Supplementary Material: One VLM to Keep it Learning: Generation and Balancing for Data-free Continual Visual Question Answering

Deepayan Das[1] Davide Talon[2] Massimiliano Mancini[1]
Yiming Wang[2] Elisa Ricci[1, 2]
[1]University of Trento [2]Fondazione Bruno Kessler
{deepayan.das, massimiliano.mancini, e.ricci}@unitn.it
{dtalon, ywang}@fbk.eu

A. Supplementary Overview

In this supplementary material we present further results and ablation analysis on the presented method GaB, implementation details and qualitative visualization of generated question-answer pairs. Specifically, while in Section B, we report extensive results on the method, Section C ablates the proposed pseudo-rehearsal strategy. We continue with Section D and Section E describing in further detail the datasets and the implementation details of the approach, respectively. Finally, Section F concludes by providing qualitatives of generated question answer pairs.

B. Extended Results

We here report extended empirical results for the proposed method GaB, including the intermediate evaluation of the sequentially training model and the robustness of the approach to different task orders.

B.1. Per-Task Performance Analysis

We evaluate the performance of the learning VQA model on the different tasks as training progresses.

Figure 1 showcases the performance variations across various tasks within the VQACL-VQAv2 benchmark in terms of AP, highlighting how each continual learning method adapts over time. As can be noted, despite building on pseudo-rehearsal samples only, throughout the entire sequential adaptation GaB achieves accuracy on par with the rehearsal strategy that leverage past real data. A similar behaviour is observed in Figure 2 detailing the sequential task performance for the CLOVE-function benchmark during sequential training.

B.2. Continual Learning Across Task Orders on CLOVE

Continual learning performance can significantly vary depending on the order in which tasks are presented. To explore this variability, we evaluate GaB across three different task orders of the CLOVE-function benchmark. Let us denote each task with its initial letter, *i.e.*, Objects (o), Attributes (a), Relations (r), Logical (l) and Knowledge (k), we consider task orders 'oarlk', 'rolak', and 'lkora'.

Table 1. The continual learning performance in terms of AP and AF on the CLOVE-function dataset considering different task orders.

	oarlk		lkora		rolak	
Method	AP (↑)	AF (↓)	AP (↑)	AF (↓)	AP (↑)	AF (↓)
Multi-task	32.26					
Rehearsal	41.82	3.14	29.75	13.61	28.23	8.57
Seq-FT	22.70	22.19	13.35	24.24	12.35	23.61
GaB w/o balancing	37.01	3.61	25.48	16.9	**29.18**	**7.97**
GaB-clustering (Ours)	**40.70**	**1.40**	**31.59**	**10.64**	26.17	9.67

The results are summarized in Table 1 in terms of Average Performance (AP) and Average Forgetting (AF). Despite the differences in AP and AF across the task sequences, similar behaviours are observed: our approach GaB-clustering performs competitively with the rehearsal strategy and it provides a large margin improvement to the Seq-FT baseline. Notably, the 'oarlk' sequence consistently shows better performance metrics compared to 'lkora' and 'rolak', suggesting that the order in which tasks are encountered can influence the efficacy of the learning process. In the 'oarlk' sequence, our GaB-clustering method demonstrated the best resilience against forgetting with an AF of 1.40, and a high AP of 40.70, underscoring its robustness in handling the challenges presented by this particular sequence. We see a similar trend in the sequence 'lkora' where our method has the best overall AP and AF. Conversely, in the 'rolak', GaB w/o balancing shows the most effective, indicating

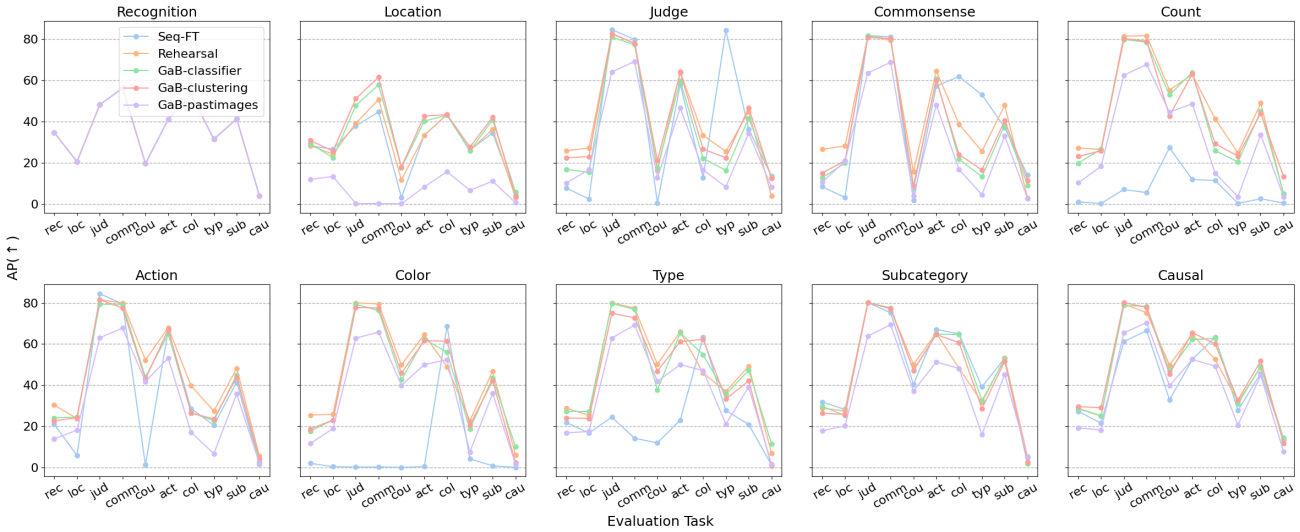

Figure 1. Per-task performance in terms of AP across different tasks in the VQACL-VQAv2 benchmark.

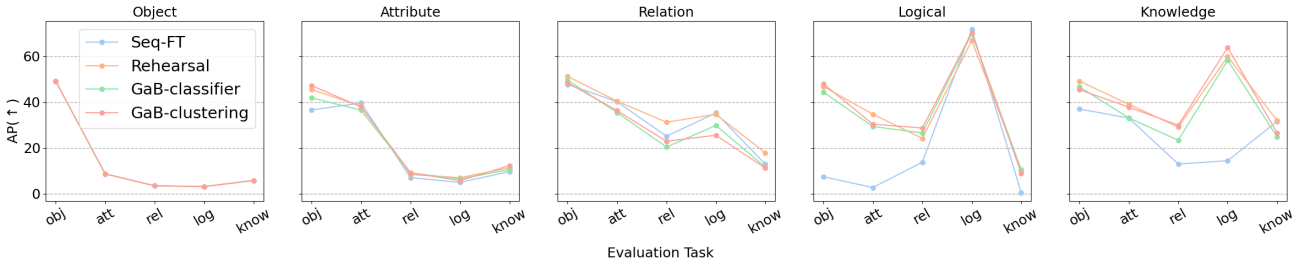

Figure 2. Per-task performance in terms of AP across different tasks in the CLOVE-function benchmark.

that pseudo-generation aids forgetting mitigation while the clustering-based balancing strategy might require more careful hyperparameter tuning, such as the number of clusters to use.

B.3. Balancing Questions

We visualize the question types distribution of rehearsal samples before (`Generated`) and after balancing (`Balanced`) with our pseudo-rehearsal balancing module, compared to the ground truth one (`Real`). These visualizations help illustrate the impact of our balancing technique on the diversity of question types generated during the pseudo-rehearsal data generation.

Figure 3 illustrates the distribution alignment for VQACL-VQAv2 demonstrating how GaB-clustering ensures no single question type dominates the training process. We show similar results on CLOVE-function in Figure 4.

C. Further ablations

We further ablate GaB on the number of balancing clusters and the use of the question-answer generation module for dynamic sampling of the rehearsal data.

C.1. Varying the number of balancing clusters

We explore the impact of varying the number of clusters in our balanced cluster strategy GaB-clustering. We ablate to determine the optimal number of clusters that yields the best performance. The results of this study are illustrated in Figure 5, which displays the average precision achieved across different cluster counts K. Our findings indicate that setting the number of clusters to 7 maximizes average perfomance.

C.2. Dynamic Pseudo-Rehearsal

Traditional rehearsal strategies for continual learning are constrained in the buffer dimension due to the limited availability of question-answers from previous tasks. Thanks to the dynamic generation of samples, pseudo-rehearsal strategies could possibly rely on larger replay buffers instead. We here explore a dynamic version of the presented approach (`GaB-dynamic`) where pseudo-samples are generated on-the-fly based on current task data. At each training step this

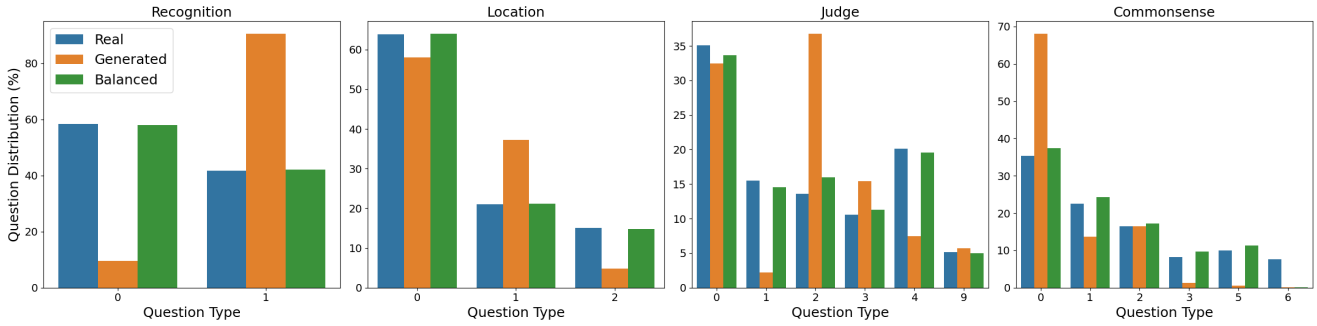

Figure 3. Question distribution before and after pseudo-rehearsal balancing for the VQACL-VQAv2 benchmark. The figure shows the distribution across different question categories for old tasks (as per plot title) generated from last task *count* visual images.

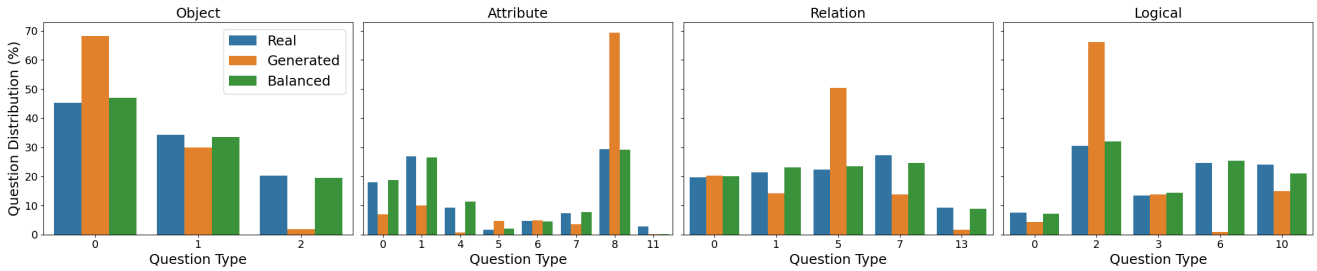

Figure 4. Question distribution before and after pseudo-rehearsal balancing for the CLOVE-function benchmark. The figure shows the distribution across different question categories for old tasks (as per plot title) generated from last task *knowledge* visual images.

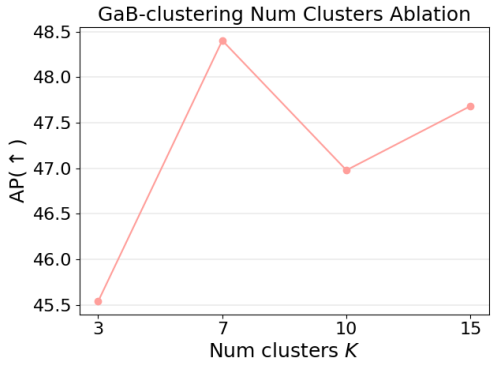

Figure 5. The continual learning performance of GaB-clustering in terms of AP when varying the number of balancing clusters K on the VQACL-VQAv2 dataset.

method involves dynamically generating a question related to previous tasks for each current batch visual image. As the alignment of the question distributions is non-trivial in the batch low-samples setting, no balancement to the generated questions-answer pairs is applied.

Table 2 compares the approaches in terms of AP and AF for the CLOVE-function setting. Despite the larger number of replay samples available to GaB-dynamic, the results indicate that GaB-clustering achieves the highest performance

Table 2. The continual learning performance in terms of AP and AF on the CLOVE-function dataset.

Method	**AP** (↑)	**AF** (↓)
GaB-dynamic	36.17	5.26
GaB-classifier (Ours)	37.97	5.25
GaB-clustering (Ours)	**40.70**	**2.26**

with a +4.53% and -3% in the AP and AF metrics, respectively. Intuitively, we observe limited variability of QA samples that could be generated within each batch with a consequent drop on the final performance.

D. Dataset Details

We provide comprehensive details on both datasets utilized in our study. The VQACL-VQAv2 benchmark is comprised of 10 distinct tasks, each with its own set of questions tailored to specific aspects of visual and textual understanding. The different tasks considered are in order: `Recognition`, `Location`, `Judge`, `Commonsense`, `Count`, `Action`, `Color`, `Type`, `Subcategory` and `Causal`. We refer the reader to the original paper for extensive details on the benchmark. On the other hand, the CLOVE benchmark includes 6 tasks. For fair comparison,

we restrict our evaluation to only 5 tasks due to the specialized architecture and auxiliary features needed for answering scene-text questions asking to OCR present text.

Each task within these datasets is associated with questions that are categorized into various types depending on their content and focus. These question types are labeled systematically to facilitate targeted training and analysis. While in VQACL-VQAv2 meta-information on question types models the initial words used in the question construction, e.g. "what type", "is the", "where is", "how many", differently, CLOVE-function auxiliary information models the property being queried and the expected answer, for instance "MaterialChoose", "activityWho" or "relVerify".

E. Implementation details

We implement our strategy in PyTorch [3] and employ the Hugging Face[1] implementation of the BLIP-2[1] architecture, specifically the `opt-2.7b` version. For textual generation, we follow standard practice and fix the *max_new_tokens* for both generated answers (2 tokens) and pseudo QA pairs (20 tokens). The repetition penalty is set to 1.2. In line with prior work, we prompt BLIP-2 for answer generation with the prompt `p="Question: <question> Answer:"`, while pseudo-QA has an empty prompt `p=""`, akin to the original BLIP-2 strategy.

Baselines. We implemented continual learning strategies following an open-source codebase for CL[2]. The strategies are applied to the same BLIP-2 architecture as GaB. For the regularization-based methods, Elastic Weight Consolidation (EWC) and Memory Aware Synapses (MAS), we set the regularization parameters to 1.0 and compute importance weights on the token classifier generating the output textual sequence. For the Learning to Prompt (L2P) approach the prompting strategy is applied to the visual encoder only. We adopt different configurations depending on the benchmark: for VQACL-VQAv2, we utilize a prompt pool size of 10 with a regularization parameter of 1.0, whereas for CLOVE-function, the prompt pool size is increased to 50 and the regularization parameter is adjusted to 0.5. These settings are carefully chosen to optimize performance across the diverse conditions presented by each benchmark.

F. Qualitatives Results

We report qualitative results on the generated question-answer pairs providing both positive and negative examples. In Figure 6-a and Figure 6-b GaB successfully generates both accurate and contextually appropriate questions and accompanying answers. For instance, questions such as "What material is the floor made of?" with the answer "tile" and "Is there a bear in the picture?" with the answer "yes" demonstrate the model's ability to understand and respond correctly based on the visual data. Differently, in Figure 6-c we highlight a scenario where the generated question is vague and lacks specificity. An example from this row includes a question like "What color is the jacket?" which, although correct (answered as "orange"), does not specify which jacket to look for as there are multiple people wearing jackets in the image. Finally, Figure 6-d presents instances where the model's generated answers are incorrect. For example, the question "How many planes are there?" receives the answer "3", whereas we can clearly see there are in fact only 2 airplanes, indicating the model's challenges in some contexts or its misinterpretation of the visual content.

[1] https://huggingface.co/

Table 3. Computational overhead in terms of training parameters and time. `Param count` indicates the fraction of trained parameters, `TFlops` indicates Tera Flops in forward+backward passes in 1 epoch, `Computational time` metrics report the processing time requirements (in seconds) per one epoch of a single task in CLOVE-function across different methods.

Method	Param count	TFlops	Computational time (s)		
			Training	Generation	Balancing
Rehearsal	0.05	2711.05	35,085	n.a.	n.a.
LAMOL*	0.1	5670.19	49,650	363	n.a.
GaB-clustering (Ours)	0.3	3568.38	42,885	1,452	273

G. Complexity overhead discussion

As a data-free rehearsal strategy GaB trades-off the prohibited access to past tasks data with increased computational complexity for sample generation. More precisely, GaB requires two passes of the employed VLM: a first forward-backward pass trains the qa projection layer $(f_{v\to qa})$ and answer projection layer $(f_{v\to a})$ while the second forward pass allows to generate rehearsal data pairs. Limited overhead is required for balancing the questions according to precomputed real-data clustering statistics, with no need for further modification. Only the task-specific projection heads and answering head are trained, while the rest of the model remains frozen, making the process computationally efficient. We evaluate the computational overhead for pseudo-rehearsal generation and balancing of the presented approach, results are presented in Tab. 3, where we can see that the approach falls behind the rehearsal baseline while being competitive with the data-free version of LAMOL*. Both GaB and LAMOL* suffer from the longer training time due to the requirements of learning how to generate question-answer pairs, however, GaB avoids replaying for the generation task, ending up with shorter training time at the cost of larger number of training parameters. No additional computation is required at inference time where task-specific heads can be discarded and the single shared answering head solves for the VQA task.

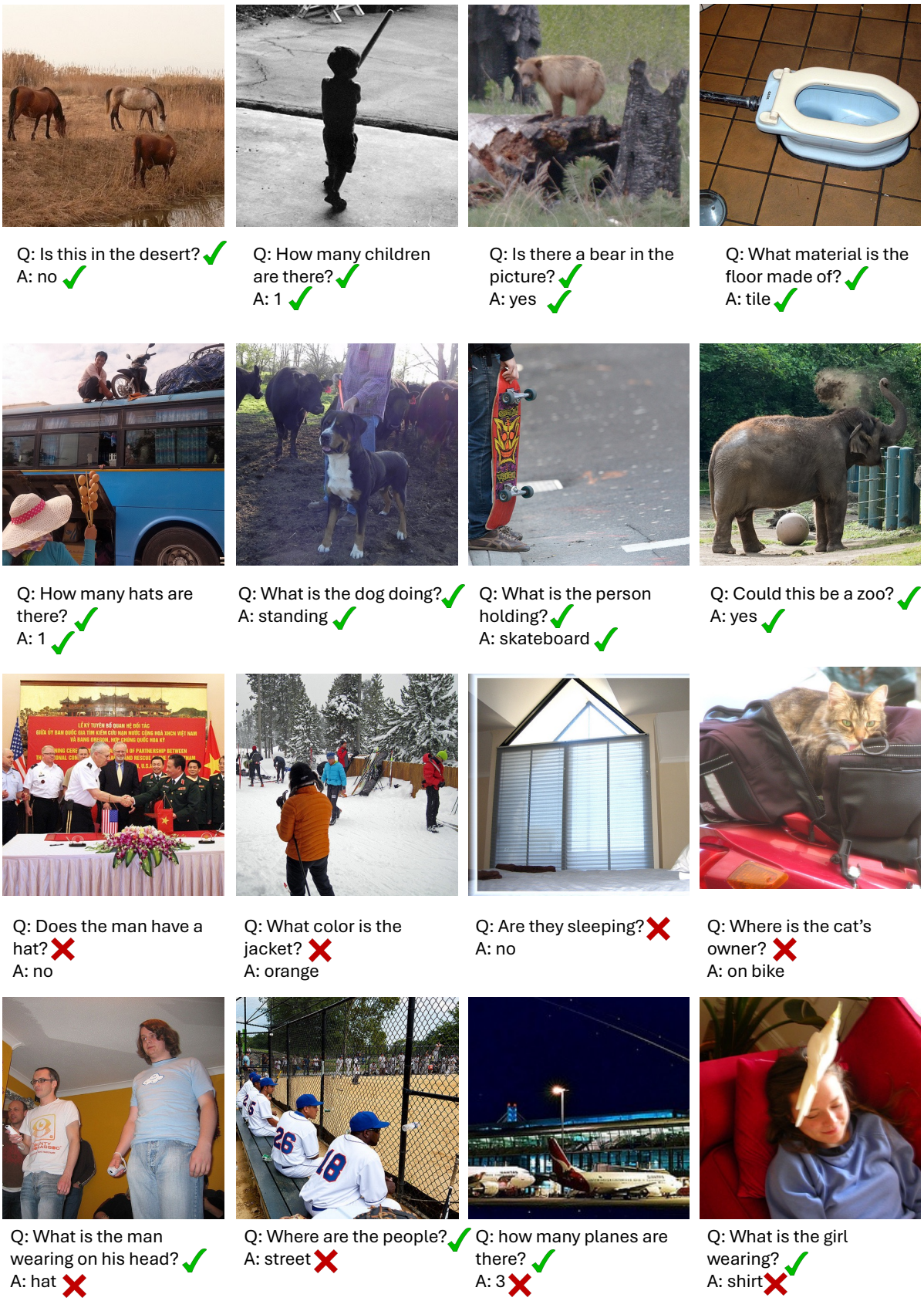

Figure 6. Qualitatative results of GaB generated QA pairs on VQACL-VQAv2 datasets. **(a)-(b)** Correctly generated questions answer pairs; **(c)** question-answer pairs with ill-posed questions; **(d)** question-answer pairs with wrong generated answers.

References

[1] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 2023. 4

[2] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido M Van de Ven, et al. Avalanche: an end-to-end library for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3600–3610, 2021. 4

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 4