Composition-Incremental Learning for Compositional Generalization

Zhen Li[1,2], Yuwei Wu[1,2], Chenchen Jing[3], Che Sun[2*], Chuanhao Li[4,1,2*], Yunde Jia[2,1]

[1]Beijing Key Laboratory of Intelligent Information Technology,
School of Computer Science & Technology, Beijing Institute of Technology
[2]Guangdong Laboratory of Machine Perception and Intelligent Computing, Shenzhen MSU-BIT University
[3]Zhejiang University of Technology
[4]Shanghai AI Laboratory
{li.zhen, wuyuwei}@bit.edu.cn, jingchenchen@zju.edu.cn, {sunche, jiayunde}@smbu.edu.cn, lichuanhao@pjlab.org.cn

Abstract

Compositional generalization has achieved substantial progress in computer vision on pre-collected training data. Nonetheless, real-world data continually emerges, with possible compositions being nearly infinite, long-tailed, and not entirely visible. Thus, an ideal model is supposed to gradually improve the capability of compositional generalization in an incremental manner. In this paper, we explore **Comp**osition-**I**ncremental **L**earning for Compositional Generalization (CompIL) in the context of the compositional zero-shot learning (CZSL) task, where models need to continually learn new compositions, intending to improve their compositional generalization capability progressively. To quantitatively evaluate CompIL, we develop a benchmark construction pipeline leveraging existing datasets, yielding MIT-States-CompIL and C-GQA-CompIL. Furthermore, we propose a pseudo-replay framework utilizing a visual synthesizer to synthesize visual representations of learned compositions and a linguistic primitive distillation mechanism to maintain aligned primitive representations across the learning process. Extensive experiments demonstrate the effectiveness of the proposed framework.

Repository — https://github.com/Lixsp11/CompIL

Introduction

Recently, compositional generalization has garnered much attention, with substantial progress in improving models' compositional generalization capability on fixed, pre-collected data (Huang et al. 2024b,a; Li et al. 2023). Given the ever-emerging nature of real-world data, *e.g.*, the recurrence of previously observed compositions and the appearance of new, unseen ones, it is essential to understand how to enrich the training data to further boost the compositional generalization capability of models. To this end, we conduct a preliminary investigation into the impact of training data on this capability in the compositional zero-shot learning (CZSL) task, which aims to recognize unseen compositions of attributes and objects (known as primitives) by leveraging knowledge from observed compositions. Specifically, we conduct comparative experiments by varying the number of samples in the training data while keeping the number of compositions fixed, or vice versa. As illustrated in Figure 1, the steeper slope of the orange line indicates that the number of compositions has a significantly impact on models' compositional generalization capability, while increasing sample size with a fixed number of compositions offers minimal benefit. More details can be found in the **supplementary material**. These findings suggest that we can improve the compositional generalization capability of models by increasing the diversity of training compositions from a data-driven perspective.

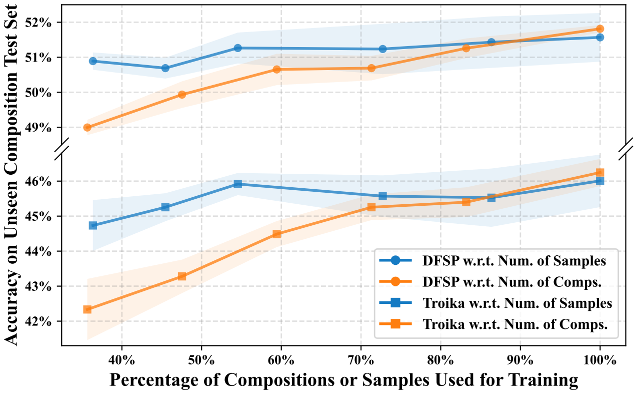


Figure 1: Accuracy of CZSL models on the unseen composition test set trained with data containing varying number of compositions or samples. Increasing training compositions boosts compositional generalization significantly more than increasing training samples.

Nevertheless, obtaining sufficiently varied compositions is both costly and time-consuming (Saini, Pham, and Shrivastava 2024; Xu et al. 2024), and the expense of training models from scratch becomes prohibitive as the dataset size scales. This raises an important question: ***Can models continually learn from an increasing number of compositions to improve their compositional generalization capability progressively?*** To answer this question, we propose a new setting: Composition-Incremental Learning for Compositional Generalization (CompIL), where models are required

[*]Corresponding author: Che Sun and Chuanhao Li

to learn sequentially on a series of tasks containing disjoint compositions. Specifically, each task contains a set of samples sharing the same primitive set, whereas the compositions of different tasks vary significantly in semantics. These semantics gaps simulate the staged data collection process in the real world, where data distribution typically follows a long-tailed pattern and continuously evolves (Gama et al. 2014; Yao et al. 2022; Li et al. 2024b). The difficulty of our setting stems from the following two challenges: (1) **Composition knowledge forgetting.** Forgetting (Li and Hoiem 2017) remains a fundamental challenge in continual learning and is even more pronounced in our setting. The vast number of compositions, coupled with the relatively small number of samples per composition, exacerbates the risk of forgetting. (2) **Primitive representation drift.** Learning semantically aligned primitive representations has been proven to enhance the compositional generalization capability of models (Li et al. 2023). However, the semantic differences between tasks foster models focusing on task-specific representations, which may not be applicable across tasks. For example, in one task, “ancient” emphasizes age, as in “ancient castle”, while in another, it emphasizes obsolescence, as in “ancient computer”.

We explore CompIL in the context of CZSL, and develop an efficient benchmark construction pipeline along with a comprehensive evaluation protocol. By formalizing the constraints in benchmark construction as an integer optimization problem, our pipeline constructs CompIL benchmarks from existing datasets via step-by-step optimization. Moreover, we introduce a hierarchical clustering strategy to enhance inter-task semantic diversity, enabling the benchmark to better align with the dynamic real world. In practice, we construct two new benchmarks MIT-States-CompIL and C-GQA-CompIL. We evaluate various existing continual learning methods and find they struggle on CompIL, often underperforming even a zero-shot baseline.

We present a pseudo-replay framework for CompIL by synthesizing pseudo-samples of past compositions and training them jointly with current task data. Recognizing that the compositions is infinite and difficult to disentangle in visual representations (Lu et al. 2023), while primitives are naturally separable in language (*e.g.*, attribute and object words), we design a visual synthesizer based on the language encoder of a pretrained vision-language model. Leveraging the vision-and-language alignment of the pretrained model, the synthesizer learns to take attribute and object words as input and synthesize corresponding visual composition representations, as pseudo-samples. Additionally, we introduce a linguistic primitive distillation mechanism. It constrains the model to maintain consistent predictions for past compositions while learning new ones, effectively mitigating primitive representation drift. Experimental results show our framework consistently improves the compositional generalization capability of models throughout the learning process.

To summarize, our contributions are as follows:

- We present a practical and challenging setting termed CompIL, where models continually learn new compositions to improve their compositional generalization capability progressively.
- We develop an efficient pipeline for constructing CompIL benchmarks for quantitative evaluation and construct two new benchmarks in the context of CZSL.
- We propose a pseudo-replay framework for CompIL by synthesizing visual representations of learned compositions and maintaining aligned primitive representations throughout learning.

Related Work

Compositional Generalization

Numerous benchmarks (Ma et al. 2023; Li et al. 2024c; Ray et al. 2024) have been proposed to evaluate compositional generalization capability, and various sophisticated model architectures and training strategies (Huang et al. 2024a; Li et al. 2024a) have been proposed to boost this capacity. A key area of compositional generalization research is compositional zero-shot learning. Benefiting from the capability of pre-trained vision-language models, *e.g.*, CLIP (Radford et al. 2021), diverse cross-model mechanisms have been proposed to enhance compositional generalization capability. For example, replacing attributes and object labels with trainable prompts (Nayak, Yu, and Bach 2023), employing cross-modal fusion to enhance feature integration (Lu et al. 2023), and using multi-branch models to better align vision-language representations (Huang et al. 2024b). These works focus on improving the compositional generalization capability on pre-collected and fixed data. Differently, our work aims to improve this capability progressively using a growing data stream with various compositions to cope with the ever-changing world.

A few works have explored extending the boundaries of compositional generalization with increasing data. VisCOLL (Jin et al. 2020) investigated the incremental acquisition of compositional phrases from streaming visual data and evaluated the compositional generalization capability after the learning process. Liao et al. (2024) focused on the multi-object compositions and proposed a compositional few-shot testing protocol for evaluating compositional generalization in continual learning. CCZSL (Zhang, Feng, and Yuan 2024) required models to continually learn compositions that include unseen primitives to expand the learned primitive set over time. CompILer (Zhang et al. 2024) also introduced a composition-incremental learning task, which separately identifies attributes and objects, aiming to mitigate forgetting of each. In contrast to the above, we explore learning continually from an increasing number of compositions within a fixed primitive set, aiming to improve models’ compositional generalization capability progressively.

Continual Learning

Continual learning is to train a single model that can incrementally update its knowledge with a continuous stream of tasks without catastrophic forgetting of previously learned tasks. Existing methods alleviated catastrophic forgetting via regularization (Kirkpatrick et al. 2017; Dhar et al. 2019), expanding models for each task (Li et al. 2019; Hu et al. 2023), or storing samples of previous tasks (Chaudhry et al. 2019; Buzzega et al. 2020; Li et al. 2024d).

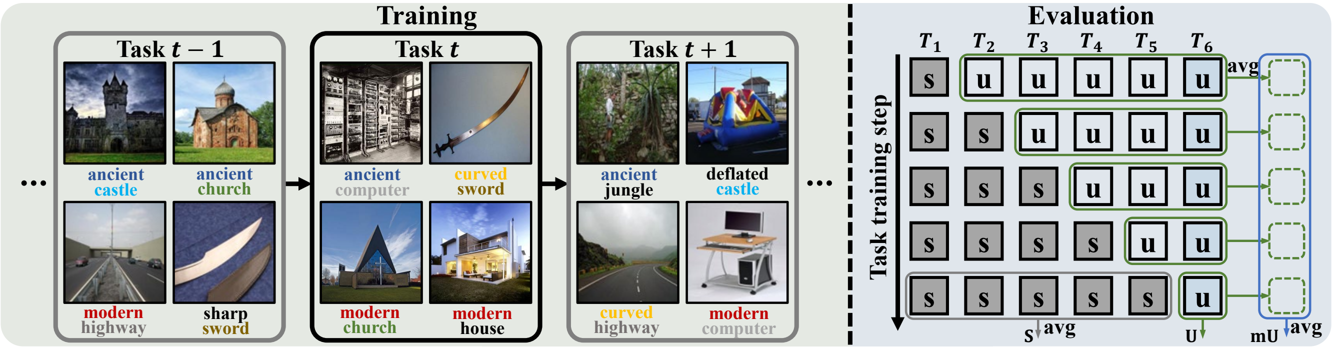


Figure 2: Illustration of our CompIL setting, taking the proposed MIT-States-CompIL benchmark as an example. The left side illustrates training samples from the tasks, where samples from different tasks vary significantly in semantics. For instance, the attribute "ancient" describes buildings in task $t-1$, outdated technology in task t, and natural landscapes in task $t+1$; the object "castle" varies likewise. The right side depicts the evaluation process, with rows representing training steps and each column indicating performance on the corresponding task.

Recently, several works have shown interest in continual learning with CLIP. Thengane et al. (2022) showed that CLIP achieves state-of-the-art performance via a zero-shot paradigm in continual learning settings. AttriCLIP (Wang et al. 2023a) leveraged a trainable attribute word bank to encode image attributes as textual prompts, enabling efficient continual learning while mitigating catastrophic forgetting. CGIL (Frascaroli et al. 2024) trained a dedicated Variational Autoencoder (Kingma 2013) for each class to generate synthetic visual features that are then used for the continual adaptation of CLIP models. Although these methods have demonstrated impressive results in mitigating catastrophic forgetting or preventing zero-shot capability degradation, their use to enhance compositional generalization capability remains under-explored. By contrast, we propose a pseudo-replay framework based on visual composition synthesis to enhance compositional generalization capability while mitigating forgetting.

Composition-Incremental Learning

Formulation

We take CZSL as a representative example to illustrate the formulation of CompIL. Given an attribute set $\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$ and an object set $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$ as the primitive concepts, the compositional label space $\mathcal{C} = \mathcal{A} \times \mathcal{O}$ is defined as their Cartesian product. CZSL divides $\mathcal{C}$ into 2 disjoint subsets, *i.e.*, $\{\mathcal{C}_1, \mathcal{C}_2\}$, aiming at learning a model from $\mathcal{C}_1$ to recognize images from novel composition set $\mathcal{C}_2$. In composition-incremental learning, the composition set $\mathcal{C}$ is further divided into $T+1$ disjoint subsets, *i.e.*, $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_T, \mathcal{C}_{T+1}\}$, where $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for any $i \neq j$ and $\bigcup_{t=1}^{T+1} \mathcal{C}_t \subseteq \mathcal{C}$. Each subset $\mathcal{C}_t$ (except for $\mathcal{C}_{T+1}$) with corresponding images $\mathcal{X}_t$ forms a task of CompIL, denoted as $\mathcal{T}_t = \{(x_i, c_i) | x \in \mathcal{X}_t, c \in \mathcal{C}_t\}$, resulting in a total of T tasks for the continual learning process.

The model is trained sequentially across these tasks. When learning the t-th task, the training images only contain compositions from $\mathcal{C}_t$, while the evaluation is performed on both seen composition set $\mathcal{C}_t^s$ and unseen composition set $\mathcal{C}_t^u$ following the standard compositional zero-shot learning, where $\mathcal{C}_t^s = \bigcup_{i=1}^{t} \mathcal{C}_i$ and $\mathcal{C}_t^u = \bigcup_{i=t+1}^{T+1} \mathcal{C}_i$, respectively. Note that $\mathcal{C}_{T+1}$ is always included in the unseen set $\mathcal{C}_t^u$ for any task t, providing a static unseen composition set for consistent and quantitative evaluation of the model's compositional generalization capability during the learning process.

Evaluation Metric

The evaluation encompasses two aspects: the model's average performance throughout the continual learning process and its final performance after the process, as illustrated in the right part of Figure 2. Specifically, after training on task t, we follow the well-established CZSL evaluation protocol by Purushwalkam et al. (2019) to evaluate the model on the seen composition set $\mathcal{C}_t^s$ and the unseen composition set $\mathcal{C}_t^u$, including the best seen accuracy S_t, the best unseen accuracy U_t, and the area under the curve AUC_t for unseen versus seen accuracy. We report the average best unseen accuracy $mU = \frac{1}{T}\sum_{i=1}^{T} U_i$ and the average area under the curve $mAUC = \frac{1}{T}\sum_{i=1}^{T} AUC_i$, which measures the model's capability when continually learning new tasks. The best seen accuracy S_t can be further divided by task as $S_{t,i}$ to quantify the performance degradation in the past tasks, which is defined as $fS = \frac{1}{T}\sum_{i=1}^{T}(S_{i,i} - S_{T,i})$. Additionally, we report the final U_T, S_T and AUC_T, denoted as U, S and AUC, to reflect the model's performance after the learning process.

Benchmark Construction

To quantitatively assess the performance of models in CompIL, we propose a pipeline that constructs CompIL benchmarks leveraging existing datasets. Besides ensuring the formulation, the pipeline also simulates the staged data collection process in the real world, where data distribution typically follows a long-tailed pattern and evolves over time. Specifically, semantically similar compositions tend to be densely observed within a period, leading to semantic differences between tasks.

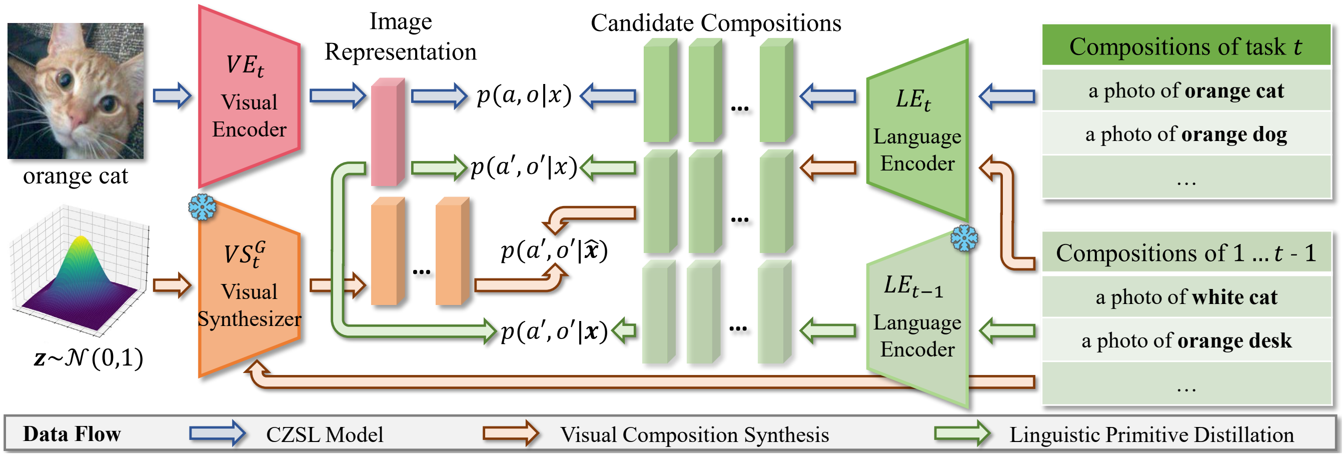


Figure 3: Overview of the pseudo-replay framework in the context of compositional zero-shot learning. The CZSL model contains a visual encoder VE_t and a language encoder LE_t. For simplicity, we omit possible connections between two components.

The pipeline employs a hierarchical clustering strategy to take the aforementioned considerations into account. Given an existing CZSL dataset containing a set of compositions and corresponding images, we first cluster the compositions into M semantically similar mini-groups $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_M\}$, using K-Means algorithm (Lloyd 1982). Each mini-group consists of compositions with the same attribute and semantically similar objects. The semantic similarity is quantified using Lin similarity (Lin et al. 1998) calculated on WordNet (Miller 1995). Next, we assign each mini-group to one of the T tasks. Considering the definition of compositional generalization, which refers to unseen compositions of seen primitives, we obtain a shared primitive set across different tasks by maximizing the objective function

$$\sum_{t=1}^{T}\left[\mathrm{N_a}\left(\bigcup_{i=1}^{M}\mathcal{G}_i \mid \mathcal{G}_i \in \mathcal{T}_t\right) + \mathrm{N_o}\left(\bigcup_{i=1}^{M}\mathcal{G}_i \mid \mathcal{G}_i \in \mathcal{T}_t\right)\right], \tag{1}$$

where $\mathrm{N_a}(\cdot)$ and $\mathrm{N_o}(\cdot)$ are functions that calculate the number of attribute and object types given the composition set, respectively. The objective function can be transformed into an integer optimization problem, and we use Gurobi Optimizer (Gurobi Optimization, LLC 2024) to find an approximate solution in a finite number of optimization steps. Finally, the unseen composition test set of the CZSL dataset is designated as task $T + 1$. We combine it with the above T tasks to form a CompIL benchmark.

We use the pipeline construct MIT-States-CompIL and C-GQA-CompIL benchmarks containing $T = 5$ tasks based on widely used CZSL datasets MIT-States (Isola, Lim, and Adelson 2015) and C-GQA (Mancini et al. 2022). Taking the MIT-States-CompIL benchmark as an example, Figure 2 illustrates our CompIL setting. Additional statistics can be found in **supplementary material**.

Pseudo-Replay Framework

The overview of the proposed framework in the context of compositional zero-shot learning is shown in Figure 3. Concretely, for a CZSL model containing a visual encoder VE_t and a language encoder LE_t, the framework integrates a visual synthesizer VS that synthesizes visual representations of past tasks. The synthesized representations are combined with current task samples to train the CZSL model jointly. Additionally, the language encoder of the CZSL model, finalized on the last task, is preserved and utilized to perform distillation with the current language encoder.

Preliminary

We first outline the pipeline of recent mainstream CZSL methods leveraging the pretrained vision-language model, *i.e.*, CLIP (Radford et al. 2021). When training on task t, given an input image x and a candidate composition set $\mathcal{C}_t$, these methods take a visual encoder VE_t and a language encoder LE_t to obtain the image representation $\boldsymbol{x} = VE_t(x)$ and candidate compositions representations $\{\boldsymbol{c} = LE_t(a,o)|(a,o) \in \mathcal{C}_t\}$, respectively. Both VE_t and LE_t are based on CLIP (Radford et al. 2021), and candidate composition representations are generated using prompt templates like "a photo of {attribute} {object}". Then, the recognition probability of the input image is calculated as

$$p(a,o|x) = \frac{\exp(\cos(VE_t(x), LE_t(a,o))/\tau)}{\sum_{i=1}^{|\mathcal{C}_t|}\exp(\cos(VE_t(x), LE_t(a_i,o_i))/\tau)}, \tag{2}$$

where τ denotes the temperature, $\cos(\cdot,\cdot)$ is the cosine similarity function, and (a,o) is the target composition. On this basis, diverse mechanisms have been proposed, such as learnable prompts (Nayak, Yu, and Bach 2023), cross-modal interaction modules (Huang et al. 2024b), and retrieval augmentation modules (Jing et al. 2024), to further enhance the compositional generalization capability of the model.

Visual Composition Synthesis

We propose a visual synthesizer that learns to synthesize visual representations of past tasks. The visual synthesizer employs the Variational Autoencoder (Kingma 2013) architecture, comprising an encoder VS_t^E and a generator VS_t^G, as illustrated in Figure 4. The encoder VS_t^E, implemented as a simple fully-connected network, encodes the image representation $\boldsymbol{x}$ into a latent code $\boldsymbol{z}$. The generator VS_t^G synthesizes image representations using the latent code $\boldsymbol{z}$ and corresponding attribute and object name (a, o).

Following Wang et al. (2023b), we adapt the language encoder LE_0 (*i.e.*, the language encoder of pretrained CLIP) for the generator VS_G, aiming to enhance the learning efficiency and the quality of the synthesizer by leveraging the aligned vision and language representations learned by the pretrained CLIP. Thus, given the latent code $\boldsymbol{z}$ and the composition (a, o), instead of synthesizing the image representation directly, the generator learns to synthesize instance-specific prompts

$$\boldsymbol{p}(\boldsymbol{z}, a, o) = [\boldsymbol{v}_1 + \boldsymbol{r}, \boldsymbol{v}_2 + \boldsymbol{r}, ..., \boldsymbol{v}_L + \boldsymbol{r}, \boldsymbol{e}_a, \boldsymbol{e}_o], \tag{3}$$

where $\{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_L\}$ are learnable prompts of length L, $\boldsymbol{e}_a$ and $\boldsymbol{e}_o$ are token embedding of the corresponding attribute and object (a, o), $\boldsymbol{r}$ is the local bias obtained from the latent code $\boldsymbol{z}$ through a fully-connected network. Then, the prompts are fed into the language encoder LE_0 to obtain the synthesized image representation $\hat{\boldsymbol{x}}$. Additionally, a lightweight adapter (Gao et al. 2024) is introduced to further bridge the modality gap. Thus, given an image representation $\boldsymbol{x}$, the synthesis process is described as

$$\boldsymbol{z} = VS_t^E(\boldsymbol{x}), \hat{\boldsymbol{x}} = VS_t^G(\boldsymbol{z}, a, o) = LE_0(\boldsymbol{p}). \tag{4}$$

The optimization of the visual synthesizer is achieved via a standard evidence-lower bound

$$\mathcal{L}_{rec} = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2, \mathcal{L}_{kl} = \mathrm{KL}(\boldsymbol{z}, \mathcal{N}(0, 1)), \tag{5}$$

where KL is the Kullback-Leibler divergence.

To ensure semantic consistency, we minimize the difference between the primitive semantic distributions of the synthesized and original visual representations. These distributions are computed via similarity to candidate primitives using simple prompt templates (*e.g.*, "an object looks attribute"). The attribute semantic distribution is denoted as

$$p(a|\boldsymbol{x}) = \frac{\exp(\cos(\boldsymbol{x}, LE_0(a))/\tau)}{\sum_{i=1}^{|\mathcal{A}|} \exp(\cos(\boldsymbol{x}, LE_0(a_i))/\tau)}, \tag{6}$$

and the object semantic distribution $p(o|\boldsymbol{x})$ is conducted similarly. Thus, the semantic loss is calculated as

$$\mathcal{L}_{sem} = \mathrm{KL}(p(a|\hat{\boldsymbol{x}}), p(a|\boldsymbol{x})) + \mathrm{KL}(p(o|\hat{\boldsymbol{x}}), p(o|\boldsymbol{x})). \tag{7}$$

overall optimization objective of the visual synthesizer is

$$\mathcal{L}_{VS} = \mathcal{L}_{rec} + \mathcal{L}_{kl} + \alpha \cdot \mathcal{L}_{sem}, \tag{8}$$

where α is the hyper-parameter that balances the objective of element-wise reconstruction and the semantic consistency.

Linguistic Primitive Distillation

Learning semantically aligned primitives improves compositional generalization (Li et al. 2023). In order to encourage the model to learn primitive semantic representations applicable to all previously seen compositions, rather than overfitting to the compositions of the current task, the framework incorporates a distillation-based mechanism. Specifically, after completing the training on task $t-1$, the language encoder LE_{t-1} is duplicated and frozen. During the training of task t, for a given image representation $\boldsymbol{x}$ (whether it is derived from an image of the current task encoded by the vision encoder VE_t or synthesized by the vision synthesizer VS), the predicted logits $\boldsymbol{l}_t$ using the current LE_t for all compositions from past tasks are computed as

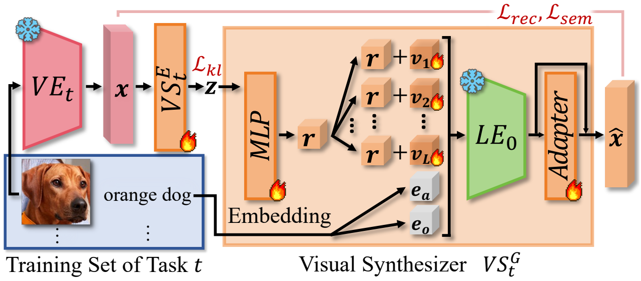


Figure 4: The architecture and training of the visual synthesizer. The visual synthesizer comprises an encoder VS_t^E and a generator VS_t^G. It is trained to synthesize visual representations conditioned on the given composition name.

$$\boldsymbol{l}_t = \{\cos(\boldsymbol{x}, LE_t(a_i, o_i))/\tau \mid (a_i, o_i) \in \bigcup_{j=1}^{t} \mathcal{C}_j\}. \tag{9}$$

Similarly, the logits $\boldsymbol{l}_{t-1}$ can be computed using the duplicated language encoder LE_{t-1}. The distillation loss

$$\mathcal{L}_{kd} = ||\boldsymbol{l}_t - \boldsymbol{l}_{t-1}||_2 \tag{10}$$

encourages consistent predictions for past compositions, ensuring that updates for new tasks retain previously learned aligned primitive representations.

Notably, benefiting from our visual synthesizer, the visual representations can correspond to any composition from task 0 to task t. This enables meaningful linguistic primitive distillation across all past tasks, highlighting the difference from vanilla knowledge distillation. Such distillation facilitates the learning of unified primitive semantic representations that generalize across tasks. Furthermore, these unified linguistic primitives ensure the synthesized visual features remain semantically aligned throughout different training stages. In other words, the visual composition synthesis and the linguistic primitive distillation can promote each other.

Optimization

To incorporate a CZSL model into the proposed framework, we begin by duplicating the language encoder of the model and utilizing it as part of the visual synthesizer. Subsequently, the model is initially trained on task 0 using the method-specific loss $\mathcal{L}_{ms}$, which depends on the selected CZSL model. After completing training on task $t-1$, we duplicate and freeze the language encoder LE_{t-1} of the CZSL model. Then, the visual synthesizer is optimized on the training set of task $t-1$ with the objective $\mathcal{L}_{VS}$. The encoder of the synthesizer VS_t^E is randomly initialized per task, while VS_t^G is retained and updated across tasks. When training begins on task t, the synthesizer is employed to synthesize visual representations of past compositions, given

Baseline	CSP (Nayak, Yu, and Bach 2023) (CLIP ViT-L/14)						Troika (Huang et al. 2024b) (CLIP ViT-B/16)					
Method	*U*	*S*	*AUC*	*mU*	*fS*(↓)	*mAUC*	*U*	*S*	*AUC*	*mU*	*fS*(↓)	*mAUC*
Zero-Shot	46.10	30.63	11.15	-	-	-	41.02	28.15	8.95	-	-	-
Joint	49.61	46.51	19.24	-	-	-	47.47	44.12	17.35	-	-	-
Vanilla	39.78	30.55	9.76	41.41	15.22	13.59	36.84	26.98	7.54	38.40	26.69	10.68
SI (Zenke, Poole, and Ganguli 2017)	46.88	37.40	14.33	45.96	8.30	16.45	36.69	27.52	7.54	38.42	26.50	10.70
EWC (Chaudhry et al. 2019)	48.35	39.62	15.70	46.81	7.21	17.30	41.39	29.71	9.41	40.70	21.62	11.80
A-GEM (Chaudhry et al. 2018)	45.71	36.22	13.46	45.71	10.17	16.32	39.54	28.87	8.67	39.71	24.31	11.34
DER++ (Buzzega et al. 2020)	45.73	37.31	13.92	43.28	6.19	14.72	39.39	30.50	9.20	38.74	**8.46**	10.59
L2P (Wang et al. 2022)	38.92	30.50	9.41	40.39	15.25	13.17	37.42	27.65	7.95	37.97	23.72	10.58
AttriCLIP (Wang et al. 2023a)	39.12	30.67	9.65	42.15	15.69	14.17	37.27	26.93	7.64	37.36	27.69	10.05
Ours	**49.11**	**41.43**	**16.77**	**47.23**	**3.28**	**17.99**	**42.60**	**32.40**	**10.69**	**41.59**	10.72	**12.26**

Table 1: Comparison with state-of-the-art continual learning methods on CZSL models on the MIT-States-CompIL benchmark.

the composition name (a', o') and the noise $\boldsymbol{z}$ sampled from the prior distribution $\mathcal{N}(0, 1)$. These synthesized representations $\hat{\boldsymbol{x}}$ are combined with the current task's training samples to train the CZSL model using the method-specific loss $\mathcal{L}_{ms}$. The overall optimization objective of the model is

$$\mathcal{L} = \mathcal{L}_{ms} + \beta \cdot \mathcal{L}_{kd}, \tag{11}$$

where the hyper-parameter β balances the stability-plasticity trade-off in primitive representation learning.

Experiments

Experiment Setting

Baseline Models. We apply the proposed framework to two CZSL models, CSP (Nayak, Yu, and Bach 2023) and Troika (Huang et al. 2024b). CSP adopts the CLIP model with learnable prompts in the language encoder, similar to the common paradigms in class-incremental learning, where the visual encoder is frozen, and only the classification head is trained. In contrast, Troika employs a more complex network architecture with additional trainable parameters, *e.g.*, the cross-modal interaction modules, reflecting the cutting-edge advancements in CZSL. We implement the two models with pre-trained CLIP ViT-L/14 and ViT-B/16 to validate the proposed framework across different model scales. More results are in the **supplementary material**. Notably, for models that require patch features during training, *e.g.*, Troika, we repeat the synthesized representations to match the size of the patch features and use them for joint training.

Implementation Details. The learning rate is set to 1e-4 for all experiments. We halve the training epochs in the original paper to prevent overfitting on individual tasks: CSP is trained for 10 epochs per task on MIT-States-CompIL and C-GQA-CompIL, while Troika is trained for 5 and 7 epochs, respectively. The prompt length L is set to 3. The hyper-parameter α and β are set to 0.1 and 0.3. All experiments are run three times under different random seeds, and the average results are reported.

Comparison Methods. Since CompIL is a newly proposed setting, there does not exist any prior works that can be used for comparison directly. Therefore, we reimplement and adapt three types of continual learning methods to integrate them with CZSL models for fair comparison. These include regularization-based approaches SI (Zenke, Poole, and Ganguli 2017) and EWC (Chaudhry et al. 2019), rehearsal-based methods A-GEM (Chaudhry et al. 2018), and DER++ (Buzzega et al. 2020), and the recent prompt-based approaches L2P (Wang et al. 2022) and AttriCLIP (Wang et al. 2023a). The memory size of rehearsal-based methods is set to 5% of the total training samples. We conduct hyperparameter search for these methods to ensure fair comparison, and provide additional results (*e.g.*, buffer sizes) in the **supplementary material**.

Method	Task Number						
	0	1	2	3	4	5	Avg
CSP	10.2	**6.2**	8.3	1.9	4.2	3.2	5.6
CSP + Ours	**10.2**	6.0	**8.4**	**2.4**	**6.3**	**3.8**	**6.2**
Troika	16.4	12.1	**23.4**	10.9	18.9	10.7	15.4
Troika + Ours	**17.2**	**13.3**	22.5	**14.0**	**22.2**	**15.6**	**17.5**

Table 2: Results of the proposed framework on the split of the C-GQA dataset introduced in CCZSL. We reported the AUC in each task and the average AUC across all tasks.

Results on Composition-Incremental Setting

The experimental results on MIT-States-CompIL and C-GQA-CompIL are listed in Table 1 and 3, where "Zero-Shot" refers to predictions from the pretrained CLIP model, "Joint" (upper bound) represents training all tasks jointly, and "Vanilla" (lower bound) represents simply performing gradient update task by task. We observe that: (1) Our framework consistently enhances two CZSL models on both seen and unseen compositions, achieving state-of-the-art overall performance as measured by *AUC* and *mAUC*. (2) Our framework significantly improves compositional generalization (*U* and *mU*), surpassing the second-best method by an average of 2%, while effectively mitigating forgetting (*S* and *fS*). (3) Existing methods struggle to continually improve compositional generalization, mostly performing worse than or similar to Zero-Shot after the learning process (see *U* metrics). Besides, prompt-based methods (L2P and AttriCLIP) fail in CompIL. We speculate this arises from conflicts be-

Baseline / Method	CSP (CLIP ViT-L/14)						Troika (CLIP ViT-B/16)					
	U	*S*	*AUC*	*mU*	*fS*(↓)	*mAUC*	*U*	*S*	*AUC*	*mU*	*fS*(↓)	*mAUC*
Zero-Shot	25.18	7.39	1.41	-	-	-	23.95	6.75	1.16	-	-	-
Joint	27.80	28.88	6.35	-	-	-	34.18	42.87	12.63	-	-	-
Vanilla	15.39	16.62	1.89	19.28	8.86	3.86	22.73	27.31	4.95	24.97	22.54	9.70
SI (Zenke, Poole, and Ganguli 2017)	18.27	18.61	2.53	20.12	6.32	4.64	22.38	28.23	5.00	24.39	21.46	9.56
EWC (Chaudhry et al. 2019)	24.48	**21.42**	3.91	24.25	**3.69**	5.10	26.05	29.26	6.33	27.17	21.50	11.17
A-GEM (Chaudhry et al. 2018)	17.66	18.69	2.53	20.45	7.06	4.36	23.51	26.83	5.01	23.90	23.52	9.11
DER++ (Buzzega et al. 2020)	22.64	21.24	3.59	22.40	4.25	4.64	21.33	32.85	5.84	20.00	**8.32**	8.45
L2P (Wang et al. 2022)	18.27	19.02	2.62	17.22	9.29	3.78	21.24	28.93	4.72	23.01	20.77	9.65
AttriCLIP (Wang et al. 2023a)	19.76	19.90	2.99	21.80	12.52	5.05	23.08	26.45	4.90	23.62	24.26	10.00
Ours	**27.54**	20.81	**4.60**	**26.71**	5.67	**5.42**	**29.55**	**34.57**	**8.40**	**27.88**	14.61	**11.98**

Table 3: Comparison with state-of-the-art continual learning methods on CZSL models on the C-GQA-CompIL benchmark.

	VS	$\mathcal{L}_{sem}$	$\mathcal{L}_{kd}$	*mU*	*fS*(↓)	*mAUC*
1				41.41	15.22	13.59
2	✓			46.64	5.63	17.33
3	✓	✓		46.92	4.83	17.43
4			✓	46.29	7.68	16.82
5	✓	✓	✓	**47.23**	**3.28**	**17.99**

Table 4: Results of different variants of the proposed framework. We use CSP as the baseline model (first row). VS denotes the visual synthesizer module.

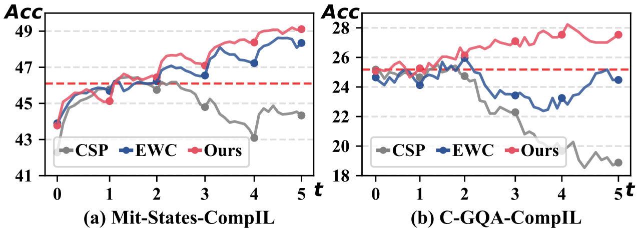
tween prompt learning and the updates of other learnable parameters in the baseline models.

Results on Primitive-Incremental Setting

Unlike our CompIL setting that focuses on composition-incremental learning within a fixed primitive set, CCZSL (Zhang, Feng, and Yuan 2024) requires models to continuously learn from compositions that include unseen primitives, thereby expanding the size of the learned primitive set over time. We conduct experiments on the split of the C-GQA (Mancini et al. 2022) dataset introduced in CCZSL, and the experimental results are shown in Table 2. We observe that our framework improves CSP and Troika across different sessions on the CCZSL split of the C-GQA dataset, with 0.6% and 2.1% absolute gains in the average AUC. Such observations suggest that our framework effectively improves CZSL models on primitive-incremental settings, though it is not explicitly designed for that.

Ablation Studies

To validate the effectiveness of each component, we conduct ablation studies on the MIT-States-CompIL benchmark using CSP as the baseline, with results shown in Table 4. Adding the visual synthesizer (VS) (row 2) yields significant gains over the baseline (row 1). Introducing semantic loss ($\mathcal{L}_{sem}$) in row 3 further enhances semantic consistency, improving performance across all metrics. Linguistic primitive distillation ($\mathcal{L}_{kd}$) also brings improvements, though not as much as the full model. Combining all components achieves the best overall results, confirming that each module contributes effectively and complementarily.



(a) Mit-States-CompIL

(b) C-GQA-CompIL

Figure 5: Compositional generalization capability of CSP, equipped with state-of-the-art methods, after training on different tasks in the compositional incremental learning process.The red line indicates CLIP's zero-shot performance.

Quantitative Studies

The accuracy of different continual learning methods on the final task of CompIL throughout the compositional incremental learning process is illustrated in Figure 5, which reflects the variation in the model's compositional generalization capability as learning new compositions. We observe that, compared to other methods, our framework significantly enhances the model's compositional generalization capability as it continually learns new compositions.

Conclusion

In this paper, we have presented a practical and challenging setting for compositional generalization, termed CompIL. The setting challenges models to continually learn new compositions, aiming to improve their compositional generalization capability progressively. We have developed a pipeline to construct CompIL benchmarks, resulting in MIT-States-CompIL and C-GQA-CompIL for quantitative evaluation. Moreover, we have proposed a pseudo-replay framework that can mitigate composition knowledge forgetting and primitive representation drift by leveraging a visual synthesizer and a linguistic primitive distillation mechanism. Extensive experiments on two CZSL models across the proposed benchmarks demonstrate its effectiveness.

Acknowledgments

This work was supported by the Shenzhen Science and Technology Program under Grant No. JCYJ20241202130548062, Natural Science Foundation of China (NSFC) under Grants No. 62172041 and No. 62176021, Natural Science Foundation of Shenzhen under Grant No. JCYJ20230807142703006, and the Key Research Platforms and Projects of the Guangdong Provincial Department of Education under Grant No.2023ZDZX1034.

References

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33: 15920–15930.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2018. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P.; Torr, P.; and Ranzato, M. 2019. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*.

Dhar, P.; Singh, R. V.; Peng, K.-C.; Wu, Z.; and Chellappa, R. 2019. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5138–5146.

Frascaroli, E.; Panariello, A.; Buzzega, P.; Bonicelli, L.; Porrello, A.; and Calderara, S. 2024. Clip with generative latent replay: a strong baseline for incremental learning. *arXiv preprint arXiv:2407.15793*.

Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; and Bouchachia, A. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4): 1–37.

Gao, P.; Geng, S.; Zhang, R.; Ma, T.; Fang, R.; Zhang, Y.; Li, H.; and Qiao, Y. 2024. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2): 581–595.

Gurobi Optimization, LLC. 2024. Gurobi Optimizer Reference Manual.

Hu, Z.; Li, Y.; Lyu, J.; Gao, D.; and Vasconcelos, N. 2023. Dense network expansion for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11858–11867.

Huang, C.; Qin, P.; Lei, W.; and Lv, J. 2024a. Towards Equipping Transformer with the Ability of Systematic Compositionality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18289–18297.

Huang, S.; Gong, B.; Feng, Y.; Zhang, M.; Lv, Y.; and Wang, D. 2024b. Troika: Multi-path cross-modal traction for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24005–24014.

Isola, P.; Lim, J. J.; and Adelson, E. H. 2015. Discovering states and transformations in image collections. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1383–1391.

Jin, X.; Du, J.; Sadhu, A.; Nevatia, R.; and Ren, X. 2020. Visually Grounded Continual Learning of Compositional Phrases. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018–2029. Online: Association for Computational Linguistics.

Jing, C.; Li, Y.; Chen, H.; and Shen, C. 2024. Retrieval-Augmented Primitive Representations for Compositional Zero-Shot Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2652–2660.

Kingma, D. P. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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.

Li, C.; Jing, C.; Li, Z.; Zhai, M.; Wu, Y.; and Jia, Y. 2024a. In-context compositional generalization for large vision-language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 17954–17966.

Li, C.; Li, Z.; Jing, C.; Jia, Y.; and Wu, Y. 2023. Exploring the effect of primitives for compositional generalization in vision-and-language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19092–19101.

Li, C.; Li, Z.; Jing, C.; Liu, S.; Shao, W.; Wu, Y.; Luo, P.; Qiao, Y.; and Zhang, K. 2024b. Searchlvlms: A plug-and-play framework for augmenting large vision-language models by searching up-to-date internet knowledge. *Advances in Neural Information Processing Systems*, 37: 64582–64603.

Li, C.; Li, Z.; Jing, C.; Wu, Y.; Zhai, M.; and Jia, Y. 2024c. Compositional Substitutivity of Visual Reasoning for Visual Question Answering. In *European Conference on Computer Vision*, 143–160. Springer.

Li, X.; Zhou, Y.; Wu, T.; Socher, R.; and Xiong, C. 2019. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International conference on machine learning*, 3925–3934. PMLR.

Li, Y.; Li, Q.; Wang, H.; Li, R.; Zhong, W.; and Zhang, G. 2024d. Towards Efficient Replay in Federated Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12820–12829.

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

Liao, W.; Wei, Y.; Jiang, M.; Zhang, Q.; and Ishibuchi, H. 2024. Does continual learning meet compositionality? new benchmarks and an evaluation framework. *Advances in Neural Information Processing Systems*, 36.

Lin, D.; et al. 1998. An information-theoretic definition of similarity. In *Icml*, volume 98, 296–304.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2): 129–137.

Lu, X.; Guo, S.; Liu, Z.; and Guo, J. 2023. Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23560–23569.

Ma, Z.; Hong, J.; Gul, M. O.; Gandhi, M.; Gao, I.; and Krishna, R. 2023. Crepe: Can vision-language foundation models reason compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10910–10921.

Mancini, M.; Naeem, M. F.; Xian, Y.; and Akata, Z. 2022. Learning graph embeddings for open world compositional zero-shot learning. *IEEE Transactions on pattern analysis and machine intelligence*, 46(3): 1545–1560.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11): 39–41.

Nayak, N. V.; Yu, P.; and Bach, S. H. 2023. Learning to Compose Soft Prompts for Compositional Zero-Shot Learning. In *International Conference on Learning Representations*.

Purushwalkam, S.; Nickel, M.; Gupta, A.; and Ranzato, M. 2019. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3593–3602.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Ray, A.; Radenovic, F.; Dubey, A.; Plummer, B.; Krishna, R.; and Saenko, K. 2024. Cola: A benchmark for compositional text-to-image retrieval. *Advances in Neural Information Processing Systems*, 36.

Saini, N.; Pham, K.; and Shrivastava, A. 2024. Beyond Seen Primitive Concepts and Attribute-Object Compositional Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14466–14476.

Thengane, V.; Khan, S.; Hayat, M.; and Khan, F. 2022. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*.

Wang, R.; Duan, X.; Kang, G.; Liu, J.; Lin, S.; Xu, S.; Lü, J.; and Zhang, B. 2023a. Attriclip: A non-incremental learner for incremental knowledge learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3654–3663.

Wang, Z.; Liang, J.; He, R.; Xu, N.; Wang, Z.; and Tan, T. 2023b. Improving zero-shot generalization for clip with synthesized prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3032–3042.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 139–149.

Xu, S.; Wang, S.; Hu, X.; Lin, Y.; Du, B.; and Wu, Y. 2024. MAC: A Benchmark for Multiple Attributes Compositional Zero-Shot Learning. *arXiv preprint arXiv:2406.12757*.

Yao, H.; Choi, C.; Cao, B.; Lee, Y.; Koh, P. W. W.; and Finn, C. 2022. Wild-time: A benchmark of in-the-wild distribution shift over time. *Advances in Neural Information Processing Systems*, 35: 10309–10324.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, 3987–3995. PMLR.

Zhang, Y.; Feng, S.; and Yuan, J. 2024. Continual Compositional Zero-Shot Learning. In Larson, K., ed., *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 1724–1732. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zhang, Y.; Qiu, B.; Jia, Q.; Liu, Y.; and He, R. 2024. Not Just Object, But State: Compositional Incremental Learning without Forgetting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.