

Visual Affordances: Enabling Robots to Understand Object Functionality

Tommaso Apicella¹, Alessio Xompero², Andrea Cavallaro^{3,4}

¹Istituto Italiano di Tecnologia, Genoa, Italy

²Queen Mary University of London, London, United Kingdom

³Idiap Research Institute, Martigny, Switzerland

⁴École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

Abstract—Human-robot interaction for assistive technologies relies on the prediction of affordances, which are the potential actions a robot can perform on objects. Predicting object affordances from visual perception is formulated differently for tasks such as grasping detection, affordance classification, affordance segmentation, and hand-object interaction synthesis. In this work, we highlight the reproducibility issue in these redefinitions, making comparative benchmarks unfair and unreliable. To address this problem, we propose a unified formulation for visual affordance prediction, provide a comprehensive and systematic review of previous works highlighting strengths and limitations of methods and datasets, and analyse what challenges reproducibility. To favour transparency, we introduce the Affordance Sheet, a document to detail the proposed solution, the datasets, and the validation. As the physical properties of an object influence the interaction with the robot, we present a generic framework that links visual affordance prediction to the physical world. Using the weight of an object as an example for this framework, we discuss how estimating object mass can affect the affordance prediction. Our approach bridges the gap between affordance perception and robot actuation, and accounts for the complete information about objects of interest and how the robot interacts with them to accomplish its task.

Index Terms—Affordance, Semantic Segmentation, Object Detection, Mass Estimation

I. INTRODUCTION

Affordances are the potential actions that objects in the scene offer to an agent¹[1]. Because of such a broad definition, the computer vision and robotics research communities cast the prediction of affordances into different formulations such as grasping detection, affordance classification, affordance segmentation, and hand-object interaction synthesis [2], [3], [4], [5]. Each of these formulations addresses a part of the affordance prediction problem. The perception of affordances enables a robot to accomplish a task, selecting which objects in the environment to interact with, what actions to perform and how to perform these actions. In human-robot collaborations, considering objects and actions that are not harmful increases the challenge of affordance prediction [6], [7].

Learning to perceive object affordances from visual data is challenging due to the varying appearance of objects based

¹Person as an agent is the most intuitive reference in this definition. Nevertheless, the reference to a robot is also commonly used, especially in robotics research, and the one we consider in this work.



Fig. 1: Examples of applications benefiting from visual affordance prediction [8]: robot alone (left image), human-robot collaboration (middle image), wearable robotics and human-to-human collaboration mediated by wearable robotics (right image).

on the setting (e.g. single object on a tabletop or presence of clutter), the limited size of training datasets, and the characteristics of the end-effector affecting the possible interactions the robot can perform with the object. The observation of an object is affected by *environmental conditions*, such as illumination, background and clutter, camera viewpoint and distance, and object material (e.g. reflective), appearance (e.g. transparency or texture), and geometry (e.g. size or shape). While estimating different physical properties of an object is difficult from an image, no less is estimating the visual affordances of that object. For example, object geometry and affordance are related: concave shapes afford the holding of a content, or sharp regions afford cutting [9], [10], [11]. An object of interest can be observed as *occluded* by other objects in cluttered scenes [12], [13], [14] or by a human hand during a manipulation [5], [15], [16], [17]. Assistive robotics can target a specific set of objects (e.g. household containers) whose physical properties can vary during a manipulation (emptying or filling the object), or the object appearance might change due to transparencies or deformability [18] (see Fig. 2).

Most of the *datasets* for visual affordance have only a few tens of thousands images due to manual annotation of the object affordances, usually limited to a pre-defined set of object categories, object instances, or affordance classes [9], [12], [19]. Small-size datasets and pre-defined classes limit the generalization capabilities of learning-based models trained on these datasets. Only a few datasets [13], [20] exceed 100,000 images, with affordances automatically annotated: off-the-shelf methods [13] or simulators [20]. Training models



Fig. 2: Challenges in visual affordance prediction. First row: different object and different categories. Second row: different objects belonging to the same category type (intra-category diversity). Overall, objects can be captured in different scenes, backgrounds, different illumination conditions, under different poses, different occlusions, and from different viewpoints and distances. Images, cropped for visualisation purpose, are sampled from the UMD [9], IIT-AFF [12], and CCM datasets [15].

on dataset annotated with off-the-shelf methods, however, presents challenges such as generalization to different actions and objects, since the only action annotated was the grasping of object handles [13]. Training models on datasets annotated using simulators limits the generalisation to real images due to the presence of mixed-reality images (sim-to-real gap) [20]. Different *environments* (or context) can imply different affordances for the same object. For example, a screwdriver can be used to insert or remove screws in a laboratory through the graspable handle. In an environment where the object does not belong to (e.g. kitchen), the screwdriver can be used to tidy up the room and the whole surface of the object becomes graspable to be picked. The physical characteristics of a *robot end-effector*, such as size, degrees of freedom, and number of fingers, can influence the interaction with the object. Given the same object and task, two different two-finger end-effectors (or grippers) can interact with the object in different ways. For example, to pick-up a cup of diameter 5 cm, a gripper with a maximum opening of 2.5 cm can grasp the cup only from the rim. On the contrary, to pick-up the same cup, a gripper with 5.5 cm opening can grasp also the lateral surface of the cup body.

In robotics, most of the methods [2], [21], [22], [23], [24], [25] performing grasping considered the task as functional to only object picking, while ignoring the grasping as one interaction functional to different tasks (see the above broad definition of affordance). In this work, we align with the affordance definition related to the functional interaction with an object predicted from a visual input [9], [26], [27]. Given a task the robot has to perform, we consider as a visual affordance the combination of the following three aspects (*what, where, how*): the potential action on the most suitable objects in the image to accomplish the task; the region where the robot will interact with the object through its end-effector; and, the most physically plausible pose of the end-effector to interact with the object. All three aspects are conditioned

to the task the robot needs to accomplish and therefore only one action, one region, and one pose are possible instead of multiple candidates. Hassanin et al.'s survey [26] analyses the challenges of predicting affordances from a visual input, discusses methods for affordance categorization, detection, and segmentation, and conceptually relates affordance understanding to function understanding. However, methods are only compared in terms of their architectures without discussing the limitations of approaches and datasets. Chen et al.'s survey [28] extends the number of tasks and methods comparing the backbones and performance, discusses more in depth the relations among affordance subtasks, and describes the datasets in terms of modalities, object categories, and affordance categories. Nevertheless, the discussion of previous methods focuses on classifying and segmenting affordances without analysing methods that predict the end-effector pose enabling the interaction between the robot and the object. Ardón et al.'s survey [29] classifies previous robotic methods based on the amount of prior knowledge used to build the relationships between affordance components (target object, action, and action effects), discussing the generalisation to unseen objects or environments, and the characteristics of main robotic datasets. Although the analysis provides a roadmap on how to include the concept of affordance in robotic problems, the formulations and the connections between the robotic problems are not discussed. Despite providing a comprehensive comparison, these surveys do not discuss the inconsistencies of training setups that undermine the reproducibility and fair comparison of affordance methods. Moreover, none of the previous surveys discusses the limitations in the formulation of each tasks and provides a unifying view of visual affordance that enables a robot to interact with objects.

In this article, we unify the formulation for visual affordance prediction across its various tasks that were treated separately or appear disconnected in previous works and surveys. Through the lens of this unified formulation, we show its redefinition in each task and systematically review related methods and datasets, highlighting similarities and limitations of the methods and datasets. We also analyse reproducibility issues of previous works and propose an Affordance Sheet as a tool inspired by Model Cards [30] to overcome the reproducibility challenges while facilitating transparency when designing new visual affordance methods. Furthermore, we present the first framework that relates the estimation of object physical properties with visual affordance prediction to enable a robot to interact with the object in a more context-aware manner. Specifically, we show how estimating the object mass as a physical property of a manipulated container from an image is relevant for visual affordance prediction but still challenging to achieve and consequently challenging to embed into our proposed generic framework².

The article is organised as follows. We detail our unified formulation of the visual affordance prediction problem in Section II. We use our formulation as a roadmap to critically

²Project webpage: <https://apicis.github.io/aff-survey>

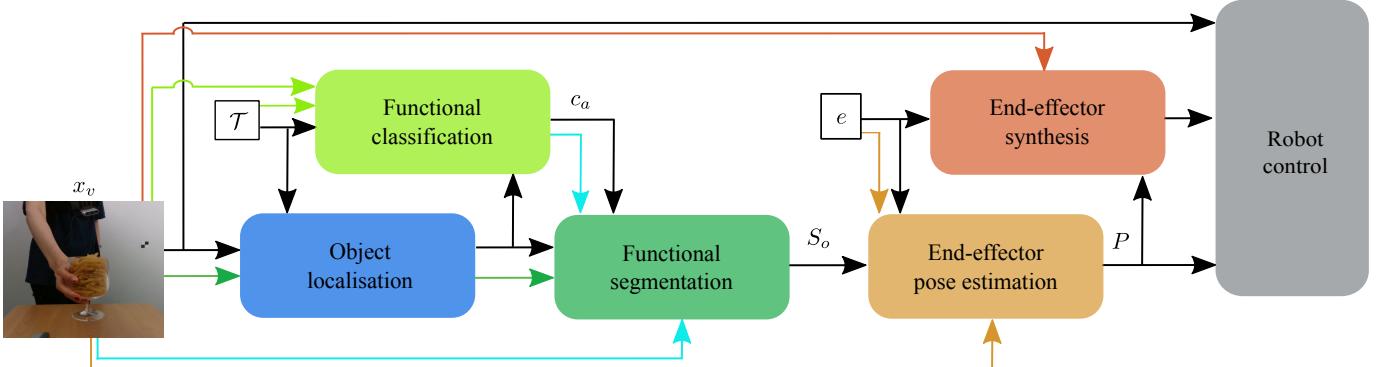


Fig. 3: Illustration of the unified visual affordance prediction framework for robotics. Previous works considered each block as a stand-alone affordance task unrelated from the other blocks, as also highlighted by the coloured lines. KEY: — Affordance classification, — Affordance detection and segmentation, — Affordance grounding, — Hand-object pose estimation, — Hand-object synthesis.

analyse the previous methods, highlighting similarities and differences, strengths and weaknesses, inside and across tasks. We analyse datasets to train and benchmark visual affordance prediction models in Section III. We discuss the performance measures to compare models in Section IV. We show a framework to estimate physical properties of objects and we discuss how the weight of an object affects the affordance prediction in Section V. In Section VI, we highlight the issue of reproducibility in previous works, highlighting characteristics and tendencies that slow down the advancements in the field along with potential solutions. We draw the conclusions in Section VII.

II. UNIFYING THE AFFORDANCE PROBLEM FORMULATION

In this section, we define a unified formulation of visual affordance prediction as the identification of the actions that a robot can perform with its end-effector on an object of interest to accomplish a task based on the visual input capturing the observed scene. We present our overall framework representing the unified formulation and comprehending the redefinitions of affordance prediction i.e., the problems tackled by previous works. We then discuss data-driven approaches that, from an RGB image, identify the object of interest in the scene, predict the object regions the robot can interact with, and predict the pose of the robot end-effector to perform the interaction.

A. Problem formulation

Let $x_v \in R^{F \times W \times H \times C}$ be the observed scene, where F is the number of frames of an image sequence, W is the image width, H is the image height, C is the number of channels ($C = 3$ for an RGB input), and v is the camera view index in a multi-camera setup. Let $\mathcal{T} = \{t_m \mid t_{m-1} < t_m < t_{m+1}\}$ be a task a robot needs to perform and represented by a sequence of steps t_m expressed as text³. For example, a robot can be instructed to perform a task with the following steps: “close the bottle” and “move the bottle to the shelf”, or only “close the bottle”. Let \mathcal{E} be the set of available end-effectors (e.g. a

bi-manual robot) that can interact with the objects, and $e \in \mathcal{E}$ encode the characteristics of the end-effector (size, number of fingers, and degrees of freedom) in a parametric model (e.g. MANO [31]). Let \mathcal{O} be the set of objects relevant for the task (objects of interest). Objects can be localised using an intermediate model of object detection from the image x_v and task \mathcal{T} . Each object $o \in \mathcal{O}$ can be represented as a bounding box $b \in \mathbb{R}^4$, indicating the position and size in x_v , an object class λ , and a confidence c : $o = [b, \lambda, c]$. Let \mathcal{A}_o be the set of potential actions that the robot performs on the object and each action $a \in \mathcal{A}_o$ can be expressed in text form. For example, for the task “close the bottle”, the robot perceives the *graspable* action of the bottle cap. Let \mathcal{S} be the set of image regions on the object o the robot can interact with to perform the action a . In general, to each action and object corresponds an interaction region $S_o^a \in \mathcal{S}$ on the objects of interest. S can be represented as a probability map $[0, 1]^{W \times H}$ having zero values in the pixels belonging to the background and values greater than zero on the object pixels. To perform the action, the robot estimates how the end-effector interacts with the object i.e., the pose of the end-effector P on the interaction region of the object, also indicating how the fingers should close. For each object and action, the pose of the end-effector can be represented as a rotation-translation matrix $P = [R|T] \in SE(3)$, where $SE(3)$ is the special Euclidean group, $R \in SO(3)$ is a 3×3 rotation matrix in the special orthogonal group, and $T \in \mathbb{R}^3$ is the translation vector in the Euclidean space. With the pose, the end-effector can be rendered on the image plane to generate $\tilde{x}_v \in R^{F \times W \times H \times C}$ representing the interaction between the end-effector and the object.

For a given task \mathcal{T} and a visual input x_v , we define a visual affordance as a region S that enables a robot with its end-effector e to perform an action a (and P) on a relevant object o . A visual affordance model is a function that maps the observed scene x_v , the task \mathcal{T} , and the end-effector e , into the objects of interest o , the potential action a , the regions of interaction

³When the cardinality of the set \mathcal{T} is 1, t_m is the task itself.

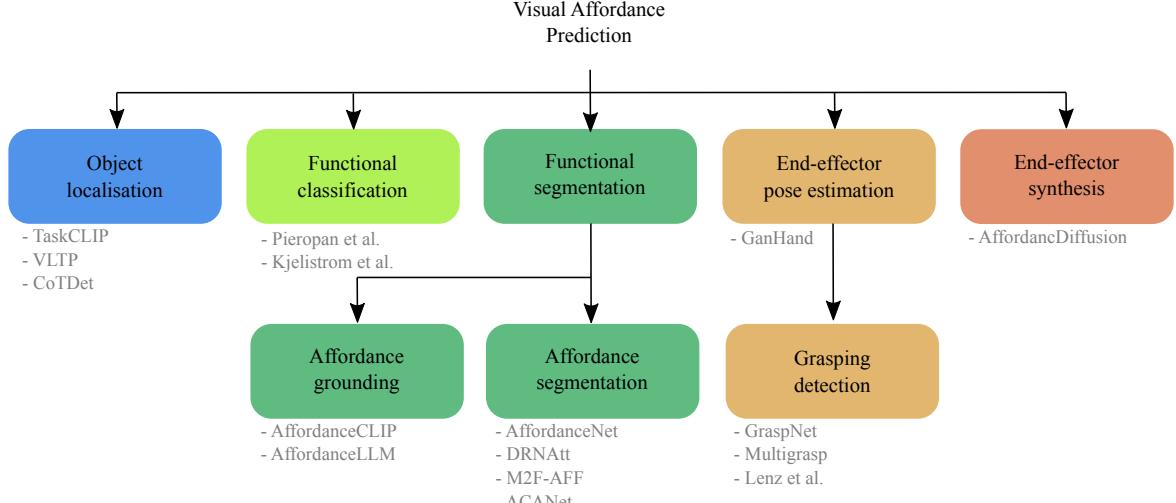


Fig. 4: Proposed taxonomy of tasks related to visual affordance prediction.

S , and the pose of the end effector P :

$$f(x_v, \mathcal{T}, e) \rightarrow \{a, o, S, P\}. \quad (1)$$

When limited to only an RGB (RGB-D) image, we simplify the visual input to $I \in R^{W \times H \times 3}$ ($I \in R^{W \times H \times 4}$). We refer the reader to other works on affordance prediction from an RGB-D input [32], [33], [34], [35], [36], [37], [38], [39] and from multi-view inputs (including stereo) [7]. Note that, in this article, we focus on methods for visual affordance prediction from an RGB image and on the single end-effector case.

Removing the end-effector e , the task \mathcal{T} , or both, increases the number of possible solutions, making the problem too generic and irrelevant for robotics. An object can offer multiple actions for the same region, multiple regions can support the same action, and the region might not be realistic or feasible for specific agents (e.g. a robot with a 2-finger gripper). Moreover, there are additional aspects that can influence the affordance: the physical properties of the robot end-effector, the physical properties of the object, and the presence of occlusions (e.g. a person holding the object with their hand). These aspects can be included as an additional input to the visual affordance model when the problem cannot be solved, for example due to the lack of annotated data or a change in the object mass due to a manipulation or presence of a content within the object.

B. Overall framework

We propose a framework that integrates the redefinitions related to affordance prediction (see Fig. 3) given the task to accomplish and the RGB(D) image. Our framework decomposes the task of the robot in the following subtasks and related components:

- 1) Localises the object of interest (*object localisation*).
- 2) Predicts the actions for each localised object (*functional classification*).

- 3) Predicts the object regions that enable to perform the action (*functional segmentation*).
- 4) Estimates the end-effector pose on the object, given the end-effector model and previous extracted information (*end-effector pose estimation*).
- 5) Renders the end-effector on the RGB(D) image (*end-effector synthesis*).
- 6) Reaches the estimated target pose, keeping into account the desired result also from a visual point of view through the rendered end-effector pose (*control process*).

We use our unified framework as a reference to critically analyse the previous works through the various subtasks that compose the affordance prediction problem. Each component of the framework instantiates one or more subtasks as shown in the proposed taxonomy of Fig. 4. For example, functional segmentation groups *affordance segmentation* and *affordance grounding*, as these two subtasks have a similar problem formulation; or *grasping detection* can be considered as a special case of *end-effector pose estimation*, assuming a 2-finger end-effector.

C. Object localisation

Given an image I and a task \mathcal{T} , the model predicts a set of bounding boxes $\{b_o\}_{o=1}^O$ with $b \in \mathbb{R}^4$ a binary segmentation mask $\{S_o\}_{o=1}^O$, with $S \in [0, 1]^{W \times H}$,

$$\{b_o, S_o\}_{o=1}^O = f(I, \mathcal{T}). \quad (2)$$

The challenges of task-driven object detection task lie in fusing information about object appearance and context. Some objects in the scene are not relevant for the task and different object categories can be used for the same task, thanks to the similarities in function.

We summarize the characteristics of task-driven object detection and segmentation methods in Table I. Methods tackle this task either with a single architecture trained end-to-end [40], [49] or with the combination of different mod-

TABLE I: Comparison of task-driven object detection and segmentation models.

Method	Source	Backbone		Output			GOR	AL	LLM	TAF
		Vision	Language	TC	BB	SEG				
GGNN [40]	GGNN [41]	RN-101 [42]	-	●	●	○	●	○	○	○
TaskCLIP [43]	CLIP [44]	ViT-H [45]	RoBERTa [46]	○	●	○	○	●	●	●
VLTP [47]	SAM [48]	ViT-H [45]	SAM (encoder) [48]	○	○	●	○	○	●	?
TOIST [49]	DETR [50]	RN-101 [42]	RoBERTa [46]	○	●	●	○	○	○	●
CoTDet [51]	DETR [50]	RN-101 [42]	RoBERTa [46]	○	●	●	○	○	●	●

KEYS – GOR: graph-based objects relationship, AL: vision-language alignment, LLM: large language model, TAF: attention-based fusion between task and vision features, TC: task classification, BB: bounding box, SEG: segmentation, RN: ResNet, ●: considered, ○: not considered.

els [43], [47], [51]. To understand the context and what objects are the most relevant to accomplish a task, GGNN [40] learns the dependency between objects in the scene using Graph Neural Networks, where each graph node represents an object. For each object, the model predicts the probability of being suitable for the task. However, the assumption of a closed set of tasks and objects limits the generalization to unseen objects and unknown tasks. TOIST [49] overcomes the limitation of closed set of objects for each task adapting DETR object detector [50] through a teacher-student training paradigm. The preference between objects is learned by replacing the object noun with an indefinite pronoun in the student task and by distilling the teacher knowledge from the output tokens. The representation of verb-pronoun is learned by replacing the pronoun tokens in the student with the closest noun token in the teacher using nearest neighbour. Other methods tackle the generalization to unseen objects and tasks, integrating Vision-Language Models or Large Language Models (LLM) [51], [43]. For example, CoTDet prompts an LLM to list the objects required to accomplish a task, the rationale that makes each object useful, and the (textual) features related to the rationale. Cross-attention combines the vision tokens with the textual features [51] to predict the object bounding box in the image. The alignment between objects and task can also be obtained adapting CLIP [44]. TaskCLIP applies cross-attention between vision and text tokens before the similarity, and a score function based on self-attention selects the objects that are more suitable for the task using the similarity matrix [43].

D. Functional classification

Functional classification, also referred to as affordance classification or affordance recognition, identifies *what* are the potential actions (or affordance classes) c that a robot can perform on an object from an input image I given a task \mathcal{T} ,

$$\{c_a\}_{a=1}^A = f(I, \mathcal{T}). \quad (3)$$

One of the main challenges of affordance classification is that the same object in the scene has multiple affordances that vary depending on the task. For example, a cup on a table can suggest the action of picking or filling, but until a task is defined (e.g. ‘move the cup’), both affordances are plausible. Another challenge is the fact that objects with similar appearances might afford different actions. For example, some models of trowel and turners might be similar in colour and

TABLE II: Characteristics and comparison of methods for function classification (or also known as affordance classification). Note that these methods use either object detection, object segmentation, object classification, or their combination as auxiliary tasks.

Reference	Source	Backbone	Depth	DET	SEG	CLS
Sun et al. [52]	PGM [53]	-	○	○	○	●
Zheng et al. [54]	Faster R-CNN [55]	VGG [56]	○	●	○	○
Pieropan et al. [3]	SVM [57]	-	●	●	●	○
Kjellström et al. [58]	FCRF [59]	-	○	○	○	●

KEYS – Source: source architecture, DET: object detection, SEG: object segmentation, CLS: object classification, PGM: probabilistic graphical model, SVM: support vector machine, FCRF: Factorial Conditional Random Field, ●: considered, ○: not considered.

shape, however the surface of a trowel is used to *scoop*, while the surface of a turner to *support*.

We summarise the characteristics of affordance classification methods in Table II. Methods learn actions that can be performed with objects in the scene either from human demonstration [3], [58], or from images of the environment [52], [54]. Sun et al. [52] used Probabilistic Graphical models to relate affordances with the appearance of objects encoding the image using Principal Component Analysis. The main limitation of the method is the limited scalability to high-dimensional signals like images and the increasing complexity of the graph structure when adding affordance categories. To limit the input dimensionality and focus only on regions of interest in the image, Zheng et al. [54] use object detection before predicting the action related to object regions. Moreover, the combination of affordance classification with auxiliary tasks such as detection and segmentation allows to group objects based on the actions they are used for (functionality), instead of their appearance [3], [54]. By training methods on data of people using objects or with the robot exploring the environment, previous works [3], [52], [58] implicitly considered as a task the functional use of the object. These methods do not enable the physical interaction of a robotic hand with the object, as the action is not associated with an interaction region in the image [52], [54], [58]. This results in the robot having multiple options (ambiguity) on how and where to perform the interaction. For example, even a simple instruction like “move the cup” can suggest a robot multiple ways to perform the interaction, such as grasping the cup by the body or by the rim.

TABLE III: Characteristics and comparison of visual affordance segmentation models [4]. Note that we report the best-performing backbone for each model, and we do not consider additional parts of the pipelines, such as a separate object detector.

Model	Architecture		Attention		Aff. Object		CRF						
	Source	Backbone	FPN	IF	Sp	Ch	Sa	Mc	C	E	C	S	L
ADOSMNet [60]	PSPNet [61]	RN-101 [42]	○	○	○	○	○	○	○	○	○	○	○
CNN [62]	SegNet [63]	VGG-16 [56]	○	○	○	○	○	○	○	○	○	○	○
RN50-F [17]	Fast-FCN [64]	RN-50 [42]	○	○	○	○	○	○	○	○	○	○	○
BB-CNN [12]	DeepLab [65]	VGG-16 [56]	○	○	○	○	○	○	○	○	○	○	●
DeepLab [16]	DeepLab [65]	RN-101 [42]	○	○	○	○	○	○	○	○	○	○	●
ACANet [20]	UNet [66]	RN-18 [42]	○	○	○	○	○	○	○	●	○	○	○
AffordanceNet [10]	Mask R-CNN [67]	VGG-16 [56]	○	○	○	○	○	○	○	●	●	○	○
4C-RPN-5C [68]	AffordanceNet [10]	SE-RNX-101 [69]	○	○	○	○	○	○	●	○	●	○	○
B-Mask R-CNN [70]	Mask R-CNN [71], [68]	RNX-101 [72]	●	○	○	○	○	○	●	○	●	○	○
A-Mask R-CNN [73]	AffordanceNet [10]	RN-50 [42]	●	○	○	○	○	○	●	○	●	○	○
GSE [74]	HRNet [75], [76]	RNS-101 [77]	○	●	○	●	○	○	○	○	○	○	○
DRNAtt [78]	DANet [79]	DRN [80]	○	○	●	●	○	○	○	○	○	○	○
SEANet [81]	DFF [82]	RN-50 [42]	○	●	●	●	○	○	●	○	○	○	○
BPN [83]	AffordanceNet [10]	RN-50 [42]	●	○	●	●	○	○	●	●	●	●	○
RANet [84]	EncNet [85]	RN-50 [42]	○	○	○	●	○	○	●	●	○	○	○
STRAP [86]	SINN [87]	RN-50 [42]	○	●	○	●	○	●	●	○	○	○	●
M2F-Aff [4]	Mask2Former [88]	RN-50 [42]	●	○	○	●	●	●	●	●	○	○	○

KEYS – Source: reference architecture, Backbone: visual encoder, Sp: spatial attention, Ch: channel attention, Mc: masked cross-attention, Sa: self-attention, Aff.: affordance, C: classification, E: edge segmentation, S: segmentation, L: localisation, RN: ResNet, RNX: ResNeXt, RNS: ResNeSt, SE-RNX: squeeze and excite ResNeXt, DRN: Dilated Residual Network, CRF: conditioned random fields, IF: intermediate feature maps fusion; ●: considered, ○: not considered.

E. Functional segmentation

The segmentation of functional regions on objects in the image identifies *where* the robot needs to perform the interaction with the object and is approached in the literature in two main ways (see Fig. 4): *affordance detection and segmentation* aims at finding the objects of interest in the image and separating the functional regions on the object, and *affordance grounding* details the task to accomplish and identifies the region on the object that should be used to perform the action.

Affordance detection and segmentation. Given an image I , the model predicts bounding boxes $\{b_o\}_{o=1}^O$ and segmentation masks of A functional regions $\{S_o\}_{o=1}^O$ for a set of objects of interest,

$$\{b_o, S_o\}_{o=1}^O = f(I, \mathcal{T}). \quad (4)$$

The segmentation mask S_o can be also formulated as the combination of the actions $\{c_a\}_{a=1}^A$ with a probability map $\{S_{o,a}\}$ where $S \in [0, 1]^{W \times H}$ indicates the region where an action takes place for each object [20], [86]. Affordance detection and segmentation formulation assumes that the objects of interest are the ones annotated in the dataset, that the task \mathcal{T} is to use the object to fulfil the purpose it was designed for [9], [10], [12], [62], and that different parts of the objects are associated with a functionality to accomplish the task. For example, a knife is designed to cut another object and therefore the handle is designed to be grasped while the blade is used for the cutting. Therefore, the models should localise the object that offers “cut” as an affordance (affordance detection) and segment the blade as a part of the knife offering the action “cut” (affordance segmentation).

Previous methods [63], [65], [79], [67], [66], [64], [61] adapt semantic and *instance segmentation* architectures to predict affordance regions on the objects. For example, A-

Mask R-CNN [73] and AffordanceNet [10] modify an instance segmentation model (Mask R-CNN [67]) to predict the affordance masks instead of the object masks for each object localised in an image. Starting from the design of AffordanceNet, BPN [83] and 4C-RPN-5C [68] align the region of interest with the feature maps at different resolutions and predict the overlapping of bounding boxes and boundaries of affordance regions. An off-the-shelf object detector localises regions of interest in the image, however inaccurate or wrong predictions can consequently result in segmenting affordance regions within image parts outside of the actual objects. When edges are blurred or not clearly defined (e.g. occlusions or transparent objects), BPN fails to predict precise affordance contours despite its additional affordance edge segmentation component in the model architecture. Instead, semantic segmentation models [17], [20], [62], [78], [81], [84], [74] avoid the dependence from an object detector and assign each pixel of the image to an affordance class during supervised training (per-pixel affordance segmentation). When objects are occluded or boundaries are not clearly defined, methods such as CNN [62], RN50-F [17], and ACANet [20] can classify affordance pixels outside the object region.

Attention mechanisms [83], [78], [81], [84], [74] are an alternative way to consider only relevant information in the image by weighing feature maps extracted from the image. For example, GSE [74] learns the channels weight with supervision of the affordances. DRNAtt [78], SEANet [81], and BPN [83], learn similarities between channels or positions in the feature maps without direct supervision, updating the weights of the layers that compute the attention map during training. For computational reasons, both DRNAtt and GSE process feature maps at low-resolutions where important details (e.g. edges) for affordance segmentation are degraded when the object is

not close to the camera. In RANet [84], the attention weights are learned with the supervision of object classes. However, in case of occlusions, mistakes in the attention weights cause the mismatch between the predicted object classes and the segmented affordances.

Previous methods [17], [20], [62], [78], [81], [84], [74] coupled the classification of the affordances and the segmentation of regions in the image. However, the two subtasks can be decoupled by performing class assignment at the level of each segmentation mask [86]. For example, STRAP [86] is designed as a multi-branch architecture that learns the affordance classification in one branch and the segmentation masks in another branch. The model is trained to segment the affordance mask with a weakly supervised supervision from a point annotation of a region [89] and by using Conditional Random Fields that process the pixel position and colour. However, this approach can lead to inaccurate segmentations when the object colour is not clearly distinguished from the background [16], [62]. STRAP also uses self-attention to process low-resolution feature maps extracted by the backbone, losing details about the object in the image when the object scale is small. To increase the resolution of processed feature maps, M2F-AFF [4] adapts Mask2Former [88]. Mask2Former added *masking* in the cross-attention mechanism to combine the features extracted from the image with learnable latent vectors, while ignoring the pixel positions outside the object region (background).

Table III summarises the characteristics of the methods we discussed for affordance detection and segmentation. Note that the segmentation of affordance regions is tackled independently from the end-effector. Affordance regions can be mapped to the robot control process to perform the actions [10], [12], [83], yet they represent a coarse information, since characteristics of end-effectors such as the number of fingers or the degrees of freedom influence the contact regions on the object (more fine-grained than affordance regions).

Affordance grounding. Given an image I and a task \mathcal{T} , the model predicts the probability map $\{S_o\}_{o=1}^O$ to identify the image region that the agent uses to interact with the object,

$$\{S_o\}_{o=1}^O = f(I, \mathcal{T}). \quad (5)$$

The task \mathcal{T} can be expressed through natural language [90], an affordance category [11], [91], [92], [93], a point in 2D [94], [95], or another image of the object of interest [96], [97], [98]. With this formulation, one-shot methods that use prior information (e.g. an image) can also be seen as affordance grounding methods.

We summarise the characteristics of affordance grounding methods in Table IV. Most of the methods use the action to be performed on the object as a prior [11], [91], [92], [95], however the prior for the affordance grounding can be a 2D query point in the image plane [94], a query image [96], [97], [98], or the task in natural language [90]. These priors enable affordance grounding methods to tackle generalization to different object categories while avoiding an explicit object detection phase. One approach to perform affordance segmentation is through weak supervision i.e., using methods for explainability

TABLE IV: Characteristics and comparison of affordance grounding methods.

Method	Prior			Vid-img		Exo-ego		CAM	Supervision	
	2D-P	IMG	CLS	Task					strong	weak
3DOI [94]	●	○	○	○	○	○	○	○	●	○
LOCATE [96]	○	●	○	○	○	●	●	○	●	●
AffCorrs [97]	○	●	○	○	○	○	○	○	○	○
Demo2Vec [98]	○	●	○	○	●	○	○	●	○	○
Hotspots [91]	○	○	●	○	●	○	●	○	●	●
Cross-View-AG [11]	○	○	●	○	○	●	●	●	○	●
OVAL-Prompt [92]	○	○	●	○	○	○	○	○	○	○
AffordanceCLIP [95]	○	○	●	○	○	○	○	○	●	○
OOAL [93]	○	○	●	○	○	○	○	○	●	○
AffordanceLLM [90]	○	○	●	●	○	○	○	●	●	○

KEYS – Vid-img: transfer from video to image, Exo-ego: transfer from exocentric to egocentric view, CAM: Class Activation Maps, 2D-P: point in image, IMG: a support image/region, CLS: action class, ●: considered, ○: not considered.

for example Class Activation Maps (CAM) [99] to obtain the region in the image that corresponds to the action [11], [91], [96]. The use of CAM, which highlights coarse image regions not bounded by object contours, limits the application of these methods to unoccluded object settings. One-shot-based methods use an image as a prior to select objects of interest [19], [100], or segment affordance regions [97], based on the similarity between the input images and the prior (query image). However, the support image is assumed to be similar to the query images, thus implying that the object category in the scene should be known in advance.

To cope with the limited amount of training images, methods learn affordances using pretrained multimodal models [90], [92], [93], [95], using knowledge transfer from video to images [91], [98] or from exocentric views of objects to the egocentric ones [11], [96]. In particular, multimodal models help generalising to unknown object categories or unknown actions (open vocabulary). For example, Affordance-CLIP adapts CLIP [44] with a learnable feature pyramid network to predict the probability map [95]. A contrastive loss encourages pixel-level embeddings within the annotated mask of the object to align with the corresponding linguistic feature. AffordanceLLM [90] processes vision and language information using a Large Language Model (LLM) to predict tokens for the affordance segmentation. This LLM generates text tokens that are converted into a description of the part used to perform the task and a mask token that is combined with the visual tokens using a transformer decoder to predict the affordance map. Few of the methods [11], [96] for affordance grounding focused on learning object affordances building correspondences from the exocentric view of an object (human using the object) to the egocentric one (object only). Both LOCATE [96] and Cross-View-AG [11] during training combine a loss to learn the affordance category with strong supervision with losses to preserve the similarity between the outputs of the exocentric and egocentric images processing. Instead of learning directly from images, methods like Demo2Vec [98] and Hotspots [91] learn to transfer the affordance from videos of humans interacting with objects in household environments e.g., oven, fridge, washing machine,

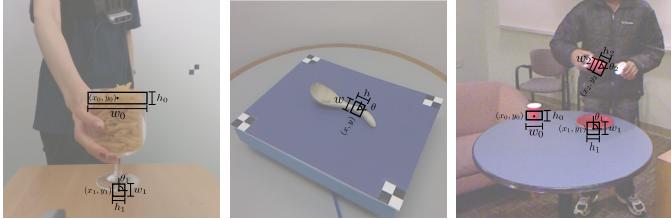


Fig. 5: The pose of a two-fingers gripper encoded as an oriented rectangle on the image plane [24]. The 5 dimensions of the rectangle encode: the centre (x , y), the opening (w), the fingers length (h), and the orientation θ .

to the images containing only the objects. Although these methods [11], [91], [96], [98] can learn the object affordances from the information of how humans performs actions with the object, the egocentric views of the objects are composed by the object on a plain background without occlusions or clutter, limiting the generalisation to in-the-wild images.

Despite generalising to different actions or task formulations, affordance grounding methods output confidence maps that are not bounded by object edges. The confidence maps could also overlap with other objects in case of clutter or with a human hand if the object is hand-held. Using a coarse confidence map when interacting with an object can lead a robot to misplace the end-effector, thus undermining the success of the interaction or harming the human.

F. End-effector pose estimation and synthesis

To perceive the visual affordance, the robot needs to predict also *how* the interaction with the object will be performed i.e., the pose of the end-effector on the object. Previous works [2], [14], [23], [24], [25], [101], [102], [103], [104] related the problem mostly to grasping rather than to visual affordances, and referred to the problem differently depending on the characteristics of the end-effector: *grasping detection* in case of a two-fingers gripper [2], [23], [24], [25], [103], *hand-object pose estimation* in case of the human hand [5], [14] (e.g. MANO model [31]), *multi-finger grasping* in case of the three fingers Barrett hand [101]. Since also the hand can be considered as a multi-finger end-effector, we merge the discussion on three fingers and human hand into *multi-finger pose estimation and interaction synthesis*. Once the end-effector model is chosen, and given the image of the object and the task, the model predicts the pose P of the end-effector on the object,

$$\{P_o\}_{o=1}^O = f(I, \mathcal{T}, e). \quad (6)$$

Predicting the pose of the end-effector, however, is challenging because the end-effector is not observed in the image, and therefore only the visual features of the object can be used.

Grasping detection. Assuming a two-fingers gripper as end-effector, the pose estimation of the end-effector on the object can be reformulated as a detection of grasping points directly on the image, encoding the parameters of the gripper as a rectangle [2]. The poses of the gripper on the object are detected using an oriented rectangle representing the 5 DoFs

TABLE V: Characteristics and comparison of grasping detection models.

Method	Backbone	D	2stages Modality fusion		Auxiliary tasks			
			EAR	MID	GLIKE	GSEG	DET	SEG
MultiGrasp [102]	AXN [105]	•	○	○	○	●	○	○
Kumra et al. [25]	RN-50 [42]	•	○	○	●	○	○	○
GraspNet [103]	-	•	○	○	○	○	●	○
Ainetter et al. [104]	RN-101 [42]	○	●	○	○	○	○	●
Lenz et al. [2]	-	●	●	●	○	○	○	○
Chu et al. [23]	RN-50 [42]	●	●	○	○	○	○	○
ROI-GD [24]	RN-101 [42]	●	●	○	○	○	●	○

KEYS – D: depth, EAR: early fusion, MID: middle fusion, GLIKE: grasp likelihood, GSEG: grasping segmentation, DET: object detection, SEG: object segmentation, AXN: AlexNet, RN: ResNet, ●: considered, ○: not considered.

of a parallel plate gripper on the image plane (see Fig. 5): 1 DoF encodes the gripper rotation with respect to the horizontal axis, 2 DoFs encodes the translation of the gripper centre (on the horizontal and vertical axis), and 2 DoFs encodes the geometry of the gripper (1 for the opening width and 1 for the fingers length). The underlying assumption is the availability of a depth map to obtain the full 7 DoFs representation of the gripper in 3D (3 DoFs for translation, 3 DoFs for rotation, and 1 DoF for the opening width). Given an RGB-D image $I \in \mathbb{R}^{W \times H \times 4}$, the model predicts a set of G oriented rectangles $\{r_g\}_{g=1}^G$ with $r \in \mathbb{R}^5$ consisting of the coordinated of the rectangle centre, the rectangle size (width and height), and the orientation. Predicting the pose of a two-fingers gripper on the surface of an object is challenging because each object has multiple grasping points, but only a part of grasping poses leads to a successful grasping. Moreover, when estimating the grasping points from a single view, only a side of the object is visible, limiting the number of feasible grasping points. Such a formulation, i.e. predicting an oriented rectangle, corresponds to the formulation of object detection and therefore object detection models, pre-trained on large-scale datasets, can be re-used and retrained or fine-tuned for grasping detection [24], [104]. However, the oriented rectangle representation can lead to a wrong detection of the grasping points if the object is in a challenging pose, and consequently the robots can fail to grasp the object.

Table V summarises methods for grasping detection. Most of the methods [2], [23], [24], [25], [103] use RGB-D images to predict grasping rectangles, as the depth information provides geometric cues to the model. Different ways of fusing visual information include learning to fuse the information in the first layers of the method [2], [23], [24], [103] (early fusion), or using a separate backbone to process RGB and depth before fusion (middle fusion) [25]. However, there are no results showing that a fusion mechanism is more effective than the others. The feature extraction is performed mainly convolutional networks like ResNet [23], [24], [25], [104] or AlexNet [102] pre-trained on ImageNet [105], to transfer the feature representations learned on large scale datasets. Similarly to object detection, methods can be categorised into single-stage and and two-stage approaches. Single-stage methods [25], [102], [103] predict the final oriented rectangles from

TABLE VI: Characteristics and comparison of multi-finger pose estimation and interaction synthesis methods.

Method	Obj. pose	Grasp		Learning	
		CLS	LOC	ADV	DIFF
Multi-FinGAN [101]	•	•	○	•	○
GanHand [14]	•	•	○	•	○
AffordanceDiffusion [5]	○	○	•	○	•

KEYS – CLS: category, LOC: location, ADV: adversarial based, DIFF: diffusion based, •: considered, ○: not considered.

the image, either directly regressing the rectangle [25], [102] or considering the rectangle as a by-product of object segmentation [103]. Two-stage methods [2], [23], [24], [104] first predict grasping candidates (coarse estimation) and then refine the predictions (fine estimation). The majority of two-stage methods adapt works for object detection (e.g. Faster R-CNN [55]) to grasping detection in different ways: separating the learning of the object location and grasp locations [24]; separating the learning of the quantized angle from the learning of the centre, width and height of the grasping rectangle [23]; separating the coarse prediction of grasping rectangles from the refinement based on the semantic segmentation of the object [104]. Auxiliary tasks, such as object detection [23] and segmentation [104], can help constrain the prediction of the grasping rectangle to the part of the image containing the object, reducing mistakes in case of cluttered scenes or when the object is not in foreground and completely visible. Other auxiliary tasks are the likelihood of an image patch (non-overlapping piece of the image) containing a grasp [102] limiting the predicted grasping rectangles number to some parts of the image, and the grasping region segmentation [103] constraining the grasping rectangle to graspable region of the object e.g., the handle of a spoon.

Such a formulation of grasping detection lacks a definition of the visual affordance, resulting in non-physically plausible solutions as the robot can grasp the object at any location on the object’s surface. For example, the rim of a cup filled with liquid (suggesting the affordance of pouring the content) might be selected as a potential grasping point without considering that the liquid might be spilled or damage the end-effector. Most of the methods for grasping detection [2], [23], [24], [25], [102], [106] use the assumption that objects are observed on a tabletop or on the floor (top-down camera view). Hence, models may fail to generalise to scenarios with different camera view-points or with occlusions (e.g. hand-occlusion in human-robot collaborations).

Multi-finger pose estimation and interaction synthesis.

Previous works [5], [14], [101] considered as visual affordance the pose of an end-effector, with more than two fingers, on objects in the scene. Given an image I , the model predicts the 6D pose of the end-effector on the object $\{[R|T]_o\}_{o=1}^O$ with $[R|T]$ representing pose of the end-effector (pose estimation) and renders an image of the end-effector, $\tilde{I} \in \mathbb{R}^{W \times H \times 3}$ (interaction synthesis).

Table VI compares the characteristics of methods for multi-finger pose estimation and interaction synthesis. Methods are

based on a coarse-to-fine approach that first locates where the end-effector is supposed to interact with the object and then refines the pose using generative adversarial networks [14], [101] or diffusion models [5]. GanHand [14] estimates objects’ shapes and locations using a sub-network, either an object 6D pose estimator or a reconstruction network. The predicted shape is projected onto the image plane to obtain a segmentation mask that is concatenated with the input image and fed to a second sub-network that predicts the grasp type, i.e. the type of interaction between end-effector and object. A module predicts the coarse pose of the end-effector from the grasp type and visual features. The network refines the end-effector parameters and obtains the final shapes and poses (i.e. MANO model [107]) by minimising an adversarial loss with a discriminator. Multi-FinGAN [101] adapts GanHand architecture to perform the pose estimation of the Barrett end-effector on the object in the image. In Multi-FinGAN the reconstruction of the object is not projected on the image to focus only on the object region, but only to refine the coarse pose of the end-effector. As a consequence, the method underperforms if multiple objects are present in the scene. AffordanceDiffusion [5] uses two diffusion models to generate the image of the end-effector interacting with the object in the image. The diffusion process is based on a end-effector prior (forearm mask) that is composed by a circle representing the end-effector and a rectangle representing the forearm. For every diffusion step, the first model (i.e. LayoutNet [5]) predicts the denoised forearm mask from the features of the forearm mask obtained in the previous step, the object image, and the forearm mask projected on the object image. The predicted layout mask (prior) and the object image are used as input to the second model (i.e. ContentNet [5]) that synthesises an image of the interaction between end-effector and object.

Methods for end-effector pose estimation and synthesis focus on a generic grasping interaction with the objects in the image, without taking into account the task that the robot performs and the affordances that the object supports. This fact can result in estimating or synthesising poses that are not aligned with the task, leading to interactions with the object that do not assist the human.

G. Future directions: AI agents and human-in-the-loop

Extension of our perception-robotic framework to multi-modal inputs, and especially the use of language, can lead to its re-design as an *agentic system* [108]⁴. The framework requires steps that are commonly refer to as understanding (perception part), reasoning (relating affordances, objects, and physical properties while being conditioned to the task to accomplish), planning (actuation part of the robot to accomplish the task), and recovering from errors when referring to AI agents. One way to recover from errors is to feed the model with previous predictions to correct the mistakes [109]. Learning to predict visual affordances for hand-object interactions can benefit

⁴<https://www.anthropic.com/engineering/building-effective-agents>
<https://blogs.nvidia.com/blog/what-is-agentic-ai/>

TABLE VII: Characteristics of datasets for visual affordance prediction grouped by task.

Task	Dataset	# Images	OBJ	AFF	Real	Tran.	3PV	HOc
OBJD	Rio [113]	40,214	-	-	●	○	●	○
	COCO-Task [40]	39,724	49	-	●	○	●	○
AFFC	Pieropan et al. [3]	~40,000	4	4	●	○	●	○
	Zheng et al. [54]	740	8	3	●	○	●	○
	Sun et al. [52]	1400	7	6	●	○	●	○
	Kjellström et al. [58]	11,500	6	3	●	○	●	●
AFFG	OPRA [98]	-	-	7	●	○	●	○
	AGD20K [11]	23,816	47	36	●	○	●	●
AFFDS	AFF-Synth [114]	30,245	21	7	○	○	●	○
	UMD-Synth [115]	37,200	17	7	○	○	●	○
	Multi-View [116]	47,210	37	15	●	○	●	○
	HANDAL [13]	308,000	17	1	●	○	●	●
	TRANS-AFF [33]	1,346	3	3	●	●	●	○
	UMD [9]	28,843	17	7	●	○	●	○
	IIT-AFF [12]	8,835	10	9	●	○	●	●
	CAD120-AFF [16]	3,090	11	6	●	○	●	●
	FPHA-AFF [17]	4,300	14	8	●	○	○	●
	CHOC-AFF [20]	138,240	3	3	●	○	●	●
GDET	Cornell grasping [117]	1,035	-	1	●	○	●	○
	GraspSeg [103]	33,188	15	1	●	○	●	○
	Jacquard [106]	54,485	-	1	○	○	●	○
	OCID [104], [118]	-	-	1	●	○	●	○
HOIS	EPIC-Kitchens [119]	-	-	1	●	○	○	○
	YCB-Affordance [14]	133,936	58	1	●	○	●	○
	HO3Pairs [5]	-	-	1	●	○	○	●

KEYS – # Images: number of images, OBJ: number of object categories, AFF: number of affordance categories, Tran.: transparency, 3PV: third person view, HOc: hand-occlusion. OBJD: task driven object detection; AFFC: affordance classification; AFFG: affordance grounding; HOIS: hand-object pose estimation and interaction synthesis; GDET: grasping detection; AFDSD: affordance detection and segmentation; ●: considered, ○: not considered, ◉: partially considered.

from human demonstrations of the actions to perform, in the same way humans prompt models with examples showing how to solve tasks [109], [110]. Our framework can be extended to include the feedback from a person at different stages (human-in-the-loop) [111], [112]. For example, by correcting the prediction mistakes of single models or also by injecting task specific knowledge in the process, overriding a specific model. Human behaviours can be collected to update the models and improve the interaction with objects.

III. DATASETS: REVIEW AND LIMITATIONS

In this section, we compare the characteristics of image-based datasets for visual affordance prediction and discuss their similarities and limitations (see Table VII), contrary to previous surveys [26], [28]. Our comparison considers

- the type of environment (indoor or outdoor);
- the camera viewpoint (third person or first person);
- the objects of interest (quantity, diversity depending on the group such as tools or containers, physical properties such as transparency);
- the type of images (real, simulated, mixed-reality);
- the presence of occlusions, caused by other objects, clutter, or hand manipulating the object; and
- the annotations of affordances (quantity, accuracy, procedure, and expertise of the annotators).

These datasets are usually split into two non-overlapping sets: one to train learning-based models (training set) and



Fig. 6: Examples of manual annotation on real data from UMD [9] and IIT-AFF [12]. First row: RGB samples. Second row: manual annotation of affordance prediction (segmentation). Mistakes highlighted with orange rectangles: not every object in the image is annotated (1st column), missing regions (2nd and 4th column), or the annotation of the graspable handle is overlapped with the hand occluding the object (3rd column).

another to evaluate the performance of the models (testing set). Dataset choices, biases in the images, and ambiguities or inaccuracies in the annotations affect and are transferred to the models during the training phase.

Annotations of affordances. Previous works proposing datasets either reuse images already available for other tasks such as object detection or image classification [11], [12], [14], [16], [17], [20], [98], or collect new images specifically for visual affordance prediction [9], [13], [33]. Target affordances in most of these datasets are *manually labelled*. For example, affordance segmentation requires the annotations to label the pixels of the object regions with an affordance class (fine-grained annotation) [9], [12], [16], [33], [114], [115], [116]. However, this procedure is time-consuming and subject to errors (see Fig. 6). Only some objects are annotated, whereas other objects are ignored because of clutter in the scene or their size in the image is small due to the distance of the object from the camera [9], [12]. Annotated segmentation masks can be incomplete (presence of holes) or not accurate close to the object boundaries. To reduce the annotation effort, a *weakly labelling procedure* requires annotators to only label points of interaction and then applies a Gaussian blur operation on the image to expand the point annotation [11], [98]. This procedure was used to annotate two datasets for affordance grounding, OPRA [98] and AGD20K [11]. The blurring operation however may cause the probability of interaction to be non-zero also outside the object boundaries. Because of ambiguities in the boundaries of visual affordances and fine-grained annotations, size of the datasets are often limited to few tens of thousands. To scale the size of the datasets, simulators can generate a large number of synthetic or mixed-reality images with *automatic annotations* while varying the illumination conditions and object models [20], [106], [114], [115]. The task-dependent effort requires the design of the simulated environments, the placement of the objects CAD model in the environment, and the annotation of the affordances (e.g. the manual labelling of the mesh or point cloud

with the affordance category) [13], [114], [115], [120]. The affordance annotations are then easily rendered on the camera frame in the simulator. For example, parts of the object mesh are associated with an affordance category and segmentation masks are obtained by ray-tracing the annotated mesh into the simulated camera frame [20], [114], [115]. For grasping detection, a robotic hand grasping the object can be simulated and the simulating tool can save the image of the object, the coordinates of the grasping attempts, and the bounding boxes as an annotation of the visual affordance [106]. However, images generated with a simulator can still differ from images captured with a real camera (sim-to-real gap), hindering the generalisation of trained models to real images, when deployed on a real robot. As an alternative to the use of simulators, an (*semi-)automatic annotation procedure* can use off-the-shelf models. For example, HANDAL [13] was annotated by using BundleSDF [121], a method that estimates the 6D pose of the objects in each frame of a video and reconstructs their CAD models, followed by manually labelling the handle of the CAD models with the affordance *graspable*, and projecting the annotated CAD model in the camera frame to obtain the annotation mask. When collecting HO3Pairs [5] to perform the synthesis of visual affordances (hand-object interactions) from egocentric images, the problem was to obtain an image containing only the object from a dataset of hand occluded objects (HOI4D [122]). In this case, annotators segmented the hand and used an image in-painter [123] to erase the hand holding the objects and reconstruct the occluded part of the object. Although the in-painting allows to obtain the image without the hand, the reconstruction causes the image quality to degrade with the presence of blurred areas, potentially affecting the performance of the trained method. The dataset RIO [113] was annotated by providing a description of the objects in the scene and an image as a prompt to an LLM, such as ChatGPT [124], that generated the description of the potential interaction between a human and the object. Using ChatGPT to automatically generate annotations requires annotators to setup the appropriate prompt and check for potential mistakes in the labels. Overall, using off-the-shelf methods can speed-up the labelling procedure, potentially scaling the size of annotated datasets [125].

Camera viewpoint: third person view and egocentric view. The majority of datasets for affordance prediction focuses on images from third person perspective [9], [11], [12], [13], [33], [98], [103], [106], [114], [116], [117]. The camera is not mounted on a robot or worn by a person, but it is in a fixed position, capturing objects from a constant distance and in a static scene (see Fig. 7). In some cases [103], [106], [117], the camera is placed in a top-down view to observe the area around a robotic arm. These conditions can limit the models trained on these dataset to generalise to other scenarios, e.g. when the view is not top-down or when there is motion blur in dynamic scenes. The first person perspective (egocentric view) includes additional challenges, such as self-occlusions due to the presence of parts of the human/robot in the collected frames and image blur due to the camera movement [5],



Fig. 7: Images from visual affordance prediction datasets [9], [12], [13], [16], [17], [33], [114], [115], [116]. Datasets have different background conditions, from single object on a plain coloured background to clutter or hand-occluded objects. Images are resized at the same height keeping the aspect ratio for visualization purpose.

[17], [52], [119]. For example, arms are observed from the bottom of an image resulting in objects highly occluded by the hands (e.g. images in FPHA-AFF [17]), or images are affected by blur while people are using kitchen tools and interacting with ingredients in a kitchen environment (e.g. EPIC-Kitchens [5], [119]). Because of these challenges, models trained on egocentric-view datasets might not generalise on images from third person perspective and vice versa.

Occlusions. Most of the datasets [5], [9], [103], [106], [115], [116], [117], [126] focus on one *unoccluded* object placed on a flat surface (e.g. tabletop or floor), and the fixed setup enabled the collection of images with a diverse and large number of object categories and objects instances. For example, UMD [9] and Multi-View [116] collected more than 15 object categories and annotated more than 5 affordance classes, while controlling the environmental conditions for all the images: same illumination and background, and objects placed on a rotating table. However, this simple and controlled setup limits the generalisation of models trained on these datasets to environments with different illumination and backgrounds, or where multiple objects are present in the scene. Only some of the datasets [11], [12], [13], [14], [16], [20] have images with occlusions caused by clutter in the scenes or human hands holding the objects (hand-occlusions). When objects are occluded, only some of their regions are visible, increasing the difficulty of learning affordance prediction because models can predict affordances only from partial information. Hand-occlusion is a main challenge in human-robot collaborations, as erroneous or inaccurate affordance predictions lead to unintended interactions with the object, potentially causing harm to the person (*human safety*) [13], [20], [127].

Objects of interest. In previous works [9], [11], [12], [13], [14], [16], [33], [103], [116], [117], the objects most suitable to accomplish a task are considered as objects of interest, and are associated with the corresponding affordances. Fig. 8 shows the size of the datasets based on the number of object and affordance categories. The majority of these datasets [3], [9], [12], [13], [16], [17], [54], [58] have fewer than 20 object categories and 10 affordance categories. Reported datasets

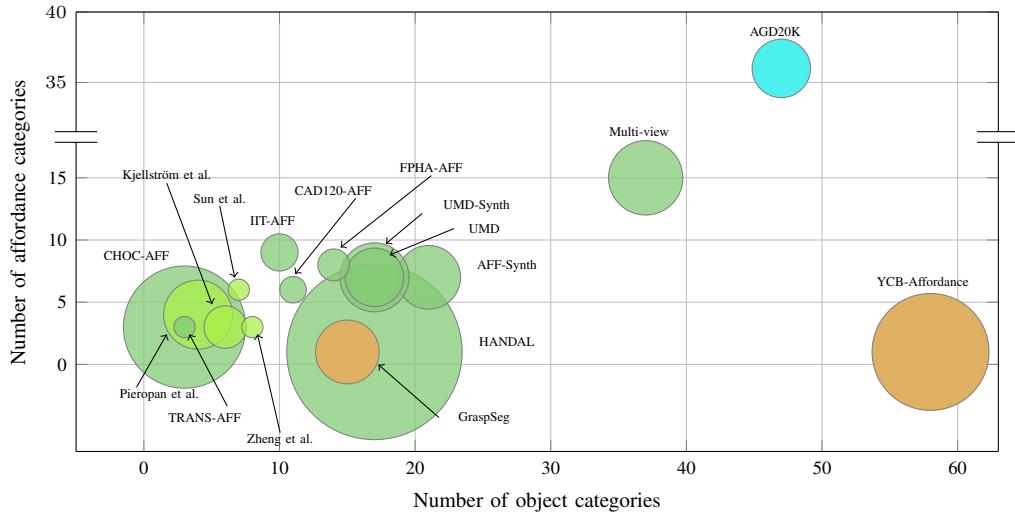


Fig. 8: Visualisation of datasets size based on number of images, number of affordance categories, and number of object categories. Note that we considered only datasets with available information. KEY: — Affordance classification, ■ Affordance detection and segmentation, ■— Affordance grounding, ■ Hand-object pose estimation.

focus on the affordances of tools and containers. Tools are usually opaque and rigid objects used in a kitchen environment (e.g. pan, fork, knife, turner) or for carpentry (e.g. hammer, shovel, saw) and consist of a graspable handle [9], [12], [13]. Compared to tools, perceiving the affordance of containers (e.g. box, cup, glasses) is more challenging, since their properties can change during a manipulation (e.g. the mass due to emptying or filling, or their appearance in case of the transparent material filled with opaque content) [18], [20], [120]. Even if a lot of containers we use in everyday life are transparent, this property is considered only in a few datasets [12], [19], [20], [33].

Diversifying the object categories, degrees of occlusions and object poses in datasets is fundamental to tackle the generalisation problem. The generalisation to diverse conditions is particularly relevant in assistive applications, where the environment is not necessarily controlled and the robot's objective is to help a human in a safe way.

A. Future direction: scaling visual affordance datasets

Available datasets are specific to each affordance redefinition rather than the unified formulation that we provided in this article for visual affordance prediction. Because of this, datasets cannot be easily re-used across different tasks or for the unified case. Moreover, the annotation of object affordances in images and videos is not trivial due to the unclear boundaries of the region on the object, the overlapping of different actions on the same region, and the difficulty of labelling the end-effector pose on objects in the scene. These challenges limit the cross-datasets evaluation of methods and the scalability of datasets for visual affordance, as manual annotations are time-consuming and ambiguous, and requires expensive resources. Because of this, the size of available datasets with curated annotations of visual affordances are

limited to less than 50,000 images. As discussed in Section III and Section VI, datasets are used for both training methods and benchmarking their performance. To scale the number of training data, datasets having similar annotation could be merged, adjusting the annotation, or adapting previous methods to provide weakly or self-supervised annotation (e.g. HANDEL [13]). To collect benchmarks, the combination of different methods could help using in-the-wild images with objects in challenging poses and with different backgrounds.

IV. PERFORMANCE MEASURES

In this section, we present the measures to evaluate the performance of models for visual affordance prediction using our framework as a reference. Given the various components of our generic framework, using a single measure is not sufficient to evaluate the performance of a model. Therefore, we discuss a set of performance measures to provide a fine-grained assessment of the various components. For a specific component or sub-task, more than one performance measure can be used to provide a complete assessment and avoid drawing partial or misleading conclusions (see Table VIII). Our discussion refers to performance measures mostly used by previous works, and highlights the characteristics and limitations of these measures (or alternatives). To evaluate our framework, we consider *per-class accuracy*, *per-class precision*, *per-class recall*, and *per-class F1 score* for functional classification; *per-class precision*, *per-class recall* and *per-class Jaccard index* or *Intersection over Union (IoU)* for functional segmentation; the *interpenetration* between end-effector and object, and the *Analytical grasp score* for end-effector pose estimation; and the *success rate* for experiments with a robot.

Functional classification. The performance measures for assessing functional classification are computed across all N

TABLE VIII: Performance measures to evaluate methods for visual affordance prediction. Highlighted in grey the measures we recommend to evaluate the framework.

Performance measure	Variable	Reference	FUNC	FUNS	EPE	EIS	ROBV
Accuracy	A	Eq. 7	●	○	○	○	○
F1 score	F	Eq. 10	●	○	○	○	○
Precision	P	Eq. 8, Eq. 11	●	●	○	○	○
Recall	R	Eq. 9, Eq. 12	●	●	○	○	○
Jaccard index	J	Eq. 13	○	●	○	○	○
Weighted F-score	F_β^w	[128]	○	●	○	○	○
Kullback-Leibler Divergence	-	[129]	○	●	○	○	○
Similarity	-	[130]	○	●	○	○	○
Normalized Scanpath Saliency	-	[131]	○	●	○	○	○
Analytical grasp score	-	[132]	○	○	●	○	○
Interpenetration volume	-	[14]	○	○	●	○	○
Contact fingers	-	[14]	○	○	●	○	○
Fréchet Inception Distance	FID	[133], Eq. 14	○	○	○	●	○
Contact Recall	-	[5]	○	○	○	●	○
Success rate	-	-	○	○	○	○	●

KEYS – FUNC: functional classification; FUNS: functional segmentation; EPE: end-effector pose estimation; EIS: end-effector interaction synthesis; ROBV: robot validation; ●: considered, ○: not considered.

samples (image, task, end-effector) of a given dataset, each associated with an affordance category a . For each class a , a true positive (TP) is a sample for which the model predicts the class a and the annotation is also a ; a false positive (FP) is a sample for which the model predicts a , but the annotation is a different class; a false negative (FN) is a sample with annotation a , but the model predicts a different class; a true negative (TN) is a sample for which both the model prediction and the annotation are a class different from a . *Per-class accuracy* (A) measures the amount of affordance class predictions matching the annotation:

$$A = \frac{\sum_{n=1}^N TP_n + TN_n}{\sum_{n=1}^N TP_n + TN_n + FP_n + FN_n}. \quad (7)$$

Per-class precision (P) measures the amount of class predictions matching the annotations among all class predictions:

$$P = \frac{\sum_{n=1}^N TP_n}{\sum_{n=1}^N TP_n + FP_n}. \quad (8)$$

Per-class recall (R) measures the amount of affordance class predictions matching the annotation relative to the total number of predictions:

$$R = \frac{\sum_{n=1}^N TP_n}{\sum_{n=1}^N TP_n + FN_n}. \quad (9)$$

Per-class F1 score (F) is the harmonic mean of per-class precision and recall:

$$F = 2 \frac{PR}{P + R}. \quad (10)$$

When evaluating the performance of affordance classification methods, previous works [3], [52], [54], [58] showed confusion matrices and computed accuracy. However, the level of detail of confusion matrices makes difficult to quantitatively compare methods. For datasets with imbalanced classes, accuracy is misleading because a high value can be obtained by predicting always the most frequent class. On the contrary,

using precision, recall, and F1 provides a complementary analysis while considering imbalanced classes, because precision focuses on false positives and recall on false negatives.

Functional segmentation. The performance measures for assessing the functional segmentation of are *per-class precision* (P), *per-class recall* (R) and *per-class Jaccard index* or *Intersection over Union* (IoU). To compute these measures, the output probability maps of the model $[0, 1]^{W \times H}$ are converted into integer values $\{0, 1\}^{W \times H}$ for example using a threshold. As for functional classification, true positives (TP), false positives (FP), and false negatives (FN) are defined for each class a . Given the model prediction \hat{S} and the segmentation annotation of the image S , a true positive is a pixel $y \in I_n$ that is predicted as 1 in \hat{S}_n and the corresponding pixel in S_n is annotated as 1; a false positive is a pixel $y \in I_n$ that is predicted as 1 in \hat{S}_n but annotated as 0 in S_n ; a false negative is a pixel $y \in I_n$ that is predicted as 0 in \hat{S}_n , but the corresponding pixel in S_n is annotated as 1. *Per-class precision* measures the percentage of true positives among all positive predicted pixels,

$$P = \frac{\sum_{n=1}^N \sum_{y \in I_n} TP_n^y}{\sum_{n=1}^N \sum_{y \in I_n} TP_n^y + FP_n^y}. \quad (11)$$

Per-class recall measures the percentage of true positive pixels with respect to the total number of positive pixels,

$$R = \frac{\sum_{n=1}^N \sum_{y \in I_n} TP_n^y}{\sum_{n=1}^N \sum_{y \in I_n} TP_n^y + FN_n^y}. \quad (12)$$

Per-class Jaccard index measures the overlap between predicted and annotated segmentation masks, and quantify how much they are similar in size,

$$J = \frac{\sum_{n=1}^N \sum_{y \in I_n} TP_n^y}{\sum_{n=1}^N \sum_{y \in I_n} TP_n^y + FP_n^y + FN_n^y}. \quad (13)$$

Note that the Jaccard index combines precision and recall, summarising their information in a single value. Therefore, we recommend to report the Jaccard index with complementary performance scores, such as precision and recall, to provide a more comprehensive evaluation and insights.

Most of affordance detection and segmentation works [10], [12], [16], [17], [62], [83], [78], [81], [84], [74], [60] evaluated the performance of methods by using the *weighted F-score* (F_β^w) [128]. Weighted F-score weighs false positives based on the Euclidean distance to the closest annotated pixels. However, F_β^w cannot be computed for the classes that are not in the annotated mask, ignoring part of the prediction mistakes. To compare the predicted probability map with the annotation, affordance grounding works [11], [90], [93], [95] used *Kullback-Leibler divergence* [129], *Similarity* [130], *Normalized Scanpath Saliency* [131]. The *Kullback-Leibler Divergence* has not an upper bound and gives more importance to false negatives compared to false positives. In particular, a false positive results in a *Kullback-Leibler Divergence* value close to 0, whereas a false negative can cause the value to

be high (potentially infinite). *Similarity* combines together the information of false positives and false negatives, assigning a low value to both errors and hence resulting in an ambiguous interpretation. *Normalized Scanpath Saliency* considers the prediction values around a neighbourhood of the annotated points. This measure can lead to misleading insights, since the false positives outside the annotation neighbourhood are discarded.

End-effector pose estimation and synthesis. A model can predict an end-effector pose that is plausible and allows a robot to complete the task even if the predicted pose is not annotated. This aspect makes the evaluation of estimated and synthesised end-effector poses challenging. We therefore recommend using *interpenetration* and *analytical grasp score* to evaluate the estimated pose, and *Fréchet Inception Distance* to evaluate the synthesised pose.

The *interpenetration* is the volume in common between object and end-effector voxels representation (the lower the better) [14]. Note that this performance measure does not consider if the pose of the hand is wrong. The measure cannot be computed if the datasets lacks the annotation of the object pose or the annotation of the end-effector pose. *Analytical grasp score* [132] computes an approximation of the minimum force to be applied to break the grasp stability by solving a quadratic program. The minimum force corresponds to the smallest Euclidean distance from the origin to any point inside the convex hull composed by all feasible forces and torques combinations. To evaluate the pose of the end-effector (hand), Corona et al. [14] also used the *average number of contact fingers*: the higher the number of fingers in contact, the stronger the grasp. This measure, however, can penalise actions or objects for which the number of contact fingers is low (e.g. when grasping a glass from the stem). *Fréchet Inception Distance (FID)* [133] quantifies the similarity between two Gaussian distributions, one fitted on the synthesised images $\hat{G} \sim (\hat{\mu}, \hat{C})$ (where μ is the mean and C the covariance) and the other on the testing set images $G \sim (\mu, C)$ (or ground truth). In particular, the two Gaussian distributions are fitted on the feature representations using the Inception network [134]. *FID* is computed as:

$$FID = \|\mu - \hat{\mu}\|_2^2 + Tr(C + \hat{C} - 2(C\hat{C})^{\frac{1}{2}}), \quad (14)$$

where Tr is the trace operator (i.e. the sum of the diagonal elements of a matrix). The first term, $\|\mu - \hat{\mu}\|_2^2$, measures the squared difference between the means of the real and generated distributions. A smaller difference indicates that the generated and real images have similar overall features. The second term, $Tr(C + \hat{C} - 2(C\hat{C})^{\frac{1}{2}})$, compares the covariances of the real and generated distributions (diversity). A low *FID* score implies high similarity between the generated images distribution and the testing ones. A high *FID* score suggests that the distribution of the generated images differs from the distribution of the testing images, either in terms of overall features (mean) or diversity of features (covariance). To evaluate AffordanceDiffusion, Ye et al. [5] also compute *contact recall* that is the amount of generated hands classified

as “in-contact” with the object in the image by an off-the-shelf method [135]. However, in case of unseen objects or unseen conditions (illumination, colour of the background), the method could misclassify whether the hands are in contact or not, leading to a mistake in the computation of *contact recall*.

Overall evaluation. If a robot is available, the whole framework can be tested in real conditions to assess model performance and identify limitations. In this case, the success rate can be used to measure the performance of the system [10], [83], [81] Reproducing experiments based on success rate is very difficult for some tasks and requires a rigorous protocol. For example, the setup should include information on the object instances, robot model, software versions, and relative poses between object and robot for the task of object lifting. The evaluation should consider separately if the grasping and the lifting are successful, also waiting a fixed amount of time to check if the object falls. When the task is part of other benchmarks [18], [136], using the available performance measures enriches the evaluation.

V. PHYSICALLY-BASED FRAMEWORK FOR ROBOT APPLICATIONS

In this section, we present a generic vision-based framework that considers the connection between the affordances and the object properties for robotic manipulation tasks [8]. The framework uses mass estimation as an example of object property. We describe the relationship between affordance and mass and how the mass influences the end-effector interaction with the object. We then discuss the results of previous methods that estimate the mass of a manipulated container regardless of the content.

A. Affordance as a function of the object physical properties

To perform the interaction with an object, the robot needs to predict the end-effector pose based on the local geometry of both the end-effector and the object. This requirement for common manipulation tasks, however, does not consider all the physical properties of an object and therefore the action can be successful only in limited conditions. Geometry alone might be insufficient for a robot to estimate a physically plausible grasp pose to complete the task. Objects can vary in their mass [137], [138] because of their material or, in other scenarios, because of a manipulation that can affect the mass. For example, household or kitchen-like objects, such as food boxes, drinking cups and glasses, bowls, bottles, vases, or cleaning products, can be filled with other solid or liquid contents. These objects require the robot to predict a grasp pose that should consider potential unwanted behaviours during the manipulation. For example, the predicted grasp pose should avoid the robot spilling the liquid out of a drinking glass. In a collaborative scenarios, the predicted pose is further constrained by the interaction that another robot or person is performing with the object. Therefore, the range of candidate poses that are commonly estimated in static scenarios (common objects placed on a table top) are not all feasible when other physical properties are also considered and in human-robot collaborations [137], [138].

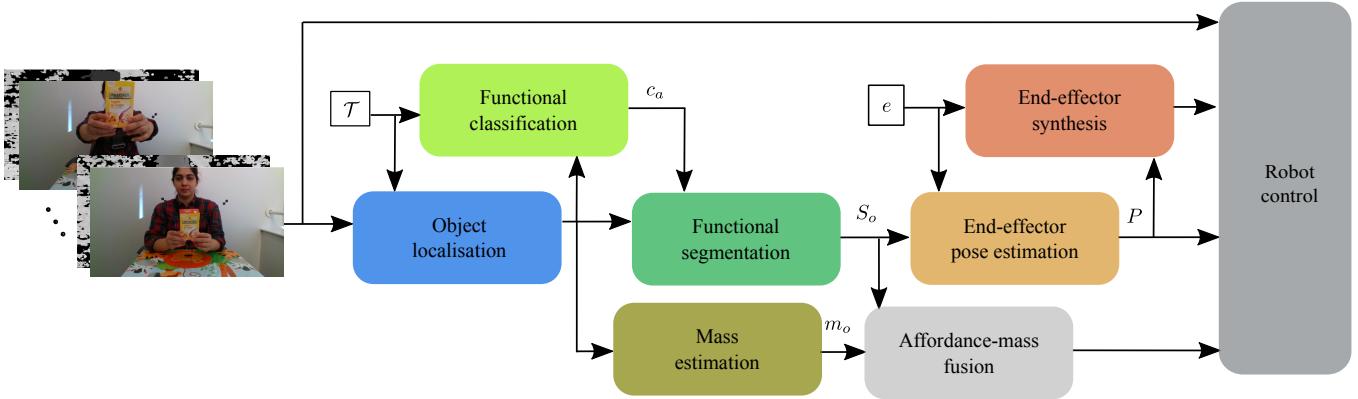


Fig. 9: Visual affordance prediction framework [8]. The vision system predicts the location, the mass and the functional regions of objects in the field of view; based on the predicted information, the control system guides the robotic hand to interact with objects.

Affordance prediction depending on relevant object physical properties can affect both motion planning and force regulation and, consequently, influence the interaction between a robot and the object. The local object properties such as shape, stiffness, and material composition, influence the magnitude of the friction force between the hand and the object. The region where to apply the grasping depends not only on the local property of friction but also on the global property of the object weight distribution. The closer is the high friction area to the center of mass, the more stable the grasping will be and the lower the chances that after closing the fingers and lifting the arm, the object will fall [139].

A possibility to consider the mass of the object in the pipeline is to predict the weight from visual data. In the literature, the mass of a filled container is defined as the sum of the mass of the content (filling mass) and the mass of the empty container [137], [138]. The two sub-problems are tackled separately with some methods focusing on the filling mass estimation [140], [141], [142] and other methods focusing on estimating the mass of the empty container [120], [143], [144]. The challenges in estimating the mass of a container range from locating the object in the scene to accurately predict the physical property using only visual data, due to changes of object appearance during manipulation. Manipulated objects may be occluded by the human hand, be altered in shape if made of deformable materials, or be subject to colour change in case the container is transparent and is filled with opaque content.

Fig. 9 shows a framework that relates the mass and the segmentation of affordance regions [8]. Detection methods localise the objects to crop the image and use the object crops as input to other specialised models. The object of interest is selected based on the specific purpose of the interaction, e.g. taking the object that is held by a human. A model specialised in mass estimation predicts the weight of the object and a model specialised in affordance segmentation predicts the regions of interaction. The two independent predictions can be fused to support the movement planning and the adjustment

of the robotic hand pose and force during the interaction (*Control*). Currently, the affordance segmentation and mass estimation are tackled separately due to the unavailability of a single unifying dataset with the appropriate annotation. We analyze the performance of methods estimating the mass to understand what is the current status and what are the main challenges related to the problem of integrating the mass estimation in an affordance prediction pipeline.

B. Example: estimating the mass of an object

We report the per-category performance of different methods predicting the mass of the container regardless of the content on the private and public testing sets of the CORSMAL Containers Manipulation (CCM) dataset [15], [137], [138]: random sampling (*M1*), average mass (*M2*), custom neural network (*M3*) [143], MobileNetV2 with Coordinate Attention (*M4*) [144], and a Neural Network fusing the image features with object aspect ratio and average depth (*M5*) [120]. *M1* uses a pseudo-random generator that samples the mass prediction from a uniform distribution in the interval [1, 351] using the Mersenne Twister algorithm [145]; *M2* computes the average of the mass annotations; *M3* [143] predicts the container mass using a custom Neural Network that combines unoccluded segmentation mask, obtained by symmetrically restoring the object mask, with geometric information; *M4* uses a MobileNetV2 [146] with Coordinate attention [147] to learn the mass from the object crops (extracted using YOVOv5⁵) through an augmentation strategy to vary the dimension of containers based on the mass and to minimize the mass variance for each object category; *M5* averages the mass from 5 candidate objects, detected by Mask R-CNN and selected based on the lowest average depth respect to the fixed frontal camera, using a linear layer that processes the concatenation of features extracted from the RGB crops, the aspect ratios of the crops and the average depth.

As performance measures, we use the relative absolute error (ϵ) and the mass estimation score (s) between the estimated

⁵<https://github.com/ultralytics/yolov5>

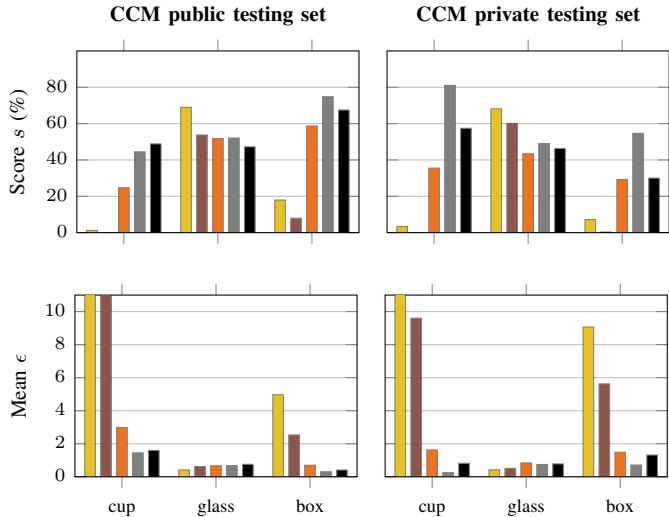


Fig. 10: Comparison of per-container-type mass estimation score (s) and mean of relative absolute error (ϵ) between random sampling (yellow), average mass (brown), custom neural network [143] (orange), MobileNetV2 with Coordinate Attention [144] (black), and neural network fusing image features with object aspect ratio and average depth [120] (grey) on the public and private testing sets of the CORSMAL Containers Manipulation dataset [15], [137]. Note the maximum y-axis value limited to 80 and 10, respectively, for visualization purpose. Random obtains 21.43 and 17.45 as mean ϵ for the class *cup* in the public and private testing sets, respectively, and the average method obtains 12.25 in the private testing set.

mass (\hat{m}^j), and the true mass(m^j) [138]. Given a set of recordings $\{v_j | j = 1, \dots, V\}$, we compute the relative absolute error as:

$$\epsilon(\hat{m}^j, m^j) = \frac{|\hat{m}^j - m^j|}{m^j}, \quad (15)$$

where j is the index of a single recording. $\epsilon \in [0, +\infty)$ has a value over 1 when the estimated mass is greater than the annotated mass, and close to 0 when the estimated mass is lower than the annotated mass. The score $s \in [0, 1]$ is used to average the relative absolute error across all V recordings:

$$s = \frac{1}{V} \sum_{j=1}^V \mathbb{1}_j e^{-\epsilon(\hat{m}^j, m^j)}. \quad (16)$$

The value of the indicator function $\mathbb{1}_j \in \{0, 1\}$ is 0 only when \hat{m} in recording j is not estimated. The score has value 1 when the estimation error is 0 (the predicted mass and annotation are equal).

Fig. 10 compares the mass estimation score and the mean relative absolute error ϵ of the methods, grouped for each container type, on the public and private testing sets of the CCM dataset. On average, models obtain a mass estimation score lower than 56% in the public testing set and 63% in the public testing set (both obtained by $M3$). This results shows that the models cannot generalise to container instances

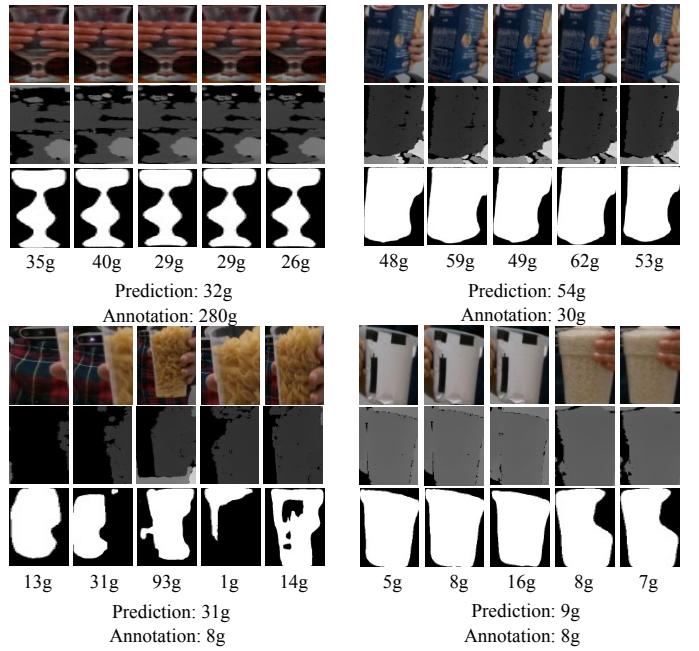


Fig. 11: Qualitative results of mass estimation predicted by the method that averages the mass predicted from 5 candidate objects ($M5$) [120]. The first three rows show the RGB, depth and semantic segmentation crops of the objects having the lowest average depth from fixed frontal camera. The prediction is the average of the mass values predicted for the single crops.

different from the ones in the training set. For both the public and private testing sets, $M1$ and $M2$ achieve higher score values than deep learning based methods for the class *glass*; in particular, $M1$ score exceeds other scores by 15 percentage points (p.p.). This result is due to the similar mass of different drinking glasses in the training and testing sets. For the class *cup*, $M5$ achieves higher score values than $M4$ with a performance increase of 24 p.p. in the public testing set and 20 p.p. in the private testing set. In the private testing set, $M3$ achieves the highest generalization to unseen container instances, outperforming deep learning methods by more than 20 p.p. in the classes *cup* and *box*, and more than 2 p.p. in the class *glass*. Compared to other methods, $M3$ uses the attention mechanisms and variance minimization to ensure consistent predictions of the same container at test time.

Fig. 11 shows $M5$ mass estimations on 4 different recordings of the CCM testing sets. In case of transparent container (1st column and 3rd column), the detection of the container might fail and also the prediction of the object mask, affecting the mass estimation. In some cases, the selection of candidate containers wrongly selects other objects in the scene, e.g. the pitcher used to fill the containers. These results show that predicting the mass from visual data is a challenging task in case of manipulated containers due to the changes of object appearance and the presence of hand occlusions. The generalization to different object instances in case of occlusions is a key concept that affects also the perception of

affordances. An inaccurate estimate of the mass and affordance of an object could lead to a wrong interaction between the end-effector and the object.

C. Future direction: multimodal estimation of object physical properties

Relating affordance prediction and estimation of object physical properties is far from easy. For example, since humans interact differently based on the objects mass [148], the affordance-mass relation can be learned from human demonstration. Humans have different ways of grasping objects depending on the action they want to perform and how the mass influences the action this section, we showed that estimating the object mass from a visual input is a challenging task and methods cannot yet generalise to unseen objects. Moreover, predicting the mass of a container or other physical properties only from images as input might not be sufficient and other modalities, such as language, audio, and haptic, can be included in our proposed framework [137], [138], [141], [143], [149]. Multimodal models have shown to better generalise to novel and different object categories [44] in tasks such as open-vocabulary object detection [150], [151], [152] and segmentation [153], [154], [155]. For visual affordance, we reviewed and discussed vision-language models addressing mainly affordance grounding and task-driven object detection. The information predicted by these models is coarse but can be used to guide the interaction between hand and object. Audio could complement the visual modality when the appearance of the object is not reliable, e.g. an opaque container whose content is not visible. Haptic could provide a feedback on the local geometry of the object and the force that the end-effector applies on the object to refine the interaction phase [139], [156].

VI. AFFORDANCE PREDICTION REPRODUCIBILITY

This section discusses the evaluation of affordance predictions and focuses on the reproducibility issues of current benchmarks, resulting in unfair and inconsistent comparisons. Reproducibility is the principle of obtaining the same results given the same conditions (i.e., data, training and testing setups, and trained model) [157], [158]. For models based on neural networks, such as CNN- and Transformer-based models discussed in this article, we also consider the reproducibility of their architecture and learned weights (or model parameters). Reproducibility allows fair comparisons across methods and helps building upon previous works while understanding their limitations. We then highlight open science practices for fair benchmarking.

A. Reproducibility challenges

Reproducibility challenges (RCs) in different redefinitions of visual affordance prediction include [4]:

- 1) data availability for benchmarking (RC1);
- 2) availability of a method's implementation (RC2);
- 3) availability of the trained model (RC3);
- 4) details of the experimental setups (RC4); and

- 5) details of the performance measures used for the evaluation (RC5).

In visual affordance prediction, no dataset is collected exclusively for benchmarking methods under specific conditions, such as illumination, clutter, or hand-occlusion (RC1). The majority of previous works [9], [13], [20], [62], [116] trains methods on a training split of data and compares their performance using the testing split of one or more datasets. Cross-dataset evaluations are mostly avoided due to partial overlapping of affordance classes or of object categories, across the selected datasets [4]. As a consequence, researchers train multiple version of the same model, adapted to the classes of a specific dataset. For example, datasets such as UMD [9], IIT-AFF [12], and Multi-View [126] share some of the object and affordance classes, but labelled with different conventions. For example in UMD the class *graspable* is associated to the class number “1”, while in IIT-AFF to the class number “5”. Additional documentation, such as metadata, help researchers train or evaluate methods only on common categories by re-ordering them. Moreover, relying only on a single benchmark can lead to limited and not generalisable considerations on model rankings. For example, images in UMD and Multi-View are collected in a laboratory environment with static conditions, such as a fixed camera oriented towards a table where an object is placed (camera-object distance is almost always the same) [9], [116]. On the contrary, in real scenarios the camera might be closer or farther from the object of interest compared to the training setting, hence the performance on the benchmark might not reflect the performance on an actual robot.

The lack of publicly available implementation of methods (RC2) [83], [78], [84], [74], the lack of publicly available of trained models (RC3) [12], [62], [83], [78], [81], [84], [74], and the lack of details of experimental setups (RC4) [10], [62], [83], [78], [84], [74] can challenge researchers in reproducing previous works for comparative evaluations. The release of the model trained weights, of the method and inference pipeline implementation, is a crucial aspect for reproducibility, especially for deep-learning based models, allowing other researchers to test the model on their own data without the need for re-training. The availability of model weights is particularly important when researchers need a comparison, as re-training the model can be too time- and resource-consuming. In case a new dataset is proposed and a previous method needs re-training, only the method implementation is sufficient, while the trained weights are not necessary. The re-implementation of methods and setup can often be time-consuming and prone to errors, and not always resulting in the expected outcome (i.e., results are not replicable or findings are not reproducible). To avoid this issue and often to save time, researchers choose to report the results from previous works as they are [83], [78], [84], [74]. This practice can result in an unfair comparison if the experimental conditions are not the same, leading to misleading findings and conclusions.

Using the same *experimental setup* to train and test affordance models allows a fair comparison whose analysis

TABLE IX: Comparison of training/testing setups used by different methods for affordance detection and segmentation on the UMD dataset [9] along with their performance [28]. We selected the backbone providing the highest F_β^w . Due to the setup inconsistencies, the straight comparison among models performance is unfair.

Training setup	Resolution	Data augmentation				Image resize procedure		F_β^w
		Flipping	Scaling	Rotating	Jittering	Training set	Testing set	
AffordanceNet [10]	1000 × 600	○	○	○	○	unknown	unknown	79.90
CNN [62]	320 × 240	○	○	○	○	centre-crop	sliding window	76.60
DRNAtt [78]	320 × 240	○	○	○	○	centre-crop	unknown	94.10
RANet [84]	224 × 224	○	○	○	○	centre-crop	unknown	86.13
GSE [74]	400 × 400	●	●	○	○	crop	unknown	85.50
BPN [83]	1000 × 600	●	●	●	●	unknown	unknown	86.21

KEYS – ●: considered, ○: not considered, Jittering: colour jittering

can better validate the technical contributions proposed by a novel work. When releasing the training and testing code is not possible, reporting all the training and testing details to reproduce a setup becomes fundamental, enabling other researchers to reimplement the method and correctly compare their solution. The experimental setup details include training hyper-parameter values, the chosen data splits, image pre-processing (normalisation and cropping procedures) and post-processing. The lack of details of the experimental setup causes methods for affordance detection and segmentation to be often not reproducible [10], [62], [83], [78], [84], [74]. For example, AffordanceNet and BPN do not include the detail of the input image resize during training and testing phases [10], [83], whereas DRNAtt, RANet, and GSE do not include these details during the testing phase. Other details often omitted are the parameters of the optimizers used during training [78], [90], [93], [94]. Apicella et al.’s work [4] showed that these lack of details in the experimental setup led to unfair and inconsistent comparisons.

Previous works evaluate the performance of different methods using scores or metrics that quantify the discrepancy between predictions and annotations. Describing a performance measure help other researchers understand if the experiment validates their claim or if a different measure should be chosen. Providing the mathematical formulation of the scores helps disambiguate similar meaning but different implementations, especially when a public evaluation toolkit is not used or referred to. For example, mean Intersection over Union (IoU) can be the average of all the IoUs between prediction and annotation, or the IoU considering the full set of predictions and annotations. Previous works evaluated a few methods with different performance measures, making comparison and ranking not possible. For example, AdaptiveNet [159] and STRAP [86] compared their performance with baselines on CAD120-AFF using IoU as a performance measure. However, these works excluded the comparison with other available models that were mostly tested on UMD but using F_β^w as a performance measure.

Affordance detection and segmentation methods are difficult to reproduce due to missing implementation and lack of setups details [10], [62], [83], [78], [84], [74]. We report the training and testing setups of affordance detection and segmentation methods on the UMD dataset in Table IX, along with the

average performance from Chen et al.’s survey [28]. Despite being trained and tested on the same dataset, models’ performance is not directly comparable due to inconsistencies in the setups such as the image resize procedure (image cropping or input resolution) and augmentation procedure during training.

Inconsistencies can also be present previous method tackling a similar problem are adapted into a baseline to compare with. For example, because of the missing annotation of the object pose in the training set, AffordanceDiffusion [5] is compared with the coarse hand prediction of GanHand [14]. Despite providing implementation details as supplementary material, the results suggest that the whole GanHand was used for comparison. However, since a part of the architecture and of the training procedure is missing, this adapted version of GanHand is only a proxy to the actual (unknown) performance of GanHand.

The redefinition of the visual affordance problem formulation (see Section II) can also result in experimental validations that ignore datasets and benchmarks of partially overlapping subtasks. For example, works on affordance grounding [11], [90], [93], [95] do not compare the performance of their proposed methods with that of methods designed specifically for affordance segmentation [62], [78], [84], [74], even if the problem formulation is similar [11], [62]. Methods for affordance segmentation output an affordance mask (binary mask) for each action in a predefined set of classes, whereas methods for affordance grounding output a confidence map describing where an action known a priori can take place in the image. Despite these differences, comparing methods for both affordance grounding and affordance detection and segmentation can explain if using action as input (affordance grounding) to a model provides any advantage.

B. In support of reproducibility: Affordance Sheets

To promote reproducibility in affordance prediction, we propose the Affordance Sheet, an organised collection of good practices that can facilitate fair comparisons and the development of new solutions (see Table X). Model cards [30] were previously introduced to improve the transparency of methods and to raise awareness about limitations of methods, by describing the characteristics of the method, the details of the experimental setup, the applications or conditions leading to underperformance. Our Affordance Sheet includes Model

TABLE X: **Affordance Sheet**. A tool inspired by Model Cards [30] to favour transparency and reproducibility of works for visual affordance predictions conditioned on robotic tasks.

MODEL NAME						
Affordance task addressed	OBJL ○	FUNC	FUNS	EPE	EIS	
Datasets (RC1)	<i>Record link</i> (where the datasets/benchmarks are stored)* <i>Licence:</i> add licence here					
Proposed method (RC2, RC3)	<i>Record link</i> (where the weights of the model are stored)* <i>Code link</i> (where model architecture and scripts are stored) <i>Model card</i> [30]: ○ <i>Licence:</i> add licence here					
Experimental setup (RC4)	<i>Data splits:</i> add splits criteria here <i>Hyperparameters:</i> add hyperparameters list here <i>Resize procedure:</i> add resize procedure here					
Performance measures (RC5)	<i>Description:</i> add measures description here <i>Formulation:</i> add math formulation here <i>Limitations:</i> add measure limitations here					
Robot validation	<i>Robot model:</i> add robot model name here <i>End-effector:</i> add end-effector here <i>Experiment description:</i> add experiment description here					

Legend: OBJL: object localisation; FUNC: functional classification; FUNS: functional segmentation; EPE: end-effector pose estimation; EIS: end-effector interaction synthesis; RC: reproducibility challenge.
Notes: *data and weights of the trained model are recommended to be placed in a repository that favours long-term persistence and accessibility.

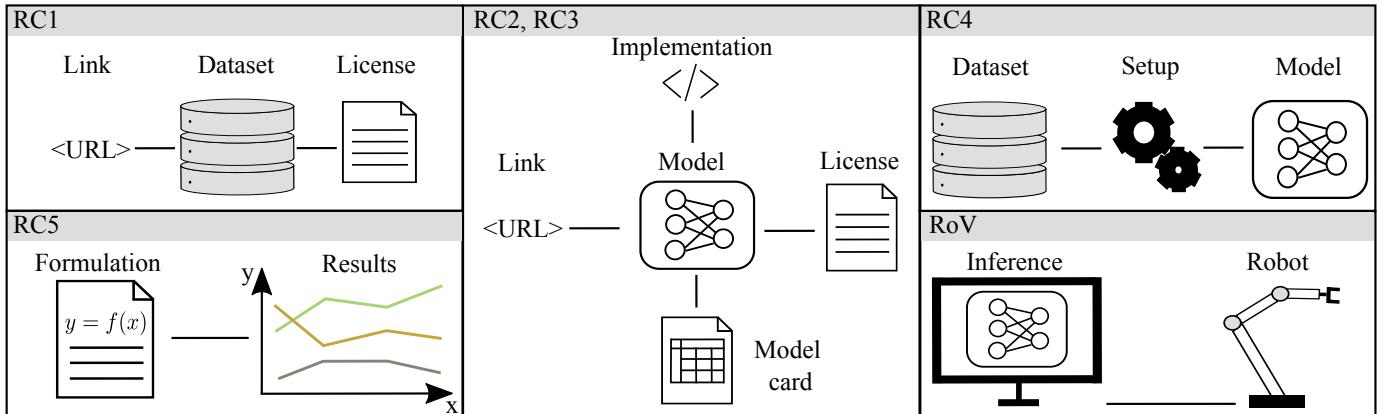


Fig. 12: Visualization of the main challenges tackled by Affordance Sheet sections. KEYS – RC: reproducibility challenge, RoV: robot validation

Cards and adds sections that complement the released information. Fig. 12 shows how each section addresses each reproducibility challenge.

The first section requires to identify which problems the affordance model tackles, helping researchers and users understand what are the competing methods and assess the performance of solutions under the same inputs and conditions. This section also implicitly highlights the considered parts of the overall framework (see Section II-B). When proposing a new problem partially overlapping with another one, previous models can be used or adapted to validate the

method. For example, selecting the channel of an affordance segmentation output based on the action considered by the affordance grounding method enables the comparison between methods for affordance segmentation and methods for affordance grounding. To compare the grounding and segmentation outputs, the grounding confidence map can be converted to a binary mask via thresholding; alternatively, the segmentation map can be converted to a confidence map by taking the segmentation output before the argmax operation or by using Gaussian blur.

The second section of the Affordance Sheet describes the

datasets (RC1) used by the proposed solutions, aiming to detail their characteristics, share the link where other researchers can find the data, and the license informing about the potential uses of the data. We recommend next benchmarks to also release a detailed description on how to use and visualize data so that researchers can get acquainted with the format. Moreover, we recommend that future benchmarks evaluate models under different conditions, such as generalization to different object instances, different object categories, different object poses, background, and clutter. Benchmarks of models for tasks different from visual affordance prediction, such as COCO for object detection and instance segmentation [160], and CORSMAL for object mass estimation [137], release only the training and validation sets while keeping a private testing set to not bias the designer of the architecture [137], [160]. The release of the testing set can lead researchers to make changes aimed at improving performance scores rather than formulating a contribution that advances the field.

The third section highlights the model characteristics (RC2, RC3) integrating information in the model card (if available). Providing model cards [30], along with the model’s implementation and trained weights, helps detail the description of models supporting other researchers to build upon. When not available, we encourage the reimplementation and retraining of the models as a contribution for the community (e.g. see a previous work that reimplemented, retrained, and release models for affordance detection and segmentation to cope with the lack of previously public available models [4], [20]). As also recommended for datasets, we also encourage providing a link to the trained model’s weights and a license detailing the allowed uses. Without a license, the code is automatically protected by copyright, hence other researchers can not directly use the method implementation to reproduce results or as baseline, as for some previous works [17], [86], [91], [94], [97].

By providing the details the experimental setup used to train and evaluate methods (RC4), the fourth section of the Affordance Sheet is fundamental to correctly use previous methods and develop a solution under the same conditions. Setup conditions include pre-processing and post-processing information such as data splits, resize procedures and data normalisation, and hyper-parameters choice. The lack of these details can result in models with significantly different parameters, and hence leading to unfair comparisons with previous works.

The fifth section of the Affordance Sheet focuses on the performance measures (RC5), the criteria used to validate and compare methods with previous solutions. Providing a stand-alone toolkit implementing the performance measures ensures the replicability of the results across different works while including new methods. For visual affordance prediction, we recommend evaluating the performance of models using more than one measure as this practice provides a more comprehensive analysis while identifying different aspects and limitations of the models. For example, in affordance segmentation, precision focuses on how many of the predicted

pixels have the correct class and recall emphasizes how many of the annotated pixels are correctly predicted. Therefore, computing more than one score (and avoiding using a single score aggregating multiple performance measures) reduces the risk of drawing misleading conclusions that are based only on partial results.

The last section describes the validation of the method through a robotic setup. In previous works, few of the methods were validated using a robotic platform [10], [12], [83], [92], [114]. Unlike previous sections of the Affordance Sheet, the robot validation depends on the availability of a robot. When a robot experiment can be performed, we recommend reporting the characteristics of the setup, the robot and end-effector models, and the description of the experiment in terms of object and conditions. This transparent reporting allows researchers to assess methods using a common platform.

C. Future direction: benchmarking visual affordance

Reproducibility and advancements in the design of novel solutions has been facilitated by available datasets, benchmarks and competitions in various computer vision tasks, such ImageNet for object recognition [161], COCO for object detection and instance segmentation [160], and BOP for object 6D pose estimation [162]. However, benchmarks for visual affordance predictions, supported by workshops and competitions to favour reproducibility and fair comparisons, are not available. The connection to a robotic task makes benchmarking and reproducibility even more challenging. Nevertheless, solutions based on our generic framework and novel methods can be designed for robotic grasping and manipulation tasks [163], such as in-hand manipulation [164], picking in clutter [165], and human-to-robot object handovers [18], whose benchmarking protocols and competitions are available. Benchmarking protocol specific to visual affordance used in a robotic task can be also designed and included in existing competitions to further promote reproducibility and engagement of both robotic and vision communities.

VII. CONCLUSION

We proposed a unified, comprehensive formulation for visual affordance prediction. Our formulation relates tasks, such as affordance segmentation, affordance detection, and grasping detection, that were defined and tackled independently in the previous literature. Using our new formulation, we link visual affordance prediction with various physical properties of an object and the context in which a robot is operating, such as the specific action to interact with the object with respect to a higher-level task to perform. Additionally, we designed a generic framework relating visual affordance prediction, physical property estimation, and robot actuation. As a case study, we compared the performance of methods estimating the mass of manipulated objects and we discussed the challenges in accurately estimating this property hence relating it to visual affordance for the proposed framework. Although current methods do not yet generalise to unseen objects, the estimation of physical properties can help bridging the gap

between robot sensors and actuators. We also introduced the Affordance Sheet, a tool to support reproducibility and fair comparisons of learning-based models for visual affordance prediction. Finally, we discussed future research directions, datasets and benchmarks.

REFERENCES

- [1] J. J. Gibson and L. Carmichael, *The senses considered as perceptual systems*. Houghton Mifflin Boston, 1966, vol. 2, no. 1.
- [2] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps,” *Int. J. Robot. Res.*, vol. 34, no. 4–5, pp. 705–724, 2015.
- [3] A. Pieropan, C. H. Ek, and H. Kjellström, “Functional object descriptors for human activity modeling,” in *IEEE Int. Conf. Robotics Autom.*, 2013.
- [4] T. Apicella, A. Xompero, P. Gastaldo, and A. Cavallaro, “Segmenting object affordances: Reproducibility and sensitivity to scale,” in *Eur. Conf. Comput. Vis. Workshops*, 2024.
- [5] Y. Ye, X. Li, A. Gupta, S. De Mello, S. Birchfield, J. Song, S. Tulsiani, and S. Liu, “Affordance diffusion: Synthesizing hand-object interactions,” in *Conf. Comput. Vis. Pattern Recognit.*, 2023.
- [6] P. Ardón, M. E. Cabrera, E. Pairet, R. P. Petrick, S. Ramamoorthy, K. S. Lohan, and M. Cakmak, “Affordance-aware handovers with human arm mobility constraints,” *IEEE Robotics Autom. Lett.*, vol. 6, no. 2, pp. 3136–3143, 2021.
- [7] Y. L. Pang, A. Xompero, C. Oh, and A. Cavallaro, “Stereo hand-object reconstruction for human-to-robot handover,” *IEEE Robotics Autom. Lett.*, 2025.
- [8] T. Apicella, “Visual affordance prediction of hand-occluded objects,” PhD thesis, University of Genoa and Queen Mary University of London, March 2024.
- [9] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, “Affordance detection of tool parts from geometric features,” in *IEEE Int. Conf. Robotics Autom.*, 2015.
- [10] T. Do, A. Nguyen, and I. Reid, “AffordanceNet: An end-to-end deep learning approach for object affordance detection,” in *IEEE Int. Conf. Robotics Autom.*, 2018.
- [11] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, “Learning affordance grounding from exocentric images,” in *Conf. Comput. Vis. Pattern Recognit.*, 2022.
- [12] A. Nguyen, D. Kanoulas, D. Caldwell, and N. Tsagarakis, “Object-based affordances detection with convolutional neural networks and dense conditional random fields,” in *IEEE Int. Conf. Intell. Robot Syst.*, 2017.
- [13] A. Guo, B. Wen, J. Yuan, J. Tremblay, S. Tyree, J. Smith, and S. Birchfield, “Handal: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions,” in *IEEE Int. Conf. Intell. Robot Syst.*, 2023.
- [14] E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, and G. Rogez, “Ganhand: Predicting human grasp affordances in multi-object scenes,” in *Conf. Comput. Vis. Pattern Recognit.*, 2020.
- [15] A. Xompero, R. Sanchez-Matilla, R. Mazzon, and A. Cavallaro, “CORSMAL Containers Manipulation,” 2020, (1.0) [Data set]. Queen Mary University of London. <https://doi.org/10.17636/101CORSMAL1>.
- [16] J. Sawatzky, A. Srikantha, and J. Gall, “Weakly supervised affordance detection,” in *Conf. Comput. Vis. Pattern Recognit.*, 2017.
- [17] S. Hussain, S. L. Liu, W. Xu, and C. Lu, “FPHA-Afford: A domain-specific benchmark dataset for occluded object affordance estimation in human-object-robot interaction,” in *IEEE Int. Conf. Image Process.*, 2020.
- [18] R. Sanchez-Matilla, K. Chatzilygeroudis, A. Modas, N. F. Duarte, A. Xompero, P. Frossard, A. Billard, and A. Cavallaro, “Benchmark for human-to-robot handovers of unseen containers with unknown filling,” *IEEE Robotics Autom. Lett.*, vol. 5, no. 2, pp. 1642–1649, 2020.
- [19] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, “One-shot affordance detection,” in *Int. Joint Conf. Artificial Intell.*, 2021.
- [20] T. Apicella, A. Xompero, E. Ragusa, R. Berta, A. Cavallaro, and P. Gastaldo, “Affordance segmentation of hand-occluded containers from exocentric images,” in *IEEE Int. Conf. Comput. Vis. Workshops*, 2023.
- [21] J. Bohg, A. Morales, T. Asfour, and D. Kragic, “Data-driven grasp synthesis—a survey,” *IEEE Trans. Robotics*, vol. 30, no. 2, pp. 289–309, 2013.
- [22] H.-S. Fang, C. Wang, M. Gou, and C. Lu, “GraspNet-1Billion: A large-scale benchmark for general object grasping,” in *Conf. Comput. Vis. Pattern Recognit.*, 2020.
- [23] F.-J. Chu, R. Xu, and P. A. Vela, “Real-world multiobject, multigrasp detection,” *IEEE Robotics Autom. Lett.*, vol. 3, no. 4, pp. 3355–3362, 2018.
- [24] H. Zhang, X. Lan, S. Bai, X. Zhou, Z. Tian, and N. Zheng, “ROI-based robotic grasp detection for object overlapping scenes,” in *IEEE Int. Conf. Intell. Robot Syst.*, 2019.
- [25] S. Kumra and C. Kanan, “Robotic grasp detection using deep convolutional neural networks,” in *IEEE Int. Conf. Intell. Robot Syst.*, 2017.
- [26] M. Hassanin, S. Khan, and M. Tahtali, “Visual affordance and function understanding: A survey,” *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–35, 2021.
- [27] F. Osiurak, Y. Rossetti, and A. Badets, “What is an affordance? 40 years later,” *Neuroscience & Biobehavioral Reviews*, vol. 77, pp. 403–417, 2017.
- [28] D. Chen, D. Kong, J. Li, S. Wang, and B. Yin, “A survey of visual affordance recognition based on deep learning,” *IEEE Trans. Big Data*, pp. 1–20, 2023.
- [29] P. Ardón, É. Pairet, K. S. Lohan, S. Ramamoorthy, and R. Petrick, “Affordances in robotic tasks—a survey,” in *arXiv:2004.07400v1 [cs.RO]*, 2020.
- [30] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru, “Model cards for model reporting,” in *Proc. Conf. Fairness, Accountability, and Transparency*, 2019.
- [31] J. Romero, D. Tzionas, and M. J. Black, “Embodied hands: Modeling and capturing hands and bodies together,” *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–17, 2017.
- [32] A. Mousavian, C. Eppner, and D. Fox, “6-DOF GraspNet: Variational grasp generation for object manipulation,” in *IEEE Int. Conf. Comput. Vis.*, 2019.
- [33] J. Jiang, G. Cao, T. Do, and S. Luo, “A4t: Hierarchical affordance detection for transparent objects depth reconstruction and manipulation,” *IEEE Robotics Autom. Lett.*, vol. 7, no. 4, pp. 9826–9833, 2022.
- [34] F.-J. Chu, R. Xu, L. Seguin, and P. A. Vela, “Toward Affordance Detection and Ranking on Novel Objects for Real-World Robotic Manipulation,” *IEEE Robotics Autom. Lett.*, vol. 4, no. 4, pp. 4070–4077, 2019.
- [35] K. Chaudhary, K. Okada, M. Inaba, and X. Chen, “Predicting part affordances of objects using two-stream fully convolutional network with multimodal inputs,” in *IEEE Int. Conf. Intell. Robot Syst.*, 2018.
- [36] S. Rezapour Lakani, A. Rodríguez-Sánchez, and J. Piater, “Towards affordance detection for robot manipulation using affordance for parts and parts for affordance,” *Auton. Robots*, vol. 43, pp. 1155–1172, 2019.
- [37] P. Rosenberger, A. Cosgun, R. Newbury, J. Kwan, V. Ortenzi, P. Corke, and M. Grafinger, “Object-independent human-to-robot handovers using real time robotic vision,” *IEEE Robotics Autom. Lett.*, vol. 6, no. 1, pp. 17–23, 2020.
- [38] W. Yang, C. Paxton, M. Cakmak, and D. Fox, “Human grasp classification for reactive human-to-robot handovers,” in *IEEE Int. Conf. Intell. Robot Syst.*, 2020.
- [39] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, “AnyGrasp: Robust and Efficient Grasp Perception in Spatial and Temporal Domains,” *IEEE Trans. Robotics*, 2023.
- [40] J. Sawatzky, Y. Souris, C. Grund, and J. Gall, “What object should I use?-task driven object detection,” in *Conf. Comput. Vis. Pattern Recognit.*, 2019.
- [41] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” in *Int. Conf. Learning Represent.*, 2015.
- [42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Conf. Comput. Vis. Pattern Recognit.*, 2016.
- [43] H. Chen, W. Huang, Y. Ni, S. Yun, Y. Liu, F. Wen, A. Velasquez, H. Latapie, and M. Imani, “TaskCLIP: Extend large vision-language model for task oriented object detection,” in *arXiv:2403.08108v2 [cs.CV]*, 2024.
- [44] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *Int. Conf. Machine Learning*, 2021.
- [45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words:

- Transformers for image recognition at scale,” *Int. Conf. Learning Represent.*, 2021.
- [46] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” in *CoRR*, 2019.
- [47] H. Chen, Y. Ni, W. Huang, Y. Liu, S. Jeong, F. Wen, N. Bastian, H. Latapie, and M. Imani, “Vltp: Vision-language guided token pruning for task-oriented segmentation,” in *arXiv preprint arXiv:2409.08464*, 2024.
- [48] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” in *IEEE Int. Conf. Comput. Vis.*, 2023.
- [49] P. Li, B. Tian, Y. Shi, X. Chen, H. Zhao, G. Zhou, and Y.-Q. Zhang, “Toist: Task oriented instance segmentation transformer with noun-pronoun distillation,” in *Adv. Neural Inf. Process. Syst.*, 2022.
- [50] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Eur. Conf. Comput. Vis.*, 2020.
- [51] J. Tang, G. Zheng, J. Yu, and S. Yang, “Cotdet: Affordance knowledge prompting for task driven object detection,” in *IEEE Int. Conf. Comput. Vis.*, 2023.
- [52] J. Sun, J. L. Moore, A. Bobick, and J. M. Rehg, “Learning visual object categories for robot affordance prediction,” *Int. J. Robot. Res.*, vol. 29, no. 2-3, pp. 174–197, 2010.
- [53] M. I. Jordan, “Graphical models,” *Statistical Science*, 2004.
- [54] X. Zheng, Z. Zeng, and J. Zhang, “High-level object affordance recognition,” in *Int. Conf. Information and Autom.*, 2018.
- [55] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Adv. Neural Inform. Process. Syst.*, 2015.
- [56] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Int. Conf. Learning Represent.*, 2015.
- [57] C. Cortes, “Support-vector networks,” *Machine Learning*, 1995.
- [58] K. Hedvig, R. Javier, and K. Danica, “Visual object-action recognition: Inferring object affordances from human demonstration,” *Computer Vision and Image Understanding*, vol. 115, no. 1, pp. 81–90, 2011.
- [59] C. Sutton, K. Rohanimanesh, and A. McCallum, “Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data,” in *Int. Conf. Machine Learning*, 2004.
- [60] D. Chen, D. Kong, J. Li, S. Wang, and B. Yin, “Adosmnet: a novel visual affordance detection network with object shape mask guided feature encoders,” *Multimedia Tools and Applications*, pp. 1–25, 2023.
- [61] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Conf. Comput. Vis. Pattern Recognit.*, 2017.
- [62] A. Nguyen, D. Kanoulas, D. Caldwell, and N. Tsagarakis, “Detecting object affordances with convolutional neural networks,” in *IEEE Int. Conf. Intell. Robot Syst.*, 2016.
- [63] V. Badrinarayanan, A. Handa, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” *arXiv:1505.07293v1 [cs.CV]*, 2015.
- [64] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, “FastFCN: Rethinking dilated convolution in the backbone for semantic segmentation,” *arXiv:1903.11816v1 [cs.CV]*, 2019.
- [65] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.
- [66] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2015.
- [67] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *IEEE Int. Conf. Comput. Vis.*, 2017.
- [68] C. N. D. Minh, S. Z. Gilani, S. M. S. Islam, and D. Suter, “Learning affordance segmentation: An investigative study,” in *Digital Image Computing: Techniques and Applications*, 2020.
- [69] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Conf. Comput. Vis. Pattern Recognit.*, 2018.
- [70] L. Mur-Labadia, R. Martinez-Cantin, and J. J. Guerrero, “Bayesian deep learning for affordance segmentation in images,” in *IEEE Int. Conf. Robotics Autom.*, 2023.
- [71] D. Morrison, A. Milan, and E. Antonakos, “Uncertainty-aware instance segmentation using dropout sampling,” in *Robotic Vision Probabilistic Object Detection Challenge (CVPR Workshop)*, 2019.
- [72] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Conf. Comput. Vis. Pattern Recognit.*, 2017.
- [73] H. Caselles-Dupré, M. Garcia-Ortiz, and D. Filliat, “Are standard object segmentation models sufficient for learning affordance segmentation?” in *arXiv:2107.02095v1 [cs.LG]*, 2021.
- [74] Y. Zhang, H. Li, T. Ren, Y. Dou, and Q. Li, “Multi-scale fusion and global semantic encoding for affordance detection,” in *Int. Joint Conf. on Neural Networks*, 2022.
- [75] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in *Conf. Comput. Vis. Pattern Recognit.*, 2019.
- [76] H. Zhang, J. Xue, and K. Dana, “Deep ten: Texture encoding network,” in *Conf. Comput. Vis. Pattern Recognit.*, 2017.
- [77] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, “Resnest: Split-attention networks,” in *Conf. Comput. Vis. Pattern Recognit.*, 2022.
- [78] Q. Gu, J. Su, and L. Yuan, “Visual affordance detection using an efficient attention convolutional neural network,” *Neurocomputing*, vol. 440, pp. 36–44, 2021.
- [79] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, “Dual attention network for scene segmentation,” in *Conf. Comput. Vis. Pattern Recognit.*, 2019.
- [80] F. Yu, V. Koltun, and T. Funkhouser, “Dilated residual networks,” in *Conf. Comput. Vis. Pattern Recognit.*, 2017.
- [81] C. Yin, Q. Zhang, and W. Ren, “A new semantic edge aware network for object affordance detection,” *J. Intelligent & Robotic Systems*, vol. 104, no. 1, pp. 1–16, 2022.
- [82] Y. Hu, Y. Chen, X. Li, and J. Feng, “Dynamic feature fusion for semantic edge detection,” in *Int. Jt. Conf. Artif. Intell.*, 2019.
- [83] C. Yin and Q. Zhang, “Object affordance detection with boundary-preserving network for robotic manipulation tasks,” *Neural. Comput. Appl.*, vol. 34, no. 20, pp. 17963–17980, 2022.
- [84] X. Zhao, Y. Cao, and Y. Kang, “Object affordance detection with relationship-aware network,” *Neural. Comput. Appl.*, vol. 32, no. 18, pp. 14321–14333, 2020.
- [85] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, “Context encoding for semantic segmentation,” in *Conf. Comput. Vis. Pattern Recognit.*, 2018.
- [86] L. Cui, X. Chen, H. Zhao, G. Zhou, and Y. Zhu, “STRAP: Structured Object Affordance Segmentation with Point Supervision,” in *arXiv:2304.08492v1 [cs.CV]*, 2023.
- [87] N. Nauata, H. Hu, G.-T. Zhou, Z. Deng, Z. Liao, and G. Mori, “Structured label inference for visual understanding,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1257–1271, 2019.
- [88] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, “Masked-attention mask transformer for universal image segmentation,” in *Conf. Comput. Vis. Pattern Recognit.*, 2022.
- [89] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov, “On regularized losses for weakly-supervised CNN segmentation,” in *Eur. Conf. Comput. Vis.*, 2018.
- [90] S. Qian, W. Chen, M. Bai, X. Zhou, Z. Tu, and L. E. Li, “AffordanceLLM: Grounding affordance from vision language models,” in *Conf. Comput. Vis. Pattern Recognit. Workshops*, 2024.
- [91] T. Nagarajan, C. Feichtenhofer, and K. Grauman, “Grounded human-object interaction hotspots from video,” in *IEEE Int. Conf. Comput. Vis.*, 2019.
- [92] E. Tong, A. Oipiari, S. Lewis, Z. Zeng, and O. C. Jenkins, “Oval-prompt: Open-vocabulary affordance localization for robot manipulation through llm affordance-grounding,” in *arXiv:2404.11000v2 [cs.RO]*, 2024.
- [93] G. Li, D. Sun, L. Sevilla-Lara, and V. Jampani, “One-shot open affordance learning with foundation models,” in *Conf. Comput. Vis. Pattern Recognit.*, 2024.
- [94] S. Qian and D. F. Fouhey, “Understanding 3D object interaction from a single image,” in *IEEE Int. Conf. Comput. Vis.*, 2023.
- [95] C. Cuttano, G. Rosi, G. Trivigno, and G. Averta, “What does CLIP know about peeling a banana?” in *Conf. Comput. Vis. Pattern Recognit. Workshops*, 2024.
- [96] G. Li, V. Jampani, D. Sun, and L. Sevilla-Lara, “Locate: Localize and transfer object parts for weakly supervised affordance grounding,” in *Conf. Comput. Vis. Pattern Recognit.*, 2023.

- [97] D. Hadjivelichkov, S. Zwane, L. Agapito, M. P. Deisenroth, and D. Kanoulas, “One-shot transfer of affordance regions? affcorrs!” in *Conf. Robot Learning*, 2023.
- [98] K. Fang, T.-L. Wu, D. Yang, S. Savarese, and J. J. Lim, “Demo2vec: Reasoning object affordances from online videos,” in *Conf. Comput. Vis. Pattern Recognit.*, 2018.
- [99] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Conf. Comput. Vis. Pattern Recognit.*, 2016.
- [100] W. Zhai, H. Luo, J. Zhang, Y. Cao, and D. Tao, “One-shot object affordance detection in the wild,” *Int. J. Comput. Vis.*, vol. 130, no. 10, pp. 2472–2500, 2022.
- [101] J. Lundell, E. Corona, T. N. Le, F. Verdoja, P. Weinzaepfel, G. Rogez, F. Moreno-Noguer, and V. Kyriki, “Multi-FinGAN: Generative coarse-to-fine sampling of multi-finger grasps,” in *IEEE Int. Conf. Robotics Autom.*, 2021.
- [102] J. Redmon and A. Angelova, “Real-time grasp detection using convolutional neural networks,” in *IEEE Int. Conf. Robotics Autom.*, 2015.
- [103] U. Asif, J. Tang, and S. Harrer, “GraspNet: An Efficient Convolutional Neural Network for Real-time Grasp Detection for Low-powered Devices,” in *Int. Joint Conf. Artificial Intell.*, 2018.
- [104] S. Ainetter and F. Fraundorfer, “End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from RGB,” in *IEEE Int. Conf. Robotics Autom.*, 2021.
- [105] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Adv. Neural Inf. Process. Syst.*, 2012.
- [106] A. Depierre, E. Dellandréa, and L. Chen, “Jacquard: A large scale dataset for robotic grasp detection,” in *IEEE Int. Conf. Intell. Robot Syst.*
- [107] J. Romero, D. Tzionas, and M. J. Black, “Embodied hands: Modeling and capturing hands and bodies together,” *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–17, 2022.
- [108] Z. Durante, Q. Huang, N. Wake, R. Gong, J. S. Park, B. Sarkar, R. Taori, Y. Noda, D. Terzopoulos, Y. Choi, K. Ikeuchi, H. Vo, L. Fei-Fei, and J. Gao, “Agent AI: Surveying the Horizons of Multimodal Interaction,” 2024, arXiv:2401.03568v2 [cs.AI].
- [109] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” in *Adv. Neural Inf. Process. Syst.*, 2022.
- [110] S. Garg, D. Tsipras, P. S. Liang, and G. Valiant, “What Can Transformers Learn In-Context? A Case Study of Simple Function Classes,” in *Adv. Neural Inf. Process. Syst.*, 2022.
- [111] T. Yu, Y. Yao, H. Zhang, T. He, Y. Han, G. Cui, J. Hu, Z. Liu, H.-T. Zheng, M. Sun *et al.*, “RLHF-V: Towards Trustworthy MLLMs via Behavior Alignment from Fine-Grained Correctional Human Feedback,” in *Conf. Comput. Vis. Pattern Recognit.*, 2024.
- [112] R. Munro, *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Manning, 2021.
- [113] M. Qu, Y. Wu, W. Liu, X. Liang, J. Song, Y. Zhao, and Y. Wei, “Rio: A benchmark for reasoning intention-oriented objects in open environments,” in *Adv. Neural Inf. Process. Syst.*, 2023.
- [114] A. D. Christensen, D. Lehotsky, M. W. Jørgensen, and D. Chrysostomou, “Learning to segment object affordances on synthetic data for task-oriented robotic handovers,” in *Brit. Mach. Vis. Conf.*, 2022.
- [115] F. Chu, R. Xu, and P. Vela, “Learning affordance segmentation for real-world robotic manipulation via synthetic images,” *IEEE Robotics Autom. Lett.*, vol. 4, no. 2, pp. 1140–1147, 2019.
- [116] Z. O. Khalifa and S. A. A. Shah, “Towards visual affordance learning: A benchmark for affordance segmentation and recognition,” in *arXiv:2203.14092v2 [cs.CV]*, 2022.
- [117] Y. Jiang, S. Moseson, and A. Saxena, “Efficient grasping from RGBD images: Learning using a new rectangle representation,” in *IEEE Int. Conf. Robotics Autom.*, 2011.
- [118] M. Suchi, T. Patten, D. Fischinger, and M. Vincze, “EasyLabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets,” in *IEEE Int. Conf. Robotics Autom.*, 2019.
- [119] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, “Scaling egocentric vision: The EPIC-KITCHENS dataset,” in *Eur. Conf. Comput. Vis.*, 2018.
- [120] T. Apicella, G. Slavic, E. Ragusa, P. Gastaldo, and L. Marcenaro, “Container localisation and mass estimation with an RGB-D camera,” in *IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2022.
- [121] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield, “BundleSDF: Neural 6-DoF Tracking and 3D Reconstruction of Unknown Objects,” in *Conf. Comput. Vis. Pattern Recognit.*, 2023.
- [122] Y. Liu, Y. Liu, C. Jiang, K. Lyu, W. Wan, H. Shen, B. Liang, Z. Fu, H. Wang, and L. Yi, “Hoi4D: A 4D egocentric dataset for category-level human-object interaction,” in *Conf. Comput. Vis. Pattern Recognit.*, 2022.
- [123] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, “Glide: Towards photorealistic image generation and editing with text-guided diffusion models,” in *Int. Conf. Machine Learning*, 2022.
- [124] OpenAI, “Chatgpt: Optimizing language models for dialogue,” 2023. [Online]. Available: <https://openai.com/chatgpt>
- [125] X. Deng, Q. Yu, P. Wang, X. Shen, and L.-C. Chen, “COCONut: Modernizing COCO segmentation,” in *Conf. Comput. Vis. Pattern Recognit.*, 2024.
- [126] S. R. Lakani, A. J. Rodríguez-Sánchez, and J. Piater, “Towards Affordance Detection for Robot Manipulation using Affordance for Parts and Parts for Affordance,” *Autonomous Robots*, vol. 43, no. 5, pp. 1155–1172, 2019.
- [127] Y. L. Pang, A. Xompero, C. Oh, and A. Cavallaro, “Towards safe human-to-robot handovers of unknown containers,” in *IEEE Int. Conf. Robot and Human Interactive Comm.*, 2021.
- [128] R. Margolin, L. Zelnik-Manor, and A. Tal, “How to evaluate foreground maps?” in *Conf. Comput. Vis. Pattern Recognit.*, 2014.
- [129] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, “What do different evaluation metrics tell us about saliency models?” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 740–757, 2018.
- [130] M. J. Swain and D. H. Ballard, “Color indexing,” *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.
- [131] R. J. Peters, A. Iyer, L. Itti, and C. Koch, “Components of bottom-up gaze allocation in natural images,” *Vision research*, vol. 45, no. 18, pp. 2397–2416, 2005.
- [132] C. Ferrari and J. Canny, “Planning optimal grasps,” in *IEEE Int. Conf. Robotics Autom.*, 1992.
- [133] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in *Adv. Neural Inf. Process. Syst.*, 2017.
- [134] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Conf. Comput. Vis. Pattern Recognit.*, 2015.
- [135] D. Shan, J. Geng, M. Shu, and D. F. Fouhey, “Understanding human hands in contact at internet scale,” in *Conf. Comput. Vis. Pattern Recognit.*, 2020.
- [136] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, “Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols,” *IEEE Robotics Autom. Magazine*, vol. 22, no. 3, pp. 36–52, 2015.
- [137] A. Xompero, S. Donaher, V. Iashin, F. Palermo, G. Solak, C. Coppola, R. Ishikawa, Y. Nagao, R. Hachiuma, Q. Liu, F. Feng, C. Lan, R. H. M. Chan, G. Christmann, J. Song, G. Neeharika, C. K. T. Reddy, D. Jain, B. U. Rehman, and A. Cavallaro, “The CORSMAL benchmark for the prediction of the properties of containers,” *IEEE Access*, vol. 10, pp. 41 388–41 402, 2022.
- [138] A. Xompero, Y. L. Pang, T. Patten, A. Prabhakar, B. Calli, and A. Cavallaro, “Audio-visual object classification for human-robot collaboration,” in *IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2022.
- [139] Q. Feng, Z. Chen, J. Deng, C. Gao, J. Zhang, and A. Knoll, “Center-of-mass-based robust grasp planning for unknown objects using tactile-visual sensors,” in *IEEE Int. Conf. Robotics Autom.*, 2020.
- [140] G. Christmann and J. Song, “2020 CORSMAL Challenge - Team NTNU-ERCReport,” 2020, https://corsmal.eecs.qmul.ac.uk/resources/challenge/2020.11.30_CORSMAL_NTNU-ERC_Report.pdf.
- [141] V. Iashin, F. Palermo, G. Solak, and C. Coppola, “Top-1 CORSMAL Challenge 2020 Submission: Filling Mass Estimation Using Multi-Modal Observations of Human-Robot Handovers,” in *IEEE Conf. Pattern Recognit. Workshops and Challenges*, 2021.
- [142] Q. Liu, F. Feng, C. Lan, and R. H. M. Chan, “VA2Mass: Towards the Fluid Filling Mass Estimation via Integration of Vision and Audio Learning,” in *IEEE Conf. Pattern Recognit. Workshops and Challenges*, 2021.

- [143] T. Matsubara, S. Otsuki, Y. Wada, H. Matsuo, T. Komatsu, Y. Iioka, K. Sugiura, and H. Saito, "Shared Transformer Encoder with Mask-Based 3D Model Estimation for Container Mass Estimation," in *IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2022.
- [144] H. Wang, C. Zhu, Z. Ma, and C. Oh, "Improving Generalization of Deep Networks for Estimating Physical Properties of Containers and Fillings," in *IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2022.
- [145] M. Matsumoto and T. Nishimura, "Mersenne Twister: a 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator," *ACM Trans. Modeling Comput. and Simulation*, vol. 8, no. 1, pp. 3–30, 1998.
- [146] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Conf. Comput. Vis. Pattern Recognit.*, 2018.
- [147] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Conf. Comput. Vis. Pattern Recognit.*, 2021.
- [148] L. Lastrico, N. F. Duarte, A. Carfi, F. Rea, A. Sciuitti, F. Mastrogiovanni, and J. Santos-Victor, "Expressing and inferring action carefulness in human-to-robot handovers," in *IEEE Int. Conf. Intell. Robot Syst.*, 2023.
- [149] R. Ishikawa, Y. Nagao, R. Hachiuma, and H. Saito, "Audio-Visual Hybrid Approach for Filling Mass Estimation," in *IEEE Conf. Pattern Recognit. Workshops and Challenges*, 2021.
- [150] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "YOLO-World: Real-Time Open-Vocabulary Object Detection," in *Conf. Comput. Vis. Pattern Recognit.*, 2024.
- [151] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection," in *Eur. Conf. Comput. Vis.*, 2024.
- [152] J. Wu, X. Li, S. Xu, H. Yuan, H. Ding, Y. Yang, X. Li, J. Zhang, Y. Tong, X. Jiang, B. Ghanem, and D. Tao, "Towards Open Vocabulary Learning: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 7, pp. 5092–5113, 2024.
- [153] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu, "Open-vocabulary semantic segmentation with mask-adapted CLIP," in *Conf. Comput. Vis. Pattern Recognit.*, 2023.
- [154] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, "Scaling open-vocabulary image segmentation with image-level labels," in *Eur. Conf. Comput. Vis.*, 2022.
- [155] A. Singh, R. Hu, V. Goswami, G. Couairou, W. Galuba, M. Rohrbach, and D. Kiela, "Flava: A foundational language and vision alignment model," in *Conf. Comput. Vis. Pattern Recognit.*, 2022.
- [156] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *IEEE Int. Conf. Robotics Autom.*, 2019.
- [157] A. Desai, M. Abdelhamid, and N. R. Padalkar, "What is Reproducibility in Artificial Intelligence and Machine Learning Research?" 2025.
- [158] J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Larivière, A. Beygelzimer, F. d'Alché Buc, E. Fox, and H. Larochelle, "Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program)," *Journal of machine learning research*, vol. 22, no. 164, pp. 1–20, 2021.
- [159] J. Sawatzky and J. Gall, "Adaptive binarization for weakly supervised affordance segmentation," in *IEEE Int. Conf. Comput. Vis. Workshops*, 2017.
- [160] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Eur. Conf. Comput. Vis.*, 2014.
- [161] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [162] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. Glent Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, "BOP: Benchmark for 6D object pose estimation," in *Eur. Conf. Comput. Vis.*, 2018.
- [163] Y. Sun, B. Calli, K. Kimble, F. wyffels, V. De Gusseme, K. Hang, S. D'Avella, A. Xompero, A. Cavallaro, M. A. Roa, J. Avendano, and A. Mavrommatis, "Robotic Grasping and Manipulation Competition at the 2024 IEEE/RAS International Conference on Robotics and Automation," *IEEE Robotics Autom. Magazine*, vol. 31, no. 4, pp. 174–185, Dec 2024.
- [164] S. Cruciani, B. Sundaralingam, K. Hang, V. Kumar, T. Hermans, and D. Kragic, "Benchmarking in-hand manipulation," *IEEE Robotics Autom. Lett.*, vol. 5, no. 2, pp. 588–595, 2020.
- [165] S. D'Avella, M. Bianchi, A. M. Sundaram, C. A. Avizzano, M. A. Roa, and P. Tripicchio, "The Cluttered Environment Picking Benchmark (CEPB) for Advanced Warehouse Automation: Evaluating the Perception, Planning, Control, and Grasping of Manipulation Systems," *IEEE Robotics Autom. Magazine*, vol. 31, no. 4, pp. 45–58, 2024.