Visually Grounded Continual Learning of Compositional Phrases

Xisen Jin Junyi Du Arka Sadhu Ram Nevatia Xiang Ren
Department of Computer Science, University of Southern California
{xisenjin, junyidu, asadhu, nevatia, xiangren}@usc.edu

Abstract

Humans acquire language continually with much more limited access to data samples at a time, as compared to contemporary NLP systems. To study this human-like language acquisition ability, we present VisCOLL, a visually grounded language learning task, which simulates the continual acquisition of compositional phrases from streaming visual scenes. In the task, models are trained on a paired image-caption stream which has shifting object distribution; while being constantly evaluated by a visually-grounded masked language prediction task on held-out test sets. VisCOLL compounds the challenges of *continual learning* (*i.e.*, learning from continuously shifting data distribution) and *compositional generalization* (*i.e.*, generalizing to novel compositions). To facilitate research on VisCOLL, we construct two datasets, COCO-shift and Flickr-shift, and benchmark them using different continual learning methods. Results reveal that SoTA continual learning approaches provide little to no improvements on VisCOLL, since storing examples of all possible compositions is infeasible. We conduct further ablations and analysis to guide future work [1].

1 Introduction

Modern NLP systems, including ones that build on pre-trained language models (Devlin et al., 2019; Radford et al., 2019), excel on a wide variety of tasks. These systems rely on offline (batch) training and have drawn recent criticism due to their inability to adapt to new contexts (Linzen, 2020). In contrast, humans acquire language from evolving environments, require a small memory footprint (McClelland et al., 1995), and can generalize their knowledge to newer tasks (Sprouse et al., 2013). It has been suggested that humans ground perceptual experience to semantically interpret symbols (Bisk et al., 2020; Harnad, 1990; Vigliocco et al., 2014).

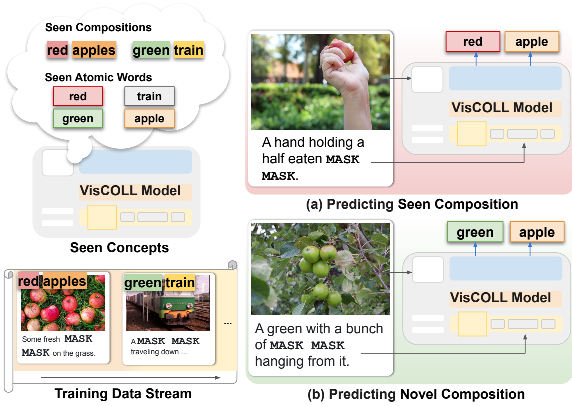


Figure 1: **Illustration of the proposed VisCOLLtask.** The end-task is Masked-Token Prediction: given an image, a model predicts the masked tokens of an associated caption in an online continual learning setup (cf. (a) in the figure). Additionally, we test composition generalization by evaluating on novel compositions (b) which are not encountered at train time.

Model model the challenge, we propose VisCOLL, a **Vis**ually-grounded **Co**ntinua**L** **L**earning setup, to acquire compositional phrases from streaming visual-linguistic data. Models receive a stream of paired image-caption data which has a shifting object distribution. As the end task, we employ masked token prediction of captions given the associated image, as illustrated in Fig. 1(a). This evaluates a model's learned knowledge on composing phrases with the given context.

VisCOLL captures two *inter-related* challenges. First, unlike previous continual learning works on image classification (Kirkpatrick et al., 2017; Zenke et al., 2017), VisCOLL requires predicting, for example, a noun with a verb or an adjective—which results in a significantly large search space. As a result of this increased search space, memory based continual methods (Robins, 1995; Aljundi

[1]Code and data: https://github.com/INK-USC/VisCOLL

et al., 2019a) cannot expect to store prototypes of each visited compositions. Second, the increased search space makes it infeasible to view all possible combinations of atomic words at train time. Therefore, to succeed on VisCOLL, models should generalize to novel compositions at test time (also called *composition generalization*) (Lake and Baroni, 2017; Keysers et al., 2020).

In this work, we extensively study the challenges associated with VisCOLL. To facilitate the research, we construct a continuously shifting data distribution to closely resemble real-word datastream and contribute COCO-shift and Flickr-shift. We benchmark these datasets using multi-modal language modeling architectures (Tan and Bansal, 2019; Su et al., 2020) which achieve state-of-art performance on multiple vision-language tasks. In particular, we don't use any pre-training, instead train randomly initialized models on streaming data using continual learning algorithms (Robins, 1995; Rolnick et al., 2019; Aljundi et al., 2019a) and evaluate their resistance to forgetting and compositional generalization. We quantify the performance and forgetfulness of trained models and evaluate on a novel test split to measure compositional generalization, as shown in Fig. 1(b).

Our proposed VisCOLL benchmark reveals that the gains observed in image classification tasks from state-of-art continual learning algorithms fail to transfer to VisCOLL even with increased memory. On the other hand, composition generalization remains challenging even for offline-training.

To summarize, our contributions are: (i) we propose VisCOLL, a task aimed at continual learning of compositional semantics from visually grounded text inputs (ii) we contribute two datasets COCO-shift and Flickr-shift to study VisCOLL and benchmark them with multi-modal language models (iii) we show that existing continual learning algorithms fail at learning compositional phrases and provide potential future research direction.

2 Related Works

Continual Learning. A major challenge in continual learning is to alleviate catastrophic forgetting (Robins, 1995). Several recent works (Greco et al., 2019; Wang et al., 2019; de Masson d'Autume et al., 2019) study the challenge in the context of NLP. Existing continual learning algorithms can be broadly classified into memory-based approaches (Lopez-Paz and Ranzato, 2017; Aljundi et al., 2019b), pseudo-replay based approaches (Shin et al., 2017), regularization based approaches (Kirkpatrick et al., 2017; Zenke et al., 2017; Nguyen et al., 2018) and architecture based approaches (Rusu et al., 2016). However, these works are mainly designed for image classification tasks where the training data has "clear" task boundaries–*i.e.,* training stream are partitioned into disjoint subsequences. In contrast, task boundaries in VisCOLL are unknown and "smooth" (i.e., with gradual transitions between tasks)–a setting that is closer to real-world situations. Thus, VisCOLL rules out many continual learning algorithms which require explicit task identity and boundary (Kirkpatrick et al., 2017; Rusu et al., 2016).

Modeling Language Compositionality. Capturing compositionality in language has been a long challenge (Fodor et al., 1988) for neural networks. Recent works explore the problem with compositional generalization on synthetic instruction following (Lake and Baroni, 2017), text-based games (Yuan et al., 2019), visual question answering (Bahdanau et al., 2019), and visually grounded masked word prediction (Surís et al., 2019). In particular, Li et al. (2020) study a closely related task of continual learning of sequence prediction for synthetic instruction following. However, their techniques for separating semantics and syntax is restricted to text-only case. Nguyen et al. (2019) investigate continual learning of image captioning, but make strong assumptions on the structure of the data stream, and do not evaluate compositionality.

Different from these, our work focuses on learning compositional language (*e.g.*, phrases) in a continual learning setup. We create realistic training streams to simulate shifting data distribution, with systematic evaluation of compositionality learned in the models. We aim at improving model's ability of acquiring language from real-world streaming data with a low-memory footprint.

3 The VisCOLL Task

Overview. There are two design considerations for VisCOLL: *compositionality* and *continual learning*. To test compositionality, we choose visually grounded masked language modeling where the model needs to compose atomic words to describe complex and novel visual scenes. To simulate a realistic continual learning setup, we construct a dynamic environment where the training data comes in as a non-stationary data stream without clear "task" boundaries. Since the goal is to simulate lan-

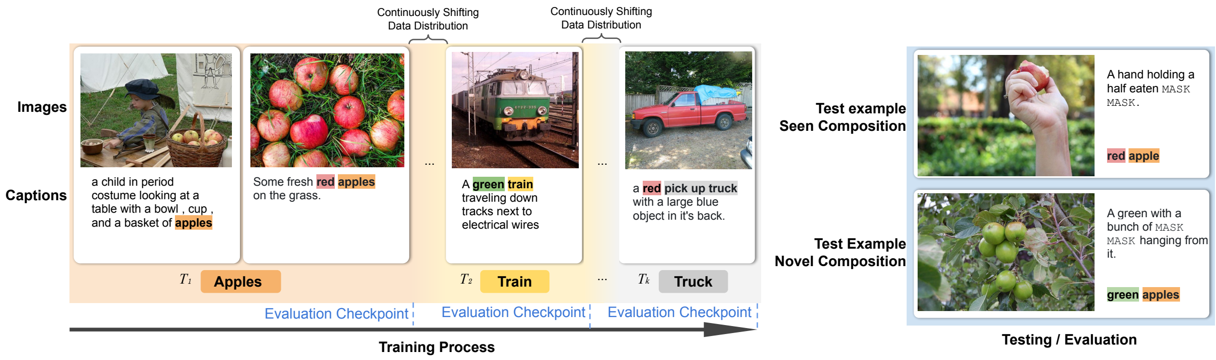


Figure 2: **An overview of training and testing process in VisCOLL.** At **train** time, the model receives a stream of masked captions (highlighted in text) with their associated image. We use the noun appearing in the masked token as the "task" subsequently used to create a continuously shifting data distribution. We further evaluate the model's performance every fixed time interval to quantify "forgetting". At **test** time, the models receives a seen composition or a novel composition of seen words.

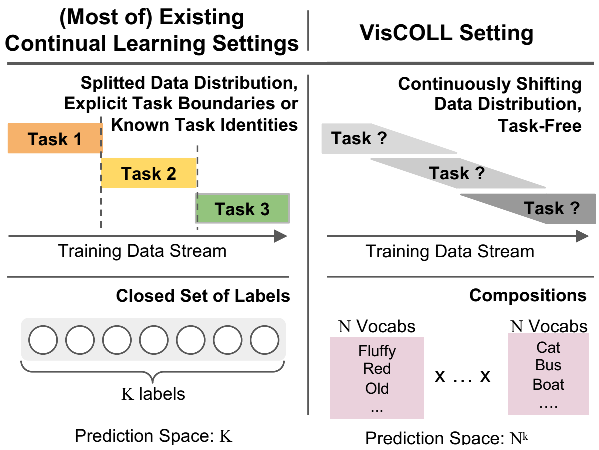


Figure 3: **Comparison of traditional continual learning setup for image classification with the setup of VisCOLL.** In most of existing continual learning settings, task identities are obtainable and can be used to define task boundaries. In contrast, models in VisCOLL is agnostic to task identities (task-free), and fed/trained by a gradually shifting data distribution.

guage acquisition from scractch, VisCOLL models shouldn't initialize weights from a pre-trained network. We introduce details of our task setup in the rest of the section.

Prediction End-Task. We employ visually grounded masked language modeling as the prediction task in VisCOLL. An input instance to this task consists of an image $\mathbf{x}_{img}$, its object bounding boxes $\mathbf{x}_{bbox}$ (without object labels), and the caption text $\mathbf{x}_{text}$, where a contiguous text span in $\mathbf{x}_{text}$ is masked with `MASK` tokens which the model needs to fill. The masked text span $\mathbf{x}_{label}$ always includes a noun and optionally includes verbs or adjectives. We define each noun, verb, and adjective as an atom, and evaluate the model in both "seen" and "novel" composition setting of nouns and verbs/adjectives. For instance, we may test whether the model successfully predicts "*red apples*" (adjective+noun) when the model has seen examples that involve "*red*" and "*apples*" separately (see Figure 2 for an illustration).

Training Data Stream. Unlike traditional offline training setups where a model visits the training examples for multiple passes, we study an online continual learning setup, where the model visits a non-stationary stream of data. We assume the data distribution changes gradually: for example, the model may see more "*apples*" in the beginning, and see less of them later. Unlike prior continual learning benchmarks, we do not assume strict task boundaries, *i.e.*, sudden distribution changes. We illustrate this distinction in Figure 3.

Formally, at each time step t, the model receives a small mini-batch of stream examples $\{\mathbf{X}_i\}_{i=0}^{B-1}$ where $X_i{=}(\mathbf{x}^i_{img}, \mathbf{x}^i_{bbox}, \mathbf{x}^i_{text}, \mathbf{x}^i_{label})$ whose distribution changes over time, *i.e.*, $P(X_i, t_i) \neq P(X_i, t_j)$ where the time step $t_i \neq t_j$. Note that our formulation rules out continual learning algorithms that make use of information about task boundaries. Section 4 formally introduces the data stream construction process.

Evaluation. In addition to the final performance, we also evaluate the model performance every fixed time interval and compute its forgetting as the performance loss over time. For compositional generalization, a novel composition split is used. Details will be covered in the following Sections 4 and 6.

Dataset	COCO-shift	Flickr-shift
Size of training set	639,592	456,299
Size of validation (dev) set	27,918	15,406
Size of regular test set	28,720	15,286
Size of compositional test set	4,426	-
Mean of instance # per task	26,288	487
Median of instance # per task	7,727	137
Average masked span length	2.497	3.380
Number of tasks	80	1,000

Table 1: **Statistics of our constructed datasets COCO-shift and Flickr-shift.**

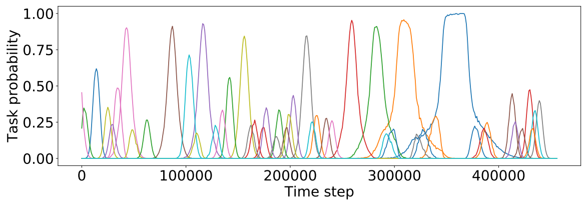


Figure 4: **Probability distributions of** 50 **tasks (the noun in the masked tokens) in Flickr-shift data stream.** Each curve corresponds to a task. x-axis shows the time step, and y-axis shows the probability of visiting the given task at a specific time step.

4 Dataset Construction

We construct our data streams using two popular vision-language datasets: COCO-captions (Chen et al., 2015) and Flickr30k Entities (Plummer et al., 2015) which provide multiple captions for each image in MSCOCO (Lin et al., 2014) and Flickr30k (Young et al., 2014) respectively. We call the resulting datasets COCO-shift and Flickr-shift (see Table 1 for dataset statistics).

Constructing a dataset for VisCOLL involves two key steps: (i) identify the phrase to be masked which involves a noun and associated verbs and adjectives (ii) create a non-stationary data-stream.

Masking Tokens. First, we append part-of-speech tags (POS) to each caption using Stanza (Qi et al., 2020). For Flickr30k Entities, we use the annotated noun-phrases as mask tokens. For COCO-captions, we identify text spans with a regular expression chunker with the following regular expression.

```
CHUNK: <DT>?<JJ|VBG|VBN>*<NN|NNS>+
            <VB|VBD|VBG|VBN|VBP|VPZ>*
```

The resulting text span always includes a noun, and optionally include a determinator and an adjective and verb before or after the noun.

To construct a data-stream, we define a "task" as the object being referred to in the textual input data. For Flickr30k Entities, this is simply the lemmatized noun in the masked span. For COCO-captions, we further map the lemmatized nouns to the 80 object categories defined in MSCOCO using a synonym table provided in (Lu et al., 2018).

Non-Stationary Data-Stream. With a set of "tasks" defined as $\mathcal{T}$, we let each task $T_i \in \mathcal{T}$ gradually introduced in the stream, then gradually fade out. We generate a random permutation of all K tasks $(T_1, T_2, T_3, ..., T_K)$ as the order in which the centroid (mode) of each task distribution arrives in the stream. Each task proposes a task distribution for itself, which is a gaussian with $\mu_i = |D_i|/2 + \sum_{k<i} |D_k|$ and $\sigma_i = |D_i|/2$, where D_i is the set of training instances for task i. μ_i and σ_i roughly determines the centroid and the spread of the distribution of each task. Finally, the algorithm greedily tries to put the proposed number of instances into each time interval to construct the stream. As a result, the constructed data stream has a gradually shifting task distribution without strict boundaries. Figure 4 illustrates the task distribution in our constructed data streams.

For COCO-shift, we separate out instances related to 5,000 images from the official validation set as our test set, and the rest as the test set. For Flickr-shift, we use the official train, validation and the test split. Note that the "task" is only used as an identifier of data distribution for constructing the dataset; the task identities are not revealed to models and the way we construct the data streams ensures there are no clear task boundaries.

Test Split of Novel Compositions. We measure compositional generalization by evaluating on a disjoint set of noun-adjective or noun-verb compositions. We use the compositional test split of COCO dataset by Nikolaus et al. (2019) and remove images related to predefined 24 concept pairs (*e.g.*, black cat, standing child) from the training, validation and the regular test set. The test split is referred to as the *compositional* test set, and the rest is referred to as the *regular* test set.

5 Methods

To benchmark on VisCOLL and study the challenges it poses on model learning, we establish several continual learning baselines. We use visual-language encoder models (Sec. 5.1) for masked token predictions. These models are trained from scratch (*i.e.*, randomly initialized) with continual learning algorithms (Sec. 5.2) to dissuade catas-

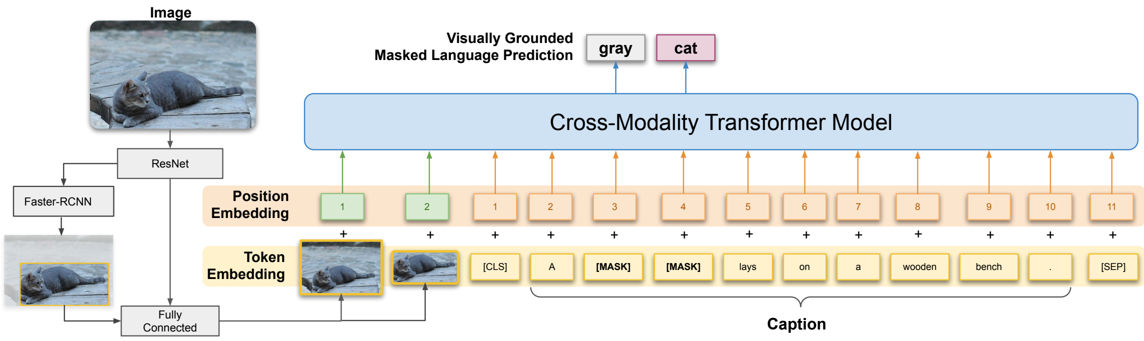


Figure 5: **Model architecture of the visual-language encoder used in VisCOLL.** For the input image, we first extract image-level and object-level features using a FasterRCNN. These features along with the masked caption are passed to a cross-modal transformer (in our case LXMERT and VLBERT) to predict the masked tokens. The model is *randomly initialized without pre-trained weights*, and trained end-to-end with cross-entropy-loss.

trophic forgetting.

5.1 Architectures of Visual Language Model

Recall that our end-task is masked token prediction where the input is an image and a caption with masked out tokens. Since the task of masked token prediction is used as a pre-training method in almost all multi-modal masked language models, we choose two such model architectures, VLBERT (Su et al., 2020) and LXMERT (Tan and Bansal, 2019), as encoder models but expect similar conclusions with other models like ViLBERT (Lu et al., 2019) or UNITER (Chen et al., 2019). Since we seek to simulate the language acquisition process, we *remove the pre-trained weights* from the models and randomly initialize the model weights for both the visual and language transformers.

For both base models, we first extract image and object features in ground truth bounding boxes with a Faster-RCNN (Ren et al., 2015) model with Resnet-101 backbone (He et al., 2015) pretrained on Visual Genome (Krishna et al., 2017) dataset using only object-detection (without attributes). The ground-truth object labels are not provided to the feature extractor model. The extracted features are then passed to the base models along with the caption with masked text span replaced with `[MASK]` tokens. Finally we compute the cross entropy loss with model predictions. We illustrate our base model in Figure 5.

5.2 Compared Methods

Non-continual Learning Comparators. The most common way of training a deep learning model is to perform gradient updates on a mini-batch of independently and identically distributed samples. Since the model has access to the complete data (offline mode), the process is repeated multiple times (multiple epochs); we call this **offline** training. To decouple the effect of training for multiple epochs, we report **offline (one pass)** where we restrict to a single pass over the data. Due to the absence of catastrohpic forgetting, the results are generally better than continual learning scenarios. Note that these two methods do not address the VisCOLL task and potentially indicate and upper-bound of performance.

Continual Learning Methods. In a continual learning setup like VisCOLL, the distribution of training examples is non-stationary, and due to limited memory only a part of the data can be stored. In general, simply performing gradient updates after receiving a mini-batch (also called **Vanilla** Online Gradient Descent), leads to catastrophic forgetting (Robins, 1995).

For VisCOLL, we focus on memory-based continual learning algorithms. These can be easily adapted to our setup as they don't require any task identifiers (not available in VisCOLL). We leave the exploration of regularization based approaches (Hsu et al., 2018) to future work.

(1) *Experience Replay* (**ER**) (Robins, 1995) randomly stores visited examples in a fix-sized memory called the replay memory, and these stored examples can later be randomly sampled for retraining. Similar techniques have been used in reinforcement learning (Schaul et al., 2016; Rolnick et al., 2019). We apply reservoir sampling (Vitter, 1985) to decide examples to store and replace from the memory. The algorithm ensures each visited example has the same probability to be stored in

	Model	VLBERT						LXMERT					
	Metrics	Final Log PPL (↓)		Final BLEU-1/2 (↑)		Forgetting (↓)		Final Log PPL (↓)		Final BLEU-1/2 (↑)		Forgetting (↓)	
	Memory sizes	1,000	10,000	1,000	10,000	1,000	10,000	1,000	10,000	1,000	10,000	1,000	10,000
COCO-shift	Vanilla	5.040		25.96 / 1.29		0.540		5.193		25.19 / 1.81		0.612	
	ER	**3.152**	**2.307**	**45.80** / 20.50	58.06 / 32.22	0.026	-0.142	**3.154**	2.426	**49.52 / 24.56**	61.01 / **35.45**	-0.069	**-0.154**
	AGEM	3.478	3.269	37.94 / 13.56	40.21 / 15.64	0.235	0.145	3.411	3.227	38.71 / 14.72	40.28 / 15.87	0.361	0.257
	ER+MIR	3.342	2.469	**45.80 / 20.87**	58.00 / 32.33	0.012	-0.133	3.162	**2.374**	48.77 / 23.72	60.79 / 35.10	**-0.076**	-0.147
	ER+MIR$_{max}$	3.344	2.473	45.53 / 20.23	**58.14 / 32.48**	**-0.056**	**-0.153**	3.218	2.378	48.03 / 23.10	**61.06** / 35.24	-0.040	-0.140
	ER-10k$_{text-only}$	3.108		47.99 / 22.51		-0.128		3.106		48.07 / 22.57		-0.112	
	Non-continual Learning Comparators												
	Offline (one pass)	1.610		65.27 / 39.61		-		1.783		61.74 / 36.03		-	
	Offline	1.443		68.93 / 44.16		-		1.503		67.53 / 42.79		-	
Flickr-shift	Vanilla	5.691		25.01 / 3.01		0.456		6.107		24.77 / 3.09		0.619	
	ER	5.016	**3.492**	**29.56 / 7.96**	40.23 / 16.80	0.229	0.023	4.949	**3.197**	31.32 / 9.58	**44.34** / 20.73	0.237	0.021
	AGEM	**4.493**	4.393	28.43 / 6.52	28.97 / 7.35	**0.004**	**-0.056**	5.246	5.072	25.18 / 3.63	24.77 / 3.09	**0.108**	0.096
	ER+MIR	5.118	3.504	29.40 / 7.48	**40.27 / 16.81**	0.268	-0.020	4.949	3.211	31.64 / 9.80	44.30 / **20.83**	0.188	0.001
	ER+MIR$_{max}$	5.057	3.555	29.43 / 7.51	40.04 / 16.64	0.233	0.009	**4.788**	3.226	**31.72 / 9.89**	43.51 / 19.95	0.191	**-0.015**
	ER-10k$_{text-only}$	3.958		35.34 / 12.06		0.070		3.461		39.07 / 16.71		-0.008	
	Non-continual Learning Comparators												
	Offline (one pass)	2.590		47.08 / 21.88		-		2.640		47.30 / 22.56		-	
	Offline	2.025		57.13 / 32.29		-		2.025		57.10 / 32.25		-	

Table 2: **Comparison of various training algorithms across two base-models (VLBERT and LXMERT) on the regular test set of COCO-shift and Flickr-shift.** Here, PPL is Perplexity, BLEU-1/2 denotes BLEU-1 and BLEU-2 respectively, metrics with (↑) imply higher is better, similarly, metrics with (↓) imply lower is better. Best performance for each metric in a dataset is emphasized.

the memory. (2) *Average Gradient Episodic Memory* (**AGEM**) (Chaudhry et al., 2019a). Unlike ER, AGEM projects the gradients to a direction of non-increasing average loss computed on a random subset in the memory to alleviate forgetting. (3) *ER with Maximally Interfering Retrieval* (**ER-MIR**) (Aljundi et al., 2019a) extends ER by selecting examples that are most likely to be forgotten in the next one update.

We further implement a method, **ER-MIR**$_{max}$ as a variation of ER-MIR specific to our compositional prediction setting; which, instead of selecting the most likely forgotten phrase, selects the phrases containing the most likely forgotten words. It prevents the importance of an example get underestimated when the example contains mostly easy words and a few forgettable words.

6 Experiments

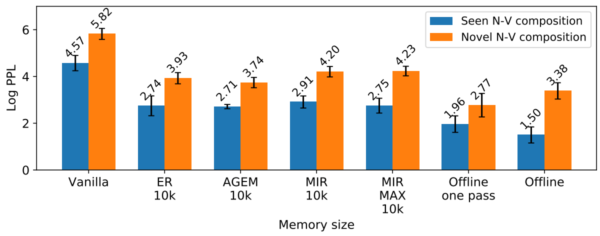
With the VisCOLL task formulation in place, we study: (i) performance of continual learning algorithms on VisCOLL. (ii) effect of the large search space on memory-based continual learning algorithms. (iii) performance on generalizing to novel compositions. (iv) effect of replay memory management strategies. We first describe our implementation details and metrics, and present our results with findings.

Implementation Details. For both VLBERT and LXMERT, we use a transformer with 6 layers, with 384 hidden units and 6 attention heads each. Note that all the parameters are learned from scratch without using pretrained weights. For all continual learning algorithms, we use a memory size of $1k$ and $10k$, corresponding to nearly 0.2% and 2% of data for the two datasets. We report average over 3 runs with the fixed stream and the same set of random seeds. See Appendix for more details.

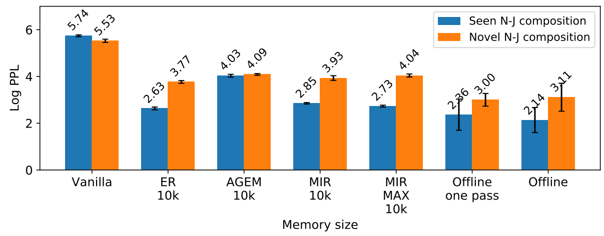
General Evaluation Metrics. We employ Perplexity (PPL) as our primary metrics (lower is better) (Mikolov et al., 2011). Given a masked text span $W{=}[w_1, ..., w_N]$ and the model's prediction probability output $P(W)$, the log-perplexity is defined as, $PPL(W) = -\frac{1}{N}\log P(W)$. We report the perplexity in the log scale. Besides, we also use sentence-level BLEU (Papineni et al., 2002).

6.1 Study of Continual Learning

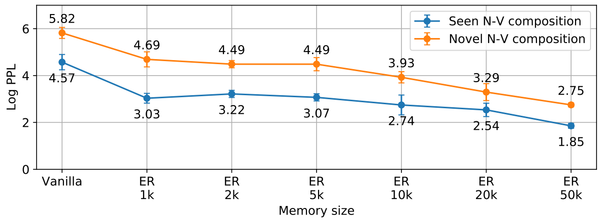
To analyze the continual learning ability of our model, we use two metrics: (i) final perplexity and BLEU: the test set perplexity and BLEU scores when the training ends (ii) forgetting metric: the averaged perplexity increase over all tasks when the training ends compared to the checkpoints (almost) all of training data of a given task is observed. Mathematically, the forgetting metric is calculated as $f_{avg} = \frac{1}{|\mathcal{T}|}\sum_{k\in\mathcal{T}} PPL_T(D_k) - PPL_{c_k}(D_k)$, where $c_k = \arg\min_{c_i\in\mathcal{C}} PPL_{c_i}(D_k)$. $\mathcal{T}$ is the set of all tasks, and $\mathcal{C}$ is the set of all checkpoints; $PPL_{c_k}(D_k)$ is the averaged perplexity over all test examples of task k at the checkpoint c_k, and T is the time step when the training ends. We expect a well-performing method to achieve low final perplexity, high BLEU scores, and low forgetting.



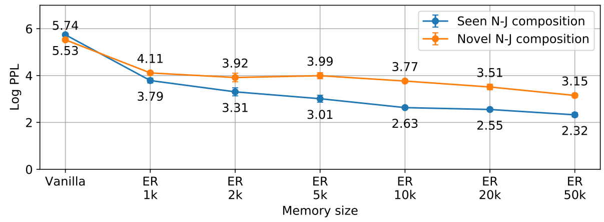
(a) Comparison across methods, Noun-Verb



(b) Comparison across methods, Noun-Adjective

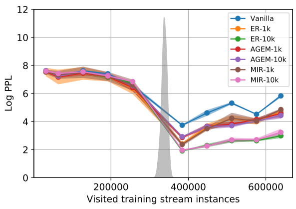


(c) Comparison across memory sizes, Noun-Verb

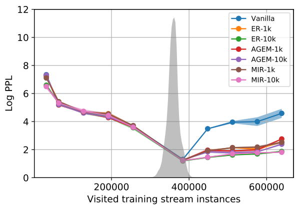


(d) Comparison across memory sizes, Noun-Adjective

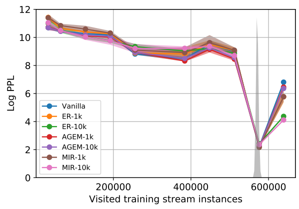
Figure 6: **Results for Compositional Generalization.** We report perplexity of seen and novel compositions across methods (a),(b) and across memory sizes (c),(d) on COCO-shift dataset on noun-verb compositions and noun-adjective compositions separately. We first average the perplexity over examples for each composition individually, then compute the mean over these averaged scores over the set of compositions.



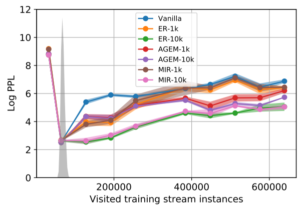
(a) Task `dog`



(b) Task `sports ball`



(c) Task `apple`



(d) Task `book`

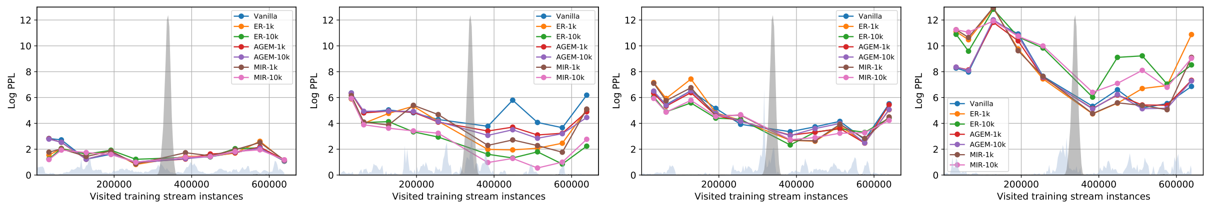
Figure 7: **Comparing effects of continual learning algorithms**, exemplified with four tasks. x-axis is the training examples visited and y-axis is the perplexity. The gray-shaded regions in show the task distribution in the stream.

Tables 2 compares base-models with various continual strategies on the corresponding regular test splits of COCO-shift and Flickr-shift. We discuss our findings below.

Non-stationary vs i.i.d Data Distribution. Across both datasets, it is evident that models trained with the non-stationary distribution (closer to what is observed in the real-world) largely under-perform compared to their i.i.d offline training counterpart at the single epoch setting (20-40 points difference in BLEU scores). This emphasizes that catastrophic forgetting is prevalent in our constructed non-stationary data stream.

Performance of Continual Learning Methods. Despite its simplicity, ER achieves extremely competitive results, scoring within 1-2 PPL of the best performing model. While AGEM achieves very appealing final perplexity on Flickr-shift dataset at the $1k$ memory setting (almost 0.5 points better than alternatives), we find the corresponding BLEU is worse. Given that perplexity evaluates over output probabilities, it is likely that AGEM makes less confident wrong predictions.

We also find ER-MIR and its variant ER-MIR$_{max}$ occasionally outperforms ER, but the improvements are inconsistent over base-models and datasets. This is in stark contrast to continual learning benchmarks on image classification where algorithms like AGEM and ER-MIR achieve SoTA performance. In Fig. 8(a)(b), we illustrate the change of perplexity over time for selected time steps in different methods. We notice that with a memory size of $10k$, on average the forgetting metric for ER is close to or less than zero most of the time. This implies the performance of each task remains constant or improves over what was initially learned.



(a) `sitting` in `black dog` (b) `running` in `big dog` (c) `black` in `black dog` (d) `big` in `big dog`

Figure 8: **Analyzing forgetting in old noun-verb or noun-adjective compositions sharing the same noun**. x-axis is the training examples visited and y-axis is the perplexity of the verb / adjective. The sharp gray-shaded regions are for the noun, while the light-blue regions near x-axis are for the adjectives.

Replay Memory Requirements Compared to Existing Benchmarks. It should be noted that even with a memory of $10k$ examples, the performance of continual learning algorithms are far from the i.i.d setting. In contrast to the popular continual learning benchmarks (Kirkpatrick et al., 2017; Zenke et al., 2017), where storing only a few examples for each class is believed to be sufficient for a good performance (Chaudhry et al., 2019b).

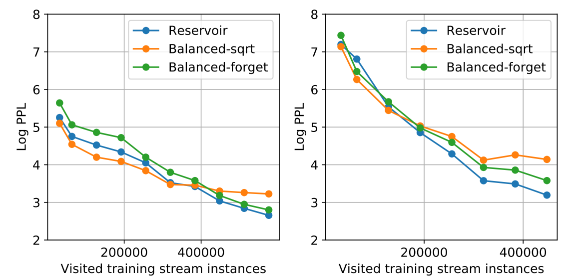
Effect of Multi-Modal Data. To decouple the gains obtained from visual and textual modality, we construct a text-only baseline by zeroing out the image-inputs in our base models and train using ER with memory size $10k$. From Table 2, we find across all cases, text-only baseline is outperformed by its multi-modal counterpart (5 points on BLEU) suggesting information from both image and captions is necessary to perform well on VisCOLL.

Our findings underscore the challenges imposed by VisCOLL and encourages closer examination towards existing continual learning benchmarks.

6.2 Study of Compositional Generalization

To measure compositionality captured by models, in addition to a regular test set, we evaluate on the compositional test set which consists of novel noun-adjective and noun-verb pairs. We compare the performance with seen compositions sharing the same set of atomic words in the regular test set. For a fair comparison with novel splits, we compare the performance on held-out novel pairs with a subset of the regular test-set sharing the same set of atomic words.

Overall Compositional Generalization Results. We plot the results in Figure 6. We note that the performance on novel compositions drops across all cases implying composition generalization is very difficult for visual language transformers. Interestingly, offline (one pass) outperforms offline training on novel compositions, suggesting the latter is over-fitting to the "seen" case.



(a) MSCOCO, VLBERT (b) Flickr, LXMERT

Figure 9: **Effect of memory management strategies, studied with ER and a replay memory of $10k$ examples.** x-axis is the training step; y-axis is the perplexity.

Analyzing Performance on Old Compositions. In an online setting, we further probe the model's predictive performance on old seen compositions. Interestingly, we find that the performance is largely dependent on the atomic words used in the compositions. For instance, the performance drop on predicting "black" in the composition "black dog" is relative small (Figure 8(c)) compared to predicting "big" in "big dog" (Figure 8(d)).

6.3 Study of Memory Management Strategies

We further study two memory scheduling strategies to account for a limited memory but large search space. Recall that the reservoir sampling applied our main experiments keeps each visited example has the same probability to be stored in the memory. We study two methods targeting storing more useful examples, aiming at: (i) storing diverse compositions, and (ii) prioritizing likely forgotten words.

We first propose target word distributions p_{tgt} in the memory. For (i), the target probability of each word is set proportional to the square root of its frequency in the visited stream. Thus, highly frequent words would take a smaller portion compared to reservoir sampling where the word distribution in the memory is linear to its frequency in the visited stream, leaving space for storing more diverse examples. We call this strategy **Balanced-sqrt**. For (ii), the target probability is proportional to its frequency in the stream multiplied by its forgetting estimated during training (*i.e.,* loss increase). We call this strategy **Balanced-forget**.

For both strategies, given the word distribution in the memory p_{mem} and target word distributions p_{tgt}, we minimize the KL-divergence $\mathrm{KL}(p_{mem}||p_{tgt})$. Thus, each time an example is received from the stream, we choose the memory example which if replaced causes the largest decrease in KL-divergence. If there are no such memory examples that let KL-divergence decrease, we discard the example.

The results are compared in Figure 9. We find that (i) diversifying storage improves performance at the early stage of the stream but not in the later stages; (ii) prioritizing words likely to be forgotten does not improve performance. Thus, future works should find a balance between storing more diverse or important examples and respecting original data distribution.

7 Conclusion

We propose VisCOLL, a novel continual learning setup for visually grounded language acquisition. VisCOLL presents two main challenges: continual learning and compositionality generalization. To facilitate study on VisCOLL, we generate continuously shifting data-stream to construct two datasets, namely COCO-shift and Flickr-shift, and establish evaluation protocols. We benchmark our proposed datasets with extensive analysis using state-of-the-art continual learning methods. Experiments reveal that continual learning algorithms struggle at composing phrases which have a very large search space, and show very limited generalization to novel compositions. Future works include looking into models and continual learning algorithms to better address the challenge.

Acknowledgements

This research is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111990059, and the DARPA MCS program under Contract No. N660011924033 with the United States Office Of Naval Research. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. We would like to thank all the collaborators in USC INK research lab for their constructive feedback on the work.

References

Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. 2019a. Online continual learning with maximally interfered retrieval. In *NeurIPS*.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. 2019b. Gradient based sample selection for online continual learning. In *NeurIPS*.

Dzmitry Bahdanau, Harm de Vries, Timothy J O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron Courville. 2019. Closure: Assessing systematic generalization of clevr models. *arXiv preprint arXiv:1912.05783*.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, M. Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph P. Turian. 2020. Experience grounds language. *ArXiv*, abs/2004.10151.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019a. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. 2019b. On tiny episodic memories in continual learning.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *ArXiv*, abs/1504.00325.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Jerry A Fodor, Zenon W Pylyshyn, et al. 1988. Connectionism and cognitive architecture: A critical analysis.

C. Greco, Barbara Plank, R. Fernández, and R. Bernardi. 2019. Psycholinguistics meets continual learning: Measuring catastrophic forgetting in visual question answering. In *ACL*.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. 2018. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.

Brenden M Lake and Marco Baroni. 2017. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *arXiv preprint arXiv:1711.00350*.

Yuanpeng Li, Liang Zhao, Kenneth Church, and Mohamed Elhoseiny. 2020. Compositional language continual learning. In *International Conference on Learning Representations*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Seattle, Washington. Association for Computational Linguistics.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *NIPS*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2018. Neural baby talk. In *CVPR*.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419.

Tomáš Mikolov, Anoop Deoras, Stefan Kombrink, Lukáš Burget, and Jan Černockỳ. 2011. Empirical evaluation and combination of advanced language modeling techniques. In *Twelfth Annual Conference of the International Speech Communication Association*.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. 2018. Variational continual learning. In *International Conference on Learning Representations*.

Giang Nguyen, Tae Joon Jun, Trung Tran, and Daeyoung Kim. 2019. Contcap: A comprehensive framework for continual image captioning. *arXiv preprint arXiv:1909.08745*.

Mitja Nikolaus, Mostafa Abdou, Matthew Lamm, Rahul Aralikatte, and Desmond Elliott. 2019. Compositional generalization in image captioning. In *CoNLL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle,

A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A {Python} Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Anthony V. Robins. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connect. Sci.*, 7:123–146.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pages 348–358.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

T. Schaul, John Quan, Ioannis Antonoglou, and D. Silver. 2016. Prioritized experience replay. *CoRR*, abs/1511.05952.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999.

Jon Sprouse, Carson T Schütze, and Diogo Almeida. 2013. A comparison of informal and formal acceptability judgments using a random sample from linguistic inquiry 2001–2010. *Lingua*, 134:219–248.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*.

Dídac Surís, Dave Epstein, Heng Ji, Shih-Fu Chang, and Carl Vondrick. 2019. Learning to learn words from visual scenes. *arXiv preprint arXiv:1911.11237*.

Hao Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP/IJCNLP*.

Gabriella Vigliocco, Pamela Perniss, and David Vinson. 2014. Language as a multimodal phenomenon: implications for language learning, processing and evolution.

Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 796–806.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78.

Xingdi Yuan, Marc-Alexandre Côté, Jie Fu, Zhouhan Lin, Christopher Pal, Yoshua Bengio, and Adam Trischler. 2019. Interactive language learning by question answering. *arXiv preprint arXiv:1908.10909*.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org.

Dataset	COCO-shift				Flickr-shift			
Method/Model	VLBERT		LXMERT		VLBERT		LXMERT	
Memory sizes	1,000	10,000	1,000	10,000	1,000	10,000	1,000	10,000
iid-online	0 h 59 min		2 h 6 min		0 h 35 min		1 h 10 min	
Vanilla	0 h 59 min		2 h 0 min		0 h 34 min		1 h 12 min	
ER	1 h 37 min	1 h 40 min	3 h 26 min	3 h 35 min	1 h 5 min	1 h 8 min	2 h 6 min	2 h 10 min
AGEM	2 h 57 min	2 h 36 min	5 h 16 min	5 h 20 min	1 h 32 min	1 h 38 min	3 h 30 min	3 h 29 min
ER-MIR	3 h 49 min	3 h 31 min	7 h 30 min	8 h 22 min	2 h 9 min	2 h 42 min	5 h 1 min	5 h 14 min
ER-MIR$_{max}$	3 h 29 min	3 h 30 min	8 h 7 min	8 h 20 min	2 h 19 min	2 h 49 min	4 h 58 min	5 h 8 min

Table 3: Average training time over a single pass of the stream.

black cat	big bird	red bus	small plane
eat man	lie woman	white truck	small cat
brown dog	big plane	ride woman	fly bird
white horse	big cat	blue bus	small table
hold child	stand bird	black bird	small dog
white boat	stand child	big truck	eat horse

Table 4: 24 compositions used for the compositional test split of COCO-split dataset.

A Details of Dataset Construction

Heldout Phrases. We put the complete list of 24 noun-verb and noun-adjective compositions used as novel compositions in Table 4, which are provided in (Nikolaus et al., 2019).

B Hyperparameter Settings

Since the complete stream should not be assumed known apriori in the online learning setting, following prior work (Chaudhry et al., 2019a), we use a small portion (10%) of the training data and the validation set to perform hyperparameter search. We use AdamW optimizer (Loshchilov and Hutter, 2019) throughout the experiements. We tune the learning rate based on the validation performance on the Vanilla method averaged over 3 runs. For a fair comparison in the online learning setup, we use the same learning rates for all methods. The learning rate is selected from $\{2e^{-4}, 1e^{-4}, 5e^{-5}\}$ and decided as $1e^{-4}$. We set the batch size to 32 throughout the experiments. For ER, ER-MIR and ER-MIR$_{max}$, at each training iteration, we replay the same number of examples from the memory as the examples received from the stream (*i.e.*, training batch size), following the convention in recent works (Aljundi et al., 2019a; Chaudhry et al., 2019b).

AGEM, unlike ER, requires a larger set of memory examples to compute regularizations. We set the numbers to 80 and 64 respectively for COCO-shift and Flickr-shift. While larger numbers can be preferable, it introduces significant time and resource consumption overhead in our problem setup, which is much larger in scale compared to existing continual learning benchmarks.

Similarly, ER-MIR and ER-MIR$_{max}$ introduce a hyperparameter for the size of the "candidate set" to retrieve examples that are most likely to be forgotten. We set the hyperparameters as 80 and 64 for COCO-shift and Flickr-shift respectively.

Task-order / Method	Vanilla	ER-1k	ER-10k
Asc. Frequency	4.31	3.22	2.42
Desc. Frequency	5.18	3.58	2.53
Random (main results)	5.04	3.15	2.31

Table 5: Performance with different task orders in COCO-shift and the VLBERT model.

C Effect of Data Order

Data order has been known to affect performance in continual learning (Greco et al., 2019). To illustrate this point, we conduct a simple experiment where the task centroids are sorted in the ascending or descending order. We show the log perplexity metrics in Table 5. The results show a significant log-perplexity gap, which verifies that data order may significantly affect performance. We leave more in-depth analysis as future works. Note that throughout our main experiments, the task order is fixed as a random order.

D Infrastructures and Statistics

We use PyTorch 1.0.0 (Paszke et al., 2019) with CUDA toolkit version 10.1. We train our models with NVIDIA 2080 Ti GPUs. Our VLBERT models have 23,564,040 trainable parameters and LXMERT models have 58,614,794 trainable parameters. We report the average training time over a single pass of the stream in table 3.