Supplementary Material for BOP-Ask: Object-Interaction Reasoning for Vision-Language Models

Vineet Bhat[1] Sungsu Kim[1] Valts Blukis[2] Greg Heinrich[2] Prashanth Krishnamurthy[1]
Ramesh Karri[1] Stan Birchfield[2] Farshad Khorrami[1] Jonathan Tremblay[2]

[1]New York University [2]NVIDIA

1. Data Generation Pipeline

The *Benchmark for 6D Object Pose Estimation (BOP)* family of datasets provides training data for 6D object pose estimation and comprises real and simulation images showcasing multiple objects and diverse setups. For example, HOPE, a BOP-based dataset, comprises 28 toy grocery objects captured in 50 scenes from 10 household/office environments. In this section, we describe our framework (Figure 1) for transforming data in the BOP format to a precise robotic dataset for large-scale 2D/3D VLM training.

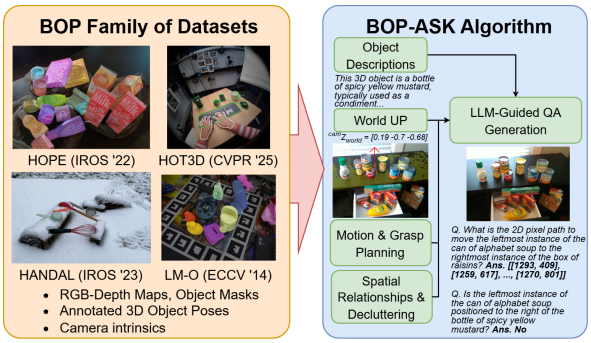

Figure 1. Our proposed data generation framework can transform all the 6D pose annotated RGB-D images within BOP into a robotics ready precise spatio-geometric reasoning benchmark.

Determining Camera Extrinsics. We propose a simple method to retrieve extrinsics from datasets that do not provide it. We first generate 2D segmentation masks from RGB images using the Molmo-72B model with a prompt like "Point to the flat surface where the objects are placed"; this point is passed to SAM 2 to generate 2D segmentation masks. We fit a RANSAC plane estimate to find the dominant planar surface normal leveraging the scene point cloud. Plane orientation ambiguity is resolved by ensuring objects lie above the fitted plane (see Figure 1). Once the world-up direction is determined, the final world rotation matrix is computed using the Rodrigues rotation formula and yaw angle refinement through principal component analysis of the projected planar inliers to align the dominant scene surface edge with the world X axis. Translation estimation places the world origin at the table surface by rotating all depth points and using the median z-coordinate of RANSAC inliers as the table height offset. The camera to world transformation ${}^{cam}T_{world}$ enables consistent world-frame object positioning essential for robotic grasping applications.

Trajectory Generation. Our framework employs the Rapidly-exploring Random Tree (RRT) planner operating in 3D cartesian space to compute collision-free pick-and-place trajectories between object locations. The RRT implementation utilizes a hybrid collision detection system that combines axis-aligned bounding box (AABB) representations of scene objects with dense point cloud-based obstacle avoidance. The planner incorporates a 10% goal bias during random sampling within adaptively computed bounds that extend 20 cm beyond the convex hull of object positions, with vertical search space biased toward regions above the workspace to encourage natural lifting motions. The raw RRT trajectory undergoes post-processing through the Ramer-Douglas-Peucker algorithm for 3D path simplification reducing waypoint density while maintaining trajectory fidelity within a 3cm tolerance threshold. The final 3D trajectory is then projected to 2D image coordinates, this dual representation enables both 3D motion execution and 2D visual verification, with the projected trajectories overlaid on RGB images to verify waypoint markers and directional arrows to indicate motion flow.

Grasp generation. We use multi-modal preprocessing to compute object grasp poses using a strong 3D grasp detection model. The scene point cloud is combined with

Table 1. Sample question-answer pairs from BOP-ASK.

Type	Question	Answer
Object poses	Locate the cuboid corners of the squeezable bottle of mayonnaise, output its bbox coordinates using JSON format. The coordinates should obey the limits of the image, and thus x coordinate should be between (0, 1920) and y coordinate should be between (0, 1080).	ẅbox¨: [[526, 498], [570, 484], [644, 439], [600, 454], [396, 261], [439, 238], [525, 207], [482, 230]]
Grasps	What are the five 2D points in image pixel space that outline the grasp plane for the bottle of tangy BBQ sauce? Your response should be as Grasp center: [], Left finger base: [], Right finger base: [], Left finger tip: [], Right finger tip: []., output its coordinates in XML format ¡points x y¿object¡/points¿. Output nothing else. The coordinates should obey the limits of the image, and thus x coordinate should be between (0, 1280) and y coordinate should be between (0, 720).	<points x=ẅ77ÿ=ẅ349ÿGrasp center</points><points x=ẅ93ÿ=ẅ367ÿLeft finger base</points><points x=ẅ64ÿ=ẅ332ÿRight finger base</points><points x=ẅ704ÿ=ẅ407ÿLeft finger tip</points><points x=ẅ674ÿ=ẅ370ÿRight finger tip</points>
Trajectories	What is the 2D pixel path to move the rightmost instance of the box of raisins to the location of the cylindrical container of grated Parmesan cheese? Your response should be a list of 2D points that show the path from target object to goal object in that order path: []., output its coordinates in XML format ¡points x y¿object¡/points¿. Output nothing else. The coordinates should obey the limits of the image, and thus x coordinate should be between (0, 1920) and y coordinate should be between (0, 1080).	<points x=ï225ÿ=ẅ821ÿpoint1</points><points x=ï076ÿ=ẅ369ÿpoint2</points>
Object rearrangement	Which objects should be moved first to create space for grabbing the box of organic whole wheat spaghetti? Your response should be a list of 2D points which mark the objects you need to move as object markers: []. Output all the object coordinates in JSON format. The coordinates should obey the limits of the image, and thus x coordinate should be between (0, 1920) and y coordinate should be between (0, 1080).	[object markers: [865, 631], [1074, 938], [802, 808]]
Spatial Reasoning	Is the box of microwave popcorn, designed for convenient preparation of popcorn in a microwave, to the left of the cylindrical container of grated Parmesan cheese?	Yes
Relative Depth Perception	Is the cooking spoon with green heads and light wooden handles farther from the camera than the whisk with a black handle and yellow wires? Your response should be a single word: yes or no.	No

annotated semantic segmentation masks to create target object point clouds. We use a dual-sampling strategy: global scene points are sampled for contextual awareness, while object-specific points are independently sampled to ensure adequate representation of target geometry. The pre-computed camera-to-world transformation ${}^{cam}T_{world}$ is used to transform the scene coordinate system which is finally passed through M2T2, a strong 3D grasp detection model with state-of-the-art performance on ACRONYM and various simulation tasks from RLBench and real-world experiments. M2T2 processes the segmented point cloud data to generate 6D grasp poses represented as 4×4 transformation matrices encoding both position and orientation for parallel gripper configurations. The model operates through multiple inference runs with stochastic point sampling to increase grasp diversity and robustness, accumulating predictions across iterations to build a comprehensive grasp candidate set. We extract top-5 candidate grasps for an object, and use camera extrinsic and intrinsic parameters to project the grasp to the image pixel space. The final grasp in 2D is represented with 5 points denoting the end-effector base and tip positions along with the grasp center.

Generating Question-Answer Pairs. For each scene in the HOPE dataset, we now have 6D object poses, grasps and 2D pixel level trajectories to move a pair of objects. We use this information to create a diverse Question-Answer dataset for training large vision-language models for improved spatial reasoning necessary for robotic deployment. BOP-Asкcomprises five broad categories of question types that cover object poses, grasps, 2D trajectories, object rearrangement for de-cluttering, spatial reasoning and relative depth perception. Geometrical awareness questions such as pose and grasp estimation tests whether the model can accurately localize an object or its affordance (specified using a grasp plane) described by a free-form text description. Trajectory prediction tests the model's collision awareness and whether the VLM is able to predict 2D waypoints describing arbitrary motion between two objects. Spatial and depth relationships denote the skill of identifying the relative position of a pair of objects (Example - "Is object A to the left of object B", etc.). We also introduce de-cluttering as an important task within our dataset, where the goal is to determine the neighboring objects around a target object to create some space, useful for grasping in a clutter. Instead of using a fixed template to generate questions as is done previous works, we use an LLM to generate diverse vocabulary in our text prompts. This allows for more expressiveness about objects and their descriptions, see Table 1 for some sample questions from our dataset and their corresponding ground truth answer formats.

Bias Analysis (M2T2/RRT vs. Human). We compared the automated priors in BOP-Asк to human annotations.

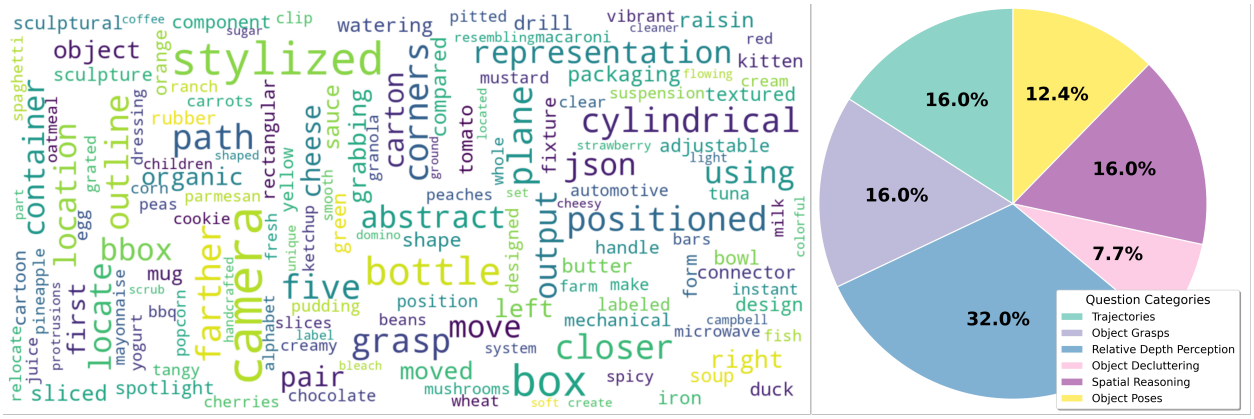

Figure 2. Distribution of free form questions and task types in BOP-Ask.

For normalized distance (where 0 indicates the object centroid and 1 the edges), M2T2 predicts more central grasps (0.56) while humans prefer edges (0.74). M2T2 also shows a bias toward top-down grasps, with a larger angular deviation from the principal axis (58°) than humans (50°). Regarding motion, RRT produces more direct paths (81.0% efficiency) than humans (77.1%), but human trajectories are smoother, with lower curvature (48°) compared to RRT (61°).

Table 2. Test-set statistics comparing quality/diversity across four spatial QA datasets. Best per row is in **bold**.RS: RoboSpatial, BL: Blink, SB: SpatialBench.

Metrics	Ours	RS	BL	SB
number of questions	**852**	350	143	34
avg length (token)	43.1	37.2	8.1	**73.6**
length std-dev	16.5	**32.4**	1.3	25.0
vocab. size	**355**	163	111	227
type-token ratio	0.01	0.01	**0.10**	**0.10**
avg obj per question	10.0	6.2	2.2	**10.9**
object vocab size	**152**	101	66	108

2. Analysis of BOP-Ask-core

In Table 2, we compare against all the questions in RoboSpatial-Home, the spatial questions in BLINK, and the position questions in SpatialBench. To quantitatively assess the diversity and complexity of the generated questions, we compute several linguistic and semantic metrics. We evaluate lexical richness and syntactic variety through vocabulary size, type-token ratio, and the average question length with its standard deviation. Semantic diversity is measured via the entropy of question types, which are automatically classified into categories (e.g., spatial configuration, context, compatibility) using keyword heuristics. Additionally, we analyze referential complexity by computing the size of the object vocabulary, the average number of unique objects referenced per question, and the prevalence of multi-object relational queries—defined as questions that reference at least two distinct objects in the presence of spatial spatial prepositions. Our test split emphasizes compositional diversity: it achieves the largest vocabulary size (230), highest question-type entropy (1.38), and the most objects per question (11.6). We also show question distribution as well as a word point cloud in Figure 2.

Table 3. Robot experiments evaluating the visual grounding, grasping and path planning capabilities of the NVILA-Base model and the NVILA-SFT model finetuned on BOP-Ask.

Method	Grounding	Grasping	Trajectories
NVILA (Base)	24.4	16.7	0
+ BOP-Ask	73.8	74.0	71.4

3. Robot Experiments

To evaluate real-world applicability, we deploy both base and fine-tuned NVILA models on a 7-DoF Franka arm across 15 tabletop tasks involving a mix of seen and unseen objects. Each task follows the template: "Move the {TARGET OBJECT} to the location of the {GOAL OBJECT}," where the VLM predicts object poses, grasp points, and motion trajectories to be executed on the physical robot. Each task is divided into three stages—visual grounding of the target and goal objects, grasp execution, and trajectory-based placement. Visual grounding is considered successful if the predicted cuboid intersects the referred object with an IoU ≥ 50, while grasp and placement success are measured by physical execution within an error of 5cm. Table 3 summarizes quantitative results across both models. Fine-tuning on BOP-Ask yields significant improvements in grounding and grasp accuracy. While the base model fails to predict motion trajectories between object pairs, the fine-tuned variant produces smooth, executable paths for seen pairs and generalizes to unseen ones—for

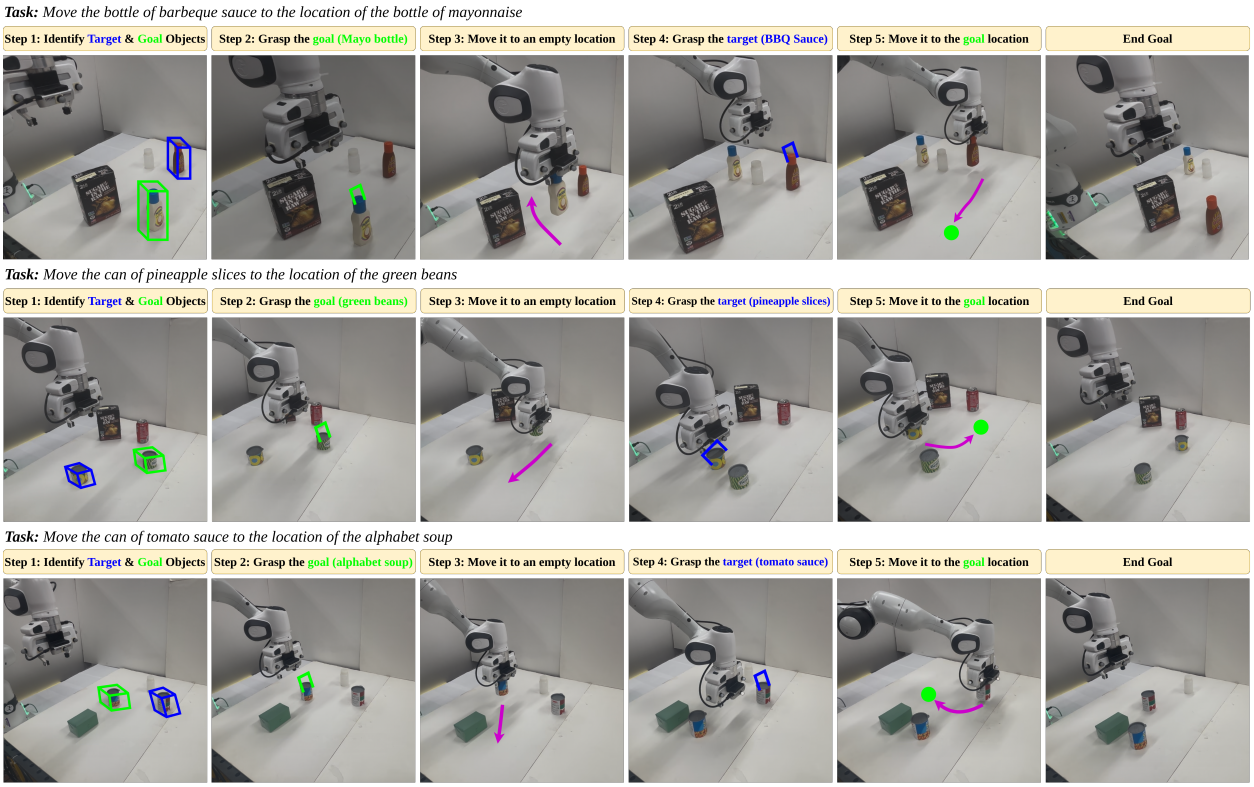

Figure 3. Real world robot experiments with a Franka arm and a ZED2 Stereo camera. VLMs fine-tuned on BOP-Ask can perform tasks such as visual grounding (step 1), grasping (step 2,4) and motion planning (step 3,6).

example, in the task *"Move the tomato sauce can to the location of the green box"*, where the green box was unseen during training. Overall, the fine-tuned model completes 10 of 15 tasks.

The predicted grasp affordances and motion paths are executed using an inverse kinematics controller that generates a smooth SE(3) trajectory through the predicted waypoints. The X and Y coordinates are derived from calibrated camera intrinsics and extrinsics, while Z-values are interpolated from object depth maps to ensure continuous motion between the source and target poses. Since the predicted model provides the positions of the left finger and right fingers of the gripper, we use the center of the grasp, which often lies on the object to get the depth value. This allows us to convert the points to 3D in the world coordinate, and thus compute the transformation matrix for the gripper. We execute the trajectories at a fixed height of 20cm above the surface to prevent collisions. We use a constant force of 5N while grasping. Some examples from our real-robot execution is shown in Figure 3. Failures primarily occur when the model misidentifies objects leading to inaccurate grasps or incomplete motion paths. Nevertheless, the fine-tuned model demonstrates strong generalization and physical reliability, showing that BOP-Ask effectively equips VLMs with transferable spatio-geometric priors for real-world manipulation.