Exploring the Challenges towards Lifelong Fact Learning

Mohamed Elhoseiny [1], Francesca Babiloni[2], Rahaf Aljundi[2], Marcus Rohrbach[1], Manohar Paluri , Tinne Tuytelaars [2]

[1]Facebook Research, elhoseiny@fb.com,mrf@fb.com,mano@fb.com

[2]KU Leuven, francesca.babiloni,rahaf.aljundi,Tinne.Tuytelaars}@esat.kuleuven.be

Abstract

So far life-long learning (LLL) has been studied in relatively small-scale and relatively artificial setups. Here, we introduce a new large-scale alternative. What makes the proposed setup more natural and closer to human-like visual systems is threefold: First, we focus on concepts (or *facts*, as we call them) of varying complexity, ranging from single objects to more complex structures such as objects performing actions, and objects interacting with other objects. Second, as in real-world settings, our setup has a long-tail distribution, an aspect which has mostly been ignored in the LLL context. Third, facts across tasks may share structure (e.g., <person, riding,wave> and <dog, riding, wave>). Facts can also be semantically related (e.g., "liger" relates to seen categories like "tiger" and "lion"). Given the large number of possible facts, a LLL setup seems a natural choice. To avoid model size growing over time and to optimally exploit the semantic relations and structure, we combine it with a visual semantic embedding instead of discrete class labels. We adapt existing datasets with the properties mentioned above into new benchmarks, by dividing them semantically or randomly into disjoint tasks. This leads to two large-scale benchmarks with 906,232 images and 165,150 unique facts, on which we evaluate and analyze state-of-the-art LLL methods.

1 Introduction

To get closer to human visual learning and to practical application scenarios, where data often cannot be stored due to physical restrictions (e.g. robotics) or policy (e.g. privacy), the scenario of lifelong learning (LLL) has been proposed. The assumption of LLL is that only a subset of the concepts and corresponding training instances are available at each point in time during training. Each of these subsets is referred to as a "task", originating from robotics applications [20]. This leads to a chain of learning tasks trained on a time-line. While training of the first task is typically unchanged, the challenge is how to train the remaining tasks without reducing performance on the earlier tasks. Indeed, when doing so naively, e.g. by fine-tuning previous models, this results in what is known as *catastrophic forgetting*, i.e., the accuracy on the earlier tasks drops significantly. Avoiding such catastrophic forgetting is the main challenge addressed in the lifelong learning literature.

Lifelong Fact Learning (LLFL). Existing works on LLL have focused almost exclusively on image classification tasks (*e.g.* [8, 2, 12, 17, 21, 22, 7]), in a relatively small-scale and somewhat artificial setup. A sequence of tasks is defined, either by combining multiple datasets (e.g., learning to recognize MITscenes, then CUB-birds, then Flowers), by dividing a dataset (usually CIFAR100 or MNIST) into sets of disjoint concepts, or by permuting the input (permuted MNIST). Instead, in our work we propose a setup with the following more realistic and desirable learning characteristics:

1. *Long-tail:* Training data can be highly unbalanced with the majority of concepts occurring only rarely, which is in contrast to many existing benchmarks (e.g., [6, 10, 19]).

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

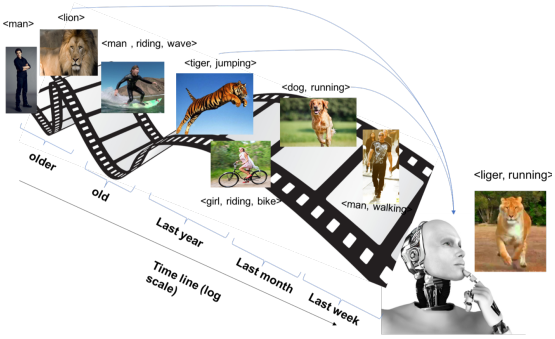

Figure 1: Lifelong Fact Learning

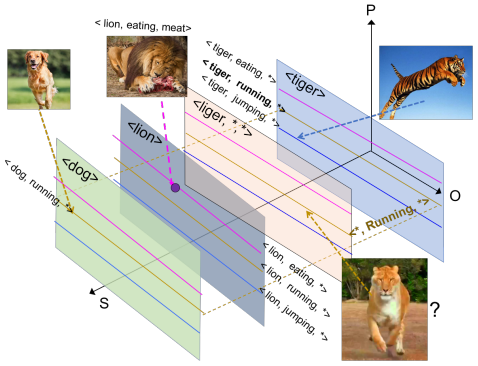

Figure 2: Structured Fact Representation

Dataset	Structured/Diverse	Long-Tail	Classes	Examples	Task Count	Split Type
MNIST	✗	✗	10	60000	2 to 5	S
CIFAR (used in [17, 22, 14])	✗	✗	100	60000	2 and 5	S
ImageNet and CUB datasets (used in [11])	✗	✗	1200	1211000	2	S
Scenes, CUB, VOC, and Flowers (used in [12, 2, 21])	✗	✗	122-526	5908-1211000	2	S
8 Dataset Sequence [1]	✗	✗	889	714387	8	S
CORe50 [13] / iCUBWorld-Transf([16]	✗	✗	10 (50)/15(150)	550/900 sessions	10	S
Our Mid-Scale LLFL Benchmark	✓	✗	186	28624	4	S & R
Our Large Scale LLFL Benchmark	✓	✓	165150	906232	8	S & R

Table 1: Comparison of some existing Task Sequences. Split Type is either S (Semantic), R (Random), or S&R (Both Semantic and Random splits are provided)

2. *Concepts of varying complexity:* We want to learn diverse concepts, including not only objects but also actions, interactions, attributes, as well as combinations thereof.

3. *Semantic and structure aware:* We want to connect semantically related visual facts. For example, if we have learned "lion" and "tiger" earlier, that can help us later in time to learn a "liger" (a rare hybrid cross between a male lion and a female tiger), even with just a few examples. Relating this to point (2) above, this further allows *compositional lifelong learning* to help recognize new facts (*e.g.* dog, riding, wave) based on facts seen earlier(*e.g.* person, riding, wave and girl, walking, dog).

A Note on Evaluation Measures. Although the performance of each task in isolation is an important characteristic which is adopted in the literature, it might be deceiving since a learnt representation could be good to classify an image in a restricted concept space covered by a single task. However, the same representation may not be able to classify the same image when considering all concepts across tasks. It is therefore equally important to measure the ability to distinguish the learnt concepts across all the concepts over all tasks. This is important since the objective of LLL is to model the understanding of an ever growing set of concepts over time. *To better understand how LLL performs in real world conditions, we advocate evaluating the existing methods across different tasks*. We named that evaluation *Generalized lifelong learning* (G-LLL), in line with the idea of Generalized zero-shot learning proposed in [3].

Advantages of a Visual-Semantic Embedding. As illustrated in Fig. 1, we expect to better understand liger, running by leveraging previously learnt facts such as lion, tiger, jumping and dog, running. We also expoit these properties in our work by leveraging semantic external knowledge using word embeddings – in particular word2vec [15]. To our knowledge, such semantic awareness has not been studied in a LLL context. To achieve this, we use a visual-semantic embedding model where semantic labels and images are embedded in a joint space. For the semantic representation, we leverage semantic external knowledge using word embeddings – in particular word2vec [15].

Contributions. First, we set up a midscale and a large scale benchmarks for lifelong fact learning, with two splits each, a random and a semantic split. The proposed approach for creating a semantically divided benchmark is general and could be applied similarly to other datasets or as more data becomes available. Second, we discuss the limitations of the current generation of LLL methods in this context, which forms a basis for advancing the field in future research. Third, we propose to focus on a more generalized evaluation (G-LLL) where test-data cover the entire label space across tasks. Our contributions further include an empirical study of existing LLL approaches in both the standard and the generalized setup. Finally,

2 Lifelong Learning Approaches

Problem Definition. Given a training set $\mathcal{D} = \{(x_i, y_i), i = 1 \cdots M)\}$, we learn from different tasks $T_1, T_2, \ldots, T_n$ over time where $T_i \subset \mathcal{D}$. y_i in our benchmarks are structured labels. For most

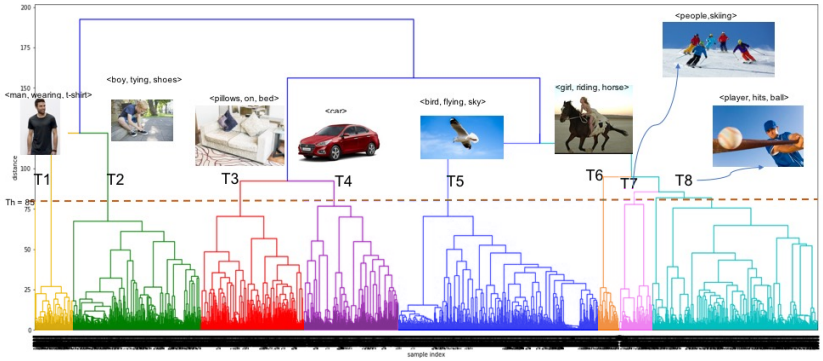

Figure 3: Lifelong Learning Semantically Divided Benchmark: 8 Tasks generated by agglomerative clustering in the semantic space of facts. The method is general and can be re-applied as more images and labels become available.

model-based approaches, we can formalize the LLL loss as follows. The loss of training the new task at time t is $L_t(\theta)$, where θ are the parameters of the network such that θ_i is the i^{th} parameter of an arbitrary neural network (a deep neural network with both convolutional and fully connected layers, in our case). $L(\theta)$ is defined as $L(\theta) = L_t(\theta|T_t) + \frac{\lambda}{2}\sum_i \omega_i^{t-1}(\theta_i - \theta_i^{t-1})^2$, where λ is a hyperparameter for the regularizer, θ_i^{t-1} the "old" network parameters at time $t-1$, and ω_i^{t-1} a weight indicating the importance of parameter θ_i for all tasks up to $t-1$. Hence, we tie strong springs on the important parameters at the previous time step (i.e., high ω_i^{t-1}) and weak springs on the non-important parameters (i.e., low ω_i^{t-1}). This way, we allow changing the latter more freely. Under this importance weight based framework, Finetuning, Intelligent Synapses [22] and Memory Aware Synapses [1] are special cases.

(1) Finetuning (FT): It does not involve any importance parameters at task $t-1$, so $\omega_i^t = 0, \forall i$.

(2) Synaptic Intelligence [22] (Int.Synapses) estimates the importance weights based on the contribution of each parameter to the change in the loss.

(3) Memory Aware Synapses [1] (MAS) defines importance of parameters in an online way based on their contribution to the change in the function output.

(5) ExpertGate [2]. ExpertGate is a data-based approach that learns an expert model for every task $E_1, E_2, \cdots E_n$, where every expert is adapted from the most related task. An auto-encoder model is trained for every task $AE_1, AE_2, \cdots AE_n$. These auto-encoders help determine the most related expert at test time given an example input x. The expert is then to make the prediction on x. Note the memory storage requirements of ExpertGate is n times bigger than earlier model-based approaches which might limit its practicality.

(4) Incremental Moment Matching [11] (IMM). Given n sequential tasks, the idea is to find the optimal parameter $\mu_{1:n}^*$ and $\Sigma_{1:n}^*$ of the Gaussian approximation function $q_{1:n}$ from the posterior parameter for each n^{th} task, (μ_n, Σ_n). At the end of the learned sequence, the obtained models are merged through a first or second moment matching.

(6) Joint Training (Joint): The data is not divided into tasks and the model is trained on the entire training data, i.e. it violates the LLL assumption. This can be seen as an upper bound for all LLL methods that we evaluate.

Adapting LLL methods to fact learning We use the joint-embedding architecture proposed in [4] as our backbone architecture to compare the evaluated methods. We chose this architecture due its superior performance compared to other joint-embedding models like [5, 9, 18] and its competitive performance to multi-class cross-entropy.

3 Experiments

Evaluation Metric (Standard vs Generalized). A central concept of LLL is that at a given time i we can only observe a subset T_i of the labeled training data $T_i \subset \mathcal{D} = \{(\mathbf{x}_k, y_k)\}_{k=1}^N$. Over time, we learn from different tasks $T_1, T_2, \ldots, T_N$. The categories in the different tasks are not intersecting, i.e., Y_i denotes all category labels in task T_i which is the union of all the labels in

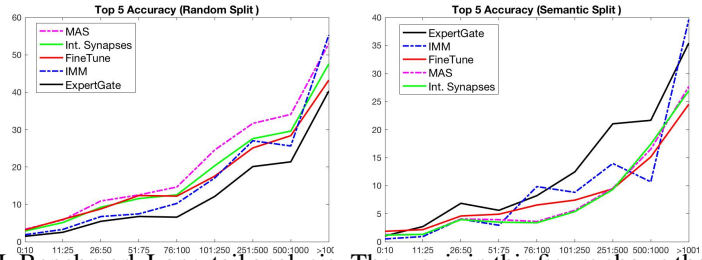

Figure 4: LLFL Benchmark Long-tail analysis. The x-axis in this figure shows the range of examples seen during training. On the left and right: the y-axis shows the generalized Top 5 Accuracy for the random and the semantic splits.

T_i, and $Y_i \cap Y_j = \emptyset, \quad \forall i \neq j$. Let $\mathcal{Y}$ denote the entire label space covered by all tasks, i.e., $\mathcal{Y} = \cup Y_i \ \forall i$.

Many existing works assume that one does not have to disambiguate between different tasks, i.e. for a predictive function $f_y : X \mapsto R$, we compute $A_{T_i \to Y_i}$ as the accuracy of classifying test data from T_i (the i^{th} task) into Y_i (the label space covered by only Y_i). The accuracy is computed per task.

$$\textbf{Standard LLL (S-LLL) Accuracy: } A_{T_i \to Y_i} = 1/M \sum_{n}^{M} 1[\hat{y_n} = \arg\max_{y \in Y_i} \quad f_y(\mathbf{x_n})] \tag{1}$$

where $\hat{y_n}$ is the ground truth label for instance $\mathbf{x_n}$. This metric assumes that at test time one knows the task of the input image. This is how most existing works are evaluated. However, this ignores the fact that determining the right task can be hard, especially when tasks are related. Therefore, we also evaluate across all tasks, which we refer to as *Generalized LLL*.

$$\textbf{Generalized LLL (G-LLL) Accuracy: } A_{T_i \to \mathcal{Y}} = 1/M \sum_{n}^{M} 1[\hat{y_n} = \arg\max_{y \in \mathcal{Y}} \quad f_y(\mathbf{x_n})] \tag{2}$$

In the generalized LLL metric, the search space at evaluation time covers the entire label space across tasks (i.e., $\mathcal{Y}$). Hence, we compute $A_{T_i \to \mathcal{Y}}$ as the accuracy of classifying test data from T_i (the i^{th} task) into $\mathcal{Y}$ (the entire label space) which is more realistic in most cases.

For each metric, we summarize results by averaging over tasks ("mean") and over examples ("mean over examples"), creating slightly different results when tasks are not balanced.

Results. For the two large-scale benchmarks, the results are reported in Table 2 and 3 for the generalized metric. Looking at these results, we make the following observations:

Random	T1	T2	T3	T4	T5	T6	T7	T8	mean	mean over examples
ExpertGate	12.99	20.77	25.19	17.72	35.17	9.62	11.64	21.75	19.36	15.34
Finetune	12.18	21.38	19.98	15.68	19.85	16.11	17.48	59.29	22.74	18.93
IMM	21.21	29.02	30.5	25.38	34.01	23.42	18.07	24.26	25.73	20.91
Int.Synapses	13.79	24.99	23.58	19.01	26.4	21.56	20.95	47.69	24.75	19.92
MAS	16.13	29.52	28.28	23.1	30.28	24.5	24.34	47.21	**27.92**	**22.48**

Table 2: Large Scale Random Split (Generalized Performance) Top 5 Accuracy

(1) The generalized LLL accuracy is always significantly lower than the standard LLL accuracy.

(2) The LLL performance of the random split is much better compared to the semantic split since the tasks has a higher similarity in the random splits and hence the representations can better adapt to future tasks with less forgetting. .

(3) ExpertGate is the best performing model on the semantic split. However, it is among the worst performing models on the random split. This is due to the setup of the semantic split, where sharing across tasks is minimized. This makes each task model behave like an expert of a restricted concept space, which follows the underlying assumption of how ExpertGate works.

(4) For the *large-scale data*, we observe that MAS is performing better than IMM on both the random and the semantic split, but especially on the random split; see Table 2. This may be because MAS has a better capability to learn low-shot classes as we discuss later in our *Few-shot Analysis*; see tables 4 and 5. This is due to the high similarity between the tasks as we go to that much larger scale.

Long-tail Analysis. We show in Fig 4 on left and middle the head-to-tail performance on the random split and the semantic split respectively. Specifically, the figure shows the Top5 generalized

Semantic	T1	T2	T3	T4	T5	T6	T7	T8	mean	mean over examples
ExpertGate	5.18	7.62	35.33	20.35	8.99	16.59	6.21	7.19	**13.43**	**14.91**
Finetune	1.58	8.56	0.07	2.06	5.88	2.86	4.77	37.9	7.96	9.75
IMM	8.34	5.06	0.18	13.27	0.52	21.48	11.21	3.26	7.91	4.15
Int.Synapses	1.71	10.82	0.22	2.84	5.87	4.77	6.36	28.26	7.61	8.70
MAS	1.79	11.35	0.64	4.25	4.76	5.36	6.2	27.35	7.71	8.54
Joint	10.20	4.86	37.71	33.52	25.09	3.17	4.83	8.43	15.98	20.68

Table 3: Large Scale Semantic Split (Generalized Performance) Top 5 Accuracy

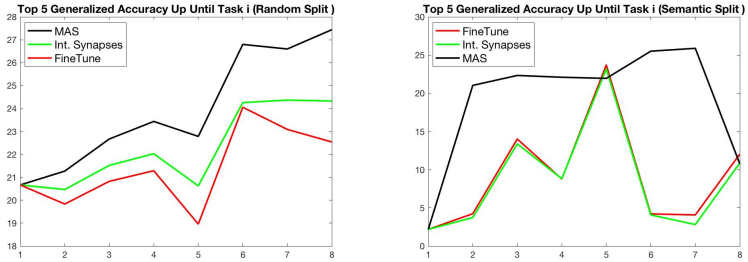

Figure 5: Gained Visual Knowledge: The x-axis shows the task number i. On the left, the y-axis shows the Random Split Top5 generalized accuracy over the entire test set up until training task i. On the right, the y-axis shows the Semantic Split Top5 generalized accuracy over the entire test set up until training task i.

	T1	T2	T3	T4	T5	T6	T7	T8	average
ExpertGate	0.33	0.88	0.57	3.27	0.69	2.04	0	0.62	1.05
Finetune	0	0.65	0	0	0.3	2.04	1.76	10.14	1.86
IMM	1.3	0.03	0	0	0.04	2.65	0	0	0.5
Int.Synapses	0.11	0.37	0	0	0.28	2.65	1.14	4.91	1.18
MAS	0.11	0.31	0	0.05	0.22	3.67	1.24	4.58	1.27
Joint	5.55	2.65	4.79	6.27	2.85	10.9	8.88	3.7	5.7

Table 4: Few-shot ($\leq$ 10) generalized Top 5 accuracy, Large-scale, Semantic Split

	T1	T2	T3	T4	T5	T6	T7	T8	average
ExpertGate	1.54	1.41	2.51	1.69	0.9	1.58	1.13	0.91	1.46
Finetune	1.48	1.62	1.4	1.73	1.47	1.77	2.66	13.17	3.16
IMM	1.79	1.58	2.1	1.9	2.88	2.07	1.76	0.95	1.88
Int.Synapses	1.4	1.55	2.25	2.95	2.94	3.09	3	5.8	2.87
MAS	1.79	2.29	3.25	3.8	4.6	3.09	3.4	4.44	3.33
Joint	3.75	4.65	6.99	4.61	3.81	8.9	10.38	2.5	5.7

Table 5: Few-shot ($\leq$ 10) generalized Top 5 accuracy, Large-scale, Random Split

accuracy over different ranges of seen examples per class (i.e., the x-axis in the figure). On the right, the figure shows the relative improvement of the model trained on the random split over the semantic split. Using the standard metrics, the head classes perform better using models trained on the semantic split compared to the random split. However as shown on Fig 4 (right), random split benefits everywhere with no clear relation to the class frequency (x-axis).

Gained Knowledge Over Time. Figure 5 shows the gained knowledge over time measured by the generalized Top5 Accuracy of the entire test set of all tasks after training each task. Figure 5 (left) shows that the LLL methods tend to gain more knowledge over time when the random split is used. This is due to the high similarity between tasks which makes the forgetting over time less catastrophic. Figure 5 (right) shows that the models have difficulty gaining knowledge over time when the semantic split is used. This is due to the low similarity between tasks which makes the forgetting over time more catastrophic. Note that the y-axis in Figure 5 left and right parts are comparable since it measures the performance of the entire test set which is the same on both the semantic and the random splits.

Few-shot Analysis. Tables 4 and 5 show few-shot results on the semantic and the random split, respectively. As already observed earlier, the performance on the random splits is better compared to the semantic splits. We can observe here that finetuning is the best performing approach on average for few-shot performance on both splits. Looking closely at the results, it is not hard to see that the main gain of finetuning is due to its high accuracy on the last task. This shows that existing LLL methods do not learn the tail and there is need to devise new methods that have a capability to learn the tail distribution in a LLL setting.

4 Conclusions

We proposed two benchmarks to evaluate fact learning in a lifelong learning setup. A methodology was designed to split up an existing fact learning dataset into multiple tasks, taking the specific constraints into account and aiming for a setup that mimics real world application scenarios. With these benchmarks, we hope to foster research towards more large scale, human-like artificial visual learning systems and studying challenges like long-tail distribution.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. *arXiv preprint arXiv:1711.09601*, 2017.

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, 2017.

[3] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European Conference on Computer Vision*, pp. 52–68. Springer, 2016.

[4] Mohamed Elhoseiny, Scott Cohen, Walter Chang, Brian L Price, and Ahmed M Elgammal. Sherlock: Scalable fact learning in images. In *AAAI*, pp. 4016–4024, 2017.

[5] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision*, 106(2):210–233, 2014.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[7] Christoph Käding, Erik Rodner, Alexander Freytag, and Joachim Denzler. Fine-tuning deep neural networks in continuous learning scenarios. In *Asian Conference on Computer Vision*, pp. 588–605. Springer, 2016.

[8] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, pp. 201611835, 2017.

[9] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[11] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pp. 4652–4662, 2017.

[12] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pp. 614–629. Springer, 2016.

[13] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, 2017.

[14] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[16] G. Pasquale, C. Ciliberto, L. Rosasco, and L. Natale. Object identification from few examples by improving the invariance of a deep convolutional neural network. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4904–4911, Oct 2016. doi: 10.1109/IROS.2016.7759720. URL `http://ieeexplore.ieee.org/document/7759720/`.

[17] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, and Christoph H Lampert. icarl: Incremental classifier and representation learning. *arXiv preprint arXiv:1611.07725*, 2016.

[18] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pp. 2152–2161, 2015.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] Sebastian Thrun and Joseph OSullivan. Clustering learning tasks and the selective cross-task transfer of knowledge. In *Learning to learn*, pp. 235–257. Springer, 1998.

[21] Amal Rannen Triki, Rahaf Aljundi, Mathew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. *arXiv preprint arXiv:1704.01920*, 2017.

[22] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3987–3995. PMLR, 06–11 Aug 2017.