GraspXL: Generating Grasping Motions for Diverse Objects at Scale

Hui Zhang[1], Sammy Christen[1], Zicong Fan[1,2], Otmar Hilliges[1], and Jie Song[1]

[1] ETH Zürich, Switzerland
[2] Max Planck Institute for Intelligent Systems, Germany

Supplementary Material

In Sec. 1, we provide qualitative results compared against our baseline. We then provide the implementation details about hyperparameters and motion objectives in Sec. 2. In Sec. 3, we show the experiment details about data preprocessing and the evaluation with generated and reconstructed objects. Finally, we provide additional experiments in Sec. 4.

1 Qualitative Results

We provide qualitative comparisons of our method with SynH2R [3] for objective-driven grasping synthesis in Fig. 1. From the figures, we can see that SynH2R either failed to establish a stable grasp due to noisy generated references, or cannot precisely follow the objectives. However, our method does not require a pre-generated reference, and can generate motions with stable grasping while satisfying motion objectives.

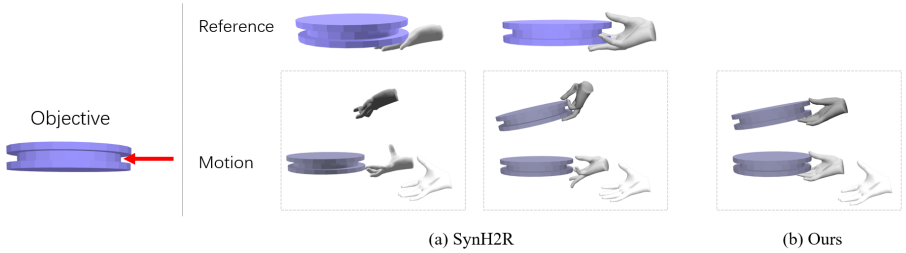

Fig. 1: Qualitative Comparison. SynH2R requires a time-consuming reference generation process, and suffers from noisy references and imperfect reference tracking, which lead to failed grasping or large objective errors.

2 Implementation Details

2.1 Training Hyperparameters

We use PPO [12] to train our policy and follow the implementation provided in [4]. We present an overview of the important parameters and weight values of the reward function in Tab. 1 and Tab. 2.

Table 1: Hyperparameters of *GraspXL*.

Hyperparameters PPO	**Value**
Epochs	1e4
Steps per epoch	3e4
Environment steps per episode	150
Batch size	2000
Updates per epoch	20
Simulation timestep	2.5e-3s
Simulation steps per action	4
Discount factor γ	0.996
GAE parameter λ	0.95
Clipping parameter	0.2
Max. gradient norm	0.5
Value loss coefficient	0.5
Entropy coefficient	0.0
Optimizer	Adam
Learning rate	5e-4
Hidden units	128
Hidden layers	2

2.2 Objectives Specification

As explained in Section 3 of the main manuscript, our framework can deal with different combinations of four kinds of objectives: the partition of graspable/non-graspable object point cloud $\{\mathbf{o}_j^+\} \cup \{\mathbf{o}_j^-\}$, the heading direction of the hand $\bar{\mathbf{v}}$, the hand wrist rotation $\bar{\omega}$, and the hand midpoint position $\bar{\mathbf{m}}$. We assume the partition $\{\mathbf{o}_j^+\} \cup \{\mathbf{o}_j^-\}$ is specified by a user to indicate the desired grasping area, such as a mug handle or a headphone earcup. By default, $\{\mathbf{o}_j^+\} = \{\mathbf{o}_j\}$ and $\{\mathbf{o}_j^-\} = \emptyset$, which means that the hand can grasp the entire object. $\bar{\mathbf{v}}$ is the only quantity that is mandatory to specify. $\bar{\omega}$ is by default 0 so that the y-axis of the hand local coordinate system (See Fig. 4 in the main manuscript) is parallel to the narrowest edge of the $\{\mathbf{o}_j\}$ projection along $\bar{\mathbf{v}}$, which represents the easiest setting for grasping with the given heading direction $\bar{\mathbf{v}}$. $\bar{\mathbf{m}}$ is by default set to be the centroid of $\{\mathbf{o}_j\}$.

Table 2: Weights of the Reward Function.

Weights	Value (1st phase)	Value (2nd phase)
w_d^+	0.3	0.3
w_d^-	0.06	0.06
$w_{\mathbf{v}}$	1.0	0.01
w_ω	1.0	0.01
$w_{\mathbf{m}}$	10.0	10.0
w_c^+	1.0	1.0
w_c^-	1.0	1.0
w_f^+	0.3	0.5
w_f^-	0.15	0.25
w_h	0.001	0.001
w_o	0.0	0.1
$w_{anatomy}$	0.2	0.1
λ	5.0	5.0

3 Experimental Details

3.1 Dataset Preprocessing

In order to compare with existing methods and to demonstrate our method's generalization capabilities, we use the three object datasets: PartNet [10], ShapeNet [1], and Objaverse [5]. Since not all objects are feasible for grasping (such as a piece of paper), we preprocess and filter the objects.

PartNet We select 80 objects from the categories scissors, knife, mug, earphone, and wineglass, which contain part-based segmentation (such as the handle and the main body of a mug). We use the individual parts of each object to specify the graspable/non-graspable areas. Objects are resized to feasible dimensions for grasping.

ShapeNet We utilize the objects of ACRONYM [6] (scaled ShapeNet [1] objects) and filter out extreme-sized objects. Specifically, we remove objects with a minimal bounding box width larger than $0.1m$ or smaller than $0.01m$, or maximal bounding box width larger than $0.3m$, or a volume smaller than $8cm^3$. This leads to 4019 objects.

Objaverse To show generalization across different scales, we resize the Objaverse [5] objects to three different scales: small, medium, and large. Specifically, for each object, we uniformly sample a small scale $s \in [3, 5]cm$, a medium scale $m \in [5, 7]cm$, and a large scale $l \in [7, 9]cm$. We then resize each object to three so that the minimum dimension of their bounding box is equal to s, m, and l, accordingly. Finally, we remove the objects with a maximal bounding box width larger than $0.3m$ or smaller than $0.05m$. This leads to 503,409 objects.

As the object meshes from the datasets contain no material information for density and friction, we calculate object masses based on the mesh volume for a given fixed density, leading to diverse masses. We use the same friction coefficient in simulation for all objects.

3.2 Reconstructed and Generated Objects

We use the eight objects generated with DreamFusion [11] which are available from their project page, and manually scale them to graspable sizes as our test set for reconstructed objects. For reconstructed objects, we use all the fourteen objects reconstructed from HO3D [8] videos and six objects reconstructed from in-the-wild videos reported in HOLD [7]. For comparison, we evaluate with the ground-truth HO3D objects with their original scales. For each object, we randomly sample 25 sets of motion objectives and report the average performance.

4 Additional Experiments

4.1 Training Set Size Effect Evaluation

To show the data efficiency of our method, we train another policy with an enlarged training set composed of 100 PartNet objects (with the graspable area partitions), and 400 ShapeNet objects. We then perform the same evaluation with the PartNet and ShapeNet test sets (See Section 4.1 in the main manuscript). The results are shown in Tab. 3. Compared with the results with a smaller training set (See Section 4.2 in the main manuscript), there is no significant improvement, which means that a training set composed of 58 objects with diverse shape is sufficient for our framework. This shows the data efficiency of our method. We hypothesize that this is because diverse shapes together with random objectives and initialization during training provide a diverse distribution of grasps and configurations.

Table 3: Comparison with Different Training Set Size.

	PartNet Test Set					ShapeNet Test Set			
Method	Suc. Rate [%] ↑	Mid. Error [cm] ↓	Head. Error [rad] ↓	Rot. Error [rad] ↓	Contact Ratio [%] ↑	Suc. Rate [%] ↑	Mid. Error [cm] ↓	Head. Error [rad] ↓	Rot. Error [rad] ↓
Ours (58 Training Objects)	**95.0**	**2.85**	**0.270**	**0.306**	86.7	81.0	**3.22**	0.292	**0.338**
Ours+ (500 Training Objects)	94.9	3.57	0.307	0.356	**88.1**	**84.3**	3.94	**0.283**	0.346

4.2 Friction Effect Evaluation

Although we use the same friction coefficient in simulation for all objects, our method can also deal with randomized frictions (± 0.3 around the default friction coefficient) as shown in Tab. 4.

4.3 Realism Evaluation

To evaluate the realism of our method, we invite 35 participants to score the realism in terms of human likeness, naturalness, smoothness, and hand-object

Table 4: Evaluation with random friction coefficients.

Settings	Suc. Rate [%] ↑	Mid. Error ↓
Original experiment	81.6	0.283
Random friction coefficient	80.5	0.285

interpenetration. They score 10 sets of rendered grasping motions from 1 (worst) to 3 (best), each containing three motions of the same object randomly chosen from ours, HO3D [9], and DexYCB [2]. The average scores are 2.12, 2.05, and 2.45, respectively. Our method has a slightly higher realism score than HO3D as HO3D has some interpenetrations and jitters caused by labeling noise, and DexYCB has the highest score due to accurate annotations.

4.4 Inference Speed Evaluation

With a 24-core Intel i9-14900 CPU and an Nvidia RTX 4090 GPU, we generate 100 sequences and calculate the average time consumption per sequence with our method, the baselines SynH2R-PD, and SynH2R [3], leading to 37.15, 37.17, and 0.14 seconds, respectively. The most time consumption for SynH2R-PD and SynH2R are caused by the optimization-based static grasping reference pose generation procedure.

References

1. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 (2015)
2. Chao, Y.W., Yang, W., Xiang, Y., Molchanov, P., Handa, A., Tremblay, J., Narang, Y.S., Van Wyk, K., Iqbal, U., Birchfield, S., Kautz, J., Fox, D.: DexYCB: A benchmark for capturing hand grasping of objects. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
3. Christen, S., Feng, L., Yang, W., Chao, Y.W., Hilliges, O., Song, J.: Synh2r: Synthesizing hand-object motions for learning human-to-robot handovers. In: IEEE International Conference on Robotics and Automation (ICRA) (2024)
4. Christen, S., Kocabas, M., Aksan, E., Hwangbo, J., Song, J., Hilliges, O.: D-Grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In: Computer Vision and Pattern Recognition (CVPR) (2022)
5. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. arXiv preprint arXiv:2212.08051 (2022)
6. Eppner, C., Mousavian, A., Fox, D.: ACRONYM: A large-scale grasp dataset based on simulation. In: International Conference on Robotics and Automation (ICRA) (2020)
7. Fan, Z., Parelli, M., Kadoglou, M.E., Kocabas, M., Chen, X., Black, M.J., Hilliges, O.: HOLD: Category-agnostic 3d reconstruction of interacting hands and objects from video (2024)

8. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: Honnotate: A method for 3d annotation of hand and object poses. In: CVPR (2020)
9. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: HOnnotate: A method for 3d annotation of hand and object poses. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
10. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: Computer Vision and Pattern Recognition (CVPR)
11. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv (2022)
12. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv:1707.06347 (2017)