

Reward Learning from Narrated Demonstrations

Hsiao-Yu Tung

Adam W. Harley

Liang-Kang Huang

Katerina Fragkiadaki

Carnegie Mellon University

5000 Forbes Ave, Pittsburgh, PA 15213

{htung, aharley, katef}@cs.cmu.edu, henrykang7177@gmail.com

Abstract

Humans effortlessly “program” one another by communicating goals and desires in natural language. In contrast, humans program robotic behaviours by indicating desired object locations and poses to be achieved [5], by providing RGB images of goal configurations [19], or supplying a demonstration to be imitated [17]. None of these methods generalize across environment variations, and they convey the goal in awkward technical terms. This work proposes joint learning of natural language grounding and instructable behavioural policies reinforced by perceptual detectors of natural language expressions, grounded to the sensory inputs of the robotic agent.

Our supervision is narrated visual demonstrations (NVD), which are visual demonstrations paired with verbal narration (as opposed to being silent). We introduce a dataset of NVD where teachers perform activities while describing them in detail. We map the teachers’ descriptions to perceptual reward detectors, and use them to train corresponding behavioural policies in simulation. We empirically show that our instructable agents (i) learn visual reward detectors using a small number of examples by exploiting hard negative mined configurations from demonstration dynamics, (ii) develop pick-and-place policies using learned visual reward detectors, (iii) benefit from object-factorized state representations that mimic the syntactic structure of natural language goal expressions, and (iv) can execute behaviours that involve novel objects in novel locations at test time, instructed by natural language.

1. Introduction

Currently, rewards or goals for behavioural policy learning are either manually coded by experts [42, 32, 31], or are learned from human supplied demonstrations (LfD, or inverse RL) [39, 40, 20, 54]. Manually coded rewards are hard to generalize across variations of the environment. Moreover, we often need a large number of demonstrations for the right reward function to be effectively communi-

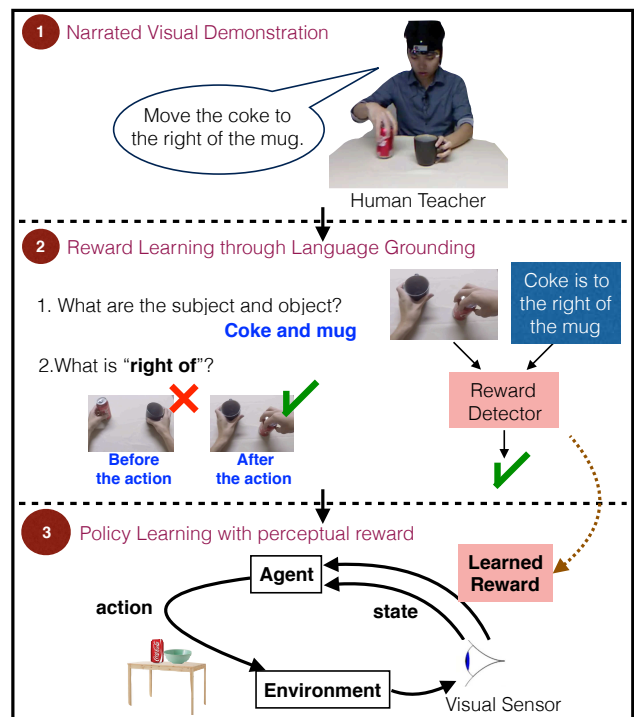


Figure 1. **Reward Learning from Narrated Demonstrations.** We begin with a narrated visual demonstration, prepared by a human (1). Our system then learns a spatial relationship detector from the visuals and audio (2). Finally, we use the learned detectors to train pick-and-place policies (3).

cated to the agent, invariant to distractors, accidental coincidences, view-dependent feature representations, speed of execution, etc. In contrast, humans effortlessly program each other’s behaviour **by conveying goals and desires in natural language**, e.g. “for a CVPR submission, the margin should be one inch on each side of the page”, or “while driving, make sure to keep a safe distance from the car in front of you.” Interestingly, in absence of natural language competence, *understanding the goal of a behaviour is often harder than learning the behaviour itself*. For example, although macaques are excellent tree climbers, incremen-

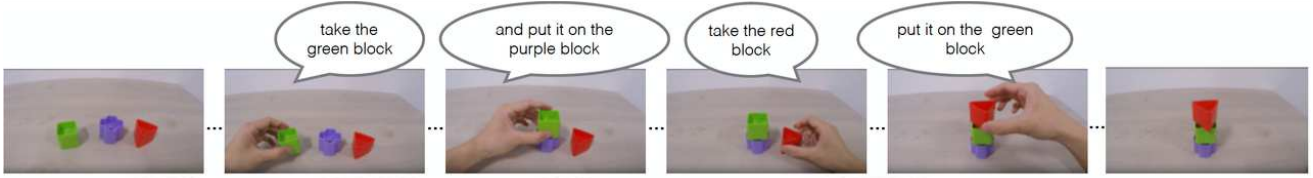


Figure 2. **Narrated visual demonstrations.** The teacher demonstrates activities and concurrently narrates them in natural language using a microphone. Many related tasks are demonstrated densely in time; temporal segmentation of the demonstration video into different tasks is easy based on natural language sentences.

tally training them to pick coconuts (with RL) is extremely laborious [34]. Humans, on the other hand, can easily understand the goal of “picking coconuts”, but are less capable of carrying it out.

This work introduces **instructable perceptual rewards**, namely, reward functions that can be both expressed in natural language and detected in the visual sensory input of the agent. It further proposes a framework for learning these rewards from **Narrated Visual Demonstrations** (NVD), which are visual demonstrations synchronized with natural language descriptions. Rather than struggling to discover essential goals of human behaviour from a large number of *silent* visual demonstrations, we instead consider *narrated* visual demonstrations, where narrations describe actions being taken, objects involved, and goals achieved, as shown in Figure 2. Given a set of NVDs, we first learn to ground natural language utterances that express activity goals—rewards, *e.g.*, “coca cola on top of the book,” to modular neural visual detectors. We then use such visual detectors to reinforcement learn policies that achieve the corresponding goals (see also Figure 1).

Narrated visual demonstrations are more data-efficient than their silent counterparts. We empirically show that learning instructable perceptual rewards and corresponding policies from NVDs results in data reduction for both reward and policy learning. This reduction comes from (i) leveraging large-scale annotated static image datasets [29] of objects and visual relationships to help ground natural language goal descriptions, (ii) demonstration dynamics, where similar objects appear with different attributes or relationships in consecutive demonstrated tasks, which forces our reward detectors to focus on the **temporal transformation of such arrangements/attributes**, as opposed to object detection and recognition, and, (iii) object-factorized state representations as input to reward detectors and policy networks, mimicking the syntax of natural language descriptions.

Collecting narrated visual demonstrations is scalable. We collect a dataset of pick-and-place activities using cameras and microphones mounted on demonstrators (human teachers) that perform activities while verbally narrating them (see Figure 2). Automated speech recognizers map the narrations to transcripts temporally synchronized with

the visual demonstrations. Each video contains multiple, diverse demonstrations, proceeding one another closely in time. Temporal segmentation of sequential demonstrations [23] is easily obtained by considering the segmentation of the transcript into verbal phrases; this alleviates the current need for demonstrations to concern a single isolated task at a time [23]. In terms of detail, deliberate demonstration and verbal narration is more scalable than post-hoc captioning [45], and allows natural language descriptions that are dense in time, without overwhelming the demonstrator. The videos in our dataset are instructional in nature, similar to instructional videos on YouTube [26, 3]. While Youtube videos target on audience with advanced language grounding capabilities, our dataset instead attempts to teach such natural language grounding, alongside the demonstrated behaviours. To the best of our knowledge, no previous work has considered narrated videos for learning rewards and policies for the demonstrated actions.

In summary, our contributions are:

- We introduce instructable perceptual rewards as modular visual detectors of natural language expressions of activity goals, and show how to learn them from few NVDs by exploiting demonstration dynamics for effective hard negative mining.
- We introduce a dataset of NVDs of daily activities. We show that pairing visual demonstration with natural language narration permits scaling up the collection of visual demonstrations, which can now be dense in time and depict diverse tasks, as opposed to being structured, isolated in time, and depicting a single task.
- We demonstrate that our agent effectively learns instructable policies using *noisy* instructable perceptual reward detectors, and can execute novel behaviours at test time, exploiting compositionality of natural language.
- We show that object-factorized state representations for our policy network generalize better than frame-centric RGB input.

2. Related Work

In the absence of a manipulation language, previous works convey goals using RGB images [19], demonstrating the desired activity itself [17], supplying desired 3D poses of the objects and end-effectors in a particular scene [6], or assuming that there is only one behaviour that can be requested [53]. This work proposes expressing goals in natural language, and builds corresponding perceptual detectors that can drive reinforcement learning for policies.

Mapping instructions to actions Numerous works have proposed learning a mapping from instructions to high level action sequences of the agent. For instance, paired examples of instruction and action sequences have been collected through Amazon Mechanical Turk [8, 36, 37, 52]. Other works attempt to learn such mapping using reinforcement learning, from pairs of instructions with desired final goal configurations [10, 35]. These models execute the predicted action sequences and evaluate whether the desired goal state is reached. Most approaches consider action sequences to be given in the task space of the agent. Instead, we consider third-person demonstrations and narrations, where the automated visual recognition needs to infer the locations of objects and their spatial configurations. We train our agent with reinforcement learning using perceptual detectors of the natural language goal expressions, as opposed to direct imitation of the corresponding action sequences.

Visual imitation learning Visual imitation learning (VIL) considers the problem of acquiring skills by observing visual demonstrations [50]. It requires inference of the “reward” (i.e., the goal of the behaviour) that the agent will attempt to match by self-practice, and adapting the demonstrations to its own degrees of freedom and workspace. Numerous works circumvent the difficult visual perception problem in VIL using special instrumentation of the environment to read off object and hand poses during video demonstrations [30], or use rewards based on **known goal 3D object configurations**. A notable exception is the work of Sermanet, Xu, and Levine [48], which learned perceptual rewards for a pouring task, using a large number of visual demonstrations. In this work, we instead propose **narrated** visual demonstrations for joint learning of natural language grounding and reward detectors. Natural language casts attention to the relevant parts of the video (e.g., the relevant objects), and facilitates the mapping of natural language descriptions to visual reward detectors. At test time, we can easily program novel behaviours by composing novel natural language goal descriptions.

Perceptually-grounded natural language Language grounding has recently attracted a lot of attention, with

the introduction of large-scale image captioning, video summarization [46, 18] and visual question answering datasets [7, 51]. Captioning models describe images and videos using natural language sentences [15] and visual question answering models answer queries about an image [4]. Such vision/language models are supervised by image captions or question/answer pairs collected from AMTurkers [45], subtitles from movies [44], or movie descriptions for the blind [43]. This paper has an orthogonal goal to the aforementioned works: we are interested in learning to ground natural language descriptions of goal configurations to visual input, and use this mapping as a reward detector for policy search. **This replaces manually-coded rewards with natural language instructions.**

Object-factorized state representations The recent work of Kansky et al. [28] showed how object-factorized state representations and dynamics can generalize across environmental variations, in contrast to frame-centric policies. In that work, it was assumed that the object identities were known beforehand. Here, we use natural language as weak supervision to focus attention to relevant objects, and use object detectors to learn reward configurations, and also during policy training and testing, to supply object-factorized states as input to our policy network. Other works [21, 9] have considered object-centric predictive models of motion under close-by interactions, and showed they generalize better than frame-centric models.

3. Instructable Reward and Policy Learning from Narrated Visual Demonstrations

3.1. Collecting Narrated Visual Demonstrations

We collect narrated visual demonstrations using GoPro cameras and microphones mounted on the head of the human demonstrator. The demonstrator names objects in the scene, describes their relationships, indicates the activities performed, explains the outcomes, and gesticulates deliberately so as to guide the learner towards the correct interpretation of the natural language description. Verbal narrations are automatically transcribed into textual descriptions using the Google speech recognition API [24]. Mistakes of the speech recognizer are rare and are corrected by hand. The sync of the narration to the video, along with the present-tense descriptions, provide a natural alignment of the semantic content to the visual stream, e.g., “*I am placing the cup on the opening of the bottle*”. Consecutive demonstrations are temporally segmented using their alignment to natural language utterances. This convenient segmentation method is only possible with narrated (rather than silent) demonstrations. In terms of human effort, the scalability of verbal narrations far surpasses annotation methods considered in previous works, such as video post-transcription

[45] or scene graph annotations [29]. Each video is between three and five minutes long and contains 14 to 30 individual demonstrations of short activities. We have thus far collected two hours of densely annotated videos. This paper uses the pick-and-place activities of the dataset (around 10 minutes in total) to learn reward detectors and train corresponding pick-and-place policies in simulation. Many more diverse activities are contained in the dataset, which we will make publicly available. We are not aware of a dataset of paired videos and natural language descriptions that addresses natural language grounding for skill policy learning, which is a gap our work attempts to cover.

3.2. Learning Instructable Perceptual Rewards through Natural Language Grounding

We learn visual reward detectors by grounding natural language descriptions of goals of pick-and-place activities (e.g., “the coke can is on top of the book”) to modular neural programs that take an image and description as input, and output a score of how well the image matches the description. These reward detectors are used to train pick-and-place policies to achieve the configuration instructed by the natural language expression.

Our visual detectors combine object detector modules and pairwise relation modules, assembled based on the syntactic structure of the natural language description, provided by a syntactic parser [12]. The architecture is depicted in Figure 3. It is comprised of two object detectors for the subject and object in the natural language expression, and a relation neural module for scoring their spatial configuration.

The object detectors build upon the state-of-the-art faster RCNN architecture [27], and have been pretrained in Visual Genome [29] and COCO datasets [13] to detect objects from 3000 categories. We use the Stanford syntactic parser [12] to parse the natural language expression into subject and object strings, and use the appropriate outputs of the object detectors to localize the mentioned object categories in the image. If the objects do not have a high enough detection score, we discard the corresponding frame. The relation module takes as input (i) a word embedding of the spatial relationship v_s , computed using a weighted average of the hidden states of a Bidirectional LSTM (BiLSTM) over the natural language expression’s words, where the weight distribution is predicted by the same BiLSTM, and, (ii) spatial locations of the object and subject, encoded as normalized pixel coordinates, and the width and height of the detected bounding boxes. This module outputs a score for the corresponding spatial relationship, as shown in Figure 3-left. The relation module is pre-trained to localize referential expressions in the Visual Genome image dataset [29], as part of the model of [25]. Although a referential expression, such as “the orange in the bowl”, is not identical in meaning

to a description, such as “the orange is in the bowl”, or to a desired post-condition, such as “the orange should be in the bowl”, in practice their learned embeddings are similar.

Our model is a variation of the referential expression detector of [25]. The difference between the two is that, instead of object detectors, their model uses two localization modules, which, take as input a weighted average of the hidden state of the BiLSTM for the subject and object, and the visual features aggregated within a bounding box proposal, and score the probability that the bounding box proposal captures the referred subject or object, respectively. Their expression detector sums the scores of the two localization modules and the relation module to score how well the two considered object proposals convey the referential expression. We instead use object detectors that take visual features as input, and predict an object category within a predefined set of categories. Linguistic variability can be handled by considering the inner product of the word embedding of the detectable object categories with the word embedding of the subject and object of the utterance, considered in the model of [25]. Thanks to its modularity, the detector generalizes better than a monolithic network trained to map a single frame or bounding box to spatial configurations scores.

Weakly-supervised metric learning with hard negative mining

In our NVD dataset, video frames are paired with corresponding pieces of transcript, as generated by the speech recognizer. We temporally segment a video sequence into individual demonstrations whenever two consecutive natural language utterances are different. Our reward detector is trained from such automatically aligned utterance-frame pairs, the same utterance covers all frames of the demonstration. We consider only the frames paired with exactly one natural language utterance and finetune the relation module of our reward detector using metric learning. Specifically, we ask our relation module to score higher in the frames paired with the considered utterance at the end of each demonstration (input to our relation module) and lower at the frames in the beginning of the demonstration.

Let f_s^k and f_o^k be the spatial features (normalized pixel coordinates, width and height) of the detected boxes for the subject and the object of the paired natural language utterance in frame k , respectively, and let v_s denote the relation embedding vector produced by the BiLSTM, given natural language expression s . For each video segment paired with natural language utterance s , let X_s^- denotes the indices for the first few frames (negative examples) and X_s^+ denotes the indices of the last few frames, (positive examples for the goal configuration), as shown in Figure 3 right. Then, the contrastive loss function for each video segment reads:

$$\mathcal{L}(s) = \sum_{k \in X_s^+} \sum_{m \in X_s^-} \max(0, \mathcal{S}(v_s, f_s^k, f_o^k) - \mathcal{S}(v_s, f_s^m, f_o^m)),$$

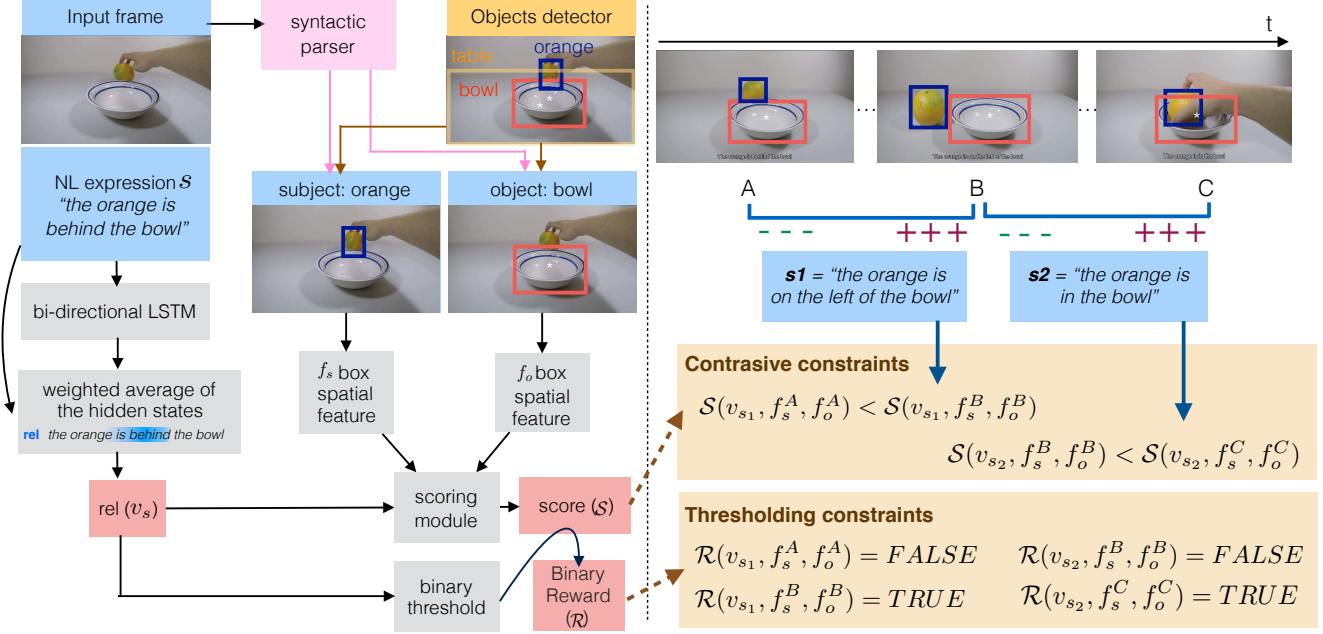


Figure 3. **Left: Visual detection of natural language spatial expressions** comprised of two object detectors and a relation module that computes a relation word embedding vector and given the spatial features extracted from the detected boxes, outputs a score for the spatial configuration. The score can be further transformed into a binary reward using predicted threshold. **Right: Hard negative mining** using spatial configurations of adjacent in time video frames, demonstrating related natural language expressions. Related in time demonstrations provide hard negative examples to our relation module, free of static image biases, that allow it to improve from very few examples.

where $\mathcal{S}(v_s, f_s^k, f_o^k)$ is the score output by our relation neural module in frame k . The score is further threshold into a binary reward $\mathcal{R}(v_s, f_s^k, f_o^k) \in \{0, 1\}$ indicating a hard decision on whether the visual inputs match with the natural language utterance. The threshold is predicted from a two-layer neural network that takes as input the relation embedding. This threshold predicting branch is trained using standard cross entropy loss for binary classification.

Natural language grounding from narrated visual demonstrations benefits from **hard negative mining** of the demonstrated natural language concepts: example frames of spatial configurations that portray the **same pair of objects, but in different spatial configurations**. This characteristic comes for free from the way people demonstrate concepts: as suggested by psychologists [49], related or opposite concepts are demonstrated/explained in temporal sequence, which much helps their **disentanglement** via providing hard negative examples to the learner. In contrast, in static images, due to photographic biases, many configurations come at stylized poses, with stylized objects, which makes it hard for the learner to disentangle the individual characteristics of the relation. As a result, even with a handful of video demonstrations (14), our relation module much improves over the pretrained model of [25], as we show in Section 4, while using similar unsupervised metric learning losses.

3.3. Policy learning with perceptual rewards

We use the learned visual reward detectors to train pick-and-place policies in simulation, replacing manually coded rewards, typically used in previous works [47, 5].

Object-factorized state representations Our reward detectors decompose the scoring of a spatial referential expression over an object-centric graph, where nodes represent object detections and edges represent their spatial relationships. We use the same object-factorized input for our policy network and show empirically that it generalizes better than frame-centric representations considered in previous works [41], where the whole frame is provided as input to a policy network. Some recent works do also consider object-centric input [14, 16]. Unlike these works, however, we additionally distinguish the roles of the objects in the scene (by mapping the subject and object in our natural language description to corresponding box hypotheses), making our object-factorized state **ordered**, as opposed to unordered.

Reward shaping via analysis-by-synthesis Our learned reward detectors from Section 3.2 take as input an RGB image and a spatial natural language expression and output a binary score, \mathcal{R} , of whether the image matches the spatial configuration. Model-free policy search with binary

rewards has notoriously high sample complexity due to the lack of informative gradients for the overwhelming majority of the sampled actions [22]. Efficient policy search requires *shaped rewards*, either explicitly [33], or more recently, implicitly [5], by encoding the goal configuration in a *continuous* space where similarity can be measured against alternative goals achieved during training.

If we were able to visually picture the desired 3D object configuration to be achieved by our pick-and-place policies, then Euclidean distances to the pictured objects would provide an effective (approximate) shaping of the true rewards. We do so using analysis-by-synthesis, where our trained detector is used to select or discard sampled hypotheses. Given an initial configuration of two objects that we are supposed to manipulate towards a desired configuration, we seek a physically-plausible 3D object configuration which renders to an image that scores high with our corresponding reward detector. Using the subject and object categories extracted from the natural language utterance, we retrieve corresponding 3D models from external 3D databases (3D Shapenet [11] and 3D Warehouse [2]) and import them in a physics simulator (Bullet). We sample 3D locations for the objects, render the scene and evaluate the score of our detector. Note that since we know the object identities, the relation module is the only one that needs to be considered for this scoring. We pick the highest scoring 3D configuration as our goal configuration. It is used at training time to provide effective shaping using 3D Euclidean distances between desired and current object locations and drastically reduces the number of samples needed for policy learning. However, our policy network takes 2D bounding box information as input, and does not need any 3D lifting, but rather operates reactively given the RGB images.

4. Experiments

We evaluate the accuracy of our reward detectors and their effectiveness for learning instructable pick-and-place policies in simulation, in place of manually coded rewards. Our experiments aim to answer the following questions:

1. How much does weak supervision from narrated visual demonstrations benefit the grounding of natural language spatial expressions, over a baseline of strongly-supervised labelled (static) image datasets?
2. How does the accuracy of learned—as opposed to manually coded—reward detectors affect the training speed and accuracy of the corresponding reinforcement-learned policies?
3. How do object-factorized policy networks compare to their frame-centric counter-parts?
4. How much does reward shaping via analysis-by-synthesis help over binary rewards for efficient policy

	in	behind	left	right	avg.
Pretrained	0.89	0.43	0.35	0.35	0.51
RandomNeg	0.50	0.50	0.50	0.50	0.50
HardNeg	0.95	0.96	0.88	0.88	0.92

Table 1. **Classification accuracy** of visual reward detectors of natural language spatial expressions trained in static images (*pre-trained*), finetuned with images using randomly selected negative examples (*RandomNeg*), finetuned with videos using hard mining negative examples (*HardNeg*) for various spatial relations.

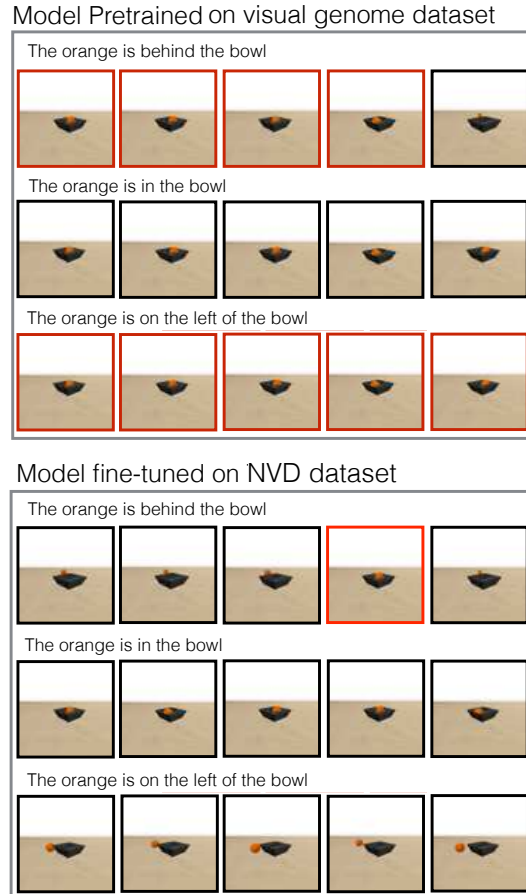


Figure 4. **Reward detectors** trained on static images alone (*top*) and on static images and narrated video demonstrations (*bottom*). We show the five highest scoring images for the two models for three spatial configurations. Red borders indicate incorrect detections. Video demonstrations improve visual detection of natural language expressions.

search?

4.1. Visual detection of natural language expressions

We generate a synthetic benchmark with 100 images for each spatial relationship. The relationships we consider are *in*, *behind*, *left*, and *right*. Each set of 100 images has 50

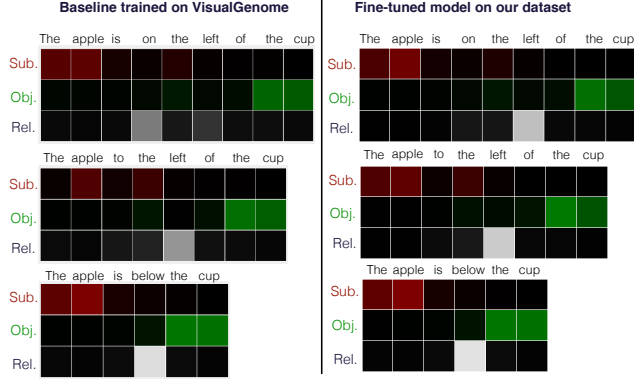


Figure 5. BiLSTM attention weights on language representation on unseen natural language descriptions. The detector trained from video demonstrations places weight on more informative keywords and generalizes to unseen sentences.

positive and 50 negative images. Ground truth annotations are generated by a hard-coded function in the simulator. In Table 1, we show classification error of the learned visual detectors. We compare the reward detector described in Section 3.2 trained on Visual Genome and finetuned on the video demonstration dataset, against the network of Hu et al. [25] trained on Visual Genome [29].

In Figure 4, we show the top retrieved images in a pool of 75 images that depict diverse spatial configurations of the same two objects (orange and bowl) using the (unthresholded) scores of our detectors. In both the classification task and the retrieval task, finetuning in our small video dataset helps the detector, despite using only 14 demonstration videos.

Finetuning in our NVD dataset clearly improves upon the pretrained model. Our video demonstrations often show **multiple spatial configurations of the same pair of objects**, and the data therefore have less biases regarding configuration-category correlations than static images. We further compare the hard negative mining from our NVD dataset against random sampling for negative examples from Visual Genome [29] in Table 1. Hard negative mining in NVD helps over random negative examples from the static image dataset (random in the absence of any information for sampling more informative negative examples).

In Figure 5, we visualize BiLSTM attention weights over the hidden states of the language representation from the pretrained and finetuned model. The finetuned model is placing weights on more informative keywords for relations, e.g., “right” and “left”, and is able to generalize to unseen (novel) natural language descriptions. Despite the fact that our model does not use the word embedding of the object or the subject, those also improve through the gradients on the relationship. In Figure 6, we show detector scores on real video sequences.

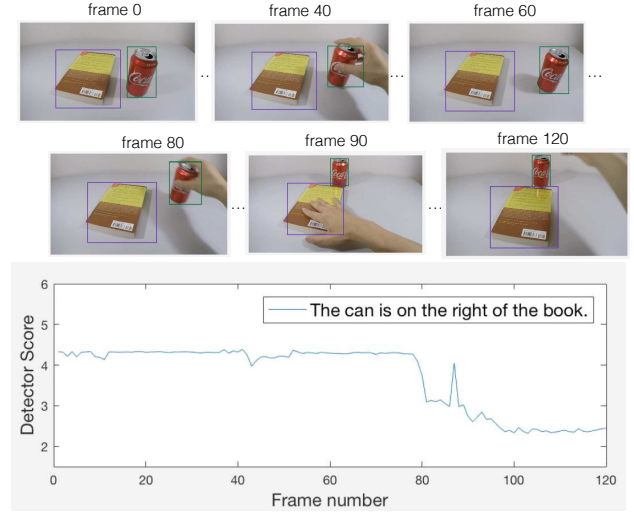


Figure 6. Reward detection on real test videos.

4.2. Policy learning with Perceptual Reward Detectors

We use our learned detectors to train instructable pick-and-place policies in the Bullet physics simulator [1]. Our policy always starts by grasping the subject of the natural language utterance as detected by our object detector. We use deep Q learning [38] over a discrete action set of {‘move forward,’ ‘move backward,’ ‘move right,’ ‘move left’} to learn a model-free policy that moves the end-effector of the Kuka IIWA robotic arm so that after an episode length of $T = 5$ action steps, the gripper opens, the object is released, and the desired configuration is achieved. Our policy network is a convolutional neural network that takes an RGB frame as input and produces a distribution over our action set. We will call this policy network RGBPolicyNet, to distinguish it from ObjectPolicyNet, which takes the spatial configuration of two objects, instead of the RGB.

Implementation details RGBPolicyNet has 5 convolutional layers and 3 fully connected layers with filter size 5×5 (stride 2), 5×5 (stride 2), 3×3 (stride 1), 5×5 (stride 2), and 3×3 (stride 1), respectively. Channel sizes are set to 32, 32, 32, 16, and 16, respectively. We use ReLU as the activation function. To reduce memory usage, we shrink the input image to 72×72 . The ObjectPolicyNet consists of three fully connected layers with size of 512, 512, and 4. We use ReLU as activation function after the first two layers. We train both networks starting from random weights using the Adam optimizer and learning rate of 0.001. The batch size in both models is set to 512. In each episode (trial), with exploration rate α , DQN takes randomly-selected actions with probability α and the action with highest score with probability $(1 - \alpha)$. The exploration

rate for DQN training is set to 0.8 and decays with the rate of 0.1 every 1000 action selections. For every five action selections, we take one gradient descent step for the DQN.

Task The task is putting an object inside a container. The containers are always facing up and initialized in a $15cm \times 40cm$ region with randomly selected orientation. The size of the container is roughly $5cm \times 5cm$. The subject indicated by our parser is initialized to be grasped by the gripper and hanging $20cm$ above the table.

Reward shaping We found that binary (oracle) rewards were not able to train successful policies with episode length larger than $T = 2$. When shaped rewards are combined with binary rewards, in terms of Euclidean distance between current and desired object 3D locations, effective policies were learned even when starting far away from the desired end-effector position. Thus, all our policy learning results in this section are obtained by combining (i) oracle shaping with oracle binary rewards (R_{gt}), or (ii) predicted shaping using analysis-by-synthesis with predicted binary rewards from our learned reward detectors (R_d).

Noisy rewards We show in Figure 7 plots of test policy accuracy against the number of episodes for RGBPolicyNet and ObjectPolicyNet using (i) oracle rewards (R_{gt}), and (ii) learned rewards (R_d) for the instruction “put the orange inside the bowl”. In a synthetic dataset of balanced successful and unsuccessful configurations, our reward detector has a classification accuracy of 95%. Table 2 shows that policy learning from noisy visual rewards for ObjectPolicyNet has 8% lower training accuracy, and much lower test performance than a policy trained with oracle rewards.

RGBPolicyNet is not strongly affected by whether the rewards are provided by an oracle or predicted by perception.

Object-factorized state representations In Figure 7 and in Table 2, we compare RGBPolicyNet and ObjectPolicyNet in their performance on seen and unseen objects. RGBPolicyNet does considerably worse, especially on unseen objects. RGBPolicyNet does not have a way to generalize to new object appearances at test time. Its worse performance during training can be explained as underfitting. It is severely hurt by resolution, since we wildly vary the configuration of the two objects during training.

5. Conclusion

In this work we introduce a paradigm for learning instructable pick-and-place policies through reinforcement from perceptual reward detectors trained through ground-narrations in narrated visual demonstrations. We show

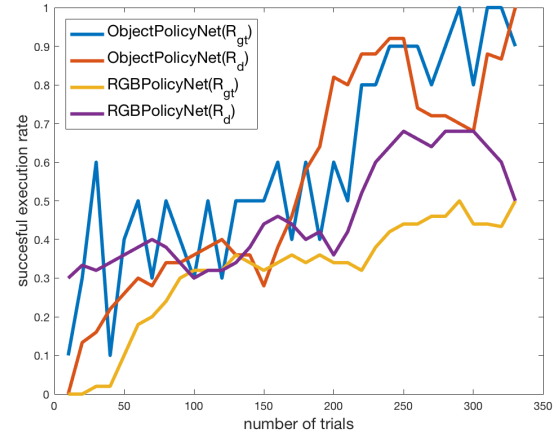


Figure 7. Policy learning with/wo noisy rewards, with/wo object-factorized input.

	accuracy (seen objects)	accuracy (unseen objects)
ObjectPolicyNet(R_{gt})	0.96	0.78
ObjectPolicyNet(R_d)	0.88	0.50
RGBPolicyNet(R_{gt})	0.71	0.27
RGBPolicyNet(R_d)	0.71	0.40

Table 2. Train/test policy accuracy (% of successful trials) for learning the task “place objects inside containers.” We consider two different policy network structures: (i) object bounding boxes and their spatial features as input (ObjectPolicyNet), and (ii) RGB image as input (RGBPolicyNet). We compare policies learned by manually-coded rewards in the simulator (R_{gt}) and by our learned reward detector (R_d). We compare policies on objects seen during training (but in novel positions), and on novel objects.

how the accuracy of the reward detectors affects the accuracy of the learned policies, and how object-factorized state representations that follow the syntactic structure of natural language help generalization of rewards and policies to novel scenes. We further show how goals instructed in natural language allow the description of novel goals and programming of corresponding novel behaviours at test time. Future work involves scaling up the vocabulary acquired for describing goals of activities, and also the corresponding skill library. Additionally, the training currently done in simulation can be done on a robotic platform. Finally, we plan to use more of the narrated demonstrations, rather than merely the final goal configurations.

Acknowledgements The authors would like to thank Hsiao-Wei Tung, Samuel Pepose, Ishu Garg, Medha Potluri, and Kanthashree Mysore Sathyendra for contributing to the NVD Dataset.

References

- [1] <http://bulletphysics.org/wordpress/>. 7
- [2] <https://3dwarehouse.sketchup.com/index.html>. 6
- [3] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2
- [4] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2016. 3
- [5] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. *CoRR*, abs/1707.01495, 2017. 1, 5, 6
- [6] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. *CoRR*, abs/1707.01495, 2017. 3
- [7] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 3
- [8] Y. Artzi and L. Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *TACL*, 1:49–62, 2013. 3
- [9] P. W. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *CoRR*, abs/1612.00222, 2016. 3
- [10] S. R. K. Branavan, H. Chen, L. S. Zettlemoyer, and R. Barzilay. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL ’09, pages 82–90, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. 3
- [11] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 6
- [12] D. Chen and C. D. Manning. A fast and accurate dependency parser using neural networks. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 4
- [13] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015. 4
- [14] C. Devin, P. Abbeel, T. Darrell, and S. Levine. Deep object-centric representations for generalizable robot learning. *CoRR*, abs/1708.04225, 2017. 5
- [15] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell. Language models for image captioning: The quirks and what works. *arXiv preprint arXiv:1505.01809*, 2015. 3
- [16] C. Diuk, A. Cohen, and M. L. Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th International Conference on Machine Learning, ICML ’08*, pages 240–247, New York, NY, USA, 2008. ACM. 5
- [17] Y. Duan, M. Andrychowicz, B. C. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba. One-shot imitation learning. *CoRR*, abs/1703.07326, 2017. 1, 3
- [18] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015. 3
- [19] C. Finn and S. Levine. Deep visual foresight for planning robot motion. *CoRR*, abs/1610.00696, 2016. 1, 3
- [20] C. Finn, S. Levine, and P. Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. *CoRR*, abs/1603.00448, 2016. 1
- [21] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik. Learning visual predictive models of physics for playing billiards. *CoRR*, abs/1511.07404, 2015. 3
- [22] X. Guo, S. Singh, H. Lee, R. L. Lewis, and X. Wang. Deep learning for real-time atari game play using offline monte-carlo tree search planning. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3338–3346. Curran Associates, Inc., 2014. 6
- [23] K. Hausman, Y. Chebotar, S. Schaal, G. S. Sukhatme, and J. J. Lim. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. *CoRR*, abs/1705.10479, 2017. 2
- [24] <https://cloud.google.com/speech/>. 3
- [25] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. 11 2016. 4, 5, 7
- [26] D. Huang, J. J. Lim, F. Li, and J. C. Niebles. Unsupervised visual-linguistic reference resolution in instructional videos. *CoRR*, abs/1703.02521, 2017. 2
- [27] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016. 4
- [28] K. Kanksy, T. Silver, D. A. Mély, M. Eldawy, M. Lázaro-Gredilla, X. Lou, N. Dorfman, S. Sidor, D. S. Phoenix, and D. George. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. *CoRR*, abs/1706.04317, 2017. 3
- [29] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalanditis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. 2, 4, 7
- [30] O. Kroemer and G. S. Sukhatme. Learning relevant features for manipulation skills using meta-level priors. *CoRR*, abs/1605.04439, 2016. 3
- [31] I. Lenz, R. Knepper, and A. Saxena. Deepmpc: Learning deep latent features for model predictive control. In *RSS*, 2015. 1

- [32] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *CoRR*, abs/1504.00702, 2015. [1](#)
- [33] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015. [6](#)
- [34] G. Lupyan and B. Bergen. How language programs the mind. In *Topics in Cognitive Science*, 2015. [2](#)
- [35] D. K. Misra, J. Langford, and Y. Artzi. Mapping instructions and visual observations to actions with reinforcement learning. *CoRR*, abs/1704.08795, 2017. [3](#)
- [36] D. K. Misra, J. Sung, K. Lee, and A. Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014. [3](#)
- [37] D. K. Misra, K. Tao, P. Liang, and A. Saxena. Environment-driven lexicon induction for high-level instructions. In *ACL*, 2015. [3](#)
- [38] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. [7](#)
- [39] K. Mülling, J. Kober, O. Kroemer, and J. Peters. Learning to select and generalize striking movements in robot table tennis. *International Journal of Robotics Research*, 32(3):263–279, 2013. [1](#)
- [40] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML ’00, pages 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. [1](#)
- [41] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel. Asymmetric actor critic for image-based robot learning. *CoRR*, abs/1710.06542, 2017. [5](#)
- [42] A. Rajeswaran, V. Kumar, A. Gupta, J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *CoRR*, abs/1709.10087, 2017. [1](#)
- [43] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [3](#)
- [44] A. Rohrbach, M. Rohrbach, S. Tang, S. J. Oh, and B. Schiele. Generating descriptions with grounded and co-referenced people. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [3](#)
- [45] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012. [2](#), [3](#), [4](#)
- [46] M. Rohrbach, Q. Wei, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. [3](#)
- [47] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In D. Blei and F. Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897. JMLR Workshop and Conference Proceedings, 2015. [5](#)
- [48] P. Sermanet, K. Xu, and S. Levine. Unsupervised perceptual rewards for imitation learning. *CoRR*, abs/1612.06699, 2016. [3](#)
- [49] L. Smith and M. Gasser. The development of embodied cognition: Six lessons from babies. *Artificial Life*, 11(1-2):13–29, 2005. [5](#)
- [50] B. C. Stadie, P. Abbeel, and I. Sutskever. Third-person imitation learning. *CoRR*, abs/1703.01703, 2017. [3](#)
- [51] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urta-sun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [3](#)
- [52] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1507–1514, San Francisco, CA, August 2011. [3](#)
- [53] A. Yahya, A. Li, M. Kalakrishnan, Y. Chebotar, and S. Levine. Collective robot reinforcement learning with distributed asynchronous guided policy search. 2017. [3](#)
- [54] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. AAAI*, pages 1433–1438, 2008. [1](#)