Modeling Verb Meaning with Trajectories

by

Dylan Ebert

B.S., University of Texas at Arlington, 2017

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2023
This dissertation by Dylan Ebert is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date____________________

Ellie Pavlick, Advisor

Recommended to the Graduate Council

4/26/2023
Date____________________

Chen Sun, Reader

Date 04/26/2023

George Konidaris, Reader

Date 04/27/2023

Yonatan Bisk, Reader

Approved by the Graduate Council

Date____________________

Thomas A. Lewis, Dean of the Graduate School
Curriculum Vitae

Name: Dylan Ebert

Education: Dylan Ebert obtained his Bachelor of Science in Computer Science from the University of Texas at Arlington in 2017. He then pursued a Master of Science in Computer Science at Brown University from 2017 to 2019, encompassed by a Doctor of Philosophy in Computer Science, also at Brown University, from 2017 to 2023.

Work: Following his Ph.D., Dylan joined Hugging Face as a Developer Advocate.
I would like to express gratitude to my advisor, Dr. Ellie Pavlick, for her guidance, support, and encouragement throughout the research journey. Her feedback and help have been invaluable to my growth as a researcher and onward in my career.

I am also grateful to my committee members, Dr. Chen Sun, Dr. George Konidaris, and Dr. Yonatan Bisk, for their suggestions and constructive criticism of that helped me improve and refine my work. I would also like to thank all members of the LUNAR lab at Brown University for their camaraderie and support.

Finally, thank you to my friends and family for their unwavering support and encouragement.
## Contents

1 Introduction 1

2 Related Work 8

2.1 NLP  
2.1.1 Environment Data Collection  18
2.1.2 Language Data Collection  20
2.1.3 Comparison to Child Directed Speech  22
2.2 Computer Vision  11
2.3 Robotics  13

3 A Visuospatial Dataset for Naturalistic Verb Learning 16

3.1 Data  17
3.1.1 Environment Data Collection  18
3.1.2 Language Data Collection  20
3.1.3 Comparison to Child Directed Speech  22
3.2 Experiments  27
3.2.1 Preprocessing  27
3.2.2 Models  28
3.2.3 Evaluation  30
3.2.4 Results and Analysis  31
3.3 Discussion  34

4 Do Trajectories Encode Verb Meaning? 35

4.1 Dataset  35
CONTENTS

4.1.1 Overview ............................................ 35
4.1.2 Data Generation and Terminology .................... 36
4.1.3 Crowdsourced Annotation ............................ 39
4.2 Dataset Analysis .................................... 41
4.2.1 Agreement ........................................ 42
4.2.2 Co-occurrence .................................... 42
4.3 Experiments ......................................... 43
4.3.1 Experimental Setup ................................ 44
4.3.2 Approaches ....................................... 45
4.4 Results ............................................... 47
4.5 Additional Experiments ................................ 48
4.6 Discussion and Conclusion ............................ 50

5 Comparing Trajectory and Vision Models for Verb Representation 51
5.1 Experimental Design .................................. 52
5.1.1 Models ............................................. 52
5.1.2 Features ........................................... 53
5.2 Results ................................................ 55
5.2.1 Main Findings ..................................... 55
5.2.2 Analysis ........................................... 55
5.3 Discussion & Limitations .............................. 57

6 Discussion ............................................. 59
6.1 Conclusion ............................................ 63
List of Tables

3.1 Object features recorded during data collection. Object appearance does not vary across frames; img_url does not vary across objects. All other features vary across object and frame. . . . . . . . . . . . . . . . . . 20

3.2 Estimates of training signal quality for nouns and verbs. N is the number of times the word occurs in the training data. P is the precision—given a 5 second clip in which the word is used, how often does the clip depict an instance of the word? Note that the verb “go” is an outlier, since it appears most often as “going to”. 26

3.3 Precision of each method with 95% bootstrapped CI. “Soft” means a prediction is correct as long as one annotator considers it to be so; “strict” means prediction is only considered correct if both annotators agree that it is correct. . . . . . . . . . . . . . . . . . 32

3.4 Analysis of model precision broken down by verb. Top-level columns are the unsupervised CNN, unsupervised obj model, and supervised obj model. For each, N is the number of times the model predicts that verb. Precision is the proportion of the time that prediction was correct. . . . . . . . . . . . . . . . . . 32

4.1 Mean Average Precision (mAP) for each approach. The pre-trained approaches outperform others on verb classification. . . . . . . . . . . . . . . . . . . . 46
5.1 Mean Average Precision (mAP) scores for each model on the verb classification task, reported with both micro and macro averaging. 95% confidence intervals are reported beside each condition. 52

5.2 Mean Average Precision (mAP) scores with 95% confidence intervals for fall and roll, which exhibit significant performance difference for the 3D Trajectory and 2D image models. For all other verbs, there was no significant difference between models. 56

5.3 Mean Squared Error (MSE) for each model on a 3D object position regression task. 57
List of Figures

2.1 Does a visual model predict “washing dishes” because of visual cues, or understanding the verb “wash”? .......................... 12

3.1 Screenshots of a person picking up a banana in each of the two kitchen aesthetics. ...................................................... 19

3.2 Comparison of word category and lexical distributions. Lexical item frequencies labels are $\times1000$. Distributions are over the most frequent categories/words according to the Brent-Siskind corpus of child-directed speech. ............................... 24

3.3 Example clips, subsampled to 6 frames. (b) is (a)’s nearest-neighbor using the Object-Based model. In each of these clips, the participant picks up an object with their right hand. (d) is (c)’s nearest-neighbor using the CNN. In each, the participant is washing dishes in a similar looking sink. .............................. 31

4.1 The *Simulated Spatial Dataset* consists of procedurally generated motion data of a virtual agent interacting with an object. In this sequence the agent (red sphere) pushes the object (blue sphere). At t=0 and t=1, the agent approaches the ball. Then, in t=2 and t=3, the agent pushes to ball. Finally, at t=4, the ball is rolling away from the agent. ................................. 36
4.2 Crowd annotation task. Crowdworkers make binary judgments on whether the verb applies to the clip. In this example, the worker is asked Does the object bounce? about the 1.5s video clip on the left. 37

4.3 Crowd annotation agreement by verb. Workers agree most on when verbs of gravity occur, such as fall, drop, bounce, and least on when verbs of rotation occur, i.e. turn, spin, tip. 39

4.4 Co-occurrence of different verbs for the same clip. For example, the majority of clips labeled bump are also labeled slap. 41

4.5 The pretraining setup. During pretraining, the model learns to encode and represent input timesteps for time-series prediction. To evaluate these learned representations, a perceptron probe is trained on the lstm outputs, without propagating gradients to the pretrained model. 44

4.6 Comparison of each approach by verb. The green-blue bars show average precision for each approach. For comparison, crowdworker annotation agreement is shown in red. 45

4.7 Comparison of using absolute trajectory inputs compared to relative trajectory inputs with the finetuned approach. The trajectory inputs (in blue) performs better overall, though the relative inputs (in orange) perform notably better for the verb “carry”. 49

5.1 The Simulated Spatial Dataset consists of procedurally generated motion data of a virtual agent interacting with an object. A) shows the camera view as the object (in red) as it falls off the counter. B) shows the corresponding 2D trajectory data, while C) shows the corresponding 3D trajectory data. 51
5.2 True and predicted frames during image-based pre-training. The frame at $t = 0$ is the final input frame, where at $t = 1..3$ are the real and predicted future frames. The predicted images suffer from mode collapse, and consistently reproduce the same reconstruction. ................................................................. 54

6.1 Screenshots of a person picking up a banana in each of the two kitchen aesthetics. ................................................................. 60

6.2 The Simulated Spatial Dataset consists of procedurally generated motion data of a virtual agent interacting with an object. In this sequence the agent (red sphere) pushes the object (blue sphere). At $t=0$ and $t=1$, the agent approaches the ball. Then, in $t=2$ and $t=3$, the agent pushes to ball. Finally, at $t=4$, the ball is rolling away from the agent. ................................................................. 60
Chapter 1

Introduction

The Friendly Robot. Imagine you’re sitting at a table. You want to check your phone, but it’s all the way across the room. Fortunately, there is a friendly robot nearby. You ask: “Hey robot, can you toss me my phone?” The robot parses these words to a sequence of actions. It locates the phone, navigates over to it, and picks it up. So far, so good. But then, the robot throws the phone at a high velocity. It hits the wall and shatters. Your phone is broken. What went wrong?

In this example, the robot didn’t understand the difference between “toss” and “throw”. This work focuses on the language understanding aspect of the problem. That is, building language representations that encode physical dynamics in verb meaning, such as the nuanced difference between “toss” and “throw”.

Verb Understanding. Verbs are an essential component of natural language understanding. They characterize actions and events, and provide information about the relationships between entities. However, verbs generally receive less attention in the literature than nouns or adjectives (Baroni and Zamparelli, 2010; Boleda and Herbelot, 2016; Cascante-Bonilla et al., 2023). Thus, it isn’t
always clear the extent to which state-of-the-art models truly understand verb meaning.

Verb understanding can range from physical interactions (i.e. “I pick up the ball”) to abstract relationships (i.e. “I miss avocados”). The work in this thesis focuses on the former. Understanding verbs that describe physical dynamics are essential to performing embodied tasks like instruction following. However, state-of-the-art approaches, like BERT (Devlin et al., 2018) and GPT (Radford et al., 2018) are based on text alone, thus often fail to capture verb meaning to an extent that allows agents to succeed at embodied tasks. These models often have difficulty understanding the complex interactions between objects and actions, and nuances of verb meaning in specific contexts (Bisk et al., 2020a; Forbes et al., 2019).

**Verb Representation.** This section provides a brief overview of the context in which the work in this thesis is situated. A more thorough overview of related work will be presented in Chapter 2.

The state-of-the-art approaches to natural language processing use large language models (LLMs) (Devlin et al., 2018; Radford et al., 2018). These are typically large transformer neural networks that are trained to predict masked language tokens in massive text corpora (Vaswani et al., 2017). This has been demonstrated to solve a wide variety of language tasks, such as sentiment analysis (Zhang et al., 2018a), named entity recognition (Devlin et al., 2018), machine translation (Edunov et al., 2018), and question answering (Zhu et al., 2021). Such models even succeed at many measures of verb understanding. Take, for example, this response from ChatGPT (OpenAI, 2023) when prompted: “What is the difference between toss and throw?”

Toss and throw are similar actions, but “toss” implies a lighter, gentler motion while “throw” implies a stronger, more forceful motion.
This appears to demonstrate a clear understanding of the physical difference between “toss” and “throw”. However, this response is based on statistical patterns in language, not the physical dynamics of how objects interact in the world. In this thesis, I will focus on physically grounded verb representations and how language representations may capture underlying verb meaning.

**Historical Context.** Verb representation has been an important area of research in NLP for decades, with early work focusing on hand-written rules and symbolic structures, such as Levin Verb Classes (Levin, 1993) and FrameNet frames (Baker and Ruppenhofer, 2002), wherein meaning is derived from logical inference and series of manipulations of symbolic representations.

In modern times, distributional models, which define the meaning of words in terms of the context in which they occur (Harris, 1954; Firth, 1957), have become the default mode of verb representation. This approach has grown in popularity in recent years, taking advantage of major advances in empirical performance brought about by deep learning (LeCun et al., 2015). This approach has scaled exponentially, taking advantage of innovations such as the transformer (Vaswani et al., 2017) architecture to scale to hundreds of billions of parameters.

These large language models (LLMs) have solved\(^1\) long-standing NLP tasks such as question answering, common sense reasoning, and sentiment analysis (OpenAI, 2023). Despite this, language understanding in general is far from solved (Mahowald et al., 2023; Marcus, 2018), though it isn’t clear what exactly are the limits of these models, or where to go next.

**Connections to Robotics.** A longstanding goal of robotics research is to have “general-purpose” robots that can complete a wide variety of tasks, from

---

\(^1\)Several models are shown to exceed human performance on SuperGLUE benchmark tasks (Wang et al., 2019).
driving, to personal assistance, to performing everyday household tasks. While
the field of robotics is primarily concerned with learning policies to carry out
these tasks, a critical part of the equation is the ability of the robot to under-
stand language.

A robot that is truly general-purpose needs to be able to communicate effect-
ively with humans, which involves understanding language and applying that
understanding in a complex physical environment. However, contrary to text-
only models in NLP, there is no general-purpose robotics model. Tasks are
typically addressed case-by-case, and language is typically mapped to symbolic
representations appropriate to the task at hand.

As large language models close the door on many long-standing NLP tasks,
the goal-post for natural language understanding moves toward an embodied
understanding of language that intersects heavily with these open problems in
robotics. While the scope of this thesis does not include robotics modeling, these
problems motivate the research direction of this thesis. That is, learning verb
representations that capture the underlying physical dynamics of verb meaning.
This work paves the way for general-purpose models that address open problems
at the intersection of NLP, vision, and robotics research.

Grounding. Despite the fact that LLMs have been a huge source of progress
in NLP, they differ significantly from how humans learn language. Understanding
this difference may be key to building models that solve the open embodied
language understanding problems of the future.

One of the most salient ways in which LLMs differ from human language
learning is that human language learning is grounded to the non-linguistic world.
Infants are shown to understand the world in terms of continuously-moving
objects, prior to language development (Spelke et al., 1995). By contrast, LLMs
learn language from text alone, with no connection to the non-linguistic world.
A growing body of research attempts to address this gap by training grounded language models, which ground text to the world in the form of vision, i.e. images and videos (Kiros et al., 2014; Sun et al., 2019; Kiros et al., 2015; Vinyals et al., 2015). However, while these are typically seen as promising research directions, they have yet to demonstrate language understanding that exceeds that of text-only LLMs. It has been argued that these models only shallowly align language and vision, failing to ground language in a meaningful way (Yun et al., 2021). For instance, image-captioning models like Show, Attend and Tell (Vinyals et al., 2015) have been found to generate plausible but incorrect captions, indicating a lack of deeper understanding of the visual scene.

These shortcomings are especially relevant in the context of verb representation. These large language-and-vision models are argued to rely on spurious correlations between verbs and their frequent context, rather than the underlying verb meaning (Zhu et al., 2020). For example, “chop” may correlate strongly with a kitchen environment, allowing models to successfully caption visual depictions of “chop”, despite failing to capture the underlying physical dynamics.

**Trajectories.** Given the limitations of existing grounded language approaches to verb representation, this work looks toward an alternate approach to representing the world - *trajectories*. Trajectories are defined in this thesis as the position and rotation of objects over time. This appeals to the intuition that encoding the underlying physical meaning of verbs like “tumble” requires a rich understanding of 3D space, which may not be captured by 2D visual modalities alone.

Representing verbs in terms of trajectories is not a novel idea. A similar idea has been proposed in computational semantics (Pustejovsky and Krishnaswamy, 2016), which demonstrates that trajectories can represent symbolic aspects of verb meaning, including path and manner. However, trajectory data has not
been explored as a modality for language representation using modern deep learning-based approaches.

One of the primary explanations for the lack of trajectory-grounding research is lack of data. Modern grounded language approaches rely on massive amounts of parallel text and image/video data. Trajectory data, on the other hand, is very sparse and difficult to obtain, requiring expensive and cumbersome sensors attached to objects (Montemerlo and Thrun, 2003). However, this is beginning to change.

Growing interest in embodied AI research, alongside advancements in simulation technology, has given rise to a number of simulated environments that are increasingly representative of the real world (Puig et al., 2018; Juliani et al., 2018; Gan et al., 2020). These environments allow rich 3D data, including trajectories, to be collected affordably and at scale. While large embodied datasets have not yet come to fruition, their imminence supports the immediate relevance of the work in this thesis.

**Thesis Statement.** This thesis tests the hypothesis that trajectory-based representations are more capable than conventional vision-based representations at capturing physical dynamics of verb meaning. This is done by building annotated datasets of rich 3D interactions, and conducting controlled experiments that evaluate the capacity of trajectory-based models to encode verb meaning. This thesis finds that while trajectory-based models successfully encode nuanced aspects of verb meaning, such as the difference between “slide” and “roll”, a conventional video-based model is able to encode these aspects comparably well, suggesting that 2D visual data alone may be sufficient to encode verb meaning. However, this thesis does not close the book on the potential of grounding language to rich 3D representations: there is always the possibility of the picture changing with increased scale and verb complexity. Nonetheless,
this thesis takes an important early step in addressing the challenging problem of language learning in embodied environments, and points toward a need for further exploration in how to build models that truly capture the underlying physical dynamics of verbs.

**Summary of Contributions.** The main contributions of this thesis are organized into three chapters.

Chapter 3 introduces the New Brown Corpus, a parallel corpus of spontaneous speech, vision, and trajectory data collected from 18 participants performing household tasks in virtual reality kitchen environments. This chapter presents preliminary results that compare trajectory and vision modalities for verb representation, but finds that more controlled experiments are necessary to evaluate the capacity of trajectory-based representations.

Chapter 4 introduces the Simulated Spatial Dataset, a procedurally-generated dataset of parallel vision and trajectory data of a virtual agent interacting with an object. This work demonstrates that a self-supervised trajectory model is able to capture the physical dynamics of verb meaning.

Chapter 5 compares trajectory-based and video-based verb representations in controlled experiments. The results show that video-based models are able to encode verb meaning comparably well to trajectory-based models, suggesting that 2D visual data alone may be sufficient to capture spatial dynamics of verb meaning.
Chapter 2

Related Work

This thesis, which takes an NLP-centric look at verb representation, is situated at the intersection of NLP, vision, and robotics research. This work is supported by an interdisciplinary body of prior work, which trends toward an increasingly unified set of goals that blurs the lines between these diverse AI fields.

This related work chapter of this thesis is organized into three sections: 2.1 NLP, 2.2 Computer Vision, and 2.3 Robotics. Each section provides an overview of relevant prior work that supports the central aim of this thesis: to build language representations that capture physical dynamics of verb meaning.

2.1 NLP

The field of Natural Language Processing (NLP) has a long history of verb representation research, with roots in linguistics and cognitive science. Verb representation research in NLP has taken various forms, from early work based in classical linguistics and logical inference, to more recent work using modern deep learning-based approaches.
Traditional Approaches to Verb Representation. Early work in verb representation takes a classical linguistics-inspired approach, where verbs are represented in terms of discrete symbols, and meaning is derived through logical inference. One notable example of this is Levin Verb Classes (Levin, 1993), which group verbs based on syntactic and semantic properties. For example, the “Break Verbs” group includes verb tokens that involve the change in material of an entity (Kipper et al., 2008).

Another traditional approach is VerbNet (Schuler, 2005), a hierarchical verb lexicon that builds upon and extends Levin Verb Classes. VerbNet aims to characterize the connection between semantics and syntax by organizing verbs into classes based on syntactic and semantic similarities. For instance, the verb “break” would be a member of the “Break-45.1” class, which includes verbs that describe something breaking into parts. This class includes the verbs break, chip, crack, crash, crush, fracture, shatter, smash, snap, splinter, split, and tear.

FrameNet (Baker et al., 1998) is another traditional approach that represents meaning in terms of semantic frames. FrameNet frames, in contrast to VerbNet verb classes, place more emphasis on the roles of the participants in an event, as well as the relationships between different frames. Each frame includes frame elements, which represent the participants and their roles in the event. For the verb “break”, these elements include the agent responsible for the change, the theme of the change, the direction of the change, the degree or extent of the change, and the scale on which the change is occurring (e.g. temperature, speed).

Finally, WordNet (Fellbaum, 2010) groups words according to sets of synonyms called synsets. These synsets represent distinct concepts which are linked to other synsets by relationships such as hyponym-hypernym relationships, i.e. all animals are dogs but not all dogs are animals, as well as troponyms and
entailments for verbs. The primary goal of WordNet is to provide a structured representation of word semantics, including verbs.

Although formal symbolic approaches have largely been replaced by modern deep learning-based approaches in downstream applications, there is ongoing research that employs traditional approaches to verb representation. For example, recent work focuses on trajectories as a means of encoding the physical dynamics of verbs in a symbolic semantic framework (Pustejovsky and Krishnaswamy, 2016, 2019). While these studies demonstrate the semantic potential of trajectories, they have not been employed at the scale required for downstream tasks. This thesis aims to contribute to the field by investigating how modern deep learning-based models can capture the physical dynamics of verb meaning.

### Distributional Models of Verb Representation.

Recent work in verb representation is dominated by distributional approaches to language understanding. These models rely on Large Language Models (LLMs) that represent verbs in terms of vectors of numbers and derive meaning from statistical relationships between language tokens. One of the early incarnations of this generation is BERT (Devlin et al., 2018), which is trained on a self-supervised masked token prediction task, where some tokens in the input are masked out and the model predicts these masked tokens. Since BERT, models have continued to grow by massively in scale, now consisting of models such as GPT-3 (Brown et al., 2020) that are trained on massive amount of language data.

Despite the success of these models in solving various NLP tasks, it has been called into question whether they truly reason about the world, or simply exploit statistical patterns. This was highlighted in the PIQA benchmark (Bisk et al., 2020b), which evaluated whether LLMs can reason physically about the world with questions such as: “What is the best way to make an outdoor pillow? Blow
into a tin can and tie with a rubber band; or, blow into a trash bag and tie with a rubber band.” and found that humans significantly outperform state-of-the-art models.

This difference in reasoning might be attributed to a lack of grounding. “We argue that the linguistic modeling task, because it only uses form as training data, cannot in principle lead to learning of meaning.” (Bender and Koller, 2020) Unlike humans, who reason about objects and their properties prior to language development (Saxton, 2017; Spelke and Kinzler, 2007), LLMs only learn from statistical patterns in text, without any connection to the non-linguistic world. Recent work has argued for the importance of grounding to approach human-like language understanding (Bommasani et al., 2021; Zellers et al., 2021a; Tan and Bansal, 2020; Sun et al., 2019). However, as stated in Chapter 1, these models have yet to demonstrate language understanding that exceeds that of text-only LLMs.

2.2 Computer Vision

The goal of vision research is to make machines understand images and videos. Vision, unlike language, is an extremely information-dense signal, with a single 1080p image containing over 2 million pixels. This is tractable when using lower-resolution images for tasks like image classification (Deng et al., 2009), but becomes prohibitively complex for tasks that require multiple frames, such as dynamics modeling. To overcome this limitation, state-of-the-art vision-and-language models (Sun et al., 2019; Kim et al., 2021) use a transformer architecture that embeds patches of images with text in a shared latent space. These models are conceptually similar to LLMs, but incorporate vision as an additional modality to solve language-and-vision tasks like question answering.

Video-language pre-training has recently gained increased attention, with
Figure 2.1: Does a visual model predict “washing dishes” because of visual cues, or understanding the verb “wash”?

models such as MERLOT (Zellers et al., 2021b), Flamingo (Alayrac et al., 2022), and VIOLET (Fu et al., 2022) integrating vision and language to develop richer understanding. However, despite recent advances in video-language understanding (Buch et al., 2022), these models still face challenges in grounding language understand beyond the shallow alignment of language and vision (Yun et al., 2021).

In the context of verb representation, vision models are susceptible to building verb representations that rely on superficial visual cues, rather than the underlying physical dynamics of verbs. For instance, a vision model may classify an image of someone washing dishes as an instance of the verb “wash” as in Figure 2.1, but this may be because of the visual context of the kitchen sink and hand, rather than a human-like understanding of the verb “wash”.

Recent work aims to address this issue by building and evaluating models
that explicitly model physical scene understanding (Li et al., 2020; Bakhtin et al., 2019). While these early efforts have not been applied in the context of language understanding, they share the common goal of building models that capture the physical dynamics of the world. However, there is still a significant amount of work to be done in terms of modeling physical dynamics on a large scale and integrating them with language models.

Self-supervised learning approaches, such as Time-Contrastive Networks (Ser-\textit{manet et al., 2018}) and ego-motion based representations (Jayaraman and Grau-\textit{man, 2016}) offer a promising direction for learning grounded representations without the need for large labeled datasets. Additionally, advancement such as VideoMAE (Tong et al., 2022) have shown the potential of masked autoencoders in data-efficient self-supervised video pretraining.

2.3 Robotics

The goal of robotics research is to develop robots that can handle a wide variety of real-world tasks, including construction (Bock, 2007), driving (Badue et al., 2021), personal aid (Javdani et al., 2018), and common household chores (Shrid-\textit{har et al., 2020}). The ultimate goal of robotics is to create a general-purpose robot that can understand the world and adapt to a wide variety of settings and tasks. This goal encapsulates NLP, requiring effective human-robot communication through language. However, current robotics research tends to focus on specific tasks independently, lacking a general understanding of language and the world. This is with noteworthy exceptions that work toward building a general understanding of language in robotics (Tellex et al., 2011; Artzi and Zettlemoyer, 2013; Chen and Mooney, 2011).

The robotics community shares many common goals with the work in this thesis. Consider the example of asking a robot to “put a washed, sliced ap-
ple on a plate.” A general-purpose robot would need to have a very robust understanding of “wash” and “slice” that adapts to diverse environments. Recent work formalizes these goals by setting up a benchmark where robots are instructed to perform common household tasks in a virtual environment (Shridhar et al., 2020). However, results are far below human performance, even in this toy virtual setting where actions are abstracted to symbolic state changes. A possible solution is to use a large pretrained robotics model, analogous to large transformer models in NLP and vision. However, there is not an abundant and normalized form of data that can be used to train general-purpose robotics models. Thus, there is a significant gap between the current state of robotics and the opportunity of general-purpose robots. This thesis aims to address this gap and contribute to a model of verb understanding that takes a step closer to developing a general-purpose robotics model.

**Simulated Environments.** The shared long-term goals of robotics, vision, and NLP have contributed to growing interest in *embodied learning*, research that aims to build models that understand and interact with the physical world. This interest has resulted in the development of simulated environments that assist with embodied research, including opportunities for data collection and model evaluation at a large scale. These include:

- VirtualHome (Puig et al., 2018)
- AI2THOR (Juliani et al., 2018)
- ThreeDWorld (Gan et al., 2020)
- Habitat (Savva et al., 2019)

The goal of these environments aligns with the work in this thesis, which also introduces two simulated environments. However, these environments are
limited in their ability to accurately model the physical dynamics of verbs. For instance, the “pick up” action in AI2THOR is abstracted to an instantaneous event that teleports the object to the agent, making it unsuitable for modeling verb dynamics. Despite these limitations, these environments represent an early step in addressing the challenges of embodied learning and general-purpose robotics. The work in this thesis builds upon this progress by introducing two new simulated environments, the first of which will be detailed in the following chapter.
Chapter 3

A Visuospatial Dataset for Naturalistic Verb Learning

This chapter describes work that takes an aspirational look at grounded distributional semantics models, based on the type of situated contexts and weak supervision from which children are able to learn much of their early vocabulary. This approach is motivated by the assumption that building computational models which emulate human language processing is in itself a worthwhile endeavor, which can yield both scientific (Potts, 2019) and engineering (Linzen, 2020) advances in NLP. Therefore, an effort is made to develop a dataset that better reflects both the advantages and the challenges of humans’ naturalistic learning environments. For example, unlike most vision-and-language models, children likely have the advantage of access to symbolic representations of objects and their physics prior to beginning word learning (Spelke and Kinzler, 2007). However, also unlike NLP models, which are typically trained on image or video captions with strong signal, children’s language input is highly unstructured and the content is often hard to predict given only the grounded context (Gillette et al., 1999).

This work presents two main contributions. First (§3.1), the creation of
the New Brown Corpus dataset, which consists of 18K words of spontaneous speech, egocentric vision, and trajectory data as participants perform everyday tasks in virtual reality kitchen environments. The data collection protocol is designed to solicit naturalistic speech that reflects data received in child language development (Frank et al., 2017). Second (§3.2), this corpus is used to compare trajectory and vision-based verb representations. The findings highlight the challenges of verb learning, with both models performing only marginally above chance, pointing toward the need for further research.

3.1 Data

The goal of the data collection is to facilitate research on grounded distributional semantics models using data that better resembles the type of input young children receive on a regular basis during language development. This is an ambitious goal, if not unattainable. Therefore, the focus is on a few aspects of children’s language learning environment that are lacking from typical grounded language datasets and that can be emulated well given current technology: 1) spontaneous speech (i.e. as opposed to contrived image or video captions) and 2) rich information about the 3D world (i.e. physical models of the environment as opposed to flat pixels).

A virtual reality (VR) environment is developed to collect this data in a controlled manner. The environment data is described in Section 3.1.1 and the language data is described in Section 3.1.2. The collection process results in a corpus of 152 minutes of concurrent video, audio, and ground-truth environment information, totaling 18K words across 18 unique speakers performing six distinct tasks each. The data is currently available for download in json format at https://github.com/dylanebert/nbc. The code needed

---

1 Our university namesake, plus paying homage to important Brown corpora in both NLP (Francis and Kucera, 1979) and Child Language Acquisition (Brown, 1973).
to implement the described environment and data recording is available at
https://github.com/dylanebert/nbc_unity_scripts.

3.1.1 Environment Data Collection

Environment Construction

A simple kitchen environment is implemented in Unity with SteamVR and ex-
periments are conducted using an HTC Vive headset. The use of VR was
chosen over alternative interfaces for simulated interactions (e.g. keyboard or
mouse control) since VR enables participants to use their usual hand and arm
motions and to narrate in real time, leading to more natural speech and more
faithful simulations of the actions they are asked to perform.

Six different kitchen environments were designed, using two different vi-
usal aesthetics (Fig. 3.1) with three floorplans each. This variation was im-
plemented to test, for example, that learned representations are not overfit to
specific pixel configurations or to exact hand positions that are dependent on
the training environment(s) (e.g. “being in the northwest corner of the kitchen”
as opposed to “being near the sink”). Each kitchen contains at least 20 com-
mon objects (not every kitchen contains every object). These objects were
selected because they represent words with low average ages of acquisition (de-
scribed in detail in §3.1.2) and were available in different Unity packages and
thus could be included in the environment with different appearances. Across
all kitchens, the movable objects used are: Apple, Ball, Banana, Book, Bowl,
Cup, Fork, Knife, Lamp, Plant, Spoon, Toy1: Bear|Bunny, Toy2: Doll|Dinosaur,
Toy3: Truck|Plane. The participant’s hands and head are also included as mov-
able objects. The following immovable objects are also included: Cabinets,
Ceiling, Chair, Clock, Counter, Dishwasher, Door, Floor, Fridge, Microwave,
Oven, Pillar, Rug, Sink, Stove, Table, Trash Bin, Wall, Window.
The environments are constructed using a combination of Unity Asset Store assets and custom models. All paid assets come from two packs: 3DEverything Kitchen Collection 2 and Synty Studios Simple House Interiors, from the Unity asset store\(^2\). These packs account for the two visual styles. VR interaction is enabled using the SteamVR Unity plugin, available for free on the Unity asset store.

**Data Recording**

During data collection, the physical state of each object in the environment is recorded a rate of 90fps (frames per second). The physical state of the user’s head and hands are also rerecorded, thanks to the accurate motion capture provided by the Vive (Borges et al., 2018). The physical features of each object are recorded as described in Table 3.1. Audio data is also collected in parallel to spatial data, using the built-in microphone. The audio is later transcribed using Google’s Cloud Speech-to-Text API\(^3\). Word-level timestamps from the API allow matching words to trajectory frames. While spatial and audio data are

---

\(^2\)https://assetstore.unity.com/  
\(^3\)https://cloud.google.com/text-to-speech/
### Table 3.1: Object features recorded during data collection.

<table>
<thead>
<tr>
<th>Name (Type)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pos (xyz)</td>
<td>Absolute position of object center, computed using the <code>transform.position</code> property; equivalent to position relative to an arbitrary world origin, approximately in the center of the floor.</td>
</tr>
<tr>
<td>rot (xyzw)</td>
<td>Absolute rotation of object, computed using the <code>transform.rotation</code> property.</td>
</tr>
<tr>
<td>vel (xyz)</td>
<td>Absolute velocity of object center, computed using the <code>VelocityEstimator</code> class included with SteamVR.</td>
</tr>
<tr>
<td>relPos (xyz)</td>
<td>Position of object’s center relative to the person’s head, computed using Unity’s built-in <code>head.transform.TransformPoint(objectPosition)</code>.</td>
</tr>
<tr>
<td>relRot (xyzw)</td>
<td>Rotation of object relative to the person’s head, computed by applying the inverse of the head rotation to the object rotation.</td>
</tr>
<tr>
<td>relVel (xyz)</td>
<td>Velocity of the object’s center, from the frame of reference of the person’s head.</td>
</tr>
<tr>
<td>bound (xyz)</td>
<td>Distance from object’s center to the edge of bounding box.</td>
</tr>
<tr>
<td>inView (bool)</td>
<td>Whether or not the object was in the person’s field of view, computed using Unity’s <code>GeometryUtility</code> to compute if an object is in the Camera renderer bounds. This is based on the default camera’s 60 degree FOV, not the wide headset FOV. The head and hands are always considered <code>inView</code>.</td>
</tr>
<tr>
<td>img_url (img)</td>
<td>Snapshot of the person’s entire field of view as a 2D image. This is computed once per frame (as opposed to the above features which are computed once per object per frame).</td>
</tr>
</tbody>
</table>

Object appearance does not vary across frames; `img_url` does not vary across objects. All other features vary across object and frame.

Recorded in real-time, video recording is not, since this would introduce high computational overhead and drop frames. Instead the spatial data is iterated over and the playback is reconstructed rerendered frame-by-frame. This approach makes it possible to render from any perspective if needed, though the provided image data is only from the original first-person perspective.

### 3.1.2 Language Data Collection

The protocol is designed to solicit the use of vocabulary items that are known to be commonly acquired by children. 20 nouns, 20 verbs, and 20 prepositions/adjectives are selected which have low average ages of acquisition accord-
ing to Frank et al. (2017) and which can be easily operationalized within the VR environment (e.g. “apple”, “put (down)”, “red”). Six basic tasks are then chosen which the participants are instructed to carry out within the environment. These tasks are: set the table, eat lunch, wash dishes, play with toys, describe a given object, and clean up toys. The tasks are intended to solicit use of many of the target vocabulary items without explicitly instructing participants to use specific words, in order to avoid coached or stilted speech as much as possible. One exception is the “describe a given object” task, in which participants are asked to describe specific objects as though a child has just asked what the object is, e.g. “What’s a spoon?”. This task is used to ensure uniform coverage of vocabulary items across environments, so that good train/test splits can be constructed across differently appearing environments.

18 participants were recruited for the data collection. Participants were students and faculty members from multiple departments involved with language research. They were asked to perform each of the tasks, one by one, while narrating their actions as if they were a parent or babysitter speaking to a young child. An illustrative example of the language in the corpus is: “okay let’s pick up the ball and play with that will it bounce let’s see if we can bounce it exactly let’s let it drop off the edge yes it bounced the ball bounced pick it up again... ”. The full data can be browsed at [https://github.com/dylanebert/nbc](https://github.com/dylanebert/nbc).

The university IRB determined that the study design was not considered human subjects research. Participants were informed about the purpose of the study and provided signatures consenting to the recording and release of their anonymized data for research purposes.
3.1.3 Comparison to Child Directed Speech

The stated goal is to collect data that better reflects the distribution of language input a young child is likely to receive. Several corpus analyses are run to assess whether this goal is achieved.

Vocabulary Distribution

The distribution of vocabulary in the collected data is compared to that observed in the Brent-Siskind Corpus, a corpus of child-directed speech consisting of 16 English-speaking mothers speaking to their pre-verbal children. For reference, the vocabulary distributions of three existing corpora which could be used for training distributional semantics models are also shown: MSR-VTT, a large dataset of YouTube videos labeled with captions, Room2Room (R2R), a dataset for instruction following within a 3D virtual world, and a random sample of sentences from Wikipedia. The primary focus is on grounded language, making MSR and R2R the more relevant points of comparison as each contains language aligned with grounded semantic information (raw RGB video feed for MSR and video+structured navigation map for R2R). Wikipedia is included to demonstrate the type of web corpus that is ubiquitous in representation learning for NLP.

Figure 3.2 displays the token- and type-level frequency distributions for each of the five corpora over major word categories and individual lexical items. The corpora are preprocessed using the SpaCy 2.3.2 preprocessing pipeline with the \texttt{en-core-web-lg} model. The entire New Brown Corpus (NBC) and Brent Corpus are analyzed, while a random sample of 5K sentences from MSR, R2R, and Wikipedia are analyzed.

NBC closely mirrors the distribution of child-directed speech, with both NBC and the Brent corpus consisting primarily of verbs (around 23% when
computed at the token level), followed by pronouns (around 19%), followed by nouns at around 17%. MSR video caption corpus and Wikipedia both contain

(a) Token-Level Frequency of Word Categories

(b) Type-Level Frequency of Word Categories

(c) Token Frequency of Individual Verbs
Figure 3.2: Comparison of word category and lexical distributions. Lexical item frequencies labels are \( \times 1000 \). Distributions are over the most frequent categories/words according to the Brent-Siskind corpus of child-directed speech. Predominantly nouns (around 40\%) and the R2R instruction dataset contains equal proportions of nouns and verbs (around 33\% each). None of the baseline corpora have significant counts of pronouns.

In terms of specific vocabulary items, NBC contains decent coverage for many of the most frequent verbs observed in CDS, while the baseline corpora are dominated by a single verb each (“go” for R2R and “be” for MSR and Wikipedia). For nouns and adjectives, NBC also has better coverage of top-CDS words compared to the other corpora analyzed, although the difference is less noticeable and the lexical items in these categories are more topically
CHAPTER 3. A VISUOSPATIAL DATASET FOR NATURALISTIC VERB LEARNING

determined.

Word-Context Alignment

It is next investigated how well the language aligns with the salient objects and events in the context of its use. This property is important because it affects the strength of the “training signal” for a model attempting to learn linguistic meaning from distributional signals. Directly estimating the quality of the “training signal” available to children is challenging. However, experiments with the Human Simulation Paradigm (HSP) (Gillette et al., 1999; Piccin and Waxman, 2007) come close. In HSP, audio and video recordings of a child’s normal activities are collected, and adults view segments of videos and predict which words are said at given points in time. This technique estimates how predictable language is given only the grounded (non-linguistic) input to which a child has access. According to Gillette et al. (1999), nouns can be predicted with 45% accuracy, and verbs with 15% accuracy.

This provides an approximate point of comparison against which to benchmark the word-to-context alignment of the collected data. Rather than trying to guess the word given a video clip, instead a short (5 second) video clip is viewed alongside an uttered word and a binary judgement is made for whether or not the clip depicts an instance of the word: e.g., yes or no, does the clip depict an instance of “pick up”? This design is chosen over the HSP design since it provides a more interpretable measure of the quality of the training signal from the perspective of NLP and ML researchers using the data. This variant of the task is expected to yield higher numbers than the HSP design, since it does not require guessing from the entire vocabulary. A sample of (up to) five instances is taken for each target noun and verb (fewer if the word occurs less often in the data) and labeled in this way. Inner annotator agreement on this task is found to be very high (91% when computed between two researchers on
CHAPTER 3. A VISUOSPATIAL DATASET FOR NATURALISTIC VERB LEARNING

the project) and thus have a single annotator label all instances.

Table 3.2 shows the results of this analysis. The expected trend is observed, in which grounded context is a considerably better signal of noun use than verb use. It’s worth noting that there is substantial variation in training signal across verbs. For example, while some verbs (e.g. “pick”, “take”, “hold”) have strong signal, other verbs (“eat”) tend to be used in contexts sufficiently detached from the activities themselves. The noisiness of this signal is one of the biggest challenges of learning from such naturalistic data, which will be discussed further in §3.2.4.

<table>
<thead>
<tr>
<th></th>
<th>Nouns</th>
<th></th>
<th>Verbs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w</td>
<td>N</td>
<td>P</td>
<td>w</td>
</tr>
<tr>
<td>table</td>
<td>81</td>
<td>1.0</td>
<td></td>
<td>go</td>
</tr>
<tr>
<td>spoon</td>
<td>76</td>
<td>1.0</td>
<td></td>
<td>put</td>
</tr>
<tr>
<td>banana</td>
<td>75</td>
<td>0.8</td>
<td></td>
<td>pick</td>
</tr>
<tr>
<td>apple</td>
<td>68</td>
<td>1.0</td>
<td></td>
<td>eat</td>
</tr>
<tr>
<td>cup</td>
<td>57</td>
<td>1.0</td>
<td></td>
<td>take</td>
</tr>
<tr>
<td>ball</td>
<td>54</td>
<td>0.6</td>
<td></td>
<td>get</td>
</tr>
<tr>
<td>toy</td>
<td>48</td>
<td>1.0</td>
<td></td>
<td>wash</td>
</tr>
<tr>
<td>fork</td>
<td>47</td>
<td>0.8</td>
<td></td>
<td>play</td>
</tr>
<tr>
<td>bowl</td>
<td>42</td>
<td>1.0</td>
<td></td>
<td>walk</td>
</tr>
<tr>
<td>knife</td>
<td>40</td>
<td>0.8</td>
<td></td>
<td>throw</td>
</tr>
<tr>
<td>book</td>
<td>25</td>
<td>1.0</td>
<td></td>
<td>hold</td>
</tr>
<tr>
<td>plant</td>
<td>22</td>
<td>1.0</td>
<td></td>
<td>drop</td>
</tr>
<tr>
<td>bear</td>
<td>18</td>
<td>1.0</td>
<td></td>
<td>stop</td>
</tr>
<tr>
<td>chair</td>
<td>16</td>
<td>0.4</td>
<td></td>
<td>give</td>
</tr>
<tr>
<td>doll</td>
<td>13</td>
<td>0.8</td>
<td></td>
<td>open</td>
</tr>
<tr>
<td>clock</td>
<td>12</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lamp</td>
<td>2</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>door</td>
<td>2</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>window</td>
<td>1</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg.</td>
<td>37</td>
<td>0.8</td>
<td>Avg.</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 3.2: Estimates of training signal quality for nouns and verbs. N is the number of times the word occurs in the training data. P is the precision—given a 5 second clip in which the word is used, how often does the clip depict an instance of the word? Note that the verb “go” is an outlier, since it appears most often as “going to”.
3.2 Experiments

Based on the data, several grounded distributional semantics models (DSM) are compared in terms of how well they encode verb meanings, with a focus in particular on differences in how the environment is represented when put in to the DSM. The hypothesis is that models will perform better if they represent the environment in terms of 3D objects and their physics rather than pixels, since work in psychology has shown that children learn to parse the physical world into objects and agents very early in life (Spelke and Kinzler, 2007), long before they show evidence of language understanding. The effect of linguistic supervision on the performance of the models is also explored, both during environment encoding and during language learning. It should be noted that the models explored are simple instantiations meant to test the parameters of interest given the small dataset. Further work on more advanced models is expected to yield improvements.

3.2.1 Preprocessing

The raw data consists of continuous video and game-engine recordings of the environment, and parallel transcriptions of the natural language narration. To convert this into a format usable by the DSM, the following preprocessing steps are performed. This preprocessing phase is common to all the models evaluated. First, the environment data is segmented into “clips”. Each clip is five seconds long\(^4\) and thus consists of 450 frames (since the VR environment recording is at 90fps), which is subsampled to 50 frames (10fps). Since the grounded DSMs require associating a word \(w\) with its grounded context \(c\), the clip immediately following the utterance of \(w\) is considered to be the context \(c\). See earlier discussion (§3.1.3) for estimates of the signal-to-noise ratio produced by this labeling.

\(^4\)The length of 5 seconds was chosen heuristically prior to model development.
method. Training clips that are not the context of any word are discarded. Two subjects’ sessions (one from each visual aesthetic) is held out for testing, and the remaining 16 subjects’ sessions are used for training.

Finally, since this verb-learning problem proves quite challenging, the analysis is scoped down to the following 14 verbs, which come from the 20 verbs specified in the initial target vocabulary (§3.1.2) less 6 which did not ultimately occur in the data: “walk”, “throw”, “put (down)”, “get”, “go”, “give”, “wash”, “open”, “hold”, “eat”, “play”, “take”, “drop”, “pick (up)”. Again, these words all have low average ages of acquisition (19 to 28 months) and thus should represent reasonable targets for evaluation. Nonetheless, in §3.2.3 it is shown that models struggle to perform well on this task; this is elaborated on the discussion in §3.3.

### 3.2.2 Models

Four different DSMs are trained and evaluated, each of which represent a word $w$ in terms of its grounded context $c$. The parameters varied are 1) the feature representation of $c$ (§3.2.2) and 2) the type of supervision provided to the DSM (§3.2.2). All models share the same simple pipeline. First, a word-context matrix $M$ is built which maps each token-level instance of $w$ to a featurized representation of $c$. Dimensionality reduction is then run on $M$. Finally, the type-level representation of $w$ is taken to be the average row vector of $M$, across all instances of $w$. All model code is available at [http://github.com/dylanebert/nbc_starsem](http://github.com/dylanebert/nbc_starsem).

### Context Encoders

**Object-Based.** The Object-Based Encoder takes a feature-engineered approach to provide the model with knowledge of the basic object physics relevant
to the semantics of the verbs targeted. It represents each clip using four feature templates (trajectory, vel, dist_to_head, relPos). The first step is to find the “most moving object”, defined as the object with the highest average velocity over the clip. The four sets of features are then computed for this object. The velocity and relPos features are the mean, minimum, maximum, start, end, and variance of the object’s velocity and relative position over the clip, respectively.

For the dist_to_head feature, the distance from the object’s center to the participant’s head is computed for each position dimension (xyz). The following values are calculated: start, end, mean, variance, minimum, maximum, minimum index, and maximum index, where the minimum/maximum index is the point at which the minimum/maximum value was reached recorded as a percentage of the way through the clip.

The trajectory features capture the shape of the object’s trajectory over the clip. For each of the position dimensions (xyz), four points are computed during the clip: start, peak (maximum), trough (minimum), and end. The max and min determine the two key points (kp1 and kp2), with kp1 defined as the point that occurs first. The following features are then calculated: kp1-start, kp2-kp1, end-kp2, and end-start.

**Pretrained CNN.** To contrast the feature-engineered approach, an encoder based on the features extracted by a pretrained CNN is also implemented. The CNN encoder has an advantage in that it has been trained on a large amount of image data, but lacks domain-specific feature engineering. The VGG16 model, a 16-layer CNN trained on ImageNet that produces a 4096-dimensional vector for each image, is used.

For each frame in the clip, the 4096-dimensional vector is computed, and the following features are calculated along each dimension to get a vector rep-
representation of the full clip: start value, end value, minimum, maximum, and mean.

**Dimensionality Reduction**

Given a matrix $M$ that maps each word instance to a feature vector using one of the encoders, dimensionality reduction is performed to obtain a 10-dimensional vector for each word instance. The dimensionality is chosen as 10 since the goal is to differentiate between 14 words, and thus the supervised LDA cannot use more than 13 dimensions.

Two settings are considered. In the unsupervised setting, vanilla SVD is used. In the supervised setting, supervised LDA is used, where the "labels" are the words spoken at the beginning of the clip, as described in Section 3.2.1.

**3.2.3 Evaluation**

The models are evaluated in terms of their precision in assigning verbs to unseen clips. For two heldout subjects, the full session is partitioned into consecutive 5-second clips, resulting in a total of 189 clips. During testing, all clips are included, even those in which the subject is not speaking, unlike during training.

For each model, each clip is encoded using the model’s encoder, and the verb with the highest cosine similarity to the encoded clip is found. We (the authors) then view each clip along with the predicted verb and make a binary judgment of whether the verb accurately depicts the action in the clip, for example, "Does the clip depict an instance of 'pick up'?"

To avoid annotation bias, all four models plus a random baseline are shuffled and evaluated together, and the annotators do not know which prediction comes from which model. The annotator agreement was high, with a 91% agreement.
CHAPTER 3. A VISUOSPATIAL DATASET FOR NATURALISTIC VERB LEARNING

3.2.4 Results and Analysis

Table 3.3 reports the main results for each model. Strict precision, in which a prediction is considered correct only if both annotators deem it correct, and soft precision, in which a prediction is considered correct if one annotator deems it correct, are both computed. The results show that no model performs exceptionally well. On average, random guessing achieves a 32% soft precision. The Object-Based model, both supervised and unsupervised, and the CNN model perform slightly better, with an average of 40% soft precision. However, the sample size is small and the differences are not significant (refer to the 95% bootstrapped confidence intervals in Table 3.3). Only the unsupervised Object-Based model performs significantly worse than all other models, with a 20% soft precision. The CNN models do not show a significant difference with the supervised dimensionality reduction. Figure 3.3 displays example clips for each encoder.

Table 3.4 presents a breakdown of model performance by verb. The performance of the CNN-based model and the Object-Based model exhibits a few intuitive differences, which are discussed below. It is important to note that

Figure 3.3: Example clips, subsampled to 6 frames. (b) is (a)’s nearest-neighbor using the Object-Based model. In each of these clips, the participant picks up an object with their right hand. (d) is (c)’s nearest-neighbor using the CNN. In each, the participant is washing dishes in a similar looking sink.
CHAPTER 3. A VISUOSPATIAL DATASET FOR NATURALISTIC VERB LEARNING

<table>
<thead>
<tr>
<th>Method</th>
<th>Soft</th>
<th>Strict</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.32 (0.25–0.39)</td>
<td>0.23 (0.17–0.29)</td>
</tr>
<tr>
<td>Obj.</td>
<td>0.20 (0.14–0.25)</td>
<td>0.13 (0.08–0.19)</td>
</tr>
<tr>
<td>CNN</td>
<td>0.40 (0.33–0.47)</td>
<td>0.29 (0.22–0.36)</td>
</tr>
<tr>
<td>Obj+Sup.</td>
<td>0.40 (0.33–0.47)</td>
<td>0.28 (0.22–0.34)</td>
</tr>
<tr>
<td>CNN+Sup.</td>
<td>0.35 (0.28–0.42)</td>
<td>0.25 (0.19–0.31)</td>
</tr>
</tbody>
</table>

Table 3.3: Precision of each method with 95% bootstrapped CI. “Soft” means a prediction is correct as long as one annotator considers it to be so; “strict” means prediction is only considered correct if both annotators agree that it is correct.

Table 3.4: Analysis of model precision broken down by verb. Top-level columns are the unsupervised CNN, unsupervised obj model, and supervised obj model. For each, N is the number of times the model predicts that verb. Precision is the proportion of the time that prediction was correct.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pick</td>
<td>0</td>
<td>0.00</td>
<td>1</td>
<td>1.00</td>
<td>4</td>
<td>1.00</td>
</tr>
<tr>
<td>take</td>
<td>0</td>
<td>0.00</td>
<td>3</td>
<td>1.00</td>
<td>12</td>
<td>0.67</td>
</tr>
<tr>
<td>hold</td>
<td>11</td>
<td>0.64</td>
<td>5</td>
<td>0.80</td>
<td>17</td>
<td>0.65</td>
</tr>
<tr>
<td>get</td>
<td>32</td>
<td>0.56</td>
<td>5</td>
<td>0.00</td>
<td>13</td>
<td>0.54</td>
</tr>
<tr>
<td>go</td>
<td>29</td>
<td>0.21</td>
<td>1</td>
<td>0.00</td>
<td>17</td>
<td>0.47</td>
</tr>
<tr>
<td>put</td>
<td>4</td>
<td>0.00</td>
<td>11</td>
<td>0.27</td>
<td>31</td>
<td>0.29</td>
</tr>
<tr>
<td>play</td>
<td>7</td>
<td>0.29</td>
<td>17</td>
<td>0.18</td>
<td>6</td>
<td>0.17</td>
</tr>
<tr>
<td>walk</td>
<td>16</td>
<td>0.44</td>
<td>33</td>
<td>0.30</td>
<td>26</td>
<td>0.15</td>
</tr>
<tr>
<td>throw</td>
<td>36</td>
<td>0.08</td>
<td>25</td>
<td>0.00</td>
<td>16</td>
<td>0.06</td>
</tr>
<tr>
<td>drop</td>
<td>4</td>
<td>0.25</td>
<td>16</td>
<td>0.06</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td>eat</td>
<td>2</td>
<td>0.00</td>
<td>2</td>
<td>0.00</td>
<td>19</td>
<td>0.00</td>
</tr>
<tr>
<td>give</td>
<td>17</td>
<td>0.00</td>
<td>35</td>
<td>0.00</td>
<td>5</td>
<td>0.00</td>
</tr>
<tr>
<td>open</td>
<td>8</td>
<td>0.00</td>
<td>30</td>
<td>0.00</td>
<td>10</td>
<td>0.00</td>
</tr>
<tr>
<td>wash</td>
<td>23</td>
<td>0.48</td>
<td>5</td>
<td>0.00</td>
<td>11</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Low-level actions. The Object-Based models achieve higher precision on low-level verbs like “pick”, “take”, and “hold”. This makes intuitive sense, since
the 3D spatial features are designed to capture these types of mechanical actions, independent of the objects with which they co-occur. The 2D visual data, on the other hand, may struggle to ground a visually diverse set of objects-in-motion to these low-level mechanical actions.

**Visual cues.** Some actions are strongly predicted by specific objects, which are well captured by visual cues. This is most obvious in the case of “wash”, on which the CNN achieves higher precision than the Object-Based models. This is again intuitive as wash tends to co-occur with a clear view of the sink, which is a large, visually-distinct part of the field of view.

**Vague actions.** Actions like “go”, “walk”, and “hold” occur frequently, even when the language signal does not reflect it. That is, in any given clip, there is a high chance that the participant walks, goes somewhere, or holds something. Thus, models which happen to predict these verbs frequently may have artificially high accuracy. For example, the unsupervised Object-Based model only predicts “go” once and “hold” 5 times, which may contribute to the unsupervised Object-Based model performing significantly worse than random, despite seeming to capture low-level actions well.

**Special cases.** It’s worth noting that some verbs are very difficult or impossible to detect given limitations of the data. In particular, “give”, “eat”, and “open” have a precision of 0 across all models, as well as in the training signal (§3.1.3). For example, “give” only occurs twice in the data (“fluffy teddy bear going to give it a little hug” and “turn on the water give it a little sore[sic] and we can let it dry there.”), but cannot occur in its prototypical sense since there is no clear second agent to be a recipient. During instances of “eat” and “open”, participants tended to mime the actions, but the in-game physics data does not faithfully capture the semantics of these verbs (e.g., containers do not
actually open). These words highlight limitations of the environment which may be addressed in future work.

3.3 Discussion

Two types of models for grounded verb learning are compared in this analysis, one based on 2D visual features and the other based on 3D symbolic and spatial features. The analysis suggests that these approaches favor different aspects of verb semantics. An open question is how to combine these differing signals and design training objectives that encourage models to choose the right sensory inputs and time scale for grounding each verb.

A small set of verbs were evaluated that are acquired comparably early by children. Nonetheless, the models perform only marginally better than random. This disconnect highlights an important challenge to be addressed by work on computational models of grounded language learning: Can statistical associations between words and contexts result in more than simple noun-centric image or video captioning, eventually forming general-purpose language models? While that question is still open, research from psychology could better inform work on grounded NLP, especially in the interest of pursuing richer, more cognitively-inspired environments in language acquisition. For example, Piccin and Waxman (2007) argues that verb learning in particular is not learned from purely grounded signal, but rather is “scaffolded” by earlier-acquired knowledge of nouns and of syntax. From this perspective, the models explored here, which are similar to what is used for noun-learning, are far too simplistic for verb learning. More research is needed on ways to combine linguistic and grounded signal in order to learn more abstract semantic concepts. The next chapter addresses these limitations, by studying verb representations in a more controlled environment.
Chapter 4

Do Trajectories Encode Verb Meaning?

In this chapter, a self-supervised pretraining approach is presented where a time-series prediction model is trained to represent trajectories in a 3D environment. The biggest advantage of the methods presented are that there is full observation of world dynamics (i.e. trajectory data), without noisy observations or distractors (i.e. multiple objects, background). Additionally, human descriptions are collected of these perceived world dynamics. The results show that the model learns to represent the physical dynamics of verbs, despite no linguistic input during pretraining. Interesting trends are observed in how well the model learns to represent different verbs, doing well on verbs like “toss” and “throw”, and poorly on verbs like “drop”.

4.1 Dataset

4.1.1 Overview

The goal is to develop a dataset that contains continuous 3D recordings of an agent interacting with an object, where each recording is annotated with verbs describing the motion of the object. For example, if the agent throws a bouncy ball across a room, the recording is expected to be annotated with
a verb sequence such as “be thrown”, “fall”, “bounce”, “bounce”, “bounce”, “roll”, “stop”. To produce such data, a simple Markovian agent is built which interacts with a variety of objects in a 3D virtual environment. The resulting trajectory of the object is recorded and then, using crowdsourcing, humans are asked to determine which verbs could accurately describe which portions of the object’s movement. An example sequence from the dataset is shown in Figure 4.1.

4.1.2 Data Generation and Terminology

This section provides details on how the data is generated, and terminology is introduced that will be used throughout the rest of the paper.

Environment. The dataset is generated in Unity, a game engine seeing increased use by researchers (Deitke et al., 2020; Gan et al., 2020) for its accessible rendering and physics simulation via the underlying Nvidia PhysX physics engine. The dataset and simulation source code are publicly available.\(^1\)

Trajectory data. Trajectory data is defined as the position and rotation of entities in space, represented with three-dimensional XYZ coordinates and four-

\(^1\)https://lunar.cs.brown.edu/simulated/
Figure 4.2: Crowd annotation task. Crowdworkers make binary judgments on whether the verb applies to the clip. In this example, the worker is asked *Does the object bounce?* about the 1.5s video clip on the left.

dimensional $XYZW$ quaternions respectively. Only these features are chosen, ignoring other possibilities like object shape or identity, in order to focus on learning generalizable aspects of verb semantics that are independent of the object.

**Sessions.** The dataset is generated in 3-minute continuous segments referred to as sessions. Within each session, several parameters are randomized, including object shape, mass, drag, friction, and bounciness.

**Action Primitives.** The data generation is driven by a Markov Chain with a set of randomly parameterized action primitives. In this Markov Chain, the States are whether the object is *Held*, *OnCounter* and *OnGround*. The
transitions between these states are action primitives like \textit{PickUp}, \textit{PutDown}, or \textit{Throw}. For example, when the object is in the state \textit{OnCounter}, the agent may execute a \textit{PickUp}, after which the object is \textit{Held}. These action primitives are intended to produce a wide range of object motions corresponding to a range of verbs, and it is not expected that the primitives will map directly to the verbs that one would use to describe the resulting object behavior. For example, when a \textit{Throw} primitive is executed, the result might be that the object flies across the room, hits the wall, falls to the floor, and bounces until it comes to a rest. The execution of each action is parameterized with action-specific parameters, e.g. the force of a throw. The combination of session- and action-level parameters can result in a wide variety of object motion from each primitive action.

\textbf{Verbs.} A distinction is made between action primitives and the \textit{high-level actions} or \textit{verbs} that emerge from them. For example, if the object is \textit{pushed}, it may then \textit{slide, bounce, roll, tumble}, or any combination thereof. These are referred to as \textit{verbs}, though only \textit{push} is an action primitive. This distinction is highlighted because this work is most interested in studying the nuanced verbs that emerge from the simulation, rather than the action primitives that drive it explicitly.

\textbf{Frames.} The atomic units of data collected are \textit{frames}, also referred to as \textit{timesteps}, which represent a single point in time. The dataset is collected at 60 fps, or 10,800 frames per session. For each frame, the position and rotation of the object is recorded, as well as the position of the agent. This is sufficient to reconstruct and render the scene from an arbitrary perspective as needed. A high framerate is chosen because it’s relatively fast and inexpensive to rapidly produce trajectory data, which can be subsampled as needed for rendering or modeling.
CHAPTER 4. DO TRAJECTORIES ENCODE VERB MEANING?

4.1.3 Crowdsourced Annotation

Labels are collected for which verbs occur in the data, and when they occur. To do this, short clips are extracted from the dataset, and crowdworkers are asked to provide binary judgments on whether a given verb occurs in the clip.

Clips. Short clips are extracted from the dataset using Hierarchical Dynamic Clustering with Motion energy-based pooling (Zhang et al., 2018b), a self-supervised action segmentation framework that can be summarized as follows:

1. The 3D space is divided into clusters using the provided trajectory data. The framework uses Hierarchical Dynamic Clustering (Zhang et al., 2018b), which is similar to k-means but shown to outperform it on human motion parsing tasks.

2. A sliding window is applied to the cluster labels for a given positional sequence. The number of transitions between clusters in a window are defined as its motion energy.

3. The subsequent motion energy curve is smoothed using a Gaussian kernel with a tuned smoothing factor.
4. The peaks of the motion energy curve are considered motion segments, with lengths varying with respect to the width of the peak.

This algorithm is shown to perform well on human motion parsing (Zhang et al., 2018b), which transfers well to the dataset when applied to object position. This yields easily identifiable patterns of motion, e.g. from the time the object is thrown to when it slows to a stop. In contrast to a random sliding window, this approach avoids cutting clips in the middle of salient patterns of motion.

A disadvantage of this approach is that the extracted segments are variable-length. To simplify the pipeline, clips are filtered to only segments of length 72 to 96, then the segment is cropped to length 90, or 1.5 seconds. Each 1.5s segment is referred to as a clip. This length is chosen to make the clip as short as possible to avoid crowdworker fatigue, but provides sufficient time for a human observer to recognize what’s happening.

**Verbs.** 24 queries are produced, each corresponding to a verb, e.g. *Does the object bounce?* To do this, a list of 24 verbs is curated which are likely to occur in the simulated data and range from general descriptions (e.g., “move”) to more subtle descriptions of object motion (e.g., “tumble”). When asking annotators whether a verb applies to a clip, the question is framed with the object as the agent. That is, when a carry event occurs, annotators are asked “is the object carried”.

Every possible (clip, query) pair is considered a potential crowdsourcing task. Conservative heuristics are applied to filter out (clip, query) pairs that are guaranteed to have a negative label. For example, if the *Held* state was never present in a clip, it isn’t asked whether the object is *carried*. This results in approximately 110k tasks, from which 100 tasks are sampled per query, for a

---

2fall, carry, fall off, fall over, bounce, drop, pick up, push, topple, bump, tumble, roll, put down, hit, throw, flip, toss, tip, stop, spin, slap, slide, start, turn
CHAPTER 4. DO TRAJECTORIES ENCODE VERB MEANING?

Figure 4.4: Co-occurrence of different verbs for the same clip. For example, the majority of clips labeled *bump* are also labeled *slap*.

total 2,400 crowdsourcing tasks, such as the one shown in Figure 4.2.

Labels. For each crowdsourcing task, responses are obtained from five workers, then the majority response is taken as the *label* for that clip. The same clip is shown for all applicable queries, resulting in a supervised dataset of 24-dimensional vectors, representing binary verb labels for each clip.\(^3\) The dataset and all unaggregated annotations are available for download.\(^4\)

4.2 Dataset Analysis

In this section, trends in the dataset annotations are analyzed, including worker agreement, and comparisons between semantically related verbs.

\(^3\)Labels are only yes or no. *Unsure* was not the majority label for any task. Tasks that were filtered out during crowdsourcing are assigned a mask value that is ignored during training and validation.

\(^4\)https://lunar.cs.brown.edu/simulated/
4.2.1 Agreement

Annotation agreement on a clip is the proportion of responses that match the majority label for that clip. Figure 4.3 shows annotation agreement by verb. A noticeable trend is that agreement is higher for particular semantic categories. Specifically, verbs that involve gravity, i.e. fall, fall off, drop, and bounce have higher agreement. On the other hand, verbs of rotation, i.e. turn, spin, tip, flip have lower agreement, alongside abstract verbs start and stop. For start in particular, feedback was received from crowdworkers that they weren’t sure whether the object started moving during the clip or not.

4.2.2 Co-occurrence

Figure 4.4 shows co-occurrence, or how often a clip labeled one verb is also labeled another verb, according to the majority label. Co-occurrence allows the answering of questions like how often is a “toss” considered a “throw”? and vice-versa. Some interesting verb relationships are highlighted below.

General co-occurrence. Verb co-occurrence is high in general. On average, when a verb occurs, there is a 48% chance any other given verb also occurs. This highlights the challenge of verb learning, as opposed to more concrete nouns and adjectives. Verbs are applicable to a wide variety of behavior, even if it isn’t a prototypical instance of that verb.

Hyponym-hypernym relationships. All dogs are animals but not all animals are dogs. These hyponym-hypernym relationships are also ascribed to verbs. In some cases, the opposite is observed than what’s expected. For example, according to WordNet (Fellbaum, 2010), toss is a hyponym of throw. However, throws are annotated as tosses more often tosses than are annotated
as throws. That is, \( p(\text{toss}|\text{throw}) = .67 < p(\text{throw}|\text{toss}) = .75 \), suggesting that throw is a hyponym of toss.

**Frequent co-occurrences.** Hit, push, and bump stand out as the most frequently co-occurring verbs, having over 90% co-occurrence with each other. These likewise occur when many other verbs do, but not reciprocally. For example, most slaps are hits, but only 41% of hits are slaps. In many cases, this can be explained by other verbs being immediately preceded by the agent making contact with the object, which gets labeled hit, push, and bump.

**Fine-grained distinctions.** Workers distinguish roll from slide - only 50% of rolls are also considered slides, and vice-versa. This validates that verbs with similar trajectories, which may be challenging for models, are indeed differentiated by humans. Additionally, verbs with similar but nuanced meanings are differentiated. For example, tip, tumble, fall over, and topple tend to co-occur around 70-80% of the time. These also fall into “verbs of rotation” category, which have the lowest annotator agreement. It isn’t clear the extent to which these are nuanced distinctions, or annotation noise.

### 4.3 Experiments

In this work, the goal is to learn, without supervision, representations of motion data that align with verb meaning. To do so, a self-supervised model is pre-trained on a time-series prediction task, then a perceptron classifier is used to evaluate its learned representations. This work is especially interested in how these representations align with English verb semantics, i.e. can the model learn to make nuanced distinctions like throw vs. toss or slide vs. roll?

To accomplish this goal, four approaches are evaluated. First off, a simple perceptron is trained to evaluate the representational capacity of the trajectory
Figure 4.5: The pretraining setup. During pretraining, the model learns to encode and represent input timesteps for time-series prediction. To evaluate these learned representations, a perceptron probe is trained on the lstm outputs, without propagating gradients to the pretrained model.

data as-is, as a comparative baseline. Secondly, a fully supervised model is trained to determine a soft upper bound on the task without pretraining. Third, the self-supervised model is evaluated. And finally, the self-supervised model is fine-tuned to determine an upper bound with pretraining. This section provides details on the experimental setup and each of the four approaches.

### 4.3.1 Experimental Setup

For all approaches, representation quality is evaluated with a multi-way verb classification task. Specifically, the verb labels are predicted for the 1.5s clips gathered through the crowdsourcing task described in Section 4.1.3.

Each input sample $X_{t_{1..90}}$ is a 90x10 matrix of position and rotation data, corresponding to 90 frames per clip and 10 spatial features\(^5\) per frame. The output $Y$ is a 24-dimensional multi-hot vector indicating whether each of the 24 verb classes apply to the clip. For verb classes where there is no crowdsourced label, a mask token is inserted that is ignored during training and evaluation.

\(^5\)Object Position XYZ, Hand Position XYZ, and Object Rotation XYZW.
CHAPTER 4. DO TRAJECTORIES ENCODE VERB MEANING?

Figure 4.6: Comparison of each approach by verb. The green-blue bars show average precision for each approach. For comparison, crowdworker annotation agreement is shown in red.

4.3.2 Approaches

Perceptron. A goal is to evaluate the representational capacity of the raw trajectory data itself. To do so, a single 24-dimensional dense layer is trained with sigmoid activation, equivalent to a perceptron for each class. While very simple, this approach gives an idea of how well trajectory data represents verbs as-is, and provides a naive comparative baseline against which to evaluate the more complex pretraining techniques.

Fully Supervised. The fully supervised approach is similar to the perceptron, but adds a dense layer and LSTM layer in-between. This is equivalent to the model shown in Figure 4.5, but trained end-to-end without pretraining. The purpose of this approach is to provide an upper bound to the experimental setup without pretraining.

Self-supervised Pretraining. To evaluate the capacity of self-supervised models to represent trajectory data, a time-series prediction model is pretrained
CHAPTER 4. DO TRAJECTORIES ENCODE VERB MEANING?

<table>
<thead>
<tr>
<th>Approach</th>
<th>mAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Stratified</td>
<td>41.4</td>
</tr>
<tr>
<td>Perceptron</td>
<td>65.3</td>
</tr>
<tr>
<td>Fully Supervised</td>
<td>72.2</td>
</tr>
<tr>
<td>Pretraining + Probe</td>
<td>76.3</td>
</tr>
<tr>
<td>Pretraining + Finetuning</td>
<td>77.4</td>
</tr>
</tbody>
</table>

Table 4.1: Mean Average Precision (mAP) for each approach. The pretrained approaches outperform others on verb classification.

on a large unlabeled dataset of 400k sessions. That is, given $n$ input frames $X_{t_1..n}$, the model is trained to predict $k$ output frames $Y_{t_{n+1..n+k}}$. The model consists of a dense layer followed by an LSTM layer unrolled $k$ timesteps, as shown in Figure 4.5.

$$\gamma_{\text{MSE}} = \sum_{t=n}^{n+k} \gamma^t (y_t - \hat{y}_t)^2 \quad (4.1)$$

A discounted mean squared error loss is used as shown in Equation 4.1, which discounts loss by how far it is into the future by factor $\gamma$. Discount factor $\gamma$ is tuned to 0.85, output length $k$ to 60, model width to 128, and batch size to 1024, using a grid search on validation performance. Input length $n$ is fixed at 90 to match the length of clips.

The concatenated LSTM outputs are considered the representation of a clip. To evaluate this representation compared to raw trajectory data, a perceptron is trained for each class.

**Fine-tuning.** To provide an upper bound for the experimental setup with pretraining, the self-supervised model is fine-tuned. This is the same as the previous approach, but allows the gradients in the perceptron step to pass through the entire model.
4.4 Results

Mean Average Precision on unseen test data is reported for each approach in Table 4.1. These are compared to random stratified predictions that are based on the class distribution of the training data.

**Perceptron.** The perceptron approach evaluates the representational capacity of raw trajectory data as-is, with a lower bound of random stratified and soft upper bound of fully supervised. The perceptron performs relatively well for its simplicity, being only 7 points below the fully supervised upper bound. This suggests that the trajectory data itself encodes a significant amount of verb meaning, but leaves plenty of room for improvement.

**Self-supervised pretraining.** The pretraining + probe approach evaluates the ability of self-supervised models to encode verb meaning from trajectory data. This is equivalent to the perceptron approach, but with learned hidden representations as input rather than raw trajectory data. The pretrained model does outperform the perceptron, as well as the fully supervised approach. Fine-tuning only improves on this slightly, highlighting that self-supervised pretraining can yield representations that successfully encode verb meaning.

**Breakdown by verb.** Figure 4.6 shows a comparison of average precision for each verb. There are some clear trends. These can be categorized as *trivial*, *tractable*, and *hard* verbs.

*Trivial* verbs are verbs that are well-represented by trajectory data as-is, i.e. those with high performance with the perceptron approach. These include *fall, fall off, fall over* and *pick up*. Many of these have high agreement, and may

---

6 *Hit, push, and bump* are excluded from trivial verbs, as these have high average precision with random stratified, showing that they are very positively skewed, not necessarily well-represented.
be explained by the object’s change in height.

*Tractable* verbs are those that see significant benefit from pretraining, including slide, roll, throw, toss, put down, turn, flip, and stop. An intuition behind this is that these verbs involve the object following a particular type of trajectory, which the time-series prediction model captures well.

*Hard* verbs are those with low performance that don’t benefit much from pretraining. These include bounce, drop, tip, topple, and spin. Many of these are verbs of rotation, which have lower agreement. Bounce, slap and spin appear to benefit a bit from both pretraining and fine-tuning, suggesting that they may be tractable with similar but more robust pretraining. Tip and topple have fairly high performance, and may almost be categorized as trivial, perhaps being explained by the object’s change in rotation. However, they are noticeably lower than other trivial verbs, despite seeing no benefit from pretraining, suggesting that there is nuance to their meaning in the dataset, which isn’t captured by any approach. Finally, drop is a great example of a hard verb, as it is similar to trivial verbs like fall. However, the transitive drop involves interaction between the agent and object that is highly agreed upon by annotators, but doesn’t appear to be captured by the approaches, despite the model receiving both object and agent data. More challenging examples may be able to unveil a similar story for other verbs of interaction like pick up and put down.

### 4.5 Additional Experiments

In addition to the experiments described in Section 4.3, additional preliminary experiments were conducted to explore methods of encoding trajectory data.

An important decision in these experiments involved choosing between the *absolute* positions of objects (i.e. the positions of objects relative to an arbitrary fixed world origin) and *relative* positions (i.e. the positions of objects relative to
CHAPTER 4. DO TRAJECTORIES ENCODE VERB MEANING?

Figure 4.7: Comparison of using absolute trajectory inputs compared to relative trajectory inputs with the finetuned approach. The trajectory inputs (in blue) perform better overall, though the relative inputs (in orange) perform notably better for the verb “carry”.

This decision was informed by preliminary experiments comparing absolute to relative positions, using the fine-tuned approach described in Section 4.3. The results of these experiments are shown in Figure 4.7. The absolute inputs perform better overall. A notable exception to this is the verb “carry”. This may be attributed to the fact that the relative position of an object being “carried” remains consistently close to zero, as the object maintains a stable position relative to the agent.

These experiments informed the decision to employ absolute positions in the experiments described in Section 4.3. However, it would be valuable research direction to look deeper into the differences between absolute and relative inputs and their respective successes and failures. Although absolute positions demonstrated superior performance in these experiments, the picture may change with
increased scale and verb complexity.

### 4.6 Discussion and Conclusion

This work introduced a self-supervised model that learns representations of trajectory data that encode verb semantics. The model was discovered to help to encode semantics of some verbs in particular, i.e. *slide, roll, throw, toss, put down, turn, flip* and *stop*, but fails to encode semantics of others, such as *drop*.

This work is a first step toward exploring ways to capture fine-grained distinctions in grounded verb semantics that are trivial for humans, but challenging for models. Recent benchmarks at the intersection of NLP, vision and robotics (Deitke et al., 2020; Shridhar et al., 2020) illuminate unsolved challenges in AI that demand a more robust understanding of verb semantics and spatial reasoning. As these benchmarks continue to be developed, and rich multimodal datasets from technologies like virtual reality become increasingly abundant, future work in this vein will be especially relevant. A question remains how this model compares to a vision-based model. This will be addressed in the following chapter.
Chapter 5

Comparing Trajectory and Vision Models for Verb Representation

Figure 5.1: The Simulated Spatial Dataset consists of procedurally generated motion data of a virtual agent interacting with an object. A) shows the camera view as the object (in red) as it falls off the counter. B) shows the corresponding 2D trajectory data, while C) shows the corresponding 3D trajectory data.

In this chapter, trajectory-based models are compared to conventional image-based models for verb representation.
CHAPTER 5. COMPARING TRAJECTORY AND VISION MODELS FOR VERB REPRESENTATION

5.1 Experimental Design

The Simulated Spatial Dataset from Chapter 4 is used for all experiments in this chapter.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP (% micro)</th>
<th>mAP (% macro)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>39.44 ±1.87</td>
<td>41.19 ±1.53</td>
</tr>
<tr>
<td>3D Trajectory</td>
<td>83.38 ±1.27</td>
<td>69.95 ±1.54</td>
</tr>
<tr>
<td>2D Trajectory</td>
<td>82.99 ±3.97</td>
<td>69.94 ±3.09</td>
</tr>
<tr>
<td>2D Image</td>
<td>81.02 ±2.74</td>
<td>68.18 ±2.95</td>
</tr>
<tr>
<td>2D Image + 2D Trajectory</td>
<td>82.38 ±1.22</td>
<td>68.80 ±1.42</td>
</tr>
<tr>
<td>2D Image + 3D Trajectory</td>
<td><strong>84.06 ±1.02</strong></td>
<td><strong>71.72 ±1.13</strong></td>
</tr>
</tbody>
</table>

Table 5.1: Mean Average Precision (mAP) scores for each model on the verb classification task, reported with both micro and macro averaging. 95% confidence intervals are reported beside each condition.

5.1.1 Models

To evaluate the effectiveness of each modality for verb representation, two steps are followed:

1. Train a self-supervised LSTM encoder using a time-series prediction task, based on models from prior work (Ebert et al., 2022). The encoder is a simple feed-forward model followed by an LSTM. The input is a 90xd matrix of time-series data, corresponding to a 90-frame (1.5s) clip, with d dimensions depending on the modality. During self-supervised pretraining, the LSTM is unrolled 60 timesteps (1s), the outputs of which are trained to approximate true future frames using a discounted mean-squared-error (MSE) loss. This loss is discounted according to how far the prediction is in the future. Hyperparameters such as batch size, learning rate, discount factor, and hidden width were tuned using a grid search on the performance of each modality on the development data. This self-supervised pretraining is performed using the 2400-hour training subset of the Simulated Spatial Dataset.
2. Fine-tune and evaluate the encoder on a supervised verb classification task. Once the self-supervised encoder is trained, it is fine-tuned on a supervised verb classification task, using the crowdsourced annotations on the Simulated Spatial Dataset. The fine-tuning process is conducted using cross-entropy loss, and evaluated using the mean Average Precision score (mAP) on the test data.

5.1.2 Features

All of the experiments use the general self-supervised training and architecture described above, varying only the input features. The features considered are described below.

**3D Trajectory.** This approach uses 10-dimensional 3D trajectory data as input. That is, the 3D euclidean XYZ position of the hand and object, and quaternion XYZW rotation of the object.

**2D Image.** This approach uses 2048-dimensional Inception-v3 (Szegedy et al., 2016) embeddings trained on ImageNet (Deng et al., 2009) as input. A convolutional encoder is also evaluated that is trained on the raw image data, but found that it fails to encode temporal relationships in the Simulated Spatial Dataset. In preliminary experiments, this was found to be a result of mode collapse, as demonstrated in Figure 5.2.

The experiments that produced the failure case shown in Figure 5.2 involved training an equivalent pretraining procedure as in other approaches. However, the linear encoder and decoder were substituted with convolutional networks, given their demonstrated efficacy in encoding raw image data (Szegedy et al., 2016). However, this training procedure suffered from mode collapse, where all reconstructions approximate the mode of the training data. Mode collapse is a well-documented issue in generative architectures (Adler and Lunz, 2018).
the context of this research, a plausible explanation for this mode collapse is the limited visual diversity of the Simulated Spatial Dataset. As a future research direction, it would be valuable to investigate the underlying issues of these failures and explore using larger, more diverse visual datasets to potentially alleviate the issue.

Figure 5.2: True and predicted frames during image-based pre-training. The frame at \( t = 0 \) is the final input frame, where at \( t = 1 \) are the real and predicted future frames. The predicted images suffer from mode collapse, and consistently reproduce the same reconstruction.

**2D Trajectory.** This approach uses 4-dimensional 2D trajectory data as input. Specifically, this is the 2D euclidean XY position of the hand and object. The purpose of this experiment is to disentangle the real performance of the 2D image-based approach from the theoretical potential of a perfect 2D object-
detection model which simply traces the trajectory of the object in 2D space.

**2D Image + 2D Trajectory.** This approach combines the 2D image and 2D trajectory modalities, encoding each together into a shared embedding space. This approximates a perfect object detection model in conjunction with additional information that may be inferred from the visual modality.

**2D Image + 3D Trajectory.** This approach combines the 2D image and 3D trajectory modalities, which points toward the potential for future work that may involve combining modalities.

5.2 Results

5.2.1 Main Findings

Table 5.1 shows the Mean Average Precision (mAP) scores for each model on the verb classification task. These results suggest that while all modalities perform significantly above random, there is little difference in the performance of the modalities. While the 2D Image + 3D Trajectory model does outperform other models on the verb classification task, with a mAP score of 84.06 at the micro level and 71.72 at the macro level, the difference in performance is not significant, with its 95% confidence interval overlapping with other approaches.

5.2.2 Analysis

Table 5.2 shows the mAP of 3D Trajectory, 2D Trajectory, and 2D Image modalities, broken down by verbs *fall* and *roll*, the only verbs which exhibit a significant difference by modality. The difference for *fall* can be explained by failure cases where the image-based model fails to encode verb meaning when the object becomes obscured or has very low contrast with the background. *Roll*, on
the other hand, is an interesting case where the image-based model outperforms trajectory-based models. In qualitative analysis, this often involves instances where a round object with low friction slides, but doesn’t actually roll, per se. It can not be determined whether this is due to annotator error or conceptual differences in the meaning of roll, but in either case this highlights the challenges of verb learning. You can view video examples of these failure cases at the link provided.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>26.78 ± 4.63</td>
</tr>
<tr>
<td>3D Trajectory</td>
<td>96.56 ± 2.63</td>
</tr>
<tr>
<td>2D Trajectory</td>
<td>95.22 ± 3.55</td>
</tr>
<tr>
<td>2D Image</td>
<td>87.67 ± 4.33</td>
</tr>
</tbody>
</table>

Table 5.2: Mean Average Precision (mAP) scores with 95% confidence intervals for fall and roll, which exhibit significant performance difference for the 3D Trajectory and 2D Image models. For all other verbs, there was no significant difference between models.

Aside from these outlier cases, there are no significant differences in performance for each modality. These results suggest that, contrary to the hypothesis, 2D image-based models encode sufficient information to capture verb semantics on par with 3D trajectories-based models.

One possible explanation for these results is that 3D trajectories can be extracted from 2D inputs. To investigate whether this is the case, a follow-up analysis is performed using probing classifiers. Specifically, for each model, the encoder is fine-tuned to predict the 3D position of the object at the final frame. Mean Squared Error (MSE) loss is reported for each approach in Table 5.3.
CHAPTER 5. COMPARING TRAJECTORY AND VISION MODELS FOR VERB REPRESENTATION

It is indeed the case that, from the 2D trajectory alone, the model is able to reconstruct the 3D trajectory reasonably well, and that adding the image to the 2D trajectory further improves the results. However, none of the 2D representations perfectly capture the 3D representation. Thus, one interpretation of this analysis in combination with the results from above is that, while 2D information is impoverished relative to 3D, the differences that are lost when moving from 3D to 2D are not differences that are central for differentiating verb semantics.

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.4104</td>
</tr>
<tr>
<td>3D Trajectory</td>
<td>0.0104</td>
</tr>
<tr>
<td>2D Trajectory</td>
<td>0.0409</td>
</tr>
<tr>
<td>2D Image</td>
<td>0.0837</td>
</tr>
<tr>
<td>2D Image + 2D Trajectory</td>
<td>0.0289</td>
</tr>
</tbody>
</table>

Table 5.3: Mean Squared Error (MSE) for each model on a 3D object position regression task.

5.3 Discussion & Limitations

The results challenge the notion that richer environment representations will necessarily yield better representations of language, specifically in the context of verbs. The experiments show that aside from narrow outlier cases discussed in Section 3.2.4, models trained on 2D Image embeddings perform similarly well to models trained on 2D and 3D Trajectory data. Combined with followup probing analysis, the results suggest that, while some information is lost when collapsing to 2D world representations, the lost information might not be critical for differentiating verb semantics.

This work also highlights the challenges of language learning in embodied environments, and points toward the need to look deeper into how models may or may not be able to capture complex aspects of verb semantics, a major bot-
tleneck in language understanding, which will be critical to building AI agents that interact effectively with humans in realistic environments.

A limitation of this work is that the data used in this study is a highly controlled environment with a relatively small number of verbs. This means that the results may not generalize to other datasets, or when scaling up to larger and more complex models. Thus, overall, these results do not close the book on the question of which world representations best support verb learning. Further research is needed to fully understand the potential and limitations of richer environment representations for language learning.
Chapter 6

Discussion

This thesis tests the hypothesis that trajectory-based representations are more capable than conventional vision-based representations at capturing physical dynamics of verb meaning. In doing so, this thesis makes several important contributions to the field of NLP and embodied learning.

First, this thesis provides an empirical evaluation of trajectory-based representations and compares them to conventional vision-based representations for capturing the physical dynamics of verb meaning. This fills a gap in the literature as to the potential of using trajectories as a modality for language grounding in a modern machine learning architecture.

Second, this thesis builds two datasets of 3D interactions. These datasets may be used by future researchers to further investigate the questions of embodied language representation.

Finally, this thesis highlights the challenges of capturing the physical dynamics of verb meaning, and points toward directions for future work.

To test the hypothesis, this thesis first introduces the New Brown Corpus dataset in Chapter 3, a dataset of parallel language, trajectory, and egocentric vision data as participants perform household tasks, as shown in Figure 6.1.
Figure 6.1: Screenshots of a person picking up a banana in each of the two kitchen aesthetics.

Figure 6.2: The Simulated Spatial Dataset consists of procedurally generated motion data of a virtual agent interacting with an object. In this sequence the agent (red sphere) pushes the object (blue sphere). At t=0 and t=1, the agent approaches the ball. Then, in t=2 and t=3, the agent pushes to ball. Finally, at t=4, the ball is rolling away from the agent.

This data is shown to reflect the inputs received during childhood language learning.

Experiments were conducted on this data to test this hypothesis in Chapter 3. Since the dataset is quite small, a naive feature-engineered approach is taken to compare trajectory-based representations to vision-based representations. However, neither approach was found to be successful. The work in Chapter 3 highlights that challenges of verb representation, and points toward a need for further research.

An additional dataset is introduced in Chapter 4 - the Simulated Spatial Dataset. This dataset addresses the limitations of the New Brown Corpus...
dataset by presenting a large amount of data in a controlled setting, as shown in Figure 6.2. This consists of procedurally generated clips of an agent interacting with an object. These clips are annotated by crowdworkers with binary verb classification labels.

Experiments are conducted on this data to evaluate the capacity of self-supervised trajectory encoders to capture physical dynamics of verb meaning. These experiments demonstrate that the trajectory-based representations are indeed able to capture verb meaning, and that the self-supervised pretraining approach successfully improves the ability to capture nuanced differences in physical dynamics, such as the difference between “toss” and “throw”.

This thesis expands on this work in Chapter 5 by comparing these trajectory-based models to conventional video-based models on the Simulated Spatial Dataset. Experiments are conducted to provide an apples-to-apples comparison of self-supervised trajectory-based and vision-based encoders, and finds that 2D visual inputs perform comparably well to trajectory inputs. This suggests that contrary to the hypothesis, 2D visual data alone may be sufficient to encode physical dynamics of verb meaning, at least in simplified settings.

These findings have several implications for the field of language learning in embodied environments. First, they suggest that conventional vision-based representations may be sufficient for verb learning, which has important implications for the design of future embodied NLP systems. Second, this thesis highlights the challenges of encoding verb dynamics. Prior work has shown the limitations of large language models in reasoning about the world, and argue for the importance of grounding to overcome future challenges in NLP. Finally, this thesis points toward a need for further exploration in how to build models that truly capture the underlying physical dynamics of verbs.

A limitation of this thesis is that the experiments conducted were on limited
datasets. While these experiments show that 2D visual data captures physical
dynamics comparably well to trajectory data, it isn’t clear that this picture will
remain the same with increased scale and verb complexity. As 3D embodied data
becomes increasingly available with the emergence of simulated environments,
it becomes increasingly practical and necessary to conduct further experiments
at a larger scale.

Future work could focus on investigating whether these findings generalize to
a wider variety of verbs and interactions. When 3D embodied data is available
at a scale comparable to the data harnessed by state-of-the-art vision models,
it will be possible to build trajectory-based models at similar scale and com-
plexity. As massively scaling LLMs have been shown to “unlock” (Thoppilan
et al., 2022) new skills for language models, a similar story may be possible for
massively scaling models that are grounded in rich embodied data. While this
is an important direction for future work, it is important to also consider the
alternative possibility.

The alternate possibility is that conventional visual inputs will remain a
sufficient input modality. Though humans are shown to represent symbolic
objects prior to language learning (Speike and Kinzler, 2007), their inputs are,
in fact, vision. Thus, the human understanding of objects may be analogous to
intermediate layers in a state-of-the-art vision model. However, this explanation
requires more investigation into what language-and-vision models are capable
of capturing. Current vision-and-language models are argued to only shallowly
align language and vision modalities (Yun et al., 2021). Even if 3D embodied
inputs such as trajectories turn out not to be the answer, there is a large gap
in research as to how a rich understanding of the world may be captured.

Overall, this thesis may be considered an important early step in addressing
the challenges of language learning in embodied environments. Much more work
is needed to explore which direction may lead to systems with a rich embodied understanding of the world, necessary for solving the next challenges in NLP.

6.1 Conclusion

This thesis has explored the challenge of connecting the meaning of verbs to the physical dynamics of objects in the world. To do so, trajectory-based encoders are compared to vision-based encoders in controlled experiments, and while trajectory-based encoders are able to successfully capture verb dynamics, vision-based encoders are able to do so comparably well. This work has taken an early step toward a future where machines can communicate naturally with humans in the real world. While this work has demonstrated a lack of advantage of a trajectory-based approach over conventional vision-based approaches, it doesn’t close the door on this approach to addressing the challenges of machines communicating naturally with humans in the real world, and points toward a need for further work that questions how we can build machines that understand language in the context of the physical world.
Bibliography


Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. 2022. Revisiting the "video" in video-language understanding.


Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,


Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and