A Contribution Statement

Kuang-Huei Lee: Proposed the project initially, implemented PI-QT-Opt, experimented with SayCan settings both in sim and real, contributed to paper writing, and led the project in general.

Ted Xiao: Implemented simulation experiments, contributed to paper writing.

Adrian Li: Designed, implemented and experimented with different options of creating the target conditioning mask for instance grasping.

Paul Wohlhart: Designed, implemented and experimented with the neural network structure for target mask conditioned instance grasping.

Ian Fischer: Helped design ablation and analysis, contributed to paper writing and editing.

Yao Lu: Implemented PI-QT-Opt, tuned and experimented with SayCan settings both in sim and real, contributed to paper writing.

B Implementation Details

B.1 The Predictive Information Auxiliary Loss

Here, we describe more details on the predictive information auxiliary loss introduced in Section 3.1. CEB [49] and InfoNCE [50] require two encoder distributions $e(z|x)$ and $b(z|y)$. Fischer [49] defines $e(z|x)$ to be the *forward encoder* from which the representation z is sampled and $b(z|y)$ to be a variational *backward encoder* that approximates the unknown density $p(z|y) = \int dx\, \frac{p(x,y,z)}{p(y)}$. In this work, we follow [51] to choose $e(z|x)$ and $b(z|y)$ to be parameterized by von Mises-Fisher (vMF) distributions, which empirically yields good performance in learning self-supervised visual representations.

The von Mises-Fisher is a distribution on the $(n-1)$-dimensional hypersphere. The probability density function is given by $f_n(z, \mu, \kappa) = C_n(\kappa)\exp(\kappa\mu^T z)$, where μ and κ are the mean direction and concentration parameter respectively. We assume κ is a constant. The normalization term $C_n(\kappa)$ is a function of κ and equal to $\frac{\kappa^{n/2-1}}{(2\pi)^{n/2} I_{n/2-1}(\kappa)}$, where I_v denotes the modified Bessel function of the first kind at order v.

As shown in [51], when the forward encoder concentration parameter κ_e approaches infinity, $\kappa_e \to \infty$, $e(z|x)$ becomes a spherical delta distribution and InfoNCE that uses vMF distributions reduces to the commonly used deterministic form of InfoNCE with cosine similarity as its distance function.

In our implementation, we parameterize $e(z|x)$ and $b(z|y)$ as follows. We select $\kappa_e = 8192$ and $\kappa_b = 7$. μ_e is a 64d vector coming from an MLP with two 512d hidden layers on top of a convolution network shared with the online Q-network. Similarly, μ_b is a 64d vector coming from an MLP with two 512d hidden layers on top of a convolution network shared with the lagged target Q-network, which is not updated using gradients. β in Equation (2) is a Lagrange multiplier that controls the strength of information compression. We choose $\beta = 0.01$.

The predictive information auxiliary objective (Equation (2)) and the Q-learning loss (Equation (3)) are weighted combined and learned with the same optimizer, with 1.0 on the Q-learning loss and 0.01 on the CEB loss.

B.2 Implementation Details of Multi-Task Context Conditioning

We introduced two implementations of task context conditioning for multi-task learning in Section 3.2: image-based and language-based. In this section, we will dive into more details about these two implementations.

Image-based Task Context Conditioning. Our image-based implementation is similar to Fang et al. [60], which utilizes image segmentation masks for task conditioning. In our setting, a manipulation task involves a robot skill (e.g. move, pick, knock) and a set of objects of interest. For any given task, we localize each object of interest with a 10x10 colored square mask in an overlay image, where the color determines the semantic skill pertaining to that object. Each square mask is

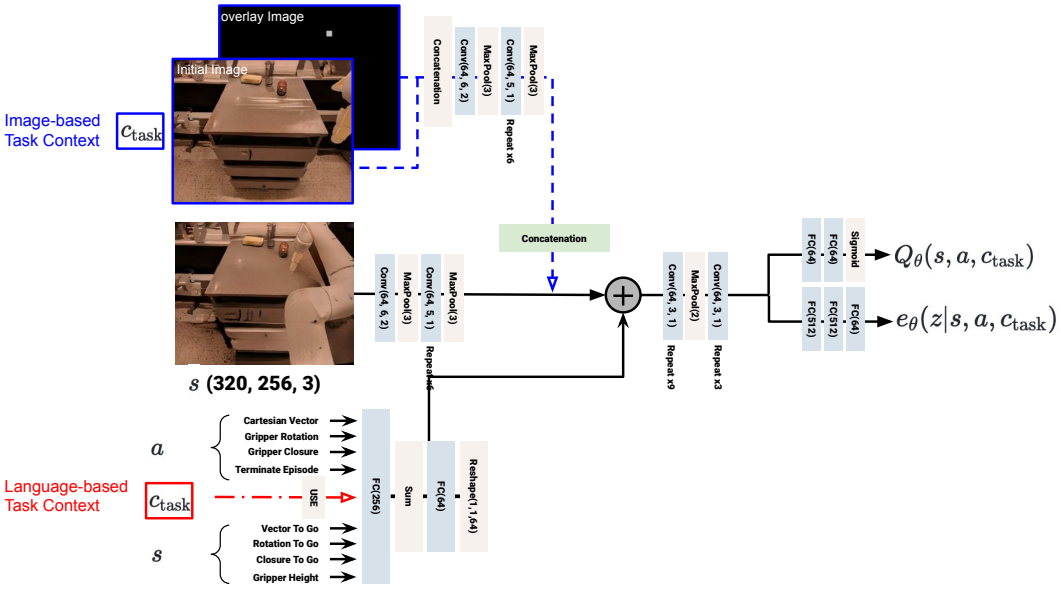


Figure 6: Network Architecture. Task context are either image-based (blue path) or language-based (red path). The convolution parameters shown in this diagram are (channels, kernel size, stride size).

centered at the the object bounding box center and agnostic of the object size. For example, the blue square in the first frame of Figure 2(f) indicates that the task is "picking up the can on the right", and the green square in the first frame of Figure 2(g) represents a task "knock down the plastic bottle in the middle". Note that the RGB color is for better presentation, and the actual implementation represents tasks in grayscale.

The reason that we use this type of task context mask instead of pixel-accurate segmentation masks [60] is as follows. While perfect pixel-accurate masks can be easily provided in simulation, when deployed in the real world (Section 5.2), pixel-accurate mask boundaries can be sensitive to many conditions including lighting, occlusions, the angle of view, and other perturbations. In practice, we find that simple 10x10 square masks tend to be more robust to these types of noise.

Notably, we only produce the task context masks one time per episode at the first frame. The same overlay image and initial frame are used as the task context throughout the entire episode. They are used only to represent a task, and not for enhancing perception during planning. This also avoids the need to run the VILD model [59] for object detection in real-time when deployed on the real robot.

Language-based Context Conditioning. Language Contexts are computed by using a pretrained and frozen Universal Sentence Encoder (USE) [61] to embed natural language task instructions, where each task corresponds to exactly one natural language instruction.

B.3 Architecture

Figure 6 shows the detailed network architecture. The convolutional and MLP blocks are similar to the network architecture used in [12]. Specifically, the first convolutional block before merging with the action or task context contains six convolutional layers and two pooling layers. The second convolutional block after merging contains nine convolutional layers. The Q-value MLP block contains two dense layers. Each convolutional and hidden dense layer is followed by a batch norm layer [62] and a ReLU activation layer.

An image-based task context consists of two images: one is the initial RGB image and the other is the corresponding grayscale overlay image. We concatenate these two images along the channel dimension and apply a convolutional block that has the same architecture as the first convolutional block that processes the current observation image. We concatenate the image-based task context embedding with the current observation image embedding along the channel dimension. Following

this step, the representation vector of action and proprioceptive state is merged with visual features by broadcasted element-wise addition.

In the alternative setting where we use the language-based task context instead of image-based task context, we use the natural language Universal Sentence Encoder (USE) [61] embedding of the current task instruction. This USE embedding is fed into an MLP with two fully connected layers, each with a batch norm layer. Finally, the processed embedding is fused with the action and proprioceptive state and eventually merged with visual features.

B.4 Comparing PI-QT-Opt and PI-SAC

We chose QT-Opt as the underlying control algorithm in this work. Compared to SAC [56] that PI-SAC [2] used, the main advantage of QT-Opt is that the action selection is pure sampling-based (CEM) [55], and thus does not require a gradient-learned actor as in SAC. This makes it possible to have complex and even dynamic action space and bounds without worrying about how to back-propagate gradients.

This makes adding safety constraints simple. Every time when we sample an action, we can clip the action according to the action bound, which can change based on the safety constraints at each specific robot state. For a gradient-based actor, such clipping zero-outs gradients, making optimization challenging. We did try training SAC and it did not learn with safety-constraint action clipping. How to make a gradient-based actor work in our setting is still an open-ended research question, for which we do not have a good answer at the moment.

On the other hand, there are certainly room for improvements of the underlying control algorithm in aspects that we do not attempt to address explicitly in this work, such as exploration.

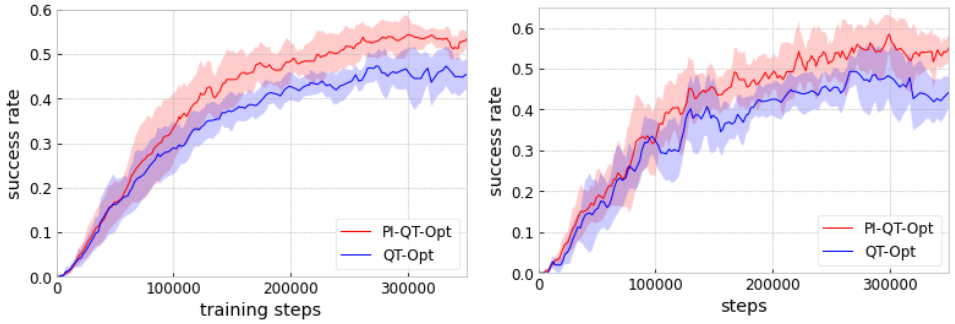
There are a few other architecture differences between PI-QT-Opt and PI-SAC. Because PI-SAC was evaluated on DM-Control [10], it follows the standard approach to stack 3 frames and, in order to avoid overlapping between the past and the future frames. The backward encoder does not share its convolutional backbone with the target Q-network. In PI-QT-Opt, because we only consider 1 past frame and 1 future frame, we are able to simplify the design, making backward encoder share the convolutional backbone with the target Q-network. In PI-SAC, the forward and backward encoder distributions are parameterized as Gaussian distributions, whereas, in PI-QT-Opt, the forward and backward encoder distributions are parameterized as von Mises-Fisher distributions.

B.5 Concurrent Control and Blocking Control

As described in Section 4.1, we utilize blocking control in the simulation-only environments (Instance Grasping and 6-Object Manipulation), where the policy waits until the previous action completes before observing the next environment state and planning the next action. Motivated by faster and more reactive robot motions for real world evaluations, we utilize concurrent control [14] in all SayCan experiments, which means that the current action is computed while the previous action is still executing. In particular, the SayCan results reported in Figure 3, Figure 4, Table 1, and Table 2 all utilize concurrent control. Intuitively, concurrent control is more challenging than blocking control, since predicting actions to execute while the robot is moving involves implicit planning compared to blocking control, where actions are guaranteed to execute in the exact same state as the observation. In Figure 7, we present PI-QT-Opt and QT-Opt results on a blocking version of the SayCan Move task set as an example. Compared with Figure 3(a) and Figure 4(a), we can observe that models learn faster and achieve better final performance when utilizing blocking control. The results also show that PI-QT-Opt outperforms QT-Opt by a similar amount on both the blocking and continuous control versions of the tasks (compare Figure 3(a) and Figure 4(a) to Figure 7(a) and (b)).

B.6 Model Training Details

We base our implementation on the distributed asynchronous QT-Opt system introduced in Kalashnikov et al. [12]. Our system uses the TF-Agents RL library [63]. For each experiment, we use 3000 data collection workers to interact with the simulation environment, 3000 "Bellman updater" jobs (Section 4.3 of [12]), a distributed replay buffer spread over 20 workers (Section F.5 of [12]), and 16 TPUv2 for model learning. We learn with stochastic gradient descent (SGD) with momentum. The



(a) On tasks that are used for agent training.

(b) On unseen novel tasks.

Figure 7: Performance on SayCan Move tasks (Blocking Control)

Table 2: Evaluations of a single model that solves SayCan 297 Tasks on the real robot.

Task	PI-QT-Opt Success Rate	QT-Opt Success Rate	Relative Change
SayCan Move	**25.0** $\pm$ 3.2%	17.4 $\pm$ 4.3%	+44.0%
SayCan Pick	**33.4** $\pm$ 12.5%	22.2 $\pm$ 9.6%	+50.1%
SayCan Knock	**52.3** $\pm$ 10.7%	39.2 $\pm$ 4.1%	+33.2%

learning rate is 9.56×10^{-3} and the momentum weight is 0.984. The model training time depends on the learning task set. For example, a SayCan 297-task model takes five to seven days to learn, while an instance grasping model takes 20 hours.

B.7 Model Evaluation Protocol Details

In simulation, for every episode, we sample a task, which involves a skill and objects of interest. The objects of interest and additional randomly sampled distractor objects are then randomly placed in the scene. If the randomly generated scene is already in a successful state, we regenerate it. The robot is always initialized with the same pose of the arm but the base position is randomly sampled within a small rectangular area in front of the counter (for SayCan tasks), the waste station (for Instance Grasping tasks), or the table (for 6-Object Manipulation tasks).

For controlled and fair evaluations in the real-world tasks, we use up to three variations of each task that are manually reset as precisely as possible for each model we evaluate (three models each for QT-Opt and PI-QT-Opt). Thus, each model gets a single attempt at each task variation, but multiple attempts at each task, and each model sees the same set of task variations. The only variation between models in evaluation is randomization of the robot's base position at the start of each episode.

C SayCan Objects

Figure 8 shows examples of the objects used in the SayCan tasks.

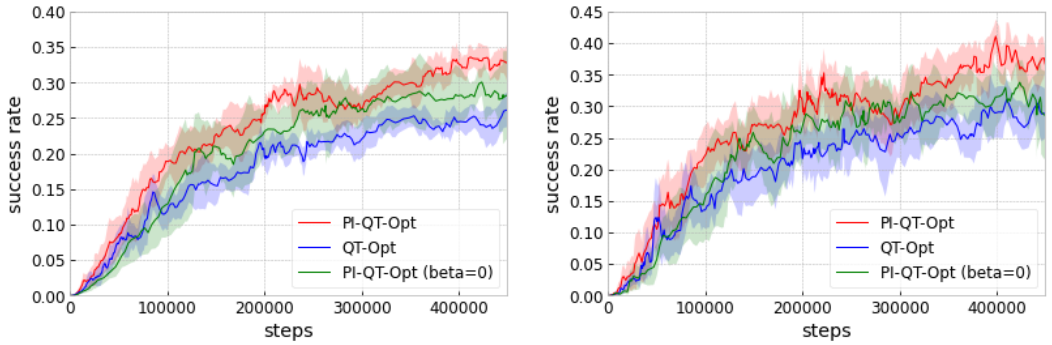
D Evaluating the SayCan-297-Task Model in the Real World

In Section 5.2, we evaluated per-skill SayCan models in the real world. Here, we additionally report real-world evaluation of the SayCan-297-Task model, which is trained on all 297 SayCan tasks. The evaluation results on each skill category are summarized in Table 2. In this set of experiments, PI-QT-Opt continues to outperform QT-Opt by a large margin.

Figure 8: Example objects used in the SayCan tasks. Objects are placed on a kitchen counter, where a robot must perform different manipulation skills such as picking objects up, moving objects, and knocking objects over. We train on 17 objects in simulation and evaluate on these 15 objects in the real world. Not shown here are a blue energy bar and an orange.

E Ablation of Predictive Information Compression

To understand the importance of compression to PI-QT-Opt, rather than just predicting the future, we compare QT-Opt, PI-QT-Opt at $\beta = 0$, and PI-QT-Opt at $\beta = 0.01$, the value used in all other experiments. When $\beta = 0$, the model still learns to predict the future (Y) from the past (X), but it no longer makes any explicit attempt to compress irrelevant information in X. In Figure 9 we compare these three models on the SayCan 297 Tasks, both for training tasks (a), and unseen evaluation tasks (b). In both cases, the compressed version of PI-QT-Opt slightly outperforms the uncompressed version, which still slightly outperforms QT-Opt without the predictive information auxiliary loss. We note, however, that compressed PI-QT-Opt often overlaps in performance with uncompressed PI-QT-Opt, so clearly the advantage of PI-QT-Opt over QT-Opt is only partially due to compression.



(a) SayCan 297 Tasks (evaluated on training tasks). (b) SayCan 297 tasks (evaluated on unseen tasks).

Figure 9: Comparison between PI-QT-Opt ($\beta = 0.01$ by default), PI-QT-Opt ($\beta = 0.0$, no explicit compression), and QT-Opt.

F Extending the Image-Based Task Context to Free-form Commands

The tasks we selected are in a structured language form of skill and target object sets. To support more complicated, natural language commands, we could encode the task description with a lan-

guage model. However, for the image-based task context described in Section 3.2 and Appendix B.2, we would also need a mechanism to visually represent the command's target(s) in the scene. We speculate that providing object detection network activations, rather than the simpler conditioning described in Section 3.2, may be sufficient to capture the relevant information for interpreting the command. We leave this generalization of our method for future work.

G Lists of 297 SayCan Tasks

In the following table we list all 297 SayCan tasks that are used for training or held-out for evaluation. Note that the knock skill target objects only include cans and bottles since other objects cannot be "knocked down" from an upright pose (See Appendix C for the object list).

Pick skill	
Training tasks	
pick 7up can	pick apple
pick blue chip bag	pick brown chip bag
pick coke can	pick green can
pick green jalapeno chip bag	pick orange can
pick pepsi can	pick redbull can
pick rxbar blueberry	pick water bottle
Held-out evaluation tasks	
pick blue plastic bottle	pick green rice chip bag
pick orange	pick rxbar chocolate
pick sponge	
Knock skill	
Training tasks	
knock 7up can over	knock blue plastic bottle over
knock coke can over	knock green can over
knock pepsi can over	knock redbull can over
knock water bottle over	
Held-out evaluation tasks	
knock orange can over	
Move skill	
Training tasks	
move 7up can near apple	move 7up can near blue chip bag
move 7up can near blue plastic bottle	move 7up can near brown chip bag
move 7up can near coke can	move 7up can near green can
move 7up can near green jalapeno chip bag	move 7up can near green rice chip bag
move 7up can near orange	move 7up can near orange can
move 7up can near pepsi can	move 7up can near rxbar blueberry
move 7up can near rxbar chocolate	move 7up can near sponge
move 7up can near water bottle	move apple near 7up can
move apple near blue chip bag	move apple near blue plastic bottle
move apple near brown chip bag	move apple near coke can
move apple near green can	move apple near green jalapeno chip bag
move apple near orange	move apple near orange can
move apple near pepsi can	move apple near rxbar blueberry
move apple near rxbar chocolate	move apple near sponge
move apple near water bottle	move blue chip bag near 7up can
move blue chip bag near apple	move blue chip bag near brown chip bag
move blue chip bag near coke can	move blue chip bag near green can
move blue chip bag near green jalapeno chip bag	move blue chip bag near green rice chip bag
move blue chip bag near orange	move blue chip bag near orange can
move blue chip bag near redbull can	move blue chip bag near rxbar blueberry
move blue chip bag near rxbar chocolate	move blue chip bag near water bottle
move blue plastic bottle near 7up can	move blue plastic bottle near apple
move blue plastic bottle near blue chip bag	move blue plastic bottle near brown chip bag
move blue plastic bottle near coke can	move blue plastic bottle near green can
move blue plastic bottle near green jalapeno chip bag	move blue plastic bottle near green rice chip bag
move blue plastic bottle near orange	move blue plastic bottle near orange can
move blue plastic bottle near pepsi can	move blue plastic bottle near redbull can
move blue plastic bottle near rxbar blueberry	move blue plastic bottle near rxbar chocolate
move blue plastic bottle near sponge	move blue plastic bottle near water bottle
move brown chip bag near 7up can	move brown chip bag near blue chip bag
move brown chip bag near blue plastic bottle	move brown chip bag near coke can
move brown chip bag near green can	move brown chip bag near green jalapeno chip bag
move brown chip bag near orange	move brown chip bag near orange can
move brown chip bag near pepsi can	move brown chip bag near rxbar blueberry
move brown chip bag near rxbar chocolate	move brown chip bag near sponge
move brown chip bag near water bottle	move coke can near 7up can
move coke can near apple	move coke can near blue chip bag
move coke can near blue plastic bottle	move coke can near brown chip bag
move coke can near green can	move coke can near green rice chip bag
move coke can near orange	move coke can near orange can
move coke can near pepsi can	move coke can near redbull can
move coke can near rxbar blueberry	move coke can near rxbar chocolate
move coke can near sponge	move green can near blue chip bag
move green can near blue plastic bottle	move green can near brown chip bag
move green can near green jalapeno chip bag	move green can near green rice chip bag
move green can near orange	move green can near orange can

move green can near pepsi can
move green can near rxbar blueberry
move green can near sponge
move green jalapeno chip bag near 7up can
move green jalapeno chip bag near blue plastic bottle
move green jalapeno chip bag near coke can
move green jalapeno chip bag near green rice chip bag
move green jalapeno chip bag near orange can
move green jalapeno chip bag near redbull can
move green jalapeno chip bag near rxbar chocolate
move green jalapeno chip bag near water bottle
move green rice chip bag near apple
move green rice chip bag near blue plastic bottle
move green rice chip bag near coke can
move green rice chip bag near green jalapeno chip bag
move green rice chip bag near redbull can
move green rice chip bag near rxbar chocolate
move green rice chip bag near water bottle
move orange can near apple
move orange can near blue plastic bottle
move orange can near green can
move orange can near green rice chip bag
move orange can near pepsi can
move orange can near rxbar blueberry
move orange can near sponge
move orange near 7up can
move orange near blue chip bag
move orange near brown chip bag
move orange near green can
move orange near green rice chip bag
move orange near pepsi can
move orange near rxbar blueberry
move orange near sponge
move pepsi can near 7up can
move pepsi can near blue chip bag
move pepsi can near brown chip bag
move pepsi can near green can
move pepsi can near green rice chip bag
move pepsi can near redbull can
move pepsi can near rxbar chocolate
move pepsi can near water bottle
move redbull can near apple
move redbull can near blue plastic bottle
move redbull can near green can
move redbull can near green rice chip bag
move redbull can near orange can
move redbull can near rxbar blueberry
move redbull can near sponge
move rxbar blueberry near 7up can
move rxbar blueberry near blue chip bag
move rxbar blueberry near coke can
move rxbar blueberry near green jalapeno chip bag
move rxbar blueberry near orange
move rxbar blueberry near redbull can
move rxbar blueberry near sponge
move rxbar chocolate near 7up can
move rxbar chocolate near blue chip bag
move rxbar chocolate near brown chip bag
move rxbar chocolate near green can
move rxbar chocolate near green rice chip bag
move rxbar chocolate near orange can
move rxbar chocolate near redbull can
move rxbar chocolate near water bottle
move sponge near blue chip bag
move sponge near brown chip bag
move sponge near green can
move sponge near green rice chip bag
move sponge near orange can
move sponge near redbull can
move sponge near rxbar chocolate
move water bottle near apple
move water bottle near blue plastic bottle
move water bottle near coke can
move water bottle near green jalapeno chip bag
move water bottle near orange
move water bottle near pepsi can
move water bottle near rxbar blueberry
move green can near redbull can
move green can near rxbar chocolate
move green can near water bottle
move green jalapeno chip bag near apple
move green jalapeno chip bag near brown chip bag
move green jalapeno chip bag near green can
move green jalapeno chip bag near orange
move green jalapeno chip bag near pepsi can
move green jalapeno chip bag near rxbar blueberry
move green jalapeno chip bag near sponge
move green rice chip bag near 7up can
move green rice chip bag near blue chip bag
move green rice chip bag near brown chip bag
move green rice chip bag near green can
move green rice chip bag near pepsi can
move green rice chip bag near rxbar blueberry
move green rice chip bag near sponge
move orange can near 7up can
move orange can near blue chip bag
move orange can near coke can
move orange can near green jalapeno chip bag
move orange can near orange
move orange can near redbull can
move orange can near rxbar chocolate
move orange can near water bottle
move orange near apple
move orange near blue plastic bottle
move orange near coke can
move orange near green jalapeno chip bag
move orange near orange can
move orange near redbull can
move orange near rxbar chocolate
move orange near water bottle
move pepsi can near apple
move pepsi can near blue plastic bottle
move pepsi can near coke can
move pepsi can near green jalapeno chip bag
move pepsi can near orange
move pepsi can near rxbar blueberry
move pepsi can near sponge
move redbull can near 7up can
move redbull can near blue chip bag
move redbull can near brown chip bag
move redbull can near green jalapeno chip bag
move redbull can near orange
move redbull can near pepsi can
move redbull can near rxbar chocolate
move redbull can near water bottle
move rxbar blueberry near apple
move rxbar blueberry near brown chip bag
move rxbar blueberry near green can
move rxbar blueberry near green rice chip bag
move rxbar blueberry near pepsi can
move rxbar blueberry near rxbar chocolate
move rxbar blueberry near water bottle
move rxbar chocolate near apple
move rxbar chocolate near blue plastic bottle
move rxbar chocolate near coke can
move rxbar chocolate near green jalapeno chip bag
move rxbar chocolate near orange
move rxbar chocolate near pepsi can
move rxbar chocolate near sponge
move sponge near 7up can
move sponge near blue plastic bottle
move sponge near coke can
move sponge near green jalapeno chip bag
move sponge near orange
move sponge near pepsi can
move sponge near rxbar blueberry
move sponge near water bottle
move water bottle near blue chip bag
move water bottle near brown chip bag
move water bottle near green can
move water bottle near green rice chip bag
move water bottle near orange can
move water bottle near redbull can
move water bottle near rxbar chocolate

Held-out evaluation tasks

move 7up can near redbull can	move apple near green rice chip bag
move apple near redbull can	move blue chip bag near blue plastic bottle
move blue chip bag near pepsi can	move blue chip bag near sponge
move brown chip bag near apple	move brown chip bag near green rice chip bag
move brown chip bag near redbull can	move coke can near green jalapeno chip bag
move coke can near water bottle	move green can near 7up can
move green can near apple	move green can near coke can
move green jalapeno chip bag near blue chip bag	move green rice chip bag near orange
move green rice chip bag near orange can	move orange can near brown chip bag
move pepsi can near orange can	move redbull can near coke can
move rxbar blueberry near blue plastic bottle	move rxbar blueberry near orange can
move rxbar chocolate near rxbar blueberry	move sponge near apple
move water bottle near 7up can	move water bottle near sponge