GraspGen+HSR: Adapting Simulation-Trained 6-DoF Grasping to Real Service Robots Without Retraining

Alexander Dvorak[*], Michael Nowak[*], Tessa Pulli, Markus Vincze

All authors are with the Automation and Control Institute, TU Wien, Vienna, Austria.
Emails: {e11912029, e12002155}@student.tuwien.ac.at; {pulli, vincze}@acin.tuwien.ac.at

Abstract

Recent diffusion-based 6-DoF grasp generation methods like GraspGen achieve state-of-the-art performance in simulation but face significant challenges when deployed on real robotic platforms. We present a unified adaptation pipeline for the Toyota Human Support Robot (HSR) that bridges these gaps without retraining the foundation model. Our approach combines symmetry-based point cloud completion to mitigate self-occlusion artifacts, three geometric feasibility filters that reduce motion planning failures from 66 % to 16 %, and a kinematic compensation for the HSR's arc-shaped gripper trajectory. We show in our experiments, that our pipeline achieves an overall success rate of 85 % which is competitive with simulation of GraspGen while outperforming baselines M2T2 (56 %) and AnyGrasp (70 %) by up to 29 percentage points. Ablation studies confirm the necessity of each component: symmetry completion improves success by +13 percentage points, while geometric filtering enables 4× more grasp candidates to reach execution. These results demonstrate that post-hoc adaptations can unlock the real-world potential of simulation-trained grasping foundation models on diverse hardware platforms. The code and repository are available at: https://github.com/Ziegenschmuggler/GraspGenforHSR

1 Introduction

Recent advances in 6-DoF grasp generation have enabled robots to predict diverse, stable grasps directly from single-view RGB-D observations [1], [2], [3], [4]. Diffusion-based methods such as GraspGen [1] achieve outstanding results in simulation. However, transferring these simulation-trained models to real-world robotic platforms remains challenging due to three primary factors: (1) hardware-specific kinematics that deviate from the parallel-jaw grippers assumed during training [5], (2) incomplete point clouds from single-view perception that cause grasp predictions on artificial boundaries [6], and (3) execution constraints that prevent motion planners from reaching many grasps that are geometrically feasible [7].

We address these challenges on the Toyota Human Support Robot (HSR) [8], a representative service robot platform with non-standard arc-shaped gripper kinematics. Our study shows that the simulation-trained GraspGen foundation model can be adapted to real hardware without retraining through a unified pipeline that implements symmetry-based point cloud completion to mitigate single-view self-occlusion, three geometric feasibility filters (plane distance, approach-from-below, approach-from-behind) to prune non-executable grasps before motion planning, and analytical kinematic compensation mapping GraspGen's parallel-jaw poses to the HSR's arc trajectory. These results demonstrate that a simulation-trained foundation model, combined with geometric constraints and platform-specific adaptations, achieves competitive real-world performance without requiring

[*]These authors contributed equally to the work.

The Third Austrian Symposium on AI, Robotics, and Vision (AIROV26).

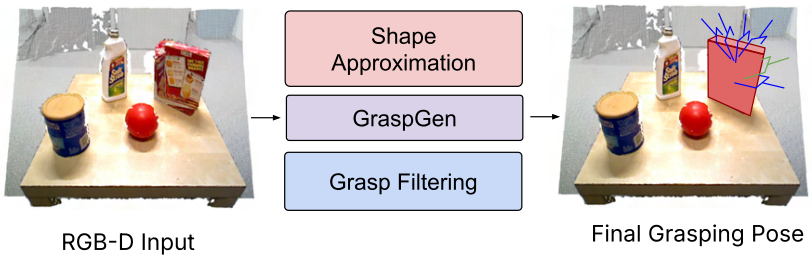


Figure 1: Overview of GraspGen+HSR: Instead of retraining, we combine GraspGen [1] with a symmetric shape approximation and grasp filtering to advance grasping performance.

retraining or collecting real-world data. The consistent success across isolated objects, cluttered scenes, and shelf grasps highlights the robustness of our unified adaptation strategy.

In summary, our contributions are the following:

- Demonstration that a simulation-trained foundation model, combined with geometric constraints and platform-specific adaptations, achieves competitive real-world performance without retraining.
- A ROS-based [9] integration of the GraspGen framework that bridges the gap between learned grasp generation and real-world execution on the HSR platform
- A symmetry-based shape completion method that creates pseudo-volumetric object representations from single-view point clouds, preventing edge grasps while preserving concave regions.
- Three lightweight filters that prune non-executable grasps prior to motion planning, reducing planning failures from two-thirds to 16 %.
- A kinematic compensation for the HSR's arc-shaped gripper trajectory, enabling direct use of GraspGen poses trained for parallel-jaw grippers.

In the following, we review related work on diffusion-based 6-DoF grasp generation and shape completion (Section 2), present our unified adaptation pipeline (Section 3), and evaluate its real-world performance on the Toyota HSR (Section 4).

2 Related Work

In this section, we review diffusion-based 6-DoF grasp generation and shape completion methods for robotic manipulation.

2.1 Diffusion-based 6-DoF grasp generation

Modern 6-DoF grasping is typically framed as generating and scoring grasp poses directly in SE(3) from 3D observations such as point clouds or depth data [10], [11], [1]. Earlier work explored autoregressive models [12] and variational autoencoders [10] to sample grasp candidates and then rank them with a learned critic, resulting in substantial improvements in diversity and success rates compared to purely analytical approaches. More recent methods introduce diffusion-based generators and combine them with discriminators that evaluate sampled poses, which have proven effective for cluttered scenes and across different object shapes [1], [2], [3], [4].

GraspGen [1], follows this line of work and combines a diffusion transformer with an on-generator discriminator, trained entirely in simulation on a large multi-gripper dataset, to achieve strong 6-DoF grasping performance across different embodiments, levels of observability, and scene complexity.

While such models provide high-quality grasps in simulation and for standard parallel-jaw grippers, they typically assume ideal point clouds and do not explicitly encode platform-specific execution constraints [1], [13]. In our work, we adopt GraspGen as the fixed generative backbone and focus on

adapting its predictions to the Toyota HSR without any retraining by adding perception, kinematic, and execution layers that are tailored to the real robot.

2.2 Perception: single-view point clouds and completion

Most 6-DoF grasp networks operate on 3D information derived either from voxel grids, implicit surfaces, or point clouds [1], [14], [15]. In practical setups, especially for mobile manipulators, grasping must often be performed from single-view RGB-D observations, which leads to partial and self-occluded object point clouds [6]. This incompleteness causes grasp generators to place contact points in inefficient locations, leading to grasping failures. To mitigate these issues, shape completion methods have been proposed that reconstruct full meshes or volumetric occupancy from partial inputs before grasp planning [16]. These methods can significantly improve planning robustness but usually require additional training data and heavy inference, which complicates deployment on resource-constrained platforms [17]. As an alternative, we introduce light-weight geometric priors to approximate the object shape. A Symmetry-based point cloud augmentation around the object centroid creates a pseudo-volumetric shell that fills in occluded geometry without requiring a learned completion network. Building on this insight, our pipeline uses a symmetry-based point cloud augmentation that is explicitly designed as a lightweight front-end to GraspGen, preserving graspable concavities while avoiding the cost of full shape completion.

3 Method

Fig. 2 illustrates our unified pipeline that adapts the pre-trained GraspGen [1] model to the HSR [8] without retraining. The system processes single-view RGB-D observations through four sequential stages: perception augmentation, grasp generation, geometric feasibility filtering, and final pose selection with kinematic compensation, followed by MoveIt! [18] motion planning.

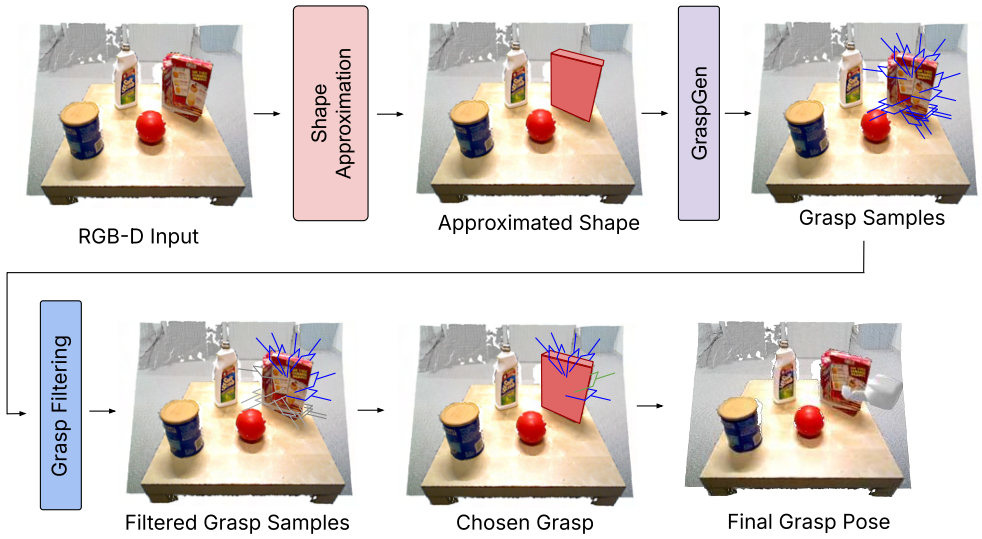
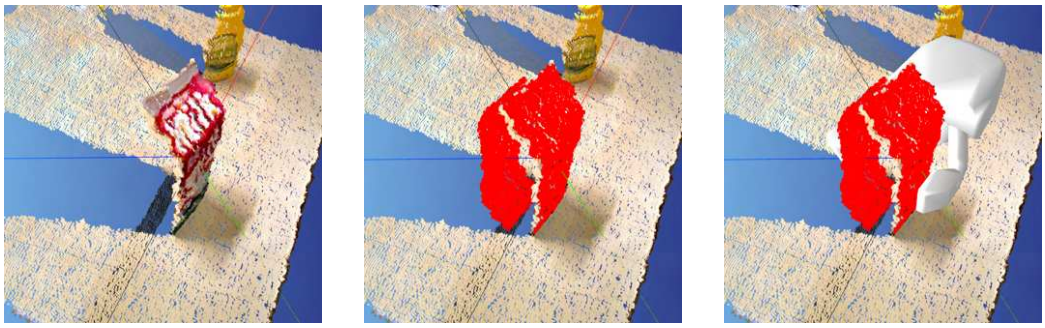


Figure 2: Single-view RGB-D input is processed through symmetry-based shape approximation to create complete object geometry for GraspGen inference. Generated 6-DoF grasp candidates undergo geometric feasibility filtering to remove non-executable poses. The highest-confidence filtered grasp receives HSR-specific kinematic compensation before MoveIt! execution.

3.1 Symmetry-Based Point Cloud Augmentation

Single-view RGB-D data produces incomplete point clouds due to self-occlusion. We create a pseudo-complete object representation by reflecting visible points around their centroid and shifting the reflected points behind the visible surface along the ray directions of the original points, see Fig. 3.

This shape approximation forms a volumetric shell that prevents GraspGen from predicting grasps on artificial depth discontinuities while preserving concave regions suitable for stable grasping.



(a) The original point cloud in real coloring with the blue ray being the view ray direction.

(b) The fully augmented point cloud in red with the symmetry-based added part.

(c) The chosen grasp based on the augmented point cloud.

Figure 3: Visualization of the symmetry-based point cloud augmentation process for creating pseudo-volumetric shells. The mirrored points are shifted by the semi-heuristic parameters s along the ray directions of each corresponding point p in the original point cloud. The view ray of the camera is the blue line visible in all images.

Formally, given a partial point cloud $P_{obs} = \{p_1, \ldots, p_n\}$ captured from a single viewpoint, we compute its centroid as:

$$c = \frac{1}{n} \sum_{i=1}^{n} p_i \tag{1}$$

We generate an augmented cloud P_{aug} by reflecting P_{obs} across its centroid c. To ensure the completed hull does not violate free-space constraints or overlap with the visible surface, we apply shifts s along the ray direction of each point. The augmented points $p' \in P_{aug}$ are defined as:

$$P_{aug} = \{p'_i \mid p'_i = 2c - p_i + s_i, \forall p_i \in P_{obs}\} \tag{2}$$

where s_i are computed as:

$$s_i = \frac{p_i}{||p_i||} \cdot \left(d + \frac{1}{n} \sum_{i=1}^{n} ||p_i - c|| \right), \tag{3}$$

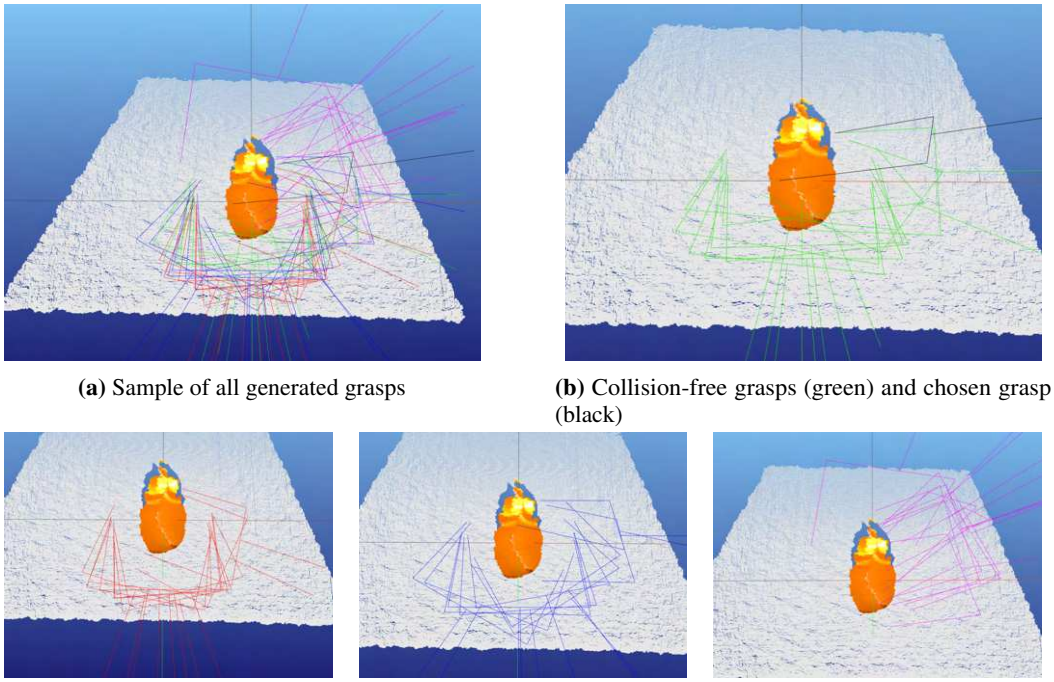
with d being a heuristic safety offset set to 5 mm ensuring that P_{aug} lies strictly behind the visible surface P_{obs}, effectively creating a pseudo-volumetric representation for more stable grasp prediction.

As symmetry assumptions are applied, the augmentation process is only reliable if the object of interest is also mostly symmetrical. Asymmetry to a certain degree is acceptable, as shown in Fig. 3a and Fig. 3b. The milk box has an asymmetric gable-top design where the cap is located. This part is mirrored to the backside bottom of the pseudo-volumetric shell and is therefore irrelevant for grasping as grippers can not get that close to the supporting surface.

3.2 Geometric Grasp Filtering

From the set of collision-free grasp candidates $G = \{g_i\}$ generated by GraspGen (Fig. 4a), we apply three lightweight geometric filters in the camera frame to discard poses that are likely to fail during execution. Each grasp g_i is represented by its position $\vec{t_i}$ and an approach vector $\vec{a}_i$ (the gripper $-z$-axis) derived from the grasp's rotation matrix $\mathbf{R_i}$. The support surface is expected to be the dominant plane and its normal $\vec{p}$ and normal distance d are estimated from the scene point cloud using RANSAC [19]. The effect of each filter is visualized in Fig. 4c–4e.

1) Distance to table: Grasps positioned in proximity to the supporting plane often result in gripper-table collisions. As all points $\vec{r}$ in the dominant plane fulfill the plane equation $\vec{r} \cdot \vec{p} = d$, the normal



(a) Sample of all generated grasps

(b) Collision-free grasps (green) and chosen grasp (black)

(c) Filtered grasps in proximity to the table

(d) Filtered grasps approaching from below

(e) Filtered grasps approaching from behind

Figure 4: Visualization of geometric grasp filtering.

distance δ between grasp position vectors and the plane can be calculated as:

$$\delta = \vec{t} \cdot \vec{p} - d \tag{4}$$

If the distance δ is lower than a given threshold Δ, the grasp is filtered out as shown in Fig. 4c. Typical threshold values Δ would be in the range of a few centimeters, depending on the size and form of the gripper.

2) Approach from below: Grasps with an upward approach vector require the gripper-joint to be positioned lower than the gripper itself, which might cause collisions with the support surface. The gripper's approach vector is defined as $\vec{a} = \mathbf{R}[0:3, 2]$. To quantify this orientation, we compute the alignment:

$$\gamma = \vec{a} \cdot \vec{p} \tag{5}$$

As shown in Fig. 4d, assuming the plane normal points upwards, candidates are pruned if they are above a given threshold $\gamma > \Gamma$, eliminating the risk of table collisions. As γ lies in the range $[-1, 1]$, a threshold of $\Gamma = 0$ would effectively remove all grasps pointing upwards. Slightly higher thresholds may be applicable if the gripper has a slim design. The lower the threshold, the more rigorous is also the exclusion of downward approaching grasps.

3) Approach from behind: To avoid trajectories that require the robot to move around the table or approach the object from the far side, we filter grasps that come from behind relative to the robot base (Fig. 4e). The robot approach direction is calculated as $\vec{r}_a = -\vec{e}_x \times \vec{p}$, assuming $\vec{e}_x$ points to the right in the camera frame. All grasps coming from behind fulfill the condition below and are filtered out, as shown in Fig. 4e.

$$\vec{r}_a \cdot \vec{a} < 0 \tag{6}$$

Applying these three constraints yields the geometrically feasible subset $G_{\text{feas}} \subset G$, which still covers diverse approach directions but significantly reduces non-executable candidates. The final selection among the remaining collision-free, feasible grasps is illustrated in Fig. 4b.

3.3 Kinematic Compensation

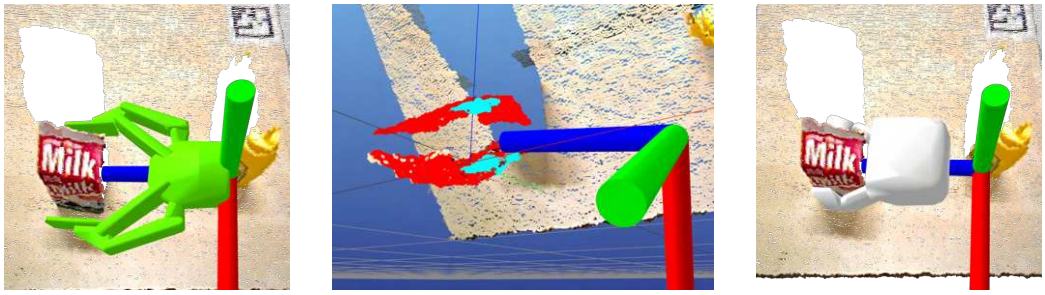
The GraspGen model predicts a grasp pose $T_{grasp} \in SE(3)$ assuming a fixed Tool Center Point (TCP). However, the HSR's arc-shaped fingers cause the physical contact point to shift along the local z-axis as a function of the gripper aperture w. To align the predicted pose with the physical hardware, we define the corrected execution pose T_{exec} as:

$$T_{exec} = T_{grasp} \cdot \text{Trans}(0, 0, \Delta z(w)) \tag{7}$$

The compensation value $\Delta z(w)$ is derived from the kinematic linkage of the HSR gripper using trigonometric relations:

$$\Delta z(w) = L \cdot \left(1 - \sqrt{1 - \left(\frac{w}{2L}\right)^2}\right) \tag{8}$$

where L represents the distance between the gripper-base and the fingertips. This transformation ensures that the finger pads align precisely with the object surface, regardless of its width, preventing collisions or shallow grasps caused by the arc-shaped closing trajectory. A comparison of the gripper kinematics and a visualization of the object's width estimation is shown in Fig. 5.



(a) Robotiq 2F-140 parallel-jaw kinematics assumed by GraspGen.

(b) Width estimation using the augmented point cloud.

(c) HSR gripper kinematics with an arc-shaped closing trajectory.

Figure 5: Gripper compensation: As the deployed gripper on the HSR has an arc-shaped closing trajectory, the TCP needs to be adjusted based on the estimated width, which defines the gripper's closing aperture w.

4 Experiments

We evaluate our integrated pipeline on a Toyota HSR in real-world grasping experiments. The experiments are designed to assess the impact of kinematic compensation, symmetry-based point cloud completion, and geometric grasp filtering across different scene configurations.

4.1 Experimental Setup

The HSR is equipped with a head-mounted RGB-D sensor used for object perception. We employ Grounded SAM [20] for instance segmentation of target objects and MoveIt! [18] for motion planning. Our test suite consists of 8 diverse household objects (e.g., bottles, cans, boxes, and small tools) spanning different geometric classes, including cylindrical, box-like, and thin elongated shapes. A *successful grasp* is defined as the robot closing the gripper on the target object, lifting it from the support surface, and maintaining a stable hold during a short transport motion.

4.2 Real Robot Experiments

We compare our full GraspGen+HSR pipeline against the unmodified GraspGen model, M2T2 [3], and AnyGrasp [4]. For the baselines, we use the official weights and default configurations as released by the authors.

We evaluate three settings that progressively increase task difficulty: grasping isolated objects on a clear tabletop, grasping a defined object from cluttered tabletop scenes, and grasping an object

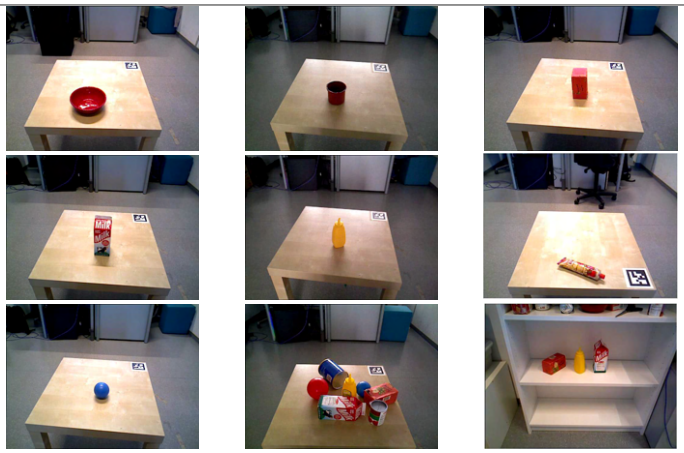


Figure 6: Experimental Setup for some single standing objects, the cluttered scene and the shelf test. The objects are from left to right and top to bottom labeled as: Bowl, Mug, Tea Box, Milk, Mustard, Tomato Tube, Blue Ball. The Small Cylinder can be seen in the cluttered scene (bottom middle) at the bottom right.

standing in a shelf compartment (Fig. 6). All isolated objects are grasped 10 times, resulting in a total of 80 trials. In the cluttered scenes and the shelf tests, the mustard bottle is always the object of interest, again being grasped 10 times in both settings. Experimental results are summarized in Tab. 1. While GraspGen slightly edges out our method on isolated objects (91 % vs. 86 %), our full pipeline clearly excels in clutter (90 % vs. 83 %) due to symmetry-based shape completion and in shelf scenarios (80 % vs. 72 %) thanks to geometric feasibility filtering.

These results show that a simulation-trained foundation model, combined with geometric constraints (approach filtering, plane distance) and platform-specific adaptations (kinematic compensation, symmetry completion), can achieve competitive real-world performance without the need for retraining or real-world data collection. The consistent success across diverse scenarios highlights the robustness of our unified adaptation strategy.

Table 1: Comparison with recent 6-DoF grasping methods across three scenarios. Our GraspGen+HSR pipeline significantly outperforms baselines, particularly in clutter and constrained shelf environments. Experimental results for GraspGen, M2T2 and AnyGrasp have been reported in [1].

Method	Isolated	Cluttered Table	Shelf
GraspGen+HSR (Ours)	86 %	**90 %**	**80 %**
GraspGen [1]	**91 %**	83 %	72 %
M2T2 [3]	81 %	75 %	14 %
AnyGrasp [4]	86 %	83 %	43 %

4.3 Ablation Study: Geometric Filtering

To quantify the effect of the geometric feasibility filters, we compare our full pipeline against a baseline that uses GraspGen generation and scene collision checking only, without the additional plane-distance and approach-direction constraints. In the baseline, the highest-confidence grasp is rejected by the motion planner as infeasible in 66% of trials, leading to long planning times and frequent failures. In Fig. 7, this metric is reported as the planning failure rate, defined as the fraction of highest-scoring grasps rejected by the planner. With our geometric filtering in place, this planning failure rate is reduced to 16 % (see Fig. 7), which significantly decreases the latency between perception and execution and increases the number of trials that reach the execution phase.

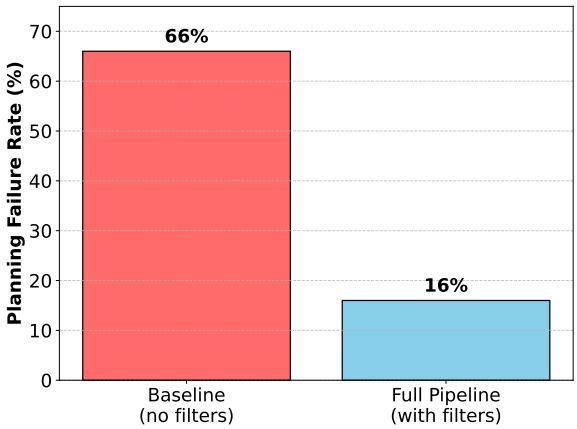


Figure 7: Effect of Geometric Filtering on Motion Planning

4.4 Grasping Performance

We conducted 100 grasp attempts (isolated objects + cluttered scenes + shelf tests) that systematically test each component of our pipeline. In isolated-object scenarios on clear tabletops, we evaluate the accuracy of our kinematic compensation by measuring grasp stability and lift success for single objects without occlusions. Cluttered scenes assess the effectiveness of symmetry-based point cloud completion under partial self-occlusion, where GraspGen would otherwise predict grasps on incomplete object boundaries. Shelf grasping scenarios challenge the geometric approach-direction filters by requiring precise top-down trajectories in vertically constrained environments.

The complete pipeline achieves an overall success rate of 86 % across all conditions. Table 2 reports per-category performance, revealing that cubic objects (Milk) benefit most from our symmetry completion. Transparent and metallic items remain challenging due to inherent perception limitations. Object detection from RGB-D sensor data with Grounded SAM [20] yielded poor point clouds, which inevitably led to poor grasping results and frequent planning failures. This is why they are not included in Tab. 2. Fig. 7 visualizes the reduction in motion planning failures enabled by our geometric filters.

Object Category	**w/o Symmetry Expansion**	**w/ Symmetry Expansion**
Bowl	80 %	90 %
Milk	50 %	90 %
Blue Ball	80 %	100 %
Mug	70 %	80 %
Mustard	100 %	90 %
Small Cylinder	80 %	80 %
Tea Box	80 %	90 %
Tomato Tube	90 %	70 %
Cluttered Scene	20 %	90 %
Shelf Tests	80 %	80 %
Total	**73 %**	**86 %**

Table 2: Grasping Success Rate Across Different Object Categories. Details about setup, shape, and size can be seen in Fig. 6

5 Conclusion

We presented a unified pipeline adapting simulation-trained GraspGen to the Toyota HSR without retraining. Our experiments demonstrate that post-hoc adaptations can unlock foundation grasping models for diverse service robots, avoiding costly retraining. Future work will extend the pipeline to a language-guided zero-shot pick-and-place method.

ACKNOWLEDGMENT

We gratefully acknowledge the support of the EU-program EC Horizon 2020 for Research and Innovation under project No. I 6114, project iChores and the EU-program EC Horizon 2020 for Research and Innovation under grant agreement No. 101017089, project TraceBot.

Use of LLMs

During the preparation of this work, the authors used ChatGPT, Google Gemini, and Perplexity to improve the language and readability of the manuscript and to assist in writing code for visualizing experimental results. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

[1] A. Murali et al., *GraspGen: A Diffusion-based Framework for 6-DoF Grasping with On-Generator Training*, 2025. arXiv: 2507.13097 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2507.13097.

[2] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, *SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion*, 2023. arXiv: 2209.03855 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2209.03855.

[3] W. Yuan, A. Murali, A. Mousavian, and D. Fox, "M2T2: Multi-Task Masked Transformer for Object-centric Pick and Place," in *7th Conference on Robot Learning*, vol. 229, PMLR, 2023, pp. 3619–3630. [Online]. Available: https://proceedings.mlr.press/v229/yuan23a.html.

[4] H.-S. Fang et al., "AnyGrasp: Robust and Efficient Grasp Perception in Spatial and Temporal Domains," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3929–3945, 2023. DOI: 10.1109/TRO.2023.3281153.

[5] C. M. Kim, M. Danielczuk, I. Huang, and K. Goldberg, "IPC-GraspSim: Reducing the Sim2Real Gap for Parallel-Jaw Grasping with the Incremental Potential Contact Model," in *ICRA*, IEEE, May 2022, pp. 6180–6187. DOI: 10.1109/ICRA46639.2022.9811777.

[6] Y.-K. Wang, C. Xing, Y.-L. Wei, X.-M. Wu, and W.-S. Zheng, "Single-View Scene Point Cloud Human Grasp Generation," in *IEEE/CVF CVPR*, 2024, pp. 831–841. DOI: 10.1109/CVPR52733.2024.00085.

[7] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, *6-DoF Grasping for Target-driven Object Manipulation in Clutter*, 2020. arXiv: 1912.03628 [cs.RO]. [Online]. Available: https://arxiv.org/abs/1912.03628.

[8] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of Human Support Robot as the Research Platform of a Domestic Mobile Manipulator," in *IEEE/RSJ IROS*, 2018, pp. 1–9. DOI: 10.1109/IROS.2018.8594344.

[9] M. Quigley et al., "ROS: An open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, vol. 3, Jan. 2009.

[10] A. Mousavian, C. Eppner, and D. Fox, *6-DoF GraspNet: Variational Grasp Generation for Object Manipulation*, 2019. arXiv: 1905.10520 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1905.10520.

[11] H. Liang et al., "PointNetGPD: Detecting Grasp Configurations from Point Sets," in *2019 ICRA*, IEEE, May 2019, pp. 3629–3635. DOI: 10.1109/icra.2019.8794435. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2019.8794435.

[12] J. Tobin et al., "Domain Randomization and Generative Models for Robotic Grasping," in *IEEE/RSJ IROS*, 2018, pp. 3482–3489. DOI: 10.1109/IROS.2018.8593933.

[13] B. Han, M. Parakh, D. Geng, J. A. Defay, G. Luyang, and J. Deng, *FetchBench: A Simulation Benchmark for Robot Fetching*, 2024. arXiv: 2406.11793 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2406.11793.

[14] T. G. W. Lum et al., *Get a Grip: Multi-Finger Grasp Evaluation at Scale Enables Robust Sim-to-Real Transfer*, 2024. arXiv: 2410.23701 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2410.23701.

[15] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, "Volumetric Grasping Network: Real-time 6 DoF Grasp Detection in Clutter," in *Conference on Robot Learning*, PMLR, 2021, pp. 1602–1611.

[16] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape Completion Enabled Robotic Grasping," in *IROS*, IEEE, 2017, pp. 2442–2447.

[17] S. S. Mohammadi et al., "3DSGrasp: 3D Shape-Completion for Robotic Grasp," in *ICRA*, 2023, pp. 3815–3822. DOI: 10.1109/ICRA48891.2023.10160350.

[18] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "MoveIt! Task Constructor for Task-Level Motion Planning," in *ICRA*, 2019, pp. 190–196. DOI: 10.1109/ICRA.2019.8793898.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. DOI: 10.1145/358669.358692.

[20] T. Ren et al., *Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks*, 2024. arXiv: 2401.14159 [cs.CV].