Semantically Grounded Object Matching for Robust Robotic Scene Rearrangement

Walter Goodwin[1], Sagar Vaze[2], Ioannis Havoutis[1] and Ingmar Posner[1]

Abstract— **Object rearrangement has recently emerged as a key competency in robot manipulation, with practical solutions generally involving object detection, recognition, grasping and high-level planning. Goal-images describing a desired scene configuration are a promising and increasingly used mode of instruction. A key outstanding challenge is the accurate inference of matches between objects in front of a robot, and those seen in a provided goal image, where recent works have struggled in the absence of object-specific training data. In this work, we explore the deterioration of existing methods' ability to infer matches between objects as the visual shift between observed and goal scenes increases. We find that a fundamental limitation of the current setting is that source and target images must contain the same *instance* of every object, which restricts practical deployment. We present a novel approach to object matching that uses a large pre-trained vision-language model to match objects in a cross-instance setting by leveraging semantics together with visual features as a more robust, and much more general, measure of similarity. We demonstrate that this provides considerably improved matching performance in cross-instance settings, and can be used to guide multi-object rearrangement with a robot manipulator from an image that shares no object *instances* with the robot's scene. Our code is available at `https://github.com/applied-ai-lab/object_matching`.**

I. INTRODUCTION

In recent years, there have been a number of successes in applying deep learning to enable manipulation skills on robots which operate directly from images. Alongside this, the 'goal image' has emerged as a convenient way to specify an instruction for a robotic task, ranging from guiding visual servoing towards a single object [1] to multi-object rearrangement settings [2]–[4]. Goal images enable specification of the goal in the same modality as the system input for a robot with vision sensors, and are a practical way to express desired spatial outcomes for objects in a scene in many settings. Consider a household task, such as laying a dining table, or an industrial task, such as kitting products into a canonical set — in both settings it can be more convenient to simply image the desired goal state rather than exhaustively define all degrees of freedom.

Furthermore, object rearrangement problems have attracted interest recently as a challenging, and quite general, robotic manipulation problem, with the goal image motivated as a key component [2], [3], [5]–[8]. There has been some success recently in the development of systems that can achieve rearrangement tasks when provided with a goal image [3], [9], generally through the integration of pre-trained object segmentation, grasping primitives or grasp prediction networks, and subsequent high-level planning at the object level. However, progress in learning both grasp prediction networks [10], [11] and segmentation networks [12] that generalise to unseen objects, mean that it is increasingly possible to both localise and grasp objects (two important stages in successful rearrangement [3], [8]).

Thus, a key outstanding challenge in specifying outcomes with images is that of robustly *matching* objects in a goal image to those in the current scene, following object detection. Successful matching is critical to successful interpretation of a goal image, and poor matching has emerged as the principal performance bottleneck in recent works on tabletop rearrangement [3], [13]. Currently, a wide range of matching methods are used, from hand crafted features such as colour histograms [14] to object-specific visual features learnt unsupervised [15], and deep feature extractors trained on large scale computer vision datasets [3], [9].

This work conducts a controlled analysis and comparison of approaches to matching under various visual shifts between the current (source) and goal (target) images, which we find to be lacking in the existing literature. We control for factors such as pose and background mismatch and report results on a number of baselines. Furthermore, we identify a fundamental limitation of the existing matching approaches, finding that matching is currently only demonstrated when source and target images contain the same *instance*, i.e two views of exactly the same object. This can be restrictive in practical settings. Consider again the table laying problem. It is unreasonable to expect the goal image to contain the precise crockery instances in the current scene. Rather, the goal image will specify a *semantically* consistent target, specifying where a mug or plate should go, regardless of their precise appearance. We thus further examine the matching problem in a cross-instance setting, in which source and target images contain different instances of the same *semantic* object, which may be visually distinct. In this setting, we find that existing matching approaches, which rely solely on visual descriptors, degrade significantly in performance.

Next, we propose a novel solution to the cross-instance matching problem via the re-purposing of the recently released CLIP model [16]. In contrast to existing matching approaches, which rely solely on visual features, the proposed model facilitates the introduction of semantic information to the matching problem, leading to a substantial boost in our proposed cross-instance matching performance. Finally, we deploy our method on a real robot system, finding that our proposed semantic matching protocol is important for cross-instance matching in the real world.

[1]Oxford Robotics Institute, [2]Visual Geometry Group, University of Oxford, [1]23/[2]25 Banbury Rd, Oxford OX2 6NN
`firstname@robots.ox.ac.uk`

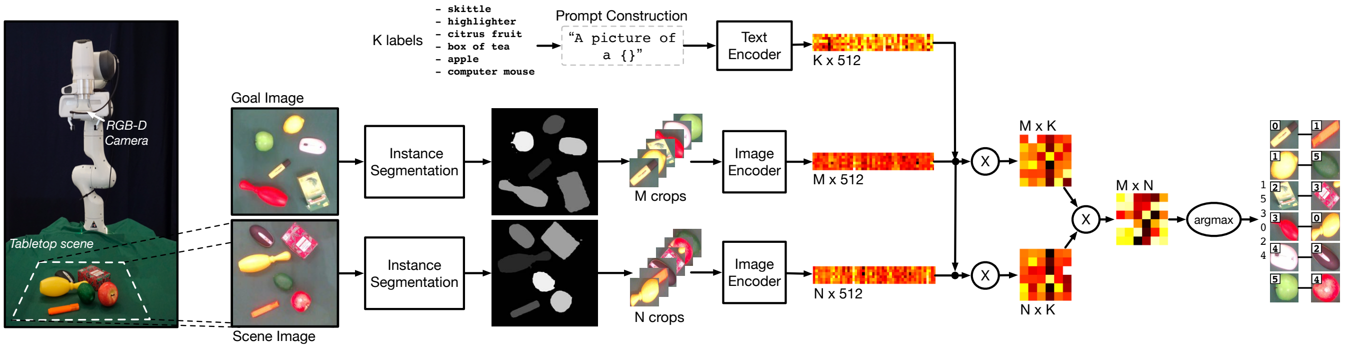

Fig. 1: Visual-Semantic Matching: For both current scene and goal images, we use an instance segmentation network to extract crops of all present objects. We then compute similarities between the crops' CLIP visual features and the semantic features of a set of K text prompts. A final $M \times N$ similarity matrix between the current and goal crops can then be computed based on the crops' similarity to each of the semantic features. The 'X' operator is $X(\mathbf{A}, \mathbf{B}) := \mathbf{A}\mathbf{B}^{\mathbf{T}}$.

II. Related Work

A number of recent works, challenges and benchmarks have considered robotic scene rearrangement with visual observations, a challenging problem for robotics and embodied artificial intelligence [2]. Simulated benchmarks have been proposed for both mobile [5], [6] and tabletop robotic manipulation [7], [17]. A common theme is the use of visual instructions in the form of goal images, as a means of communicating a target scene configuration. Recent attempts at tabletop rearrangement on *real* robot platforms take a modular approach, where object detection is executed in the current (source) scene and the goal (target) image. Objects are then matched across the scenes, and planning over pick-and-place actions brings about the desired object displacements [3], [8], [9], [13]. Existing works have tackled particular parts of this pipeline, with work on improved collision checking [8], high-level planning [3], [9], [18], and vision-based RL approaches [4], [19]. Several of these works note failures in successful matching [3], [9] due to the challenges of accurately comparing objects from only single views. It is notable that, with the exception of some invariance to viewpoint shift [9], goal-conditioned works are restricted to goal images generated in the exact setting, and with the exact object instances, that will be encountered by the robot at deployment. Matching approaches handle only *instance-level* correspondence, and cannot infer *semantic* matches between scenes. We review existing matching approaches, and attempts to exploit object semantics, in robotics.

A. Object matching in robotics

Hand-crafted visual descriptors such as colour histograms were originally proposed for this task [14] and remain widely used today [20]. More recently, features extracted from the backbone of a CNN classifier have been applied to tackle the matching problem [3], [9]. These features have been trained on large scale datasets such as ImageNet [21] and demonstrate invariance to nuisance factors such as lighting and 2D rotation. However, outside of training classes, these features have not been trained to be invariant to *instance-shift*, and hence degrade in matching performance when the source and target images contain different instances of semantically identical objects. Object matching is related to the mature field of template matching, but where only one single view of the object exists as the 'template'. Solutions to single-view instance recognition have been proposed. [22] fine-tune a CNN pre-trained on ImageNet on multiple views of many 3D objects to learn pose-invariant representations which is finetuned on a single image of a new object for pose-invariant object detection. They show that this performs far better than a large range of pre-existing template matching techniques for a single template image. Other deep learning methods reduce the number of views required [23]–[25]. However, while these methods are often able to build models robust to *pose* shift of objects, they require that the same *instance* of the target object is being sought. [26] trained a cross-domain image matching technique to enable products viewed by the robot to be matched to a database of Amazon product photos, but this *instance*-matching method requires a known, *systematic* shift between source and target. In contrast, in this work, through leveraging visual-semantic grounding, we handle a much more general case, in which arbitrarily different instances of objects with the same underlying semantic description can be successfully paired.

B. Visual-semantic object picking

There have been numerous attempts to enable language-guided robotic manipulation by grounding language instructions to visual observations. While a different mode of instruction to the goal image, these works are related in their use of semantics in aiding robotic disambiguation of the visual world, and in their use of vision-language models. Work concurrent with this *conditions* imitation learning on CLIP embeddings of text instructions to improve generalisation [27]. Several recent works present systems to guide a robot to pick a particular object from a scene with language prompts viewed as *referring expressions* [28]–[30]. Referring expression comprehension is an area of computer vision research that seeks to ground an unstructured text prompt from a human that *refers* to an item visually present in an image, and locate the item on this basis. When brought to robotic systems, standard models [31] trained on a dataset with a limited number of classes tend to be used [28]–[30]. While this work also proposes language as a mechanism for resolving visual ambiguities, we do not rely on pre-training and are able to handle arbitrary object classes.

III. Visual Object Matching

In this section we describe the core matching models we compare in this paper. We first describe the proposed visual-semantic models based on CLIP [16], before summarising the baseline matching models against which we compare.

Preliminary notation A common operation when comparing objects is, given a set of inputs, $\mathcal{T}$, and a feature extraction function, $f(\cdot)$, to build a set of normalised feature vectors. Here $\mathcal{T}$ could refer to a set of image crops or text prompts. Formally, we define the operation $\mathcal{F}$, as:

$$\mathcal{F}(f, \mathcal{T}) = \left\{ \frac{f(t)}{|f(t)|} \quad \forall t \in \mathcal{T} \right\} \tag{1}$$

A. Visual-Semantic Models

In this work, we propose *semantic matching* through the recently released CLIP model [16]. CLIP consists of a pair of neural network embeddings which jointly map text-image pairs into a common feature space. The model is trained on web-scale data and is thus capable of interpreting a wide range of semantic objects. In this way, by finding similarities between a set of text (semantic) prompts and a given image, one can identify the most likely category of an object within the set in a 'zero-shot' fashion.

1) Object Matching: The proposed visual-semantic object matching process is shown in Figure 1. Leveraging unseen object instance segmentation [12], object crops are taken, and all M objects in the source and the N objects in the target are passed through the image encoder. The set of K possible object categories are passed through the text encoder. This allows us to construct two classification matrices, C_s and C_t, which describe the model's confidence that each object belongs to each of the K categories. Next, based on these confidences for each object, we compute a similarity between each object in the source and the target.

Consider Φ_v and Φ_s as the deep image and text CLIP encoders respectively. Consider $\mathcal{X}$ as the set of cropped object patches in either the source, $\mathcal{X}_s$, or target, $\mathcal{X}_t$, and $\mathcal{Y}_K$ as the set of text prompts. Normalised visual features are constructed for the crop sets as $v_s = \mathcal{F}(\Phi_v, \mathcal{X}_s)$ and $v_t = \mathcal{F}(\Phi_v, \mathcal{X}_t)$, and semantic features as $s = \mathcal{F}(\Phi_s, \mathcal{Y}_K)$.

The classification matrix for either source or target, C, is then constructed as $C_{ik} = \langle v_i, s_k \rangle$ where $\langle \cdot, \cdot \rangle$ represents an inner-product. The similarity matrix S between source and target is computed as $S = C_t C_s^T$. The final step is performing assignment based on the similarity matrix, which can either be performed with the `argmax()` operation or through minimum weight matching with the Hungarian algorithm. We experiment with two variants of the CLIP-Semantic model: first we pass all K semantic labels for matching (**CLIP-SemFeat-K**); we also pass only the semantic labels which we know to be present in the target image (**CLIP-SemFeat-N**). These two settings both correspond to practical scenarios. In many tasks, there might not be prior knowledge of exactly what will be in the scene, but we know that the relevant objects can be described by a subset of K labels. For instance, a robot laying a table might enumerate the names of all items of tableware, even if only a fork and spoon are present. On the other hand, in a warehousing setting, the exact objects present may be known, giving a set of N labels. Intuitively, if $K > N$, **CLIP-SemFeat-K** considers some labels irrelevant to the scene.

2) Prompt Engineering: We seek to optimise CLIP to ground the textual descriptions of the objects to their corresponding objects across the considered datasets (Section IV), to ensure that the model can provide meaningful proximity measures between crops of objects across the images, and their text descriptions, through the shared embedding space. To this end, we engineer the text prompts used in the semantic embedding function. For each object class, we take a small number of reference crops, and type around 5 short descriptions of the object. Using CLIP, we compute the cosine similarity between the visual embedding of the crops, and text embeddings of these descriptions (including the original description). We take the descriptions with highest similarity as the improved set of semantic prompts, with the entire process taking a couple of minutes for a single user.

B. Baselines

In our experiments, we compare to a number of baseline matching approaches used in recent rearrangement works. All methods rely on extracting purely visual descriptors of crops based on the pixel values, before constructing a similarity matrix between the N source and M target crops. Formally, the visual features are extracted from the crops as $v = \mathcal{F}(f, \mathcal{X})$ for a given feature extraction function $f(\cdot)$. The similarity matrix is computed solely based on the visual descriptors as $S_{mn} = \langle v_m, v_n \rangle$.

Colour Histograms: the winning submission from 59 teams to the OCRTOC Tabletop Rearrangement Challenge 2020 used the cosine similarity between colour histograms for object recognition, and deployed nearest neighbour voting against a dataset of multiple images of each potential object to determine the identity of an object in a scene [20]. While the goal-image setting provides only one reference crop for each object to be matched, we consider cosine distance on colour histograms as a first baseline. Here, $f(\cdot)$ involves concatenating histograms for both the hue and saturation values across all pixels in an image. Hue and saturation are projections of RGB pixel values into a frame more in line with visual cues of interest to human observers [14].

AlexNet-S: in [9], which achieves multi-object rearrangement with goal-image matching for up to 12 cubes on a tabletop, the authors use distances between the output of the conv3 layer of AlexNet pre-trained on ImageNet. The objects considered, though, are all cubes with distinct colours. Here, $f(\cdot)$ extracts the flattened spatial features of the conv3 layer, which retain some of the spatial structure of the input crops.

ResNet50-(S/G): most recently, ResNet50 features have been used in [3], where they were used for finding object matches between scenes of 2 to 5 objects. In this case, we experiment with two settings, with spatial features, **ResNet50-S** (before the final max-pool layer), and with global features **ResNet50-G** (immediately after the final max-pool layer).

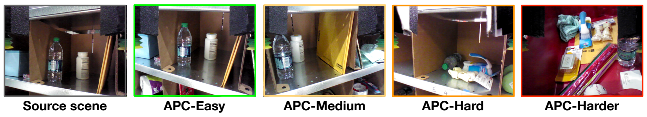

Fig. 2: Examples of the 4 same-instance matching settings we construct from the APC dataset.

Model	Top-1	Top-5
Random Guess	2.6	12.8
CLIP	30.7	54.6
CLIP+	**38.0**	**65.2**
CLIP++	35.0	61.5

TABLE I: Zero-shot classification performance of the vision-language model (CLIP) across the APC dataset. **CLIP** uses the exact same wording as the original product names from APC [32], formatted as "A picture of a {...}". **CLIP+** uses the same formatting but with 'better' labels chosen as described in V-B. **CLIP++** additionally ensembles over all labels and additional prompt formats, as described in V-B.

CLIPVisual: to allow us to isolate the effect of the *semantic* information encoded in the full CLIP model, from the effect of the CLIP model's visual feature extractor alone, we also embed the crops through only the visual CLIP backbone, $f = \Phi_v$. We use these features identically to the other purely visual descriptors.

IV. Datasets

Our aim is to investigate matching performance in two distinct settings: *same-instance matching* and *cross-instance matching*. We first leverage the APC dataset [32], which allows investigation of the instance matching setting while controlling for degrees of visual shifts in pose and background. We then use the LVIS dataset [33] to look at cross-instance matching, as it contains multiple instances of the same semantic class in different settings.

A. Amazon Picking Challenge (APC) 2016 dataset

We use the dataset collected by the MIT team for their entry to the Amazon Picking Challenge (APC) 2016 [32]. This comes with accurate predicted instance segmentation masks for the 39 objects considered in APC 2016, and contains 7,281 images from 452 distinct scenes. Scenes contain between one and twelve objects arranged in two different settings: a shelf, pictured from the side and a plastic tote box, pictured from above. Each scene type was recorded in two different locations, with different lighting conditions. Each shelf scene is imaged from 15 different views, and each tote scene from 18 views. The same 39 object instances occur throughout, though between scenes they vary in pose, occlusion relationships with other objects, background and lighting conditions. From these conditions we are able to form four different object matching settings, which we hypothesise - and empirically confirm - pose progressively more challenging conditions for matching.

APC-Easy: we set up matching problems where both source and target image are drawn from exactly the same scene, but from different views. In this condition, we consider only pairs over views that are close. Objects will in general retain substantial visual similarity, but will be viewed from different angles, and occlusion conditions may change slightly.

APC-Medium: as with **APC-Easy**, except we consider source/target pairs for matching that are maximally dissimilar i.e. viewed from diagonally opposite corners. While still matching within-scene, relative object poses are substantially shifted, and occlusion conditions will vary.

APC-Hard: we formulate source-target pairs from *different* scenes but of the same setting (e.g. shelf). All scene pairs in which there is at least one valid match to be made are matched. Object poses are different between scenes, and some objects will not have any valid match - we do not count these towards the reported accuracy.

APC-Hardest: as with **APC-Hard**, but we sample source and target scenes from opposite settings e.g. shelf vs tote.

We remove any trivial examples from these partitions, where both source and target scene contain just one object.

B. Large Vocabulary Instance Segmentation (LVIS) dataset

A key limitation of the current matching tasks demonstrated in the robotics literature is that they only consider same-instance correspondence. This is evident when looking at recent work [2]–[4] and instance-specific challenges and datasets such as APC [32], and is likely due to the open nature of the cross-instance matching problem. However, achieving this goal would enable carrying out robotic manipulation tasks that leverage a goal image without the restrictive constraint that this goal (target) image needs to contain visually very similar objects to - i.e. the same object instance as - the current (source) image. We present our lab-based experiments in Section V-D, but for a more comprehensive, in-the-wild, and unbiased assessment of visual-semantic matching performance on cross-instance settings, we use the Large Vocabulary Instance Segmentation (LVIS) dataset [33], a densely annotated object recognition and segmentation dataset, with a training set of 1.27M annotations over 1203 object classes, across 100,170 images. Annotations consist of segmentation masks and class id. To formulate a large set of matching problems relevant to robotics applications, we select a subset of 40 objects that could be feasibly grasped by a tabletop manipulator, from the top 200 most-occurring classes. All annotations that have a pixel area of less than 32^2 are disregarded, and and only the first instance of a given object class from each image is kept. This leaves around 36000 annotated instances, with a mean of 900 per class. Unlike in the APC dataset, the open-world settings and high number of possible classes in LVIS means most pairs of scenes have either zero or a small intersection of classes present. These scene pairs would present trivial matching settings, and so we formulate matching problems synthetically. For an N-way matching problem, we sample N labels from the set of 40 classes, and then sample a pair of different object crops C_i^S, C_i^T for each label $i \in N$. We then seek to match the set of source crops $\{C_i^S\}_1^N$ against the set of target crops $\{C_i^T\}_1^N$. We crop based on ground-truth masks, as we focus on matching and not detection.

model	Same-Instance (APC) [32]				Cross-Instance (LVIS) [33]	
	Easy [%]	Medium [%]	Hard [%]	Hardest [%]	8-way [%]	20-way [%]
RandomGuess	32.3	33.1	25.4	26.3	12.4	5.0
ColourHist	89.4	57.7	48.0	42.2	20.3	9.8
AlexNet-S	95.8	60.6	46.3	37.7	30.5	17.6
ResNet50-S	**97.5**	64.6	54.2	47.8	46.9	32.6
ResNet50-G	96.9	**67.7**	**61.3**	**55.0**	50.6	35.5
CLIPvisual	91.2	60.7	50.9	46.1	40.3	27.4
CLIPsemfeat-N	81.8	60.0	52.7	51.8	**58.8**	**40.1**
CLIPsemfeat-K	90.6	63.7	54.2	52.1	52.9	37.8

TABLE II: Matching success accuracy for baselines and visual-semantic models. The settings for same instance and different instance matching are as described in Sections IV-A, IV-B. Best performing approaches are bold-faced. For increasingly difficult matching settings, the performance of approaches based on purely visual features deteriorates rapidly, with the **CLIP-SemFeat** approaches outperforming all baselines in cross-instance settings.

model	Same- [%]	Cross-instance [%]
RandomGuess	15.7	15.3
ColourHist	69.6	13.7
AlexNet-S	59.0	19.4
ResNet50-S	57.7	37.9
ResNet50-G	**76.5**	55.6
CLIPvisual	72.9	49.8
CLIPsemfeat-N	70.1	74.2
CLIPdiscrete-N	59.0	**77.4**

TABLE III: Source to target image object accuracy for baselines and visual-semantic models across same-instance and different-instance robotic rearrangement scenes.

V. Experiments

A. Same-Instance Matching Results

We run all baselines and variants of our method across the APC same-instance matching settings (Section IV-A), with results given in the left side of Table II. Accuracy is reported as the total correct matches over the total number of possible correct matches, which we take to be the intersection of the labels of objects visible in any two considered scenes. We first note that all proposed matching models do indeed experience performance degradation as difficulty is increased. We further note the surprisingly competitive performance of colour histograms on the same-instance problem, suggesting the relatively simple visual nature of current robotics rearrangement challenges. We observe that the **ResNet50-S** spatial features perform best on the **Easy** APC setting, suggesting that when source and target views are very similar, spatial correspondences between features are useful. For the **Medium**, **Hard** and **Hardest** settings, the **ResNet50-G** global features, which contain no spatial information, perform better. We further note that, across the same-instance matching settings, ResNet50 features trained on ImageNet beat **CLIP-SemFeat** models. Intuition for this can be gained by analysing the results of **CLIPVisual**, which uses only the visual embedding of the CLIP model for matching, and which also consistently under-performs ResNet50 ImageNet features. This is likely because ImageNet models are trained with strong data augmentation, and thus explicitly learn features invariant to the kind of visual shifts observed in the same-instance matching setting. In contrast, CLIP is trained only with random-crop data augmentation, with invariances in the visual embedding learned only implicitly through the scale of the training data. Finally, it is notable that even in these same-instance settings, giving the CLIP model access to semantics boosts its performance except in the most trivial setting, and improves robustness under increasing difficulty versus purely visual approaches. When looking at the **APC-Easy** setting, **CLIP-SemFeat** models under-perform **ResNet50-G** by as much as 16%, while this is gap is reduced to just 3% for the **APC-Hardest** setting.

B. Text Prompt Engineering

Objects in the APC dataset are described by their product catalogue names. The descriptions are highly specific and not designed to be semantically discriminative. For instance, 82% of APC labels include a brand name - something that carries little intrinsic semantic value. Through K-way classification experiments, we investigate whether such untailored semantic text prompts are well understood by vision-language models such as CLIP, with results in Table I denoted **CLIP**. For comparison, we run K-way classification again using 'improved' prompts following Section III-A.2. The classification results using these labels are given by **CLIP+** in Table I. We find that **CLIP+** outperforms **CLIP** by 7.3% in top-1 accuracy. This result affirms the significance of the choice of semantic description used in CLIP zero-shot classification, and by extension our visual-semantic matching approaches. Finally, we examine the effect of ensembling over multiple text prompts for each class, which has been shown to be beneficial for CLIP-based classification [16]. We use all of the short descriptions in **CLIP+**, and format them into the prompts "*A picture of a* {...}", "*A picture of a* {...}, *a product*", "*A* {...}, *a product*", "{...}". This results in around 20 text prompts per class for ensembling (**CLIP++**). We see no further boost from prompt ensembling, and use the **CLIP+** prompts in our APC dataset matching experiments.

C. Cross-Instance Matching Results

We run all baselines and variants of our method across 20,000 matching scenarios drawn from the LVIS setting described in Section IV-B, and present results on the right of Table II. For text prompts for our visual-semantic matching methods, we use LVIS' object class names (mean length of 1.3 words), formatted as "*A picture of a* {...}". The LVIS setting enables matching up to 40 different object classes at once, and we run with $N = 8$, in line with the higher end of APC matching problems, and $N = 20$,

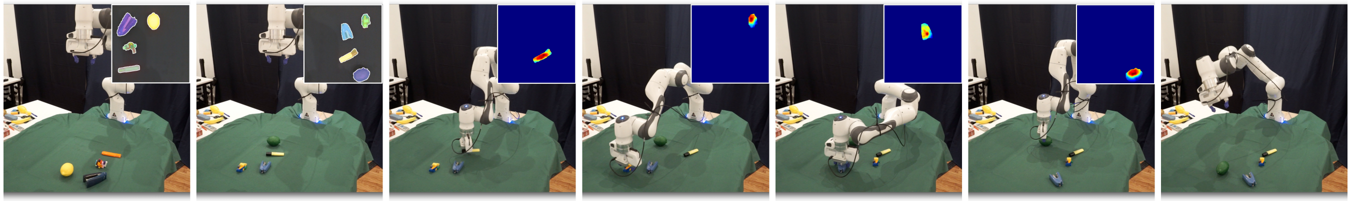

Fig. 3: Successful completion of a different-instance rearrangement task using **CLIP-SemFeat-N** with labels `{highlighter, stapler, lego blocks, citrus fruit}`. *Inset*: segmentation masks (leftmost frames) and masked GG-CNN grasp predictions. A video of this task is available at `https://youtu.be/9soVArIXJlM`.

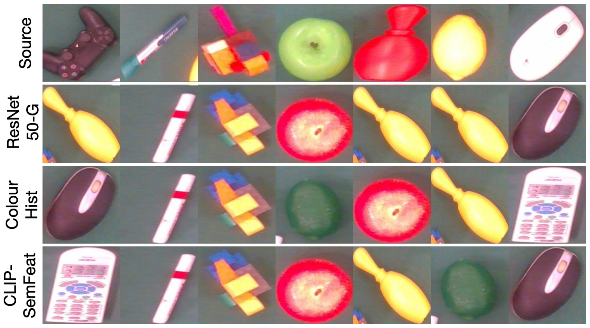

Fig. 4: Matching results in the real robot setting for three different matching models. Only **CLIP-SemFeat** is able to match all crops accurately.

to investigate the effect of a harder matching setting on matching performance. Average accuracies across methods are reported in Table II. **CLIP-SemFeat-K** uses all 40 object class names, while **CLIP-SemFeat-N** uses only those of the objects present. We note that *all* methods relying purely on visual descriptors perform worse in this setting when compared to the **APC-Hardest** results, while the **CLIP-SemFeat** methods achieve the highest accuracy on these cross-instance matching problems. These results confirm that purely visual descriptors struggle to perform good matching across different object instances, reinforcing the importance of leveraging semantics in this setting. The improvement in accuracy of **CLIP-SemFeat** methods between **APC-Hardest** and the 8-way LVIS experiments, despite the cross-instance shift, is helped by the clear semantic nature of the LVIS class labels, providing prompts for CLIP that are likely to be within its training distribution.

D. Real Robot Deployment

We assess the practical impact of visual-semantic matching across a number of multi-object, multi-step tabletop object rearrangement tasks with a Franka Emika Panda robot arm. As before, we consider both the same-instance and cross-instance matching setting. Critically, we show that in the latter case, we can conduct robotic object rearrangement that satisfies a goal image in which all object instances are substantially visually distinct to those in the scene, through the use of our semantically grounded matching method.

We collect a set of 25 household objects, and sample 20 scenes of objects at random, with 2 to 10 objects per scene. For each object set, we throw the objects into the robot's workspace, with semi-random placement: while we seek occasional occlusions, handling dense clutter is not the focus of this work. An RGB-D image is taken with a robot-wrist-mounted Intel RealSense D435i camera, and segmented with an instance segmentation network [12]. This is taken to be the goal image. We then conduct 3 further rearrangements of the scene to simulate different starting conditions, taking RGB-D images and producing segmentation masks of each. This gives us 60 source-target image pairs with 395 valid object matches. From a 2nd set of 'twinned' objects for assessing cross-instance matching, consisting of 10 *pairs* of household objects that are different instances of the same item, we produce 20 image pair problems with 2 to 10 objects per scene. Table III shows results for both settings. CLIPDISCRETE assigns labels to objects in the images independently, and takes same-labelled crops as matched. We use the GG-CNN for grasping [10], and write a mask-based collision planner for planning pick-and-place sequences. Figure 3 shows an example of a successful rearrangement task.

E. Qualitative Matching Model Comparisons

Figure 4 provides an example of matching results in the real-robot setting. Unsurprisingly, **Colour Histograms** fail when the colour of the source and target objects differ. For instance, the green apple is matched to the green citrus, while the yellow citrus is matched to the yellow bowling pin. The **ResNet50-G** features overcome this somewhat, matching mice and apples of different colours, exhibiting some semantic understanding (notably, both classes feature in the ImageNet training data). However, the **ResNet50-G** features are not reliably discriminative, and assign the yellow bowling pin in the target image to multiple source image crops. Finally, the CLIP model is most consistently able to leverage semantic information in source and target crops, matching correctly despite significantly different visual appearance.

VI. CONCLUSIONS

We examine the problem of object matching for robotic rearrangement tasks instructed with goal images, and characterise the deterioration of existing matching approaches as domain shift between source and target scene increases. We propose a novel approach to matching that makes use of semantic grounding, via a large pre-trained vision-language model, providing additional information about object similarity between scenes. We demonstrate, on both a large-scale dataset and a set of objects in the lab, that this approach enables successful matching even when the objects in source and target scenes are different instances. We integrate our approach as part of a robotic tabletop object rearrangement system that can leverage cross-instance goal images. We believe that our results motivate further exploration of semantics as a disambiguating factor in vision-based robotic manipulation tasks.

References

[1] F. Sadeghi, A. Toshev, E. Jang, and S. Levine, "Sim2Real Viewpoint Invariant Visual Servoing by Recurrent Control," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4691–4699.

[2] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, M. Savva, and H. Su, "Rearrangement: A Challenge for Embodied AI," Tech. Rep., 2020.

[3] A. Qureshi, A. Mousavian, C. Paxton, M. Yip, and D. Fox, "NeRP: Neural Rearrangement Planning for Unknown Objects," in *Robotics: Science and Systems*, 2021.

[4] O. Groth, C.-M. Hung, A. Vedaldi, and I. Posner, "Goal-Conditioned End-to-End Visuomotor Control for Versatile Skill Primitives," in *International Conference on Robotics and Automation (ICRA)*, 2021.

[5] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training Home Assistants to Rearrange their Habitat," pp. 1–32, 2021.

[6] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi, "Visual Room Rearrangement," in *CVPR*, 2021.

[7] Z. Liu, W. Liu, Y. Qin, F. Xiang, M. Gou, S. Xin, M. A. Roa, B. Calli, H. Su, Y. Sun, and P. Tan, "OCRTOC: A Cloud-Based Competition and Benchmark for Robotic Grasping and Manipulation," in *ICRA workshop*, 2021.

[8] M. Danielczuk, A. Mousavian, C. Eppner, and D. Fox, "Object Rearrangement Using Learned Implicit Collision Functions," in *International Conference on Robotics and Automation (ICRA)*, 2021.

[9] Y. Labbe, S. Zagoruyko, I. Kalevatykh, I. Laptev, J. Carpentier, M. Aubry, and J. Sivic, "Monte-Carlo Tree Search for Efficient Visually Guided Rearrangement Planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3715–3722, 2020.

[10] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 183–201, 2020.

[11] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies Between Affordance and Geometry: 6-DoF Grasp Detection via Implicit Representations," in *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, 2021.

[12] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, "Learning RGB-D Feature Embeddings for Unseen Object Instance Segmentation," no. CoRL, pp. 1–10, 2020.

[13] Z. Zhang, R. Newbury, K. He, S. Martin, G. Suddrey, J. Kwan, P. Corke, and A. Cosgun, "Tabletop Object Rearrangement: Team ACRV's Entry to OCRTOC," 2021.

[14] M. J. Swain and D. H. Ballard, "Color Indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[15] Y. Wu, O. P. Jones, M. Engelcke, and I. Posner, "APEX: Unsupervised, Object-Centric Scene Segmentation and Tracking for Robot Manipulation," in *IROS*, 2021.

[16] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," 2021.

[17] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "RLBench: The Robot Learning Benchmark & Learning Environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[18] H. Song, J. A. Haustein, W. Yuan, K. Hang, M. Y. Wang, D. Kragic, and J. A. Stork, "Multi-object rearrangement with monte carlo tree search: A case study on planar nonprehensile sorting," in *IEEE International Conference on Intelligent Robots and Systems*. Institute of Electrical and Electronics Engineers Inc., 2020, pp. 9433–9440.

[19] W. Yuan, J. A. Stork, D. Kragic, M. Y. Wang, and K. Hang, "Rearrangement with Nonprehensile Manipulation Using Deep Reinforcement Learning," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2018, pp. 270–277.

[20] H. Zhao, J. Guo, D. Liu, H. Wang, Yecan Yin, and Dongsheng Ge, "Team iRobotCNC's winning submission to OCRTOC 2020 competition," Tech. Rep., 2020.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[22] D. Held, S. Thrun, and S. Savarese, "Robust single-view instance recognition," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2152–2159.

[23] J.-P. Mercier, M. Garon, P. Giguere, and J.-F. Lalonde, "Deep Template-based Object Instance Detection," in *WACV*, 2021, pp. 1506–1515.

[24] J. P. Mercier, L. Trottier, P. Giguère, and B. Chaib-Draa, "Deep object ranking for template matching," in *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*. Institute of Electrical and Electronics Engineers Inc., 2017, pp. 734–742.

[25] P. Ammirato, C.-Y. Fu, M. Shvets, J. Kosecka, and A. C. Berg, "Target Driven Instance Detection," 2018.

[26] A. Zeng, S. Song, K. T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morena, P. Qu Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., 2018, pp. 3750–3757.

[27] M. Shridhar, L. Manuelli, and D. Fox, "CLIPort : What and Where Pathways for Robotic Manipulation," no. CoRL, pp. 1–24, 2021.

[28] H. Zhang, Y. Lu, C. Yu, D. Hsu, X. Lan, and N. Zheng, "INVIGORATE: Interactive Visual Grounding and Grasping in Clutter," in *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, 2021.

[29] O. Mees and W. Burgard, "Composing Pick-and-Place Tasks by Grounding Language," in *Springer Proceedings in Advanced Robotics*, 2021, vol. 19, pp. 491–501.

[30] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, and J. Tan, "Interactively Picking Real-World Objects with Unconstrained Spoken Language Instructions," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2018, pp. 3774–3781.

[31] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, "MAttNet: Modular Attention Network for Referring Expression Comprehension," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1307–1315.

[32] A. Zeng, K. T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2017, pp. 1386–1393.

[33] A. Gupta, P. Dollar, and R. Girshick, "LVIS: A Dataset for Large Vocabulary Instance Segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2019-June. IEEE, 2019, pp. 5351–5359.