THESIS

EXPLORING CORRESPONDENCES BETWEEN GIBSONIAN AND TELIC AFFORDANCES FOR OBJECT GRASPING USING 3D GEOMETRY

Submitted by

Aniket Tomar

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2023

Master's Committee:

Advisor: Nikhil Krishnaswamy

Nathaniel Blanchard
Benjamin Clegg

ABSTRACT

EXPLORING CORRESPONDENCES BETWEEN GIBSONIAN AND TELIC AFFORDANCES FOR OBJECT GRASPING USING 3D GEOMETRY

Object affordance understanding is an important open problem in AI and robotics. Gibsonian affordances of an object are actions afforded due to its physical structure and can be directly perceived by agents. A telic affordance is an action that is conventionalized due to an object's typical use or purpose.

This work explores the extent to which a 3D CNN analogue can infer grasp affordances from only 3D shape information. This experiment was designed as a grasp classification task for 3D meshes of common kitchen objects with labels derived from human annotations. 3D shape information was found to be insufficient for current models to learn telic affordances, even though they are successful at shape classification and Gibsonian affordance learning.

This was investigated further by training a classifier to predict the telic grasps directly from the human annotations to a higher accuracy indicating that the information required for successful classification existed in the dataset but was not effectively utilized.

Finally, the embedding spaces of the two classifiers were compared and found to have no significant correspondence between them. This work hypothesizes that this is due to the two models capturing fundamentally different distributions of affordances with respect to objects, one representing Gibsonian affordances or shape information, and the other, telic affordances.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Nikhil Krishnaswamy for his patient guidance and constant, continuous encouragement throughout this long and often challenging process. I would also like to thank Dr. Nathaniel Blanchard and Dr. Benjamin Clegg for taking the time to be on my committee.

I want to thank my friends especially Tushar, Paras, Anurag, Aditi, and Ramya for their help, advice, and support. Additionally, I would like to thank David, DJ, Hannah, and Roxana for accepting me into their home and their hearts and keeping me in their prayers. I would like to thank my family for always having my back and believing in me.

DEDICATION

To my friends and family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

One sign of human intelligence is the degree, complexity, and specificity with which we use tools. This differentiates humans from the rest of the animal kingdom where tool use is uncommon and at best primitive. Humans not only use existing objects as tools, but also fashion new elaborate tools from these objects for specific or general uses. Socially, we even develop canonical uses for objects as tools. But how do we know about the use of objects? How do we determine the properties that a tool must possess to be useful for the purpose we are making it for? These questions have long been important for understanding cognition and how tool use shaped human evolution. With recent advances in AI, however, these questions have become increasingly important in computer science and robotics as well for achieving the long-standing goal of developing general and robust robots. Such robots would need to have the ability to infer and understand the uses of objects by perceiving them in the real world and to create tools they need using this understanding. Neural networks have therefore been looked upon as a viable approach toward object affordance understanding owing to their ability to learn flexible representations, but despite advances in neural approaches to human-object interaction, the problem remains challenging.

In 1977, J. J. Gibson [1] introduced the concept of affordances to describe the functional and ecological relationship between organisms and their environment. To say an object "affords" an action is to say that the object facilitates the action being taken with it. Affordances in the classic *Gibsonian* sense are those behaviors that are afforded due to the physical object structure and can be directly perceived by animals. Eg., for a monkey as well as a human, an iPhone is *throw-able* but not so for a dolphin. In 2013, James Pustejovsky [2] introduced the notion of a *telic* affordance or behavior that is conventionalized due to an object's typical use or purpose. Telic affordances

are also likely to be mediated by *habitats*, or conditioning environments that enable or inhibit a particular afforded behavior. Eg., a cookie is *edible* in general but not if it is *inside a trashcan.*

Affordance learning remains an important unsolved problem in AI even with the recent successes of Large Language Models. This is because affordances are rarely explained or even mentioned in corpora as humans do not learn affordances by discussing them or thinking about them. Humans learn about objects' Gibsonian affordances by learning the correspondences between their appearance and structural properties and by interacting with them. However, we usually learn the telic affordances of objects by watching others use them in the context of their typical conventionalized usages [3]. Thus, significant research effort has focused on using imitation learning for affordance learning [4–7] and other related tasks using modalities other than text with some success. Modalities such as image, video, and 3D mesh data allow for the Neural Networks to represent the objects more faithfully and to an increased degree of fidelity. However, the degree to which this increased fidelity is sufficient for the model to learn Gibsonian, as well as telic affordances without imitation, is uncertain. This research aims to tackle this question.

1.2 Research Questions

This work explores the following research questions:

- **[RQ1]** To what extent can the 3D representation learning models in current research learn Gibsonian and telic grasping affordances with only static 3D geometry information?
- **[RQ2]** Is there a fundamental difference in the representations of the 3D data learned by the model for the telic affordance learning task vs. the Gibsonian affordance learning task?

1.3 Challenges

This task presents several challenges:

1. Defining object affordances for an agent in the abstract when the object is static and devoid of any habitat is challenging.

2. Most research on representation learning has been done on text or 2D data and the insights gained from these cannot be directly or easily applied to learning from 3D data.

3. 3D objects can be represented as point clouds or 3D polygon meshes. 3D point cloud data is easier to collect in large numbers but is difficult to work with. On the other hand, 3D polygon meshes are of higher quality and can be used with existing 3D modeling and manipulation tools, however, they need to be constructed manually and are therefore difficult to acquire at a scale large enough to be used as learning data for neural networks.

   Even with the relatively higher quality of 3D meshes, often many meshes are not good enough to be used with current neural networks and need to be discarded. Quite often 3D meshes gathered from the internet require extensive cleaning, preprocessing, and manual inspection before they can be used as input. Thus, datasets that can be used to train neural networks are few and limited in size.

4. Creating a learning task and evaluating a model on learning Gibsonian affordances is challenging because an object affords different behaviors to a non-human AI than a human. An AI also has a different perception than humans and there may be affordances that the model can perceive that a human cannot and vice versa. Thus, creating a Gibsonian affordance learning task might end up having a human bias. It is also very difficult to conceptualize what Gibsonian affordances can mean for an AI that is disembodied, cannot interact with objects, and has no goals.

5. The difficulty in defining an affordance for a disembodied non-human agent can be overcome for telic affordances if the typical use of the object is defined as the use that is typical in humans. However, this makes creating and evaluating a learning task for telic affordance learning challenging because a way is needed to establish a consensus on the typical use of an object.

6. Another challenge is to find a metric for the performance of the model, as well as a way to interpret and analyze this performance on the task.

1.4 Overview of Approach

This thesis has been designed to overcome these challenges in the following ways:

1. Object grasping was used as a proxy for affordance because object grasping depends only upon the geometry of the object to a large extent and is not usually influenced substantially by the object's habitat. Object grasping is also one of the most foundational affordances that need to be learned for a model to learn higher-order affordances. If a model cannot learn grasps it will likely not be able to learn other higher-order affordances. Moreover, object grasping is an important unsolved problem in robotics.

2. A prominent 3D analogue of CNN was used. This CNN analogue extends the convolution and pooling operations to 3D data and can be used for a variety of 3D tasks just like a 2D CNN.

3. 15 common kitchen objects that are graspable by one hand were selected to be included in the dataset. 3D polygon meshes were collected for these objects. This was done by iteratively collecting meshes from many sources on the internet and then cleaning and preprocessing them. The meshes that were unusable by the model or had lost their identity were discarded and this information was used to guide the next iteration of mesh collection. A dataset was created from the meshes that remained.

4. A small preliminary grasp classification experiment was designed as a proxy for the Gibsonian affordance learning task. The model performed well as measured by the test accuracy but the validity of the learning task and evaluation metric was difficult to establish. It was difficult to parse whether the model was learning just the shape of the object or the Gibsonian affordance and it was also difficult to account for human-induced biases in the experiment. This experiment however informed the approach for future experiments and analysis.

5. A grasp classification task was designed for telic grasp learning. The grasps were divided into classes based on a survey of human subjects. A dataset was created from the collected

meshes. The 3D Neural Network model seemed to give middling results. This was validated by using a classifier directly on the survey data which performed better.

6. To study the reasons for the middling performance of the 3D Neural Network for telic affordance learning, the learned representations of the Human Annotation classifier and the 3D Neural Network classifier were compared after attempting a linear mapping between the two embedding spaces. The linear mapping was unsuccessful indicating that the models had learned different representations. Visualizing and comparing the two embedding spaces indicated that the 3D Neural Network classifier was learning 3D geometries or Gibsonian affordances even in the telic affordance learning task while the human annotations classifier was learning telic grasps.

1.5 Research Contributions

This research demonstrates that 3D Neural Networks rely too heavily on 3D geometry information for learning which is not sufficient to learn telic grasp affordances. This work further presents a detailed analysis of the reasons for this inadequacy by analyzing the representations learned by a 3D Convolutional Neural Network and comparing them to those learned by a classifier trained on grasp survey data.

The important contributions of this research can be itemized as:

1. Conducted a Grasp Similarity Survey and used it in a novel way to create a dataset for classifying telic grasps.

2. Created a novel 3D polygon mesh dataset for grasp affordance learning.

3. Designed and implemented an experiment to learn grasp affordances from polygon meshes using MeshCNN.

4. Demonstrated that classifiers trained using 3D information to learn telic grasp affordance learn different representations than those trained using human annotations with no clear correspondence between embedding spaces.

5. Hypothesized that this difference in representations is because the classifiers learn Gibsonian vs telic affordances.

1.6 Organization

The following Chapter 2 describes the existing related research work and provides the requisite background including the 3D Neural Network model used for this thesis. Chapter 3 describes the data collection process and the datasets for all the different experiments in this thesis. Chapter 4 describes the methodology and the experiment design for the different experiments. Chapter 5 describes the results of the experiments. Chapter 6 describes the analysis of the results and model performance and discusses the interpretation of these results. Chapter 7 discusses the conclusions of this research and the future directions that can be explored.

Chapter 2

Related Work and Prerequisites

2.1 Affordance Learning

2.1.1 Language Models

Humans more often learn about affordances *(e.g., "cups contain things," "spoons are used for stirring", "grasp knife from the handle end")* by using objects or watching them being used rather than being told about or reading about them. Hence this information is often absent from or sparsely distributed in linguistic corpora. Leveraging recent advances in NLP for learning affordances has been difficult because of the very small signal in corpora.

Further, for generalization of affordance understanding the models need to also be able to do commonsense reasoning about affordances. This has been a significant challenge. Learning from corpora has again proved difficult. Approaches based on word embeddings for example have been shown to give low vector similarity scores between object word vectors and their associated action word vectors *(e.g., "stir" and "spoon")*. Using symbolic learning or hardcoded knowledge has had more success in reasoning over affordances. Encoded knowledge of habitats and affordances has been shown to be useful, even over small sample sizes, at determining similarities between objects based on their known behaviors, and at acquiring partial information about novel objects [8]. However, this encoded knowledge is usually hand-crafted (e.g., in VoxML [9]), and difficult to acquire at scale.

Studies have shown that Large Language Models possess some commonsense world knowledge and can *"guess"* the affordances and properties of many objects, but they cannot reason about the relationship between these properties and affordances [10]. For example, BERT [11] *"knows"* that people can walk into houses, and that houses are big, but it cannot infer that houses are bigger than people. It would then seem that if a house was smaller than a person BERT would still suggest that it can be walked into [12].

Moreover, some of BERT's world knowledge comes from learning stereotypical associations [13]. There have been many suggestions that ungrounded LLMs do not have the ability to do these kinds of reasoning that is characteristic of humans because of the multimodality of human sensory inputs and our ability to ground reasoning in multiple modalities. For example, to fit a wide table through a door, GPT-3 suggests cutting it in half using a table saw [14]. This clearly misunderstands what a table saw is and the nature of the problem while also not being able to reason how to solve the problem: "You are having a small dinner party. You want to serve dinner in the living room. The dining room table is wider than the doorway, so to get it into the living room, you will have to remove the door. You have a table saw, so you *cut the door in half and remove the top half*."

Recently much Larger Language Models [15] have been shown to possess significantly better reasoning abilities than previously thought. However, these abilities need to be activated using intelligent chain-of-thought prompting [16] to allow the model to use its output as a working memory to reason. These approaches in the future may give better results but as of now, little research has been done to reason about object affordances using these approaches.

There have also been attempts to ground Large Language Models by using multimodal inputs like images with captions [17] but little research on their ability to reason about affordances has been conducted to our knowledge. Recently, Large Language Models have been used as a foundation, combined with robots, that have defined affordances to create a complete system [18] that can follow general instructions or perform actions in the real world to solve a problem described in natural language. Here, the language models are used to interpret user input and suggest a few actions to the robot which then responds with the actions it is capable of. Based on this the language model updates the action plan, and the robot performs it. This system has been deployed in a limited real-world setting with success, but the robot has a very limited set of affordances available to it. Moreover, the robot still has to be able to use its perception to determine which of the affordances available to it can be performed successfully in a given situation. Addressing the

problem of learning and generalization of these affordances from perception in 3D world is also the motivation for this work.

2.1.2 Other Modalities

Significant research effort has focused on using imitation learning for learning affordances [4–7] as well as for related tasks such as action recognition [19, 20], Human-Object Interaction [21–25] in a generalizable manner. Many successful efforts have been based on using imitation learning on video data. However, performance is highly dependent on demonstration quality [26] and gathering and labeling large quantities of the required high-quality data for this is a challenge [27]. Additionally, model performance on 2D image or video data often does not translate to real-world performance in 3D.

Moreover, although using imitation may be useful in action recognition, Human-Object Interaction, or even for learning meaningful representations, it is uncertain the extent to which it allows learning of affordances themselves according to Gibson's formulation. Most datasets focus on Action Recognition [28–31] or Human-Object Interaction [32–35] tasks and allow for any type of action that can be taken with an object, rather than a specific relation denoting *what the object offers the agent* a la Gibson. Fewer datasets exist to specifically learn affordances [36, 37] and affordance learning tasks are relatively less well defined. Most of these datasets are 2D image or video datasets and the models trained to learn affordances from these datasets cannot explicitly use 3D geometry information in affordance learning. Incorporating 3D object data as a modality along with other modalities can aid affordance learning and can cover some of the limitations of using 2D video data because apart from color and texture properties, 3D data explicitly encodes perceivable geometric properties which are often the most important properties in inferring affordances when no supplemental information about human actions is given. However, as we don't yet have large-scale 3D mesh temporal data for imitation learning, a salient and useful question is to what extent can static 3D data help in learning affordances. Datasets [38–40] for affordance learning using 3D data that existed prior to or those published during this research were not suitable for this

research as they did not allow for the fine-grained grasp classes derived from human annotations or as many single-hand graspable objects as in the dataset in this research. Further, the objects in these datasets were unsuitable to be used by the 3D CNN analogue in this research due to a variety of reasons including a large variation in object sizes. Hence, a new dataset had to be created for this research.

2.2 Cognition and Embodiment

Training an AI to learn affordances is part of the larger goal of training AI to solve problems in an environment by interacting with it as is done by embodied cognitive agents, it is thus useful to study embodied cognition while tackling this problem. However, in cognitive science, there are several notions of embodiments [41] as well as several theories of mental simulation in humans [42] such as the ability of humans to predict others' mental states, use mental simulations, logical reasoning, and exploration and experimentation have been theorized to underpin behavior, social interactions, decision making, learning, and problem-solving [42]. It is important to grapple with these different ideas of embodied cognition to help inform the problem description, and the design of the learning task, and to reveal inherent limitations in the ability of an AI in learning to solve such problems. However, it is very challenging to interpret what a neural network has learned in terms corresponding to these theories of mental simulation and this was not attempted.

Experiments in cognitive science have shown that cognition extends beyond the mind and is embodied, i.e., it can be influenced by the states of the body [43] or even the environment [44]. Experiments have also shown that even abstract cognitive states are grounded in states of the body and using abstract cognition can affect the state of the body [45].

In this research, the AI trained is disembodied. Since affordances are defined for embodied agents, it is challenging to create a disembodied learning task for a disembodied AI agent. Moreover, even when the problem of affordance learning is associated with embodied agents these agents are asocial such as robots, while the concept of telic affordances is defined socially for a group of embodied human agents. Thus, more care needs to be taken when trying to develop an

affordance learning task for an individual asocial disembodied AI agent. Given these challenges, I focus on using embodied cognition literature to inform the design of my learning tasks.

Although the usual way of designing neural network learning tasks assumes that most of the cognitive work involved in solving the task will be done by the learned complex representations of the large neural network, one amongst the many co-existing notions of embodiment [46] theorizes that the brain is not the sole cognitive resource that can be utilized to solve problems and in fact, it is the perceptually guided motion of the body and the state of the environment that does much of the work in solving problems. In this way, cognitive resources to solve a problem are distributed across the brain, body, and the environment, the latter two replacing the need for complex internal mental representations as described in [47] including for robots in [48]. This would raise the possibility of the task of affordance learning using a disembodied cognitive agent being intractable if not meaningless. However, the paper also describes a procedure to analyze the task to identify the cognitive requirements and the mental, and environmental resources available to fill these requirements. They suggest four key questions to ask:

1. What is the task to be solved?
2. What are the resources that the organism has access to in order to solve the task?
3. How can these resources be assembled so as to solve the task?
4. Does the organism, in fact, assemble, and use these resources?

This idea of cognition and method of analysis of identifying the resources available to the agent were suitable for analyzing the design of the learning tasks because it does not necessitate that the agent has a body as long as cognitive resources available to the agent can be identified. This analysis was incorporated into the broader analysis of my learning tasks and their implications for my experiments and the task of affordance learning in computer science research in general.

2.3 MeshCNN

Learning to predict object affordances based on its perceptible attributes and not directly from corpora is also important for many use cases such as autonomous robot learning. Predicting possible affordances can inform robot planning and action selection. Methods that learn object affordances from visual features exist, but few attempts have been made to use 3D data like polygon meshes or point clouds that can allow for direct access to information such as the structure of the object to ground such LLMs. Polygon meshes explicitly and efficiently capture both shape surface and topology in intricate detail. Although high-quality 3D polygon mesh data is difficult to acquire at scale unlike images there has been a significant push towards it and recent advances in capturing 3D data or synthesizing views from a few images, in stylizing a sample mesh based on a text prompt and in converting low quality acquired 3D point cloud data into high-quality polygon meshes. This suggests a trend that such difficulties in acquiring high-quality polygon mesh data might be mitigated in the future.

CNNs have been extremely successful at a wide range of computer vision tasks. A significant reason for this is that the inductive biases in a CNN are very well suited to images and the significant amount of image data available allows for the learning of very good task-independent representations. This success of CNNs in 2D perception might suggest that they can also be exploited in learning 3D representations of objects from images. However, this has proven to be a significantly challenging task. The limited availability of 3D data makes it challenging to learn 3D representations using a CNN. Directly using the convolution operation on 3D data is also challenging because of the absence of an implicit neighborhood and uniformity as in images and non-Euclidean geometry of 3D data.

MeshCNN [49] is an adaptation of convolutional neural networks for the analysis of 3D triangular meshes. MeshCNN uses specialized convolution and pooling operators analogous to the convolution and pooling operators of conventional CNNs, thereby importing the benefits of these well-understood models to 3D meshes. These operators are designed such that they can directly operate on mesh edges (akin to how conventional CNNs can operate on pixels) in a task-aware

fashion, unlike previous work on making the convolution operation intrinsic to the mesh [50–54], or using a convolution operation on point cloud-based representations [55–57], or work that involved first transforming 3D data into a regular representation on which a convolution operator could be applied [58–63].

A convolution operator can be applied to unambiguously ordered input features in the neighborhood so that the learned features are invariant. The conventional image convolution operator can be directly applied to images because images are represented as a regular grid with the inherent neighborhood, features, and order, which is not true for irregular and non-uniform 3D triangular meshes. To design the convolution operator for meshes in MeshCNN, the authors define a neighborhood for each edge that the operator operates on as edges contained in the faces incident on that edge. The vertices are ordered counter-clockwise. This ordering is ambiguous, which the authors address by defining input features of an edge as a 5-dimensional feature—the dihedral angle, two inner angles, and two edge-length ratios between the edge and the perpendiculars for each face from the edge. Further, the authors aggregate the four incident edges that make a ring around the edge being operated on into two pairs of edges that have ambiguity and generate new features by applying simple symmetric functions like summation on each pair. Thus, the neighborhood, features, and order are defined in a way that a convolution operator can be applied to the edges and can learn invariant features. The pooling operation is defined as the collapse of incident edges to a point on the edge being operated on. The edges are put in a priority queue and edges with features having the smallest norm are pooled first making the pooling operation task-aware.

2.3.1 Advantages of MeshCNN

Using the operators described above, MeshCNN has demonstrated good performance on a number of different learning tasks including segmentation and classification [64–66], including using fewer parameters and compute time than comparable methods.

The pooling operation that MeshCNN provides is task aware and gives priority to the edges that have features with the lowest norm for a task while pooling. As a mesh passes through successive

pooling layers of the network this process leaves the edges that are important to a task intact while those that are not are pooled. Thus, this provides us a way to analyze what the model is learning by visualizing the mesh after it has undergone successive pooling stages and comparing them with each other. The features that are important for the task would be present in a pooled mesh and those that are not will not. We can also see which features are being learned in each convolutional layer by comparing the pooled mesh with the input mesh to the layer.

These advantages were the reason MeshCNN was selected for this task.

2.3.2 Preliminary Experiments and challenges with MeshCNN

I conducted some preliminary experiments with MeshCNN to understand how it could be best used for my experiments.

Before collecting 3D data for MeshCNN, I conducted an experiment to find out the resolution of simple object meshes that MeshCNN would work best for. I used the original SHREC16 dataset [67] that contains simple meshes of 500 faces each and created 2 more datasets from it with meshes modified to have 250 and 750 faces respectively. MeshCNN performed better for lower-resolution meshes than the higher-resolution meshes. This helped inform my data collection and I tried to collect the lowest resolution of 3D data I could.

I also conducted an experiment with a dataset containing varying resolutions of meshes. MeshCNN's performance worsened significantly if the dataset had large variations in mesh resolution.

Given these findings, during data collection, I preferred the lowest resolution of 3D triangular meshes for the objects selected to be in the dataset. Given the wide variety of objects selected to be in the dataset, it was infeasible to collect meshes of the same resolution for different objects. So, the collected meshes were resized to be of comparable resolution.

MeshCNN requires that input meshes pass certain validity checks which required the meshes to be preprocessed. These resizing and preprocessing steps are discussed in detail in Section 3.1.1.

MeshCNN has been demonstrated with known benchmark datasets, e.g., the SHREC dataset, but when training and testing on new meshes, there is a chance that the MeshCNN pooling op-

eration in one of the early convolutional layers will cause the resulting feature map equivalent to be nonmanifold when it enters later layers. Therefore, I trained MeshCNN on each mesh for 10 epochs on a dummy classification task with only 1 class, but the initial mesh and pooling resolution set to the same values to be used in the actual classification task. I then discarded any mesh that threw an error due to the aforementioned property of the pooling operator.

This difficulty with the pooling operation was compounded by the fact that the meshes were collected from many different public repositories and then resized and operated on for preprocessing before being passed as input. Performing these operations on anyway low-quality freely available meshes meant that they could not be pooled significantly by the network without the pooling resulting in non-manifold meshes. This not only meant that a larger network with more computational resources had to be trained because of the inability to discard irrelevant information and reduce the size of later layers by pooling but also that an alternative method for analyzing what the model was learning as the internal pooled representations would not change across the layers and visualizing them would not provide any information. The solution to this was to train a classifier on human survey data and then compare its learned representations with that of MeshCNN. This is described in detail in Sections 4.3, 6.1, and 6.3.

Chapter 3

Datasets

The following chapter describes the creation of the datasets for the different tasks and experiments. It is divided into three sections, Sections 3.1 and 3.2 based on the type of data *(Meshes or Human Annotations)* used for creating the datasets, and Section 3.3 describes the derivation of classes for Telic Affordance Learning classification task. The Meshes Section 3.1 is further divided into task-specific subsections - *Gibsonian Affordance Learning Task (Section 3.1.2)*, *Telic Affordance Learning Task (Section 3.1.3)* as well as into subsections expanding on other relevant information like mesh preprocessing (Section 3.1.1).

For both Gibsonian as well as Telic affordance learning, grasping was used as a proxy for affordances in genetal. So, while creating the dataset I focused on objects from a domain where the primary affordance is *grasp*, which is an important domain for applications, is a common domain both in real-world human interaction and in research, is sufficiently difficult and which will have sufficient 3D data available for creating a dataset. I saw common kitchen objects as fitting these criteria.

MeshCNN like a general convolutional network is invariant to scale, and the meshes by themselves don't provide any information about scale. This implies that MeshCNN would process a large cylinder in much the same way as a smaller cylinder. Here it meant that the network would not have any information to distinguish between a glass and a trashcan for example. To limit the impact of this on the experiment only small objects that can be grasped by a single hand were used.

I first began by selecting a test set of common household objects graspable by a single hand. These objects, all found in a kitchen and therefore reminiscent of common problems in this domain [28], include: *bottle*, *mug*, *knife*, *bowl*, *plate*, *wine glass*, *pen*, *apple*, *jar*, *spoon*, *fork*, *glass*, *teapot*, *banana*, *pan*.

3.1 Meshes

For creating a dataset of meshes to train MeshCNN for each of the two tasks, I collected 3D meshes (e.g., see Figure 3.2) from public repositories. These meshes were all converted to the Wavefront 3D Object file format (.obj) because MeshCNN uses .obj files as input.

3.1.1 MeshCNN Preprocessing

As discussed before, MeshCNN necessitates that the input meshes contain roughly the same number of edges just like a CNN requires that input images be of roughly the same resolution. It also requires that the input meshes be manifold (continuous meshes or meshes not violating properties of Euclidean space at the close resolution, such as by having crossing edges), free of islands (faces unconnected to other faces), and free of zero faces (3 edges bounding a topological one-dimensional hole).

Thus, I standardized the varying number of faces in each mesh by sub-triangulation to create additional faces or edge fusion to remove edges as necessary. I standardized the faces to approximately 2,000 for Preliminary Gibsonian Learning Task from a starting number of faces in the range of 1,600-2,800 and to approximately 8,000 for Telic Affordance Learning Task. This difference in the standardized number of faces between meshes for the two tasks was due to the fact that the Telic Affordance Learning Task had a larger variety of objects that had a greater range of an initial number of faces and many of these objects could not be reduced in resolution using further without destroying their identity. Following this process, each mesh used for the Telic Affordance Learning had approximately 15,000 edges. Following this process, for each task, I used the open-source MeshLab tool to clean up each mesh by removing islands (faces unconnected to other faces), zero faces (3 edges bounding a topological one-dimensional hole), and non-manifold meshes (non-continuous meshes or meshes violating properties of Euclidean space at close resolution, such as by having crossing edges). I then visually inspected each mesh to make sure that this process did not cause the mesh to be deformed beyond recognition of its original identity: that is, a cleaned mesh of a bowl still needed to visually resemble a bowl to a human observer in

order to make valid comparisons to MeshCNN's capabilities. Finally, I validated each mesh with MeshCNN itself.

3.1.2 Gibsonian Grasp Affordance Mesh Dataset

The preliminary experiment on Gibsonian Affordance Learning using MeshCNN was designed as a classification task and conducted with a small dataset consisting of 4 grasp classes represented by only one object each *(bottle, bowl, knife, mug)* with 15 training and 4 test meshes each. I decided that this was an appropriate size for a preliminary experiment because these numbers of per-class training and test meshes were comparable to the SHREC16 dataset that MeshCNN had been benchmarked against although it had many more (30) classes. The conceptual difficulties associated with designing this preliminary experiment as described in Section 4.1 resulted in the reformulation of this research into its current form.

3.1.3 Telic Grasp Affordance Mesh Dataset

For each of the 15 objects, I collected, preprocessed, and validated 40 3D meshes from public repositories. As described in Section 3.3 5 classes were derived for Telic Affordance Learning and these 15 objects were divided into the 5 classes as shown in Table 3.1. The meshes were divided into the 5 derived classes with each class containing 60 training and 10 test meshes distributed approximately equally among the objects within a class. For eg., class 0 has only 2 objects - Apple, Banana. Thus, each object will contribute 30 meshes in the training set of the class. In comparison, class 4 has 5 objects - Spoon, Fork, Knife, Pen, and Pan. Thus, each object will contribute only 12 meshes in the training set of the class. The meshes to be included were selected at random.

3.2 Human Annotations Dataset for Telic Grasp Affordance Learning

I created a survey to elicit the canonical grasp pose for each object. The survey was posed as a multiple-choice questionnaire: "*Consider how your hand is posed while grasping each object for typical use. Then, for each object, select all other objects which are grasped using a similar hand pose.*" This phrasing, particularly the phrase "for typical use," was chosen to elicit the *telic* affordance for each of these objects. For each object, annotators could select which of the 15 objects satisfied the question, allowing for multiple objects to be selected as being grasped similarly to the one in question.

I had 28 annotators take the survey in total, resulting in 28 15-dimensional k-hot vectors. Each object was assumed to be grasped like itself, allowing us to keep indices constant across all objects. I used standard statistical techniques for identifying outliers, such as z-score filtering (with a z-score of 5) and normalization [68]. Because a single outlier can make the standard deviation large, it is common to use the median of all absolute deviations from the median (MAD) as a more robust measure of the scale [69]. I computed the Kraemer kappa reliability score [70], to account for more than two annotators and the variable number of choices each annotation allowed for, resulting in $\kappa \approx .32$, indicating "fair agreement" according to Landis and Koch [71] [1].

3.3 Deriving Grasp Classes for Telic Affordance Learning

In order to assess MeshCNN's ability to classify objects according to their graspability, I needed to organize my ground truth object annotations into classes according to their grasp poses.

I first summed the 28 vectors for each object into single vectors for that object. The resulting matrix can then be treated as a co-occurrence matrix. For each object (i.e., each row), I computed the Positive Pointwise Mutual Information (PPMI) with each other object (i.e., each column), ac-

[1] While the Landis and Koch scale is the widely-accepted scale for assessing Kappa values, it was designed only for Cohen's Kappa on single-choice, 2-evaluator problems and therefore its assessment of other kinds of Kappa on multi-evaluator problems is usually artificially deflated (in other words kappa of .32 on a problem a multi-evaluator multi-choice problem is probably considerably better than "fair" agreement as the scale would suggest)

cording to $PPMI(a,b) = max\left(ln\left(\frac{P(a,b)}{P(a)P(b)}\right), 0\right)$, and then used Euclidean distance as a similarity measure to find the similarity between each object.

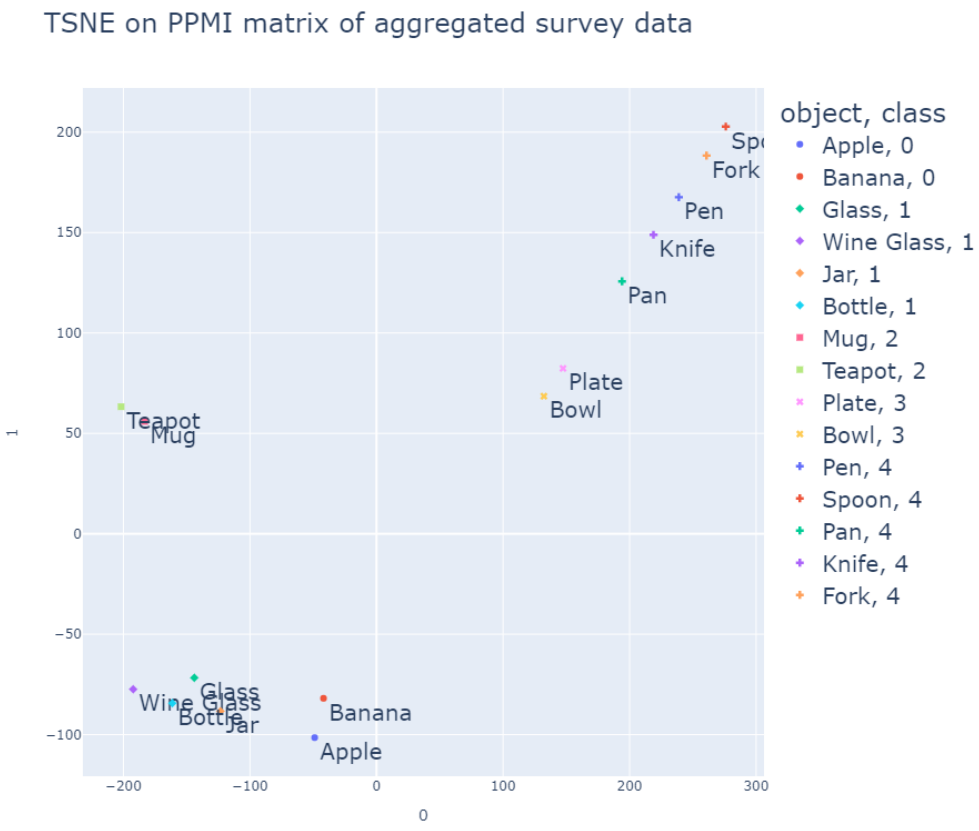

Figure 3.1: TSNE plot of object PPMI vectors.

I then ordered object pairs based on the computed similarity scores and assigned the objects in the pairs starting with the most similar to the same class. For pairs where one object was already assigned to a class but the second object was not, I put the second in the same class. When both objects in a pair had not yet been assigned to a class, I assigned both to a new class and repeated this until all the objects were assigned to a grasp class. To check the validity of this assignment I plotted a TSNE plot for the aggregated object PPMI vectors (Figure 3.1).

Table 3.1 shows the resulting grasp classes, which can be denoted by a description of the hand pose. Following terminology used in occupational therapy [72], which has since been adopted by

Table 3.1: The 5 classes derived from the assignment scheme.

Grasp Class	Objects
0	Apple, Banana
1	Bottle, Wine Glass, Glass, Jar
2	Mug, Teapot
3	Bowl, Plate
4	Spoon, Fork, Knife, Pen, Pan

the robotics community [73] Class 0 is a *spherical* grasp class, holding a fruit as if for consumption. Class 1 is the similar *cylindrical* grasp, a canonical pose when holding a glass for drinking. Class 2 contains the only objects in the dataset with "ear" handles joined to the object at both ends, which both use the *hook* grasp. Class 3 contains two objects typically held from the side or bottom (e.g., Figure 3.2), which use the *palmar pinch*. Class 4 is a *tripod grasp*, and contains objects where the hand is held as if eating with a spoon or writing with a pen.

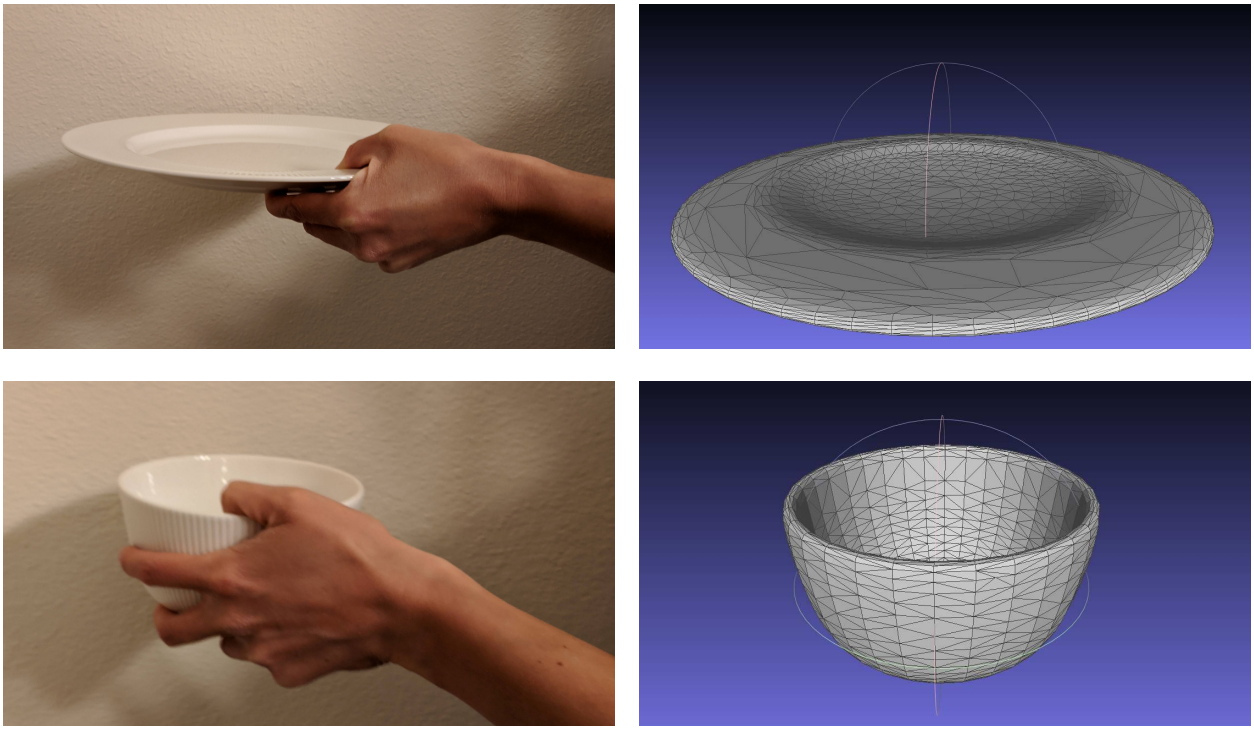

Figure 3.2: Plate and bowl grasping images along with their respective geometries. Plate and bowl form class 3

Chapter 4

Methodology

4.1 Gibsonian Affordance Learning & Embodiment Problem

To explore the research question **[RQ1]** - *To what extent can I use the current research in 3D learning to learn Gibsonian and telic grasping affordances with only static 3D geometry information?* I decided to design a small preliminary experiment to check if an advanced 3D machine learning algorithm *(MeshCNN)* could learn Gibsonian grasp affordances from just 3D mesh data. I formulated the task as a simple classification task using the small dataset described in Section 3.1.2 - 4 classes containing a single object each with 15 training and 4 test meshes in each class. The best hyperparameters were found by a grid search and are given in Table 4.1. The MeshCNN classifier achieved a high **test accuracy of 95%**.

Table 4.1: MeshCNN Hyperparameters for Gibsonian Grasp Affordance Learning Task.

Hyperparameter	**Value**
`pool res`	3500, 3500, 3500
`# conv filters`	32, 64, 128
`# neurons in FC layer`	50
`normalization`	group
`# resnet blocks`	1
`flip edges`	0.2
`slide vertices`	0
`# augmentations`	20
`# epochs with initial LR`	50
`# epochs with LR decay`	30
`# input edges`	3750
`optimizer`	Adam
Accuracy	95%

However, many conceptual challenges were encountered during the experiment design. Creating a learning task and evaluating a model on learning Gibsonian affordances is challenging because an object affords different behaviors to a non-human disembodied AI than a human. An

AI also has different sensory perception than humans and there may be Gibsonian affordances that the model can perceive that a human cannot and vice versa. Thus, creating a Gibsonian affordance learning task might end up having a human bias. It is also very difficult to conceptualize what Gibsonian affordances can mean for an AI that is disembodied, cannot interact with objects, and has no goals.

Identifying the cognitive resource requirements and their availability by analyzing the task using the procedure described in Section 2.2 suggested that the agent had the resources required to complete the task if the task was defined as - *Classify the meshes of four unique objects into corresponding four classes based on human-labeled grasp classes*. However, the analysis was difficult if the task was defined as - *Classify the meshes of four unique objects into corresponding four Gibsonian grasp affordance classes* - due to the difficulty in defining Gibsonian affordance for the AI.

1. *What is the task to be solved?*
    - Classify the meshes of four unique objects into corresponding four classes based on human-labeled grasp classes
2. *What are the resources that the organism has access to in order to solve the task?*
    - Human-labeled 3D mesh training data
3. *How can these resources be assembled so as to solve the task?*
    - MeshCNN can learn the shapes from object mesh data and since there is only one object per class and the objects differ in shape, MeshCNN can classify the learned shapes correctly
4. *Does the organism, in fact, assemble, and use these resources?*
    - Yes, this is discussed in detail in the discussion, Section 6.3.

4.2 MeshCNN Classifier

To classify the meshes according to grasp class, I trained an instance of MeshCNN using empirically-derived hyperparameters (shown in Table 4.2). `flip edges` refers to a data augmentation technique used by MeshCNN where a percentage of edges in the mesh are selected randomly and flipped[2]. `slide verts` refers to a similar data augmentation technique achieved by sliding vertices along the mesh surface. With the meshes in my dataset, this was a cause of the problems with non-manifold intermediate representations that I encountered during preprocessing (see Section 3.1.1), and so I did not use this hyperparameter.

Table 4.2: MeshCNN Hyperparameters for telic Grasp Affordance Learning Task.

Hyperparameter	**Value**
`pool res`	15000, 15000, 15000, 15000
`# conv filters`	32, 64, 128, 128
`# neurons in FC layer`	200
`normalization`	group
`# resnet blocks`	1
`flip edges`	0.2
`slide vertices`	0
`# augmentations`	20
`# epochs with initial LR`	100
`# epochs with LR decay`	50
`# input edges`	15600
`optimizer`	Adam
Accuracy	54%

Identifying the cognitive resource requirements and their availability by analyzing the task using the procedure described in Section 2.2 suggested that the agent may not have had the resources required to complete the task defined as - *Classify the meshes of the 15 objects into corresponding 5 telic grasp affordance classes*. The important resource here is the information encoded in the mesh dataset from the Human Annotations survey. Although the telic grasp classes were extracted based on the Human Annotations, the labeled meshes themselves either provide a relatively small signal when compared to the high learning signal in the similarity matrix of Human Annotations

[2]What is meant by "edge flipping" is well beyond the scope of this thesis but a detailed treatment is given by [74].

or MeshCNN is too biased towards using shape information for learning. This suggests that an embodied agent that can access and use the environment cognitive resource of human annotation survey information dataset should fare better. This experiment is conducted in Section 4.3.

1. *What is the task to be solved?*

    - Classify the meshes of the 15 objects into corresponding 5 telic grasp affordance classes

2. *What are the resources that the organism has access to in order to solve the task?*

    - Human-labeled 3D mesh training data

3. *How can these resources be assembled so as to solve the task?*

    - It seems that these resources can be assembled by MeshCNN to solve the task. It can learn the shapes from mesh data but struggles to classify the learned shapes with similarly shaped objects being in different classes and multiple objects in each class. The telic grasp information from the similarity matrix is either not fully captured by the small number of meshes or MeshCNN relies too much on 3D shape data to classify telic grasps.

4. *Does the organism, in fact, assemble, and use these resources?*

    - MeshCNN does use and assemble the meshes as resources however it is uncertain if it is able to fully exploit the telic grasp information contained within the mesh dataset or if it just uses the shape information.

4.3 Human Annotations Classifier

I trained a classifier to classify every vector representing an object from every human annotation into one of the 5 different grasp classes. Because the grasp classes had a different number of objects, the classes also had a different number of training vectors, making the dataset imbalanced. To correct this imbalance, I randomly discarded excess training vectors from each class to achieve

a sample balanced across classes, resulting in 56 training vectors per class (280 training object vectors total). I then divided the data into 82% training and 18% test splits, corresponding to 46 samples per class in the training set (230 samples total) and 10 samples per class in the test set (50 samples total). I built an MLP classifier in PyTorch, using grid search to tune hyperparameters, arriving at the hyperparameter set shown in Table 4.3.

Table 4.3: Hyperparameters for Human Annotation classifier.

Hyperparameter	Value
`input size`	15
`hidden layer size`	200
`# classes`	5
`# epochs`	60
`learning rate`	0.2
`batch size`	46*5
`optimizer`	Adam

Identifying the cognitive resource requirements and their availability by analyzing the task using the procedure described in Section 2.2 suggested that although the agent may not have had the resources required to complete the task defined as - *Classify the vectors of the 15 objects into corresponding 5 telic grasp affordance classes* it is still able to access and utilize more relevant environment resources then MeshCNN. The important resource here is the high signal telic information encoded in the Human Annotations survey. This agent may be considered a more embodied agent than MeshCNN as it is able to better use the important environment cognitive resource. This is also discussed in Section 6.3.

1. *What is the task to be solved?*
    - Classify the vectors of the 15 objects into corresponding 5 telic grasp affordance classes
2. *What are the resources that the organism has access to in order to solve the task?*
    - Vector training data derived from Human-Annotation survey
3. *How can these resources be assembled so as to solve the task?*

- The classifier can learn to classify the vectors based on their similarity.

4. *Does the organism, in fact, assemble, and use these resources?*

    - Yes, the classifier does assemble and use these resources. It does a better job of accessing and utilizing these resources than MeshCNN.

Chapter 5

Results

5.1 MeshCNN Classifier Results

Using the given architectures and hyperparameter combinations above, the MeshCNN classifier, despite the extensive preprocessing, achieved only a **test accuracy of 54%**. Figure 5.1 shows the confusion matrix for the MeshCNN classifier. Here there are only two classes that achieve high classification accuracy: Class 0 (the spherical grasp class) and Class 2 (the hook grasp class). It is observed that topologically the two objects in each class have an obvious correspondence: both apples and bananas are topological spheroids while both teapots and mugs are topological toroids.

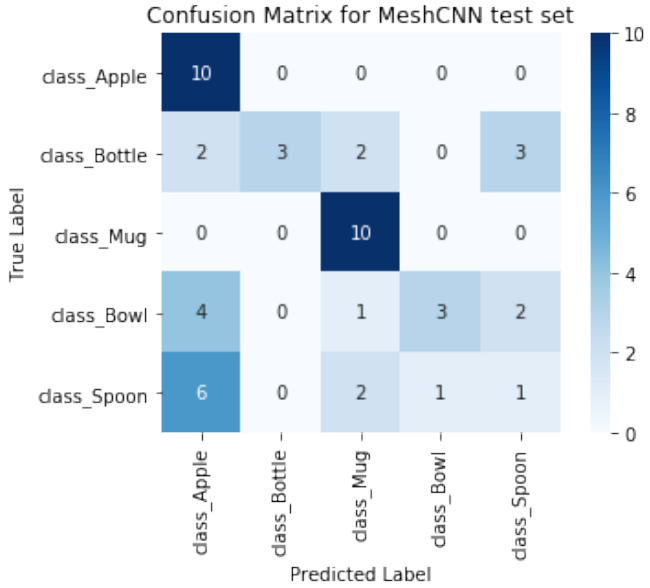

Figure 5.1: Confusion matrix of the MeshCNN classifier test output.

5.2 Human Annotations Classifier Results

Using the architecture and hyperparameter combination described in Table 4.3, the human annotation classifier achieved a **test accuracy of 72%**. This result was validated by training a Random Forest classifier as well.

Chapter 6

Analysis & Discussion

6.1 Linear Mapping between Embedding Spaces

The poor performance of MeshCNN on this task necessarily raised questions about what the network was learning in this case. MeshCNN has advertised effectiveness on related tasks, such as classifying whether a vase has a handle [49] using similar data sizes, which made the relative difficulty of the grasp class task curious. In addition, the comparatively better but still middling performance of the classifier over the human annotations, combined with the relatively low agreement between annotators suggested that different humans use different heuristics when assessing the telic qualities of objects.

Previous research [75] into the properties of embedding spaces has demonstrated that, in closed-set tasks where a fixed set of final-layer labels is shared between two networks A and B, some level of interchangeability is in fact expected, up to a matrix $M_{A \to B} \in \mathbb{R}^{d_A} \times \mathbb{R}^{d_B}$ that minimizes the distance between paired points in $\mathbb{R}^{d_A} \times \mathbb{R}^{d_B}$ feature space that correspond to the same label. In a geometric sense, this is equivalent to asking, given two objects A and B, are they likely to be the "same" when deformed under, at most, a warping or affine transformation? Here, however, the objects are not meshes a la a MeshCNN object type classification task, but points in high dimensional vector space. If the grasp classes I derived can in fact be represented as roughly equivalent subspaces in both the human-annotation MLP embedding space and the MeshCNN embedding space, then the poor test performance of MeshCNN could simply be attributed to overfitting to the training data, I set out to evaluate if this was in fact the case.

The hypothesis here was that if the MeshCNN embeddings can be transformed into the MLP embedding space such that the R^2 coefficient of determination is high enough for the training pairs, then poor performance can be attributed to overfitting to the training data. If not, then the

more likely explanation is that the two representation spaces are underlyingly different due to fundamental differences in what the training data itself represents.

I retrieved all 200-dimensional embeddings for each input (training and test) from each of the two classifiers. Because I now wanted to evaluate equivalency between the *trained* embedding spaces, I use all data, including the embeddings representing the training inputs to the respective classifiers, to compute the mapping, in order to minimize the distance between points that each classifier "knows" belong to the correct class. Because there were 350 training inputs to the MeshCNN classifier and only 280 inputs to the human-annotation classifier, I discarded 70 randomly selected extra inputs until I had 1-to-1 paired embeddings representing 56 pairs each per grasp class. I divided these into the same 82:18 train/test split used in the MLP classification, resulting in 46 train and 10 test embedding vectors per grasp class.

To compute the linear mapping, I used an MLP regressor with no hidden layer (i.e., a multivariate linear model) as an affine mapping from one embedding space to another. The MeshCNN embeddings were the inputs and the MLP embeddings, as the classifier with higher accuracy and therefore presumably better-defined subspaces, serve as the outputs. This process maps the individual MeshCNN representations as closely as possible to their MLP-space equivalents. Since all embeddings come from a network whose final layer is a 5-node softmax activation representing the 5 grasp classes, this attempts to align the embedding spaces in which the grasp classes are represented by the respective models as closely as possible.

6.2 Linear Mapping Results

The regressor process was largely unsuccessful in aligning the two embedding spaces. $R^2 = 0.06$ on the training set and when the test embeddings were premultiplied by the computed mapping matrix, $R^2 = 0.02$. Even after mapping, the two embedding spaces remained almost entirely orthogonal. indicating that the two classifiers had learned fundamentally different representations and that the poor test performance of MeshCNN was most likely not due to overfitting on the training data (more on this in Section 6.3).

Figure 6.1 shows a 3D TSNE plot of the vectors extracted from the two embedding spaces *after* the MeshCNN embeddings were mapped into the MLP embedding space.

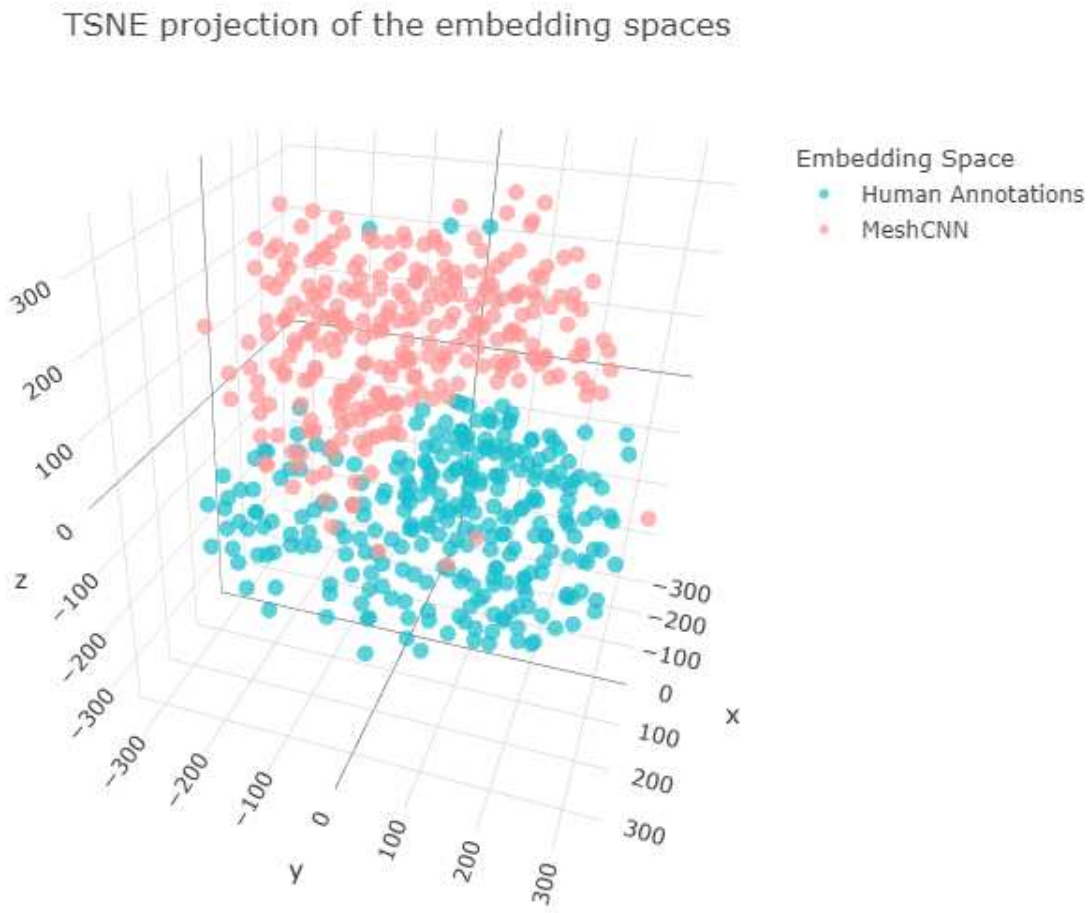

Figure 6.1: 3D TSNE plot of all embedding vectors, colored by original embedding space.

6.3 Discussion

Recall that the ground-truth grasp classes were derived from human annotations of objects that were designed to specifically elicit judgments on the *use or purpose* of the object, i.e., the telic affordance (Section 3.2). 3D meshes alone, however, capture none of this information. There may be some geometric correspondences that correlate with typical purpose-denoting grasps, such as handles, but the geometry itself, being a representation of *structure*, is naturally Gibsonian.

In Figure 6.1, I see two almost completely separable regions. If the MeshCNN classifier had actually overfitted to its training data, I would expect to see far more of the MeshCNN embeddings—the pink points—map closely to a subset of the human annotation embeddings—the blue points—because that is the same data that the output labels were derived from. Instead, the MeshCNN embeddings were mapped into the MLP embedding space (as seen by the fact that the entire convex hull of the MLP embeddings, including outliers at the top of the plot, encompasses nearly all the MeshCNN embeddings), but remain neatly separated from the bulk of the MLP embeddings, including the paired outputs that the input embeddings were trained against.

This suggests that MeshCNN learned certain representations of Gibsonian affordances but, being trained against telic affordance labels, did not have the information available in the structure of the mesh itself to learn appropriate Gibsonian-telic correlations. Meanwhile, the MLP trained over human annotations of telic affordances learned different information. I surmise, therefore, that Gibsonian and telic affordances represent related but fundamentally different ways of interpreting the same sets of objects.

This can be further confirmed by examining the nearest neighbors for specific objects within each embedding subspace.

Figure 6.2 shows the 55 nearest neighbors[3] of a representative *bowl* object (a member of grasp class 3) in each of the respective embedding spaces. In the bottom cluster, marked with circles, showing the data from the human annotation MLP embedding space, I see that the vast majority of nearest neighbor objects are either other bowls or plates (the other member of grasp class 3). There are a few other neighbors belonging to other grasp classes, which I attribute to the disagreement the human annotators themselves showed, but these are overwhelmingly outnumbered by neighbors that belong to the correct cluster.

Meanwhile in the top cluster, marked by squares, the nearest neighbors of a representative *bowl* object are much more diverse, roughly equally divided between more bowls and plates but also bottles, jars, and even teapots.

[3]Because there are 56 objects in a class when training and testing sets are put together.

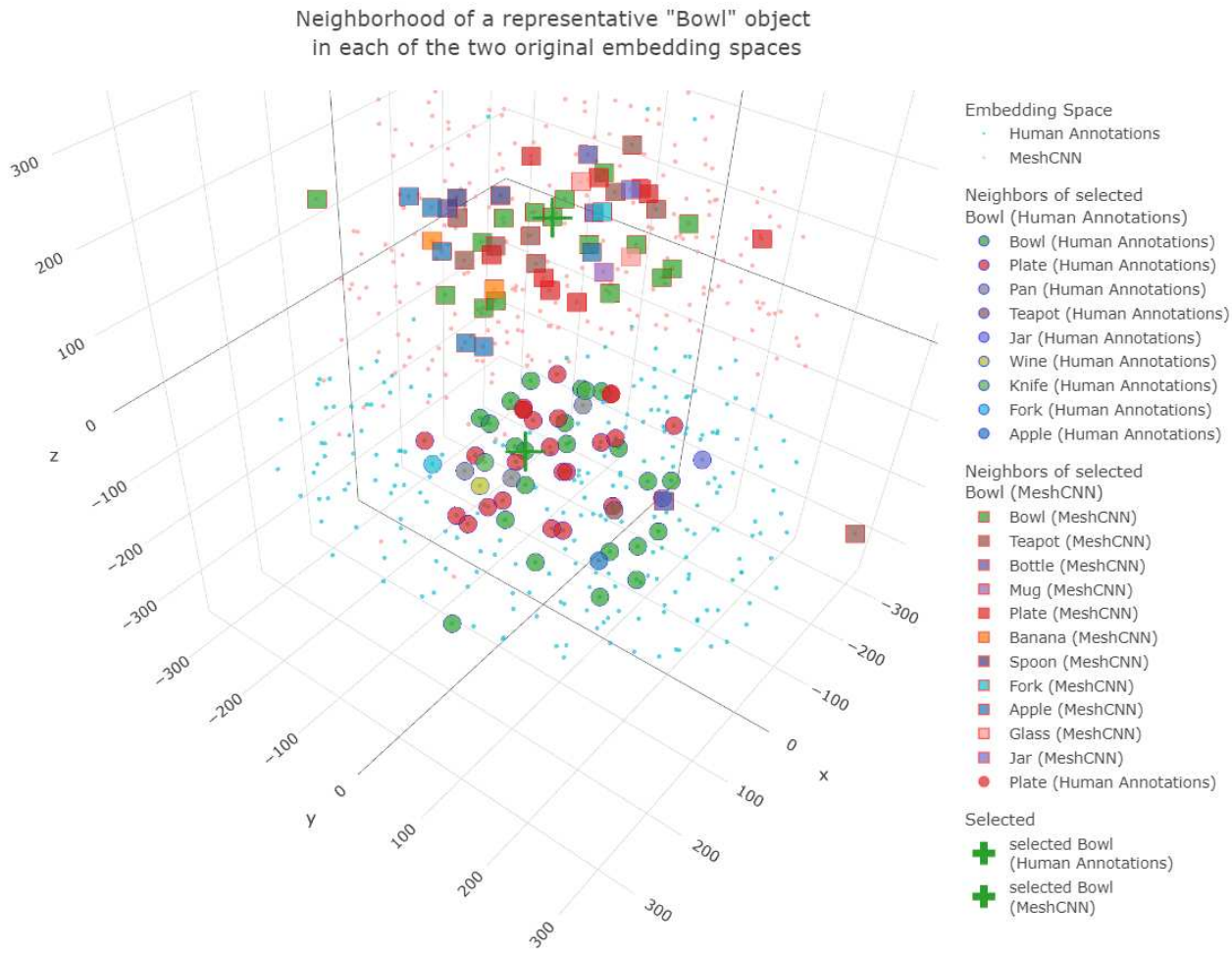

Figure 6.2: Nearest neighbors of representative *bowl* object across both embedding types.

Figure 6.3: Bottle being grasped.

Nevertheless, when grasped for typical use or purpose as the human annotators were asked (e.g., drinking from a bottle vs. filling a bowl), the actual hand pose is markedly different (e.g., see Figure 6.3).

Figure 6.4 shows the nearest neighbors of a *mug* object, which is a member of class 2, one of the classes on which MeshCNN actually performed well. Most neighbors here, among both types of embeddings, are mugs and teapots (the other class 2 member). In fact, among the nearest neighbors of the MeshCNN embeddings are *MLP embeddings* of teapots and mugs, indicating that MeshCNN did learn a representation of the handle geometry correlated with that type of grasp.

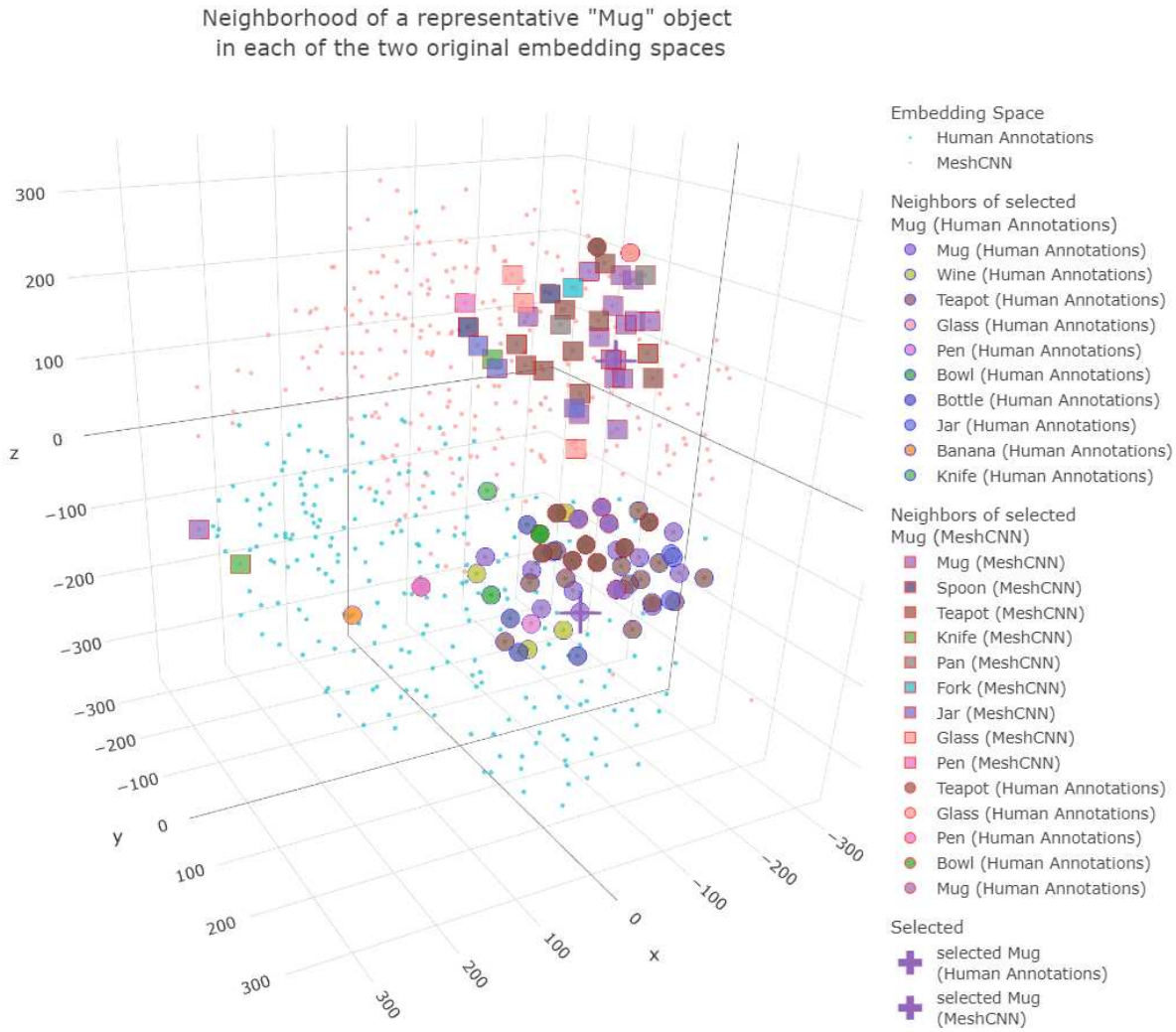

Figure 6.4: Nearest neighbors of representative *mug* object across both embedding types.

Grasping is typically a Gibsonian affordance, based on object structure, as would be encoded in a geometric mesh representation, but grasping for a particular use or purpose implies a *telic* affordance. Encoding information this way, as the human annotations did, appears to result in a markedly different representation from the geometric representation learned by MeshCNN. Even with the same output labels, models trained on this differing data do not appear to be learning equivalent representations that can be correlated to relationships between Gibsonian and telic affordances.

6.4 Implications for Action Recognition

An affordance is not just any action taken with an object (i.e., not every human-object interaction exploits the object's affordances). An affordance is a distinct action possibility that an object allows an agent to take, that is more particular to that object than would arise by chance. In particular, for *human*-object interaction, the human manipulators, i.e., the hands, play a critical role in determining how an object is used. Therefore simple object detection is not enough, and false positives on human-object interaction detection tasks are often the result of detecting the presence of an object in an image when it is *not* being held for typical use or purpose, or in a telic-enabling fashion (e.g., see the discussion of false positives in [76].) The ability to recognize and detect Gibsonian vs. telic affordances and, critically, the difference between the two, will be an important point in future successes in activity recognition and evaluation of human-object interaction tasks.

Chapter 7

Conclusion and Future Work

In this thesis, I hope to have demonstrated that an embodied task like affordance classification is still difficult for even specialized models like MeshCNN that can operate directly over 3D data. I have shown that the problem becomes more difficult if affordances are characterized by a telic/Gibsonian distinction and that even a single afforded behavior such as grasping, when thought about in telic terms, carries quite different information from the same affordance viewed from a purely Gibsonian perspective. A broader implication is that Gibsonian and telic affordances may carry fundamentally different information about an object. Shape, the correlate for Gibsonian affordances encoded in geometries, underspecifies use, or telic affordances,

One specific angle for future work is examining the possibility of getting telic affordance information out of a mesh, which would make both Gibsonian and telic affordances encodable using meshes. When comparing the human annotations to the 3D mesh, one piece of information explicitly singled out by the human annotations was the pose of the hand. This information was nowhere to be found in the 3D meshes. Think about a cup on the table. It can potentially afford anything such as drinking, pushing, etc. The final action cannot be predicted, e.g., by an HOI classifier, unless coupled with a grasp pose. Methods like ContactOpt [77] and HandsFormer [78] offer the possibility of fitting a 3D mesh of a hand to an object in either image or mesh format. Retrieving either the mesh or the joints of the fitted hand may potentially provide mesh or mesh-like 3D information that could be provided to a method like MeshCNN to increase performance on tasks like affordance classification.

Bibliography

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

[2] James Pustejovsky. Dynamic event structure and habitat theory. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)*, pages 1–10, 2013.

[3] Michael Tomasello. Learning through others. *Daedalus*, 133(1):51–58, 2004.

[4] Renaud Detry, Carl Henrik Ek, Marianna Madry, and Danica Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. 05 2013.

[5] John Sweeney and Roderic Grupen. A model of shared grasp affordances from demonstration. pages 27 – 35, 01 2008.

[6] Dirk Kraft, Renaud Detry, Nicolas Pugeault, Emre Baseski, Justus Piater, and Norbert Krüger. Learning objects and grasp affordances through autonomous exploration. volume 5815, pages 235–244, 10 2009.

[7] Yueh-Hua Wu, Jiashun Wang, and Xiaolong Wang. Learning generalizable dexterous manipulation from human grasp affordance, 04 2022.

[8] James Pustejovsky and Nikhil Krishnaswamy. Embodied human computer interaction. *KI-Künstliche Intelligenz*, 35(3):307–327, 2021.

[9] James Pustejovsky and Nikhil Krishnaswamy. Voxml: A visualization modeling language. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4606–4613, 2016.

[10] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

[13] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. E-bert: Efficient-yet-effective entity embeddings for bert. pages 803–818, 01 2020.

[14] Gary Marcus. Gpt-3, bloviator: Openai's language generator has no idea what it's talking about, Dec 2020.

[15] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[16] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[17] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

[18] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

[19] Alessandro Carfì, Timothy Patten, Yingyi Kuang, Ali Hammoud, Mohamad Alameh, Elisa Maiettini, Abraham Itzhak Weinberg, Diego Faria, Fulvio Mastrogiovanni, Guillem Alenyà, Lorenzo Natale, Véronique Perdereau, Markus Vincze, and Aude Billard. Hand-object interaction: From human demonstrations to robot manipulation. *Frontiers in Robotics and AI*, 8, 2021.

[20] Chao Zhuang, Hongjun Zhou, and Shigeyuki Sakane. Learning by showing: An end-to-end imitation leaning approach for robot action recognition and generation. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 173–178, 2016.

[21] Ayumu Sasagawa, Kazuki Fujimoto, Sho Sakaino, and Toshiaki Tsuji. Imitation learning based on bilateral control for humanrobot cooperation. *IEEE Robotics and Automation Letters*, PP:1–1, 07 2020.

[22] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, 2021.

[23] Kanishk Gandhi, Siddharth Karamcheti, Madeline Liao, and Dorsa Sadigh. Eliciting compatible demonstrations for multi-human imitation learning. In *6th Annual Conference on Robot Learning*, 2022.

[24] Manuel Mühlig, Michael Gienger, and Jochen J. Steil. Interactive imitation learning of object movement skills. *Autonomous Robots*, 32:97 – 114, 2011.

[25] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[26] Boyuan Zheng, Sunny Verma, Jianlong Zhou, Ivor W. Tsang, and Fang Chen. Imitation learning: Progress, taxonomies and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–16, 2022.

[27] Xin Chen, Sam Toyer, Cody Wild, Scott Emmons, Ian Fischer, Kuang-Huei Lee, Neel Alex, Steven H Wang, Ping Luo, Stuart Russell, Pieter Abbeel, and Rohin Shah. An empirical investigation of representation learning for imitation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[28] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

[29] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common

sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[30] Khurram Soomro, Amir Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 12 2012.

[31] Hilde Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb51: A large video database for human motion recognition. pages 2556–2563, 11 2011.

[32] Yu-Wei Chao, Yunfan Liu, Michael Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. 02 2017.

[33] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. 05 2015.

[34] Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. Hico: A benchmark for recognizing human-object interactions in images. pages 1017–1025, 12 2015.

[35] Francesco Ragusa, Antonino Furnari, Salvatore Livatino, and Giovanni Farinella. The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain. pages 1568–1577, 01 2021.

[36] Wei Zhai, Hongcheng Luo, Jing Zhang, Yang Cao, and Dacheng Tao. One-shot object affordance detection in the wild. *International Journal of Computer Vision*, 130:2472 – 2500, 2021.

[37] Zeyad Osama Khalifa and Syed Afaq Ali Shah. Towards visual affordance learning: A benchmark for affordance segmentation and recognition. *ArXiv*, abs/2203.14092, 2022.

[38] Chao Xu, Yixin Chen, He Wang, Song-Chun Zhu, Yixin Zhu, and Siyuan Huang. Partafford: Part-level affordance discovery from 3d objects. *ArXiv*, abs/2202.13519, 2022.

[39] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. pages 1778–1787, 06 2021.

[40] Fenggen Yu, Kun Liu, Mei Yan, Chenyang Zhu, and Kai Xu. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. pages 9483–9492, 06 2019.

[41] Tom Ziemke. What's that thing called embodiment? *Proceedings of the 25th Annual meeting of the Cognitive Science Society*, 6, 01 2003.

[42] A.I. Goldman and Oxford University Press. *Simulating Minds: The Philosophy, Psychology, and Neuroscience of Mindreading*. Oxford scholarship online. Oxford University Press, USA, 2006.

[43] Anita Eerland, Tulio M. Guadalupe, and Rolf A. Zwaan. Leaning to the left makes the eiffel tower seem smaller: Posture-modulated estimation. *Psychological Science*, 22(12):1511–1514, 2011. PMID: 22123776.

[44] Hajo Adam and Adam D. Galinsky. Enclothed cognition. *Journal of Experimental Social Psychology*, 48(4):918–925, 2012.

[45] Lynden K. Miles, Louise K. Nind, and C. Neil Macrae. Moving through time. *Psychological Science*, 21(2):222–223, 2010. PMID: 20424050.

[46] Margaret Wilson. Six views of embodied cognition. *Psychonomic bulletin review*, 9:625–36, 01 2003.

[47] Andrew Wilson and Sabrina Golonka. Embodied cognition is not what you think it is. *Frontiers in Psychology*, 4, 2013.

[48] Marinus Maris and R. Boeckhorst. Exploiting physical constraints: heap formation through behavioral error in a group of robots. pages 1655 – 1660 vol.3, 12 1996.

[49] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[50] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. 12 2015.

[51] Mikael Henaff, Joan Bruna, and Yann Lecun. Deep convolutional networks on graph-structured data. 06 2015.

[52] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. 05 2016.

[53] Ayan Sinha, Jing Bai, and Karthik Ramani. Deep learning 3d shape surfaces using geometry images. volume 9910, pages 223–240, 10 2016.

[54] Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir Kim, and Yaron Lipman. Convolutional neural networks on surfaces via seamless toric covers. *ACM Transactions on Graphics*, 36:1–10, 07 2017.

[55] C. Qi, L. Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

[56] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy Jyoti Mitra. Pcpnet learning local shape properties from raw point clouds. *Computer Graphics Forum*, 37, 2018.

[57] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, 2018.

[58] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. 05 2015.

[59] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3d shape segmentation with projective convolutional networks. pages 6630–6639, 07 2017.

[60] Charles Ruizhongtai Qi, Hao Su, Matthias NieBner, Angela Dai, Mengyuan Yan, and Leonidas Guibas. Volumetric and multi-view cnns for object classification on 3d data. pages 5648–5656, 06 2016.

[61] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. pages 1912–1920, 06 2015.

[62] Andrew Brock, T. Lim, James Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. 08 2016.

[63] Lyne Tchapmi, Chris Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. pages 537–547, 10 2017.

[64] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh: A self-prior for deformable meshes. *ACM Trans. Graph.*, 39:126, 2020.

[65] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. Cnns on surfaces using rotation-equivariant features. *ACM Transactions on Graphics (TOG)*, 39:92:1 – 92:12, 2020.

[66] Vitalis Vosylius, Andy Wang, Cemlyn Waters, Alexey Zakharov, Francis Ward, Loïc Le Folgoc, John R. G. Cupitt, Antonios Makropoulos, Andreas Schuh, Daniel Rueckert, and Amir Alansary. Geometric deep learning for post-menstrual age prediction based on the neonatal white matter cortical surface. In *GRAIL@MICCAI*, 2020.

[67] Danilo Avola, Marco Bernardi, Luigi Cinque, Gian Luca Foresti, and Cristiano Massaroni. Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. *IEEE Transactions on Multimedia*, 21(1):234–245, 2018.

[68] Peter J Rousseeuw and Mia Hubert. Robust statistics for outlier detection. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 1(1):73–79, 2011.

[69] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of experimental social psychology*, 49(4):764–766, 2013.

[70] Helena Chmura Kraemer. Extension of the kappa coefficient. *Biometrics*, pages 207–216, 1980.

[71] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[72] Noriko Kamakura, Michiko Matsuo, Harumi Ishii, Fumiko Mitsuboshi, and Yoriko Miura. Patterns of static prehension in normal hands. *The American journal of occupational therapy*, 34(7):437–445, 1980.

[73] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M Dollar, and Danica Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on human-machine systems*, 46(1):66–77, 2015.

[74] Siu-Wing Cheng and Jiongxin Jin. Edge flips in surface meshes. *Discrete & Computational Geometry*, 54(1):110–151, 2015.

[75] David McNeely-White, Benjamin Sattelberg, Nathaniel Blanchard, and Ross Beveridge. Exploring the interchangeability of cnn embedding spaces. *arXiv preprint arXiv:2010.02323*, 2020.

[76] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8359–8367, 2018.

[77] Patrick Grady, Chengcheng Tang, Christopher D Twigg, Minh Vo, Samarth Brahmbhatt, and Charles C Kemp. Contactopt: Optimizing contact to improve grasps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1471–1481, 2021.

[78] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Handsformer: Keypoint transformer for monocular 3d pose estimation ofhands and object in interaction. *arXiv preprint arXiv:2104.14639*, 2021.