Appendix

The GMD dataset and code will be released at https://github.com/SteveHao74/GMD at soon.

A Grasp Metrics

A.1 Posteriori Metrics Drawbacks

For instance, gravity and contact friction (information cannot be observed from pure visual input) of objects partially affect the grasp execution results. Once failures caused by those factors are recorded with negative labels in the dataset, the trained models will be forced to build up spurious correlations between grasp quality with other irrelevant but observable factors rather than those relevant yet unobservable factors (a kind of overfitting).

A.2 Hybrid Metric Group

The hybrid metric group consists of 36 metrics, including 16 priori metrics, 17 posteriori metrics, and 3 comparative metrics evaluating the difference between priori and posteriori metrics. The description of all the metrics is shown in the following table. Notations: i) the original grasp refer to those initial grasp candidate generated by dense sampling (introduced in Section 5.1); ii) the analytical grasp refer to those metrics calculated by intersection line analysis; iii) the simulative grasp refer to those metrics collected after the grasp trial in simulation.

Kind	**ID**	**Metrics**	**Defination**
Priori	0	bound_x	length of x-axis of object Oriented Bounding Box (OBB)
	1	bound_y	length of y-axis of OBB.
	2	bound_l	length of the shortest edge of OBB.
	3	bound_r	ratio between the length of the longest and shortest edges of OBB.
	4	g_dis	euclidean distance between the original grasp center to the center of mass of the object.
	5	g_dis_norm	g_dis metric normalized by bound_l metric.
	6	g_quality	Ferrari&Canny's L1 metric[9] (also known as ϵ-metric): the largest perturbation wrench that the grasp can resist in any direction).
	7	g_coll	whether there are collisions with the original grasp by collision detection. (binary)
	8	real_d_width	width of the analytical grasp calculated by the intersection analysis.
	9	real_d_width_norm	real_d_width normalized by bound_l.
	10	real_d_dis	euclidean distance between object center of mass and center of grasp generated by the intersection analysis.
	11	real_d_dis_norm	real_d_dis normalized by bound_l.
	12	real_d_offset_l	euclidean distance between the center of analytical grasp and original grasp.
	13	real_d_offset_a	orientation deviation between the center of analytical grasp and original grasp.
	14	real_d_offset_l_norm	real_d_offset_l normalized by bound_l.
	15	real_d_quality	epsilon quality of the analytical grasp calculated by the intersection analysis.
Posteriori	16	real_s_width	width of the simulative grasp.
	17	real_s_width_norm	real_s_width normalized by bound_l.
	18	real_s_dis	distance between the geometric center of object and center of simulative grasp.

	19	real_s_dis_norm	real_s_dis normalized by bound_l.
	20	real_s_offset_l	distance between the center of simulative grasp and original grasp.
	21	real_s_offset_l_norm	real_s_offset_l normalized by bound_l.
	22	real_s_offset_a	orientation deviation between the center of simulative grasp and original grasp.
	23	real_s_quality	the epsilon quality of the analytical grasp.
	24	clo_ori_comfort	orientation change of object during the gripper closing processes.
	25	lif_ori_stability	orientation change of object during the lifting process.
	26	moved	object shift during gripper closing process.
	27	moved_norm	object shift during gripper closing process normalized by bound_l metric.
	28	sim_width	width of actual simulative grasp attempt.
	29	sim_width_norm	width of actual simulative grasp attempt normalized by bound_l metric.
	30	sim_success	success rate of five grasp trials in simulation.
	31	sim_coll	whether there are collisions that occurred during five simulative grasp attemps. (binary)
	35	force	value of external force needed to push out the object from the gripper after being lifted.
Comparative	32	g_offset_l	distance between the center of simulative grasp and analytical grasp.
	33	g_offset_l_norm	g_offset_l normalized by bound_l.
	34	g_offset_a	orientation deviation between the center of simulative grasp and analytical grasp.

A.3 Feature Selection

To reveal the contribution of each metric in approximating the human grasp decision, a series of typical feature selection methods in traditional machine learning is imposed to give an importance ranking of all the metrics used: XGBoost(the same model in our fine screening model), random forest, low variance filter, forward feature selection, high correlation filter, backward feature elimination. Table 5 shows the top 5 important metrics ranked by each feature selection method on the expert training set. Concluding votes from all of the methods, the comprehensive importance order is given in Table 6. As we can see, the top 8 important metrics contain both priori and posteriori metrics, which imply the necessity of a combination of two kinds of metrics.

Table 5: Top 5 important metrics selected by 6 kinds of feature selection methods.

Order	**XGB**	**RF**	**LVF**	**FFS**	**HCF**	**BFE**
1	g_dis	g_dis	real_d_width_norm	real_d_dis_norm	real_d_dis_norm	moved_norm
2	sim_width_norm	real_d_dis	bound_r	real_d_dis	real_d_quality	real_d_dis_norm
3	real_d_quality	moved	real_s_quality	g_dis_norm	clo_ori_comfort	sim_width_norm
4	clo_ori_comfort	real_d_dis_norm	real_d_quality	g_dis	moved_norm	clo_ori_comfort
5	moved	g_dis_norm	g_quality	real_d_quality	g_offset_l	sim_success

Table 6: Comprehensive importance calculated by votes of 6 kinds of feature selection methods.

Order	Metric	Kind
1	real_d_dis_norm	priori
2	g_dis	priori
3	real_d_quality	priori
4	real_d_dis	priori
5	sim_width_norm	posteriori
6	clo_ori_comfort	posteriori
7	moved_norm	posteriori
8	real_d_width_norm	priori

A.4 Metrics Correlation

Since we take grasp metrics as the features for the grasp evaluator, it is important to analyze the correlation between the features to examine the diversity and completeness. Thus, the Pearson correlation indexes among the hybrid metric group are calculated and visualized in Figure 7.

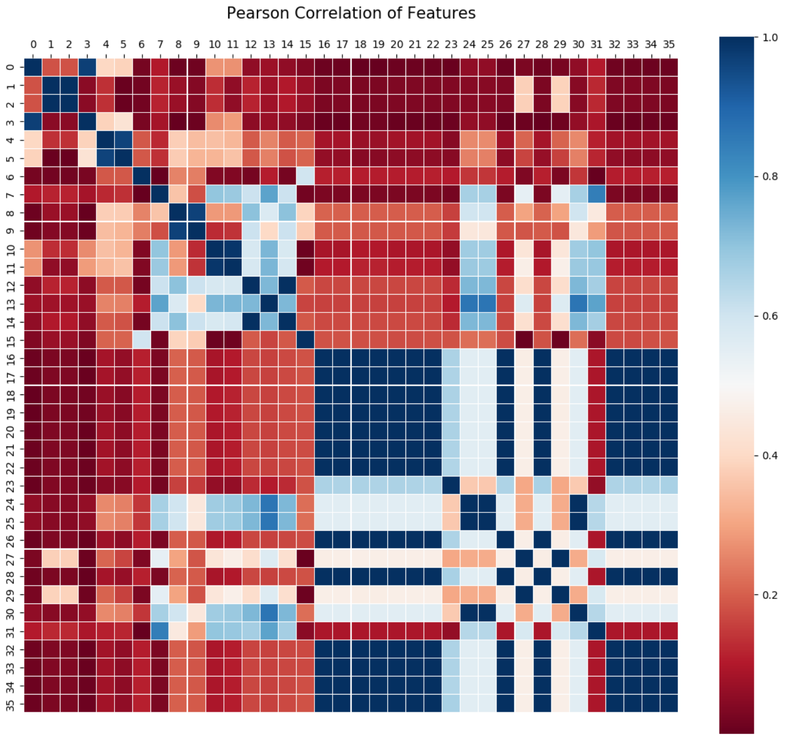

Figure 7: Pearson correlation indexes within the metric group. The corresponding metric id can be found in the long table of Appendix A.2 .

B Grasp Evaluator

To train the three kinds of machine learning models, we split the expert set into a training set(90%) and a testing set(10%). For the training set, we first calculate the mean value and deviation value for each metric and then normalize different metrics into the same scale. Finally, we use the GridSearchCV module from the scikit-learn tool to automatically search the best hyper-parameter configurations for each model. The final parameters of each model will be introduced in the following subsections.

B.1 Expert Set Collection

We select 50, 13, 63, 50 and 80 objects from 3D-NET [27], Adversarial [5], test set of EGAD [28], Kit [29], Shapenet [30], for a total of 256 object CAD models. These object models are densely sampled and grasp metrics proposed in Section 3.1 are calculated for each grasp candidate with a depth image rendered from the top-down view. Then multiple experts are invited to manually label each image with the best grasp in their confidence through a self-developed annotation tool. During the labeling process, high precision rather than recall is what we pursue following human intuition to do more conservative decisions. The final expert set consists of 256 depth images and 2400 grasp annotations.

B.2 Evaluator models

B.2.1 Decision Three

We have improved the decision tree model in the following two aspects: i) data entropy considering sampling cost: In the process of data entropy calculation during tree generation [31], we added the metric sampling cost $C(a)$ as punishment, which was the average calculation time of each metric on the Intel i9-9900K with a single CPU core. The modified formula is

$$Gain\_ratio(D, a) = \frac{Gain\left(D, a\right) - \beta C\left(a\right)}{IV\left(a\right)} \tag{3}$$

ii) maximum recall pruning: As the coarse screening model, the decision tree is hoped to quickly exclude as many as possible negative samples. To avoid false-negative cases, we prune the tree model based on the maximum recall principle. For each leaf node in the tree model with a negative label, if there is at least a positive sample in the validation set wrongly classified, pruning will be conducted and the leaf node returns to its parent node. Executing the pruning through recursion, until there isn't any false-negative case on the validation set. The final hyper-parameters of the decision tree are shown in Table 7.

Table 7: Hyper-parameters for decision tree training.

min_samples_split	**min_samples_leaf**	**textbf_criterion**
2	4	improved informative gain ratio
min_impurity_decrease	**ccp_alpha**	
0	0	

B.2.2 SVM

The final hyper-parameters of SVM model are shown in Table 8.

B.2.3 XGBoost

The final hyper-parameters of XGBoost model are shown in Table 9.

Table 8: Hyper-parameters for SVM training.

kernel	C	gamma	accuracy
RBF	12.30088027	0.06482219	0.959169464

Table 9: Hyper-parameters for XGBoost training.

n_estimators	max_depth	learning_rate	subsample
105	4	0.52	0.733333333
colsample_bytree	**min_child_weight**	**accuracy on testing set**	
0.713333333	1	0.970462163	

C Dataset Annotation Framework Details

C.1 Objects Preparation and Simulation Scene Generation

We use 20k object models collected from SHREC [32], YCB [33] and DeX-Net1.0 [11], 8k object models from Shapenet [30] and EGAD [28] as the object sources to generate GMD dataset. For each object, we conduct the following preprocessing: i) the shortest side of the bounding box on the xoy plane in all stable poses is limited to 5 cm through scaling, which ensures available places for grasp remained in the object; ii) Origin of the object coordinate system is translated to coincide with the center of mass; iii) mass is unitedly set as 0.1kg and the contact friction coefficient is set as 0.8. The Pybullet physics simulator [23] is used to build up a grasping platform for the subsequent grasping simulation and datasets synthesis. Each object model is first loaded and conducted with a free fall over the platform in a random initial posture. Then the stable pose and position after landing will be used for subsequent grasping sampling, metrics calculation, and depth image rendering.

C.2 Annotation Cost Analysis

The computation cost of our hybrid metrics is intermediate between the pure priori and pure posteriori metrics methods. For the Intel i9-9900K we used, it takes about 3.7 seconds to compute all 36 metrics for each grasp using a single CPU core. The computation cost of our method is relatively low compared with the other two kinds of grasp annotation methodologies (hand-annotated and physical trial) as described in Figure 1. While among the methods of metrics-based methodology, the computation cost of our hybrid metrics-based method is unavoidably higher than those pure priori metrics-based methods, since they have sacrificed the accuracy and information completeness at the cost.

D Grasp Detection Models Training

D.1 Hyper-parameters setting about Model Training

Since the scale of Cornell is far smaller than both Jacquard [6], Dex-Net2.0 [5] and GMD, we use the online data augmentation strategy to randomly rotate and shift each sample. Then we increase training epochs during model training on Cornell to make sure that the Cornell dataset has equivalently the same different samples as other datasets. GQCNN [5] model training hyper-parameters are shown in Table 10. GR-ConvNet [26] model training hyper-parameters are shown in Table 11.

D.2 More Details about GR-ConvNet Training

GR-ConvNet belongs to the pixel-wise grasp parameter prediction methodology, and the model training needs the object-wise heatmaps as supervision. Thus, each training sample for GR-ConvNet (a heatmap) is synthesized by all of the feasible grasps of a specific object. Therefore, the number of objects is controlled with the number of training samples to be equal across different datasets. Notations: As for the case that there are different render views of the same objects as samples in the

Table 10: Hyper-parameters for GQCNN training.

initial learning rate	**optimizer**	**momentum rate**	**L2 regularizer**
0.015	momentum	0.9	0.005
batch size	**metric threshold**	**drop out**	**decay step multiplier**
16	0.002	0	0.2
decay rate			
0.95			
training set	**batches per epoch**	**epoch**	**augmentation**
Cornell	100	250	1
Jacquard	500	50	0
Dex-Net2.0	500	50	0
GMD	500	50	0

Table 11: Hyper-parameters for GR-ConvNet training.

initial learning rate	**optimizer**	**batch size**	**drop out**
0.001	Adam	8	0
GR-ConvNet	**batches per epoch**	**epoch**	**augmentation**
Cornell	99	200	1
Jacquard	1416	15	0
GMD	1416	15	0

dataset, we think that different views of the same object can be roughly regarded as different objects in the level of geometric diversity and affordance since the partial geometric information recorded from different views will be different (especially for those irregular objects).

E Experiment Setup

E.1 Real World Experiment Setup

For testing set, there are 25 objects, including 9 simple objects with regular geometric, 7 3D-printed objects with adversarial geometric [5], 8 unseen household objects close to YCB [33] and APB [34] object sets. The experiment platform is built up with UR5 manipulator [1] with Robotiq 2f-85 gripper[2] and an Intel RealSense D435i RGB-D Camera[3] with eye-in-hand configuration(shown in Figure 4.C).

F GMD Dataset Visualization

Our dataset is composed of depth images for each scene and corresponding grasp annotation described under 6 image coordinates. Part of the GMD dataset is uploaded here, the rest of them will be released after the paper acceptance. We compress the dataset into npz format. Numpy tools can be used to extract the grasp annotation and images. The visualization of GMD dataset samples is shown in Figure 8.

[1] https://www.universal-robots.com/products/ur5-robot/

[2] https://robotiq.com/products/2f85-140-adaptive-robot-gripper

[3] https://www.intelrealsense.com/depth-camera-d435i/

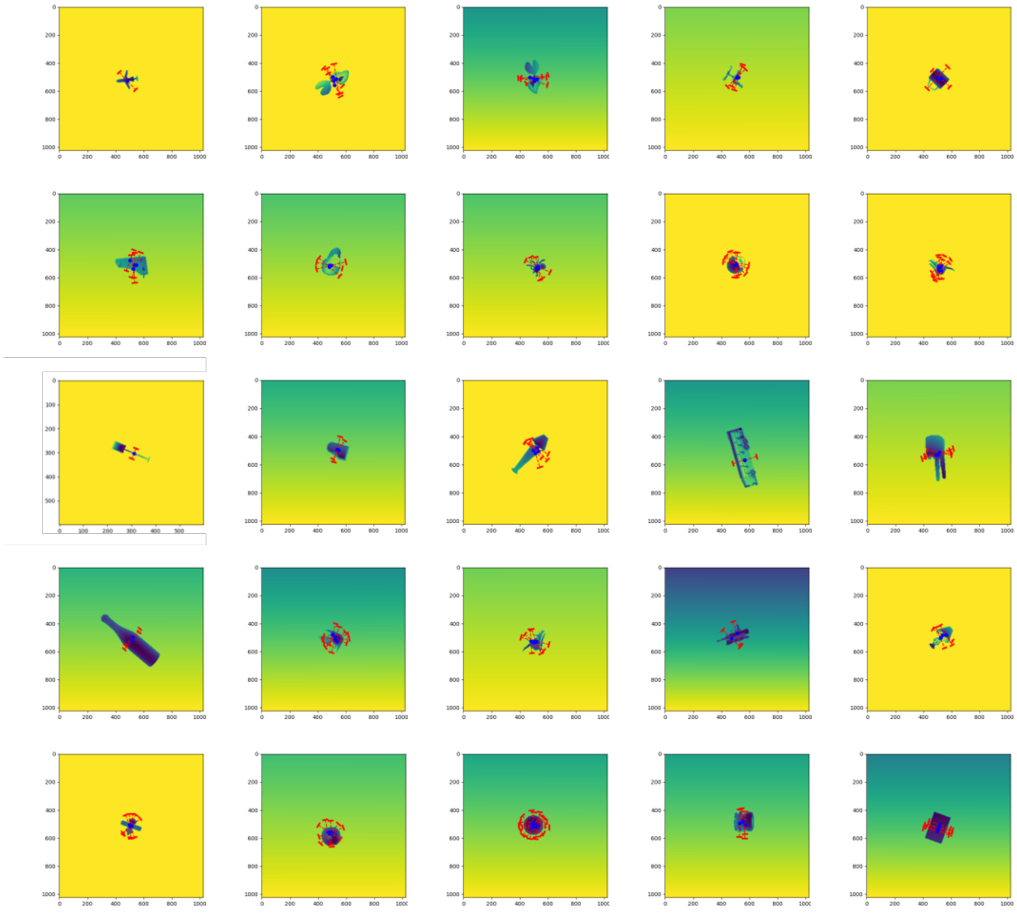

Figure 8: Visualization of GMD dataset.