

Supplementary Material for Dexonomy: Synthesizing All Dexterous Grasp Types in a Grasp Taxonomy

Jiayi Chen^{1,2*}, Yubin Ke^{1,2*}, Lin Peng² and He Wang^{1,2,3†}

Abstract—This material first introduces the evaluation metrics and provides more experimental results of our proposed grasp synthesis pipeline for both fingertip grasps and more complex grasp types. Then, we show an application of our algorithm for annotating grasps via a UI. Next, we introduce the synthesized dataset and the proposed type-conditional generative model for grasp synthesis from single-view point clouds. Finally, a detailed time analysis is performed.

I. EXPERIMENT

A. Evaluation Metrics

The following metrics are used for a comprehensive evaluation of the synthesis pipeline and grasp quality. All distances are measured using collision meshes in MuJoCo.

Grasp Success Rate (GSR) (unit: %): The percentage of successful grasps relative to the attempt number. For our method, one attempt is defined as one valid result output by the global alignment stage. A grasp succeeds only if it resists six external forces in MuJoCo and does not have severe penetrations (> 1 cm), since the penetration may cause simulation failure and prevent the object from moving. The object mass is 100g, and the success criteria for the object pose are 5cm and 15° .

Object Success Rate (OSR) (unit: %): The percentage of objects that have at least one successful grasp. If the object scales are fixed, different scales of the same object are treated as separate objects.

Speed (S) (unit: second^{-1}): The maximum number of attempts completed per second on a server with 8 NVIDIA RTX 3090 GPUs and 2 Intel Xeon Platinum 8255C CPUs (48 cores, 96 threads). We report the time running on a server because our method utilizes both GPUs and CPUs. This metric excludes simulation validation.

Contact Link Number (CLN): The number of hand links whose distance to the object surface is within 2 mm.

Contact Distance Consistency (CDC) (unit: mm): The delta between the maximum and minimum signed distances across all fingers. This metric quantifies the variation in contact distance across different fingers and is invariant to penetration.

Penetration Depth (PD) (unit: mm): The maximum intersection distance between the hand and object for each grasp.

Self-Penetration Depth (SPD) (unit: mm): The maximum self-intersection distance among different hand links.

Diversity (D) (unit: %): The proportion of total variance explained by the first principal component in PCA, computed

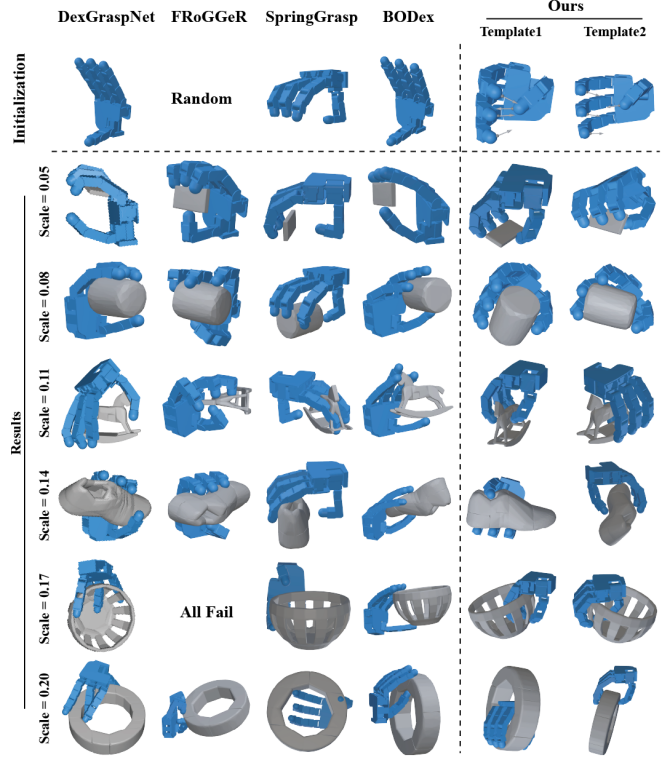


Fig. 1: **Visualization of Synthesized Fingertip Grasps.** Our method synthesizes human-like and stable grasps, even for objects with complex geometries (e.g., object scales = 0.11, 0.17, and 0.20).

as the ratio of the first eigenvalue to the sum of all eigenvalues. PCA is performed on data points that include grasp translation \mathbf{T}_g , rotation \mathbf{R}_g (in the axis-angle representation), and joint angles \mathbf{q}_g .

B. Type-unaware Grasp Synthesis

1) *Visualization Comparison*: Figure 1 illustrates the initial hand pose and some synthesized grasps for each method. Our method consistently synthesizes human-like and stable grasps, even for objects with complex geometries (e.g., for scales 0.11, 0.17, and 0.20). Notably, the synthesized grasp for template 1 and object scale 0.05 requires high precision and is challenging for previous methods. Furthermore, the grasp for template 2 and object scale 0.14 shows a much larger thumb-to-other-tip distance than the initial human-annotated template, demonstrating our method’s ability to adjust hand joint angles across a large range.

¹Peking University. ²Galbot. ³Beijing Academy of Artificial Intelligence.

*Equal contribution. [†]Corresponding author: hewang@pku.edu.cn.

Method	Attempt Number	DGN object [1]		Objaverse [5]	
		GSR \uparrow	OSR \uparrow	GSR \uparrow	OSR \uparrow
BODex	20	14.79	71.30	6.92	43.48
BODex	100	14.80	89.84	6.91	73.53
Ours	20	27.16	91.28	18.25	84.17
Ours	100	27.18	95.13	18.34	94.63

TABLE I: **A Harder Benchmark for Fingertip Grasp Synthesis.** This benchmark uses smaller friction coefficients and more diverse objects, and our method consistently outperforms the baseline. DGN indicates DexGraspNet.

For baseline methods, DexGraspNet [1] shows high uncertainty, partly due to its randomness in selecting contact points. While it occasionally generates good grasps (e.g., for scales 0.08 and 0.14), it often results in twisted fingers (e.g., for scales 0.17 and 0.20) or large thumb-to-object distance (e.g., for scale 0.05). FRoGGeR [2] performs well on simple objects but almost always fails on objects with complex geometries. It also tends to generate grasps with different contact normals for each fingertip (e.g., for scales 0.05 and 0.08), an issue encouraged by many previous force closure metrics. SpringGrasp [3] suffers from severe penetration and inconsistent contact distances, especially for the thumb. Additionally, their grasps lack diversity, and their thumb joint frequently exceeds the feasible range, which is not executable in both MuJoCo and the real world. Although BODex [4] demonstrates high success rates in simulation, their synthesized grasps rarely involve finger bending, resulting in unnatural poses.

2) *Comparison using a harder benchmark:* The benchmark in the main paper uses large friction coefficients and many simple objects, which do not fully reflect the ability of each method to synthesize very high-quality grasps in more complex scenarios. To address this, we introduce a more challenging benchmark by reducing the tangential and torsional friction coefficients from 0.6 and 0.02 to 0.3 and 0.002, respectively, and randomly selecting 5000 additional objects from Objaverse [5] for testing. To mitigate the increased difficulty, we allow each method more attempts per object (from 20 to 100). We compare only with BODex, as other baselines exhibit significantly lower success rates and slower speeds.

As shown in Table I, our method significantly outperforms BODex. Notably, our method achieves an object success rate exceeding 94%, successfully grasping nearly all scaled objects, while BODex fails on about 27% of the Objaverse objects. This highlights our method’s stronger generalizability to complex in-the-wild objects. Additionally, our grasp success rate continues to improve with more attempts, benefiting from continuous updates to our template repository, whereas the performance of BODex remains unchanged. This further demonstrates the adaptability of our approach.

C. Type-aware Grasp Synthesis

1) *Visual Comparison:* Unfortunately, we didn’t acquire the code of suitable baselines for comparison, as existing methods either do not support robotic hands (e.g.,

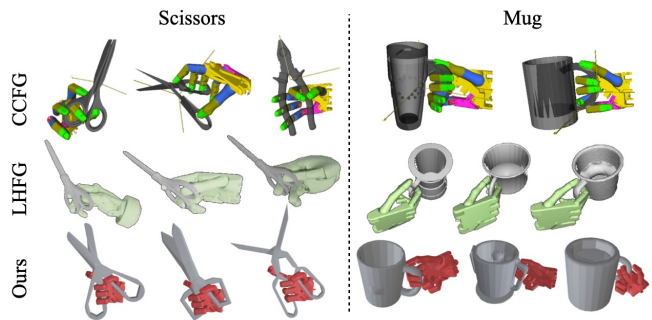


Fig. 2: **Comparison with Functional Grasp Transfer Baselines.** Our grasps involve more contact points while ensuring no penetration, indicating higher stability, particularly for scissors.

	GSR(%) \uparrow		OSR(%) \uparrow		CLN \uparrow	D(%) \downarrow
	Normal	Hard	Normal	Hard		
Power	24.2	12.8	81.9	68.3	9.1	24.7
Intermediate	23.0	6.6	79.9	69.4	4.8	27.6
Precision	36.0	11.4	95.9	85.6	4.2	25.8

TABLE II: **Statistics of Grasp Synthesis for the GRASP Taxonomy.** The success rate is lower than fingertip grasps because many flexible grasp types are suitable only for specific objects, e.g., *Lateral* (#16) grasps for flat objects.

Oakink [6]) or have not made their code publicly available (e.g., LHFG [7] and CCFG [8]) despite our requests through emails. Consequently, we can only perform qualitative comparisons using figures from their papers. As shown in Fig. 2, previous baselines mainly use fingertips to grasp the object, particularly for scissors. In contrast, our method achieves significantly more contact points (approximately 10 for scissors and 7 for mugs), resulting in more stable and human-like grasps. Additionally, CCFG’s grasps show noticeable penetrations especially with mugs, while LHFG reports a maximum penetration of about 1 cm in their paper. In contrast, our grasps do not have any penetration.

2) *Statistics analysis of our pipeline:* In the absence of a suitable baseline for comparison, we provide some quantitative results in Table II, which were gathered while synthesizing our Dexonomy dataset in Section III. The grasp types are categorized into three large groups, namely power, intermediate, and precision grasps, according to the GRASP taxonomy [11].

The overall success rate is considerably lower than that of fingertip grasp synthesis, as many flexible grasp types are designed for specific object shapes. For instance, the *Lateral* (#16) grasp is only used for flat and small objects. Among different grasp types, precision grasps exhibit the highest success rate under *normal* test conditions (i.e., with friction coefficients of 0.6 and 0.02), since these grasps typically involve only the fingertips and suit more objects. However, the success rate of precision grasps drops more rapidly than that of power grasps when the friction coefficients are reduced to 0.3 and 0.002 (i.e., the *hard* test conditions), indicating that power grasps offer higher stability due to more

Dataset	Hand	Sim./Real	Objects	Grasps	Grasp Types	Force Closure	Data Type	Method
DexGraspNet [1]	Shadow	IsaacGym	5.4k	1.32M	Random	✓	Grasp pose	Optimization
RealDex [9]	Shadow	Real	52	59k	Random	✗	Motion	Teleoperation
GraspXL [10]	Multiple	RaiSim	500k	10M	Random	✗	Motion	RL
BODex [4]	Shadow	MuJoCo	2.4k	3.62M	Fingertip	✓	Pre-grasp, grasp poses	Optimization
Dexonomy (Ours)	Shadow	MuJoCo	10.7k	9.5M	31 types	✓	Pre-grasp, grasp, squeeze poses	Sampling+opt.

TABLE III: **Dexterous Grasp Dataset Comparison.** Our large-scale dataset aims to support the study of data-driven methods for type-aware grasp synthesis.

Method	Dataset	GSR↑	OSR↑	CDC↓	PD↓	D↓
Type-uncond.	DGN [1]	8.32	44.3	20.5	15.9	29.1
	BODex [4]	54.0	84.4	11.7	6.2	32.0
	Ours-type1	55.5	85.9	10.8	8.4	31.5
	Ours-all	24.5	73.2	15.6	11.6	28.0
Type-cond.	Ours-all	63.9	91.3	13.9	8.6	25.7

TABLE IV: **Learning-based Grasp Synthesis from Single-View Object Point Clouds in Simulation.** Our type-conditional model trained on our Dexonomy dataset significantly outperforms baselines.

contact with the object. Additionally, the overall diversity of grasps is better than previous work reported in the main paper, owing to the inclusion of many distinct grasp types.

D. Learning-based Grasp Synthesis in Simulation

In this section, we compare the influence of both the grasping dataset and the learning method in simulation. The 10.7k objects in our Dexonomy dataset are randomly split into training and test sets with a 4:1 ratio. While the object scales used for training vary, we fix the scales during testing, using the same six scale levels as described in Section ???. To ensure a fair comparison, we also regenerate a dataset for BODex using our objects and scales, resulting in 0.7M valid grasps. *Ours-type1* includes only the *Large Diameter* (#1) grasp type from the Dexonomy dataset and contains 0.4M data points, while *Ours-all* uses the full 9.5M dataset. For the type-conditional model, we additionally train a classifier to select the best grasp type based on each object’s point cloud. For each object, 100 candidate grasps are predicted and ranked by their associated probabilities, with the top 10 selected as the final outputs.

As shown in Table IV, our type-conditional model trained on the Dexonomy dataset significantly outperforms the BODex baseline by around 10%, further highlighting the value of our dataset. Notably, even when using only a single grasp type with less data, the learned model still outperforms its counterpart trained on BODex. Without type-conditional features, the model struggles to learn from the diverse grasp data and performs poorly. In contrast, the type-conditional model successfully synthesizes the intended grasp types, as visualized in Figure ???. The model trained on our dataset exhibits slightly higher penetration, likely due to the fact that our grasps are more contact-rich. Contact distance consistency is also higher for *Ours-all*, as this metric considers all fingers, while some grasp types do not involve every finger.

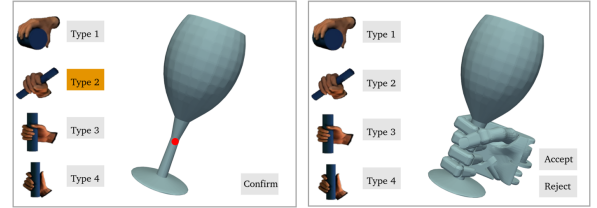


Fig. 3: **An Annotation UI based on Our Algorithm for Collecting Functional Grasp.** (Left) The user *clicks twice* to specify a contact point on the object and a grasp type. (Right) A high-quality grasp is synthesized according to the user’s needs within seconds.

II. APPLICATION: ANNOTATION UI

Although our algorithm is semantic-unaware and cannot directly synthesize grasps to touch object regions specified by human language commands, it can be used to develop an efficient annotation system for collecting semantic dexterous grasp data. Unlike widely used teleoperation methods, which often require well-trained annotators and hardware dependencies like data gloves, our annotation system has minimal requirements, relying only on simple mouse clicks.

As shown in Figure 3, the annotator only needs to click twice: once to specify a contact point on the object and once to select a desired grasp type. Our algorithm will automatically sample nearby object points and grasp templates from existing libraries, and synthesize valid grasps, with the best results displayed in the GUI within seconds. For a full demonstration, please refer to our supplementary video. We plan to continue improving this tool and hope it facilitates future research on semantic grasping.

III. DEXONOMY DATASET

Using our proposed grasp synthesis pipeline, we construct a large-scale dataset for Shadow hand covering 31 grasp types from the GRASP taxonomy [11]. This dataset is designed to support research on data-driven methods for type-aware grasp synthesis. Two grasp types in the taxonomy, *Distal Type* (#19) and *Tripod Variation* (#21), are excluded due to their specificity to object categories, namely scissors and chopsticks, respectively.

As shown in Table III, our dataset comprises 10.7k object assets, including 5,697 objects from DexGraspNet [1] and 5,000 new objects randomly selected from Objaverse [5]. All objects are normalized such that the diagonal of their axis-aligned bounding box is 2 meters, with scales ranging between [0.05, 0.2]. Only successful grasps are retained,

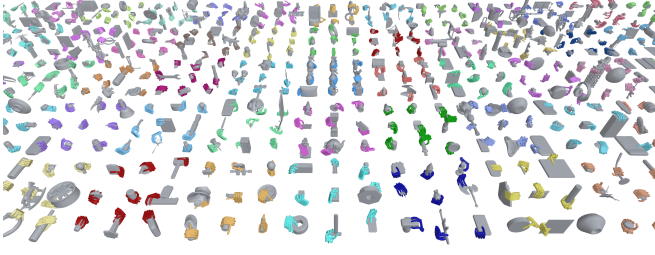


Fig. 4: **Dexonomy Dataset Visualization.** Each color corresponds to a different grasp type.

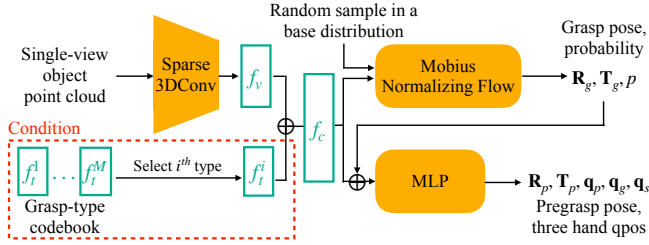


Fig. 5: **Type-Conditional Grasp Generative Model.** Without the grasp-type codebook in the red dashed box, the model becomes type-unconditional and is similar to previous works [4], [12].

resulting in 9.5M data points. The entire dataset was synthesized in less than 3 days on a server with 8 NVIDIA RTX 3090 GPUs. Additional statistics are provided in Table II.

Each data point includes three key poses:

- **Grasp pose**, obtained via local refinement.
- **Pre-grasp pose** for collision-free motion planning, generated after the grasp pose by enforcing a 2cm contact margin in MuJoCo—pushing the hand away if it is within 2cm of the object.
- **Squeeze pose**, derived from the control signal used for simulation validation, to apply force through hand-object contacts.

These poses provide the minimal requirements for generating a complete grasping trajectory (including reaching and squeezing) and are compatible with diverse robot arms and initial hand configurations.

IV. TYPE-CONDITIONAL GRASP GENERATIVE MODEL

To generate grasps from partial observations for real-world deployment, data-driven methods are essential. Although learning is not the main focus of this paper, we present a simple model as an initial try. The model architecture is very similar to previous works [12], [4], with the key difference being the grasp-type codebook added as a conditional input to specify a grasp type.

The input to the model consists of a single-view object point cloud and a type feature f_t^i selected from the grasp-type codebook. The point cloud is encoded into a feature f_v using a Sparse3DConv network with MinkowskiEngine [13]. This vision feature f_v , along with the type feature f_t , are concatenated to form a conditional feature f_c . Conditioned

on f_c , the Mobius normalizing flow [14] maps a random sample in a base distribution to a grasp pose R_g and T_g , and calculates a probability p indicating the pose quality. The predicted grasp pose is then concatenated with f_c and passed through an MLP to predict a pre-grasp pose R_p , T_p , and three hand qpos q_p , q_g , and q_s for the pre-grasp, grasp, and squeeze poses, respectively. The whole model is trained end-to-end and the type feature f_t^i is also optimizable.

V. TIME ANALYSIS

The times reported in Table 1 of the main paper represent the maximum speed for synthesis **without simulation validation**. This section provides a more detailed time breakdown of our proposed grasp synthesis pipeline.

First, the *lightweight global alignment* stage processes over 100,000 initial samples in approximately 3 seconds on a single 3090 GPU. The maximum number of intermediate results generated for the next stage can be controlled via a hyperparameter. We typically process 10 objects in parallel, with 10 results per object. For grasp types that are commonly suitable for many objects, this stage is usually not the bottleneck, as it can synthesize over 200 intermediate results per second using 8 GPUs. However, for more challenging grasp types that are hard to match, this stage can become the bottleneck, because there may be less than 20 results per second.

The optimization step consistently takes around 1.2 seconds, while the time cost of calculating the grasp quality metric for post-filtering varies significantly, ranging from 0.3 to 1.5 seconds. When many samples are filtered out, leaving only around 5,000 for energy calculation, the process takes approximately 0.3 seconds. This speed is achieved by using the batched Relu-QP [15] algorithm as in BODex [4], whereas traditional CPU-based QP solvers are significantly slower. The other operations are very fast.

Next, the *simulation-based local refinement* stage requires 200 simulation steps, which take less than 0.1 seconds. This stage is highly efficient, easily synthesizing more than 200 grasps per second when utilizing 32 threads.

Finally, the *simulation validation* stage often becomes the speed bottleneck, as it involves approximately 3,000 simulation steps (6 external force directions, with 500 steps per direction). Despite employing early-stop strategies to handle failure cases, this stage can only process about 40 grasps per second using 48 threads. The slow speed has nothing to do with our proposed contact-aware control strategy and is consistent for other synthesis baselines if they want to test in MuJoCo. Future work may try to use GPU-based MuJoCo or other faster physics simulators for testing.

REFERENCES

- [1] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, “Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 359–11 366.
- [2] A. H. Li, P. Culbertson, J. W. Burdick, and A. D. Ames, “Frogger: Fast robust grasp generation via the min-weight metric,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 6809–6816.
- [3] S. Chen, J. Bohg, and C. K. Liu, “Springgrasp: An optimization pipeline for robust and compliant dexterous pre-grasp synthesis,” *arXiv preprint arXiv:2404.13532*, 2024.
- [4] J. Chen, Y. Ke, and H. Wang, “Bodex: Scalable and efficient robotic dexterous grasp synthesis using bilevel optimization,” *arXiv preprint arXiv:2412.16490*, 2024.
- [5] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. Vander-Bilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, “Objaverse: A universe of annotated 3d objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 142–13 153.
- [6] L. Yang, K. Li, X. Zhan, F. Wu, A. Xu, L. Liu, and C. Lu, “Oakink: A large-scale knowledge repository for understanding hand-object interaction,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 20 953–20 962.
- [7] W. Wei, P. Wang, S. Wang, Y. Luo, W. Li, D. Li, Y. Huang, and H. Duan, “Learning human-like functional grasping for multi-finger hands from few demonstrations,” *IEEE Transactions on Robotics*, 2024.
- [8] R. Wu, T. Zhu, X. Lin, and Y. Sun, “Cross-category functional grasp transfer,” *arXiv preprint arXiv:2405.08310*, 2024.
- [9] Y. Liu, Y. Yang, Y. Wang, X. Wu, J. Wang, Y. Yao, S. Schwertfeger, S. Yang, W. Wang, J. Yu, *et al.*, “Realdex: Towards human-like grasping for robotic dexterous hand,” *arXiv preprint arXiv:2402.13853*, 2024.
- [10] H. Zhang, S. Christen, Z. Fan, O. Hilliges, and J. Song, “Graspxl: Generating grasping motions for diverse objects at scale,” in *European Conference on Computer Vision*. Springer, 2025, pp. 386–403.
- [11] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, “The grasp taxonomy of human grasp types,” *IEEE Transactions on human-machine systems*, vol. 46, no. 1, pp. 66–77, 2015.
- [12] J. Zhang, H. Liu, D. Li, X. Yu, H. Geng, Y. Ding, J. Chen, and H. Wang, “Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes,” in *8th Annual Conference on Robot Learning*, 2024.
- [13] C. Choy, J. Gwak, and S. Savarese, “4d spatio-temporal convnets: Minkowski convolutional neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3075–3084.
- [14] Y. Liu, H. Liu, Y. Yin, Y. Wang, B. Chen, and H. Wang, “Delving into discrete normalizing flows on so (3) manifold for probabilistic rotation modeling,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 264–21 273.
- [15] A. L. Bishop, J. Z. Zhang, S. Gurumurthy, K. Tracy, and Z. Manchester, “Relu-qp: A gpu-accelerated quadratic programming solver for model-predictive control,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 285–13 292.