2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

FOGL: Federated Object Grasping Learning

Seok-Kyu Kang[1] and Changhyun Choi[2]

***Abstract*— Federated learning is a promising technique for training global models in a data-decentralized environment. In this paper, we propose a federated learning approach for robotic object grasping. The main challenge is that the data collected by multiple robots deployed in different environments tends to form heterogeneous data distributions (i.e., non-IID) and that the existing federated learning methods on such data distributions show serious performance degradation. To tackle this problem, we propose federated object grasping learning (FOGL) that uses cross-evaluation in a general federated learning process to assess the training performance of robots. We cluster robots with similar training patterns and perform independent federated learning on each cluster. Finally, we integrate the global models for each cluster through an ensemble inference. We apply FOGL to various federated learning scenarios in robotic object grasping and show state-of-the-art performance on the Cornell grasping dataset.**

I. INTRODUCTION

Robotic object grasping has been actively explored and shown significant progress recently [1], [2]. As robot grasping systems are being deployed in real environments, an emerging problem is: how to exploit the power of multiple deployed robots and the data continuously collected by them. While there have been a few studies [3], [4] on how multiple robots collect object grasping data in parallel, the data was gathered in a central server for a batch training. However, the importance of data privacy has been increased recently [5], [6] and transmission of extensive data collected from multiple robots is often prohibited due to communication and storage costs. For instance, we would not want domestic and factory robots to share image data of personal items and proprietary parts with a central data server, respectively. As robots become more universal and personalized, this issue is becoming more prevalent.

Federated learning is a method of training models in a situation where data is decentralized while ensuring data privacy. Since this method does not need to transmit local robot data to the central server, there are several advantages such as guaranteeing data privacy and efficient training in a distributed data environment [7]. Lately, federated learning has been considered as one of the important methods when it comes to effectively training of multi-robots [8], [9], [10], [11], [12], [13].

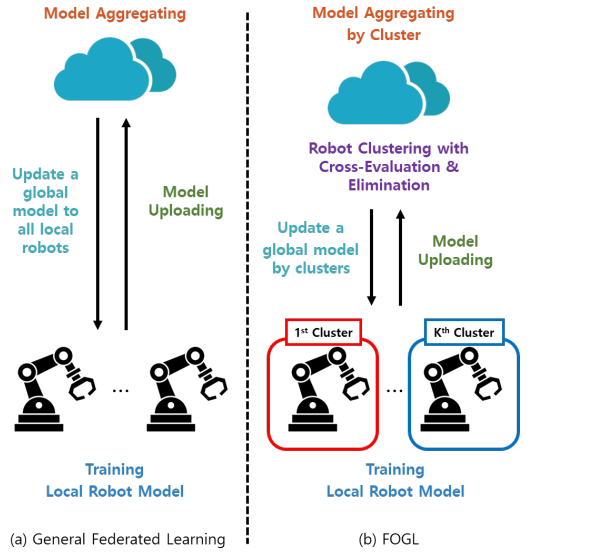

Fig. 1. While (a) general federated learning methods aggregate local models into a global model, (b) our approach, FOGL, clusters the local models via cross-evaluation and elimination and then performs federated learning on each cluster to generate a global model through an ensemble technique.

As shown in Fig. 1 (a), in typical federated learning, the weights of the individually trained neural network model in each robot are transmitted to the central server, and the server aggregates the weights and distributes them to each robot again. When the data of each local robot is not independent and identically distributed (non-IID), the trained models are very dissimilar, and hence existing federated learning algorithms suffer from performance degradation and unstable learning. Although this situation is quite common in the real-world, this problem has not been sufficiently addressed yet.

In this paper, we propose federated object grasping learning (**FOGL**), a federated learning method that effectively trains multi-robots in grasping operations even in situations where inter-robot data is *heterogeneous* and *not transferable*. The core idea of FOGL is to perform independent federated learning for each cluster of robots showing similar learned models and hence comparable performance. General federated learning methods aggregate local models into one global model. If the local models are trained on heterogeneous data, these approaches hinder the convergence of the global model, and hence the global model underperforms. Our proposal,

This research was supported by the Republic of Korea government (MSIT, Ministry of Science and ICT), under the High-Potential Individuals Global Training Program (IITP-2021-0-02132) and AI Graduate School Support Program (Sungkyunkwan University) (IITP-2019-0-00421) supervised by the IITP (Institute of Information and Communications Technology Planning & Evaluation). Special thanks to William Boulanger for the English proofreading of this paper.

[1]Seok-Kyu Kang was with the Department of Artificial Intelligence, Sungkyunkwan University, Suwon, Gyeonggi, South Korea kittyhyen@skku.edu and is currently with Korea Shipbuilding & Offshore Engineering Co., Ltd. (KSOE), HD Hyundai Group, Seoul, South Korea seokkyu.kang@ksoe.co.kr

[2]Changhyun Choi is with the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities, Minneapolis, USA cchoi@umn.edu

FOGL, performs clustering through cross-evaluation and elimination as shown in Fig. 1 (b). FOGL clusters similar local models and performs federated learning on each cluster to generate a set of global models. We create a final global model by combining the generated global models into one using an ensemble technique. The global model learned through our FOGL reaches state-of-the-art performance on a standard object grasping dataset.

To summarize, our main contributions are as follows:

- We propose FOGL, federated object grasping learning. By assessing the performance of multiple robots, it clusters the learned grasping models and selects promising models for effective learning.
- We experimentally prove that FOGL shows a significant performance improvement on the Cornell robot grasping dataset.
- We show that FOGL can be effectively applied in various training scenarios such as when the local robot data is heterogeneous, an auxiliary dataset is utilized from a central server, and a pretrained model is applied.

II. Related Work

A. *Federated Learning*

When data collected from multiple machines is sensitive to privacy or too big to share, it is often not possible to send the data to a server in order to train all at once using common centralized training methods. McMahan et al. [7] proposed a federated learning method that trains a shared model by leaving distributed training data on multiple local machines and aggregating locally computed updates. More generally, the purpose of federated learning is to create a global model in such a way that a central server can efficiently process data distributed on all local machines without accessing them. However, there is a degradation of performance when the data on the local machines is heterogeneous, and a number of federated learning studies have been proposed to solve this problem as follows.

*1) **Average-Based Federated Learning**:* Previous studies on federated learning have mainly tried to solve the performance degradation in non-IID situations using methods based on averaging [7], [14], [15]. FedAvg [7] is an algorithm that simply averages the respective weights of the models on the central server. FedProx [14] improved on FedAvg by adding a proximal term. FedNova [15] is a normalized averaging method that eliminates objective inconsistency while preserving fast error convergence. Average-based federated learning has the advantage of not requiring data on the server during training. However, since this approach is average-based, it is vulnerable to the existence of local robot models with dissimilar weights or poor relative performance.

*2) **Knowledge-Based Federated Learning**:* Some studies on federated learning have investigated knowledge distillation. FedDfusion [16] proposed ensemble distillation for model fusion, training the central classifier through auxiliary data on the outputs of the models. FedDistill [17] also proposed learning the usage of knowledge distillation and assumed that private user data samples could be used for model training. Knowledge-based federated learning requires an auxiliary (proxy) dataset of the same distribution. Therefore, it is only used in limited situations, and there is a serious performance degradation problem if the auxiliary data is not evenly distributed.

B. *Federated Learning in Robotics*

A few studies investigated federated learning in distributed robot problems. Majcherczyk et al. [8] proposed Flow-FL, multi-robot federated learning systems and setting for spatio-temporal predictions of robot swarms. Yu et al. [9], [11] applied federated learning to a vision-based obstacle avoidance problem for distributed mobile robots. Zhang et al. [10] proposed a federated learning-based dynamic map fusion framework for intelligent networked vehicles. J.S.Nair et al. [12] proposed dFRL, decentralized federated reinforcement learning in multi-robot scenarios. Ho et al. [13] applied a weighted federated learning for a warehouse task scheduling problem.

While federated learning has been explored in the aforementioned robot problems, the study of federated learning on grasping training has been under-explored. Previous research on robot grasping [1], [2], [18], [19] mainly suggest methods for more accurate and efficient training in a single robot. Some studies on multi-robot data collection [3], [4] have been proposed, but these studies assume that all data is shared with a central server. Also, they do not consider the problem of multiple deep learning model training, heterogeneous data distribution, and privacy issues.

III. Proposed Method

We propose FOGL, federated object grasping learning, for the multi-robot system training. FOGL consists of two Steps. Step 1 performs the general federated learning with cross-evaluation, and Step 2 conducts the clustered federated learning and ensemble. Fig. 2 shows the overall training flow of FOGL.

A. *Step 1. Federated Learning with Cross-Evaluation and Elimination*

In Step 1, the central server performs general federated learning, for which the central server collects models trained by local robots for aggregation. We use the local robot models collected from every communication round to identify the relative performance of the models through cross-evaluation. The robot that hinders learning is removed from the training process through elimination, and the local models are clustered. We consider two scenarios depending on the existence of data on the central server.

*1) **Scenario 1: No data on server**:* Each local robot spares 10% of its training set for the validation set. Each robot trains using its own training set for a set local epoch in the same way as in general federated learning and then transmits the trained model to the central server. After that, the central server sends the collected local models to each robot for cross-evaluation and elimination. The central server encrypts the order and the name of the local models so that

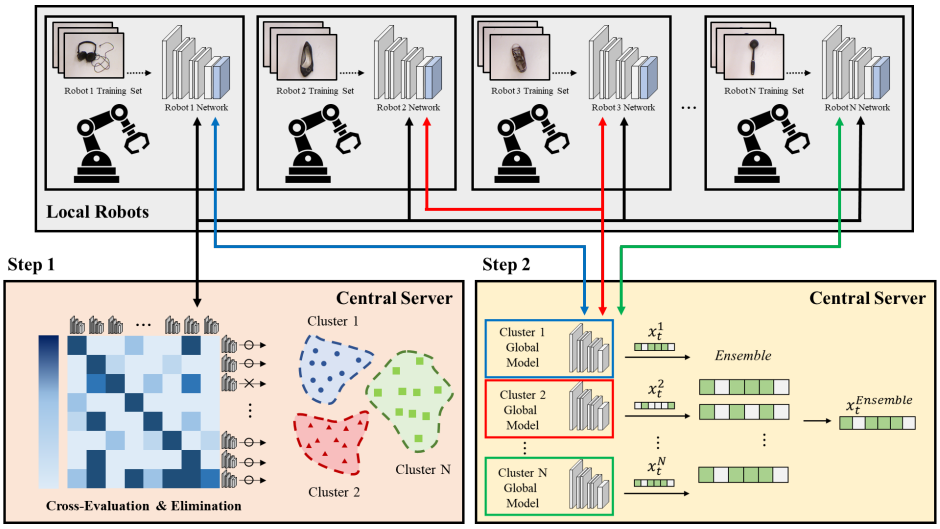

Fig. 2. Overview of federated object grasping learning (FOGL).

each local robot does not know identity of the other shared models. Each robot evaluates its own model and the models it received from the server with its validation set. The results of the inference process are then sent back to the central server.

The central server selects some local robot models by the cross elimination rule as follows. Let $P_{i\to j}$ denote the performance of the robot i's model evaluated on the robot j's validation set. If both of the following conditions are satisfied, robot model j is eliminated and replaced with robot model i.

$$\begin{cases} \text{i) } P_{j\to j} < P_{i\to j} \text{ and } P_{j\to i} < P_{i\to i} \\ \text{ii) } \beta < |P_{j\to j} - P_{i\to j}| \end{cases} \tag{1}$$

where β is a hyperparameter that sets the elimination patience, which can be adjusted according to cross-evaluation performance between robots (default = 10).

More generally, we create the cross-evaluation matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$, where N is the number of training robots and $M_{ij} = P_{i\to j}$, through cross-performance evaluation of each trained model and validation set. Similarly, the following conditions are employed to keep and eliminate the robot model i and j, respectively:

$$\begin{cases} \text{i) } M_{jj} < M_{ij} \text{ and } M_{ji} < M_{ii} \\ \text{ii) } \beta < |M_{jj} - M_{ij}| \end{cases} \tag{2}$$

If the first condition is satisfied, it means that the model i performs better than the model j in the validate sets of the robot i and j. The second condition makes sure if the model i is significantly better than the model j.

The central server clusters the selected local robot models that show similar performance on the same validation data based on the cross-evaluation matrix $\mathbf{M}$. We use the mean shift algorithm [20] for clustering that is a procedure for locating the maxima of a density function's modes given discrete data sampled from that function [21]. The mean-shift algorithm has the advantage of being fast and easy to apply among non-parametric clustering algorithms. We record the result of the clustering process in each communication round, and if the same results are shown more than γ times (default = 3), the general federated learning with cross-evaluation step (Step 1) is finished.

*2) **Scenario: Auxiliary data on server**:* It is often assumed that some dataset is available on the central server [16], [17], [22]. FOGL exploits an auxiliary dataset on the sever to facilitate the aforementioned cross-evaluation, elimination, and clustering. Specifically, FOGL evaluates the performance of the local models on the auxiliary dataset and appends the poor local models to the elimination list (e.g., 10%p lower than the average local model). The local models showing similar performance on the auxiliary dataset are further considered in the clustering process as surrogate information.

Since the auxiliary dataset enables FOGL to simplify the cross-evaluation and elimination, we introduce a light version of FOGL, FOGL-l, that does not distribute the local models to each robot for cross-evaluation. As FOGL-l uses only the dataset on the central server, communication and computation costs can be significantly reduced.

B. Step 2. Clustered Federated Learning and Ensemble

After Step 1 is finished, the central server receives the elimination and clustering list for local robots and a global model that is resulting from the last communication round. The central server excludes local robots in the elimination list from training and performs independent federated learning for each cluster in the clustering list. The central server obtains as many global models as the number of clusters. Finally, we integrate these global models into a global ensemble model as the final output. We use an ensemble method that averages the final output of each model [23].

Note that existing federated learning algorithms do not exclude robots with relatively poor performance from training. If a local robot model with relatively poor performance

is continuously used for federated learning, it is difficult to expect high performance of the global model. FOGL eliminates the robots that hinder training through cross-evaluation so that federated learning can be more effective.

In addition, the existing federated learning algorithms show good performance in IID situations. However, in situations where the data of each robot is heterogeneous (i.e., non-IID), each local robot model is trained heterogeneously. If the central server aggregates such heterogeneous local robot models, there is a high possibility of causing serious performance degradation. Since FOGL performs federated learning among similar local robot models, it significantly alleviates the problem that current federated learning algorithms have in non-IID data situations.

IV. Experiment Setup

A. Robot Grasping Dataset

For evaluation, we use the extended version of the Cornell grasping dataset [24], which is one of the widely used robotic grasping datasets. We also use the Jacquard dataset [25] as an auxiliary dataset (see Section V-C).

The extended version of the Cornell grasping dataset [24] consists of 1035 RGB-D images of 240 objects. Each image is annotated with ground truth positive and negative grasping rectangle representations. To simulate the non-IID data distributions, we recategorized the images of the dataset into two large categories, 'home' and 'office', consisting of 885 images in total. These categories were then further divided into several smaller categories totaling 20 as shown in Table I. The parentheses in Table I indicate the number of images in each category. In addition, we performed image augmentation through random crops, zooms, and rotations for effective training.

TABLE I
RECATEGORIZED CORNELL GRASPING DATASET

Large Category	Small Category
home(416)	bottle(71), box(28), cooking(35), cup&bowl(44), fruits(76), shaver(24), snack(27), sponge(16), toothbrush(24), toothpaste(24), can(12), others(35)
office(469)	cap(22), electronics(86), glasses(36), scissors(28), lock(17), pen(44), shoes(40), tape(31), tools(32), others(133)

For the non-IID situation, we assigned 20 categories to each robot with the ratio of the non-IID data, $r = \{0.25, 0.5, 0.75, 1.0\}$. For example, $r = 0.25$ represents that each robot holds 25% of the data in one of the 20 categories, while the remaining 75% holds IID data.

The Jacquard dataset [25] is built on a subset of ShapeNet [26], a large dataset of CAD models. It contains both 54k RGB-D images and annotations of successful grasping positions obtained in a simulated environment. We used it in the scenarios where auxiliary data exists on the server and pretrained models are initially deployed.

B. Federated Learning Setup and Configuration

We set up 20 local robots and one central server for a federated learning experiment. We used 1 DGX Station (8× NVIDIA A100 GPUs) as a central server. We also set up 8 NVIDIA RTX2080Ti GPUs and 12 NVIDIA GTX1080Ti GPUs as local robots. For the backbone, we adopt GR-ConvNet [1] that is one of the state of the art model in robotic grasping. As baselines, we used FedAvg [7], FedProx [14], and FedNova [15] which are average-based algorithms. We also used FedDistill [17] and FedDfusion [16] as knowledge-based baselines, which consider the auxiliary data in the central server. We set up the Adam optimizer, SGD (learning rate=0.01, momentum=0.9), 5 local epochs, and 50 communication rounds.

V. Experiment Results

We evaluate the grasp prediction accuracy [24] of the existing federated learning algorithms and our proposed method (FOGL, FOGL-l) on the Cornell grasping dataset in various scenarios.

A. Exp 1: Scenario where no data on server

We consider a scenario where there is no data set on the central server and data exists only in each local robot. Table II shows the results of applying our proposed FOGL to FedAvg, FedProx, and FedNova, which are average-based federated learning algorithms. The better accuracy than the baselines is bold-faced. Oracle shows the upper bound performance where the server has all the robot data and trains a global model. Ensemble represents another baseline that simply ensembles the inference results of the local robot models using averaging [23] without federated learning.

TABLE II
EXP1: GRASPING ACCURACY ON CORNELL DATASET IN A SCENARIO WHERE THERE IS NO DATASET ON THE SERVER

Algorithm	IID	Non-IID (r = 0.25)	Non-IID (r = 0.5)	Non-IID (r = 0.75)	Non-IID (r = 1.0)
Oracle	98.87	-	-	-	-
Ensemble	92.24	82.58	67.86	52.97	52.33
FedAvg [7]	92.16	84.67	69.12	54.96	54.35
FedProx [14]	93.12	85.39	68.16	56.91	55.20
FedNova [15]	93.48	88.12	71.06	63.41	61.78
FOGL-Avg	92.11	**85.29**	**73.53**	**62.94**	**66.29**
FOGL-Prox	**93.15**	**86.84**	**73.68**	**65.17**	**68.42**
FOGL-Nova	**94.74**	**91.18**	**76.31**	**67.64**	**69.21**

As the rate of non-IID increases, the grasping accuracy of all the algorithms decreases. The federated learning algorithms applied with FOGL show better grasping accuracy in most situations except for a few IID cases. The performance of the baselines tends to decrease as the data becomes more heterogeneous. For baselines to which FOGL is applied, the higher the rate of data heterogeneity, the better the performance improvement they achieve. This verifies that FOGL

TABLE III

EXP 2: GRASPING ACCURACY ON CORNELL DATASET IN A SCENARIO WHERE THERE IS THE CORNELL DATASET ON THE SERVER

Algorithm	IID	Non-IID (r = 0.25)	Non-IID (r = 0.5)	Non-IID (r = 0.75)	Non-IID (r = 1.0)
FedAvg [7]	92.16	84.67	69.12	54.96	54.35
FedProx [14]	93.12	85.39	68.16	56.91	55.20
FedNova [15]	93.48	88.12	71.06	63.41	61.78
FedDistill [17]	93.53	88.81	72.25	66.05	63.47
FedDfusion [16]	93.58	89.47	72.82	66.71	64.46
FOGL-Avg-l	92.11	**85.38**	**72.93**	**65.02**	**69.15**
FOGL-Prox-l	**93.15**	**86.29**	**72.61**	**65.22**	**69.49**
FOGL-Nova-l	**94.74**	**88.23**	**74.80**	**68.13**	**68.31**
FOGL-Distill-l	92.13	**90.16**	**76.51**	**67.47**	**68.00**
FOGL-Dfusion-l	**93.71**	**91.03**	**75.49**	**68.50**	**69.43**
FOGL-Avg	92.10	**85.47**	**73.94**	**66.24**	**69.29**
FOGL-Prox	**93.15**	**86.84**	**73.75**	**66.20**	**69.43**
FOGL-Nova	**94.74**	**92.16**	**78.95**	**69.42**	**69.21**
FOGL-Distill	**95.16**	**90.50**	**76.25**	**66.73**	**70.24**
FOGL-Dfusion	**94.73**	**91.17**	**76.47**	**69.01**	**69.45**

TABLE IV

EXP 3: GRASPING ACCURACY ON CORNELL DATASET IN A SCENARIO WHERE THERE IS THE JACQUARD DATASET ON THE SERVER

Algorithm	IID	Non-IID (r = 0.25)	Non-IID (r = 0.5)	Non-IID (r = 0.75)	Non-IID (r = 1.0)
FedAvg [7]	92.16	84.67	69.12	54.96	54.35
FedProx [14]	93.12	85.39	68.16	56.91	55.20
FedNova [15]	93.48	88.12	71.06	63.41	61.78
FedDistill [17]	89.47	79.41	63.55	53.15	51.81
FedDfusion [16]	91.10	80.89	65.24	52.80	50.00
FOGL-Avg-l	92.11	**85.41**	**72.28**	**61.63**	**60.05**
FOGL-Prox-l	92.75	**85.40**	**72.02**	**64.31**	**60.92**
FOGL-Nova-l	93.21	**87.03**	**75.84**	**66.73**	**61.63**
FOGL-Distill-l	88.92	78.32	60.92	**58.09**	**52.80**
FOGL-Dfusion-l	90.94	80.10	**67.00**	**53.81**	**54.17**
FOGL-Avg	92.11	**85.43**	**73.68**	**65.31**	**66.29**
FOGL-Prox	92.76	**86.25**	**73.53**	**65.02**	**68.43**
FOGL-Nova	**94.74**	**89.84**	**76.32**	**68.50**	**69.21**
FOGL-Distill	88.92	**80.41**	**67.26**	**55.11**	**53.15**
FOGL-Dfusion	90.94	**82.13**	**69.50**	**53.67**	**55.24**

is effective in the robotic federated learning scenarios where the data is non-IID. In particular, in the case of non-IID (r=1.0), it achieves a maximum performance improvement of 13.22%p (i.e., percentage point) in FOGL-Prox.

B. Exp 2: Scenario where similar auxiliary dataset exists on server

The scenario where some auxiliary datasets exist on the central server is commonly considered in federated learning studies [16], [17], [22]. We further consider FedDistill and FedDfusion as baselines that are federated learning algorithms using auxiliary datasets for training. In addition, we compare the performance of FOGL-l, a lighter variant of FOGL that does not perform cross-evaluation and elimination, not requiring additional communication with local robots.

We first set up a scenario where the same Cornell grasping dataset exists in the central server. We assign the 'others' in both 'home' and 'office' categories in Table I as an auxiliary dataset on the server. The size of the auxiliary dataset is $168 = 35 + 133$ which corresponds to about 19% of the total training dataset, and it was further enriched by performing data augmentation, resulting in a dataset of 1008 (168×6) images. Note that the auxiliary dataset shares the same background (white tabletop) as the training set, but the categories of objects are different.

Table III shows the results on the Cornell dataset as auxiliary data on the central server. When a similar auxiliary dataset exists on the server, FOGL shows an average performance improvement of 4.7%p compared to the baseline. It also shows an additional performance improvement compared to EXP 1 in which there is no data on the server. In the extreme cases (IID and non-IID with r=1.0), there is no significant difference, but in the non-IID cases (r=0.25, 0.5, 0.75), the average performance improvement is 1.7%p compared to that of EXP 1. Therefore, if datasets of similar domains are available on the central server, we can use them for marginal performance improvement.

On the baselines without FOGL, the knowledge distillation-based federated learning algorithms (i.e., FedDistill and FedDfusion) perform better than the average-based federated learning algorithms by 4.32%p on average. However, with FOGL, the difference is reduced to an average of 1.25%p.

Compared to FOGL, FOGL-l only shows an average performance drop of 0.90%p in all cases, and there is also no significant difference in performance, especially in IID and non-IID (r=1.0) cases. When comparing the performance of FOGL-l and that of the baselines, the average performance improvement is 3.8%p. Through this experiment, we notice that if a similar auxiliary dataset exists in the central server, FOGL-l shows similar performance to FOGL while minimizing communication between the local robots and the central server, providing a good alternative solution for systems with high communication costs.

C. Exp 3: Scenario where dissimilar auxiliary dataset exists on server

We perform the same experiment as EXP 2 for the scenario where the auxiliary dataset on the central server is in a different domain from the local robot datasets. To test this situation, the Jacquard dataset was used in the central server. The Jacquard dataset consists of different objects and backgrounds from the Cornell grasping dataset. Compared to EXP 2, the amount of auxiliary data size is about 10.71 times bigger (10.8k).

TABLE V

EXP 4: GRASPING ACCURACY ON CORNELL DATASET WITH DIFFERENCE INITIAL MODELS

	From scratch		Cornell pretrained		Jacquard pretrained	
Algorithm	IID	Non-IID (r = 1.0)	IID	Non-IID (r = 1.0)	IID	Non-IID (r = 1.0)
FedAvg [7]	92.16	54.35	95.50	65.24	91.23	50.01
FedNova [15]	93.48	61.78	95.87	65.27	93.48	53.25
FedDfusion [16]	-	-	97.05	66.16	90.87	50.00
FOGL-Avg-l	-	-	**95.50**	**70.16**	**90.38**	**58.40**
FOGL-Nova-l	-	-	**96.82**	**72.49**	**92.24**	**60.88**
FOGL-Dfusion-l	-	-	**97.24**	**74.53**	**89.02**	**50.73**
FOGL-Avg	92.11	**66.29**	**95.51**	**71.73**	**92.50**	**65.02**
FOGL-Nova	**94.74**	**69.21**	**96.63**	**72.66**	**95.55**	**67.52**
FOGL-Dfusion	-	-	**97.75**	**76.27**	90.58	**55.04**

TABLE VI

EXP 5: GRASPING ACCURACY ON CORNELL DATASET WITH DIFFERENCE BACKBONE MODELS

	AlexNet [18]		ResNet-50 [19]		GR-ConvNet [1]	
Algorithm	IID	Non-IID (r = 1.0)	IID	Non-IID (r = 1.0)	IID	Non-IID (r = 1.0)
Oracle	88.00	-	96.03	-	97.71	-
FedAvg [7]	80.90	51.81	89.89	51.06	92.16	54.35
FedNova [15]	82.24	53.67	89.88	52.33	93.48	61.78
FedDfusion [16]	81.58	54.17	88.76	55.11	93.58	64.46
FOGL-Avg-l	80.01	**57.35**	**89.47**	**57.05**	**92.11**	**69.15**
FOGL-Nova-l	**84.21**	**59.13**	**88.37**	**61.30**	**94.74**	**68.31**
FOGL-Dfusion-l	**86.84**	**61.81**	**90.11**	**66.29**	**93.71**	**69.43**
FOGL-Avg	**81.47**	**57.83**	**90.13**	**58.39**	**92.10**	**69.29**
FOGL-Nova	**86.51**	**60.89**	**90.26**	**63.67**	**94.74**	**69.21**
FOGL-Dfusion	**87.64**	**63.48**	**89.38**	**67.47**	**94.73**	**69.45**

Table IV summarizes the results on the Cornell dataset with the Jacquard dataset as auxiliary data on the server. FedAvg, FedProx, and FedNova, which are average-based federated learning, do not have a big performance difference even if the domains of the datasets in the server are different. However, in the case of FedDistill and FedDfusion, which use the dataset in the server directly in the federated learning training process, the average performance of FedDistill and FedDfusion is lower than that of EXP 2. That is because they are greatly affected by the domain gap. When FOGL is applied, their performance improvement is 1.89%p compared to the baseline. Average-based federated learning applied with FOGL shows an average performance that is 8.2%p higher than theirs.

This performance degradation of the knowledge distillation-based algorithm becomes more exacerbated when using FOGL-l. The average-based FOGL-l algorithms show an average decrease in performance of 2.08%p compared to EXP 2, whereas the knowledge distillation-based FOGL-l algorithms show an average decrease in performance of 10.74%p in performance compared to EXP 2. The knowledge-based algorithms are greatly affected by domain changes even when FOGL is applied. Given the results of EXP 2 and EXP 3, it is advantageous to apply FOGL to the average-based federated learning algorithm when the domain of the auxiliary data on the server is quite different from that of data on the robots.

D. *Exp 4: Scenario of deploying a pretrained model*

In the previous experiments, we trained models from scratch by distributing randomly initialized models. However, initiating training with pretrained a model in a similar domain or other domain can also be considered.

We deploy a pretrained initial model in the situation where the central server does not hold data (EXP 1), similar domain data (EXP 2), and other domain data (EXP 3). Baselines used FedAvg, the most basic algorithm for federated learning, FedNova, one of the average-based federated learning algorithms, and FedDfusion, one of the knowledge distillation-based federated learning algorithms. The results of EXP 4 are shown in Table V.

In Table V, 'From scratch' column shows grasping accuracy when the dataset does not exist on the server. Since FedDfusion and FOGL-l are methods that presuppose the auxiliary data, these two cases are excluded from the results. In the 'Cornell pretrained' and 'Jacquard pretrained' columns, Cornell and Jacquard datasets exist on each server, and models are pretrained using the data and then trained.

According to the results, the Cornell pretrained model has the highest performance. Deploying a model pretrained with other domain data (i.e., Jacquard dataset) has the effect of lowering performance. Given the results, if we plan to distribute the initial pretrained model to each local robot, it is advantageous to distribute the pretrained model in the same domain. If the data collected by the central server is more likely to be data from other domains, we can expect higher performances by training from scratch.

E. *Exp 5: Evaluation with different backbone models*

Finally, we test how the initially deployed backbone model can affect the performance in the scenario of EXP 2. We use three well-known deep learning models, AlexNet [18], ResNet-50 [19], and GR-ConvNet [1], as backbones, which are typically used in robot grasping studies. The results are shown in Table VI. The performance of baselines, FOGL and FOGL-l tends to be proportional to the performance of the backbone model, but FOGL shows the best accuracy on average.

VI. CONCLUSIONS

We proposed FOGL, federated object grasping learning, that can be effectively applied to existing federated learning algorithms. We trained a global model using FOGL, performing federated learning with the cross-evaluation, elimination, and ensemble. We experimentally proved that FOGL shows good performance improvement in various data distributions on local robots and the central server.

REFERENCES

[1] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9626–9633, IEEE, 2020.

[2] S. Ainetter and F. Fraundorfer, "End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13452–13458, IEEE, 2021.

[3] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection," in *Proceedings of International Symposium on Experimental Robotics (ISER)*, 2016.

[4] C. Finn and S. Levine, "Deep Visual Foresight for Planning Robot Motion," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[5] B. Yankson, "An empirical study: Privacy and security analysis of companion robot system development," in *ICCWS 2021 16th International Conference on Cyber Warfare and Security*, p. 409, Academic Conferences Limited, 2021.

[6] M. Qiu, H.-N. Dai, A. K. Sangaiah, K. Liang, and X. Zheng, "Guest editorial: Special section on emerging privacy and security issues brought by artificial intelligence in industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2029–2030, 2020.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[8] N. Majcherczyk, N. Srishankar, and C. Pinciroli, "Flow-fl: Data-driven federated learning for spatio-temporal predictions in multi-robot systems," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8836–8842, IEEE, 2021.

[9] X. Yu, J. P. Queralta, and T. Westerlund, "Towards lifelong federated learning in autonomous mobile robots with continuous sim-to-real transfer," *arXiv preprint arXiv:2205.15496*, 2022.

[10] Z. Zhang, S. Wang, Y. Hong, L. Zhou, and Q. Hao, "Distributed dynamic map fusion via federated learning for intelligent networked vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 953–959, IEEE, 2021.

[11] Y. Xianjia, J. P. Queralta, J. Heikkonen, and T. Westerlund, "Federated learning in robotic and autonomous systems," *Procedia Computer Science*, vol. 191, pp. 135–142, 2021. Publisher: Elsevier.

[12] J. S. Nair, D. D. Kulkarni, A. Joshi, and S. Suresh, "On decentralizing federated reinforcement learning in multi-robot scenarios," in *2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pp. 1–8, 2022.

[13] T. M. Ho, K.-K. Nguyen, and M. Cheriet, "Federated Deep Reinforcement Learning for Task Scheduling in Heterogeneous Autonomous Robotic System," *IEEE Transactions on Automation Science and Engineering*, pp. 1–13, 2022.

[14] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[15] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[16] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 2351–2363, 2020.

[17] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data," Nov. 2018. arXiv:1811.11479 [cs, stat].

[18] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 1316–1322, IEEE, 2015.

[19] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[20] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[21] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[22] H.-Y. Chen and W.-L. Chao, "Fedbe: Making bayesian model ensemble applicable to federated learning," in *International Conference on Learning Representations*, 2021.

[23] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123–140, 1996. Publisher: Springer.

[24] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[25] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3511–3516, IEEE, 2018.

[26] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.