IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Incorporating Object Intrinsic Features within Deep Grasp Affordance Prediction

Matthew Veres, Ian Cabral, Medhat Moussa

Abstract— **Robotic grasping systems often rely on visual observations to drive the grasping process, where the robot must be able to detect and localize an object, extract features relevant to the task, and then combine this information to plan a manipulation strategy. But what happens when some of the most impactful features are not observed by the robot? Without context on an objects center-of-mass, for example, a robot may make assumptions such as uniform density that do not hold, and which may in turn guide the robot into perceiving a sub-optimal set of grasping configurations. In this work, we examine how having prior knowledge of an object's intrinsic properties influences the task of dense grasp affordance prediction. We investigate a simple, constrained grasping task where object properties heavily regulate the space of successful grasps, and further evaluate how learning is affected when generalizing across unseen weight configurations and unseen object shapes.**

I. Introduction

Grasping within unstructured environments is a major challenge for many robotic systems, largely due to the inherent complexity of the task. There are many factors involved in this process: object characteristics (both intrinsic and extrinsic), the gripper used, and the overall environment all play a role in determining whether a grasping operation is successful or not. As such, while traditional analytical methods have been explored within grasping for over 30 years, there has been a significant push towards *learning* approaches — a shift from explicitly calculating grasp quality metrics, to learning how to grasp based on real or simulated grasping experiments.

Recent deep-learning based methods have demonstrated particularly strong success generalizing across the visual appearance of objects. These models have been applied to many traditional areas of grasping, such as: object detection and recognition (e.g. [1], [2]), pose estimation [3], and in generating grasps for both rigid [4], [5], [6], [7] and deformable [8] objects. Yet static, visual observations can only identify some of the object's characteristics; in this case, mostly extrinsic ones such as shape and texture. Planning grasps without understanding the full scope of the object can lead to strategies that are neither optimal, nor actually applicable given the objects true underlying context.

Intrinsic object properties such as center of mass (CoM), surface friction, and rigidity play a significant component in many grasping applications. Suction grippers, for example, are used in a wide variety of scenarios due to their ability to grasps objects in the presence of clutter and with only one contact surface (e.g. [7]). Yet suction grasps are highly sensitive to the relative placement of the contact points with respect to the object's CoM. In some instances, this CoM may be dynamic if the object contains fluid, is non-rigid, or is even considered as part of a larger system with multiple components. Without accounting for these properties within the grasping process, there is a much higher likelihood of a grasp failing. This holds particularly well for objects that have very similar shapes, but different internal material density and structure.

Over the years, different models have been proposed that not only seek to identify these intrinsic properties, but also to investigate how they can be used within a grasping context. Works such Standley et al. [9] leverage deep learning to estimate an object's mass through RGB images and size information. Kannabiran, Essa, and Liu [10] learn control policies for estimating an object's mass distribution. In [11], an object's visible appearance is used to learn different material properties. Other works [12] have also been proposed that bootstraps the prediction of object properties, by modeling real-world scenarios within simulated environments, and observing the objects behaviour. Local surface characteristics have been used to learn different grasp quality metrics [13], and in [14] capture context about a grasped object for planning adjustments to a grasp. In other instances, force and torque data has been used to help guide a humanoid robot to selecting grasps close to an objects CoM [15]. Contact feedback has also been used within reinforcement learning to find control policies for stably grasping objects [16].

Finally, the overall task is also a critical factor in any grasping operation. Grasp affordances reflect the myriad of ways that a robot can grasp an object and complete a task [17], [18], [19], [7]. Recent work in learning grasp affordances has also looked at tasks that require a more sophisticated process, such as object throwing [20]. Important to highlight about [20] is the notion that a robot can perceive grasp affordances differently, depending on the requirements of a task and the supervision a network is given. Researchers have also studied how an object's CoM affects *human* grasp selection, given objects with different visible materials [21].

In this work, we are interested in developing a general learning framework where the robot can not only predict successful grasping configurations given an object image and surrounding environment, but also learn about the object's intrinsic properties, and how they may influence the space of grasp affordances. Without this knowledge, it is possible

This work was supported by the Natural Sciences and Engineering Research Council of Canada

Authors are with the School of Engineering, University of Guelph, 50 Stone Road East. Guelph, Ontario, Canada. {mveres, icabral, mmoussa}@uoguelph.ca

that a robot may perceive affordances that do not accurately reflect the true set of action possibilities, and which in turn may result in a sub-optimal task performance.

A. Paper Contributions

Our work is focused on learning a complete object model, that includes both intrinsic and extrinsic features. In [22], Veres et al. proposed a deep learning framework that learns an object motor image, which links a visual object image with grasp configurations. In this work, we extend this concept by incorporating information about an object's *intrinsic* properties; namely, it's center of mass. Our contributions are:

- We propose a learning framework that incorporates CoM *implicitly* into grasp affordance prediction through force and torque readings from a robot's wrist. Together, passive and active observations are linked to form a grasp motor image of the object.
- We show how grasp affordances can be learned given only a few prior experiments, by adapting the framework of [23] to the problem of robotic grasping – using prior experiments as "support" examples for conditioning affordance prediction.
- We demonstrate the proposed framework by investigating a constrained grasping task where an object's density and center-of-mass heavily regulate whether a grasp will be successful or not. Within this setting, a robot must physically interact with the object in order to understand how the object will behave once grasped. All investigation is performed in real grasping experiments using a suction gripper and wrist-mounted force sensor. Our dataset can be found at: `https://doi.org/10.5683/SP2/YCBUSR`.

II. Methodology

Self-supervised grasping is an appealing approach for learning how to grasp, largely due to the autonomy of the process and the lack of human bias in determining whether a grasp will be successful or not. Yet one of the challenges of this approach is that labels that are collected for a single attempt are inherently sparse by nature: A robot typically observes information relative to specific environmental conditions, and grasps areas of an object that may be highly localized. Below, we outline how different sources of local information can be fused to predict global suction affordances following an explore-then-act paradigm.

A. Problem Setup

We wish to predict a dense, grasp affordance map for a suction cup gripper that has a limited maximum suction force. The objects we are interested in are simple, planar objects that are visible to an overhead camera. Each object belongs to one of nine object classes, and contains some unknown mass distribution. Because the object shapes and appearances are so similar (by design), and because the mass distribution cannot be sensed without physically interacting with the object, this problem represents a real-world scenario that "just because two object's may look the same, does not mean they are the same". When combined with the limited grasping force, the robot must carefully reason about the object in order to predict successful grasp candidates.

Similar to [7], we refer to a grasp affordance map $\mathcal{G}$ as a pixel-wise probability; that performing a top-down grasp at pixel locations within in some query image I_q will lead to a successful outcome. Here, we focus only on pixel locations where the object is present. For any given object, the robot is tasked with predicting $\mathcal{G}$ based on a small set of previously-collected attempts grasping the same object, i.e. $p(\mathcal{G}|I_q, e_1, \ldots, e_k)$. Each prior experience e_i contains an image I_s of the object, the location where the object was grasped p (in pixel coordinates), the experienced wrist forces f and torques t in the local sensor frame, and a boolean o that denotes whether the explored grasp was successful or not. Thus, $e_i = \{I_{s,i}, p_i, f_i, t_i, o_i\}$.

B. Approach

In order to predict $\mathcal{G}$, our approach is to train a neural network to learn how intrinsic and extrinsic features of prior grasping attempts can be used to recognize other grasping candidates. To accomplish this, we extend the *guided networks* approach of [23] to the problem of robotic grasping with sensory feedback. Guided networks, at their core, perform tasks such as semantic segmentation by learning how to predict pixel-wise class association, conditioned on *average* representations of both the target class ("positive" support examples) and of other classes ("negative" support examples). Importantly, these representations only require a few sparse labels to be constructed. This is an important characteristic for robotic grasping, where it might be difficult to obtain large scale data from actual experiments.

In this work, our support examples refer to the set of prior grasping experiences e defined above — where the target class represents either a successful or a failed grasp, and where the single pixel-location of the grasp attempt is known. The guided network therefore is tasked with extracting meaningful context from these support examples, such that it can predict $\mathcal{G}$ for a new image observation of the same object. To include additional context about the support examples, we extend the architecture of [23] with an additional network branch that learns to encode sensory feedback previously captured during each grasp. Figure 1 highlights our model, with the sensory branch in yellow.

C. Model Description

We discuss our model setup with respect to Figure 1 below. Training, hyperparameters, and discussion on how support examples are chosen is discussed further in Section IV-B.

Visual Query Encoder: Given a query image x of an object, a convolutional network (CNN) first encodes x through several layers of convolution & pooling operations. The output of this branch is a 3D grid of size $11 \times 11 \times 64$, which represents an $8\times$ down sampling of the query image, with 64 learned features at each location.

Visual Exploration Encoder: In parallel, a second CNN encodes the support *images* that were collected during each

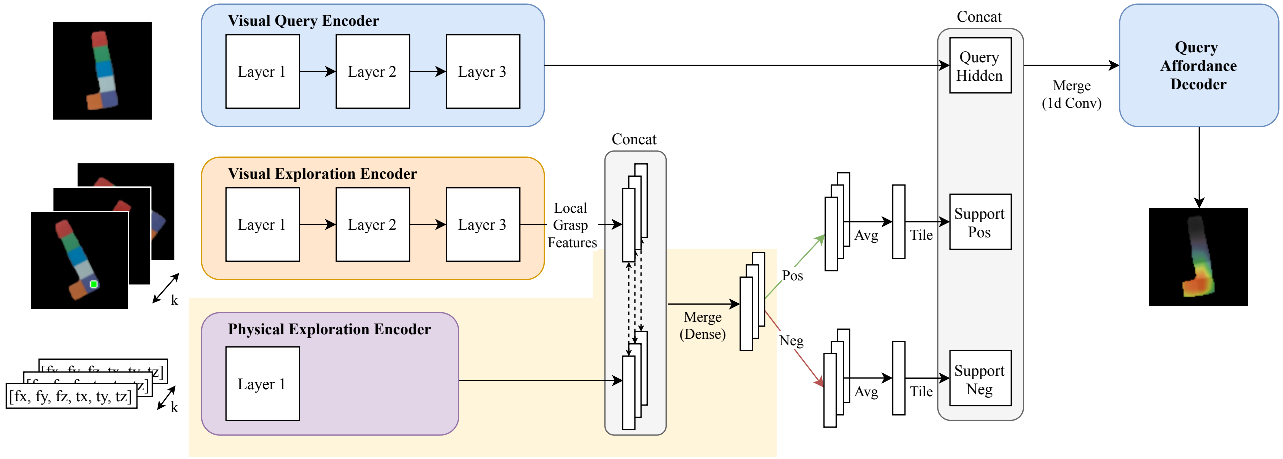


Fig. 1: Affordance prediction model that leverages both visual information and sensory feedback. A CNN model (Visual Query Encoder) first encodes an RGB object image through several convolution + pooling operations. The network is also shown k previous grasping attempts of the exact same object, in the form of RGB images & force / torque readings, which are encoded by another CNN (Visual Exploration Encoder) and MLP (Physical Exploration Encoder) respectively. The network produces an affordance map with the same resolution as the input image, where each pixel location on the object represents the probability of a grasp succeeding (red=high probability, black=low probability). The location and outcome of the first prior grasp (a success) can be seen superimposed on the input sample. Additional details are provided in Section II-C.

of the prior k grasping attempts (for the same object). The output for each support is also a 3D grid of size $11 \times 11 \times 64$. To show the network where each support image was grasped (and to extract features local to this location), we follow a late-fusion strategy as in [23]. First, we plot the original grasp location as a single pixel active in a binary mask (size 88×88 pixels), and then downsample the mask using bilinear interpolation [24] to a resolution of 11×11. We then perform an element-wise multiplication and summation with the respective encoded support image. The output of this step is a feature vector with a size of 64.

Physical Exploration Encoder: Once the local grasp features have been extracted from each support image, we then encode the wrist forces and torques through a small MLP with a single hidden layer, before merging through another hidden layer with the extracted local grasp features.

Query Affordance Decoder: After merging the visual & physical encoded support features, as with guided networks we separate each feature vector into positive and negative classes, based on whether the support grasp was successful or not. Given these groups, we then compute an average representation for both classes, yielding two vectors with 64 units. These vectors are then replicated 11×11 times and tiled to form an $11 \times 11 \times 64$ grid. We then concatenate and merge the query, positive, and negative representations through a series of 1d convolutions, and then up-sample using a bilinear interpolator network [24] to match the original image size. The output of this stage is a 1d channel image with a binary probability at each pixel location indicating whether a grasp at that location is likely to succeed or not.

III. Data Collection Procedure

Objects can have many different intrinsic properties. In this work, we focus on how an object's mass, and mass distribution affects the performance of grasping with a suction cup gripper. To account for real-world phenomenon (e.g. the flexibility of a gripper's suction pad), we collect data in the real-world using a 6-DoF Fanuc 200iC arm, along with nine custom 3d printed object shapes as shown in Figure 2.

A. *Object Descriptions*

Many real-world objects do not readily allow access to their intrinsic properties, nor do they easily allow for these properties to be re-configured in any consistent way. We originally thought to construct different shapes through e.g. lego-like parts, but found it difficult to build objects that were able to support additional weight *and* consistently hold their shape after being dropped by the gripper. By 3d printing the objects, we were able to standardize these parameters.

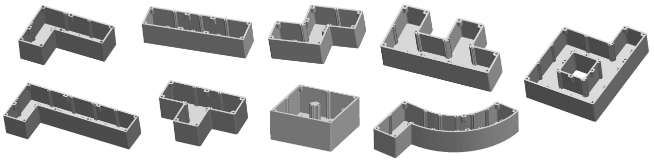


Fig. 2: CAD models of the nine 3d printed object shapes.

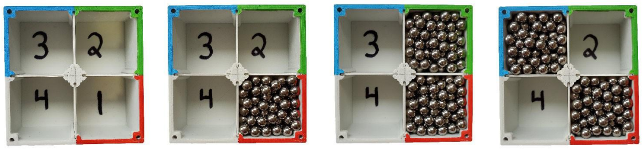
Our objects are simple, planar objects, and share some similarity to those used in [21]. Our goal however is in cases where the CoM cannot easily be detected through visual means alone. Five of the nine objects used this in work represent *tetrominoes* from the classical game of Tetris (L, Straight, S, T, Square). One object resembles a longer version of the L-shaped object, and the remaining three "F", "Banana" and "d" shapes were constructed by taking into consideration object symmetry, and a similar planar structure. Each shape is composed of multiple $40\,\mathrm{mm} \times 40\,\mathrm{mm} \times 40\,\mathrm{mm}$ hollow cells, and contains three different components: a solid outer shell, a removable back cover, and a set of dividers that can be inserted or removed to partition the individual cells. Dimensions for each object can be seen in Table I.

After printing, we first smooth the object's surface by applying a top coat of nail polish, and then paint each cell (on the outer shell) a different colour. Painting the object this way enables *us* to quickly ground-truth where the different weights are located, and provides context to the robot (i.e. extrinsic features) to help localize prior grasps.

To emulate different masses and mass distributions, we fully partition each object and fill select cells with $1/4\,\mathrm{inch}$ diameter carbon steel ball bearings as illustrated in Figure 3. Filling a single cell of an objects typically adds around $202\,\mathrm{g}$ to the overall mass, apart from the banana shape (which has a slightly modified cell definition) which adds around $140\,\mathrm{g}$.

TABLE I: Object X, Y, Z dimensions and number of cells.

Object	Cells	X (mm)	Y (mm)	Z (mm)
L	4	120	80	40
Straight	4	160	40	40
S	4	120	80	40
T	4	120	80	40
Square	4	80	80	40
Long L	6	200	80	40
F	6	160	80	40
Banana	7	218	80	40
d	9	160	120	40



0 cells filled　1 cell filled　2 cells filled　2 cells filled

Fig. 3: Permutations of which cells were filled with weight during data collection for the square object. The numbers inside each cell represent a different way for ground-truthing.

B. Data Collection Parameters

Our robot cell can be seen in Figure 4. Force and torque readings are captured by a Robotiq FT 150 sensor attached to the robot's wrist, while an Intel RealSense D435 camera (also mounted on the wrist) is used to collect RGB-D images.

When choosing which internal object cells are filled weight, we evaluate different configurations where either the object has no additional weight added to it (0-cells filled), configurations where 1 cell is filled with weight, or configurations where 2-cells are filled with weight. For the {Square, T, S, L, Straight, Long-L} objects, we collect grasping experiments using all unique permutations of [0, 1, 2]-cell filled instances (e.g. the square object in Figure 3). The larger {F, Banana, d} objects are grasped with an empty configuration, and two random instances of 1 & 2-cell filled configurations each. The choice to restrict these grasps was made as we generally observed grasps became much more unstable as the object's size was increased.

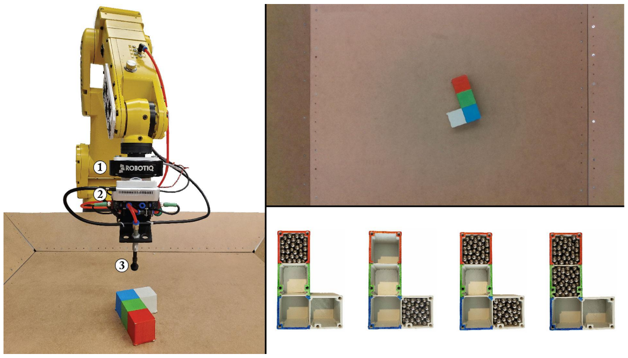


Fig. 4: Overview of the robot cell (1. Force sensor; 2. RGB-D camera; 3. Suction cup gripper), along with a sample view taken from the gripper-mounted camera, and a sample of cell configurations used for the L-shaped object. These configurations are unknown to the robot prior to any contact.

Each grasp is performed by the robot in a "top-down" manner. Our suction cup diameter is $17.5\,\mathrm{mm}$, and the maximum suction force is constrained to 60 PSI, which was found empirically to produce a modest amount of both successful and failed grasps in the (heaviest) 2-cell filled configurations (Figures 3 & 4). When lifting the object, we set a threshold height of $+18\,\mathrm{cm}$ along the world z axis from the grasped location. An object still in the gripper at this location is deemed to be a successful grasp; otherwise, the grasp is recorded as a failure.

C. Data Collection

Data collection begins with an object's shape and mass distribution first chosen from a predefined list (Section III-B). We then fill the corresponding object cells with weight, and place the object flat within the workspace bin.

Once placed in the bin, the robot's wrist is moved to a home location perpendicular to the table's surface, and an RGB-D image of the object is recorded. We subsequently segment the object from the background (using a combination of colour thresholding, depth thresholding, and a simple fully-convolutional network trained to predict noisy object masks), and then select a grasp location by randomly sampling from the visible points across the object's surface. When choosing where to grasp, we mask the object's edges from being selected by computing a distance transform for the object mask, and then thresholding it to keep the pixels within 40% of the maximum distance value. We also apply an offset of $2\,\mathrm{cm}$ along the world z to the chosen grasp location to ensure the suction cup forms a complete seal with the object. Parameters for the offset and distance transform were found empirically to work well with our gripper.

If the suction cup is unable to form a proper seal on the object (i.e. the pressure experienced between the object and the suction cup fails to cross a threshold of $-50\,\mathrm{kPa}$), the experiment resets with neither a success or a fail, and the above process is repeated. Otherwise, the robot attempts to lift the object and records the forces and torques during the lift at a sampling rate of 20 ms. If the object falls any time during the lift, the end of the trial is marked – with the outcome recorded as a failure and the robot reset to the home position. If the robot is still grasping the object at the height of the lift, the trial is marked as a success, and the object's pose is randomized prior to deactivating suction and returning to the home position. A total of 50 grasp attempts are collected per cell configuration, per object.

Depending on the object's shape and cell configuration, deactivating the suction while the object is above the table could result in resting poses for the object that are counter-intuitive to what we would expect without knowing the object's intrinsic characteristics. We generally tried to avoid these during data collection by manually resetting the object pose when this phenomenon was noticed, though some poses do remain present in the data. Our pre-processed dataset (Section IV-A) can be accessed online at: `https://doi.org/10.5683/SP2/YCBUSR`.

IV. Learning

Our task is akin to semantic segmentation, with some mild differences. Similar to [7], given an RGB image of an object, we seek to predict a class value for every pixel in the input image. In this work we deal with binary classes [0, 1] which corresponds to whether a top-down suction grasp is likely to fail or succeed at each pixel location, respectively. Because the object's intrinsic properties are unknown prior to grasping however, we require additional context (through previous grasping attempts) to guide the prediction process.

A. Dataset Pre-processing

A total of 3,649 grasp attempts were collected across all object-cell configurations, where one attempt was removed for missing a pre-grasp observation. The average grasp success rate was 0.4713 (std=0.3306). Within the dataset, we noticed it was possible for multiple grasp attempts to be repeated within any given object-cell configuration, simply due to the finite amount of variability and task setup. Prior to learning, we filter these attempts by first measuring the Jaccard similarity between any two centered object masks (threshold = 0.75), and then by measuring the pixel-wise grasp distance. Objects that have similar poses, grasp locations, and outcomes were removed from the dataset. Following this procedure, a total of 2,868 samples remain for learning. An overview of the data is shown in Figure 5.

RGB Images: We first segment the object from the full 720 × 1280 resolution image, then subsequently remove the depth channel and center the object by taking a fixed-size crop of 352 px. After centering, we downscale the image by a factor of 4, and normalize the values to [-1, 1].

Target Label: As with semantic segmentation, our target label t is represented as image with the same height and width as the RGB image. On this image, we toggle a single pixel at the grasped location (see Section III-C) to have an intensity of 1 if the grasp was a success or a 0 if the grasp was a failure. All other locations are treated as masked regions, which are not optimized for during training. The label is centered and downscaled identically to the RGB image.

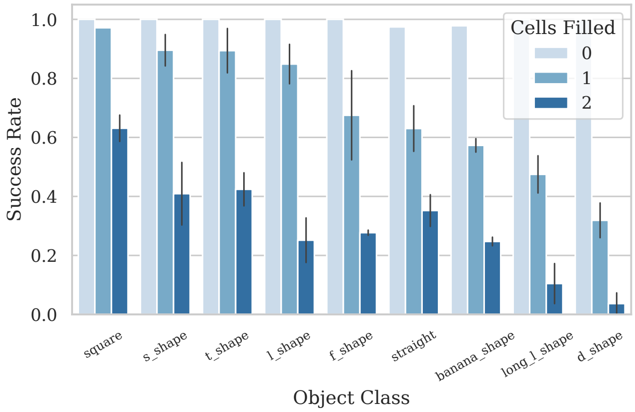


Fig. 5: Overview of the filtered data as a fraction of successful grasps per object class, averaged over all object-cell configurations. Note that the f_shape, banana_shape, and d_shape objects only had a select few weight configurations tested due to their larger shape complexity.

Sensory Feedback: For the sensory data, we use the force and torque readings with respect to the local frame, and then compute an average of the final 100 ms of the lift. Both force and torque data is standardized to have zero mean and unit variance across the individual x, y, z channels.

B. Optimization

Our objective function is the binary cross-entropy loss:

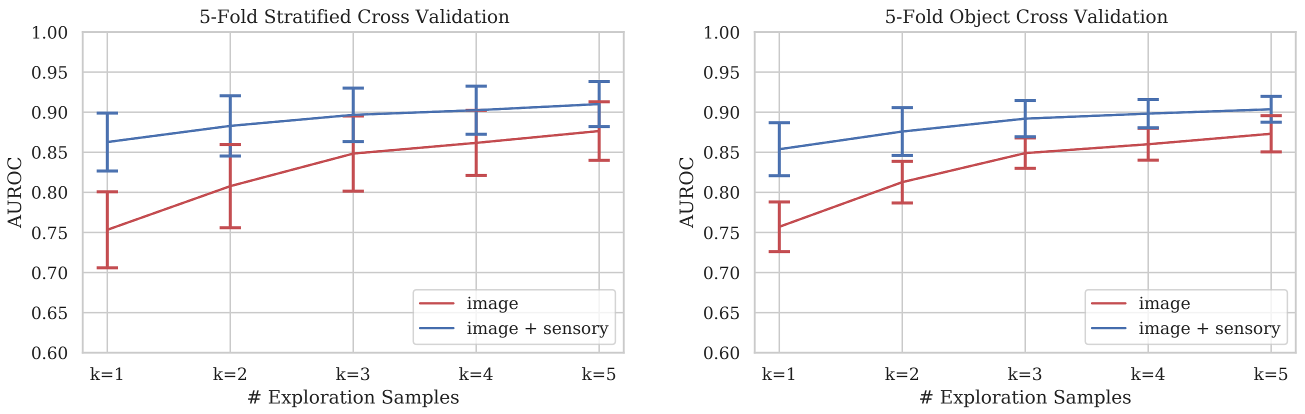
$$\mathcal{L} = t \log(\mathcal{G}) + (1 - t) \log(1 - \mathcal{G}) \tag{1}$$

where t (the target) and $\mathcal{G}$ (the predicted affordance map) are both represented spatially. We update network weights using the Adam optimizer [25], with a learning rate of $5e^{-4}$ and $\beta = (0.5, 0.999)$. We use a small amount of weight decay for regularization ($2e^{-4}$) and apply dropout to the sensory feedback encoder with $p = 0.5$, and convolution filters with $p = 0.1$. During training, we also apply random horizontal and vertical flips to the query image with $p = 0.5$.

To construct a mini-batch of training examples, for every example, we first randomly choose an object class, and then randomly choose a 0, 1, or 2-cell filled configuration belonging to that class. Given this configuration, we then randomly sample a query instance, along with k other grasping attempts to be used as the corresponding supports. We do not discriminate sampling between successful and failed grasps when choosing support instances, as it is possible that a dataset may or may not contain one or the other.

C. Experimental Setup

We evaluate our models based on (1) how well a robot is able to predict affordances if it has seen an *object class* before (but has not seen a particular object-cell configuration), and (2) on how well the robot can generalize to objects it has never seen before. Both settings employ cross-validation



(a) AUROC for seen objects with unseen cell configurations.

(b) AUROC for unseen objects.

Fig. 6: Mean & standard deviation of AUROC evaluated across both 5-fold stratified CV (partitioned by the 50 attempts per object-cell configuration), and 5-fold CV split by object class. Our models are trained using $k = 5$ support examples (except for the baseline); also reported here are the effects of using fewer than k samples during the inference procedure. The baseline model reaches a (mean, std) AUROC of 0.67 ± 0.06 and 0.64 ± 0.04 for the stratified and object CV respectively.

(CV): the former uses stratified-sampling to balance the object classes across each fold, while the latter employs a leave-one-object-out scheme, in which every held-out fold contains two novel objects (apart from the final fold).

Within both settings, we evaluate a *baseline* architecture that predicts affordances without considering any prior grasp attempts, an *image-only* architecture that predicts affordances using only visual observations, and an *image + sensory feedback* architecture that mimics the structure in Figure 1. The baseline architecture uses the visual query encoder and bilinear decoder network only. The image-only architecture discards the sensory branch + merge layer, and immediately splits the grasp feature vectors into their positive and negative representations. With respect to Figure 1, both the query and exploration networks are composed of three layers, having 2× Conv-ReLU-Dropout blocks each. We use 32 filters in the first layer, and 64 in both the second and third. Each convolution uses a kernel size of 3 and stride 1. Parameters between the query and exploration encoders are shared.

V. Results

We report our results as the mean and standard deviation of the Area Under the Receiver-Operating Characteristic Curve (AUROC) [26] across all CV folds. Figure 6 highlights our results, and sample predictions can be seen in Figure 7. Across both cross-validation strategies, using sensory feedback improves the models performance. The mean AUC scores *with* sensory feedback are also generally more stable across different k's then when compared to using images only – including those cases where fewer support examples are used. For seen objects, the performance bands are larger than for unseen objects, reflecting performance fluctuations. The baseline model does not consider prior grasping attempts, and performance is limited on the testing sets.

To illustrate the effects of the support examples when making a prediction, Figure 8 presents sample outputs when $k = 0$. These predictions are obtained by zeroing out the latent representations for both the positive and negative support examples in Figure 1, right before merging with the query representation. In general, the system favours predicting successful grasps near the object's geometric center.

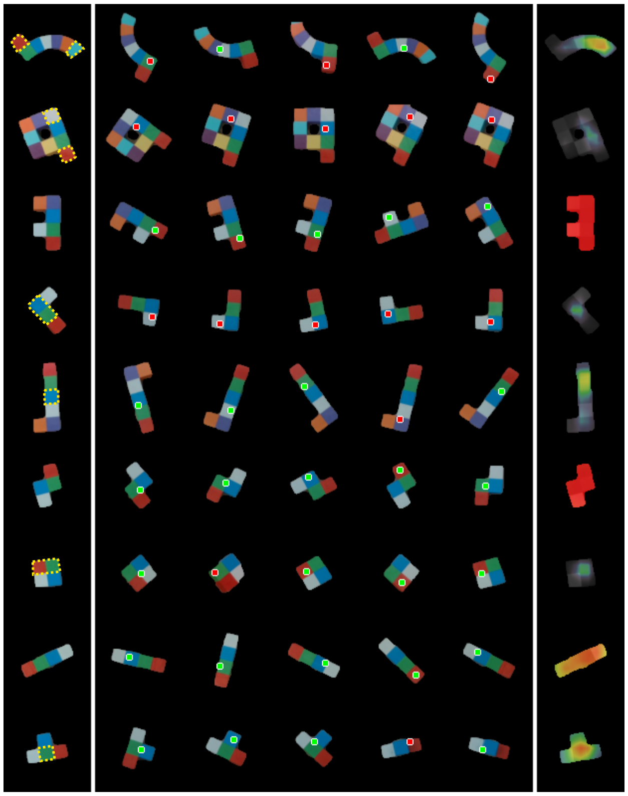


Fig. 7: Sample affordances predicted with $k = 5$ exploration samples. **Left Column**: query image & which cells have weight (circled in yellow). **Middle**: k exploration samples, along with their grasp location and outcome (red=failed grasp, green=successful grasp). **Right**: Predicted affordances on the object. The more red the colour, the more likely a grasp is to succeed at that location. Best viewed in colour.

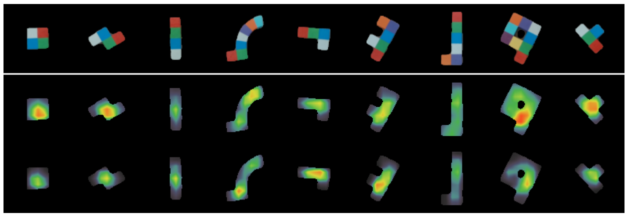


Fig. 8: Sample "default" affordance predictions when $k = 0$

Figure 9 presents an example for how predictions change as $1 \leq k \leq 5$ support examples are presented to the trained networks. When comparing between the image-only and image + sensory feedback rows, one can observe important performance advantages for incorporating force / torque values. For example, when no cells are filled (i.e. the object can be grasped almost anywhere successfully), the performance is excellent even with $k = 1$ when using sensory feedback, versus those from the image-only architecture for any k. With one cell filled, with sensory feedback the predictions appear to be much more localized to where the support grasps are occurring (in this case, near the CoM).

Finally with two cells filled, the predictions become stable near the CoM with $k \geq 2$ both with and without sensory feedback. However, we note that while both architectures predict grasps around the same location, the prediction confidence is different. With image-only, the grasp location is predicted to be successful with a very strong probability (i.e. a dark red colour). On the other hand, with image + sensory feedback the same grasp location is predicted, but not with the same probability of success (i.e yellow / green colour). This is an important distinction given that Figure 5 shows that failure rate is higher with two-cell filled cases.

These observations show that using sensory feedback can lead to a more accurate grasp prediction with fewer grasping experiments. Given the wide range of object materials and density distributions, it is not practical to expect large datasets to be available for each intrinsic object property. Figure 9 shows that this can be avoided.

Figure 9 provides further insight into how the proposed learning framework operates. Comparing the performance of the one cell versus two cell-filled cases, one can observe that the first support examples for each are positive and negative respectively. Given that every prediction made by our affordance network is made based on some past history (k examples) grasping the exact same object, prediction accuracy tends to improve once positive examples are provided to condition the predictions. As such, it is important for a model to understand the relation between the context of the historical grasps, and the current observation.

This raises an interesting question — how do we predict affordances that *are not* near the sampled prior grasps? Do we just exhaustively grasp the object in order to build an object model? If only visual observations are provided to the network, there is a motivation for our model to overfit on object shapes and learn average local representations where grasps are likely to succeed. With these learned biases, one way for the model to leverage prior grasping experiences is to use them to simply refine these assumptions on where the CoM is likely to be. On the other hand, when sensory feedback is available the model now has access to a signal directly correlated with the object's CoM, which has the potential to discover graspable locations, or phenomenon such as *anywhere* on the object can be grasped much quicker then through visual observations alone.

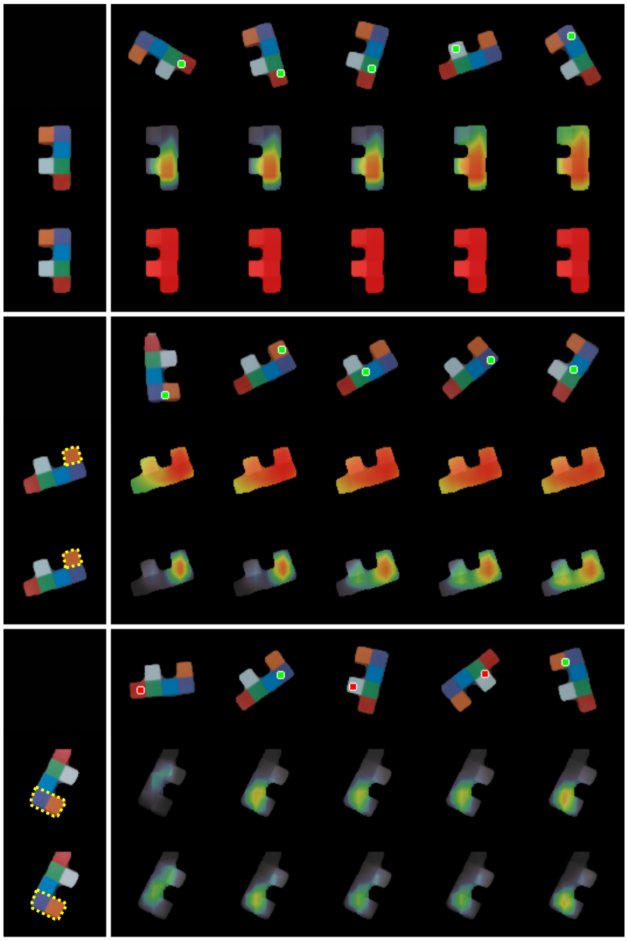


Fig. 9: Predicted suction affordances with $k = [1, 2, 3, 4, 5]$ explored grasps. The **top** row of each segment shows the exploration samples used (along with grasp location & outcome), while the **middle** row shows predictions from the image-only model, and the **bottom** row is from image + sensory feedback model. Moving left to right, the networks see all previous exploration grasps when making a prediction.

VI. Discussion

In terms of failure cases, we found it was possible that if the support examples were low quality, or grouped in a single location, the "default behaviour" (Figure 8) could suggest incorrect grasp locations. We also noticed failure cases where images of the support objects and query were vastly different (e.g. an object sitting upright vs. flat); in this instance, more geometric reasoning about the object may be required.

When building a grasp motor image of the object [22], incorporating more intrinsic object characteristics and task requirements in a motor image raises questions surrounding scalability and the number of data points needed. By adapting the late fusion architecture of [23] to our task, we can quickly learn to predict affordances from only a few prior examples (e.g. Figure 9) rather then collecting and training on a new, large dataset for every novel object.

While the context of our experiments was limited to suction grasps and a finite set of objects, we argue that the presented results are not solely limited to this context due to the way various experimental parameters were set. The suction force for example, was determined to allow both failed and successful grasped to be observed given the objects — A much stronger suction force would have resulted in all successful grasping experiments, while a much weaker suction would have resulted in all failed grasping experiments. The same applies to the kind of objects used and how the CoM was changed. Many objects can be assumed to have uniform density and symmetric shape, resulting in a center of mass closer to the object geometric center. Some of the objects in the object set exhibit this characteristic. Other objects, however, have an asymmetric shape and the weight distributions often create an off-(geometric) center, CoM location. This generalizes to all types of objects in real-world with non-uniform density or shape.

VII. Conclusion

Intrinsic characteristics are present in every single object we encounter in the real world and it is important to account for them during the modeling process. In this work, we present a scenario where a robot must account for intrinsic properties (in the form of an object's CoM) when planning suction grasps. We view this as a step for an affordance model that incorporates many different intrinsic properties, such as an object's surface friction and rigidity.

VIII. Acknowledgements

We would like to thank Patrick Wspanialy for designing and building the first iterations of the objects used in this study, and Terrance DeVries for useful feedback.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[2] E. Jang, C. Devin, V. Vanhoucke, and S. Levine, "Grasp2vec: Learning object representations from self-supervised grasping," *Proceedings of Machine Learning Research*, vol. 87, p. 99–112, 2018.

[3] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Conference on Robot Learning*, 2018, pp. 306–316.

[4] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[5] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *IEEE International Conference on Robotics and Automation*. IEEE, 2016, pp. 1957–1964.

[6] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz, and S. Levine, "Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods," in *2018 IEEE International Conference on Robotics and Automation*. IEEE, 2018, pp. 6284–6291.

[7] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE International Conference on Robotics and Automation*. IEEE, 2018, pp. 1–8.

[8] T. Kurutach, A. Tamar, G. Yang, S. J. Russell, and P. Abbeel, "Learning plannable representations with causal infogan," in *Advances in Neural Information Processing Systems*, 2018, pp. 8733–8744.

[9] T. Standley, O. Sener, D. Chen, and S. Savarese, "image2mass: Estimating the mass of an object from its image," in *Conference on Robot Learning*, 2017, pp. 324–333.

[10] N. K. Kannabiran, I. Essa, and C. K. Liu, "Estimating mass distribution of articulated objects through physical interaction," *arXiv preprint arXiv:1907.03964*, 2019.

[11] K. Takahashi and J. Tan, "Deep visuo-tactile learning: Estimation of tactile properties from images," in *2019 International Conference on Robotics and Automation*. IEEE, 2019, pp. 8951–8957.

[12] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," in *Advances in neural information processing systems*, 2015, pp. 127–135.

[13] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *2015 IEEE International Conference on Robotics and Automation*, May 2015, pp. 4304–4311.

[14] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.

[15] D. Kanoulas, J. Lee, D. Caldwell, and N. Tsagarakis, "Center-of-mass-based grasp pose adaptation using 3d range and force/torque sensing," *International Journal of Humanoid Robotics*, vol. 15, p. 25, 03 2018.

[16] H. Merzić, M. Bogdanović, D. Kappler, L. Righetti, and J. Bohg, "Leveraging contact forces for learning to grasp," in *2019 International Conference on Robotics and Automation*, May 2019, pp. 3615–3621.

[17] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater, "Learning grasp affordance densities," *Paladyn, Journal of Behavioral Robotics*, vol. 2, no. 1, pp. 1–17, 2011.

[18] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Detecting object affordances with convolutional neural networks," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2016, pp. 2765–2770.

[19] M. Kokic, J. A. Stork, J. A. Haustein, and D. Kragic, "Affordance detection for task-specific grasping using deep learning," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 91–98.

[20] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *Proceedings of Robotics: Science and Systems (RSS)*, 2019.

[21] L. K. Klein, G. Maiello, V. C. Paulun, and R. W. Fleming, "How humans grasp three-dimensional objects," *bioRxiv*, p. 476176, 2018.

[22] M. Veres, M. Moussa, and G. W. Taylor, "Modeling grasp motor imagery through deep conditional generative models," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 757–764, 2017.

[23] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, and S. Levine, "Few-shot segmentation propagation with guided networks," *arXiv preprint arXiv:1806.07373*, 2018.

[24] E. Shelhamer and K. Rakelly, "revolver," https://github.com/shelhamer/revolver, 2019.

[25] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of 3rd International Conference on Learning Representations*, 2015.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.