

Object-agnostic Affordance Categorization via Unsupervised Learning of Graph Embeddings

Alexia Toumpa

School of Computing, University of Leeds, UK

A.TOUMPA@LEEDS.AC.UK

Anthony G. Cohn

School of Computing, University of Leeds, UK

The Alan Turing Institute, UK

Tongji University, China

A.G.COHN@LEEDS.AC.UK

Abstract

Acquiring knowledge about object interactions and affordances can facilitate scene understanding and human-robot collaboration tasks. As humans tend to use objects in many different ways depending on the scene and the objects' availability, learning object affordances in everyday-life scenarios is a challenging task, particularly in the presence of an open set of interactions and objects. We address the problem of affordance categorization for class-agnostic objects with an open set of interactions; we achieve this by learning similarities between object interactions in an unsupervised way and thus inducing clusters of object affordances. A novel depth-informed qualitative spatial representation is proposed for the construction of *Activity Graphs* (AGs), which abstract from the continuous representation of spatio-temporal interactions in RGB-D videos. These AGs are clustered to obtain groups of objects with similar affordances. Our experiments in a real-world scenario demonstrate that our method learns to create object affordance clusters with a high V-measure even in cluttered scenes. The proposed approach handles object occlusions by capturing effectively possible interactions and without imposing any object or scene constraints.

1. Introduction

In the literature, the meaning of the term *affordance* of an object differs depending on the context. In robotic applications, *e.g.* robot manipulation tasks, the definition of *affordance* is bound to the part of a tool, which can be afforded in a specific way, *e.g.* the handle of a hammer has the affordance of 'hold' whereas the head has the affordance of 'hit', the inner surface of a cup has the affordance of 'contain' although the cup's handle has the affordance of 'hold'. In contrast, in human-object interaction recognition tasks, *affordance* is defined as the way an object can be utilized by the human in a scene, *e.g.* if a human uses a cup for containing something then the cup will have the affordance of 'contain'. Moreover, an object may have more than one affordance as it depends on the purpose it is being used for, *e.g.* a pizza box can have the affordance of 'contain' when it is being utilized as a container of a pizza or 'support' when it plays the role of a tray. Such multi-labeled affording objects can be recognized by considering their interactions with other objects.

In a human-robot collaboration scenario, acquiring knowledge of the affordances of the objects in a scene is crucial for aiding the human, *e.g.* assisting the human when performing a physically hard task. This becomes challenging when the scene comprises an open-set of objects and the affordance space enlarges. Moreover, in human action prediction tasks, the

affordances of the objects carry useful information for the prediction of the future action, and are highly correlated with the rest of the objects the human interacts with. Nevertheless, such knowledge is not easy to obtain as humans tend to use the same object in different ways depending on the task, thus changing its primary affordance.

Object *affordances* were first formally defined by Gibson (1977); however the concept of object affordances as it is understood in Computer Science has not been explicitly defined in the literature, causing some confusion. For this purpose we propose the definition:

Definition 1 *An affordance is a property of an object arising from its interaction with another entity, i.e. agent, object. It is correlated with the occurring interaction as every interaction exploits at least one object affordance. An object may have multiple affordances based on the various interactions it may have with other entities.*

Based on this definition, we address the problem of affordance inference by exploiting object interactions through RGB-D video data. Hence, we rely on the way objects are being utilized by the human agents in a scene, allowing objects to support different kinds of affordances at the same time. Thus, we focus on learning groups of high-level object interactions, which take into account their spatio-temporal relations from extracted visual appearances. Graphs are able to represent both static and dynamic high level relations between objects. Accordingly, we embody these object interactions through a high-level graphical structure, the *Activity Graph*, abstracting from the continuous spatio-temporal representation and acquiring depth-informed qualitative spatial relations between object pairs. With these high-level qualitative relations, our graphical structures for capturing interactions are not dependant on scene-specific characteristics, *i.e.* the objects’ labels, distance between objects, etc.

Definition 1 states that “*every interaction exploits at least one object affordance*”; affordances of objects are inferred from these high-level graphs representing pair-wise object interactions. Affordance clusters are formed in an unsupervised way by exploiting graph similarity using the cosine similarity measure of their embeddings in a learned latent space. By clustering graph structures, a hierarchical tree representation is produced demonstrating their similarity. Since our approach is based on learning a high-level representation of interactions, it is not limited to any number or kind of affordances, scenes, and objects.

To obtain a richer set of spatial relationships than those possible from a sequence of purely 2D frames, we exploit the depth information assuming the presence of RGB-D video data. The depth cues allow some inference about the morphology of the objects in the scene and thus the possible ways they can interact with other objects, *e.g.* non-concave objects can not act as *containers*.

Hence, the objectives of this work are ¹:

- to propose a depth-informed set of qualitative relations, which can describe a wide variety of object configurations regarding a real-world scenario as well as differentiating between object occlusions and interactions, for detecting effectively spatial relationships of objects;

1. Preliminary versions of this work in unpublished workshop papers can be found in Toumpa and Cohn (2019) and Toumpa and Cohn (2020).

- to introduce a novel unsupervised affordance categorization framework, which handles an open-set of interactions and affordances, considering high-level information of the human-object and object-object interactions;
- to present a novel human-object interaction dataset, which focuses on different object affordances.

2. Literature Review

Qualitative spatial relations Qualitative relations are mostly used to represent semantically meaningful properties of a perceived *event*, *i.e.* interaction, deformation, whilst abstracting away from any non-relevant information, *i.e.* pixel values, numeric measurements. Spatial relation formalisms comprise: mereotopology, direction, distance, shape, relationships between static spatial entities, and moving objects (Chen et al., 2015; Dylla et al., 2017; Landsiedel et al., 2017). Due to the nature of the problem we address in this work, we focus on mereotopological information for representing human-object and object-object interactions and shape information for defining possible concave regions on the detected objects.

The Region Connection Calculus (RCC) (Cohn et al., 1997; Gerevini & Renz, 2002) and the n-intersection model (Egenhofer, 2005; Egenhofer & Franzosa, 1991, 1995; Egenhofer & Herring, 1990; Egenhofer & Sharma, 1993; Egenhofer & Vasardani, 2007; Egenhofer et al., 1994, 1994) are the most well-known and well-studied approaches for representing topological information. However, as both sets of relations only rely on the spatial location of the detected entities, *i.e.* objects, no occlusion can be differentiated from an actual interaction of containment or support. Galton’s *Lines of Sight* (Galton, 1994), the *Region Occlusion Calculus* (Randell et al., 2001), and *The Occlusion Calculus* (OCC) (Köhler, 2002) aim to solve the occlusion problem in 2D space. However, relations as ‘contain’ and ‘support’ are still not detectable. Thus, shape characteristics are crucial for detecting more relevant relations to object interactions.

Qualitative shape information can be obtained by using region-based methods (Cohn, 1995) or boundary-based methods (Leyton, 1988; Galton & Meathrel, 1999; Gottfried, 2003a, 2003b, 2004). Both kind of methods depend on the 2D shapes of the detected entities. By exploiting the depth cues of the entities, concave regions are discovered in the 3D space, indicating possible containment.

Inspired by the RCC8 spatial relations and the *Process-Grammar* (Leyton, 1988), in this work we propose a novel set of mereogeometrical relations that exploit the concave regions indicated by the depth cues of the detected entities. Our new set of relations define more complex and concave-informed relations, such as ‘contain’ and ‘support’.

Object affordances Several methods have been proposed for detecting functional object parts and their corresponding affordance labels. These works involve the detection of object affordance parts by considering their visual and geometric features (Deng et al., 2021; Wang & Tarr, 2020). One of the early works in this direction focused in the detection of graspable object areas by creating local visual descriptors of grasping points and estimating the probability of the presence of a graspable object based on the Bernoulli trial (Montesano & Lopes, 2009). New approaches employ Convolutional Neural Network (CNN) models to produce classes of functional object parts from RGB (Nguyen et al., 2017; Do et al.,

	Input data type	Information exploited ^a				Solve task ^b		
		L.F.	H.A.	H.F.	O.T.	O.O.	O.S.A.	O.S.O.
(Montesano & Lopes, 2009)	2D & Keypoints	✓						
(Zhao & Zhu, 2013)	2D & 3D	✓						
(Myers et al., 2015)	2.5D	✓				✓		
(Nguyen et al., 2016)	2.5D	✓						
(Nguyen et al., 2017)	2D	✓						
(Kokic et al., 2017)	Synthetic	✓				✓		
(Sawatzky et al., 2017)	2D & Keypoints	✓						
(Do et al., 2018)	2D	✓						
(Wang & Tarr, 2020)	2D	✓						
(Deng et al., 2021)	3D	✓				✓		
(Xu et al., 2021)	2.5D & Keypoints	✓				✓		
(Turek et al., 2010)	2D				✓		✓	✓
(Qi et al., 2018)	3D			✓				
(Kjellström et al., 2011)	2D		✓			✓	✓	
(Yao et al., 2013)	3D		✓					
(Qi et al., 2017)	2.5D		✓	✓				
(Gkioxari et al., 2018)	2D		✓					
(Fang et al., 2018)	2D			✓				
(Chuang et al., 2018)	2D		✓	✓				
(Tan et al., 2019)	2D		✓	✓				
(Wu et al., 2020)	Synthetic		✓					
(Hou et al., 2021)	2D		✓					
(Sridhar et al., 2008)	2D			✓			✓	✓
(Aksoy et al., 2010)	2D			✓			✓	
(Aksoy et al., 2011)	2D			✓			✓	
(Pieropan et al., 2013)	2.5D			✓			✓	
(Pieropan et al., 2014)	2D			✓	✓			
(Moldovan & De Raedt, 2014)	Synthetic					✓		
(Liang et al., 2016)	2.5D			✓	✓	✓		
(Liang et al., 2018)	2.5D				✓	✓		
This work	2.5D			✓		✓	✓	✓

^a L.F.: Low-level features H.A.: Human Actions H.F.: High-level Features (graph representations, embeddings) O.T.: Object tracks

^b O.O.: Object Occlusions O.S.A.: Open-set of affordances O.S.O.: Open-set of objects

Table 1: Related works on object affordance detection.

2018; Sawatzky et al., 2017) and synthetic data (Kokic et al., 2017). However, depth cues along with the RGB information have demonstrated a greater detection accuracy in this task (Nguyen et al., 2016; Myers et al., 2015; Xu et al., 2021). Additionally, incorporating knowledge about the scene and context in which an object is being used boosts the prediction accuracy even more (Zhao & Zhu, 2013).

However, processing static visual information restricts the number of affordances assigned to an object to be the ones correlated only with its visual features. For this purpose, many works have considered exploiting the correlation of human actions and the detected objects in a scene (Gkioxari et al., 2018; Yao et al., 2013; Fang et al., 2018; Kjellström et al., 2011; Hou et al., 2021; Wu et al., 2020). Depending on the human-object interaction being held, a different affordance is detected. Also, prediction of object affordances, cast as human-object interactions, utilize graph-based convolutional neural networks, which are trained on Object Affordance Graphs (OAG), constructed by the spatio-temporal relations between human and objects/scenes across the input video data (Tan et al., 2019). Similarly, a spatio-temporal And-Or graph (ST-AOG) exploits the context captured by objects, actions, and affordances to solve the task of activity understanding (Qi et al., 2017). Another approach predicts action-object affordances based on the contextual information of the scene, whilst deploying a graph to represent the objects in the scene as the nodes and their spatial relations as the edges (Chuang et al., 2018). These works demonstrate that by fusing knowledge about the scenario in which an interaction takes place facilitates the prediction of affordances, however limits the generalizability across different domains.

To facilitate domain independence, high-level graph representations of interactions are sometimes employed. Recent works introduce such graphical structures in synthetic indoor environments focusing on the prediction of furniture areas the human is most likely to interact with (Qi et al., 2018), whereas outdoors scenes are examined by considering the behavior of moving objects around them (Turek et al., 2010). Nevertheless, these approaches only consider a one-to-one mapping of affordances and objects, hence the objects are bound to a single kind of interaction, not permitting multi-labeled object affordances.

Though graphical structures are able to extract high-level information about the interactions in the scene, their structure might be the cause of domain restriction (Aksoy et al., 2010, 2011; Pieropan et al., 2014, 2013). For this purpose, qualitative spatio-temporal relations are exploited for their construction (Sridhar et al., 2008). One of the fundamental obstacles in these works is object occlusion. To mitigate this problem, the tracks and visual appearances and disappearances of non-deformable objects are considered (Liang et al., 2018, 2016; Moldovan & De Raedt, 2014). However, these approaches are restricted to the detection of a single object affordance, *i.e.* containment, and prone to false positive detections of the containment relation due to occlusions. Table 2 presents the related works of object affordance detection and prediction.

In this work, we propose a novel method for predicting object affordances in real-world scenarios with object occlusions. Different from any other published work, our approach is not restricted to a predefined set of objects, interactions, or scenes. High-level graphs assist in considering an open-set of interactions and no object labels restrict the generalizability of our method. Our experiments demonstrate that, without any supervision, we acquire homogeneous and complete clusters of object affordances by exploiting qualitative information about their interactions and their shapes.

RCC2	Definition	Description
$C(\alpha, \beta)$	$mask(\alpha) \cap mask(\beta) \neq \emptyset$ [†]	α and β are connected
$DC(\alpha, \beta)$	$\neg C(\alpha, \beta)$	α is disconnected from β

[†] $mask(z)$ defines object z 's region in the image plane.

Table 2: RCC2

3. Background

3.1 Activity Graphs

Relational graph structures represent high-level information by abstracting from the continuous space of the exploited relations. From definition 1: “*It (an affordance) is correlated to the occurring interaction as every interaction exploits at least one object affordance.*”, hence an object-object and human-object interaction reveals each object’s affordances. Relational graph structures of object interactions aid in representing an open set of interactions. Thus, in this work we exploit the representation of *Activity Graphs (AGs)* (Sridhar et al., 2010), which are relational graphs that describe the interaction between entities, *i.e.* objects and human body parts, of a scene by considering their qualitative spatio-temporal relationships.

An *AG* consists of three layers of vertices, where each layer comprises a single type of node and only nodes in adjacent layers can be connected with each other. The three layers of an *AG* are: 1) the *entity* layer, which contains the set of vertices of the entities which interact (V_{ent}), 2) the *spatial* layer, which consists of vertices with the spatial relations (V_{spat}) which describe the spatial interactions of the entities in V_{ent} , and 3) the *temporal* layer, which specifies the temporal relations between the *spatial* layer nodes (V_{temp}).

In this work, *AGs* are deployed to capture object-object and human-object interactions, thus two different qualitative spatial relations (QSRs) are considered, one for each kind of interaction. A set of novel QSRs, introduced in §5, is employed to describe the spatial relationships between objects, whereas the RCC2 set of relations (Table 2) (Randell et al., 1992; Cohn et al., 1997) are used for representing interactions between objects and humans. These two spatial relational sets comprise the V_{spat} layer of our *AGs*, whilst *Allen’s temporal algebra* (Allen, 1983) is exploited to express the temporal relationships between the spatial relations (V_{temp}). This qualitative temporal set consists of the relations²: ‘before’ (<, >), ‘meets’ (m, mi), ‘overlaps’, (o, oi), ‘starts’ (s, si), ‘during’ (d, di), ‘finishes’ (f, fi), and ‘equals’ (=). The set of qualitative spatial relations for every interaction is obtained from the presence of *episodes*. An *episode* is the maximum period of time where a single spatial relation between two entities holds, while a different spatial relation occurs before and after the defined time period. By considering *episodes* instead of individual frames for detecting the spatial relationships of the objects, our method can effectively generalize across different video fps and is not video length dependent.

We define an *Activity Graphlet (AGraphlet)* of an object as a sub-graph of an *AG*, which carries the spatial and temporal information (V'_{spat}, V'_{temp}) of the respective object’s

2. Inverse relations end in i.

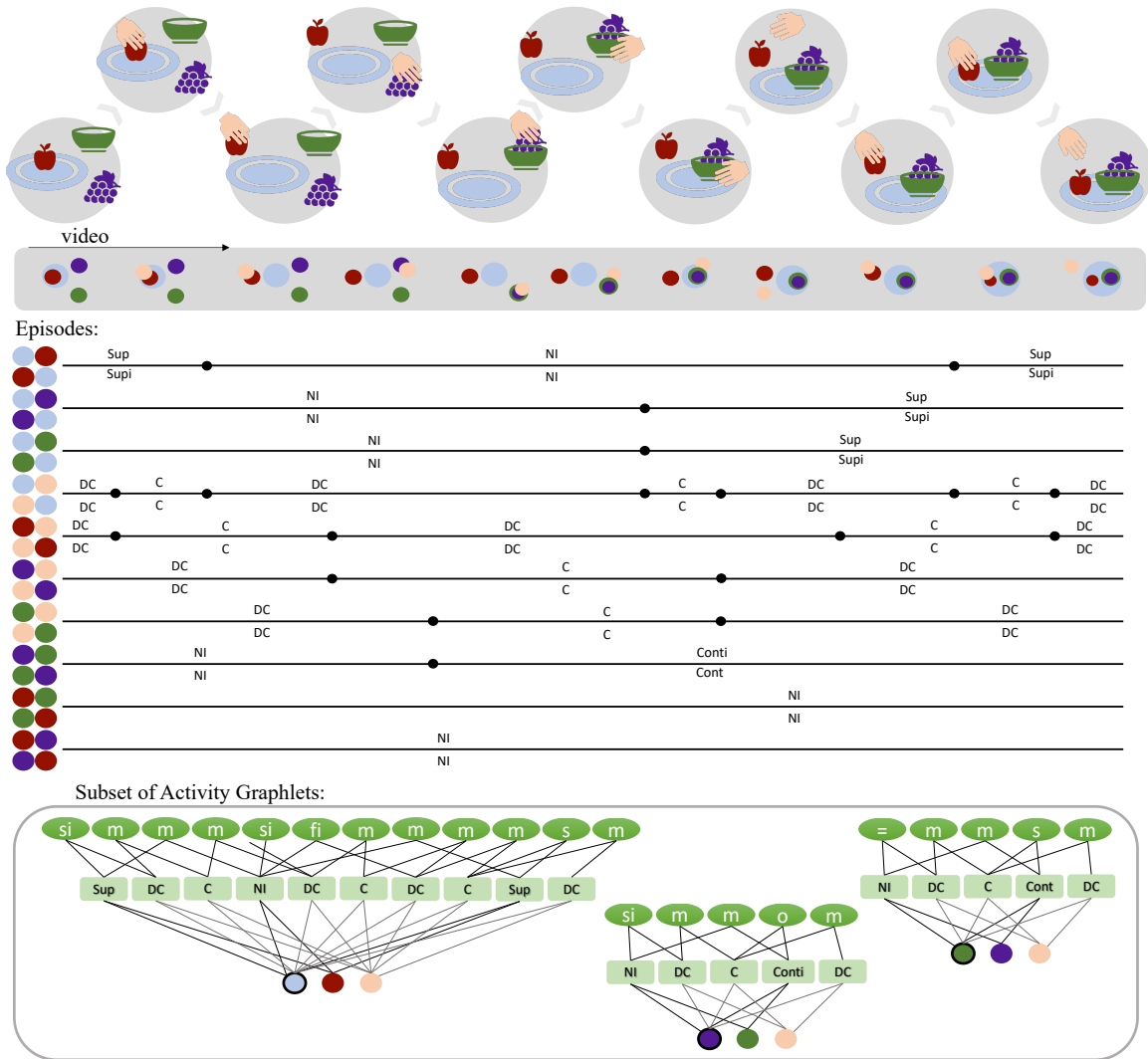


Figure 1: (best viewed in color) Qualitative spatial relations are extracted from detected episodes in the temporal domain of a video. *Activity Graphlets* are constructed using these qualitative spatio-temporal relations for individual detected objects (encircled) in the scene describing their interaction with another object and a human body part, *e.g.* human hand.

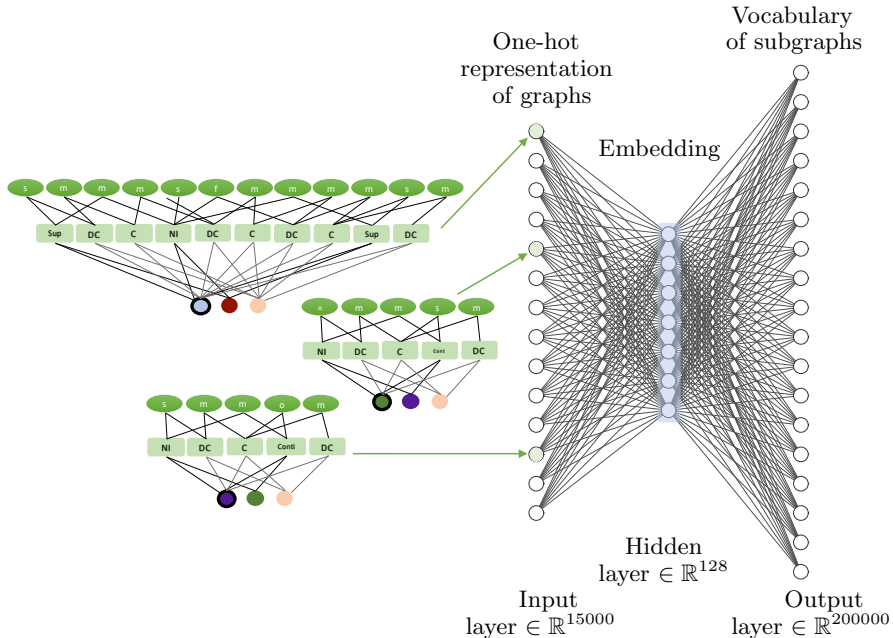


Figure 2: *Graph2vec* network; the size of the depicted network is specific to one of the folds used for training.

interactions with another object and a human body part (V'_{obj}). Figure 1 illustrates an example where *AGraphlets* are extracted from a video sequence of three interacting entities.

3.2 Graph Embedding

Graph embeddings are d -dimensional vector representations of projections of graphs in a d -dimensional latent space. By projecting graphs in a multi-dimensional space we are able to find similarities/dissimilarities between them by considering distance measures on their vector representations.

Inspired by the Natural Language Processing literature, we exploit the *graph2vec* network (Narayanan et al., 2017) to learn graph embeddings in an unsupervised way and project the *AGraphlets* in the learned latent space. The network exploits the notion of context, where context is defined as a fixed number of subgraphs comprising every *AGraphlet*. *AGraphlets* with similar subgraphs represent similar affordances. Figure 2 shows an illustration of the *graph2vec* network with the *AGraphlets* as input data.

We refer the reader to §8.4 for more details about the hyper-parameter’s selection and training of the *graph2vec* network.

4. Overview

The proposed approach solves the problem of object affordance categorization in an unsupervised way, by exploiting depth-informed qualitative information and representing object interactions in a high-level latent space. Figure 3 summarizes the proposed approach.

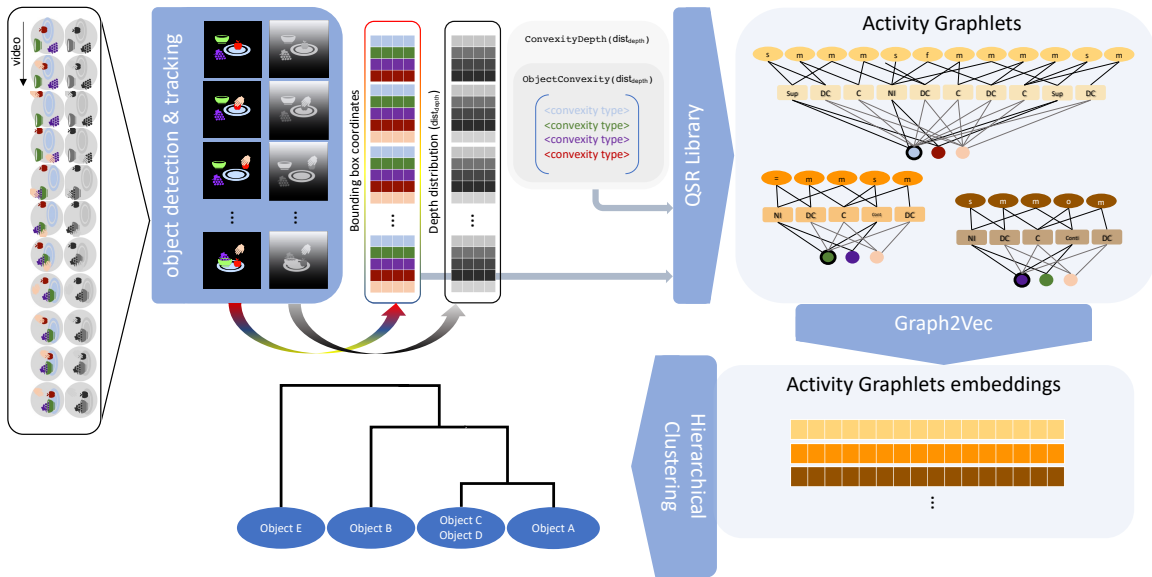


Figure 3: Overview of the proposed approach. Object coordinates and their convexity type are extracted from RGB-D video data per frame and employed for the construction of *AGraphlets* capturing the proposed set of relations. The Graph2Vec network is trained to project these graphs on a latent space. Graph embeddings of *AGraphlets* are then hierarchically clustered to create groups of objects with similar affordances.

Firstly, objects of interest are localized in the input frames of a video by utilizing any visual-based object detector which provides class-agnostic object bounding boxes. We capture object-object and human-object interactions and we effectively represent qualitative spatial relationships between them by exploiting a novel set of depth-informed spatial relationships (§5) and the RCC2 relation set. Such qualitative spatial information constitutes the spatial layer of an *AGraphlet*. *AGraphlets* are used as a high-level representation of object interactions, thus achieving a generalization across various scenarios. These graphical structures are then employed to train the *graph2vec* network, which projects each *AGraphlet* to a latent space. Hierarchical clustering, which is described in §6, is employed for learning in an unsupervised way clusters of detected object interactions. By exploiting graph embeddings, which are projections of *AGraphlets* onto the latent space, the clustering produces a dendrogram of object affordances. As, our method is not object-dependent and not scene-constrained, we perform evaluation on the newly presented Leeds Object Affordance Dataset (LOAD), where the groundtruth affordances do not correlate to the objects’ primary function.

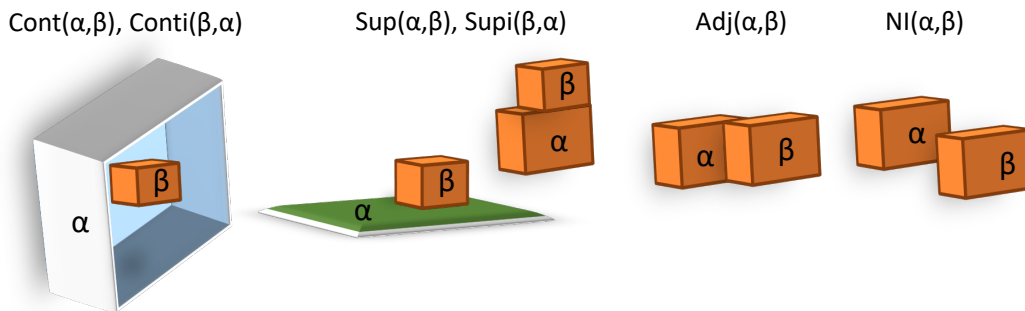


Figure 4: DiSR representations in 3D space.

5. Depth-informed QSR

5.1 Formulation of Depth-informed Spatial Relations (*DiSR*)

Effectively, detecting simple and discrete spatial interactions *e.g.* ‘touching’, ‘not touching’ as well as more complex ones *e.g.* ‘supporting’, ‘containing’, in the 2D image plane, is challenging, especially when there is object occlusion. To address this limitation we propose a depth-informed set of qualitative spatial relationships, which enable the differentiation between objects’ visual occlusion and actual interaction, and take into account the object’s convexity-type. The object’s convexity-type carries information about the object’s affordance, hence the kind of interaction which can hold between that object and another. For example, the relation *contains* can hold between a concave object, due to its concavity, and another object when interacting in a specific way. However in an interaction between objects without any visual concave curve, such a relation will not hold.

The spatial information employed to describe the object interactions in the spatial domain consists of a discrete set of spatial relations that can describe any spatial configuration of the objects in a scene. These depth-informed interactions are exploited for the construction of more accurate qualitative graphical structures, achieving more homogeneous and complete affordance clusters.

Previous work on affordances using QSRs (Sridhar et al., 2008) has used the purely mereotopological RCC set of relations (Randell et al., 1992; Cohn et al., 1997). Here we propose a more expressive set of relations (DiSR), which exploit knowledge that can be garnered from a 2.5D representation: objects’ convexity type and their orientation in the vertical plane. We propose the set of *Depth-informed Spatial Relations (DiSR)* (Fig. 4)³: ‘supports’ (Sup, Supi), ‘contains’ (Cont, Conti), ‘adjacent’ (Adj), ‘not interacting’(NI).

Depth information is considered for reasoning about the spatial relative positions of objects in 3D space, hence enabling the distinction between occlusions and interactions. Furthermore, from depth cues we can effectively infer the convexity-type of the detected objects, and along with the corresponding depth differences, DiSR relations are extracted. For a spatial interaction to hold, an overlap on the depth dimension must be evident as well as the x, y dimension, and depending on the convexity-type of the interacting objects and

3. The inverse of a relation is denoted with an i at the end, wherever an inverse exists, *e.g.* the inverse of Sup is Supi.

DiSR	Definition	Description
$\text{Sup}(\alpha, \beta)$	$(\text{DPO}(\alpha, \beta) \wedge \text{Surface}(\alpha)) \vee \text{On}(\beta, \alpha)$	α supports β
$\text{Cont}(\alpha, \beta)$	$\text{P}(\beta, \alpha) \wedge \text{Concave}(\alpha) \wedge \text{DPP}(\beta, \alpha)$	α contains β
$\text{Adj}(\alpha, \beta)$	$\text{O}(\alpha, \beta) \wedge \text{DPO}(\alpha, \beta) \wedge \neg \text{Cont}(\alpha, \beta) \wedge \neg \text{Cont}(\beta, \alpha) \wedge \alpha$ is adjacent to β $\neg \text{Sup}(\alpha, \beta) \wedge \neg \text{Sup}(\beta, \alpha)$	
$\text{NI}(\alpha, \beta)$	$\neg \text{Sup}(\alpha, \beta) \wedge \neg \text{Sup}(\beta, \alpha) \wedge \neg \text{Cont}(\alpha, \beta) \wedge \neg \text{Cont}(\beta, \alpha) \wedge \alpha$ does not interact with β $\neg \text{Adj}(\alpha, \beta) \wedge \neg \text{Adj}(\beta, \alpha)$	

P and O of RCC relations

$\text{P}(\alpha, \beta)$	$\forall z (\text{C}(z, \alpha) \rightarrow \text{C}(z, \beta))^\dagger$	α is a part of β
$\text{O}(\alpha, \beta)$	$\exists z (\text{P}(z, \alpha) \wedge \text{P}(z, \beta))$	α overlaps β

† C relation is defined in Table 2.

Table 3: DiSR

in which part of the depth information the overlap appears, a different qualitative spatial relation holds from the set of DiSR.

DiSR spatial relations take into account the depth information of the detected object masks in the scene as well as their 2D location considering the bounding box enclosing the detected mask of every object. We present in Table 3 the definition of every DiSR relation inspired by the RCC set, whilst exploiting the RCC relations: ‘overlap’ (O) and ‘part’ (P, Pi) computed in the 2D image plane.

The convexity type of an object, being either *concave*, *surface* or *convex*, is described further in §5.2. ‘Depth Overlap’ (DPO) and ‘Depth Proper Part’ (DPP, DPPi) are primitive relations which hold between the objects’ depth information, defined as:

$$\begin{aligned} \text{DPO}(\alpha, \beta) &\equiv ((dmax_\alpha \geq dmin_\beta) \wedge (dmax_\alpha < dmax_\beta) \wedge \\ &\quad (dmin_\alpha < dmin_\beta)) \vee ((dmax_\beta \geq dmin_\alpha) \wedge \\ &\quad (dmax_\beta < dmax_\alpha) \wedge (dmin_\beta < dmin_\alpha)) \\ \text{DPP}(\alpha, \beta) &\equiv (dmax_\alpha > d_cmin_\beta) \wedge (dmax_\alpha \leq d_cmax_\beta) \wedge \\ &\quad (dmin_\alpha \geq d_cmin_\beta) \wedge (dmin_\alpha < d_cmax_\beta) \end{aligned}$$

where $dmax$ and $dmin$ are the maximum and minimum depth values, respectively, by considering the depth cues of the detected object’s mask, and d_cmax and d_cmin are the maximum and minimum depth values of the detected concave curve area. Figure 5 (right) illustrates a concave object’s concave curve area ($[d_cmin, d_cmax]$), where ascending-ordered depth-pixel values are on the x -axis and their depth values on the y -axis. The values of d_cmin and d_cmax are obtained by Alg. 2, which is further explained in §5.2.

Moreover, the primitive spatial relation On is defined as:

$$\begin{aligned} \text{On}(\alpha, \beta) &\equiv \text{O}(\alpha, \beta) \wedge ((ymax_\alpha \geq ymax_\beta) \wedge \\ &\quad (ymin_\alpha \geq ymin_\beta) \wedge (xmax_\alpha \leq xmax_\beta) \wedge (xmin_\alpha \geq xmin_\beta)) \end{aligned}$$

where $(xmin, ymax)$ and $(xmax, ymin)$ are the top-left and bottom-right corners of the detected object’s bounding box (Fig. 5 (left)).

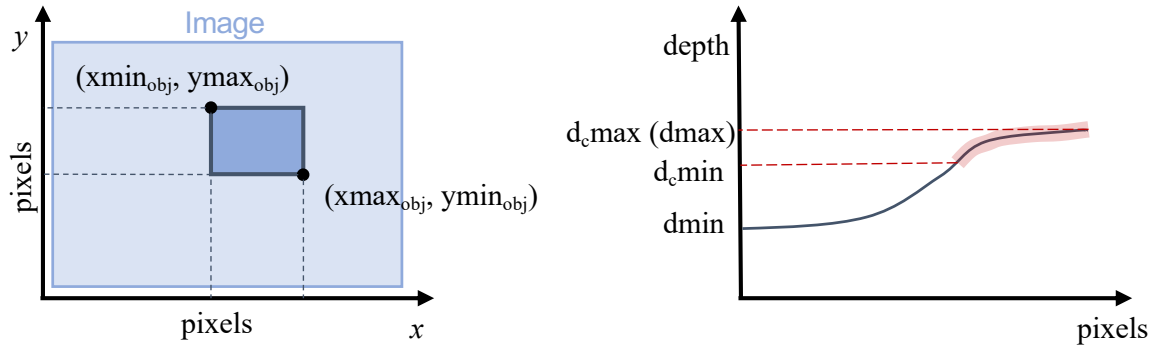


Figure 5: (left) Detected object bounding box coordinates. (right) Object depth distribution and the depth information signifying potential concave curve detection between the values d_{cmin} and d_{cmax} (red).

Algorithm 1 Compute the convexity type of an object.

Given: $thresh_{convex}$

- 1: **procedure** OBJECTCONVEXITY($dist_{depth}$)
- 2: $dmax \leftarrow max(dist_{depth}); dmin \leftarrow min(dist_{depth})$
- 3: $C \leftarrow ContourHierarchy(dist_{depth})$
- 4: **if** $((dmax - dmin) < thresh_{convex}) \ \& \ (C.child() \text{ exists})$ **then**
- 5: $object_{type} \leftarrow concave$
- 6: **else**
- 7: **if** $(dmax - dmin) < thresh_{convex}$ **then**
- 8: $object_{type} \leftarrow surface$
- 9: **else** $object_{type} \leftarrow convex$
- 10: **return** $object_{type}$
- 11: **end**

The definitions proposed are an approximation to the meaning of the English descriptions in Table 3; it is possible to satisfy the definitions with configurations which do not accord with intuitive meaning of the English word. *E.g.* in the case where three objects are stacked the one on top of the other, then in the 2D image plane $On(top\ object, bottom\ object)$ will be True even though there is a *middle object* in between. However, these definitions are easy to compute and work well in the everyday scenes we have considered so far. In future work we may refine these relationships to more closely correspond to the semantics of the English words.

5.2 Object Convexity Type Detection

By considering the depth information we obtain knowledge about the *indentation area* (m-) and *protrusion area* (M+) of an object, as defined in the *Process-Grammar* (Leyton, 1988). More specifically, visually ‘convex’ type objects do not appear to have any *indentation areas* whereas ‘concave’ type objects are characterized by their concavity curve, which

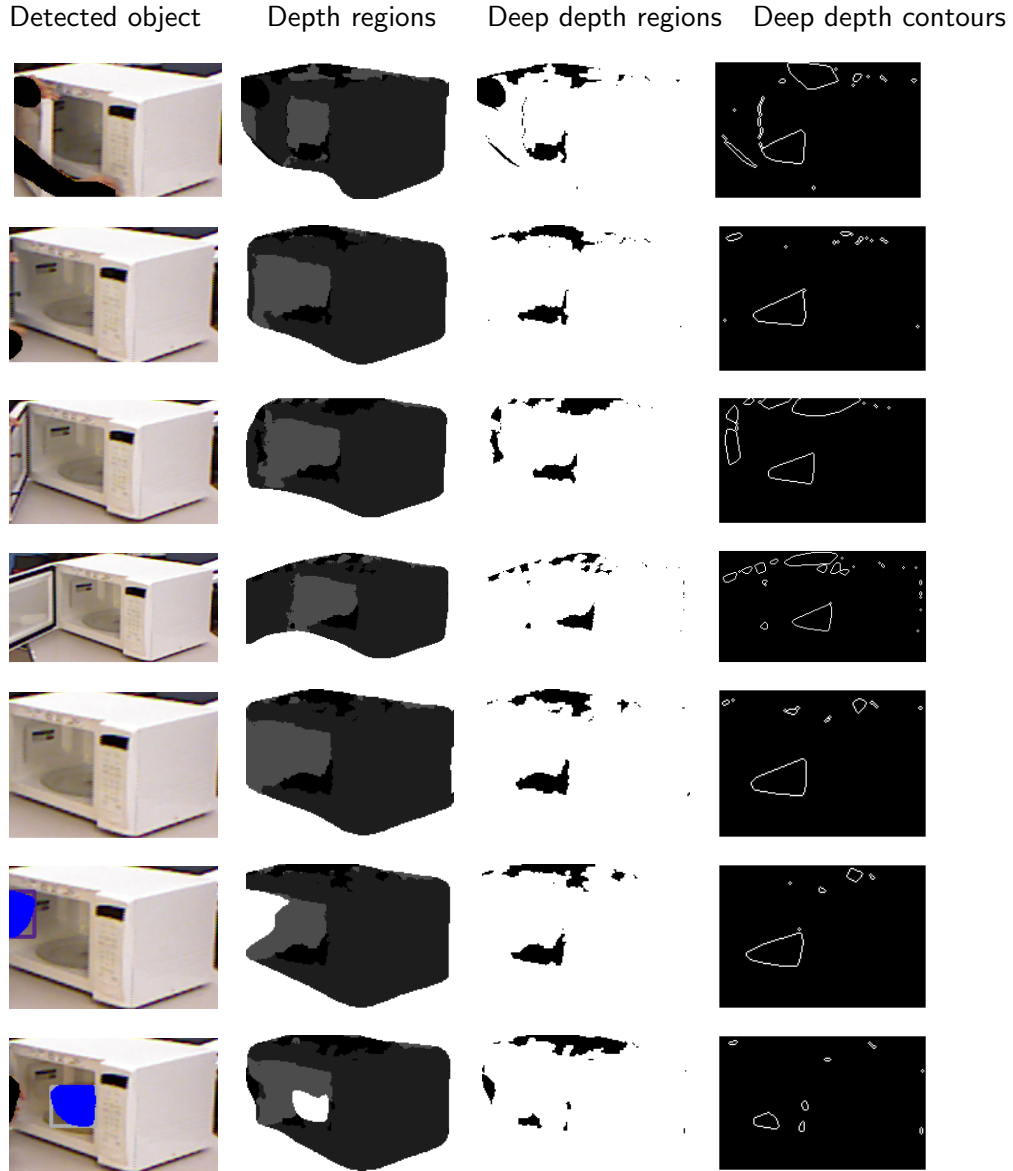


Figure 6: Processing steps for extracting depth contours. The bounding boxes of the detected objects in the RGB frames are mapped onto the depth data from where we extract all the depth information partitioned into sections depending on their depth values. Finally, from that depth data, deep regions are considered from which depth contours are created indicating potential *indentation areas*, thus this object is classified as ‘concave’. The extraction of deep depth regions is based on the employment of the $thresh_{convex}$ threshold. Any information indicating another detected object occluding the primary one, is discarded; the last two rows illustrate the presence of a bowl object (blue) whose depth information is not considered as shown in the ‘Depth regions’ column.

Algorithm 2 Define min and max depth of an object’s concavity.

Given: h, n

- 1: **procedure** CONVEXITYDEPTH($\text{dist}_{\text{depth}}$)
- 2: $d_{\text{max}} \leftarrow \text{max}(\text{dist}_{\text{depth}})$; $d_{\text{min}} \leftarrow \text{min}(\text{dist}_{\text{depth}})$
- 3: $\text{object}_{\text{type}} \leftarrow \text{OBJECTCONVEXITY}(\text{dist}_{\text{depth}})$
- 4: **if** $\text{object}_{\text{type}} = \text{concave}$ **then**
- 5: $\text{sections} \leftarrow (d_{\text{max}} - d_{\text{min}})/h$
- 6: $d_c \text{max} \leftarrow d_{\text{max}}$
- 7: $d_c \text{min} \leftarrow d_{\text{max}} - (n * \text{sections})$
- 8: **else** $d_c \text{max} \leftarrow d_{\text{max}}$; $d_c \text{min} \leftarrow d_{\text{min}}$
- 9: **return** $d_c \text{max}, d_c \text{min}$
- 10: **end**

morphologically appears as a bay-formation described as M+m-M+. Such information is critical to ascertain a relation when two objects interact in the 2D image plane, *e.g.* a Cont DiSR occurs between one or more ‘concave’ type objects when the depth information of the *containee* confirms that is between m- and M+ areas of the *container*. In this section we will demonstrate how we derive the convexity type of the detected objects from the depth cues.

We identify three primary convexity type objects: ‘concave’, ‘convex’, and ‘surface’ by employing Alg. 1. For the determination of the objects’ convexity type the increasing-ordered-by-value depth information across the pixels defining each object is considered (Fig. 5(right)). Alg. 1 is based on a convexity depth threshold ($\text{thresh}_{\text{convex}}$), which defines the upper boundary of depth range information of a ‘convex’ type object. Objects with depth range (considering the minimum and maximum values of the $\text{dist}_{\text{depth}}$) greater than $\text{thresh}_{\text{convex}}$ are subject to be grouped under ‘concave’ or ‘surface’ depending on their depth contour hierarchies (*ContourHierarchy*). A depth contour is the contour created by the depth information exceeding $\text{thresh}_{\text{convex}}$ (Fig. 6). Contour hierarchies establish a tree structure of contour inclusion, where every node of the tree stands for a contour and every parent includes its children. In this case, the root of the tree is always the frame of the image. Contours which are not directly included in the root contour are not considered its children. We eliminate potential contour noise by considering the relative difference of the size of the contours with respect to the detected object. The remaining contours are then examined in terms of hierarchy. A *child contour* indicates an *indentation area*. Thus, the detection of a child contour in the depth domain, indicates the presence of a concave curve, therefore a ‘concave’ type object, and ‘surface’ type otherwise⁴. Figure 7 summarizes the process of the detection of a concave object.

However, defining only the type of an object is not sufficient for reasoning about the spatial interactions between objects. Detecting complex spatial object relations effectively, *i.e.* contains, is challenging without considering potential concave curves of objects. Thus, we propose Alg. 2 to infer the boundaries of the m- and M+ areas, as a direct generalization of the *Process-Grammar* to 2.5D, indicating a concave curve in 3D space, with respect to the

4. This approach considers this restricted representation for detecting concave objects as the depth resolution varies depending on the objects’ distance from the visual sensor.

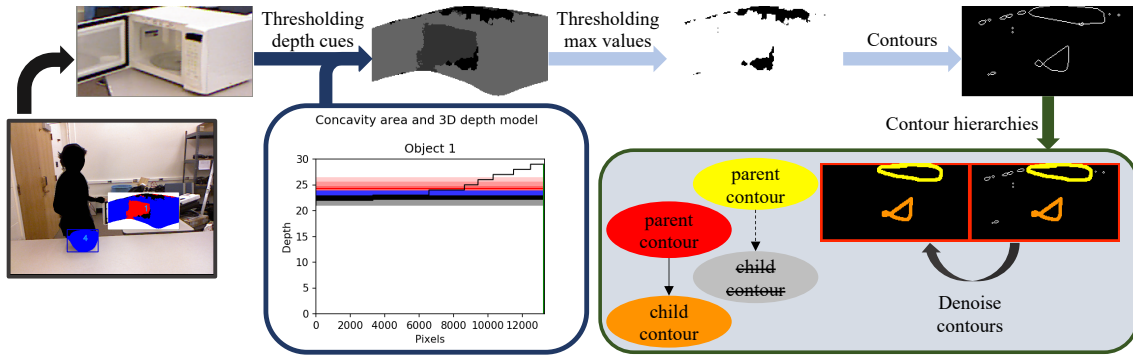


Figure 7: Process steps for detecting ‘concave’ type objects. From an RGB video frame objects are detected using an object detector; numbers are shown in the image frame for every detected object. The bounding boxes proposing an object detection are mapped on the depth domain to extract the depth data relevant to that object detection. The depth data is then partitioned and deep depth regions are exploited to produce depth contours of possible concave curve areas. Finally, depth contours are a hierarchical set depending on each contour’s inclusion, *e.g.* the orange contour is included in the red contour hence it is a child of the red contour. If a child contour exists then the region it encapsulates is detected as a concave curve area.

object’s depth information and convexity type. For a ‘concave’ type object we partition the ordered-by-value depth information into h sections for distinguishing the *indentation* from the *protrusion area*. The n sections with the highest depth values are estimated to capture its concave curve. We set these depth boundaries of such objects to enclose the potential concave curve’s depth information for detecting the relation **Cont**. The parameterization of h , n , and $thresh_{convex}$ was conducted in an empirical study and is further explained in §8. Depth boundaries of ‘convex’ and ‘surface’ type objects are not processed due to concave curve absence. d_{cmax} and d_{cmin} set the boundaries of the potential concave curve area of a ‘concave’ type object and are further employed for detecting a *contains* relation between a pair of objects.

6. Learning Object Affordances

We learn to categorize object affordances by clustering *AGraphlets* whilst employing a hierarchical approach. Every *AGraphlet* represents the interaction between a pair of objects in the scene. We consider an interaction between two objects as the spatio-temporal sequence of relations holding during an activity. By clustering such graph structures we produce a hierarchy of similar affordances, which are closely related with the way every object is being used in an activity. Hence, our method does not pose any constraints on the number of affordance clusters an object can be assigned to, whilst every object has as many *AGraphlets* as detected interactions. *E.g.* consider the scenario where an agent picks a bowl from a table and places it in the microwave. The clustering mechanism considers the interaction

of the bowl with the table and the bowl with the microwave as two different interactions; two *AGraphlets* will be created for the bowl implying that it is a *supportable* as well as a *containable* object.

As *AGraphlets* are high-level object interaction representations defined in the space of the qualitative relations being captured, computing the similarity between them is based on the spatio-temporal relations occurring. To effectively cluster these representations we project the *AGraphlets* in a d -dimensional latent space and perform clustering on that space instead. The *graph2vec* network is employed for this purpose, creating d -dimensional graph embeddings for each *AGraphlet*.

To compute the similarity/dissimilarity of these graph embeddings we exploit a cost function using the cosine similarity measure, defined as:

$$C(A, B) = 1 - \cos(A, B) = 1 - \frac{A \cdot B}{\|A\| \times \|B\|} \quad (1)$$

where the values of the $\cos()$ function range from -1 to 1, where 1 indicates that the vector embeddings are exactly the same, 0 that they are orthogonal, and -1 means that the vectors are exactly opposite. In the *AGraphlet* space, a 1 cosine value stipulates that the two graphs carry the same spatio-temporal information, hence the same affordance, whereas a value of -1 suggests that the two graphs have no spatio-temporal information in common, thus each one of them represents a completely different affordance. A cosine value of 0 denotes that the two graphs are opposite, each one carrying information of the two objects that interact.

We perform a hierarchical clustering on the graph embeddings, using the cost function defined in Eq. 1, to produce a dendrogram of similarities of the extracted *AGraphlets*. Clusters are formed by grouping the leaves of the dendrogram with respect to their hierarchy. Figure 11(top) illustrates a subset dendrogram of the complete hierarchical clustering output. Such a hierarchy reveals the similarities of the different interactions occurring in the data, which thus yields a set of affordance clusters. Different set of clusters are formed based on a tunable threshold in our hierarchy. The higher in the hierarchy we consider creating the clusters, the more general the clusters become in terms of affordances.

7. Leeds Object Affordance Dataset (*LOAD*)

We present the Leeds Object Affordance Dataset or *LOAD* dataset, which effectively extends the domain of activities existing in all the publicly available datasets to date. This newly presented dataset comprises of 58 RGB-D videos with an average number of 258 frames per video, capturing various human-object interactions, which are not present in the existing datasets visualizing human-object interactions. *LOAD* dataset aims to push the boundaries of affordance distinction and categorization through more complex real-world activities. Sample frames presenting the different object affordances captured can be found in the appendix. Figure 8 illustrates the percentage of the affordance labels occurring in the *LOAD* dataset and Figure 9 presents the percentage of different kinds of activities. Moreover, Figure 10 displays the percentage of occurrence of the affordances in every kind of activity. The main benefit of the *LOAD* dataset, as shown in these figures, is that the same affordance is present in different kind of activities and between different objects. From Figure 10, though some affordances are present in only a single activity, *i.e.* ‘drinkable’, there are

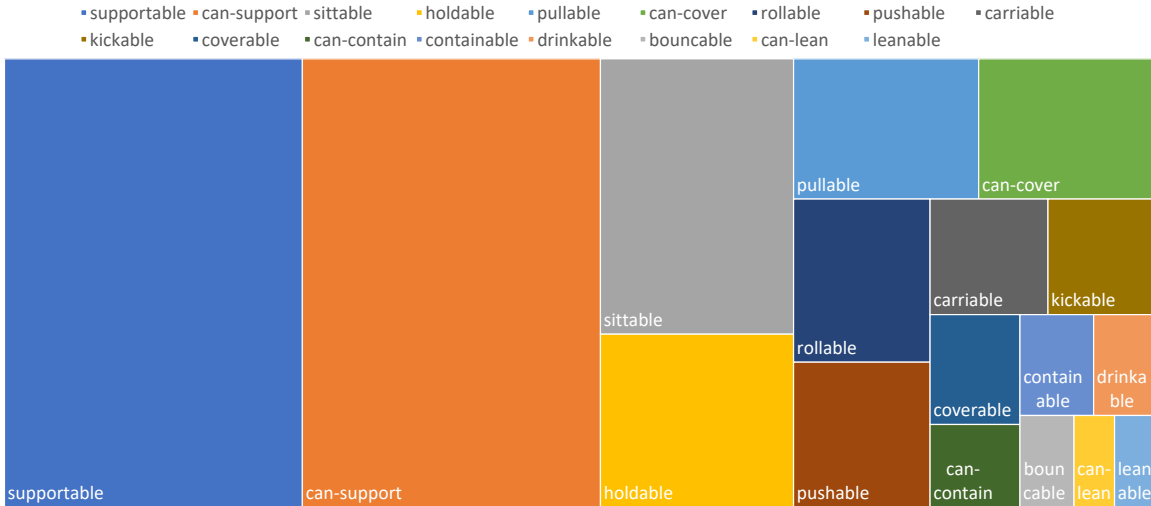


Figure 8: Affordance labels in the LOAD dataset; area of rectangle reflects abundance in the dataset.

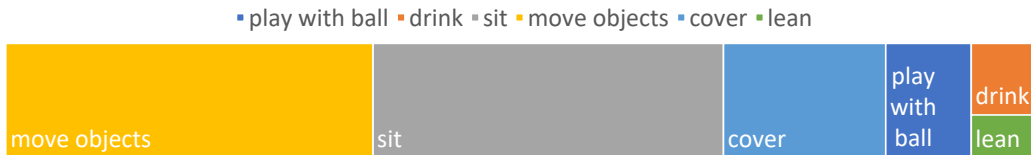


Figure 9: Coarser grouping of activities in LOAD dataset.

others that are spread throughout different kind of activities, *i.e.* ‘support’, ‘can-support’, ‘holdable’, ‘rollable’, ‘coverable’, ‘can-cover’. In this dataset, apart from the interactive objects, we also consider the ‘floor’ as an entity that holds an affordance, thus the affordance labels of ‘can-support’ and ‘supportable’ dominate in the ground truth. Also, there are several affordance labels, which do not have a complementary affordance. This is due to affordances related to human-object interactions, *e.g.* ‘kickable’, ‘pushable’, ‘pullable’, etc.

8. Experimental Setup

8.1 Datasets

For the evaluation of our method we found the publicly available CAD-120 dataset (Koppula et al., 2013) (CAD) suitable for the task of object affordance categorization. Moreover, to prove generalizability across different data distributions we also tested our method on the Watch-n-Patch (Wu et al., 2015) (WnP), and on the LOAD dataset (LOAD) newly presented here. All three datasets comprise activities of everyday-life scenarios with various configurations of the objects in the scene as well as different camera orientations. Table 4 summarizes the characteristics of the datasets employed in this study. Moreover, the affordance labels present in the LOAD dataset are indicated in Figure 8, whereas the labels

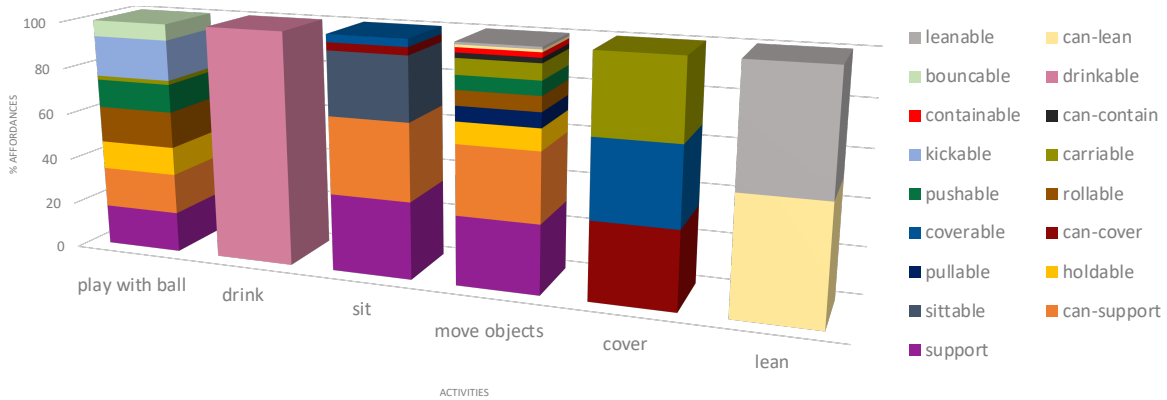


Figure 10: Analysis of the percentage of affordances for every activity, considering a coarser grouping of activities.

for both the CAD-120 and Watch-n-Patch datasets are: ‘can-contain’, ‘containable’, ‘can-support’, ‘supportable’, ‘drinkable’, and ‘holdable’.

We used 80% of the CAD-120 dataset to determine the method’s parameters and train the *graph2vec* network, resulting to 18,072 *AGraphlets*, and then evaluated the proposed approach on the remaining 20% unseen videos (24 videos), which comprise of 5,682 *AGraphlets*. Experiments on the Watch-n-Patch and LOAD datasets were conducted on 24 and 15 hand-picked videos, or 6,900 and 1,212 *AGraphlets* respectively, where the predictions of the object detector, after visual inspection, were sufficient for capturing interacting objects. For defining the parameters of the algorithm, we exploited hand-picked videos from the training set for the CAD-120 dataset, and for the LOAD and Watch-n-Patch datasets we hand-picked videos different from the ones used for testing the proposed method.

8.2 Human/Object Detection and Tracking

To obtain human skeletal data in the scene we compared the output of the Convolutional Pose Machine (CPM) (Wei et al., 2016) model with the kinect skeletal data. Through an empirical study we found that the CPM predictions are more accurate, especially when human-joint occlusion occurs or the human agent stands on their side.

Object locations and depth information are provided from the predicted objects’ masks. We employ the state-of-the-art Mask R-CNN framework (He et al., 2017) trained on the COCO dataset (Lin et al., 2014), since the training data distribution is similar to the data distribution present in the datasets we use in this work, and it is a generic dataset with more object classes than those included in the target task. The box enclosing the object’s mask corresponds to the object’s bounding box. This implementation is based on Mask R-CNN predictions, however the proposed method is not object-specific and any class-agnostic proposal method can be used.

Whilst Mask R-CNN objects’ bounding boxes predictions are temporally sparse, we enrich the objects tracks by considering the CSRT tracker’s predictions (Bolme et al., 2010) from the latest object bounding box occurrence, for frames where Mask R-CNN failed to

Dataset	data	#vid.	avg.fr. ^a	a.p.v. ^b	a.r. ^c	activities
CAD-120	2.5D	120	525	1-2	3	‘arranging objects’, ‘cleaning objects’, ‘having meal’, ‘making cereal’, ‘microwaving food’, ‘picking objects’, ‘stacking objects’, ‘taking food’, ‘taking medicine’, ‘unstacking objects’
Watch-n-Patch	2.5D	457	170	1-2	3	‘turn on monitor’, ‘turn off monitor’, ‘walking’, ‘play computer’, ‘reading’, ‘fetch book’, ‘put back book’, ‘take item’, ‘put down item’, ‘leave office’, ‘fetch from fridge’, ‘put back to fridge’, ‘prepare food’, ‘microwaving’, ‘fetch from oven’, ‘pouring’, ‘drinking’, ‘leave kitchen’, ‘move kettle’, ‘fill kettle’, ‘plug in kettle’
LOAD	2.5D	58	258	3-4	3	‘pull chair’, ‘push chair’, ‘sit on chair’, ‘sit on the floor’, ‘sit on the table’, ‘push table’, ‘pull table’, ‘hold bottle’, ‘hold umbrella’, ‘carry bag’, ‘drinking’, ‘put item down’, ‘kick ball’, ‘roll ball’, ‘bounce ball’, ‘throw ball’, ‘cover’, ‘put into’, ‘take out’, ‘lean against the wall’, ‘roll bottle’, ‘roll chair’, ‘roll suitcase’, ‘carry chair’

^a average number of frames (per video)

^b activities per video

^c activity repetition

Table 4: Datasets

detect the object, hence creating tracks of class agnostic-masks. CSRT tracker solely relies its predictions of the visual characteristics of the initial object mask predictions, considering only past instances on mask detections. Hence, it is able to track non-rectangular objects, which aids in having more accurate tracking predictions by considering the objects mask’s rather than their bounding boxes. A threshold of minimum 0.5 IoU overlap, determined from an empirical study, is required to assign a bounding box prediction as the predicted location of a detected object.

The predicted object mask’s depth information is employed to infer the object’s convexity type. We exclude any pixel, which is part of a human detection, by retrieving a human mask from the DensePose framework (Alp Güler et al., 2018). Whilst Mask R-CNN produces object masks for every object separately, overlapping objects have overlapping masks, thus the intersected mask area may cause problems in determining the object’s convexity type. A *semantic depth map* is constructed for every frame of the video data, consisting

of all the predicted object masks while eliminating any detected intersection mask area. This is achieved by assigning every pixel of such area to the object with the highest mask detection score.

8.3 Parameters

We employed the QSRLib library (Gatsoulis et al., 2016) for the construction of *AGraphlets*. For the h and n values which are used in defining the *indentation area* of a concave object, we selected the values of 5 and 3 respectively, after conducting an empirical study for $h \in \{2, 3, 4, 5, 6, 7\}$ and $n \in \{1, \dots, h - 1\}$. We also set the $thresh_{convex}$ value to be 4 for the CAD-120 dataset after evaluating on the values 1,2,3,4,5,6,7,8,9,10, 0.3 for the Watch-n-Patch after consideration of the values in the range 0.1 to 2.0 with step 0.1, and 10.0 for the LOAD after studying the values in the range 8 to 20 with step 1.

For the creation of clusters we threshold the produced hierarchy at 0.02 for all three datasets. The value of 0.03 was employed for the RCC5(+ON) experiment (§9.1), though for the sED (§9.1) the value 1.0 was used. This hierarchy threshold was defined by evaluating the Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) on the training set for different heights of the hierarchy.

8.4 *graph2vec* Network

We trained from *graph2vec* network with the *AGraphlets* of the training set to create a 128-dimensional latent space where the interaction graphs can be projected on. We employed the network architecture provided by (Narayanan et al., 2017) and we trained it using Stochastic Gradient Descent, a batch size of 512, and setting the learning rate to 0.5. Also, we set the WL subgraph extraction value to 14 which, through an empirical study, we found it was sufficient enough to create all the different subgraphs from the *AGraphlets*.

The selection of the embedding size and learning rate was made from two empirical studies detailed in the appendix.

9. Experimental Evaluation

9.1 Quantitative Evaluation

In this section, we will analyze the experimental results by inspecting the clusters of affordances formed, and evaluating their homogeneity and completeness. The datapoints clustered consist of detected objects with their interactions with any other object and the human agent, allowing multiple datapoints to point to the different interactions a single object might hold. We evaluate the clusters by reporting the normalized *V-measure*, *homogeneity*, and *completeness* scores, with higher values implying a better clustering. Homogeneity captures how homogeneous the clusters are with respect to the groundtruth annotations, whereas completeness expresses how complete the clusters are in terms of the groundtruth affordance labels, *i.e.* encapsulating as many as possible of the same groundtruth labels into a single cluster. The V-measure represents an average of the homogeneity and completeness scores.

	Method	V-measure	Homogeneity	Completeness
CAD	RCC5(+ON)	0.80	0.93	0.70
	sED	0.57	0.67	0.49
	Ours	0.87	0.99	0.77
WnP	RCC5(+ON)	0.49	0.83	0.34
	sED	0.14	0.13	0.16
	Ours	0.57	0.99	0.40
LOAD	RCC5(+ON)	0.62	0.78	0.52
	sED	0.40	0.44	0.37
	Ours	0.70	0.99	0.54

Table 5: Baselines experiments.

9.1.1 BASELINES EXPERIMENTS

We compare our method with the baselines presented in Table 5, in which we evaluate the impact of using depth information to enhance 2D qualitative spatio-temporal relations, and projecting *AGraphlets* into a latent space. The baselines are:

- RCC5(+ON): we evaluated our method with the RCC5 (Randell et al., 1992), which consists of the relations: ‘discrete’ (DR), ‘partially overlapping’ (PO), ‘proper part’ (PP, PPI), and ‘equal’ (EQ), along with the addition of the On spatial relation defined in §5.1. We aim to investigate how the lack of depth information affects the affordance categorization. For this study we keep the method’s configuration unchanged and conduct the experiments altering only the exploited spatial relations;
- sED: we performed hierarchical clustering on the *AGraphlet* without projecting them in a latent space learned by the *graph2vec* network. This baseline evaluates the clustering performance by considering a vanilla measure for graph comparison and aims to showcase the contribution of projecting *AGraphlets* in a latent space. Hence we exploit a *set Edit Distance* measure (Eq. 2, where c_{spat} and k_{spat} are 0.5 from an ablation study) to create clusters of similar affordances;

$$\begin{aligned}
 sED = c_{spat} \sum_{v \in V'_{DiSR}} v + c_{temp} \sum_{v \in V'_{DiSR_{temp}}} v + \\
 k_{spat} \sum_{v \in V'_{RCC2}} v + k_{temp} \sum_{v \in V'_{RCC2_{temp}}} v
 \end{aligned} \tag{2}$$

where $c_{temp} = 1 - c_{spat}$ and $k_{temp} = 1 - k_{spat}$

$$\begin{aligned}
 V'_R{}^{\alpha,\beta} = \{v : v \in \{V'_R{}^\alpha \setminus V'_R{}^\beta\} \cup \{V'_R{}^\beta \setminus V'_R{}^\alpha\}\} \\
 \text{where } R \in \{DiSR, DiSR_{temp}, RCC2, RCC2_{temp}\}
 \end{aligned} \tag{3}$$

	Method	V-measure	Homogeneity	Completeness
CAD	Sridhar et al.	0.40	0.66	0.29
	Aksoy et al.	0.71	0.84	0.61
	Ours	0.87	0.99	0.77
WnP	Sridhar et al.	0.39	0.65	0.28
	Aksoy et al.	0.58	0.70	0.50
	Ours	0.57	0.99	0.40
LOAD	Sridhar et al.	0.63	0.82	0.51
	Aksoy et al.	0.61	0.77	0.50
	Ours	0.70	0.99	0.54

Table 6: Experimental comparison with related works.

From this study, it is evident that our method outperforms the RCC5(+ON) and sED baselines in all reported metrics and datasets. The comparison between the experimental results of the proposed approach and the RCC5(+ON) baseline, demonstrate that the employment of depth information for inferring object affordances has a considerable improvement of the V-measure in comparison to the primary RCC5 set. This improvement results from more accurate spatial relationships of interactions, enabling the creation of more accurate graphical structures for describing them. Also, the scores’ improvement involving the employment of the *graph2vec* network, in comparison to the sED baseline, indicates the enhancement of the clustered datapoint representation by projecting the *AGraphlets* in the latent space.

9.1.2 RELATED-WORKS COMPARISON

Related works exploit the distance between objects as a measure for perceiving their interactions, or simplistic qualitative relations, which only describe the relation of the objects with the human hand, *e.g.* ‘hand approaching’, ‘hand leaving’, ‘in use’, ‘idle, and ‘close to’. Both approaches are not able to capture complex object interactions. For this reason, our evaluation considers methods, which leverage qualitative spatial relations for describing human-object and object-object interactions, which are more descriptive in the spatial domain.

We compare our approach with two state-of-the-art methods that use qualitative spatio-temporal relations to describe in a high-level way human-object and object-object interactions, similar to the proposed method. These works are:

- Sridhar et al.: we conducted experiments using a re-implementation of the approach of Sridhar et al. (2008) which, to the best of our knowledge, is the most closely related work that exploits QSRs to capture functional object clusters;
- Aksoy et al.: we also performed a comparison of the proposed method with a re-implementation of the approach of Aksoy et al. (2010), which exploits high-level

qualitative graphs without the employment of QSRs, to describe interactions between objects.

Both of these works employ a set of spatial qualitative relations. However, these relations are not descriptive enough to distinguish between complex object interactions, *i.e.* containment and support. Hence, differentiation of affordances due to different object types is not present, resulting to a coarser-grain of affordance categorization. Moreover, occlusion is not handled, thus capturing many false positive object interactions.

Table 6 shows the quantitative results of the experiments we conducted to demonstrate the performance of the proposed approach against the Sridhar et al. (2008) and Aksoy et al. (2010) baselines. Our experiments demonstrate a significant increase in all the reported metrics against the Sridhar et al. (2008) work, which results from the representation of *AGraphlets* that exploit more information about the interactive objects. We also achieve higher scores than the Aksoy et al. (2010) work in the CAD-120 and LOAD datasets, and obtain comparable V-measure score on the Watch-n-Patch dataset. This is due to the fact that not many interactions are evident in the video scenes of the Watch-n-Patch, and the set of relations being captured are more coarse-grained, resulting to a higher completeness score, however degrading the homogeneity of the clusters.

Our results indicate that our method creates more fine-grained groups of affordances than the groundtruth annotations, *i.e.* there can be more than one group of datapoints indicating a ‘supportable’ affordance. Hence, we can distinguish between different objects with the same convexity type.

9.2 Qualitative Evaluation

A representative instance of our object affordance categorization method is illustrated in Figure 11. A subset of *AGraphlets* from the test set of the CAD-120 dataset is employed in this figure as an input to the proposed methodology. Groups of similar *AGraphlet* embeddings in the latent space form clusters of object affordances. Clusters are color-coded in the presented hierarchy.

The depicted clusters denote the different kind of affordances in the present subset of a video. To facilitate better understanding of the results, by considering the dominant ground truth affordance label in each group of data points, we are able to give an affordance name to every cluster (Fig. 11 (bottom right)). Some clusters have been assigned the ‘unknown label’, which denotes an affordance that is not present in the groundtruth hence, creating a fine-grained affordance hierarchy. Also, some clusters have two labels of affordances, which indicate that the objects in those clusters carry both these affordance labels. Our clustering hierarchy of affordances takes no account of any object labels, thus a cluster is able to contain all objects that have similar interactions in reference to their *AGraphlets* embeddings.

Figure 12 shows the clustering output of a video from the CAD-120 dataset. Groups of objects with similar affordances form the clusters of the hierarchy. Groundtruth affordance labels are assigned to every cluster by inspecting the objects and their *AGraphlets* the cluster comprise of.

An important aspect of our method, as illustrated in this figure, is that we achieve a fine-grained affordance categorization by differentiating between different kinds of the same

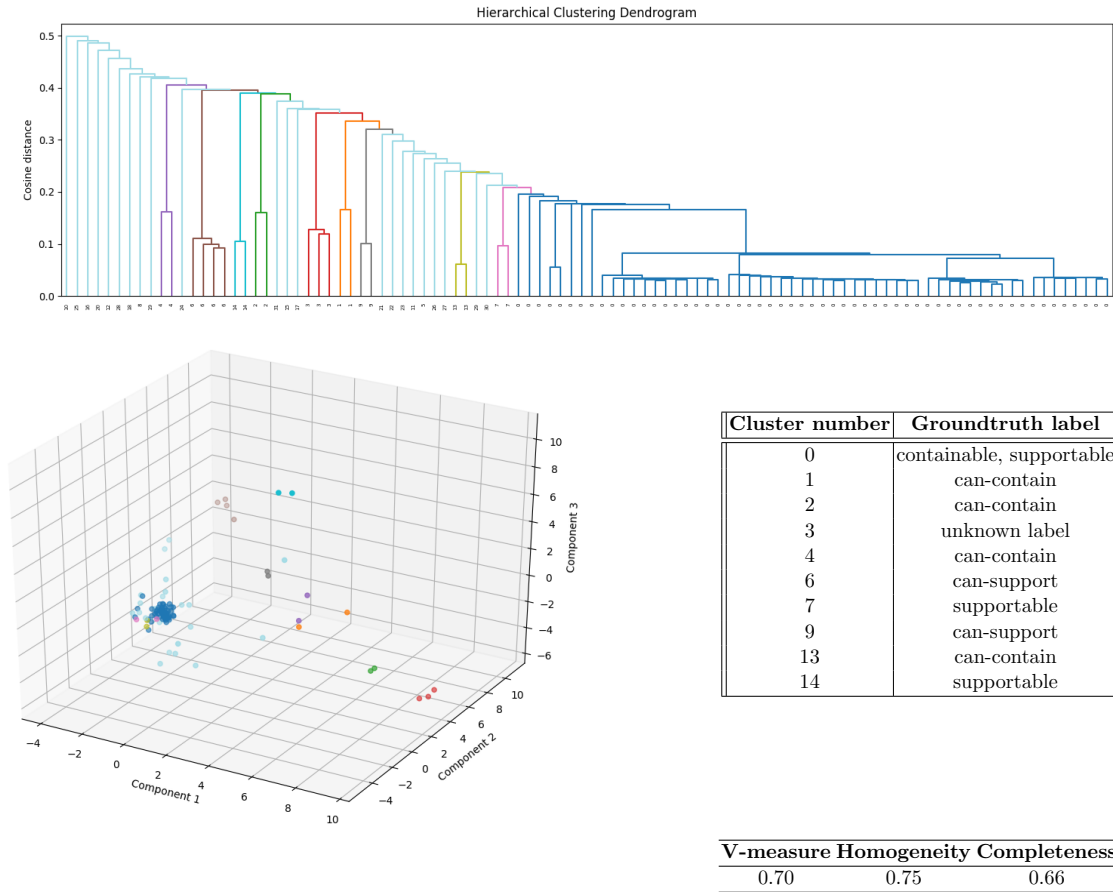


Figure 11: Hierarchical clustering of 100 *AGraphlets* captured from the CAD-120 dataset. (top) The hierarchy of clusters where the y -axis corresponds to the distance and the x -axis shows the cluster id every leaf node is assigned to; edges of the leaf nodes are colored depending on the cluster the leaf node is assigned to. (bottom left) Visualization in 3D space of the graph embeddings of the clustered data, after applying PCA. Coloring is associated with the clusters they are assigned to. (bottom right) Table mapping the clusters id with groundtruth labels.

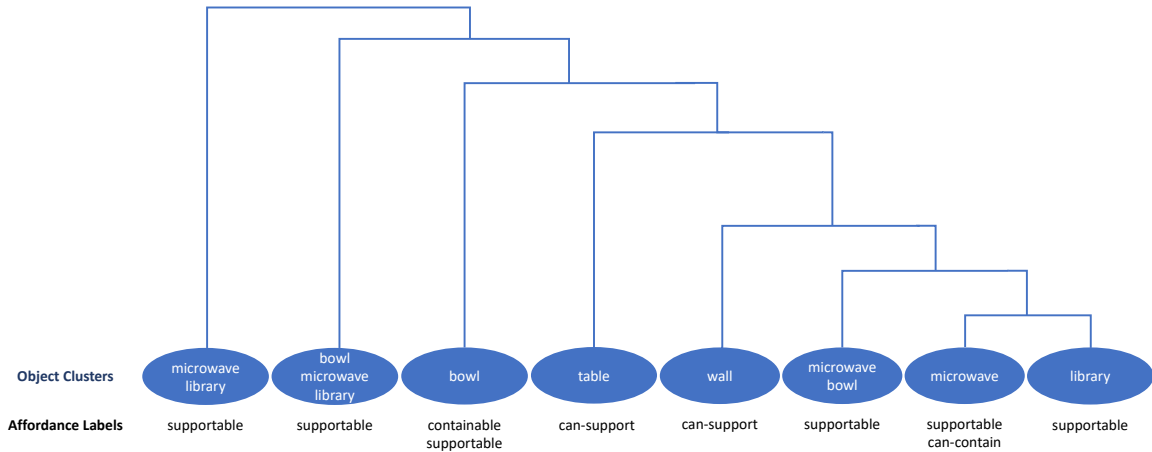


Figure 12: Clusters' hierarchy for a video from the CAD-120 dataset.

convexity type objects, *i.e.* the bowl and the microwave in this example. As both the bowl and the microwave are containers, the bowl plays the role of a containee. Whilst, we can also group together different convexity type objects if they carry the same interaction.

The hierarchical tree presented, is generalizable to the whole dataset. More hierarchical trees of other videos demonstrating the generalization of our approach across different datasets, thus videos, are illustrated in the appendix.

10. Discussion

10.1 Failure Cases

Object detections are retrieved from an off-the-shelf object detector, trained on the COCO dataset. However, some object configurations, present in the datasets used for the evaluation of this work, are out of the object detector's training distribution. Hence, false positive object detections are evident in cases where the primary objects change shape and visual appearance, *i.e.* opening a closed microwave. In such cases, false positive relations between detected objects are captured. *E.g.* opening the door of a closed microwave produces a detection for the main body of the microwave object as well as a different one for its door, resulting to the presence of *AGraphlets* between the door and the rest of the objects in the scene. Hence, the proposed approach is heavily reliant to the quality of object proposals.

10.2 Limitations

Our definitions of spatial interactions model spatial states of the interactive objects, *e.g.* a cup is on the table. However, some kind of affordances are derived from interactions holding as a transition from one state to another, *e.g.* 'pourable', 'able to be poured' are inferred from the transition of the state $\text{Cont}(\text{bowl1}, \text{liquid})$ to $\text{Cont}(\text{bowl2}, \text{liquid})$. Our method is limited to only detect state-based relations; hence affordances as 'pourable', 'able to be poured', 'throwable', and 'able to be thrown' are not detectable in the current pipeline. Although the proposed framework is generic, it is currently limited to the set of detectable

and defined relations. Moreover, an enhancement of detecting interactions of more than two objects is necessary for affordances which are inferred from the interaction of multiple objects, *e.g.* learning ‘stirable’ requires both a liquid and a stirrer to be contained in a concave object.

11. Conclusion

In this work, we presented a novel depth-informed qualitative representation with the ability to handle occlusions by considering depth information, and efficiently detecting relative spatial relationships between objects in the 2D image plane. We also addressed the problem of affordance categorization by creating embeddings of graphs of human-object and object-object interactions using the *graph2vec* network, and clustering these graphs in an unsupervised way, forming groups of similar affordances in the latent space. Moreover, we introduced a new object affordance dataset which expands the affordance domain compared to existing datasets. Our experiments demonstrate that exploiting the proposed relations produces more accurate qualitative graphs, describing the interactions, resulting in more homogeneous and complete clusters of affordances, and higher V-measure scores.

The enhancement of the graph representations with additional object information, such as the size of the objects, is a future direction for expanding the possible affordances our method can detect. Also, in our future work, we aim to enhance the qualitative spatial relations, by detecting multi-object relations and enabling the construction of sequence-based relations, which allow the description of interactions that occur during a series of episodes, *e.g.* ‘throwing’.

Acknowledgements

The authors would like to thank Panagiotis Magkafas for his valuable assistance in re-implementing the Aksoy et al. baseline work, used for comparison with our approach. This work has been partially supported by the European Union’s Horizon 2020 research and the innovation program under grant agreement No. 825619 (AI4EU), and by an Alan Turing Institute Fellowship to the second author.

Appendix A.

A.1 LOAD Dataset

ball: kickable, supportable



ball: holdable, bouncable



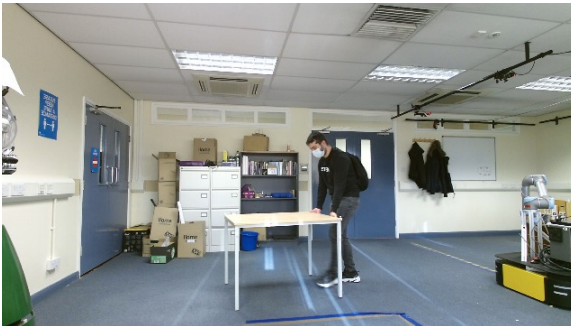
suitcase: holdable, carriable



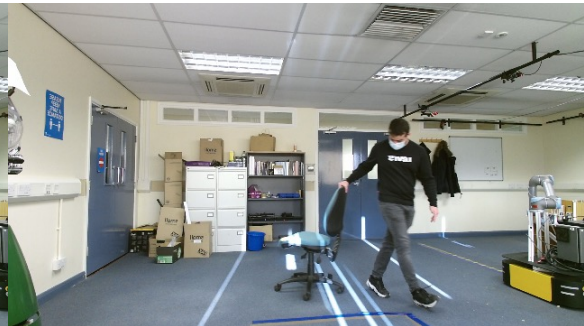
chair: holdable, carriable



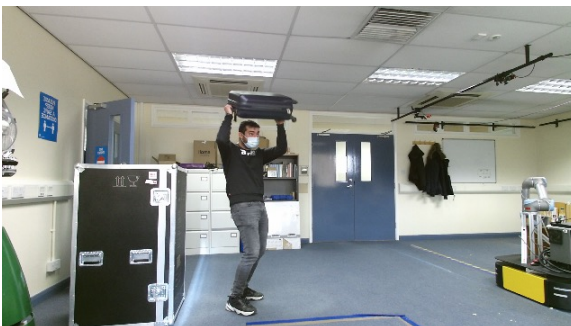
table: supportable, pushable



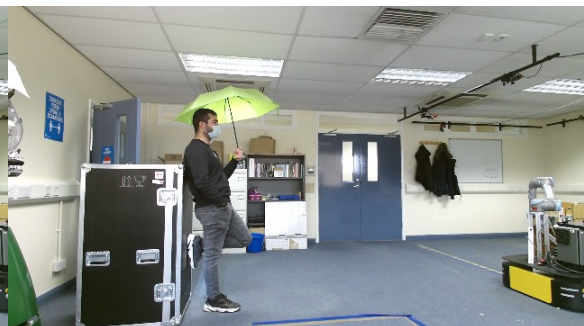
chair: supportable, pullable



suitcase: holdable, carriable, can-cover



umbrella: holdable, can-cover



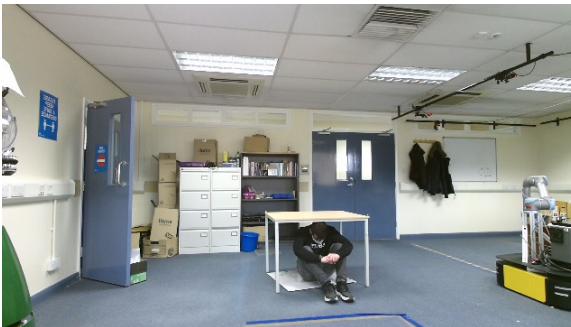
box: supportable, can-support, coverable, sittable



chair: supportable, sittable, coverable



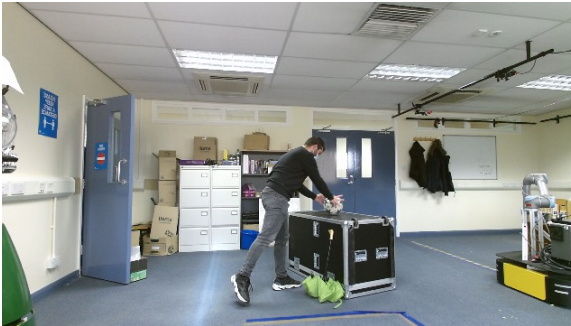
table: supportable, can-cover



bottle: holdable, drinkable



box: supportable, can-support, leanable



suitcase: supportable, can-contain



bottle: supportable, containable, holdable



suitcase: supportable, can-support, coverable, rollable



Some sample image frames from the LOAD dataset are presented here, along with the affordance labels for one of the detected objects in every scene.

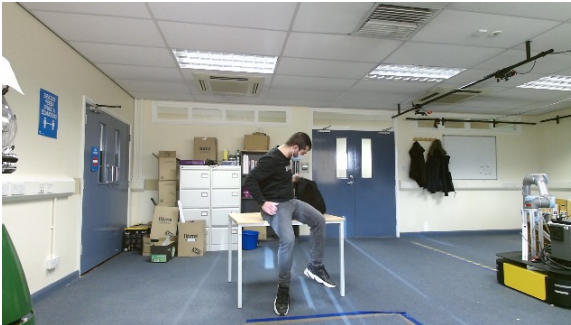
table: supportable, sittable, coverable



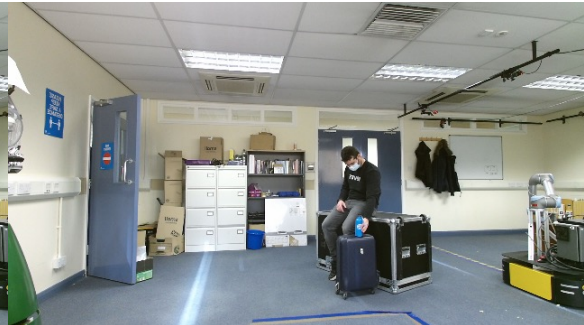
chair: supportable, can-support



table: supportable, sittable, can-support



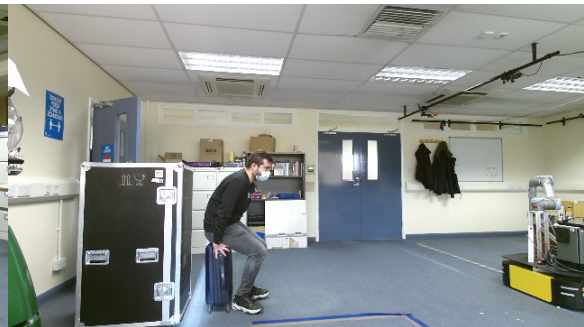
suitcase: supportable, can-support



chair: supportable, sittable



suitcase: supportable, sittable



chair: supportable, can-support



box: supportable, sittable



A.2 *graph2vec* Hyper-parameter Tuning

For the selection of the embedding size we trained the *graph2vec* network with the embedding size values 128, 256, 512, and concluded that an embedding of size 128 produces the smallest

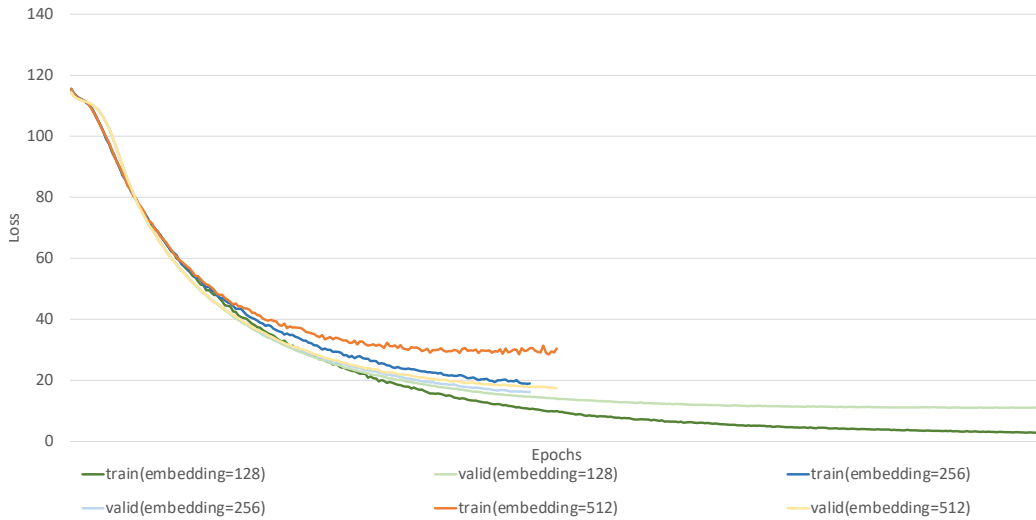


Figure 13: (best viewed in color) Training and validation loss of *graph2vec* network for different embedding sizes. The experiments were done with batch size 512 and learning rate 0.5. The loss history does not update if a validation loss value greater than the latest reported has been calculated.

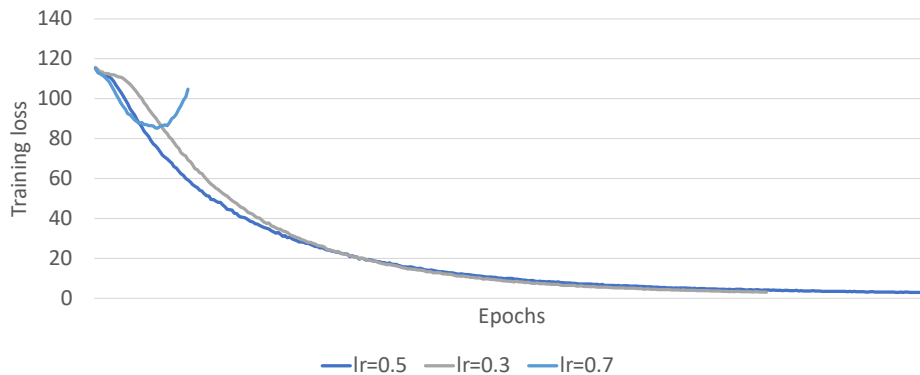


Figure 14: (best viewed in color) Training loss of *graph2vec* network with embedding size 128 for different learning rates. The experiments were done with batch size 512. The training loss history does not update if a validation loss value greater than the latest reported has been calculated.

training and validation loss (Fig. 13). Batch size and learning rate remained unchanged in this experiment. Moreover, we evaluated the output after training the network with different learning rates (0.3, 0.5, 0.7). As illustrated in Figure 14, the best training results were produced with learning rate 0.5. The batch and embedding sizes remained the same throughout this study. The batch size was selected through an empirical study.

A.3 Qualitative Results

Further qualitative results are presented in this section (Fig. 15, Fig. 17). We also, present the cluster hierarchy formed, illustrating groups of object affordances (Fig. 16, Fig. 18). Every cluster forms a group of objects that share the same affordance. Some clusters comprise a single object, however with multiple affordances.

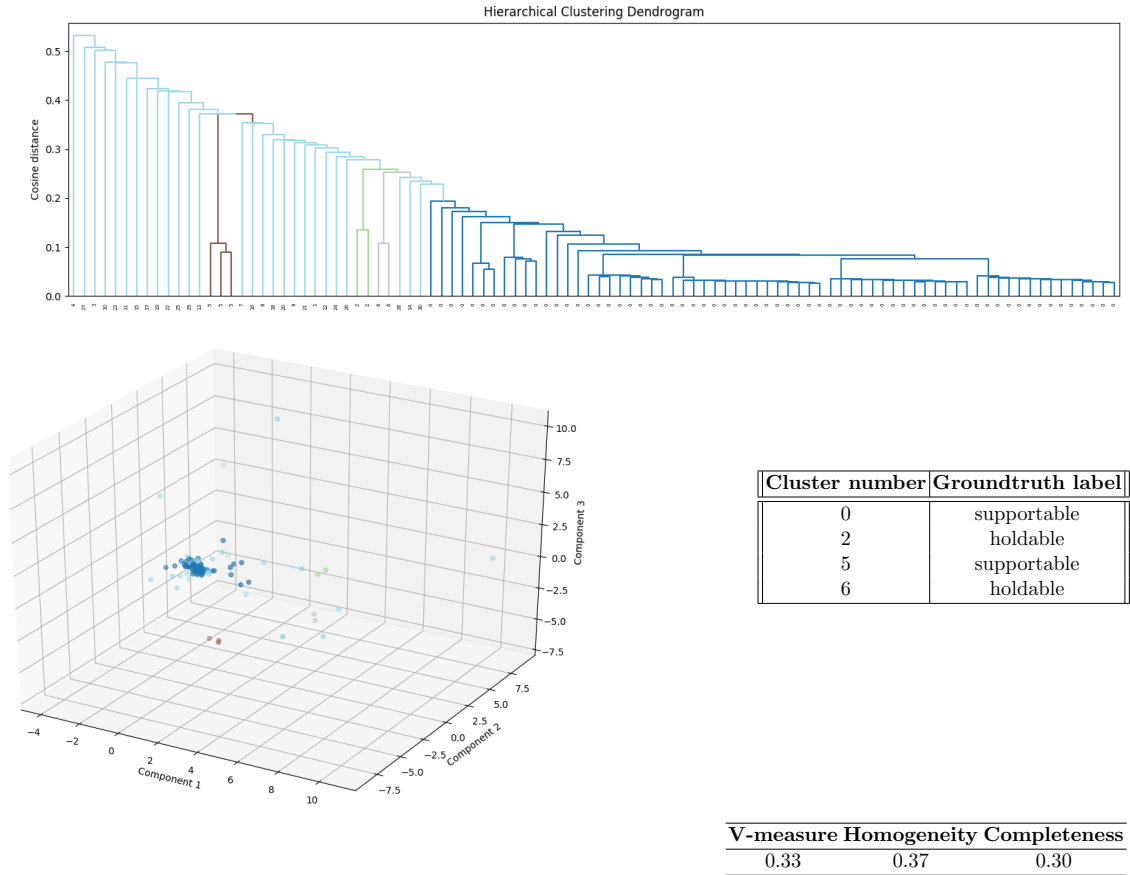


Figure 15: Qualitative results from the Watch-n-Patch dataset.

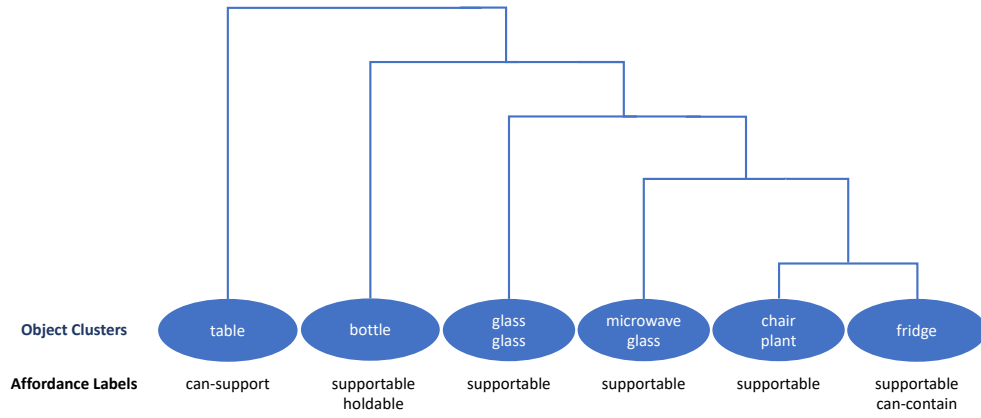


Figure 16: Clusters hierarchy for a video from the Watch-n-Patch dataset.

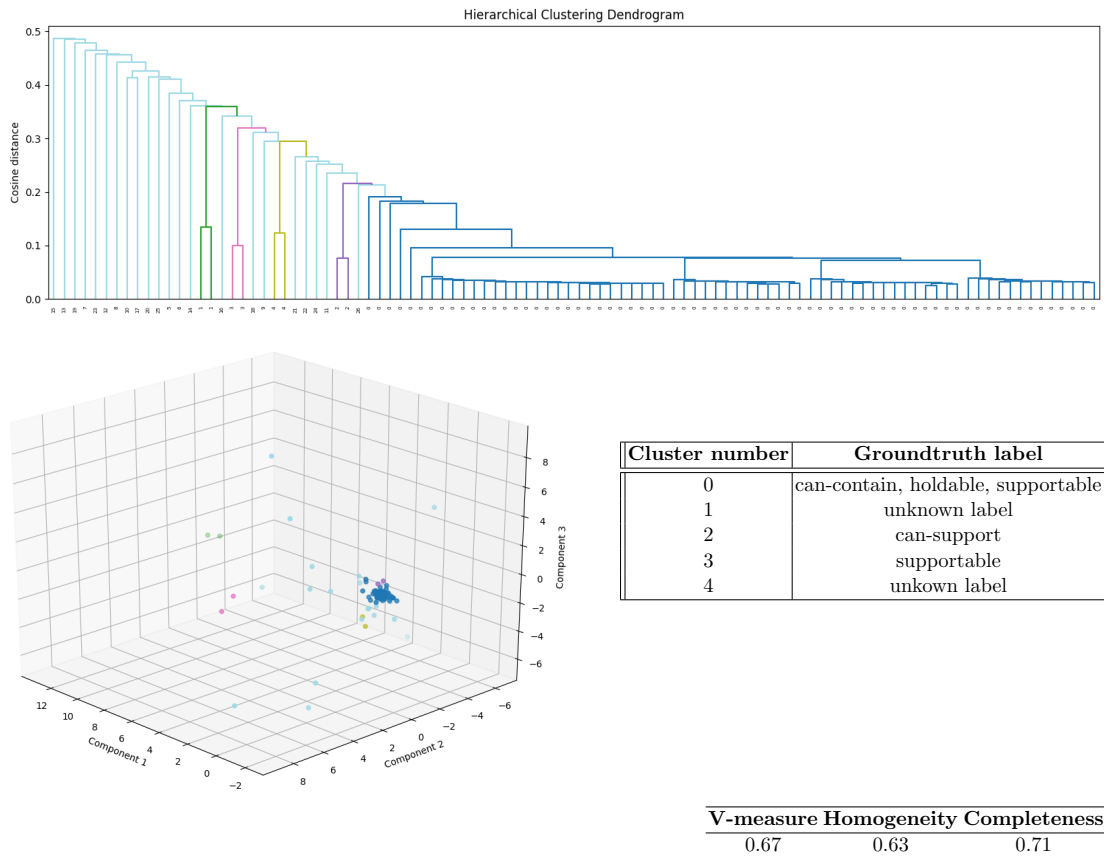


Figure 17: Qualitative results from the LOAD dataset.

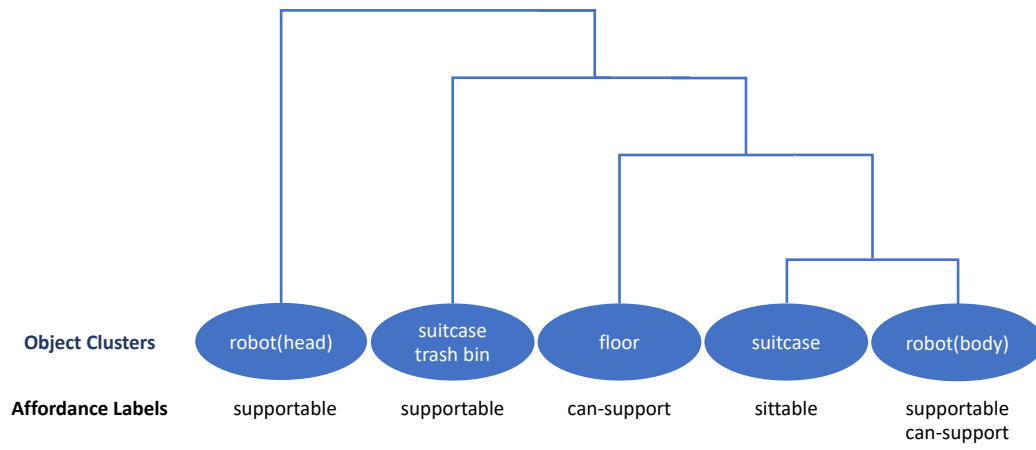


Figure 18: Clusters hierarchy for a video from the LOAD dataset.

References

- Aksoy, E. E., Abramov, A., Dörr, J., Ning, K., Dellen, B., & Wörgötter, F. (2011). Learning the semantics of object–action relations by observation. *The International Journal of Robotics Research*, 30(10), 1229–1249.
- Aksoy, E. E., Abramov, A., Wörgötter, F., & Dellen, B. (2010). Categorizing Object-Action Relations from Semantic Scene Graphs. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 398–405. IEEE.
- Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11), 832–843.
- Alp Güler, R., Neverova, N., & Kokkinos, I. (2018). Densepose: Dense Human Pose Estimation in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7297–7306.
- Bolme, D. S., Beveridge, J. R., Draper, B. A., & Lui, Y. M. (2010). Visual Object Tracking using Adaptive Correlation Filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 2544–2550. IEEE.
- Chen, J., Cohn, A. G., Liu, D., Wang, S., Ouyang, J., & Yu, Q. (2015). A Survey of Qualitative Spatial Representations. *The Knowledge Engineering Review*, 30(1), 106–136.
- Chuang, C.-Y., Li, J., Torralba, A., & Fidler, S. (2018). Learning to Act Properly: Predicting and Explaining Affordances from Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 975–983.
- Cohn, A. G. (1995). A hierarchical representation of qualitative shape based on connection and convexity. In *International Conference on Spatial Information Theory*, pp. 311–326. Springer.
- Cohn, A. G., Bennett, B., Gooday, J., & Gotts, N. M. (1997). Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. *GeoInformatica*, 1(3), 275–316.
- Deng, S., Xu, X., Wu, C., Chen, K., & Jia, K. (2021). 3D AffordanceNet: A Benchmark for Visual Object Affordance Understanding. In *arXiv preprint arXiv:2103.16397*.
- Do, T.-T., Nguyen, A., & Reid, I. (2018). AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1–5. IEEE.
- Dylla, F., Lee, J. H., Mossakowski, T., Schneider, T., Delden, A. V., Ven, J. V. D., & Wolter, D. (2017). A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties. *ACM Computing Surveys (CSUR)*, 50(1), 1–39.
- Egenhofer, M. J. (2005). Spherical Topological Relations. In *Journal on Data Semantics III*, pp. 25–49. Springer.
- Egenhofer, M. J., Clementini, E., & Di Felice, P. (1994). Topological relations between regions with holes. *International Journal of Geographical Information Science*, 8(2), 129–142.

- Egenhofer, M. J., & Franzosa, R. D. (1991). Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2), 161–174.
- Egenhofer, M. J., & Franzosa, R. D. (1995). On the equivalence of topological relations. *International journal of geographical information systems*, 9(2), 133–152.
- Egenhofer, M. J., & Herring, J. (1990). Categorizing binary topological relations between regions, lines, and points in geographic databases. *The*, 9(94-1), 76.
- Egenhofer, M. J., Mark, D. M., & Herring, J. (1994). The 9-Intersection: Formalism and Its Use for Natural-Language Spatial Predicates. In *Report*, Vol. 9, p. 76.
- Egenhofer, M. J., & Sharma, J. (1993). Topological relations between regions in ρ^2 and \mathbb{Z}^2 . In *International Symposium on Spatial Databases*, pp. 316–336. Springer.
- Egenhofer, M. J., & Vasardani, M. (2007). Spatial reasoning with a hole. In *International Conference on Spatial Information Theory*, pp. 303–320. Springer.
- Fang, K., Wu, T.-L., Yang, D., Savarese, S., & Lim, J. J. (2018). Demo2Vec: Reasoning Object Affordances from Online Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2139–2147.
- Galton, A. (1994). Lines of Sight. In *AISB Workshop on Spatial and Spatio-Temporal Reasoning*, Vol. 35, pp. 37–39.
- Galton, A., & Meathrel, R. C. (1999). Qualitative outline theory. In *IJCAI*, pp. 1061–1066.
- Gatsoulis, Y., Alomari, M., Burbridge, C., Dondrup, C., Duckworth, P., Lightbody, P., Hanheide, M., Hawes, N., Hogg, D., & Cohn, A. (2016). QSRlib: A Software Library for Online Acquisition of Qualitative Spatial Relations from Video. In *In Workshop on Qualitative Reasoning (QR16), at IJCAI*.
- Gerevini, A., & Renz, J. (2002). Combining Topological and Size Information for Spatial Reasoning. *Artificial Intelligence*, 137(1-2), 1–42.
- Gibson, J. J. (1977). The Theory of Affordances. *Hilldale, USA*, 1(2).
- Gkioxari, G., Girshick, R., Dollár, P., & He, K. (2018). Detecting and Recognizing Human-Object Interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8359–8367.
- Gottfried, B. (2003a). Tripartite line tracks–bipartite line tracks. In *Annual Conference on Artificial Intelligence*, pp. 535–549. Springer.
- Gottfried, B. (2003b). Tripartite line tracks qualitative curvature information. In *International Conference on Spatial Information Theory*, pp. 101–117. Springer.
- Gottfried, B. (2004). Reasoning About Intervals in Two Dimensions. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, Vol. 6, pp. 5324–5332. IEEE.
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2980–2988. IEEE.
- Hou, Z., Yu, B., Qiao, Y., Peng, X., & Tao, D. (2021). Affordance Transfer Learning for Human-Object Interaction Detection. In *arXiv preprint arXiv:2104.02867*.

- Kjellström, H., Romero, J., & Kragić, D. (2011). Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115(1), 81–90.
- Köhler, C. (2002). The Occlusion Calculus. In *Cognitive vision workshop*, pp. 420–450. Citeseer.
- Kokic, M., Stork, J. A., Haustein, J. A., & Kragic, D. (2017). Affordance Detection for Task-Specific Grasping using Deep Learning. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 91–98. IEEE.
- Koppula, H. S., Gupta, R., & Saxena, A. (2013). Learning Human Activities and Object Affordances from RGB-D videos. *The International Journal of Robotics Research*, 32(8), 951–970.
- Landsiedel, C., Rieser, V., Walter, M., & Wollherr, D. (2017). A review of spatial reasoning and interaction for real-world robotics. *Advanced Robotics*, 31(5), 222–242.
- Leyton, M. (1988). A Process-Grammar for Shape. *Artificial Intelligence*, 34(2), 213–247.
- Liang, W., Zhao, Y., Zhu, Y., & Zhu, S.-C. (2016). What Is Where: Inferring Containment Relations from Videos. In *IJCAI*, pp. 3418–3424.
- Liang, W., Zhu, Y., & Zhu, S.-C. (2018). Tracking Occluded Objects and Recovering Incomplete Trajectories by Reasoning about Containment Relations and Human Actions. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In *European conference on computer vision*, pp. 740–755. Springer.
- Moldovan, B., & De Raedt, L. (2014). Occluded Object Search by Relational Affordances. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 169–174. IEEE.
- Montesano, L., & Lopes, M. (2009). Learning Grasping Affordances from Local Visual Descriptors. In *2009 IEEE 8th international conference on development and learning*, pp. 1–6. IEEE.
- Myers, A., Teo, C. L., Fermüller, C., & Aloimonos, Y. (2015). Affordance Detection of Tool Parts from Geometric Features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1374–1381. IEEE.
- Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., & Jaiswal, S. (2017). graph2vec: Learning Distributed Representations of Graphs. In *arXiv preprint arXiv:1707.05005*.
- Nguyen, A., Kanoulas, D., Caldwell, D. G., & Tsagarakis, N. G. (2016). Detecting Object Affordances with Convolutional Neural Networks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2765–2770. IEEE.
- Nguyen, A., Kanoulas, D., Caldwell, D. G., & Tsagarakis, N. G. (2017). Object-Based Affordances Detection with Convolutional Neural Networks and Dense Conditional Random Fields. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5908–5915. IEEE.

- Pieropan, A., Ek, C. H., & Kjellström, H. (2013). Functional Object Descriptors for Human Activity Modeling. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1282–1289. IEEE.
- Pieropan, A., Ek, C. H., & Kjellström, H. (2014). Recognizing Object Affordances in Terms of Spatio-Temporal Object-Object Relationships. In *International Conference on Humanoid Robots, November 18-20th 2014, Madrid, Spain*, pp. 52–58. IEEE conference proceedings.
- Qi, S., Huang, S., Wei, P., & Zhu, S.-C. (2017). Predicting Human Activities Using Stochastic Grammar. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1164–1172.
- Qi, S., Zhu, Y., Huang, S., Jiang, C., & Zhu, S.-C. (2018). Human-Centric Indoor Scene Synthesis Using Stochastic Grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5899–5908.
- Randell, D., Witkowski, M., & Shanahan, M. (2001). From Images to Bodies: Modelling and Exploiting Spatial Occlusion and Motion Parallax. In *IJCAI*, pp. 57–66.
- Randell, D. A., Cui, Z., & Cohn, A. G. (1992). A Spatial Logic based on Regions and Connection. *KR*, 92, 165–176.
- Sawatzky, J., Srikantha, A., & Gall, J. (2017). Weakly Supervised Affordance Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2795–2804.
- Sridhar, M., Cohn, A. G., & Hogg, D. C. (2008). Learning Functional Object Categories from a Relational Spatio-Temporal Representation. In *ECAI 2008: 18th European Conference on Artificial Intelligence (Frontiers in Artificial Intelligence and Applications)*, pp. 606–610. IOS Press.
- Sridhar, M., Cohn, A. G., & Hogg, D. C. (2010). Relational Graph Mining for Learning Events from Video. In *Proceedings of the 2010 conference on STAIRS 2010: Proceedings of the Fifth Starting AI Researchers Symposium*, pp. 315–327.
- Tan, H., 0003, L. W., Zhang, Q., Gao, Z., Zheng, N., & Hua, G. (2019). Object Affordances Graph Network for Action Recognition. In *BMVC*, p. 145.
- Toumpa, A., & Cohn, A. (2019). Relational Graph Representation Learning for Predicting Object Affordances. In *NeurIPS Workshop on Graph Representation Learning*.
- Toumpa, A., & Cohn, A. G. (2020). Depth-informed Qualitative Spatial Representations for Object Affordance Prediction. In *33rd International Workshop on Qualitative Reasoning*.
- Turek, M. W., Hoogs, A., & Collins, R. (2010). Unsupervised Learning of Functional Categories in Video Scenes. In *European Conference on Computer Vision*, pp. 664–677. Springer.
- Wang, A. Y., & Tarr, M. J. (2020). Learning Intermediate Features of Object Affordances with a Convolutional Neural Network. In *arXiv preprint arXiv:2002.08975*.

- Wei, S.-E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional Pose Machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724–4732.
- Wu, C., Zhang, J., Savarese, S., & Saxena, A. (2015). Watch-n-Patch: Unsupervised Understanding of Actions and Relations. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 4362–4370. IEEE.
- Wu, H., Misra, D., & Chirikjian, G. S. (2020). Is That a Chair? Imagining Affordances Using Simulations of an Articulated Human Body. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7240–7246. IEEE.
- Xu, R., Chu, F.-J., Tang, C., Liu, W., & Vela, P. A. (2021). An Affordance Keypoint Detection Network for Robot Manipulation. *IEEE Robotics and Automation Letters*, 6(2), 2870–2877.
- Yao, B., Ma, J., & Fei-Fei, L. (2013). Discovering Object Functionality. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2512–2519.
- Zhao, Y., & Zhu, S.-C. (2013). Scene Parsing by Integrating Function, Geometry and Appearance Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3119–3126.