Spot-Compose: A Framework for Open-Vocabulary Object Retrieval and Drawer Manipulation in Point Clouds

Oliver Lemke*, Zuria Bauer, René Zurbrügg, Marc Pollefeys, Francis Engelmann†, and Hermann Blum†

***Abstract*— In recent years, modern techniques in deep learning and large-scale data sets have led to impressive progress in 3D instance segmentation, grasp pose estimation, and robotics. This allows for accurate detection directly in 3D scenes, object- and environment-aware grasp prediction, as well as stable and repeatable robotic manipulation. This work aims to integrate these cutting-edge methods into a comprehensive framework for robotic interaction in human-centric environments. Specifically, we leverage high-resolution point clouds from a commodity scanner for open-vocabulary instance segmentation, alongside grasp pose estimation, to demonstrate dynamic picking of objects and opening of drawers. We show the performance and robustness of our model in two sets of real-world experiments evaluating dynamic object retrieval and drawer opening, reporting a 51% and 82% success rate respectively. To encourage further development of and experimentation with our framework, we make the code and videos available at https://spot-compose.github.io/.**

I. INTRODUCTION

One of the pinnacle achievements in the field of robotics is to develop systems capable of understanding and navigating spaces designed for humans. This task poses a significant challenge due to the high variability and complexity of human-centric environments, requiring good semantic understanding and precise manipulation. Nonetheless, achieving this goal is considered a significant milestone in technological evolution, bringing with it strong efficiency and accessibility improvements. Recent works in high-resolution 3D scanning technologies, sophisticated perception models, and intricate manipulation algorithms have collectively facilitated a leap in robotic abilities, enabling more nuanced and effective interactions within our daily spaces. This study introduces a framework that aims to utilize these advancements, leveraging modern models for instance segmentation and grasp pose estimation on top of the potent Boston Dynamics Spot robot. Key contributions of this paper include:

- Introduction of a modular framework on top of the Spot SDK, providing a flexible platform for the integration of cutting-edge machine perception techniques.
- The framework uses advanced models for perception and grasp estimation, enabling versatile interactions with diverse objects in human-centric environments.
- We demonstrate the practical applicability of our framework through real-world experiments, including dynamic object retrieval and drawer manipulation tasks.

All authors are with ETH Zürich.
*Corresponding Author: Oliver Lemke <olemke@ethz.ch>
†equal advising, This research was partially supported by the ETH AI Center and ETH Zürich Career Seed Award

Fig. 1. **Overview of the Spot-Compose pipeline.** Given a previously acquired point cloud, we segment the scene and localize the wanted object via a natural language query. For object retrieval (top), we isolate the object to determine the most effective grasp. For drawer manipulation (bottom) we use the cabinet position to point our camera for 2D drawer detection.

II. RELATED WORK

3D instance segmentation. Given point cloud inputs, 3D instance segmentation assigns each point a class label, distinguishing between different instances [1]–[8]. Mask3D [3] directly predicts instance masks from point clouds, learning instance queries through iterative attention to multi-scale features. Recently, much research has been done in the field of open-set segmentation. [9]–[11]. Takmaz et al. [9] expand on Mask3D and leverage class-agnostic 3D instance masks and multi-view fusion of CLIP-based image embeddings to segment objects. In this work, we rely on OpenMask3D for the localization of based on natural language queries.

Grasp pose estimation in point clouds. Given an object, robotic grasping determines the most effective ways for robotic two-finger grippers to grasp objects in various situations [12]–[17]. Early work by Ten Pas et al. [18] and others [19], [20] proposed sampling-based grasp estimation, usually with the consequence of long inference times. More recently, approaches have experimented with end-to-end learning to address this issue [21], [22]. With AnyGrasp [23], Fang et al. predict two-finger grasps directly on 3D point cloud representations. It utilizes a dense supervision strategy combining real perception and analytic labels in the spatial-temporal domain, including awareness of objects' center-of-mass for improved grasp stability. We utilize AnyGrasp due to its high performance and built-in environment awareness, which simplifies grasp estimation significantly.

Robot task planning. Robot task planning refers to the process of creating and organizing a series of actions for a robot to achieve specific goals [14], [24]–[26]. In "*ASC: Adaptive Skill Coordination for Robotic Mobile Manipulation*" [27] presents a method for performing long-horizon tasks like mobile pick-and-place using three key components:

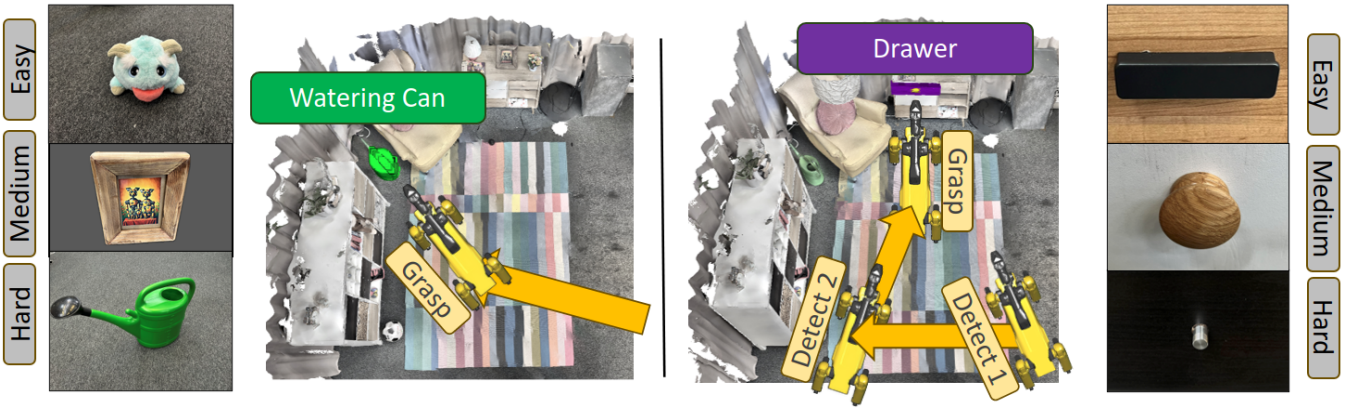

Fig. 2. **Adaptive grasping and drawer interaction pipeline.** On the left, we illustrate the grasping sequence initiated by the successful localization of the watering can through 3D instance segmentation. Following this, an optimal robot positioning is computed by the navigation planner and the object is grasped. The right side of the figure details the drawer detection and manipulation process. Multiple images are captured for robust detection. Subsequently, the robot is maneuvered into position to facilitate drawer opening. This dual-phase approach demonstrates the integration of object detection, navigation planning, and execution within a dynamic scene. On the respective sides we illustrate example objects and handles in various levels of difficulty.

a library of basic visuomotor skills, a skill coordination policy, and a corrective policy for adapting skills in novel situations. *OK-Robot* [28] presents an Open Knowledge-based robotics framework integrating Vision-Language Models with navigation and grasping primitives for efficient pick-and-drop operations in home environments. In contrast to *OK-Robot*, we employ modern segmentation techniques and directly rely on the 3D scene representation. We further demonstrate interaction with articulated objects.

III. TECHNICAL COMPONENTS AND METHOD

Our objective is twofold: firstly, to enable the picking of objects across a wide range of shapes and sizes; and secondly, to facilitate access to concealed spaces, defined as areas that become accessible only by manipulating elements such as drawers or doors, and are otherwise inaccessible or hidden from view. To do this, we require a variety of capabilities. These are (A) semantic point cloud segmentation to identify objects that can be interacted with, (B) point cloud-dependent grasp pose estimation, (C) adaptive navigation, and (D) dynamic drawer detection and subsequent estimation of the axis of motion. Finally, in (E) we will provide a brief overview of additional functionalities that can be readily implemented with the aforementioned skills.

We utilize Boston Dynamics Spot [29].The framework is built upon the associated SDK [30]. We wish to highlight the modular nature of our approach, which would allow for the integration of more advanced models as the field progresses.

A. Object localization via 3D instance segmentation

One integral element of our approach is an open-vocabulary instance segmentation model, which we employ to map our environment and then pinpoint any object of interest specified through a natural language query.

Acquisition of 3D environmental data. We assume the availability of a pre-scanned 3D representation of the environment. These representations can be obtained through various methods, including the use of modern smartphones equipped with LiDAR scanners and an associated scanning apps. We conduct all of our environment scans using an iPhone 13 Pro Max [31] using the *3D Scanner App* [32].

Open-vocabulary 3D Instance Segmentation. Contrary to previous approaches, recent developments in 3D instance segmentation enable the localization of any specified object directly from the 3D point cloud data. This represents a significant advancement over traditional 2D instance segmentation by facilitating the extraction of a detailed, object-specific mask at the point level. We propose, that utilizing this mask should aid in subsequent planning and picking processes, allowing for more precise object manipulation.

In this paper, we deploy OpenMask3D [9], which on top of instance segmentation, allows querying of segmentation masks based on natural language input. This enhances the robot by allowing intuitive command input in natural language, broadening accessibility and user-friendliness.

B. Adaptive grasping

The crucial stage in robotic object manipulation lies in the grasp pose estimation. We implement the AnyGrasp system [23] for this purpose. Fang et al. describe AnyGrasp as a "unified system for fast, accurate, 7-DoF and temporally-continuous grasp pose detection, using a parallel gripper". This method is notably aware of the object's center of gravity and and its environment, filtering grasps that would be rendered impossible by surrounding obstacles.

Inference. By default, AnyGrasp is tuned to identify grasp poses based on the frontal view of the given point cloud. To expand our grasp detection capabilities and encompass all potential grasp poses, we run multiple detection iterations, each with distinct initial rotations of the object. For each perspective, we obtain the top k grasps for post-processing. Subsequently, we filter the poses based on a set of criteria, namely (1) have a positive confidence score and (2) be located on the object point cloud.

C. Adaptive navigation and joint optimization

Before grasping, we need to decide where to position the robot, such that it has a good vantage point. For this, we

sample a set of positions radiating outward from the grasp item. For each position, we check whether it (1) lies within the scene and (2) has a direct line of sight to the object. Over the remaining body and grasp poses we now decide on the best combination via joint optimization.

$$s = s_{\text{grasp}} + \lambda_{\text{body}} \cdot s_{\text{body}} + \lambda_{\text{align}} \cdot s_{\text{align}} \tag{1}$$

where, s_{grasp} is the confidence score of the grasp, while

$$s_{\text{body}} = d_{\text{obstacles}} - \lambda_{\text{item}} d_{\text{item}} \tag{2}$$

defines a metric on the body pose, such that $d_{\text{obstacles}}$ denotes the distance to the nearest non-grasp item object in the environment, while d_{item} denotes the distance to the grasp item. By adjusting λ_{item}, this metric strikes a good trade-off between avoiding collisions with the environment, while staying close enough to the object to allow for easy grasping.

Finally, s_{align} represents

$$s_{\text{align}} = \tanh\Big(T \cdot \frac{\vec{x}_{\text{rt}}}{||\vec{x}_{\text{rt}}||} \cdot \frac{\vec{x}_{\text{g}}}{||\vec{x}_{\text{g}}||}\Big), \tag{3}$$

i.e. the dot product between the normalized vectors $\vec{x}_{\text{rt}}$, pointing from robot to target, and $\vec{x}_{\text{g}}$, pointing in the grasp direction, scaled by some temperature T. This term encourages body and grasp positions to be aligned with each other. We add a tanh term to allow for slight misalignments .

We experimentally set $\lambda_{\text{body}} = 0.01$, $\lambda_{\text{align}} = 0.02$, $\lambda_{\text{item}} = 0.5$, and $T = 1$. Note that not all scores s have the same initial magnitude. This configuration enables the model to focus on the following aspects in order of importance: (1) finding the best grasp, (2) choosing a body position best aligned with that grasp, and (3) choosing a body position distanced from any obstacles. We have found these parameters to work well for our environment, however encourage experimentation with these values for new locations.

D. Dynamic drawer detection and motion estimation

The second key capability we introduce involves the dynamic detection and manipulation of drawers, comprising three subtasks: drawer and handle detection, estimation of the axis of motion, and actual grasp planning. This skill is significant, enabling the robot to access spaces that are typically concealed, such as when searching for lost objects.

Drawer and handle detection. To initially identify all cabinets within the environment, we apply the method outlined in Section III-A. However, for the detection of individual drawers and associated handles, we find that 3D instance segmentation falls short. The lack of distinct geometrical features in drawers and the insufficient resolution of our point cloud render it ineffective for accurately segmenting handles. Instead, we opt to leverage the RGBD camera embedded in the robot's gripper to capture images of the cabinet and localize handles in the RGB image using 2D object detection techniques. The final handle pose is computed by backprojecting into 3D using the associated depth value. For 2D handle detection, we finetune a YOLOv8 model [33] on the DoorDetect dataset [34]. To increase the robustness of our detection, we capture multiple images, utilizing Iterative Farthest Point Sampling (see Fig. 2).

Axis of motion estimation. A crucial aspect of the drawer pulling action is determining the axis of motion, which dictates the direction in which the drawer must be opened. We find that this motion typically aligns with the normal of the drawer front. To identify 3D points related to the drawer front, one might consider selecting points within a specific constant offset from the handle detection. However, this approach fails to generalize effectively due to the variable distance to the camera and the drawer's shape. Instead, we leverage the fact that our model successfully identifies both drawers and handles, and employ Hungarian Matching [35] to pair the two detections. The matching cost is defined as

$$C_{\text{Hungarian}}(i,j) = -\big(\kappa \cdot \text{IoA}(h_i, d_j) + \text{Conf}(d_j)\big), \tag{4}$$

where h_i and d_j denote the i^{th} handle and the j^{th} drawer instance, respectively. Here, $\text{Conf}(d_j)$ is the confidence score of the drawer prediction, and $\text{IoA}(h_i, d_j)$ ("Intersection over Area"), represents the proportion of the handle's bounding box that overlaps with the drawer's bounding box,

$$\text{IoA}(h_i, d_j) = \frac{\text{Intersection}(h_i, d_j)}{\text{Area}(h_i)}. \tag{5}$$

This cost function is prioritizes handle bounding boxes that fall within the drawer bounding boxes, using the confidence of the drawer detection as a secondary criterion. The constant $\kappa = 10$ balances the significance of these two factors.

To estimate the axis of motion, we project 3D depth capture into the image, select all points that lie within the drawer's, but not the handle's, bounding box, and employ RanSAC [36] plane estimation [37].

Refinement and impedance-based pulling. After estimating all the necessary components for the pulling motion, the remaining step is execution. We position the robot at a predetermined distance in front of the cabinet, aligning it parallel to the axis of motion. Given that our predictions are unlikely to be perfectly accurate, we implement two additional refinements. The first involves capturing another image in front of the drawer to refine both the coordinate and the axis of motion from a closer proximity. Secondly, an impedance-based pulling motion allows the robot some flexibility in directions orthogonal to the axis of motion.

E. Expansion of capabilities

With the foundational framework established, extending the system to incorporate additional functionalities becomes a relatively straightforward task.

Playing fetch. Combining an appropriate detection model, such as VitPose [38] with the object retrieval function described in Section III-B, we can deliver objects to humans.

Search. The ability to open and close drawers is only one building block. When this functionality is integrated with the camera located on the end-effector, alongside a contemporary open-vocabulary object detection approach (such as OWLv2 [39]), it becomes straightforward to develop a mobile search robot capable of exploring concealed areas.

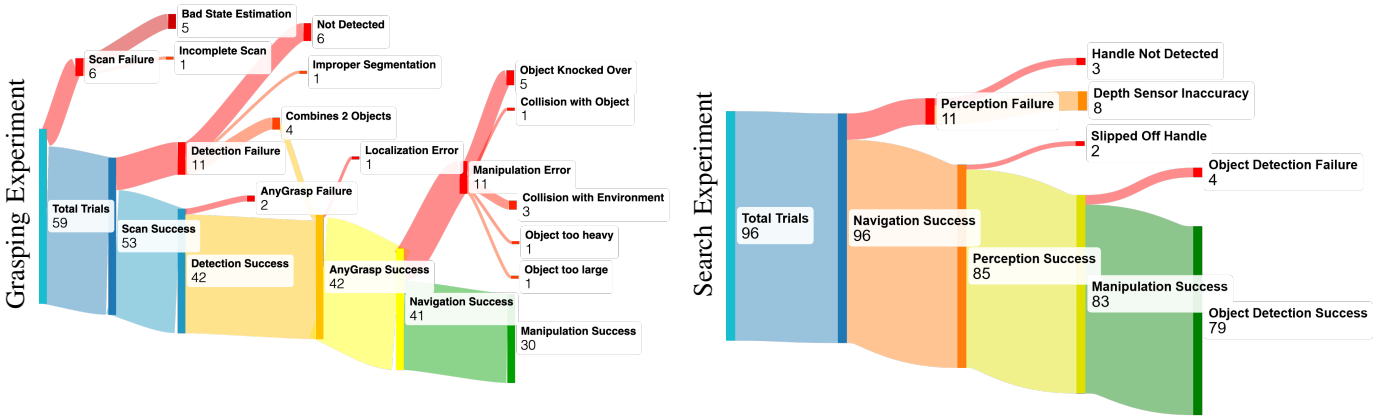

Fig. 3. **Grasping experiment results.** To evaluate the grasping capability of our framework, we conduct 59 trial runs across six different scenes and with 13 distinct objects. The test includes items and placements of varying difficulty. We observed an overall success rate of 51%, with the highest failure rate occurring in detection and manipulation. **Search experiment results.** For this evaluation, we conduct 16 runs with six individual drawers. Moreover, we explore combinations of 15 handles and 19 objects, experimenting with various pairings. We observe a 82% success rate, with the majority of failure cases being connected to bad perception, especially an inaccurate Time-of-Flight depth sensor.

IV. EXPERIMENTS

We assess our framework on two separate experiments: (1) grasping, and (2) search. The experiments are designed to evaluate the high level functions of our model, while highlighting multiple the facets of the framework.

Grasping. To test the dynamic grasping ability of our model, we evaluate it on a set of 6 different scenes with a varying difficulty levels of both object graspability and placement. For an example of different object difficulties, please refer to Fig. 2. The level of difficulty was decided based on deformability of the object and amount of possible grasps. All objects must be placed in a free-standing manner. Each run is repeated once, to test for robustness and our results are illustrated in Fig. 3 and Table I. We observe an overall success rate of 51%, where the ease of grasping a respective object is the main predictor of a successful overall grasp. While we are able to manipulate even very difficult objects, such as watering cans, and navigate difficult locations, robustness in these cases tends to suffer.

TABLE I

SUCCESS RATE BREAKDOWN: OBJECT VS. PLACEMENT DIFFICULTY. WE ILLUSTRATE THE MANIPULATION SUCCESS RATE ACROSS DIFFERENT DIFFICULTY LEVELS OF OBJECTS AND PLACEMENTS, OBSERVING AN INVERSE CORRELATION WITH HIGHER DIFFICULTY.

Object \ Placement	Easy	Medium	Hard
Easy	75%	100%	100%
Medium	90%	75%	50%
Hard	83%	25%	40%

Search. We aim to evaluate the drawer localization and 2D detection capabilities of our approach. The setup is as follows: We initialize the robot with the position of the two cabinets derived via instance segmentation of the environment, as well as a natural language search term. Subsequently, the robot is tasked with detecting all drawers within the cabinets, opening them to inspect their contents, and identifying any items within. Should the sought-after object be found among the contents, the robot must record the drawer in which the object was located in 3D space.

The results are illustrated in Fig 3. We report an 82% success rate, with the majority of failure cases being related to inaccurate depth sensing. By capturing multiple perspectives of a given cabinet, we are able to robustly detect even small, color-matching, or oddly-shaped handles in the vast majority of cases. However, this is at the cost of additional inference time, representing a trade-off for real-work applications.

Inference times. We list the expected inference times for localization (0.221s), one-time 3D instance segmentation (271s) grasp pose estimation (13.7s), navigation planning (24.0s), joint optimization (0.3ms), drawer detection (0.84s) and zero-shot object detection (2.85s). All times except for localization and optimization include latency, as they are executed on an external NVIDIA RTX 4090 GPU [40].

V. CONCLUSION

In this paper, we have presented a comprehensive framework to efficiently build new functionality for the Spot robot, increasing accessibility for researchers beyond the field of robotics. We utilize it to enhance the capabilities of robots to interact within human-centric environments, specifically through dynamic object retrieval and drawer manipulation tasks. Our work leverages state-of-the-art techniques in 3D instance segmentation, grasp pose estimation, and object detection to enable mobile manipulation in real-world settings.

The experiments underline the practical applicability of our approach, showcasing the potential for robots to perform complex tasks in environments designed for humans. Finally, the modular nature of our framework allows for the seamless integration of future advancements in perception and manipulation technologies.

Current limitations of our approach include long latencies arising from 3D instance segmentation and 360° grasp pose estimation. In future work we plan to focus on efficient grasp trajectory planning and more sophisticated joint optimization to enhance the robustness of our model. We encourage everybody to build on top of our framework, enabling more streamlined implementation and sophisticated actions.

REFERENCES

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[2] K. Genova, X. Yin, A. Kundu, C. Pantofaru, F. Cole, A. Sud, B. Brewington, B. Shucker, and T. Funkhouser, "Learning 3d semantic segmentation with only 2d image supervision," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 361–372.

[3] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3d: Mask transformer for 3d semantic instance segmentation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8216–8223.

[4] J. Sun, C. Qing, J. Tan, and X. Xu, "Superpoint transformer for 3d scene instance segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 2393–2401.

[5] H. Lei, N. Akhtar, and A. Mian, "Spherical kernel for efficient graph convolution on 3d point clouds," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3664–3680, 2020.

[6] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer, 2016, pp. 424–432.

[7] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, 2018.

[8] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4096–4105.

[9] A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann, "Openmask3d: Open-vocabulary 3d instance segmentation," *arXiv preprint arXiv:2306.13631*, 2023.

[10] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. D. Mello, "Open-vocabulary panoptic segmentation with text-to-image diffusion models," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2955–2966, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257405338

[11] X. Chen, S. Li, S. N. Lim, A. Torralba, and H. Zhao, "Open-vocabulary panoptic segmentation with embedding modulation," *ArXiv*, vol. abs/2303.11324, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257632184

[12] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9626–9633.

[13] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *The International Journal of Robotics Research*, vol. 41, no. 7, pp. 690–705, 2022.

[14] A. Hundt, V. Jain, C.-H. Lin, C. Paxton, and G. D. Hager, "The costar block stacking dataset: Learning with workspace constraints," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1797–1804.

[15] K. Karunratanakul, J. Yang, Y. Zhang, M. J. Black, K. Muandet, and S. Tang, "Grasping field: Learning implicit representations for human grasps," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 333–344.

[16] S. Ainetter and F. Fraundorfer, "End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 452–13 458.

[17] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[18] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[19] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2901–2910.

[20] H. Duan, P. Wang, Y. Huang, G. Xu, W. Wei, and X. Shen, "Robotics dexterous grasping: The methods based on point cloud and deep learning," *Frontiers in Neurorobotics*, vol. 15, p. 658280, 2021.

[21] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.

[22] D.-C. Hoang, J. A. Stork, and T. Stoyanov, "Voting and attention-based pose relation learning for object pose estimation from 3d point clouds," *IEEE Robotics and Automation Letters*, vol. 7, pp. 8980–8987, 2022.

[23] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, 2023.

[24] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *arXiv preprint arXiv:2002.06289*, 2020.

[25] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, S. Y. Min, K. Shah, C. Paxton, S. Gupta, D. Batra *et al.*, "Goat: Go to any thing," *arXiv preprint arXiv:2311.06430*, 2023.

[26] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[27] N. Yokoyama, A. W. Clegg, E. Undersander, S. Ha, D. Batra, and A. Rai, "Adaptive skill coordination for robotic mobile manipulation," *arXiv preprint arXiv:2304.00410*, 2023.

[28] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "Ok-robot: What really matters in integrating open-knowledge models for robotics," *arXiv preprint arXiv:2401.12202*, 2024.

[29] Boston Dynamics, "Spot: The agile mobile robot," https://bostondynamics.com/products/spot/, 2023, accessed: 2024-03-10.

[30] ——, "Spot sdk documentation," https://dev.bostondynamics.com/, 2024, accessed: 2024-03-10.

[31] Apple Inc., "iphone 13 pro max - technical specifications," https://support.apple.com/kb/SP848?locale=en_US, 2023, accessed: 2023-12-26.

[32] Laan Labs, "3d scanner app," https://apps.apple.com/us/app/3d-scanner-app/id1419913995, 2023, accessed: 2023-12-26.

[33] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[34] M. Arduengo, C. Torras, and L. Sentis, "Robust and adaptive door operation with a mobile robot," *Intelligent Service Robotics*, May 2021. [Online]. Available: http://dx.doi.org/10.1007/s11370-021-00366-7

[35] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[36] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[37] Open3D, "Plane segmentation - open3d documentation," https://www.open3d.org/docs/latest/tutorial/Basic/pointcloud.html#Plane-segmentation, 2023, accessed: 2024-03-02.

[38] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," in *Advances in Neural Information Processing Systems*, 2022.

[39] M. Minderer, A. Gritsenko, and N. Houlsby, "Scaling open-vocabulary object detection," 2023.

[40] NVIDIA, "Geforce rtx 4090 graphics cards for gaming — nvidia," https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4090/, 2024, accessed: 2024-03-11.