Improving Robotic Grasping Ability Through Deep Shape Generation

Junnan Jiang[1], Yuyang Tu[2], Xiaohui Xiao[3], Zhongtao Fu[4], Jianwei Zhang[2], Fei Chen[†5], Miao Li[†1]

Abstract—Data-driven approaches have become a dominant paradigm for robotic grasp planning. However, the performance of these approaches is enormously influenced by the quality of the available training data. In this paper, we propose a framework to generate object shapes to improve the grasping dataset quality, thus enhancing the grasping ability of a pre-designed learning-based grasp planning network. In this framework, the object shapes are embedded into a low-dimensional feature space using an AutoEncoder (encoder-decoder) based structure network. The rarity and graspness scores are defined for each object shape using outlier detection and grasp-quality criteria. Subsequently, new object shapes are generated in feature space that leverages the original high rarity and graspness score objects' features, which can be employed to augment the grasping dataset. Finally, the results obtained from the simulation and real-world experiments demonstrate that the grasping ability of the learning-based grasp planning network can be effectively improved with the generated object shapes.

Index Terms—Data Augmentation, Shape Generation, Robotic Grasping, Feature Embedding

I. INTRODUCTION

Grasping is a fundamental ability for robots. Despite significant progress in the area of grasp planning, it is still a difficult task to plan a grasp for unknown objects in general [1], [2]. During the past decade, data-driven approaches have demonstrated significant advantages over traditional model-based approaches [3] in the grasp planning area. Since the performance of data-driven approaches is greatly limited by the quality of the available training data, how to improve the quality of training data is one of the crucial questions to improve the grasping ability.

The intuitive idea is to expand the size of the training dataset. Numerous grasping datasets containing more shapes, more annotated grasps and more sensory information have been proposed [4]–[9] and are trying to include more diverse data for network training. However, except for expensive and time-consuming dataset collection, a well-trained grasp
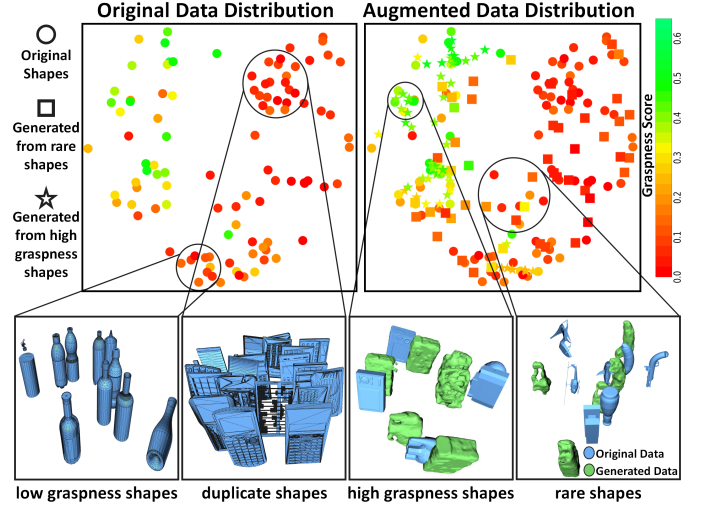

Fig. 1. Original and augmented data distribution comparison. We use t-SNE [12] to project the shape feature vectors to a 2D plane, where the Euclidean distances between scattered points represent their features' similarity, and the colors represent their graspness scores. We generate new data that leverages the features of original high rarity and graspness shapes to improve the quality of the dataset.

planning network on existing datasets may still fail when facing some special pre-grasped objects. Simply adding a small amount of "failed objects" to the original large amount of training data is difficult to improve the quality of the training dataset effectively, which further leads to only a small improvement in the grasping ability of the grasp planning network.

Another more efficient approach is data augmentation. Similar to computer vision, there are also some data augmentation methods applied in grasping datasets using random image transformation or shape generation [10], [11]. However, the effect of these random operations on grasping ability improvement is not certain, and these operations also do not take into account the grasp property. Furthermore, for a grasping dataset shown in Fig. 1, where the Euclidean distances between scattered points represent their shape similarity and the colors represent their graspness scores defined in IV-A, the data distribution may be duplicated in object shapes and uneven in different graspness scores. Randomly augmenting the training data, i.e., generating new data using the features of all data equally, may lead to a larger amount of duplicate shapes and higher uneven distribution of these shapes' graspness scores, which may result in grasp planning network overfitting on these duplicate and uneven data and cannot generalize to other unseen data.

[1] Junnan Jiang and Miao Li are with the Institute of Technological Sciences, Wuhan University, Wuhan 430072, China. (e-mail: jiangjunnan@whu.edu.cn, miao.li@whu.edu.cn)

[2] Yuyang Tu and Jianwei Zhang are with the Department of Informatics, University of Hamburg, Hamburg 20146, Germany

[3] Xiaohui Xiao is with the School of Power and Mechanical Engineering, Wuhan University, Wuhan 430072, China.

[4] Zhongtao Fu is with the School of Mechanical and Electrical Engineering, Wuhan Institute of Technology 430205 Wuhan, China

[5] Fei Chen is with the Department of Mechanical and Automation Engineering, The Chinese University of HongKong, Hongkong 999077, China

To address these issues, we augment the dataset for the purpose of making it more diverse, and generate data leveraging the features of rare data in the original dataset. In detail, an AutoEncoder-based network is firstly proposed to encapsulate object shape information into a low-dimensional feature space. In this feature space, shapes in the original dataset can be effectively encoded, interpolated and generated. Moreover, in this low-dimensional feature space, two grasp-related metrics are proposed to find the rare data in the grasping dataset. Finally, the features of these rare data are used to generate new shapes that can further improve the quality of the original dataset and thus improve the grasping ability of a pre-defined learning-based grasp planning network. The comparison between the original and augmented data distribution is shown in Fig. 1: Our newly generated shapes fill the vacant area of the original data distribution and cause a more even graspness score distribution.

The main contributions of this paper are summarized as:

- An AutoEncoder-Critic network is proposed to map a voxelized shape into a low-dimensional feature space.
- Two grasp-related metrics are proposed to find the rare data in the grasping dataset.
- A systematic approach is proposed to generate new shapes that can improve the grasping ability of a pre-designed learning-based grasp planning network.

The remainder of this paper is structured as follows. Section II presents related work in grasping datasets and data augmentation. Section III describes the methodology of our whole data augmentation pipeline, including an object shape encoding method for shape generation, two grasp-related metrics for shape selection, and an augmentation method by the generated object shapes. Section IV describes both simulation and real-world experiments, with a final discussion and conclusion in Section V.

II. RELATED WORK

A. Grasping Dataset

With the great success achieved by data-driven grasp planning methods [1], many grasping datasets have been proposed [4]–[9]. Though existing grasping algorithms can perform well on one dataset, they may still fail when facing unseen objects. Since optimizing the grasping network architectures requires expert experience, it is more desirable to improve the grasping dataset quality and retrain the network, such as expanding the dataset with "failed objects" or performing some data augmentation tricks. But it is still difficult to answer whether newly added shapes will improve the dataset quality in terms of enhancing the grasping ability. To solve this problem, the EGAD dataset [8] methodically generates shapes with a richer range of shape complexity and grasp difficulty. However, since the shapes generated in EGAD are very different from existing real-world shapes, it is more difficult for a pre-defined network to learn a good grasping policy with an EGAD dataset than with a real-world dataset. This means that the EGAD dataset can only be used for evaluation, but is difficult to apply in real-world scenarios.

B. Data Augmentation

Data augmentation [13], [14] is a prevalent approach in data-driven approaches because it effectively improves the quality of the dataset and reduces network overfitting problems. Handcrafted methods [15] have been widely used in computer vision, such as shifting, scaling and rotating. Similarly, randomly rotating and cropping images [10] or randomly combining different shapes to generate a new shape [11] can also be used to augment the grasping dataset. However, since it is difficult to evaluate what these random operations bring to the dataset, the effect of these dataset augmentation methods can only be known after retraining the network on the augmented dataset.

With the development of generation methods such as AutoEncoder [16] and generative adversarial network (GAN) [17], data can be generated more flexibly and even with specific objectives. DeVries et al. [18] augment different-domain data in feature space only with the same AutoEncoder-based network. Wang et al. [19] generate adversarial grasp objects by evaluating the generated objects' grasp difficulty and regularizing the generation network to generate difficult-to-grasp objects. Mitrano et al. [20] formalize data augmentation as an optimization problem and propose some objective functions to sample better generated data for manipulation tasks. Inspired by these methods, in this work we focus on grasping dataset augmentation and generate data similar to the rare data in a grasping dataset, thereby improving the grasping ability for a pre-defined network.

III. OBJECT SHAPE ENCODING

To leverage shapes' features of the original dataset and to ensure that generated shapes are more realistic, we propose an AutoEncoder-Critic (AE-Critic) network. The AutoEncoder (AE) can embed object shapes into a low-dimensional feature space to better generate shapes by interpolation, and the Critic can regularize the generated shapes to be more realistic.

A. Network Architecture

In order to generate new shapes leveraging the features of the original object shapes, we propose an AE-Critic network and show its structure in Fig. 2. We use a voxel grid to represent a shape, which maps a shape to a $64 \times 64 \times 64$ binary matrix. The AutoEncoder [16] of AE-Critic contains an Encoder and a Decoder. The Encoder maps a voxel grid x to a 128-dimensional feature vector z, containing five 3D convolution layers and two fully connected layers. The convolution layers use $4 \times 4 \times 4$ kernel size and two strides, with batch normalization and ReLU layers added in between, mapping the voxel grid to a $512 \times 4 \times 4 \times 4$ sized feature map. The fully connected layers have 32768 and 128 neurons separately, and map the feature map to a 128-dim feature vector. The Decoder mirrors the Encoder, maps a 128-dim feature vector to a $64 \times 64 \times 64$ reconstructed voxel grid $\hat{x}$, and contains two fully connected layers and five 3D transposed convolution layers [21]. The layer configurations are the same as the Encoder. Using AutoEncoder, we can generate new shapes leveraging the original shapes' features by changing

their feature vectors, like interpolating between real samples. In detail, based on the formula $z_{mix} = \alpha z_1 + (1-\alpha)z_2$, we can obtain z_{mix} by interpolating two feature vectors z_1, z_2, which are encoded by two shapes x_1, x_2, with the interpolated weight α. Then we decode the mixed feature vector z_{mix} to generate an interpolated shape $\hat{x}_\alpha$. Theoretically, by changing the interpolated pairs x_1, x_2 and weights α, we can generate infinite interpolated shapes $\hat{x}_\alpha$.
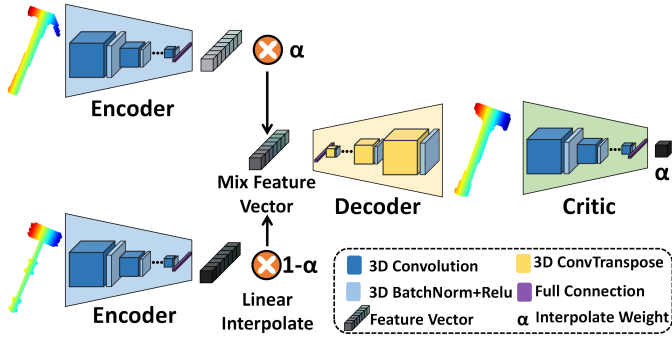

Fig. 2. AE-Critic network architecture. The Critic tries to estimate the interpolated weight α corresponding to an interpolated shape and thus can regularize the AutoEncoder (Encoder+Decoder) to generate more realistic interpolated shapes by fooling the Critic to output a smaller interpolated weight.

In addition, to make the shapes generated by interpolation more realistic, we add a Critic inspired by Berthelot et al. [22], which aims to estimate the interpolated weight α corresponding to an interpolated shape $\hat{x}_\alpha$, and to regularize the training process of the AutoEncoder. In detail, the AE is trained to generate shapes to fool the Critic to output a smaller interpolated weight, which means that the Critic is more willing to consider the interpolated input as a non-interpolated original shape. Therefore, more realistic shapes that are similar to the original shapes can be generated. The structure of the Critic is similar to the Encoder, only with one more fully connected layer to map the 128-dim feature vector to a 1-dim interpolated weight.

B. Network Training

The Critic is trained to minimize the loss function given in Eq. (1), its first term trains the Critic C to recover interpolation weight α from interpolated voxel grid $\hat{x}_\alpha$ with mean square loss. For the second term, γ is a scalar hyperparameter and $\hat{x}$ is a reconstructed voxel grid by AutoEncoder from an original voxel grid x. This enforces the Critic C outputs 0 for non-interpolated inputs, which means the mixture of the original and reconstructed voxel grid in shape space but not in feature space, thus making the Critic's training process more stable.

$$L_C = ||C(\hat{x}_\alpha) - \alpha||^2 + ||C\{\gamma x + (1-\gamma)\hat{x}\}||^2 \quad (1)$$

The AutoEncoder is trained to minimize the loss function given in Eq. (2), where L_B represents binary cross-entropy loss and λ is a scalar hyperparameter to balance the magnitude of the two loss terms. Its first term trains the AutoEncoder to generate a reconstructed voxel grid $\hat{x}$ similar to the original voxel grid x, the same as the original AutoEncoder training

process. The second term, serving as a regularization term, encourages the interpolated voxel grid $\hat{x}_\alpha$ to fool the Critic to output 0, which means that the Critic considers the generated interpolated voxel grids to be realistic non-interpolated voxel grids. And this can finally regularize the AutoEncoder to generate more realistic voxel grids.

$$L_{(E,D)} = L_B(\hat{x}, x) + \lambda||C(\hat{x}_\alpha)||^2 \quad (2)$$

After the training process, for each object represented as a voxel grid, our AE-Critic network can map it to a 128-dim feature vector. To better visualize the distribution of all the feature vectors, we use t-SNE [12] to project feature vectors to a 2D plane. The distribution of shapes can be seen in Fig. 1. As shown, it is clear that similar shapes are located close to each other. Moreover, all the shapes are not uniformly distributed in the space and there are many vacant areas, which means no shapes are located there. These areas can be filled by interpolated shapes. The next chapter will explain in detail which vacant areas need to be filled to improve the grasping ability of a given dataset.

IV. DATA AUGMENTATION FOR ROBOTIC GRASPING

The purpose of improving the quality of the grasping dataset to improve the grasping ability is to allow the grasp planning network to learn more diverse data. Therefore, we first define rarity and graspness metrics for shapes, where the higher the metric score, the rarer the data. Then, through the AE-Critic network, new shapes are generated using the features of high-scoring data, which are further used to augment the original dataset. The whole grasping dataset augmentation pipeline is shown in Fig. 4.

A. Shape Rarity and Graspness Metrics

Shape Rarity:
We assume that rare shapes are those whose features are distinct from others. Thus, we use outlier detection [23] to evaluate each feature vector. For one shape which is rarer, the score of outlier detection is higher. In detail, we use the Euclidean distance $dist$ between two feature vectors to measure their similarity. For one shape O, we select its k-nearest shapes $N_k(O)$, and the local reachability density D can be defined by Eq. (3), which is the reciprocal of the mean of distances between feature vectors from shape O to its k neighbors. This can be used to measure the density around each shape; higher scores indicate greater density.

$$D(O) = \frac{1}{\frac{1}{k}\sum_{P \in N_k(O)} dist(O,P)} \quad (3)$$

With the local reachability density metric D, we can compute the score of local outlier factor R for each shape's rarity score by Eq. (4). It is the average ratio of the D of each shape O to the D of its k nearest neighbors P. A lower object O's D score and higher D scores of its neighbors will result in a higher R score, indicating that shape O is rarer.

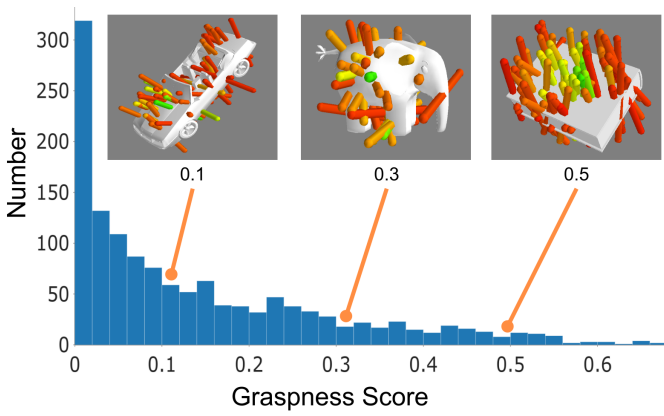$$R(O) = \frac{1}{k} \sum_{P \in N_k(O)} \frac{D(P)}{D(O)} \quad (4)$$

Fig. 3. The graspness score histogram of all objects in the 3dnet dataset [24]. Three objects with different graspness scores are shown above, each cylinder representing an antipodal grasp. The color of the cylinder indicates the grasp quality of each grasp, ranging from red to green.

Shape Graspness:

We define a shape's graspness score as the level of difficulty to find a stable grasp for an object. Firstly, using the Dex-Net analytical grasp planner [6], a number of antipodal grasps on a shape's surface can be sampled. Then, we use robust Ferrari-Canny [3] to compute each grasp quality Q:

$$Q = \min_w LQ(w) \tag{5}$$

where LQ is a local quality metric that measures how efficiently a given wrench w can resist disturbances given applied forces f and the approximated friction cone FC:

$$LQ(w) = \max_f \frac{||w||}{||f||} \tag{6}$$
$$\text{s.t.} \quad f \in FC$$

Finally, we use a threshold of 0.002 [25] for the grasp quality to distinguish whether a grasp is successful or not, and define the proportion of successful grasps of an object to all its sampled grasps as its graspness score. The lower the graspness score, the harder the object is to grasp. Fig. 3 is the histogram of graspness scores of all objects in the 3dnet dataset [24]. The lack of objects with a high graspness score verifies that they are insufficient and need to be generated.

B. Shape Generation

Based on the AE-Critic network and two defined metrics, we can finally generate shapes leveraging the features of insufficient data to augment the dataset. The overall pipeline is shown in Fig. 4.

After calculating the rarity and graspness scores of all objects, we only select every two objects whose scores are higher than $t\%$ of all scores in each metric as a generation pair. Then we will have to avoid the feature vectors of shapes in generation pairs being too close, causing shapes to be duplicated, or too far apart, which would cause the intermediate properties to disappear. As this would mean that the intermediate interpolated shape's properties are not similar
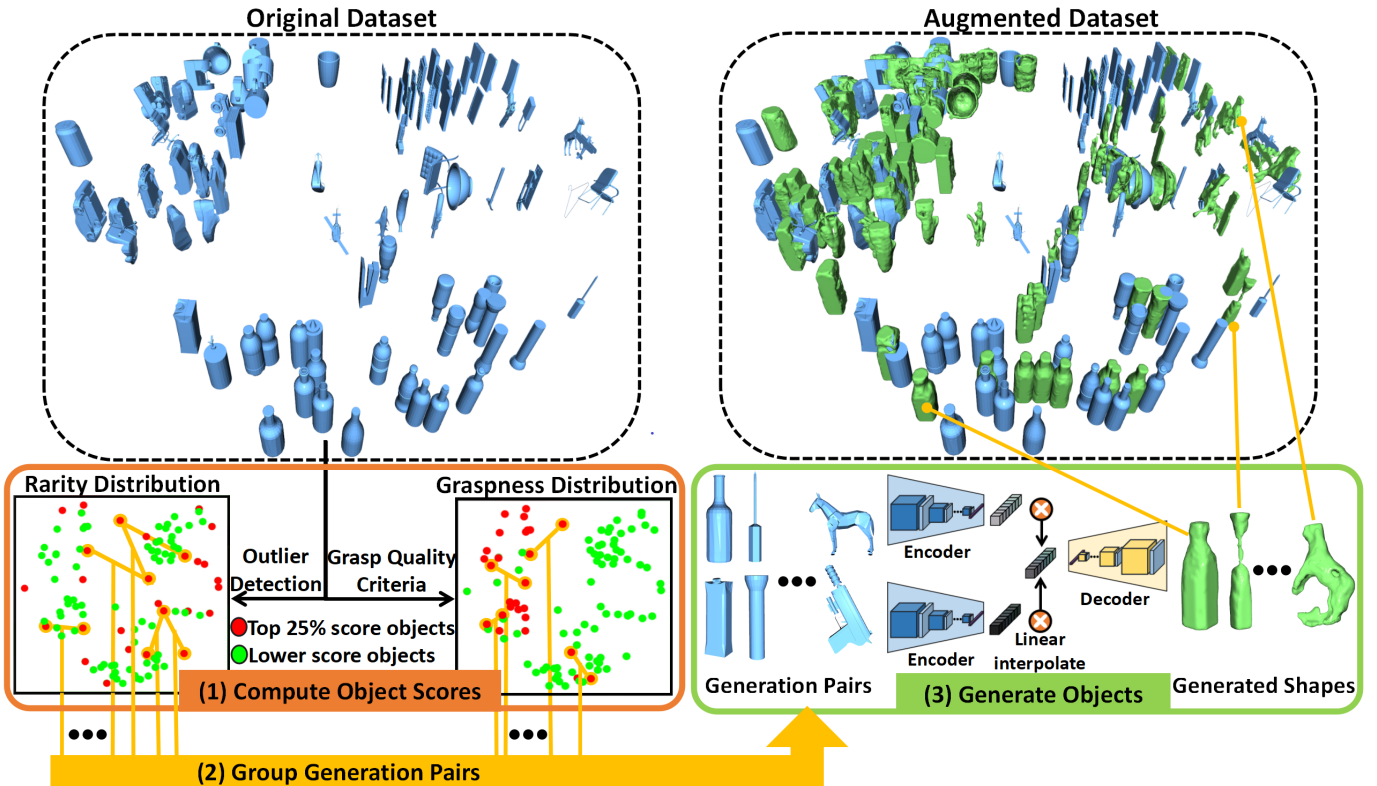

Fig. 4. The whole pipeline of shape generation for grasping dataset augmentation. The original shapes' rarity and graspness scores are firstly computed through outlier detection and grasp-quality criteria. Then, every two high-scoring nearby objects are grouped as a generation pair. Finally, the AE-Critic network is used for shape generation through interpolation between two shapes' feature vectors.

to the generation pair's properties, we group the nearest N-th to $(N+K)$-th neighbors into generation pairs. With these generation pairs, we linearly interpolate with interpolation weight α between each generation pair's feature vectors and decode the mixed feature vector to a newly generated shape. The generated shapes, represented in a voxel grid, are converted to a triangle mesh representation using marching cubes [26] and smoothing [27].

Up to this point, we leverage the features of shapes with high rarity and graspness score to generate new shapes and get a higher quality grasping dataset. The generated number of augmented shapes depends on the parameters t, N, K and α.

V. EXPERIMENTS

A. Experiment Setup

All our experiments are based on the 3dnet dataset [24] and the GQ-CNN [6] grasp planning algorithm. Since our augmentation method only expands the amount of shapes, other grasping datasets and grasp planning networks can also be used. We randomly select 1000 shapes as the training dataset and 363 shapes as the test dataset. Considering the network is needed to classify the uneven distribution of successful and failed grasps, we use the Average Precision (AP) score on the test dataset to measure the performance of a network. To evaluate the augmentation effect in real-world applications, we also set up a grasping system with a Franka Emika Panda robot arm and an Intel Realsense D435 depth camera.

B. Critic Regularization Effect

To compare the effect of Critic regularization on shape generation, we evaluate the generated shapes' completeness by AE and AE-Critic networks. In detail, we first train the AE network on 3dnet using the Adam [28] optimizer with a 0.001 learning rate and use the trained AE parameters as a pre-trained network for AE-Critic, and train AutoEncoder and Critic in AE-Critic with a 0.0001 and 0.001 learning rate, respectively. Then, we perform DBSCAN [29], a point cloud clustering method, for each shape to obtain all their clusters. All points in one cluster are contiguous, and the cluster with the highest number of contiguous points is considered to be the major part of a shape, while the points in the other clusters are considered to be outlier points. The percentage of outlier points to all points in a shape is used to evaluate its completeness; the lower the outlier percentage, the more complete the shape is. Finally, we generate 409 shapes with different interpolated weights from 200 randomly selected shapes in the 3dnet dataset, and calculate the percentage of the outlier in the generated shapes. The outlier percentages of two networks based on different interpolated weights are shown in Fig. 5.

The results show that higher interpolated weights lead to higher outlier percentages, but the Critic regularization method can reduce the percentage of outlier points in the generated objects. The generated objects are shown in Fig. 6, with the major part of the shapes in green, and their outlier points in red. And in each set of interpolated shapes, the top row is
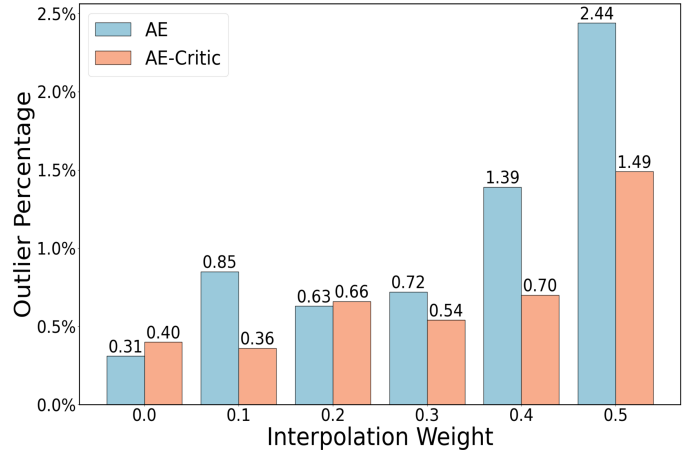

Fig. 5. Generated shapes' outlier percentage comparison between AE and AE-Critic network on different interpolated weights. Due to the symmetry of interpolation, we only generate shapes with interpolated weights in the range $[0, 0.5]$.

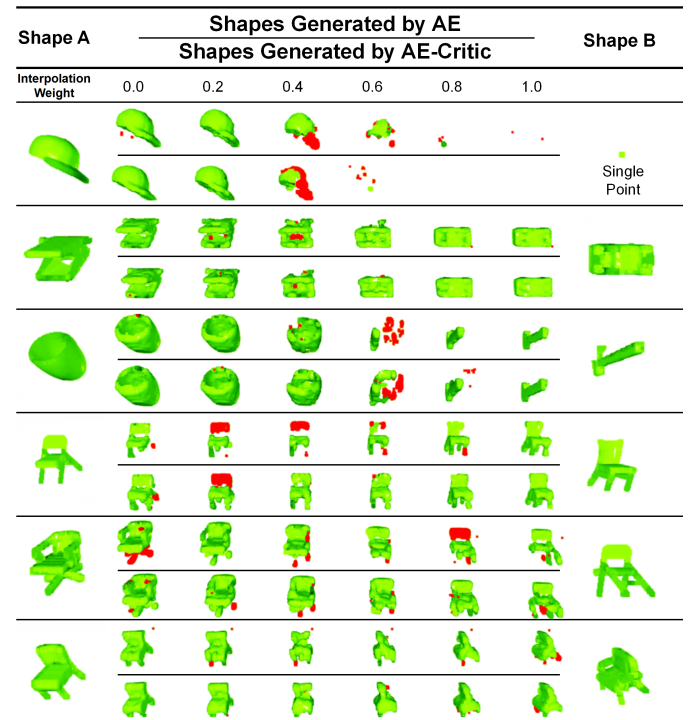generated by AE, and the bottom row is generated by AE-Critic.

Fig. 6. Example of interpolated data from 3D-Net [24], produced by AE and the AE-Critic network. The red indicates the outlier points and the green indicates the major part of the shape. In each set of interpolated objects, the top row is generated by AE and the bottom row is generated by AE-Critic.

C. Augmentation Ratio and limitation

To find the optimal augmentation ratio and augmentation limitation, we randomly select 50, 100, and 200 shapes from 1000 training data, and generate new shapes with 1:0, 1:0.5, 1:1, 1:1.5, and 1:2 augmentation ratios. This means that for 50 original shapes, 0, 25, 50, 75, and 100 generated shapes similar

to both in high rarity and graspness score shapes are used for augmentation, the same as 100 and 200 shapes. Then the whole 15 augmented datasets are used for GQ-CNN training, and the 15 trained GQ-CNN networks are tested through the same test dataset mentioned in Section V-A, and their corresponding AP scores are shown in Fig. 7.
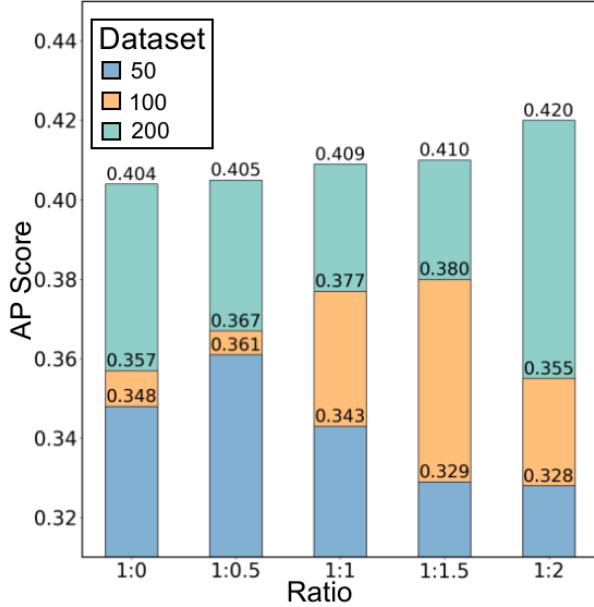

Fig. 7. 15 GQ-CNN networks trained on 15 augmented datasets and their corresponding AP scores on the same test dataset. The 15 datasets are augmented by different augmentation ratios and the amount of original datasets.

Although the results show that our augmentation methods can improve the accuracy of the network on the test dataset, the augmentation ratios allowed by different amounts of data are different. The 50, 100, and 200 data achieve the highest AP values at 1:0.5, 1:1.5, and 1:2, respectively. This means that blindly increasing generated data will lead the network overfitting to the generated data and cause a bad performance on the original dataset.

D. Improvement from Generated Data

To see the detailed changes brought from our generated data, we compare the correlation between the selected data for data generation, the newly generated data, and the overall network AP improvement on the test dataset caused by training with the augmented data. Specifically, selected data is the data with the top 25% rarity or graspness scores from the randomly selected 200 shapes in the 3dnet dataset. Generated data is the data generated by leveraging the features of the selected data and using them for data augmentation. 190 and 219 data are generated from the selected high-scoring rarity and from graspness data separately. Both the selected and generated data's distribution histogram of rarity and graspness scores are computed for visualization. We also calculate the AP improvement value of each object with an amount of 363 in the test dataset mentioned in Section A. The AP improvement refers to the improvement of the network on the test dataset after training the GQ-CNN [6] with the augmented data. We

sort them into different rarity and graspness score intervals and calculate the average AP score of objects in each score interval. Thus, we can plot the distribution histogram of AP improvement relative to rarity and graspness scores. For the convenience of visualization, all histograms normalize the total number of their distribution to 1 and are plotted together in Fig. 8.
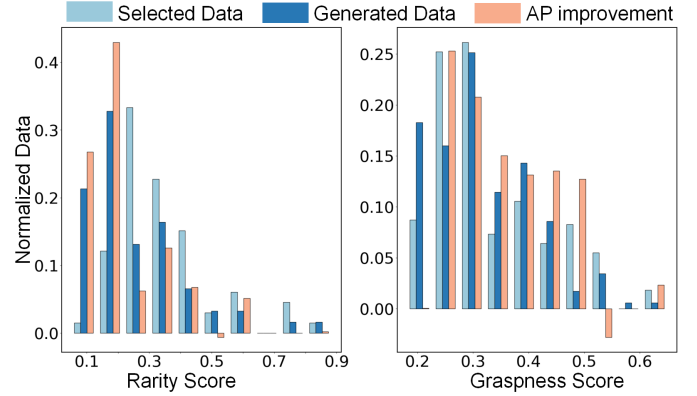

Fig. 8. The histograms between selected high-scoring data, generated data and AP improvement on the test dataset with different rarity or graspness scores.

The histograms show that more selected data will result in more generated data with the same rarity or graspness score, which means that the generated data has the same property as the original selected data to a certain extent. And the generated data at the same time will lead to a greater AP score improvement.

E. Real-world Validation

To validate the augmentation effect in the real-world applications, we augment 200 original shapes with 409 generated shapes. Both original and generated shapes are the same in Section V-D. Then two GQ-CNN [6] networks are trained on the before and after augmentation dataset, and deployed
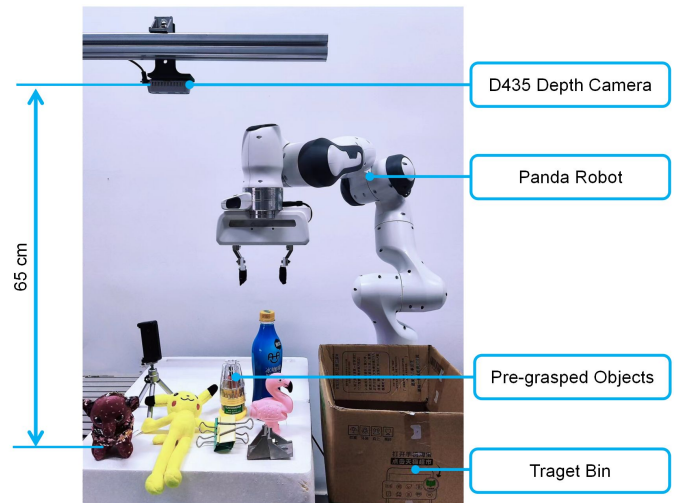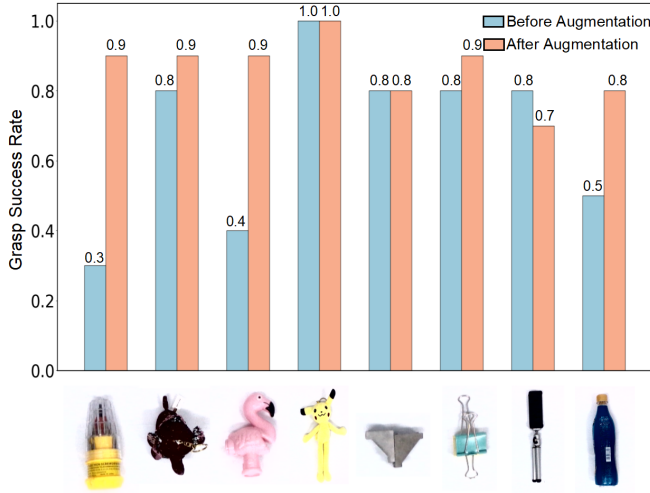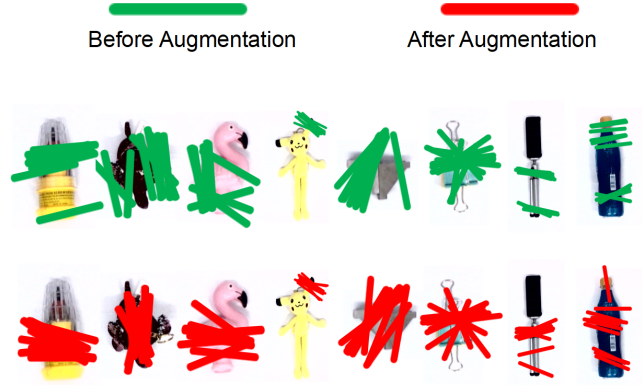

Fig. 9. The robotic grasping system.

(a) Grasp success rate comparison

(b) All grasp attempt results

Fig. 10. Real-world experiment results. Fig. 10a compares the success rate of each object's 10 grasp attempts between, before, and after the augmentation GQ-CNN network. [6]. The average grasp success rate increases from 68% to 86%. Fig 10b shows each object's 10 grasp attempts' results produced separately before and after the augmentation GQ-CNN network. Green and red lines indicate the grasp attempt results before and after the augmentation GQ-CNN network.

in the grasping system shown in Fig. 9. The D435 depth camera is fixed 65cm above the grasping platform and eight everyday objects are selected for grasping. A Panda robot performs 20 grasp attempts for each object, ten attempts before and ten after the augmentation GQ-CNN network. For each grasp attempt, a depth image is captured from a fixed viewpoint and the object is placed in the same pose. A grasp is considered successful only when the object is grasped and placed in the target bin. The ten times grasp success rate for each object before and after the augmentation GQ-CNN network is shown in Fig. 10, and the grasping process is shown in Fig. 11. All the experimental videos are available at https://youtu.be/Pn6tpSVu5aU. Experimental results show that the average grasp success rate increases from 68% to 86% using our augmentation method, and validates our augmentation method in real-world scenarios.
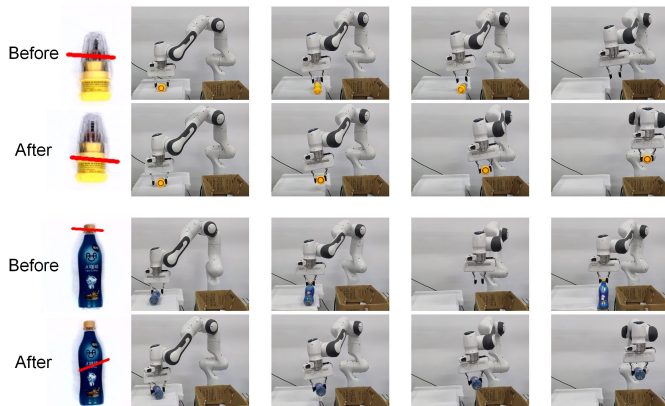

Fig. 11. Grasp process comparison between before and after augmentation GQ-CNN network [6].

VI. DISCUSSION AND CONCLUSION

A. Discussion

Although our generated shapes can improve the network's grasping ability in both simulation and real-world experiments, there are still some limitations to this paper. First, in terms of shape generation, either the network structure, shape representation method or feature vector dimension may not be optimal, and a better generation method may result in more realistic and diverse generated shapes, thus further improving the quality of the grasping dataset. Second, in terms of shape selection metrics, the definition and calculation of metrics are not unique. For example, the size and orientation of objects are not taken into account in this paper, and real experimental results can also be used for the calculation of the graspness metric. Finally, shape generation is now only used once for data augmentation, but the shape generation process can be a lifelong process, which means we can continuously use the features of grasp-failed objects for subdivision interpolation and shape generation. By learning more and more shapes, the grasping algorithm may gradually improve its capabilities, and this lifelong learning method is what we hope to investigate more in future work.

Meanwhile, compare to our initial conference paper [30], we also investigate the regularization effect of the Critic network and the augmentation method effect on real robots in this paper. And both supplementary experiments have proved the effectiveness of our methods.

B. Conclusion

In this paper, we present a systematic pipeline for grasping dataset augmentation. Objects are encoded into feature vectors using the AE-Critic network, and generated objects, which are generated by leveraging the features of original high rarity and graspness score objects, are used to augment the original

grasping dataset. Experimental results show that our generated data improves the quality of the original grasping dataset, and thus improves the ability of the pre-designed learning-based grasp planning network.

REFERENCES

[1] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[2] M. Li, K. Hang, D. Kragic, and A. Billard, "Dexterous grasping under shape uncertainty," *Robotics and Autonomous Systems*, vol. 75, pp. 352–364, 2016.

[3] C. Ferrari and J. F. Canny, "Planning optimal grasps." in *ICRA*, vol. 3, no. 4, 1992, p. 6.

[4] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *2011 IEEE International conference on robotics and automation*. IEEE, 2011, pp. 3304–3311.

[5] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.

[6] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[7] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.

[8] D. Morrison, P. Corke, and J. Leitner, "Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4368–4375, 2020.

[9] R. Gao, Z. Si, Y.-Y. Chang, S. Clarke, J. Bohg, L. Fei-Fei, W. Yuan, and J. Wu, "Objectfolder 2.0: A multisensory object dataset for sim2real transfer," *arXiv preprint arXiv:2204.02389*, 2022.

[10] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International journal of robotics research*, vol. 39, no. 2-3, pp. 183–201, 2020.

[11] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[12] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[13] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

[14] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[15] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, *et al.*, "imgaug," https://github.com/aleju/imgaug, 2020, online; accessed 01-Feb-2020.

[16] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological cybernetics*, vol. 59, no. 4, pp. 291–294, 1988.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[18] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," *arXiv preprint arXiv:1702.05538*, 2017.

[19] D. Wang, D. Tseng, P. Li, Y. Jiang, M. Guo, M. Danielczuk, J. Mahler, J. Ichnowski, and K. Goldberg, "Adversarial grasp objects," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 241–248.

[20] P. Mitrano and D. Berenson, "Data augmentation for manipulation," *arXiv preprint arXiv:2205.02886*, 2022.

[21] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2528–2535.

[22] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, "Understanding and improving interpolation in autoencoders via an adversarial regularizer," *arXiv preprint arXiv:1807.07543*, 2018.

[23] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[24] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze, "3dnet: Large-scale object class recognition from cad models," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 5384–5391.

[25] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4304–4311.

[26] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, "Efficient implementation of marching cubes' cases with topological guarantees," *Journal of graphics tools*, vol. 8, no. 2, pp. 1–15, 2003.

[27] J. Vollmer, R. Mencl, and H. Mueller, "Improved laplacian smoothing of noisy surface meshes," in *Computer graphics forum*, vol. 18, no. 3. Wiley Online Library, 1999, pp. 131–138.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[30] J. Jiang, X. Xiao, F. Chen, and M. Li, "Learning grasp ability enhancement through deep shape generation," *arXiv preprint arXiv:2206.09353*, 2022.