frontiers in Artificial Intelligence

BRIEF RESEARCH REPORT
published: 04 February 2022
doi: 10.3389/frai.2022.824655

Catastrophic Forgetting in Deep Graph Networks: A Graph Classification Benchmark

Antonio Carta[1]*, *Andrea Cossu*[1,2], *Federico Errica*[1] and *Davide Bacciu*[1]

[1] Computer Science Department, University of Pisa, Pisa, Italy, [2] Scuola Normale Superiore, Pisa, Italy

In this work, we study the phenomenon of catastrophic forgetting in the graph representation learning scenario. The primary objective of the analysis is to understand whether classical continual learning techniques for flat and sequential data have a tangible impact on performances when applied to graph data. To do so, we experiment with a structure-agnostic model and a deep graph network in a robust and controlled environment on three different datasets. The benchmark is complemented by an investigation on the effect of structure-preserving regularization techniques on catastrophic forgetting. We find that replay is the most effective strategy in so far, which also benefits the most from the use of regularization. Our findings suggest interesting future research at the intersection of the continual and graph representation learning fields. Finally, we provide researchers with a flexible software framework to reproduce our results and carry out further experiments.

Keywords: continual-learning, lifelong-learning, catastrophic-forgetting, deep-graph-networks, benchmarks

OPEN ACCESS

Edited by:
Andrea Soltoggio, Loughborough University, United Kingdom

Reviewed by:
Pawel Ladosz, Ulsan National Institute of Science and Technology, South Korea
Andri Ashfahani, ITS, Indonesia

***Correspondence:**
Antonio Carta
antonio.carta@di.unipi.it

Specialty section:
This article was submitted to Machine Learning and Artificial Intelligence, a section of the journal Frontiers in Artificial Intelligence

Received: *29 November 2021*
Accepted: *11 January 2022*
Published: *04 February 2022*

Citation:
Carta A, Cossu A, Errica F and Bacciu D (2022) Catastrophic Forgetting in Deep Graph Networks: A Graph Classification Benchmark. Front. Artif. Intell. 5:824655. doi: 10.3389/frai.2022.824655

1. INTRODUCTION

Building a robust machine learning model that incrementally learns from different tasks without forgetting requires methodologies that account for drifts in the input distribution. The Continual Learning (CL) research field addresses the catastrophic forgetting problem (Grossberg, 1980; French, 1999) by devising learning algorithms that improve a model's ability to retain previously gathered information while learning across multiple steps. Each step in a CL scenario constitutes a new learning experience providing new data to the model, whose distribution may be different with respect to the previously encountered ones. As of today, CL methods have been studied from the perspective of flat data (data without a strong temporal or geometrical structure) (Kirkpatrick et al., 2017; Shin et al., 2017; Maltoni and Lomonaco, 2018) and, to a lesser extent, sequential data (Ehret et al., 2020; Sodhani et al., 2020; Cossu et al., 2021). In particular, the literature on CL revolves around three main families of strategies aimed at tackling catastrophic forgetting (Parisi et al., 2019): regularization strategies, architectural strategies and replay strategies. Though not entirely comprehensive, this taxonomy includes most of the currently used CL strategies.

Regularization strategies add a penalization to the standard loss function to enforce the stability of existing parameters. For example, the penalization may force parameters deemed important for a specific task not to change much during training (Kirkpatrick et al., 2017), or it may impose stability of the output activations during different tasks through distillation (Li and Hoiem, 2016). **Architectural strategies** try to mitigate forgetting by enhancing the model's plasticity. Typically, they expand the network by adding more units (Marsland et al., 2002; Draelos et al., 2017), an entirely new module (Rusu et al., 2016; Cossu et al., 2020), or by expanding and then compressing

the resulting architecture (Hung et al., 2019; Srivastava et al., 2019). Finally, **replay strategies** mix input patterns from the current step with patterns from previously encountered steps (Isele and Cosgun, 2018; Rolnick et al., 2019). Replay memory management is crucial because it is not feasible to store all the patterns from previous steps. Generative replay, instead, overcomes this problem by training a generative model (with fixed space occupancy) that provides on-demand previous patterns (Shin et al., 2017; Wang et al., 2019; van de Ven et al., 2020).

Graph Representation Learning (GRL) is the study of machine learning models that can make predictions about input data represented as a graph. GRL methods naturally find application in social sciences (Nechaev et al., 2018), recommender systems (Bobadilla et al., 2013), cheminformatics (Micheli et al., 2007), security (Iadarola, 2018), and natural language processing (Marcheggiani et al., 2018), where each graph has a potentially different topology (Micheli et al., 2007).

There is a long and consolidated history of works that discuss these problems in static scenarios, where data is completely available from the beginning (Sperduti and Starita, 1997; Frasconi et al., 1998; Micheli, 2009; Scarselli et al., 2009). Nowadays, the models that can process a broad spectrum of graphs by means of local and iterative processing of information are called Deep Graph Networks[1] (DGNs) (Bacciu et al., 2020). Generally speaking, DGNs propagate nodes' information across the graph by stacking several graph convolutional layers on top of each other. Each layer works by aggregating each node's neighboring information, and it ultimately produces node representations that can be used to make predictions about nodes, links, or entire graphs. For the sake of brevity, we refer the reader to recent works that summarize the state of the art (Bronstein et al., 2017; Battaglia et al., 2018; Bacciu et al., 2020; Wu et al., 2020).

At present, the literature lacks an analysis of catastrophic forgetting in models that deal with graphs. The few existing works focus on new approaches which are not compared to existing CL strategies on challenging benchmarks (Wang et al., 2020; Zhou and Cao, 2021). This work makes the first step in this direction by carrying out continual learning experiments on graph classification benchmarks in a robust and controlled framework. In this context, we investigate whether specific GRL regularization strategies can mitigate catastrophic forgetting by enforcing structural information preservation.

Our contribution is two-fold. First of all, we study whether CL techniques for flat data still work on the graph domain. If that is not the case, the results will call for different and novel approaches to be developed. Secondly, we provide a robust and reproducible framework to carry out Continual Learning experiments on graph-structured data. Indeed the GRL field has suffered serious reproducibility issues that impacted chemical and social benchmarks (Shchur et al., 2018; Errica et al., 2020). By publicly releasing our code and adopting a clear experimental evaluation, we prevent common malpractices such as the usage of custom data splits for model selection and model assessment, the absence of a model selection, and incorrect evaluations of the estimated risk on the validation (rather than test) set.

[1]This term disambiguates the more common "Graph Neural Networks" (GNN), which refers to the work of Scarselli et al. (2009).

2. METHOD

We now detail the CL strategies and deep graph networks used to evaluate catastrophic forgetting in the domain of graph-structured data. To the best of our knowledge, this is one of the first studies to investigate this particular aspect. To keep the discussion clear, we will focus on regularization and replay strategies applied to simple architectures for graphs, deferring more complex techniques to future studies.

2.1. Continual Learning Strategies

2.1.1. Elastic Weight Consolidation

Elastic Weight Consolidation (Kirkpatrick et al., 2017) is a regularization technique which prevents changes in parameters that are important for previous steps. Formally, EWC adds a squared penalty term $\mathcal{R}$ to the classification loss at training time:

$$\mathcal{R}(\Theta, \Omega) = \lambda \sum_{i=1}^{n-1} \Omega_{\mathbf{i}} \|\Theta_{\mathbf{i}} - \Theta_{\mathbf{n}}\|_2^2, \tag{1}$$

where $\Theta_{\mathbf{n}}$ is the vector of parameters of current step n, $\Theta_{\mathbf{i}}$ is the vector of parameters from previous step i and $\Omega_{\mathbf{i}}$ is the vector of parameter importances for step i. The hyperparameter λ controls the trade-off between classification accuracy on current step and stability of parameters. The importance for step n is computed at the end of training on step n, through a diagonal approximation of the Fisher Information Matrix:

$$\Omega_{\mathbf{n}} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\in\mathcal{D}}\left[(\nabla_{\Theta_{\mathbf{n}}} \log p_{\Theta_{\mathbf{n}}}(\mathbf{y}|\mathbf{x}))^2\right]. \tag{2}$$

The computation of importance values requires an additional pass over the training data $\mathcal{D}$ and the estimation of the log probabilities $\log p_\Theta$ represented by the network outputs. Following Schwarz et al. (2018), we keep a single importance matrix for all steps, by summing the importance on the current step with the previous values. In order to prevent the unbounded growth of importance values we normalize between 0 and 1 when computing importance on the current step.

2.1.2. Learning Without Forgetting

Learning without Forgetting (LwF) (Li and Hoiem, 2016) is a regularization technique which preserves the knowledge of previous steps by fostering stability at the activation level through knowledge distillation (Hinton et al., 2015). The method adds a regularization term $\mathcal{R}$ to the loss during step n as follows:

$$\mathcal{R}(\Theta_{\mathbf{n}}, \Theta_{\mathbf{n-1}}; \mathbf{x}, \mathbf{y}) = \alpha\ \mathrm{KL}[p_{\Theta_{\mathbf{n}}}(\mathbf{y}|\mathbf{x}) \,||\, p_{\Theta_{\mathbf{n-1}}}(\mathbf{y}|\mathbf{x})], \tag{3}$$

where α controls the regularization strength. The KL-divergence term prevents current activations to diverge too much from the ones of the model at previous step.

2.1.3. Replay

Replay of previous patterns during training is a very effective technique against forgetting of existing knowledge (Hayes et al., 2018; Aljundi et al., 2019; Chaudhry et al., 2019b; Rolnick et al., 2019). We leveraged a replay memory which stores a fixed number of patterns for each class. During training on each step, the replay memory is concatenated with the training set. The resulting dataset is shuffled and used for training the model. Therefore, replay patterns are spread uniformly over the training set.

2.1.4. Naïve

The Naïve strategy trains the model continuously without applying any CL technique. This strategy is heavily subjected to catastrophic forgetting. Therefore, it can be used as a baseline to compare the performance of more effective CL strategies, which should perform significantly better in terms of forgetting.

2.2. Deep Graph Networks Models

We define a graph as a tuple $g = (\mathcal{V}_g, \mathcal{E}_g, \mathcal{X}_g, \mathcal{A}_g)$ where $\mathcal{V}_g$ is the set of *nodes*, $\mathcal{E}_g$ is the set of oriented *edges* connecting ordered pairs of nodes, whereas $\mathcal{X}_g$ (respectively $\mathcal{A}_g$) denotes node (edge) features. The neighborhood $\mathcal{N}_v$ of a node v is the set of all nodes u for which an edge (u, v) directed toward v exists.

2.2.1. Structure-Agnostic Baseline

To assess whether continual learning strategies have an impact when working with graphs, we must first devise a baseline that ignores the structural information and relies only on node features. The most common baseline we find in the literature (Dwivedi et al., 2020; Errica et al., 2020) is a multi-layer perceptron (MLP) that is invariant to the ordering of the nodes. Formally, the baseline compute a node representation $\mathbf{h}_v$ as follows

$$\mathbf{h}_v = \psi(\mathbf{x}_v), \;\; x_v \in \mathcal{X}_g, \tag{4}$$

$$\psi(x_v) = \mathbf{W}_L^T(\sigma(\dots(\sigma(\mathbf{W}_1^T x_v + \mathbf{b}_1)\dots) + \mathbf{b}_L), \tag{5}$$

where $\psi(\cdot)$ is an MLP of L layers, the symbol $\mathbf{W}$ denotes a weight matrix and $\mathbf{b}$ is the bias. As the tasks under consideration in this paper deal with graph classification, an additional *readout* phase is necessary, in which we aggregate all node representations into a single graph representation $\mathbf{h}_g$:

$$\mathbf{h}_g = \Psi_g\Big(\{\mathbf{h}_v \mid v \in \mathcal{V}_g\}\Big), \tag{6}$$

where Ψ_g is a permutation invariant function; in this work we will use the *mean* function as the baseline's readout.

2.2.2. Deep Graph Networks

While DGNs usually adopt the same readout scheme as the one of Equation 6, the fundamental difference lies in its graph convolutional layer. If we assume a deep network of L layers, the node representation at layer $\ell < L$, that is, $\mathbf{h}_v^\ell$ is obtained by aggregating the neighboring information of all nodes using another permutation invariant function Ψ_n:

$$\mathbf{h}_v^{\ell+1} = \phi^{\ell+1}\Big(\mathbf{h}_v^\ell, \; \Psi_n(\{\psi^{\ell+1}(\mathbf{h}_u^\ell) \mid u \in \mathcal{N}_v\})\Big), \tag{7}$$

where ϕ and ψ are usually implemented as linear layers or MLPs.

In our experiments, we define Ψ_n as the *mean* operator for digit classification tasks and sum for the chemical ones.

2.2.3. Structure-Preserving Regularization Loss

We believe it is worth investigating whether a structure-preserving regularization loss such as the one of Kipf and Welling (2017) affects catastrophic forgetting when used alongside the various CL strategies. The catch is that regularization will help preserve the output of previously seen classes when similar structural patterns appear in the new training samples. In general, the interplay between GRL and CL regularization strategies opens appealing research directions for the future. In case the chosen regularization does not help, this may indicate that the distribution of neighbor states of patterns belonging to a new class is radically different from those seen before.

3. RESULTS

This section provides a thorough description of the experimental details necessary to reproduce our experiments and of the results we obtained. The code is made publicly available to reproduce the results and carry out novel robust evaluations of different continual learning strategies[2].

In all our experiments, we performed model selection on the validation set using a grid-search strategy for all the implemented models. Regardless of the dataset or continual learning technique used, we selected the number of layers in $\{2, 4\}$ for the DGN and 4 for the baseline. In both cases, the dimension of the hidden layer was chosen in $\{64, 128\}$. The number of epochs was set to 200 (patience = 20) for the Baseline and to 1,000 for DGN and DGN+Reg (patience = 50). The learning rate was set to 0.001, and the optimizer chosen was Adam. We used the "sum" version of the EWC combined with normalized importance scores. Being LWF very sensible to the hyper-parameters, we chose $\alpha \in \{0.5, 1.0, 2.0\}$ and the temperature in $\{0.5, 1.0, 2.0\}$.

3.1. Datasets

The evaluation is carried out on three different large graph classification datasets. The former two, namely MNIST and CIFAR10, are the standard digit classification benchmarks used in the CL literature. However, here the digits are represented as graphs of varying dimension and shape (Dwivedi et al., 2020). The nodes are "superpixels" obtained through a specific coarsening process, and the adjacency information is constructed using the k-nearest neighbor algorithm. We defer the specifics of this process to the original paper. The third dataset is OGBG-PPA (Hu et al., 2020), a dataset of undirected protein association neighborhoods taken from protein-protein interaction graphs. Here, the task is to classify each input as one of 37 different taxonomy groups. Here, node features are missing but edges contain information. As such, we treat edges as nodes in the structure-agnostic baseline. We use the same data splits as those provided in the original papers, thus performing standard hold-out model selection and assessment. We also use the

[2]https://github.com/diningphil/continual_learning_for_graphs

TABLE 1 | Summary of the datasets statistics.

	MNIST	CIFAR10	OGBG-PPA
Size	70,000	60,000	158,100
Node attrs.	3	5	0
Edge attrs.	0	0	7
Classes	10	10	37
Avg $\lvert\mathcal{V}_g\rvert$	70.57	117.63	243.4
Avg $\lvert\mathcal{E}_g\rvert$	564.63	941.07	2266.1
Data split	55K/5K/15K	45K/5K/15K	49%/29%/22%
Class split	2+2+2+2+2	2+2+2+2+2	17+5+5+5+5

"Class split" refers to how we group classes in the Split CL experiment.

TABLE 2 | Mean accuracy and mean standard deviation among all steps.

	Model	Strategy			
		Naïve	EWC	Replay	LWF
MNIST	Baseline	$19.56_{\pm 0.1}$	$19.39_{\pm 0.1}$	$86.13_{\pm 4.5}$	$33.16_{\pm 13.1}$
	DGN	$19.19_{\pm 0.1}$	$18.95_{\pm 0.3}$	$79.52_{\pm 1.9}$	$32.64_{\pm 5.0}$
	DGN+reg	$19.31_{\pm 0.1}$	—	$81.42_{\pm 2.4}$	—
CIFAR10	Baseline	$17.49_{\pm 0.1}$	$17.49_{\pm 0.1}$	$42.87_{\pm 3.7}$	$26.77_{\pm 5.1}$
	DGN	$17.11_{\pm 0.2}$	$17.10_{\pm 0.2}$	$39.55_{\pm 2.3}$	$24.13_{\pm 4.1}$
	DGN+reg	$17.13_{\pm 0.1}$	—	$46.61_{\pm 3.5}$	—
OGBG-PPA	Baseline	$14.53_{\pm 0.5}$	$13.90_{\pm 0.8}$	$55.96_{\pm 3.0}$	$20.83_{\pm 6.1}$
	DGN	$14.47_{\pm 0.3}$	$14.15_{\pm 0.5}$	$56.34_{\pm 2.5}$	$18.46_{\pm 5.4}$
	DGN+reg	$15.18_{\pm 0.8}$	—	$57.27_{\pm 3.2}$	—

Smaller accuracy results in larger forgetting of previous knowledge. Replay results are related to memory size of 1,000. Results are averaged over 5 final runs. We treat the regularization loss as a separate strategy.

readily available version of all datasets provided by the Pytorch Geometric library (Fey and Lenssen, 2019). **Table 1** summarizes some useful dataset statistics.

3.2. Continual Learning Evaluation Protocol

We evaluated each model in the class-incremental scenario, a popular continual learning setting where new classes arrive over time (van de Ven and Tolias, 2018). When a new step arrives, the model is trained on the new data without using data from the previous steps (except for the replay buffer, when used). We use single-head models, where the entire output layer is used at each step. This is one of the most challenging scenarios for the mitigation of catastrophic forgetting in CL. **Table 1** shows the class splits for each dataset, highlighting how many new classes are present in each step. We monitor the metric $\text{ACC} = \frac{1}{T}\sum_{t=1}^{T} R_{T,t}$, introduced in Lopez-Paz and Ranzato (2017), where $R_{T,t}$ is the accuracy on step t after training on step T.

We reported the average ACC and its standard deviation computed over 5 runs. Larger final accuracy corresponds to a smaller degree of forgetting, sometimes also referred to as Negative Backward Transfer in the continual learning literature (Lopez-Paz and Ranzato, 2017). We evaluated the performance by computing the mean accuracy over all the steps after training on all steps.

3.3. Catastrophic Forgetting Analysis

The empirical results suggest that Deep Graph Networks trained continuously are subjected to catastrophic forgetting of previous knowledge. **Table 2** reports the average ACC across all steps (see also **Figure 1** for an intuitive visualization of results). We also extend the results presented in Lesort et al. (2020) to Deep Graph Networks: importance-based regularization strategies are not able to prevent forgetting in class-incremental scenarios. In fact, in our experiments EWC always performs comparably to the *Naïve* strategy.

Interestingly, Deep graph networks do not provide significant performance improvements with respect to a structure-agnostic baseline. This is a surprising result, which might have two complementary explanations. The first is that the neighboring states' distribution of different classes varies, thus making the previously trained graph convolutions inadequate for subsequent tasks. The second, instead, relates to the nature of the class-incremental scenario. Since the model sees few classes at a time, each training task becomes so simple that the model ends up relying on node features only to discern between the two classes. This is confirmed by the fact that, when encouraged to retain structural information *via* the regularization term, DGN shows a slight increase in performance with the replay strategy. We believe that addressing both points in more detail could constitute interesting future work at the intersection of the two research fields.

3.3.1. Sensitivity of LwF to Hyperparameters

Not all regularization strategies are, however, subjected to forgetting. In fact, we show that LwF is able to recover part of the original knowledge, outperforming both Naïve and EWC. We also found LwF to be very sensitive to the choice of the hyperparameters. In particular, the softmax temperature and the hyperparameter α, which controls the amount of knowledge distillation heavily influence the final performance. In order to best show the sensitivity of LwF to the choice of hyperparameters, we computed the mean ACC and its standard deviation across all runs of model selection (**Figure 2**). Then, we compared the results with the best performance we found during model assessment. The difference highlights the high sensitivity of Lwf which could partially limit its applicability in real world applications, where it may be impossible to perform appropriate model selection in continual learning scenarios (Chaudhry et al., 2019a).

3.3.2. Effectiveness of Replay

Replay strategy is considered among the strongest CL strategies available. In our experiments, replay consistently outperforms all the other strategies. **Supplementary Material** shows ACC values for increasing replay memory sizes. Deep graph networks and baseline models require a comparable amount of replay to obtain

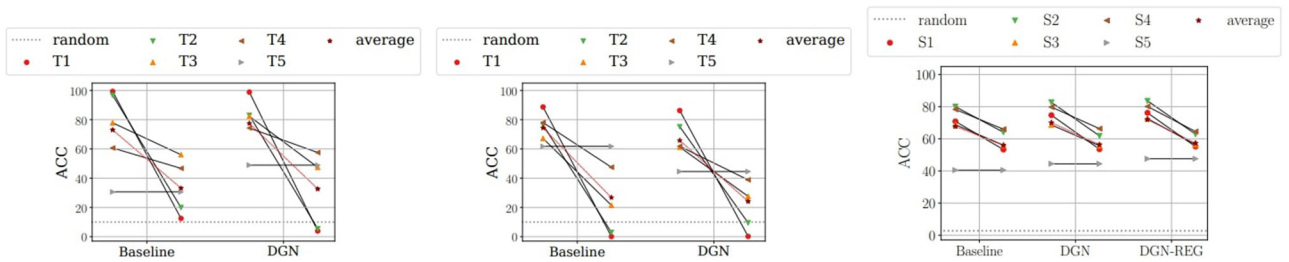

FIGURE 1 | Paired plots showing the ACC on each step for different models for LWF and MNIST **(left)**, LWF and CIFAR10 **(middle)**, Replay and OGBG-PPA **(right)**. Complete plots in the **Supplementary Material**. Each column refers to a model and it is composed by pairs of connected points. Each pair refers to a specific step. The leftmost point in the pair represents ACC after training on that specific step. The rightmost point represents ACC after training on all steps. The more vertical the line connecting the points, the larger the forgetting effect. The dashed horizontal line indicates the performance of a random classifier. The red star represents the average performance over all steps.

the same level of performance. Therefore, replay seems to behave as a good model-agnostic strategy even in the domain of graphs.

4. DISCUSSION

Learning from a data stream in a continual fashion is a key property of many biological systems, including the human brain. In fact, continual learning is often considered as a necessary condition for the development of artificial intelligent agents operating in the real world. Even though there exist continual learning strategies loosely inspired by neuroscience, most of them do not take into consideration that we, as humans, act in a world filled with structured and complex interactions between different objects. On one side, applying graph representation learning techniques to continual learning may lead to the acquisition of durable knowledge, since each piece of information is related to many others and it may be therefore more difficult to forget it. On the other side, our results highlighted that before being able to achieve this objective, it will be necessary to design *ad-hoc* continual learning strategies which explicitly take into consideration the structure and relations present in the data.

Our empirical evaluation of continual learning strategies with graph-structured data focused on the catastrophic forgetting phenomenon which affects deep graph networks in class-incremental scenarios. We evaluated a number of existing CL approaches and we discussed whether they are able to retain previous knowledge when applied to deep graph networks.

Interestingly, while graph networks outperform feedforward baselines during offline training, our results show that this advantage disappears in continual learning scenarios. This suggests that structure-preserving regularization techniques may help DGNs to mitigate catastrophic forgetting. Nonetheless, the results are still far from the performance achieved in the offline setting, where all data is available at the beginning of training. This can be easily seen by looking at the performance of the replay strategy, which largely improves over all the other strategies. Since replay approximates the offline training regime for large replay memory sizes, its performance can be considered as an upper-bound for the other continual learning strategies. Unfortunately, storing previous patterns is not always possible in real-world environments (e.g., due to privacy reasons or memory constraints). The design of *ad-hoc* DGNs and regularization techniques constitutes a valid, replay-free alternative. However, it would be interesting to limit the disadvantages of replay without sacrificing its performance. For example, latent replay approaches do not store raw input patterns to rehearse previous knowledge, but only latent and hidden activations of the model. In the presence of graph-structured data, storing few activations may be enough to reconstruct significant portions of the others. In the future, as supported by Hayes et al. (2021), latent replay on structured data can be empowered by better understanding the role played by replay in biological systems and in the human brain, where partial stimuli are often sufficient to reconstruct previous experiences in detail.

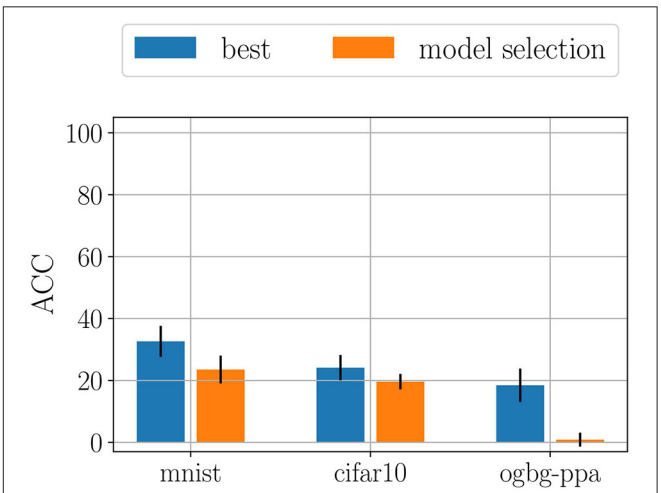

FIGURE 2 | Comparison of performances between model selection (averaged across all configurations) and model assessment (averaged across 5 final training runs). The difference highlights the sensitivity of LwF to the choice of hyperparameters. Most of the configurations cause larger forgetting effects with respect to the best configuration found by model selection.

By releasing the code of our experiments, and by providing a robust evaluation protocol for continual learning on some graph classification tasks, we hope to contribute to further progresses in the understanding of how novel continual learning strategies can be applied to the domain of graphs.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

ACa, ACo, and FE contributed to the definition of the benchmark, the experimental setup, and the evaluation of the experiments. DB contributed to writing the paper and to provide feedbacks on the future impacts of the work. All authors contributed to the article and approved the submitted version.

FUNDING

This work has been partially supported by the European Community H2020 programme under project TEACHING (Grant No. 871385).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2022.824655/full#supplementary-material

REFERENCES

Aljundi, R., Belilovsky, E., Tuytelaars, T., Charlin, L., Caccia, M., Lin, M., et al. (2019). "Online continual learning with maximal interfered retrieval," in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc.), 11849–11860. doi: 10.1109/CVPR.2019.01151

Bacciu, D., Errica, F., Micheli, A., and Podda, M. (2020). A gentle introduction to deep learning for graphs. *Neural Netw*. 129, 203–221. doi: 10.1016/j.neunet.2020.06.006

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv [Preprint]*. arXiv:1806.01261.

Bobadilla, J., Ortega, F., Hernando, A., and Gutierrez, A. (2013). Recommender systems survey. *Knowl. Based Syst*. 46, 109–132. doi: 10.1016/j.knosys.2013.03.012

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond Euclidean data. *IEEE Signal Process. Mag*. 34, 18–42. doi: 10.1109/MSP.2017.2693418

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. (2019a). "Efficient lifelong learning with A-GEM," in *ICLR*.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H. S., et al. (2019b). On tiny episodic memories in continual learning. *arXiv [Preprint]*. arXiv:1902.10486.

Cossu, A., Carta, A., and Bacciu, D. (2020). "Continual learning with gated incremental memories for sequential data processing," in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN 2020)* (Glasgow: IEEE). doi: 10.1109/IJCNN48605.2020.9207550

Cossu, A., Carta, A., Lomonaco, V., and Bacciu, D. (2021). Continual learning for recurrent neural networks: an empirical evaluation. *Neural Netw*. 143, 607–627. doi: 10.1016/j.neunet.2021.07.021

Draelos, T. J., Miner, N. E., Lamb, C., Cox, J. A., Vineyard, C. M., Carlson, K. D., et al. (2017). "Neurogenesis deep learning," in *IJCNN*.

Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., and Bresson, X. (2020). Benchmarking graph neural networks. *arXiv [Preprint]*. arXiv:2003.00982.

Ehret, B., Henning, C., Cervera, M., Meulemans, A., Oswald, J. V., and Grewe, B. F. (2020). "Continual learning in recurrent neural networks," in *International Conference on Learning Representations* (Zürich).

Errica, F., Podda, M., Bacciu, D., and Micheli, A. (2020). "A fair comparison of graph neural networks for graph classification," in *Proceedings of the 8th International Conference on Learning Representations (ICLR)* (Pisa).

Fey, M., and Lenssen, J. E. (2019). "Fast graph representation learning with PyTorch Geometric," in *Workshop on Representation Learning on Graphs and Manifolds, International Conference on Learning Representations (ICLR)* (Dortmund).

Frasconi, P., Gori, M., and Sperduti, A. (1998). A general framework for adaptive processing of data structures. *IEEE Trans. Neural Netw*. 9, 768–786. doi: 10.1109/72.712151

French, R. (1999). Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci*. 3, 128–135. doi: 10.1016/S1364-6613(99)01294-2

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychol. Rev*. 87, 1–51. doi: 10.1037/0033-295X.87.1.1

Hayes, T. L., Cahill, N. D., and Kanan, C. (2018). "Memory efficient experience replay for streaming learning," in *IEEE International Conference on Robotics and Automation (ICRA)* (Montreal, QC: IEEE). doi: 10.1109/ICRA.2019.8793982

Hayes, T. L., Krishnan, G. P., Bazhenov, M., Siegelmann, H. T., Sejnowski, T. J., and Kanan, C. (2021). Replay in deep learning: current approaches and missing biological elements. *Neural Comput*. 33, 2908–2950. doi: 10.1162/neco_a_01433

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv [Preprint]*. arXiv:1503.02531.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., et al. (2020). Open graph benchmark: datasets for machine learning on graphs. *arXiv [Preprint]*. arXiv:2005.00687.

Hung, S. C. Y., Tu, C.-H., Wu, C.-E., Chen, C.-H., Chan, Y.-M., and Chen, C.-S. (2019). "Compacting, picking and growing for unforgetting continual learning," in *NeurIPS*, 13669–13679.

Iadarola, G. (2018). *Graph-based classification for detecting instances of bug patterns* (Master's thesis). University of Twente, Enschede, Netherlands.

Isele, D., and Cosgun, A. (2018). "Selective experience replay for lifelong learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 3302–3309.

Kipf, T. N., and Welling, M. (2017). "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. U.S.A*. 114, 3521–3526. doi: 10.1073/pnas.1611835114

Lesort, T., Stoian, A., and Filliat, D. (2020). Regularization shortcomings for continual learning. *arXiv [Preprint]*. arXiv:1912.03049.

Li, Z., and Hoiem, D. (2016). "Learning without forgetting," in *European Conference on Computer Vision* (Springer), 614–629. doi: 10.1007/978-3-319-46493-0_37

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

Maltoni, D., and Lomonaco, V. (2018). Continuous learning in single-incremental-task scenarios. *arXiv [Preprint]*. arXiv:1806.08568. doi: 10.1016/j.neunet.2019.03.010

Marcheggiani, D., Bastings, J., and Titov, I. (2018). "Exploiting semantics in neural machine translation with graph convolutional networks," in *Proceedings of*

the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (New Orleans), 486–492. doi: 10.18653/v1/N18-2078

Marsland, S., Shapiro, J., and Nehmzow, U. (2002). A self-organising network that grows when required. *Neural Netw*. 15, 1041–1058. doi: 10.1016/S0893-6080(02)00078-3

Micheli, A. (2009). Neural network for graphs: a contextual constructive approach. *IEEE Trans. Neural Netw*. 20, 498–511. doi: 10.1109/TNN.2008.2010350

Micheli, A., Sperduti, A., and Starita, A. (2007). An introduction to recursive neural networks and kernel methods for cheminformatics. *Curr. Pharmaceut. Design* 13, 1469–1496. doi: 10.2174/138161207780765981

Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2018). SocialLink: exploiting graph embeddings to link DBpedia entities to Twitter profiles. *Prog. Artif. Intell*. 7, 251–272. doi: 10.1007/s13748-018-0160-x

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: a review. *Neural Netw*. 113, 54–71. doi: 10.1016/j.neunet.2019.01.012

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T. P., and Wayne, G. (2019). "Experience replay for continual learning," in *NeurIPS*, 350–360.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., et al. (2016). Progressive neural networks. *arXiv [Preprint]*. arXiv:1606.04671.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Trans. Neural Netw*. 20, 61–80. doi: 10.1109/TNN.2008.2005605

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., et al. (2018). "Progress & compress: a scalable framework for continual learning," in *International Conference on Machine Learning*, 4528–4537.

Shchur, O., Mumme, M., Bojchevski, A., and Gannemann, S. (2018). "Pitfalls of graph neural network evaluation," in *Workshop on Relational Representation Learning, Neural Information Processing Systems (NeurIPS)*.

Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems 30*, eds I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc.), 2990–2999.

Sodhani, S., Chandar, S., and Bengio, Y. (2020). Toward training recurrent neural networks for lifelong learning. *Neural Comput*. 32, 1–35. doi: 10.1162/neco_a_01246

Sperduti, A., and Starita, A. (1997). Supervised neural networks for the classification of structures. *IEEE Trans. Neural Netw*. 8, 714–735. doi: 10.1109/72.572108

Srivastava, S., Berman, M., Blaschko, M. B., and Tuia, D. (2019). "Adaptive compression-based lifelong learning," in *BMVC*.

van de Ven, G. M., Siegelmann, H. T., and Tolias, A. S. (2020). Brain-inspired replay for continual learning with artificial neural networks. *Nat. Commun*. 11, 4069. doi: 10.1038/s41467-020-17866-2

van de Ven, G. M., and Tolias, A. S. (2018). "Three scenarios for continual learning," in *Continual Learning Workshop NeurIPS*.

Wang, C., Qiu, Y., and Scherer, S. (2020). Lifelong graph learning. *arXiv [Preprint]*. arXiv:2009.00647.

Wang, Z., Subakan, C., Tzinis, E., Smaragdis, P., and Charlin, L. (2019). Continual learning of new sound classes using generative replay. *arXiv [Preprint]*. arXiv:1906.00654. doi: 10.1109/WASPAA.2019.8937236

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst*. 32, 4–24. doi: 10.1109/TNNLS.2020.2978386

Zhou, F., and Cao, C. (2021). Overcoming catastrophic forgetting in graph neural networks with experience replay. *arXiv [Preprint]*. arXiv:2003.09908.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.