Anticipating Future Object Compositions Without Forgetting

Zahran, Youssef; Burghouts, Gertjan; B. Eisma, Yke

DOI
10.1007/978-3-031-78113-1_18

Publication date
2025

Document Version
Final published version

Published in
Pattern Recognition

Anticipating Future Object Compositions Without Forgetting

Youssef Zahran[1,2(✉)], Gertjan Burghouts[2], and Yke B. Eisma[1]

[1] Department of Cognitive Robotics, Delft University of Technology, Delft, The Netherlands
y.zahran09@gmail.com, Y.B.Eisma@tudelft.nl

[2] Intelligent Imaging Department, TNO, The Hague, The Netherlands
gertjan.burghouts@tno.nl

Abstract. Despite the significant advancements in computer vision models, their ability to generalize to novel object-attribute compositions remains limited. Existing methods for Compositional Zero-Shot Learning (CZSL) mainly focus on image classification. This paper aims to enhance CZSL in object detection without forgetting prior learned knowledge. We use Grounding DINO and incorporate Compositional Soft Prompting (CSP) into it and extend it with Compositional Anticipation. We achieve a 70.5% improvement over CSP on the harmonic mean (HM) between seen and unseen compositions on the CLEVR dataset. Furthermore, we introduce Contrastive Prompt Tuning to incrementally address model confusion between similar compositions. We demonstrate the effectiveness of this method and achieve an increase of 14.5% in HM across the pretrain, increment, and unseen sets. Collectively, these methods provide a framework for learning various compositions with limited data, as well as improving the performance of underperforming compositions when additional data becomes available.

Keywords: compositional zero-shot learning · prompt tuning · incremental learning

1 Introduction

Although humans have never seen a blue apple, they can easily picture it. This is due to the inherent human ability to generalize to novel concepts by combining the known entity “apple” with the color “blue”. However, do computer vision models possess this capability? This question has motivated the development of Compositional Zero-Shot Learning (CZSL) [5,21,22,26]. In CZSL, the goal is to recognize unseen object-attribute combinations, referred to as compositions, based on the compositions seen during training. For this, models should understand the attributes and objects that compose these compositions to generalize to all possible compositions.

Vision Language Models (VLMs), pretrained on large-scale image-text pairs, are promising for CZSL due to their ability to understand the relationship between the visual content and the textual description [16,22,26]. For object

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15330, pp. 265–279, 2025.
https://doi.org/10.1007/978-3-031-78113-1_18

detection, VLMs such as Grounding DINO [10] and GLIP [7] learn to associate regions of text with regions of images by pulling the embeddings of paired image regions and text descriptions close while pushing others away [25]. These models perform cross-modality fusion throughout the whole architecture, which makes the textual features image-aware and the visual features text-aware. Liu et al. [10] argue that VLMs benefit from frequent cross-modality fusion, making Grounding DINO superior to GLIP. Therefore, throughout this paper, we will solely focus on Grounding DINO.

Unfortunately, these models tend to be biased towards object categories rather than attributes, which makes it suffer from feature misalignment when used directly for attribute recognition [3]. Fine-tuning VLMs can solve this but often leads to catastrophic forgetting of prior knowledge [28], thereby compromising their generalization ability. To address this, we explore how to fine-tune VLMs to perform well in CZSL without forgetting any prior knowledge. Nayak et al. [15] introduced Compositional Soft Prompting (CSP), which combats catastrophic forgetting by adding auxiliary tokens for all words in a given dataset and training only these tokens. This approach preserves the model's original embeddings, allowing it to retain and revert to its initial knowledge when necessary, unlike full fine-tuning, which alters the model's parameters. CSP improves model performance in CZSL for image classification. We incorporate CSP in Grounding DINO, to leverage it for object detection.

We consider CSP as a baseline and improve it for CZSL by introducing Compositional Anticipation (CA), which recognizes that additional compositions may exist beyond those present during training. In this context, the term "anticipation" does not refer to actively predicting new compositions. Instead, it involves enhancing the model's ability to handle potential new compositions by adjusting how it processes partially correct predictions through Compositional Smoothing and by guiding the model to disentangle attributes from objects via Compositional Independence. Compositional Smoothing prepares the model for novel compositions by assigning soft labels when predictions are partially correct, e.g., the object is correct but the attribute is different. This approach deviates from conventional Label Smoothing [20], which assigns soft labels to all classes. Compositional Independence disentangles objects from attributes through Separation and Decorrelation. Separation introduces a separation loss to maximize the distinction between object and attribute classes by applying intra-class separation within objects and attributes and an inter-class separation between objects and attributes. Decorrelation minimizes the correlation between objects and attributes to reduce dependency between the two.

For incremental learning on newly added compositions, we use prior knowledge to address specific mistakes related to confusion between similar compositions. Inspired by recent developments in prompt tuning [17,29,30], we introduce a novel method called Contrastive Prompt Tuning, specifically tailored for object attributes. Contrastive Prompt Tuning addresses cases where the model confuses similar compositions, such as mistaking a blue apple for a red apple, by adding a trainable prompt in front of the confused class: "is not red apple but is blue apple". This approach utilizes our prior knowledge to harness the ability of a VLM to exploit language.

In summary, our main contributions are:

1. We incorporate CSP [15] into Grounding DINO [10] and extend it with Compositional Anticipation. Compositional Anticipation consists of:
   – Compositional Smoothing, which assigns soft labels when predictions are partially correct.
   – Compositional Independence, which disentangles objects from attributes.
2. We develop Contrastive Prompt Tuning, a method that adds a learnable prompt for compositions that are confused with each other during training. This technique harnesses the power of language and our understanding of the model to improve performance beyond simply training with additional data.

2 Related Work

In this section, we review literature related to our work. We cover Compositional Zero-Shot Learning (CZSL), Prompt Tuning in Vision-Language Models (VLMs), and Class Incremental Learning (CIL). Our research focuses on improving the CZSL capabilities of Grounding DINO [10], a VLM designed for object detection, by utilizing prompt tuning. Additionally, we address underperforming compositions in a class-incremental manner to further improve model performance.

Compositional Zero-Shot Learning. The main objective of CZSL is to recognize unseen compositions from the compositions encountered during training. In CZSL, individual objects and attributes are referred to as primitives. Misra et al. [13] use a limited set of compositions to learn linear classifiers for each primitive. Then, they learn a transformation network that takes these classifiers as input and composes them to produce a classifier for their combination. Since then, multiple works [8,12,14,19] have been proposed to tackle the CZSL task.

Recent works focus on adapting pretrained VLMs for CZSL by fine-tuning primitive tokens. While CSP [15] only trains these tokens, others [11,21] also introduce prompt disentangled tuning. This technique addresses entanglement, where optimizing one primitive's embedding affects another. Prompt disentangled tuning divides the process into three phases with different prompts: one for the entire composition, one for the attribute, and one for the object. This ensures attributes and objects learn their optimal parameters independently.

While [11,21] improve upon [15] with an average performance increasement of 1.7% and 2.3%, respectively, the gains are marginal relative to the increased complexity. Our work is closely related to [11,15] as we adapt CSP for object detection and address entanglement through Compositional Independence.

Prompt Tuning in VLMs. Ever since CLIP [16] demonstrated that prompt templates such as "a photo of a [CLASS]" improve the results of VLMs compared to using only the classname, several other works [17,24,29,30] have been introduced to replace the hand-crafted prompt with learnable soft prompts. CoOP [30] introduces soft prompts that are shared across all classes, resulting in prompts like $[v_1], [v_2], \dots [v_M]$ for all images. CoCoOp [29] improves upon this by proposing soft prompts that are image-conditioned, generating prompts such as $[v_1(x)], [v_2(x)], \dots [v_M(x)]$ for each image x. Building upon these advancements, Rao et al. [17] use contextual information from the image to prompt the language model.

Our work is closely related to these works but is unique in its focus on improving the performance of confused compositions using learnable prompts that are initialized based on our knowledge of the model's errors.

Class Incremental Learning. Class-Incremental Learning (CIL) refers to learning new classes while retaining previously learned classes [27,28]. In typical CIL scenarios, learning occurs through a sequence of training tasks, each of which introduce new classes without any overlap of the classes from previous tasks. The main challenge is avoiding catastrophic forgetting, where learning new classes leads to a loss of knowledge from previous tasks. Our approach bears resemblance to Blurry CIL [1,2], where former classes can be revisited during training. Similarly, we train incrementally with underperforming compositions while allowing former compositions to be revisited.

3 Method

3.1 Problem Definition

Compositional Zero-Shot Learning. We follow [21,22,26] and formalize the CZSL task as follows. Let $\mathcal{A}$ denote the set of attributes, and $\mathcal{O}$ the set of objects, and $\mathcal{C} = \mathcal{A} \times \mathcal{O}$ the set of all compositions. $\mathcal{T} = \{(x_j, c_j)\}_{j=1}^{N}$ denotes the train set where $x_j \in \mathcal{X}$ is a sample in the input (image) space $\mathcal{X}$ and $c_j \in \mathcal{C}_s$ is a composition in the subset $\mathcal{C}_s \subseteq \mathcal{C}$. The seen set $\mathcal{C}_s \subseteq \mathcal{C}$ consists of all compositions encountered during training, whereas the unseen set $\mathcal{C}_u \subseteq \mathcal{C}$ consists of compositions not seen during training. Let $\mathcal{C}_s$ and $\mathcal{C}_u$ be two sets such that $\mathcal{C}_s \cap \mathcal{C}_u = \emptyset$. While $\mathcal{C}_s$ and $\mathcal{C}_u$ are disjoint, the objects $\mathcal{O}_u$ and attributes $\mathcal{A}_u$ are defined such that $\mathcal{O}_u \subseteq \mathcal{O}_s$ and $\mathcal{A}_u \subseteq \mathcal{A}_s$

Catastrophic Forgetting. VLMs, such as Grounding DINO [10], are known for their ability to generalize well across diverse tasks due to the extensive and varied data used during pre-training. However, fine-tuning these models on a new dataset often compromises their generalization capability, as the rich features learned during pre-training are replaced by features specific to the new dataset. This can lead to catastrophic forgetting, where the model's performance on previously learned tasks significantly deteriorates. In the context of

CZSL with VLMs, catastrophic forgetting is particularly problematic. While the model may perform well on the specific compositions present in the new dataset it was fine-tuned on, it risks becoming overly specialized. This specialization may result in a model that loses its ability to generalize to other compositions, objects, or concepts, and instead becomes exceptionally good at predicting the compositions seen during training. Such a limitation is especially undesirable for open-set object detectors like Grounding DINO, which are meant to recognize a wide range of concepts.

3.2 Incremental CZSL

In practical settings, models often encounter new data or need to improve performance on underperforming compositions after the initial training phase. To address this, we introduce an increment set $\mathcal{C}_i \subseteq \mathcal{C}$ to CZSL. Let $\mathcal{C}_p$ be the set used for the initial fine-tuning of the model, with $\mathcal{C}_p = \mathcal{C}_s$. After introducing $\mathcal{C}_i$, the set of seen compositions becomes $\mathcal{C}_s = \mathcal{C}_p \cup \mathcal{C}_i$. The increment set $\mathcal{C}_i$ consists of compositions introduced after the initial fine-tuning to improve performance on underperforming compositions. Improving these underperforming compositions with additional compositions is challenging because $\mathcal{C}_p$ is designed to cover $\mathcal{A}$ and $\mathcal{O}$ with the minimum number of compositions. Extending $\mathcal{C}_s$ with $\mathcal{C}_i$ makes the attributes and objects in $\mathcal{C}_i$ overrepresented in $\mathcal{C}_s$, which can bias the model towards these attributes and objects. In this paper, we focus solely on improving performance on compositions $c_j \in \mathcal{C}$ without extending the attribute set $\mathcal{A}$ or the object set $\mathcal{O}$.

3.3 Compositional Soft Prompting

To prevent catastrophic forgetting in Grounding DINO [10], we follow CSP [15] and modify it for object detection. Objects and attributes that form compositions are treated as learnable tokens within the VLMs vocabulary. Each attribute $a_j \in \mathcal{A}$ and each object $o_j \in \mathcal{O}$ is represented as an auxiliary token t_{a_j} and t_{o_j} respectively, where $t_{a_j}, t_{o_j} \in \mathbb{R}^d$, with d being the dimension of the vocabulary embedding. During training, only these auxiliary tokens are tuned, resulting in $(|\mathcal{A}| + |\mathcal{O}|) \times d$ learnable parameters.

To illustrate, CSP creates auxiliary tokens for each attribute and object such as t_{blue} for the attribute "blue" and t_{apple} for the object "apple". These tokens are adjusted during training while the rest of the weights, such as those of the encoder and decoder in Grounding DINO, remain unchanged. By doing this, CSP prevents catastrophic forgetting and preserves the pretrained weights of the model.

3.4 Compositional Smoothing

To combat bias during training for $\mathcal{C}_s$, where the model becomes overly confident with the seen classes, we assign soft labels rather than hard labels (0 and 1) in

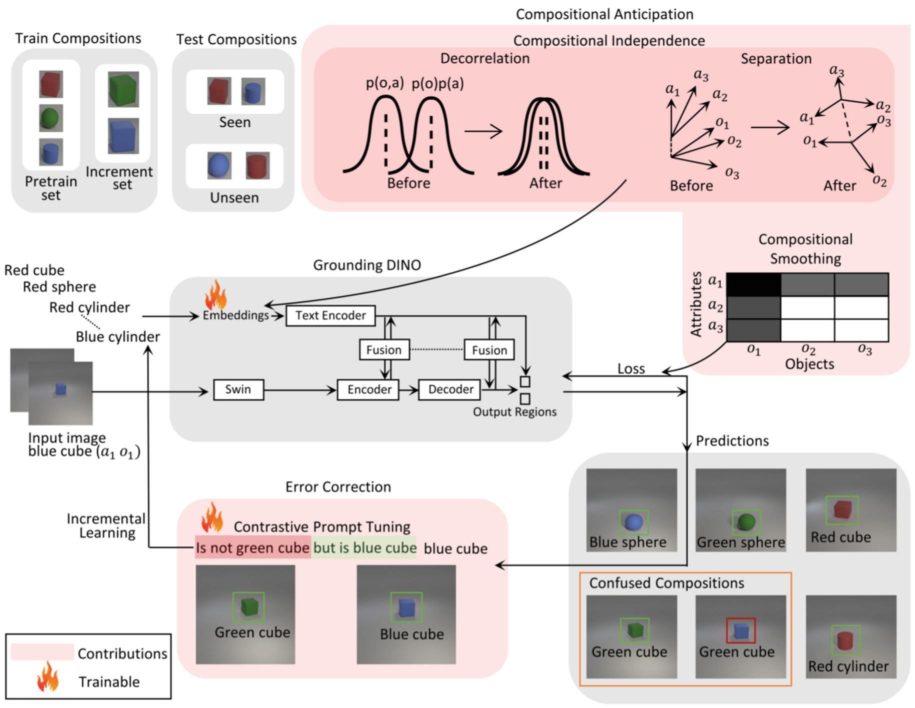


Fig. 1. Our method anticipates unseen, future object-attribute compositions through Compositional Independence and Compositional Smoothing. Forgetting is mitigated by creating auxiliary tokens for the language embeddings and refining only these tokens. Errors in compositions are incrementally corrected using Contrastive Prompt Tuning, which contrasts confused compositions.

the classification loss. This is referred to as Label Smoothing [20], and it prevents the model from becoming overly confident in its predictions, thereby improving its generalization capability. Conventional Label Smoothing adjusts the target labels by distributing a small portion of the probability mass to all other labels. For a given true label y in a classification problem with k classes, the smoothed label y_{smooth} is defined as:

$$y_{\text{smooth}} = (1 - \epsilon)y + \frac{\epsilon}{k}, \tag{1}$$

where ϵ is the smoothing parameter, and the term $\frac{\epsilon}{k}$ distributes the smoothing equally among all classes.

We deviate from conventional Label Smoothing [20] and assign soft labels based on the correctness of the object, attribute, or the entire composition. We refer to this as Compositional Smoothing. Let $p_{\mathcal{O}}$, $p_{\mathcal{A}}$, and $p_{\mathcal{C}}$ represent the probabilities for object, attribute, and overall composition predictions, respectively. For a given true composition c_t composed of object o_t and attribute a_t, and predicted composition c_p composed of object o_p and attribute a_p, the smoothed label $y_{\text{o,a}}$ is defined as:

$$y_{\text{o,a}} = \begin{cases} p_{\mathcal{O}} & \text{if } o_p = o_t \wedge a_p \neq a_t, \\ p_{\mathcal{A}} & \text{if } a_p = a_t \wedge o_p \neq o_t, \\ p_{\mathcal{C}} & \text{if } a_p = a_t \wedge o_p = o_t, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Compositional Smoothing ensures that there is a difference between having partial correctness and no correctness in the prediction, guiding the model to learn what the compositions are composed of rather than learning the compositions themselves. This, in turn, should lead to better performance on $\mathcal{C}_u$.

Figure 1 (top right) illustrates Compositional Smoothing. For a given ground truth label (a_1, o_1), predictions where both the attribute and object are correct are shown in black, and the smoothed label becomes $p_{\mathcal{C}}$. Partial correctness is depicted in gray, with the smoothed label being either $p_{\mathcal{O}}$ or $p_{\mathcal{A}}$. When both the object and attribute are completely wrong, no smoothing is applied.

3.5 Compositional Independence

In CZSL, it is important to disentangle objects from attributes and have clear distinctions withing each category. For example, a cube and a cylinder should be easily distinguishable to prevent confusion. Additionally, colors should be distinguished from specific objects, such as cubes, to ensure their independence. This prevents similar attributes or objects to be confused with each other and helps the model treat attributes and objects as distinct concepts.

We achieve this independence through two components: Separation and Decorrelation. Separation enforces orthogonality within the embeddings of objects and attributes and maximizes the distance between their mean embeddings. Decorrelation minimizes the correlation between the embeddings of objects and attributes. This is achieved using the Hilbert-Schmidt Independence Criterion (HSIC) [4], a kernel statistical test commonly used to measure independence between two random variables, which proved to be effective for CZSL image classification [18] and is leveraged here for object detection.

Separation. To help the model differentiate between similar attributes or objects, we introduce an orthogonality loss. We achieve orthogonality within the groups of attributes and objects by minimizing the average absolute similarity between the normalized embeddings within each group:

$$\mathcal{L}_{\text{orth}}(\mathbf{E}) = \frac{1}{|\mathbf{E}|^2 - |\mathbf{E}|} \sum_{i=1}^{|\mathbf{E}|} \sum_{\substack{j=1 \\ j \neq i}}^{|\mathbf{E}|} |\mathbf{e}_i \cdot \mathbf{e}_j| \tag{3}$$

where $\mathbf{E}$ is the set of normalized embeddings, and $\mathbf{e}_i$ and $\mathbf{e}_j$ are embeddings within this set. The summation $\sum_{\substack{j=1 \\ j \neq i}}$ ignores self-similarity, and $\frac{1}{|\mathbf{E}|^2 - |\mathbf{E}|}$ ensures that self-similar terms are excluded during normalization. This orthogonality loss is applied to both the attributes and objects:

$$\mathcal{L}_{\mathcal{A}} = \mathcal{L}_{\text{orth}}(\mathbf{E}_{\mathcal{A}}) \tag{4}$$

$$\mathcal{L}_{\mathcal{O}} = \mathcal{L}_{\text{orth}}(\mathbf{E}_{\mathcal{O}}) \tag{5}$$

where $\mathbf{E}_{\mathcal{A}}$ and $\mathbf{E}_{\mathcal{O}}$ represent the sets of normalized embeddings for the attributes and objects, respectively.

Additionally, to enforce a clear distinction between attributes and objects, we ensure that the mean embeddings of attributes and objects are significantly separated:

$$\mathcal{L}_{\text{distance}} = -\log(\|\mu_{\mathcal{A}} - \mu_{\mathcal{O}}\|_2) \tag{6}$$

where $\mu_{\mathcal{A}} = \frac{1}{|\mathcal{A}|}\sum_{j=1}^{|\mathcal{A}|} \mathbf{e}_{\mathcal{A}_j}$ and $\mu_{\mathcal{O}} = \frac{1}{|\mathcal{O}|}\sum_{j=1}^{|\mathcal{O}|} \mathbf{e}_{\mathcal{O}_j}$ represent the mean embeddings of attributes and objects, respectively. The distance is computed using the L_2 norm between the mean embeddings of the two groups.

The total Separation loss is a weighted combination of the orthogonality and mean separation components:

$$\mathcal{L}_{\text{separation}} = \lambda_1 \mathcal{L}_{\text{distance}} + \lambda_2 \mathcal{L}_{\mathcal{A}} + \lambda_3 \mathcal{L}_{\mathcal{O}} \tag{7}$$

where λ_1,λ_2 and λ_3 are hyperparameters controlling the contribution of $\mathcal{L}_{\text{distance}}$, $\mathcal{L}_{\mathcal{A}}$ and $\mathcal{L}_{\mathcal{O}}$ to the final loss, respectively.

Decorrelation. To further ensure the independence between object and attribute embeddings, we introduce Decorrelation by using HSIC [4]. For an object o_j with attribute a_j, we formulate the HSIC loss as follows:

$$\mathcal{L}_{\text{hsic}} = \lambda_h \text{HSIC}(o_j, a_j) \tag{8}$$

Here, λ_h is a hyperparameter that controls the contribution of the HSIC term to the total loss.

3.6 Compositional Anticipation

Our method, which we refer to as Compositional Anticipation (CA), consists of both Compositional Smoothing and Compositional Independence. While CA does not actively predict unseen compositions, it prepares the model by refining how it handles potential new compositions and disentangles attributes from objects. Figure 1 shows how we implement CA in Grounding DINO [10].

3.7 Contrastive Prompt Tuning

To improve the performance of some underperforming composition after training with $\mathcal{C}_p$, we extend $\mathcal{C}_s$ with an additional set $\mathcal{C}_i$ to improve performance. Our approach begins with analyzing the predictions to identify compositions that are frequently confused with each other. For instance, if c_j and c_k are often mixed-up, both compositions are included in $\mathcal{C}_i$, and a trainable prompt is added in front of the underperforming class(es). For example, if c_j performs poorly, we add the following prompt in front of the class: *"is not c_k but is c_j"*. This prompt contains both a negative and an affirmative component.

We refer to this method as Contrastive Prompt Tuning and it does not modify any of the tokens present in the sets $\mathcal{A}$ and $\mathcal{O}$. Instead, it focuses solely on the learnable prompt, which leads to fewer changes in the performance of other compositions and mitigates catastrophic forgetting. By doing this, we exploit the ability of a VLM to understand language and use a semantically meaningful initial prompt to learn to distinguish between similar compositions. This step is depicted as Incremental Learning in Fig. 1.

4 Experiments

4.1 Evaluation

Dataset. We evaluate our approach using a synthetic dataset generated following the CLEVR framework [6]. This dataset consists of three types of objects: cube, cylinder, and sphere. Each object is associated with six attributes: blue, red, green, purple, brown, and yellow.

The dataset intentionally excludes non-visual attributes (e.g., heavy) and attributes that exhibit significant variation across different objects (e.g., wet in wet dog versus wet car). This yields a dataset that is reliable for assessing a model's performance in the CZSL task. Given that there are no ambiguous attributes present in this dataset, a poorly performing model would indicate that the model is bad in the CZSL task.

Train-Test Split. Throughout this section, all experiments for the CZSL task are trained using the set: {red cube, blue cube, green sphere, purple sphere, brown cylinder, yellow cylinder} as $\mathcal{C}_p$ with 10 shots per composition. This split ensures that $\mathcal{C}_s$ covers the entire set of objects $\mathcal{O}$ and attributes $\mathcal{A}$. Testing is performed with the whole set of composition $\mathcal{C}$ with 60 samples per composition.

Evaluation Metric. We adopt the NMS mAP evaluation metric introduced by Yoa et al. [23]. In this work they argued that the traditional COCO mAP [9] is deceiving for open vocabulary detection models, such as Grounding DINO [10]. Consider an image annotated with two ground-truth instances: a purple cylinder and a green cylinder, assuming these are the only cylinder categories in the model. These models tend to be able to detect and locate the presence of all cylinders in the image, but they struggle with the contextual description. They would predict two overlapping bounding boxes for each object, mistakenly assigning both 'green' and 'purple cylinder' labels to each object. All four of these boxes would be predicted with a high confidence score. Additionally, the highest scoring label is not necessarily the correct one. Consequently, the AP for each category would misleadingly be 0.50, despite the model failing to correctly comprehend the target objects. Yao et al. [23] refer to this as the 'inflated AP problem'.

To address this issue, Yao et al. [23] propose applying class-agnostic Non-Maximum Suppression (NMS) before calculating the mAP. This method suppresses redundant bounding boxes, ensuring that only the prediction with the highest confidence score is used in the calculation of the mAP. We adopt this NMS mAP metric to provide a more realistic measure of our model's performance.

4.2 CSP Base

We adapt CSP [15] and modify it for Grounding DINO [10] and this integration serves as our baseline method. To assess its performance, we begin by training it with $\mathcal{C}_p = \mathcal{C}$. This yields an NMS mAP of 87.2 ± 6.8, demonstrating that good performance can be achieved by only training the embeddings of $\mathcal{O}$ and $\mathcal{A}$.

Table 1. Compositional Anticipation improves both object detection performance and generalization to unseen compositions. Compositional Smoothing contributes the most to these improvements, followed by Separation and Decorrelation.

Compositional Anticipation (CA)			Seen	Unseen	HM
Compositional Smoothing	Separation	Decorrelation			
✗	✗	✗	81.4 ± 7.6	4.5 ± 4.6	8.0 ± 8.1
✗	✗	✓	81.3 ± 7.7	10.8 ± 6.7	18.2 ± 10.5
✗	✓	✗	82.5 ± 6.8	15.1 ± 4.2	25.4 ± 6.1
✗	✓	✓	84.4 ± 7.1	20.8 ± 4.7	33.1 ± 6.4
✓	✗	✗	86.2 ± 6.8	64.3 ± 5.9	73.5 ± 5.3
✓	✗	✓	92.4 ± 3.0	61.6 ± 5.7	73.8 ± 4.5
✓	✓	✗	86.0 ± 6.1	67.7 ± 4.7	75.7 ± 5.0
✓	✓	✓	88.7 ± 4.9	70.6 ± 7.4	**78.5** ± **6.0**

Table 2. Our model does not forget. It achieves good performance on the fine-tuned CLEVR [6] dataset while preserving performance on MS-COCO [9], whereas conventional fine-tuning of Grounding DINO [10] leads to forgetting on MS-COCO.

	CLEVR [6]		MS-COCO [9]	
Model	Before	After	Before	After
Grounding DINO [10]	23.4	91.0 $^{\uparrow 67.6}$	41.1	11.8 $^{\downarrow 29.3}$
+ CSP [15] + CA (ours)		76.6 $^{\uparrow 53.2}$		41.1 $^{=0.0}$

4.3 CZSL Comparison

We compare the CSP [15] baseline with our proposed method, Compositional Anticipation (CA) which extends CSP with Compositional Independence and Compositional Smoothing. The results, averaged over 13 experimental runs, are shown in Table 1 and are denoted using the NMS mAP metric [23]. Our results show that our method substantially improves upon the CSP baseline, with the harmonic mean (HM) between seen and unseen compositions improving by 70.5%. This improvement is predominately achieved on the unseen compositions, which improved by 66.1%.

Additionally, we showcase that our method does not suffer from catastrophic forgetting by evaluating its generalization ability compared to the conventional fine-tuning of Grounding DINO [10]. We compare the results on the MS-COCO [9] dataset before and after fine-tuning on the CLEVR [6] dataset for the CZSL task. Table 2 shows that conventional fine-tuning of Grounding DINO [10] achieves a 67.6% improvement on CLEVR, whereas our method achieves a 53.2% improvement. This suggests that conventional fine-tuning of Grounding DINO is superior in CZSL. However, conventional fine-tuning of Grounding DINO leads to a 29.3% performance drop on MS-COCO, whereas our method maintains stable performance with no drop at all. This demonstrates that conventional fine-tuning suffers from catastrophic forgetting, while our method does not.

4.4 Improving Incrementally

In this experiment, we incrementally learn new classes using the model initially trained with CSP [15] extended with Compositional Anticipation. We continue training the model using a dataset that includes both $\mathcal{C}_p$ and $\mathcal{C}_i$.

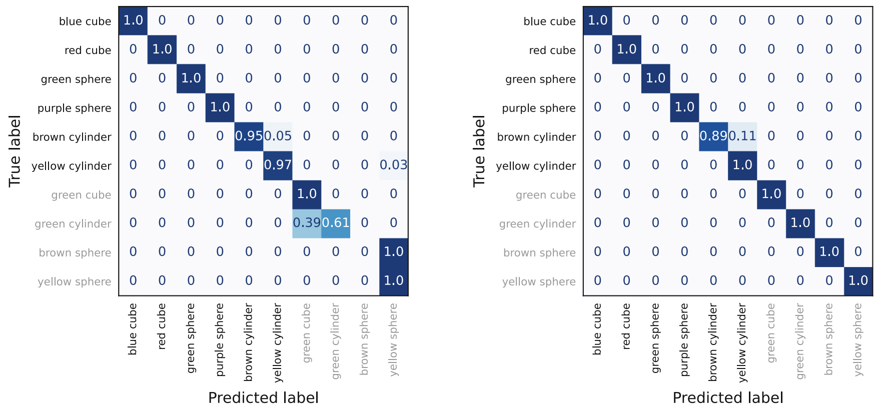
We explore two different fine-tuning methods: fine-tuning class-specific tokens and our proposed method, Contrastive Prompt Tuning. For fine-tuning class-specific tokens, we compare CSP [15] with CSP extended with Compositional Anticipation. Additionally, we conduct this fine-tuning in two ways: (1) allowing the fine-tuning of all tokens in the sets $\mathcal{O}$ and $\mathcal{A}$, and (2) fine-tuning only objects and attributes present in $\mathcal{C}_i$, specifically $\mathcal{O}_i$ and $\mathcal{A}_i$. For Contrastive Prompt Tuning, all tokens are frozen and only the prompt is fine-tuned. The prompt is initialized with semantically meaningful information, including both an affirmative and a negative component. For instance, if "green cylinder" is often confused with "green cube", the prompt is initialized as "is not green cube but is green cylinder". We also analyze the individual contributions of each component to the overall performance enhancements.

To evaluate performance, we compare the model's results before and after introducing $\mathcal{C}_i$. Specifically, we determine performance across the sets $\mathcal{C}_p$, $\mathcal{C}_i$, and $\mathcal{C}_u$. Initially, $\mathcal{C}_u$ is defined as $\mathcal{C} - \mathcal{C}_p$. After introducing $\mathcal{C}_i$, $\mathcal{C}_u$ becomes $\mathcal{C} - \mathcal{C}_p - \mathcal{C}_i$. The results on these sets after introducing $\mathcal{C}_i$ are shown in Table 3, with the absolute changes compared to the initial values indicated with arrows.

Our results show that our method, Contrastive Prompt Tuning, which fine-tunes a prompt initialized with prior knowledge to address specific mistakes

Table 3. Our Contrastive Prompt Tuning is effective for incremental learning. It improves performance across all classes including the unseen ones.

Method	Tunable Tokens	Pretrained	Increment	Unseen	HM
Class-Specific Tuning					
CSP	$\mathcal{O} + \mathcal{A}$	88.1 ± 4.5 $^{\downarrow 2.9}$	72.9 ± 20.3 $^{\uparrow 13.6}$	0.0 ± 0.0 $^{\downarrow 74.3}$	0.0 ± 0.0 $^{\downarrow 71.7}$
+ CA	$\mathcal{O} + \mathcal{A}$	80.9 ± 5.0 $^{\downarrow 10.1}$	60.5 ± 18.2 $^{\uparrow 1.2}$	76.5 ± 8.4 $^{\uparrow 2.1}$	69.6 ± 7.0 $^{\downarrow 2.1}$
CSP	$\mathcal{O}_i + \mathcal{A}_i$	89.0 ± 2.3 $^{\downarrow 2.0}$	64.1 ± 20.9 $^{\uparrow 4.8}$	69.4 ± 8.3 $^{\downarrow 4.9}$	70.6 ± 8.2 $^{\downarrow 1.1}$
+ CA	$\mathcal{O}_i + \mathcal{A}_i$	89.2 ± 3.9 $^{\downarrow 1.8}$	62.4 ± 19.6 $^{\uparrow 3.1}$	78.6 ± 4.9 $^{\uparrow 4.3}$	73.3 ± 9.1 $^{\uparrow 1.5}$
Compositional Prompt Tuning (ours)					
Affirmation	Prompt	88.8 ± 5.5 $^{\downarrow 2.2}$	82.8 ± 17.8 $^{\uparrow 23.5}$	74.8 ± 6.6 $^{\uparrow 0.5}$	80.2 ± 7.9 $^{\uparrow 8.5}$
Negation	Prompt	91.2 ± 2.0 $^{\uparrow 0.2}$	81.9 ± 13.8 $^{\uparrow 22.6}$	76.6 ± 5.8 $^{\uparrow 2.3}$	82.1 ± 6.0 $^{\uparrow 10.4}$
Both	Prompt	92.6 ± 1.5 $^{\uparrow 1.6}$	93.7 ± 2.1 $^{\uparrow 34.4}$	75.2 ± 5.0 $^{\uparrow 0.9}$	**86.2 ± 2.1** $^{\uparrow 14.5}$



(a) Before incremental learning (b) After incremental learning

Fig. 2. Our Contrastive Prompt Tuning method is effective in incremental learning. It improves performance on the increment compositions (in gray) while preserving performance of the pretrained compositions (in black). (Color figure online)

related to confusion between similar compositions, is superior to the class-specific tuning strategy. With Contrastive Prompt Tuning, we achieve a 12.9% enhancement in the HM across the pretrain, increment, and unseen sets compared to the best class-specific tuning method. This improvement is predominately achieved across the increment set, which improved by 31.3% compared to the best class-specific tuning method. Furthermore, Contrastive Prompt Tuning benefits from both the affirmative and negative components of the prompt.

Figure 2 shows the effects of Compositional Prompt Tuning on the model's predictions. Figure 2a shows that before incremental learning, 39% of all instances of "green cylinder" are misclassified as "green cube", and all instances of "brown sphere" are misclassified as "yellow sphere". Figure 2b demonstrates that after applying Compositional Prompt Tuning, "green cube", "green cylinder", "yellow sphere", and "brown sphere" are classified correctly on all instances.

5 Conclusion

In this paper, we demonstrated that conventional fine-tuning of Grounding DINO achieves an NMS mAP of 91.0 when fine-tuned on the CLEVR dataset for CZSL. However, this approach suffers from catastrophic forgetting, as confirmed by a 29.3% decrease in performance on the MS-COCO dataset post fine-tuning. To address this, we proposed incorporating CSP into Grounding DINO to mitigate forgetting by only fine-tuning auxiliary tokens. However, we observed that using CSP alone resulted in an NMS mAP of only 8.0 for the HM between seen and unseen compositions. Therefore, we extended CSP with Compositional Anticipation, which improved the HM by 70.5%. While our method improves upon the CSP baseline, it does not surpass conventional fine-tuning of Grounding DINO. Additionally, we introduced Contrastive Prompt Tuning to incrementally improve compositions that are confused with each other during training. With Contrastive Prompt Tuning, we improve performance on the HM across the pre-train, increment, and unseen sets by 12.9% compared to the best class-specific tuning method.

Given these findings, we recommend conventional fine-tuning of Grounding DINO for applications where performance on a specific dataset is prioritized, and our method for scenarios emphasizing overall performance across datasets. However, we acknowledge that our experiments are limited to the CLEVR dataset, and it remains unclear how the proposed methods will perform on real-world datasets beyond this toy dataset.

Considering that Grounding DINO excels in CZSL, likely due to the cross-modality fusion between image and text embeddings and our proposed methods involve strategically guiding the positioning of embeddings in the embedding space. Having demonstrated the benefits of our approach for CZSL, further investigation into the positioning of embeddings in the fused embedding space could potentially yield results approaching those achieved by conventional fine-tuning of Grounding DINO, but without encountering catastrophic forgetting.

References

1. Bang, J., Kim, H., Yoo, Y., Ha, J.W., Choi, J.: Rainbow memory: continual learning with a memory of diverse samples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8218–8227 (2021)
2. Bang, J., Koh, H., Park, S., Song, H., Ha, J.W., Choi, J.: Online continual learning on a contaminated data stream with blurry task boundaries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9275–9284 (2022)
3. Chen, K., et al.: Ovarnet: towards open-vocabulary object attribute recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 23518–23527 (2023)
4. Gretton, A., Fukumizu, K., Teo, C., Song, L., Schölkopf, B., Smola, A.: A kernel statistical test of independence. In: Advances in Neural Information Processing Systems, vol. 20 (2007)

5. Huang, S., Wei, Q., Wang, D.: Reference-limited compositional zero-shot learning. In: Proceedings of the 2023 ACM International Conference on Multimedia Retrieval, pp. 443–451 (2023)
6. Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: a diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2901–2910 (2017)
7. Li, L.H., et al.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10965–10975 (2022)
8. Li, X., Yang, X., Wei, K., Deng, C., Yang, M.: Siamese contrastive embedding network for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9326–9335 (2022)
9. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
10. Liu, S., et al.: Grounding Dino: marrying Dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
11. Lu, X., et al.: DRPT: disentangled and recurrent prompt tuning for compositional zero-shot learning. arXiv preprint arXiv:2305.01239 (2023)
12. Mancini, M., Naeem, M.F., Xian, Y., Akata, Z.: Open world compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5222–5230 (2021)
13. Misra, I., Gupta, A., Hebert, M.: From red wine to red tomato: composition with context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1792–1801 (2017)
14. Nagarajan, T., Grauman, K.: Attributes as operators: factorizing unseen attribute-object compositions. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 172–190. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_11
15. Nayak, N.V., Yu, P., Bach, S.H.: Learning to compose soft prompts for compositional zero-shot learning. arXiv preprint arXiv:2204.03574 (2022)
16. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
17. Rao, Y., et al.: Denseclip: language-guided dense prediction with context-aware prompting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18082–18091 (2022)
18. Ruis, F., Burghouts, G., Bucur, D.: Independent prototype propagation for zero-shot compositionality. In: Advances in Neural Information Processing Systems, vol. 34, pp. 10641–10653 (2021)
19. Saini, N., Pham, K., Shrivastava, A.: Disentangling visual embeddings for attributes and objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13658–13667 (2022)
20. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
21. Wang, H., Yang, M., Wei, K., Deng, C.: Hierarchical prompt learning for compositional zero-shot recognition. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, pp. 1470–1478 (2023)

22. Wang, Q., et al.: Learning conditional attributes for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11197–11206 (2023)
23. Yao, Y., et al.: How to evaluate the generalization of detection? A benchmark for comprehensive open-vocabulary detection (2024)
24. Yao, Y., Zhang, A., Zhang, Z., Liu, Z., Chua, T.S., Sun, M.: CPT: colorful prompt tuning for pre-trained vision-language models. AI Open **5**, 30–38 (2024)
25. Zhang, J., Huang, J., Jin, S., Lu, S.: Vision-language models for vision tasks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. (2024)
26. Zheng, Z., Zhu, H., Nevatia, R.: Caila: concept-aware intra-layer adapters for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1721–1731 (2024)
27. Zhou, D.W., Wang, Q.W., Qi, Z.H., Ye, H.J., Zhan, D.C., Liu, Z.: Deep class-incremental learning: a survey. arXiv preprint arXiv:2302.03648 (2023)
28. Zhou, D.W., Zhang, Y., Ning, J., Ye, H.J., Zhan, D.C., Liu, Z.: Learning without forgetting for vision-language models. arXiv preprint arXiv:2305.19270 (2023)
29. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16816–16825 (June 2022)
30. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. Int. J. Comput. Vision **130**(9), 2337–2348 (2022)