When Continual Learning Meets Robotic Grasp Detection: A Novel Benchmark on the Jacquard Dataset

Rui Yang[1,2][a], Matthieu Grard[1][b], Emmanuel Dellandréa[2][c] and Liming Chen[2][d]
[1]*Siléane, 17 Rue Descartes, Saint-Etienne, France*
[2]*Liris, Ecole Centrale de Lyon, 36 Av. Guy de Collongue, Ecully, France*

Keywords: Robots-Grasp, Continual Learning.

Abstract: Robotic grasp detection is to predict a grasp configuration, e.g., grasp location, gripper openness size, to enable a suitable end-effector to stably grasp a given object on the scene, whereas continual learning (CL) refers to the skill of an artificial learning system to learn continuously about the external changing world. Because it corresponds to real-life scenarios where data and tasks continuously occur, CL has aroused increasing interest in research communities. Numerous studies have focused so far on image classification, but none of them involve robotic grasp detection, although extending continuously robots with novel grasp capabilities when facing novel objects in unknown scenes is a major requirement of real-life applications. In this paper, we propose a first benchmark, namely Jacquard-CL, that uses a small part of the Jacquard Dataset with variations of the illumination and background to create a NI(new instances)-like scenario. Then, we adapt and benchmark several state-of-the-art continual learning methods to the grasp detection problem and create a baseline for the issue of continual grasp detection. The experiments show that regularization-based methods struggle to retain the previously learned knowledge, but memory-based methods perform better.

1 INTRODUCTION

Dexterous manipulation has been a long-standing challenge in AI and robotics. Already Aristotle noted that the hand is the "tool of tools", and Anaxagoras held that "man is the most intelligent of the animals because he has hands". Thus, intelligence has long been understood to come together with dexterity, and the lack thereof in artificial systems explains why current robots are mostly limited to pre-programmed tasks in known environments.

As such, a huge research effort (Kumra et al., 2020) has been focused on the so-called robotic grasp detection problem, also known as grasp synthesis, which aims to predict from the observation of a scene a grasp configuration, *e.g.*, grasp location, gripper openness size, orientation, to enable a suitable end-effector to stably grasp a given object laid on the scene.

State of the art has so far featured a number of

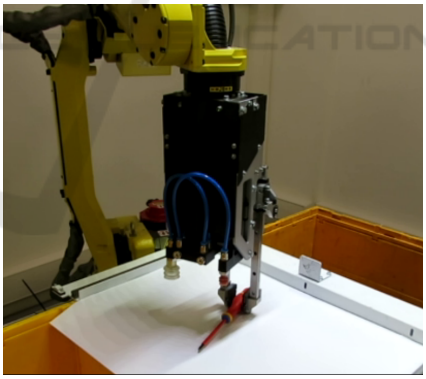

Figure 1: A robotic arm is trained to grasp an object (Depierre et al., 2021).

CNN-based methods for grasp detection from either RGB or RGB-D images. Most of them can achieve satisfactory performances (Zhou et al., 2018a), (Depierre et al., 2018), (Morrison et al., 2018) (Kumra et al., 2020). For instance, GR-ConvNet (Kumra et al., 2020) achieves 97.7% and 94.6% on the Cornell and Jacquard datasets, respectively(Kumra et al., 2020). However, all these models are trained with a large dataset through an offline scenario where the entire training data is available, and the training and test-

[a] https://orcid.org/0000-0002-2102-4306
[b] https://orcid.org/0000-0003-3477-3696
[c] https://orcid.org/0000-0001-7346-228X
[d] https://orcid.org/0000-0002-3654-9498

Yang, R., Grard, M., Dellandréa, E. and Chen, L.
When Continual Learning Meets Robotic Grasp Detection: A Novel Benchmark on the Jacquard Dataset.
DOI: 10.5220/0011874700003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP*, pages 703-710
ISBN: 978-989-758-634-7; ISSN: 2184-4321

ing data are assumed i.i.d. But this scenario is quite unrealistic for real-world applications. For example, a service robot may encounter different scenes and lighting conditions during its life span, and it is therefore difficult, if not impossible, to retrain the robot with the whole dataset each time it encounters novel scenes in its life span, and this for two reasons: 1). Previous data may not be available anymore, especially due to the reason of data privacy and protection; 2). Retraining the model each time of a novel scene with the entire dataset is also time-consuming. Nevertheless, if the robot is only trained with the data of new scenes, it often suffers from a well-known phenomenon, namely *catastrophic forgetting* (Goodfellow et al., 2014), thereby performing poorly in previous scenes. As a result, how to train the robot to learn new knowledge efficiently without *forgetting* becomes a key issue for real-world applications. This is exactly the aim of the research line on *continual learning*.

Last years have witnessed many scenarios and methods in continual learning (CL), but most of them only focus on image classification. In this paper, we extend CL to the issue of robotic grasp detection. In CL, different learning scenarios can be considered. De Ven and Tolias (van de Ven and Tolias, 2019) propose the following three CL scenarios: 1). Task-incremental learning where task identity is always provided in both training and evaluation; 2). Domain-incremental learning where task identity is not provided, and the model needs not to predict task identity during the inference; 3). Class-incremental learning where task identity is not provided, and the model needs to predict task identity during the inference. More specifically, for object recognition, Lomonaco and Maltoni (Lomonaco and Maltoni, 2017) propose to consider the following three scenarios: 1). New Instances (NI) where new unseen instances of the same class appear in the subsequent task; 2). New Classes (NC) where new unseen classes appear in the subsequent task; 3). New Instances and Classes (NIC) where new unseen objects and instances appear in the subsequent task. For a classification problem tackled with a deep neural network, each output neuron is related to a class. Thus, when a new class occurs, the output space is modified. This is not the case for the problem of grasp prediction, since the output space does not change. As a result, in this paper we focus on the scenario NI or domain-incremental learning. This means the neural network architecture doesn't change during the training. We consider novel instances of objects within different scenes and illumination conditions in subsequent tasks.

State of the art on CL has so far featured three main approaches, namely regularisation-based, memory-based, and the use of dynamic architectures. Regularisation-based methods (Chaudhry et al., 2018) (Zenke et al., 2017) (Kirkpatrick et al., 2017) evaluate the importance of each parameter and add some constraints to prevent these parameters from changing too much during training. These approaches work well for task-incremental learning. However, they all fail when applied to domain-incremental or class-incremental learning (van de Ven and Tolias, 2019). For our purpose of grasp detection, these methods can be directly applied without modification. Memory-based techniques (Li and Hoiem, 2016), (Shin et al., 2017), (Lopez-Paz and Ranzato, 2017), (Chaudhry et al., 2019), (Goodfellow et al., 2014), (Bang et al., 2021), (Douillard et al., 2020) store a small part of exemplars of previous tasks or create pseudo-exemplars. They are efficient at preventing forgetting, but the performance is limited by the replay buffer size or the generative model's quality. Some methods can be directly extended to our purpose, but others targeting class-incremental learning are not suitable for grasp detection. The methods based on dynamic architectures (Rusu et al., 2016) (Aljundi et al., 2017) (Verma et al., 2021) propose to add an auxiliary architecture or fix a part of the network for a specific task. They mainly target task-incremental learning. Therefore, they are not suitable for our purpose.

The contributions of this work are twofold :

- We propose a new benchmark based on Jacquard dataset for continual grasp detection.
- We expand some works of continual learning to grasp detection and create a baseline.

2 RELATED WORK

Grasp Detection Architectures. Recent CNN-based models usually use 2D representation for a grasp. It can be described as $g = \{x, y, h, w, \theta\}$ where (x, y) is the center, (h, w) represents the dimensions, and θ the orientation. As a result, Redmond and Angelova work(Redmon and Angelova, 2015) treats grasp detection as a pure regression problem. It uses AlexNet as the backbone and predict these five values directly in the output layer. This model can only predict one grasp from one image and can not fully use the dataset. Inspired by the research on object detection in computer vision (CV), (Zhou et al., 2018b) (Depierre et al., 2021) treat a grasp as a bounding box and introduce the notion of reference anchor box. They do not directly predict five parameter values of a grasp, but a deformation of a reference box. (Depierre et al., 2021) achieves 85.74% on the Jacquard

Dataset. (Morrison et al., 2018) (Kumra et al., 2020) use a generative model to predict a grasp at the pixel level. The output here is four images of the same size as the input. GR-ConvNet(Kumra et al., 2020) reaches the state-of-the-art performance of 94.6% on the Jacquard Dataset. In this paper, GR-ConvNet is used as the backbone.

GR-ConvNet. GR-ConvNet (Kumra et al., 2020) is based on GG-CNN (Morrison et al., 2018). Both networks use the depth image or RGB-D image as input and generate three pixel-wise maps: grasp quality, angle, and gripper width, respectively. Then they search local maximums in the grasp quality map for the positions for potential grasps. A complete grasp is generated with the value of angle and gripper width maps at these local maximum positions.The main difference between GR-ConvNet and GG-CNN resides in the fact that GR-ConvNet uses a deeper network and residual blocks, thereby resulting in a higher accuracy.

Continual Learning (CL) Scenarios. As discussed in the introduction section, different CL scenarios are possible. De Ven and Tolias (van de Ven and Tolias, 2019) makes use of the notion of task identity to distinguish three CL scenarios: task-incremental, domain-incremental or class-incremental CL. Many CL methods (Aljundi et al., 2017) (Rusu et al., 2016) target task-incremental learning, where task identity is provided during training and inference. It gives the possibility to add some specific parameters' responses to the corresponding task. In such a situation, a model is always composed of multiple submodels, and each corresponds to a task. However, in the case of grasp detection, the task identity is unknown, and we need a model that can react correctly to the input without task identities. Therefore, our use-case is closer to domain-incremental or class-incremental learning. The difference between them resides on whether the model predicts the task identity or not. If the output space changes between tasks, Class-incremental is more suitable. For example, adding new classes(for classification) means adding new neurons at the output layer, thereby changing also the output labels. However, if the output space does not change and there is no apparent boundaries between tasks, domain-incremental scenario is more appropriate. More specifically to classification, Core50 (Lomonaco and Maltoni, 2017) proposes three CL scenarios, namely NI, NC, and NIC. Compared with previous work, NI is similar to domain-incremental, NC similar to class-incremental. NIC is a kind of NC where task boundaries are fuzzy between tasks. For grasp detection, the output space doesn't change. Therefore, Domain-incremental or NI is more appropriate to our use-case. Specifically to NI, OpenLoRIS (She et al., 2020) changes the distributions of different tasks by adjusting some parameters, *e.g.*, illumination, occlusion, or clutter.

Continual Learning (CL) Methods. Most of the existing methods are aimed at the class-incremental learning. Therefore, they don't match our use-case. For grasp detection, the output of the network for the five grasp parameters doesn't change over tasks. As a result, only the CL methods or techniques using a static architecture is suitable for us.

EWC (Kirkpatrick et al., 2017) is the first work to estimate the importance of all parameters for previously learned tasks and penalize their changes in the future. SI (Zenke et al., 2017), Riemannian walk (Chaudhry et al., 2018) share the same ideal but propose different estimation methods. These methods can retain the previous knowledge, but they possibly penalize too many parameters important to the previous task, and it degrades the ability to learn new tasks.

LWF (Li and Hoiem, 2016) introduces the notion of distillation in continual learning. It needs the previously learned model to label the data of the current task (without memory) or previous task (with memory), then use these data together with current data in training. An essential aspect of this method is *knowledge distillation* which makes use of the whole output of the previous model to label the data, resulting in the so called "soft targets"(van de Ven and Tolias, 2019). Soft targets usually contain richer information about the previous model. But if the distribution of the current task is much different from the distribution of the previous task, this kind of label may not be useful. Based on the idea of distillation, PodNet (Douillard et al., 2020) labels the data not only using the output layer but also all the intermediate layers. This method performs better when the learning sequence is longer. (Ebrahimi et al., 2021) uses Grad-cam to label the data while (Kurmi et al., 2021) relies on uncertainty to label the data.

A-GEM (Chaudhry et al., 2019) stores a part of previous data called as episodic memory. When training, it computes a reference gradient by using a random batch from the episodic memory, then adjusting the current gradient to ensure the gradient to optimize does not violate the reference gradient. Thus, the loss of the previous task will not increase.

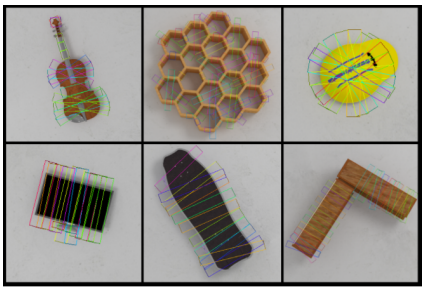

Figure 2: Samples from the Jacquard Dataset. Each rectangle represents a possible grasp. In this dataset, each object has five poses, and each pose has hundreds of possible grasps.

3 CONTINUAL GRASP DETECTION BENCHMARK

The Jacquard Dataset illustrated in Figure 2 (Depierre et al., 2018) is a large-scale synthetic dataset containing RGB-D images and rich annotations of successful grasps from a simulated environment. The Jacquard dataset comprises 54 485 different images from 11 619 distinct objects with a total of 4 967 454 grasps annotations.

3.1 Dataset

Because the Jacquard Dataset is based on ShapeNet (Chang et al., 2015), we can use WordNet synsets to regroup the dataset into 160 categories. As a result, the datatset can also be used for classification. The aim of this paper is not to achieve the state-of-the-art performance on grasp detection, but generate a continual learning benchmark. For this purpose, we choose from the Jacquard dataset five categories (airplane, desk lamp, bowl, soda can, pencil) that contain the most images and 150 images for each category, resulting in 750 images in total. From the developer version of the Jacquard Dataset, we can obtain the mask of an object and change the background and illumination conditions. Specifically, we keep the size and the pose of the object so that the ground truth grasp doesn't change but modify the background or the illumination conditions to create different tasks. Each task thus contains the same five categories and 750 images but with a different background or different illumination conditions. As such, ten different tasks have been created. The model is trained sequentially to these ten tasks.

3.2 Evaluation Metrics

3.2.1 Grasp Detection Metric

To determine whether a grasp is successful or not, we use the rectangle metric proposed by (Jiang et al., 2011). A successful grasp should satisfy the following two conditions as shown in Figure 4:

- The intersection over union(IOU) ratio between the prediction and the ground-truth grasp rectangle is over 25%.
- The predicted angle is less than 30° compared with the ground-truth.

We then propose two criteria to evaluate the performance of grasp detection :

- **One Grasp Accuracy (1-GA):** From one output of our model, we generate only one grasp. After passing all test images, we accumulate all successful grasps, then divide them by the number of test images.
- **Ten Grasps Accuracy (10-GA):** Here we generate ten grasps for one output. After passing all test images, we accumulate all successful grasps, then divide them by the number of attempted grasps.

Because the output of our model is an image of the same size as the input, **Ten grasps accuracy (10-GA)** can get much more detailed information and evaluate the quality of generated images more precisely.

3.2.2 Continual Learning Metric

Average Accuracy (a). Let $a_{i,j}$ be the accuracy evaluated on the data of *j-th* task after training the model incrementally from tasks 0 to *i*. Then the average accuracy after training on *i-th* task is then defined as $A_i = \frac{1}{10}\sum_{j=0}^{9} a_{i,j}$ there are ten tasks in total. The higher is A, the better is the model, but this does not provide any information about forgetting.

Backward Transfer (BF). the backward transfer after training on *i-th* task is defined as $BF_i = \frac{1}{i}\sum_{j=0}^{i}(a_{j,j} - a_{i,j})$ and $i \neq 0, j < i$. This criterion can estimate the influence that learning a task i has on the performance of a previous task $j < i$ (Chaudhry et al., 2018). The lower is BF, the better the model can overcome the forgetting.

4 EXPERIMENTS

GR-ConvNet (Kumra et al., 2020) has been chosen as the backbone model as it represents the state-of-the-art on grasp detection. It consists of basic blocks,

Figure 3: There are 10 tasks in our benchmark, T_i represents the i-th task. The difference between tasks is the background and illumination. T_5 correspond to the original Jacquard dataset. $T_0, T_2, T_3, T_4, T_6, T_8$ use a different background.T_1, T_7 use a different illumination. In T_9, we apply a gaussian filter to blur the input image.

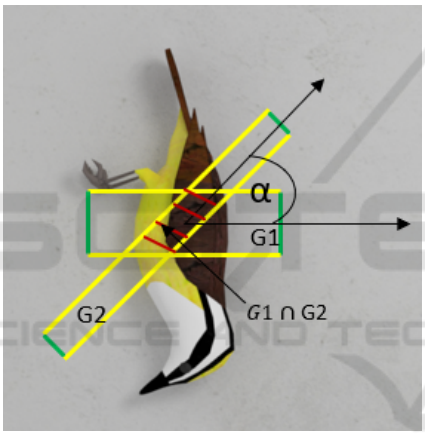

Figure 4: We present two grasps for an object given. G1: the ground truth and G2: the prediction. α represents the included angle.$G1 \cap G2$ represents the intersection of two grasps rectangles.

e.g., convolutional layers, batch normalization layers, *etc.* As a result, it is simple to adapt continual learning methods used for classification to fit GR-ConvNet chosen as backbone. To enable a model to predict both grasps and classes from an input RGB-D image, we add an extra head to predict the class label. The network architecture is shown in Figure 5.

Original RGB-D images are rendered in 224x224 resolution, then normalized. The annotations are transformed into four images via the method proposed in (Morrison et al., 2018). For each task, 416 images are used for training, 104 images for validation and 130 images for testing. Because the task order influence much, we run three experiments with different task orders and then output the average accuracy.

4.1 Methods

Using the NI scenario, we compare the following CL methods:

- Offline: The model is trained using the data from all the ten tasks at once. This is also called *joint training*, and the performance it achieves can be seen as an upper bound for CL methods.

- Fine-Tuning: The model is sequentially trained in the standard way. It uses the weights of the previously trained model as the initialization for the current task. Its performance constitutes the lower bound of CL methods.

- Online EWC / SI: One hyperparameter, namely λ, is introduced by these methods, to control the regularization strength as in the following formula : $L_{total} = L_{current} + \lambda L_{regularization}$. The value of λ is set by a grid search.

- LWF / PodNet with or Without Memory: The CL variant *without memory* means that the previously trained model is used to label the data of the current task. So for each training mini-batch, $L_{total} = \lambda_c L_{current} + \lambda_m L_{memory}$. We set $\lambda_c = \frac{1}{N_{tasks-so-far}}$ and $\lambda_m = (1 - \frac{1}{N_{tasks-so-far}})$ based on (van de Ven and Tolias, 2019). The CL variant *with memory* means that we store a small part of previous data into a memory buffer. We then random sample a mini-batch from the buffer when training the model for a new task. After training a task, 15 images from the task are randomly

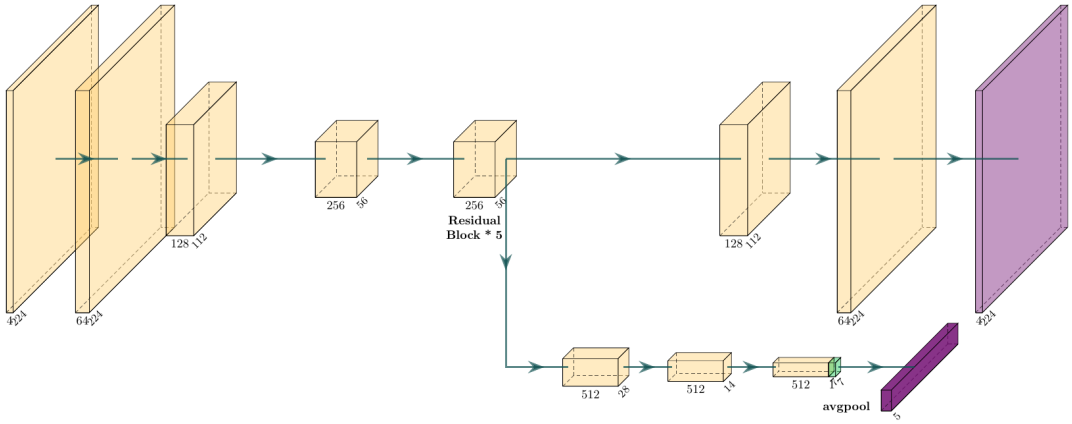

Figure 5: Architecture of the GR-ConvNet-based network used in our experiments. The input is a 224x224 RGB-D image. There are two outputs: the classification output with five classes, and the grasp output with a quality image, a width image, and two angle images, respectively as in (Morrison et al., 2018).

selected into the replay buffer. The rest of the algorithm stay the same.

- Memory Replay: Based on fine-tuning, we add a replay buffer that is the same as in **LWF with memory**. During training, a mini-batch is randomly sampled from the replay buffer and is used to compute a memory loss. The formula and the hyperparameters are the same as in **LWF with memory**. This method can be seen as a lower bound of CL algorithms that use memory.

- A-GEM: It uses the same replay buffer like **LWF with memory**. There is no hyperparameter, and we use the same strategy as (Chaudhry et al., 2019).

4.2 Training Details

For all tasks, we turn on the data augmentation that includes random rotations (0,$\pi/2$,π,$3\pi/2$) and random zooms from 0.6 to 1.0. All models are trained 20 iterations per task using the ADAM-optimizer and the model that performs the best on the validation set of the current task is selected for testing. The learning rate is initially set to 0.00005 and a learning rate decay $lr = 0.9 * iteration^2$ is used. For each task, the optimizer is reinitialized.

5 RESULTS

As can be seen from Table 1, the CL methods without memory struggle on our benchmark for grasp detection, as they display almost the same performance or even worse compared to the naive fine-tuning. The methods using memory perform well and are very close to the upper bound, but these methods have no obvious advantage over the Native memory replay4.1 method. Our benchmark evidences that there exists a big gap between methods with and without memory. EWC and SI perform better in overcoming the forgetting (the lowest BF) for methods without memory, but LWF and PodNet are nearly helpless and even have negative effects. In our benchmark, the change of background can lead to a drift of distribution, especially from a complex background to a simple one. This severe gap between distributions of different tasks can make LWF and PodNet useless, because these methods use the previous model to label the current data. For methods with memory, PodNet achieves the best score in terms of BF. In this case, PodNet uses the previous model to label previous data by using all intermediate layers' output, and suggests that this kind of annotation has rich information about the previous model, thereby helpful to overcome forgetting.

6 LIMITATIONS

Our benchmark are primarily designed for changes in background and illumination conditions over different tasks. However, for a real-work scenario, the classes and the poses may also increase incrementally.

7 CONCLUSION

We proposed a novel CL benchmark for vision-based grasp detection. We overviewed some continual learning methods and extended them to our benchmark to test their performance. Our experimental re-

Table 1: Comparison of state-of-the-art continual learning methods on our benchmark. For each method, we run three experiments with different task orders. A1: one-grasp accuracy; A10: ten-grasp accuracy; Ac: classification accuracy; BF1: one-grasp backward transfer; BF10: ten-grasp backward transfer; BFc: classification backward transfer. From the table, we find that there is no obvious improvement by using the state-of-the-art incremental learning methods compare with using Fine-tuning and Memory Replay.

Methods without memory	A1 (%)	A10 (%)	Ac (%)	BF1 (%)	BF10 (%)	BFc (%)
Offline(upper-bound)	80.18	73.34	97.34			
Fine-tuning(ft)	73 ± 4	50 ± 12	76 ± 15	-1 ± 2	13 ± 12	11 ± 13
Online EWC (Kirkpatrick et al., 2017)	$\mathbf{74 \pm 4}$	50 ± 10	$\mathbf{77 \pm 15}$	$\mathbf{-1 \pm 2}$	12 ± 13	10 ± 12
SI (Zenke et al., 2017)	73 ± 4	$\mathbf{51 \pm 10}$	76 ± 12	0 ± 2	11 ± 13	11 ± 12
LWF without memory (Li and Hoiem, 2016)	65 ± 2	44 ± 2	50 ± 10	1 ± 1	3 ± 6	$\mathbf{8 \pm 1}$
PodNet without memory (Douillard et al., 2020)	66 ± 2	46 ± 2	49 ± 7	1 ± 1	$\mathbf{2 \pm 6}$	9 ± 1

Methods with memory (20 images for each task)	A1 (%)	A10 (%)	Ac (%)	BF1 (%)	BF10 (%)	BFc (%)
Memory Replay	77 ± 1	65 ± 5	92 ± 3	-3 ± 2	-3 ± 5	-6 ± 2
A-GEM (Chaudhry et al., 2019)	76 ± 1	62 ± 8	90 ± 5	-2 ± 2	-1 ± 6	-3 ± 5
LWF (Li and Hoiem, 2016)	77 ± 1	66 ± 3	$\mathbf{93 \pm 3}$	-4 ± 2	$\mathbf{-4 \pm 3}$	$\mathbf{-7 \pm 1}$
PodNet (Douillard et al., 2020)	77 ± 1	$\mathbf{67 \pm 3}$	92 ± 2	-4 ± 2	-4 ± 5	-6 ± 2

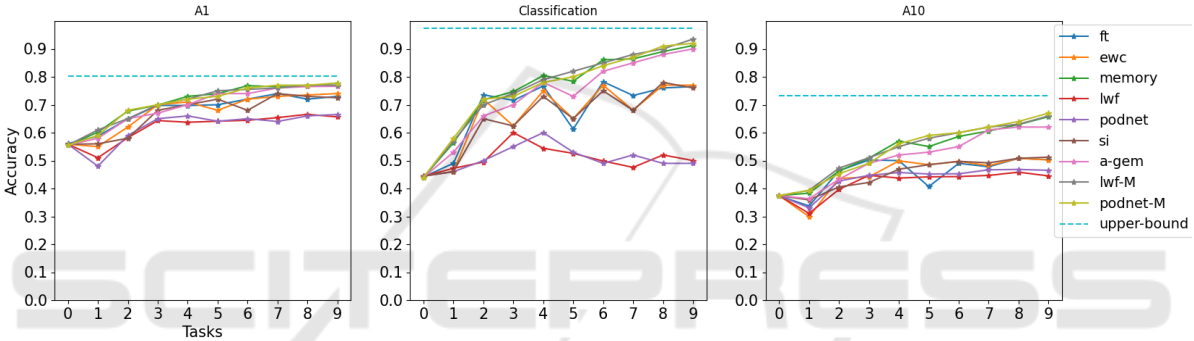

Figure 6: Validation accuracy (averaged on 3 runs) after training on each task for different methods.

sults show that these CL methods don't display significant improvement over the lower bound, thereby suggesting the necessity of a replay buffer to retain knowledge. However, in real-life situations, past data are not always available, therefore, how to get a comparable accuracy without memory is an issue for future research.

ACKNOWLEDGMENTS

This work was in part supported by the French national program of investment of the future and the regions through the PSPC FAIR Waste project, and the French Research Agency, l'Agence Nationale de Recherche (ANR), through the projects Learn Real (ANR-18-CHR3-0002-01), Chiron (ANR-20-IADJ-0001-01) and Aristotle (ANR-21-FAI1-0009-01), as well as the 4D Vision project funded by the Partner University Fund (PUF), a FACE program.

REFERENCES

Aljundi, R., Chakravarty, P., and Tuytelaars, T. (2017). Expert gate: Lifelong learning with a network of experts. *CVPR*.

Bang, J., Kim, H., Yoo, Y., Ha, J.-W., and Choi, J. (2021). Rainbow memory: Continual learning with a memory of diverse samples. *CVPR*.

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., and Yu, F. (2015). Shapenet: An information-rich 3d model repository.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. S. (2018). Riemannian walk for incremental learning: Understanding forgetting and intransigence. *CVPR*, page 556–572.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. (2019). Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*.

Depierre, A., Dellandréa, E., and Chen., L. (2018). Jacquard: A large scale dataset for robotic grasp detection. *CoRR*.

Depierre, A., Dellandréa, E., and Chen, L. (2021). Scoring

graspability based on grasp regression for better grasp prediction. *ICRA*.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. (2020). Podnet: Pooled outputs distillation for small-tasks incremental learning. *ECCV*.

Ebrahimi, S., Petryk, S., Gokul, A., Gan, W., Gonzalez, J. E., Rohrbach, M., and Darrell, T. (2021). Remembering for the right reasons: Explanations reduce catastrophic forgetting. *ICLR*.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. (2014). An empirical investigation of catastrophic forgeting in gradientbased neural networks. In *In Proceedings of International Conference on Learning Representations (ICLR*.

Jiang, Y., Moseson, S., and Saxena, A. (2011). Efficient grasping from rgbd images: Learning using a new rectangle representation. In *2011 IEEE International Conference on Robotics and Automation*, pages 3304–3311.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

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

Kurmi, V. K., Patro, B. N., Subramanian, V. K., and Namboodiri, V. P. (2021). Do not forget to attend to uncertainty while mitigating catastrophic forgetting. *WACV*.

Li, Z. and Hoiem, D. (2016). Learning without forgetting. *ECCV*, abs/1606.09282.

Lomonaco, V. and Maltoni, D. (2017). Core50: a new dataset and benchmark for continuous object recognition. *CoRR*, abs/1705.03550.

Lopez-Paz, D. and Ranzato, M. (2017). Gradient episodic memory for continual learning. *NIPS*.

Morrison, D., Corke, P., and Leitner, J. (2018). Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach. In *Proc. of Robotics: Science and Systems (RSS)*.

Redmon, J. and Angelova, A. (2015). Real-time grasp detection using convolutional neural networks. *ICRA*.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *CoRR*, abs/1606.04671.

She, Q., Feng, F., Hao, X., Yang, Q., Lan, C., Lomonaco, V., Shi, X., Wang, Z., Guo, Y., Zhang, Y., Qiao, F., and Chan, R. H. M. (2020). Openloris-object: A robotic vision dataset and benchmark for lifelong deep learning. *ICRA*.

Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). Continual learning with deep generative replay. *NIPS*.

van de Ven, G. M. and Tolias, A. S. (2019). Three scenarios for continual learning. *CoRR*, abs/1904.07734.

Verma, V. K., Liang, K. J., Mehta, N., Rai, P., and Carin, L. (2021). Efficient feature transformations for discriminative and generative continual learning. *CVPR*.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. *ICML*.

Zhou, X., Lan, X., Zhang, H., Tian, Z., Zhang, Y., and Zheng, N. (2018a). Fully convolutional grasp detection network with oriented anchor box. *IROS*, abs/1803.02209.

Zhou, X., Lan, X., Zhang, H., Tian, Z., Zhang, Y., and Zheng, N. (2018b). Fully convolutional grasp detection network with oriented anchor box. *IROS*.