Understanding Multi-compositional learning in Vision and Language models via Category Theory

Sotirios Panagiotis Chytas[1], Hyunwoo J. Kim[2], and Vikas Singh[1]

[1] University of Wisconsin – Madison
[2] Korea University

Abstract. Pre-trained large language models (and multi-modal models) offer excellent performance across a wide range of tasks. Despite their effectiveness, we have limited knowledge of their internal knowledge representation. To get started, we use the classic problem of Compositional Zero-Shot Learning (CZSL) as an example, and first provide a structured view of the latent space that any general model (LLM or otherwise) should nominally respect. We obtain a practical solution to the CZSL problem that can deal with both Open and Closed-World single-attribute compositions as well as multi-attribute compositions with relative ease, where we achieve performance competitive with methods designed solely for that task (i.e., adaptations to other tasks are difficult). Then, we extend this perspective to analysis of existing LLMs and ask to what extent they satisfy our axiomatic definitions. Our analysis shows a mix of interesting and unsurprising findings, but nonetheless suggests that our criteria is meaningful and may yield a more structured approach for potential incorporation in training such models, strategies for additional data collection, and diagnostics beyond visual inspection. The code is available at https://github.com/SPChytas/CatCom.

1 Introduction

In the theory of Forms, Plato argues that every entity in the *physical* world – animals, persons, and all other entities – correspond to a Form (or Idea) that answers "What is that?" [49]. In fact, the objects in our world are merely *imitations* of these "Forms", their impure realizations. The Idea on the other hand refers to the *non-physical* essence of all things. For instance, we can only observe the Idea "car", via an imitation of its *attributed* Form, *red car*, *small car*, *sports car*, and so on. We are almost never shown the Idea, by itself. Still, we inherently possess the ability to perform such compositional classification, and more importantly, recognize novel combinations of attributes and Ideas.

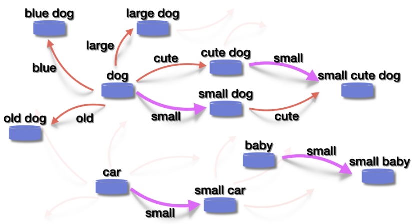


Fig. 1: The implicit structure of concepts should be reflected in a model too. Arrows "add" new information to concepts and are shared among different concepts.

Since we observe the world in this compositional view of attributes and Ideas, it is reasonable to ask whether a model can be trained to perform such compositional learning or whether a model already possesses this ability. The reader will acknowledge that despite intensive (and impressive) ongoing work in foundation and generative models [6, 37, 53, 54], their ability to compose known concepts in novel ways is a work in progress [13,32]. For the remainder of this paper, we will refer to the Ideas as primitives.

A simple example. Consider two objects, a *red car* and a *sports car*. These two objects have something in common: the primitive. How can we leverage this knowledge? What about two other objects: a *red car* and a *big red car*? How are the objects related? How about a *small blue car*? We can appreciate that we are traversing multiple compositions: from *car* to *red car* to *big red car*, and these traversals can often involve backtracking: *big red car* to *red car* to *small red car*. If the primitives are commonly occurring, we may find that a pre-trained diffusion model [54] can already provide such capabilities out of the box. When it fails for a primitive with multiple nested attributes, say for a less common primitive, it is not easy to diagnose such cases in advance and/or simply re-run the model with additional guidance.

This "compositional" problem setting is not unique or novel to our work – indeed, many existing approaches [9, 38, 42, 44, 46, 55] have variously approached the task of compositional learning. However, even now, when Foundation models dominate nearly every popular benchmark, compositionality remains an open question and contemporary architectures can fail on even simple compositionality tasks [16,22,31]. In the compositional task above, the main focus is on the *relationships* between the samples, often with various levels of nesting. The individual samples are relevant but only as a "seed". For the "relations", what constraints should be checked when operating in the latent spaces of large models? Given recent discussions surrounding emergent capabilities in large models, can we check to what extent the latent spaces support compositions?

What is required for understanding compositions is a mechanism to handle, operate on, and reason with structured inclusion relations (i.e., directed) between entities (e.g., car in the example above) during or post- training. It turns out that Category theory provides us with precisely these types of tools. Category Theory [35] is a general (but abstract) mathematical theory of structures and of *systems* of structures [20, 41, 43] and unifies seemingly unrelated fields (such as Group Theory and Topology) using Categories as its building blocks. One of the basic operations in category theory is to compose more complicated objects/systems via simpler ones – when these objects/systems correspond to "relations" or "structure", we directly obtain the main ingredients we will need to bake into our learning model. The high-level goal of our paper is to check what these tools can reveal about the latent space of foundation models commonly used in the community. In doing so, we will see that solutions for compositional zero shot learning (CSZL) fall out directly by a simple instantiation of basic axioms from category theory.

Contributions and paper organization: On the *technical* side, using the setting in compositional zero-shot learning as a jumping off-point, we give a succinct mathematical formulation based on Category Theory, as introduced in §2. To impose structure, we study a simpler formulation that allows a direct generalization to multi-attribute compositional learning tasks (§3). On the *practical* side, we propose a simple, attention-based, instantiation of our formulation that can handle both single and multi-attribute compositions with no adjustments (§3). After demonstrating how our machinery can handle a well-studied problem setting (CSZL), we study what the formulation can say in the context of the latent space of contemporary LLMs (§4).

2 Category Theory: A brief review

A Category consists of **(i) Objects** that correspond to individual entities (e.g., sets) and **(ii) Morphisms** that represent the connections (or arrows) between the Objects (e.g., functions). A Category follows two basic axioms: each Object has an identity Morphism (i.e., a self-loop), and the Morphisms compose. To be precise, consider a Category C and any three Objects $a, b, c \in C$. If $\exists f : a \to b$ and $\exists g : b \to c$ then the composition of these two Morphisms also exists in C and is denoted as $g \circ f : a \to c$. As the above description suggests, Category theory deals with morphisms and *not* the objects; it discourages worrying about what a, b and c actually are. It is highly diagrammatic, with graphical descriptions common in both definitions and proofs.

Definition 1. *Functors are structure-preserving maps between Categories. For two Categories $\mathscr{A}$, $\mathscr{B}$, we define the Functor $F : \mathscr{A} \to \mathscr{B}$ as a mapping with the properties:*

- $F(id_a) = id_{F(a)},\ \forall a \in \mathscr{A}$
- $F(g \circ f) = F(g) \circ F(f),\ \forall f : a_1 \to a_2, g : a_2 \to a_3$ *in* $\mathscr{A}$

A Functor transforms not just the objects but *also* their morphisms across categories.

Example 1. Consider the set of ImageNet [14] images and $\mathbb{R}^n$ as two categories. An image encoder, e.g., ResNet [26], which maps each image to an embedding vector in $\mathbb{R}^n$ is an example of a Functor from the Category of Images to the Category of Vectors.

Example 2. Consider the category $\mathscr{E}ng_{to}\mathscr{F}r$ where each object represents a sentence in English or French, and there is a unique morphism $(translation)$ between corresponding English and French sentences. Mapping this category to an embedding space involves finding a functor $\mathcal{E}nc$ that preserves the structure. Assuming $\mathcal{E}nc$ has an inverse morphism ($\mathcal{D}ec = \mathcal{E}nc^{-1}$), we recover the standard language translation model.

Definition 2. *For two Categories $\mathscr{A}$, $\mathscr{B}$, define the Product Category $\mathscr{A} \times \mathscr{B}$ as the Category with the properties: **(i)** Its Objects are pairs $(a, b)\ \forall a \in \mathscr{A}, b \in \mathscr{B}$ **(ii)** Its Morphisms are the pairs $(f, g) : (a_1, b_1) \to (a_2, b_2)\ \forall f : a_1 \to a_2, g : b_1 \to b_2$ **(iii)** The composition of Morphisms is defined element-wise as $(f_2, g_2) \circ (f_1, g_1) = (f_2 \circ f_1, g_2 \circ g_1)$ for all composable Morphisms.*

For each Product Category $\mathscr{A} \times \mathscr{B}$, we can also define two Functors $P_{\mathscr{A}} : (\mathscr{A} \times \mathscr{B}) \to \mathscr{A}$, $P_{\mathscr{B}} : (\mathscr{A} \times \mathscr{B}) \to \mathscr{B}$ that correspond to the "projections":

- $P_{\mathscr{A}}\big((a, b)\big) = a$; $P_{\mathscr{A}}\big((f, g)\big) = f$
- $P_{\mathscr{B}}\big((a, b)\big) = b$; $P_{\mathscr{B}}\big((f, g)\big) = g$

for each Object, Morphism of $\mathscr{A} \times \mathscr{B}$ respectively. We can think of the operation as the "backtracking" discussed in §1. A key feature of Category Theory, useful for clarity, is the graphical representation of the concepts, see Fig. 2.

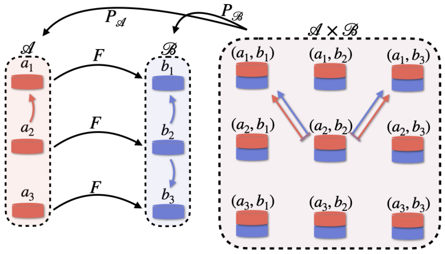


Fig. 2: A Functor $F : \mathscr{A} \to \mathscr{B}$ and the Product Category $\mathscr{A} \times \mathscr{B}$ shown graphically (self-loops omitted). The Functors $P_{\mathscr{A}}, P_{\mathscr{B}}$ also map the Morphisms between $(a_2, b_2) \to (a_1, b_3)$ and $(a_2, b_2) \to (a_1, b_1)$, i.e., the Morphisms $a_2 \to a_1$ and $b_2 \to b_1, b_2 \to b_3$.

Why Category Theory? Category Theory is used in formal methods for reasoning about types and structure [20]. With functional programming approaches gaining prominence in deep learning, practical uses of category theory, can become feasible. A category theory based formulation streamlines the creation of compositional constructs. Ongoing efforts in formulating deep learning concepts via category theory, while nascent, are yielding interesting new algorithms [11, 12, 19, 23, 24, 56]. Since our focus here is to provide a general treatment of compositional learning (as will become clear shortly), we find that Category Theory is an ideal fit. It allows cleaner and more general results, while maintaining the mathematical correctness and rigor desired.

3 Proof of concept: Compositional Zero-shot Learning

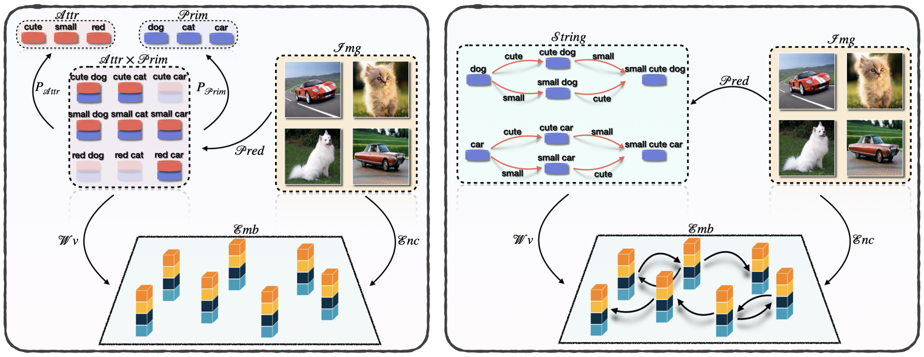


Fig. 3: Left: the typical formulation of CZSL, in Category Theory terms. The opaque pairs correspond to implausible pairs that are not considered under the Closed-World setting, resulting in a smaller space. Right: an alternative formulation that allows for a direct generalization to the multi-attribute setting, by morphisms' composition.

To test drive the concepts above, we start with a simple problem setting. Single-attribute Compositional Zero-shot learning [38, 46] is defined as follows. Assume a labeled (image) dataset of the form $\mathbb{T} = \Big\{\big(x, (a, p)\big) | x \in \mathcal{X}, (a, p) \in \mathcal{Y}_t\Big\}$ where each image's label has two parts; $a \in \mathscr{A}$ is an attribute (e.g., small) and $p \in \mathscr{P}$ is a primitive (e.g., car). Using $\mathbb{T}$, CZSL seeks to accurately characterize unseen images whose labels are novel pairs of attributes and primitives.

3.1 The "product" formulation and its limitations

Fig. 3(left) restates a common formulation in CZSL, represented as a category-theoretic diagram. The approach roughly involves: **(a)** Dealing with attribute-primitive pairs in CZSL, expressed as a discrete Product Category ($\mathscr{A}ttr \times \mathscr{P}rim$). **(b)** Transitioning from words and pairs using various design choices (mostly involving pretrained word embeddings such as GloVe [48]), modeled as a single Functor ($\mathcal{W}v : \mathscr{A}ttr \times \mathscr{P}rim \to \mathscr{E}mb$) in the embedding space Category $\mathscr{E}mb$. **(c)** Associating each image with a label of the form (attribute, primitive), represented as a discrete Category of Images ($\mathscr{I}mg$) with a "prediction" Functor $\mathcal{P}red : \mathscr{I}mg \to \mathscr{A}ttr \times \mathscr{P}rim$. **(d)** Closing the circuit by training CZSL methods to learn representative image embeddings, modeled as a Functor $\mathcal{E}nc : \mathscr{I}mg \to \mathbb{R}^n$ which encodes each image into the same embedding space, ensuring commutative operations,

$$\mathcal{E}nc = \mathcal{W}v \circ \mathcal{P}red \tag{1}$$

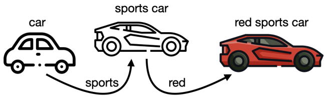
Applicable to LLMs? Although at a high level, this formulation may look too simple to capture what is going on in larger models, this is not the case. Consider each input sentence as a multi-product of words (or more accurately tokens). We indeed learn a mapping of this product to an embedding space, using training tasks such as next-word prediction or masked predictions [15, 52]. Of course, images are missing in this description but we can check that CLIP [51] closely follows the formulation of Fig. 3(left) by aligning a multi-word string (caption) to an image.

No interactions. The "product" formulation is direct and intuitive, but by re-casting everything in Category Theory language, we recognize a key shortcoming; all Categories are missing a critical component: Morphisms. No interaction is assumed between the different (attribute, primitive) pairs or the embedding vectors, e.g., the Objects *small car* and *red car* have no relationship. The result in [46] recognized this issue and used a graph between all primitives, attributes, and pairs. Incorporating this missing component, obvious from our category-theoretic reinterpretation, led to state-of-the-art results in [46]. This hints at the potential benefits of the general categorical approach.

3.2 The Categorical Composer

Building on the observations of §3.1, we present a simple reformulation, which overcomes the issues identified above. In Fig. 3(right), we present an alternative model; we call it the "morphism" formulation (as opposed to the "product" formulation in §3.1).

The "morphism" formulation. Similar to §3.1, we will use Fig. 3(right) as a guide, and walk through the main components, step by step. **(a)** We assume that each attribute is a Morphism, instead of an Object. Each attribute "acts" upon an Object and changes it in a well-defined way. According to this view, different concepts are intrinsically related and not disjoint from each other (similar to the real world).



Interestingly, we can define all multi-attribute concepts (e.g., *red sports car*) in this way simply as the *composition* of Morphisms. There is no need to define a new Category (for a multi-Product of attributes and primitives). **(b)** Based on this view, we define the $\mathscr{S}tring$ Category. Its Objects are all strings of zero or more attributes together with a primitive (e.g., *car*, *sports car*, *red sports car*). **(c)** Since each attribute is a Morphism, there is a (directed) *arrow* from each string that *does not* include this attribute to a string that *does* include it (e.g., τ_{red} : *sports car*$\xrightarrow{red}$*red sports car*). **(d)** Similar to the "product" formulation, we define the Functor $\mathcal{W}v : \mathscr{S}tring \to \mathscr{E}mb$ that maps this structure to Category $\mathscr{E}mb$ that corresponds to an embedding space. We can assume that $\mathscr{E}mb$ is the commonly used $\mathscr{S}et$ Category where each Object is a *set* of one or more vectors in $\mathbb{R}^n$ and each Morphism is a function $\tau : \mathbb{R}^n \to \mathbb{R}^n$. The mapping of the attributes to these functions is what allows traversing the embedding space and perform multi-attribute composition (not possible with the "product" formulation where both attributes and primitives are Objects, i.e., there are no Morphisms). **(e)** Finally, we define the Category of images ($\mathscr{I}mg$) along with the Functors $\mathcal{P}red : \mathscr{I}mg \to \mathscr{S}tring$ and $\mathcal{E}nc : \mathscr{I}mg \to \mathscr{E}$, in a similar way. However, notice that the label can easily be a multi-attribute string such as *big red sports car*.

What is the embedding of a multi-word string? Embedding multiple words often relies on heuristics like concatenation or addition of word embeddings. Even with language models, the use of heuristics such as the mean of output word embeddings to define the entire string's embedding is common. Consider a simple example: *small cute dog*. In both cases, determining $\mathcal{W}v(\textit{small cute dog})$ is important. The "morphism" formulation indirectly defines it, using the commutative property of Functors (Def. 1). So,

$$\mathcal{W}v(\textit{small cute dog}) = \tau_{\textit{small}} \circ \tau_{\textit{cute}} \circ \mathcal{W}v(\textit{dog}), \tag{2}$$

simplifies the learning process to focus on primitives' embeddings and one morphism for each attribute.

Use of Special Variables. Each $\mathscr{S}tring$ Object representing a primitive (e.g., *dog*) is a *Limit* Object, signifying it as the purest representation of all Objects corresponding to that primitive paired with one or more attributes. Assuming attribute order insignificance, each attribute pair forms a *Pushout*, and with existing inverse morphisms (e.g., cute^{-1}, small^{-1}), it becomes a *Pullback*. Assuming inverse morphisms in our $\mathscr{S}tring$ Category (as in [38]), Objects for the same primitive, irrespective of attributes, are *Isomorphic*. This embodies the notion that attributes preserve the "essence" of a primitive.

Implementation details. Here, we provide high-level details of a specific instantiation of our formulation (denoted CatCom) by modeling each concept with a neural network module. Similarly to existing works, we use a pre-trained (on ImageNet [14]) ResNet model [26] together with a trainable feedforward network on top, as the $\mathcal{E}nc$ Functor and pretrained word embeddings (transformed through an MLP) as inputs to the morphisms (Fig. 4).

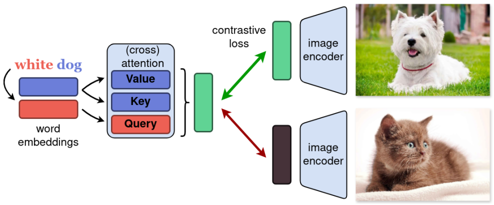


Fig. 4: CatCom overview. Slightly different from Self-Attention [57], the Query vector is derived from another input (the attribute). The entire architecture is trainable, using a contrastive loss.

We use Cross-Attention, used successfully in other scenarios (e.g., Stable Diffusion [54]), to merge the information of a primitive and an attribute (i.e., act as our morphisms). Using the attention mechanism, we obtain a vector that encapsulates the information of both inputs. This architecture allows us to optimize our model end-to-end, similar to CGE [46] and OADis [55], by maximizing the cosine similarity for the correct pair while minimizing the similarity with all other pairs, using the cross-entropy loss. We choose cosine similarity over ℓ_2-loss (also permissible in our model) because the ℓ_2-loss implies that all images with the same label must also have exactly the same embedding. The choice of ℓ_2-loss in [38] may be due to the use of group actions.

3.3 Experimental results

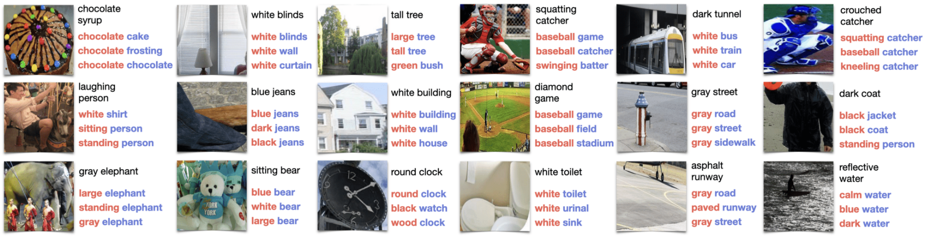


Fig. 5: Qualitative results for C-GQA. In black, we give the true label and below the top-3 predictions. Even from this small sample of images, the difficulty of the CSZL task is apparent. In many cases, our model makes a prediction that is accurate but not the exact ground truth label. Interestingly, we observe that in many cases our model predicts a pair that is more suitable than the actual label (e.g., in the case of the sitting bear).

In this section, we test our method to both single (CW and OW) and multi-attribute CZSL. We follow the evaluation protocols of existing works [38, 46] and test against the current state-of-the-art methods in all cases.

Baselines. We evaluate our method against strong baselines, including: **(a) SymNet** [38]: Handles both single and multi-attribute cases. **(b) CompCos** [42]: Operates in the OW setting, regularizing all possible combinations by assessing similarity between attributes and primitives. **(c) CGE** [46]: Utilizes a GNN for composition embeddings, excelling in the CW single-composition case with a trainable backbone. **(d) OADis** [55]: Implements a triplet loss, works well for CW single-composition by fine-tuning the last ResNet [26] layers. Our baselines, as well as our CatCom, are all based on ImageNet-trained vision models, in contrast to more recent works (e.g., [47]) that use larger models (e.g., CLIP [51]) which may violate the zero-shot nature of the problem.

			MIT-States				UT-Zappos				C-GQA			
		Method			Best				Best				Best	
			AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
“Closed-World”	“product”	CompCos [42]	4.5	16.6	25.8	24.1	19.8	35.2	58.9	42.7	2.6	12.5	28.9	10.8
		CGE [46]	5.1	17.2	28.0	25.2	24.7	38.9	58.8	61.0	2.5	11.9	27.5	11.7
		OADis$_t$ [55]	5.9	18.9	28.9	25.0	30.0	44.4	59.5	65.5	3.5	14.8	31.0	13.7
		CGE$_t$ [46]	**6.5**	**21.4**	31.6	**27.5**	33.0	47.3	61.8	66.3	3.6	14.5	31.4	14.0
	“morph”	SymNet [38]	4.1	16.3	26.2	23.1	27.7	42.5	58.8	61.0	1.8	9.8	25.2	9.2
		CatCom	5.7	18.5	30.8	26.4	29.2	43.0	60.2	64.4	3.5	14.3	**31.6**	**14.2**
		CatCom$_t$	6.3	19.5	**32.1**	**27.5**	**35.3**	**48.8**	**64.7**	**70.1**	**4.3**	**15.8**	**34.1**	**16.0**
“Open-World”	“product”	CompCos [42]	1.5	8.4	25.4	10.0	17.8	32.9	56.1	44.6	0.2	1.0	18.0	0.9
		CGE [46]	1.4	8.0	26.2	8.1	15.4	30.9	58.6	35.9	0.3	1.9	26.8	1.5
		OADis$_t$ [55]	2.1	10.3	29.6	11.6	21.5	**37.4**	56.0	**50.6**	0.5	3.6	29.9	2.9
		CGE$_t$ [46]	**2.2**	**10.6**	**31.6**	10.4	16.0	30.3	60.0	37.5	0.6	3.2	30.0	2.7
	“morph”	SymNet [38]	0.8	5.8	21.4	7.0	20.2	36.8	54.7	44.7	0.2	0.8	14.7	1.7
		CatCom	1.9	9.5	28.8	10.4	17.7	33.0	55.6	43.8	**0.7**	**4.0**	28.8	**3.1**
		CatCom$_t$	**2.2**	**10.6**	31.1	**12.1**	**22.1**	**37.4**	**62.6**	48.0	**0.8**	**4.6**	**31.6**	**3.1**

Table 1: Results for Closed and Open World. The subscript t stands for a trainable backbone. Even our frozen version (denoted CatCom) achieves results comparable to the state of the art methods (with a trainable backbone) and it is better than other methods that use a frozen backbone. When we also train the backbone (denoted CatCom$_t$) then we get a performance similar to CGE$_t$ which is the state-of-art-method. In C-GQA specifically, the most challenging dataset, the simplicity of our method results in much better results than all other methods.

For multi-composition tasks, we consider: **(a) GALM** [10]: Performs multi-label classification across all attributes, with an automated architecture-building procedure. **(b) FMT** [40]: Similar to GALM, automatically designs a multi-task network for multi-attribute classification. **(c) AMT** [25]: Designs a multi-task network, considering correlations between attributes (*dark* and *black*). Note that these methods predict attribute sets through multi-label classifiers, a task simpler than CZSL.

Metrics. We use the same evaluation protocol used in all baselines [46]. We use the same bias scheme in our predictions, so we report **(i)** the Area Under the Curve (**AUC**), **(ii)** The best **Seen** and **Unseen** Accuracy, and **(iii)** the Harmonic Mean (**HM**) between the Seen and the Unseen values. We refer to [46] for more details of the evaluation protocol. In the multi-composition case, since the above metrics are not directly applicable, we consider **mAUC** so that we are consistent with existing works for this problem.

Datasets. We evaluate using three prominent datasets for single attributes: **(i) MIT-States** [29]: The oldest dataset with 245 primitives, 115 attributes, and 28K pairs (only 1262 seen). **(ii) UT-Zappos** [60]: Focuses on shoe images, featuring 192 pairs in total. **(iii) C-GQA** [28, 46]: Recent larger dataset with over 200K pairs derived from GQA [28]. In all datasets, we use the Generalized CZSL splits from prior works [46, 50]. For multi-attribute, we use: **(i) aPY** [18]:A dataset with 64 attributes and 32 primitives, created from PASCAL VOC 2008 [17]. **(ii) SUN** [58]: A larger multi-attribute dataset (102 attributes, 700 primitives).

Main results/observations. Our results are in Tab. 1 and Tab. 2 while qualitative results are depicted in Fig. 5.

Method	mAUC	
	aPU	SUN
AMT [25]	84.5	82.5
FMT [40]	70.5	75.5
GALM [10]	84.2	86.5
SymNet$_s$ [38]	79.9	86.7
SymNet$_m$ [38]	83.4	**88.4**
CatCom	**84.7**	87.9

Table 2: Results for multi-attribute composition. CatCom is comparable to SymNet$_m$ despite our simple evaluation scheme. SymNet$_s$ underperforms in both cases.

4 Compositionality in Language Models

With a general recipe for thinking in a "compositional manner" in hand, we examine if compositionality exists in contemporary large models. This analysis, possible because of the general nature of our formulation, allows us to provide a mechanism to better understand and interpret LLMs, a relevant but underdeveloped topic. We know that models such as BERT [15] and CLIP [51] have been trained with "simpler" loss functions, enabled by huge training datasets. The question then is: "Are these models able to shape their latent space in a way that resembles what a compositional learning algorithm would provide out of the box?"

Quantifying the performance of diversely-trained LLMs remains an active topic of research, with metrics such as the hallucination index [3]. Here, we consider multiple large-foundation models and examine what insights, if any, our structured modeling of the latent space can provide. The reader will notice that at a minimum, our formulation can yield a "compositionality index" and we can check whether large violations are indicative of anything at all.

Dataset, Models and Setup. Using the same terminology as before, we define our Forms to correspond to common everyday primitives (the full set of classes of

ImageNet-1K [14] and CIFAR100 [33]). To assess structure in the latent space, we consider the four most common attributes that will act upon the Form alone (e.g., for images, it does not affect the background): size (e.g., small), color (e.g., green), texture (e.g., shiny), and age (e.g., old), and we examine the latent space of multiple widely used models: **(i) BERT** [15] **(ii) Albert** [34] **(iii) Roberta** [39] **(iv) Deberta** [27] **(v) CLIP** [51] **(vi) GPT2** [52] Following existing works [59], in all models we use the average of all the output embeddings (i.e., one per input token) to form a single embedding for the given prompt.

A note on model selection. We chose the above models since these are some of the most widely used text encoders in practice, for multiple and diverse text-based applications. In fact, our experiments will offer some guidance on why these models are widely used and why, in some cases can be a suitable choice (relative to models like Mistral LLM/Zephyr). To highlight the differences between the two classes of models, we also consider the following three LLMs: **(i) Phi-2** [4] **(ii) Zephyr** [1] **(iii) Mistral** [30] .

In all our experiments, we find the difference vector between the following two quantities: **(i) plain embedding** which corresponds to the embedding of the expression "an image of a(n) $\langle\text{object}\rangle$", and **(ii) attributed embedding** which corresponds to the embedding of the expression "an image of a(n) $\langle\text{attribute}_1\rangle, \cdots,$ and $\langle\text{attribute}_n\rangle$ $\langle\text{object}\rangle$". The plain embeddings correspond to the Forms (the Limits of our String Category) and the differences between the two embeddings correspond to our morphisms, the τ arrows (or functions). We will use the same notation here: the difference will be $\tau^{\text{object}}_{\text{attr}_1\&\cdots\&\text{attr}_n}$.

What are we looking for? Notice that $\tau^{\text{object}}_{\text{attr}}$ are exactly the morphisms in §3.2, and $\tau^{\text{object}}_{\text{attr}_1\&\cdots\&\text{attr}_n}$ is exactly the composition of these morphisms (i.e., $\tau^{\text{object}}_{\text{attr}_1\&\cdots\&\text{attr}_n} = \tau^{\text{object}}_{\text{attr}_n} \circ \cdots \circ \tau^{\text{object}}_{\text{attr}_1}$). Based on this observation, we define **compositionality** to reflect the property that the composition of multiple *atomic* attributes is equivalent to a complicated expression that includes all of them. LLMs may be operating in one of the three following regimes, **(i)** (extreme 1) LLMs have no internal notion of compositionality (i.e., they are "pure" black boxes) and we cannot link them back to our world's structure, **(ii)** (extreme 2) LLMs are able to perfectly pick the compositionality of our world, based on its extensive training data, or **(iii)** (middle) Contemporary training procedures offer a partial "understanding" of compositionality and different training schemes lead to models that align more or less with this view.

Scope of experiments. We will use our formulation to ask the following questions for the three models. How similar are the morphisms for different Forms? What about morphisms for different attributes? How well do multiple morphisms compose?

Remark 1. Different attribute types affect plain embeddings with *different* magnitudes.

Consider a primitive, say, a car (either in text or joint image+text), and assume two variations are available: in the first, it is larger and the second, shows it in a different color. In the latent space, which is more similar to the original? All models concur that the change in size is *less significant* than the change in color.

Remarkably, all models share the same attribute order (of how much "work" τ does), despite being trained on diverse data and objectives. We find that "size" is unanimously the attribute with the smallest magnitude, while "color", "texture", and "age" have similar magnitudes (although usually “age” is slightly smaller). This relationship can be seen in Fig. 6 (top) also, where we project the embeddings to the two-dimensional space using T-SNE. Besides the intra-attribute ordering, the models show an inter-attribute ordering that is consistent with our world-view. In Fig. 6 (bottom), we show the angle between τ_{normal} and τ_{attr} for all other size attributes, for GPT2 and BERT. Both models are able to understand the relative relationships between the different size attributes.

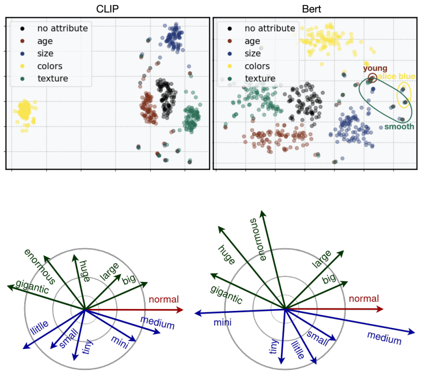


Fig. 6: (Top) T-SNE plot of the clusters that different attributes form, for all objects. Even in this lossy projection, we can observe that color (yellow cluster) leads to a cluster further from the black cluster (plain embeddings) compared to age, size, and texture. (Bottom) the relationship between all the size morphisms reflects how we would also arrange them in real life.

Morphism outliers. From our analysis, we observed the existence of some outliers, for each type of attributes. These outliers correspond to pairs of (atttibute, primitives) that are rare, or even unlikely to ever occur. For instance, some of the pairs are (knitted, bowl), (magenta, baby), and (appicot, woman). This is expected to happen – during training, there is no categorical structure embedded in their latent space, any unknown compositions will be arbitrary and will not behave well.

Besides the primitive-specific magnitude, we observe that, on average, simple and common attributes such as “old”, “young”, “rough”, “cracked”, “big”, and “little” appear to have, on average, a smaller magnitude than more rare/exotic attributes such as “gigantic”, “childlike”, “embossed”, and “time-worn”.

Remark 2. The Yoneda perspective reveals clusters of similar models.

The Yoneda perspective states that an object is completely characterized by its “neighbors” and the morphisms by which they are connected. A similar characterization has recently occurred in [45]. There, the authors calculate the characteristic embeddings using the so-called “anchor points”, leading to a construction that has been formally defined in Category Theory as *Yoneda Embeddings*.

To this end, we construct the Yoneda embeddings [8]. The Yoneda embedding can be thought of as a succinct representation of all neighbors of an object. In our case, we form the vector whose elements are the magnitudes of each morphism (i.e., $\|\tau_{attr}^{object}\|$). We would expect these vectors to be *similar* enough for different models, since all of them were trained on data that resemble the real world. Indeed, when we consider the Pearson Correlation [21] we can observe that all of them are sufficiently similar to each other. In Fig. 7 (top) we can observe the pairwise Correlation between all models. All models have a high, positive correlation, confirming our hypothesis that these models are, in a way, isomorphic views of the same world.

Several intriguing patterns emerge from the analysis. First, the formation of two distinct clusters is evident: Bert aligns closely with Albert, while Roberta exhibits strong alignment with GPT2 and all three LLMs. This suggests a degree of similarity in their embedding spaces. Surprisingly, Roberta's embedding space appears almost isomorphic to that of GPT2, indicating its efficacy in practice. This finding corroborates Roberta's widespread adoption in practical applications, e.g., as a popular choice in Kaggle competitions on text-data.

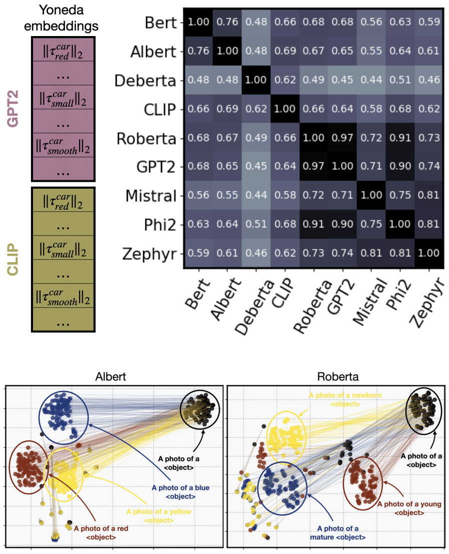


Fig. 7: (Top) The pairwise correlation of the Yoneda embeddings between all models. On the left, we show how we form the Yoneda embedding for each model. The correlation map reveals that Bert and Albert are clustered together, as well as that Roberta, GPT2, and the three LLMs have a high-to-absolute correlation. (Bottom) The latent space of the models is clustered into distinct regions that correspond to each attribute while the homogeneity of each attribute (colored lines) is apparent.

The observation regarding Deberta is interesting. Despite its effectiveness, Deberta stands out as an outlier, displaying a significantly lower correlation with other models. This distinctiveness also aligns with its use in practice, where Deberta is often utilized as a complementary component in ensemble solutions due to its unique properties. This characteristic was particularly evident in the recent Kaggle competition titled "LLM - Detect AI Generated Text" [2] where Deberta featured prominently in all of the top solutions. Overall, these insights provide a nuanced understanding, enabled by our categorical casting, of the relationships between various language models, shedding light on their comparative performance and potential synergies in practical applications.

Remark 3. Models are homogeneous w.r.t. how the same attribute affects different objects and **homogeneity** (i.e., the similarity of an attribute's embedding across different objects) is a unique property that cannot be estimated by the dataset size, the model size, as well as the training type alone.

One major assumption of our framework is the fact that an attribute (e.g., *small*) acts the same way on each primitive (hence the knowledge sharing we exploit in CZSL). For LLMs, this would imply that the morphisms (τ vectors) would be (almost) the same for a specific attribute across all different objects. To this end, we examine the cosine similarity across different objects for the same attribute (Fig. 8 (right)). While all models have relatively high homogeneity, the value differs in a way that no clear correlation exists with the model's size, its training type, or even the release date. This indicates that homogeneity is a unique property that can provide an alternative view of the models. This will become apparent in Remark 5, in which we consider positional attributes also. In Fig. 7 (bottom) we can also observe the 2D projection of 6 attributes.

Even when we "squeeze" the high-dimensional embeddings into two dimensions, the homogeneity of the attributes is preserved.

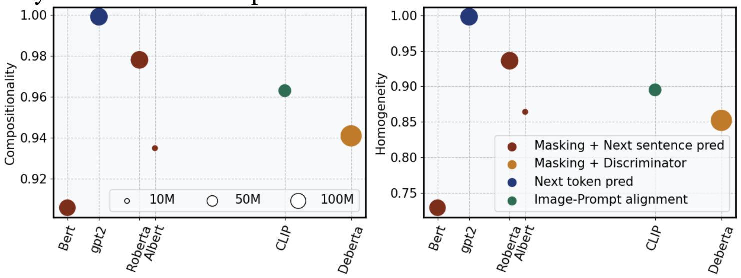


Fig. 8: Compositionality (left) and Homogeneity (right) of each text encoder. The models are arranged in chronological order, based on their release date. We observe that all models have the same profile with respect to both metrics, and that compositionality/homogeneity can not be explained solely by the model's size, or the training type.

Object-specific attributes. An interesting observation emerges from the range of color attributes considered. Our list encompasses a broad spectrum, including less common colors like "pear", "salmon" and "lemon". However, these colors are also used as food nouns. Consequently, when paired with the primitive "can", all models tend to interpret the prompt as referencing a can of lemons, salmon, etc., rather than a can of that specific color. This discrepancy affects the calculation of $\tau^{\text{can}}_{\text{attr}}$, rendering them dissimilar to the τ vectors of other primitives. Additionally, we noticed variations in size-related attributes, particularly with "medium" hinting at its broader usage beyond indicating object size (overloaded).

Remark 4. Compositionality is mostly preserved and can be predicted by attribute homogeneity. Its violation indicates regimes where training data were scarce.

After single-attributed prompts, we turn our focus to multi-attribute prompts. One of the basic questions, and a major thrust of our formulation, is that compositions must be preserved and their violation can be checked. In the two-attribute case, this implies that: $\tau^{\mathrm{X}}_{\mathrm{attr}_1\&\mathrm{attr}_2} = \tau^{\mathrm{X}}_{\mathrm{attr}_1} + \tau^{\mathrm{X}}_{\mathrm{attr}_2}$, $\forall$ object X. In Fig. 8 (left), we show how well each model preserved our compositionality constraint, as a function of model size, training type, as well as release date. Interestingly, all models show a good compositional behavior on average, although many outliers hint that compositionality is not universally respected in these models. A striking observation is the fact that compositionality can be accurately estimated by attribute homogeneity (Fig. 8). While the values may differ, the ordering between the models is the same across the two metrics. In practice, this means that compositionality can be easily estimated quickly since it does not require the quadratic complexity of compositions.

When does compositionality fail? Our evaluation of the constraint violation suggests that compositionality does not hold for either specific, rare attributes or for far-fetched compositions. For instance, an attribute that led to a consistently low compositionality score was the color "puce", no matter the type of the other attribute. Another such attribute was the size-related attribute "mini". As is also apparent from Fig. 6 (bottom), τ_{mini} is quite different than the other size attributes. For CLIP specifically, we believe this is due to the fact that most of the “mini” images correspond to images that depict the car *Mini Cooper* and not a mini version of a typical object. Similarly, for “puce”, it is very likely that images (or even texts) linked to “puce” correspond to the Franco-Belgian comic “Zig et Puce” [5] and not the color “puce”. Finally, odd compositions such as young, bumpy otter and youthful, pear bear lead, as perhaps expected from *Remark* 1, to a low compositionality score.

Remark 5. LLMs have an in-built understanding of viewpoint invariance, but their latent spaces are fundamentally different.

So far, we considered “external” attributes, such as size and color. However, we can also consider attributes that do not really change the essence of the primitive. One such category of attributes is the positional attributes that leads to expressions such as “an image of a car from the *front*” or “an image of a dog from *above*”. Invariance with respect to 3D rotations is an active topic of research and remains an open question. We consider a list of 13 positional attributes and we compute the embedding for each combination of object and position, using the prompt “a photo of a ⟨object⟩ from ⟨position⟩”. If we denote each embedding as $\epsilon^{\text{object}}_{\text{position}}$, we can estimate the viewpoint invariance of each model by calculating the cosine similarity between all the embeddings $\epsilon^{\text{object}}_{\text{position}_1}$ and $\epsilon^{\text{object}}_{\text{position}_2}$ for all combinations of positions and objects.

While we observe that the average cosine similarity is very high (more than 0.95), this does not imply that all models are the same. In Fig. 9 we depict the latent space (T-SNE 2D plot) of two of the models under consideration: the smaller, widely used text-encoder Deberta [27], and a popular more recent LLM Mistral [30]. We can observe an interesting behavior. While Deberta’s latent space is such that (while it achieves viewpoint invariance) the embeddings of each positional attribute are clustered together, this is not the case for Mistral. In Mistral, the different viewpoints are clustered together for each object, leading to this “spotted” view of the latent space, which may not be that useful in practice. We believe that this is one of the properties that make Deberta a widely used text encoder in contrast to LLMs (such as Mistral) that, although show strong performance on multiple NLP tasks, are not a default choice as a text encoder in vision-language tasks.

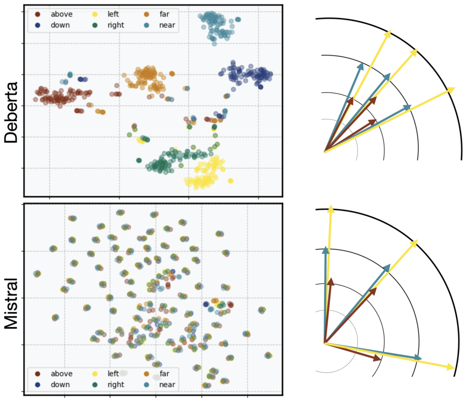


Fig. 9: How viewpoint invariance manifests to different models. In Deberta (top) there are clear clusters for each position, hinting at a latent space in which different object embeddings have a small angle with each other, while different viewpoints lead to embeddings with the same direction but with big differences in the norm. This leads to this type of clustering when we consider a 2D projection. On the contrary, in Mistral (bottom) each object’s viewpoints are clustered together, hinting at a latent space in which each object occupies a very different area.

4.1 Compositionality in Large Language Models

Our main focus so far was in language models that are widely used in multiple applications, such as Deberta, GPT2, and Roberta. However, recently, we have witnessed the emergence of LLMs; billion-parameter models that are dominating multiple, diverse benchmarks. While the LLMs are capable of performing NLP tasks in which smaller models (such as the ones we consider here) fail, this does not imply that they are universally more useful for all types of text-based applications (although this is being actively studied [7,36]). According to the Yoneda perspective (Remark 2), LLMs such as Zephyr (3B) and Mistral (7B) do not differ significantly from smaller models such as GPT2 and Roberta ($\tilde{1}$00M). This implies that, at least for certain applications, it may be beneficial to use a smaller model compared to a larger one, or, alternatively, the use of a larger model may offer minimal improvements.

Additionally, the viewpoint invariance (Remark 5) reveals that, while all models have an in-built notion of viewpoint invariance, their latent space is not equally “useful”. As we observe in Fig. 9, Mistral’s latent space is formed in a way that may not be ideal for a text encoder. This statement agrees with what has been observed in practice too, where Deberta is one of the main choices for a text encoding model, in contrast to larger LLMs. Of course, our experiments do not intend to understate the improvements that various LLMs have achieved, but rather provide a more critical and informed view that shows that huge models are not necessarily well-suited for all applications. The Yoneda perspective as well as compositionality/homogeneity can serve as guidance for selecting models appropriate for the task at hand. Here, we considered a generic set of attributes (age, color, size, texture, as well as positions) but for a specific application, it is straightforward how the categorical view can be applied to a different set of attributes and provide a quick proxy of a model’s performance.

5 Conclusions

We showed how compositional learning can be reinterpreted in Category Theory terms. This perspective allows us to identify strategies to concurrently improve as well as simplify existing formulations. The experimental results suggest that our CatCom model yields a performance profile comparable to the state of the art for Compositional Zero-Shot Learning. The same core ideas offer a mechanism to dig deeper into the latent space of LLMs, from a categorical perspective. Our tests allow us to evaluate less well-studied properties of LLMs, and the potential role the Yoneda perspective and the compositionality index may offer. These metrics can provide insights into how the model is trained and what extra training examples can help. Our multi-level analysis of multiple contemporary LLMs revealed multiple anecdotes that hold in the community of NLP practitioners. Our categorical view, along with the specific metrics we presented, can be used for text-based applications and provide a “ranking” of the candidate models.

Acknowledgments. We are grateful to Loris D’Antoni and Gabe Orlanski for their valuable discussions and insights. This research was supported in part by the Vilas Board of Trustees through a Vilas Professorship (VS) and a Bodosaki scholarship (SPC). HJK received support from the ICT Creative Consilience Program through the IITP grant funded by the Korea government (MSIT) (IITP-2024-2020-0-01819).

References

1. Introducing stable lm zephyr 3b: A new addition to stable lm, bringing powerful llm assistants to edge devices. https://stability.ai/news/stablelm-zephyr-3b-stability-llm
2. Llm - detect ai generated text. https://www.kaggle.com/competitions/llm-detect-ai-generated-text
3. Llm hallucination index. https://github.com/rungalileo/hallucination-index
4. Phi-2: The surprising power of small language models. https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/
5. Zig et puce. https://www.coolfrenchcomics.com/zigpuce.htm
6. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. In: International Conference on Learning Representations (2022), https://www.microsoft.com/en-us/research/publication/beit-bert-pre-training-of-image-transformers/
7. BehnamGhader, P., Adlakha, V., Mosbach, M., Bahdanau, D., Chapados, N., Reddy, S.: Llm2vec: Large language models are secretly powerful text encoders. arXiv preprint arXiv:2404.05961 (2024)
8. Bradley, T.D., Terilla, J., Vlassopoulos, Y.: An enriched category theory of language: from syntax to semantics. La Matematica **1**(2) (2022)
9. Chen, C.Y., Grauman, K.: Inferring analogous attributes. IEEE Conference on Computer Vision and Pattern Recognition (2014)
10. Cheng, Z.Q., Wu, X., Huang, S., Li, J.X., Hauptmann, A., Peng, Q.: Learning to transfer: Generalizable attribute learning with multitask neural model search. Proceedings of the 26th ACM international conference on Multimedia (2018)
11. Chytas, S.P., Lokhande, V.S., Singh, V.: Pooling image datasets with multiple covariate shift and imbalance. In: International Conference on Learning Representations (2024), https://openreview.net/forum?id=2Mo7v69otj
12. Cruttwell, G.S.H., Gavranović, B., Ghani, N., Wilson, P., Zanasi, F.: Categorical foundations of gradient-based learning (2021), https://arxiv.org/abs/2103.01931
13. Cui, Y., Niekum, S., Gupta, A., Kumar, V., Rajeswaran, A.: Can foundation models perform zero-shot task specification for robot manipulation? In: Proceedings of The 4th Annual Learning for Dynamics and Control Conference. vol. 168. PMLR (2022), https://proceedings.mlr.press/v168/cui22a.html
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (2009). https://doi.org/10.1109/CVPR.2009.5206848
15. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). Association for Computational Linguistics (2019), https://aclanthology.org/N19-1423
16. Drozdov, A., Scharli, N., Akyuurek, E., Scales, N., Song, X., Chen, X., Bousquet, O., Zhou, D.: Compositional semantic parsing with large language models. arXiv preprint arXiv:2209.15003 (2022)
17. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision (2010)

18. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.A.: Describing objects by their attributes. IEEE Conference on Computer Vision and Pattern Recognition (2009)
19. Fong, B., Spivak, D., Tuyéras, R.: Backprop as functor: A compositional perspective on supervised learning. In: 2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (2019). https://doi.org/10.1109/LICS.2019.8785665
20. Fong, B., Spivak, D.I.: Seven sketches in compositionality: An invitation to applied category theory (2018), https://arxiv.org/abs/1803.05316
21. Freedman, D., Pisani, R., Purves, R.: Statistics (international student edition). Pisani, R. Purves, 4th edn. WW Norton & Company, New York (2007)
22. Furrer, D.P., van Zee, M., Scales, N., Schärli, N.: Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. arXiv e-prints p. arXiv:2007.08970 (2020)
23. Gavranović , B.: Learning functors using gradient descent. Electronic Proceedings in Theoretical Computer Science (sep 2020), https://doi.org/10.4204%2Feptcs.323.15
24. Gavranović, B.: Compositional deep learning (2019), https://arxiv.org/abs/1907.08292
25. Hand, E.M., Chellappa, R.: Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In: AAAI Conference on Artificial Intelligence (2017)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016). https://doi.org/10.1109/CVPR.2016.90
27. He, P., Gao, J., Chen, W.: Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. arXiv preprint arXiv:2111.09543 (2021)
28. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. IEEE Conference on Computer Vision and Pattern Recognition (2019)
29. Isola, P., Lim, J.J., Adelson, E.H.: Discovering states and transformations in image collections. In: IEEE Conference on Computer Vision and Pattern Recognition (2015)
30. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b. arXiv preprint arXiv:2310.06825 (2023)
31. Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., Momchev, N., Sinopalnikov, D., Stafiniak, L., Tihon, T., Tsarkov, D., Wang, X., van Zee, M., Bousquet, O.: Measuring compositional generalization: A comprehensive method on realistic data. In: International Conference on Learning Representations (2020)
32. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners (2022), https://arxiv.org/abs/2205.11916
33. Krizhevsky, A.: Learning multiple layers of features from tiny images. University of Toronto (2012)
34. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
35. Lawvere, F.W., Schanuel, S.H.: Conceptual mathematics: a first introduction to categories. Cambridge University Press (2009)
36. Lee, C., Roy, R., Xu, M., Raiman, J., Shoeybi, M., Catanzaro, B., Ping, W.: Nv-embed: Improved techniques for training llms as generalist embedding models. arXiv preprint arXiv:2405.17428 (2024)
37. Li, J., Xu, Y., Lv, T., Cui, L., Zhang, C., Wei, F.: Dit: Self-supervised pre-training for document image transformer. In: ACM Multimedia 2022 (2022), https:

//www.microsoft.com/en-us/research/publication/dit-self-supervised-pre-training-for-document-image-transformer/

38. Li, Y.L., Xu, Y., Xu, X., Mao, X., Lu, C.: Learning single/multi-attribute of object with symmetry and group. IEEE Transactions on Pattern Analysis and Machine Intelligence (12) (2022). https://doi.org/10.1109/TPAMI.2021.3119406
39. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
40. Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T., Feris, R.: Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In: IEEE Conference on Computer Vision and Pattern Recognition (2017). https://doi.org/10.1109/CVPR.2017.126
41. MacLane, S.: Categories for the working mathematician. Springer, New York, NY (2014)
42. Mancini, M., Naeem, M.F., Xian, Y., Akata, Z.: Open world compositional zero-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2021). https://doi.org/10.1109/CVPR46437.2021.00518
43. Marquis, J.P.: Category Theory. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, Fall 2021 edn. (2021)
44. Misra, I., Gupta, A.K., Hebert, M.: From red wine to red tomato: Composition with context. IEEE Conference on Computer Vision and Pattern Recognition (2017)
45. Moschella, L., Maiorca, V., Fumero, M., Norelli, A., Locatello, F., Rodolà, E.: Relative representations enable zero-shot latent space communication. In: International Conference on Learning Representations (2023), https://openreview.net/forum?id=SrC-nwieGJ
46. Naeem, M.F., Xian, Y., Tombari, F., Akata, Z.: Learning graph embeddings for compositional zero-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (June 2021)
47. Nayak, N.V., Yu, P., Bach, S.: Learning to compose soft prompts for compositional zero-shot learning. In: International Conference on Learning Representations (2023), https://openreview.net/forum?id=S8-A2FXnIh
48. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Conference on Empirical Methods in Natural Language Processing (2014)
49. Plato: The Republic (1994), http://classics.mit.edu/Plato/republic.html
50. Purushwalkam, S., Nickel, M., Gupta, A.K., Ranzato, M.: Task-driven modular networks for zero-shot compositional learning. IEEE International Conference on Computer Vision (ICCV) (2019)
51. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139. PMLR (2021), https://proceedings.mlr.press/v139/radford21a.html
52. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019), https://api.semanticscholar.org/CorpusID:160025533
53. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
54. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)

55. Saini, N., Pham, K., Shrivastava, A.: Disentangling visual embeddings for attributes and objects. In: IEEE Conference on Computer Vision and Pattern Recognition (2022). https://doi.org/10.1109/CVPR52688.2022.01329
56. Shiebler, D., Gavranović, B., Wilson, P.: Category theory in machine learning (2021), https://arxiv.org/abs/2106.07032
57. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
58. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: IEEE Conference on Computer Vision and Pattern Recognition (2010). https://doi.org/10.1109/CVPR.2010.5539970
59. Ye-Bin, M., Kim, J., Kim, H., Son, K., Oh, T.H.: Textmania: Enriching visual feature by text-driven manifold augmentation. In: IEEE International Conference on Computer Vision (2023)
60. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2014). https://doi.org/10.1109/CVPR.2014.32