
Same Object, Different Grasps: Data and Semantic Knowledge for Task-Oriented Grasping

Adithyavairavan Murali¹, Weiyu Liu², Kenneth Marino¹, Sonia Chernova^{2,3}, and Abhinav Gupta^{1,3}

¹The Robotics Institute, Carnegie Mellon University

²Institute for Robotics and Intelligent Machines, Georgia Institute of Technology

³Facebook AI Research

Abstract

Despite the enormous progress and generalization in robotic grasping in recent years, existing methods have yet to scale and generalize task-oriented grasping to the same extent. This is largely due to the scale of the datasets both in terms of the number of objects and tasks studied. We address these concerns with the TaskGrasp dataset which is more diverse both in terms of objects and tasks, and an order of magnitude larger than previous datasets. The dataset contains 250K task-oriented grasps for 56 tasks and 191 objects along with their RGB-D information. We take advantage of this new breadth and diversity in the data and present the GCNGrasp framework which uses the semantic knowledge of objects and tasks encoded in a knowledge graph to generalize to new object instances, classes and even new tasks. Our framework shows a significant improvement of around 12% on held-out settings compared to baseline methods which do not use semantics. We demonstrate that our dataset and model are applicable for the real world by executing task-oriented grasps on a real robot on unknown objects. Code, data and supplementary video could be found <https://sites.google.com/view/taskgrasp>.

1 Introduction and Related Work

We have seen tremendous progress in the fundamental task of robotic grasping in recent years. State-of-the-art grasping algorithms have shown generalization to object instances [1, 2, 3, 4], viewpoints [5], DOF constraints [6, 7, 8], unknown environments [9] and even adversarial objects [10]. The key reason for the success of these approaches is large-scale learning. Typically data is sampled from analytical approaches in simulation [1, 7] or using a self-supervised framework [4, 5]. Despite these recent successes, there is still a significant gap between how humans grasp objects and how robots perform picking. Most techniques plan for stable grasps assuming grasping to be the end goal. However, when humans grasp an object, we do so with a particular purpose in mind and grasping is just the first step as a means to that end. For example, when humans grasp a cup, we use the handle to drink from it though several other stable grasps exist. Humans also use objects creatively, such as scooping with a bowl or hammering with a heavy mug. Different tasks may require completely different grasps for the same object. To effectively operate in human homes and complete multiple tasks, a personal robot would have to learn from humans to generalize grasping to several tasks. Towards this goal, we study not just stable grasping or grasping for an object’s primary use-case but rather how to grasp depending on both the task and the object.

The biggest bottleneck in task-oriented robotic grasping is the need for human-labeled data. Unlike self-supervised or analytical approaches for which force sensing or contact models can provide labels for stable grasps, here we need humans to identify how an object can be grasped for multiple tasks. There has been a lot of recent work in this area, including [11, 12, 13]. Brahmabhatt et al. [11] used thermal imaging in a curated setup to study human grasping contacts on 50 3D printed objects for two tasks. Fang et al. [13] proposed to jointly learn a task-oriented grasping network and



Figure 1: Example point clouds and grasps from our TaskGrasp dataset. Column 7-9 shows how grasps vary with tasks for a salad tongs (with higher diversity) and a rolling pin (with lower diversity). Green and Red means successful and incorrect task-oriented grasps respectively.

manipulation policy in simulation with reinforcement learning and demonstrated the framework on two goal tasks with two object categories. Liu et al. [12] proposed a data-driven approach to learning the complex relationships between grasps, objects, tasks, and broadened semantic contexts. However, their approach required pixel-wise affordance segmentation [14] for a small set of known object categories. Despite this progress in learning from human grasping, existing datasets are limited in the number of tasks and object classes collected.

As such, the first key contribution of this work is the collection of a large-scale dataset which we call TaskGrasp. We increase the number of real objects from the current best of 50 in prior works [11] to 191, and collect RGB-D point cloud observations and object-centric 6-DOF grasps for the task-oriented grasping problem. We also scale the number of object classes from 40 [11] to 75 and resolve each of these to the standard WordNet ontology [15]. Most importantly, we scale the number of tasks from 2 – 7 in prior works [12, 11, 13] to 56 everyday tasks that impose different semantic constraints on grasping. We use crowdsourcing to annotate 250K stable grasps on whether a grasp is appropriate for each particular task. This expanded dataset both gives a better benchmark for task-oriented grasping and allows us to study generalization by expanding the number of object categories and tasks. TaskGrasp will be publicly released upon publication and more details of the dataset are provided in the supplementary material.

In order to generalize to a new object or task, we need to have some prior semantics about it. Our second contribution is the GCNGrasp framework which incorporates semantic knowledge into the end-to-end learning of task-oriented grasping from object point clouds. In particular, we use a Graph Convolutional Network (GCN) [16] to reason about a knowledge graph that encodes relations between objects and tasks, and further leverage word embeddings trained on large-scale language tasks to provide additional prior information. Our GCNGrasp model shows a significant improvement of 12% and 3.5% on held-out tasks and object categories, respectively, compared to baselines which do not incorporate semantics. We also show that our method and dataset are applicable for actual robots by executing task-oriented stable grasps on a 7-DOF Sawyer Robot on unknown objects.

2 Task-Oriented Grasping with Semantic Knowledge

We consider the problem of generating grasps for task-oriented grasping given the object point cloud and task constraints. Specifically, we want to estimate the grasp distribution $P(G^*|X, \mathcal{T})$, where X is the point cloud input, \mathcal{T} are the constraints imposed by goal tasks, and G^* is the space of successful grasps. Following convention in related work [8, 7], we represent grasps $g \in G^*$ as the grasp pose $(R, T) \in SE(3)$ of a parallel-jaw gripper with its fingers open which when closing will lead to a stable grasp. We further factorize the estimation of $P(G^*|X, \mathcal{T})$ into 1) task-agnostic grasp sampling $P(G^*|X)$ and 2) task-oriented grasp evaluation $P(S|X, \mathcal{T}, g)$. The primary benefit of this factorization is that it allows us to leverage prior work in stable grasp generation.

In this section, we describe our GCNGrasp method as shown in Fig 2. It is comprised of: (1) a Grasp and Object Encoder built on a PointNet++ architecture [17] to encode the object point cloud and grasp (2) a Graph Convolutional Network [16] which takes the encoded object and grasp as input as

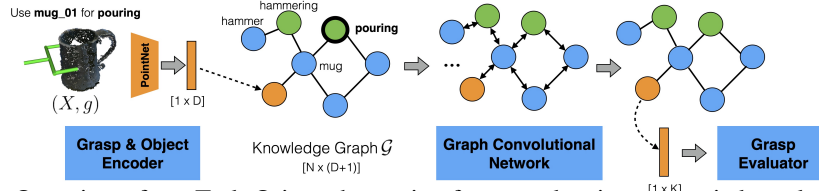


Figure 2: Overview of our Task-Oriented grasping framework using semantic knowledge graphs. well as a knowledge graph \mathcal{G} encoding the semantic relationships between object categories, tasks and hierarchies and (3) a Grasp Evaluator which outputs the final grasp prediction.

Grasp and Object Encoder: Our input observations are object point clouds and we want to reason about $SE(3)$ grasps. Qi et al. [17] proposed the PointNet++ architecture to efficiently represent 3D data which we use to learn a representation for the object point cloud and 6-DOF grasp poses. The grasp g is defined in the object frame and six control points are selected on the gripper to form a gripper point cloud X_g . Similar to Mousavian et al. [7], X_g is concatenated with the object point cloud X with an extra latent indicator vector to represent whether a point is part of the gripper or the object. The PointNet layer reasons about the relative spatial information between the grasp and the object. It outputs an embedding which is used to initialize the grasp node (orange in Fig 2) in the graph.

Graph Convolutional Network: We use the standard Graph Convolutional Network (GCN) model from Kipf and Welling [16], which is a neural network structured on the shape of the input graph. By structuring a neural network to pass information between adjacent nodes, we use the input graph to correctly reason about the relationship between the object classes and the target task. The first input of a GCN is the graph itself $\mathcal{G} = (V, E)$. In our application, we use a knowledge graph constructed from two sources: the task-object class relationships in our dataset and the object hierarchy from WordNet [15]. The grasp nodes (orange in Fig 2) are added online to the existing knowledge graph \mathcal{G} by connecting edges to the corresponding object class nodes. The graph is represented as a binary adjacency matrix A , which we normalize to obtain \hat{A} following [16]. The next input to each node of the GCN is a $D + 1$ -dimensional embedding vector, where the extra dimension is used to indicate the target task. Except for the grasp node, all other nodes are initialized with word embeddings [18] corresponding to each concept in the knowledge graph (e.g. “mug”). The embedding vectors are stacked across nodes to get the input matrix $\mathcal{X} \in \mathbb{R}^{|V| \times (D+1)}$. The output of the GCN are K -dimensional (with $K=128$) embeddings for each node $\mathcal{Z} \in \mathbb{R}^{|V| \times K}$. The node embeddings are propagated to their neighbours using message passing in each convolutional layer $H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)})$, where σ is the ReLU activation function, $H^{(0)} = \mathcal{X}$ and $H^{(L)} = \mathcal{Z}$ where $L=6$ is the number of layers. Implementation details are provided in the supplementary.

Grasp Evaluator: After the GCN, we are left with a node-level embedding \mathcal{Z} . We use the embedding corresponding to the grasp node z_g to train the final grasp evaluator $P(S|z_g)$, where S is the grasp score. This module has three fully connected layers with $K=126$ units and a final sigmoid layer. The entire model is optimized with ADAM using a binary cross entropy loss.

3 Experimental Evaluation and Discussion

A central goal of our dataset and method is to show that we can learn task-oriented grasping models which generalize to novel objects, classes and tasks. To test this, we focus on three different held-out test settings of increasing difficulty: held-out object instances, classes and tasks. For each setting, we perform k -fold cross validation ($k=4$), such that each category (a task, object class, or object instance based on the setting) will be held out exactly once. We report mean Average Precision (mAP) averaged over all object instances, mAP over classes, and mAP over tasks as shown in Table 1.

Baselines: We compare our approach to the following models: (1) Random, which represents grasping strategies that focus on grasp stability and ignore task constraints. Results are averaged over five random seeds. (2) Semantic Grasp Network (SGN), which learns to reason about context of grasps (e.g., constraints imposed by objects and tasks) from data. This model is adapted from [12], with the difference that the input to the model is replaced with geometric embedding from our shape encoder and word embeddings of the task and the object class. Note that embeddings of tasks and object classes are both learned from training data. (3) SGN + *word embedding*, which uses ConceptNet [18] numberbatch as pretrained word embeddings for object classes and tasks.

Analysis: First, we see that random grasp prediction achieves approximately 50-60% accuracy, establishing a floor for the other methods. Because the number of positive and negative grasps in

Table 1: Results on TaskGrasp

Model	Test Performance (mAP)		
	Instances	Classes	Tasks
Random	59.75	58.73	52.37
SGN [12]	78.51	72.95	63.35
SGN + word embedding	79.74	75.51	70.55
GCNGrasp (ours)	80.25	76.57	76.01

Table 2: Ablation on Semantic Knowledge

Model	Held-out Setting		
	Task	Class	Instance
GCN + tasks + WordNet	76.01	76.57	80.25
GCN + tasks	77.54	75.86	81.46
GCN + WordNet	71.77	70	78.66

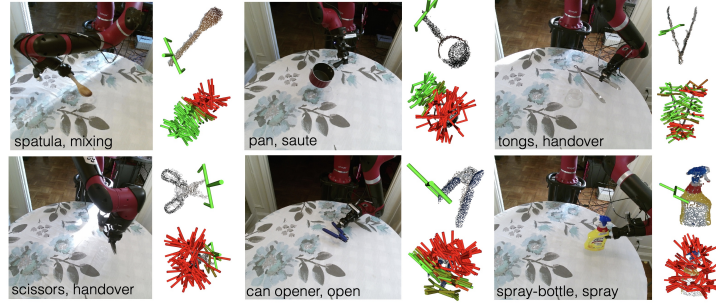


Figure 3: For each execution, the top visualization shows the grasp that was executed (which had the best score) and the bottom shows the stable grasp candidates colored by their scores (green is higher).

the dataset is about even, random guessing is able to achieve a seemingly high mAP. In a dataset with more negatives we would expect this number to be much lower. Our method outperforms baselines in all three settings. This confirms that our method can effectively leverage the knowledge graph to generalize to novel object instances, classes, and tasks. SGN + *word embedding* also outperforms SGN, suggesting that implicit distributional knowledge provides a prior that is useful for generalization. Despite the benefit of distributional knowledge, it still only represents semantic similarities between concepts. In contrast, the knowledge graph directly stores relations between the relevant objects and tasks, and exploiting this additional knowledge allows our model to achieve better zero-shot generalization than SGN + *word embedding*. When comparing our method with SGN and SGN + *word embedding*, we observe increasingly larger margins in performance from the held-out instance to the held-out class setting. As objects from different classes have more variance in terms of geometric features than objects from the same class, semantic knowledge becomes more important in unifying these objects. The difference in performance between our method and these two baselines on the held-out task setting reached 12.6% and 5.46% respectively, affirming that semantic knowledge is especially crucial for generalizing disparate constraints from different tasks.

Ablations on Knowledge Graph: We compared the default knowledge graph with a graph containing only the semantic hierarchy of objects and one containing only the relations between object classes and tasks. The results are summarized in Table 2. We observe that edges between object classes and tasks were the most important knowledge for generalizing to novel tasks and instances, though every task we tested was valid for the target object class. This suggests that knowledge about which objects could generally be used for which tasks provide important information for discovering similarities between tasks. In the held-out class setting, additional knowledge from the object hierarchy helped to generalize to novel classes by associating known and novel classes through the WordNet hierarchy.

Real Robot Evaluation: We run experiments to show that our approach and dataset transfer to a real robot and novel objects in unknown poses. We evaluate on our best performing GCNGrasp model from the held-out task ablations (Table 1). Fig 3 shows the executed task-oriented grasps on the Sawyer robot with a parallel-jaw gripper. Based on the grasp evaluator scores from Fig 3, our model is able to interpolate between modes in the continuous $SE(3)$ space to reason about task-oriented grasping. One failure mode of our work is that it does not generalize to categories (like the spray bottle in Fig 3 in the bottom right) with limited training data.

4 Conclusion

We present the TaskGrasp dataset to study generalization in Task-Oriented grasping. We also present a framework for jointly learning from geometric and semantic knowledge to generalize to new object instances, classes and even tasks. While we collected real point cloud data of objects, we could convert the point clouds to meshes or acquire shape models from large online repositories to use in physics simulators. This could expand the scope of the dataset for sim2real transfer and to even learn task policies in simulation conditioned on the task-oriented grasps like in prior work [13].

References

- [1] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *RSS*, 2017.
- [2] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *Conference on Robot Learning*, 2018.
- [3] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.
- [4] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [5] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *International Symposium on Experimental Robotics (ISER)*, 2016.
- [6] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-dof grasping for target-driven object manipulation in clutter. *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [7] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-DOF GraspNet: Variational grasp generation for object manipulation. *International Conference on Computer Vision*, 2019.
- [8] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017.
- [9] Abhinav Gupta, Adithyavairavan Murali, Dhiraj Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *Neural Information Processing Systems (NeurIPS)*, 2018.
- [10] David Wang, David Tseng, Pusong Li, Yiding Jiang, Menglong Guo, Michael Danielczuk, Jeffrey Mahler, Jeffrey Ichnowski, and Ken Goldberg. Adversarial grasp objects. In *Conference on Automation Science and Engineering*. IEEE, 2019.
- [11] Samarth Brahmabhatt, Cusuh Ham, Charlie Kemp, and James Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [12] Weiyu Liu, Angel Daruna, and Sonia Chernova. Cage: Context-aware grasping engine. *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [13] Kuan Fang, Yuke Zhu, Animesh Garg, Andrey Kurenkov, Viraj Mehta, Li Fei-Fei, and Silvio Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision. *Robotics Science and Systems*, 2018.
- [14] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 ICRA*, pages 1–5. IEEE, 2018.
- [15] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.
- [16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017.
- [17] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

- [18] Robyn Speer, Joshua Chin, and Catherine Havasi. ConceptNet 5.5: An open multilingual graph of general knowledge. pages 4444–4451, 2017. URL <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972>.
- [19] Adithyavairavan Murali, Tao Chen, Kalyan Vasudev Alwala, Dhiraj Gandhi, Lerrel Pinto, Saurabh Gupta, and Abhinav Gupta. Pyrobot: An open-source robotics framework for research and benchmarking. 2019. URL <https://arxiv.org/abs/1906.08236>.
- [20] Andreas ten Pas and Robert Platt. Using geometry to detect grasp poses in 3d point clouds. In *Robotics Research*, pages 307–324. Springer, 2018.
- [21] Justus J Randolph. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss’ fixed-marginal multirater kappa. *Online submission*, 2005.

A Appendix

This document is the supplementary materials for the paper, along with the video. We first provide more details regarding the TaskGrasp dataset, implementation details for the framework and metrics for experimental evaluations. Second, we demonstrate that models trained on our TaskGrasp dataset transfer to other task-oriented grasping datasets, namely SG14000 proposed by Liu et al. [12]. Third, we show more analysis on our dataset and model predictions on different tasks. Lastly, we describe our procedure and graphical interface for annotating task-oriented grasps on Amazon Mechanical Turk (AMT).

A.1 Dataset

In this section we describe our dataset: TaskGrasp, specifically its properties, collection and annotation methodology. As shown in Table 3, TaskGrasp is the largest and most diverse dataset for task-oriented grasping to date with respect to number of objects, categories and tasks.

TaskGrasp contains 191 individual household and kitchen objects comprising 75 distinct object categories and varying in size, geometry, material, and visual appearance. Figure 4 shows the class of each object and its proportion in the dataset. We collect RGB-D pointclouds for each object, and automatically annotate 250K stable grasps using crowdsourcing. We also curate a list of 56 everyday tasks that impose different semantic constraints on grasping and annotate for each grasp whether that grasp is appropriate for each particular task. The details of the data acquisition process and analysis are provided in the supplementary material.

A.1.1 Data Acquisition on a Robot

After selecting our 191 objects by browsing various homegoods stores, we scan the objects to acquire their point clouds. A Realsense D415 eye-in-hand camera mounted on a LoCoBot [19] is used for 3D scanning. The object is placed on a transparent mount in front of the robot, which is commanded to different poses along the object approach direction to capture point clouds from multiple viewpoints. This setup helps to capture more of the object geometry under self-occlusion, which in turn increases the coverage of grasp samples. The multi-view observations are registered using robot kinematics and further refined with the iterative closest point algorithm. After table plane segmentation, 600 object-centric stable grasps are then sampled [20] from the object point cloud. 25 grasps are selected with farthest point sampling (to maximize grasp coverage) for annotation. These grasps are chosen as a representative, albeit limited, grasp set for the object to trade off between dataset size and budget.

A.1.2 Data Annotation by Crowdsourcing

We use Amazon Mechanical Turk (AMT) to crowdsource labels for the 250K stable grasps. Instead of exhaustively labelling each task-object combination ($\sim 10K$), we reduce the annotation cost with a two-stage procedure. We use the insight that the pre-condition for a task-oriented grasp is that the object has to be capable of the task in the first place. First, we gather labels for whether a task is suitable for each object. Second, for this filtered subset of task-object combinations, we collect labels for the 25 task-oriented grasps per object. To ensure annotation quality, we assign each labeling task to three annotators and use gold standard questions (questions that we know the answers to) to filter

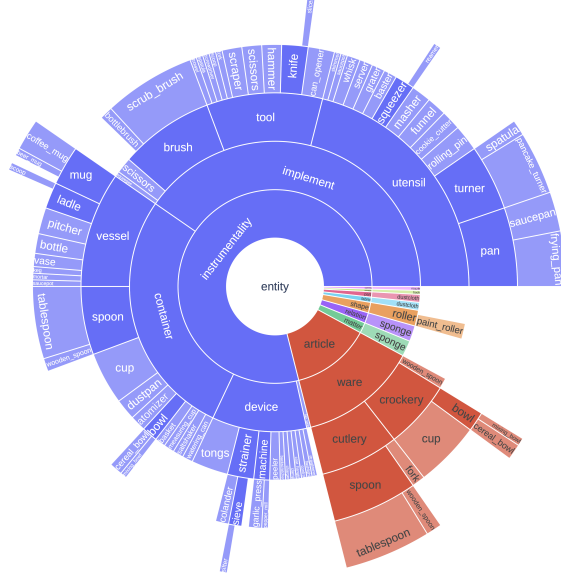


Figure 4: Semantic hierarchy of objects. Each level of the hierarchy is represented by one ring with the innermost circle as the root of the hierarchy. The angle of each segment is proportional to the number of objects.

annotators with low accuracy. For both stages, we take a majority vote between the annotators. We measure agreement with Randolph’s free-marginal multirater kappa [21]. Kappa values for the two stages are 0.65 and 0.62 respectively (0.0 meaning agreement equal to chance, and 1.0 indicating perfect agreement above chance), which suggests good agreement between annotators.

A.1.3 Analysis

In Figure 5 we show prototypical examples from TaskGrasp. Additional examples are given in the original paper.

Diversity of Grasps: As a result of the large number of objects and tasks, TaskGrasp contains a wide variety of task-oriented grasps. On average, each object is suitable for 7 tasks. As shown in Fig 5, these tasks involve both prototypical (a spatula for sauteing) and creative use of objects (mug for scooping), imposing drastically different semantic constraints on grasping. These examples also demonstrate the complex geometries presented in real world objects, which pose another challenge for generalization.

We also quantitatively measure grasp diversity by analyzing the effect of tasks on grasps. Since different tasks provide different labels for the same set of stable grasps on each object, we compute Randolph’s kappa [21] on these labels as a measure of agreement between tasks, i.e., how likely grasps for one task (e.g., stir) agree with grasps for another task (e.g., cut). Ranging from 0.19 to 0.93, kappa values of the objects suggest that the effect of tasks vary greatly for different objects. Column 7-9 in Figure 1 (in the main submission draft) shows how grasps vary with tasks for a salad tongs with a kappa value of 0.38 and a rolling pin with kappa value of 0.97. In TaskGrasp, 25% of the objects have kappa values lower than 0.5 and these objects require significantly different grasps for different tasks.

Semantic Knowledge of Objects and Tasks: We also provide semantic knowledge about objects and tasks in the dataset. Objects are manually mapped to WordNet synsets [15] which represent a semantic hierarchy, as shown in Figure 4. Each of the 75 leaf synsets in the hierarchy represents a distinct object class and is linked to 2.5 objects on average. Building on the hypernym paths from WordNet, the semantic hierarchy includes a rich set of object concepts interlinked by “Is-A” relations. This provides useful semantic knowledge for task-oriented grasping as objects in the same subtree of the hierarchy often share similar functionalities or geometric properties. For example, mug, ladle, and bottle are in the vessel subtree and can all be used to hold liquid. In addition, we connect a task to an object class through “Used-For” relations if any object in the class is considered suitable for the

Table 3: Comparing recent Task-Oriented Grasping Datasets

	ContactDB [11]	SG14000 [12]	TOG-Net [13]	TaskGrasp (Ours)
Semantic Knowledge	X	X	X	✓
Object Categories	40	5	2	75
Objects	50	44	18K (synthetic)	191
Tasks	2	7	2	56
Grasps	3750	14K	1.5M	250K
Grasp Type	Contact Map	$SE(3)$	Planar	$SE(3)$

task from the first stage of our crowdsourcing. We provide a thorough breakdown of object counts, class hierarchies and used-for relations in the supplementary materials.

A.2 Implementation Details for GCNGrasp

The point clouds were all downsampled to 4096 points during training. They were also mean centered and unit-scaled. The PointNet module consists of three set abstraction layers and the number of points sampled are 512, 128 and all points. The set abstraction layers are followed by three fully connected layers with sizes $[1024, 512, D]$. Each set abstraction layer has three fully connected layers to learn features. The point clouds were perturbed with random rotations, jitter and dropout for data augmentation and to build robustness when testing on novel objects in unknown poses. We choose $D=300$ and $K=128$, and $L=6$ as the parameters for our GCN network, as introduced in the main paper.

A.3 Experimental Details and Evaluation Metrics

In this section, we explain further the metrics used to evaluate generalization in the dataset. Since k -fold cross validation in any held-out setting will evaluate all grasps in the dataset, we can compute Average Precision (AP) scores for any category, i.e., any object instance, object class, or task. We then compute an mAP averaged over object instances, mAP averaged over object classes, and mAP over tasks as shown in Table 1 in the main paper. In each fold, grasps from 25% of the categories will be used for testing while remaining grasps will be used for training and validation.

In all experiments, we only evaluate tasks that are valid for a given input object class. This makes sense from an evaluation perspective as it separates the problem of predicting applicable tasks for objects from task-driven grasping. It also makes the comparison to methods using object-task information fair since the models do not have to decide whether the object-task pair is valid.

A.4 Comparison to SG14000

We want to demonstrate that grasping models trained on our GCNGrasp dataset generalize to other task-oriented grasping datasets. We show transfer learning results on SG14000, since it has the most similar setting by providing objects with their corresponding point clouds and grasps in $SE(3)$. Since SG14000 does not come with any semantic knowledge, we use the Semantic Grasping Network (SGN) + *word embedding* as the backbone model instead of GCNGrasp. SG14000 is significantly smaller and less diverse with 14K grasps. The five object categories and seven tasks were resolved to WordNet synsets to have complete overlap with TaskGrasp. The test dataset was held-out based on grasps, hence may include known object classes and tasks during evaluation. The model trained on SG14000 performed well when tested on itself. However, it failed to generalize to the more diverse TaskGrasp with only a 17% increase over a random baseline. It is not surprising that the model trained on TaskGrasp was able to generalize to the held-out test set in TaskGrasp. It also performed well when tested on the SG14000 test set though it did not outperform the model trained on SG14000. This is owing to several reasons. First, the point clouds in SG14000 were incomplete with a lot of self-occlusions (since objects were scanned from just a single view) whereas our point clouds are constructed based on scans from multiple view points. This could affect the performance of the Object and Grasp Encoder based on PointNet [17]. Second, SG14000 has a dataset bias since it models the effects of material and object state on grasps, while we focus on object geometry. Another reason could be dataset imbalance in TaskGrasp as we do not have sufficient quantities of certain categories (bowls, bottles) in comparison. Lastly, SG14000 has some grasps in free space (which we filtered out in our dataset) where our model predicts a high score. This can be corrected by adding unstable grasps as hard negatives during training, similar to prior work [7, 6].

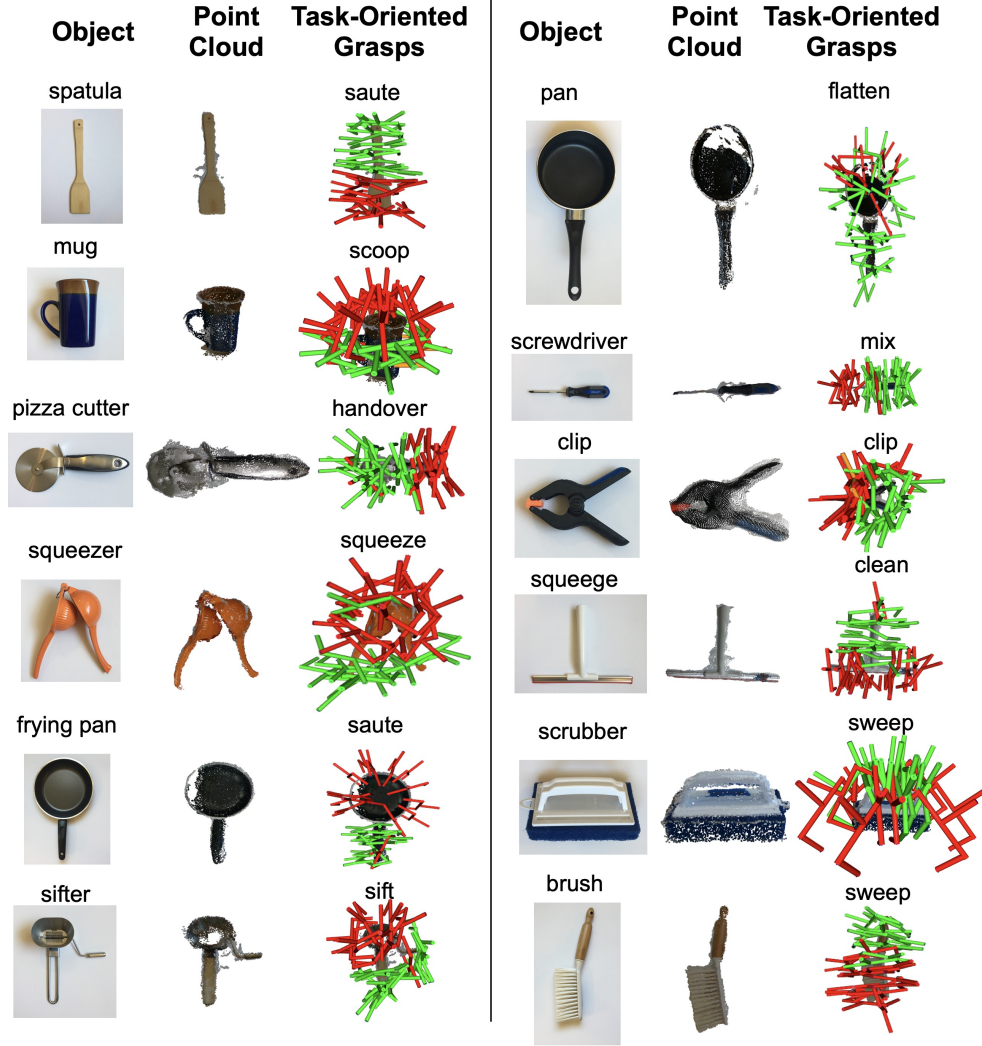


Figure 5: Example point clouds and grasps from our TaskGrasp dataset. Grasps colored in green and red are successful and incorrect respectively for task-oriented grasping.

Table 4: Cross generalization on TaskGrasp and SG14000

Train Dataset	Held-out Test Grasps (mAP)	
	TaskGrasp	SG14000 [12]
TaskGrasp	76.2	52.3
SG14000	25.1	62.7
Random	7.9	24.8

A.5 Additional Dataset Examples and Analysis

Additional examples of task-oriented grasps for several objects are shown in Fig 5. Fig 6 shows a histogram of the number of object instances per object category. In TaskGrasp, each object category is resolved to a WordNet synset [15] (displayed on the vertical axis). On average, each object category has approximately 2.5 instances. There are three main sets of categories based on the distribution: categories with high (cups, spatulas etc.), medium (paint roller, knife, etc.) and low (sifter, mixing bowl, etc) incidence. It is also noteworthy that our dataset is restricted to objects that can be effectively scanned by a depth sensor i.e. objects that are not transparent or too specular.

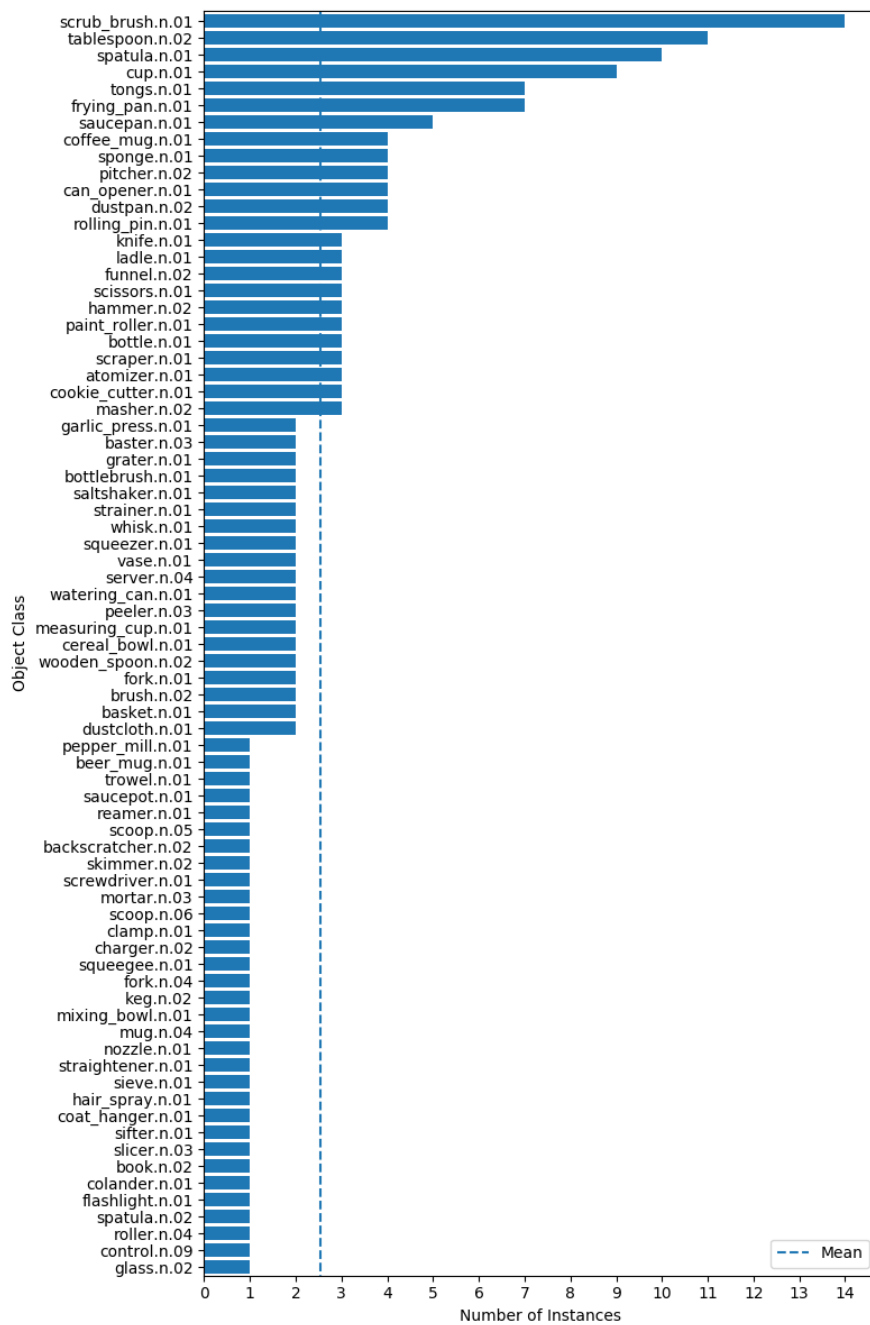


Figure 6: Number of object instances for each object category. The dotted vertical line is the average number of instances/category. The category is represented by its WordNet synset on the y-axis.

There are two main types of semantic knowledge provided in our dataset. The first type of knowledge are “Used-For” relations between tasks and object categories. We list all the tasks suitable for each object category in Table 5. As explained in the paper, these task-object category combinations were collected in the first stage of annotation before labelling the task-oriented grasps. The second type of knowledge is the semantic hierarchy for objects, which includes a diverse set of object concepts interlinked by “Is-A” relations. The hierarchy provides useful information for task-oriented grasping as objects in the same subtree of the hierarchy may share similar geometry or functionality.

A.6 Analysis on GCNGrasp Predictions

Next we visualize AP scores for each task from GCNGrasp predictions trained on TaskGrasp in Fig 7. The AP scores for all tasks were computed with cross validation as detailed in the main paper. The red bar corresponds to AP score with predictions from a random model (averaged over five seeds) while the red and blue bar cumulatively represents the model AP score. Overall, GCNGrasp performed better than random predictions, though some tasks are more challenging than others. For instance, juice, saute and screw are harder tasks (with low random prediction scores) compared to handover and poke. Tasks that represent more creative than prototypical uses of an object are typically more ambiguous and challenging to label. Yet, our model is able to improve over random predictions even in these challenging tasks.

A.7 Annotation Interface on Amazon Mechanical Turk

We now describe the process for annotating task-oriented grasps for a given object and task. Instead of manually labelling the dataset like prior work [12], we scale the labelling effort for our larger dataset using Amazon Mechanical Turk (AMT). As explained in the main paper, we have a two-stage procedure to reduce the annotation cost.

In the first stage, we gathered suitable tasks for each object. In each of our labeling tasks, we presented annotators with the image of the object, the object category, and the task, as shown in the example in Fig 9.

In the second stage, we collected labels for task-oriented grasping only if the task applies to the object (filtered from the first stage). For each grasp, we presented the annotators with an image of the object, visualizations of the object point cloud and the grasp from 3 different angles, the object category, and the task. Examples of this graphical interface are shown in Fig 10. Since labelling on 3D data from a 2D interface is highly ambiguous and challenging, we presented visualizations from multiple views.

In our pilot study, we found that annotators do not always agree with each other since the notion of task-oriented grasping for robots is generally ambiguous to non-expert users. As a result, we provided them with seven guidelines concerning different aspects of task-oriented grasps, such as functionality, stability, creative uses of objects and safety. We empirically found that most crowd workers were able to understand the annotation task, though it takes some time to internalize the object shape and task description before deciding on the label. We used a qualification test to recruit crowd workers. The test had 26 questions which the authors annotated the ground-truths for. Fig 8 shows the results from the qualification test. We rank the test participants and qualify the top 30% for the final annotation round. To improve annotation quality, each task-oriented grasp is presented to three annotators and the final label is decided on a majority vote. As shown in Fig 8, we found that having a redundant set of three crowd workers for each question was a good trade-off between annotation cost (which scales linearly with redundancy) and label quality. Both the guidelines and the qualification test effectively improved the quality and consistency of labels.

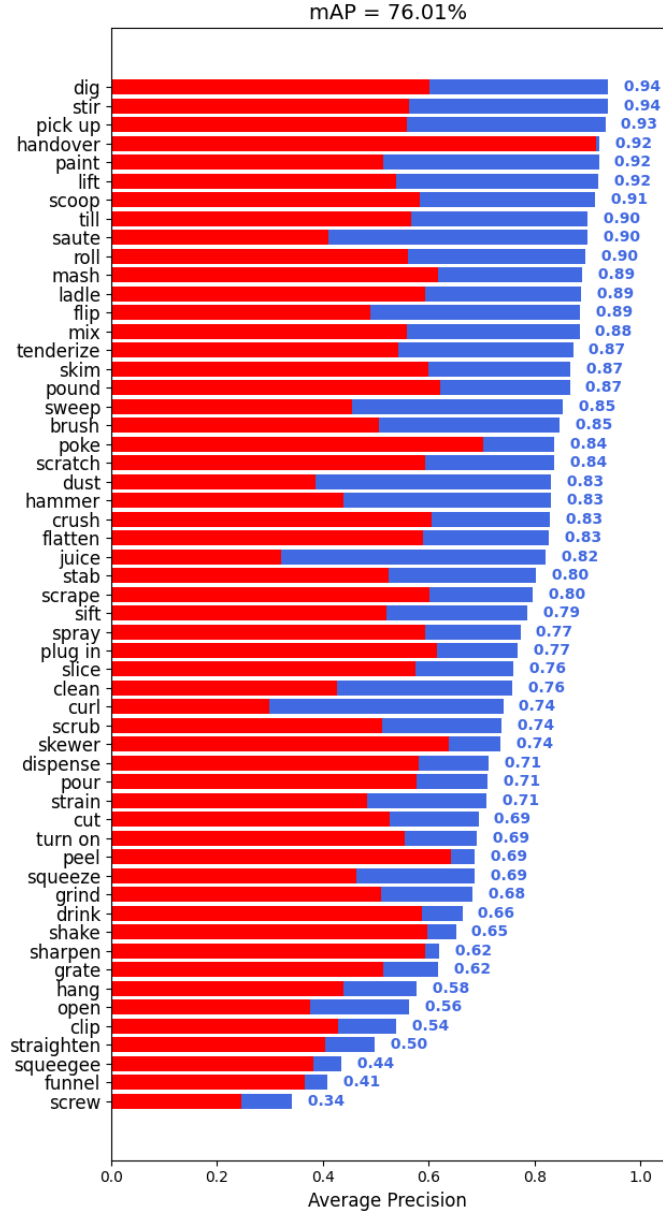


Figure 7: mAP across tasks for GCNGrasp predictions. The red bar is for AP predictions by a random model while the red and blue cumulatively represents the model prediction.

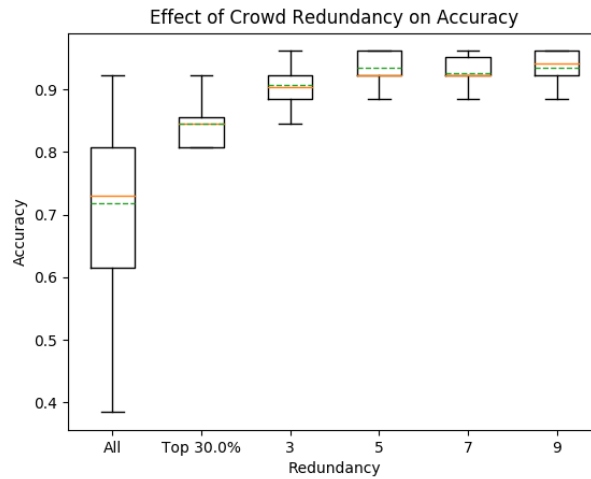


Figure 8: Results of qualification test. The left-most entry is the results for all participants in the test. The box for each entry extends from the lower to upper quartile value, with the dotted and solid line at the mean and median accuracy respectively. The second entry from the left is the top 30% of the crowd workers who are recruited for the actual annotation. For the remaining entries, 10 sets of crowd turkers were randomly selected from the top 30% cohort with a redundant set size of $K=3,5,7,9$. The average accuracy for each set size is plotted with the performance saturating after $K=3$. As such, we used three annotators for labelling each task-oriented grasp.

Commonsense Knowledge for Assistant Robots

The purpose of this task is to provide examples for an assistive robot to intelligently use household objects. For each example object such as a mug, we will give you an image and a task such as pouring something, and you will tell us whether you think that object is suitable for that task.

Example



1. Is the object **mug** suitable for **pouring something**?

A: **Yes**

2. Is the object **mug** suitable for **cutting something**?

A: **No**

Questions

1. Is the object **bowl** suitable for **squeezing something**?



☐ Yes ☐ No

Figure 9: Example from the first annotation stage to gather labels for task suitability for each object.

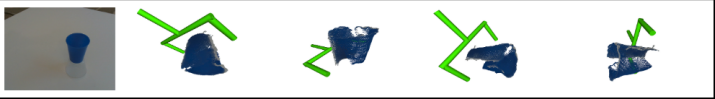
Table 5: Object Task Combinations

Object Class	Suitable Tasks
atomizer.n.01	pour, clean, squeeze, dispense, handover, spray
backscratcher.n.02	pick up, turn on, shake, scrape, handover, scoop, scratch, mix
basket.n.01	pick up, dispense, lift, handover, scoop, hang
baster.n.03	poke, dispense, squeeze, handover, drink, spray
beer.mug.n.01	pour, scoop, ladle, flatten, dispense, hammer, lift, mash, handover, crush, drink, pound
book.n.02	pound, crush, handover
bottle.n.01	pour, clean, juice, squeeze, poke, shake, flip, flatten, dispense, hammer, straighten, handover, pound, drink, spray, mix
bottlebrush.n.01	clean, dust, scrub, scrape, straighten, brush, mash, handover, crush, scratch
brush.n.02	handover, brush, sweep, paint
can.opener.n.01	cut, open, squeeze, poke, pick up, squeeze, screw
cereal.bowl.n.01	pour, scoop, ladle, pick up, dispense, handover, drink, mix
charger.n.02	plug in, turn on
clamp.n.01	clip, straighten, lift, squeeze, crush, hang
coat.hanger.n.01	hang, straighten, pick up, handover
coffee.mug.n.01	pour, skim, clean, scoop, ladle, pick up, grind, shake, flatten, dispense, dig, lift, handover, drink, pound
colander.n.01	pour, skim, juice, poke, dig, funnel, handover, crush, sift, scoop, strain, pound
control.n.09	turn on
cookie.cutter.n.01	slice, cut
cup.n.01	pour, skim, clean, scoop, ladle, till, saute, poke, pick up, shake, tenderize, flatten, dispense, dig, lift, crush, handover, pound, drink, mix
dustcloth.n.01	clean, dust, brush, sweep
dustpan.n.02	clean, dust, pick up, flip, dispense, lift, handover, scoop, sweep
flashlight.n.01	turn on, handover
fork.n.01	skewer, juice, ladle, poke, stir, pick up, handover, stab, flip, grate, dig, scrape, curl, lift, funnel, mash, scoop, scratch, strain, mix
fork.n.04	till, stir, stab, dig, scrape, handover, scratch
frying.pan.n.01	pour, saute, stir, pick up, flip, tenderize, flatten, dispense, hammer, lift, mash, handover, crush, pound, scoop, mix
funnel.n.02	pour, scoop, pick up, stab, dispense, scrape, squeeze, funnel, roll, strain, mix
garlic.press.n.01	grind, flatten, hammer, squeeze, mash, handover, crush, scratch
grater.n.01	cut, slice, grind, tenderize, grate, scrape, scratch, strain
hair.spray.n.01	roll, spray, handover
hammer.n.02	tenderize, flatten, hammer, straighten, mash, crush, pound
keg.n.02	flatten, dispense, drink, pour
knife.n.01	cut, peel, poke, slice, stab, clip, scrape, sharpen, scratch
ladle.n.01	pour, skim, scoop, ladle, poke, saute, stir, pick up, dispense, hammer, scrape, lift, handover, sift, roll, drink, strain, sweep, mix
masher.n.02	cut, juice, poke, stir, grind, tenderize, flatten, hammer, mash, handover, crush, pound, mix
measuring.cup.n.01	pour, scoop, ladle, pick up, dispense, dig, lift, handover, drink
mixing.bowl.n.01	pour, dispense, pick up, mix
mortar.n.03	pour, stir, grind, tenderize, flatten, pound, mash, handover, crush, sift, scoop, mix
mug.n.04	pour, drink, scoop, handover
nozzle.n.01	dispense, spray
paint.roller.n.01	squeeze, paint, tenderize, flatten, dispense, brush, handover, roll, pound
pancake.turner.n.01	cut, skim, ladle, pick up, crush, scoop, saute, turn on, flatten, scrape, handover, mix, pound, pour, stir, poke, flip, dig, lift, mash, sift
peeler.n.03	peel, slice, grate, scrape, scratch
pepper.mill.n.01	grind, crush, handover
pitcher.n.02	pour, scoop, ladle, stir, pick up, shake, flatten, dispense, lift, handover, drink, mix
reamer.n.01	plug in, juice, scrape, mash, handover
roller.n.04	roll, clean, lift, handover
rolling.pin.n.01	poke, tenderize, flatten, hammer, straighten, mash, handover, crush, roll, pound
saltshaker.n.01	pour, dispense, crush, handover, strain, shake
saucepan.n.01	pour, scoop, ladle, saute, stir, pick up, shake, flatten, dispense, dig, lift, mash, handover, crush, drink, mix
saucepot.n.01	pour, saute, dispense, lift, mash, handover, crush, drink, mix
scissors.n.01	cut, open, poke, slice, handover, stab, clip, scrape, curl, straighten, sharpen, scratch
scoop.n.05	clean, pick up, flip, dig, lift, handover, sift, scoop, strain, mix
scoop.n.06	clean, stir, pick up, dig, lift, handover, scoop, pound
scraper.n.01	peel, clean, squeeze, slice, stir, stab, flatten, dig, scrape, straighten, lift, handover, scoop, scratch
screwdriver.n.01	skewer, open, poke, stab, dig, screw, hang, scratch, mix
scrub.brush.n.01	clean, dust, paint, poke, stir, scrub, shake, stab, flip, tenderize, flatten, scrape, straighten, mix, funnel, handover, brush, scratch, sweep, pound
server.n.04	ladle, stir, pick up, curl, lift, handover, sift, scoop, hang, mix
sieve.n.01	sift, dispense, strain, skim
sifter.n.01	sift, dispense, strain
skimmer.n.02	skim, ladle, saute, stir, pick up, flip, scrape, handover, sift, scoop, strain
slicer.n.03	cut, peel, open, juice, slice, saute, grate, screw, mix
spatula.n.01	skim, poke, saute, stir, pick up, scrub, flip, dig, lift, crush, handover, scoop, scratch, mix
spatula.n.02	skim, stir, pick up, flip, flatten, scrape, lift, mix
sponge.n.01	skim, clean, squeeze, dust, poke, scrub, scrape, brush, handover, drink, scratch, sweep
squeeze.n.01	clean, squeeze, scrub, scrape, handover
squeezer.n.01	juice, flatten, squeeze, mash, handover, crush, drink, pound
straightener.n.01	flatten, pick up, straighten
strainer.n.01	skim, stir, pick up, shake, flip, dispense, lift, funnel, handover, crush, sift, scoop, strain, sweep, mix
tablespoon.n.02	skim, ladle, pick up, dispense, curl, scoop, scratch, saute, turn on, stab, flatten, scrape, squeeze, handover, mix, pound, pour, stir, poke, flip, dig, lift, mash, sift, drink, strain
tongs.n.01	squeeze, pick up, clip, dispense, crush, scoop, scratch, saute, turn on, stab, scrape, squeeze, funnel, handover, shake, skewer, mix, stir, straighten, poke, flip, lift, roll
towel.n.01	till, slice, stir, poke, stab, flip, flatten, hammer, dig, scrape, lift, crush, scoop, scratch, mix
vase.n.01	pour, scoop, tenderize, dig, straighten, lift, handover, drink, shake
watering.can.n.01	pour, scoop, poke, dispense, funnel, drink, shake
whisk.n.01	mix, stir, brush, handover
wooden.spoon.n.02	skim, ladle, poke, saute, stir, pick up, flatten, dig, scrape, lift, mash, handover, pound, scoop, mix

[Questions](#)


Questions

1. The robot will grasp the **cup** as shown below. Is this grasp suitable for **pouring something**?




☐ Yes
 ☐ No
 ☐ Unclear

2. The robot will grasp the **spray bottle** as shown below. Is this grasp suitable for **dispensing something**?



☐ Yes
 ☐ No
 ☐ Unclear

3. The robot will grasp the **watering can** as shown below. Is this grasp suitable for **scooping something**?



☐ Yes
 ☐ No
 ☐ Unclear

4. The robot will grasp the **spoon** as shown below. Is this grasp suitable for **poking something**?




Figure 10: Example questions from the second stage of annotation to label task-oriented grasps.