TIGR: a Mixture-of-Foundation-Model-Experts for 3D-informed Task-Aware Grasping

Philipp G. Knestel[1], Andrea Sipos[1], Ashok Meenakshi Sundaram[1], Freek Stulp[1], Samuel Bustamante[1]

***Abstract*—Task-aware grasping of unknown objects involves interpreting natural-language instructions to identify the correct target object within a scene, localizing the task-relevant part, and generating collision-aware grasps from both observable and occluded approach directions. We present TIGR (Task-aware Intelligent Grasping for dexterous Robots), a pipeline for task-aware grasping. It combines 3D reconstruction with a mixture of foundation-model experts whose predictions across rendered viewpoints of the reconstruction are fused into a unified 3D representation. This representation lets TIGR recover functional parts beyond the camera's field of view and generate grasps from all collision free approach directions. We evaluate on a real-robot benchmark of 20 everyday objects, each paired with 2–3 task formulations and tested across 5 viewpoints, totaling 809 trials across TIGR, GraspMolmo, and ShapeGrasp. TIGR outperformed GraspMolmo and ShapeGrasp, achieving 97.8% object-identification accuracy, 82.6% task-relevant segmentation success, and 52.6% grasp success, with at least one successful grasp on every object in the set.**

[1] German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchener Str. 20, 82234 Weßling, Germany. This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy by means of the project SMiLE-AI and by the German Research Foundation (DFG) within the Collaborative Research Center EASE under Grant SFB 1320

I. Introduction

Recent advances in deep learning and foundation models have raised expectations for robots to solve open-vocabulary manipulation tasks zero-shot in unknown environments. One milestone is *task-aware grasping*, a paradigm shown in Fig. 1 in which the robot (i) interprets the task from open-vocabulary, (ii) identifies the correct target object in the scene, (iii) localizes the task-relevant parts of that object, (iv) decides a part to grasp, and (v) generates a physically executable grasp on that part.

Recent task-aware grasping methods such as GraspMolmo [1], ShapeGrasp [2], UAD [3], Task-aware grasping [4] and AffordGrasp [5] operate exclusively within the camera's RGB-D field of view. This has two consequences: first, segmentation and grasp generation are restricted to the visible parts of the objects, even when a stable task-relevant grasp would require

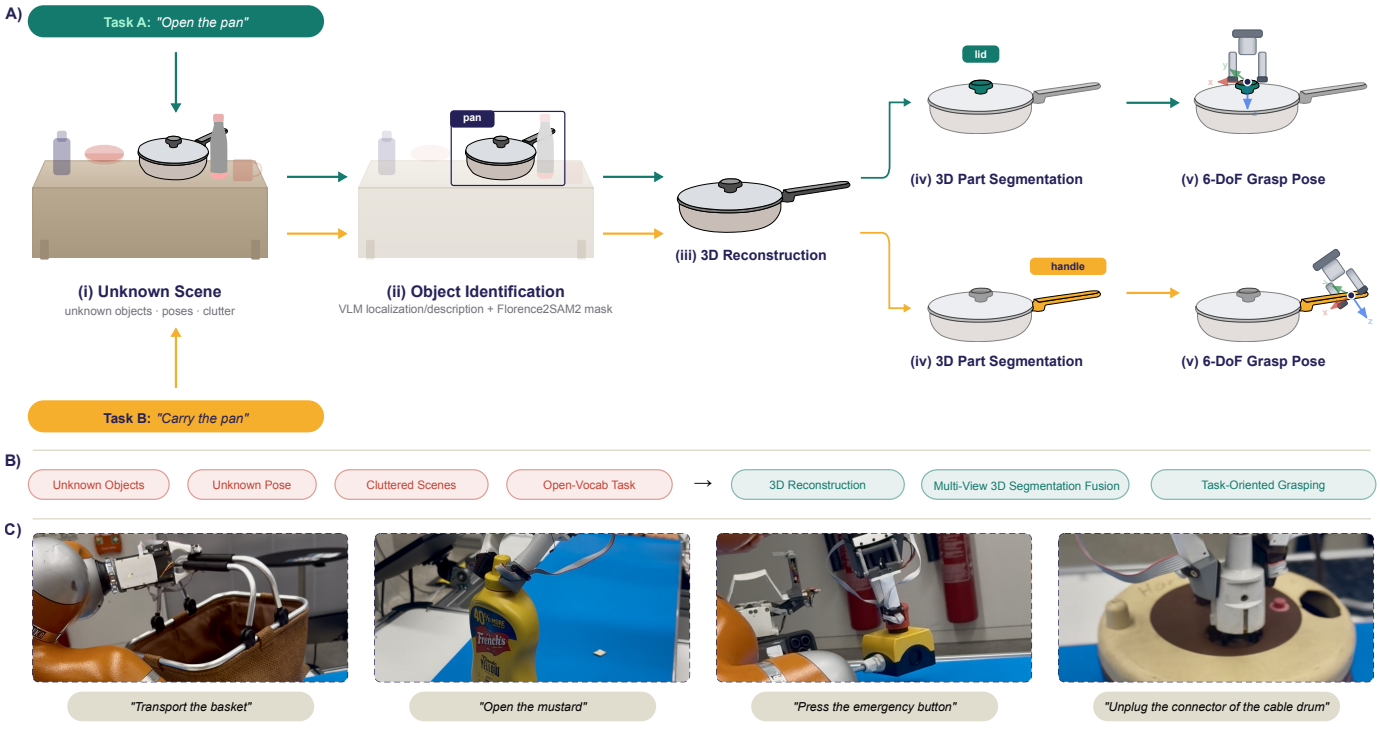

Fig. 1. TIGR pipeline overview. From an RGB-D observation and a natural-language task, TIGR reconstructs the target object, renders it from multiple viewpoints, aggregates part predictions from complementary foundation-model experts into a 3D task-relevant region, and plans collision-aware grasps restricted to that region.

access to a more hidden geometry, for instance, the full handle of a mug or broom. Second, these methods rely on a single foundation model to localize the task-relevant part and inherit its dominant failure modes. Vision Language Model (VLM)-based pointing models hallucinate, open-vocabulary maskers can segment the full object instead of the requested part, and any single Large Language Model (LLM) can occasionally produce incorrect outputs with no redundancy to catch them.

Our approach: TIGR reconstructs the target object in 3D, delegates per-viewpoint part localization to a *mixture of complementary foundation-model experts* (frozen), and fuses their output into a single 3D task-relevant region on which grasps are planned.

Contributions:

1) A mixture-of-foundation-model-experts formulation for task-relevant part localization.
2) A 3D fusion strategy that lifts per-viewpoint 2D expert output into 3D via Point-SAM [6] and aggregates them with a STAPLE [7] variant, making the final task-relevant region robust to individual expert mispredictions on any single view.
3) A real-robot evaluation across 20 objects, 2–3 task formulations, and 5 viewpoints (809 trials across TIGR and two baselines), demonstrating that multi-view fusion over reconstructed geometry enables reliable task-relevant grasping across explicit, functional, and abstract task phrasings.

II. Method

As shown in Fig. 1A, TIGR takes a single RGB-D observation and a natural-language task description and outputs collision-aware grasps on the task-relevant part. The pipeline has five stages:

Scene understanding: Qwen3-VL [8], [9] parses the task and identifies the target object in the scene by name (Fig. 1A(i)). It then enumerates all visible parts of the target object and reasons about which part is most suitable to grasp in order to fulfill the given task, including a rationale for its recommendation.

Object reconstruction: The object name returned by the VLM is passed to Florence-2 [10] to produce a 2D segmentation mask of the target object (Fig. 1A(ii)). This mask, together with the RGB-D image, is fed to SAM3Dobjects [11], which reconstructs the object as a full 3D mesh and estimates its pose in the scene. To refine the pose and scale, Any6D [12] is additionally queried, and the better of the two pose estimates is selected via a 2D Intersection over Union (IoU) comparison. The resulting mesh is then rendered virtually from a set of viewpoints, enabling TIGR to reason about object parts that are partially occluded in the original camera view.

Mixture-of-experts part localization: For each rendered viewpoint, two complementary experts predict the task-relevant region (Fig. 1A(iii)): a pointing VLM (Moondream3 [13]), and the masking model Florence2+SAM2 [10], [14]. Crucially, Florence2+SAM2 is equipped with a guardrail that suppresses output when the target part is not visible in a given view, rather than returning a spurious full-object mask.

TABLE I
Overview of models used in TIGR.

Model	Usage
Qwen3-VL [8], [9]	Object identification, grasp hint generation, object part identification, semantic reasoning, and selection of the object part to be grasped.
Florence2+SAM2 [10], [14]	Object mask generation in the scene and masking of the object part of interest.
SAM-3DObjects [11]	Object reconstruction and 6D pose estimation.
Any6D [12]	Refinement and rescaling of the 6D object pose estimate.
Moondream3 (Preview) [13]	Pointing to the object part of interest.
Point-SAM [6]	3D point cloud segmentation.

3D part segmentation: The 2D predictions are lifted into 3D by raycasting, followed by the 3D point cloud segmentation model Point-SAM [6], producing multiple candidate 3D segmentations (Fig. 1A(iv)). These candidates are fused with a STAPLE-based [7] scheme that distinguishes consensus cases (experts agree) from coverage cases (experts see different fractions of the part).

Grasp generation and selection: Candidate grasps are generated on the fused segmented point cloud with an antipodal grasp sampler [15], [16] and ranked by a combination of grasp-wrench-space force-closure [17], center-of-mass proximity, and finger-pad overlap with the segmented region (Fig. 1A(v)). For each ranked grasp, a set of approach directions is sampled around the grasp axis and scored for reachability and top-down preference. Each candidate approach is then collision-checked against the reconstructed object and environment mesh at both the pre-grasp and the final grasp pose, and the first feasible grasp is executed.

Throughout the five-stage pipeline, the guardrails catch the dominant failure modes of the individual foundation models. An IoU-based check rejects full-object masks, a back-projected-IoU check selects the better of two pose estimates, and the STAPLE3D fusion suppresses single-view mispredictions when other views disagree.

III. Experiments

A. Experimental setup

- **Robot:** KUKA LBR 4+.
- **Vision:** Azure Kinect DK (top view), single picture.
- **Gripper:** Linear closing gripper with single-plane motion, actuated by a variable stiffness actuator [18].
- **Local GPUs:** A GPU server on premises (using an NVIDIA Hopper architecture) provided the Qwen model. All other models were served on a 2x NVIDIA RTX 6000 Ada (49GB) GPU computer.

See Fig. 1C for pictures of the robot and the experimental setup. We used only locally available models exclusively, and

TABLE II
EXPERIMENT PROTOCOL

Task	Example task prompt	Views
1	"carry the bottle, the cap is not pressed tight"	5
2	"open the bottle, it is already being held"	5
3	"give me something to drink"	5
# Objects		20
Executed trials (TIGR / GraspMolmo / ShapeGrasp)		270 / 269 / 270

the specific instances are listed in Table I. We highlight that all these models were frozen, i.e., we did not perform any fine-tuning on our data.

B. Trials

We evaluate TIGR, GraspMolmo [1] and ShapeGrasp [2] on a real-robot grasping benchmark comprising 20 everyday objects across up to three task prompt formulations, five camera viewpoints each. In total we conducted 809 grasping trials (TIGR(270); GraspMolmo(269); ShapeGrasp(270)), see Table II.

Unlike evaluations based on standardized object benchmarks such as YCB, our object set is intentionally composed of uncurated, off-the-shelf household and domestic items collected from everyday environments, including IKEA. The object set spans a wide range of geometries, sizes, and materials, from small household objects (ketchup bottle, mug) to large articulated objects (broom, umbrella, cable drum). The task description ranges from explicit instructions ("carry the mug") through functional descriptions ("use the broom") to abstract requests ("it is dusty, give me something to tidy up"), see Table II for more examples. We note that for some objects we only conducted only 2 tasks instead of 3 due to limited functional prompts.

Procedure: Each object was placed individually on an uncluttered tabletop with a fixed RGB-D camera. No background covering was applied, leaving the lab environment visible in the camera's field of view and introducing natural visual noise, though no clutter was present in the immediate vicinity of the object. Across the five viewpoints for each task prompt, the object's position and orientation were varied to expose different visible surfaces and placements. Each method received the same natural-language task prompt per viewpoint.

C. Evaluation

Each trial is scored by the experimenter along three independent binary criteria: (1) whether the correct target object is identified, (2) whether the task-relevant object part is segmented or pointed at (GraspMolmo [1]) in a manner that could solve the task, and (3) grasp success. A grasp was deemed successful if the robot grasped the object at the task-relevant part, lifted it 10 cm, held it for 5 seconds, and returned it to the tabletop without dropping it. We score them independently because their decoupling captures meaningful

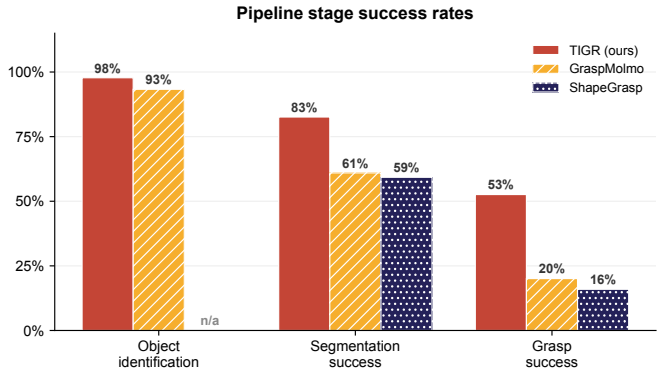

Fig. 2. Stage-wise success rates for TIGR (n=270), GraspMolmo (n=269), and ShapeGrasp (n=270) across all trials. Object identification is not defined for ShapeGrasp.

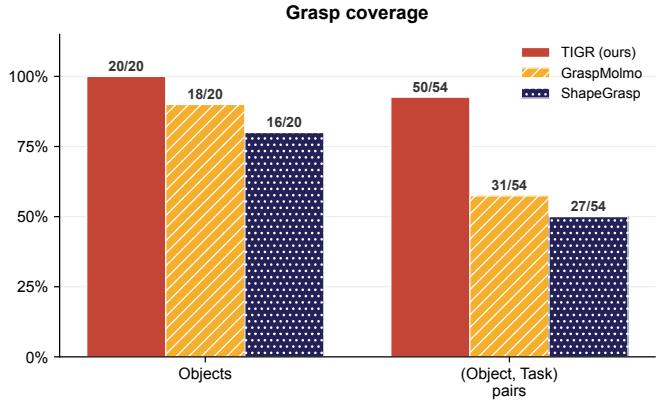

Fig. 3. Grasp coverage across pipelines: proportion of objects and (object, task) pairs with at least one successful grasp.

system behavior: a stable grasp on the wrong part is a grasp failure even if the gripper closes, and a successful grasp on the correct part remains a grasp success even if the segmentation over-covers the object.

D. Results and Discussion

1) Overall success rates: As shown in Fig. 2, across all 270 trials, TIGR correctly identifies the task-relevant object in 97.8% of cases and produces a correct task-aware segmentation in 82.6% of cases. The executed grasp succeeds in 52.6% of trials overall. The high identification and segmentation rates show that TIGR's VLM-guided scene analysis and multi-view part segmentation reliably translate the task description into a concrete grasp target, even for abstract task formulations. The 30% drop between segmentation and grasp success reflects the difficulty of estimating an accurate 6D pose on the reconstructed mesh and selecting a suitable grasp frame from the candidate set. Compared to the single-view baselines GraspMolmo (20%) and ShapeGrasp (16%), TIGR more than doubles the overall grasp success rate, reflecting the benefit of reasoning on a complete 3D reconstruction rather than on a single camera view.

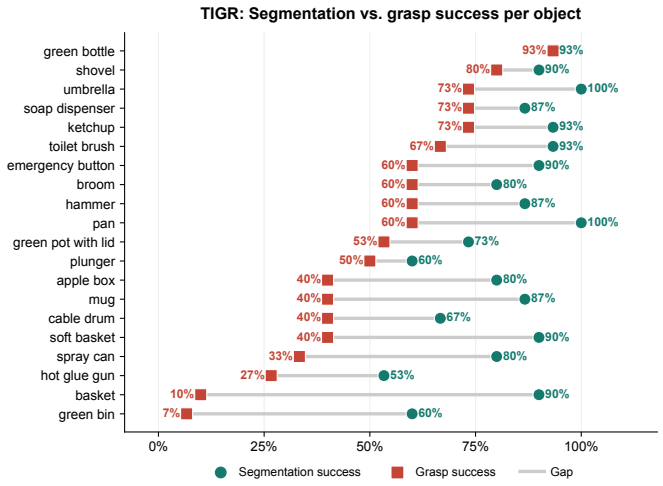

Fig. 4. Per-object comparison of segmentation and grasp success. The grey connector shows the performance gap between perception and grasp execution.

2) Grasp coverage: As Fig. 3 shows, TIGR achieves at least one successful grasp on all 20 objects in the set. At the finer granularity of *(object, task) pairs*, TIGR succeeds at least once on 50/54 combinations. This breadth demonstrates that TIGR's 3D reconstruction and multi-view part segmentation generalize across diverse object geometries and task types without requiring object-specific templates. For reference, the baselines achieve successful grasps on a smaller fraction of the *object set* (GraspMolmo: 18/20, ShapeGrasp: 16/20) as well as on a smaller fraction of the *(object, task) pairs set* (GraspMolmo: 31/54, ShapeGrasp: 27/54), highlighting that a single viewpoint is often insufficient to recover a graspable pose for geometrically complex or partially occluded objects. As Fig. 3 shows, TIGR successfully produces at least one grasp for 50 out of 54 object-task pairs, spanning explicit, functional, and abstract task formulations alike. This confirms that TIGR's VLM-based task decomposition reliably resolves high-level instructions into task-relevant object parts without requiring the instruction to name the target part directly.

3) TIGR: Per-object analysis (Fig. 4): Grasp success varies across objects, reflecting the inherent difficulty of task-oriented grasping for geometrically and functionally diverse objects. Objects with clearly defined functional parts (green bottle: 93%, shovel: 80%) achieve high success rates, while objects requiring precise part segmentation on complex geometries (green bin: 7%, basket: 10%) remain challenging. Comparing segmentation and grasp success per object reveals that segmentation is consistently more stable than grasp execution. For most objects, the segmentation rate substantially exceeds the grasp rate, indicating that the primary bottleneck lies in geometry estimation of the reconstructed mesh and grasp frame selection rather than in segmentation. Inaccurate pose estimates cause the gripper to slip or miss the correctly segmented region entirely, as seen in cases such as pan (100% segmentation, 60% grasp) and umbrella (100% segmentation, 73% grasp), where the task-relevant part is reliably identified but translating the segmentation into a stable grasp frame proves difficult. Additionally, selecting the appropriate grasp frame from a large candidate set remains non-trivial, particularly for objects with thin or geometrically complex targets (basket: 10%, spray can: 33%). Conversely, objects where both rates are low (green bin, hot glue gun) indicate cases where the pipeline already fails at segmentation, leaving no opportunity for downstream recovery.

IV. Conclusion

We presented TIGR, a task-aware grasping pipeline combining 3D reconstruction with a mixture of complementary foundation-model experts fused in 3D. On a real-robot benchmark of 270 trials, TIGR reliably identifies the correct target (97.8%) and segments the task-relevant part (82.6%) across explicit, functional, and abstract task phrasings, achieving at least one successful grasp on every object in the set. We further benchmark GraspMolmo and ShapeGrasp as single-view baselines. Both show a comparable gap between segmentation and grasp success: GraspMolmo is sensitive to depth noise and occlusion from a single depth map, while ShapeGrasp introduces variability through depth-based translation estimates and LLM-selected grasp angles. TIGR's dominant failure source remains depth-dependent pose and scale estimation. However, reasoning over the full object geometry yields usable poses even when depth is locally noisy, and multi-view aggregation over the reconstructed mesh admits tens to hundreds of candidate grasp frames, giving TIGR more flexibility in reachability and viewpoint than single-view methods. This comes at a computational cost, as the sequential multi-model architecture results in significantly longer inference times (40-80 seconds), motivating future work on parallelization or distillation.

In future work, we will consider improved grasp frame selection as well as VLM-based grasp failure detection. We will also extend the evaluation to cluttered scenes where occlusion poses additional challenges for both segmentation and planning. Finally, we will consider alternative paradigms for task-aware grasping beyond object parts, for instance by segmenting affordance heatmaps [3].

References

[1] A. Deshpande, Y. Deng, J. Salvador, A. Ray, W. Han, J. Duan, R. Hendrix, Y. Zhu, and R. Krishna, "Graspmolmo: Generalizable task-oriented grasping via large-scale synthetic data generation," in *Conference on Robot Learning*. PMLR, 2025, pp. 2983–3007.

[2] S. Li, S. Bhagat, J. Campbell, Y. Xie, W. Kim, K. Sycara, and S. Stepputtis, "Shapegrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 10 527–10 534.

[3] Y. Tang, W. Huang, Y. Wang, C. Li, R. Yuan, R. Zhang, J. Wu, and L. Fei-Fei, "Uad: Unsupervised affordance distillation for generalization in robotic manipulation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 3822–3831.

[4] A. X. Appius, É. Garrabé, F. Hélénon, M. Khoramshahi, M. Chetouani, and S. Doncieux, "Task-aware robotic grasping by evaluating quality diversity solutions through foundation models," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 1035–1040.

[5] Y. Tang, S. Zhang, X. Hao, P. Wang, J. Wu, Z. Wang, and S. Zhang, "Affordgrasp: In-context affordance reasoning for open-vocabulary task-oriented grasping in clutter," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 9433–9439.

[6] Y. Zhou, J. Gu, T. Yen Chiang, F. Xiang, and H. Su, "Point-sam: Promptable 3d segmentation model for point clouds," *arXiv e-prints*, pp. arXiv–2406, 2024.

[7] S. K. Warfield, K. H. Zou, and W. M. Wells, "Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation," *IEEE Transactions on Medical Imaging*, vol. 23, no. 7, pp. 903–921, 2004.

[8] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond," *arXiv preprint arXiv:2308.12966*, 2023.

[9] Q. Team, "Qwen3 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2505.09388

[10] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan, "Florence-2: Advancing a unified representation for a variety of vision tasks," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024, pp. 4818–4829.

[11] S. D. Team, X. Chen, F.-J. Chu, P. Gleize, K. J. Liang, A. Sax, H. Tang, W. Wang, M. Guo, T. Hardin, X. Li, A. Lin, J. Liu, Z. Ma, A. Sagar, B. Song, X. Wang, J. Yang, B. Zhang, P. Dollár, G. Gkioxari, M. Feiszli, and J. Malik, "Sam 3d: 3dfy anything in images," 2025. [Online]. Available: https://arxiv.org/abs/2511.16624

[12] T. Lee, B. Wen, M. Kang, G. Kang, I. S. Kweon, and K.-J. Yoon, "Any6D: Model-free 6d pose estimation of novel objects," in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025.

[13] M87 Labs, "Moondream 3 (preview)," Hugging Face Model Hub, 2025, model card for `moondream/moondream3-preview`.

[14] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint*, 2024.

[15] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.

[16] NVIDIA, "Isaac Sim." [Online]. Available: https://github.com/isaac-sim/IsaacSim

[17] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2290–2295 vol.3.

[18] W. Friedl, "Evaluation of different robotic grippers for simultaneous multi-object grasping," *Frontiers in Robotics and AI*, vol. Volume 11 - 2024, 2024. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1351932