Symbolic Replay: Scene Graph as Prompt for Continual Learning on VQA Task

Stan Weixian Lei[1]*, Difei Gao[1]*, Jay Zhangjie Wu[1], Yuxuan Wang[1], Wei Liu[2], Mengmi Zhang[3], Mike Zheng Shou[1]†

[1]Show Lab, National University of Singapore
[2]Tencent Data Platform
[3]CFAR and I2R, Agency for Science, Technology, and Research (A*STAR), Singapore
{weixian.lei,zhangjiewu,yuxuan.wang}@u.nus.edu, {daniel.difei.gao,mike.zheng.shou}@gmail.com,
wl2223@columbia.edu, Mengmi@i2r.a-star.edu.sg

Abstract

VQA is an ambitious task aiming to answer any image-related question. However, in reality, it is hard to build such a system once for all since the needs of users are continuously updated, and the system has to implement new functions. Thus, Continual Learning (CL) ability is a must in developing advanced VQA systems. Recently, a pioneer work split a VQA dataset into disjoint answer sets to study this topic. However, CL on VQA involves not only the expansion of label sets (new **A**nswer sets). It is crucial to study how to answer questions when deploying VQA systems to new environments (new **V**isual scenes) and how to answer questions requiring new functions (new **Q**uestion types). Thus, we propose CLOVE, a benchmark for **C**ontinual **L**earning **O**n **V**isual qu**E**stion answering, which contains scene- and function-incremental settings for the two aforementioned CL scenarios. In terms of methodology, the main difference between CL on VQA and classification is that the former additionally involves expanding and preventing forgetting of reasoning mechanisms, while the latter focusing on class representation. Thus, we propose a real-data-free replay-based method tailored for CL on VQA, named Scene Graph as Prompt for Symbolic Replay. Using a piece of scene graph as a prompt, it replays pseudo scene graphs to represent the past image, along with correlated QA pair. A unified VQA model is also proposed to utilize the current and replayed data to enhance its QA ability. Finally, experimental results reveal the challenges in CLOVE and demonstrate the effectiveness of our method.

Introduction

In recent years, we have witnessed tremendous successes in achieving state-of-the-art performance on VQA tasks by CV and NLP communities (Anderson et al. 2018; Lu et al. 2019; Su et al. 2019; Jiang et al. 2020; Gao et al. 2020; Chen et al. 2020). Despite the remarkable success, current VQA systems are usually trained on specific datasets and then fixed for use. However, in real applications, user's demands are always updated, as is shown in Fig. 1 (a). The VQA system is expected to continuously learn the knowledge, when it is deployed to new scenes or is required to add new functions.

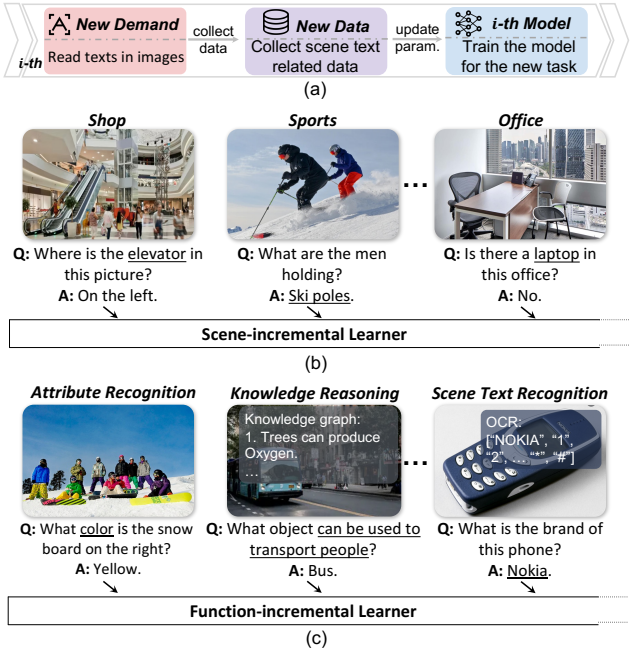

Figure 1: (a) An AI continuously receives new demands and is updated with collected data. (b) A scene-incremental learner adapts to new scenes for deployment. (c) A function-incremental learner acquires new functions over time.

In the CV community, various approaches have been studied actively to tackle continual learning for image classification (Rebuffi et al. 2017), where a model is trained sequentially on a set of images with disjoint labels. In this setting, a model continually enhances one ability (*i.e.*, recognition) over one modality (*i.e.*, vision) and mainly learns the representation for each class. In VQA, a model is required to adapt to new environments (*e.g.*, shop, office, etc.) or learn new abilities (*e.g.*, attribution recognition, knowledge reasoning, etc.) according to the changing demands, as shown in Fig 1 (b) and (c). As such, CL in VQA is different from the aforementioned image classification that a model continually learns new ablities over multi-modalities and focuses more on the reasoning. Thus, it is crucial to study Continual

*These authors contributed equally.
†Corresponding author.

Learning for Visual Question Answering (CLVQA).

Only a few pioneer works attempted to explore CLVQA. Greco et al. studied the continual learning on two subsets, wh- and yes/no questions, of synthesized CLEVR dataset (Johnson et al. 2017) split by themselves. However, using the above setting is still hard to evaluate the continual learning of new scenes or functions we are interested in. Specifically, for the image domain, images for both wh- and yes-no questions are about geometric spheres. For functions, both types of questions are about the object properties and multi-hop reasoning upon them. The only difference is the way to ask, *e.g.*, *what color is the cube?*, *is the cube in red?* actually evaluating the same function.

Thus, we reorganize existing VQA datasets (Hudson and Manning 2019; Gao et al. 2019; Singh et al. 2019) to construct CLOVE, a novel benchmark devised for **C**ontinual **L**earning **O**n **V**isual qu**E**stion answering with two continual learning settings, as shown in Fig. 1 (b) and (c). (1) *Scene-incremental setting* mimics the scenario where a VQA agent is adapted to new scenes for deployment. Our CLOVE-scene contains 6 scenes: *ShopAndDinning*, *Workplace*, *HomeOrHotel*, *Transportation*, *SportAndLeisure* and *Outdoors*. (2) *Function-incremental setting* tests a model's ability in acquiring new functional abilities over time. Our CLOVE-function contains 6 functions: *object recognition*, *attribute recognition*, *relation reasoning*, *logic reasoning*, *knowledge reasoning* and *scene text understanding*.

In terms of methodology, traditional continual learning methods on image classification (Rebuffi et al. 2017) are specially designed to prevent forgetting the representation of vision modality. They could be hard to perform well on CLVQA, which requires multi-modal reasoning. Specifically, regularization-based methods (Kirkpatrick et al. 2017; Aljundi et al. 2018) might fail in estimating the importance and balancing the learning of past and new tasks due to the complicated model design in VQA models. Some other works preserve historical knowledge through the replay. And since, in the CL setting, real data usually cannot be saved due to privacy concerns, many works retain knowledge by generating pseudo samples (Shin et al. 2017). However, replaying pseudo samples in CLVQA could be extremely challenging. The images could come with complicated visual scenes and fine-grained details, which could be hard to be precisely generated by the state-of-the-art image GAN model (Sauer, Schwarz, and Geiger 2022). Generated images in low quality also limits the quality of generated question-answer pairs. All these issues pose a dilemma for generating *image-question-answer* for pseudo-replay.

In this paper, we introduce Scene Graph as Prompt for symbolic replay (SGP), a real-data-free replay-based method for CLVQA. SGP overcomes the aforementioned limitations of replayed methods by leveraging the scene graph, a concise and structured representation of visual information, as an alternative to images for replay. Specifically, SGP consists of a symbolic replay model (SRM) and a unified VQA model (UniVQA). The SRM, which belongs to a language model (Radford et al. 2019), continuously captures the symbolic reasoning mechanism and learns the task-specific mapping between scene graph and QA pairs. During inference, SRM replays the *scene-graph-question-answer* triplet for knowledge revisiting, prompted by a randomly sampled scene graph relationship. We call this "symbolic replay". Besides, the UniVQA is designed to adapt a wide range of input modalities for different VQA tasks. Trained with the mix of current task samples and symbolic-replayed samples, UniVQA is capable of learning a new task while retaining the previously acquired knowledge. Moreover, since the past real data is not saved, our framework can be employed to various situations involving privacy concerns.

Extensive experiments with various types of existing CL methods and our model show the difficulties of our benchmark and demonstrate the effectiveness of our method.

Related Work

Visual Question Answering. VQA is a general task aiming to answer any vision-related question. It requires AI to achieve a vast set of functions to answer questions, ranging from fine-grained recognition, object detection, activity recognition to commonsense reasoning, etc. (Antol et al. 2015; Goyal et al. 2017) introduced the VQA benchmarks for understanding the common visual concepts in real world. (Johnson et al. 2017) built a synthetic dataset for testing visual reasoning abilities, *e.g.*, multi-hop and logic reasoning. (Hudson and Manning 2019) constructed a VQA dataset sourced from Visual Genome (Krishna et al. 2017) by leveraging the annotated scene graph, aiming to test the model's compositional reasoning capability in real images. (Gao et al. 2019; Marino et al. 2019) proposed benchmarks where a model should resort to external knowledge for reasoning. (Singh et al. 2019) built a dataset where a model should understand the text in images to answer questions.

The development of VQA shows that the function set of VQA always need to be continuously expanded with new demands. However, few benchmarks focus on continual learning on VQA. Thus, we propose a benchmark to mirror real-world scenarios where an AI is required to be deployed to new environments or learn new functions in a CL manner.

Continual Learning Benchmarks. In CV community, most of works study continual learning under three settings: (1) Class-incremental learning, where a classification model learns to classify increasing number of classes over time (Li and Hoiem 2017; Rebuffi et al. 2017). (2) Task-incremental learning, where task identity of newly included task (a set of classes) remains known during inference (Aljundi, Chakravarty, and Tuytelaars 2017; Serra et al. 2018). (3) Domain-incremental learning, where a model sequentially learns to solve tasks with shifts in input distributions (Rebuffi, Bilen, and Vedaldi 2017). In NLP, CL is conducted on tasks with different domains (Chen, Ma, and Liu 2015; Lee 2017) or on cross-task benchmarks (Biesialska, Biesialska, and Costa-jussà 2020; Hu et al. 2020a; Sun, Ho, and Lee 2020). All the aforementioned CL tasks focus on one single modality. For CV, it focuses only on one single ability, *e.g.*, image classification or detection. In contrast, CLVQA considers multiple modalities, vision and language, and involves mutiple abilities.

Similar to ours, (Greco et al. 2019) proposed a CL benchmark on VQA on the synthesized CLVER dataset. It selected Wh- and Yes/No-type questions and studied CL in the two yielded task orders. This setting is similar to class-incremental learning that the increments are on the disjoint answer set. Unlike CLOVE, neither shift in image domain nor expansion of acquired functions set is reflected. The challenges in CLVQA might be underestimated.

Continual Learning Methods. Continual learning studies the methods that can learn new knowledge without forgetting the past knowledge. Existing CL methods can be grouped into the following categories: (1) Replay-based method, which reminds models of knowledge from previous tasks through experience replay. iCaRL (Rebuffi et al. 2017) tackles CIL by selecting the nearest-mean-examples from previous tasks and combining rehearsal and distillation strategies. GEM (Lopez-Paz and Ranzato 2017) preserves a subset of real samples from previous tasks. Utilizing these real samples during optimization helps somewhat constrain parameter gradients. (Shin et al. 2017; Sun, Ho, and Lee 2020) synthesized pseudo samples with generative models to mitigate catastrophic forgetting. (2) Regularization-based method which tries to keep the weights that are important for the previous tasks. EWC (Kirkpatrick et al. 2017) uses the Fisher Information Matrix to estimate the importance while MAS (Aljundi et al. 2018) estimates by measuring how small changes in the parameters affect the output of the model. (3) Architecture-based method, where different tasks are associated with different modules of the whole model, reduces the interference between tasks (Mallya and Lazebnik 2018; Liu, Schiele, and Sun 2021).

Most of the above methods mitigate catastrophic forgetting from the perspective of representation. However, VQA tasks inherently involve two types of abilities: representation learning and reasoning. Thus, we propose symbolic replay, which helps continuously learn the representation and reasoning skills over a sequence of multi-modal VQA tasks.

CLOVE Benchmark

Here, we detail how we create CLOVE, a benchmark for **C**ontinual **L**earning **O**n **V**isual qu**E**stion answering. It contains scene-incremental setting and function-incremental setting, named as CLOVE-scene and CLOVE-function.

Task Formulation

In our VQA continual learning framework, we define the sequence of N VQA tasks to be solved as $\mathbf{T} = (T_1, T_2, \cdots, T_N)$, where task T_i is to optimize a model towards an objective on the task-specific dataset $D_i = \{\mathbf{d}_1^i, \cdots, \mathbf{d}_n^i, \cdots, \mathbf{d}_{|D_i|}^i\}$. Here, $\mathbf{d}_n^i$ represents an image-question-answer triplet $\{\mathbf{v}_n^i, \mathbf{q}_n^i, \mathbf{a}_n^i\}$. The image and question are inputs to a VQA model, and the ground-truth answer is the desired output.

CLOVE-scene Setting

An advanced AI agent is expected to be capable of answering questions from different scenes. It may need to adapt to a novel visual environment with new concepts, and meanwhile, remember the past knowledge.

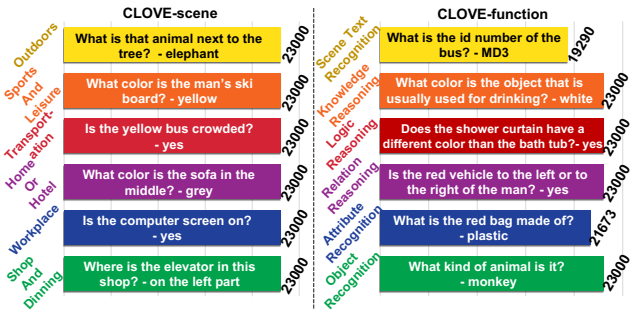

Figure 2: Sample numbers with one QA example for each task in CLOVE-scene and CLOVE-function.

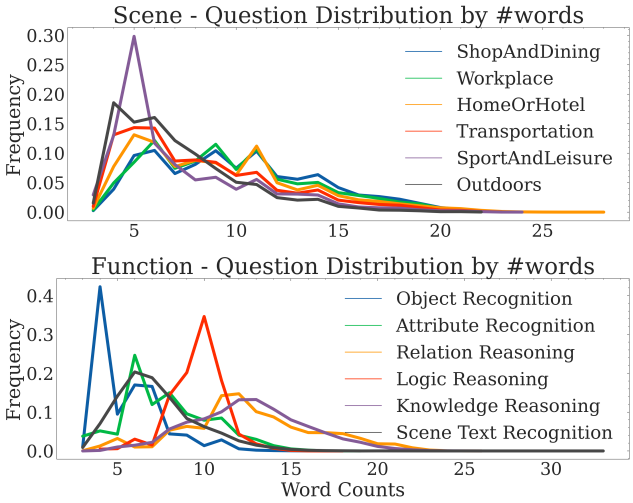

Figure 3: Distribution of question length for CLOVE-scene (up) and CLOVE-function (bottom).

Setting definition. We refer to the taxonomy in the SUN database (Xiao et al. 2010) and classify the sourced images in GQA. We obtain six classes from the second level of scene hierarchy defined in the SUN database: *ShopAndDinning*, *Workplace*, *HomeOrHotel*, *Transportation*, *SportAndLeisure* and *Outdoors*.

Image sourcing. To obtain images for each task, we resort to a *sota* scene classification model to obtain an initial partition. Then, we apply two post-processing strategies to improve the qualities of the selected images: (1) filter images with a low classification confidence score; (2) filter images with limit frequent objects in that scene, which is given in the SUN database. Finally, we randomly sample 100 images from each task and ask 3 human workers to evaluate the accuracy. Result shows that the yielded splitting achieves a mean accuracy of 91.0%.

Question-answer pairs sourcing. When deployed to a new scene, the model may face two type of questions: (1) Questions related to a unique object in that scene, requiring the model to predict the object's name as a scene-specific

answer. (2) Questions related to general concepts shared among different scenes, *e.g.*, color and material, requiring the model to predict common answers. Our scene incremental setting includes both two types of questions. Concretely, we maintain a set of common answers for all tasks and sets of unique answers for each task. Specifically, a common answer can appear in different tasks (*e.g.*,"red" can appear in different tasks) while a unique answer is only allowed to appear in its corresponding task (*e.g.*, "computer monitor" only appears in *Workplace*). For each task, we collect similar number of samples with common answers and with unique answers. In addition, we balance the number of samples among different tasks and follow GQA to smooth the answer distribution within each task to avoid dataset bias.

CLOVE-function Setting

A VQA system requires a vast set of functions to answer a question – to name a few, object detection (e.g. "How many birds are there?"); activity recognition (e.g., "Is the man running?") , knowledge base reasoning (e.g., "Is this a vegetarian pizza") and scene text recognition ("What is the name of this book?"). It is common that an AI agent is required to develop different functions when encountering different use cases over time. To mimic such a scenario, we create CLOVE-function, by sampling and reorganizing data from GQA (Hudson and Manning 2019), CRIC (Gao et al. 2019) and TextVQA (Singh et al. 2019).

Setting definition. Based on the functions defined and introduced in GQA, CRIC and TextVQA, we collect six tasks for function incremental setting: *object recognition*, *attribute recognition*, *relation reasoning*, *logic reasoning*, *knowledge reasoning* and *scene text recognition*.

Sample sourcing. We source data from GQA for *object recognition*, *attribute recognition*, *relation reasoning* and *logic reasoning*, while directly sourcing samples for *knowledge reasoning* and *scene text recognition* from CRIC and TextVQA, respectively. Thanks to the rich annotations provided in GQA and CRIC, for each question we can obtain a functional program that specifies the reasoning steps having to be taken to answer it. We then define a unique set of function operations for each task. A question with functional program containing a specific operation set is assigned to the corresponding stage, as is shown in Tab. 1. Note that the function operations sets of all tasks are not exclusive, as they may share some basic functions of VQA.

Distribution smoothing. To facilitate the study of CLVQA, we create comparable number of samples for each task to avoid the potential issues caused by the imbalanced data.

We showcase the number of samples for each setting in Fig 2. The distributions of question length in Fig. 3 show the similarities among the tasks of CLOVE-scene, and the question variations in CLOVE-function.

Evaluation Metric

For each task in CLVQA, we follow VQA v2 (Goyal et al. 2017) to calculate the accuracy for a question, which is measured via soft voting of the 10 answers. The accuracy is defined as $\text{Acc(ans)} = \min\left\{\frac{\#\text{ans in annotation}}{3}, 1\right\}$. To measure a model's performance on CLVQA, we use the following metrics (Chaudhry et al. 2018; Lopez-Paz and Ranzato 2017).

Function	Operation	Argument
Object Recognition	`Select, Query, Choose`	`name`
Attribute Recognition	`Query, Verify, Choose, Filter`	`color, material, weather` $\cdots$
Relation Reasoning	`Relate, Verify, Choose`	`rel`
Logic Reasoning	`Different, Same, Common, Choose`	`same color, healthier,` $\cdots$
Knowledge Reasoning	`Find with Knowledge Graph`	–
Scene Text Recognition	`scene text recognition`	–

Table 1: Function operations with argument examples. We assign each question to the corresponding task following these rules.

Average accuracy. Let $a_{k,j}$ denote the accuracy evaluated on the held-out testset of $T_j (j \leq k)$ after training a continual learner from T_1 to T_k. The average accuracy at T_k is defined as $A_k = \frac{1}{k}\sum_{j=1}^{k} a_{k,j}$.

Method

Overview of Continual Learning Pipeline

In this section, we introduce our SGP framework for CLVQA, which contains a Symbolic Replay Model (SRM), denoted as S, and a unified VQA model (UniVQA), denoted as U. As is shown in Fig. 4, training the whole continual learner involves two independent procedures of training S and U. To train S, S itself firstly replays scene graph (SG) as the representation of $\mathbf{v}'$ and generates related $\mathbf{q}'$ and $\mathbf{a}'$, which is prompted by a random-sampled relationship (SG-prompt). Combining the annotated scene graph (SG-GT) of $\mathbf{v}$, $\mathbf{q}$ and $\mathbf{a}$ from the current task, S learns the potential scene graph representation and QA patterns from the mix of pseudo-replayed and real samples. To train U, we again use the mix of the pseudo-replayed and real samples, enabling U to learn knowledge from both current and previous tasks.

Leveraging scene graph. Given the difficulties in generating an image with complicated scenes and fine-grained details, replaying highly correlated *image-question-answer* triplets in CLVQA could be intractable. Thus, we resort to scene graphs and leverage a language model (our SRM), DistilGPT2 (Radford et al. 2019), for pseudo-replay. Scene graph is a graphical representation for images and is similar to the form widely used in knowledge base representations (Krishna et al. 2017). Therefore, it plays a role as the bridge connecting vision and language. In addition, scene graphs can be used to power a question engine to generate diverse questions over an image (Hudson and Manning 2019), thus enabling a language model to learn the potential question-answering pattern.

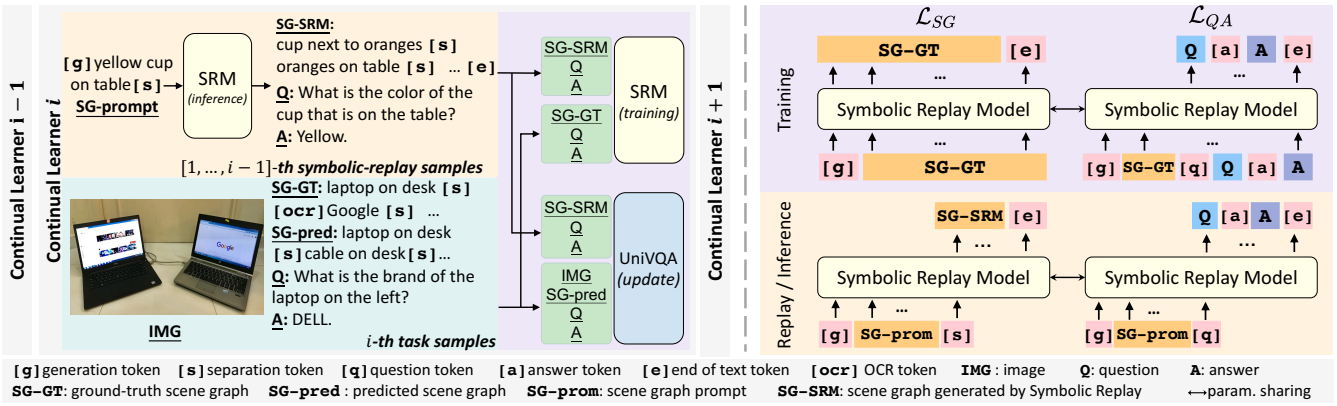

Figure 4: Left: Sequential training for the continual learner. Before training on T_i, the SRM takes SG-prompt as input and generates *SG-SRM-question-answer* triplet for replay. During training, both SRM and UniVQA are trained with the mix of current and replayed samples. Right: Details of SRM. During training, we apply the next token prediction task on the GT scene graph sequence(SG-GT) and supervise question-answer generation using the related scene graph relationships. During inference, the SRM takes an scene graph relationship (SG-prompt) as input, and outputs the completed scene graph and generated question-answer pair. A detailed example is shown on the left part.

Symbolic replay. Combining scene graphs and the language model, we propose SRM, a model which uses an SG-prompt for scene graph replay ($\mathbf{v}'$) and question-answer pair replay ($\mathbf{q}'$,$\mathbf{a}'$). We name this symbolic replay.

Scene graph completion. With the associated scene graph annotations from images, we sequentialize the scene graph for each image and apply next token prediction over the sequence, enabling S to learn the structure of the image. Let $\mathcal{G} = (g_1, g_2, \cdots, g_M)$ denote the scene graph from current or replayed data. During training, we minimize:

$$\mathcal{L}_{SG}(\boldsymbol{\theta}) = -\sum_{|D|}\sum_{m} \log P(g_m|g_1, \cdots, g_{m-1}; \boldsymbol{\theta}),$$

where $P(g_m) = \text{softmax}(S(G_m))$, G_m represents the context tokens of g_m, $\boldsymbol{\theta}$ is the model parameters of S, and D is the training data. We organize the input sequence as shown in Fig. 4: the input is the concatenation of the generation token `[g]` and the scene graph as input, and the output is obtained by shifting a word of the input sequence and appending an end of text token `[e]`. Note that within the input scene graph, a separation token `[s]` is between two individual relationships. During inference for scene graph replay, we construct the input as the concatenation of `[g]`, a randomly sampled SG-prompt and `[s]`, then S completes the SG, denoted as SG-SRM, in an autoregressive manner.

Question-answer generation. We also adapt S for the supervised question-answer generation task. Let G_{qa} denote the scene graph relationships used to generate the GT question-answer pair. During training, we minimize:

$$\mathcal{L}_{QA}(\boldsymbol{\theta}) = -\sum_{|D|} \log P(q, a|G_{qa}; \boldsymbol{\theta}),$$

where $P(q, a) = \text{softmax}(S(G_{qa}))$. As is shown in Fig. 4, we concatenate `[g]`, the question-answer related scene graph relations, question token `[q]`, the question, answer token `[a]`, and the answer as the input. S learns to decode the sequence of question, `[a]`, answer and `[e]`. During inference for question-answer pair replay, the input to S is the concatenation of `[g]`, a random-sampled SG-prompt and `[q]`, and S decodes the question, `[a]`, answer and `[e]`.

Joint-training loss. The loss function for jointly training S is formulated as $\mathcal{L}_{SRM} = \mathcal{L}_{QA} + \lambda\mathcal{L}_{SG}$, where λ is the weight of scene graph completion loss.

Use the pseudo-replayed samples. The current S uses the SG-SRM as input for the next token prediction task and uses the randomly-sampled SG-prompt, generated question and answer to supervise question-answer generation.

Scene-graph prompt for symbolic replay. During training S, our method does not explicitly save any real scene graphs as external memory, as it may raise privacy issues in real-world applications. Instead, we maintain a scene graph database for each specific task. Concretely, we go through the training set in task T_i, and calculate the frequencies of the objects, attributes, relations. Then, we randomly sample one to three scene graph items for replaying based on the statistics, and save them as SG-prompt for further replay.

Unified VQA Transformer

In CLVQA, a VQA model should be able to continuously learn to tackle different types of questions. As a result, it may encounter inputs of different modalities at different stages. *E.g.*, a VQA model might need to read the text and copy the OCR token for answering a question after it is taught how to verify an object's attribute. Therefore, we propose a general VQA model based on Multi-modal Transformer (Hu et al. 2020b). Details are shown in Fig. 5.

Feature extraction. We extract features from general fields (e.g., question word features and object features) and task-specific fields (e.g., OCR token features from an external OCR system when tackling VQA needs reading texts in an image). For language based feature extraction, we use a pre-trained BERT model (Vaswani et al. 2017) for extraction.

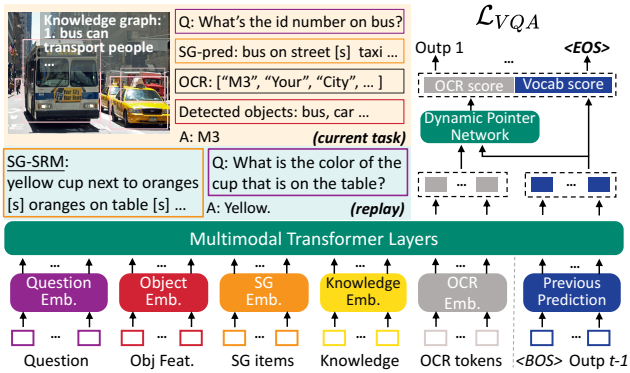

Figure 5: Architecture of UniVQA. It extracts features of all inputs and projects them into a common space. Then, it applies multimodal transformer layers with dynamic pointer network to auto-regressively predict the answer, where each word could be an OCR token or a word in the vocabulary.

We apply this extraction to question, scene graph and knowledge. For object feature extraction, we obtain a set of visual objects through the Faster R-CNN model (Ren et al. 2015) and combine their appearance feature and location feature. For OCR tokens, we follow (Hu et al. 2020b) to extract and utilize their feature representations.

Multi-modal fusion and answer prediction. We apply a stack layers of transformers over the list of all features. Then we follow (Hu et al. 2020b) to decode answer words in an autoregressive manner, where each decoded word could be either an OCR token copied by the dynamic pointer network, or a word from the answer vocabulary.

Data input scheme. In CLVQA, for the current task T_i, we extract object features, question words features, and some task-specific token features (*e.g.*, OCR token features for *scene text recognition* and knowledge feature for *knowledge reasoning*). However, the replayed samples generated by S_{i-1} do not contain images or visual features. To reduce the gap of inputs, we extract the scene graph of the image of the current task via an offline scene graph generation model (Tang et al. 2020), viewing it as plain text and extracting its language-based feature using BERT. We feed this feature of scene graph into the multi-modal transformer as well. For replayed samples, we extract the language feature for SG-SRM. Notably, we mask object features from the current task with a probability of 15% in practice, forcing the U to perceive visual information from the scene graphs.

Training. In task T_i ($i > 1$), U is trained on the mix of current samples and generated samples. In experiments, we generate $\gamma|D_i|$ samples from S_{i-1} and all previous tasks have the same share. That is, when beginning training U_i, we generate $\frac{\gamma}{i-1}|D_i|$ samples for each of the previous $i-1$ tasks. Formally, the loss function of the i-th VQA model is:

$$\mathcal{L}_{VQA}(\phi) = \mathbb{E}_{\{\mathbf{v},\mathbf{q},\mathbf{a}\}\sim D_i}\left[\mathcal{L}\left(U\left(\mathbf{v},\mathbf{q};\phi\right),\mathbf{a}\right)\right] + \mathbb{E}_{\{\mathbf{v}',\mathbf{q}',\mathbf{a}'\}\sim S_{i-1}}\left[\mathcal{L}\left(U\left(\mathbf{v}',\mathbf{q}';\phi\right),\mathbf{a}'\right)\right],$$

where ϕ is U's parameter and $\mathcal{L}$ is answer prediction loss.

Experiments

Baselines

 # Finetune: It finetunes the UniVQA model sequentially without any other specific design.

 # EWC (Schwarz et al. 2018): We apply online EWC, a transformed version of EWC, which accumulates the importance of the stream of tasks.

 # MAS (Aljundi et al. 2018): A regularization based method estimating importance via the gradients of model outputs.

 # LAMOL-m (Sun, Ho, and Lee 2020): LAMOL tackles CL in NLP by generating pseudo samples for experience replay using a language model. Here, we modify its pipeline to adapt to CLVQA task. Specifically, we apply MSE loss for object feature regression and use the same QA loss as in the original LAMOL.

 # VQG (Krishna, Bernstein, and Fei-Fei 2019): It saves part of images and answers from previous tasks and use a visual question generation model to generate coherent questions for replay. A pseudo-replayed sample consists of an image and an answer from a previous task, and the generated question from VQG.

 # Real data replay: It finetunes model augmented with real samples saved into an episodic memory. We adopt two strategies to choose the real samples. (1) **Real-rnd**: randomly choose samples and update the memory with the latest samples; (2) **Real-Kmeans**: we add a learnable token for UniVQA and used its output feature from the multimodal transformer to represent the input sample of the current task. We then use these features to conduct K-means and select the samples closest to each cluster's centroid, following (Huang et al. 2021). In the experiments, we set the memory buffer size to be equal to our SRM's model size plus the saved SG-prompts.

Results and Analyses

Experiments on different task orders. For both CLOVE-scene and CLOVE-function settings, we randomly sample 6 task orders from all possible permutations for evaluation. We use the following abbreviation scheme to simplify the task order notation: *oarlks* denotes *object recognition* → *attribute recognition* → *relation reasoning* → *logic reasoning* → *knowledge reasoning* → *scene text recognition* and *abcdef* denotes *ShopAndDining* → *Workplace* → *HomeorHotel* → *Transportation* → *SportAndLeisure* → *Outdoors*. Besides, for both settings, we set $\gamma = 1.5$ for SGP and report the average accuracy by default.

From the results in Tab. 2, we can find that our method outperforms baselines and SOTA methods on other CL tasks without saving real data by a large margin. Compared to LAMOL-m, our SRM might replay better visual representation than that of LAMOL-m, where object feature regression is applied. Besides, although VQG additionally saves real images, answers and the corresponding task labels as input to generate questions, our SGP still obviously outperforms the VQG. This is probably because by using the scene graph, we can better capture the symbolic relations between visual content and QA pairs. The model can better learn to ask questions by using scene graphs than images.

Method	CLOVE-scene						CLOVE-function					
	abcdef	*bdfcae*	*beacfd*	*beadcf*	*bedfca*	*ecdfab*	*oarlks*	*roslak*	*rklsao*	*rsolak*	*lkosra*	*kaorls*
Finetune	27.53	27.98	28.39	27.71	24.49	25.42	27.60	29.33	21.12	30.65	25.43	22.82
EWC	27.59	27.64	28.47	29.18	24.03	25.48	29.26	30.87	21.87	28.69	23.58	23.27
MAS	27.41	27.15	28.19	27.34	25.40	26.78	28.73	31.59	28.62	28.57	24.26	26.73
VQG	29.15	29.74	30.02	30.27	27.28	28.66	32.78	33.16	29.55	33.82	30.17	28.67
LAMOL-m	29.40	28.52	29.45	29.86	26.52	27.82	28.42	29.04	24.16	32.17	26.94	26.92
SGP (Ours)	**32.21**	**33.72**	**34.37**	**33.18**	**31.84**	**32.98**	**45.97**	**41.80**	**39.05**	**42.95**	**38.65**	**43.62**
Real-rnd	*36.60*	*37.69*	*35.50*	*36.51*	*35.86*	*36.84*	*44.83*	*42.62*	*39.28*	*43.37*	*40.85*	*40.08*
Real-kmeans	*36.91*	*38.15*	*37.01*	*38.30*	*37.93*	*34.86*	*40.28*	*41.19*	*38.49*	*42.21*	*38.39*	*36.29*
Offline	48.45						57.53					

Table 2: Summary of average accuracy(%) for different methods under six task orders in CLOVE-scene and CLOVE-function respectively. We use models at last iteration of last task for testing. Offline training considered as an upper bound is shown at the bottom. Our method outperforms all other real-data-free baselines and achieves comparable performance with real data replay. Rows with italics use the real data while replaying.

Moreover, compared to real data replay methods, although they save the real samples, SGP still achieves comparable performance under the functional setting and is even slightly better on some task orders. Note that on other CL benchmarks, the pseudo-replay *sota* methods usually still has a non-negligible gap with the real-replay methods (Van de Ven and Tolias 2018; Sun, Ho, and Lee 2020). Thus, we believe that symbolic replay could be a very promising direction. Under the CLOVE-scene, our method underperforms real data replay methods. The reason could be that the disjoint image splits in CLOVE-scene increase the difficulty in combating the forgetting in the visual modality. Thus, the replayed scene graphs generated by SRM are less informative than images in retaining visual knowledge.

We also find that the performances of all previous methods without using real data are not satisfactory. In some task orders, regularization-based methods (*i.e.*, EWC & MAS) are even worse than Finetune. It indicates that simply measuring the importance of multi-modal network's parameters might fail to decouple the forgetting in complex multiple modalities and lead to a sub-optimal regularization. LAMOL-m and VQG outperform Finetune in both settings, indicating that the pseudo-replay based methods work for CLVQA, while their improvements are relatively limited.

For the real data replay baselines, we can find that Real-kmeans works better under the CLOVE-scene, while in CLOVE-function, the naive real-rnd works slightly better. We conjecture that Kmeans relies on discriminative sample representations. Image features used in previous works are usually well discriminative, but it might not be the same for the complex multi-modal features in VQA, especially for CLOVE-function, which involves more diverse types of multi-modal reasoning. This indicates that how to represent a VQA sample is still an open problem.

Importance of scene graph replay. We remove the next token prediction task when training SRM, and only generate QA pairs during replay. Comparing #1 and #2 in Tab. 3, we notice an obvious drop in final average accuracy when scene graphs aren't replayed, showing that our replayed scene graph can effectively retain visual contexts from previous tasks.

No.	Prompt	Replay	Scene	Function
#1	Random	Q + A	29.52	40.24
#2	Random	SG + Q + A	32.08	44.21
#3	GT	SG + Q + A	35.09	47.01

Table 3: Comparison of different types of SG-prompt and replay elements (reported under $\gamma = 0.9$).

Using GT scene graph for prompting. As mentioned before, for symbolic replay in SRM, we randomly sample a few scene graph items as a prompt to complete the scene graph and generates corresponding QA pairs. Here, we would like to see the effect of the random prompt sampling on CLVQA. Thus, we propose a variant of SRM, which saves GT scene graphs used to generate QA pairs after training each task and samples the GT scene graph prompts for symbolic replay. Comparing #2 and #3 in Tab. 3, SGM with GT prompts can boost the average accuracy by 3.01% and 2.8% in CLOVE-scene and CLOVE-function settings respectively. It indicates that improving the sampling strategy could be one direction to obtain better CL performance.

Potential of SGP when generating more precise QA pairs. To explore how well an ideal SGP can combat forgetting in CLOVE-scene and CLOVE-function, we compare our SGP with (1) SGP prompt by GT SG-prompt, which means that we save the ground truth scene graph relationship for question generation from the training set of each task, and use the saved SG-prompt as prefix for replay. (2) Replay data with saved predicted scene graph from an offline scene graph predictor, and ground truth QA pair. This can be viewed as a pseudo upper bound of SGP as it replays the real QA pair and corresponding scene graph. We should the results of $\gamma = [0.1, 0.3, 0.5, 0.7, 0.9]$ under *abcdef* in CLOVE-scene and *oarlks* in CLOVE-function. Results are shown in 6. We can observe that for both the CLOVE-scene and CLOVE-function setting, under different numbers of re-

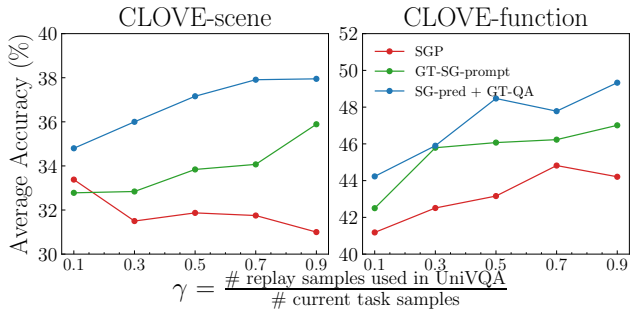

Figure 6: Comparing different replay schemes of SGP. SGP: use randomly sampled SG-prompt to generate scene graph, question and answer. GT-SG-prompt: use ground-truth SG-prompt to generate scene graph, question and answer. SG-pred + GT-QA: save the predicted scene graph of an image by an offline scene graph predictor, ground truth question and answer.

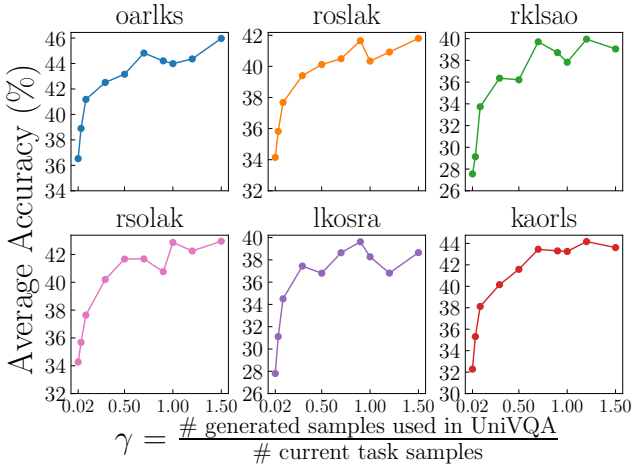

Figure 7: Results of using different number of generated samples in training UniVQA on CLOVE-function.

played samples (γ), (1). Using ground-truth scene graphs outperforms using randomly sampled SG-prompt, indicating that sampling or sourcing better SG-prompt could be a potential solution. (2). Saving scene graphs, questions and answers outperforms all others. This finding indicates that replaying an oracle *scene-graph-quesiton-answer* triplet could further enhance the performance, and this can be a promising direction for real-data-free replay-based method.

Using different number of generated samples in training UniVQA. We set different γ in this experiment to inspect the impact of adopting different amounts of generated samples in training UniVQA. Fig. 7 illustrates that the performance is relatively low at small γ, indicating that a small number of replayed data might not be sufficient against forgetting. Also, the performance reaches stable when $\alpha > 0.7$. The reason could be that when the sample reaches a certain number, generating more data will not increase the diversity of the samples, but lead to imbalanced distribution of the replay

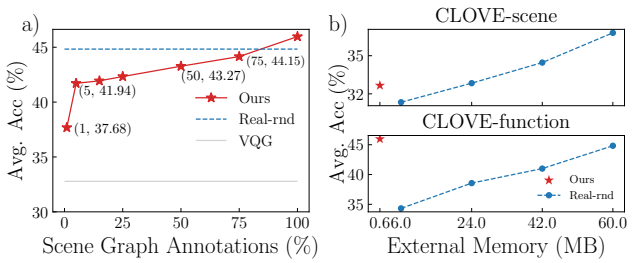

Figure 8: a) Results of using different proportional annotations in training SRM under CLOVE-function-*oarlks*. b) Compare SGP with Real-rnd with different memory size.

and current samples and further jeopardize the performance.

Using less scene graph annotations in training SRM. SRM requires scene graph annotation, which may need extra annotation costs. However, here, we would like to illustrate that SRM is not data-hungry. It can achieve good results with just a small amount of data. Specifically, we reduce the annotations used in SRM training by randomly sampling $r|D_i|$ samples for each task. We showcase the result in Fig. 8 (a). We can see that only using 50% data can lead to an average accuracy of 43.27%, which is still close to the Real-rnd method. Also, the SRM with only 1% training data (i.e., only about 200 VQA samples) still outperforms VQG, the best model without saving real data, by 4.9%.

Comparing external memory size. We compare our method with the real data replay method with different memory sizes for saving data. For our method, SGP only needs to save the sampled prompts, so we calculate the memory size of saving the largest number of samples in our experiments (*i.e.*, $\gamma = 1.5$). For real data replay, we adopt memory of different storage sizes. As is shown in Fig. 8 (b), our method is memory-efficient in terms of saving external data: by saving only 612KB SG-prompts, it achieves comparable performance with Real-rnd saving 24MB (40.2×) real data under CLOVE-scene setting, and Real-rnd saving 60MB (100.4×) real data under CLOVE-function setting.

Conclusion

This paper proposes CLOVE benchmark investigating the CL of VQA under scene- and function-incremental settings. We also propose a framework, Scene Graph as Prompt for symbolic replay, which retains past knowledge by using replayed scene graphs and correlated QA pairs. Extensive experiments show the superiority of our method over other real-data-free CL methods. Besides, we find that previous CL methods face several unique challenges in CLOVE: 1) How regularization methods decouple multi-modal information from network's parameters. 2) How real data replay methods obtain a discriminative representation of a VQA sample suitable for sample selection. 3) How pseudo replay methods generate plausible images or alternatives of images. Finally, we hope CLOVE can provide an enabling resource for future VQA systems with powerful CL ability.

Acknowledgements

This project is supported by the National Research Foundation, Singapore under its NRFF Award NRF-NRFF13-2021-0008, and Mike Zheng Shou's Start-Up Grant from NUS. The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore. Mengmi Zhang was supported by NRF AISG award AISG2-RP-2021-025.

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 139–154.

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T. 2017. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3366–3375.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6077–6086.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Biesialska, M.; Biesialska, K.; and Costa-jussà, M. R. 2020. Continual Lifelong Learning in Natural Language Processing: A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6523–6541. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 532–547.

Chen, Y.-C.; Li, L.; Yu, L.; El Kholy, A.; Ahmed, F.; Gan, Z.; Cheng, Y.; and Liu, J. 2020. Uniter: Universal image-text representation learning. In *European conference on computer vision*, 104–120. Springer.

Chen, Z.; Ma, N.; and Liu, B. 2015. Lifelong Learning for Sentiment Classification. *international joint conference on natural language processing*.

Gao, D.; Li, K.; Wang, R.; Shan, S.; and Chen, X. 2020. Multi-modal graph neural network for joint reasoning on vision and scene text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12746–12756.

Gao, D.; Wang, R.; Shan, S.; and Chen, X. 2019. CRIC: A vqa dataset for compositional reasoning on vision and commonsense. *arXiv preprint arXiv:1908.02962*.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Greco, C.; Plank, B.; Fernández, R.; and Bernardi, R. 2019. Psycholinguistics Meets Continual Learning: Measuring Catastrophic Forgetting in Visual Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3601–3605. Florence, Italy: Association for Computational Linguistics.

Hu, J.; Ruder, S.; Siddhant, A.; Neubig, G.; Firat, O.; and Johnson, M. 2020a. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, 4411–4421. PMLR.

Hu, R.; Singh, A.; Darrell, T.; and Rohrbach, M. 2020b. Iterative Answer Prediction with Pointer-Augmented Multimodal Transformers for TextVQA. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Huang, Y.; Zhang, Y.; Chen, J.; Wang, X.; and Yang, D. 2021. Continual learning for text classification with information disentanglement based regularization. *arXiv preprint arXiv:2104.05489*.

Hudson, D. A.; and Manning, C. D. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6700–6709.

Jiang, H.; Misra, I.; Rohrbach, M.; Learned-Miller, E.; and Chen, X. 2020. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10267–10276.

Johnson, J.; Hariharan, B.; van der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; and Girshick, R. 2017. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Krishna, R.; Bernstein, M.; and Fei-Fei, L. 2019. Information Maximizing Visual Question Generation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1): 32–73.

Lee, S. 2017. Toward Continual Learning for Conversational Agents. *arXiv: Computation and Language*.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.

Liu, Y.; Schiele, B.; and Sun, Q. 2021. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2544–2553.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Mallya, A.; and Lazebnik, S. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 7765–7773.

Marino, K.; Rastegari, M.; Farhadi, A.; and Mottaghi, R. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 3195–3204.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rebuffi, S.-A.; Bilen, H.; and Vedaldi, A. 2017. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Sauer, A.; Schwarz, K.; and Geiger, A. 2022. Stylegan-xl: Scaling stylegan to large diverse datasets. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings*, 1–10.

Schwarz, J.; Czarnecki, W.; Luketina, J.; Grabska-Barwinska, A.; Teh, Y. W.; Pascanu, R.; and Hadsell, R. 2018. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, 4528–4537. PMLR.

Serra, J.; Suris, D.; Miron, M.; and Karatzoglou, A. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, 4548–4557. PMLR.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

Singh, A.; Natarjan, V.; Shah, M.; Jiang, Y.; Chen, X.; Parikh, D.; and Rohrbach, M. 2019. Towards VQA Models That Can Read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8317–8326.

Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; and Dai, J. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Sun, F.-K.; Ho, C.-H.; and Lee, H.-Y. 2020. LAMOL: Language Modeling Is All You Need for Lifelong Language Learning. In *International Conference on Learning Representations*.

Tang, K.; Niu, Y.; Huang, J.; Shi, J.; and Zhang, H. 2020. Unbiased Scene Graph Generation from Biased Training. In *Conference on Computer Vision and Pattern Recognition*.

Van de Ven, G. M.; and Tolias, A. S. 2018. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 3485–3492. IEEE.