Does CLIP Bind Concepts?
Probing Compositionality in Large Image Models

Anonymous ACL submission

Abstract

Large-scale neural network models combining text and images have made incredible progress in recent years. However, it remains an open question to what extent such models encode compositional representations of the concepts over which they operate, such as correctly identifying *red cube* by reasoning over the constituents *red* and *cube*. In this work, we focus on the ability of a large pretrained vision and language model (CLIP) to encode compositional concepts and to bind variables in a structure-sensitive way (e.g., differentiating *cube behind sphere* from *sphere behind cube*). To inspect the performance of CLIP, we compare several architectures from research on compositional distributional semantics models (CDSMs), a line of research that attempts to implement traditional compositional linguistic structures within embedding spaces. We benchmark them on three synthetic datasets – single-object, two-object, and relational – designed to test concept binding. We find that CLIP can compose concepts in a single-object setting, but in situations where concept binding is needed, performance drops dramatically. At the same time, CDSMs also perform poorly, with best performance at chance level.

1 Introduction

Good semantic representations are generally assumed to require, at a minimum, *compositionality* and *groundedness*. That is, meanings of sentences should be functions of the words they contain and the syntax via which those words are combined (Partee, 1995) (*compositionality*), and such meanings should be at least in part responsible for reference to the real world, e.g., via truth conditions (*groundedness*). The current state-of-the-art of semantic representation consists of vectors extracted from very large neural networks trained either on text alone (Devlin et al., 2019; Brown et al., 2020; Touvron et al., 2023) or a mix of text and images (Radford et al., 2021; OpenAI, 2023). It remains a wide-open question whether such models constitute good semantic representations (Pavlick, 2022), with empirical evidence and in-principle arguments simultaneously supporting claims that models are and are not compositional (Marcus and Millière, 2023), and that they are and are not grounded (Piantadosi and Hill, 2022; Bender and Koller, 2020; Mollo and Millière, 2023).

In this paper, we focus on vision-and-language models [1] (specifically CLIP and fine-tuned variants of CLIP), and seek to answer, in a controlled setting, whether such models meet basic tests of grounded compositionality. Specifically, we consider three basic types of linguistic compositions: combining a single adjective and noun (*red cube*), combining two adjectives with respective nouns (*red cube and blue sphere*), and relating two nouns (*cube behind sphere*). All three of these settings require some degree of compositionality and groundedness, with the latter two exemplifying a more abstract type of compositionality (pervasive in language) which depends not only on recognizing a conjunction of constituents but an ability to bind meaning representations to abstract syntactic roles. Recently, there has been a significant interest in the community to benchmark the compositional capabilities of CLIP and other VLMs (Ma et al., 2022; Yuksekgonul et al., 2023; Thrush et al., 2022). However, Hsieh et al. (2023a) shows that these datasets are 'hackable' as the incorrect labels may not be meaningful and do not require the image to predict the correct label. For example, an image of *a horse eating the grass* can have the distractor *the grass eating a horse*. In contrast, we are less

[1] There is significant debate about whether text-only language models can be considered "grounded". It is often assumed that models trained on multimodal data will circumvent this debate, but this should not be taken for granted. Our findings add to work which shows that VLMs don't necessarily learn a grounded semantics of the type traditionally sought in linguistics; further work and debate is necessary to make normative claims about the representations that VLMs learn.

prone to such "hackable" artifacts as we include meaningful distractors that require both the image and the labels for the final prediction. We therefore provide a controlled setting for benchmarking compositionality in CLIP.

We situate our work within the tradition of research on *compositional distributional semantics models* (CDSMs) (Erk and Padó, 2008; Mitchell and Lapata, 2010; Baroni and Zamparelli, 2010; Coecke et al., 2010; Boleda, 2020), which seek to bridge the gap between distributional models and formal semantics by building architectures which operate over vectors yet still obey traditional theories of linguistic composition. We adapt several such models to the grounded language setting, and compare the performance of CLIP's text encoder (tuned in various settings) to the performance of these explicitly compositional models. Overall, we see that on single adjective-noun compositions (*red cube*), CLIP performs better than any of the more explicitly compositional CDSMs. In the other settings, which rely on the ability to bind variables, we see that using CDSMs for the text encoder sometimes improves performance, but not always, and that, across all models, performance is essentially at chance in the best case. These results suggest that CLIP's representation of the visual world is poorly suited for compositional semantics, and suggest that future work on improving these representations is a necessary next step in advancing work on grounded compositional distributional semantics.

In summary, we make the following contributions:

- We provide a controlled analysis of the ability of CLIP and fine-tuned variants to perform compositional visual reasoning tasks.
- We adapt a variety of traditional compositional distributional semantics (CDS) architectures to the grounded language setting.
- We show that all our models perform poorly on generalization settings that require abstract variable binding, suggesting major limitations in the way CLIP represents the visual world.

2 Models

In this work, we are interested in comparing contemporary "end to end" methods for training neural networks with explicitly compositional models of the type developed in compositional distributional semantics (Erk and Padó, 2008; Mitchell and Lapata, 2010; Baroni and Zamparelli, 2010; Coecke et al., 2010; Boleda, 2020) (henceforth CDSMs for "compositional distributional semantics models"). Below, we describe the models we compare, including baselines, explicitly compositional models, and contemporary vision-and-language models.

2.1 Setup

We describe a unified setup that we use to represent compositions in CLIP-based models as well as in CDSMs. For each compositional task, we are given a dataset $\mathbb{S} = \{(x, y)\}$ where x is the image and $y \in \mathbb{Y}$ is a *phrase* which correctly describes the image where $\mathbb{Y}$ is the set of all phrases. We use CLIP (Radford et al., 2021) to get image embeddings for all input images. Embeddings for the phrases are generated either using the text encoder in CLIP (possibly fine-tuned) or using CDSMs.

We train different CLIP variants and CDSMs in order to encode each of the phrases. We deal with two types of phrases, namely, adjective-noun and subject-relation-object phrases. Let $\mathbb{A} = \{a_1, \ldots, a_n\}$ be the adjectives and $\mathbb{N} = \{n_1, \ldots, n_m\}$ be the nouns in an adjective-noun phrase. The models produce the adjective-noun phrase embedding $\mathcal{T}(a, n)$ in the joint semantic space where $a \in \mathbb{A}$ and $n \in \mathbb{N}$. Letting $\mathrm{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$ be the relations, the model generates the relational phrase embedding $\mathcal{T}(s, \mathcal{R}, o)$ where the subject is $s \in \mathbb{N}$, the relation is $\mathcal{R} \in \mathrm{R}$, and the object is $o \in \mathbb{N}$. All models, with the exception of frozen CLIP, are trained to update phrase embeddings based on the training data. For the compositional models, the word embeddings that are composed to form the phrase embedding are updated. For more details, see Section 4.

2.2 CLIP and Variants

We examine the performance of CLIP (Radford et al., 2021), fine-tuned CLIP, and a compositional variant (Nayak et al., 2023) on the tasks.

CLIP CLIP (Radford et al., 2021) is a pretrained vision-and-language model trained with a contrastive loss objective on 400 million image-text pairs. The architecture includes two key components: an image encoder and a text encoder that produce vector representations for images and texts in the joint semantic space. The text encoder accepts prompts in natural language to produce zero-shot classifiers. We get the final prediction by taking the cosine similarity between the image and the text vectors and choosing the text with the highest similarity score. This ability enables us to test CLIP out-of-the-box on compositional tasks. We set the following prompt templates for the adjective-noun and subject-relation-object setting:

$$\mathcal{T}(a, n) = \phi(\texttt{a photo of adj noun})$$
$$\mathcal{T}(s, \mathcal{R}, o) = \phi(\texttt{a photo of sub rel obj})$$

where ϕ is the CLIP pretrained text encoder, `adj noun` is replaced with the adjective and noun pairs, and `sub rel obj` is replaced with nouns and relations from the dataset. We consider frozen CLIP and a fine-tuned variant CLIP-FT (Section 4).

Compositional Soft Prompting CSP or compositional soft prompting (Nayak et al., 2023) is a parameter-efficient learning technique designed to improve the compositionality of large-scale pretrained models like CLIP. They focus on real-world adjective-noun datasets which contain images of a single object associated with an adjective. They fine-tune embeddings of tokens corresponding to adjective and object concepts on a set of seen classes while keeping other parameters of the text and the image encoders frozen. During inference, they recompose adjective and object tokens in new concatenations for zero-shot inference. In this work, we systematically evaluate CSP on different types of compositional tasks (Section 4). We set the following prompt templates for the adjective-noun and subject-relation-object setting:

$$\mathcal{T}(a, n) = \phi(\texttt{a photo of [adj] [noun]})$$
$$\mathcal{T}(s, \mathcal{R}, o) = \phi(\texttt{a photo of [sub] [rel] [obj]})$$

where ϕ is the pretrained text encoder in CLIP, `[adj] [noun]` are the fine-tuned token embeddings for adjectives and nouns and `[sub] [rel] [obj]` are the fine-tuned token embeddings for nouns and relations in the dataset.

2.3 Compositional Distributional Semantics Models (CDSMs)

We consider a number of compositional distributional semantics models, which have been proposed in past work but have not been applied to a grounded language setting. Each of these models trains embeddings (vectors, matrices, or tensors) for each word in the class, and then composes them together to produce a compositional phrase embedding. All models are trained to learn the phrase embeddings by aligning them with the frozen image embeddings from CLIP.

Syntax Insensitive Models (Add, Mult, Conv) We consider three simple compositional models that are insensitive to order. The first two are Add, consisting of combining word vectors by addition, and Mult, where word vectors are combined by pointwise multiplication (Mitchell and Lapata, 2010; Grefenstette and Sadrzadeh, 2011). Lastly, we use circular convolution (Conv) (Plate, 1995). For $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{c} = \mathrm{Conv}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a} \circledast \boldsymbol{b}$ means that $\boldsymbol{c}_i = \sum_{j=0}^{n-1} \boldsymbol{a}_j \boldsymbol{b}_{i-j}$ where $i - j$ is interpreted as modulo n.

Type-logical model (TL) Type-logical approaches to distributional semantics map grammatical structure into vector space semantics (Baroni and Zamparelli, 2010; Coecke et al., 2010). Concretely, we represent the nouns as vectors, adjectives as matrices, and the composition of an adjective and a noun is given by matrix-vector multiplication. Following Kartsaklis et al. (2012), we represent transitive verb or relation as a matrix, and the composition of the noun-relation-noun is given by matrix-vector multiplication followed by pointwise vector multiplication, i.e.:

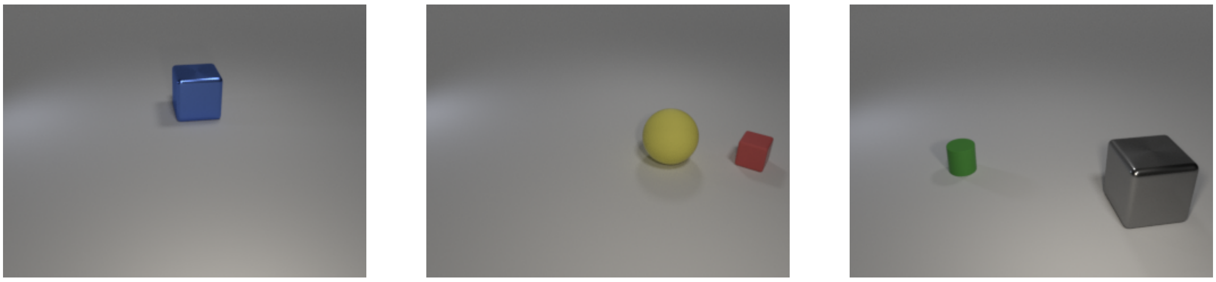
$$\mathcal{T}(a, n) = \boldsymbol{A} \cdot \boldsymbol{n}, \quad \mathcal{T}(s, \mathcal{R}, o) = \boldsymbol{s} \odot (\boldsymbol{R} \cdot \boldsymbol{o})$$

where $\boldsymbol{n}$, $\boldsymbol{s}$, and $\boldsymbol{a}$ are learnable embeddings, $\boldsymbol{A}$ and $\boldsymbol{R}$ are learnable weight matrices, $\cdot$ is matrix-vector multiplication and $\odot$ is pointwise multiplication .

Role-filler model (RF) Introduced in Smolensky (1990), role-filler-based representations provide a means of representing structure using vectors. A symbolic structure can be represented as a collection of role-filler bindings, instantiated within a vector space. Consider *red cube* which is rendered as $\mathbf{red} \circledast \mathbf{adj.} + \mathbf{cube} \circledast \mathbf{noun}$ where $\mathbf{adj.}$ and $\mathbf{noun}$ are role vectors, $\mathbf{red}$ and $\mathbf{cube}$ are filler vectors, and circular convolution $\circledast$ is a binding operator (Plate, 1995). Formally, we learn an embedding for each filler, of type noun, adjective, or relation, and another set of embeddings for each role:

$$\mathcal{T}(a, n) = \boldsymbol{a} \circledast \boldsymbol{r}_a + \boldsymbol{n} \circledast \boldsymbol{r}_n$$
$$\mathcal{T}(s, \mathcal{R}, o) = \boldsymbol{s} \circledast \boldsymbol{r}_s + \boldsymbol{R} \circledast \boldsymbol{r}_R + \boldsymbol{o} \circledast \boldsymbol{r}_o$$

where all of $\boldsymbol{a}$, $\boldsymbol{n}$, $\boldsymbol{s}$, $\boldsymbol{R}$, $\boldsymbol{o}$, $\boldsymbol{r}_a$, $\boldsymbol{r}_n$, $\boldsymbol{r}_s$, $\boldsymbol{r}_R$, and $\boldsymbol{r}_o$ are learnable embeddings and $\circledast$ is the circular convolution operation.



(a) Single-object dataset. Example true label and distractors are: {blue cube, yellow sphere, gray cube, purple cylinder, cyan cylinder}

(b) Two-object dataset. Example true label and distractors are: {yellow sphere, yellow cube, red sphere, blue cube, purple cylinder}. *yellow cube* and *red sphere* are 'hard' distractors.

(c) Relational dataset. Example true label and distractors are: {cylinder left of cube, cube left of cylinder, cylinder right of cube, sphere left of cube, cylinder left of sphere}.

Figure 1: Example images and label sets from each dataset. The texts in Green are the true classes and Red are the distractors. Unlike the two-object and relational datasets, the single-object dataset does not require concept binding.

	Train		Validation		Generalization	
Dataset	# Examples	# Classes	# Examples	# Classes	# Examples	# Classes
Single-object	5598	14	799	2	3195	8
Two-object	20000	14	20000	2	20000	8
Relational	40000	20	20000	2	20000	2

Table 1: Summary of the statistics of the datasets in the concept binding benchmark.

3 Concept Binding Benchmark

We introduce the concept binding benchmark to evaluate the compositional generalization capabilities of VLMs. In this benchmark, we introduce three datasets: single-object, two-object, and relational (see Figure 1). Following Johnson et al. (2017), we use Community (2018) to generate synthetic datasets with objects of simple shapes and colors. Each dataset contains train, validation, and generalization sets with no overlap in the true class labels. Class labels are of the form *adjective-noun* or *subject-relation-object*. All individual nouns, adjectives, and relations are included in the training sets such that we can train models on the training set and test for compositional generalization on held-out classes in the validation and generalization set. Unlike prior work that introduces datasets with a focus on concept binding (Yuksekgonul et al., 2023; Ma et al., 2022; Thrush et al., 2022), our synthetically generated datasets contain both semantically meaningful and hard labels and provide a controlled setting to evaluate the compositional capabilities of VLMs. Table 1 shows the statistics of the datasets.

Single-object dataset The dataset consists of images of exactly one object of a given shape and color (see Figure 1a). We consider the following shapes and colors: cubes, spheres, and cylinders and blue, gray, yellow, brown, green, purple, red, and cyan with a total of 24 possible combinations. The validation set includes brown cube and green cylinder and the generalization set includes green cube, purple cube, red cube, cyan cube, blue cylinder, gray cylinder, yellow cylinder, and brown cylinder. The remainder of the combinations are included in the training set. The correct label for the image is an adjective-noun label. Four distractors are sampled from the other possible adjective-noun combinations.

Two-object dataset The dataset contains images with two objects of different shapes each associated with a different color (see Figure 1b). Following the single object experiments, we use the same shape-color combinations in the train, validation, and generalization split. A correct label for a given image is again an adjective-noun label. However, we manually choose "harder" distractors by switching the adjective and object compositions. For example, in Figure 1b we have two classes *red cube* and *yellow sphere*. When *red cube* is the positive label, we set two of the four distractors to be *red sphere* and *yellow cube*. The other two distractors are randomly sampled from the pool of negative labels, say *blue sphere* and *red cylinder*. We follow the same procedure when *yellow sphere* is the positive example.

Relational dataset This dataset contains images with two objects. A correct label for an image is given by a phrase of the form *subject relation object*. We consider the following objects and relations: cube, sphere, and cylinder and left, right, front, and behind. This means there are 24 possible combinations of spatial relations of the form $a\mathcal{R}b$ where $\{a, b\}$ are objects and $a \neq b$ and $\mathcal{R}$ is the relation. For each image, the distractor labels are constructed as $\{b\mathcal{R}a, a\mathcal{S}b, a\mathcal{R}c, c\mathcal{R}b\}$ where $c \notin \{a, b\}$ is an object type other than a or b and $\mathcal{S}$ is the relation opposite to $\mathcal{R}$. The validation set includes images of cubes in front of spheres (equivalently, spheres behind cubes), and the generalization set includes images of cylinders in front of cubes (equivalently, cubes behind cylinders). All the other 20 image types are seen in the training set, and note that shapes can appear on either side of the image. Figure 1c shows an example from the training set with a *cylinder behind cube*.

4 Experiments and Results

To understand the compositional capabilities of CLIP, we benchmark CLIP and the compositional models from Section 2 on the three datasets described in Section 3. Detailed training setup and parameters are given in Appendix A. The code, included in the supplementary, will be released.[2]

4.1 Single Adjective-Noun Composition

We test the ability of our models to correctly classify the composition of objects with properties (e.g., "red cube") in the single-object dataset.

Results In Table 2, we see that frozen CLIP outperforms all the models. CLIP achieves 97.75% on the validation set and 92.39% on the generalization set. After fine-tuning, CLIP's performance drops to 89.06% on the validation set and 78.54% on the generalization set. We observe a similar trend in CSP, i.e., the performance on the validation set reduces to 84.58% but achieves slightly better performance on the generalization set with 88.74%. We suspect this drop is because the model overfits to the true compositions in the training set.[3] Out of the CDSMs, Add and RF both perform well on training and generalization sets, achieving 80.64% and 85.16% on the generalization set respectively. We see that Conv, Mult, and TL are unable to generalize to the validation and the generalization sets. These three models can achieve high performance (high 90s) on the training set after several epochs but at the expense of performance on the validation set (not included in Table 2 as we report accuracy based on best performance on the validation set).

A breakdown of errors on the generalization set

Model	Train	Val	Gen
CLIP	94.23	97.75	92.39
CLIP-FT	$98.98_{1.02}$	$89.06_{5.84}$	$78.54_{4.41}$
CSP	$94.98_{0.45}$	$84.58_{0.16}$	$88.74_{0.34}$
Add	$99.77_{0.03}$	$44.98_{1.32}$	$85.16_{0.96}$
Mult	$43.27_{13.9}$	$4.48_{4.08}$	$5.38_{2.66}$
Conv	$41.10_{14.3}$	$7.33_{2.90}$	$4.11_{1.53}$
TL	$99.98_{0.02}$	$1.08_{0.44}$	$0.92_{0.24}$
RF	$98.87_{0.11}$	$59.52_{6.12}$	$80.64_{1.36}$

Table 2: Results for all models on single adjective-noun composition, training epoch chosen by performance on validation set. We report the average accuracy for all the methods on 5 random seeds and the standard error.

Model	Adj	Noun	Both
CLIP	83.47	14.87	1.65
CLIP-FT	$0.12_{0.12}$	$92.95_{4.09}$	$6.94_{3.98}$
CSP	$85.19_{0.72}$	$12.57_{0.72}$	$2.24_{0.05}$
Add	$94.85_{0.51}$	$1.13_{0.22}$	$4.02_{0.43}$
Mult	$33.47_{3.17}$	$14.70_{2.62}$	$51.84_{5.75}$
Conv	$29.59_{3.19}$	$13.12_{1.84}$	$57.29_{4.25}$
TL	$39.18_{0.72}$	$21.64_{0.27}$	$39.17_{0.50}$
RF	$64.01_{2.70}$	$10.99_{1.08}$	$24.99_{2.50}$

Table 3: Percentages assigned to each type of error for the single-object color task, generalization split. Here, Adj means the model predicted the adjective incorrectly but the noun correct; Noun means the opposite error; and Both means the model predicted neither the adjective nor the noun correctly. We report the average error proportions for all the methods on 5 random seeds and the standard error.

[2] anonymous github url

[3] Calibrating predictions on the validation set is a common practice in zero-shot learning to reduce bias towards seen classes. We find calibration improves CSP from 88.74% to 96.31% on the single-object setting. This shows fine-tuned variants of CLIP can generalize better than frozen CLIP. However, calibration in the two-object setting does not improve generalization accuracy suggesting this setting is harder as it requires *binding* adjectives to objects. Details in Appendix C.

Model	Train	Val	Gen
CLIP	27.02	7.17	31.40
CLIP-FT	86.91 $_{8.15}$	6.31 $_{3.31}$	0.25 $_{0.10}$
CSP	37.59 $_{1.54}$	20.98 $_{0.22}$	11.15 $_{2.03}$
Add	32.46 $_{0.11}$	15.38 $_{0.89}$	21.37 $_{0.60}$
Mult	86.65 $_{8.93}$	4.66 $_{1.35}$	0.13 $_{0.03}$
Conv	46.26 $_{0.53}$	7.11 $_{2.18}$	0.28 $_{0.14}$
TL	99.41 $_{0.17}$	21.23 $_{4.08}$	0.08 $_{0.07}$
RF	25.23 $_{1.08}$	25.13 $_{3.99}$	20.36 $_{1.36}$

Table 4: Results for all models on adjective-noun binding task, training epoch chosen by performance on validation set. We report the average accuracy for all the methods on 5 random seeds and the standard error.

Model	Adj	Noun	Both
CLIP	53.08	45.40	1.51
CLIP-FT	47.63 $_{0.26}$	46.89 $_{1.20}$	5.48 $_{1.01}$
CSP	49.22 $_{0.54}$	48.25 $_{0.72}$	2.53 $_{0.17}$
Add	53.57 $_{0.16}$	44.32 $_{0.25}$	2.11 $_{0.23}$
Mult	48.51 $_{0.03}$	46.43 $_{1.13}$	5.06 $_{1.15}$
Conv	44.27 $_{0.19}$	38.20 $_{0.35}$	17.53 $_{0.43}$
TL	48.76 $_{0.03}$	47.85 $_{0.12}$	3.39 $_{0.15}$
RF	50.64 $_{0.91}$	41.32 $_{1.26}$	8.04 $_{1.46}$

Table 5: Percentages assigned to each type of error for the two-object setting. Here, Adj means the model predicted the adjective incorrectly but the noun correct; Noun means the opposite error; and Both means the model predicted neither the adjective nor the noun correctly. We report the average error proportions for all the methods on 5 random seeds and the standard error.

is reported in Table 3. We see that CSP, Add, and RF have similar types of errors, i.e., these models often predict the incorrect adjective but predict the correct noun. CLIP-FT, however, predicts the adjective (color) correctly but gets the noun wrong.

4.2 Two-Object Adjective-Noun Binding

In this task, we test whether CLIP can *bind* concepts together. Given two objects, can CLIP bind adjectives to correct objects as opposed to merely representing the image as a "bag of concepts"? For example, in Figure 1b, can CLIP predict that the image contains a *red cube* rather than a *yellow cube*?

Results This task is more challenging for all models (Table 4). Frozen CLIP performs at a level close to chance. After fine-tuning, we see that CLIP-FT overfits to the training set, achieving good training accuracy (86.91%), but falling much lower on validation and generalization (6.31% and 0.25% respectively). At the epoch with the best accuracy on the validation set, CSP has a lower performance on the training set and slightly higher on the validation and generalization sets compared to CLIP-FT. However, as training progresses, we observe that CSP also overfits to the training set (not reported in the table). We see that Conv, Mult and TL also exhibit the same pattern of overfitting to the training data, with high training accuracy and low validation and generalization accuracy. The additive models, Add and RF, underfit the training set and show random accuracy on validation and generalization sets.

Table 5 shows that the errors are similar across the models. For most models, the errors are evenly split between the adjectives and the nouns while only a small proportion of the errors get both incorrect. However, we find that Conv incorrectly predicts both the adjective and noun. For the best performing models, Add and RF, there is a slight bias towards getting the adjective wrong rather than the noun.

4.3 Relational Composition

In this task, we test understanding of spatial relationships between objects, i.e., can our models *bind* objects to positions? This task requires the models to encode an order or relation between two arguments. For example, in Figure 1c, can CLIP differentiate between *cube behind cylinder* and *cylinder behind cube*, even though they have the same words?

Results Frozen CLIP performs slightly better than chance on the training set, but worse on the validation and generalization sets, indicating that these may be more difficult (Table 6). After fine-tuning, CLIP-FT improves to around 50% on the training set, but is completely unable to generalize. This pattern is also seen for CSP and TL. All the other CDSMs perform slightly above chance. This is to be expected for Add, Mult, and Conv because they are commutative. Surprisingly, RF is unable to perform better than chance in this setting. We suspect that RF has a lower capacity as RF only fine-tunes the role and filler parameters. Fine-tuning the image encoder along with the role and filler parameters will increase the complexity of the model and potentially improve the performance on

Model	Train	Val	Gen
CLIP	26.80	14.99	0.00
CLIP-FT	49.59 $_{0.44}$	0.00 $_{0.00}$	0.00 $_{0.00}$
CSP	30.40 $_{0.11}$	0.12 $_{0.01}$	0.03 $_{0.00}$
Add	25.41 $_{0.13}$	26.03 $_{0.07}$	25.47 $_{0.18}$
Mult	25.67 $_{0.12}$	25.95 $_{0.09}$	25.78 $_{0.09}$
Conv	24.83 $_{0.06}$	26.36 $_{0.55}$	24.95 $_{0.11}$
TL	67.19 $_{0.26}$	0.00 $_{0.00}$	0.00 $_{0.00}$
RF	25.18 $_{0.28}$	24.89 $_{0.73}$	22.78 $_{0.20}$

Table 6: Results for all models on relational composition. We report the average accuracy for all the methods on 5 random seeds and the standard error.

Model	$b\mathcal{R}a$	$a\mathcal{S}b$	$a\mathcal{R}c$	$c\mathcal{R}b$
CLIP	50.00	50.00	0.00	0.00
CLIP-FT	37.54 $_{7.60}$	45.97 $_{2.41}$	12.19 $_{7.78}$	4.30 $_{1.94}$
CSP	49.75 $_{0.01}$	49.77 $_{0.01}$	0.40 $_{0.01}$	0.08 $_{0.00}$
Add	34.21 $_{0.08}$	65.79 $_{0.08}$	0.00 $_{0.00}$	0.00 $_{0.00}$
Mult	34.41 $_{0.17}$	65.57 $_{0.17}$	0.01 $_{0.01}$	0.01 $_{0.01}$
Conv	32.98 $_{0.27}$	66.14 $_{0.11}$	0.54 $_{0.24}$	0.34 $_{0.10}$
TL	49.06 $_{0.55}$	49.44 $_{0.33}$	1.07 $_{0.64}$	0.44 $_{0.27}$
RF	53.09 $_{0.46}$	46.18 $_{0.32}$	0.48 $_{0.14}$	0.26 $_{0.08}$

Table 7: Percentages assigned to each type of error for the relational task. We report the average error proportions for all the methods on 5 random seeds and the standard error.

the various splits.

Table 7 gives a breakdown of errors. Recall that the distractors have a specific structure: if a correct caption for the image is $a\mathcal{R}b$, then the given distractors are: $b\mathcal{R}a$, $a\mathcal{S}b$, $a\mathcal{R}c$, $c\mathcal{R}b$. We note that CLIP, CSP, and TL have a very similar pattern of errors: each model is able to distinguish objects perfectly, and almost all errors are split between $b\mathcal{R}a$ and $a\mathcal{S}b$ - tuples that have been seen in training. The three commutative models, Add, Mult, and Conv, also have a distinctive error pattern. Errors are again focused on $b\mathcal{R}a$ and $a\mathcal{S}b$, with approximately a 1:2 split. This indicates that the models select the relation $\mathcal{R}$ 50% of the time, and $\mathcal{S}$ the other 50%. When $\mathcal{R}$ is selected, the predictions are split again between $a\mathcal{R}b$ and $b\mathcal{R}a$, since these cannot be distinguished by the commutative models. Although the overall performance of RF is similar to these models, the pattern of errors is more similar to that of CLIP, CSP, and TL. Finally, CLIP-FT has another different pattern of errors, in which more of the error is now on the objects, rather than the relation. We also note that these errors are much noisier than for the CDSMs.

5 Discussion

Our work highlights the limitations of CLIP as a basis for compositional language representations. We show that CLIP is capable of disassociating objects and adjectives, enabling it to behave compositionally in the single-object setting. However, it appears to lack a richer structure necessary for compositions that require more abstraction, such as syntax-sensitive variable binding. We find that fine-tuning CLIP or training composition-aware models (CDSMs) does not help the model generalize better on the unseen classes for two-object and relation settings. Our results show that among the CLIP variants, CLIP-FT overfits to the training set and achieves high training accuracy while hurting the generalization accuracy. CSP can show improved training accuracy over CLIP and sometimes show increases in validation and generalization accuracy but not always. Among the syntax insensitive models, we see that Add, Mult, and Conv improve on the training accuracy on the single-object and the two-object settings but only Add generalizes to held-out classes in the single-object setting. As expected, these models cannot represent order and achieve accuracy close to chance on the relational dataset. Our results with type-logical models (TL) have high training accuracy but validation and generalization accuracy are usually close to 0. Finally, RF can learn to generalize to classes in the single-object dataset but achieves chance on the two-object and the relational dataset. Our experiments focus only on CLIP, and thus should be interpreted conservatively. Newer visual encoders trained with different training objectives may produce better results, even with the same text encoders we use in the paper. Or, perhaps, progress on compositionality both in visual and text encoding will be necessary to alleviate the problems highlighted here. Overall, our results motivate the need for pretraining methods in VLMs that account for binding for better compositionality.

We also shed light on the benchmarking datasets used in compositional zero-shot learning. Typical benchmarking datasets for this task are MIT-States (Isola et al., 2015), UT-Zappos (Yu and Grauman, 2014), and C-GQA (Mancini et al., 2021). CLIP and CSP show strong performance compared to several existing methods on these datasets (see

Section 5 in Nayak et al. (2023)). However, these datasets do not explicitly test for binding of adjectives to nouns, i.e., they are restricted to a single-object setting. While this setting captures one important aspect of composition, it does not require models to encode an abstract, order-aware syntax, a critical component of linguistic composition. In our experiments, we find that CLIP and CSP show high accuracy on the single-object dataset (Section 3) but the performance drops dramatically on the two-object dataset (Section 4.2) and relational dataset (Section 4.3). Challenging datasets like ARO (Yuksekgonul et al., 2023) show that fine-tuning CLIP with harder negative images and captions can improve CLIP's accuracy on the relational split that accounts for the order of objects. Our training setup shares similarities as we include hard negative captions for each image. However, we do not see improved performance after fine-tuning. Recent work (Hsieh et al., 2023b) shows that the ARO benchmark includes test examples that can be solved without the visual encoder which could explain the possible improvement in performance. These findings motivate the need for more realistic and challenging benchmarks that test for binding and order.

6 Related Work

Compositionality in Language Our work contributes to the extensive body of work in compositionality and language spanning several decades (Smolensky, 1990; Plate, 1995; Baroni and Zamparelli, 2010; Coecke et al., 2010; Socher et al., 2012; McCoy et al., 2019; Smolensky et al., 2022). Key models of composition used in language include simple elementwise composition (Mitchell and Lapata, 2010), neural models of composition (Socher et al., 2012), type-logical models of composition (Baroni and Zamparelli, 2010; Coecke et al., 2010), and role-filler modes of composition (Smolensky, 1990; Plate, 1995; McCoy et al., 2019). We focus on type-logical and role-filler models of composition. In the area of type-logical models, our work extends models from Maillard and Clark (2015); Wijnholds et al. (2020); Nagarajan and Grauman (2018) to learn from both images and text and to handle a wider range of compositions. Within the area of role-filler approaches, recent work has looked at approaches to reasoning (Chen et al., 2020), mathematics (Russin et al., 2021), and whether recurrent neural networks can be emulated using role-filler approaches (McCoy et al., 2019). In particular, McCoy et al. (2019) use tensor product representations to show that sentence encoders (Conneau et al., 2017; Kiros et al., 2015) can be well approximated by a "bag of words" model. In this work, we show that CLIP image embeddings behave like a "bag of concepts".

Compositionality in Vision There is a growing interest in compositionality and vision (Misra et al., 2017; Nagarajan and Grauman, 2018; Naeem et al., 2021; Mancini et al., 2021; Lovering and Pavlick, 2022; Nayak et al., 2023; Yun et al., 2022; Tull et al., 2023). Several architectures have been proposed to improve benchmark results on compositional zero-shot learning datasets (Yu and Grauman, 2014; Isola et al., 2015; Mancini et al., 2021). However, these datasets are often restricted to an adjective-noun setting, ignoring concept binding. Recently, datasets such as CREPE (Ma et al., 2022), ARO (Yuksekgonul et al., 2023), and Winoground (Thrush et al., 2022) study compositionality in VLMs including concept binding, but may not provide a faithful and controlled environment benchmark (Hsieh et al., 2023b). In contrast, we build a controlled setup without potential confounders that arise with real-world images to carefully study compositional visual reasoning. Concurrently, Clark and Jaini (2023) compared the performance of frozen CLIP and Imagen, a text-to-image model, on a task similar to our two-object dataset. They find that Imagen, in some cases, performs more strongly, suggesting that generative models are better at binding concepts.

7 Conclusion

We investigate the ability of CLIP and variants and CDSMs in a controlled environment to perform compositional visual reasoning tasks. Our results show that CLIP performs well on the single adjective-noun compositions but struggles on compositional tasks that rely on the ability to bind variables. Some of the CDSMs perform well on single adjective-noun composition but show performance closer to chance in the two-object and relational tasks. Our work not only sheds light on the limitations of CLIP but also suggests that the pretraining of VLMs should account for binding and order for better compositional generalization.

8 Limitations and Risk

8.1 Models

We run our experiments on one major VLM (CLIP) and compare these results with a set of compositional models. Results on the benchmarking datasets we propose may differ for other VLMs. The compositional models we test do not include some types of model such as Recursive Neural Networks (Socher et al., 2012), but we do compare key types of model (type-logical and role-filler) from the compositional literature.

8.2 Datasets

The Concept Binding Benchmark that we propose studies concept binding with artificially generated shapes. While the simplicity of our datasets strengthens the findings, we suspect that the results may differ with more realistic images.

8.3 Language

The language we look at is limited to English. For the CLIP models that we use, we are limited to English, however, for the compositional models, it would be possible to use other languages, including alternative grammatical structures and word orderings. The kind of language used in the labels is very simple, and further work could include more complicated descriptions of the images.

8.4 Risk

This research presents limited risk, due to the abstract nature of the datasets and the limited domain of investigation. All previously existing artefacts have been used within the limits of their original purpose.

9 Ethical Considerations

The abstract nature of the datasets we use means that ethical implications of the type of modeling done are minimal. We do use English as a language, however, the methods we propose for the CDSMs could be applied to other languages, as in Moortgat and Wijnholds (2017). The training methodology involves fine-tuning a VLM with a large number of parameters (see Table 8), however use of this model can be minimized by saving out frozen image embeddings and using these to train CDSMs.

References

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1183–1193, USA. Association for Computational Linguistics.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Gemma Boleda. 2020. Distributional Semantics and Linguistic Theory. *arXiv:1905.01896 [cs]*. ArXiv: 1905.01896.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. 2016. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European conference on computer vision (ECCV)*.

Kezhen Chen, Qiuyuan Huang, Hamid Palangi, Paul Smolensky, Ken Forbus, and Jianfeng Gao. 2020. Mapping natural-language problems to formal-language solutions using structured neural representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1566–1575. PMLR. ISSN: 2640-3498.

Kevin Clark and Priyank Jaini. 2023. Text-to-image diffusion models are zero-shot classifiers.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical Foundations for a Compositional Distributional Model of Meaning. *Lambek Festschrift, Linguistic Analysis*, 36.

Blender Online Community. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from

Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP ’08, pages 897–906, USA. Association for Computational Linguistics.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental Support for a Categorical Compositional Distributional Model of Meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. 2023a. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. 2023b. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. In *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Phillip Isola, Joseph J. Lim, and Edward H. Adelson. 2015. Discovering states and transformations in image collections. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1383–1391. IEEE Computer Society.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. 2012. A Unified Sentence Space for Categorical Distributional-Compositional Semantics: Theory and Experiments. In *Proceedings of COLING 2012: Posters*, pages 549–558, Mumbai, India. The COLING 2012 Organizing Committee.

Jamie Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Advances in neural information processing systems*, volume 28.

Charles Lovering and Ellie Pavlick. 2022. Unit testing for concepts in neural networks. *Transactions of the Association for Computational Linguistics*, 10:1193–1208.

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. 2022. Crepe: Can vision-language foundation models reason compositionally? *arXiv preprint arXiv:2212.07796*.

Jean Maillard and Stephen Clark. 2015. Learning Adjective Meanings with a Tensor-Based Skip-Gram Model. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 327–331, Beijing, China. Association for Computational Linguistics.

M Mancini, MF Naeem, Y Xian, and Zeynep Akata. 2021. Open world compositional zero-shot learning. In *34th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Gary Marcus and Raphaël Millière. 2023. Compositional Intelligence Research Group. `https://compositionalintelligence.github.io/`.

R. Thomas McCoy, Tal Linzen, Ewan Dunbar, and Paul Smolensky. 2019. RNNs Implicitly Implement Tensor Product Representations. In *ICLR 2019 - International Conference on Learning Representations*.

Ishan Misra, Abhinav Gupta, and Martial Hebert. 2017. From red wine to red tomato: Composition with context. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1160–1169. IEEE Computer Society.

Jeff Mitchell and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429.

Dimitri Coelho Mollo and Raphaël Millière. 2023. The vector grounding problem.

Michael Moortgat and Gijs Wijnholds. 2017. Lexical and derivational meaning in vector-based models of relativisation.

Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. 2021. Learning graph embeddings for compositional zero-shot learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 953–962.

Tushar Nagarajan and Kristen Grauman. 2018. Attributes as Operators: Factorizing Unseen Attribute-Object Compositions.

Nihal V. Nayak and Stephen H. Bach. 2022. Zero-shot learning with common sense knowledge graphs. *Transactions on Machine Learning Research (TMLR)*.

Nihal V. Nayak, Peilin Yu, and Stephen H. Bach. 2023. Learning to compose soft prompts for compositional zero-shot learning. In *International Conference on Learning Representations*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Barbara Partee. 1995. Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311–360.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Ellie Pavlick. 2022. Semantic structure in deep learning. *Annual Review of Linguistics*, 8:447–471.

Steven T. Piantadosi and Felix Hill. 2022. Meaning without reference in large language models. *ArXiv*, abs/2208.02957.

T.A. Plate. 1995. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641.

Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. 2019. Task-driven modular networks for zero-shot compositional learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3592–3601. IEEE.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR.

Frank Ruis, Gertjan J Burghouts, and Doina Bucur. 2021. Independent prototype propagation for zero-shot compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34.

Jacob Russin, Roland Fernandez, Hamid Palangi, Eric Rosen, Nebojsa Jojic, Paul Smolensky, and Jianfeng Gao. 2021. Compositional Processing Emerges in Neural Networks Solving Math Problems. *CogSci ... Annual Conference of the Cognitive Science Society. Cognitive Science Society (U.S.). Conference*, 2021:1767–1773.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1-2):159–216.

Paul Smolensky, Richard McCoy, Roland Fernandez, Matthew Goldrick, and Jianfeng Gao. 2022. Neurocompositional computing: From the central paradox of cognition to a new generation of ai systems. *AI Magazine*, 43(3):308–322.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic Compositionality through Recursive Matrix-Vector Spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea. Association for Computational Linguistics.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Sean Tull, Razin A. Shaikh, Sara Sabrina Zemljic, and Stephen Clark. 2023. Formalising and Learning a Quantum Model of Concepts. ArXiv:2302.14822 [quant-ph, q-bio].

Gijs Wijnholds, Mehrnoosh Sadrzadeh, and Stephen Clark. 2020. Representation Learning for Type-Driven Composition. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 313–324, Online. Association for Computational Linguistics.

Aron Yu and Kristen Grauman. 2014. Fine-grained visual comparisons with local learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 192–199. IEEE Computer Society.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2023. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*.

Tian Yun, Usha Bhalla, Ellie Pavlick, and Chen Sun. 2022. Do vision-language pretrained models learn primitive concepts? *arXiv preprint arXiv:2203.17271*.

A Training Details

We provide the training details and hyperparameters used in the experiments. We build the training and evaluation pipeline in PyTorch (Paszke et al., 2019). The models are trained on a single NVIDIA RTX 3090, A40, or V100 GPU depending on their availability. The models are trained for 20 epochs which takes about 6-20 minutes per epoch depending on the dataset. Table 8 shows the number of trainable parameters in all the models used in our experiment.

We have three categories of models: CLIP, CLIP variants, and CDSMs (Add, Mult, Conv, TL, RF). All the models use pre-trained CLIP ViT-L/14 in the experiments [4]. These methods except CLIP are trained with a cross entropy loss on the train split using an Adam optimizer. We use frozen CLIP to predict the classes for the images in the datasets. During training, we set the batch size of 32 and weight decay of 10^{-5}. CLIP (FT) fine-tunes all the model parameters including the vision and text encoder with a learning rate of 10^{-7}. In CSP, we initialize the token embeddings by averaging the embeddings of all the tokens in the English name of the adjective, noun, or relation to get one initial token embedding per concept. Then, we fine-tune them on the training split with a learning rate of 10^{-6}. In CDSMs, we randomly initialize the model parameters and train them with a learning rate of $5 \cdot 10^{-4}$. We train all our models on the train split and use the validation split to select the final model for testing based on accuracy.

	Dataset	
Method	Single/Two-object	Relational
CLIP-FT	429M	429M
CSP	8,448	5,376
Add	8,448	5,376
Mult	8,448	5,376
Conv	8,448	5,376
RF	9,984	7,680
TL	4.7M	2.3M

Table 8: The number of trainable parameters in each experiment.

[4] https://github.com/openai/CLIP/blob/main/model-card.md.

B Training Algorithm

We describe the algorithm used to train the models. Models are trained to align the caption vectors with the image vectors. Algorithm 1 shows the training algorithm for adjective-noun phrases. We follow a similar procedure to train relational phrases.

Algorithm 1: Algorithm to train the model on the adjective-noun compositions.

Input : Training dataset $\mathbb{S}$, image encoder $\mathcal{I}$, composition encoder $\mathcal{T}$, learnable parameters $\boldsymbol{\theta}$, adjectives $\mathbb{A}$, nouns $\mathbb{N}$, λ weight decay, number of distractors D, number of epochs M

Output : The model parameters $\boldsymbol{\theta}$

for $i \leftarrow 1$ *to* M **do**

  foreach $x, y = (a, n) \in \mathbb{S}$ **do**

    $\boldsymbol{x} \leftarrow \mathcal{I}(x)$; get the image vector

    $\mathbb{Y}^D_{\text{neg}} \leftarrow$ sample D distractors from $\mathbb{Y}_{neg} = \mathbb{Y} \setminus \{y\}$

    $l_{\text{pos}} \leftarrow \boldsymbol{x} \cdot \mathcal{T}(a, n)$

    $l_{\text{neg}} \leftarrow \sum_{y_{\text{neg}} \in \mathbb{Y}^D_{\text{neg}}} \boldsymbol{x} \cdot \mathcal{T}(y_{\text{neg}})$

    $p_{\boldsymbol{\theta}}(y = (a, n)|x) \leftarrow \frac{\exp{(l_{\text{pos}})}}{\exp{(l_{\text{pos}} + l_{\text{neg}})}}$

    $\mathcal{L} \leftarrow -\log p_{\boldsymbol{\theta}}(y|x) + \lambda||\boldsymbol{\theta}||_2$; cross entropy loss with weight decay

    $\boldsymbol{\theta} \leftarrow$ update all learnable parameters

  end

end

return $\boldsymbol{\theta}$; the learned model parameters

C Calibrated Stacking

Calibrated stacking is a standard practice in zero-shot learning (Chao et al., 2016; Nayak and Bach, 2022). Zero-shot models tend to be overconfident or biased towards seen classes because they only see the unseen classes as negatives or they are excluded from the training altogether. We can fix this overconfidence by simply calibrating the predictions on validation data. Following prior work in zero-shot learning, we add a calibration coefficient to lower the cosine similarity score of the seen classes. During testing, we use the calibration coefficient and calculate the accuracy.

Setup To test whether calibrated stacking improves generalization accuracy, we experiment with CSP on the single object dataset but modify the train set. To find a calibration coefficient, we need a validation set to include seen and unseen classes. Since our validation set contains only unseen classes as the positive labels, we need a additional validation set with seen classes. To fix this issue, we randomly sample 10% of the train set and use that as the seen validation set. We train our model on the remaining 90% of the data with the same training details (see Section 4). Next, we compute the cosine similarity scores for the seen and the unseen validation sets and search for the calibration coefficient. Next, we get the highest cosine similarity $l_{\max}$ and vary the calibration $-l_{\max}$ to $+l_{\max}$ with a step size of $l_{\max}/100$ and choose the coefficient with the highest harmonic mean of the seen and the unseen accuracy. Finally, we use the calibration coefficient on the generalization set and report the performance.

Method	Generalization
CLIP	92.39
CSP	88.74
CSP + calib.	96.31

Table 9: The results for single-object setting on the generalization split. For CSP and CSP + calib., we report the average accuracy on 5 random seeds.

Results Table 9 shows that CSP with calibration improves by 8 points on the generalization split. We also see that CSP improves over CLIP by 4 points showing that the model has learned to generalize to unseen adjective-noun compositions. This shows that fine-tuned models, including the CSDMs, could potentially generalize better than frozen CLIP with calibration. These results are in line with the literature in compositional zero-shot learning that calibrate the predictions and show improved results on the adjective-noun datasets (Purushwalkam et al., 2019; Ruis et al., 2021). However, we find that calibrating the predictions in the two-object setting does not improve the generalization performance the same way. This may be due to the construction of the two-object dataset. In the validation split we have the classes *brown cube* and *green sphere*. The "hard distractors" for these classes are *brown sphere* and *green cube*. However, these hard distractors come from the generalization set, i.e., they are unseen classes. This means the calibration does not decrease the cosine similarity of the hard distractors, making it difficult to calibrate the validation set. Finally, calibration is not applicable to the relational dataset because we consider only two classes in the generalization split, *cube behind cylinder* and *cylinder behind cube*, that are equivalent. This means, we only see one class at a time and simply setting the probability of the distractors to 0, we can get 100% accuracy on the generalization set. For this reason, we do not calibrate on the relational dataset and leave the experiment for future.

D License

All the code including the models and the datasets used in this work are released under open-source licenses. Blender is released under the GNU GPL License, CLIP is released under the MIT license, and CSP is released under the BSD-3 license. Upon acceptance, we will release the concept binding benchmark dataset under the Apache 2 license.