Slot-Level Robotic Placement via Visual Imitation from Single Human Video

Anonymous submission

Paper ID

Abstract

The majority of modern robot learning methods focus on learning a set of pre-defined tasks with limited or no generalization to new tasks. Extending the robot skillset to novel tasks involves gathering an extensive amount of training data for additional tasks. In this paper, we address the problem of teaching new tasks to robots using human demonstration videos for repetitive tasks (e.g., packing). This task requires understanding the human video to identify which object is being manipulated (the pick object) and where it is being placed (the placement slot). In addition, it needs to re-identify the pick object and the placement slots during inference along with the relative poses to enable robot execution of the task. To tackle this, we propose SLeRP, a modular system that leverages several advanced visual foundation models and a novel slot-level placement detector Slot-Net, eliminating the need for expensive video demonstrations for training. We evaluate our system using a new benchmark of real-world videos. The evaluation results show that SLeRP outperforms several baselines and can be deployed on a real robot.

1. Introduction

Humans demonstrate exceptional skill in performing fine-grained manipulation tasks with high precision in their daily lives. From arranging eggs in an egg carton to sorting utensils in an organizer, we excel at tasks that require identifying and reasoning about which objects to pick up and how to place them into confined slots. Cognitive and motor development theories suggest that we develop such skills at a young age, based on early experiences like playing with shape sorter toys [49]. However, current robotic and automated systems are not yet as adept as humans at perceiving and performing these fine-grained manipulation tasks.

Slot-level manipulation is crucial in various industrial, logistics, and domestic contexts. For example, in industrial settings, machine tending [57] requires placing components precisely into machine slots for assembly or processing. In logistics, sorting and packaging tasks, such as organizing parcels in a warehouse or placing products into shipping containers, demand efficient and precise placement to optimize space and minimize damage. In domestic environments, future home assistant robots will need to perform slot-level manipulation tasks such as organizing items in cabinets, placing dishes in a dishwasher, and even preparing meals by accurately arranging ingredients in a pan.

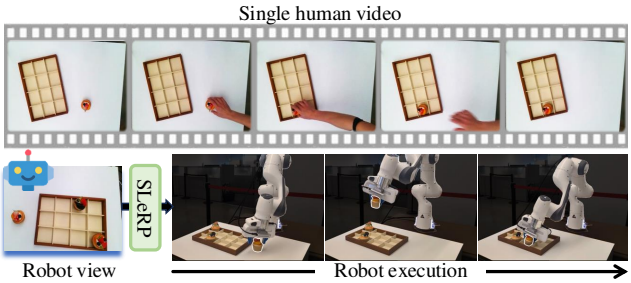


Figure 1. We introduce the novel problem of imitating slot-level robotic placement from a single human video. Given a human demonstration video showing an object being placed in a slot, and a new robot-view image captured by the robot wrist camera (may feature varied camera and object poses, changed scenes), SLeRP is able to find the corresponding object and similar slots in the robot view, and provide the 6-DoF transformation matrix for each detected slot to guide the robot in placing the object accurately.

The task of programming robots to perform slot-level placement remains arduous. Traditional methods [20, 67] often require manual programming with domain expertise and assume that the object models and slot locations are known beforehand. Learning-based approaches [9, 30] show promise in alleviating the burden of programming; however, collecting robot data through tele-operation remains tedious and inefficient and can be particularly brittle for high-precision tasks due to embodiment gaps. Learning from human demonstration videos has recently emerged as a promising approach due to its ease, speed of collection, and potential to capture slot-level details. However, previous research [3, 6, 26, 76, 86] has generally been limited to coarser object-level tasks and often requires large amounts of training data to learn how to parse human demonstrations and translate them into robot policies.

[*] Work done during internship at Nvidia. [†] Primary mentor.

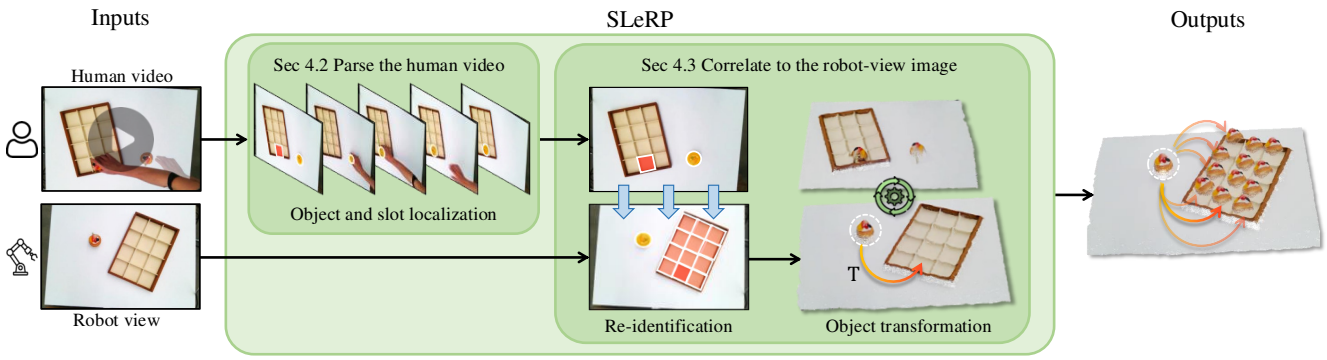


Figure 2. **Method Overview.** The system begins by analyzing the input human video, tracking the object (highlighted in yellow) throughout the sequence and identifying the placement slot (highlighted in red). Next, we re-identify the object and the slot in the robot's view by correlating the human-view and robot-view images. Using depth images, we reconstruct the observations in 3D and compute a single 6-DoF object transformation T in the robot's view, enabling the robot to transfer the object into the slot. If more than one slot is present, we detect all applicable slots and compute one 6-DoF object transformation for each slot. Finally, such 6-DoF object transformations are sent to the downstream robot planning and control pipeline for real robot pick-and-place execution.

In this paper, we study the novel problem of recognizing slot-level object placement from a *single* human video, and estimating 6-DoF transformations for robot imitation. As shown in Fig. 1, the task takes two visual inputs: (1) a single human RGB-D video in which a person demonstrates picking up an object (*e.g.*, a muffin) and precisely placing it into a slot within a placement object (*e.g.*, a tray), and (2) a single RGB-D image captured from the robot's wrist camera, representing the new setup for the robot to operate in with possible varying camera and object poses compared to the human video. The outputs aim to detect the object and all empty slots in the robot's view similar to the placement slot in the human video, as well as compute the 6-DoF object transformations necessary for the robot to transfer the object from its initial position to each of the slots.

We propose a novel modular approach called SLeRP (*i.e.*, **S**lot-**Le**vel **R**obotic **P**lacement), to tackle the problem. As shown in Fig. 2, SLeRP starts by analyzing the input human demonstration video, tracking the manipulated object across the video frames and identifying the placement slot. Next, within the robot's view, SLeRP re-identifies both the object and the slot by correlating the human-view images with the robot-view images. By lifting the observations in 3D using the depth sensing and camera parameters, SLeRP calculates a 6-DoF transformation matrix for the robot to transfer the object from its initial location to the desired slot in the robot's view. If multiple slots are present, SLeRP detects all slots that are similar to the one in the human video and computes the object transformations for all of them. Finally, the computed 6-DoF object transformations are sent to the downstream robot planning and control pipeline for robot pick-and-place execution.

A key component of SLeRP is the detection of placement slots. Currently, no existing method is specifically designed for this task, and simple image differencing or change detection [18] does not effectively solve the problem. Therefore, we propose a new slot-level placement detector, Slot-Net, that takes two image frames from a human demonstration video—one before and one after placement—and outputs a 2D mask outlining the placement slot on the images. Unlike common vision tasks, collecting a sizable training dataset for slot-level placement detection is challenging. To address this challenge, we introduce a generative AI-based data creation pipeline that expands the training set by bootstrapping from a small set of images.

For evaluation, we introduce a new dataset comprising 288 real-world videos targeted at studying this novel problem. We compare our method against several baseline approaches, including ORION [86], a state-of-the-art method for object-level pick-and-place from a single human video; CLIPort [61], an end-to-end imitation-learning-based language-conditioned policy for tabletop tasks; adapted versions of both for the novel slot-level task; and a custom baseline leveraging cutting-edge vision-language models like GPT-4o [25]. Our results demonstrate that SLeRP outperforms baselines in accurately predicting placement slots and computing 6-DoF transformations across diverse real-world tasks. Our ablation studies further validate several key components and design choices in our system. Finally, we conduct real-robot experiments that successfully apply the system in real world scenarios.

In summary, the core contributions of this paper are:

- Studying the novel task of slot-level object placement by learning from a single human demonstration video;
- Designing the modular approach SLeRP and the slot-level placement detector Slot-Net to tackle this problem;
- Introducing a new benchmark and several baseline methods to systematically evaluate system performance;
- Demonstrating that SLeRP achieves strong performance in real-world and real-robot evaluations.

2. Related Work

Object Placement in Robotics. Identifying where and how to place an object after picking it up is a crucial step in robotic pick-and-place tasks [37]. Early works [5, 22, 23] analytically search for flat features on the object and the placement surface. Modern learning-based methods estimate placement locations and poses using learned features, focusing mainly on flat surfaces [46], such as tabletop [13, 43, 82] and furniture shelves [44]. Researchers have also explored tabletop object placement under spatial and semantic constraints given other objects [35, 42, 52].

In the more challenging case of placing an object on another non-flat object, prior work [16, 28, 50, 59, 62, 63, 66, 73, 86] has explored tasks like putting one spoon in a cup or hanging a mug on a rack. Our work extends these studies by focusing on placing objects into all empty, fine-grained, tight-fitting slots (*e.g.*, all egg slots in a carton), a task that demands greater precision in recognition and prediction, as well as handling multiple placement locations. Additionally, unlike previous work [24, 61, 75, 83, 84] that addresses a 2D planar setting and requires task-specific training from a few robot demonstrations, our approach tackles this problem in 3D, learning from a single human demonstration to enable one-shot generalization to novel tasks.

Imitation Learning from Human Videos. Human videos serve as a natural, information-rich, and easily accessible source of data for learning robotic manipulation. Previous work has explored diverse methods to extract, represent, and apply knowledge from human videos to support robot manipulation learning. These approaches include pre-training latent visual representations [15, 39, 45, 58], inferring action trajectories or plans [3, 32, 47, 71, 79], learning value or reward functions [10, 38], reconstructing human hand or hand-object interaction [41, 51, 54, 60, 64], parsing interaction goals and affordance [4, 29, 36, 78], learning point tracks for human-to-robot transfer [7, 74, 76, 77, 81], etc. While the primary goals of these works are typically learning robot trajectories or manipulation policies, our work explores a novel perspective by recognizing fine-grained placement slots as visual imitation targets.

Additionally, we tackle robot imitation learning from a single human video. Previous work has investigated one-shot [14, 26, 27, 40, 80] and even zero-shot learning from human videos [6]; however, these approaches often require extensive human video datasets, sometimes paired with robot videos, to span multiple tasks during training. In contrast, our approach leverages existing visual foundation models, eliminating the need for large-scale training videos. A notably similar work ORION [86] relies on text to recognize task-relevant objects and primarily focuses on object-level pick-and-place. In contrast, our method exclusively extracts information from a single human video to perform more fine-grained slot-level placement tasks.

3. Problem Formulation

We formulate the novel problem of recognizing slot-level object placement from a single human video, and estimating 6-DoF transformations for downstream robot imitation.

Inputs. The task takes the following inputs:

- a single RGB-D human demonstration video with n frames, denoted as $\mathcal{H} = \{\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_n\}$, recording a person picking up an object O from the scene and placing it in a slot S within a placement object;
- a single RGB-D robot-view image $\mathbf{R}$ that captures the robot's observation, often taken from the robot wrist camera and possibly with different camera and object poses, or scene layouts.

Outputs. The task outputs, in the robot's view, are:

- an object mask $\mathbf{M}^O_{\mathbf{R}}$ over the robot image $\mathbf{R}$ that segments the object O to pick;
- a list of slot masks $\{\mathbf{M}^{S_0}_R, \mathbf{M}^{S_1}_R, \cdots, \mathbf{M}^{S_k}_R\}$ over the robot image $\mathbf{R}$ that marks all empty slots on the placement object similar to the demonstrated placement slot in the human video $\mathcal{H}$;
- a list of 3D 6-Degree-of-Freedom (DoF) object transformation matrices $\{\mathbf{T}_0, \mathbf{T}_1, \cdots, \mathbf{T}_k \mid \mathbf{T}_i \in SE(3)\}$ in the robot's coordinate frame, for the robot to transfer the object O from its initial position to all the detected slots.

Passing the detected object and slot masks, as well as the calculated 6-DoF object transformatrion matrices, downstream robot pick-and-place pipeline is able to execute slot-level object placement as shown in Fig. 1.

4. Method

In this section, we present the technical designs of SLeRP. We begin with an overview (Sec. 4.1) and then dive into more details in parsing the input human video (Sec. 4.2) and correlating to the robot's view image (Sec. 4.3).

4.1. System Overview

Taking as inputs a human demonstration video $\mathcal{H}$ and a robot-view image $\mathbf{R}$, our method SLeRP (Fig. 2) starts with parsing the input human video (Sec. 4.2) by tracking the object O throughout the video frames and identifying the placement slot S. After this process, we obtain an object mask $\mathbf{M}^O_{\mathbf{H}_1}$ and a slot mask $\mathbf{M}^S_{\mathbf{H}_1}$ over the first frame of the human video $\mathbf{H}_1$. Next, by leveraging this information, SLeRP correlates the human-view and robot-view images (Sec. 4.3), and re-identify the object mask $\mathbf{M}^O_{\mathbf{R}}$ and the slot mask $\mathbf{M}^{S_0}_{\mathbf{R}}$ in the robot-view image $\mathbf{R}$, as observed in the first human frame. If multiple similar slots are present, the system detects other empty slots $\{\mathbf{M}^{S_1}_R, \cdots, \mathbf{M}^{S_k}_R\}$ as well. Then, the system lifts the human and robot observations in 3D using the depth sensing and camera intrinsics, and computes a single 6-DoF object transformation matrix $\mathbf{T}_i \in SE(3)$ for each detected slot $\mathbf{M}^{S_i}_R$.

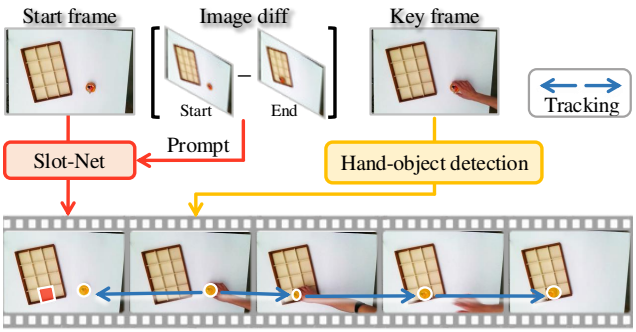


Figure 3. **Parse Human Video.** Given the input human video (bottom), we run state-of-the-art hand-object detector (yellow) and tracker (blue) to obtain the pick object mask (yellow) and train a novel network Slot-Net (red) to identify the slot mask (red).

4.2. Parsing the Human Demonstration Video

The input human video precisely demonstrates what is the pick object O and where is the placement slot S. As shown in Fig. 3, our method utilizes powerful hand-object detection and tracking systems to identify the object mask $\mathbf{M}^O_{\mathbf{H}_1}$ and proposes a novel network Slot-Net for estimating the slot mask $\mathbf{M}^S_{\mathbf{H}_1}$ over the first frame of the human video.

Object detection and tracking. We use a hand-object detector [11] to detect frame-wise hands and in-contact objects, enabling us to locate the pick object O in the human video. As the detector operates on a per-frame basis, there may be temporally inconsistent predictions. To refine the detection results, we apply MASA's matching algorithm [34] to generate smooth trajectories for the hand and pick object across the hand-object contact frames. We then identify a confident key frame, when the hand and object first interacts, and use SAM2 [55] to track through the video, producing per-frame object segmentation $\mathbf{M}^O_{\mathbf{H}_i}$.

Placement slot detection (Slot-Net). Since no prior work has studied the problem of detecting the placement slot given a human pick-and-place video, we propose our own novel network Slot-Net for this purpose. We leverage the SAM architecture [31] as the backbone given its powerful capability in segmentation. Slot-Net takes the starting frame $\mathbf{H}_1$ of the pick-place video as the input, together with the absolute image difference in gray-scale between the starting and end frame $|\mathbf{H}_1 - \mathbf{H}_n|$ as the visual prompt, and is tasked to output a slot segment in the starting human frame image $\mathbf{M}^S_{\mathbf{H}_1}$. We leverage SAM's large-scale pretraining by preserving most of its designs (*e.g.*, the image encoder, the mask decoder) and we use the same image encoder as the prompt embedder to process image difference prompt. Since we find that the SAM pretraining does not directly work on such customized new task, finetuning over such slot-level placement data is necessary.

Slot-Net data generation. Training our SAM-based SLeRP requires a lot of data, yet collecting fine-grained slot-level placement data in the real world is expensive.

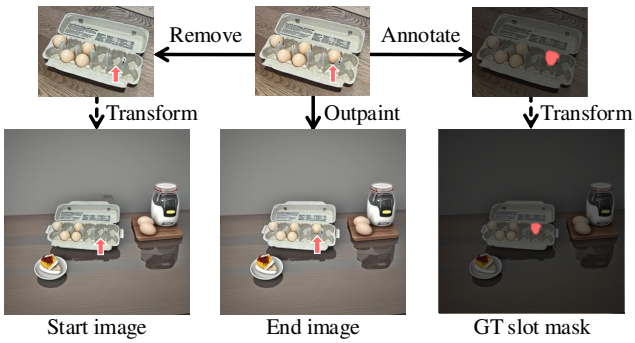


Figure 4. **Slot-Net Data Generation.** Given an object-centric image (top middle), we inpaint to remove an object and reveal its slot (top left) and manually annotate the slot mask (top right). We then outpaint these images with a scene background (bottom) to create a starting and end image pair with a ground-truth slot mask.

However, recent generative models have demonstrated great capabilities in generating realistic images [53], excelling at tasks such as object removal and image outpainting. We therefore propose a semi-automatic synthetic data generation pipeline (Fig. 4). Given a collected object-centric image of a placement object with many slots, we utilize a state-of-the-art object removal model (SDXL [70] and Cleanup.pictures [12]) to remove one pick object from one slot and manually annotate the slot mask for the removed object using TORAS [65]. Then, we employ a powerful image outpainting generative model (Hugging Face Outpainting script [1]) to expand the image canvas, generating 100 images in diverse backgrounds, prompted with Llama [69] generated text prompts, for each object-centric image.

In this manner, we obtain a large number of annotated starting and end image pairs to train SLeRP. We crowdsourced and collected 2,138 object-centric images of items with slots, spanning 67 object categories, by capturing them in everyday environments. We applied 100 augmentations for each slot on the object-centric image, resulting in 156K images for training, with the rest left for testing and validation. See supplementary for more details.

4.3. Correlating to the Robot-view Image

After we obtain the pick object mask $\mathbf{M}^O_{\mathbf{H}_1}$ and the slot mask $\mathbf{M}^S_{\mathbf{H}_1}$ from the human video, the next step is to correlate this information to the robot's view $\mathbf{R}$. As shown in Fig. 5, SLeRP first re-identifies the object mask $\mathbf{M}^O_{\mathbf{R}}$ and a list of empty slot masks $\{\mathbf{M}^{S_0}_{\mathbf{R}}, \mathbf{M}^{S_1}_{\mathbf{R}}, \cdots, \mathbf{M}^{S_k}_{\mathbf{R}}\}$ similar to the human demonstrated placement slot. Then, 2D keypoint matching and 3D lifting enable the calculation of a single 6-DoF object transformation matrix $\mathbf{T}_i \in SE(3)$ for each detected slot $\mathbf{M}^{S_i}_{R}$ for downstream robotic pick-and-place.

Object and slot re-identification. Taking as input a short video with only two frames $\{\mathbf{H}_1, \mathbf{R}\}$, SAM2 [55] is employed to output the object mask $\mathbf{M}^O_{\mathbf{R}}$ and one best-matched slot mask $\mathbf{M}^{S_0}_{\mathbf{R}}$ over the robot image $\mathbf{R}$, given the detected

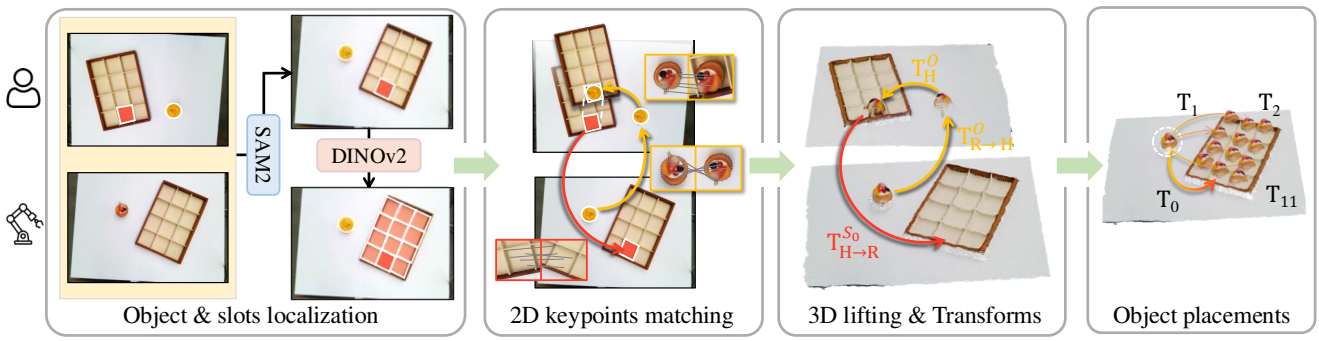


Figure 5. **Correlate with robot view.** Given the object and slot mask detected in the human video, we first re-identify the corresponding object and slot in robot view, and also find all similar empty slots. With corresponding object masks and slot masks, we first compute 2D keypoint matching among the detected object and mask local patches and then lift the observations to 3D to compute 6-DoF transforms.

2D object mask $\mathbf{M}^{O}_{\mathbf{H}_1}$ and the slot mask $\mathbf{M}^{S}_{\mathbf{H}_1}$ on the human first frame image $\mathbf{H}_1$. If multiple similar slots are present in the robot image, we leverage SAM [31] to propose segment candidates and use DINOv2 [48] to collect additional slot masks $\{\mathbf{M}^{S_1}_{\mathbf{R}}, \cdots, \mathbf{M}^{S_k}_{\mathbf{R}}\}$ that share similar DINOv2 features with the detected slot mask $\mathbf{M}^{S_0}_{\mathbf{R}}$. Empirically, we find that SAM2 and DINOv2 provide good enough performance on our data.

2D keypoint matching. With two corresponding masks in the human view and robot view, we use MASt3R [33] to detect 2D keypoint correspondences by expanding the masks into local 2D bounding boxes. As illustrated in Fig. 5 (middle left), we compute the 2D keypoint matching on two pairs of object local patches (between the object mask $\mathbf{M}^{O}_{\mathbf{R}}$ in the robot frame and the object mask $\mathbf{M}^{O}_{\mathbf{H}_1}$ in the initial human frame, and between the object mask $\mathbf{M}^{O}_{\mathbf{H}_1}$ in the initial human frame and the object mask $\mathbf{M}^{O}_{\mathbf{H}_n}$ in the last human frame) and one pair of slot local patches (between the slot $\mathbf{M}^{S}_{\mathbf{H}_1}$ in the initial human frame and the slot $\mathbf{M}^{S_i}_{\mathbf{R}}$ in the robot view for any slot S_i to place).

3D lifting and transformation calculation. Using the depth sensing and the camera intrinsic parameters, we can lift all human and robot view images into 3D point cloud observations. Then, we are able to lift the 2D keypoint correspondences into 3D correspondences. Equipped with the 3D correspondences, we use Procrustes analysis [21] with RANSAC [19] to calculate three 6-DoF transformation matrices for the aforementioned three local patch pair matchings. We denote the three computed 6-DoF transformations as $T^{O}_{\mathbf{R}\to\mathbf{H}}$, $T^{O}_{\mathbf{H}}$, and $T^{S_i}_{\mathbf{H}\to\mathbf{R}}$ respectively. Fig. 5 (middle right) illustrates their geometric meanings: the object transformation from the robot scene to the human scene at the start of the human video, the transformation applied by the person to the picked object in the human video, and the slot transformation from the human scene to the robot scene.

Final object placement transformations. As clearly illustrated in Fig. 5 (middle right and rightmost), by chaining up the three 6-DoF transformation matrix explained above, we can compute the final desired 6-DoF transformation matrix for the robot to execute in order to transform the pick object O from its initial position to any target slot i in the robot coordinate frame as the following

$$\mathbf{T}_i = \mathbf{T}^{S_i}_{\mathbf{H}\to\mathbf{R}}\mathbf{T}^{O}_{\mathbf{H}}\mathbf{T}^{O}_{\mathbf{R}\to\mathbf{H}}. \quad (1)$$

5. Experiments

We propose a new dataset and present an extensive evaluation of our system in Sec. 5.1, where our system, SLeRP, outperforms the baselines by a large margin. In Sec. 5.2, we present an in-depth ablation over Slot-Net and additional design choices in SLeRP. In Sec. 5.3, we show that SLeRP is effective with real-world robots.

5.1. System Evaluation

Given the novelty of the problem we address, existing evaluation benchmarks are unavailable, and there are no baseline methods with which to make direct comparisons. Consequently, we have curated a dataset comprising real-world videos and established a benchmark specific to this problem by developing suitable baselines and metrics.

Dataset. We collected 288 real-world RGB-D videos spanning 9 different object-in-slot task scenarios. For each scenario, variations were introduced in the background and the inclusion of distractor objects, camera positions, and slot occupancy conditions. The suite of tasks includes challenging, common daily activities such as *putting bread into a toaster*, *arranging eggs in an egg steamer*, and *setting mugs on coasters*. All the objects are unseen to Slot-Net during training, and 3 out of the 9 tasks encompass previously unseen task categories. Visualizations of the tasks and their varying settings are provided in the supplementary material.

Benchmark setup. Given that our task necessitates paired data comprising a human demonstration video and a novel image for the robot's view, we construct test pairs by repairing the videos in our dataset. Each pair comprises videos depicting the same object being placed into the same slot, albeit with potential variations in background, camera angle, and initial slot occupancy. We designate the first

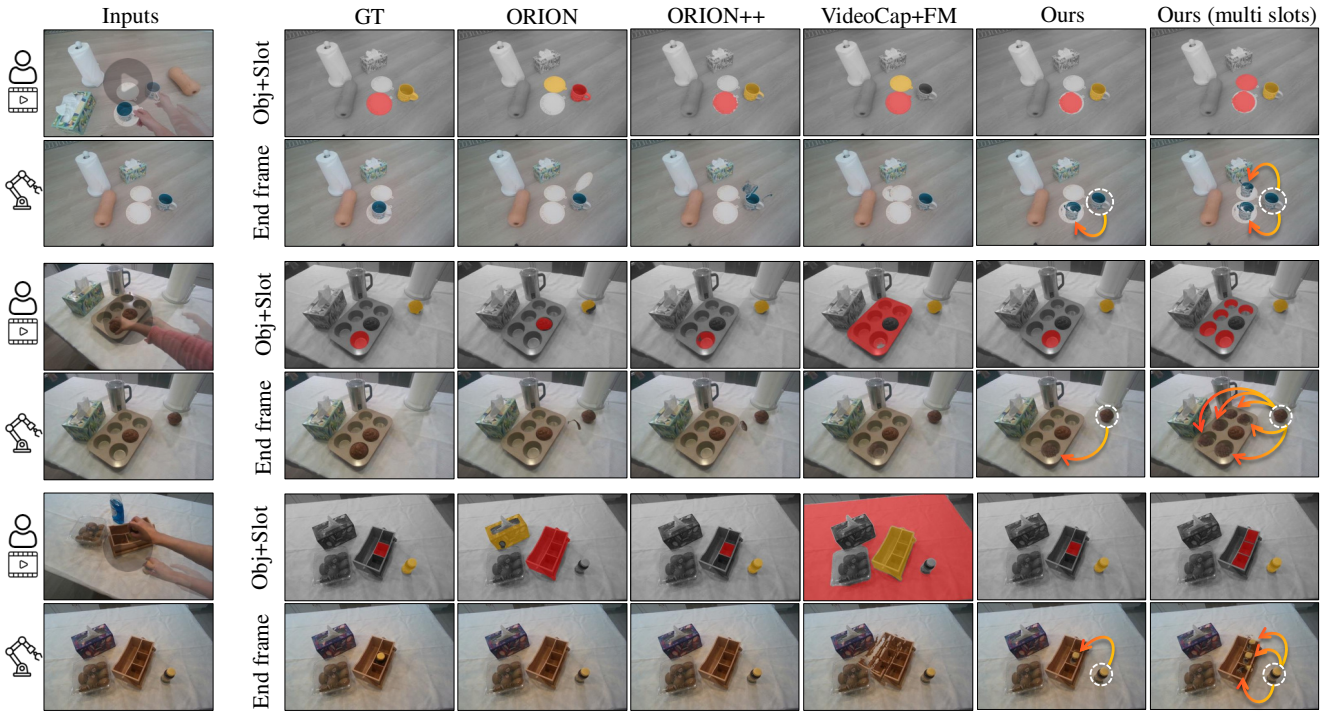


Figure 6. **Qualitative Comparison.** We compare our method to baselines and present side-by-side results on three examples. For each example, the first column shows the input human video at the top and robot-view image in the bottom. The top row displays 2D re-identification results (object in yellow, slot in red), while the bottom row shows 6-DoF relative pose predictions by projecting the object point cloud onto the slots. Unlike the baselines that can only predict one exact slot, our approach can also identify multiple slots. These results clearly demonstrate that our system outperforms the baselines, achieving accurate slot and transformation predictions.

video in each pair as the human demonstration and employ the initial frame of the second video as the robot's view, as illustrated in Fig. 6. During video data collection, we ensure that human hands are absent in the first frame. We generate three distinct test splits, introducing variations in viewpoint (288 video pairs), background (720 video pairs), and slot occupancy (288 video pairs).

Metrics. We evaluate the accuracy of the predicted 2D masks and the 6-DoF transformation matrix using five distinct metrics. For the evaluation of 2D masks, we calculate (1) the intersection-over-union (IoU) for the object mask and (2) the IoU of the exact slot mask in the robot view, comparing them to their respective ground-truth masks. We assess the accuracy of the 6-DoF transformation by transforming and projecting the object's point cloud onto the robot view, then measuring (3) the precision of the mask against the ground-truth mask. For 3D evaluation, we compute (4) the Chamfer Distance and (5) Earth Mover's Distance [17] between the transformed object point cloud and the ground-truth object point cloud at placement. If no mask or transformation output is predicted for any method, we use a default empty mask and identity matrix as the fallback predictions. To establish ground truths, we annotate the 2D masks of the object and exact slot in the start video frame (*i.e.*, robot's view), alongside the object's mask post-placement in the end frame.

Baselines. We design four baselines for comparison.

- *ORION.* Zhu *et al.* [86] perform vision-based human-to-robot imitation learning focused on object-level placement. We adapt ORION to our slot-level setting by providing the required ground-truth object and slot names. In contrast, our method automatically detects in-contact objects and slots without the need of explicit name inputs.
- *ORION++.* We leverage the object and slot detection results from SLeRP to enhance ORION, thereby establishing a stronger baseline.
- *CLIPort++.* CLIPort [61] is an end-to-end imitation-learning-based language-conditioned policy for tabletop tasks. However, the original method requires videos with action labels for training, whereas ours does not. To construct a comparison, we randomly split the tasks into training and test sets, ensuring that tasks, objects, and scenes are unseen during testing, and use the training split to train CLIPort (more details in Supplementary).
- *VideoCap+FMs.* We test whether our proposed tasks can be effectively solved using state-of-the-art foundation models. We utilize Qwen2VL [72], a video captioning model, to summarize the video, and then employ GPT4o [2] to identify the object and slot names. Subsequently, we use grounding-SAM2 [56] to generate a

Method	Different view					Different background					Different slot occupancy				
	Obj↑	Slot↑	Prec.↑	CD↓	EMD↓	Obj↑	Slot↑	Prec.↑	CD↓	EMD↓	Obj↑	Slot↑	Prec.↑	CD↓	EMD↓
ORION [86]	0.00	0.11	0.41	0.0949	0.0552	0.45	0.13	0.00	0.0932	0.0540	0.40	0.12	0.82	0.0952	0.0559
ORION++ [86]	10.21	8.89	2.10	0.1058	0.0584	7.53	5.97	0.75	0.1113	0.0597	8.83	7.44	2.95	0.1021	0.0576
CLIPort++ [61]	1.54	0.47	18.75	0.3887	0.1663	1.51	0.35	1.71	0.1152	0.0615	3.06	0.34	12.50	0.1348	0.0681
VideoCap+FMs	2.35	8.61	13.45	0.1918	0.0987	2.12	6.28	13.61	0.1743	0.0917	2.64	9.84	7.25	0.1508	0.0837
Ours	**73.85**	**54.37**	**36.40**	**0.0282**	**0.0182**	**70.27**	**44.70**	**25.39**	**0.0573**	**0.0323**	**68.12**	**47.04**	**30.30**	**0.0334**	**0.0223**

Table 1. **Quantitative System Evaluation.** We compare SLeRP with baselines and report the 2D detection and the 3D object transformation accuracy: IoU for the object mask prediction (*Obj*); IoU for the slot mask prediction (*Slot*); mask precision of the predicted object after placement onto the slot projected to the camera plane (*Prec.*); and Chamfer distance (*CD*) and Earth-Mover distance (*EMD*) between the predicted and ground-truth target object point clouds after placement. We evaluate in three different settings with the robot's views having different camera viewpoints, scene backgrounds, and initial states of the placement slot occupancy compared to the input human videos. We find that SLeRP substantially outperforms the baselines by large margins across all the metrics in all the three evaluation settings.

Method	Synthetic		Real (seen tasks)		Real (unseen tasks)	
	F1 ↑	IoU ↑	F1 ↑	IoU ↑	F1 ↑	IoU ↑
Image difference	44.10	31.34	32.69	20.04	32.20	19.53
Change detection [18]	2.27	1.58	0.06	0.03	6.30	4.19
Object mask	60.16	48.73	65.90	52.10	58.67	42.98
Object-box mask	-	-	43.39	37.74	39.37	34.65
GPT4o+SAM	0.91	0.49	4.82	2.84	4.45	2.68
Slot-Net (end image)	**83.44**	74.57	48.83	38.68	38.19	28.37
Slot-Net (ours)	82.89	**74.62**	**73.27**	**61.59**	**66.50**	**54.26**

Table 2. **Slot Segmentation Results.** We compare Slot-Net against various alternative approaches on slot detection. We evaluate on test synthetic images and our collected real-world images (seen and unseen tasks). Dashes note that the method cannot be evaluated for synthetic data given no video inputs. We can observe cleary that our Slot-Net performs the best.

bounding box and employ SAM2 [55] to produce the object and slot mask, with the transformation matrix computed using modules in the same way as in our system.

Results. Table1 presents a quantitative evaluation comparing SLeRP to four baseline methods. The results indicate that SLeRP significantly outperforms all baselines by considerable margins. Fig.6 offers qualitative comparisons, showcasing 2D object and slot mask predictions alongside 3D object transformation estimations. We observe that SLeRP generates more accurate 2D object and slot detection results, as baseline methods such as ORION and VideoCap+FMs frequently struggle to describe slot names in natural language for subsequent visual recognition (e.g., incorrectly detecting the entire placement object or table). Additionally, SLeRP achieves more precise 3D object transformations compared to baselines like CLIPort, which are primarily designed for top-down 2D predictions.

Lastly, our method can fill multiple slots, whereas other methods generate output for only a single slot. In a subset of the data with ground-truth annotations for multiple slots, SLeRP achieves IoU scores of 67.70 and 42.62 for 2D object and slot segmentation. Additionally, it attains scores of 23.14, 0.0575, and 0.0350 for 3D transformation predictions in terms of slot projection precision, Chamfer Distance (CD), and Earth Mover's Distance (EMD), respectively. These results are comparable to the single-slot placement evaluations reported in Table 1. See the supplementary materials for further details.

Method (diff. view)	SlotNet	SAM2	Mast3r	Obj ↑	Slot ↑	Prec. ↑	CD ↓	EMD ↓
Base design	✗	✗	✗	28.53	28.23	24.72	0.2289	0.1183
Ours w/o SlotNet	✗	✓	✓	**73.85**	29.63	27.84	0.0486	0.0289
Ours w/o SAM2	✓	✗	✓	31.05	35.98	23.75	0.1219	0.0667
Ours w/o Mast3r	✓	✓	✗	**73.85**	**54.37**	32.74	0.0321	0.0205
Ours	✓	✓	✓	**73.85**	**54.37**	**36.40**	**0.0282**	**0.0182**

Table 3. **Ablation Study.** All metrics follow Table 1. Results show all the key modules help. See supplementary for the full table.

5.2. Ablation Study

We evaluate the effectiveness of Slot-Net for placement slot detection in Sec. 5.2.1 and additional ablations on other components like object and slot re-identification and keypoint matching in Sec. 5.2.2.

5.2.1. Slot-Net Ablations

Baselines. We consider the following alternative and ablation approaches to replace Slot-Net.

- *Image difference.* We use the difference image between the gray-scale start and end frame, then apply thresholding to the difference image to obtain a mask.
- *Change detection.* We use an off-the-shelf change detection model [18] given two frames for the masks.
- *Object mask.* We directly use the ground-truth pick object mask as the slot mask prediction.
- *Object-box mask.* We take the pick object bounding box detected in the tracking procedure and query SAM for a proxy placement slot mask.
- *GPT4o+SAM.* We query GPT4o with start and end frames for slot bounding boxes and query SAM for masks.
- *Slot-Net (end image).* We use the end frame to replace the difference image as the prompt for Slot-Net.

Benchmark and metrics. We use our newly proposed real-world video dataset along with held-out synthetic images for evaluation. For the real images, we evaluate two splits: *seen tasks*, which involve seen object categories during training but novel object instances, and *unseen tasks*, featuring object categories not encountered during training.

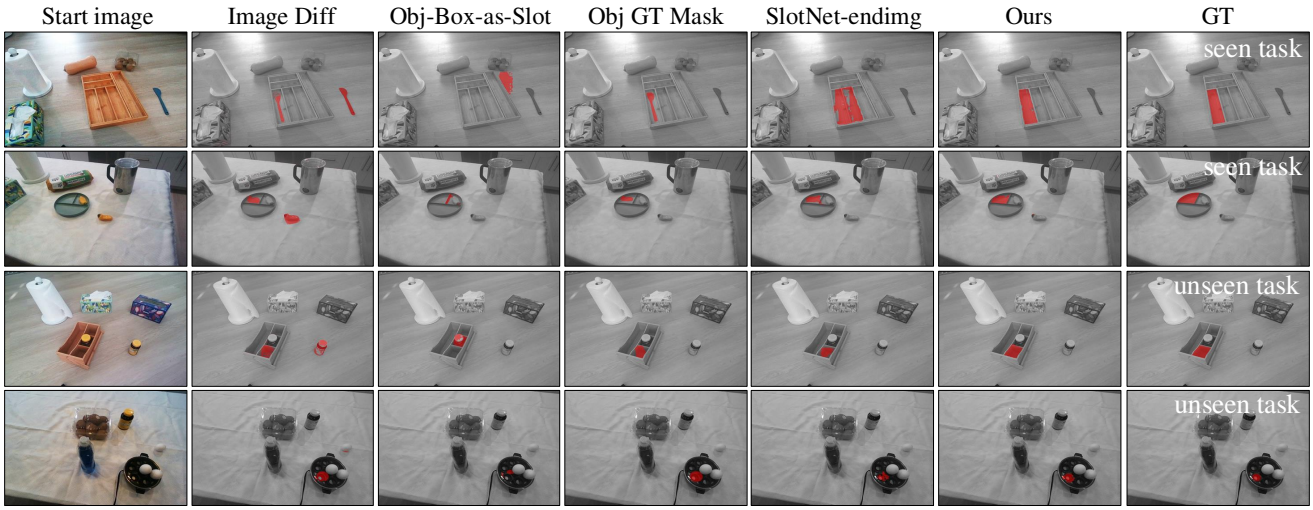


Figure 7. **Slot Detection Comparisons.** We compare Slot-Net to many alternative approaches, and results show that ours performs better.

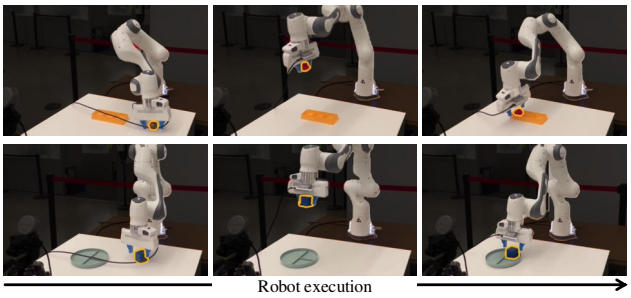


Figure 8. **Real-Robot Experiments.** We show real-robot experiments for "block into a container" and "strawberry into an organizer". See supplementary for videos and more examples.

For each real-world video, we pair the starting and ending frames as input and evaluate predictions against the ground-truth slot mask from the starting frame. The slot mask prediction performance is assessed using IoU and F1 scores.

Results. Table 2 shows that Slot-Net, when trained on synthetic data, generalizes effectively to real images, outperforming alternative approaches. Fig. 7 provides side-by-side comparisons of different methods, revealing that Slot-Net excels in identifying slot boundaries of various shapes. This underscores the necessity of training a custom model for slot detection and demonstrates that our model is both well-designed and effective.

5.2.2. Other Ablations

Beyond ablating Slot-Net, we further validate two additional key design elements in our system: utilizing SAM2 for object and slot re-identification and employing MASt3R for keypoint matching. In the absence of SAM2, we rely on DINOv2 feature similarity between the slot mask and all SAM-generated masks in the robot image. To replace MASt3R, we employ DINOv2 features for Hungarian matching. Table 9 shows the necessity of these modules.

5.3. Real-Robot Experiments

As shown in Fig. 8, we perform real-robot experiments with a Franka robot and show that SLeRP is effective for real robots. The manipulation system employs a wrist-mounted RGB-D camera (Realsense D415) and an external RGB-D camera (Realsense L515). The camera intrinsics and extrinsics relative to the robot are known. The wrist-mounted camera provides observations for SLeRP, while the external camera observes the entire scene and aids in planning collision-free trajectories. The system utilizes Contact-Graspnet[68] to generate grasps and plans collision-free trajectories using methods described in [13, 44].

6. Conclusion

In this paper, we address the novel problem of slot-level object placement by learning from a single human demonstration video. We propose a modular system to tackle this problem, which operates without requiring additional training video data and features a unique slot-level placement detector. To evaluate the system's performance, we introduce a new benchmark consisting of real-world videos and compare our system against key baseline methods. Our results demonstrate that SLeRP outperforms these baselines and functions effectively in real-robot experiments.

Limitations and future work. Given the novel problem formulation, there is potential for further research in fine-grained slot-level object placement with minimal or no human demonstrations. Future work could focus on relaxing current system assumptions, such as the static camera, single-handed interaction, and minimal motion of the placement object. Moreover, advancements in visual foundation models could enhance the robustness of our system, as they play a crucial role in this work.

References

[1] Advanced inference: Outpainting. https://huggingface.co/docs/diffusers/en/advanced_inference/outpaint. Accessed: 2010-09-30. 4, 13, 14

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[3] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022. 1, 3

[4] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023. 3

[5] Johannes Baumgartl, Tim Werner, Per Kaminsky, and Dominik Henrich. A fast, gpu-based geometrical placement planner for unknown sensor-modelled objects and placement areas. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1552–1559. IEEE, 2014. 3

[6] Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6904–6911. IEEE, 2024. 1, 3

[7] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision (ECCV)*, 2024. 3

[8] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 13

[9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 1

[10] Annie S Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from" in-the-wild" human videos. *arXiv preprint arXiv:2103.16817*, 2021. 3

[11] Tianyi Cheng, Dandan Shan, Ayda Sultan Hassen, Richard Ely Locke Higgins, and David Fouhey. Towards a richer 2d understanding of hands at scale. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 4

[12] Cleanup.pictures. https://cleanup.pictures. 4, 12, 13

[13] Michael Danielczuk, Arsalan Mousavian, Clemens Eppner, and Dieter Fox. Object rearrangement using learned implicit collision functions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6010–6017. IEEE, 2021. 3, 8

[14] Sudeep Dasari and Abhinav Gupta. Transformers for one-shot visual imitation. In *Conference on Robot Learning*, pages 2071–2084. PMLR, 2021. 3

[15] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*, pages 1183–1198. PMLR, 2023. 3

[16] Ben Eisner, Yi Yang, Todor Davchev, Mel Vecerik, Jonathan Scholz, and David Held. Deep se (3)-equivariant geometric reasoning for precise placement tasks. In *The Twelfth International Conference on Learning Representations*. 3

[17] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 6

[18] Sheng Fang, Kaiyu Li, and Zhe Li. Changer: Feature interaction is what you need for change detection. *arXiv preprint arXiv:2209.08290*, 2022. 2, 7

[19] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5

[20] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):265–293, 2021. 1

[21] Colin Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(2):285–321, 1991. 5

[22] Kensuke Harada, Tokuo Tsuji, Kazuyuki Nagata, Natsuki Yamanobe, and Hiromu Onda. Validating an object placement planner for robotic pick-and-place tasks. *Robotics and Autonomous Systems*, 62(10):1463–1477, 2014. 3

[23] Joshua A Haustein, Kaiyu Hang, Johannes Stork, and Danica Kragic. Object placement planning and optimization for robot manipulators. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7417–7424. IEEE, 2019. 3

[24] Haojie Huang, Dian Wang, Robin Walters, and Robert Platt. Equivariant transporter network. In *Robotics: Science and Systems*, 2022. 3

[25] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 2

[26] Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, et al. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. *arXiv preprint arXiv:2403.12943*, 2024. 1, 3

[27] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022. 3

[28] Yun Jiang, Marcus Lim, Changxi Zheng, and Ashutosh Saxena. Learning to place new objects in a scene. *The International Journal of Robotics Research*, 31(9):1021–1043, 2012. 3

[29] Aditya Kannan, Kenneth Shaw, Shikhar Bahl, Pragna Mannam, and Deepak Pathak. Deft: Dexterous fine-tuning for hand policies. In *Conference on Robot Learning*, pages 928–942. PMLR, 2023. 3

[30] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 4, 5

[32] Jangwon Lee and Michael S Ryoo. Learning robot activities from first-person human videos using convolutional future regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–2, 2017. 3

[33] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 5

[34] Siyuan Li, Lei Ke, Martin Danelljan, Luigi Piccinelli, Mattia Segu, Luc Van Gool, and Fisher Yu. Matching anything by segmenting anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18963–18973, 2024. 4

[35] Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6322–6329. IEEE, 2022. 3

[36] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1118–1125. IEEE, 2018. 3

[37] Tomás Lozano-Pérez, Joseph L. Jones, Emmanuel Mazer, and Patrick A. O'Donnell. Task-level planning of pick-and-place robot motions. *Computer*, 22(3):21–29, 1989. 3

[38] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*. 3

[39] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36:655–677, 2023. 3

[40] Zhao Mandi, Fangchen Liu, Kimin Lee, and Pieter Abbeel. Towards more generalizable one-shot visual imitation learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2434–2444. IEEE, 2022. 3

[41] Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning*, pages 651–661. PMLR, 2022. 3

[42] Oier Mees, Alp Emek, Johan Vertens, and Wolfram Burgard. Learning object placements for relational instructions by hallucinating scene representations. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 94–100. IEEE, 2020. 3

[43] Chaitanya Mitash, Rahul Shome, Bowen Wen, Abdeslam Boularias, and Kostas Bekris. Task-driven perception and manipulation for constrained placement of unknown objects. *IEEE Robotics and Automation Letters*, 5(4):5605–5612, 2020. 3

[44] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Adam Fishman, and Dieter Fox. Cabinet: Scaling neural collision detection for object rearrangement with procedural scene generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1866–1874. IEEE, 2023. 3, 8

[45] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR, 2023. 3

[46] Rhys Newbury, Kerry He, Akansel Cosgun, and Tom Drummond. Learning to place objects onto flat surfaces in upright orientations. *IEEE Robotics and Automation Letters*, 6(3): 4377–4384, 2021. 3

[47] Anh Nguyen, Dimitrios Kanoulas, Luca Muratore, Darwin G Caldwell, and Nikos G Tsagarakis. Translating videos to commands for robotic manipulation with deep recurrent neural networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3782–3788. IEEE, 2018. 3

[48] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*. 5

[49] Helena Örnkloo and Claes von Hofsten. Fitting objects into holes: on the development of spatial cognition skills. *Developmental psychology*, 43(2):404, 2007. 1

[50] Chuer Pan, Brian Okorn, Harry Zhang, Ben Eisner, and David Held. Tax-pose: Task-specific cross-pose estimation for robot manipulation. In *Conference on Robot Learning*, pages 1783–1792. PMLR, 2023. 3

[51] Austin Patel, Andrew Wang, Ilija Radosavovic, and Jitendra Malik. Learning to imitate object interactions from internet videos. *arXiv:2211.13225*, 2022. 3

[52] Chris Paxton, Chris Xie, Tucker Hermans, and Dieter Fox. Predicting stable configurations for semantic placement of novel objects. In *5th Annual Conference on Robot Learning*. 3

[53] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and

Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*. 4

[54] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022. 3

[55] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4, 7

[56] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, Yuda Xiong, Hao Zhang, Feng Li, Peijun Tang, Kent Yu, and Lei Zhang. Grounding dino 1.5: Advance the "edge" of open-set object detection, 2024. 6

[57] Brian Rooks. Machine tending in the modern age. *Industrial Robot: An International Journal*, 30(4):313–318, 2003. 1

[58] Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. In *8th Annual Conference on Robot Learning*. 3

[59] Aditya Sharma, Luke Yoffe, and Tobias Höllerer. Octo+: A suite for automatic open-vocabulary object placement in mixed reality. In *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, pages 157–165. IEEE, 2024. 3

[60] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, pages 654–665. PMLR, 2023. 3

[61] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021. 2, 3, 6, 7

[62] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022. 3

[63] Anthony Simeonov, Yilun Du, Yen-Chen Lin, Alberto Rodriguez Garcia, Leslie Pack Kaelbling, Tomás Lozano-Pérez, and Pulkit Agrawal. Se (3)-equivariant relational rearrangement with neural descriptor fields. In *Conference on Robot Learning*, pages 835–846. PMLR, 2023. 3

[64] Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos. *arXiv preprint arXiv:2409.08273*, 2024. 3

[65] Toronto Annotation Suite. https://aidemos.cs.toronto.edu/toras. 4, 12, 13

[66] Yu Sun, Shaogang Ren, and Yun Lin. Object–object interaction affordance learning. *Robotics and Autonomous Systems*, 62(4):487–496, 2014. 3

[67] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Millane, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. CuRobo: Parallelized collision-free robot motion generation. In *ICRA*, 2023. 1

[68] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Fox Dieter. Contact-graspnet: Efficient 6-dof grasp generation in clutteredscenes. *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 8

[69] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 4, 13, 14, 18, 19

[70] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 4, 12, 13

[71] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023. 3

[72] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6

[73] Yixuan Wang, Mingtong Zhang, Zhuoran Li, Tarik Kelestemur, Katherine Rose Driggs-Campbell, Jiajun Wu, Li Fei-Fei, and Yunzhu Li. D3 fields: Dynamic 3d descriptor fields for zero-shot generalizable rearrangement. In *8th Annual Conference on Robot Learning*, 2024. 3

[74] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems*, 2024. 3

[75] Hongtao Wu, Jikai Ye, Xin Meng, Chris Paxton, and Gregory S Chirikjian. Transporters with visual foresight for solving unseen rearrangement tasks. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10756–10763. IEEE, 2022. 3

[76] Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7834. IEEE, 2021. 1, 3

[77] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In *8th Annual Conference on Robot Learning*. 3

[78] Yezhou Yang, Yi Li, Cornelia Fermuller, and Yiannis Aloimonos. Robot learning manipulation action plans by" watch-

ing" unconstrained videos from the world wide web. In *Proceedings of the AAAI conference on artificial intelligence*, 2015. 3

[79] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024. 3

[80] Tianhe Yu, Chelsea Finn, Sudeep Dasari, Annie Xie, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *Robotics: Science and Systems XIV*, 2018. 3

[81] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024. 3

[82] Wentao Yuan, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. M2t2: Multi-task masked transformer for object-centric pick and place. In *7th Annual Conference on Robot Learning*. 3

[83] Kevin Zakka, Andy Zeng, Johnny Lee, and Shuran Song. Form2fit: Learning shape priors for generalizable assembly from disassembly. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9404–9410. IEEE, 2020. 3

[84] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021. 3

[85] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 13

[86] Yifeng Zhu, Arisrei Lim, Peter Stone, and Yuke Zhu. Vision-based manipulation from single human video with open-world object graphs. *arXiv preprint arXiv:2405.20321*, 2024. 1, 2, 3, 6, 7

Contents

1. Introduction 1

2. Related Work 3

3. Problem Formulation 3

4. Method 3
- 4.1. System Overview . . . 3
- 4.2. Parsing the Human Demonstration Video . . 4
- 4.3. Correlating to the Robot-view Image 4

5. Experiments 5
- 5.1. System Evaluation . . . 5
- 5.2. Ablation Study . . . 7
  - 5.2.1 . Slot-Net Ablations . . . 7
  - 5.2.2 . Other Ablations . . . 8
- 5.3. Real-Robot Experiments . . . 8

6. Conclusion 8

A. Slot-Net Data Generation 12
- A.1. Details, Statistics and Examples . . . 12
- A.2. Synthetic Data Generation Process . . . 13
- A.3. Limitations . . . 14

B. Evaluation Videos 14
- B.1. Visualization of 9 tasks . . . 14
- B.2. Visualization of 3 settings . . . 14

C. Additional Implementation Details 14
- C.1. Slot-Net . . . 14
- C.2. CLIPort++ . . . 14

D. Additional Results of SLeRP 14
- D.1. Full Ablation Study Table . . . 14
- D.2. Multi-slots analysis. . . . 14
- D.3. More baseline comparison results . . . 15
- D.4. More ablation comparison results . . . 15
- D.5. Limitations . . . 15

E. Additional Results of Slot-Net 15
- E.1. Additional results of Slot-Net . . . 15
- E.2. Limitations . . . 15

A. Slot-Net Data Generation

A.1. Details, Statistics and Examples

Object-with-slots image collection. We collect object-centric images of items with slots by photographing them in everyday environments. These images serve as the *end* object crops, and we generate corresponding *start* images by removing objects, thus creating (*start, end*) image pairs for a pick-and-place action. During data collection, participants were instructed to partially fill the slots. This approach ensures that slots are not entirely empty, facilitating object removal, while avoiding fully filled slots, which can complicate inpainting during object removal. In total, we have collected 2,138 object-centric images spanning 67 categories, as detailed in Tab. 4.

Pick-object removal. The captured image serves as an object-centric *end* image. To create an object-centric *start* image, where objects are absent from the slot, we employ the SDXL [70] inpainting model to remove the pick-object. For cases where SDXL does not successfully remove the object, we use Cleanup.pictures [12] in a subsequent round to achieve more effective removal results.

Slot mask annotation. After obtaining the (*start, end*) object-centric images, we annotate the slot mask by comparing each pair using TORAS [65]. We perform mask annotation prior to augmentation and automatically transform these masks onto the canvas, enabling us to generate labels for a large dataset.

Task Name	#(obj centric)	#synthetic
battery in batter compartment	20	50
bead in bead organizer	41	4,291
block in a toy vehicle	12	102
book in book holder	22	2,207
bottle in box	28	2927
bottle in organizer	87	7,810
bread in toaster	84	3,896
can in box	5	50
can on board	15	557
candle in organizer	5	450
capsule in tray	20	1,966
cards in organizer	5	51
choclate in organier	22	200
coin in organizer	6	451
cup in cardboard tray	101	10,541
cup on board	5	245
cup on coaster	5	199
cylinder in organizer	49	1,705
egg in egg carton	205	36,758
egg in egg steamer	13	649
egg in tray	22	2,116
flower in organier	15	250
food in organizor	123	9,117
food in tray	10	1,250
fruit in box	5	450
fruit in food organizer	35	3,146
fruit in organizer	155	11,303
glass in organizer	5	400
glass on coaster	7	200
glove in organizer	12	400
ice cube in ice tray	31	661
jewerly in organizer	18	960
key in organizer	6	900
lip stick in organizer	14	256
muffin in muffin tray	63	4,150
mug on board	10	310
mug on coaster	66	2,453
notepad in box	15	901
peach in box	12	903
peg in wood base	8	1,413
pen in basket	15	793
pen in cup	20	994
pen in organizer	103	8,690
pen in pen holder	76	3,305
pen on book	30	1,335
pepper in tray	10	492
pill in pill organizer	15	462
plant in vase	12	387
rectangle box in tray	35	366
tangerine in muffin tray	24	1,700
tool in organizer	61	400
toy car in organizer	14	603
tube in tray	15	1,400
utensil in bowl	20	850
utensil in cup	19	2,311
utensil in utensil organizer	146	14,976
wood in organizer	9	1,100

Table 4. Statistics for object-centric images. This table shows the task names for the object-centric images we collected (col 1), the amount of object-centric images (col 2), and the amount of corresponding generated images (col 3).

Figure 9. **Data augmentation.** We show 9 synthetic data augmentation examples for the "egg in egg carton" task. In training data generation, we use x100 augmentation for each slot in the object-centric images.

Augmentation with outpainting. For each object-centric slot object, we create a larger scene by first sampling random locations on a 1024x1024 canvas and then applying outpainting with various generated prompts. Using a short category name, such as "bread in toaster", we employ Llama [69] to enrich the text prompts with descriptions of the environments. We apply outpainting to the background following the Hugging Face Outpainting script [1] with enriched texts to create the *end* image. The outpainting script incorporates ControlNet [85], SDXL [70], and ZoeDepth [8]. Subsequently, the start object crop images and slot masks are subjected to the same transformation to create the outpainted (*start, end*) image pairs and the corresponding ground-truth slot masks. In total, we apply 100 augmentations for each slot on the object-centric image, resulting in 156,000 images for training. Examples of "egg in egg carton" augmentation are shown in Fig. 9.

A.2. Synthetic Data Generation Process

We present examples of our synthetic generation process in Tab. 5 and Tab. 6. Each row in the table, from left to right, illustrates the step-by-step process for one example.

Beginning with a task name (col 1) and an object-centric image (col 3), we first enhance the description using Llama [69] (col 2). Next, we perform object removal from the slots through two rounds of image inpainting using SDXL [70] and Clean.pictures [12] (col 4) to ensure more effective results. We annotate the slot mask on the object-centric image utilizing the Toronto Annotation Suite (TORAS) [65] (col 5).

For generating diverse daily backgrounds, we transform

the object-centric image (col 3) onto a 1024x1024 canvas and outpaint the background using the Hugging Face Outpainting script [1] with the enriched text generated by Llama [69] (col 2) as a prompt to create the *end* image. Finally, the *start* image and ground-truth mask undergo the same transformation onto a 1024x1024 canvas to produce the (*start*, *end*, mask) triplet samples for training.

A.3. Limitations

Our data generation pipeline for creating (*start*, *end*) pairs with minimal data collection and annotation is highly efficient, enabling us to train SAM with manageable effort.

Despite its efficiency, the generated images have certain limitations. First, the diversity of the generated image styles is insufficient. Although we employ Llama to enrich text prompts for backgrounds and request diverse styles, the generated images lack sufficient diversity and realism. Second, the generated images sometimes defy physical plausibility. In the process of outpainting the object-centric images, we sample random locations and rotations on the canvas. Occasionally, these locations or rotations are challenging to outpaint while adhering to physical principles, resulting in images that appear unrealistic.

The purpose of outpainting data augmentation is to generate more varied backgrounds, thereby aiding Slot-Net in achieving generalization. In this context, the appearance of the background does not adversely affect our use case. We anticipate that advances in generative AI models will mitigate these limitations.

B. Evaluation Videos

In this paper, we developed a pick-and-place video dataset containing 288 videos spanning 9 tasks for evaluating both Slot-Net and SLeRP.

For Slot-Net evaluation, we utilize the *start* and *end* images from each video. Among the 9 tasks, 3 are categorized as unseen for Slot-Net evaluation: "bottle in organizer", "cup on saucer", and "egg in egg steamer".

For SLeRP evaluation, we select two videos of the same task, one serving as the human demonstration and the other as the robot-view video. The robot-view video has its final frame depicting what the robot's view will look like upon completion of the action, providing the ground-truth *end* frame appearance. The three different settings we use to pair the videos are: (1) different views (captured from different camera angles), (2) different backgrounds (captured in different environments with setups on distinct tables), and (3) different slot occlusions (captured with varying slot occlusions, where one video has all other slots empty while the other has some slots filled).

In this section, we provide visualizations of the 9 tasks and 3 settings in Tab. 7 and Tab. 8.

B.1. Visualization of 9 tasks

We show one video for each of the 9 tasks in Tab. 7. For each video, we out the (*start*, *pick*, *place*, *end*) frames to illustrate the action.

B.2. Visualization of 3 settings

We show two example videos for each of the 3 settings (different views, different backgrounds, and different slot occlusions) in Tab. 8.

C. Additional Implementation Details

C.1. Slot-Net

We follow SAM's training recipe for finetuning the ViT-based encoder using a combination of Dice Loss and Cross-Entropy Loss. We train the model on an A100 GPU for 72K iterations. We use a learning rate of 1×10^{-5} with a batch size of 8.

C.2. CLIPort++

We start with the released code from the authors. To prepare labels for training from our RGBD videos, we treat our videos as a one-step action with the corresponding task name as the language goal. And we treat the center of the object mask as the pick action location and the center of the slot mask as the place action location. We use 80 videos for training and 64 videos for evaluation, ensuring there is no overlap in tasks, objects, and backgrounds.

D. Additional Results of SLeRP

D.1. Full Ablation Study Table

Tab. 9 presents the full ablation study results evaluated all all the three different settings.

D.2. Multi-slots analysis.

Tab. 10 further provides 1-to-another slot placement results, placing into a different slot as human videos, and shows ours generalizes well.

	Diff Slot Placement				
	Obj↑	Slot↑	Prec.↑	CD↓	EMD↓
Ours	67.70	42.62	23.14	0.0575	0.0350

Table 10. Object placement into a different slot as in the human video.

To evaluate DINOv2+SAM 1-to-N slot re-identification, we annotated all N slots, and Table 11 shows that our method does reliably well in finding all equivalent slots.

	Diff View		Diff Bg		Diff Slot Occ.	
	mIoU	AP	mIoU	AP	mIoU	AP
Ours	68.3	46.3	66.8	44.2	66.0	43.3

Table 11. Multi-slot mask identification evaluation.

D.3. More baseline comparison results

We show more qualitative comparison results for baselines in Fig. 10.

D.4. More ablation comparison results

We show more qualitative comparison results for ablations in Fig. 11.

D.5. Limitations

Our system, SLeRP, demonstrates superior performance compared to baseline methods. However, there are areas that require improvement. Firstly, SLeRP is modular, with each module relying on the output of the preceding one, which may result in compounding errors. If a preceding module fails to provide accurate outputs, subsequent results may be adversely affected. Secondly, while the modules currently employed are not without flaws, they can be easily replaced with more advanced methods.

For instance, Slot-Net necessitates that the *start* and *end* images have minimal changes aside from the pick object; future methodologies could relax this requirement, enabling slot detection in more general settings. Additionally, SAM2 is a tracking method repurposed for re-identification, and methods specifically designed for re-identification would likely yield better results. Lastly, while Mast3r is effective for object-level matching, it struggles with slot-level matching, indicating that improved slot-level matching algorithms could enhance accuracy.

E. Additional Results of Slot-Net

E.1. Additional results of Slot-Net

We provide more Slot-Net comparison results in Tab. 12 and random results in Tab. 13.

E.2. Limitations

Slot-Net has limitations in slot segmentation across varying conditions due to its training data constraints: (1) It requires that the input (*start*, *end*) images have minimal changes in viewpoint. (2) It necessitates that the pick object is the sole changing element between (*start*, *end*) images, and neither humans nor human hands are present. Slot-Net may exhibit the following issues: (1) While it can generalize to unseen tasks, its performance may degrade. (2) The predicted slot may encompass multiple slots or extend beyond the ground truth. (3) It may incorrectly identify pick objects as slots.

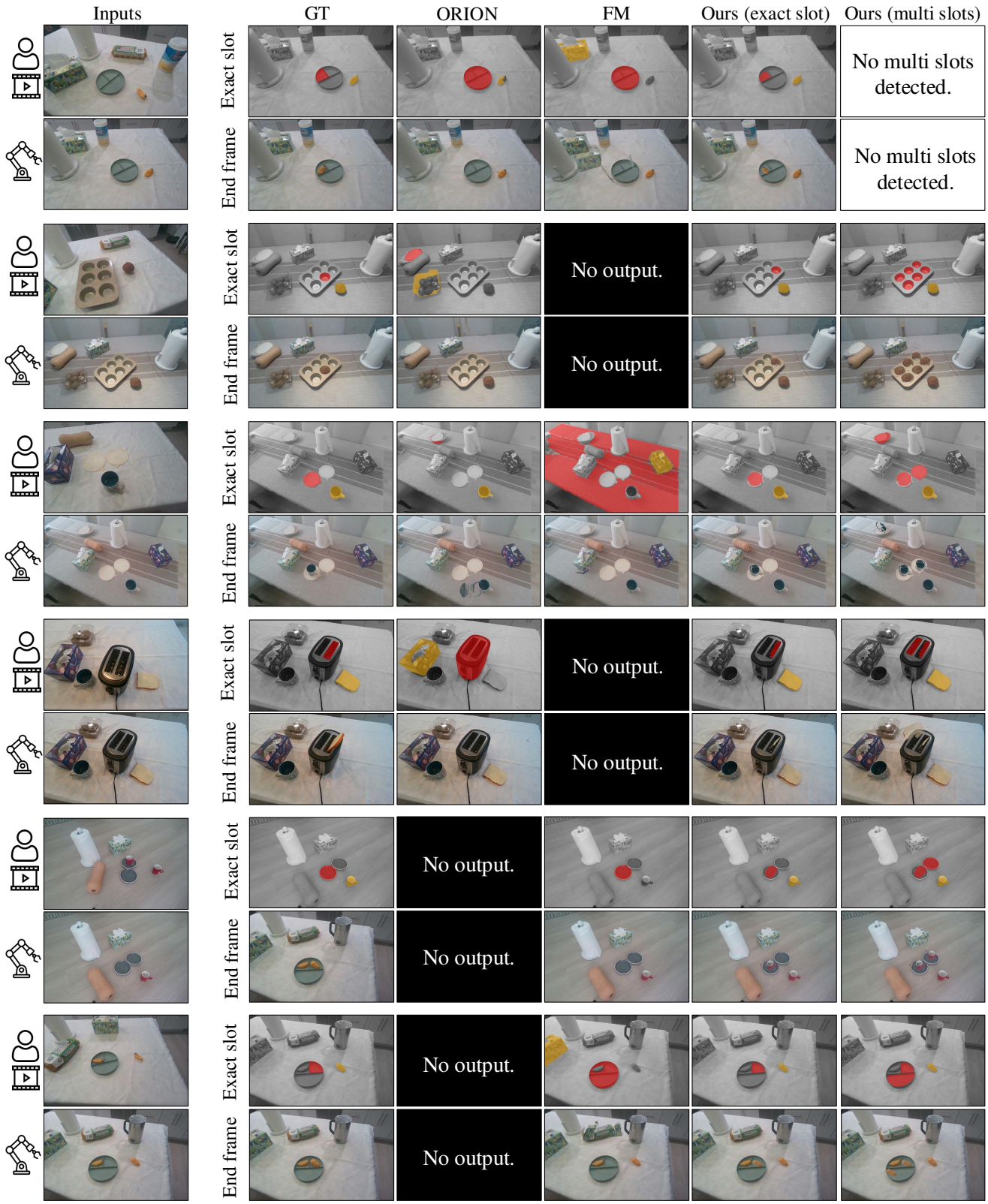


Figure 10. **Qualitative Comparison for baselines.** We compare our method to baselines and present side-by-side results on six examples. For each example, the top row displays 2D slot prediction results, while the bottom row shows 6-DoF relative pose predictions by projecting the object point cloud onto the slots.

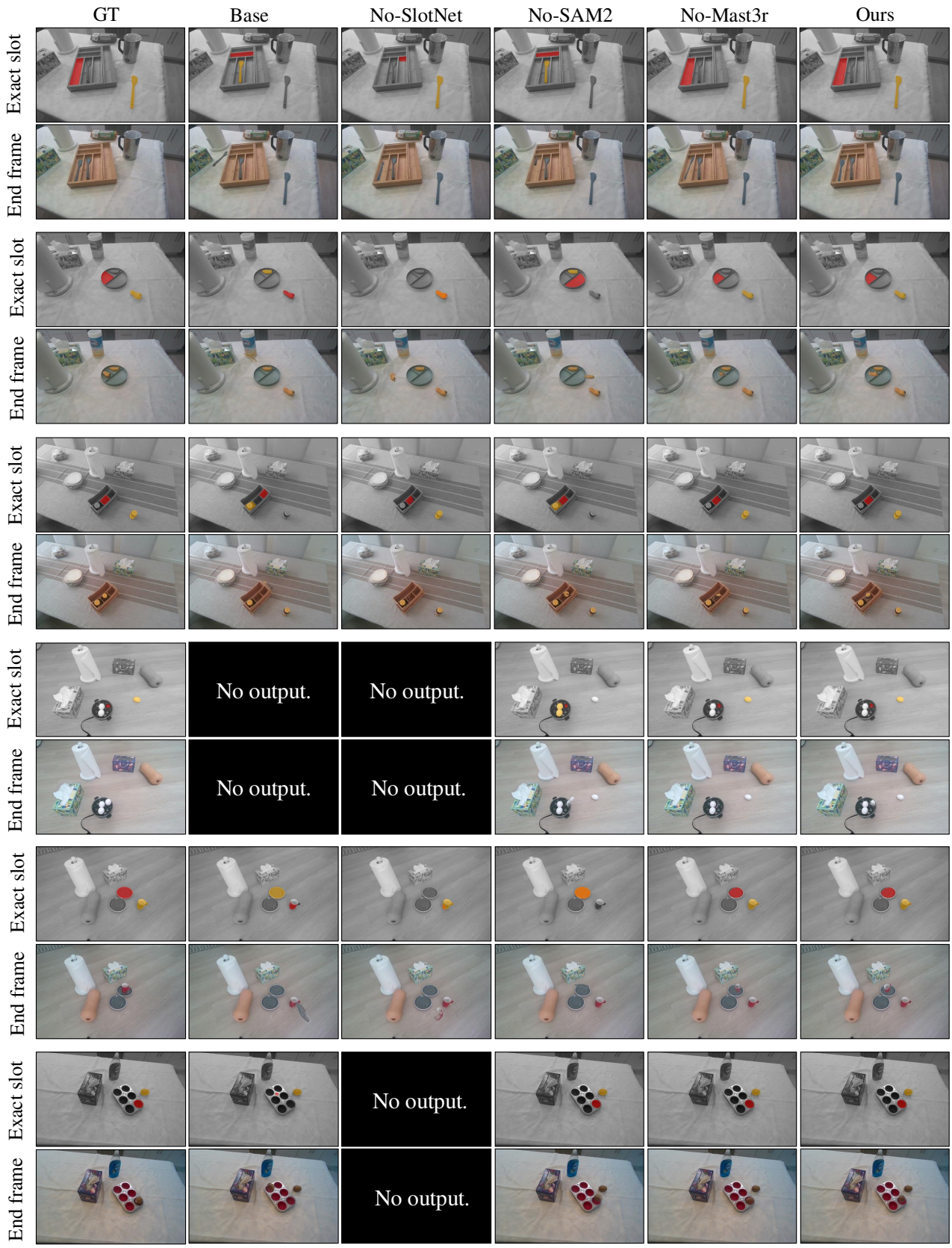


Figure 11. **Qualitative Comparison for ablations.** We compare our method to ablations and present side-by-side results on six examples. For each example, the top row displays 2D slot prediction results, while the bottom row shows 6-DoF relative pose predictions by projecting the object point cloud onto the slots.

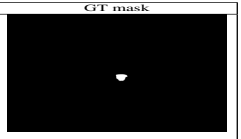
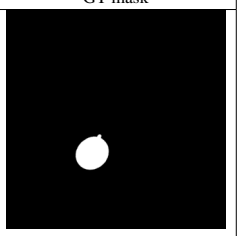
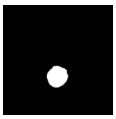
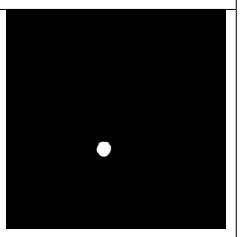
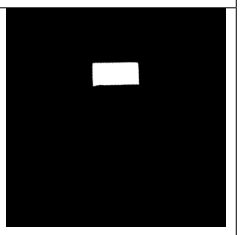
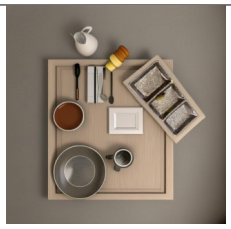
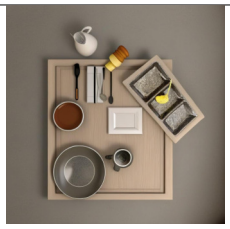
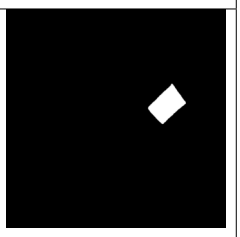
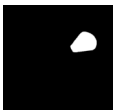
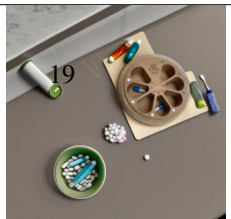
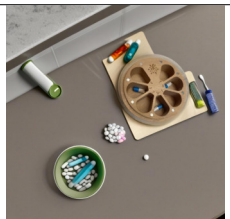
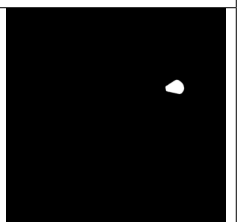
Task	Enrich text	Obj-centic img	Obj removal	Annotation	*Start*	*End*	GT mask
egg in egg carton	'top': 'egg', 'bottom': 'egg carton', 'style': 'High-quality and photorealistic, modern, bright, morning, 45-degree look-down view, 2020s', 'objects': 'A large, brown egg with a smooth, glossy texture, positioned in the center of a compact, rectangular egg carton with a textured, ribbed surface, nestled snugly in its compartment.', 'background': 'A realistic, everyday kitchen scene, with a plain, white countertop, a glass jar of jam, a toaster, a stack of plates, a wooden cutting board, a wood countertop, and soft, creamy white walls.'						
bottle in box	'top': 'box', 'bottom': 'bottle is inside the box', 'style': '2020s contemporary style with bright and luminous brightness, set in the morning time with an eye-level shot', 'objects': 'The bottle is a sleek, modern glass bottle with a smooth texture and a minimalist design. It has a simple label and a narrow neck. The box is a sturdy, corrugated cardboard box with a textured surface and a classic design.', 'background': 'A realistic, everyday living room scene with a few daily supplies around the box and bottle, including a coffee table with a vase, a few books, and a magazine on the floor. The ground, with neutral-colored walls and a light-colored rug on the floor.'						
bread in toaster	'top': 'toaster', 'bottom': 'bread', 'style': 'A 2020s, modern, bright, and luminous style with a 45-degree look-down view, capturing the morning scene with a sense of coziness and warmth.', 'objects': 'The toaster, a sleek and modern appliance with a stainless steel finish and a compact design, sits proudly on the countertop. The bread, a soft and fluffy white loaf, is nestled inside the toaster, ready to be toasted to perfection.', 'background': 'The countertop is a warm and inviting beige color, with a few subtle scratches and scuffs to give it a sense of character. A few daily supplies are scattered around the toaster, including a jar of jam, a carton of eggs, and a loaf of bread, adding to the sense of a busy morning routine.'						
egg in egg steamer	'top': 'egg', 'bottom': 'egg steamer', 'style': 'High-quality and photorealistic, modern, 2020s, bright, noon afternoon, 45-degree look-down view', 'objects': 'The egg is a large, brown egg with a smooth, glossy texture, placed in the egg steamer basket. The egg steamer is a stainless steel, modern kitchen appliance with a sleek and shiny surface, and a simple, geometric design.', 'background': 'A modern kitchen with a white countertop, a stainless steel sink, and a few kitchen utensils and appliances around. A few kitchen towels and a fruit bowl are nearby, adding to the daily living environment.'						
fruit in food organizer	'top': 'fruit', 'bottom': 'food organizer', 'style': 'A contemporary 2020s style with a bright and luminous ambiance, a 45-degree look-down view, and a medium shot of the objects to capture both the fruit and the organizer in detail.', 'objects': 'The fruit, a vibrant and juicy apple, is placed neatly in the organizer's container, showcasing its natural texture and color. The organizer itself has a sleek and modern design, with a matte finish and a subtle sheen to it. The material appears to be a high-quality plastic, with a slight give when touched.', 'background': 'The background is a realistic representation of a modern kitchen, with a plain white wall behind the fruit and the organizer. There's a wooden kitchen table to the left, with a few utensils and a water bottle scattered around it. The overall ambiance is bright and inviting, with a glimpse of a cityscape outside.'						
muffin in muffin tray	'top': 'muffin', 'bottom': 'muffin tray', 'style': 'A high-quality, photorealistic style with a 2020s modern aesthetic, bright and warm and cozy atmosphere, a 45-degree look-down view, and a medium shot distance.', 'objects': 'The muffin in the muffin tray is a delicious treat with a soft, fluffy texture and a crispy edge. The muffin tray, made of durable, stainless steel, has a sleek, modern design with a slight sheen to it, giving it a premium look.', 'background': 'The background is a realistic, everyday kitchen scene, complete with a stainless steel refrigerator, a wooden kitchen island, and a few kitchen utensils scattered about.'						

Table 5. Data generation process (Part 1/2). Start from a task name (col 1) and an object-centric image (col 3), we first get an enriched detailed description using [69] (col 2), then remove one object from the slots (col 4) and annotate the slot mask on the object-centric image (col 5). To put the object-centric image into various daily backgrounds, we transform the object-centric image (col 3) onto a 1024x1024 canvas and outpaint the background with the enriched text (col 2) to create *End*. Finally, *Start* and GT mask follow the same transformation onto the canvas.

Task	Enrich text	Obj-centic img	Obj removal	Annotation	*Start*	*End*	GT mask
mug on coaster	'top': 'mug', 'bottom': 'coaster', 'style': 'High-quality and photorealistic, 2020s, modern, bright, morning time, 45-degree look-down view', 'objects': 'A ceramic mug with a textured, matte finish, having a simple yet elegant design with a subtle sheen to it, and a wooden coaster with a natural, rustic texture, having a subtle wood grain pattern', 'background': 'A realistic, real-world scene of a kitchen table with a few daily supplies around, such as a jar of coffee beans, a sugar bowl, and a few coffee cups, with a wooden table finish and light, neutral-colored walls'						
peach in box	'top': 'peach', 'bottom': 'box', 'style': 'mid-century modern, high-quality and photorealistic, bright and cozy, morning, 45-degree look-down view', 'objects': 'The peach has a smooth, juicy texture, and a vibrant orange color. The box has a wooden texture, with a natural wood grain pattern, and a soft brown color.', 'background': 'A realistic kitchen scene, with a wooden table and chairs, and a few daily supplies such as a toaster, a coffee maker, and a vase with fresh flowers.'						
pen in organizer	'top': 'pen', 'bottom': 'organizer', 'style': 'high-quality and photorealistic, modern European-style, bright and cozy, morning, 45-degree look-down view', 'objects': 'The pen is a sleek and modern writing instrument with a silver finish and a rubberized grip. The organizer is a compact and functional desk accessory with a wooden base and a mesh pocket.', 'background': 'The background is a realistic depiction of a modern home office, with a wooden desk and a comfortable office chair. There are several daily supplies around the organizer, including a notebook, a stapler, and a cup of coffee.'						
pepper in tray	'top': 'pepper', 'bottom': 'tray', 'style': 'high-quality and photorealistic, 2020s, contemporary, bright, luminous, morning, 45-degree look-down view', 'objects': 'a medium-sized, green bell pepper with a glossy texture, placed in the center of a rectangular, stainless steel tray with a smooth, matte finish', 'background': 'a wooden kitchen table with a few daily supplies, a warm, beige-colored kitchen wall, cabinets, and a window that lets in natural light'						
pill in							

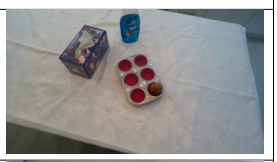
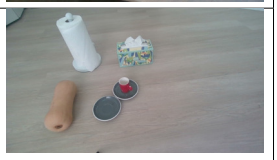
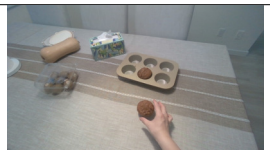
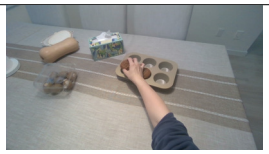
No.	Task	#Total	Start Frame	Pick Frame	Place Frame	End Frame
1	bread in toaster	32				
2	bottle in organizer	32				
3	fruit in tray	32				
4	cup in saucer	32				
5	mug in coaster	32				
6	egg in egg steamer	32				
7	fruit in organizer	32				
8	utensil in utensil organizer	32				
9	muffin in muffin tray	32				

Table 7. Evaluation videos for 9 tasks. We show one example video for each of the 9 tasks from our evaluation video dataset. The 4 frames showing for each video are the start frame, pick frame, place frame, and end frame in the video. We get the pick and place frame indexes from the in-contact object trajectory, where the pick frame index is when the in-contact object is first detected, and the place frame index is when the in-contact object is last detected in the video.

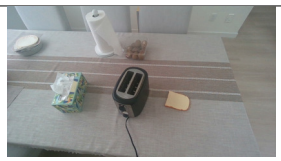
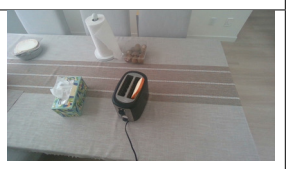
Settings	Video 1 - Start Frame	Video 1 - End Frame		Video 2 - Start Frame	Video 2 - End Frame
Different view					
Different Background					
Different Slot Occlustion					
Settings	Video 1 - Start Frame	Video 1 - End Frame		Video 2 - Start Frame	Video 2 - End Frame
Different view					
Different Background					
Different Slot Occlustion					

Table 8. Evaluation videos in 3 settings. We show two example videos ("bread in toaster", "muffin in muffin tray") in 3 different setting. For each task of our evaluation videos, we captured videos for 3 different settings to evaluate system performance when the human demonstration video and its paired robot-view video were in different conditions. The 3 different settings we captured are (1) different views (two videos are captured from different camera views), (2) different backgrounds (two videos are captured from different backgrounds where we set up the two scenes on two different tables), and (3) different slot occlusions (two videos are captured from different slot occlusions, where one video with all other slots empty while the other video with some slots full).

Method	SlotNet	SAM2	Mast3r	Diff. View					Diff. Background					Diff. Slot Occu.				
				Obj ↑	Slot ↑	Prec. ↑	CD ↓	EMD ↓	Obj ↑	Slot ↑	Prec. ↑	CD ↓	EMD ↓	Obj ↑	Slot ↑	Prec. ↑	CD ↓	EMD ↓
Base design	✗	✗	✗	28.53	28.23	24.72	0.2289	0.1183	33.27	23.40	22.22	0.1217	0.0643	27.71	27.05	26.47	0.1499	0.0809
Ours w/o SlotNet	✗	✓	✓	**73.85**	29.63	27.84	0.0486	0.0289	**70.27**	23.91	18.80	0.0630	0.0367	**68.12**	26.33	25.06	0.0520	0.0315
Ours w/o SAM2	✓	✗	✓	31.05	35.98	23.75	0.1219	0.0667	33.98	30.87	16.99	0.0870	0.0505	28.62	37.48	22.97	0.1012	0.0575
Ours w/o Mast3r	✓	✓	✗	**73.85**	**54.37**	32.74	0.0321	0.0205	**70.27**	**44.70**	**28.93**	**0.0540**	**0.0318**	**68.12**	**47.04**	27.38	0.0675	0.0374
Ours	✓	✓	✓	**73.85**	**54.37**	**36.40**	**0.0282**	**0.0182**	**70.27**	**44.70**	25.39	0.0573	0.0323	**68.12**	**47.04**	**30.30**	**0.0334**	**0.0223**

Table 9. **System Ablation Studies.** We conduct ablation studies on the key modules in SLeRP. All metrics and test splits follow Table 1. Results show all the key modules help, and our full system performs the best on average across different settings.

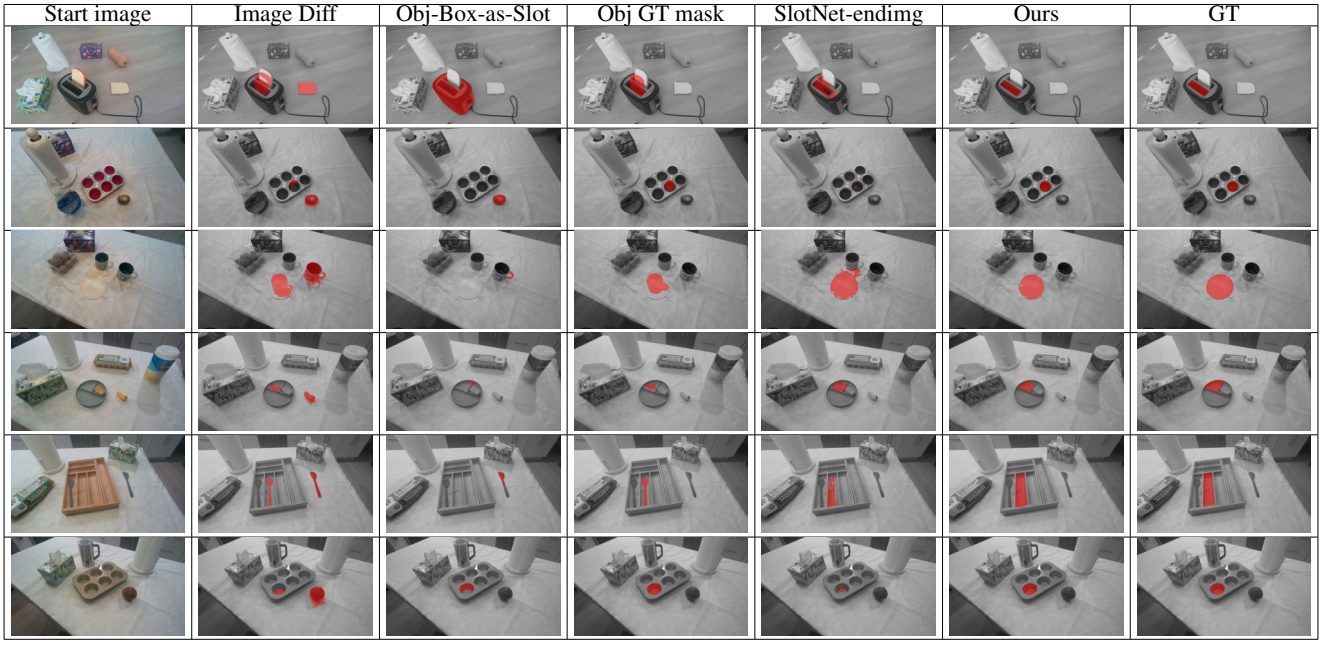
Start image	Image Diff	Obj-Box-as-Slot	Obj GT mask	SlotNet-endimg	Ours	GT
						

Table 12. More comparison results for Slot-Net on seen tasks. We show more examples for comparison results between Slot-Net and other baselines or heuristics.

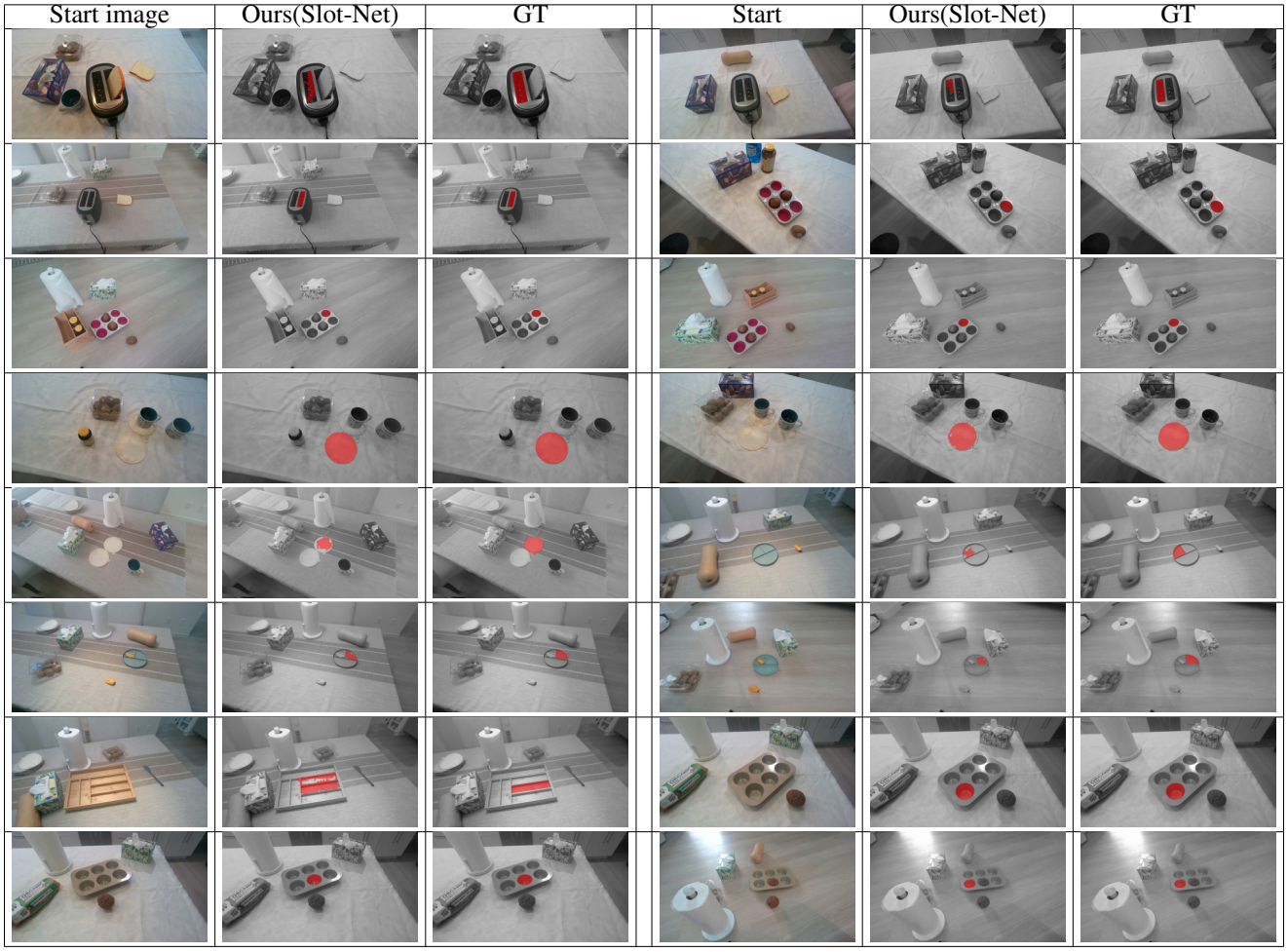
Start image	Ours(Slot-Net)	GT	Start	Ours(Slot-Net)	GT
					

Table 13. Random results for Slot-Net on seen tasks. We show randomly sampled results from Slot-Net on seen tasks.