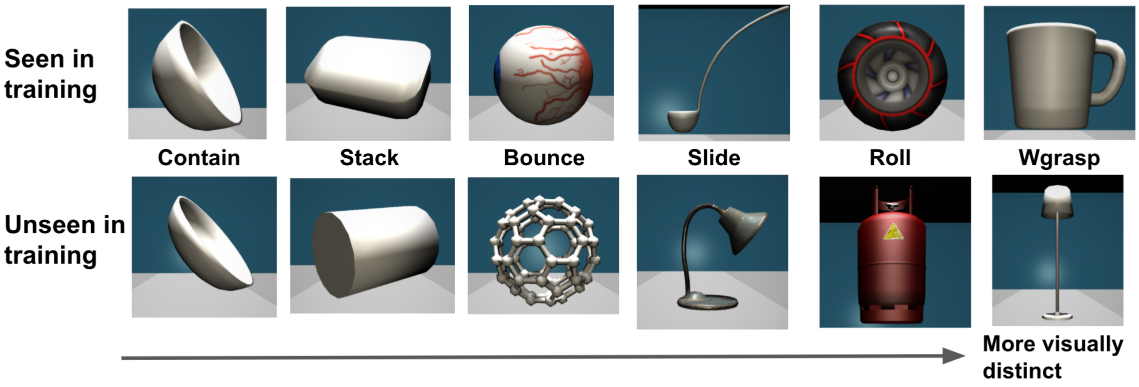


Figure 3: Examples of objects seen during training and an unseen object counterpart, ordered with respect to visual similarity between the objects in each pair to show the varying degree to which the zero-shot objects differ from seen objects. Both depicted objects are positively labeled for the corresponding affordance.

we average the encoder output tokens s_i to form s. In the Multiview model case, we only average the encoder output tokens of the multiview image tokens v_i to form representation mv.

5 Experiments

Our goal is to demonstrate that our trajectory-prediction pretraining objective captures rich object affordance information that text and vision representations alone do not. In all cases, we examine encodings in the zero-shot setting to show that our model generalizes to unseen data. We probe our intermediate representations for evidence of affordance information (Section 5.1) and illustrate that our model organizes the latent space into human-interpretable clusters of affordances (Section 5.3). Finally, in Section 6 compares performance in affordance prediction of our representations to text and vision embeddings. All experiments use encodings of unseen object types U_{enc}^{mv}, unless otherwise specified. Unseen objects are chosen so that they have similar affordance labels to seen objects, but vary in appearance and shape (Figure 3). If the model learns to generalize about the kinds of shapes of objects that afford different actions, we would expect high quality representations regardless of the shape of a novel object.

5.1 Probing for Affordances

We use probing classifiers (Veldhoen et al., 2016; Adi et al., 2017; Conneau et al., 2018; Hupkes et al., 2018; Hewitt and Manning, 2019) to measure the extent to which the intermediate representations of our model encode the desired affordance information. 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 network trained to classify whether the object in the sequence affords a certain action. We train a separate classifier probe for each of the six affordance classes on inputs from unseen object types only. As shown in Table 1, our multiview model is able to infer affordance information from intermediate representations with an accuracy that significantly exceeds random chance.

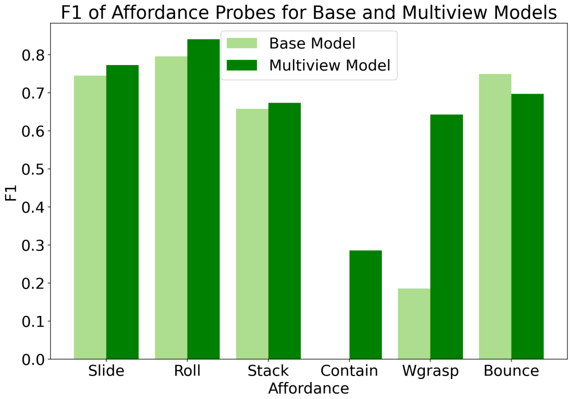
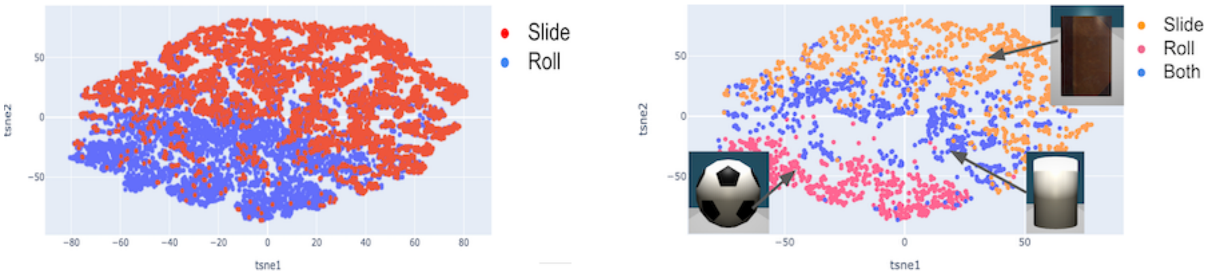


Figure 4: Including images of the object generally improves probe classification performance, especially in the case of `wgrasp` and `contain`.

For example, the model is able to classify whether an object slides based on its intermediate representation 74% of the time, which is 21 percentage points above chance (53% of the objects afford rolling over sliding). The model has the highest accuracy predicting sliding and rolling behavior, and the worst predicting containing and wgrasping. It makes sense for the model to capture this distinction because in essentially every

Affordance	Majority (%)	Accuracy (%)	Precision	Recall	F1
Slide	52.9	74.4	0.73	0.82	0.77
Roll	64.8	78.9	0.82	0.86	0.84
Stack	64.8	78.5	0.72	0.64	0.67
Contain	76.4	77.84	0.55	0.19	0.29
W-Grasp	58.8	72.98	0.70	0.59	0.64
Bounce	82.2	89.2	0.71	0.69	0.70

Table 1: Probe classifier performance for representations of objects extracted from the Multiview Model.



(a) All Sequences that afford either rolling or sliding

(b) Subset of sequences of objects that afford only rolling, only sliding, or both

Figure 5: t-SNE projections of model representations. The left panel shows the separation between rolling and sliding across all sequences and objects. The right panel portrays a subset of objects that afford rolling and sliding. Objects that afford both populate the center area between the single-affordance encodings.

sequence, the object either slides or rolls along the ground. `contain` and `wgrasp` are the opposite in the sense that they are never directly observed in our setting. Containing is only weakly connected to shape, and our dataset would need to have sequences with multiple objects in order to observe a containing action.`Wgrasp`, can never be observed because our disembodied agent has no hands.

5.2 The Effect of Visual Information

The basic version of our model relies exclusively on location traces. We find that, overall, the results improve when we include visual information from the multiview object images, especially in the case of `contain` and `w-grasp`, which require some reasoning about shape or multiple object interactions. Our results are shown in Figure 4 and indicate that the biggest improvements stem from those two cases. These findings are promising for the argument that grounding to different domains provides a more complete “understanding” of objects.

5.3 Qualitative Analysis

We analyze how the representations of the motion sequences cluster. If our model is learning intuitive concepts of affordances, we expect different sequences of actions to form distinct clusters. Based on a t-SNE projection of the sequence representations in U_{enc}^{base}, we find that the most salient pattern the model encodes is the distinction between rolling and sliding, with the model generally grouping sequence representations into one of these two groups. Interestingly, objects that afford both sliding and rolling (e.g., a `can` slides when upright, but rolls on its side) have representations that span both groups, as shown in Figure 5. Because nearly every sequence includes an object sliding or rolling along the ground at some point, we believe this is the most fundamental characteristic for the model to encode in order to predict the withheld frames.

6 Comparison to Language and Vision Representations

Statistical language models struggle to encode affordance information from text alone (Forbes et al., 2019), making it difficult to reason about physical properties of objects. We show interaction pretraining, however, consistently encodes this knowledge into representations of unseen object types. Additionally, we compare to encodings from pretrained vision models (He et al., 2015) to demonstrate that the choice of grounding domain matters. Although each model operates on different types of input,

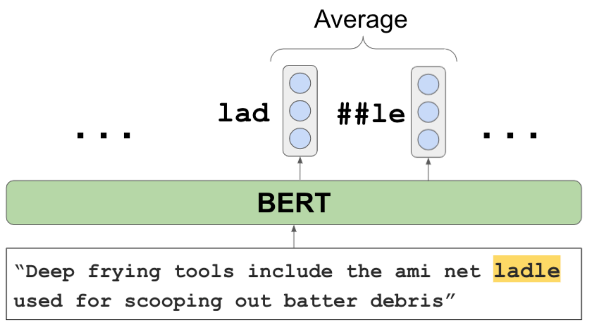

Average
. . .
lad
##le
. . .
BERT
"Deep frying tools include the ami net ladle used for scooping out batter debris"

Figure 6: For the text setting of our affordance prediction task, we find sentences from the Wikitext corpus that mention object types from our dataset and use only the contextualized embeddings of those objects as inputs to our model.

this comparison shows how well each modality encodes affordances. We find that our model's representations outperform both text and vision baselines, which are derived from much larger models. Our results indicate that in general, representations from language modeling and image classification pretraining do not encode affordance information as richly as interaction pretraining.

6.1 Data

To test our claim that interaction pretraining encodes more robust object affordance information, we first need datasets for each modality containing representations of each object type in our interaction environment.

Text Data We collect sentences from the Wikitext-103 corpus (Merity et al., 2017) that mention an object from our dataset. We reduce our set of objects into a set of 18 common labels that accurately describe each class of object. For example, `Bottle1` and `Bottle2` both map to `bottle` so that we can find mentions in the corpus. We run each sentence through a pretrained BERT-base model (Devlin et al., 2019) and average the contextualized word embeddings for each WordPiece token that corresponds to a mention of the object of interest (see Figure 6). In total, we collect 38,969 sentences for an average of 2,165 sentences per object.

Vision Data To keep the information in the visual domain consistent with that of our trained model, we use images of the objects from the Unity environment as our vision data. Images of objects are taken at random angles and positions in the environment. In all cases, the objects are in the center of the frame. We preprocess the images in the same way as for the multiview images, but with the addition of RandAugment (Cubuk et al., 2019) to add more variation to the training data. We run these images through Resnet-34, and extract 512-dimensional encodings from the penultimate layer.

Interaction Data To test the model's capacity for generalization to object types that were never encountered during training, we use U^{mv}_{enc}, the unseen object encodings in the Multiview model as our interaction inputs.

6.2 Affordance Prediction Model

Similar to our probing experiments, we want to determine if encodings of objects contain physical affordance information. However, we change the task slightly to account for the inherent fuzziness of labeling objects for affordances. Although a coin could roll if thrown at exactly the right angle, it is much more likely for a soccer ball to roll. To account for this, we adjust the task to express preference between encodings of two different objects, indicating which one is *more likely* to afford an action. For each modality m ∈{text, vision, interaction} and each affordance $a \in A$ we sample from the pretrained model corresponding to m one encoding that is positively labeled for a and one which is negatively labeled. We train a simple multilayer perceptron f^m_a to output a likelihood of affording the action Each model is trained using a margin ranking loss (Equation 3) with margin $m = .25$ to rank the positive example higher than the negative one.

$$L(x1, x2, y) = max(0, -y * (x1 - x2) + m) \quad (3)$$

We believe this is a fairer evaluation of the data points that we have for the text and vision components, and allows us to train on more combinations of examples.

6.3 Text and Vision Baseline Results

We find that grounding to object interactions allows a model to encode intuitive affordance information much more saliently than models that are trained without this kind of grounding. The networks trained on the interaction data, as encoded by the Multiview model, attain the highest accuracy on the majority of affordances, determined by the number of positive-negative pairs the model ranked correctly. Our results in Table 2 show that this performance difference is considerable. We

Model	Slide	Roll	Stack	Contain	W-Grasp	Bounce
Text	43.0	24.5	74.3	63.4	**86.6**	57.0
Vision	57.1	73.3	68.5	**68.3**	65.4	74.3
Interaction (ours)	**76.1**	**77.5**	**78.0**	51.6	60.3	**85.5**
<Text, Vision>	70.1	48.7	79.1	66.0	76.7	72.2
<Text, Inter.>	**70.9**	47.3	65.2	56.7	**80.8**	**79.6**
<Text, Vision, Inter.>	70.7	**53.1**	**80.6**	**69.5**	66.3	75.2

Table 2: Accuracy for each affordance prediction model for each modality. Interaction based object representations perform best in all cases except for those affordances for which our model had the weakest signal to learn from. Concatenation of vectors from multiple modalities shows improvement in some cases.

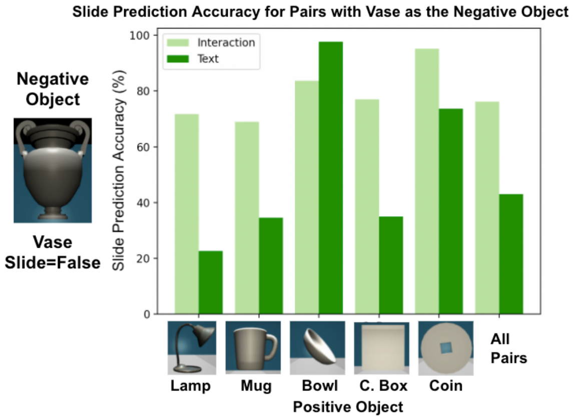


Figure 7: The affordance prediction model picks between two objects which is more likely to afford an action (in this case, `slide`). Text representations are highly inconsistent and performance depends on the object pair.

find that the biggest gains are in the `slide`, `roll`, and `bounce` actions. Direct observation of these actions is inherently missing from vision and text pretraining, but are clearly apparent in object trajectories. As evidenced by previous experiments, our pretraining setup does not have positive inductive bias for learning what “containing” or “grasping” are, and subsequently yields poor performance for these affordances.

We tend to see a much larger variance in performance in the purely text-based model, depending on the object pair used. As an example, let us inspect interaction and text models on `slide` prediction, when pairing `vase` negatives to various positive object types. We can observe that text representations do not consistently encode affordance information, while interaction pretraining is robust to all unseen object types. Our results are shown in Figure 7. Even though the BERT model has seen these words before, its contextualized text embeddings are either only encoding affordance information some of the time, or not encoding affordance information at all and relying on learning some heuristic difference between the two objects. Contrasting this, our model yields much more consistent high-performing results even though the objects are outside of the training distribution.

The knowledge encoded during interaction pretraining has the potential to improve physical reasoning in existing pretrained language models. Motivated by this possibility, we explore concatenating text embeddings (768d) with our interaction representations (100d) and retrain the models. We compare these results with text concatenated with vision representations (512d) as well as all three together as shown in Table 2. Although pure interaction models perform the best overall, we see some improvements in the `stack` and `contain` affordances. More work is needed in how to effectively combine these sources of knowledge.

7 Conclusion

This paper proposes an interaction-based self-supervised pretraining scheme for learning object trajectories from observations of interactions in a 3D virtual environment. We show that object affordance information can be encoded in the intermediate representations of our model more robustly than in those from models pretrained for either language modeling or image classification. Our model differs from those of previous works in that it learns rich representations from raw interactions, allowing it to generalize to unseen object types while requiring very little preprocessing and no human annotation. The effectiveness of this approach encourages future follow-up research into the optimal integration of interaction-based pretraining into language models to improve physical reasoning performance in downstream applications.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks.

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 W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. 2016. Interaction networks for learning about objects, relations and physics. *ArXiv*, abs/1612.00222.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

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.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Yue Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph P. Turian. 2020. Experience grounds language. In *EMNLP*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*. ArXiv: 2005.14165.

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.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2019. Randaugment: Practical automated data augmentation with a reduced search space.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Thanh-Toan Do, Anh Nguyen, and Ian Reid. 2018. Affordancenet: An end-to-end deep learning approach for object affordance detection.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense?

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.

Saurabh Gupta and Jitendra Malik. 2015. Visual semantic role labeling.

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. 2015. Deep residual learning for image recognition.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure.

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

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

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843.

William Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand?

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.

Thao Nguyen, Nakul Gopalan, Roma Patel, Matt Corsaro, Ellie Pavlick, and Stefanie Tellex. 2020. Robot object retrieval with contextual natural language queries.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv:1910.10683 [cs, stat]*. ArXiv: 1910.10683 version: 1.

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.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Sara Veldhoen, Dieuwke Hupkes, and Willem Zuidema. 2016. Diagnostic classifiers: revealing how neural networks process hierarchical structure.

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 Yun, Chen Sun, and Ellie Pavlick. 2021. Does vision-and-language pretraining improve lexical grounding?

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*.

A Appendix A

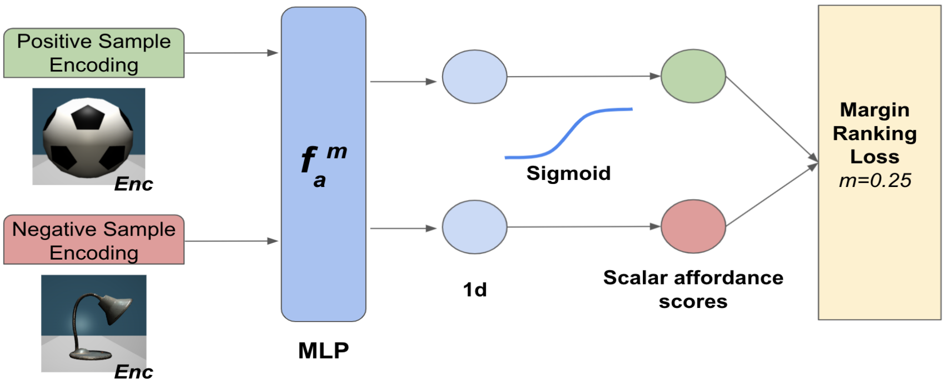


Figure 8: The affordance prediction architecture for comparing interaction pretraining with vision and text representations. For each modality m (e.g., vision) and each affordance a (e.g., `roll`) we train a network that learns to predict which of two objects is more likely to afford a. Each sample is some representation of an object that either does or does not afford a.

Object	**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 3: All objects in the dataset and their associated affordances

Affordance	Number of Objects
Slide	22
Roll	23
Stack	17
Contain	8
Wrap-grasp	13
Bounce	7

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