Language Embedded Radiance Fields for Zero-Shot Task-Oriented Grasping

Adam Rashid*, Satvik Sharma*, Chung Min Kim, Justin Kerr, Lawrence Yunliang Chen
Angjoo Kanazawa, Ken Goldberg

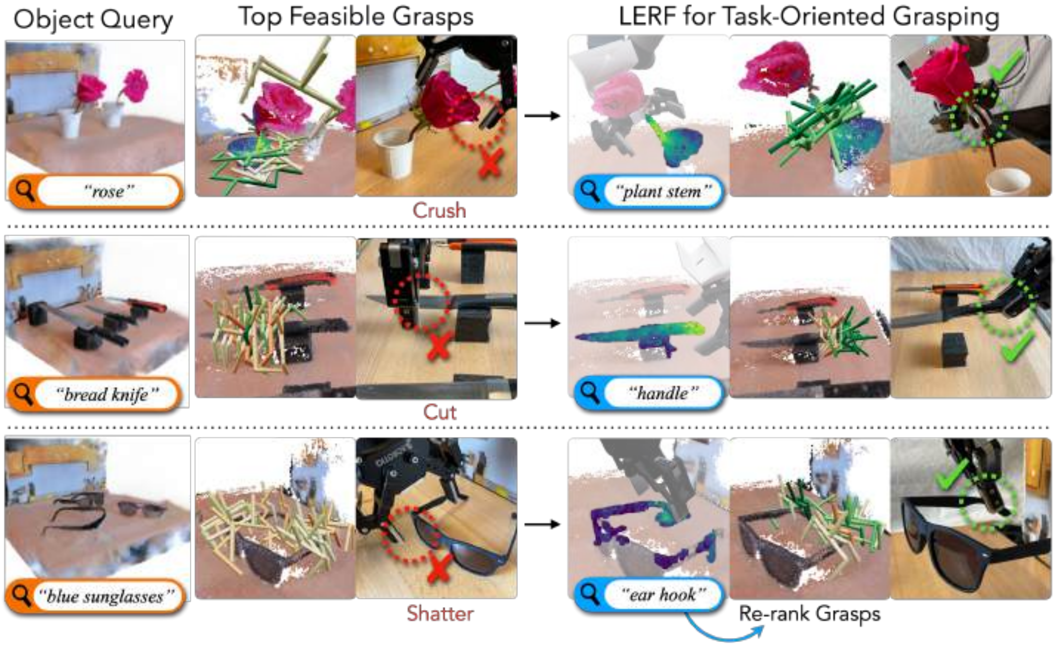


Figure 1: Learning-based grasp planners primarily consider object geometry, potentially leading to dangerous grasps for robots and their environment. LERF-TOGO uses natural language to select the target object with LERF [1] (in **orange**), and resamples grasps towards specific object parts using conditional LERF queries (in **blue**) for safe, task-oriented grasps.

Abstract: Grasping objects by a specific part is often crucial for safety and for executing downstream tasks. Yet, learning-based grasp planners lack this behavior unless they are trained on specific object part data, making it a significant challenge to scale object diversity. Instead, we propose LERF-TOGO, Language Embedded Radiance Fields for Task-Oriented Grasping of Objects, which uses vision-language models zero-shot to output a grasp distribution over an object given a natural language query. To accomplish this, we first reconstruct a LERF of the scene, which distills CLIP embeddings into a multi-scale 3D language field queryable with text. However, LERF has no sense of objectness, meaning its relevancy outputs often return incomplete activations over an object which are insufficient for subsequent part queries. LERF-TOGO mitigates this lack of spatial grouping by extracting a 3D object mask via DINO features and then conditionally querying LERF on this mask to obtain a semantic distribution over the object with which to rank grasps from an off-the-shelf grasp planner. We evaluate LERF-TOGO's ability to grasp task-oriented object parts on 31 different physical objects, and find it selects grasps on the correct part in 81% of all trials and grasps successfully in 69%. See the project website at: lerftogo.github.io

Keywords: NeRF, Grasping, Semantics, Natural Language

* Denotes Equal Contribution, Alphabetically ordered
UC Berkeley

7th Conference on Robot Learning (CoRL 2023), Atlanta, USA.

1 Introduction

Many common objects must be grasped appropriately to avoid damage or facilitate performing a task: a knife by its handle, a flower by its stem, or sunglasses by their frame. Learning-based grasping systems exhibit impressive robustness on grasping arbitrary objects [2, 3, 4, 5, 6, 7, 8, 9, 10], but these systems typically measure grasp success based on whether the object was lifted [11, 12, 13, 14, 15]. Critically, these methods ignore an object's semantic properties: even if a robot could locate your favorite sunglasses, rather than safely grasp at the frame it may shatter the lenses. This ability to grasp an object part based on a desired task and constraints is called *task-oriented grasping*, and while well-studied [16, 17, 18, 19, 20, 21], previous methods collect specific object affordance datasets and struggle to scale to a diverse set of objects. Instead, the flexibility of natural language has the potential for specifying what and where to grasp. In this work, we propose *LERF* for *Task-Oriented Grasping on Objects* (LERF-TOGO), a method which enables task-oriented grasping through natural language by using large vision-language models in a zero-shot manner.

LERF-TOGO takes as input an object and a task-orienteed object part name in natural language (i.e. *"flower; stem"*), and outputs a ranking over viable grasps on this object from which the robot should grasp. We build on recent work Language Embedded Radiance Fields (LERF) [1], which takes in calibrated RGB images and trains a standard NeRF in tandem with a scale-conditioned CLIP [22] feature field. Given a sentence prompt query, it outputs a 3D *relevancy* heatmap representing similarity to the query. However, these heatmaps may fail to highlight the full object (e.g., highlight only the bristles of a brush), which may cause issues when directly deployed to a task-oriented grasping task (grasp the "handle" of a brush). LERF-TOGO improves upon LERF's capabilities by predicting a 3D object mask using 3D DINO [23] features explicitly during inference. We propose a method of conditional LERF querying which restricts an object sub-part query to the object mask, leveraging the multi-scale nature of LERF to isolate specific regions within an object. LERF-TOGO then uses GraspNet [15] to generate grasps, re-ranking them based on the geometric and semantic distributions.

We implemented a system with appropriate regularizations which allows LERF-TOGO to operate on a physical robot and evaluate its semantic grasping capabilities on 39 common household objects. In experiments, 96% of generated grasps are on the correct object, 82% on the correct object part, and 69% result in a successful grasp.

This work contributes LERF-TOGO, an algorithm for producing task-based semantic grasp distributions over an object by first extracting a coarse 3D object mask with DINO [23] features to locally grow a relevant region, then conditioning a LERF query on this mask to isolate sub-parts of an object. We design a robotic system which integrates LERF-TOGO on a physical robot to reconstruct a LERF of a scene, then execute task-oriented grasps through natural language to grasp semantically meaningful object parts. See the project website at: lerftogo.github.io

2 Related Work

2.1 Task-Oriented Grasping

Task oriented grasping studies how to grasp objects by specific parts based on a use case. It has been studied by probabilistically modeling human grasps [19], extracting geometric features from labeled object parts [21], training on part-affordance datasets in simulation [24], or transferring category-specific part grasps to new instances [25]. Recent works [26, 17] train object-part grasp networks by leveraging object part and manipulation affordance datasets for a range of household objects. In contrast, LERF-TOGO's usage of off-the-shelf vision-language models trained at scale does not require training on affordance datasets, capturing long-tail objects more easily. Given 3D grasp regions on an object, Song et al. [27] demonstrate that biasing an off-the-shelf grasp planner towards these regions is a viable approach to sampling task-oriented semantic grasps. Kokic et al. [18] use videos of humans interacting with objects to guide grasps towards the same part. Decomposing objects into parts has also long been studied as a co-segmentation task in vision [28, 29]. Recent

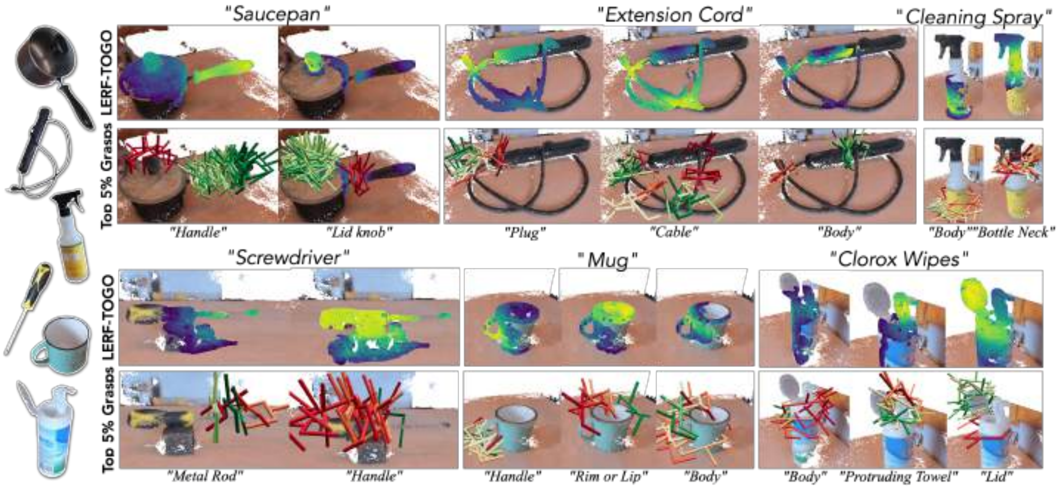


Figure 2: **Task-Oriented Grasps**: LERF-TOGO grasps target objects by different parts with different natural language queries (in quotes). Left: Crops of target objects. Right: Top row visualizes the relevancy distribution across the object mask, and bottom row shows top 5% of resampled grasps.

approaches use pretrained vision features to discover common parts within sets of objects [30]. This technique has been applied at scale to segment parts of objects based on a canonical object [31] or detect object affordances from example images of human usage [16]. Though effective, it assumes access to a canonical image of each object and pre-existing part labels or demonstrations, which are restrictive in real-world applications. LERF-TOGO instead uses free-form language queries to isolate parts of objects in a 3D reconstruction.

2.2 Neural Radiance Fields (NeRF)

Neural Radiance Fields (NeRF) [32] are an attractive representation for high quality scene reconstruction from pose RGB images, with an explosion of recent work on visual quality [33, 34, 35, 36, 37, 38, 39], large-scale scenes [40, 41, 42], optimization speed [43, 44, 45, 46], dynamic scenes [47, 48, 49], and more. Because of its high-quality reconstruction and differentiable properties, NeRF has been widely explored in robotics for navigation and mapping [33, 50, 51, 52], manipulation [53, 54, 55, 56], and for synthetic data generation [57]. This work is most similar to works such as Evo-NeRF [55] which use NeRF as a real-time scene reconstruction to grasp objects. However, in contrast to previous works which only use RGB information, in this work we must include additional semantic information in 3D to select grasps falling on relevant target objects.

Several prior works explore using semantic outputs inside NeRF. Semantic-NeRF [58], Panoptic Lifting [59], and Panoptic Neural Fields [60] distill semantic categories from semantic segmentation networks into 3D to improve the 3D consistency of labels, particularly noting the denoising effect of averaging multiple views. Other works such as Distilled Feature Fields [61] or Neural Feature Fusion Fields [62] distill feature vectors from DINO and LSeg [63], and show they can be used for editing and scene segmentation. We build off of LERF [1], which is described in the next section.

2.3 LERF Preliminaries

Language Embedded Radiance Fields (LERF) [1] is a recent representation that distills CLIP features into a NeRF. LERF inputs RGB images with camera poses and outputs a 3D field of DINO embeddings as well as a scale-conditioned CLIP field. This supports querying points in 3D for CLIP embeddings at different physical scales, capturing different semantics given different amounts of context. Given a text query, a relevancy value (from 0 to 1) can be generated at any 3D point by calculating the cosine similarity between LERF-queried embeddings and the CLIP embedding of query text. During this query process, a grid search on the scale parameter retrieves the scale with the highest activation. LERF is particularly attractive for task-oriented grasping because 1) its multi-scale parameterization allows queries at both object-level and part-level scales 2) LERF uses outputs from a pre-trained CLIP model without fine-tuning, which supports a variety of long-tail object queries not included in object or part segmentation datasets. LERF, however, tends to produce nonuniform activations on object queries because it lacks spatial grouping as shown in Fig. 5. In this work we show how to explicitly use the DINO feature field to obtain object masks to enable down-stream object part queries related to task-oriented grasping.

2.4 Open-Vocabulary Detectors

Open-vocabulary object detection attempts to output masks or bounding box detections given text prompts as input. OVD [64] proposes a two-stage training pipeline, where they first learn a visual-semantic stage using image-text pairs and then learn object detection using object annotations from a set of base classes. OpenSeg [65] employs a similar joint learning framework for simultaneously training a mask prediction model and learning visual-semantic alignments for each mask by aligning captions and masks. Some methods [66, 67, 68] distill CLIP embeddings into a visual encoder by using labels from detection or classification datasets, or fine-tune a lightweight detection head on top of a pretrained backbone [69]. However, fine-tuning on detection datasets severely bottlenecks the language understanding capability of detection models [1, 61, 70], and as noted by the authors of OV-Seg [67], these networks face challenges in decomposing hierarchical components of the original masks, including object parts. LERF circumvents the usage of region proposals or fine-tuning by incorporating language embeddings densely in 3D, making it an attractive approach for effective part decomposition.

2.5 Natural Language in Robotics

With the advent of large pretrained language and vision-language models, several works have explored building 3D map representations to guide robot navigation. VL-Maps [71] and OpenScene [72] build a 3D language embedding from pretrained open-vocab detectors [73, 63], which can be used to navigate to target queries. CLIP-Fields [74], ConceptFusion [75], and NLMaps-SayCan [76] take more of a region-proposal based approach, querying CLIP on the outputs of some region proposal methods and fusing them into 3D point clouds for downstream navigation tasks. Region-based zero-shot methods retain more language understanding than fine-tuned features but run the risk of missing objects by insufficiently masking input images. Semantic Abstraction [77] avoids this by extracting relevancy from vision-language models using Chefer et al. [78] and uses these for composing multiple language queries with spatial relationships. Language has also been studied in the context of robot manipulation. Mees and Burgard [79] use the MAttNet [80] vision-language model for object rearrangement, and CLIPort [81] uses language understanding from CLIP to train a language-conditioned pick and place module from demonstrations. PerAct [82] uses language-conditioned demonstrations with a 3D scene transformer to learn diverse tasks, MOO [83] uses the outputs from Owl-ViT to condition a manipulation policy for grasping objects, and large-scale demonstration datasets like RT-1 [84] train on massive language-conditioned demonstration trajectories. In contrast to many other language-conditioned approaches, LERF-TOGO uses internet-scale vision models purely zero-shot and does not require fine-tuning on demonstrations or robot exploration.

3 Problem and Assumptions

Given a planar surface (table or workbench) containing a set of objects, the objective is for a robot to grasp and lift a target object specified using natural language. This query (e.g., *"sunglasses; ear hooks."*) includes both the object query (*"sunglasses"*) and the object part query, which specifies the part to grasp the object by (*"ear hooks"*). We experiment with lifting this assumption in Sec. 5 by leveraging an LLM for providing part queries. We assume access to a robot manipulator with

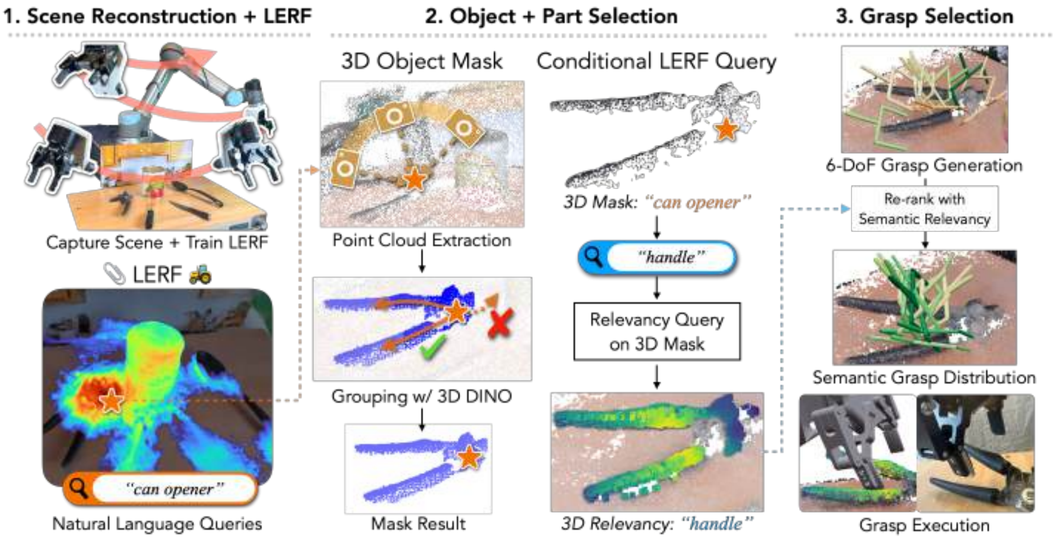


Figure 3: **LERF-TOGO Pipeline**: After reconstructing the scene with a wrist-mounted camera, we render an object-centric point cloud around the highest LERF activation. We next extract a 3D object mask by flood-filling the DINO features in this point cloud, condition an object part query on this object mask. Finally, we sample grasps and re-rank them according to 3D object part relevancy.

a parallel jaw gripper and calibrated wrist-mounted RGB camera, and the objects in the scene are graspable by the robot. We also assume the object query specifies a single object that is present in the scene.

4 Method

Given an object and object part query, LERF-TOGO outputs a ranking of viable grasps on the object part. To accomplish this, it first captures the scene and reconstructs a LERF. Given a text query, LERF can generate a 3D relevancy map that highlights the relevant parts of the scene (Sec. 2.3). Second, a 3D object mask is generated using the LERF relevancy for the object query and DINO-based semantic grouping (Sec. 4.1). Third, a 3D part relevancy map is generated with a conditional LERF query over the object part query and the 3D object mask (Sec. 4.2). The part relevancy map is used to produce a semantic grasp distribution.

4.1 3D Object Extraction

An important limitation of LERF is its lack of spatial grouping within objects: for example given *"can opener"*, LERF tends to highlight regions of the object that most obviously identify that object (e.g. the metal cogs on the can opener as shown by the orange star in Fig. 3). However, since the region that visually identifies the object and the desired grasp location (e.g. handle) can differ significantly, this is problematic. LERF inherently exhibits such local behavior because it trains on local crops of input images, causing CLIP embeddings surrounding the handle to be unaware if it belongs to a can opener. LERF-TOGO overcomes this by finding a 3D object mask given a language query, which groups the object part together with the LERF activation. To create the object mask, we leverage the 3D DINO embeddings [23] (self-**DI**stillation with **NO** labels) present within LERF during inference, because DINO embeddings have been shown to exhibit strong object awareness and foreground-background distinction [30, 23, 85].

First, we obtain a coarse object localization from LERF by rendering a top-down view of the scene and querying the object. We produce a foreground mask by thresholding the first principal component of the top-down rendered DINO embeddings, and constrain the relevancy query to this mask to

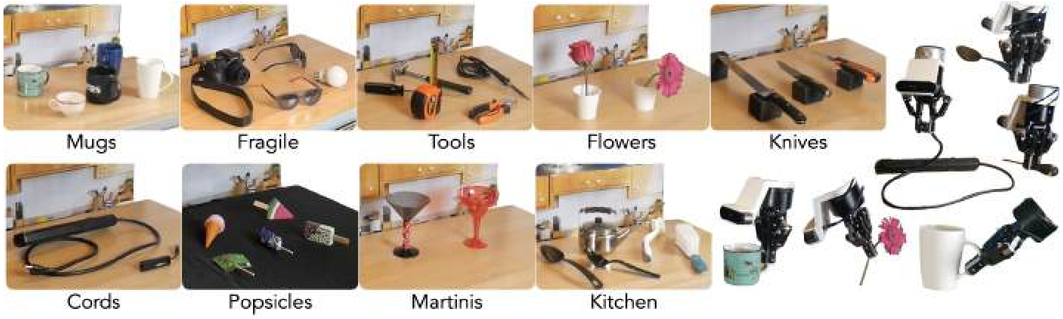


Figure 4: **Left**: Select scenes used during experiments, **Right**: example grasps using LERF-TOGO.

find the most relevant 3D point. We then refine this single-point localization into a complete object mask. We render an object-centric point cloud around this 3D point by deprojecting NeRF depth from multiple views, and then iteratively grow the object mask by including neighboring points to the frontier which lie within a threshold DINO similarity (similar to floodfill). The output of this process is a set of 3D points lying on the target object. See the Appendix for more details.

4.2 Conditional LERF Queries

Another important challenge of using CLIP is its tendency to behave as a bag-of-words [22]: the activation for *"mug"* behaves very similarly to *"mug handle"* because CLIP latches onto individual words, not the grammatical structure of sentences. To mitigate this phenomenon, LERF-TOGO introduces a conditional method of querying LERF relevancy by composing two related queries, similarly to how composing prompts has shown promise in generative modeling for guiding specific properties [86]. Because LERF is scale-conditioned, during inference it searches over scales for a given query and returns the relevancy at the scale with the highest activation. To condition a LERF query, LERF-TOGO searches only on the points within the 3D object mask. Intuitively, this results in a distribution over the object's 3D geometry representing the likelihood that a given point is the desired object part, which can be used for biasing grasps towards this region.

4.3 Grasping

Grasp Sampling Ensuring complete coverage of grasps on objects is critical to avoid missing specific object parts. We use GraspNet [15], which can generate 6-DOF parallel jaw grasps from a monocular RGBD point cloud, but from a single view it often misses key grasps on target object parts. To mitigate this, and to leverage the full 3D geometry available within NeRF, we create a hemisphere of virtual cameras oriented towards the scene's center. For every virtual camera, we convert the scene's point cloud to the camera coordinate frames before providing it as input to the pretrained GraspNet model. To obtain the final set of grasps for the scene, we combine the generated grasps from the virtual cameras using non-maximum suppression to remove duplicates.

Grasp Ranking Given the grasps sampled in the previous step (the *geometric* distribution), we now combine it with the semantic distribution across an object obtained from LERF-TOGO. The semantic score s_{sem} for a given grasp is computed as the median LERF relevancy of points within the grasp volume. The geometric score s_{geom} is the confidence output from GraspNet, indicating grasp quality based on geometric cues. To balance relevance and success likelihood, we combine the grasp score $s = 0.95 s_{sem} + 0.05 s_{geom}$ to ensure that we consider the most relevant grasps while slightly biasing towards confident grasps.

4.4 Scene Reconstruction

The robot uses a wrist-mounted camera to capture the scene with a hemispherical trajectory centered at the workspace, similar to Evo-NeRF [55]. The capture has a radius of 45 cm and arcs from $\pm 100°$

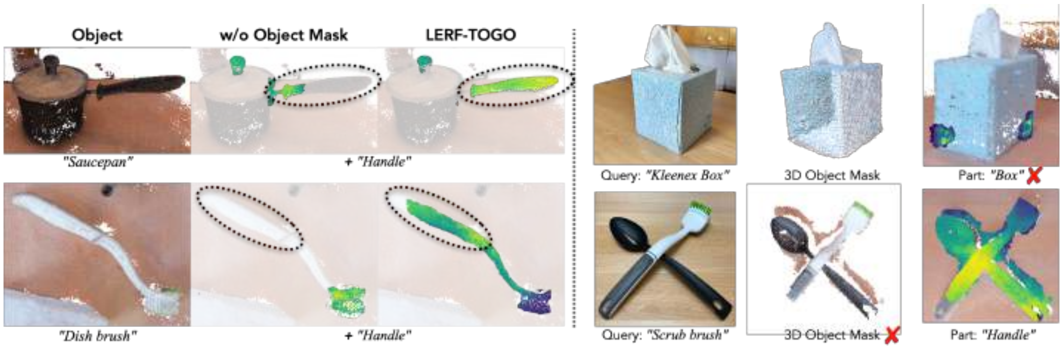


Figure 5: **Ablation and Limitations**: *Left*: Without 3D object masking and conditional querying, LERF cannot capture oblong object shapes. *Right*: CLIP can sometimes fail on generic prompt queries, like the poor activation on the box. Additionally, LERF-TOGO struggles with groups of connected objects as the 3D object mask groups them all together.

	ConceptFusion [70]	LERF-TOGO
Correct Object	77%	96%
Correct Part	39%	82%
Successfuly Lifted	–	69%

Table 1: **Part-Based Grasping Results:** Results are reported across 49 different prompts and 12 scenes. See the Appendix for a complete list of scenes and queries.

around the workspace horizontally and an inclination range of 30° to 75°. We capture images while the arm moves at 15 cm/sec at a rate of 3 hz, resulting in around 60 images per capture. We discard blurry images by analyzing the variance of the image Laplacian, ensuring the images are high quality. While the robot moves, we pre-process each image to extract DINO features, multi-scale CLIP, and ZoeDepth [87], which are used during LERF training. See Appendix B for additional training details.

5 Experiments

Part-Oriented Grasping We evaluate LERF-TOGO on a wide variety of 31 different objects and 49 total object parts to grasp (Fig. 4). For each object, we select an object query by describing it sufficiently to unambiguously differentiate between other objects in the scene. We use semantic descriptions when possible, and add visual descriptions *only* when such descriptions are ambiguous (i.e using color to differentiate multiple mugs in a scene). We provide a part query for each object by describing a natural place for a robot to grasp and lift (i.e. "handle", "plant stem", "ear hook", "frame"). In addition, several objects include multiple different grasp locations. A grasp is successful if it lifts the correct object using the appropriate subpart at least 10cm vertically, and the object remains securely within the gripper jaws throughout. For each query, we measure 1) whether the selected grasp was on the correct *object*, 2) whether the selected grasp was on the correct object *part*, and 3) whether the grasp successfuly *lifted* the object from the table. Every scene is reconstructed once in the beginning, after which the objects are removed sequentially (i.e., objects are removed one-by-one) with no updates in the scene representation. For a full list of object queries see the Appendix.

Task-Oriented Grasping LERF-TOGO accepts a natural language part query as input, allowing it to be used alongside large language models (LLMs) to generate parts based on the task. To investigate if the LLM can also generate the object part, we use an LLM (ChatGPT) to generate the object and part query automatically via few-shot prompting. Results are shown in Table 3. The prompt and all tasks are included in the Appendix. Given the task and the list of objects in the scene, the LLM is tasked with generating the correct object and object part pair (object, part). We used a

	SemAbs [77]	OWL-ViT [69]
Correct Object	80%	85%
Correct Part	35%	50%

Table 2: **Single View Comparisons:** Results are reported across 20 different prompts and 5 scenes. See the Appendix for a complete list of scenes and queries.

majority voting scheme to query the LLM. Given the task, the LLM provides seven candidates that we use to select the pair (object, part) that appears in a majority of the responses.

Integration with an LLM Planner LERF-TOGO can integrate as a module with an LLM planner to combine task-oriented grasps for robotic manipulation tasks. We define a set of robotic manipulation primitives (grasp, press, twist, pick&place, pour) and prompt the LLM to output the correct primitive for a given task. We use the same majority voting scheme in the previous section to select both the correct robotic primitive and the pair (object, part). Now, given a task (e.g. 'uncork the wine'), an LLM can specify the action to accomplish the task ('grasp') and the pair of object and object part (e.g. 'wine' and 'cork'). The prompt and all tasks are included in the Appendix.

5.1 Comparisons

ConceptFusion [70] generates a multimodal point cloud of a scene by fusing RGBD images and their extracted features together. We provide ConceptFusion with depth generated from the NeRF, which results in the same high-quality point cloud we use for grasping. To fairly represent the paper, we use the OpenCLIP ViT-H/14 model, which is several times larger than the ViT-B/16 model for LERF-TOGO. To query ConceptFusion, we provide it with the concatenated object and part prompt (i.e. *"mug handle"*) and rank grasps via the highest similarity. We report the object and part success without physical evaluation.

Semantic Abstraction [77] takes a single RGBD frame as input and a text query and outputs a relevancy heat map over the image. Since the method takes a single image, we provide the method with an input image observing all object parts for a fair comparison. We provide it with part queries 2 ways and take the best performance: 1) the concatenated object and part prompt (i.e. *"mug handle"*), and 2) the object and part as separate queries. A query is a success if the majority of the heatmap overlaps with the object part. In instances where activations are detected on other objects, it is considered successful if the highest activation is on the desired object part. Detailed results can be found in Table 2 and the Appendix.

OWL-ViT [69] is an open-vocab detector which takes in an RGB image and text prompts and outputs segmentation maps. We provide OWL-ViT a single input image that encompasses all object parts for a fair comparison. To obtain an object mask, we use the object prompt to establish an initial bounding box. This box serves as a region to identify the highest-scoring part within the region. In order to deem the part box as successful, we visually confirm that it aligns with the object part. Results can be found in Table 2 and the Appendix.

6 Results

Part-Oriented LERF-TOGO overall achieves a 69% success rate for physically grasping and lifting objects by the correct part. The selected grasp was located around the correct object part 82% of the time, with the remaining failures being grasp execution failures. For context, the highest confidence geometric grasp on an object mask only lies on the correct part 18% of the time, suggesting LERF-TOGO meaningfully biases the grasp distribution to the object part. Selected task-oriented queries are visualized in Fig. 2: the distribution of grasps drastically shifts based on the given part query, and can focus task-oriented grasps on multiple different regions per object based on the language prompt. LERF-TOGO shows strong language understanding performance for object selection (96%), able to

	Human-Provided Part	LLM-Provided Part
Correct Object	96%	96%
Correct Part	82%	71%

Table 3: **Task-Oriented Success:** Results using an LLM to choose the object part given a task specification. Results are reported across 49 different prompts and 12 scenes. A success measures if the grasp chosen by LERF-TOGO surrounds an object part which facilitates the task.

differentiate between very fine-grained language queries like color, appearance ("matte" vs "shiny"), or semantically similar categories (*"popsicle"* vs *"lolipop"*). It also can recognize long-tail object queries like *"ethernet dongle"*, *"cork"*, or *"martini glass"*, owing to its usage of CLIP zero-shot.

Task-Oriented Combining few-shot LLM prompting with LERF-TOGO identifies the correct primitive with 92% success and produces grasps on the correct object with 71% success across 49 tasks on 39 different objects. The LLM was able to correctly identify the object in the scene with the same success rate as the human, giving the correct object and part pairs for common tasks like "scrub the dishes" and "cut the steak". However, the LLM had a lower success rate (71%) compared to the human (82%) for object part selection. This is because CLIP, and by extension LERF-TOGO, can be sensitive to subtle variations in wording like "body" vs. "base" resulting in different LERF activations and thus grasps.

Comparisons LERF-TOGO out-performs ConceptFusion by 43% at task-oriented grasping because it can capture multi-scale semantics, while ConceptFusion is limited to one CLIP embedding per point. This makes hierarchical querying difficult, and is reflected by the fact that ConceptFusion performs similarly to LERF-TOGO at selecting the correct object, but suffers at selecting the right object part. Due to its lack of scale-conditioning, ConceptFusion frequently emphasizes large sections of the table due to the inclusion of both the objects and the table itself in the mask proposals (Fig. 6).

Semantic Abstraction achieves an overall object detection rate of 80% and part detection of 35%. The method tends to produce empty relevancy responses when queried for specific object parts, potentially owing to its averaging across multiple scales which drowns out smaller part features. When presented with the concatenated object and object part, the method highlights the entirety of the object, owing to CLIP's bag of words behavior, a characteristic addressed by LERF-TOGO's compositional queries. (Fig. D.2).

OWL-ViT achieves 85% accuracy for object localization, struggling on very long-tail objects that were not encountered within the detection datasets. This behavior is amplified for object part queries, where queries tend to be especially long-tail such as *"measuring tape"* and *"ethernet dongle"* (Fig. 12).

Failures The primary failure modes of LERF-TOGO are mistaking visually similar object parts for one another (eg teapot spout for a handle), missing subtle geometries like the small teacup handle or spray trigger, or confusing very close categories like steak and bread knives. We also observe prompt sensitivity with part queries: for example *"bottle neck"* more strongly localizes grasps than *"neck"*, and without more prompt tuning *"body"* sometimes fails to highlight the bases of bottles.

Object Extraction Ablation: Without 3D object masking and conditional querying, LERF-TOGO suffers with oblong objects, as shown in Fig. 5. We compare against querying LERF individually for the object and part, and multiplying their results together. This produces fragmented results which can ignore relevant parts of the object for part queries.

7 Limitations and Future Work

One limitation of LERF-TOGO is speed: the entire end-to-end the process takes a few minutes which can be impractical for time-sensitive applications. Future work on additional regularizations and optimizations to LERF training may reduce computation time. Another key limitation of LERF-TOGO is with groups of connected foreground objects, for example a bouquet of multiple flowers.

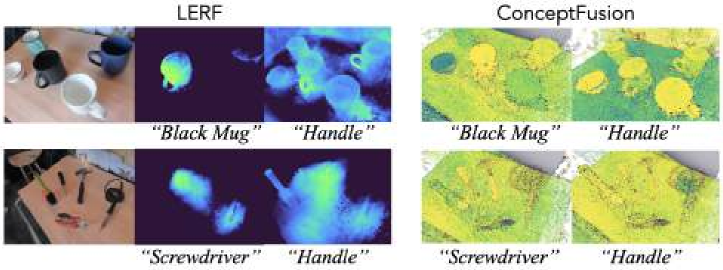


Figure 6: **Comparison with LERF and ConceptFusion**: ConceptFusion performs well on object-level queries, but struggles with sub-object part queries because of its lack of multi-scale semantics.

The DINO flood-fill will create a foreground group containing all flowers, after which isolating the stem of a specific individual type of flower (i.e *"daisy"* vs *"rose"*) is challenging. Supporting hierarchy within foreground groups is critical to enable such cases. If there are multiple objects that match the prompt, the system will arbitrarily choose only one of them. This is also true if the object query cannot disambiguate multiple instances of a similarly colored object (e.g., "mug" object query in the mugs scene). We also note that LERF-TOGO is not designed for referring/comparative expressions (e.g., "mug next to the plate", "biggest mug"). In addition, though we present a method for obtaining object part queries from input task descriptions via LLMs, in future work we will evaluate its performance on a diverse set of tasks.

8 Conclusion

This paper presents LERF-TOGO, a method for using vision-language models zero-shot with Language Embedded Radiance Fields to grasp objects and their parts via language. By improving the spatial grouping of LERF relevancy outputs, LERF-TOGO can support hierarchical part queries conditioned on the full object. Results indicate it performs strongly at language-guided grasping, with grasps landing on the correct object 96% of the time, and furthermore can direct grasps to the correct object parts 81% of the time. All code and datasets will be released after the submission process.

9 Acknowledgements

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab. The authors were supported in part by donations from Toyota Research Institute, Bosch, Google, Siemens, and Autodesk and by equipment grants from PhotoNeo, NVidia, and Intuitive Surgical. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Sponsors. We would like to thank our colleague Brent Yi for his work on Viser, the visualization tool of our experimental setup. We thank our colleagues who provided helpful feedback and suggestions, in particular Simeon Adebola and Julia Isaac.

References

[1] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik. Lerf: Language embedded radiance fields. *ICCV*, 2023.

[2] K. Kleeberger and M. F. Huber. Single shot 6d object pose estimation, 2020.

[3] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke. Using simulation and domain adaptation to improve efficiency of deep robotic grasping, 2017.

[4] S. Song, A. Zeng, J. Lee, and T. Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *Robotics and Automation Letters*, 2020.

[5] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world, 2017.

[6] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation, 2018.

[7] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018. doi:10.1177/0278364917710318.

[8] A. Mousavian, C. Eppner, and D. Fox. 6-dof graspnet: Variational grasp generation for object manipulation, 2019.

[9] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.

[10] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.

[11] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2015.

[12] A. Depierre, E. Dellandréa, and L. Chen. Jacquard: A large scale dataset for robotic grasp detection. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3511–3516. IEEE, 2018.

[13] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[14] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015.

[15] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020.

[16] G. Li, V. Jampani, D. Sun, and L. Sevilla-Lara. Locate: Localize and transfer object parts for weakly supervised affordance grounding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2023.

[17] A. Murali, W. Liu, K. Marino, S. Chernova, and A. Gupta. Same object, different grasps: Data and semantic knowledge for task-oriented grasping. In *Conference on Robot Learning*, 2020.

[18] M. Kokic, D. Kragic, and J. Bohg. Learning task-oriented grasping from human activity datasets. *IEEE Robotics and Automation Letters*, 5(2):3352–3359, 2020.

[19] D. Song, C. H. Ek, K. Huebner, and D. Kragic. Task-based robot grasp planning using probabilistic inference. *IEEE transactions on robotics*, 31(3):546–561, 2015.

[20] R. Detry, J. Papon, and L. Matthies. Task-oriented grasping with semantic and geometric scene understanding. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3266–3273. IEEE, 2017.

[21] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1374–1381. IEEE, 2015.

[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[23] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[24] F.-J. Chu, R. Xu, and P. A. Vela. Learning affordance segmentation for real-world robotic manipulation via synthetic images. *IEEE Robotics and Automation Letters*, 4(2):1140–1147, 2019. doi:10.1109/LRA.2019.2894439.

[25] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour. Part-based grasp planning for familiar objects. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 919–925, 2016. doi:10.1109/HUMANOIDS.2016.7803382.

[26] Y. Song, P. Sun, Y. Ren, Y. Zheng, and Y. Zhang. Learning 6-dof fine-grained grasp detection based on part affordance grounding. *arXiv preprint arXiv:2301.11564*, 2023.

[27] J. Song, A. Tanwani, J. Ichnowski, M. Danielczuk, K. Sanders, J. Chui, J. A. Ojea, and K. Goldberg. Robust task-based grasping as a service. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 22–28. IEEE, 2020.

[28] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *2009 IEEE 12th International Conference on Computer Vision*, pages 269–276. IEEE, 2009.

[29] Y. Chai, V. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for image classification. In *2011 International conference on computer vision*, pages 2579–2586. IEEE, 2011.

[30] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022.

[31] P. Sun, S. Chen, C. Zhu, F. Xiao, P. Luo, S. Xie, and Z. Yan. Going denser with open-vocabulary part segmentation. *arXiv preprint arXiv:2305.11173*, 2023.

[32] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[33] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

[34] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[35] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[36] L. Ma, X. Li, J. Liao, Q. Zhang, X. Wang, J. Wang, and P. V. Sander. Deblur-nerf: Neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12861–12870, 2022.

[37] X. Huang, Q. Zhang, Y. Feng, H. Li, X. Wang, and Q. Wang. Hdr-nerf: High dynamic range neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18398–18408, 2022.

[38] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20626–20636, 2023.

[39] J. Philip and V. Deschaintre. Radiance field gradient scaling for unbiased near-camera training. *arXiv preprint arXiv:2305.02756*, 2023.

[40] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, et al. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023.

[41] P. Wang, Y. Liu, Z. Chen, L. Liu, Z. Liu, T. Komura, C. Theobalt, and W. Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4150–4159, 2023.

[42] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023.

[43] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[44] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su. Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 333–350. Springer, 2022.

[45] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023.

[46] A. Yu, S. Fridovich-Keil, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021.

[47] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[48] Z. Li, Q. Wang, F. Cole, R. Tucker, and N. Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4284, 2023.

[49] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[50] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12786–12796, June 2022.

[51] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6229–6238, October 2021.

[52] A. Rosinol, J. J. Leonard, and L. Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields, 2022.

[53] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba. 3d neural scene representations for visuomotor control. In *Conference on Robot Learning*, pages 112–123. PMLR, 2022.

[54] D. Driess, I. Schubert, P. Florence, Y. Li, and M. Toussaint. Reinforcement learning with neural radiance fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[55] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg. Evo-neRF: Evolving neRF for sequential robot grasping of transparent objects. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=Bxr45keYrf.

[56] J. Ichnowski*, Y. Avigal*, J. Kerr, and K. Goldberg. Dex-NeRF: Using a neural radiance field to grasp transparent objects. In *Conference on Robot Learning (CoRL)*, 2020.

[57] A. Byravan, J. Humplik, L. Hasenclever, A. Brussee, F. Nori, T. Haarnoja, B. Moran, S. Bohez, F. Sadeghi, B. Vujatovic, and N. Heess. Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields, 2022.

[58] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021.

[59] Y. Siddiqui, L. Porzi, S. R. Buló, N. Müller, M. Nießner, A. Dai, and P. Kontschieder. Panoptic lifting for 3d scene understanding with neural fields, 2022.

[60] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. Guibas, A. Tagliasacchi, F. Dellaert, and T. Funkhouser. Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation. In *CVPR*, 2022.

[61] S. Kobayashi, E. Matsumoto, and V. Sitzmann. Decomposing nerf for editing via feature field distillation. In *NeurIPS*, volume 35, 2022. URL https://arxiv.org/pdf/2205.15585.pdf.

[62] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022.

[63] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

[64] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang. Open-vocabulary object detection using captions, 2021.

[65] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin. Scaling open-vocabulary image segmentation with image-level labels, 2022.

[66] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. Open-vocabulary object detection via vision and language knowledge distillation, 2022.

[67] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip, 2023.

[68] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision, 2022.

[69] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby. Simple open-vocabulary object detection with vision transformers, 2022.

[70] K. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, J. Tenenbaum, C. de Melo, M. Krishna, L. Paull, F. Shkurti, and A. Torralba. Conceptfusion: Open-set multimodal 3d mapping. *arXiv*, 2023.

[71] C. Huang, O. Mees, A. Zeng, and W. Burgard. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*, 2022.

[72] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. *arXiv preprint arXiv:2211.15654*, 2022.

[73] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021.

[74] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022.

[75] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.

[76] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler. Open-vocabulary queryable scene representations for real world planning. *arXiv preprint arXiv:2209.09874*, 2022.

[77] H. Ha and S. Song. Semantic abstraction: Open-world 3D scene understanding from 2D vision-language models. In *Proceedings of the 2022 Conference on Robot Learning*, 2022.

[78] H. Chefer, S. Gur, and L. Wolf. Generic attention-model explainability for interpreting bimodal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406, 2021.

[79] O. Mees and W. Burgard. Composing pick-and-place tasks by grounding language, 2021.

[80] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1307–1315, 2018.

[81] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[82] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[83] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, B. Zitkovich, F. Xia, C. Finn, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.

[84] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[85] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[86] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum. Compositional visual generation with composable diffusion models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 423–439. Springer, 2022.

[87] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[88] Guangcong, Z. Chen, C. C. Loy, and Z. Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. *Technical Report*, 2023.

[89] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs, 2021.

A LERF-TOGO

A.1 Implementation Details

We implement LERF-TOGO on top of the Nerfacto method from Nerfstudio [40]. For faster convergence and smoother optimization, we modify several parameters from the original LERF paper. We use a smaller hashgrid with 16 levels and a maximum resolution of 256, and find that using larger MLPs for the density, color, and transient output heads of NeRF results in faster convergence and better ability to handle specularities and robot shadows. We introduce weight decay of 1e-7 to the LERF network which smooths training. In addition, we compress the DINO embeddings into 128 dimensions and supervise on these vectors rather than the original DINO outputs, but we do not normalize the resulting vectors like Tschernezki et al. [62]. While constructing the CLIP embedding pyramid for LERF, we use crops ranging from 5% to 35% of image height with 6 pyramid levels, biasing the pyramid to smaller crops as LERF-TOGO is primarily interested in object and part queries.

B Robot Capture

Robot Capture Region Size The robot captures the scene along a hemispherical trajectory arcing $\pm 100°$ around the workspace horizontally (“1/2 hemisphere” in Fig. B.) When this horizontal sweep angle is reduced to a fraction of the range, the quality of the 3D object mask degrades, sometimes selecting the incorrect object altogether. LERF’s semantic field is supervised on features of the scene images, thus the quality is heavily correlated with the distribution of images viewing the object. This lowers the quality of the 3D DINO embeddings used for the mask generation.

LERF Training Steps In our experiments the LERF scene representation is trained to 2k steps. As shown in Fig. B, objects/object parts (e.g., “spray nozzle”, “bottle”) can be detected in steps as low as 1k, but more fine-grained or smaller parts (e.g., “handle”) may take longer (2-3k steps). This is consistent with what LERF reports: ”fine-grained features take more steps [to emerge]”.

3D Object Extraction Given the initial 3D point with the highest LERF activation, we create an object-centric point cloud by rendering six different views looking at the 3D coordinate. The views are $\pm 90°$ around the upwards vector through the 3D point. For DINO floodfill, the threshold DINO similarity is defined as first projecting the current DINO embedding onto the first PCA component of the top-down image, then taking the L2 norm of the difference between the current embedding and the DINO embedding at the initial 3D point.

NeRF Regularization NeRF encounters difficulties in reconstructing texture-less planar surfaces, especially in the presence of specularities. This limitation is prominent in our table-top scenes, where the glossy surface and metallic objects can result in depth renderings with jagged missing regions. These missing regions can cause LERF renderings to spuriously activate and degrade the performance of grasp networks, so we apply depth regularization to mitigate this issue. We adopt the local depth ranking loss proposed in SparseNeRF [88] and use ZoeDepth [87] as the underlying depth model. We found this performs better than smoothness priors [89, 46] because it retains more fine-grained geometry. Additionally, we use the gradient scaling approach from Philip and Deschaintre [39], which significantly reduces the number of near-camera floaters and enables more robust grasping directly from point clouds rendered from the NeRF.

Poses obtained from cameras in motion are slightly inaccurate, which we found could result in oversmoothed geometry with depth regularization. To overcome this, we optimize the NeRF for the first 500 steps without any regularization to allow the camera poses to settle, then anneal the depth regularization loss term from 0 to 100% over the next 1500 steps. Interestingly, we find staged training not only preserves thin features better but also speeds LERF optimization. We hypothesize this is because supervising the language field on un-converged density in free space results in a poor network initialization, while beginning LERF optimization after geometry has been largely removed from free space allows a smoother learning signal.

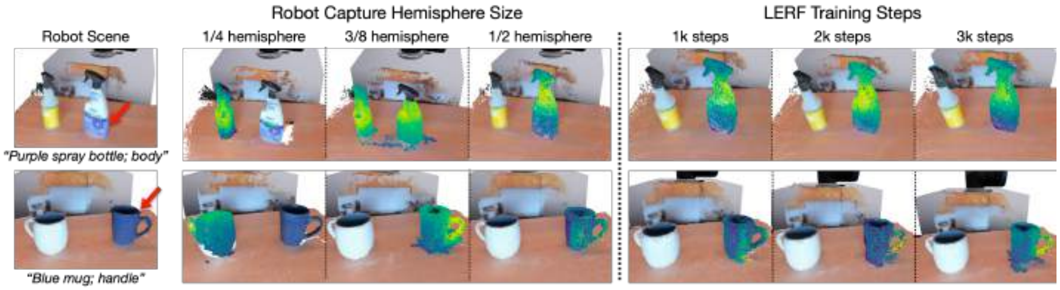


Figure 7: Left: Decreasing the robot scan region degrades the quality of 3D relevancy map generated by LERF-TOGO. Right: LERF-TOGO relevancy map converges earlier for larger objects and parts.

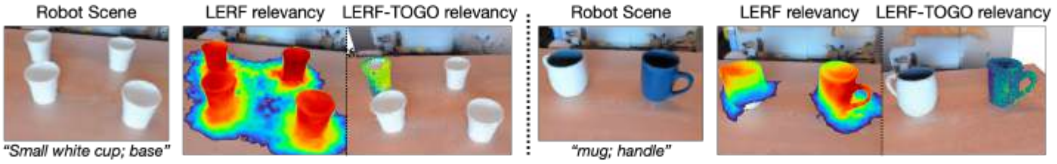


Figure 8: In the case of identical objects or ambiguous object queries, LERF-TOGO picks a single object in the scene but does not propose grasps for all the objects in the category.

C Grasping

Point Cloud Extraction To extract a scene-wide point cloud for grasping, we use the method in Nerfstudio [40], which deprojects randomly sampled rays' depth from the train camera poses, then filters with outlier rejection. We then crop the point cloud to the workspace of the robot. For object centric point clouds, we deproject depth from views radially surrounding the object of interest.

Motion Planning A grasp is considered feasible if the robot can perform a collision-free trajectory with the following poses: the pre-grasp, grasp, and post-grasp configurations. The pre-grasp pose is positioned 5cm along the z-axis of the robot end effector, which allows the gripper to approach the target grasp pose with minimal additional motion. The post-grasp pose is located 10cm above the grasp pose, along the z-axis of the world frame. The UR5's IK solver calculates the set of viable joint configurations at these poses, and we calculate the trajectory as a linear interpolation between them. We additionally allow for a 180 degree rotation at the last wrist joint, as parallel-jaw grasps are rotationally symmetric. This facilitates the motion planning process, as the robot's camera mount is prone to colliding with the robot arm.

D Experiments

D.1 Setup Details

We use a UR5 arm with a Logitech BRIO webcam at 1600x896 resolution, with all camera settings frozen before each capture prevent color discrepancies among images. The camera mount points orthogonally to the gripper axis, to maximize the reachable workspace of the camera while pointing towards the workspace center. During robot capture, pre-computation of DINO, CLIP, and ZoeDepth is parallelized across 3 NVIDIA 4090 GPUs to achieve real-time performance, and all subsequent operations are carried out on a single 4090. Capturing a scene takes 30 seconds, training the LERF to 2k iterations takes 78 seconds, and finally querying LERF-TOGO takes 10 seconds.

D.2 LLM Interface

We provide the full prompt to the LLM below. For any given task and scene the OBJECT_LIST is replaced with a list of objects within the scene and TASK is replaced with the desired task:

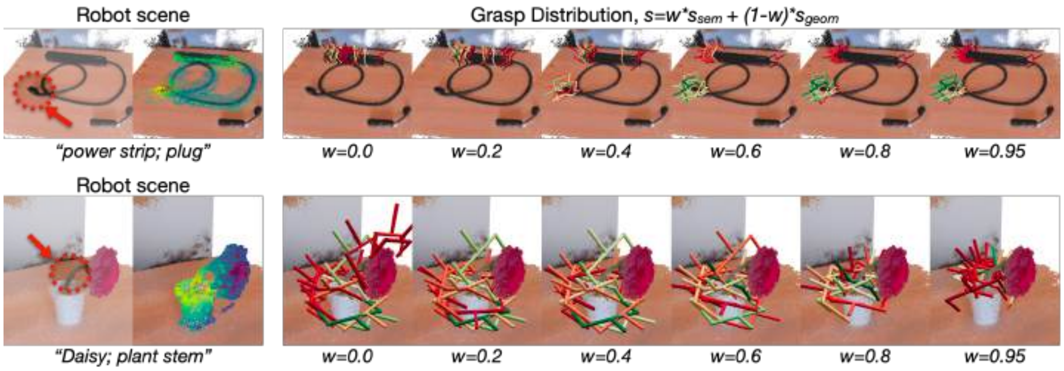


Figure 9: **Grasp score weighting**: Varying weight between geometric grasp score and semantic grasp score shifts the grasp distribution. A high semantic grasp weight ($w = 0.95$) is required, since geometric grasps may be biased away from small and fine-grained object parts of interest. Both geometric and semantic scores are in the range [0, 1].

```
  Answer the question as if you are a robot with a parallel jaw gripper that has
  access to only the objects in the object list. Follow the exact format.
  First line should describe what basic action is needed to do the task from
  the following set of actions: press, grasp, twist, pick & place.
  The second line should only be an object from the object list followed by 1 object
  part that the robot would touch to do this task. VERY IMPORTANT: If the basic
  action is pick & place, only then have a third line with ’Place: ’
  to specify the object to place on. \
Object list: [’pot’, ’knife’, ’spoon’, ’black pan’] \
Q: How can I safely pick up a pan? \
Basic Action: grasp \
Sequence: 1. black pan 2. handle \
\
Object list: [’mechanical keyboard’, ’knife’, ’TV’, ’camera’] \
Q: How can I safely hit the spacebar on a keyboard? \
Basic action: press \
Sequence: 1. mechanical keyboard 2. spacebar\
 \
Object list: [’green mug’, ’blue spoon’, ’fork’, ’knife’] \
Q: How can I cut a block of cheese? \
Basic action: grasp \
Sequence: 1. knife 2. handle \
 \
Object list: [’salt shaker’, ’knife’, ’fork’, ’white pan’] \
Q: How can I safely lift a salt shaker? \
Basic Action: grasp \
Sequence: 1. salt shaker 2. base \
\
Object list: [’red cup’, ’blue cup’, ’mug’, ’bowl’] \
Q: How do I stack the red cup on the blue cup? \
Basic action: pick & place \
Sequence: 1. red cup 2. rim \
Place: blue cup \
 \
Object list: [’door knob’, ’black mug’, ’green dish brush’, ’shiny knife’] \
Q: How do I open a door knob? \
```

Scene	Object Query ; Part Query
Kitchen	(black matte spoon, handle), (shiny black spoon, handle), (teapot, handle), (dish scrub brush, handle), (dust brush, handle)
Flowers	(daisy, plant stem), (rose, plant stem)
Mugs	(blue mug, handle), (pink teacup, handle), (turqouise mug, handle), (white mug, handle), (black mug, handle)
Tools	(Measuring tape, base), (screwdriver, handle), (wire cutters, handle) (soldering iron, handle), (hammer, handle)
Knives	(bread knife, handle), (steak knife, handle), (box cutter, handle)
Martinis	(red martini glass, stem), (grey martini glass, stem)
Fragile	(camera, strap), (pink sunglasses, earhooks) (blue sunglasses, earhooks), (lightbulb, screw)
Cords	(power strip, plug), (power strip, base), (ethernet dongle, usb) (ethernet dongle, ethernet)
Messy	(ice cream, cone), (green lollipop, stick), (blue lollipop, stick)
Pasta	(wine, cork), (wine, bottle neck), (saucepan, lid knob)
Cleaning	(saucepan, handle), (corkscrew, handle) (clorox, wet towel), (clorox, lid), (clorox, body) (tissue box, box), (tissue box, tissue)
Bottles	(meyer's cleaning spray, spray trigger), (meyer's cleaning spray, bottle neck) (meyer's cleaning spray, body) (purple cleaning spray, body) (purple cleaning spray, spray trigger), (purple cleaning spray, bottle neck)
Electronics	(e-stop, red button), (computer mouse, scroll wheel), (controller, buttons), (spacemouse, button), (white controller; button) (controller, joystick), (e-stop, red button)

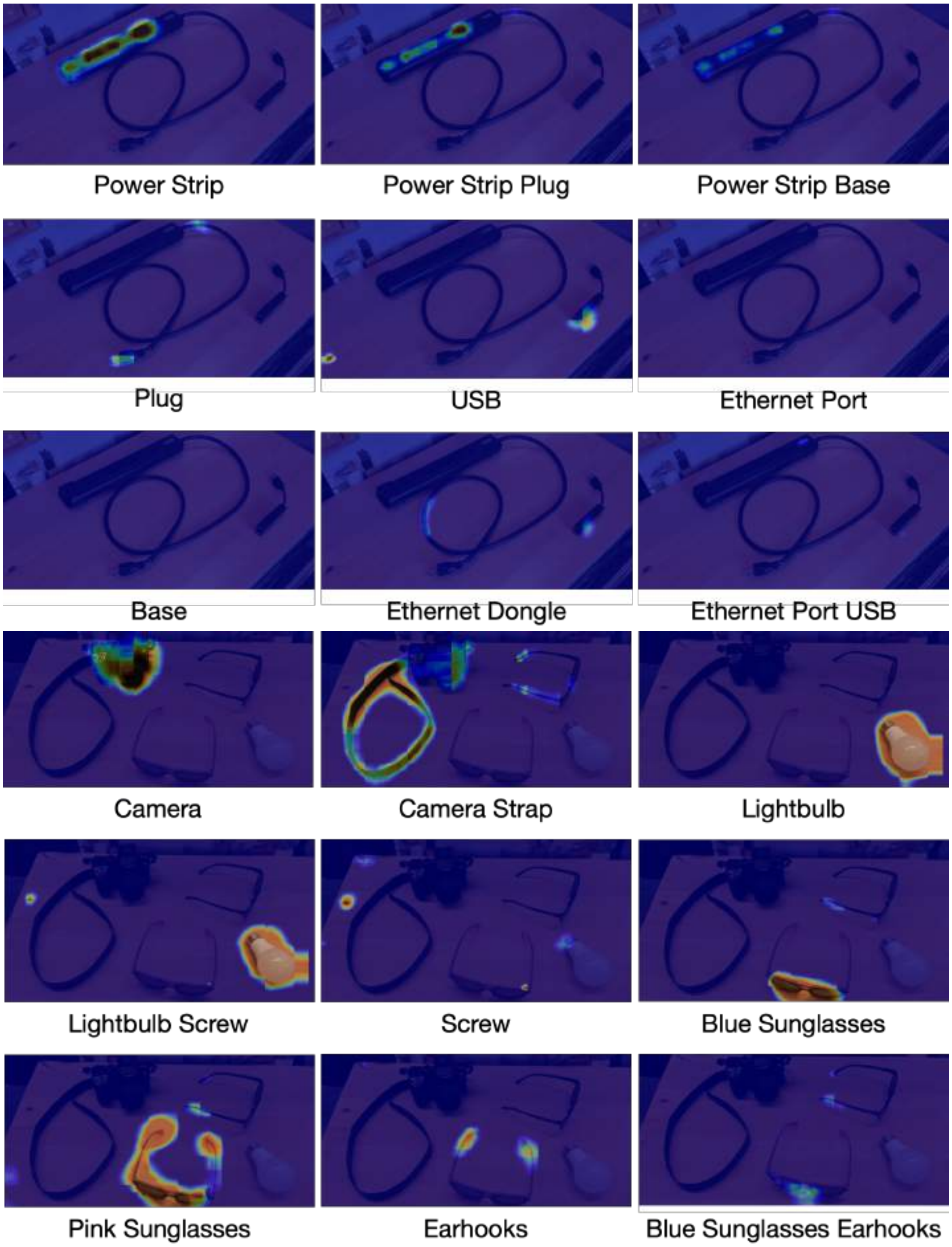
Table 4: **Complete list of object and part queries**

```
Basic action: twist \
Sequence: 1. door knob 2. rim \
  \
Object list: ['dryer', 'washing machine', 'sunglasses'] \
Q: How do I turn on the washing machine? \
Basic action: twist \
Sequence: 1. washing machine 2. dial \
  \
Object list: ['paper towel roll', 'mug', 'teacup', 'headphones', 'pen'] \
Q: How do I grab a paper towel? \
Basic action: grasp \
Sequence: 1. paper towel roll 2. paper towel \
  \
Object list: ['magnifying glass', 'blue spoon', 'fork', 'knife'] \
Q: How do I pick up a magnifying glass? \
Basic action: grasp \
Sequence: 1. magnifying glass 2. handle \
  \
Object list: ['teddy bear', 'toy block', 'mouse', 'saucepan', 'hammer'] \
Q: How do I grab a teddy bear? \
Basic action: grasp \
Sequence: 1. teddy bear 2. head \
```

Scene	Tasks
Kitchen	’grab the matte spoon’, ’pass the shiny spoon’, ’grab the teapot’, ’scrub the dishes’, ’dust the book’
Flowers	’give the daisy’, ’give the rose’
Mugs	’grab the blue mug’, ’grab the pink teacup’, ’grab the turquoise mug’ ’grab the white mug’, ’grab the black mug’
Tools	’grab the measuring tape’, ’give the screwdriver’, ’cut the wire’ ’hold the soldering iron’, ’swing the hammer’
Knives	’cut the bread’, ’cut the steak’, ’cut the box’
Martinis	’deliver the grey martini glass’, ’deliver the red martini glass’
Fragile	’take a picture’, ’wear the pink sunglasses’ ’wear the blue sunglasses’, ’move the lightbulb’
Cords	’grab the power strip’, ’plug in the power strip’, ’plug the ethernet into the ethernet dongle’, ’plug in the usb into the ethernet port’
Messy	’eat the ice cream’, ’eat the green lollipop’, ’eat the blue lollipop’
Pasta	’uncork the wine’, ’grab the saucepan’, ’open the saucepan’, ’grab the corkscrew’, ’lift the wine’
Cleaning	’grab a wet towel from the clorox’, ’close the box of clorox’ ’hold the clorox’, ’get a tissue’, ’hold the tissue box’
Bottles	’grab the meyers cleaning spray’, ’open the meyers cleaning spray’, ’spray the meyers cleaning spray’, ’grab the purple cleaning spray’, ’open the purple cleaning spray’, ’spray the purple cleaning spray’
Electronics	’hold down the e-stop’, ’hit the scroll wheel’, ’click the controller’, ’click on that website with the spacemouse’, ’click the white controller’, ’turn video game character’, ’release the e-stop’

Table 5: **Complete list of tasks for each scene**

```
\
Object list: [’green mug’, ’blue spoon’, ’fork’, ’knife’] \
Q: How do I put the mug in the cabinet? \
Basic action: pick & place \
Sequence: 1. green mug 2. handle \
Place: cabinet \
\
Object list: {OBJECT_LIST} \
Q: How can I safely {TASK}? \
Basic action: "
```



Power Strip

Power Strip Plug

Power Strip Base

Plug

USB

Ethernet Port

Base

Ethernet Dongle

Ethernet Port USB

Camera

Camera Strap

Lightbulb

Lightbulb Screw

Screw

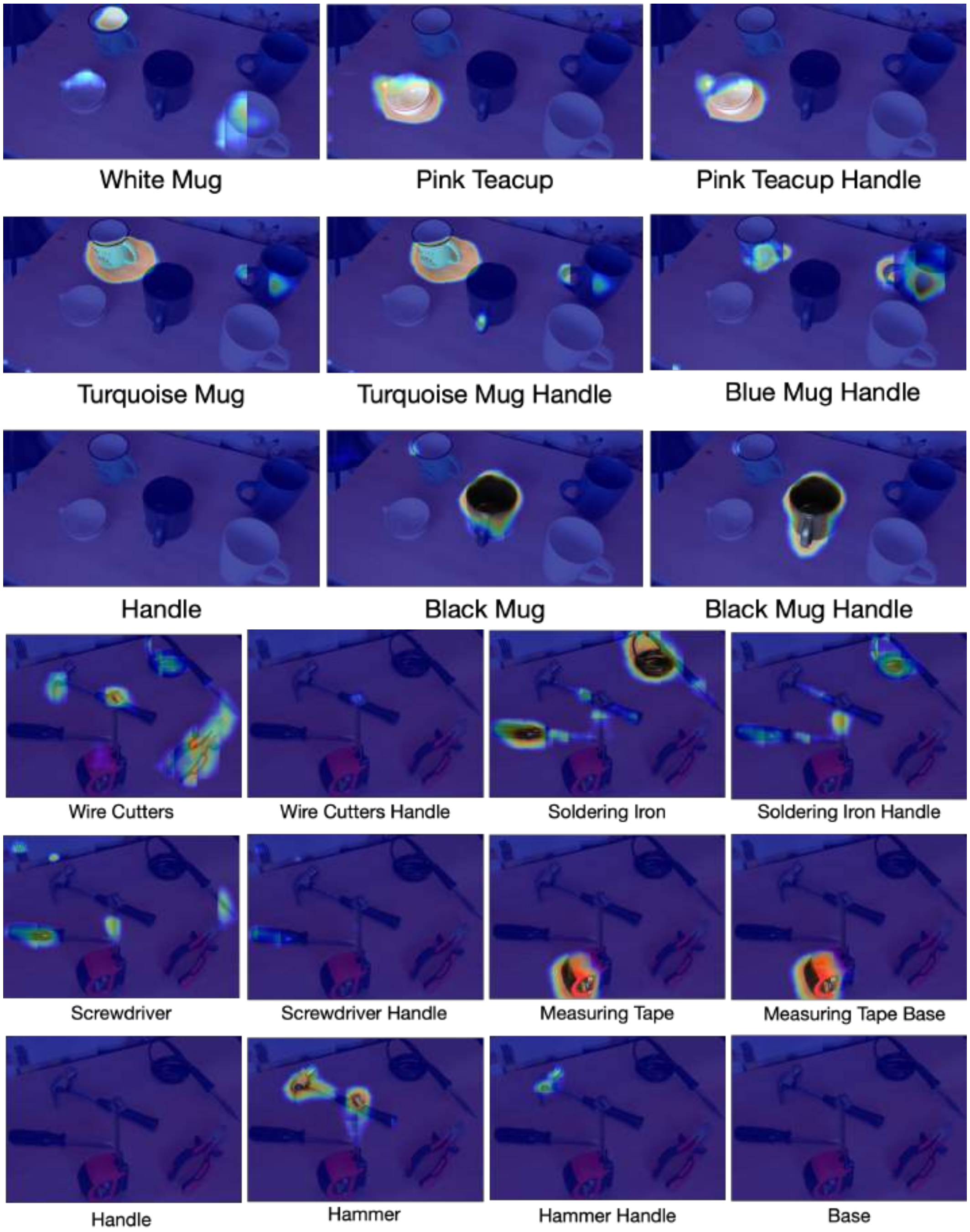
Blue Sunglasses

Pink Sunglasses

Earhooks

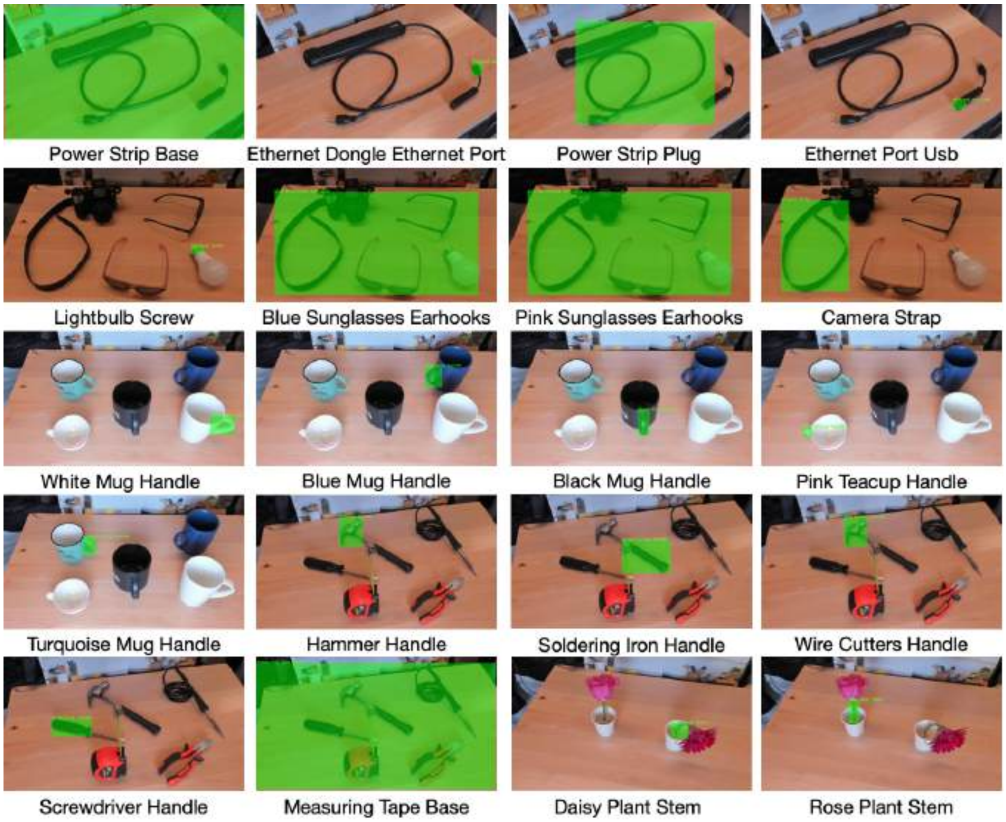
Blue Sunglasses Earhooks

Figure 10: Semantic Abstraction results for object and object part localization (cont.)



White Mug
Pink Teacup
Pink Teacup Handle
Turquoise Mug
Turquoise Mug Handle
Blue Mug Handle
Handle
Black Mug
Black Mug Handle
Wire Cutters
Wire Cutters Handle
Soldering Iron
Soldering Iron Handle
Screwdriver
Screwdriver Handle
Measuring Tape
Measuring Tape Base
Handle
Hammer
Hammer Handle
Base

Figure 11: Semantic Abstraction results for object and object part localization



Power Strip Base

Ethernet Dongle Ethernet Port

Power Strip Plug

Ethernet Port Usb

Lightbulb Screw

Blue Sunglasses Earhooks

Pink Sunglasses Earhooks

Camera Strap

White Mug Handle

Blue Mug Handle

Black Mug Handle

Pink Teacup Handle

Turquoise Mug Handle

Hammer Handle

Soldering Iron Handle

Wire Cutters Handle

Screwdriver Handle

Measuring Tape Base

Daisy Plant Stem

Rose Plant Stem

Figure 12: OWL-ViT results for object and object part localization